Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für
Technische Informatik und
Kommunikationsnetze

SimBet

Semester thesis on SimBet

Leo den Hartog

Semester Thesis SA-2008-06
March 2008 until May 2008
Advisors: Dr. Thrasyvoulos Spyropoulos
Supervisor: Prof. Dr. Bernhard Plattner

Abstract

This thesis is about investigating the benefits of adding social metrics to DTN routing on the example of a well-known social-based protocol called SimBet. We show that SimBet performs very well in synthetic scenarios, which differ much from the real-traces used in the proposed implementation. By designing a model, we were able to test SimBet on several different scenarios, in order to perform further research on SimBet's performance on specific situations. After running several simulations, we could pinpoint to three weaknesses in the implementation, which we found to be not optimally solved. On those weaknesses, a sensitivity study has been done to see the effects of the individual parameters in SimBet. Finally we show some approaches to solving them and run preliminary test, to try and find directions for future research.

Contents

Chapter 1

Introduction

Routing in Delay Tolerant Networks has been intensively researched. The concept of routing based on social properties such as closeness of nodes though is relatively new. DTN networks are created in real environments and therefore behave a lot like social networks, which have been well studied and many metrics and community algorithms exist. Nonetheless, routing protocols today are mostly examined on a Random Waypoint Mobility model, which neglects knowledge about future node movement. Many real world traces, like for example the haggle traces, are created by giving nodes to people at conferences or even larger scales and examine their interaction [1]. Because most scenarios, where DTN networks are applied on human movement and interaction, there is an obvious benefit from using that additional information. Initial studies [5] have shown a significant beneficial effect of adding social properties into DTN Routing protocols. There exist a few routing protocols that make their decisions based on social properties for example Prophet [2] and SimBet [6].

The Prophet algorithm makes routing decisions based on delivery probability. Its calculations are based on a time of last encounter vector, which is updated on every node contact. Nodes not only update their contact history, they also mutually exchange it in order to have some transitive information. The idea behind Prophet is, that in a social network people/nodes that met only a short while ago have a higher probability of meeting again sooner than two nodes that haven't met for a long time. This is a reasonable assumption and has been shown in several papers on social networks [5]. Using this approach Prophet is indirectly a social protocol, albeit not relying on any explicit social metrics. The main advantage is that it has proven to be able to deliver beneficial effects without the cost of neither more calculation time nor much additional traffic in the single-copy version. The original, flooding-based version is problematic in that aspect.

By neglecting social metrics, Prophet is not taking full advantage of the social information it collects, and is therefore not able to make real "smart" decisions. SimBet is the first routing protocol to directly implement social metrics. It uses two different metrics, similarity and betweenness, operating on the same information as Prophet, on which it bases its routing decisions. Similarity is a measurement of how socially close two nodes are, by comparing mutual contacts. Betweenness calculates the interconnection of a node, providing an estimation on how important a node is for inter-group routing. By applying those metrics and using both for forwarding decisions, SimBet tries to achieve two goals; the betweenness value is used first for optimal inter-group routing, and then to find a destination group, the similarity metric is used to find the destination using intra-group routing.

Our approach is first the implementation of SimBet in an open-source simulator and comparing it against other protocols like Prophet and Spray and Wait [3] in a number of synthetic "social" scenarios. We created a modification of the Random Waypoint Model to implement a social factor, which gives us the possibility to test SimBet on scenarios different from the real-traces, on which it has already been tested. With this we want to further prove, that adding social metrics into a protocol has beneficial effects on routing in different scenarios and needs to be looked into more deeply. With the simulator, we are able to adjust the mobility model to our needs concerning sociality, rather than use only one instance of a real environment. We are also concerned with the precision of simulations using real-traces, due to an effect called path-explosion [8], which shows on the same traces that different routing protocols perform very similar. Secondly we wanted to make a sensitivity analysis on the several parameters SimBet uses, because no clear justification is provided in the paper for the various parameter choices. We believe that some of those values might not be optimal. To explore how adaptive SimBet is for different scenarios, we ran several scenarios with different sets of values for the parameters of interest and examine the impact on the delivery probability. Out of the experience gained there, we wanted to locate potential weaknesses and propose possible improvements for fixing them.

To summarize, the contributions of this work are the following; first, we were able to clearly show in a range of simulation scenarios that a DTN routing protocol greatly profits from the use of social network based metrics. SimBet outperforms Prophet in every scenario, does very well against 2-copy Spray and Wait and achieves only a slightly smaller delivery probability than Epidemic [4] with a much smaller overhead. At the same time, our results indicate that using redundant copies, even in this social context, can have a great beneficial effect to routing, Secondly we could pinpoint to three potential weaknesses in the SimBet protocol, we think are not addressed well in the original protocol. We tried to separate the weak points and run individual sensitivity studies on the specific parameters. Based on the sensitivity analysis, we proposed some approaches on how the weaknesses could be fixed and ran some preliminary simulations on our proposals. Still, further research has to be done on improving the weaknesses. Our approaches can be seen as initial directions, which would have to be tested more extensively on different types of simulations and real-traces.

The remaining of the paper is structured as follows. In the next section, a new mobility model is introduced, which is based on Random waypoint with an added "social" pattern of the individual nodes to model the effect of social routing. Section 4 explains in detail how SimBet works. In section 5, SimBet is compared to other DTN Routing Protocols like "Prophet" and "Spray and Wait" in different scenario settings, using our mobility model. Based on these comparisons we identify some potential weaknesses in section 6. For each weakness a sensitivity study is performed and approaches on improving them are given. Section 7 analyzes our results and concludes the paper.

Chapter 2

Mobility Model

The mobility model for all the scenarios is based on Random Waypoint, enhanced with a social component, which is shown to capture trace behavior well [7]. We used a synthetic movement model because we wanted to have full flexibility regarding the association of the nodes. The SimBet paper has shown advantages for real-traces, but we wanted to control the setting in order to test performance on very different kind of scenarios. As a recent work [8] suggests, differences in performance on the traces the SimBet paper uses, can be less pronounced due to a phenomenon called "Path Explosion". Our mobility model also gives us the ability to test some individual properties of the protocol in more detail, because we are able to generate the scenario accordingly.

