The Evolution of a DTN Routing Protocol – PRoPHETv2

Samo Grasic
Luleå Univ. of Tech.
[samo.grasic@ltu.se]

Elwyn Davies
Folly Consulting
[elwynd@folly.org.uk]

Anders Lindgren
SICS
[andersl@sics.se]

Avri Doria
Independent Consultant
[avri@acm.org]

ABSTRACT

Research within Delay- and Disruption Tolerant Networks (DTN) has evolved into a mature research area. PRoPHET is a routing protocol for DTNs that was developed when DTN research was in its infancy and which has been studied by many. In this paper we investigate how the protocol can evolve to meet new challenges that has been identified through research and practical experience.

We propose some minor modifications to the routing metric calculations done in PRoPHET which has potential to alleviate some issues and improve the performance of the protocol. Using these modifications, we define an updated version of the protocol called PRoPHETv2. We run simulations to verify the operation of the protocol and compare its performance against the original version of the protocol as well as some other routing protocols. The evaluations are done using both traces from an existing DTN deployment and a synthetic mobility model. Since the basic mechanisms of the protocol remain the same, migrating existing implementations to the new version of PRoPHET is possible with limited effort.

Categories and Subject Descriptors

C.2.2 Network Protocols [**Routing protocols**]:

General Terms

Algorithms, Design

Keywords

DTN, routing

1. INTRODUCTION

Over the past decade, DTN research has gone from being a small niche research area and evolved into a mainstream research topic within networking. The area is now more mature than in its early days, and many lessons have been learned over the years.

In order to exploit available scheduled and opportunistic ad-hoc connectivity in the deployed area, an efficient routing scheme is needed. Much of the focus of DTN research has been in routing protocols, with a multitude of protocols presented in research papers. The PRoPHET routing protocol [6] was one of the first routing protocols to be defined in the early days of DTN research. It is also the only DTN routing protocol that has a well-defined specification of the protocol details in an IRTF Internet Draft [5]. Therefore, it has often been used as a benchmark and in comparison to evaluation of other DTN routing protocols [2], as well as in some real-world tests and deployments[1]. PRoPHET uses the history of previous encounters with other nodes as well as the transitive properties of the network to optimize the bundle forwarding over the network by pruning the epidemic distribution tree.

Based on evaluations of PRoPHET and our own experience with the protocol in deployments and simulations, we realize the need for the protocol to evolve to meet new challenges and improve its performance. In this paper, we look at where previous research has shown room for improvement, discuss some situations where this may have been due to poor experimental setups, but also find situations where minor modifications to PRoPHET can address the concerns raised and improve performance. We then describe a proposed update to the protocol as PRoPHETv2 and evaluate the updated protocol through simulations and compare it against the original PRoPHET and some other routing protocols.

2. BACKGROUND

The PRoPHET routing protocol [5] uses the history of previous encounters and the transitive property inherent in human contacts to estimate a *delivery predictability* (referred to as DP for the rest of the paper), $P(A, B)$, at each node A for all known destinations B, kept in a *DP table*. As nodes meet, they exchange DP tables and then update their own DP table based on the identity of the other node and the received table according to the following equations. For the encountered node B, node A increases its DP using 1 where P_{enc} is a configurable parameter. This is used so that nodes that are frequently encountered will get a high DP (proposed value in [5] is 0.75).

$$P(A, B) = P(A, B)_{old} + (1 - P(A, B)_{old}) \times P_{enc} \quad (1)$$

The DPs for all other destinations i known by the encountering nodes are also updated based on the values in the table sent to node A from node B due to the transitive property of the DP (if node A frequently meets node B and node B frequently meets node i, then node A is a good carrier for messages destined for node i even if it doesn't meet i itself as it can send it through B). This transitive update is done using 2 where β is a configurable parameter (proposed value in [5] is 0.25):

$$P(A, i) = P(A, i)old + (1 - P(A, i)old)P(B, i)P(A, B) \quad (2)$$

CHANTS'11, September 23, 2011, Las Vegas, Nevada, USA.
Copyright 2011 ACM 978-1-4503-0870-0/11/09 ...$10.00.

