# Promoting Congestion Control in Opportunistic Networks

Andrew Grundy
Department of Computer Science,
University of Nottingham
Jubilee Campus, Wollaton Road,
Nottingham, NG8 1BB, UK
amg@cs.nott.ac.uk

Milena Radenkovic
Department of Computer Science,
University of Nottingham
Jubilee Campus, Wollaton Road,
Nottingham, NG8 1BB, UK
mvr@cs.nott.ac.uk

*Abstract*—**This paper is concerned with congestion aware forwarding algorithms within opportunistic networks. We remove the reoccurring assumption of unlimited storage, and make it evident that congestion is a prominent problem that needs to be addressed. We propose a distributed congestion control algorithm that adaptively chooses the next hop based on contact history and statistics, as well as storage statistics. We aim to distribute the load away from the storage hotspots in order to spread the traffic around. We perform an extensive set of trace driven simulations for "several-to-many" communication patterns in opportunistic networks. We show that congestion control is an essential component in the transfer of data in opportunistic networks, and can be achieved in a fully open loop manner and by only local dissemination of statistics of nodes availability and connectivity. Our results with real connectivity traces show that by using novel availability heuristic we achieve higher levels of sent and delivered packets and outperform current opportunistic forwarding protocols, such as SimBetTS and FairRoute.**

*Index Terms*—**Opportunistic Forwarding, Congestion**

## I. INTRODUCTION

Recent developments in social-based opportunistic forwarding [1], [2] have identified that load is unfairly distributed towards nodes which are better connected. By making informed forwarding decisions based on a heuristic that favours connectivity, delivery probabilities increase, but load distribution becomes more unbalanced. Unfair load distribution and high unrestricted volumes of traffic produce congestion. This is particularly challenging in cases when opportunistic networks get adjacent to other networks (i.e. in order to leverage more capacity out of pure cellular networks) because the connection hotspots can get overloaded and become unusable.

Traditional connection-oriented protocols, such as TCP, are built around the assumption of contemporaneous end-to-end connectivity and they use closed loop control as they rely on acknowledgments. TCP congestion control mechanisms are integral to the stability of the traditional Internet, but are not usable in the non-Internet like scenarios (where the assumption of end to end connectivity does not hold) that are tolerant to delays. Open-loop congestion control protocols are much better suited to opportunistic networks, but they have not yet been extensively studied and integrated in current opportunistic protocols. It is clear that in order to be robust opportunistic networks need to be concerned with congestion control. More specifically, due to the fragmented nature of opportunistic networks, our concern needs to be with the distribution and self organisation of congestion control.

The key problem is that nodes within opportunistic networks only have access to the available bandwidth information for their next hop and due to the changing nature of the topology disseminating this information is not productive. Without new forwarding techniques, the packets will inevitably collect at bottleneck nodes (e.g. more popular nodes and hotspots) and be dropped. For example, if hotspots get congested during opportunistic bulk data transfer, congestion aware forwarding is needed that would offload the load away from these hotspots and spread it around.

The rest of the paper is organized as follows. Section II describes related work. Section III identifies the meaning of congestion in opportunistic networks (in comparison to the traditional networks) and identifies the mechanisms needed to alleviate it. Section IV proposes two new metric Receptiveness and Retentiveness, that can help measure how utilised a node is in terms of transfer bandwidth and buffer capacity respectively. We propose heuristic to factor in a cost metric associated with storage so that traffic is spread away from centrality or ego-network-centric paths and only paths with more storage and lower delays are chosen. We make use of the diversity of forwarding paths in human contact networks [6] by forwarding packets along multiple paths similarly to the resource pooling principal [5].

Section V describes extensive set of experiments with several to many multipoint traffic pattern and real connectivity traces and gives our results. By performing real trace driven evaluation of our protocol and using multipoint traffic pattern we aim to be congruent with the real envisaged application scenarios such as publish/subscribe applications in wireless opportunistic podcasting applications [4]. Section VI concludes and outlines future work.

## II. RELATED WORK

### A. MANET Congestion Control

A survey on congestion control for MANETs [14] argues that the majority of work is concerned with improving the performance of TCP in MANET scenarios. [15] indicates that

TCP-like congestion control mechanisms suffer fundamental problems in the presence of wireless interference. Unlike the majority of congestion control protocols for MANETs, EXACT [16] and ATP [17] are clean-slate approaches to congestion control. Both EXACT and ATP require stable, contemporaneous connectivity and are thus not applicable to opportunistic networks.

