Opportunistic Routing with Minimum Latency for MANETs with Intermittent Link Failures

VIJAYALAKSHMI Gurunathan FUJITA Satoshi

I. Introduction

According to the explosive growth of network technologies, it increases the demand for the mobile communication in the environment with no stationary infrastructure such as 5G cellular and wireless mesh networks. Although mobile ad hoc networks (MANETs) have been widely used as a key technology which can cope with such a situation, most MANET protocols assume the existence of stable inter-node links [4], [5], [7], [10], [11], [12], and it is not easy to keep a high availability due to various reasons, such as rapid changes in radio wave conditions and the battery exhaustion of relaying nodes. The concept of **delay tolerant networks** (DTNs) was proposed about 20 years ago to model wireless networks which incorporate such intermittent link failures [2], [3], [6], [8], [9], [13], [14], [15], [16].

[*** Insert a brief history of DTN ***]

PROPHET [9] (Probabilistic ROuting Protocol using History of Encounters and Transitivity) is a message routing protocol proposed for DTNs. The basic idea of PROPHET is to determine the next node on the delivery route of a message with the notion of **delivery predictability** of the nodes[1], where for each message transmission, the receiver of the message must be within the transmission radius of the transmitting node to successfully forward the message to the next node.

Let $p_{a,b}$ denote the delivery predictability of node a concerned with destination b, where it should be noted that it is *not* an attribute of messages, but an attribute of nodes. In PROPHET, the value of $p_{a,b}$ is dynamically updated according to the following three types of events. At first, when node a encounters with node b, it is updated as

$$p_{a,b} := p_{a,b} + (1 - p_{a,b})\alpha, \tag{1}$$

where α is a parameter which is typically fixed to 0.75 [9]. Since the second term in the formula is positive, and it holds $p_{a,b} < 1.0$ for any $0 < \alpha < 1$, the value of $p_{a,b}$ geometrically approaches to 1.0 as repeating encounters. Next, if a does not encounter with b for a certain time period, $p_{a,b}$ is updated as

$$p_{a,b} := p_{a,b} \times \gamma, \tag{2}$$

where γ is a parameter called aging which is typically set to 0.98 [9]. Note that aging can occur several times for a long duration of time; namely, it gradually converges to zero. The above two update rules merely consider the direct encounter of a and b, but in actual mobile communication, the message delivery is realized by repeating message transfer through intermediate nodes, since even if a does not meet b in person, it is possible to hand a message to b through a common friend c. The third update rule reflects the transitive nature of such data transfer. Specifically, when node a encounters with node $c(\neq b)$, $p_{a,b}$ is updated as

$$p_{a,b} := p_{a,b} + (1 - p_{a,b}) \times p_{a,c} \times p_{c,b} \times \beta, \tag{3}$$

where β is an appropriate parameter.

Outline of our contribution ***

- the delivery predictability of a node introduced in PROPHET does not represent the expected latency of the message delivery through the node.
- of course, the delivery of a message to a designated destination through many intermediate nodes would cause a small delivery predictability of the next node, but it is indirect (in contrast, many proactive routing protocols such as AODV explicitly count the number of hops towards the destination)
- we would like to introduce the notion of (expected) latency to the heuristic evaluation value of nodes concerned with the designated destination.

II. Related Work

Several extensions of PROPHET are proposed in the literature. Priority-enhanced PROPHET [1] prioritizes messages exchanged among nodes so as to allow urgent messages to reach their destination faster, while it is still realized in the best effort manner. The urgency of messages is acquired through natural language processing, and an important use case of this extension is the information sharing in an area suffer from disaster. In the original PROPHET, each relay node deletes message contained in the buffer after passing enough time (e.g., few minutes) so as to guarantee the reach of the message to the destination. Wang *et al.* [14] proposed a technique to increase the size of free space in the buffer by forcing the destination of messages to explicitly returns an ACK message to relay nodes.

There are many previous works concerned with the selection policy of the next node. EA-PROPHET proposed in [2] takes into account the buffer availability and the life time of the buttery of nodes in addition to the delivery predictability to the destination. PROPHET+ [6] extends PROPHET so as to take into account the buffer size, buttery capacity, node location, and the node popularity. The consideration of the buttery capacity is also done in [3]. PROPHET-CLN [15] extends PROPHET to take into account the congestion level of nodes,

[1] While there are several options in the usage of the delivery predictability in routing protocols, [9] proposed a simple protocol so that: if node a holds a message towards destination b and if it encounters with node $c(\neq b)$ with $p_{c,b} > p_{a,b}$, then a forwards a copy of the message to c.

which could be detected by monitoring the free space of the buffer. DiPROPHET (Distance-based PROPHET) proposed in [13] takes into account the physical distance to the destination by adopting a cross-layer approach to acquire the RSSI value of received messages.

PROPHETv2 [8] extends PROPHET so as to take into account the encounter intervals in updating the delivery predictability. This allows us to effectively eliminate misleading cases in which a node repeatedly encounters with another node in a very short time period as in the car movement in a parking area. Finally, [16] extends PROPHET to reduce the variance of jitter.

