

Exploiting Social Preferences for Congestion Control in Opportunistic Networks

Dimitrios-Georgios Akestoridis, Nikolaos Papanikos, and Evangelos Papapetrou
Department of Computer Science and Engineering, University of Ioannina, Greece
Email: {dakestor, npapanik, epap}@cs.uoi.gr

Abstract—It has been observed that opportunistic networks exhibit a highly unbalanced traffic load distribution, mainly because of the heterogeneity in mobility and the greedy routing decisions, leading to packet drops due to storage constraints. The existing strategies rely either on fairness techniques or on diverting traffic to alternative routes in order to control congestion. The result is a dilemma between performance and fairness. In this work, we introduce a congestion control mechanism that provides a tunable trade-off between efficiency and fairness. We rely on the social preferences of the nodes for dynamically tuning the aforementioned trade-off. Our simulations show that the proposed algorithm achieves high delivery ratio, combined with low end-to-end delay and routing cost, without sacrificing fairness under high traffic load.

Index Terms—Congestion control, social preferences, fairness, opportunistic networks, delay-tolerant networks.

I. INTRODUCTION

Connectivity in opportunistic networks appears in the form of contacts between nodes, where an end-to-end path between the source and the destination is rarely available [1]. As a result, the prominent routing approach is to make greedy decisions on a contact basis, according to a utility metric that captures the fitness of the contacting nodes for delivering and/or forwarding a packet to its destination [2]. Several researchers [2]–[7] have provided evidence suggesting that the greedy routing approach usually results in a highly unbalanced loading of the nodes, i.e. a small fraction of nodes handles most of the traffic load. This phenomenon has been observed for various routing protocols in a variety of traces from experimental networks [3]–[6]. The main reason for this behavior is the heterogeneity of node utilities, i.e. the presence of few nodes with high utility values. In many cases this is due to the heterogeneity of node mobility [2], which is also present in opportunistic networks that are driven by human mobility [1]. In the latter case, the network exhibits the structural characteristics of social networks [8], such as the presence of few nodes that act as hubs. The result is again a highly unbalanced distribution of utilities among the nodes. This is because most of the metrics used in social network analysis depend on the importance of a node in the network. Finally, note that the aforementioned phenomena aggravate when a destination-independent utility is used [7], i.e. a utility that captures the node’s ability to act as a forwarder regardless of the destination. Keeping in mind the limited storage resources of a node, it becomes evident that implementing a congestion control mechanism is of paramount importance in order to minimize the degradation of delivery rate due to packet drops. This is especially true for single-copy

routing protocols since a packet drop completely erases that packet from the network.

In this work, we focus on the challenging task of providing an efficient congestion control mechanism for single-copy routing schemes. The proposed strategies can be classified based on the type of information that they use in order to make decisions [9]. However, another useful approach is to classify them based on the performance characteristics that each scheme targets, besides the maximization of delivery rate. More specifically, two types of strategies can be identified; (a) those that aim at enhancing fairness, and (b) those that aim at avoiding storage congestion with the minimum possible routing cost. The rationale in the first category is that energy is a limited resource as well, therefore fairness is required in using the nodes as relays [10]. However, fairness is achieved at the expense of delay. The algorithms of the second category [11] are more performance-oriented. The idea is to exploit alternative paths in order to bypass the area of congestion. However, this approach is usually poor in fairness, while at the same time induces an increased cost in terms of transmissions in order to reduce delay and vice versa.

The motivation of this work has been the observation that a profitable trade-off between the two aforementioned approaches can be achieved. That is, to devise a congestion control mechanism that, besides maximizing delivery rate, it can achieve low end-to-end delay at the expense of a reasonably low cutback in fairness. To this end, we propose the *Congestion Control with Adjustable Fairness (CCAF)* algorithm. CCAF incarnates the described functionality in a generic manner that can be incorporated to virtually any utility-based routing protocol. Our contributions are as follows:

- We propose a congestion control mechanism that provides, through a tunable parameter, a trade-off between fairness and end-to-end delay without impacting and in some cases improving the delivery ratio (Section III).
- We provide a method for dynamically adjusting the tunable parameter based on the social preferences of the nodes (Section IV).
- We show the validity of our approach using different traces from real networks. More specifically, the overall performance is significantly improved while fairness is practically unaffected when most needed, i.e. when the offered traffic load increases (Section V).

