Ad Hoc Networks 8 (2010) 640–653

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Probabilistic proactive routing with active route trace-back for MANETs

Keyvan Kashkouli Nejad *, Shawish Ahmed, Xiohong Jiang, Susumu Horiguchi

Department of Graduate School of Information Sciences, Tohoku University, Aobayama 6-3-09, Sendai 980-8579, Japan

ARTICLE INFO

Article history:
Received 27 August 2009
Received in revised form 11 January 2010
Accepted 15 January 2010
Available online 22 January 2010

Keywords:
MANET
Routing
Probabilistic protocol
Proactive
Scalable

ABSTRACT

Mobile Ad Hoc network (MANET) is a very attractive networking technology for providing flexible communication in an anytime and anywhere fashion. However, MANET is infrastructure-less and highly dynamic, so the design of an efficient routing strategy for reliable end-to-end communication in such a network has been a challenging issue. The available routing protocols and their embedded information retrieval methods for MAMETs work well only for networks with certain limited assumptions of number of nodes, density of nodes and mobility. With the continuous expansion of the MANETs in real applications, it is now imperative to develop a new routing protocol for MAMETs that is more scalable and topology independent. In this paper, we introduce such a routing scheme for MANETs, which works well under a wide range of network topologies, nodes-density, coverage area size and nodes-mobility. The proposed scheme is based on a novel enhancement of the hint-based probabilistic protocol. Instead of broadcasting extensive control packets for network topology information retrieval as that of conventional routing schemes, the proposed scheme carefully reuses the feedback information carried in unicast packets for this purpose without introducing any extra overhead. The efficiency of the proposed scheme is demonstrated through both mathematical analysis and an extensive simulations study.

1. Introduction

Mobile Ad Hoc networks (MANETs) are very attractive for providing flexible communication support in some extreme scenarios like disaster areas or battle fields, where no predetermined network structure is known and today's Internet-based communication paradigms are no longer applicable [1]. Among main factors a MANET can be characterized by are nodes mobility and network size.[1] The nodes mobility describes the average move velocity of network's nodes, while the network size is usually defined in terms of average number of nodes in the network coverage area, or equally the node-density and network coverage area size.

In MANETs, retrieval and maintenance of network topology information are usually performed through information broadcast. Freshness and efficiency of such broadcast-based information is actually very sensitive to nodes-density and nodes mobility, because high mobility may cause fast and frequent changes in network topology, and sudden increase in node-density can cause a dramatic growth in topology information broadcast but the decrease of node-density or expansion of coverage area size will cause information gathering problem for faraway nodes.[2] Therefore, the retrieval and maintenance of the current topology information and corresponding routing issues in infrastructure-less MANETs are very challenging [2].

There are different classifications of routing protocols in MANETs. But the routing protocols and their embedded information retrieval can be roughly reclassified into

* Corresponding author. Tel.: +81 22 795 7181.
E-mail address: keyvan@ecei.tohoku.ac.jp (K.K. Nejad).

[1] As this paper contributes to routing and topology information retrieval, we focus on finding the locations of nodes, and not issues such as congestion and traffic pattern.

[2] Due to power and health constraints, signal strength and thus its range cannot be too large.

1570-8705/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2010.01.002

deterministic ones and probabilistic ones [1,3], as there has been a recent trend towards probabilistic protocols for MANETs. Deterministic protocols require instant network topology information to construct and maintain the routes [4–6]. Due to the dynamic changes of network topology in MANETs, however, such a protocol must constantly acquire new information to exactly reflect the network topology up to date, which will trigger a high overhead (especially in presence of high mobility and large number of nodes) and significantly limit its practicability [7–12]. Therefore, the available deterministic protocols can carry out the routing process only under a group of restricted constraints on the network coverage area size, nodes-density or mobility.

Probabilistic protocols, on the other hand, approximately predict the network topology using only probabilistic and traceable information [13–18]. Therefore, they can significantly reduce the amount of overhead in comparison with the deterministic protocols and thus are more scalable and topology independent.

Recently, Beraldi et al. [19,20] proposed a novel Hint-Based Probabilistic (HBP) routing protocol for Ad-Hoc networks. The main idea of HBP is to use meta-information in form of hints to direct the packet probabilistically toward its destination. This protocol belongs to the category of gossip based routing protocols, where nodes tend to gossip about the possible location of a destination and intermediate nodes try to direct the packet toward its destination using gossips they have heard. The hint-based probabilistic protocol can achieve a good network performance in terms of latency and packet delivery probability even under high node-mobility, so it has a good mobility resiliency. However, the main problem of this protocol is that it does not work well under sparse networks deployed in a large area, because no sufficient information is provided for faraway nodes.

In this paper, we propose an enhanced hint-based probabilistic routing protocol by introducing a new information retrieval method in it, such that the new protocol can work efficiently under different network characteristics: nodes-density and coverage area size as well as nodes-mobility. The main idea of the new information retrieval method is to reuse the already allocated feedback information carried in unicast packets to retrieve extra information for the routing process instead of broadcasting extensive control packets. Although we choose HBP routing protocol to guarantee nodes-mobility resiliency and low latency, but the basic idea of this new topology information retrieval method can be applied to other probabilistic gossip-based routing protocols. Mathematical and extensive simulation-based analysis has been conducted to verify the efficiency of the enhanced routing scheme in coping with different network sizes and topologies as well as nodes-mobility. Each experiment was inspected in terms of packet delivery probability, route length, and latency.

The rest of this paper is organized as follows. Section 2 introduces the related works. In Section 3, we introduce the proposed routing protocol. In Section 4 an analytical model for studying the packet hint-distance correlation is presented. Section 5 presents the experimental simulation analysis of our proposed routing protocol. Finally, we conclude this paper in Section 6.

2. Related works

In general, routing protocols in MANETs can be classified into deterministic and probabilistic protocols. Here, we present an overview of these protocols.

For routes discovery and recovery processes, the deterministic protocols are designed to keep instant information about the network topology [4–6]. However, due to the random and continuous mobility of the network's nodes, such a protocol must periodically perform inspection process using control packets, which presses a continuous overhead on the network bandwidth [7–9]. We can easily notice that any sudden increase in the nodes-mobility can cause more frequent need of network topology inspection and thus an extensive amount of overhead [10–12]. This problem becomes even worse with the increase of the network size (i.e., nodes-density and network coverage area size), since a huge amount of information will need to be constantly and frequently inspected. Therefore, the available deterministic protocols can carry out the routing process only under some restricted constraints on the network coverage area size, nodes-density and mobility. Some deterministic approaches try to enhance the existing deterministic protocols to get more scalable algorithms. For example, SMORT introduced in [4] is a scalable deterministic routing algorithm. It exploits secondary paths to recover broken paths, and thus reduces the overhead produced in route recovery procedure of AODV protocol. In presence of only a few sessions, this protocol provides good scalability for different sizes of the network by adopting fail-safe multiple paths. However, the route discovery there involves vast network flooding, so as the number of concurrent sessions increase, it will trigger more discovery attempts and thus tends to extensively increase the overhead.

Instead of frequently acquiring new information for each time the network topology changes, the Probabilistic protocols try to statistically predict these changes based on pre-collected information [13–18]. Therefore, the frequency of updating and also the amount of control information for each update is reduced in comparison with the deterministic protocols.

In general, the route discovery in the probabilistic protocols can be further classified into reactive and proactive approaches. In the probabilistic reactive approach, the route discovery or recovery is initiated on request via a wide flooding of control packets. In a large network area, finding the route to a far away destination without any pre-collected information about its previous location is very time consuming. Moreover, due to the extensive flooding, which will be initiated as result of the continuous route breakage under high nodes-mobility condition, the limited bandwidth of the Ad Hoc network is aggressively consumed. Again, we can easily notice that this class of routing schemes can perform the routing process only under a group of limited assumptions on network coverage area size and nodes-mobility. Currently, the routing protocols in this subclass are mainly based ant or agent techniques.

