ISSN:0975-9646

Mohsin Naziruddin / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, 7841-7843

Impact of queuing policy variations on MaxProp DTN routing protocol

Mohsin Naziruddin

Department of Computer Science and Engineering, SRM University
Kancheepuram District,
Kattankulathur – 603203,
Tamil Nadu, India

***Abstract*— A Delay/Disruption Tolerant Network is a network where temporary or intermittent connectivity among all nodes exist. Conventional routing protocols require end to end connectivity whereas DTN does not require such connectivity. This leads to the critical problem of how to route a packet from one node to another, in such a network. This problem becomes more complex, when the node mobility is also considered. In real mobility patterns the nodes always move in some predictable fashion rather than the random movement.This paper aims at visualizing the impact of the queuing policies on the MaxProp[1] routing protocol. MaxProp is based on prioritizing both the schedule of packets transmitted to other peers and the schedule of packets to be dropped. These priorities are based on the path likelihoods to peers according to historical data and also on several complementary mechanisms, including acknowledgments, a head-start for new packets, and lists of previous intermediaries.**

The Maxprop protocol follows a buffer split strategy to increase the delivery ratio. It makes the transmission as well as deletion of messages (in case of buffer overflows) based on the threshold separated by means of hop counts and costs. Simulations are conducted to analyze the impact of queuing policy by prioritizing on hopcounts or costs only and without acknowledgements.

***Keywords*— DTN, Maxprop, queuing policy, reachability**

I. INTRODUCTION

Routing in delay-tolerant networking concerns itself with the ability to transport, or route, data from a source to a destination, which is a fundamental ability all communication networks must have. Delay- and disruption-tolerant networks (DTNs) are characterized by their lack of connectivity, resulting in a lack of instantaneous end-to-end paths. In these challenging environments, popular ad hoc routing protocols such as AODV and DSR fail to establish routes. This is due to these protocols trying to first establish a complete route and then, after the route has been established, forward the actual data. However, when instantaneous end-to-end paths are difficult or impossible to establish, routing protocols must take to a "store and forward" (main backbone technology for DTN) approach, where data is incrementally moved and stored throughout the network in the hope that it will eventually reach its destination. A common technique used to maximize the probability of a message being successfully transferred is to replicate many copies of the message hoping that it will succeed in reaching its destination. There are many characteristics DTN protocols, including routing, must take into consideration.

A first consideration is if information about future contacts is readily available in scheduled or predictable contacts as in the case of interplanetary routings. On the contrary, in disaster recovery networks the future location of communicating entities, such as emergency responders, may not be known. These types of contacts are known as intermittent or opportunistic contacts.

A second consideration is if mobility can be exploited and, if so, which nodes are mobile. There are three major cases, classifying the level of mobility in the network. First, it is possible that there are no mobile entities. In this case, contacts appear and disappear based solely on the quality of the communication channel between them. For instance, in interplanetary networks, large objects in space, such as planets, can block communicating nodes for a set period of time. Second, it is possible that some, but not all, nodes in the network are mobile. These nodes, sometimes referred to as Data Mules, are exploited for their mobility. Since they are the primary source of transitive communication between two non-neighbouring nodes in the network, an important routing question is how to properly distribute data among these nodes. Third, it is possible that the vast majority, if not all, nodes in the network are mobile. In this case, a routing protocol will most likely have more options available during contact opportunities, and may not have to utilize each one. An example of this type of network is a disaster recovery network where all nodes (generally people and vehicles) are mobile. A second example is a vehicular network where mobile cars, trucks, and buses act as communicating entities.

A third consideration is the availability of network resources. Many nodes, such as mobile phones, are limited in terms of storage space, transmission rate, and battery life. Others, such as buses on the road, may not be as limited. Routing protocols can utilize this information to best determine how messages should be transmitted and stored to not over-burden limited resources.

II. BACKGROUND

MaxProp was developed at the University of Massachusetts, Amherst by Burgess et al. [1]. MaxProp is flooding-based in nature, in that if a contact is discovered, all messages not held by the contact will attempt to be replicated and transferred (often called as summary vector exchange). MaxProp tries to determine the priorities of message transfer, i.e. which messages should be transmitted first and which messages should be dropped first. MaxProp maintains an ordered queue based on the destination of each message, ordered by the estimated likelihood of a future transitive path to that destination.

