Congestion Aware Forwarding in Delay Tolerant and Social Opportunistic Networks

Milena Radenkovic, Andrew Grundy
School of Computer Science and IT
University of Nottingham, UK
mvr,amg@cs.nott.ac.uk

***Abstract* - We propose an approach for opportunistic forwarding that supports optimization of multipoint high volume data flow transfer while maintaining high buffer availability and low delays. This paper explores a number of social, buffer and delay heuristics to offload the traffic from congested parts of the network and spread it over less congested parts of the network in order to keep low delays, high success ratios and high availability of nodes. We conduct an extensive set of experiments for assessing the performance of four newly proposed heuristics and compare them with Epidemic, Prophet, Spay and Wait and Spay and Focus protocols over real connectivity driven traces (RollerNet) and with a realistic publish subscribe filecasting application. We look into success ratio of answered queries, download times (delays) and availability of buffer across eight protocols for varying congestion levels in the face of increasing number of publishers and topic popularity. We show that all of our combined metrics perform better than Epidemic protocol, Prophet, Spray and Wait, Spray and Focus and our previous prototype across all the assessed criteria.**

I. INTRODUCTION

Over the last few years there has been a growing interest in DTN routing protocols that aim to improve the performance of DTNs under finite storage and/or transmission constraints. While DTN nodes are typically assumed to take part in routing messages of the others in order to help them achieve better success ratios and delay constraints, it is important that they keep control over the expected impact this has on their own resources. In this paper we address social opportunistic networks that are a class of DTNs that exhibit *the small world* phenomenon in which the individuals are linked by a short chain of acquaintances. Previous work by Hsy and Helmy [19] showed that node encounters in the real world analysis on real world traces of different university campus wireless networks are sufficient to build a connected relationship graph which is a small world graph. Since then a number of social analysis techniques have been successfully applied in delay-tolerant social opportunistic networks in order to increase the "probability to deliver" to a destination node and to minimise delays. Our previous work [1][15][16] and also [2] have identified that using social metric only leads to the load unfairly being distributed towards nodes which are better connected. Unfair load distribution and high unrestricted volumes of traffic can produce single node and network wide congestion. This becomes critical when opportunistic networks get adjacent to other networks because the connection hotspots can get overloaded and become unusable. In Gass's and Diot's recent work [14], in-motion proxies were proposed to efficiently take advantage of Wi-Fi access points while in-motion by utilizing transient connections while a user is moving. The goal of the in-motion client is to connect to an AP and transfer as much data as possible before moving out of range. [14] does not consider congestion that is likely to happen at the APs. If not handled within the routing protocol, congestion in opportunistic and delay tolerant networks may take the form of persistent storage exhaustion [4]. There are typically several solutions to the storage congestion problem such as: slowing sending rates of the sources, using alternative routes, discarding traffic, or migrating messages to alternative storage locations. We focus on congestion aware forwarding algorithms that adaptively choose the next hop based on contact history, predictive storage and delay analysis of a node and its ego network in order to distribute the load away from the storage hotspots and spread the traffic around. We define ego network (EN (X)) of a node X as a set of all (or some) of the contacts that a node X has "met". We extend and refine our forwarding heuristics proposed in [15,16] to take into consideration three aspects of a node congestion such as delay, buffer availability and congesting rate, and we introduce and define three aspects of node's ego network congestion such as delay, buffer availability and congestion rate of the ego networks. We argue that ego network metric and additional node metric help forwarding protocol handle more reliably and flexibly congested parts of the network by finding alternative routes. Our approach is particularly suitable to social opportunistic networks as we build on the huge diversity of forwarding paths in human contact networks as show in [7] by forwarding packets along multiple paths. In [6], the authors argue that if the end systems in the traditional Internet could spread their load across multiple paths in the right way, with the right reaction to the right congestion signals from the network, then traffic would quickly move away from congested or failed links in favour of uncongested links. [6] claim that resource pooling via multipath routing is the only way the phone network can achieve high reliability, greater than the reliability of the individual switches and links. We argue that the nodes in the opportunistic networks should also work together towards detecting and reacting to congestion in a self organized manner and as a single pooled resource. In this way the opportunistic networks would be even more reliable in the face of local surges in traffic and have higher utilization.

The remainder of the paper is organized as follows. Section II describes related work in the area of congestion control and forwarding in opportunistic networks and DTNs. Section III introduces our new forwarding heuristics that includes three parts: *social driven part* that exploits social relationships to allow optimal directionality and delivery probability of a node; *node congestion driven part* that considers and avoids nodes that have lower availability and higher congesting rates; and *ego network driven* that detects and avoids congested parts of the network to

minimise packet losses, delay and maximise success ratio and network availability. In Section IV, we perform an extensive set of trace driven simulations in ONE simulator [24] with RollerNet dataset [11] for opportunistic file casting communication. We show that the utilities of the new heuristics we propose increase robustness of content distribution in terms of shorter download times, higher ratio of queries being successfully solved and higher availability in the face of increasing topic popularity and increasing number of publishing nodes. We compare four combined metrics of our forwarding protocol, with node congestion awareness (based on buffer, delay and congesting rate of a node), ego network congestion awareness (based on buffer, delay and congesting rates of ego networks) and combination of both node and ego network congestions parameters against Epidemic, Prophet, Spray and Wait and Spray and Focus protocols.