In order to eliminate stale information from the network, the DP table is periodically aged according to 3 for all destinations i, where γ is a constant (proposed value in [5] is 0.998) and T is the number of time units since the last time the aging process was run.

$$P(A,i) = P(A,i)_{old} \times \gamma^{T} \tag{3}$$

In the rest of this section, we will analyze some situations where places for improvements in PRoPHET have been found to find the reason for the observed behavior and consider what can be done to improve the current state.

2.1 Experiences from N4C Deployments

The N4C project deployed real DTN systems in remote areas of the Swedish mountains. One component of the N4C system tested during annual summer deployments, was an implementation of PRoPHET. After the first week of the summer test in 2009, it was noticed that one data mules refused to forward bundles destined for the Internet gateway to the fixed node at the helicopter base. Investigations showed that the DP of the data mule for the gateway node was actually higher than the DP of the node in the helicopter base, despite the fact that the data mule was never in direct contact with the gateway. In the process of checking the DPs of other nodes, we found that they were all very high and that they did not correspond to the actual network topology. A reason for this unpredicted protocol behavior was found in the transitive update equation 2 of PRoPHET. When the frequency of encounters is not evenly spread over the network, and encounters are frequent enough that the aging mechanism does not reduce the DP rapidly enough, such problems can occur. In our case, two regions were formed, one where the encountering frequency of the nodes (in the village) was high, and the other where the encountering frequency was low (data mules, border nodes and gateway).

2.2 Parking Lot Problem

In its initial form, there were instances when PRoPHET would produce inappropriate evolution of its DPs if a group of nodes came together and repeatedly exchanged their sets of DPs. This was named the *Parking Lot Problem* as the situation that was discussed involved a number of users with DTN devices congregating in the supermarket car park or âĂŸparking lotâĂŹ. On analyzing the data from N4C deployments it became clear that even without clusters of devices, we saw many encounters separated by only a short time while human patterns would be expected have time scales of hours or days. Many of these encounters are in fact reconnections due to flaky WiFi conditions, and counting each of them as a new encounter between nodes overestimates the number of “real” encounters. If PRoPHET treats these reconnections as new encounters, the DPs for the encountering nodes will increase too much and the the mobility pattern model will be distorted. This is very similar to the theoretically postulated parking lot problem.

2.3 A Chain Topology Example

To showcase a problem that has been noticed in the evolution of DPs over time in certain settings, we ran some simulations in the ONE simulator. The topology shown in Figure 1, was used to analyze and illustrate an observation regarding the transitive update equation in PRoPHET. This simple network consists of four fixed nodes A, B, C and D that form a chain topology. The links between the nodes are only up for short periods of time, and only one link at a time is up in the network.

Nodes A and B, representing a network region with low encounter frequency, will connect only once at the beginning of the simulation. After this, the links $B - C$ and $C - D$ will periodically go up and down, representing a network region with a higher frequency of node encounters. Using this simple scenario, we investigate how the information about the initial connection between nodes A and B spread through the network and how it evolves in the high encounter frequency region of the network. In order to do this, we focused only on the DPs to node A of nodes B, C and D.

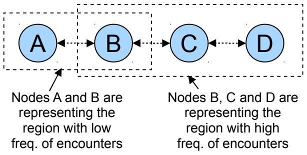
A
B
C
D
Nodes A and B are representing the region with low freq. of encounters
Nodes B, C and D are representing the region with high freq. of encounters

Figure 1: Chain topology.

As shown in Figure 2, shortly after the initial encounter between A and B, $P(B,A)$ keeps growing even though there was only a single encounter between the nodes. After the 22nd iteration, $P(C,A)$ even becomes larger than $P(B,A)$ even though nodes C and A have never met. As a result, any data flow for node A would not be forwarded from node C to B according to the PRoPHET forwarding policies. This is due to the additive property of the transitive equation 2 in PRoPHET in combination with the fact that only a single number is used to represent the DP, making it impossible to determine where a certain piece of information has been received from (this was a conscious design choice in order to reduce complexity and the amount of state stored). Therefore, when B and C meet and information that C has earlier received from B is then passed on back to B and vice versa, their DPs for A are further reinforced even though no additional encounters have occurred.

When the information about the initial encounter between nodes A and B reach the high encounter region, it gets amplified in an inappropriatly. This problem is less critical when the encounter frequency is more even across the network and can be solved partly through a more aggressive aging policy.