In the mobility model we use, nodes are split into groups; every group has an assigned range for speed, pause-time and "sociality". Pause time is chosen randomly (out of a distribution) every time a destination is reached, while speed and sociality are assigned once on start of the simulation randomly out of their respective value range. Therefore different movement patterns exist inside a group, which corresponds to different movement patterns in real life scenarios. Sociality has been realized by the creation of communities, where every group of nodes is assigned two communities. With a certain probability μ , called sociality, a node chooses its next destination randomly inside its communities as opposed to anywhere in the scenario. Communities are rated equally, so there is no favorable "home-community". By this, we were able to create a DTN Network with an underlying mobility model that gives reasonable ground for socially inspired routing.

For the simulations in part 4, we used three different mobility scenarios, which roughly represent three different types of envisioned social network interaction and respective mobility patterns. Rectangles represent communities, edges represent groups (sociality is the probability, a node chooses a new destination inside one of its communities. So with sociality smaller than one, every node of a group has the whole scenario as a potential target, edges only represent the two communities the group is assigned to).

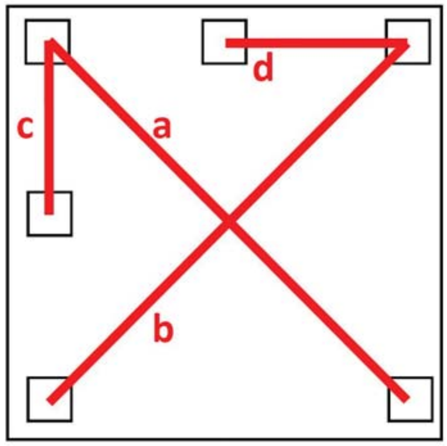

Figure 2.1: Scenario 1 (1000x1000) - Group-Community affiliation

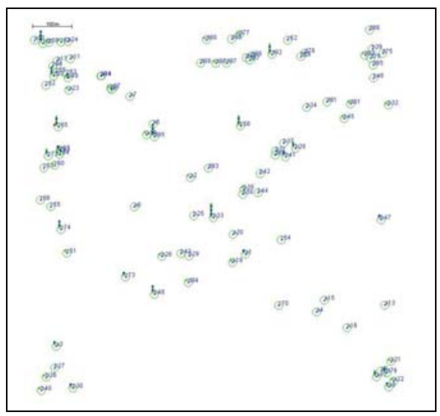

Figure 2.2: Scenario 1 (1000x1000) - Runtime screenshot

Scenario 1 is designed to investigate the overall performance of SimBet in a generic setting, because it includes strong familiarity within the small communities as well as a crossing point between unaffiliated groups.

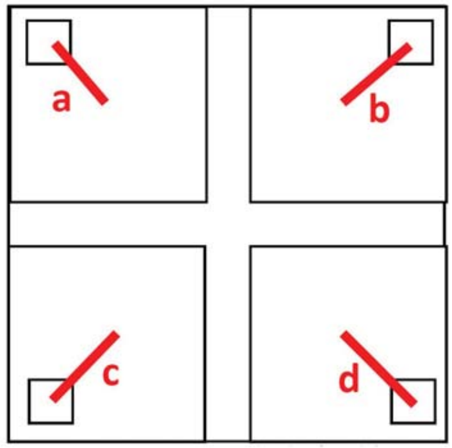

Figure 2.3: Scenario 2 (1000x1000) - Group-Community affiliation

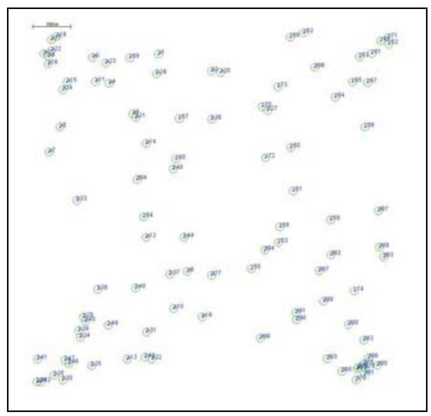

Figure 2.4: Scenario 2 (1000x1000) - Runtime screenshot

Scenario 2 is created as a very social setting. Nodes of different groups meet only outside their communities. This is a scenario where the similarity metric should be very important.

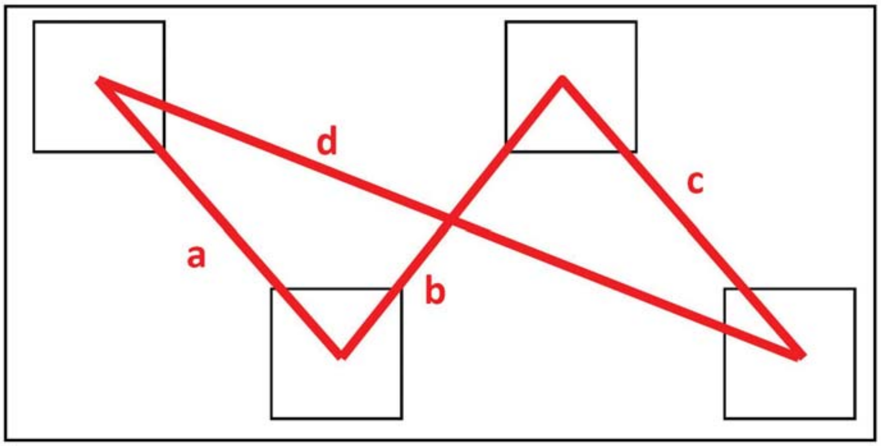

Figure 2.5: Scenario 3 (2000x1000) - Group-Community affiliation

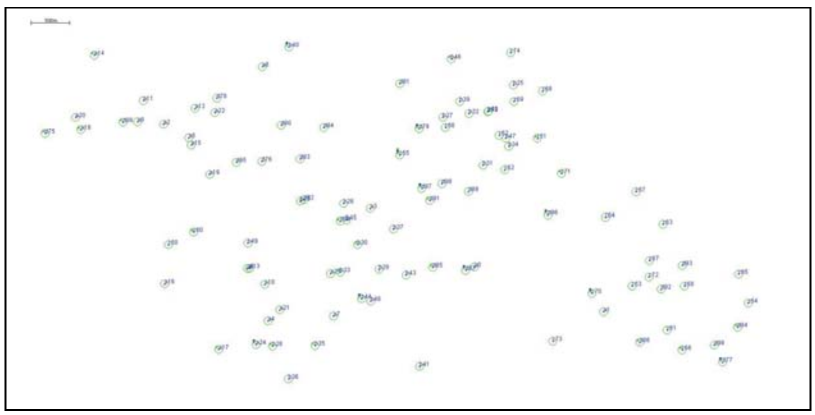

Figure 2.6: Scenario 3 (2000x1000) - Runtime screenshot

Scenario 3 is sparser than the others and the communities are bigger. Communities are further away from each other and travelling distances increase. This is a test on SimBet's betweenness calculation.