The ant-based probabilistic routing in MANETs has been widely discussed in [13,16,14]. One example of ant-based routing, Ant-colony Based Routing Algorithm (ARA), was introduced by Gunes et al. in [14]. In this algorithm, when a connection request arrives at any node, a forward ant (FANT) with a unique sequence number is launched from the source node to search for paths to the destination node. Once a FANT reaches the destination, its information is extracted and a backward ant (BANT) is launched to search for a backward path to the source node. At the arrival of the BANT at the source node, the path is established and data packets are sent. This approach suffers from high setup delay and also a high overhead, especially in presence of high mobility of nodes. Various probabilistic routing algorithms using agents in the routing process of MANETs have also been introduced, see, for example [21–23,18]. In [18], Roth et al. proposed a probabilistic routing algorithm by exploiting swarm intelligence, called Termite. In this algorithm, a network node is equipped with probabilistic pheromone table that contains the selection probability of neighbor node when a packet is moving toward its destination node. As a packet is dispatched from the source to destination, it follows the pheromone trail for its destination through the network while leaving pheromone for its source. Route requests, on the other hand, perform a random walk over the network until a node is found which contains some pheromone for the requested destination. It is notable that parameter tuning in agent-based and ant-based routings is not a trivial task.

The probabilistic proactive approach, on the other hand, continuously keeps track of the network topology information for routing and therefore it does not suffer from the high route setup latency as the reactive approach. However, as the network topology information is continuously updated to cover all possible routes, the amount of control information aggressively consumes the network resources with the increase of nodes-density and network coverage area size, even if none of those routes are actually used by the ongoing traffic [24]. Again, we can easily notice that this kind of routing schemes is constrained to some strict assumptions [19,20]. In the following, we introduce some examples of this subclass of protocols which is most related to our work in this paper.

Beraldi et al. presented some preliminary idea of the hint-based routing for Ad-Hoc networks in [24], exploiting the duration of time passed since the last time nodes encountered with the destination, namely the encounter age. When forwarding a packet, a node chooses an approximate next-hop node from the set of nodes at distance less than or equal to two hops from itself according to their encounter ages. The approximate information is gathered only in a vicinity of two hops away, so as the area of the network increases, the forwarding policy is likely to fail when the source is very far from the destination. This problem is enhanced in HBP [19,20] to some extent, but we will show that HBP routing protocol still suffers from this problem, which we will try to overcome using the proposed scheme.

3. Probabilistic proactive routing with active route trace-back

The enhanced hint-based probabilistic routing protocol and the new information retrieval method for mobile Ad-Hoc networks are presented in this section.

As we explained in Section 1, we base our scheme on top of HBP scheme due to its good properties in terms of mobility resiliency and low latency. In HBP routing protocol, each node i has a hint table that contains hints towards any possible destination. These hints are originated (produced) by other nodes, not farther than a specific hop distance (which is called LookAhead or L) from i. On the other hand, node i is in charge of computing the hint h_{id} for any possible destination d, and dispersing hints into its locality (up to L hops away). Node i uses the hints in its hint table when selecting the next hop node among its neighbor nodes for forwarding the packets.

We use hints produced the same way as the HBP protocol (time vector hints) using a small value of LookAhead, and therefore, reduce the amount of control packets. The main difference between the proposed scheme and HBP is that additional hints according to the feedback information of the unicast packets are produced. These additional hints would be compatible and comparable in value with the original time vector hints, and therefore, can be stored in the same hint table with them. The header of each unicast packet in HBP has a vector V of visited nodes. According to this information, additional hints regarding the active route in which this packet is traversing through will be calculated. Using this information retrieval method we can get information about destinations far away without introducing any new overhead. We now describe the algorithm in more details.

3.1. Hint table structure

In our algorithm a node i with k neighbors has a hint table, HT^i, that has $N-1$ (N is the number of nodes in the network) rows and $k+1$ (one column for its own hint towards each destination) columns.

In the hint table, each row corresponds to a destination node and each column corresponds to a neighbor node, with one additional column reserved for the hints a node calculates itself to be broadcasted to its neighbor nodes (own hints). Each cell in this table, let us say of row d and column n contains multiple of tuples with form of (h, hop, g), where h is the hint towards destination d received through neighbor node n, generated hop hops away by node g. A hint is a value that represents the probable distance between the node that generated the hint (g) and a destination (d). The smaller the hint the closer the generating node might be to the destination, so, after receiving a packet traveling towards that destination, a node might want to forward the packet towards the generating node of the smallest hint it has in its hint table. This should be done through the neighbor node between the current node and the generating node, the neighbor through which the current node received the hint. A simple network topology at time t is illustrated in Fig. 1. Table 1

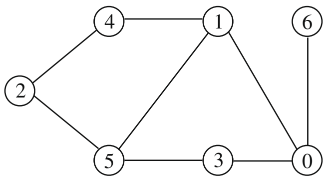


Fig. 1. A simple network topology at time t.

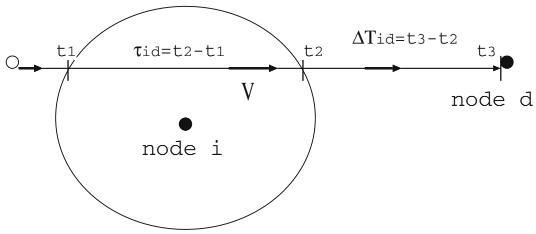


Fig. 2. Illustration of time vector information.

Table 1
Hint table of node 2 with $L = 2$.

Destinations	Own Hint	4	5
0	0.9, 0, 2	0.8, 1, 4 0.0, 2, 1	0.6, 1, 5 0.0, 2, 3 0.0, 2, 1
1	0.3, 0, 2	0.0, 1, 4	0.0, 1, 5 0.6, 2, 3
...	...	...	...
6	0.4, 0, 2	0.7, 1, 4 0.5, 2, 1	0.7, 1, 5 0.5, 2, 1 0.2, 2, 3

shows the corresponding hint table with $L = 2$ of node 2 at a specific time t of this network with entries in form of (h, hop, g).

3.2. Hint computation

The main difference between the proposed scheme and the original HBP protocol lies in the way hints are produced. In our algorithm we calculate hints in two ways, one is the same method used in HBP exploiting the time vector of nodes contact, and the other is the proposed method using the feedback information of data packets, we will call the former Time Vector Hints and the latter Packet Hints later on in this paper. Both types of hints are compatible with each other, and apart from computation, their process of storing and inquiring need not be differentiated.

3.2.1. Time vector hint computation

In HBP in each node i the encounter time t_{start} and departing time t_{break} with any other node d is stored in a time vector. Using the time vector values the hint towards a node d is computed according to the following procedure: hint $h_{id}(t)$ computed at time t is zero if at this time i and d are neighbors, while it is ∞ if they never came in contact before t. If, however, they were 1-hop neighbors in the past and their last contact was lost at time t^*, then the hint is given by

$$H_{id} = \frac{t - t^*}{\tau_{id}} = \frac{\Delta T_{id}}{\tau_{id}}, \quad (1)$$

where ΔT_{id} is the period of time passed since d was i's neighbor last time, and τ_{id} is the length of time d was within i's transmission range (see Fig. 2). The hint values could be computed without using the value of τ_{id} as well. But the computed hints are to be sent in control packets to other nodes, and thus, they should be fit in few bits. For doing so we need to normalize them in a specific range. This could be done by dividing ΔT_{id} by a constant value but this constant value will not be efficient for networks with different nodes' speeds (e.g. networks with low nodes' speed would need different value to networks with high nodes' speeds). As ΔT_{id} and τ_{id} are both influenced by the nodes' corresponding speed, dividing by τ_{id} can eliminate the effect of different speeds (e.g. high or low speeds) in the process of storing hint values.

Hints are dispersed by broadcasting control messages within beacon packets. Each node i broadcasts a heartbeat packet every $\Delta T_B s$ [19]. This packet encapsulates hints generated by node i itself and nodes located at distance at most $L - 1$ hops from itself for all destinations (the value L is called the look-ahead of the protocol). Therefore, a node receiving the control packet will update its hint table by hints generated at most L hops away from itself.

In this method the gathered information (hint) looses its validity sooner when we move farther from the destination the smaller the L, and the amount of overhead is proportional to the value of L. In environments that collecting information about a node far away is difficult (e.g. sparse networks or networks with large network area), to guarantee that for any possible destination enough routing information is collected inside hint table, we have to use big values of L, in which case the overhead increases extensively (e.g. the amount of overhead using value of $L = 4$ will be three times of that amount when value $L = 2$ [19]), while for dense networks with large number of nodes, we have to use small values of L. This is due to the fact that control packet size is also directly related to the number of nodes and average number of neighbors [19], and so in a network with large network area and high density of nodes very small values of L should be used not to suffer the bandwidth extensively. But as we explained we need a more general algorithm applicable for different network topologies.