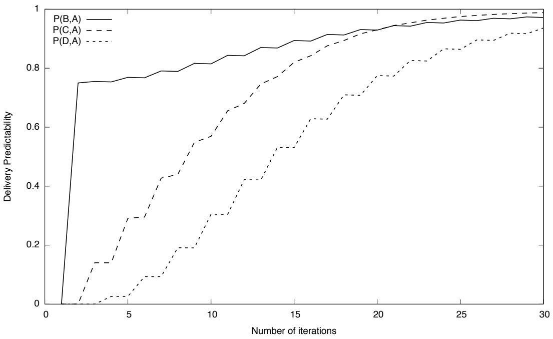

Figure 2: DP evolution over time using legacy PRoPHET

2.4 PRoPHET Evaluations in the Literature

In this section, we look at some work by other that have shown poor performance for PRoPHET and try to understand these results. At Helsinki University of Technology [4, 3], PRoPHET was compared to other protocols using various synthetic mobility scenarios. KerÃd’nen et al. [4] found PRoPHET to be sub-optimal for Random Waypoint (RWP) types of scenarios, partly because RWP mobility does not provide predictable mobility patterns that PRoPHET can leverage. When more realistic mobility was used, PRoPHET outperformed Epidemic Routing. It is noteworthy that in the more realistic scenarios, the delivery rate of PRoPHET and Spray-And-Wait were close to a single-hop direct delivery method. This indicates that the mobility is such that most nodes will eventually meet a given destination, thus not ideal for showcasing the strengths a multi-hop protocol. An interesting observation in both this work and in [3] is that PRoPHET had its best performance when β was set to zero. We will consider this later on in the paper.

In evaluations of the RAPID routing protocol [2], PRoPHET was found to perform worse than other protocols evaluated. As the evaluation was done using real-world traces and not random movement, we wanted to review the PRoPHET behaviour in this case further. The authors of [2] generously shared their traces and source code, facilitating this process. A couple of mistakes in the implementation of PRoPHET in the simulator was found (use of γ instead of β in transitive update equation, and a problem with the way routing information is exchanged). In addition, all the routing information and the traffic that was gathered was cleared out of the network at the end of each simulated day. This is inappropriate for a protocol like PRoPHET that makes estimations based on a long term history of encounters. As a result, PRoPHET's DP values could not converge and the protocol was kept in the learning phase throughout.

2.5 The Need for Transitivity

As noted above, several reports claim that the best results of PRoPHET was achieve when β is set to zero, effectively disabling the transitive properties. Thus, one might argue that the inclusion of a transitive property in the calculation of the routing metric is not beneficial and should thus not be used. So, should a new version of PRoPHET disregard transitivity? We believe the answer to this to be a resounding no. First of all, many of the results that have shown negative results for the transitivity are likely to be due to the issues discussed in previous sections, which will be addressed in the proposal in the following section. Further, we believe that this transitive property is vital for good operation of the protocol in certain settings where the network might otherwise become partitioned so that some nodes are not able to communicate. This will happen when the network consists of multiple communities that are mostly disjoint, with only a few nodes moving between the edges of the communities. A typical example is the network previously discussed and shown in Figure 1. If only direct contacts are used to build the routing metric in such a network, node C will never disseminate any knowledge about node A, and node D will thus never be able to know that there is a potential path towards A through C. A correctly functioning transitive protocol will create a gradient through the network towards the destination.

3. PROPOSED IMPROVEMENTS

Based on the issues discussed in the previous section, we will now describe some improvements to the mechanisms of PRoPHET, and finally propose a new version of the protocol, PRoPHETv2.

Having identified the parking lot problem, and the issues that can arise if the protocol believes that a lot of encounters are happening due to intermittent wireless connections that frequently go up and down, a new formula was developed for the evolution that increases the DP of a node that is encountered. This is shown in 4. The mechanism used in the formula solves this issue and avoids this distortion without storing large amounts of extra information for each node. This is done by taking into account the time since an information exchange was last done with this node and reducing the amount that the DP is increased for nodes that were recently encountered.

$$P(A,B) = P(A,B)_{old} + (1 - P(A,B)_{old}) \times P_{enc} \quad (4)$$

Where P_{enc} is calculated in the following way:

$$P_{enc} = P_{max} \times (Intvl_B / I_{typ}); if 0 \leq Intvl_B \leq I_{typ}$$
$$P_{enc} = P_{max}; otherwise$$

Where I_{typ} is a parameter that is set to the expected typical time interval between connections in the application scenario relevant to the network deployment. $Intvl_B$ is the time since the last encounter with node B.