II. Related Work

Forwarding algorithms for DTNs and opportunistic networks vary from epidemic replication of all the messages to every node, through to single copy forwarding. Multi-copy protocols typically aim to limit the number of replicas of the message in order to leverage a tradeoff between resource usage and probability of message delivery. Flooding-based protocols with unlimited replicas of messages cause high demand on network resources, such as storage and bandwidth and cause congestion. We review recent work on how congestion is handled by both multicopy and singlecopy DTN routing protocols.

Spray and Wait [17] is a quota-based protocol where an upper bound on the number of replicas allowed in the network is fixed during message creation. In a Binary spray and wait, any node X that has more than a single copy of the message, and encounters another node Y with no copies of the message, hands over to node Y half of its copies. When a node is left with only a single message, the message can be offloaded to the destination only. A follow-up protocol called Spray and Focus [18] uses a similar spray phase, followed by a focus phase, where single copies can be forwarded to a node that has "seen" the destination most recently in order to help maximize a utility function. Both of these use fixed number of copies per message.

[13][12] observe that overloading of a single node in a DTN does not indicate that there is network-wide congestion and that the number of copies allowed for the messages needs to be adaptive.

[13] propose EBR, a quote based replication protocol, where each node tracks its rates of encounters in order to intelligently decide how many replicas of a message a node should transfer during a contact opportunity. The appropriate fraction of message replicas the nodes should exchange when they meet is determined by the relative ratio of their respective rates of encounters.

[12] develop a dynamic, local approach to detect and respond to congestion by adjusting the copy limit for new messages. In their work DTN nodes use implicit indicators to detect congestion based on gathered network metrics from their contacts with other nodes. They investigate four implicit indicators of network-wide congestion: ACKS of generated messages, duplicate ACKs, timed out messages and dropped messages. The protocol proposes the nodes to create their own congestion view (CV) as the ratio of drops and duplicate deliveries and compare it to the congestion threshold. Depending on the comparison the copy limit for new messages is lowered or raised following a back-off algorithm. [12] consider that there is congestion once the number of drops exceeds duplicates. The target copy limit is set using additive increase multiplicative decrease on the current copy limit. This work has assumed a uniform network with random waypoint mobility. In reality the networks are likely to be non-uniform and the level of congestion may between different regions of the network. Even though this work identifies the need for congestion control algorithm that can adjust the number of message copies at any node - not only the sender, this work has not proposed how the network adjustments would compensate for differing local conditions.

[11] propose DA-SW (Density-Aware Spray-and-Wait), that is a measurement-oriented variant of the spray-and-wait algorithm that dynamically determines the number of a messages disseminated in the network in order to achieve constant delay. DA-SW relies on the current average node degree in the roller tour. Whenever a node has a bundle to transmit, it computes its current connectivity degree and refers to the abacus to determine the exact number of copies that is expected to lead to some expected delay. The authors define connectivity degree as a number of neighbours a node has "seen" in the last 30 sec. The authors did not address the impact of this measurement window on the performance of their system. This work does not consider dealing with resource constraints such as node buffers, bandwidth and energy consumption.

[4] address the problem of handling storage congestion at store-and-forward (DTN) nodes by migrating stored data to neighbours. They propose a set of algorithms to determine which messages should be migrated to which neighbours and when. This is based on the extension of the DTN custody transfer mechanism that enables a "pull" form of custody transfer where a custodian may request custody of a message from another custodian. This approach allows to decouple the problem of storage allocation among a relatively proximal group of storage nodes from the overall problem of path selection across a larger network. [4] select eligible storage neighbours using a function of available storage and incident link characteristics. [4] show how migrating custodian storage in this fashion can improve message completion rate by as much as 48% for some storage-constrained DTN networks.

[10] support the observation that encounters between nodes in real environments do not occur randomly [23] and that nodes do not have an equal probability of encountering a set of nodes. Combining Similarity, Betweeness and TieStrength (SimBetTS) for social routing metric was shown to result in improved overall delivery performance with the additional advantage that the load on central nodes is reduced and better distributed across the network. [10] show that SimBetTS achieves delivery performance comparable to Epidemic Routing, without the additional overhead and that it outperforms the PRoPHET routing protocol in terms of overall delivery performance. Replication can be used in SimBetTS to increases the probability of message delivery and the number of messages that are replicated depends on the relative SimBetTS utility value of each node. Consequently, the node with the higher utility value receives a higher replication value. This means that higher traffic will inevitably aggregate at and congest the points in the network that have higher SimBetTS social utility value.

FairRoute [2] argue that considering only contact histories to define contact duration, frequency and interaction strength cannot achieve balanced traffic distribution. In order to produce a fair distribution of load, FairRoute proposes nodes queue length to be evaluated in order to allow nodes to only forward to nodes with a bigger queue size. Similarly to our strategy FairRoute considers nodes buffer size. Contrary to our design, FairRoute defines a large queue size as a high social status and therefore a more desirable next hop. The only restriction that alleviates huge social clustering is that forwarding can be done to nodes with equal or higher status (queue size). This approach, as previous work such as SimBetTS, does not avoid congesting popular nodes and will inevitably lead to packets being dropped.

III. CONGESTION AWARE FORWARDING ALGORITHM

We propose an approach for congestion aware opportunistic forwarding that supports optimization of high volume multipoint data flows transfer while maintaining high buffer availability. Detecting and reacting to congestion in opportunistic networks is a difficult problem and was addressed in our earlier work [15,16] and also in [12,13]. In our previous work [15] we proposed a retentiveness and receptiveness utility driven forwarding protocol as an effective way of decongesting popular nodes in a social opportunistic network and distributing the traffic via different routes. In this paper we propose and investigate new metric for analyzing and integrating node buffer and delay behaviour with node's ego network buffer and delay behaviour in a number of new heuristics for different forwarding strategies. The new forwarding strategies aim to allow avoiding the nodes and the parts of the network with high congesting rates with the aim to keep high success ratio, low delays and good network efficiency even at times of increasing congestion.