3.2.2. Packet hint calculation

The main drawback in HBP is that the topology information which is broadcasted in expensive control packets, is gathered for all possible destinations simultaneously. We will solve this problem by gathering additional information (Packet Hints) regarding only the active destinations (active routes), and doing so we can avoid large values of L. Moreover, gathering and dispersing of this information is not done through broadcasting control packets, but rather by reusing the feedback information already available in the header of unicast packets.

In HBP every unicast packet p is equipped with a vector V of the already visited nodes to avoid loops in the routing process. This information can reveal the probable distance of the last visited node from the node originating the packet (although in some protocols such list is not available, usually in the header of the unicast packets at least the originator ID, the sender ID, the destination ID, and number of hops the packet traversed are mentioned), which would be especially useful when the destination has a packet to send back to destination. As the HBP does not support any acknowledgment back to the originating node, the packet coming back could be the acknowledgment packet sent back in a TCP-like connection, or uplink flow packets in case of voice or any other kind of a two-way connection.

Let us assume that node w received a packet p from neighbor node n, and for this packet p we have $V = \{v_0, v_1, \ldots, v_k\}$ where $v_0 = org$ and $v_k = n$, and n is a neighbor node of w. It is very probable that the more hops the packet traversed to reach neighbor n the farther such node is from the originating node, for example with a high probability v_1 is closer to org than v_4, mainly due to the fact that at each hop we try to forward the packet as close as possible to its destination using hints. Now we can calculate new hints called Packet Hints according to the hop distance of a node $v_l (1 \leqslant l \leqslant k)$ from org. But these new hints should be compatible with time vector hints to be able to be put into use in the hint table.

As we explained each hint towards a destination d comes in a tuple $\langle h, g, hop \rangle$ and is received through a neighbor node *neigh*, so we have to define each of these values for the Packet Hints we are going to produce, to be able to be put in the hint table. Let us say we want to calculate the hint that node v_l generates towards org. The generating node is v_l, the neighbor would be n, and hop would be the distance of v_l from w, which would be $k + 1 - l$, d would be org (because we are calculating the hint towards org). The only problem here is how to calculate Packet Hint (h) itself such that it would be comparable in value with the time vector hints.

The time vector hints are calculated according to Eq. (1), and its values are illustrated in Fig. 2 We try to rewrite this equation and values in another way. In [25] it was shown that the joint mobility problem can be transformed into an equivalent problem involving the movement of a single node by fixing the frame reference of one node to the other. For each movement of this node, the other node is translated an equal distance in the opposite direction. Therefore, if we assume i is stable and d moves towards it and passes through its transmission region (a circle with radius R around d, where R is the transmission range) with average velocity V, we have

$$H_{id} = \frac{\Delta T_{id}}{\tau_{id}} = \frac{D_{id}/V}{d_{id}/V} = \frac{D_{id}}{d_{id}}, \tag{2}$$

where D_{id} is the probable distance of i from d's transmission region, and d_{id} is the probable distance node d traversed in node i's neighborhood (see Fig. 3).

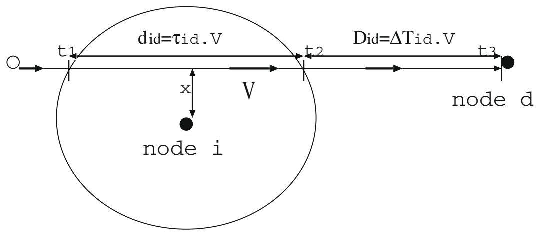


Fig. 3. Distances traversed in nodes encounter and separation.

So if we put $i = v_l(d = org)$, all we have to do now is to compute the D_{id} and d_{id} according to the packet's traversed distance in hops.

For a packet at each hop the probable distance from the originating node is increased in average by amount ΔD, which is the average distance of a node from its neighbor. If we assume the position of the node as the center and its neighbor node as $\{x, y\}$. We can calculate ΔD by

$$\Delta D = \frac{\int\int_{\sqrt{x^2+y^2}<R} \sqrt{x^2 + y^2} \cdot d_x d_y}{\int\int_{\sqrt{x^2+y^2}<R} d_x d_y} = (2/3) \cdot R. \tag{3}$$

So the probable distance of the node v_l from the signal region of the originating node org is

$$D_{id} = (l - 1) \cdot \Delta D = (2/3) \cdot (l - 1) \cdot R. \tag{4}$$

Please note we use $l - 1$ instead of l, this is because the first hop the packet traversed was inside the transmission range of the originating node.

Instead of d_{id}, we normalize this amount by the average value $\overline{d_{id}}$, which is the average distance a node d can traverse in a node i's transmission range before they loose contact. Without loosing generality we can assume node d passes through node i's transmission region horizontally. We can get

$$\overline{d_{id}} = \frac{\int_{-R}^{R} 2\sqrt{R^2 - x^2} \cdot dx}{\int_{-R}^{R} dx} = \frac{\pi R}{2}, \tag{5}$$

where x is the distance of the node d from node i in the closest point, when passing through node i's transmission region (Fig. 3).

Now that we computed D_{id} and d_{id} we can calculate the new correspondent Packet Hint as follows:

$$PH_{v_l d} = \frac{4 \cdot (l - 1)}{3\pi}. \tag{6}$$

The packet hints can be calculated correspondent to different generating nodes up to L hops away from node i in the same way as time vector hints. For example if $L = 2$ then we will have

$$PH^1 = \frac{4 \cdot (k - 1)}{3\pi}, \tag{7}$$

$$PH^2 = \frac{4 \cdot (k - 2)}{3\pi}, \tag{8}$$

where PH^j correspondents to the hint generated by node v_{k-j+1}, j hops away from the current node (w). All these hints are assumed to be received by node w. If in the hint table of node w a hint for the same destination and hint originator existed, then the one with the smaller value will replace the other.

3.2.3. Packet forwarding

As explained above, we assume that each data packet includes a list of visited nodes V. Upon the arrival of a packet p destined to d in a node i at time t, it determines an ordered list of possible next hop nodes (list of its neighbors excluding nodes listed in V). The order is determined by the value of hints in the Hint Table. Each time i tries to forward the packet to the node at the head of the list, and selects the next node in the list if the forwarding process was not successful. This will be repeated until i received the acknowledgement for packet p from a neighbor node n, or the list is empty, and in either case the packet will be dropped. Like this, in our algorithm we provide for multiple paths [4] at all intermediate nodes which is shown to improve the performance of multipath routing protocols in [26], without any need for route recovery process.

3.3. Discussion

Here we discuss some main issues we might encounter while applying this new information retrieval method.

In the algorithm description we assumed unicast packets are equipped with a vector V of the already visited nodes to avoid loops in the routing process. Although this is true for protocols like HBP, some protocols main not provide such list. However, usually in the header of the unicast packets at least the originator ID, the sender ID, the destination ID, and number of hops the packet traversed are mentioned (even in some deterministic protocols). In such case the new packet–hint values can be calculated, but the only problem is that the hint generator node will not be known. This problem can be solved with a slight modification in the algorithm, and packet–hints can be calculated and stored to be used in process of routing. In case even such values as number of hops, sender ID and origination ID are not mentioned in the unicast packet, the proposed method cannot be applied.

One other issue is that some sort of two-way communication was considered (either some kind of ACK or data packets going back to origination node). In case there are no packets going back to origination node from destination, the newly computed hints might be useless. This is the main drawback of this method. However as the computation process for this new information is very simple, and they add no extra overhead to the protocol, this will only cause a slight redundant computation. Also if any other communication will regard the origination node of the communication discussed above as its destination, again the computed packet–hints can be put into use.

Lastly for computing new packet–hints, some constant average velocity was assumed in order to make the new hints compatible with the original time–vector–hints. This assumption of constant average speed vector, which has also been applied in the original HBP routing protocol, can not hold true for long distances. As a matter of fact, when packets traverse longer distances hints computed by such assumptions start to lose their validity, but this case happens for the original hints as well. On the other hand, we will try to show that the validity of packet–hints is more stable for long distances compared to original time–vector–hints.

