

Efficient and Adaptive Congestion Control for Heterogeneous Delay-Tolerant Networks

Milena Radenkovic, Andrew Grundy

Department of Computer Science, University of Nottingham, Jubilee Campus, Wollaton Road, Nottingham, NG8 1BB, UK

Abstract

Detecting and dealing with congestion in Delay Tolerant Networks (DTNs) is an important and challenging problem. Current DTN forwarding algorithms typically direct traffic towards more central nodes in order to maximise delivery ratios and minimise delays, but as traffic demands increase these nodes may become saturated and unusable. We propose CafRep, an adaptive congestion aware protocol that detects and reacts to congested nodes and congested parts of the network by using implicit hybrid contact and resources congestion heuristics. CafRep exploits localized relative utility based approach to offload the traffic from more to less congested parts of the network, and to replicate at adaptively lower rate in different parts of the network with non-uniform congestion levels. We extensively evaluate our work against benchmark and competitive protocols across a range of metrics over three real connectivity and GPS traces such as Sassy[44], San Francisco Cabs[45] and Infocom 2006[33]. We show that CafRep performs well, independent of network connectivity and mobility patterns, and consistently outperforms the state-of-the-art DTN forwarding algorithms in the face of increasing rates of congestion. CafRep maintains higher availability and success ratios while keeping low delays, packet loss rates and delivery cost. We test CafRep in the presence of two application scenarios, with fixed rate traffic and with real world Facebook application traffic demands, showing that regardless of the type of traffic CafRep aims to deliver, it reduces congestion and improves forwarding performance.

Keywords:

Delay Tolerant Networking, Congestion Control, Multi-Path Forwarding, Load-Dependant Routing, Open-Loop

1 Introduction

Over the recent years the pervasiveness of mobile computing devices has increased significantly and the possibility of communication without an existing network infrastructure has become a reality. Hsu and Helmy [18] showed that real world traces of different university campus wireless networks node encounters are sufficient to build a connected relationship graph that can be successfully used for data transmission despite the absence of contemporaneous end-to-end paths. Routing in these networks is a challenging problem and has been primarily concerned with providing maximum throughput and minimal delays while typically assuming unlimited storage and transfer bandwidth i.e. using forwarding algorithms that greedily select best connected nodes as their next hop [14, 28]. As mobile devices have limited resources, key nodes in the network quickly become congested and unusable and cause even more disconnections and even lower delivery rates.

Newly emerging work on congestion control in DTNs attempts to rectify this by proposing adaptive forwarding [14, 31, 15, 30] or adaptive replication management [39, 38, 26, 36, 37] techniques. In our earlier work we proposed an adaptive forwarding protocol, Café [14, 31, 15, 30], that diverts the load away from its conventional centrality-driven path by considering congestion heuristics that route the traffic away from the congesting areas. This approach is particularly suitable to social opportunistic networks as they have been shown to exhibit the path explosion phenomenon [9]. The main drawback of adaptive forwarding is that it does not decrease the total level of traffic in the network but only redirects it. This is suboptimal at times when finding alternative non-congested parts of the network is not possible. Adaptive replication management techniques [39, 38] can decrease the traffic in response to congestion, but they cannot adaptively offload the traffic from more congested to less congested parts of the network. This means that they cannot efficiently deal with scenarios when congestion affects some regions of the network (e.g. hotspots, roadside units) and not the others.

In this work we propose to unify adaptive forwarding and adaptive replication into a common congestion control framework for DTNs (CafRep) that manages to both decrease the load on the network and offload the traffic to the parts of the network that are less congested. We achieve this by using a local based implicit heuristic based on contact and resource statistics that extends our previous work on adaptive forwarding [10-12]. More specifically, we propose to dynamically combine three types of heuristics: node centrality and contact analysis driven heuristics that exploits contact relationships to allow optimal directionality and delivery probability of a node; node resource driven and ego network driven heuristics to adapt to the nodes and parts of the network's buffer availability, delays or congesting rates. As a result,

our new framework adaptively changes forwarding and replication behaviour to best manage tradeoffs across multiple contact and resource attributes of nodes in real network scenarios with different mobility, and connectivity patterns.

Previous work on resource aware DTN protocols has either focused on the homogeneous networks where the authors assumed nodes to have the same amount of resources on their devices [51] or looked at only one connectivity trace [15, 39]. However, it is not realistic to assume network homogeneity as mobile devices can be used in different ways, move in different ways and do not consume their resources at the same rates. In this paper we explore how a more intelligent resource aware model, reflective of the heterogeneity of devices' resources, and non-uniform connectivity patterns, and application demands can improve data delivery and dissemination in the network.

Section 2 discusses the lessons learned and shortcomings of DTN forwarding, replication, load distribution and congestion control approaches further, together with state of the art work from related areas such as MANET congestion control, resource pooling, Peer-to-Peer (P2P), as well as an algorithmic game theory perspective of selfish forwarding and routing.