In particular, our design combines: *social driven* part that aims to enable the most direct route to a destination node by selecting the intermediaries with higher probability of meeting the destination according to a social metric; *node resources driven* part that aims to detect and avoid the nodes that have low buffer availability, high delays or high congesting rate; *ego network driven* part that aims to detect and avoid parts of the networks that have low buffer and increased delay. In this way our protocol works as a local forwarding protocol that diverts the load from its conventional social aware path at times of congestion and directs it via a different path that decreases the load of hotspots and end-to-end delays while keeping high success ratios. When a potential intermediary node or its ego network (contacts that it has "seen") is about to get increasingly congested, we determine the load and expected delays of a set of neighbours and their ego networks, and choose alternative targets for offloading the messages. We perform statistical analysis of nodes' contacts, storage and delay history in order to make a decision as to whether to offload messages to it, or not. A number of heuristics are discussed and proposed to select the alternative node depending on how much knowledge the nodes have accumulated until that moment. These heuristics may select path with higher hop counts but lower load or delay, and higher success ratio when answering queries or offloading content.

A. Social driven Delivery Probability Considerations

In this paper our message delivery probability calculation, relies on calculating a metric based on the social metric such as connectivity, similarity, betweeness and tie strengths to decide on the probability of encountering a certain node, and using that to support the decision of whether or not to forward a packet to a certain node. We apply SimBetTS [10] utility metrics to our opportunistic forwarding as it is a suitable utility for distributed systems where global topology information is unavailable and where the underlying networks exhibit small-world characteristics. We utilise SimBetTS Routing metric as it comprises of both a node's centrality and intermediary's social similarity to the destinations and it was shown to work well both in cases of known and unknown destinations. Our work is not intended to be limited to using only SimBetTS metric but could be integrated with other social metrics as well.

B. Node resource considerations

1) Node Buffer

In our previous work we defined *current* Retentiveness as the percentage of remaining storage capacity and our congestion control was based only on discouraging the use of nodes that have lower levels of availability and promoting the use of nodes which have a slightly less desirable social forwarding heuristic, but have greater level of storage Retentiveness.

In this paper we extend and modify our buffer considerations in order to include: calculating *retentiveness* at time t as a percentage of the available buffer, smoothing retentiveness using exponential moving average to take in account longer term buffer behaviour in terms of buffer levels and congesting rate of a node that aims to be indicative of the rate of a node's buffer getting full.

We define node X congestion rate as follows. We denote node X's percentage of buffer availability at time t1 as $Ret_{t1}(X)$. Each nodes keeps track of the percentage of time it has been full $T_{\%Full}(X)$ (as shown in Equation 1) and the average time between its fullness periods $T_{AT}(X)$ (as given in Equation 2). We use CR(X) as an indication of how fast the buffer is to get filled up and we refer to it as *congesting-rate* of node X (given in Equation 3). On encounter, each node reports its current buffer availability $Ret_{curr}(X)$, its smoothed buffer availability Ret(X) and its congesting rate CR(X). Each node receives the same from the other nodes. Equation 4 shows an exponentially weighted moving average (EWMA) of retentiveness in order to smooth out short-term fluctuations and highlight longer-term trends without storing the history values. Equation 4 can be used for predicting the contact's remaining buffer where λ is the weight that identifies the degree of response, σ is the standard deviation of the buffer levels, W is a random number with zero mean and equal variances.

$$T_{\%Full}(X) = 100 \cdot \frac{T_{FullBuffer}(X)}{T_{TotalTime}(X)} \quad (1)$$

$$T_{AT}(X) = \frac{1}{N} \cdot \sum_{i=1}^{N} T_i end(X) - T_i start(X) \quad (2)$$

$$CR(X) = \frac{T_{\%Full}(X)}{T_{AT}(X)} \quad (3)$$

$$\mathrm{Re}t(x) = \lambda \cdot \mathrm{Re}t_{old}(x) + (1-\lambda) \cdot \mathrm{Re}t_{curr}(x) + \sigma \cdot W.\sqrt{1-\lambda^2} \quad (4)$$

We calculate the following relative utilities that allow us to compare *current* and *smoothed* buffer levels between the two nodes to allow comparisons between short and long terms buffer behaviour, and to compare *congesting rates* that predicts which of the two nodes is more likely to get filled up faster. We define the retentiveness utility and congesting rate utility in Equations 5 and 6 respectively. Equation 6 allows the nodes to explicitly avoid nodes that congest at a faster rate than the sending nodes. Equation 5 allows the sender to avoid nodes that have historically higher load than the sender. In our experiments we show that integrating CRUtil metric improves performance to the forwarding algorithms across all measured criteria.

$$\mathrm{Re}tUtil(X) = \frac{\mathrm{Re}t(X)}{\mathrm{Re}t(X) + \mathrm{Re}t(Y)} \quad (5)$$

$$CRUtil(X) = \frac{CR(X)}{CR(X) + CR(Y)} \quad (6)$$

2) Node Delay

We model in-network node delays as follows. Each node keeps track of when it receives and forwards the messages so that it could calculate the time per packet that the node has had it for. In [15] we considered only the duration of the delay message *p* has incurred due to being stored in X by keeping track of times messages were received ($T^{x}_{receive}(p)$) and forwarded ($T^{x}_{send}(p)$, and calculating the delay ($T^{x}_{send}(p)$ -($T^{x}_{receive}(p)$). For messages that are stored but not yet forwarded, we calculate $T_{current} - T_{received}$. This allows us to take into account the on-going delays of the packets that have not yet been forwarded and provide more accurate delay of the node X. We investigate if penalising a node for dropping messages by sharply raising it current packet delay (we double the current delay for the packet that gets dropped) will improve the performance of our forwarding algorithms. By doing this, we aim to make the nodes that drop messages less desirable next hops.