Chapter 3

The SimBet protocol

SimBet is a DTN routing protocol. Each node keeps a contact history vector consisting of the time of last encounter. On every contact, a contact history matrix is mutually updated with the contact history vector of the other node. Based on that individual matrix, SimBet nodes calculate two social network based metrics, similarity and betweenness.

Similarity is individually calculated by adding up the number of common nodes the node and the destination have met in the past, which makes a reasonable estimation of how socially connected the two nodes are.

$$Sim_t(d) = \mid N_t \cap N_d \mid \; ; \; with\ N_x = Contact\ history\ vector\ of\ node\ x \tag{3.1}$$

Betweenness is a measurement for how interconnected a node is; its calculation is based on the number of nodes the specific node has met, that haven not met each other.

$$Bet_t = \sum A^{'}_{i,j} \; ; \; with\ A^{'} = A^2 * (1 - A)\ and\ A = Contact\ history\ matrix \tag{3.2}$$

For the comparison, both values are averaged, and weighed by two parameters α and β.

$$SimUtil_t(d) = \frac{Sim_t(d)}{Sim_t(d) + Sim_o(d)} \; ; \; with\ Sim_x(y) = Similarity\ of\ node\ x\ to\ node\ y\ and\ t = This\ node\ ; \ o = Other\ node \tag{3.3}$$

$$BetUtil_t = \frac{Bet_t(d)}{Bet_t(d) + Bet_o(d)} \; ; \; with\ Bet_x = Betweenness\ of\ node\ x \tag{3.4}$$

$$SimBetUtil_t(d) = \alpha * SimUtil_t(d) + \beta * BetUtil_t \;\; with\ \alpha = 0.5\ and\ \beta = 0.5 \tag{3.5}$$

Forwarding decisions are made based on the comparison of the two SimBetUtil values

$$If(SimBetUtil_t(d) < SimBetUtil_o(d))\ Then(forward\ message\ to\ node\ o) \tag{3.6}$$

Chapter 4

Comparison of social and non-social DTN protocols

All the comparisons are made running the One Simulator [9]. This simulator was chosen, because it is open-source, therefore assumed to be bug-fixed and tested, and because it has been developed purposely to evaluate DTN scenarios better. The developers included the "Epidemic", "Spray And Wait" and Prophet Router. The SimBet Router has been implemented as stated in the Paper [6]. For all comparisons, buffer size and bandwidth are set to infinite, so no messages are lost due to physical limitations. Therefore we use Epidemic Routing as an upper bound for performance with respect to delivery probability, because in these conditions Epidemic Routing always finds the optimal path. Message creation is based on a randomly generated event list consisting of 550 sender-receiver pairs, in order to be able to compare the protocols on the exact same conditions, but having a statistically significant enough amount of messages to average the results. This ensures that the different protocols are compared in an appropriate way. For all scenarios, settings are set like following:

- Transmission range: 10
- Number of Nodes: 100
- Pause time between targets: uniform random number between 100-200
- Scenario Time: 25'000
- Sociality: uniform random number between 0.8-1

The parameter for speed varies between the three scenarios as indicated. Speed range has been adapted to size and density of the scenario, so that Epidemic is able to perform reasonably in terms of delivery probability in order to make sure that the number of possible paths does not bias the comparisons.

Spray and Wait runs in two-copy mode. This is a relatively small amount of copies for Spray and Wait, but because Prophet and SimBet are single-copy, we chose a very small number to ensure Spray and Wait differs enough from Epidemic in a scenario with a relatively small number of nodes. Similar to Epidemic, one copy could be seen as a lower border for a social routing protocol, because it is just waiting until sender and destination meet.

For the various graphs, we compared delivery probability and overhead ratio (the number of messages transmitted per successful delivery) of the four protocols for different TTLs. (Time to Live)

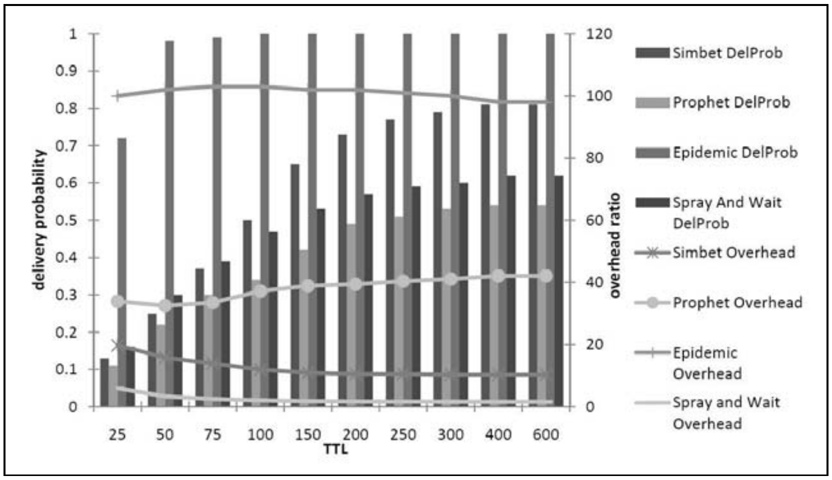

Figure 4.1: Scenario 1 (Speed: 0.5-2)

Scenario 1 was designed, to represent a social scenario with a neutral meeting point (crossing in the middle) to stress both similarity and betweenness, which clearly worked, because SimBet performs significantly better than Prophet and Spray and Wait. Compared to Epidemic, SimBet achieved about 80% delivery ratio at an overhead ratio of ten times less. Against Prophet, delivery probability was about 40% higher at an overhead ratio of four times less. Interestingly, for TTL smaller than 100 Spray and Wait performs better than SimBet. We explain this with the fact, that multi-copy protocols have the bigger benefit than social protocols due to more opportunities. If only little time is available, social factors are less important than the gain of multiple copies.