4. Packet–hint distance correlation

In gossip-based protocols (e.g. HBP) nodes tend to gossip about the possible location of a destination. Here we show mathematically that exploiting the active route trace-back (packet–hints) above some gossip-based algorithm can improve the performance, especially in a network deployed in a large area. We assume that a packet is directed towards its faraway destination based on gossips and on its way the intermediate nodes store the trace-back information (packet–hints). We also consider the fact that gossips tend to lose their accuracy the farther we move from a destination.

The process of our analysis in this section is as follows: first we show that the newly collected information can reflect the locations of nodes in the network area. For doing so we will derive the conditional expected distance of the node holding the packet and the origination node, assuming that the packet reached this node after k hops using gossips. If there is a logical relation between the expected distance and number of hops, then the number of hops can be used as way to show the distance of node from an origination node, which would be helpful in case the origination node becomes a destination of a packet (e.g. two-way communications). Second, we show that the newly gathered information can improve the probability of directing of the packet towards nodes closer to the destination at each iteration of the algorithm. In the gossip-based protocol (e.g. HBP), at each node the gossips (hints) about a destination were collected from nodes whose hop distance are not more than L hops away. In the process of forwarding the packet, the node tries to direct the packet towards a node with a best gossip (hint) about the destination, i.e. a node which could be the nearest to the destination. We will show mathematically that the probability of choosing a node nearer to destination, when choosing between two nodes to direct the packet to, will increase using packet–hints in parallel to original gossips (considering that the original gossips lose their accuracy the farther we move from a destination). In other words, the probability of choosing a better node in process of forwarding the packet increases.

This section describes a discrete-hop–distance discrete-space analytical model for studying the packet–hint distance relationship. The model is inspired by the so called Manhattan-like topology, which is used to represent a city with major streets running east–west and north–south. In this topology, the user can move along either the horizontal or the vertical direction.

4.1. Model

We capture the network topology at a specific period of time, during which a packet travels from its originating node towards its destination, with a 2-D torus with $2H \times 2H$ points (Fig. 4). At each point of the torus a node

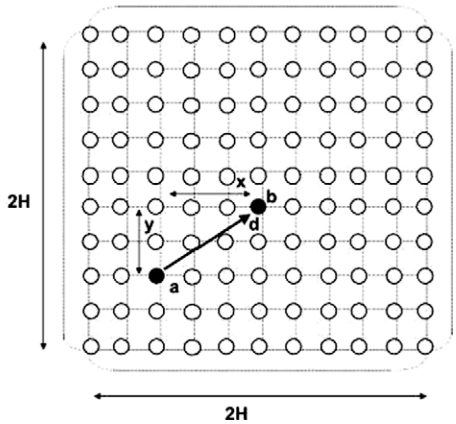


Fig. 4. A $2H \times 2H$ torus.

is located. As the period of a packet moving is a matter of milliseconds and the period of time necessary for a node movement is a matter of seconds, we can assume that the topology does not change in the period of time we inspect the network.

The relative position of a node with the originating node is expressed by a vector of random variables, (x, y), where x and $y \in [-H, \ldots, 0, \ldots, H]$ (recall that the topology is wrapped), and so node (x, y) denotes the node located in position (x, y). The transmission range is R (here we assume $R = 2$) and a node (k, l) is a member of neighbors of node $(i,j), N(i,j)$, if their Euclidean distance is less than or equal to R (here we do not consider the shadowing effect). A packet can move discretely in this torus. At time tick i a node (x_i, y_i) holding the packet decides to forward the packet to a node (x_{i+1}, y_{i+1}).

On the other hand the distance between two nodes (i,j) and (k, l) is the Manhattan distance metric $|i-k| + |j-l|$, which represents the distance for node (i,j) to traverse to reach to the vicinity of node (k, l). We will denote with $dist(i,j)$ the distance of the node (i,j) from a the packet origination node located at $(0, 0)$. The destination however, is located at $(dest_x, dest_y)$, and with $d(i,j)$ we denote the distance of a node from destination.

The packet is originated at node (x_0, y_0) (0th hop), and at ith hop, the node holding the packet (x_i, y_i), decides to forward the packet to one of the nodes in $N(x_i, y_i)$. The decision is based on some pre-calculated information called gossips (at the first phase this information only includes time–vector–hints introduced in [19], because there are no packet–hints calculated yet). Let the distance between the node holding the packet and the origination node at a random hop be the r.v. M. According to this information similar to the analysis made in [20] we can suppose that at each hop the probability that a node holding the packet forwards it to any of its neighbors nearer to the destination than itself is P_F, and the probability that it forwards the packet any of the other neighbors is $P_B < P_F$, and $Acc = P_F - P_B$ $(P_F + P_B = 1)$. It can be realized from analysis performed in [19] that the farther a node is from the destination, and the smaller the look-ahead, the hints reliability and as a result the gap between P_F and P_B, Acc, will be smaller and this decline resembles a negative exponential function. In the look-ahead zone of a destination, i.e. nodes up to L hops away from it, the original hints can make sure that the packet is forwarded to a node nearer to the destination in other words $Acc = 1$. For farther nodes such value drops down dramatically as with distance from the destination. Therefore, for a node say (x, y) we assume the Acc to be:

$$Acc_{x,y} = exp^{1-\frac{\sqrt{(dest_x-x)^2+(dest_y-y)^2}}{L \cdot R}}. \tag{9}$$

Let us now calculate the conditional probability of the location of the packet at the kth hop being (x, y) assuming that at hop $h = 0$ it was located in $(0, 0)$, Such a probability will be denoted as $Pl(k, x, y)$.

Consider the discrete-time Markov chain whose states correspond to locations of the packet. The initial state of the chain is (0,0), and $dest_x, dest_y$ is the absorbing state. $Pl(k, x, y)$ is the probability that the state of the chain at time (hop) k is (x, y). Therefore, we have

$$Pl(k, x, y) = \sum_{(i,j) \in N(x,y)} Pl(k-1, i, j) \cdot P_{(i,j),(x,y)}, \tag{10}$$

where $P_{(i,j),(x,y)}$ is the probability of transition from state (i,j) to state (x, y). This probability is one if $(i,j) = (x, y) = (dest_x, dest_y)$ (this is the absorbing state), and zero if $(x, y) \notin N(i,j)$. For the other cases (i.e., $(x, y) \in N(i,j)$) we can write

$$P_{(i,j),(x,y)} = \begin{cases} p_{f(i,j)} & \text{if } (x, y) \in N_F(i,j), \\ p_{b(i,j)} & \text{if } (x, y) \in N_B(i,j), \end{cases} \tag{11}$$

where $N_F(i,j) = \{(v, w) \in N(i,j) | d(i,j) > d(v, w)\}$, $N_B(i,j) = \{(v, w) \in N(i,j) | d(i,j) \leqslant d(v, w)\}, p_{f(i,j)}$ is the probability of transition from the state (i,j) to one of the states in $N_F(i,j)$, and $p_{b(i,j)}$ is the probability of transition from the state (i,j) to one of the states in $N_B(i,j)$. We can write

$$p_{f(i,j)} = \frac{P_{F(i,j)}}{|N_F(i,j)|}, \tag{12}$$

$$p_{b(i,j)} = \frac{P_{B(i,j)}}{|N_B(i,j)|}. \tag{13}$$

So now we have Eq. (14).

$$Pl(k, x, y) = \sum_{(i,j) \in N_B(x,y)} Pl(k-1, i, j) \cdot p_{f(i,j)} + \sum_{(i,j) \in N_F(x,y)} Pl(k-1, i, j) \cdot p_{b(i,j)}. \tag{14}$$

4.2. Numerical results

We now give some numerical result assuming $dest_x = 30, dest_y = 30$. Fig. 5 shows the probability that, after k hops the the distance between the mobiles is l, namely $\sum_{x+y=l} Pl(k, x, y)$. The curves are calculated for $k = 10, 50, 80, 100$. For a large k such a probability suddenly increases for the value $l = 60$, which is the distance between origination node and destination node, due to the fact that there is a high probability that the packet reaches the destination. Please note that the destination point is the absorbing state in our Markov chain, and therefore, packets reaching this

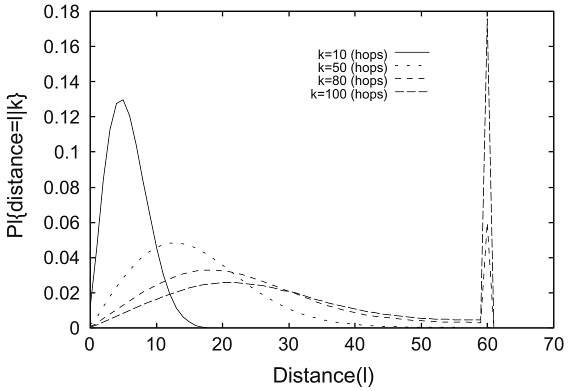


Fig. 5. Probability distribution of traversed distance for different traversed hops $k = 10, 50, 80, 100$.