In the rest of the paper, Section II provides an overview of congestion control strategies, while Section VI summarizes our

findings and provides some directions for future work.

II. RELATED WORK

Several approaches have been proposed in the literature for overcoming the problem of storage congestion in opportunistic networks [9]–[13]. They can be broadly categorized to those applicable to multi-copy routing protocols and those applicable to single-copy schemes. In the multi-copy case, the proposed approaches take advantage of the fact that packet drops can be tolerated to some extent since another node may carry a replica of the dropped packet. Therefore, the focus is on dynamic replication techniques and/or efficient dropping policies.

The problem of congestion control in the single-copy case is far more challenging since the delivery rate automatically degrades whenever a node is forced to drop a packet. The proposed strategies fall into two categories [9]: (a) those using an economic model, and (b) those using network traffic levels to make their decisions. The most representative algorithm of the first category is Autonomous Congestion Control (ACC) [11]. In this algorithm, the remaining storage space of each node is regarded as money and the packet transmissions as financial activities. A node will accept to carry a packet only if it has enough storage space and the potential risk of carrying it is low. The risk of accepting a packet depends on its residual time-to-live, the projected buffer growth rate of the node, and the previous risks that it took. Therefore, in order to apply this approach, each node has to keep a detailed record of its buffer occupancy history. A typical algorithm of the second category is FairRoute [10]. Besides the proposed routing mechanism, FairRoute uses a queue control module to distribute the traffic load fairly among the nodes of the network. The idea is to use the queue length of each node as an equivalent of its social status, so that each node will receive packets only from nodes of equal or higher social status. Although this approach may not imply a reduction in terms of delivery ratio due to the diversity of paths among the nodes [14], it may increase the average delay significantly.

Besides the aforementioned classification of algorithms, an interesting and orthogonal approach is to distinguish them using the performance characteristics that they focus on. For example, FairRoute focuses on fairness, while ACC focuses on diverting traffic to alternative paths. In this latter category, Storage Routing [12] relies on the mitigation of packets from the congested nodes to their neighbors, in order to avoid as many packet drops as possible. This method may provide significant throughput, but unfortunately it is extremely costly in terms of packet transmissions. Another approach, called CAFé [13], utilizes buffer and network statistics to predict whether the transmission of a packet will later cause congestion or not. However, a non-trivial fine-tuning of the parameters is required for collecting statistics.

III. CONGESTION CONTROL WITH ADJUSTABLE FAIRNESS

A. The Need for Adjustable Fairness

As previously mentioned, several researchers have illustrated that, under many and diverse conditions, opportunistic networks

suffer a highly unbalanced distribution of traffic load among the nodes. As a result, it is possible for some nodes to exhaust their limited storage and energy resources. Traditional approaches that handle congestion by diverting traffic to alternative paths are not oriented towards economizing on both storage and energy. Therefore, the most effective approach for tackling the problem is to increase the fairness in the network. By distributing the traffic load among all the nodes in the network, we can achieve a notable reduction in the total number of packet drops. Unfortunately, if the packets are distributed only based on fairness criteria, an increase in delay is the most likely outcome [15]. The reason is that, under this strategy, a node misses opportunities to forward a packet to a better node just to keep the fair allocation of resources. Although several alternative paths to reach the destination may exist [14], the delivery time may increase significantly. We make the observation that following fairness-related incentives is useful, however it is desirable to combine them with utility-based routing principles. In this way, it is possible to increase the likelihood of delivering packets through shorter paths, thus reducing the traffic load in the network and alleviating congestion. Therefore, our approach is to be as fair as possible, while taking advantage of the high-utility nodes wisely.