Equation 7 shows that we calculate receptiveness as the exponential moving average of the delay this node has added to the packets it has held, this value is updated each time a packet is successfully forwarded or is dropped. The intuition behind this is that the longer this node has held onto the packet for the more congested the network is. We calculate receptiveness utility as in 8.

$$\mathrm{Re}c_{new}(x) = \lambda \cdot \mathrm{Re}c_{old}(x) + (1-\lambda) \cdot \mathrm{Re}c_{curr}(x) \quad (7)$$

$$\mathrm{Re}cUtil(X) = \frac{\mathrm{Re}c(X)}{\mathrm{Re}c(X) + \mathrm{Re}c(Y)} \quad (8)$$

C. Ego network

We believe that it is important to consider node's ego-network wide congestion metric such as buffer availability or buffer congesting rates across all or some of the contacts a node has met. This will allow the sending node to offload its data even to the nodes that are currently and historically worse than the sender but are better in terms of meeting the nodes that are congesting at the lower rate. Similarly, when the sender meets a node with higher total current utility, the sender may not always forward to it as in our previous work. Rather, it will check the contact's ego network and predicted congesting rate. If the potential next hop is only better currently but does not have lower prediction for congesting rate nor it is connected to the ego network that has lower congesting rates, the sender may not forward to it.

1) Ego Network Buffer consideration

We argue that it is important to consider buffer levels (Equation 9) and congesting rates in node's ego network (Equation 10 and 11) as it provides a wider perspective of the network than considering only a single node resources while it can be easily locally gathered by each node. More specifically, this would allow the nodes to detect and avoid more congested parts of the networks and move to freer parts of the network.

Equation 9 defines ego-network-smoothed-buffer-availability as a sum (smoothed) of average buffer availabilities of the nodes that the nodes encountered.

We define *congestion rate of ego network* of a node X in Equation 11 as an average (or EWMA) sum of congesting rates of all the encountered nodes for node X.

We define $EN_{\%Full}(X)$ in Equation 10 as a percentage of full (congested) nodes a node X meets on its way. Each node keeps track of the percentage of filled up nodes it meets on its way. This can be calculated over the entire time but is more effective if calculated either using EWMA in order to take into account smoothing, or across the most recent contacts, or most frequent contacts.

$$EN_{\mathrm{Re}t}(X) = \frac{1}{N}\sum_{1=1}^{N} \mathrm{Re}t(c_i(X)) \quad ...(9)$$

$$EN_{\%Full}(X) = \frac{N_{Full}(C_1(X)..C_N(X))}{N(X)} \quad (10)$$

$$EN_{CR} = \frac{1}{N}\sum_{1=1}^{N} CR_i(X) \quad (11)$$

We propose to calculate the following relative utilities: ego network average buffer utility (Equation 12) and ego network congesting rate utility (Equation 15) that allow us to check which of the subnetworks has higher storage and to predict which of the ego networks of the two nodes is more likely to get filled up faster.

$$EN_{\mathrm{Re}t}Util(X) \frac{EN_{\mathrm{Re}t}(X)}{EN_{\mathrm{Re}t}(X) + EN_{\mathrm{Re}t}(Y)} \quad (12)$$

2) Ego Network Delay consideration

We propose to consider the delays for ego networks as they can also provide a valuable insight on delay of a part of the network that can be locally gathered. We calculate this as an exponential weighted average sum of in-network node delays that a node meets on its way (Equation 13). Similarly to the ego network buffer considerations, we can calculate the averaged sum of the delay of the recently met nodes or most frequently met nodes as it may be more accurately adapting to the dynamic topology. This metric allows nodes to move from the parts of the network with higher delays to other parts with lower delays.

$$EN_{\mathrm{Re}c} = \frac{1}{N}\sum_{1=1}^{N} \mathrm{Re}c_i(X) \quad (13)$$

Equations 14 define the ego network receptiveness utility of an ego network as:

$$EN_{\mathrm{Re}c}Util(X)\frac{EN_{\mathrm{Re}c}(X)}{EN_{\mathrm{Re}c}(X)+EN_{\mathrm{Re}c}(Y)} \quad (14)$$

$$EN_{CR}Util(X)\frac{EN_{CR}(X)}{EN_{CR}(X)+EN_{CR}(Y)} \quad (15)$$

D. Forwarding Strategies

Utilising only a single forwarding strategy based on one node utility function as proposed in our previous work [15,16] may lead to potentially suboptimal next hop choices when the network connectivity is very dynamic. In this section we use the utilities we defined in sections III B and III C to explore the forwarding algorithms that include a subset (or all) of the following aims: avoid the nodes that have higher congesting rates or higher delays or lower retentiveness, or avoid nodes whose ego networks have higher congesting rates, lower storage or higher delays, and the combination of all of them.