MaxProp releases the bias toward short-distance destinations, by using hop counts in packets, serving at the same time as a measure of network resource fairness. Though acknowledgments are propagated network-wide, and not just to the source, MaxProp addresses stale data. Finally, MaxProp stores a list of previous intermediaries to prevent data from propagating twice to the same node. While these ideas were simple, the authors' experiments [1] show they significantly raise the delivery rate and lower latency in a wide variety of scenarios as compared to previous approaches.

The initial MaxProp [1] scenario assumes that each peer has an effectively unlimited buffer for messages produced, but a fixed-size buffer for carrying and relaying messages originated by others. Transfer opportunities are assumed to be limited both in duration and bandwidth. Furthermore, nodes are assumed have no a priori knowledge of network connectivity, no control over their movement, no knowledge of geographic location. In a real network, the opportunistic communication proceeds roughly through three stages.

- Neighbor Discovery, where Peers must discover one another before a transfer opportunity can begin; and they do not know when the next opportunity will begin.
- Data Transfer. When two peers meet, the amount of data they can transfer is limited. Peers do not know the duration of each opportunity.
- Storage management. As packets are received from a neighbor, each peer must manage its finite local buffer space by selecting packets to delete according to some criterion or algorithm.

Messages that are destined for a receiving peer are passed up to the application layer and removed from the buffer. Each peer carries all messages until a subsequent meeting occurs. A peer will continue to forward a message to any number of other peers until its copy of the message times out, it is notified of delivery by an acknowledgment, or the message is dropped due to a full buffer. Regarding priorities packets that are ranked with highest priority are the first to be transmitted during a transfer opportunity. Packets ranked with lowest priority are the first to be deleted to make room for an incoming packet. When two packets have destinations with the same cost, the tie is broken by giving the packet that has traveled fewer hops higher priority.

When two peers discover each other, MaxProp proceeds to exchange packets along the following steps [2]:

- All messages destined to the neighbor peer are transferred.
- routing information is passed between peers: This is done through a vector listing estimations of the probability of meeting every other node.
- acknowledgments of delivered data are transferred, regardless of source and destination. An acknowledgment consists of a cryptographic hash of the content, source, and destination of each message. (Note that this mechanism aims to clear out buffers in the network of old data at a low overhead cost given that the ACK messages are small—compared to data packets). In the original paper evaluation [1], peers would not spend more than 1% of the historical average connection duration on sending acknowledgments.
- packets that have not spent many hops in the network are given priority. This is because estimating the delivery likelihood can favor packets that have a high chance of reaching a destination, causing some packets to never get a chance to be transmitted. Therefore, MaxProp attempts to give new packets a “head start” in the network by giving them a higher priority. The effect of this approach is that newer packets are transmitted at several transfer opportunities when they are generated, thus expanding fast toward the destination. To implement this strategy, MaxProp splits the buffer in two logical sub-buffers, according to whether the packets have a hop count less than a threshold. Packets below the threshold are sorted by hop count. Since a static threshold assignment would be arbitrary and might not work in all environments. MaxProp takes an adaptive approach to setting the threshold. In environments where the average number of bytes transferred per encounter is much smaller than the buffer size, MaxProp prioritizes low hop count packets. As the size of transferred batches grows, the threshold is progressively reduced to the difference between the two values. When transfer batch size is larger than the buffer size, the threshold is completely removed since it is no longer of any effect.
- the remaining, untransmitted packets are sent in an order the Estimating Delivery Likelihood, which in turn is based on variation of Dijkstra's algorithm.
- packets that have already been sent to the node are not sent again. A hop list in each packet stores peers that the packet has already traversed, including peers to which the current node has sent the packet.

As a summary, MaxProp removes acknowledged packets instantly, followed by packets that have crossed the threshold of t intermediate hops with minimum scores, followed by packets with the maximum hops below threshold t.

III. EVALUATION

The effect of the change in queuing policies is identified by means of a simulation experiment.