B. Combining Forwarding and Custody Acceptance Criteria

In order to produce the functionality that we just described, our first goal is to determine the *importance of each forwarding step*. Let us consider the case where node i carries a packet and it encounters node j . Furthermore, let $u_i(d)$ and $u_j(d)$ denote the utilities of i and j , while d denotes the destination of the packet. Obviously, when j has a much higher utility value than i , the likelihood of successful delivery is significantly increased if the packet is moved from i to j . However, we also have to take into account the cases where both $u_i(d)$ and $u_j(d)$ are close to 0. The most unfavorable situation is when a node carries packets for which it has a utility value equal to 0. In such occasions, it is critical to forward these packets to nodes that have a higher utility value, even if the difference is marginal. This is to increase the chance of finding other nodes that will be able to deliver the packets to their destinations. To capture all the aforementioned cases, we use a normalized utility value $U_{i,j}(d)$ which is defined as:

$$U_{i,j}(d) = \frac{u_j(d) - u_i(d)}{u_j(d) + u_i(d)} \quad (1)$$

Note that $U_{i,j}(d) = 1$ only when $u_i(d) = 0$ and $u_j(d) > 0$. Additionally, $U_{i,j}(d) \rightarrow 1$ when $u_j(d) \gg u_i(d)$, which means that j is much closer to the destination than i . On the contrary, $U_{i,j}(d) \rightarrow 0$ indicates that both i and j have almost equal utility values. Observe that $U_{i,j}(d)$ can be used in any utility-based routing protocol, without affecting its performance. This is because the traditional requirement for forwarding a packet, i.e. $u_j(d) > u_i(d)$ can be replaced by $U_{i,j}(d) > 0$.

The next step in our approach is to define a criterion for accepting the custody of a message. Therefore, we examine at which extent the buffer of a node is occupied. To this end, it

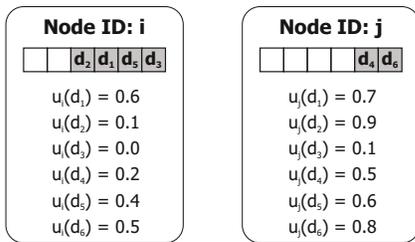


Fig. 1. Two nodes in contact with known utility values and buffer contents.

is reasonable to use the remaining storage space so that when i and j meet, they both know how many more packets each one can carry, avoiding unnecessary packet drops. However, since it is possible that nodes have different storage capabilities, we use the *normalized residual space*. More specifically, the normalized residual space of node i is defined as R_i/B_i , where R_i denotes the remaining storage space and B_i the total storage capacity. The intuition behind this approach is to protect nodes with high storage capacity from receiving an excessive number of packets, which would result in draining their batteries faster.

The final step of our approach is to combine the forwarding and custody acceptance criteria in order to decide when to transfer a message from node i to node j . More specifically, upon contact of node i with node j , a packet destined for d is forwarded from i to j provided that:

$$(U_{i,j}(d) > 0) \wedge \left(\frac{R_j}{B_j} > \left(1 - (U_{i,j}(d))^\delta\right) \frac{R_i}{B_i} \right) \quad (2)$$

and the remaining storage space R_j is enough to store the packet. The rationale behind using $1 - (U_{i,j}(d))^\delta$ in the storage-related criterion is to be able to *relax* or *enforce* it based on the importance of the forwarding opportunity. The tunable parameter δ allows us to adjust our approach either towards high efficiency or absolute fairness. To understand this, consider the case that $\delta \in (0, 1)$. This is equivalent to further relaxing the forwarding condition when the normalized utility value $U_{i,j}(d)$ is high. On the other hand, when $\delta > 1$, we mostly concentrate on maintaining the load balancing among the nodes. In general, δ can be used to fine-tune the performance of the network in order to meet the needs of a certain application.

To further illustrate the intuition behind (2), let us consider the example depicted in Fig. 1 where two nodes, i and j , carry some packets for the destination nodes d_1, \dots, d_6 . Without any congestion control mechanism, node i would forward all the packets that it carries to node j , simply because node j has a higher utility value for each one of the destinations. According to the strategy proposed in the congestion control module of FairRoute [10], node i would refrain from forwarding packets to node j when the traffic load is distributed fairly. However, in our approach, we are able to relax the requirement for fair traffic load distribution in order to favor the forwarding of a packet when j is a far better candidate than i for delivering the packet to its destination, i.e. $U_{i,j}(d) \rightarrow 1$. For example, we would not forward the packet destined for node d_1 , because both nodes have almost equal utility values, but we may forward the packet destined for node d_2 , since node j has a much higher utility

```

upon contact of node  $i$  with node  $j$  do
1: obtain information regarding  $U_{i,j}$ ,  $R_j$ , and  $B_j$ 
2: for each  $pkt \in Buffer_i$  do
3:   if  $U_{i,j}(pkt.dst) > 0$  then
4:     if  $R_j \geq pkt.size$  then
5:        $NRS_i \leftarrow R_i/B_i$ 
6:        $NRS_j \leftarrow R_j/B_j$ 
7:        $MF \leftarrow 1 - pow(U_{i,j}(pkt.dst), delta)$ 
8:       if  $NRS_j > MF \times NRS_i$  then
9:         forward  $pkt$  to node  $j$ 
10:         $R_i \leftarrow R_i + pkt.size$ 
11:         $R_j \leftarrow R_j - pkt.size$ 
12:      end if
13:    end if
14:  end for
15: end for