The problem with the original transitive update equation 2 is that as long as $\beta > 0$, the DP for every known node k will increase regardless of whether any node in the network has recently met node k or not. Although these transitive increments are usually small, they will inappropriately raise the DP values in the case of a higher encounter frequency (for nodes other than k). When this spurious information spreads through a high encounter frequency part of the network, the problems that occurred in the N4C summer test in 2009 can be expected.

As a solution to this problem we propose a new transitive update equation 5 that does not have the same additive property as the original equation. Instead, we compare the old DP value $P(A,i)_{old}$ and a product of $P(B,i) \times P(A,B) \times \beta$ values (where we suggest the default β value to be set to 0.9) and then choose the maximum value as a new DP value $P(A,i)$.

$$P(A,i) = max(P(A,i)_{old}, P(B,i)P(A,B) \times \beta) \quad (5)$$

By taking the maximum we allow for the evolution of the DP value without subjecting it to exaggerated growth when no new information is available.

3.1 PRoPHETv2

Using the improvements to the DP evolution mechanisms outlined above, we have defined a new version of the PRoPHET routing protocol, PRoPHETv2. The new version of the routing protocol maintains the original ideas and design of the long standing PRoPHET protocol, with only these minor modifications to the evolution calculations. It should be possible to upgrade operational code for the original PRoPHET to match the PRoPHETv2 specification with only small changes to the code . The current detailed specification of PRoPHETv2 is available as an Internet Draft at [5]. In the following section, we will evaluate this new version of the protocol to show the benefits over the previous version.

4. EVALUATION

In order to evaluate the performance of PRoPHETv2, we implemented it in the ONE simulator. Initially, we wanted to investigate how the new transitivity update equations affected the development of the DPs over time and space. We used the same set up of four nodes in the chain topology as seen in Figure 1 and the same connectivity model described in Section 2.3 and compare the results to those shown in the study in that section. In Figure 3, we can see that using the new transitive update equation defined in 5 with new default parameters P_{enc} and β, the DPs of node C to A does not overgrow the DP of node B (as happened with legacy PRoPHET). Instead, the nodes reach a steady state where the desired âĂIJgradientâĂİ through the network is present so that nodes closer to the destination have a higher DP. Thus, messages destined for node A would flow in the correct direction in the network.

After confirming that the new equation for the transitive update of DPs does indeed provide an improvement upon their spread through the network and creates a valid gradient towards the destination, we also wanted to see what impact this has on the performance of the protocol in terms of message delivery.

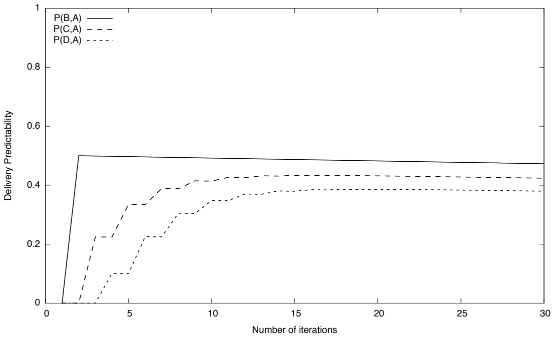

Figure 3: DPs evolution over time using PRoPHETv2.

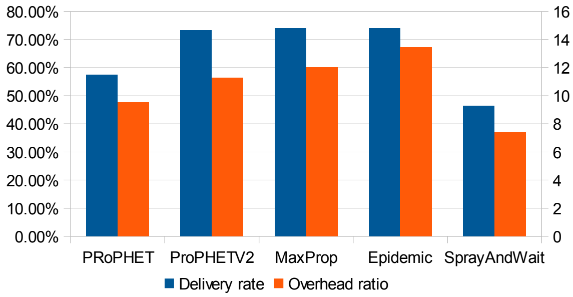

Figure 4: Results from simulations using N4C traces.