*1) EXplicit rAte-based flow ConTrol (EXACT) [16]:* EXACT is in-network, rate based flow control. Nodes have dedicated state variables for flows they are participating in. Each node determines a current bandwidth that they can supply to neighbouring nodes, calculating fair bandwidth shares for each flow. Explicit rate information is updated by the intermediate nodes in each passing packet, where the intermediate nodes available bandwidth is lower than the rate currently specified in the packet header.

*2) Ad-hoc Transport Protocol (ATP) [17]:* ATP strictly separates congestion control and reliability mechanisms and only requires a small amount of feedback from the receiver. In contrast to EXACT, ATP does not require any flow-specific state variables in the intermediate nodes. All nodes calculate an exponential average of the delay of all packets passing through them. This delay consists of the time a packet had to wait in the node's local queue and of the time to wait for a free medium before it could be transmitted. These values are independent of the flows the packets belongs to.

### B. DTN Congestion Control

The majority of research in congestion control for DTNs is concerned with buffer management [18], [19], [20], [21]. We briefly describe some of the major approaches below.

*1) Storage Routing (SR) [21]:* SR avoids congested nodes dropping packets by sending a set of messages out to a set of available neighbours. When the congested nodes manage to reduce its buffer capacity, any messages that were previously migrated are retrieved. SR operates as a local routing protocol to divert messages from their conventional routing path, for later forwarding. All the nodes in the k-neighbourhood of a congested node with connectivity are eligible to be selected as alternative custodians. The major shortcoming of this protocol is that it only temporarily alleviates congestion as it relies on contacts that are present not suffering from congestion themselves.

*2) Autonomous Congestion Control (ACC)[18]:* ACC implements congestion control by propagating buffer utilisation stress backwards through the network to the source nodes. This is accomplished by intermediary nodes declining to take custody of packets, forcing the sending node to retain packets that results in a demand for buffer space, which in turn forces another refusal of custody, and this continues back to the source node. ACC apply a financial model to buffer space management. Unoccupied buffer space is regarded as money free to invest and routing network traffic is regarded as the daily financial activities of an investment banker. The constraint on the activity of the router is in essence "cash flow", which limits his ability to accept ("purchase") new

packets. This work holds similarities with our concept of Retentiveness, but they do not overcome intermediary / source nodes filling their own buffers and isolating peripheral nodes.

### C. DTN Load Distribution

*1) FairRoute:* 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 allowing 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. THE PROBLEM

In this section we aim to give an outline of what congestion refers to in opportunistic networks. Even though most of the forms of congestion collapse identified for the Internet can be applicable to opportunistic networks, the causes and the methods for detecting and correcting congestion differ vastly.

The work by Floyd and Fall [7] outlines five different forms of congestion collapse in the Internet: classical congestion collapse, congestion collapse from undelivered packets, fragmentation based congestion collapse, congestion collapse from increased control traffic and congestion collapse from stale or unwanted packets. We here discuss the causes of congestion and their applicability to opportunistic networks.

Classical congestion collapse is the result of retransmitting packets that are already en route or have already been successfully delivered. This becomes an issue if our system is closed-loop and therefore relies on acknowledgements. Congestion collapse from undelivered packets is the result of bandwidth being wasted by packets that are dropped before arriving at their destination. Packet cache timeouts are critical in opportunistic networks and it is clear that a node should not occupy its storage with a packet for longer than is necessary.

Fragmentation based congestion collapse is the result of fragments of packets being discarded en route to the destination, such packets are then discarded by the receiver as they will not reassemble. We use the DTN bundle ideology for message dissemination as we ensure the whole message is transferred at each hop and the bundle is not fragmented. The Haggle model [8], unlike the DTN model, sends bundles as packets. Because the Haggle bundles are fragmented, Haggle has to avoid fragmentation based congestion collapse by sending most of the packets as self describing fragments.

Congestion collapse from increased control traffic can result from an increase in fraction of control data transmitted on a congested path. As dynamic topology of opportunistic networks often lead to extensive nodes reporting, it is essential

that we filter our observations and balance the benefits of keeping knowledge fresh with the cost of doing so. We therefore have to keep the control overheads below a fraction of any contact opportunity.

The final form of Internet congestion collapse is the result of stale or unwanted packets caused by large delays between the request and receipt of data. It is easy to see that stale or unwanted packets are likely to occur frequently in opportunistic networks. One way to overcome this is by taking the open-loop approach and setting the cache timeouts in order to clear out stale and unwanted data.