```

Fig. 2. Pseudo-code of CCAF.

value than node i .

To summarize, in the case where $U_{i,j}(d) \rightarrow 0$, the packet will be forwarded to node j only if it also contributes at balancing the traffic load between the two nodes. When $U_{i,j}(d) \rightarrow 1$, we may relax the load balancing to forward the packet to node j , thus notably increasing the chance of successful delivery. All the packets for which $U_{i,j}(d) = 1$ will be forwarded to node j , as long as j has enough space to store them. It should be noted that when $\delta \rightarrow \infty$, CCAF resembles the queue control of FairRoute and when $\delta \rightarrow 0$ it operates as if there was no congestion control mechanism, but without overfilling the buffers of the nodes. A pseudo-code of our algorithm is given in Fig. 2.

IV. TOWARDS AN EFFICIENT SELF-CONFIGURING METHOD

To illustrate the ability of CCAF to provide a trade-off between fairness and efficiency, we performed simulations for five different values of δ and various levels of traffic load. The Reality Mining [16] trace was used, while SimBet [17] was the underlying routing protocol. More details about the simulation setup can be found in Section V-A. Fig. 3 presents the obtained results. As expected, assigning a low value to δ results in fast packet delivery, while using a high value increases fairness among the nodes. Note that a profitable trade-off is present when $\delta \in (0.25, 1]$. More specifically, the best performance for both delay and delivery ratio is witnessed in the aforementioned range of values regardless of the offered traffic load. Furthermore, the cutback in fairness is minimal when the offered traffic load is high, which is an important feature since fairness is more critical in such conditions.

It is clear that a real-life implementation requires a dynamic self-configuring method for δ . To this end, we visualize δ as a parameter that describes the *social preferences* [18] that each node has towards allocating its resources for the sake of others. When $\delta \rightarrow 0$, the nodes behave altruistically because they are contributing all of their resources to help others achieve communication. Selfish nodes would choose a high δ value, so that they could avoid using their resources to help other nodes.

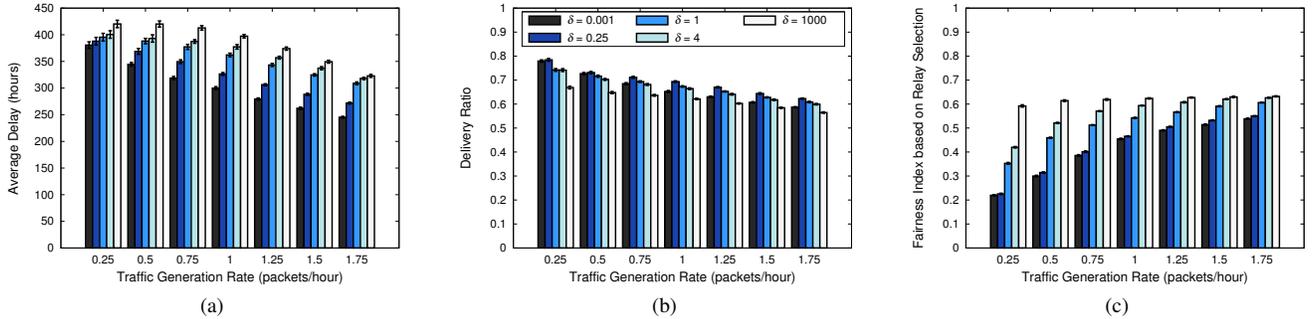


Fig. 3. The impact of δ : (a) Average Delay, (b) Delivery Ratio, and (c) Fairness Index based on the number of times that a node is used as a relay.