The introduction of node resource utility and ego network resource utility allows us to 1) move our traffic from more overloaded part of the network to less overloaded part of the network, and 2) avoid greedy choices of more available nodes that may later on congest or are highly connected to nodes with higher congestion rates. The decisions of how strict a node should be and how greedy it should be are not trivial to make and generally should depend on the state of the network. In some cases it might be sensible to only give a packet to nodes that have a high buffer and delay utility. On the other hand, when encountering a node with a lower current buffer and delay but that is connected to better nodes, it may be better to use it rather than a node that has current good resources utility on its own but is connected to the nods worse than itself. While [14] argues that it is good to offload as much data as a node can before it goes out of range of an intermediary, we argue that at times of high traffic rates this may not be the most suitable strategy as it would create congestion and cause more disconnections as we will show in section IV.

We define several combined utility functions that aim to integrate different subsets of individual utilities defined in sections III B and III C, and manage the tradeoffs across multiple criteria. The combined social and resource driven utilities allow us to use directionality heuristic for quick delivery SimBetTS [10] as long as the network is not congested, When it gets congested, the combined utility allows us to move to an entirely different dissimilar network that is less congested. Social only forwarding utility proposed in [10] gives more copies to the nodes with higher social utility value with the destination. This will congest the nodes that are more similar to the destination or that have higher centrality values. EBR in [13] will similarly target to gives higher number of copies to the nodes better connected. We target to de-cluster individual nodes and parts of the network by leveraging social metric with resource constraints.

$$SRRUtil(X) = SUtil(X) + \mathrm{Re}cUtil(X) + \mathrm{Re}tUtil(X) \quad (16)$$

The aggregate utility given in the Equation 16 integrates node's Social utility, Retentiveness utility and Receptiveness utility in order to allow the nodes to avoid the nodes with high delays, low buffer availability or low Social utility.

$$SRRCRUtil(X) = SUtil(X) + \mathrm{Re}cUtil(X) + \mathrm{Re}tUtil(X) + CRUtil(X) \quad (17)$$

In addition to taking into account nodes's centrality and social similarity, retentiveness and receptiveness as in (16), Equation 17 takes into account the relative utility of the nodes' congesting rates. This will allow us to choose the nodes that congest slower.

$$SEN_{CR}Util(X) = SUtil(X) + EN_{CR}Util(X) \quad (18)$$

Equation 18 takes a radically different approach and looks only at relative nodes' ego network congesting rates in addition to the Social utility of the nodes. This is important in order to allow us to avoid nodes that are connected to more congested nodes (part of the network) and move to the other parts of the network that are less congested.

$$SEN_{\mathrm{Re}t}Util(X) = SUtil(X) + EN_{\mathrm{Re}t}Util(X) \quad ...(19)$$

Similarly to (18), Equation 19 takes into account the relative nodes' ego networks average buffer availability and social utility of nodes. This is important to allow us to avoid nodes that are connected to less available nodes (more congested part of the network) and move to the other more available parts of the network.

$$SEN_{\mathrm{Re}c}Util(X) = SUtil(X) + EN_{\mathrm{Re}c}Util(X) + EN_{CR}Util(X) + EN_{\mathrm{Re}t}Util(X) \quad (20)$$

Equation 20 defines a combined ego network utility that allows us to avoid nodes that are connected to a set of

nodes that congest faster, have higher delays or low buffer availability.

$$TotalUtil(X) = SUtil(X) + \mathrm{Re}cUtil(X) + \mathrm{Re}tUtil(X) + CRUtil(X) + EN_{\mathrm{Re}c}Util(X) + EN_{CR}Util(X) + EN_{\mathrm{Re}t}Util(X) \quad (21)$$

Equation 21 defines a combined total utility that takes into account all utilities: node retentiveness, receptiveness and congesting rate, and ego network's congesting rate, retentiveness and delays.

Our algorithm functions as follows. After we calculate the Utility of all the neighbours of the sending node X there are a number of options that are not trivial to decide about. In our early prototype, a sender chooses the bestFit node that has the higher total utility compared to its other neighbours, and sent messages to it if it was not full. This is simplified approach that can result in suboptimal forwards, increased delays and packet losses as we will show in section IV. With the use of ego network statistics and node congesting rate statistics, a sender may not always offload to the bestFit (e.g. if the bestFit's ego network has worse metric than sender's ego network) and may offload to betsFit that has lower current metric to it (if its predicted congestion rate and ego network metrics are better). Also, in this work, we discuss two options on how many copy of the message the node will offload when it has found the best contact to forward to: First we will consider offloading a single copy, and then multiple copies of a message. Single copy is more resource efficient but typically has lower delivery rations than multicopy protocols. In our experiments we show that even a single copy protocol, our protocol can outperform other multicopy protocols. In case of multicopy strategy, we propose to utilise EN_{Ret} to determine the correct number of replicas in the following way. If there are M replicas of a message at node Y, Equation 22 shows the number of messages that are sent to X. This allows higher number of messages to be offloaded to the freer parts of the network.

$$M \cdot \frac{EN_{\mathrm{Re}t}(X)}{EN_{\mathrm{Re}t}(X) + EN_{\mathrm{Re}t}(Y)} \quad (22)$$

IV. Experiment Setup and Results

This section conducts trace driven experiments of multiple forwarding heuristic we proposed in our section III. We investigate performance of four combined heuristics in terms of success ratios, delays and available buffer of the nodes under increasing congestion levels in a realistic multipoint publish-subscribe podcasting application. In our previous work we used Infocom 2005 traces [21] for testing our early Café prototype [15,16]. Now we choose to use RollerNet [11] connectivity traces and show that we outperform our early prototype, Prophet, Spray and Wait and Spray and Focus in terms of availability, delays and success ratio of answered queries. We show that all our metrics have higher availability of nodes, lower delays and that the rate of buffers filling up is slower compared to the Epidemic routing, Spray and Wait and Prophet. We also show that combined total utility is usually the best out of three metrics we test. More interestingly, we show that ego network-only considerations can have very good performance and be better than Prophet, Spray and Wait and Epidemic routing protocols.