Add more previous works: papers after 2017, epidemic-based schemes, and the learning of probability distribution from past experiences.

III. Network Model

Let V be a set of **stationary** nodes with fixed locations and U be a set of **mobile** nodes which can move freely in a given space. Stationary and mobile nodes can directly communicate with each other through short-range wireless communication such as Wi-Fi and Bluetooth as long as their locations are close enough (e.g., few meters). The time required for the message transmission is negligibly short and, for simplicity, the moving speed of mobile nodes is assumed to be a constant. A message emanating from node i to node j is picked up by a mobile node passing near i, travels with the mobile node, and is dropped near j. Each message emanated from a source node can pass through other stationary nodes and change to other mobile nodes encountered along the way to the destination. Such a message delivery is realized by repeating store-and-forward operation, where each node has a local buffer of enough size to store all incoming messages so that we do not need to care about message drops due to buffer full.

Each link becomes *unavailable* if the distance between two incident nodes is longer than the transmission radius of BLE (e.g., 20m), and such an intermittent disconnection occurs according to a certain probability distribution. In this paper, we consider the problem of delivering a given message to the destination with as small maximum delivery time as possible.

A. Contact Possibility Graph

In this paper, we assume that any two stationary nodes are located far apart so that the direct communication between them is not possible, and the contact of two mobile nodes could *not* be predicted beforehand, although it enables the direct communication between them. Each mobile node v has a unique **range of activity** (e.g., around the residence for retired person and between residence and school for primary school children), and let $R(v)$ denote the set of stationary nodes contained in the activity range of v. Such a possibility of contacts between stationary and mobile nodes is modeled by a bipartite graph $G = (A, B, E)$ with vertex set $A \cup B$ and edge set E so that: $u \in A$ and $v \in B$ are connected by an edge in E iff $u \in R(v)$. In the following, we call such G the **contact possibility graph** for the given wireless network.

IV. Proposed Method

A. Directed Bipartite Graph with Edge Weight

In this paper, we are interested in the problem of routing a message from source node s to destination node t as quickly as possible. Such a message routing can be regarded as the process of message forwarding towards destination t by changing mobile nodes as needed. If the contact possibility graph G is explicitly given, then we can realize the minimum hop count routing to t by each node forwarding the received message in the direction so that the minimum hop count to t becomes smaller (similar to AODV).

On the other hand, if we wish to minimize the *actual latency* of the message delivery (rather than the number of hops), we should take into account the *waiting time* for a mobile node to arrive at a given stationary node and the *travel time* to the next stationary node from the given stationary node. Although it might be possible to calculate an optimal scheduling based on the complete knowledge about the behavior of individual mobile nodes by using a generic integer programming solver, in this paper, we would like to take a macroscopic approach to treat it as a stochastic process, since it could better reflect fluctuations in travel routes and travel times (remind that primary school children often take a detour even if the school dismisses at a fixed time). In order to reflect the above two characteristics (i.e., the waiting time and the travel time), we will extend G to a directed graph $\hat{G}$ in the following manner:

1) for each $v \in B$, replace mobile node v by a collection of $|R(v)|$ virtual nodes so that each virtual node corresponds to a stationary node $u \in R(v)$;
2) for each $v \in B$ and $u \in R(v)$, connect u and all virtual nodes w derived from v by a directed edge (u, w) with the weight corresponding to the expected waiting time for v at node u.
3) for each $v \in B$ and $u, u' \in R(v)$, connect virtual node w corresponding to u' and stationary node u by a directed edge (w, u) with the weight corresponding to the expected travel time from u' to u.

We then apply an AODV-like approach to realize the message delivery towards the given destination along a path with the minimum expected latency (note that this calculation depends on the *linearity of expectation*).

B. Estimation of Edge Weight

The remaining task is to estimate the expected waiting and travel times as accurately as possible. The followings are hints to improve the accuracy of estimation.

- The number of sample data is generally very small.
- If two mobile nodes have similar range of activities, we could merge them into a single mobile nodes with more sample data and shorter waiting time.

V. Distributed Implementation

Network routing protocols can be roughly divided into two categories called distance-vector type and link-state type. The message routing using the Dijkstra's algorithm is suitable for the link-state protocol, in which link weights are collected to a specific place such as server and each source determines the route of a message based on them. However, in general DTNs which are the target of PROPHET, the state of links changes dynamically, so that it often changes even during aggregation, or an efficient aggregation itself is difficult.

Therefore, in this study, we will consider the message routing of distance-vector type, in which the next node on the delivery route is adaptively selected by each relay node by referring to the routing table locally held by each node. AODV is one typical protocol to maintain such a routing table in a wireless network. In the proposed method, the weight of each link is acquired in a distributed manner similar to AODV, in the following manner:

- Each node computes the link encounter probability p for each neighbor based on local observations and takes the inverse of the probability as the link weight (if we want to take into account the message transfer time, we can increase the link weight by that amount. Please be mind that the dimension of weight is time).
- The source node floods request packets to each destination, and fills the table entries by aggregating reply packets returned by the destination (or its neighboring nodes). During the aggregation, we can try to leave the next hop on the path with the lowest cumulative weight.