Recall that, in the previous experiments, the best performance results when $\delta \in (0.25, 1]$, which in general corresponds to a *cooperative* behavior. However, we make the observation that a more efficient solution may result from a mixture of selfish, prosocial, and altruistic nodes. Therefore, each node should choose its own δ value in accordance to its social preferences, which could change over time. Our approach for providing a self-configuring method for δ is based on the idea that the nodes would prefer to allocate their resources according to their social ties. In other words, each node would prefer to be selected as a relay node for packets that are destined for its friends rather than strangers. From the network’s point of view, such a behavior is beneficial in terms of delivery ratio, delay, and overhead. This is because each node carries packets for smaller amounts of time, since delivering a message to a friend is easier than delivering it to a stranger. To produce the aforementioned behavior, we use *similarity* [17] to measure “friendship”. Then, we define $\delta_{i,j}(d)$ as:

$$\delta_{i,j}(d) = e^{-(\text{Similarity}_j(d) - \text{Similarity}_i(d))} \quad (3)$$

Note that, $\delta_{i,j}(d) \rightarrow 0$ when the “friendship” between nodes j and d is much closer than that between i and d . In other words, nodes will act altruistically for their friends and selfishly for strangers. This user-oriented approach is also suitable for cases of social selfishness in opportunistic networks [19], where most of the nodes are willing to exchange packets only with those whom they have social relationships. Assuming that the willingness information of each node is available, we can use it to tune the δ parameter, replacing the similarity metric.

V. EXPERIMENTAL EVALUATION

A. Simulation Setup

To evaluate the performance of CCAF we have developed a custom event-driven simulator, which is capable of processing contact-based mobility traces. In the following experiments, SimBet [17] is used as the underlying routing protocol. More specifically, we compare three implementations for congestion control: (a) Autonomous Congestion Control (ACC) [11], (b) FairRoute’s congestion control module [10], and (c) CCAF with the self-configuring method proposed in Section IV. Moreover, we simulate SimBet without any congestion control mechanism and use it as a reference protocol.

We selected two of the most widely used real-world mobility traces for our simulations, i.e. Reality Mining [16] and Cambridge [20], which are available in the CRAWDAD archive [21]. The Reality Mining data set consists of contacts from 97 students and faculty members at the MIT, over the course of 9 months. The Cambridge data set contains contacts from an experiment that lasted 11 days, involving 36 students from the University of Cambridge.

The network traffic load of each simulation consists of fixed-sized packets that are generated with a random pair of source and destination nodes. Because some users did not participate throughout the whole experiment, each node can be the source or the destination only for packets that were generated during its presence in the network. We examined seven different scenarios, where the network traffic load ranges from low to high, based on the characteristics of each data set. Each node could store up to 20 packets in its buffer, that are processed in a FIFO and Drop Front order.

For each scenario, we simulated 25 trials and report average values as well as the 95% confidence intervals for a variety of evaluation metrics. To avoid statistical bias, the results were collected after a warm-up and before a cool-down period, each of which lasts as much as 20% of the total simulation time, so that the network would be in its steady state.

B. Simulation Results

Fig. 4 illustrates the performance of all algorithms for both traces under various levels of offered traffic load. More specifically, we report: (a) the average delay normalized to the delay of SimBet without congestion control, (b) the delivery ratio, and (c) the overhead ratio, i.e. the number of transmissions for each generated packet. The first metric is indicative of the impact of congestion control mechanisms, which is the increased average delay, since routes of longer delay are used in principal in order to reduce packet drops. Nevertheless, as we can observe (Figs. 4a and 4d), CCAF achieves a much lower normalized average delay compared to ACC and FairRoute regardless of the traffic load. As expected, FairRoute presents the highest average delay because it focuses on achieving a fair traffic load distribution among the nodes. Figs. 4b and 4e clearly demonstrate the impact of storage constraints on the network’s performance as well as the need for congestion