In addition to the congestion collapse already identified, opportunistic networks suffer from in-network congestion. This is due to each node being potentially a source, a sink and/or a router. This is challenging as a node has a conflict of interests when it is required to forward packets for others, while having packets to forward for itself. This conflict of interests can be seen in Figure 3 and is discussed further in Section IV-B2.

Internet congestion control mechanisms are typically sender or receiver driven and closed-loop as they rely on delivery acknowledgements. In contrast, opportunistic networks require in-network congestion control and should be open-loop as this best addresses lack of information available to the senders and receivers, and the delay and cost associated with reporting. This section shows that a new and a different type of congestion control protocol that interacts with the routing system in the network is needed. This is described in the Section IV

## IV. CONTEXT / SOCIALLY AWARE FORWARDING ALGORITHM (CAFÉ) FOR OPPORTUNISTIC NETWORKS

We briefly describe architectural and functional overview of our protocol.

We have designed and built our Context Aware Forwarding Algorithm (CAFé) around two core components the Contact Manager and the Congestion Manager that work together in order to select which packet should be sent next and to which nodes. The Congestion Manager is concerned with the Availability metric of network nodes while The Contact Manager is concerned with the Forwarding Heuristic used at the nodes. All nodes contain two packet buffers, the Forwarding Buffer and the Sender Buffer, in order to allow a fair distribution between sending the nodes own data and forwarding the data from others. Each node, on meeting, exchanges their availability information, statistics about their interactions with other nodes and their locally calculated centralities. From the information received and calculated from the beacon messages the Contact Manager and the Congestion Manager are able to make forwarding decisions.

When a node receives a contact beacon the Contact Manager updates the contact duration, contact frequency and the last seen timestamp for each contact, while allowing for interference as discussed in our previous work [9]. The Contact Manager also updates the neighbours centrality (including degree, similarity and betweeness) and the neighbors contact set (including statistics for each contact), in order to calculate the

Forwarding Heuristic. The beacon also contains information about the percentage of available buffer and average delay for each neighbour that are updated in the Congestion Manager as Figure 1 illustrates.
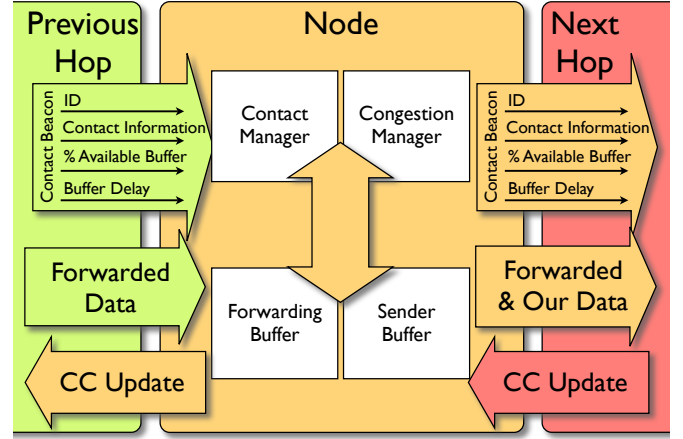


Fig. 1: Overview of the Context / Socially Aware Forwarding Algorithm for Opportunistic Networks

While the nodes buffer is not empty, the node calculates the contacts forwarding heuristic for the each packet in its buffer. This heuristic is then weighted based on the contacts availability information. All the first hop neighbours are compared and the packet is offloaded to the best node in terms of both availability and connectivity.

### A. Forwarding Heuristic

Our forwarding heuristic is based on a utility driven approach that combines smoothed locally calculated centrality (Equation 1), a filtered common neighbour metric (Equation 2), and an interaction strength between two nodes (Equation 3). $Sim_x(D)$ refers to similarity of contacts between node $x$ and destination $D$. Similarity utility $SimU_y(D)$ and tie strength utility $TSU_y(D)$ of node $y$ for delivering packet to destination $D$ are given in 2 and 3. Aggregate utility in 4 is obtained as a sum of Similarity, Centrality and TS utility.

$$d^+U = \frac{Sd_t^+(y)}{Sd_t^+(y) + Sd_t^+(x)} \quad (1)$$

Where $Sd_t^+(y)$ refers to EWMA smoothed out-degree of node $y$ at a given time $t$ based on the current out-degree observation $d_t^+$ and the previous smoothed out-degree value $Sd_{t-1}^+$.