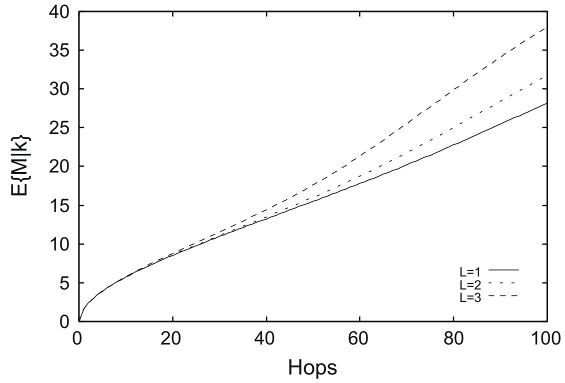


Fig. 6. Conditional expected distance vs. traversed hops.

point even before number of k hops were traversed, are assumed to stay there for the rest of hops until we calculate the probability for number of hops equal to k. Otherwise, the peaks shown would be much lower than the ones shown in the figure. Fig. 6 plots the expected distance of the node holding the packet from the origination node assuming the packet has traversed k hops, with different values of L. The expected value increases with number of hops and approaches distance between the origination and destination nodes, and with larger values of L this happens faster. For smaller values of k the expected distance.hops correlation is almost linear. The hop difference during which an appreciable linear relation between these two values exists can be taken as a measure of the lifetime of the packet–hint distance correlation.

was received by node (x_j, y_j). Now consider that these two nodes' hop distances are k_1 and $k_2 = k_1 + \delta k$. What is the probability, say Popt, that n_1 is closer than n_2 to the destination for all possible pairs of (n_1 and n_2)? This value tells us the probability that, when choosing between two nodes to direct the packet towards, the node providing the lower hop distance (lower packet–hint value) is the correct choice given that their hop distances were k_1 and k_2. We get Eq. (16).

$$Popt = \frac{\sum_{(x_1,y_1),(x_2,y_2)\in Z_L(x_j,y_j),|x_1|+|y_1|<|x_2|+|y_2|} Pl(k_1,x_1,y_1)\cdot Pl(k_1+\delta k,x_2,y_2)}{\sum_{(x_1,y_1),(x_2,y_2)\in Z_L(x_j,y_j)} Pl(k_1,x_1,y_1)\cdot Pl(k_1+\delta k,x_2,y_2)}, \tag{15}$$

where $Z_L(x_j, y_j)$ is the set of nodes of the torus than can be at most L hops away from (x_j, y_j). Proof:

$$\begin{aligned} Popt &= P(d(x_1,y_1) < d(x_2,y_2)|k(x_1,y_1) = k_1 \wedge k(x_2,y_2) = k_2 \wedge (x_1,y_1),(x_2,y_2) \in Z_L(x_j,y_j)) \\ &= \frac{P(d(x_1,y_1) < d(x_2,y_2) \wedge k(x_1,y_1) = k_1 \wedge k(x_2,y_2) = k_2|(x_1,y_1),(x_2,y_2) \in Z_L(x_j,y_j))}{P(k(x_1,y_1) = k_1, k(x_2,y_2) = k_2|(x_1,y_1),(x_2,y_2) \in Z_L(x_j,y_j))} \\ &= \frac{\sum_{(x_1,y_1),(x_2,y_2)\in Z_L(x_j,y_j)} P(d(x_1,y_1) < d(x_2,y_2) \wedge k(x_1,y_1) = k_1 \wedge k(x_2,y_2) = k_2)}{\sum_{(x_1,y_1),(x_2,y_2)\in Z_L(x_j,y_j)} P(k(x_1,y_1) = k_1 \wedge k(x_2,y_2) = k_2)}. \end{aligned} \tag{16}$$

Consider now the case that the origination sent packets to the destination, and each node (i, j) on the way registers the hop distances from the origination node in a local variable, say $k(i, j)$. Now the destination has a packet to send back to origination node, and therefore, the origination node, node(0,0), plays the role of the destination of this packet.

Consider that the packet is currently at a node (x_j, y_j) and this node wants to forward the packet towards node(0,0). Further assume that this node can decide between two nodes, say n_1 and n_2 to direct the packet towards. As the look-ahead of the algorithm is L each of these nodes (n_1 and n_2) can be a node up to L hops away who have produced a hint (packet–hint), and that hint

If we substitute the definition of Pl, we get Eq. (16).

Fig. 7 shows such a probability for the $(x_j, y_j) = (15, 15)$ and $L = 2$. The probability that n_1 is the correct choice increases with δk and as k_1 decreases. The figure also reports the probability for a random selection. (In the random selection next hop node is selected only based on the precalculated gossips or in HBP's case the time–vector–hint information.)

We have also conducted a simulation-based analysis of to show the value Popt empirically as shown in Fig. 8. The values were estimated by simulating nodes moving in a square area with edge 4000 m according to random walk with wrapping mobility model, with minimum speed 1 m/s and maximum speed 20 m/s. There results are cal-

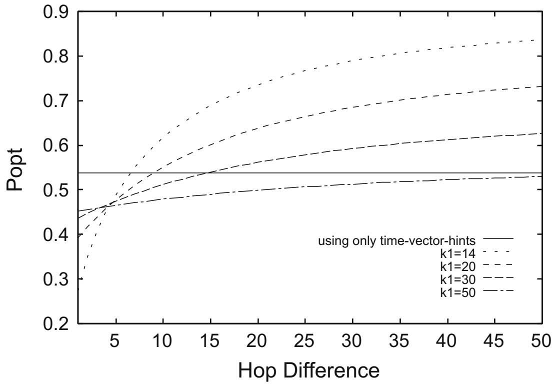


Fig. 7. Probability that n_1 is closer than n_2, given that $k_2 = k_1 + \delta k$.

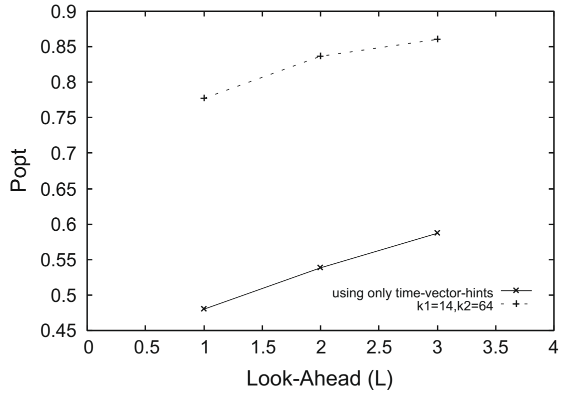


Fig. 9. Probability that n_1 is closer than n_2 as a function of L, given that $k_1 = 14$ and $k_2 = 64$.

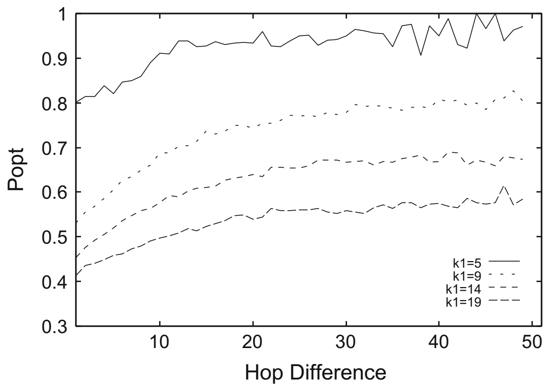


Fig. 8. Empirical results: probability that n_1 is closer than n_2, given that $k_2 = k_1 + \delta k$.

culated for one origination node sending packets every 50 m/s toward its destination. The value Popt was calculated for different values of $k_1 = 4, 9, 14, 19$ and $\delta k = 1, \ldots, 49$, for any node whose euclidean distance with the origination node is larger than $5 \times R$ and smaller than $6 \times R$. (The simulation analysis showed similar results for other distances as well.) Where R, the transmission radius is 250 m. The simulation was repeated 100 times and the duration of each simulation was 700 s, with 200 s as warm up time before the origination nodes starts sending packets. The simulation results show similar behavior as the mathematical results.