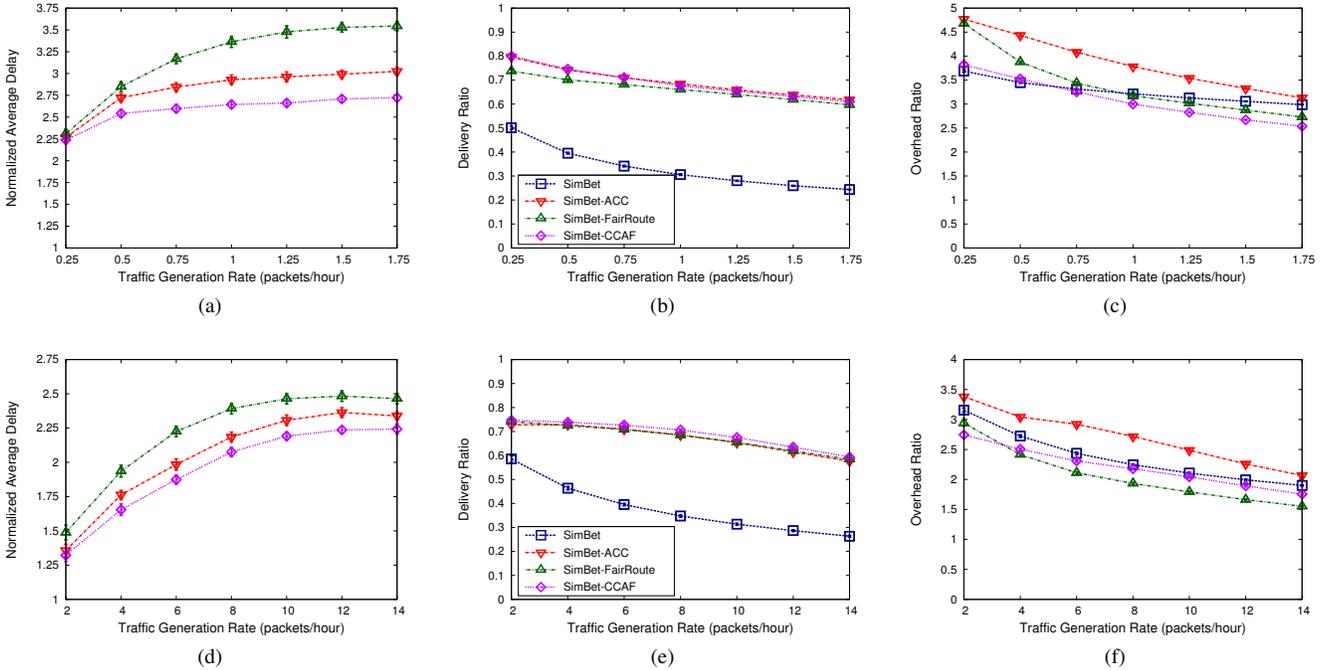


Fig. 4. Efficiency evaluation on the (a)–(c) Reality Mining and (d)–(f) Cambridge data sets.

control. Note that all the congestion control strategies deliver about the same number of packets under any traffic load. However, CCAF is able to deliver them a lot faster, by breaking the fair traffic load distribution to take advantage of the most critical forwarding opportunities. Finally, ACC clearly induces the highest cost in the network (Figs. 4c and 4f) while CCAF performs fewer transmissions. This is a clear indication of a more successful strategy in identifying a good relay node. At the same time FairRoute presents a performance similar to CCAF, however this is the result of its conservative approach regarding forwarding. A confirmation of this is its increased delay.

In Figs. 5a and 5d, we use *Jain's Fairness Index* [22] to capture the fairness of each scheme in selecting nodes as relays. Note that the Fairness Index is bounded between 0 and 1, where 0 corresponds to definite unfairness and 1 represents absolute fairness. The figures clearly indicate FairRoute's success in providing high levels of fairness, however this happens at the expense of delay. On the other hand, CCAF presents an interesting performance; under low traffic load its fairness index is similar to that of plain SimBet, while under high traffic load its fairness level significantly increases and approaches that of FairRoute. In other words, the fairness in choosing a relay increases as the burden of being a relay increases (due to the increased traffic load). A similar behavior is witnessed for ACC. However, ACC clearly involves a higher burden of packet transmissions (Figs. 4c and 4f). Interestingly enough, as illustrated in Figs. 5b and 5e, in many occasions the top 10 nodes perform more forwards under ACC than under plain SimBet. Finally, Figs. 5c and 5f are another indication of the need for congestion control. Plain SimBet is forced to

an excessive number of packet drops while the other three algorithms manage to significantly reduce them. FairRoute presents a small advantage because of its smoother traffic load distribution. However, again this advantage comes at the cost of increased delay. Furthermore, note that besides the slightly increased number of packets compared to FairRoute, CCAF still remains the most efficient approach not only in terms of delay and overhead but also in terms of delivery ratio.