$$SimU_y(D) = \frac{Sim_y(D)}{Sim_y(D) + Sim_x(D)} \quad (2)$$

$$TSU_y(D) = \frac{TS_y(D)}{TS_y(D) + TS_x(D)} \quad (3)$$

$$Util_y(D) = SimU_y(D) + d^+U + TSU_y(D) \quad (4)$$

We improve the common neighbour metric integrated in SimBetTS [10], by only informing neighbouring nodes about the nodes that have strong values of connectivity frequency and duration (Equation 5), we therefore reduce the quantity of data exchanged between nodes.

$$Sim_y(D) = |C_y \bigcap C_D| \qquad (5)$$

Many different centrality metrics have been identified and used in graph theory to index vertices. This value represents a degree of importance in the network, what attribute is chosen depends on what is actually important. We observe that the use of a centrality index, such as Degree Centrality[11] or Ego-Betweenness[10] Centrality will add value to our forwarding heuristic. In our previous work [1] we identify that by smoothing Degree Centrality we increase the size of centrality subsets, resulting in a more even spread of load (Equation 6).

$$Sd_t^+(x) = \lambda \cdot d_t^+(x) + (1 - \lambda) \cdot Sd_{t-1}^+(x) \qquad (6)$$

[12] identified that by analysing mobile network connectivity you are able to extract social interactions - identifying friends as people who are long duration contacts; acquaintances as people who are contacts more frequently, but for shorter periods of time; and strangers as very loosely associated contacts. It is clear that the stronger a neighbour's association with the destination is, the greater the probability they will be able to deliver the packet, SimBetTS [10] called this tie strength.

$$TS_x(y) = \frac{f(y)}{F(x) - f(y)} + \frac{d(y)}{D(x) - d(y)} + \frac{l(y)}{L(x) - l(y)} \qquad (7)$$

In Equation 7 $f(y)$ is the contact frequency experienced with node $y$, $F(x)$ is the total frequency of contacts with node $x$, $d(y)$ is the contact duration experienced with node $y$, $D(x)$ is the total duration of contacts with node $x$, and $l(y)$ is the contact duration experienced with node $y$, $L(x)$ is the total duration node $x$ has been a part of the network.

### B. Availability

In our previous work [1] we identified and proposed the addition of two congestion control attributes with the intention of favouring nodes that are able to both receive and retain the packets sent to them. In this section we expand on these attributes.

*1) Retentiveness:* We define retention as the nodes ability to retain the packets that are sent to them. This is an important attribute to consider because of the store and forward nature of opportunistic networks. We see Retentiveness as the percentage of remaining storage capacity. Our congestion control deters the use of nodes that have lower levels of availability and promoting the use of nodes which have a slightly less desirable forwarding heuristic, but have greater level of storage Retentiveness.

$$Av(y) = \lambda \cdot \frac{B_f^{t-1}(y)}{B_c^{t-1}(y)} + (1 - \lambda) \cdot \frac{B_f^t(y)}{B_c^t(y)} + \sigma \cdot \sqrt{1 - \lambda^2} \cdot W \quad (8)$$

Equation 8 shows our method for predicting the contact's remaining buffer. $\lambda$ is the weight that identifies the degree of response, $\sigma$ is the standard deviation of the buffer levels, $W$ is a random number with zero mean and equal variances; and $B_f(y)$ and $B_c(y)$ are node $y$'s free buffer and buffer capacity respectively, where $t$ represents time.