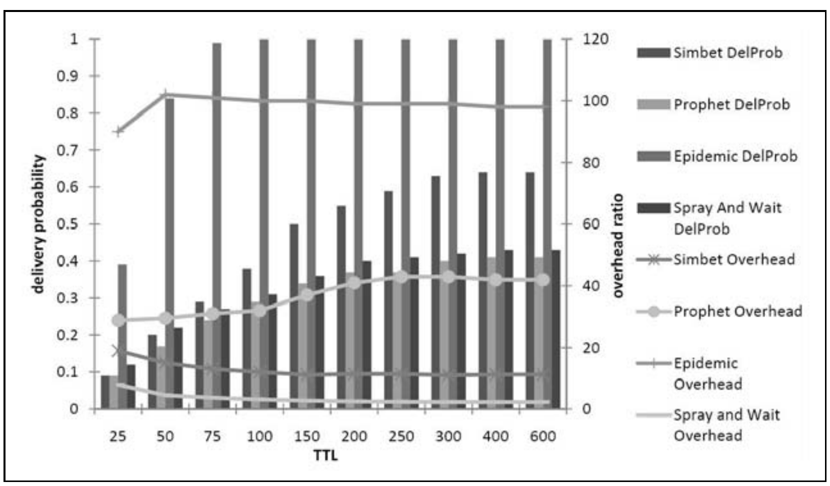

Figure 4.2: Scenario 2 (Speed: 0.5-1.5)

The second scenario should represent a very familiar environment, where nodes seldom meet nodes of another group. Relatively to the other scenarios, Prophet performs worse here, which makes sense, because Prophet only takes last encounters into consideration, which is rather unsuited for inter-community routing based on random waypoint (because groups don't have overlapping communities). SimBet performs better, mainly because it combines the element of finding the way out of a community (by betweenness) as well as routing the message back into a community (by similarity). Still it gets outperformed by Spray and Wait for very low TTLs (<75) which encourages the idea, that multi-copy is performing better if minimal time is available. The fact that SimBet starts performing better than Spray and Wait earlier than in scenario 1 is in our opinion due to the relatively small opportunities for inter-community routing that exist, which apparently SimBet is better at recognizing.

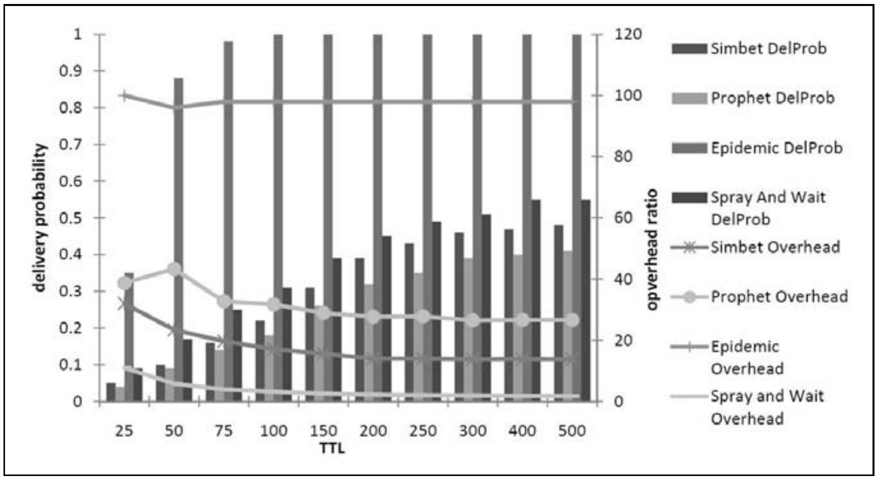

Figure 4.3: Scenario 3 (Speed: 0.5-3)

The third scenario shows, that SimBet still outperforms Prophet, but looses against Spray and Wait. Delivery probability for Spray and Wait is uniformly higher than for SimBet at a much lower overhead ratio. One can still detect the even bigger difference for lower TTLs, but for higher TTLs the results were rather surprising. In our opinion, this is because the communities were quite big (so similarity is less strong, because nodes of the same group don't necessarily meet, even if they are in the same community) and speed of the nodes was set quite high in order to still get a high contact rate. It makes another very strong argument for the use of a multi-copy version of any protocol.

The gathered data clearly shows that SimBet outperforms Prophet in every Scenario. Delivery probability is consistently higher at an overhead of about half. Especially the first two scenarios are very "social" therefore it is clear that SimBet has a much lower overhead, because of its smarter forwarding decisions. We conclude that routing based on social metrics has clear advantages, but especially in scenario 3, a multi-copy version of the SimBet protocol has to be considered.

Chapter 5

Identifying weaknesses of SimBet and proposing solutions

As we saw in the previous section, SimBet outperforms the other social protocol by a significant margin, but is still no match for Epidemic. This indicates that there can still be room for improvement in the published SimBet version, which we want to examine and integrate into the protocol:

- One potential issue might be the setting of the α and β parameters, which are arbitrary fixed at .5 and .5 in the proposed implementation of the scheme. In section 5.1. we perform a sensitivity study on different values of alpha/beta in order to check whether there is any improvement potential. On top of that, we try a dynamic approach of choosing alpha/beta in order to make the protocol more flexible.
- Another imprecise thing in the paper is the aging of the contact history values. We tried in 5.2. different values for aging and got very different results. In our opinion, some kind of aging has to be implemented into the protocol, because at some point every node would have seen every other node at least once and any benefit of a social protocol would be ruined. We did a study on a simple way of aging, but also experimented with other ways of aging, which involved slightly changing the metrics for an added benefit.
- Finally, we started to implement an idea from a different perspective and tried in 5.3. to implement a simple community calculation. All the relevant data is available at the nodes, so it seems logical to implement a community algorithm, in order to make SimBet more dynamic for specific scenarios.

5.1 Making alpha and beta flexible

5.1.1 Sensitivity study

First we ran scenarios 1 and 2 for several different ratios of alpha and beta. They are run with the same settings as in the comparisons, but with a fixed TTL of 250. That value was chosen, because the curve starts to converge at that value.