VI. CONCLUSIONS AND FUTURE WORK

As it has been pointed out, traditional utility-based routing protocols overuse a small subset of nodes and therefore result in many unnecessary packet drops when the resources are limited. To overcome this situation we made the observation that, unlike other approaches, it is critical to combine utility-based routing principals with custody acceptance criteria that are oriented towards fairness. In this way, we were able to propose a congestion control algorithm that provides a trade-off between high efficiency and absolute fairness. This trade-off can be fine-tuned to match the desired network performance through the tunable parameter δ . Furthermore, we provided a dynamic self-configuring method which relies on the social preferences of the nodes, so that each node could set its own δ value accordingly. We showed that our approach results in significant performance gains with a limited cutback in fairness.

In future work, we intent to investigate other methods to define δ , based on local and network-wide information. A possible extension, which could be used for multi-copy routing protocols, is to rely on an estimation of the number of replicas in the network. This approach would favor the spread of packets with few replicas in the network, which in turn would increase

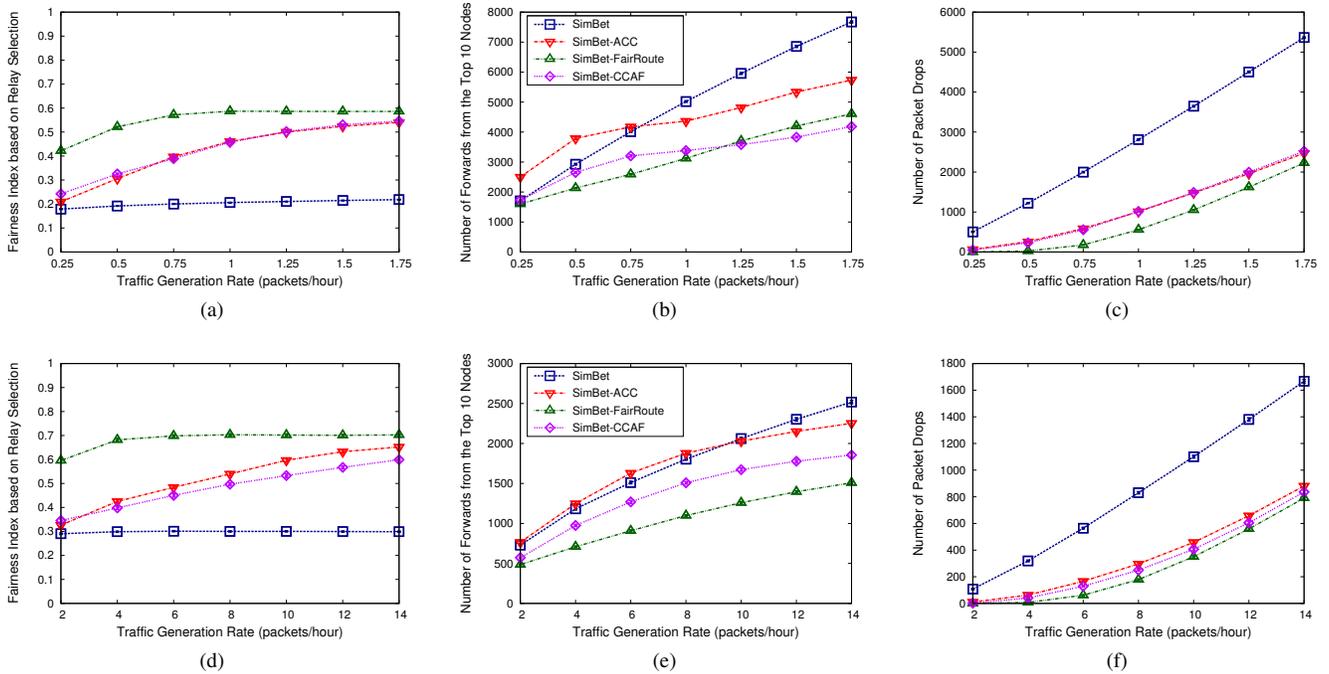


Fig. 5. Fairness evaluation on the (a)–(c) Reality Mining and (d)–(f) Cambridge data sets.

their chance to meet their destinations significantly. Another possibility for determining δ could be the battery level of each device, so that we can avoid node overloading in cases where there are energy constraints.