Section 3 specifies the particular challenges in opportunistic delay tolerant networks that prevent successful data delivery and identifies our criteria that guide our proposal for congestion control framework for adaptive replication and forwarding - CafRep. We present an analytical model that identifies fundamental problems regarding traffic distribution and shortest path forwarding heuristics in relation to the traffic flow and the price of anarchy. Modelling the behaviour of opportunistic networks is a challenging because of time varying network topology and congestion, and thus we integrate the concepts of a time varying networks [12] and dynamic flows [13]. We describe the design space of our proposal, give multi-layer architectural overview of our conceptual model and CafRep pseudo code. We identify and describe the core congestion signals, heuristics and techniques that are at the core of our proposal and allow adaptive dissemination of messages throughout the network such that we address our criteria i.e. allow spreading the traffic across multiple paths whilst avoiding congested regions and minimising delays and network overheads. We extend our previous work by discussing the impact of a number of different CafRep utility weighting schemes on the protocol performance over varying network topologies.

Section 4 describes our evaluation methodology. As nodes' encounter patterns can greatly differ for social and vehicular networks, it is important to evaluate our CafRep protocol across different real traces from CRAWDAD [1] in order to gain better understanding of our protocol performance. Section 4 begins by discussing the heterogeneity of the three chosen real connectivity and GPS traces. We then motivate the use of two different application scenarios: publish subscribe podcasting and real Facebook application.

Section 5 discussed our extensive evaluation of CafRep against some of the major state of the art DTN protocols including a benchmark DTN forwarding algorithm Prophet [23], an adaptive forwarding algorithm Café [31], a fixed replication algorithm Spray and Focus (SF) [37] and two adaptive replication algorithms Encounter Based Routing (EBR) [26] and Retiring Replicants (RR) [38]. We consider seven different metrics for the performance analysis of the protocols including success ratio, delay, buffer availability, packet loss rates, number of forwarded and replicated packets, and delivery cost. We show that our protocol outperforms all four other algorithms across majority of metrics across the three chosen heterogeneous traces that have different mobility and connectivity patterns. We identify differences in protocols' performances across the different traces and discuss multiple reasons for causing that. We briefly describe the results with the different Facebook traffic types that show that regardless of the type of traffic CafRep aims to deliver, it reduces congestion and improves forwarding performance in the network.

Section 6 concludes the work and presents the open questions for future work.

2 Related Work

2.1 DTN Forwarding, Replication, Load Distribution and Congestion Control

This section discusses the most recent advances in message replication and forwarding for DTNs. Early work in this area focuses primarily on the challenge of reducing the delivery latency and cost with the underlying assumption of unlimited transfer and storage capacity [40, 23, 17, 8, 36, 37]. The majority of research in congestion control for DTNs is concerned with buffer management [6, 22, 24, 34], but recent developments have been concerned with replication management [34, 39, 30] and distribution [26, 28, 15].

Encounter Based Routing (EBR) proposed in [26] is a quota based replication protocol that aims to forward more copies of a message to nodes that are better connected. The introduction of limited resources and increasing traffic demands cause key components in the network to become congested, which in turn leads to messages being dropped.

Retiring Replicas (RR) [38] is an adaptive replication protocol that adjusts a nodes' copy limit based on the locally perceived global level of congestion. RR 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. This work assumes a uniform network with random waypoint mobility. In reality the networks are likely to

be non-uniform and the level of congestion may vary between different regions of the network. This work has not proposed how the network adjustments would compensate for differing local conditions.

In Cafe [15] we proposed and examined several combined social and resources heuristics in order to detect congested parts of the network and move the traffic away towards less congested parts. These heuristics include social, delay and buffer metric of nodes and their ego networks. The total combined utility function we propose is at the core of our adaptive forwarding protocol that is dynamic and flexible as it operates as a pure social (contact driven) protocol at times of low congestion but is highly resource driven at times of high congestion. We show that our single copy adaptive forwarding protocol achieves better performance in comparison to multi-copy protocols such as spray and wait (SW)[36], spray and focus [SF][37], Prophet[20] and epidemic protocols in terms of decreased delays, higher availability of nodes and higher success ratios.

In [30] we built an interest-driven P2P content dissemination overlay on the top of our congestion aware forwarding protocol. Both caching and forwarding policies are decided based on the interest, availability, social closeness and numbers of interested nodes. Our results show that our adaptive overlay manages to maintain high success ratio of answered queries, high availability of intermediary nodes and short download times for a P2P file casting application running in the face of increasing number of file publishers and topic popularity.

FairRoute[28] argues that considering only contact histories to define contact duration, frequency and interaction strength cannot achieve balanced traffic distribution. FairRoute proposes nodes queue length to be evaluated in order to allow nodes to only forward to contacts with a bigger queue size, but this does not avoid congesting popular nodes and leads to packets being dropped.

Storage Routing (SR)[34] avoids congested nodes dropping packets by sending a set of messages out to neighbours with available storage, when buffer capacity is free these are retrieved. This protocol only temporarily alleviates congestion as it relies on contacts that are present not suffering from congestion themselves.