$$F_y(D) = Util_y(D) \cdot Av(y) \qquad (9)$$

Equation 9 shows that the forwarding heuristic $F_y(D)$ is the result of taking the social network utility $Util_y(D)$, adding the receptiveness utility and weighting the overall utility based on the predicted percentage of remaining buffer $Av(y)$. Because a node weights each node by its availability, when a contacts buffer is high, the probability of forwarding data to that node will be low. If the node itself it congested, its forwarding requirements are relaxed making a wider range of possibilities for a next hop.
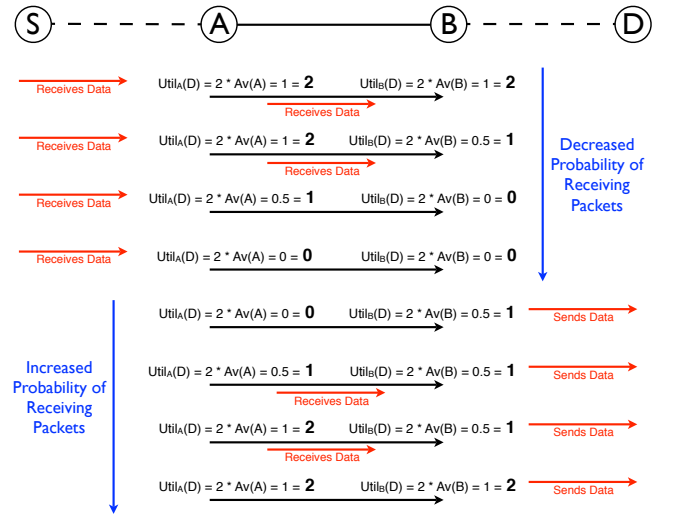


Fig. 2: An example of weighting the forwarding heuristic by Retentiveness

Figure 2 shows an example of congestion control between nodes S and D, using only Equation 9. The upper half of Figure 2 shows that initially node A receives and forwards packets to node B and that node B is unable to forward these packets from the outset - this may be due to node B having a high forwarding value which is not contestable (i.e. node B is more likely to have direct contact with the destination), or due to no neighbouring node being currently available, either due to being out of range or due to having no remaining storage. Eventually node B becomes an undesirable next hop forcing node A's retention to depreciate, this process will continue through the network, either back to the sender or to a node in the network that is able to branch the flow off along another route.

The lower half of Figure 2 shows the events that occur when node B is able to forward packets after a period of disconnection. The first packet that node B sends allows it

to become a potential next hop, prompting node A to utilise it. Node B stays in a frail state until it has cleared the backlog of packets, this behaviour benefits the network, because in the event a node with a similar forwarding heuristic but a lower state of congestion connects with node A the stress is relieved from node B.

The problem with this solution is that congestion partitions the network (Figure 3) - the core of the network becomes congested and the peripheral nodes in the network are essentially secluded. This problem is only made more complex by the fact that the routers between the source and the destination are generating traffic and therefore competing for resources.



Fig. 3: Partitioned network as a consequence of congestion

Figure 3 illustrates an example in which the nodes labeled S cannot forward packets to node D as the nodes labeled B are congested with packets they are generating for themselves. This identifies that introducing the retention metric as a congestion prevention does not solve the problem caused by intermediary nodes congesting themselves while in a disconnected state - this is resolved by queueing generated and third-party packets separately.

*2) Receptiveness and Rate Adaptation:* We define receptiveness as the nodes capacity as a link, we measure this capacity as the exponential moving average delay this node adds to the packets travelling by its aid.

$$Rec^t(x) = \frac{1}{\alpha \cdot Rec^{t-1} + (1-\alpha) \cdot T_d(P_x^t)} \quad (10)$$

Equation 10 shows that we calculate receptiveness as the inverse of 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, the duration of the delay will also depend on the level of connectivity in the network as well as the degree of congestion.

$$SR(x,y) = B^f(y) \cdot \frac{1}{Rec(x) + Rec(y)} \cdot Rec(y) \quad (11)$$

We calculate the Sending Rate (Equation 11) based on the remaining buffer of $y$, and the Receptiveness (rate of throughput) that node $y$ is experiencing relative to node $x$. We use relative in-network delay to determine the percentage of the available buffer of our contacts that we can use.

## V. SIMULATION AND EVALUATION

We have implemented SimBetTS [10], FairRoute [2], and an early prototype of our algorithm CAFé in our own custom built trace driven opportunistic simulator. We have chosen SimBetTS and FairRoute as benchmark algorithms because

SimBetTS is a state of the art social forwarding algorithm, while FairRoute focuses on load distribution fairness in social opportunistic networks.

Our prototype currently integrates Forwarding Heuristic and the Retention Availability described in Section IV. For our real connectivity traces we use Infocom 2005 dataset [13]. The dataset contains logs for 41 Bluetooth devices (iMotes) which were carried by attendees for 3 to 4 days. We have randomly selected 10 different groups of 5 publishing nodes and 20 subscribing nodes. All 20 nodes are interested in the data published by all 5 publishing nodes during the whole time of experiment. Each experiment for each group is run with all three different protocols. All connections are assumed to be capable of transferring data at the peak Bluetooth data rate, and each sending node generates enough data per second to constantly consume one of these links. All nodes have storage limited to 1000MB. When the sending nodes drop packets because their sending buffer is full we count this as a non-queued packet to differentiate from a packet that is dropped in transit and those dropped due to connectivity.