As expected, the optimal α and β ratio varies between different scenarios. In Scenario 1, there are two peaks, at 0.6 and at 4, with the trend line (4th polynomial)

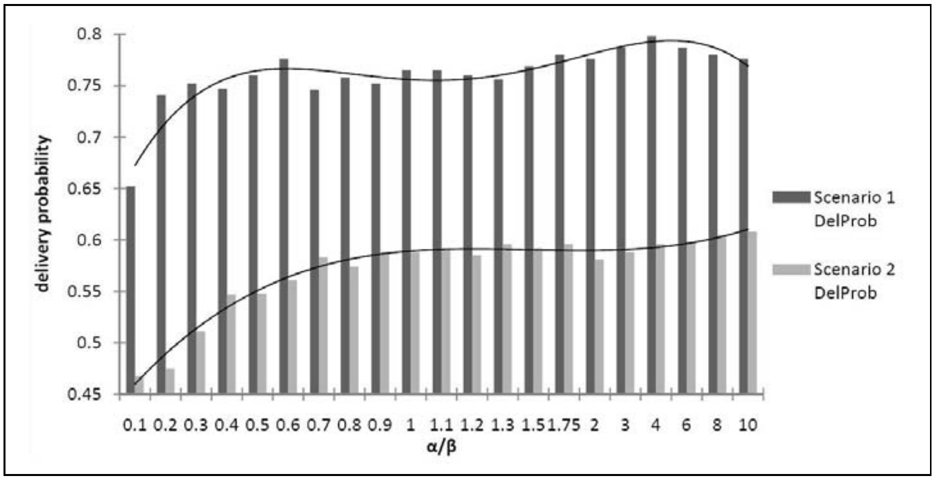

Figure 5.1: Sensitivity study

showing 4 as the optimal value, with slight improvement towards higher values of α over the traditional SimBet. This makes sense, because in a scenario with clear community sharing between different groups of nodes, similarity becomes more important than betweenness. We explain the first peak (where betweenness is stressed) with an improvement in routing because of increased choice of between nodes, which is more favorable in a quite sparse scenario with rather big differences in speed. The values for scenario 2 favor α even more, which was to be expected in a scenario where similarity is the key factor for routing. Even with more extensive testing on different scenarios, there wasn't a consistently better α and β. Even at a ratio of 0.1, which means routing almost entirely based on betweenness, SimBet still outperforms Prophet.

Generally, we believe that α should be weighed higher than β. From the proposed calculation, the values for SimUtil are close, whereas the values for BetUtil usually are far apart, because it is calculated in a different way. This means, that averaging for the two SimUtils works, but averaging the BetUtils concludes in a value of mostly 0 or 1. In order to weigh the utilities appropriately, α has therefore be weighed more, because it generally has smaller differences between the two nodes.

5.1.2 Proposed solution

An idea for improvement is to make α and β individual per node. By definition of a social setting, different nodes have different characteristics. According to another paper [8] one can categorize nodes depending on their behavior into the groups "IN","OUT" and "NORMAL". As metrics we used similarity and betweenness for the reason that they are already implemented into SimBet. For that we introduce two new calculations, a sort of similarity-rating and betweenness-rating.

Similarity-Rating: $R_s = \frac{V_s}{T_s}$ where V_s equals the number of similarity comparisons won ($\Leftrightarrow$ Similarity metric > 0.5) and T_s equals the total number of similarity comparisons.

Betweenness-Rating: $R_b = \frac{V_b}{T_b}$ where V_b equals the number of betweenness comparisons won ($\Leftrightarrow$ Betweenness metric > 0.5) and T_b equals the total number of betweenness comparisons.

Based on those two ratings, we first tried to group the nodes into three categories:

- IN ($R_s > \delta$ & $R_b > \delta$)
- Normal (($R_s > \delta$ & $R_b < \delta$) or ($R_s < \delta$ & $R_b > \delta$))
- OUT ($R_s < \delta$ & $R_b < \delta$)

According to their relative chance of forwarding a message, a node is either IN, Normal or OUT at the moment. We were hoping to find a significant set of values for improving SimBet. For each node, we dynamically varied the individual α and β according to which group a node belongs to at the time. We compared results to the original SimBet, which did not return remarkable results. We believe the main reason is that SimBet performs quite well for that set of nodes. OUT nodes will always forward the message, because they lose in similarity as well as betweenness, whilst IN nodes mostly keep the message because they "win" at both metrics. The only improvement could have been to make IN nodes more sensible for meeting someone of a specific community, but we already achieved that effect with a higher value of alpha.

The second idea was to group the nodes in a way that nodes which have very different R_s and R_b are separated.

- Social: ($R_s > \delta$ & $R_b < \delta$)
- Normal: All other nodes
- Travel: ($R_s < \delta$ & $R_b > \delta$)

We implemented the SimBet Calculation so, that nodes automatically compensate (or worsen) their weaknesses according to which group they belong, meaning that the ratio for a/ß for a social node equals the ratio of $\frac{\beta}{\alpha}$ for a node belonging to the travel group. The nodes belonging to the normal group had the standard settings of α=0.5 and β=0.5.

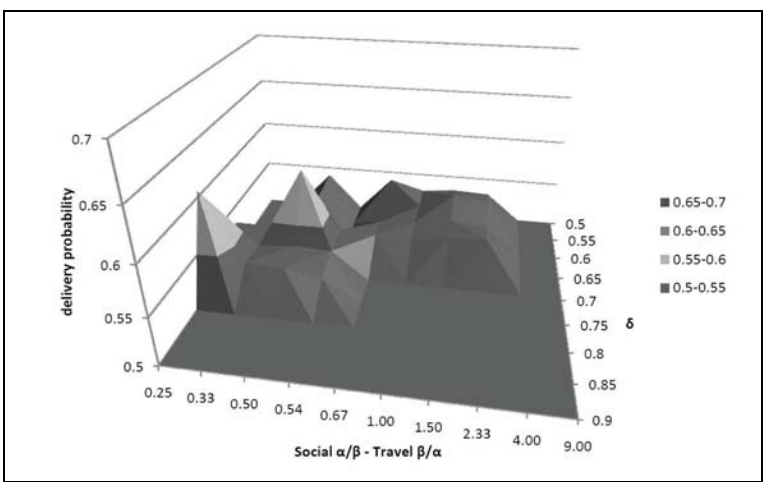

Figure 5.2: Scenario 1 (Speed: 0.5-3; Sociality: 0.5-1)

As is shown in the figure above, this delivered quite good results for a high threshold (only a few nodes are "Social" or "Travel") and a high $\frac{\beta}{\alpha}$ ratio (nodes that are "Social" or "Travel" differ significantly from "Normal"). This is understandable, because Social nodes need to push betweenness to get the message out of their community, and on the other hand do Travel nodes need to take similarity more into account, to make sure that the message gets forwarded into the destination's community. This benefits particularly the Travel nodes, because as stated in 6a, they have relatively high betweenness and similarity is underrated in the proposed solution, which causes inappropriate forwarding decisions if not taken into account.