A. Real traces

In this paper we use RollerNet [11] dataset as it represents a class of DTNs that follows a pipelined shape because it has extreme dynamics in the mobility pattern of a large number of nodes. The fluctuations in the motion of the rollerbladers cause a typical *accordion* phenomenon – the topology expands and shrinks with time, thus influencing connection times and opportunities between participants. [11] show that the resulting connectivity graphs exhibit a number of properties that had not been previously observed. The accordion phenomenon occurs in pipelined sets of interconnected systems and means that a variation in the state of one system can greatly impact the states of the other systems.

B. Publish-Subscribe Filecasting Application

We have built a fully distributed file casting application on the top of the forwarding protocols we are testing. Our multipoint publish subscribe application works as follows. Each node that has content it wants to publish will send that content to the nodes that are interested in it and/or the nodes that "know" the nodes that are interested in it as long as they have availability. Our content is organized as in previous filecasting work [5]: it contains topics and each topic has chunks that can be exchanged when the two nodes meet. Each chunk has a unique ID and the topic has the total number of chunks. We randomly assign topics to share and we choose random number of publishers. Nodes randomly choose to be interested in certain topics. Each node has a queue size of 1000 units. Podcasting nodes send at the rate of 5 chunks a second. We have run four combined heuristics for our congestion aware forwarding algorithm proposed in Section III D: SRRUtil, SRRCRUtil, EN_{Ret}Util and TotalUtil against Prophet, Spray and Wait, Spray and Focus and Epidemic routing protocols on RollerNet connectivity traces. Our aim is to explore how successful node-only metric (SRRUtil and SRRCRUTil), ego-network-only (EN_{Ret}Util metric or total combined metric (TotalUtil) are in terms of success ratio of answered queries, download times (delays) and average buffer availability. We have run eight experiments first with increasing number of publishers ranging from 9% to 70%, then with increasing number of subscribers from 9% to 70%. All simulations are repeated five times with different random subscribers and publishers. We compare the performance of all eight protocols (Prophet, Spray and Wait (with three copy per message), Spray and Focus (with three copies per message), epidemic routing, and our forwarding protocols with SRRUtil, SRRCRUtil, SEN_{Ret}Util and TotalUtil)) in terms of delays, success ratio and buffer availability. The results are discussed in section IV C.

C. Results

Figure 1 shows average node availability for increasing number of publishers and six subsribers. Total and SRRCUUtil metrics are having the highest query success ratio with Total metric being slightly higher than SRRCU. SEN_{Ret}Util performs as good as or better than SRRUtil

protocol. All four new tested metrics have better availability than other four protocols. For low congestion rates up to 20% of publishers at maximum rates, SRRUtil, SRRCRUtil and TotalUtil are about 80% higher than the Spray and Focus and more than 30% better than Epidemic routing, Prophet, and Spray and Wait. For medium congestion levels, from 20% to 60% of publishers, SRRCRUtil and TotalUtil are up to 60% higher then Spray and Wait and Spray and Focus, 20% better than SRRUtil and 10% better than EN_{Ret}Util. For high congestion levels, above 70% publishers, are up to 500% higher than the Spray and Wait and Spray and Focus and two times better than Prophet and Epidemic routing.

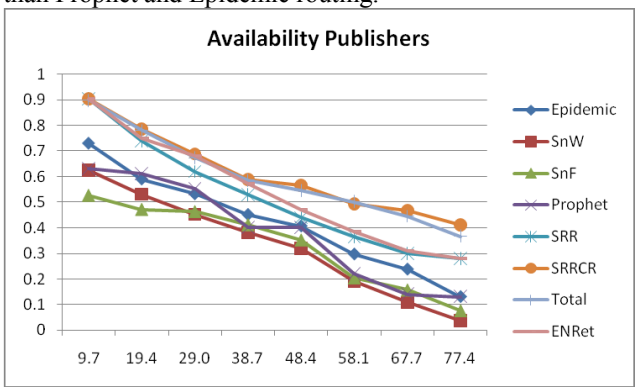

Figure 1. Availability for Increasing number of publishers

Figure 2 shows average node availability for increasing number of subscribers and six publishers. Our total metric is significantly higher than other metrics for all the congestion levels and always above 60%. SRRCRUtil is close to TotalUtil for low congestion levels, but then worsens up to 15% compared to the TotalUtil as the congestion increases. SEN_{Ret}UtilEgo is always higher than node only multi-metric SRRCRUtil (from 30-50% better) as the congestion increases. It is interesting to see that Total metric two times better SRRUtil metric and 50% better than ENRet Util metric as the congestion inceases.