Fig. 9 shows such the probability Popt as a function of L for a given k_1 and k_2. The value of the probability of a random choice is also reported.

Although the model studied above is simple, we guess that it is able to capture some general properties of the communication in the environment and hence can be used to derive some general principles (which are confirmed by the simulation-based analysis). If $k_1(k_2)$ are the hop information of the node $n_1(n_2)$ w.r.t. a target node, then we can assume that, on the average, the probability that n_1 is closer than n_2 to the target is higher than the one associated to a random choice between the two nodes when:

- k_1 is lower than a critical value K (hop–distance correlation is still valid),
- and $k_2 - k_1 > 0$. (The packets moving towards destination have traversed fewer hops when reaching n_1 than n_2.)

Moreover, such a difference increases if: (i) k_1 decreases; (ii) $k_2 - k_1$ increase; (iii) the LookAhead L increases.

It can be noticed that although the nodes considered for calculating Popt in the mathematical results are not far from the origination node, we have considered packets that reach these nodes after several hops (e.g. 50 hops). There reason behind this is that the results are taken in the stage where nodes only use original hints for forwarding the packets, and as we have discussed such information is not sufficient. Therefore some packets will not follow the shortest paths to the destination. As a result such packets will wander in the network for several hops without getting much closer to the destinations at each hop. This fact was confirmed by the empirical results shown in Fig. 8.

5. Simulation and results analysis

In this section we verify the performance of our algorithm based on simulation of a random Ad Hoc network. We compare the performance of the proposed algorithm with the promising HBP algorithm.

5.1. Simulation model and assumptions

In this part we talk about simulation model and assumptions. Here we have used a simple simulator as our goal in this paper is solely to introduce an inexpensive information retrieval method to find a more exact location of the destination, which is the main task of routing. But if traffic patterns are or other issues not addressed here are to be considered, which is of our concern in the future works, using a more sophisticated simulator would be required.

5.1.1. Transmission primitives

Packet transmissions are governed by an ideal scheduler. A FIFO buffer of 20 packets in size is used at each node. A packet reception is notified to a sender's neighbor provided that they remained for the whole duration of the transmission within each others transmission range and such that no collisions with other transmissions occurred in the meanwhile. At the end of the transmission, the scheduler checks whenever other packets queued in the sending buffers can be served. Transmission speed is 11 Mbps. If the sender attempts to send a packet to one of its neighbors and this attempt failed, it goes into a back-off time, and tries again after the back-off time finished. It makes maximum of seven attempts, where *i*th back-off time is $50T$ (μs), and T is chosen randomly in range $[0,\ldots,2^i]$. If all attempts failed it considers the link towards that neighbor as broken and removes that neighbor from its neighbor list.

5.1.2. Mobility

We apply our mobility model within a square shaped area of $E[m]$ edge. At the beginning of the simulation nodes are located uniformly at random in the area. After that each node either decides to stop at the same point or to move, with the same probability. In case of moving, it chooses a random direction (north, east, west or south), a random speed V in range $[1,\ldots,Vmax]$ (m/s), and a random moving time MT in range $[1,\ldots,MTmax]$ (s), then it starts moving in that direction with speed V for MT seconds. If it decided to stop at the same point it chooses a halt time HT in range $[1,\ldots,HTmax]$ (s) and stays there for HT seconds. When the moving time or halt time passed it continues doing the same process.

This mobility model was inspired by random walk with wrapping mobility model discussed in [27], which is very similar to random waypoint mobility model [28]. These class of mobility models are all special cases of random trip mobility model [27].

5.1.3. Traffic

We adopt the constant bit rate traffic model widely used in performance analysis of MANETs. The constant bit rate sources always send packets of 512 bytes in length to the same destination, and the destination sends back packets to the same originating node. The number of source–destination pairs is 10% of the number of nodes in the network. The default value of simulation's main parameters are reported in Table 2.

Table 2
Default simulation parameters.

Parameter	Values
Simulation time	1500 (s)
Default number of nodes	100
Nodes' speed range	$[1,\ldots,V_{max}]$ (m/s)
Default V_{max}	20 m/s
Pause time	$[1,\ldots,HT_{max}]$ (s)
Default HT_{max}	10 (s)
Moving time	$[1,\ldots,MT_{max}]$ (s)
Default MT_{max}	100 (s)
Transmission radius, R	250 (m)
Default area edge length, E	2000 (m)
Look-ahead zone, L	2 (hops)
Message length	512 (bytes)
Transmission speed	11.0 (Mbps)
Sending buffer	20 (packets)
Update interval, ΔT_B	500 (ms)
Allowed number of missed heartbeats, M	1

5.2. Performance metrics

The following metrics were estimated during a simulation:

- Delivery probability, ratio of the number of data packets delivered to the destinations to those generated by the traffic sources.
- Average path length, given in number of hops a packet traverses until it reaches its destination.
- End to end packet delay, the time elapsed from when a packet is generated by the source until it is delivered to the destination.

Due to the fact that in our algorithm no additional control packets is introduced to the HBP algorithm discussed in [19], our results would not be including this metric.

Each experiment is conducted with an initial warm up time of 200 s before collecting statistical data.

5.3. Results analysis

In this section, we analyze the performance of our proposed routing protocol in comparison with the original HBP protocol under different simulation scenarios.

In all of the conducted simulations, the look-ahead value,[3] L, is set to 2. The reasons of such settings were explained in details in Section 3.

5.3.1. Performance versus nodes density

In this part, we investigate the effect of the nodes-density variation on the performance of the proposed scheme. In our simulations we increased the number of nodes in the network area (i.e., square shaped area of 2 Km^2), while keeping a high nodes-mobility with $V_{max} = 20$ m/s.

The results in Fig. 10a shows that the delivery probability of the proposed scheme outperforms that of HBP significantly in sparse networks scenario while both protocols report a similar performance under the dense networks condition.

In the dense network scenario, the number of neighbors increases and thus, even with a small look-ahead value, any node can easily collect enough information about the network topology using the HBP protocol. In sparse network scenario, the same task becomes harder due to the lack of neighbor nodes. On the other hand, due to the low overhead and the newly exploit trace back information, the proposed scheme shows efficient performance in both dense and sparse network scenarios without introducing any extra overhead.

[3] The maximum number of hops a hint is forwarded from its originating node.

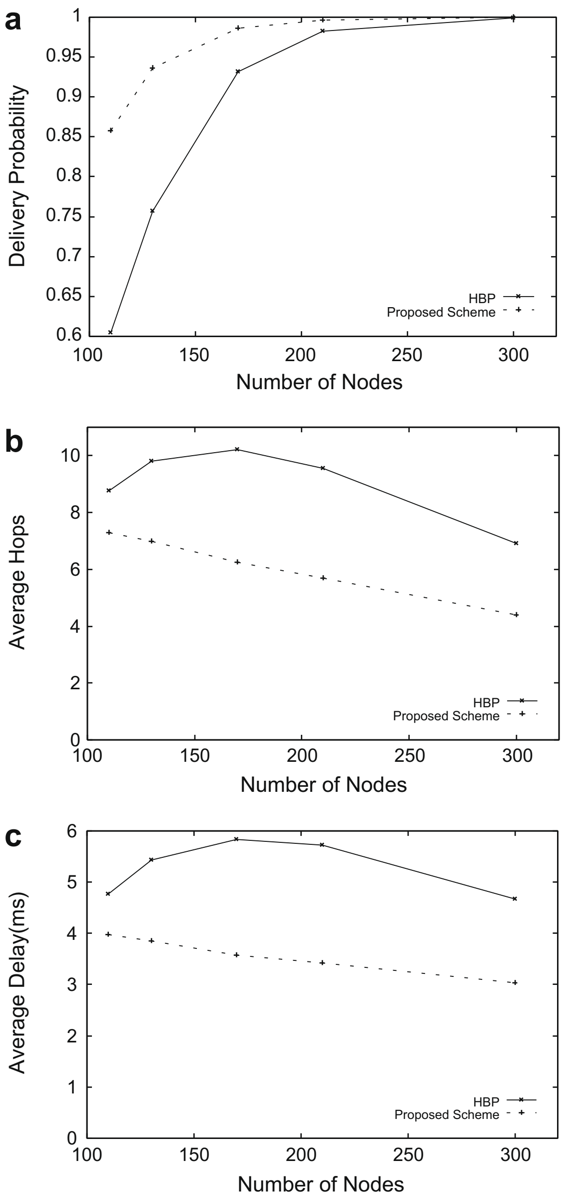


Fig. 10. Performance as a function of nodes density when network coverage area size = 2000 (m) and $V_{max} = 20$ m/s.