A. AODV

The proposed method uses AODV (Ad hoc On-Demand Distance Vector) to find a route from the source to the final destination. AODV is a reactive routing protocol designed for mobile ad hoc network (MANETs) [**?**]. More concretely, a source which wishes to find a route to the final destination broadcasts RouteRequest packet to its nearby nodes, which contains source identifier SrcID, destination identifier DestID, source sequence number SrcSeqNum, destination sequence number DesSeqNum, broadcast identifier BcastID and TTL. SrcSeqNum and DesSeqNum indicate the freshness of the route accepted by the source at that time and the freshness of the route to the destination, respectively (the larger the newer). TTL is used to limit the times of packet forwarding.

A node u receiving RouteRequest packet from a neighbor v returns RouteReply packet to v if either u is DestID or u knows a correct route to the destination, where a route is said to be correct if it has a larger DesSeqNum than the packet. Otherwise u forwards the received packet to the other neighbors after decrementing TTL by one (as long as it has positive TTL, of course). It then locally caches DesSeqNum and BcastID in the packet for a certain period of time so that they can be used for the matching with RouteReply packet sent back from neighbors. If u receives RouteRequest packet with the same SrcID and BcastID several times, it discards later packets to reduce the exploration cost.

RouteReply packet records the number of hops to the destination and is delivered to the originator of the corresponding RouteRequest packet along the route of the packet in the reverse direction. After receiving RouteReply packets from neighbors, the node selects the route with the smallest number of hops and updates the routing table so that the next node along the selected route is recorded for each destination.

VI. Evaluation

VII. Concluding Remarks

References

[1] S. Bhattacharjee, S. Basu, S. Roy, and S. D. Bit. "Best-effort delivery of emergency messages in post-disaster scenario with content-based filtering and Priority-enhanced PROPHET over DTN." in *Proc. 8th Int'l Conf. on Communication Systems and Networks (COMSNETS)*, 2016.

[2] B. B, Bista and D. B. Rawat. "EA-PROPHET: An Energy Aware PROPHET-Based Routing Protocol for Delay Tolerant Networks." in *Proc. 31st Int'l Conf. on Advanced Information Networking and Applications (AINA)*, IEEE, 2017, pages 670–677.

[3] S. J. Borah, S. K. Dhurandher, S. Tibarewala, I. Woungang, and M. S. Obaidat. "Energy-Efficient PROPHET-PRoWait-EDR Protocols for Opportunistic Networks." in *Proc. GLOBECOM*, IEEE, 2017.

[4] T. W. Chen and M. Gerla. Global state routing: a new routing scheme for ad-hoc wireless networks. in *Proc. IEEE Int'l Conf. on Communications*, 1998. p. 171–5.

[5] T. Clausen, G. Hansen, L. Christensen, and G. Behrmann. The optimized link state routing protocol, evaluation through experiments and simulation. in *Proc. IEEE Symp. on wireless personal mobile communications*, 2001.

[6] T.-K. Huang, C.-K. Lee, and L.-J. Chen. "PROPHET+: An Adaptive PROPHET-Based Routing Protocol for Opportunistic Network." in *Proc. 24th Int'l Conf. on Advanced Information Networking and Applications (AINA)*, 2010, pages 112–119.

[7] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, 1996:153–81.

[8] H.-J. Lee, J.-C. Nam, W.-K. Seo, Y.-Z. Cho, and S.-H. Lee. "Enhanced PROPHET routing protocol that considers contact duration in DTNs." in *Proc. Int'l Conf. on Information Networking (ICOIN)*, 2015.

[9] A. Lindgren, A. Doria, and O. Schelén. "Probabilistic Routing in Intermittently Connected Networks." in *Dini P., Lorenz P., de Souza J.N. (eds) Service Assurance with Partial and Intermittent Resources. SAPIR 2004*, Lecture Notes in Computer Science, vol 3126. Springer, Berlin, Heidelberg.

[10] S. Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mobile Networks and Applications*. 1:183–97, 1996.

[11] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *ACM SIGCOMM Computer Communication Review*, 24:234–44, 1994.

[12] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. in *Proc. 2nd IEEE Workshop on mobile computing systems and applications (WMCSA'99)*, 1999. pages 90–100.

[13] P. Sok, S. Tan, and K. Kim. "PROPHET Routing Protocol Based on Neighbor Node Distance Using a Community Mobility Model in Delay Tolerant Networks." in *Proc.10th Int'l Conf. on High Performance Computing and Communications & Int'l Conf. on Embedded and Ubiquitous Computing*, IEEE, 2013.

[14] P. Wang and F. Shen. "Method to Improve the Performance of PROPHET Routing Protocol." in *Proc. 2nd Int'l Workshop on Education Technology and Computer Science*, 2010.

[15] G. Wang, J. Tao, H. Zhang, and D. Pan. "A improved PROPHET routing based on congestion level of nodes in DTN." in *Proc. 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, IEEE, 2017.

[16] J. Xue, J. Li, Y. Cao, and J. Fang. "Advanced PROPHET Routing in Delay Tolerant Network." in *Proc. Int'l Conf. on Communication Software and Networks*, 2009.