Figure 4 shows the average number of sent, dropped and delivered packets from the 10 different sender / destination combinations for Café (a), FairRoute (b) and SimBetTS (c) over 70 hours. Café manages to send 1.8 times more packets than SimBetTS and 4.4 times more than FairRoute. All the algorithms have the same level of demand, enough data per second to constantly consume one link, this builds up at the source when the node is disconnected, the reason Café has a higher level of sent packets is due to the way the algorithm brings nodes with a high forwarding inline with node with a lesser value then the high ranking node becomes less available, distributing the load of the network better. Because Café is managing to send more packets it is essentially able to deliver packets that would otherwise be dropped at the source.

Figure 5 shows Box-plots of the levels of dropped, delivered, forwarded and the success ratio for each algorithm, the results are not normalised due to the potentially significant variations in connectivity over the runs. On average Café has 1.4 times better success ratio, delivers 1.4 times more packets and drops 1.3 times less packets than SimBetTS. On average Café delivers 2 times more packets than FairRoute, but drops 0.3 times more packets. We also observe that Café increases the number of forwards in order to achieve this and we evaluate this further in Figure 6.

Figure 6 shows the average delay, the mean absolute deviation of the delay, the average and mean absolute deviation of the hop count of delivered packets. On average packets take 1.3 and 1.2 times longer to arrive when using Café rather than SimBetTS and FairRoute respectively. Packets traversing the network by the guidance of Café also on average travel via 1.3 times more hops than both SimBetTS and FairRoute. We believe that the additional delay is a justifiable cost as the algorithm has a higher success ratio, higher number of sent and delivered packets and fewer dropped packets.