5.2 Changing the aging

5.2.1 Sensitivity study

Some improvement potential clearly lies in the aging of the contact history matrix. The authors of SimBet don't age the values at all, but that is an important factor to the success of any social routing protocol. If one doesn't age the values or keeps too big a history without some kind of rating between different values, the whole benefit of sociality is lost. If a simulation runs for an arbitrary long time, most nodes (in realistic scenarios) have encountered each other at least once, so if no aging is done on the contact history matrix, eventually all benefits from social based routing is lost. On the other hand, if aging is done too fast, contact history is based more on chance, than actual social contacts.

First we want to show the effects of different values for a simple aging algorithm: After an aging threshold Ω, a value is deleted from the contact history matrix.

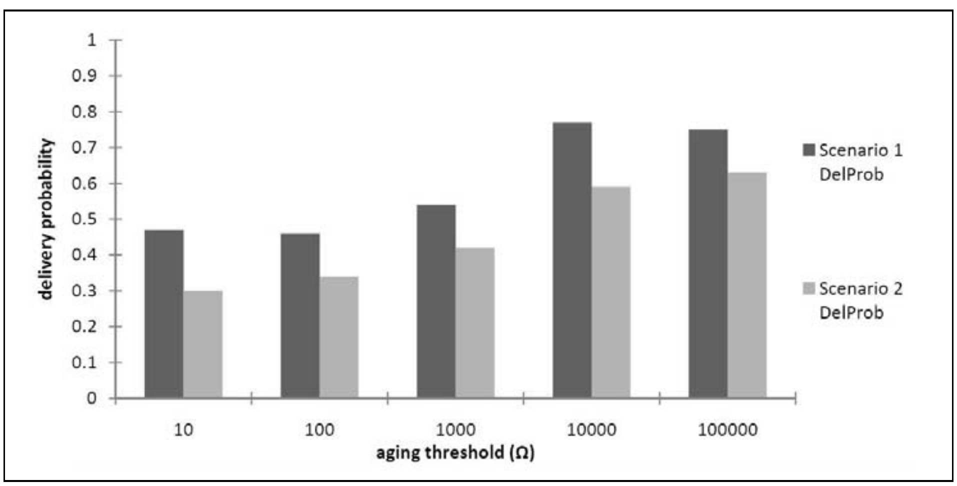

Figure 5.3: Comparing different values of Ω

This shows, that SimBet is quite dependant on a good value for the aging threshold. It is clear that arbitrary long aging will return even worse results. It is seen from the logarithmic graph that the peak of both scenarios is in the range of 10'000-end of scenario. This is a fairly close match, but that is only because the two scenarios are quite similar in matters of node speed and density, and that the scenarios are too sparse and short to feel the effect of too long aging that much. In scenario 1, the effect starts towards the end, but in a scenario as social as scenario 2, it will take quite some time until the effect becomes big. One problem about a fixed aging threshold is the variation in speed and density of a scenario. Imagine a scenario of the size 100x100 with speed and transmission range accordingly, the graph would shift to right by a factor 100 and delivery probability would be half on a scenario with the same relative settings. To be able to get more consistent results in different scenarios, there need to be a way to age the contact matrix dynamically, depending on individual factors like density and contact rate.

5.2.2 Feasibility study for improvement approaches

One approach would be to introduce some metric that tries to identify the scenario and adapts the threshold accordingly. We tried to find a good relation between ”node speed”-”scenario size”-”aging threshold”, but there's was no reliable correlation found, which could have been calculated by a node dynamically. This is however certainly a topic that might provide benefits and needs to be further looked into. The way that seemed most promising to us was, to rate the different contacts depending on how recent they happened. Therefore we changed the binary aging into a sliding process. Instead of just deleting an entry after a certain time, we let them degrade slowly. Before every access to the contact history matrix, we decreased all values by an aging constant (which we set to $\frac{1}{\Omega}$) multiplied by the time of last access. This resulted in a contact matrix, which had values slowly degrading from 1 to 0. To keep SimBet working, the Similarity Calculation was adapted, in order not to compare integers, but adding up the multiplication of every pair values that was above a certain threshold δ (δ=0.1).

$$\begin{aligned} Sim_{new} &= \sum (X_{me,n} * X_{other,n} \mid X_{me,n} > \delta \;\&\; X_{other,n} > \delta) \\ with\ X_{y,z} &= Entry\ in\ contact\ history\ matrix\ at\ position\ (y; z) \end{aligned} \tag{5.1}$$

Instead of

$$Sim = \sum (1 \mid X_{me,n} = 1 \;\&\; X_{other,n} = 1 \tag{5.2}$$

The simple aging algorithm had slightly higher peaks than ours, but we managed to get slightly more consistent results with the new aging algorithm. Especially in scenario 2, the new algorithm performed better at short aging intervals than the simple one. One reason might have been the sparseness of our scenarios, for which in the new algorithm, even lower values have been achieved for the similarity, which would have needed a higher $\frac{\alpha}{\beta}$ ratio to compensate. We believe in the benefits of an aging mechanism that, similar to the Prophet approach, takes the age of last encounter and at the same time the contact rate into consideration, by changing the binary contact history matrix into a weighed one. This should make SimBet more resistant to different kind of simulations, if the threshold is not adapted to a certain scenario and deliver a benefit to the protocol.