A. *Comparisons*

1) *Buffer split without acknowledgement:* The original MaxProp implementation without the deletion of delivered messages at encounter time (at peer nodes) to check for its impact on reachability. The queuing policy of buffer split of

hop counts and cost metrics while disabling acknowledgement functionality.

2) *Hop count alone without acknowledgement*: The Maxprop implementation relying completely on hop counts favouring the time of packet's generation as priority while disabling acknowledgement functionality.

3) *Cost alone without acknowledgement*: The Maxprop implementation relying completely on costs favouring the delivery likelihood estimate of packet as priority while disabling acknowledgement functionality.

4) *Buffer split with acknowledgement:* The original MaxProp implementation with the queuing policy of buffer split of hop counts and cost metrics while enabling acknowledgement functionality.

B. Experiment

The experiment has been performed on ONE (Opportunistic Network Environment) simulator [3]. The scenario with Helsinki city map and parameters (as mentioned in Table 1) has been simulated for a period of 6 hours for the varying queuing policies for Maxprop protocol.

TABLE I
SIMULATION PARAMETERS

Parameters	**Values**
Simulator used	ONE Simulator
Simulation time	21600s (6 hours)
Number of nodes	80 pedestrians 40 cars 6 trams
Movement model	Shortest path map based movement (Helsinki city map)
Buffer size	50MB (Trams) 5 MB (Pedestrians and cars)
Message size	500 KB
Message TTL	5 hours
Generation interval	Equal intervals for entire simulation time
Number of packets	100,200,300,400,500

C. Results

The simulation experiments for the four queuing policy variations of the Maxprop routing protocol has been conducted and the result obtained is shown in Fig.1.

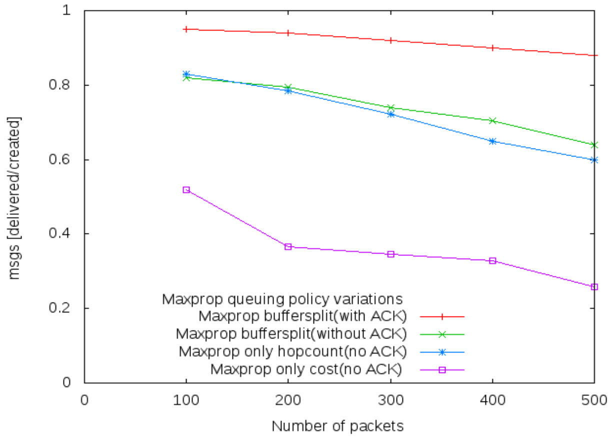

Fig. 1 A graph showing the reachability for varying queuing policies of Maxprop

The graph shows maximum reachability for the original implementation. The priority of queuing policy based on hopcounts as well as the protocol without acknowledgement seems to show similar reachability. The cost based queuing policy shows the least reachability.

IV. CONCLUSIONS

The impact of queuing policy variations on MaxProp DTN routing protocol has been verified by means of the simulation experiment conducted. The Maxprop routing protocol with buffer split and acknowledgement happens to give maximum reachability since the deletion of acknowledged messages from the buffer enables accommodation of new messages. Without acknowledgement the protocol seems to considerably decrease its performance for the simulation performed. Moreover the queuing based on hop count alone seems to favour and shows similar reachability with that of the protocol without acknowledgement. The protocol based on only costs shows the least reachability. Thus the buffer split is a requirement for balancing the tradeoff between costs and hopcount queuing policies.

REFERENCES

[1] Burgess J, Gallagher B, Jensen D, Levine B "MaxProp: routing for vehicle-based disruption-tolerant networks", *in the proceedings of the 25th IEEE annual joint conference of the IEEE computer and communications societies (INFOCOM 2006)*

[2] Woungang I, Ddurandher S.K, Anpalagan A, Vasilakos A.V, *Routing in Opportunistic Networks* (Springer)2013,XVI,411 p.83 illus.

[3] Ari Keränen, Jörg Ott, Teemu Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation", *Simutools '09 Proceedings of the 2nd International Conference on Simulation Tools and Techniques*

[4] (2014) The IEEE website. [Online]. Available: http://www.ieee.org/