Figure 7 shows per-destination success ratio for all the three algorithms for one of the 4 day experiments, with 5 sources
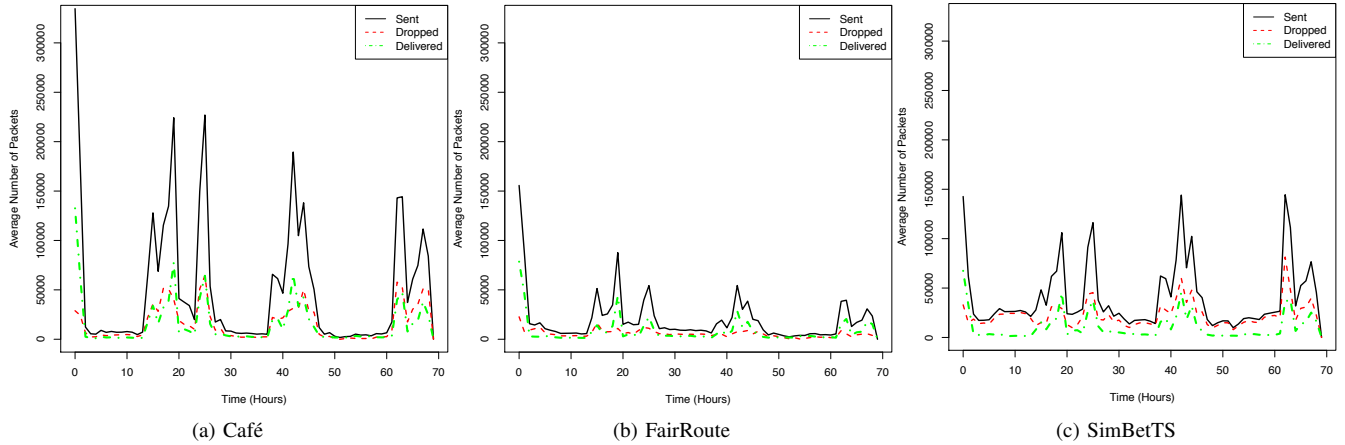
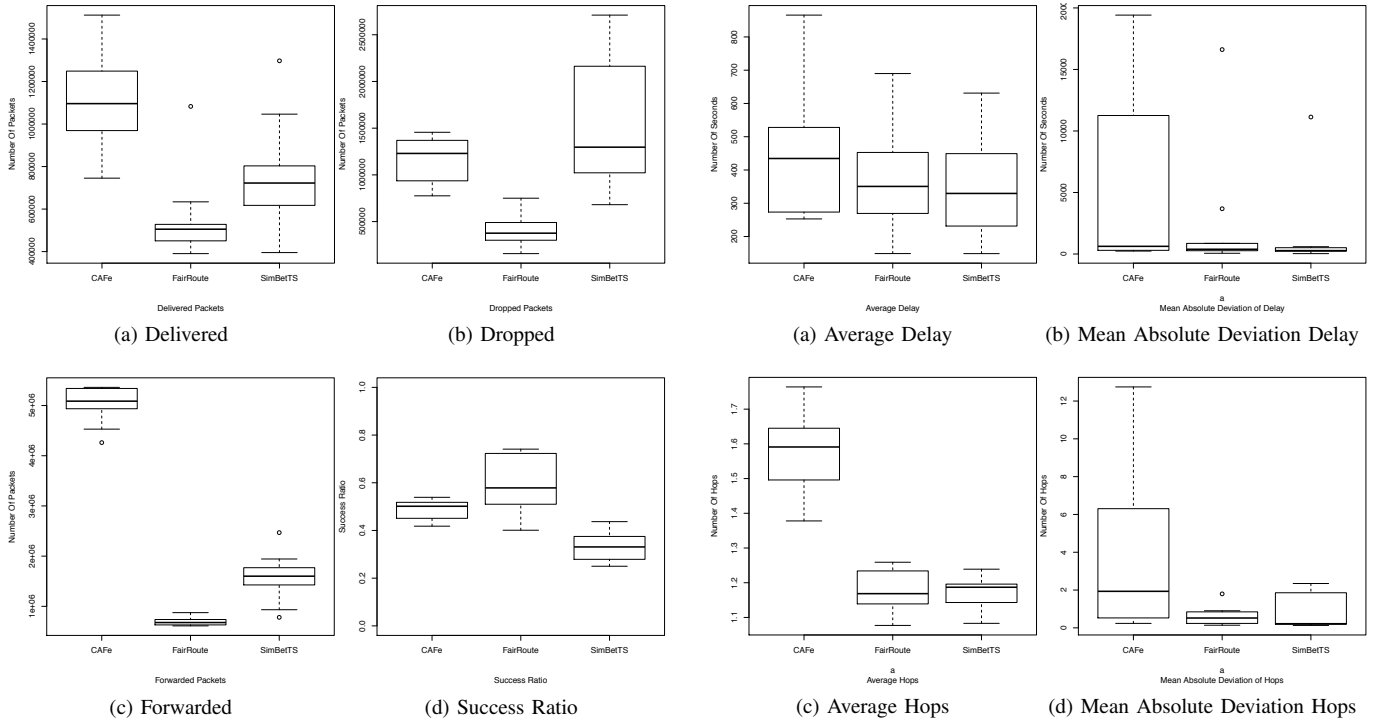Fig. 4: Average sending rate, number of dropped and forwarded packets per simulated hour



Fig. 5: Boxplots to show success ratio, along with the levels of dropped, delivered and forwarded packets



Fig. 6: Boxplots to show the average delay, mean absolute deviation delay and the average and mean absolute deviation hop count of delivered packets

and 20 destinations. We have run the same per destination analyses for 10 experiments (40 days), with randomly chosen five sources and twenty destinations. When comparing the per-destinations success ratio of the three algorithm for the worst performing 20% of the nodes we were seeing an improvement of over 200% for Cafe over FairRoute, and an improvement of above 140% for Cafe over SimBets.

## VI. Conclusion and Future Work

This paper has identified a number of objectives a congestion control algorithm has to satisfy in order to function within an opportunistic network. We have proposed and presented initial evaluation of CAF congestion aware forwarding algorithm that adaptively chooses the next hop based on contact history and statistics, and buffer statistics. In our extensive trace driven experiments we showed that congestion control is an essential component to be included in the transfer of data
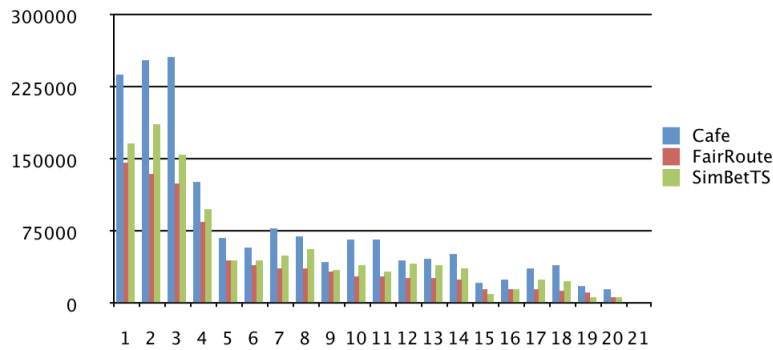
Fig. 7: Per-destination success ratio for one of the 4 day experiments

in opportunistic networks. Our future work will comprise of further development and evaluation of the availability metric presented in this work but also focus more on adding in-network delay prediction in order to extend our model.