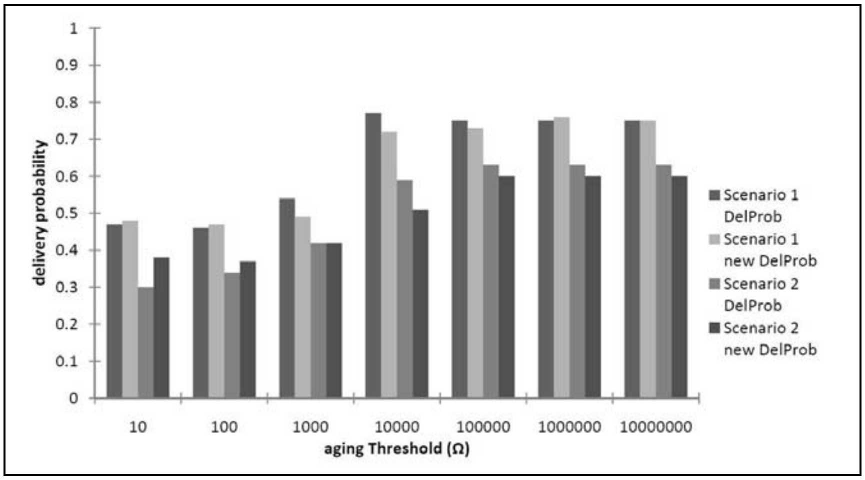

Figure 5.4: Comparing modified SimBet for different values of Ω

5.3 Using Similarity to calculate communities

5.3.1 Feasibility study

Our last approach of improving SimBet is the idea of including some means for a community detection into the protocol. In social scenarios (like our simulations) it should be an advantage to know when a message reaches a certain community. To assess the opportunities for improvement, we first labeled the nodes according to their group. As soon as a message is received by a member of the same group as the destination it is flagged. We added a final forwarding decision step, which prevents flagged messages to be routed to nodes outside the group. With this addition, we ran the simulations from section 5 again:

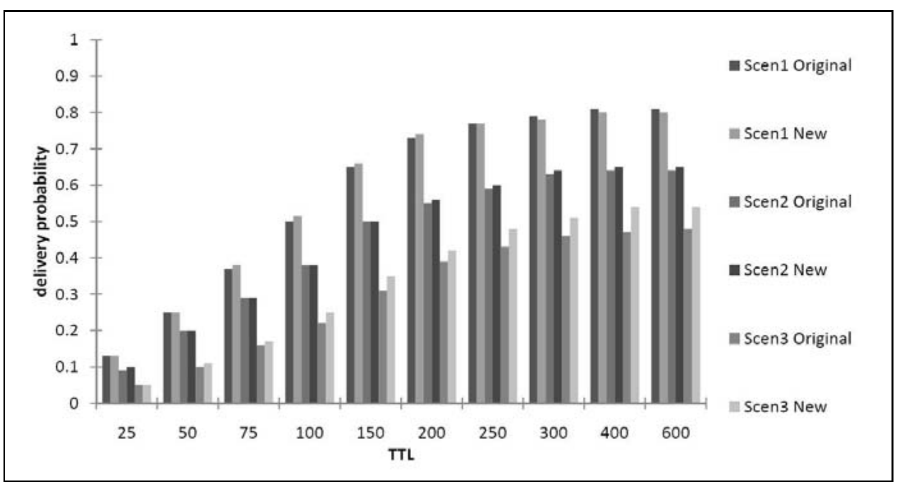

Figure 5.5: Feasibility study for community calculation

One can see from the graph that the addition might help a bit, but in general SimBet does a very good job with its similarity metric. This shows again that SimBet that greatly profits from its social based metrics, because at least in scenario 1 and 2 it was able to perform equally without the community flag. We believe that in scenario 1, forcing messages to stay in a group might not help more than it hurts, because most communities are shared by several groups and routing opportunities are missed, which SimBet might have found. In Scenario 2, the difference is also very small, mostly because here the similarity calculation is high enough to give a very distinguished similarity value depending on group affiliations, so the flag is not of great help anymore. In scenario 3 the benefit is clear, because of higher node movement and higher average speeds, messages can travel fast inside a group and have a significantly higher delivery probability, and SimBet has bigger troubles of finding a destinations group, because of the less distinct similarity values.

5.3.2 Implementing a simple community calculation

In order to be able to identify groups without using labels, we thought of an implementation using the similarity rating introduced in section 6a.2) to determine whether a node is in the same group as the destination node. We implemented an individual variable $Sim_{t,max}$ to the nodes, which stored the highest similarity value achieved so far. Our assumption is, that any node belonging to the same group would have a similarity calculation in the same range as the maximal similarity value achieved so far. So for all other similarity calculations above a threshold ? multiplied with $Sim_{t,max}$, we assume that those nodes are also in the same group.

$$Sim_t(d) > \gamma * Sim_max \ ; \ with \ \gamma \in [0.2; 0.8] \tag{5.3}$$

So we dynamically changed the $\frac{\alpha}{\beta}$ ratio in those cases, where a node achieves a similarity value high enough to assume it is in the same group to a value that inhibits forwarding to nodes not belonging to the group. For intra-community routing similarity is the important metric and in our experience, lower betweenness values are even beneficial for intra-community routing. Therefore we came to the conclusion that it would even be reasonable to punish betweenness in order to make sure the message stays in its group. We tried a few different value sets in scenario 3 at a TTL of 250 in figure 5.6. The optimal value we found was at $\frac{\alpha}{\beta}$= -10 and γ= 0.6. With those values, we compared the new protocol to the existing one and the version that doesn't allow the message to leave the destinations group in figure 5.7. In most of the cases, the new version performed even worse than the original. This means our algorithm did not work properly on this scenario, presumably because the benefits of the flagged variant lie in the cases were the similarity is very low. If the similarity value is high enough for the improved version to detect group membership, the original SimBet would route the same way anyways. We showed that adding a community calculation has possibilities of improving SimBet, but it would probably need to depend on a different calculation, otherwise the benefits are too small, because of the very reliable way SimBet handles similarity.