REFERENCES

- [1] M. Conti and S. Giordano, "Mobile ad hoc networking: Milestones, challenges, and new research directions," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 85–96, 2014.
- [2] T. Spyropoulos, R. N. B. Rais, T. Tulletti, K. Obraczka, and A. Vasilakos, "Routing for disruption tolerant networks: taxonomy and design," *Wireless Networks*, vol. 16, no. 8, pp. 2349–2370, 2010.
- [3] 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 (MobiHoc '08)*, 2008, pp. 241–250.
- [4] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, "Delegation forwarding," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, 2008, pp. 251–260.
- [5] A. J. Mashhadi, S. B. Mokhtar, and L. Capra, "Fair content dissemination in participatory DTNs," *Ad Hoc Networks*, vol. 10, no. 8, pp. 1633–1645, 2012.
- [6] B. Soelistijanto and M. Howarth, "Traffic distribution and network capacity analysis in social opportunistic networks," in *Proceedings of the 8th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '12)*, 2012, pp. 823–830.
- [7] T. Spyropoulos, T. Tulletti, and K. Obraczka, "Routing in delay-tolerant networks comprising heterogeneous node populations," *IEEE Transactions on Mobile Computing*, vol. 8, no. 8, pp. 1132–1147, 2009.
- [8] T. Hossmann, T. Spyropoulos, and F. Legendre, "A complex network analysis of human mobility," in *Proceedings of the 2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2011, pp. 876–881.
- [9] B. Soelistijanto and M. P. Howarth, "Transfer reliability and congestion control strategies in opportunistic networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 538–555, 2014.
- [10] J. M. Pujol, A. L. Toledo, and P. Rodriguez, "Fair routing in delay tolerant networks," in *Proceedings of the 28th IEEE International Conference on Computer Communications (INFOCOM '09)*, 2009, pp. 837–845.
- [11] S. Burleigh, E. Jennings, and J. Schoolcraft, "Autonomous congestion control in delay-tolerant networks," in *Proceedings of the 9th AIAA International Conference on Space Operations (SpaceOps '06)*, 2006.
- [12] M. Seligman, K. Fall, and P. Mundur, "Alternative custodians for congestion control in delay tolerant networks," in *Proceedings of the 2006 SIGCOMM Workshop on Challenged Networks (CHANTS '06)*, 2006, pp. 229–236.
- [13] M. Radenkovic and A. Grundy, "Congestion aware forwarding in delay tolerant and social opportunistic networks," in *Proceedings of the 8th International Conference on Wireless On-Demand Network Systems and Services (WONS '11)*, 2011, pp. 60–67.
- [14] 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 (IMC '07)*, 2007, pp. 161–174.
- [15] A. Mtibaa and K. A. Harras, "Fairness-related challenges in mobile opportunistic networking," *Computer Networks*, vol. 57, no. 1, pp. 228–242, 2013.
- [16] N. Eagle and A. Pentland, "Reality mining: sensing complex social systems," *Personal and Ubiquitous Computing*, vol. 10, no. 4, pp. 255–268, 2006.
- [17] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant MANETs," in *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '07)*, 2007, pp. 32–40.
- [18] W. B. G. Liebrand, "The effect of social motives, communication and group size on behaviour in an n-person multi-stage mixed-motive game," *European Journal of Social Psychology*, vol. 14, no. 3, pp. 239–264, 1984.
- [19] Q. Li, S. Zhu, and G. Cao, "Routing in socially selfish delay tolerant networks," in *Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM '10)*, 2010, pp. 1–9.
- [20] J. Leguay, A. Lindgren, J. Scott, T. Friedman, and J. Crowcroft, "Opportunistic content distribution in an urban setting," in *Proceedings of the 2006 SIGCOMM Workshop on Challenged Networks (CHANTS '06)*, 2006, pp. 205–212.
- [21] "CRAWDAD: A community resource for archiving wireless data at dartmouth." [Online]. Available: <http://crawdad.cs.dartmouth.edu/>
- [22] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," Digital Equipment Corporation, Tech. Rep. DEC-TR-301, 1984.