Paper Review: Routing in a Delay Tolerant Network

Rachit Kumar - 2017CS10364

November 4, 2019

1 Historical Context:

In the 1990s and early 2000s, Inter-Planetary Internet (IPN) architecture was being developed which introduced the need of networking technologies that can cope with the significant delays and packet corruption. Hence from here, Delay Tolerant Networking emerged which was the extension of the ideas in IPN architecture to terrestrial networks. The main motivation of development of this field was due to widespread use of wireless protocols in the 1990s when mobile ad-hoc networking (MANET) became a field of increasing interest due to emergence of personal computers and mobile phones in the market. Earlier, packet radios were popular but they were bulky, had slow data rate and were unable to maintain links if the mobility was high. Hence, this led to the need of new techniques and hence MANET were born. This research paper was hence written to provide an overview and comparison between the then existing routing techniques for routing in a delay tolerant network.

2 Introduction:

Delay tolerant networks are the networks designed to operate mostly over large distances with long latency when there is lack of connectivity and hence lack of availability of end-to-end simultaneous connections. Routing in a delay tolerant network is a form of quickest transshipment problem. When instantaneous end-to-end paths are difficult or impossible to establish, routing protocols have to take to a ”store and forward” approach, where data is incrementally moved and stored throughout the network in hopes that it will eventually reach its destination. In the paper, they have first formulated the problem in a known environment, then have provided a framework for evaluating and showed a simulation-based comparison of various routing algorithms with each other and with an optimal algorithm based on a linear programming approach.

3 The Delay Tolerant Network Model:

A Delay Tolerant Network has been modeled as a directed multi-graph since the set of edges in the graph must capture both time-varying capacity and propagation delay as well as multiple parallel edges. The opportunity to communicate is called a contact which includes the duration of time, a capacity, a propagation delay and buffering constraints. In their framework, a contact is simply an edge and the corresponding time interval when the edge is available, i.e. the edge capacity is positive. Another important part of the model is the message tuple (source, destination, time, size).The storage in the model is exclusively used for holding in-transit data. The routing algorithm determines which edge will be chosen next such that it minimises the delay of a message which indirectly improves the probability of message delivery.

4 Algorithms:

The routing algorithms mentioned use per-hop routing against source routing as they need to utilise local information at each hop. Also, reactive routing has been used at least in the non-basic algorithms. They have tried to explain what information is required by the algorithms using knowledge oracles(Contacts Summary Oracle, Contacts Oracle, Queuing Oracle and Traffic Demand Oracle). The broad category of algorithms discussed are:

- Routing with Zero knowledge: First Contact (FC)-This uses none of the oracles (and hence zero knowledge), a message is forwarded along an edge chosen randomly among all the current contacts.

- Routing with Partial Knowledge: Each message is routed independently of the future demand because the traffic oracle is not used. These are all based upon assigning costs to edges and computing a form of minimum-cost ("shortest") path. Costs are assigned to edges (by consulting the available oracles) to reflect the estimated delay of the message in taking that edge. But only a single path to a destination is derived. These aren't globally optimal as they don't consider traffic demand and buffer constraints. These include Minimum Expected Delay (MED) with only the Contacts Summary oracle, Earliest Delivery with Contacts oracle, Earliest Delivery with Local Queuing (EDLQ) with Contacts oracle but taking into account the "local" queue occupancy and Earliest Delivery with All Queues (EDAQ) with Contacts and Queuing oracle.

- Routing with Complete Knowledge: They've presented a Linear Programming formulation that uses all the oracles. This is similar to the problem which involves balancing flow during a set of disjoint time intervals. There are some constraints applied by LP which may cause the messages to split.

5 Framework for Performance Comparison:

They've developed a custom simulator for DTN environments. It is a discrete event simulator written in Java. The two key components of the simulator are the nodes and the links, which can be created and destroyed dynamically (temporarily or permanently). Nodes have finite storage capacity(analogous to buffers). Links are attached to nodes and are directional by default and have finite propagation delay and finite bandwidth. It is different from normal simulators as it can distinguish between Complete link failure and Link close at the source. It supports reactive fragmentation which is necessary for simulating the LP algorithm above.

6 Experimental Results:

Overall, smarter algorithms which required more knowledge performed much better than the simpler algorithms, both in terms of delay and delivery ratio. This difference is more prominent when the network is more disconnected, and hence verifying the need of smarter routing techniques in sparse networks as in the motivation. Also, as the load is increased, EDLQ and EDAQ perform better than ED as it doesn't take into account the queueing factor. The interesting thing is that EDLQ performs comparably to EDAQ and hence implementing the queueing oracle isn't really required. Also, sometimes LP's maximum delay sometimes exceed that of EDAQ because LP optimizes for the average delay. Hence, global knowledge may not be required for good performance in all cases.

7 Conclusions and Remarks:

The paper was like an experience study in which they took some existing algorithms on routing in delay tolerant networks and did an apples-to-apples comparison on a common framework and assessed the techniques on the basis of delay/performance and delivery ratio. Hence, these experiments are replicable and would make sense anytime. But, as now the network connectivity has increased and networks have reached to almost all parts of the globe and are have considerably many more users, so the difference in the performances may not be as much as it was then and hence some more factors will need to be included to compare these algorithms effectively. Also, in real-world situations, the realisation of the knowledge oracles may sometimes prove to be very difficult and nowhere have they discussed anything about how to handle them. The networks may also not be entirely predictable as they have assumed. In many networks, entities are mobile and hence a dynamic topology is needed and hence the algorithms and the framework would have to be modified in such cases. Most importantly, now as the computation power has increased greatly, hence the LP simulation can be applied for much larger networks too to solve and verify their experiment.