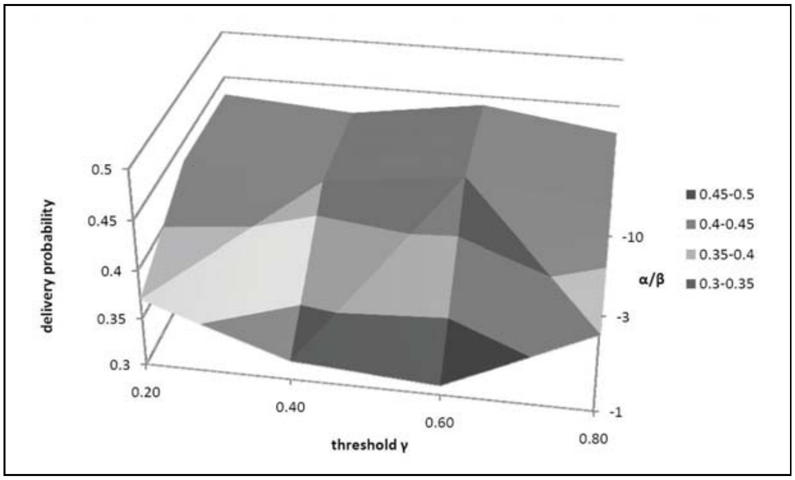

Figure 5.6: Delivery probability for different values of γ and $\frac{\alpha}{\beta}$

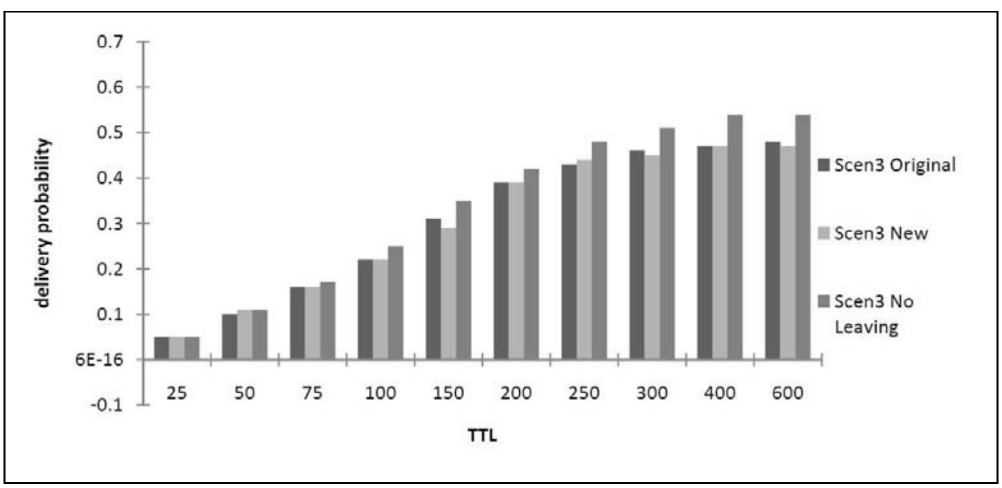

Figure 5.7: Comparison on scenario 3 with optimal values

Chapter 6

Conclusion

The clear conclusion of this paper is that it is of great benefit to use social metrics in DTN routing. SimBet outperformed Prophet in every scenario and achieved very good results against 2-copy Spray and Wait. Comparing to Epidemic Routing it has a reasonable delivery probability compared to a much lower overhead. From the several tests on different properties of SimBet, we can conclude that SimBet clearly benefits from its implementation of social metrics. As compared to Prophet, which also bases its routing decision on a social base, it definitely behaves much better in social environments. From the overhead the two (both single-copy) protocols incur to deliver messages, one can see that SimBet makes the wiser decisions and is able to achieve better results with less traffic.

As for the detailed analysis of SimBet, we were able to pinpoint at three weaknesses in the proposed implementation of SimBet. The choice of α and β is not optimal and could be chosen more wisely depending on the scenario. Our sensitivity study showed, that the proposed values achieve pretty good results, but there still lies some room for improvement in that field. Especially with an adapted aging algorithm as is proposed in section 5.2, α and β need to be adapted to the resulting values of the SimBet utilities. Aging was neglected completely, which is, as we have proven, a very important part for a social based routing protocol. Without aging, many benefits of social based routing are lost due to imprecise contact information. The proposed solution doesn't entirely solve the problem, but points to a direction that has to be further researched. As for the community calculation, our study showed that little gain is to be expected on any community calculation based on the already implemented metrics. However, there's still potential for improvement in that field, because the contact history matrix is stored in every node individually anyway, so there might be different metrics that would lead to better results.

We certainly gained a lot of experience in the field of DTN Routing and it would make a lot of sense to follow up on those ideas. One direction of further research should probably be the implementation of real-traces into the ONE simulator in order to verify our proposals on the original scenario as well as performing further tests on different synthetic scenarios. There also needs to be some sort of aging implemented into the algorithm to make it more robust against longer and possibly more dense scenarios. What became very clear out of our research, are the beneficial effects a multi-copy version would have; further studies should definitely look into an implementation of a multi-copy version of SimBet. Because of the mathematical approach of the protocol, one could use the existing metrics to implement a more sophisticated version of the binary distribution mode.

Chapter 7

References

[1] C. Diot et al. Haggle project, http://www.haggleproject.org, 2004.

[2] E. P. C. Jones, L. Li, and P. A. S. Ward. Practical routing in delay-tolerant networks. In Proc. WDTN, 2005.

[3] T. Spyropoulos, K. Psounis, and C. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In Proc. WDTN, 2005.

[4] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, April 2000.

[5] S. Okasha. Altruism, group selection and correlated interaction. British Journal for the Philosophy of Science, 56/4, 703/725, December 2005.

[6] E.Daly and M.Haahr. Social Network Analysis for Routing in Disconnected Delay-Tolerant MANETs. MobiHoc'07, ACM 978-1-59593-684-4/07/0009, September 2007

[7] Wei-Jen Hsu, Thrasyvoulos Spyropoulos, Konstantinos Psounis and Ahmed Helmy. Modeling Time-variant User Mobility in Wireless Mobile Networks. IEEE INFOCOM, May 2007

[8] Vijay Erramilli, Mark Crovella, Augustin Chaintreau, Christophe Diot. Diversity of Forwarding Paths in Pocket Switched Networks. IMC'07, ACM 978-1-59593-908-1/07/00, Octobre 2007

[9] One simulator, http://www.netlab.tkk.fi/tutkimus/dtn/theone/