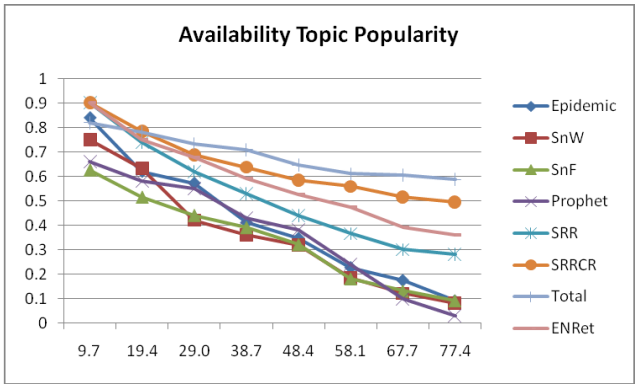

Figure 2. Availability of nodes for increasing topic popularity

Figures 3 and 4 show success ratios for increasing topic popularty and number of publishers respectivley. In both scenarios our total metric keeps the success ratio close to 70% during all levels of congestion, and all our metrics are above 350% (fow low congestion levels) and up to 700% (for high congestion levels). Prophet, epidemic, Spray and Focus and Spray and Wait protocols perform significantly worse as they hardly reach 15% of success ratio for low congetion levels and close to zero for high congestion levels.

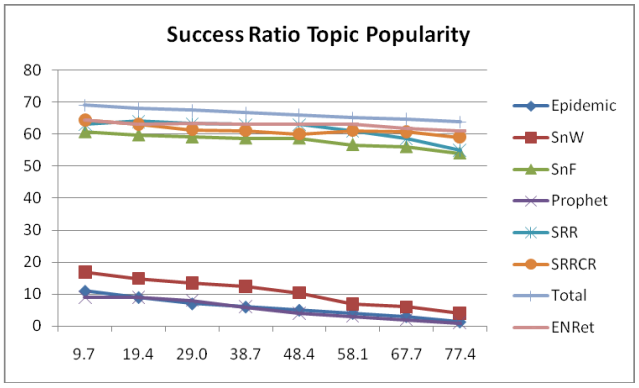

Figure 3. Success ratio for incresing topic popularity (number of subscribers)

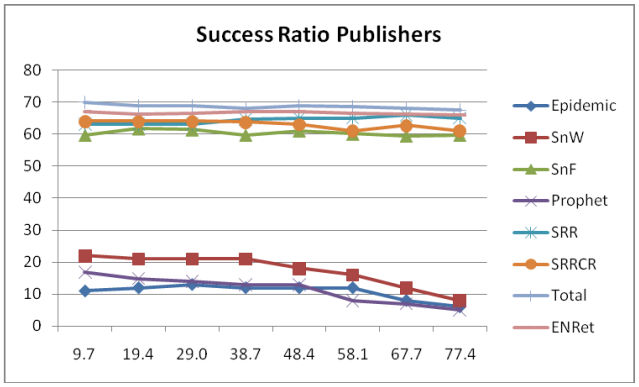

Figure 4. Success ratio for increasing number of publishers

Figure 5 shows average delays for increasing number of publishers for all 8 protocols. We see that our total metric has the lowest delays for all congetion levels and the delays are kept under 30 seconds. SRRCRUtil has similar delays as Total but starts to increase for 70% of publishers. SEN_{Ret}Util performs better than SRRUtil for all congestion levels. All of our four metrics lower delays than the other protocols: SRRUtil is 50% better than Spray and Focus; TotalUtil that is 300% better than Spray and Wait and TotalUtil is 700% better than Prophet for high congestion levels.

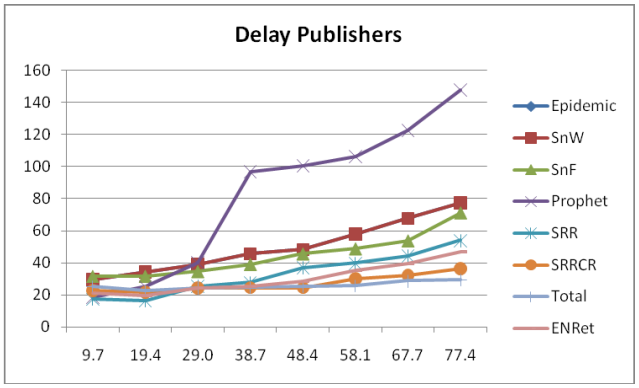

Figure 5. Delay for increasing number of publishers

Figure 6 shows average delays for increasing topic popularity. We see that TotalUtil metric has higher delays than it does for increasing number of publishers when the congestion levels are above 50% but it still never exceeds 30 seconds. SRRCRUtil and ENRetUtil perform similary but SRRCR has slightly lower delays. SRRUtil has the higher delays out of our four metrics but still considerably lower than Spray and Focus (by 200%) and Spray and Wait

(above 300%). The delays of epidemic routing gets lower towards very high levels of congestion because its success ratio is very low, and Prophet has lower delays as its success ratio is very low for the same scenario.

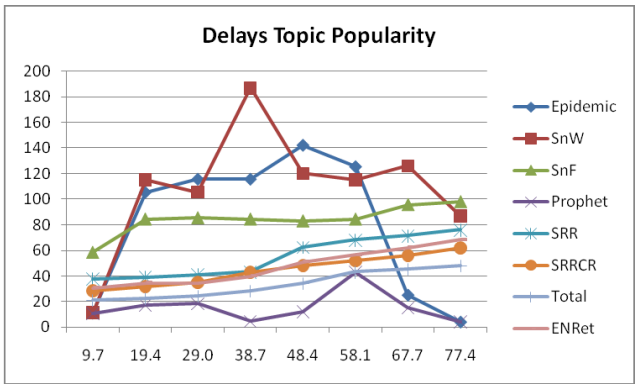

Figure 6. Delays for increasing topic popularity (number of subscribers)

Our experiments show that statistics of ego network retentivness and node congesting rates are highly valuable for forwarding in social opportunistic networks. Even a simple ego network retentivness for the majority of time performs better than more sophisticated analysis of one node only. Combined metric (node and its ego network statistics) allows better adaptation to the dynamic conditions and higher congestions levels.