With respect to the route length and latency, the proposed scheme reports a significant enhancement in both metrics in comparison with the HBP as shown in Fig. 10b and c, respectively. This is because the performance of the HBP algorithm mainly depends on the nodes-density. With the decrease of nodes-density, the number of neighbors decreases and hence the source nodes became able to find routes to nearby destinations only, while they cannot do the same with the faraway destinations, due to the lack of topology information. With the increase of nodes-density, those faraway destinations become reachable, and therefore, the route length and latency increase. However, this information is still not enough to find the shortest possible routes. With the further increase of nodes-density, the source nodes become able to find alternative and shorter routes and hence decrease the route length and latency.

On the other hand, the proposed scheme does not suffer from the same lack of information as the HBP scheme does. The acquired information about the network topology is always enough to find routes to the near and faraway destinations. With the further increase of neighbors the performance was naturally enhanced due to the availability of shorter alternative routes. This is why the increase of nodes-density (and hence the number of neighbors) always results in reducing the route length and latency.

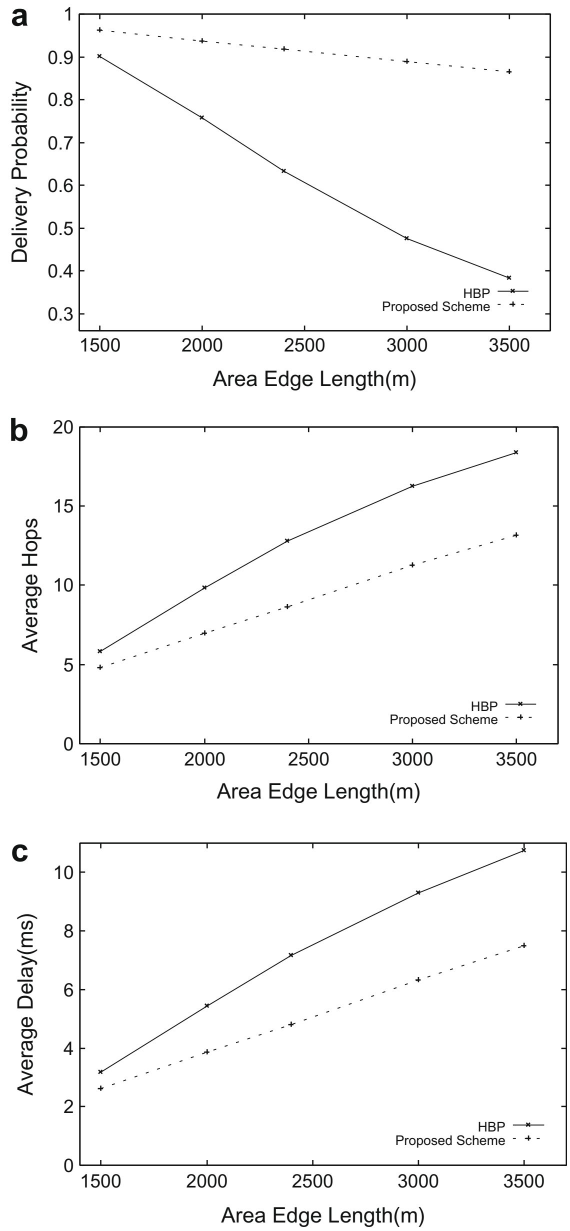


Fig. 11. Performance as a function of network coverage area size when $V_{max} = 20$ m/s.

In general, the conducted simulation studies demonstrate that the proposed scheme outperforms the available one not only under spares network conditions, but also in case of dense networks.

5.3.2. Performance over network area size

Here, we investigate the influence of the network area size under the worst network conditions (i.e., low nodes-density with a low look-ahead $L = 2$ and high node mobility). We keep the average number of neighbors for each node to a constant number of seven neighbors (i.e., the nodes-density is fixed) and high mobility is applied ($V_{max} = 20$ m/s).

The results in Fig. 11 report that the delivery probability of the proposed scheme is not only higher than the original one, but also the route length and the latency are lower. This enhancement increases with the expansion of the network area size.

As we explained before to reduce overhead we apply a low value of look ahead ($L = 2$). For HBP in a small network area, a small L may be enough to discover a big portion of the overall network topology, while using the same L with the expansion of network area may lead to a significant lack of information and hence a severe degradation in the quality of routing. However, the use of a big value of L to overcome the lack of information will result in an extensive overhead especially in a dense network case. On the other hand, due to the efficient and inexpensive information retrieval method, the proposed scheme shows a robust performance even with the increase of the network area size.

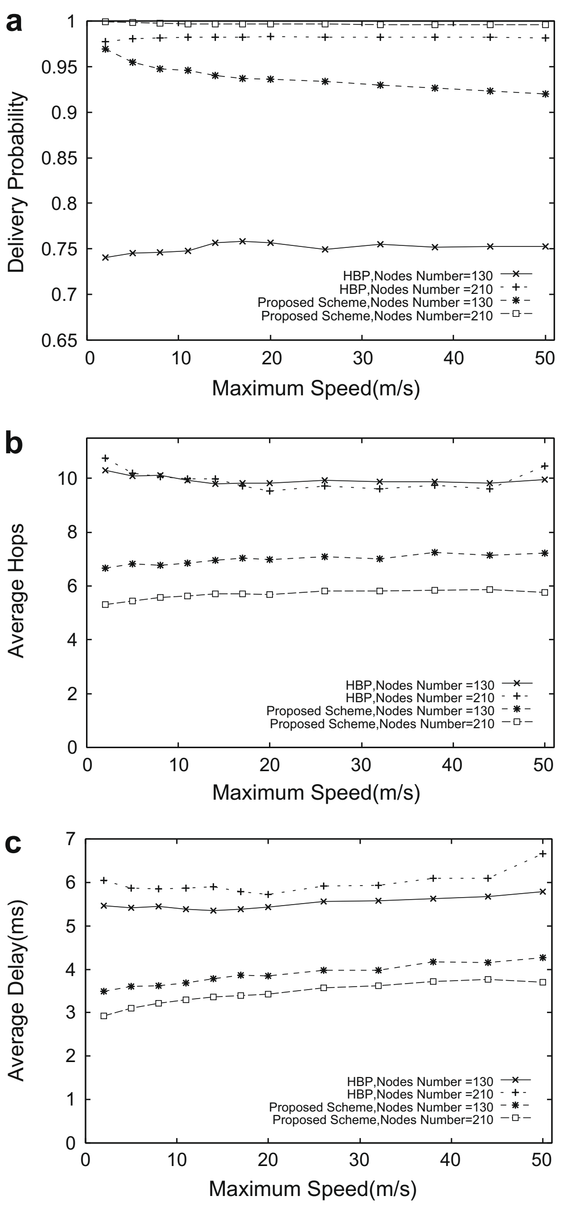


Fig. 12. Performance as a function of nodes mobility when network area edge size = 2000 m and nodes number = 130, 210.

5.3.3. Performance over nodes-mobility

Now as we showed that our scheme is robust under different network sizes (i.e., different nodes-densities and different area sizes). Now, we show that our scheme does not sacrifice the mobility resiliency, which was already supported by the original HBP scheme. Therefore, the following set of simulation studies has been conducted to evaluate the mobility resiliency of our proposed scheme under different network settings: the node density was ($N = 130, 210$) in an area of default size $E = 2000$ m.

As shown in Fig. 12, the obtained results in terms of delivery probability, route length, and latency indicate that the proposed scheme is also resilient to the nodes-mobility.