One of our concerns is to do with the overheads associated with the control packets used to select a next hop. In our previous work [1] we suggest filtering this information prior to disseminating it. Our future work will be concerned with finding the optimal balance between throughput and overheads. We aim to incorporate our fully distributed ad hoc reputation mechanism [22] into our forwarding heuristic (Section IV) in order to allow detection and isolation of malicious nodes performing a blackhole attack.

## REFERENCES

[1] A. Grundy and M. Radenkovic, "Decongesting opportunistic social-based forwarding," in *Seventh Annual Conference on Wireless On demand Network Systems and Services (WONS)*, Kranjska Gora, Slovenia, 2 2010.

[2] J. Pujol, A. Toledo, and P. Rodriguez, "Fair Routing in Delay Tolerant Networks," in *Proceedings of 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 *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*. ACM, 2007, p. 234.

[4] M. May, V. Lenders, G. Karlsson, and C. Wacha, "Wireless opportunistic podcasting: implementation and design tradeoffs," in *Proceedings of the second ACM workshop on Challenged networks*. ACM, 2007, p. 82.

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

[6] 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.

[7] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Transactions on Networking (TON)*, vol. 7, no. 4, pp. 458–472, 1999.

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

[9] A. Grundy and M. Radenkovic, "Routing in Wireless Networks of Varying Connectivity," in *Proceedings of the 2009 Fifth International Conference on Wireless and Mobile Communications-Volume 00*. IEEE Computer Society, 2009, pp. 18–23.

[10] E. Daly and M. Haahr, "Social network analysis for information flow in disconnected Delay-Tolerant MANETs," *IEEE Trans. Mob. Comp*, vol. 8, no. 5, pp. 606–621, 2009.

[11] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay tolerant networks," in *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*. ACM New York, NY, USA, 2008, pp. 241–250.

[12] N. Eagle and A. Pentland, "Reality mining: sensing complex social systems," *Personal and Ubiquitous Computing*, vol. 10, no. 4, pp. 255–268, 2006.

[13] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD data set cambridge/haggle (v. 2009-05-29)," Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle, May 2009.

[14] C. Lochert, B. Scheuermann, and M. Mauve, "A survey on congestion control for mobile ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 7, no. 5, pp. 655–676, 2007.

[15] V. Raghunathan and P. Kumar, "A counterexample in congestion control of wireless networks," *Performance evaluation*, vol. 64, no. 5, pp. 399–418, 2007.

[16] K. Chen and K. Nahrstedt, "EXACT: An explicit rate-based flow control framework in MANET (extended version)," *University of Illinois at Urbana-Champaign*, 2002.

[17] K. Sundaresan, V. Anantharaman, H. Hsieh, and R. Sivakumar, "ATP: a reliable transport protocol for ad-hoc networks," in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. ACM New York, NY, USA, 2003, pp. 64–75.

[18] S. Burleigh, E. Jennings, and J. Schoolcraft, "Autonomous congestion control in delay-tolerant networks," *American Institute of Aeronautics and Astronautics*, 2007.

[19] Y. Li, L. Zhao, Z. Liu, and Q. Liu, "N-Drop: congestion control strategy under epidemic routing in DTN," in *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*. ACM, 2009, pp. 457–460.

[20] A. Lindgren and K. Phanse, "Evaluation of queuing policies and forwarding strategies for routing in intermittently connected networks," *Proc. of IEEE COMSWARE*, 2006.

[21] M. Seligman, K. Fall, and P. Mundur, "Storage routing for dtn congestion control," *Wireless Communications and Mobile Computing*, vol. 7, no. 10, pp. 1183–1196, 2007.

[22] S. Zakhary and M. Radenkovic, "Reputation-based security protocol for manets in highly mobile disconnection-prone environments," in *Seventh Annual Conference on Wireless On demand Network Systems and Services (WONS)*, Kranjska Gora, Slovenia, 2 2010.