To ensure that PRoPHETv2 would work under conditions similar to the ones in which the problems with the old transitive update equations were found, we used real-world connectivity and traffic traces, collected during a N4C deployment in 2010. The traces come from one week when the mobility conditions resembled the ones in the N4C summer test of 2009. During that period (July 27-August 3, 2010) only one data mule carrier was used and the network had high usage. The total number of DTN nodes was 18 and the number of bundles sent was 1407. The lifetime of these bundles was set to three days. To show that the old transitive update equation 2 in certain condition can cause low delivery rate regardless of lack of DTN resources (storage and bandwidth) were met or not, unlimited bandwidth and storage capacity were set in the simulator. Thus, the delivery rate from the Epidemic routing can be read as an optimal solution in this scenario above.

Only one helicopter flight with data mules was scheduled per day and some of the bundles that were sent in the last days had to be dropped at the end of simulations, so the delivery rate of Epidemic routing was 74%. As seen in Figure 4, legacy PRoPHET delivered only 57% of the traffic despite unlimited network resources, while PRoPHETv2 could deliver the same number of bundles as Epidemic Routing with a lower overhead. MaxProp achieved similar delivery rate as PRoPHETv2 and Epidemic with a slightly higher overhead than PRoPHETv2, and Spray-and-Wait had the lowest delivery rate (but also the lowest overhead).

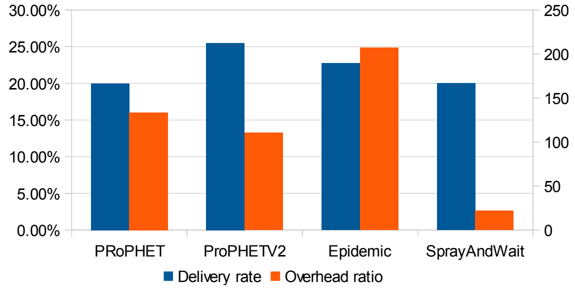

Figure 5: Results from simulations using WDM mobility.

In the second evaluation we used the Working Day Model (WDM) [4] with the default settings for WDM in the ONE simulator. Due to a likely memory leak in the ONE implementation of MaxProp, the simulations for MaxProp never terminated, so MaxProp results are not included for this scenario. The simulation time was set to one week and the expiry time for the bundles was set to 1430 minutes. The number of nodes was 500, and a total of 17000 bundles were sent. In this set up, the buffer space was limited to 100MB per node and the available bandwidth was 100 kbit/s.

As seen in Figure 5, despite limited resources (buffer and bandwidth) PRoPHETv2 performed better than Epidemic Routing and legacy PRoPHET in this scenario. Likewise, the delivery rate of PRoPHETv2 was significantly higher than for Spray-and-Wait, but also with higher overhead. The fact that PRoPHETv2 outperformed Epidemic Routing shows that the protocol makes wise decisions on what bundles to forward and how to use the limited resources.

5. CONCLUSIONS

Drawing on our experience with the PRoPHET routing protocol, as well as results shown by others, we analyzed these studies and defined a new transitive update equation 5 and direct encounter update equation 4. This updated functionality was used to define PRoPHETv2. Simulation evaluations show that PRoPHETv2 perform better than the old PRoPHET routing protocol, especially in cases of heterogeneous network mobility scenarios. PRoPHETv2 also achieves similar or better performance to MaxProp and Spray-and-Wait, but has the added advantage of having a well-documented and detailed specification that can be used to create independent interoperable implementations of the protocol.

Our aim with new PRoPHETv2 routing protocol has been to make an improvement of the protocol with only minor changes in specification facilitating the continued usefulness of current implementations of PRoPHET routing protocol.

6. REFERENCES

[1] Networking for communications challenged communities (N4C) project website, http://www.n4c.eu.

[2] A. Balasubramanian, B. N. Levine, and A. Venkataramani. DTN Routing as a Resource Allocation Problem. In *ACM SIGCOMM*, August 2007.

[3] J. Karvo and J. Ott. Time scales and delay-tolerant routing protocols. In *CHANTS 2008*, September 2008.

[4] A. Keränen and J. Ott. Increasing reality for dtn protocol simulations. Technical report, Helsinki University of Technology, July 2007.

[5] A. Lindgren, A. Doria, E. Davies, and S. Grasic. Probabilistic routing protocol for intermittently connected networks. Internet Draft draft-lindgren-dtnrg-prophet-09.txt, April 2011.

[6] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. In *SAPIR 2004*.