6. Conclusion

In this paper, we provided an overview of the available routing schemes and their limitations in mobile Ad Hoc networks. We then proposed a novel topology information retrieval scheme. Based on this method we introduced an enhanced Hint-based Probabilistic Protocol which has the ability to overcome such limitations and work efficiently under any network topology: nodes-density and coverage area size, and also mobility unlike the available schemes, which assume specific settings of some of these factors. The extensively conducted simulation-based analysis have not only verified the enhanced scheme has the capability to work under any network setting, but also demonstrates the significant enhancement of the routing process in terms of delivery probability, route length, and latency with a high resiliency to the nodes-mobility. On the other hand, this scheme assumes two way communications. In case the communication is only from origination node to destination, such scheme triggers slight additional computation with no considerable enhancement in the routing process.

Acknowledgement

This work was supported in part by Global COE program, Tohoku University Center of Education and Research for Information Electronics Systems, and JSPS Grant-In-Aid of Scientific Research (B) 21300018, Japan.

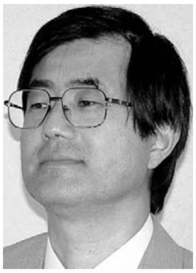
References

[1] E.M. Belding-Royer, C.E. Perkins, Evolution and future directions of the ad hoc on-demand distance-vector routing protocol, Ad Hoc Networks 1 (1) (2003) 125–150.
[2] C. Sengul, R. Kravets, Bypass routing: an on-demand local recovery protocol for ad hoc networks, Ad Hoc Networks 4 (3) (2006) 380–397.
[3] M. Abolhasan, T. Wysocki, E. Dutkiewicz, A review of routing protocols for mobile ad hoc networks, Ad Hoc Networks 2 (1) (2004) 1–22.
[4] L. Reddeppa Reddy, S.V. Raghavan, SMORT: scalable multipath on-demand routing for mobile ad hoc networks, Ad Hoc Networks 5 (2) (2007) 162–188.
[5] G. Pei, M. Gerla, T.W. Chen, Fisheye state routing: a routing scheme for ad hoc wireless networks, in: Communications, 2000. ICC 2000. 2000 IEEE International Conference on, vol. 1, 2000.
[6] C.K. Toh, Associativity-based routing for ad hoc mobile networks, Wireless Personal Communications 4 (2) (1997) 103–139.
[7] C.E. Perkins, E.M. Royer, Ad-hoc on-demand distance vector routing, in: Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, vol. 2, February, 1999, pp. 90–100.
[8] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, Kluwer International Series in Engineering and Computer Science (1996) 153–179.
[9] C.E. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, ACM SIGCOMM Computer Communication Review 24 (4) (1994) 234–244.
[10] R.G. Ogier, F.L. Templin, B. Bellur, M.G. Lewis, Topology broadcast based on reverse-path forwarding (TBRPF), draft-ietf-manettbrpf-05.txt, 1, 2002.
[11] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, L. Viennot, Optimized link state routing protocol for ad hoc networks, IEEE INMIC 1 (2001) 28–30.
[12] V.D. Park, M.S. Corson, A highly adaptive distributed routing algorithm for mobile wireless networks, in: IEEE INFOCOM, vol. 3, Institute of Electrical Engineers Inc (IEEE), 1997, pp. 1405–1413.
[13] L. Rosati, M. Berioli, G. Reali, On ant routing algorithms in ad hoc networks with critical connectivity, Ad Hoc Networks 6 (6) (2008) 827–859.
[14] I. Bouazizi, ARA-the ant-colony based routing algorithm for MANETs, in: Proceedings of the 2002 International Conference on Parallel Processing Workshops, IEEE Computer Society Washington, DC, USA, 2002.
[15] H.F. Wedde, M. Farooq, T. Pannenbaecker, B. Vogel, C. Mueller, J. Meth, R. Jeruschkat, BeeAdHoc: an energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior, in: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, ACM New York, NY, USA, 2005, pp. 153–160.
[16] G. Di Caro, F. Ducatelle, L.M. Gambardella, AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks, European Transactions on Telecommunications 16 (5) (2005) 443.
[17] M. Roth, S. Wicker, Performance evaluation of pheromone updates in swarm intelligent MANETs, in: The Sixth IFIP IEEE International Conference on Mobile and Wireless Communication Network, 2005.
[18] M. Roth, S. Wicker, Termite: ad-hoc networking with stigmergy, in: Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE, vol. 5, 2003.
[19] R. Beraldi, L. Querzoni, R. Baldoni, A hint-based probabilistic protocol for unicast communications in MANETs, Ad Hoc Networks 4 (5) (2006) 547–566.
[20] R. Beraldi, The polarized gossip protocol for path discovery in MANETs, Ad Hoc Networks 6 (1) (2008) 79–91.
[21] R.R. Choudhury, K. Paul, S. Bandyopadhyay, MARP: a multi-agent routing protocol for mobile wireless ad hoc networks, Autonomous Agents and Multi-Agent Systems 8 (1) (2004) 47–68.
[22] N. Migas, W.J. Buchanan, K.A. McArtney, Mobile agents for routing, topology discovery, and automatic network reconfiguration in ad-hoc networks (2003).
[23] S. Marwaha, C. Khong, T.D. Srinivasan, Mobile agents based routing protocol for mobile ad hoc networks, in: Proceedings of IEEE Globecom, 2002.
[24] R. Beraldi, On message delivery through approximate information in highly dynamic mobile ad hoc networks, in: The Seventh International Symposium on Wireless Personal Multimedia Communications, Italy, vol. 12, 2004, p. 15.
[25] A.B. McDonald, T.F. Znati, A mobility-based framework for adaptive clustering in wireless ad hoc networks, IEEE Journal on Selected Areas in communications 17 (8) (1999) 1466–1487.
[26] A. Nasipuri, R. Castañeda, S.R. Das, Performance of multipath routing for on-demand protocols in mobile ad hoc networks, Mobile Networks and Applications 6 (4) (2001) 339–349.
[27] J. Le Boudec, M. Vojnovic, Perfect simulation and stationarity of a class of mobility models, in: IEEE INFOCOM, vol. 4, Citeseer, 2005, pp. 2743–2754.
[28] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, Kluwer International Series In Engineering and Computer Science (1996) 153–179.

Keyvan Kashkouli Nejad received his M.S. and B.S. degree from Tohoku University, Sendai, Japan, in 2007 and 2009, in engineering and computer sciences respectively. He is currently working toward Ph.D. degree in the Graduate School of Information Science in Tohoku University, Japan. His research interests include mobile ad-hoc networks, and passive RFID identification.

Ahmed Shawish received the B.S. and M.S. degree from Ain Shams University, Cairo, Egypt in 1997 and 2002, all in Computer Sciences. Since 2005, he was on a fellowship/mission provided by the Egyptian government to get the Ph.D. degree from the Graduate school of Information Science, Tohoku University, Japan. He accomplished this mission in 2009. His current research focuses on supporting the Voice over Internet-Protocol (VoIP) applications over wired and wireless networks. He is a student member of the IEEE.

Xiaohong Jiang received his B.S., M.S. and Ph.D degrees in 1989, 1992, and 1999 respectively, all from Xidian University, Xi'an, China. He is currently an Associate Professor in the Department of Computer Science, Graduate School of Information Science, TOHOKU University, Japan. Before joining TOHOKU University, Dr.Jiang was an assistant professor in the Graduate School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), from October 2001 to January 2005. He was a JSPS (Japan Society for the Promotion of Science) postdoctoral research fellow at JAIST from October 1999–October 2001. He was a research associate in the Department of Electronics and Electrical Engineering, the University of Edinburgh from March 1999–October 1999. His research interests include optical switching networks, routers, network coding, WDM networks, VoIP, interconnection networks, IC yield modeling, timing analysis of digital circuits, clock distribution and fault-tolerant technologies for VLSI/WSI. He has published over 130 referred technical papers in these areas. He is a senior member of IEEE.



Susumu Horiguchi (M'81-SM'95) received the B. Eng. the M. Eng. and PhD degrees from Tohoku University in 1976, 1978 and 1981 respectively. He is currently a Full Professor in the Graduate School of Information Sciences, Tohoku University. He was a visiting scientist at the IBM Thomas J. Watson Research Center from 1986 to 1987. He was also a professor in the Graduate School of Information Science, JAIST (Japan Advanced Institute of Science and Technology). He has been involved in organizing international workshops, symposia and conferences sponsored by the IEEE, IEICE, IASTED and IPS. He has published over 150 papers technical papers on optical networks, interconnection networks, parallel algorithms, high performance computer architectures and VLSI/WSI architectures. He Horiguchi is members of IEICE, IPS and IASTED.