V. CONCLUSION

We have proposed multiple node and ego network buffer and delay utility metrics for enabling congestion aware forwarding of high volume multi point flow transfer in delay tolerant and social opportunistic networks. Our combined utilities allow the forwarding protocol to be more dynamic and flexible as it operates as a pure social protocol at times of low congestion and as fully resource driven protocol at times of high congestion**.** We have done extensive real trace driven experiments in ONE in order to compare the performance of four of our new metrics versus epidemic, Prophet, Spray and Wait and Spray and Focus protocols. All our metrics perform better than epidemic, Prophet, Spray and Wait, Spray and Focus in terms of delay, buffer availability and success ratio in the face of increasing congestion levels for a realistic multipoint publish subscribe file casting application. Our combined ego network and node congestion metric performs better in all the cases than the other three of our metrics but can similar to SRRCR for low and medium congestion levels. SEN_{Ret} metric is similar to CRRCR metric and always better than SRR metric. We plan to extend our work with multicopy forwarding (that we proposed in Equation 22) and investigate its impact on the protocol performance when compared to [12,13]. We aim to explore in greater detail ego network resource metric proposed in III C and III D and adaptive weighting of the utilities used in the TotalUtility. We believe that the heuristics proposed in this paper can be effective for many application scenarios that are outside the area of social based networks only and can include other complex types of networks.

REFERENCES

[1] A. Grundy and M. Radenkovic, "Decongesting opportunistic social based forwarding," in WONS, Slovenia, 2010

[2] J. Pujol, A. Toledo, and P. Rodriguez, "Fair Routing in Delay Tolerant Networks," in IEEE INFOCOM, 2009

[3] E. Yoneki, P. Hui, S. Chan, and J. Crowcroft, "A socio-aware overlay for publish/subscribe communication in delay tolerant networks," in Proc. ACM MSWiM, 2007

[4] M. Seligman, K. Fall, P. Mundur, "Storage Routing for DTN congestion control", Wireless Communications and Mobile Computing, Vol 7, Issue 1,0, p1183-1196

[5] M. May et all, "Wireless opportunistic podcasting: implementation and design tradeoffs," in Proc. of ACM Workshop on Challenged networks. ACM, 2007, p. 82.

[6] D. Wischik, M. Handley, and M. Braun, "The resource pooling principle," ACM SIGCOMM Computer Communication Review, vol. 38, no. 5,pp. 47–52, 2008.

[7] V. Erramilli, A. Chaintreau, M. Crovella, and C. Diot, "Diversity of forwarding paths in pocket switched networks," in Proceedings of the 7th ACM SIGCOMM conference on Internet measurement. ACM, 2007, p. 174.

[8] J. Scott, P. Hui, J. Crowcroft, and C. Diot, "Haggle: A networking architecture designed around mobile users," IFIP WONS, vol. 2006, 2006.

[9] J. Ott, Mikko Pitkanen, "DTN-based Content Storage and Retrieval", In Prof. WOWMOM 2007

[10] E. Daly and M. Haahr, "Social network analysis for information flow in disconnected Delay-Tolerant MANETs," IEEE Trans. Mob. Comp, 2009.

[11] Pierre-Ugo Tournoux Jeremie Leguay, et al, "The Accordion Phenomenon: Analysis, Characterization, and Impact on DTN Routing", in Proc of INFOCOM 2010, 1116-1124

[12] N. Thompson, R Kravets, "Poster Abstract: Understanding and Controlling Congestion in Delay Tolerant Networks", in ACM SIGMOBILE Mobile Computing and Communications Review, Poster session MOBICOM 2008, 13/3, 2009, pp 42-45

[13] S.C. Nelson, M. Bakht, R. Kravets, S. HarrisIII: "Encounter: based routing in DTNs", Mobile Computing and Communications Review 13(1) 56-59 (2009)

[14] R. Gass and C. Diot. "Eliminating Backhaul Bottlenecks for Opportunistically Encountered Wi-Fi Hotspots". VTC 2010. Taipei, May 2010.

[15] A. Grundy; M. Radenkovic, "Promoting Congestion Control in Opportunistic Networks", In the Proceedings of IEEE WiMob 2010, Niagara Falls, Canada

[16] M. Radenkovic, A. Grundy, "Congestion Aware Data Dissemination in Social Opportunistic Networks", Poster in the ACM Mobicom/Mobihoc 2010, Chicago, USA

[17] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. "Spray and wait: An efficient routing scheme for intermittently connected mobile networks". In WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, 2005

[18] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility". In Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, 2007

[19] W.-J. Hsu and A. Helmy, "Impact: Investigation of Mobile-User Patterns across University Campuses Using WLAN Trace Analysis," Proc. IEEE INFOCOM, Aug. 2005

[20] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau,

[21] "CRAWDAD data set cambridge/haggle (v. 2009-05-29)," Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle, May 2009.

[22] Y. Jiao et al**, "**Data Dissemination in Delay and Disruption Tolerant Networks Based on CoIntent Classification", in proc of MSN'09

[23] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket Switched Networks and Human Mobility in Conference Environments," Proc. ACM SIGCOMM Workshop Delay-Tolerant Networking (WDTN '05), pp. 244-251, Aug. 2005.

[24] ONE simulator. http://www.netlab.tkk.fi/tutkimus/dtn/theone