Autonomous Congestion Control (ACC)[6] implements congestion control by apply a financial model to buffer space management, in order to propagate buffer utilisation stress backwards through the network to the source nodes. This work does not overcome intermediary / source nodes filling their own buffers and isolating peripheral nodes.

[38] propose DA-SW (Density-Aware Spray-and-Wait), that is a measurement-oriented variant of the spray-and-wait [36] 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 did not address the impact of their static measurement window (30sec) on the performance of their system. This work does not consider dealing with resource constraints such as node buffers, bandwidth and energy consumption.

2.2 Resource Pooling

[42] believe that the natural evolution of the Internet should be to harness multipath-capable end systems in order to achieve resource pooling. [9] show that opportunistic networks typically exhibit the path explosion phenomenon. In our work, the nodes benefit from pooling the capacity of their many and varied contacts, making effective use of the network resources available to them. If traffic is spread across the resources of a node's many and varied contacts in the right way, with the right reaction to the right congestion signals from the network, then traffic can quickly move away from congested regions.

2.3 Peer-to-Peer Replication Strategies

We observe the similarities between content dissemination in opportunistic networks and in the related field of peer-to-peer (P2P) content dissemination and storage systems. Although P2P networks operate in the application level we believe lessons can be learned from the work in this area. In applications such as BitTorrent, peers replicate each other's data in order to increase data availability [32], also resulting in the pooling of the upload capacity of many network nodes [42].

[32] studied the problem of replica placement in a P2P system intending to optimise availability and/or the number of replicas. [32] show that centralised control of resource placement is a NP-hard problem and that if the control is fully decentralised the peers selfishness can greatly alter the results leading to performance inequities that can render the system unreliable and thus ultimately unusable [32, 29]. The most common approach to P2P replication is the random distribution of copies [7, 5]. [4] analyse how many randomly placed replicas are required to achieve a desired level of availability. [32] argue that replication should not be random, but be based on cliques of peers replicating each other's data, limiting the selfishness of the participants.

Table 1 presents a summary of the techniques discussed in this section in terms of seven criteria. Note that none of the existing approaches have support for adaptive forwarding and replication, fully localized, social and resource aware congestion control in DTNs.

	EBR	Fairroute	Storage Routing	DA-SW	RR	ACC	P2P	Cafe
Adaptive forwarding	No	Yes	Limited (offloading)	Yes	No	No	Yes	Yes
Adaptive Replication	Yes	No	No	Yes	Yes	No	Yes	No
Congestion Aware	No	No	Limited	Limited (relies on the abacus)	Yes	Yes	Yes	Yes
Social/ Contact Aware	Yes	Yes	Limited	Yes	Yes	No	No	Yes
Resource Aware	No	No	Yes	No	No	Yes	Yes	Yes
Fully localised	Yes	Yes	Yes	No (relies on the abacus)	Yes	No	Yes	Yes
Delay Tolerant	Yes	Yes	Yes	Yes	Yes	No	No	Yes

Table 1 Overview of the techniques that deal with efficient data dissemination in DTNs

3 Congestion Aware Forwarding and Replication for DTNs

3.1 Challenges

Varying mobility patterns, topology changes, disconnections and resource restrictions pose many challenges for the design and implementation of congestion aware data transmission in DTNs. This section systematically outlines particular challenges that motivate our criteria described in Section 3.2 that guide our proposal.

Distributed decisions: The limited connectivity and fragmented nature of opportunistic networks mean that it is neither efficient to obtain and maintain a global level of knowledge of the network nor to rely on closed-loop (acknowledgment based) techniques for any decisions. We aim to allow each device in the DTN network to act independently and base their decisions on the limited localised knowledge with the aim to enable better performance of the whole network.

Limited resources: As buffer capacity, transfer bandwidth and battery life are limited, the messages can only be transferred successfully if the efficient coordination between the devices competing for resources is provided.

Network density and localised surges in traffic: Node connectivity in DTNs is sporadic and islands of connectivity may range in size, from sparse to dense. Sparse networks present limited forwarding options at any given time, while dense networks are prone to suffering from transmission collisions due to wireless interference. We aim to propose a protocol that could work well across these highly different network scenarios.

3.2 Criteria

In response to the challenges described in Section 3.1, we focus on the following research question:

How can disconnection prone nodes with different mobility and connectivity patterns and with limited resources, communicate in an efficient, adaptive and robust manner when there are a large number of data sources and destinations?

We address this question by identifying the following criteria that guide our proposal.

Efficiency: We define efficiency in terms of providing support for minimising the traffic latency and optimising utilisation of network resources. More specifically, for DTN forwarding algorithms traffic latency is a key concern, as the freshness of data is important and it is persistently challenged by disconnections and congestion of intermediaries that prevent efficient store-carry-and-forward routing. Similarly, DTN forwarding algorithms identify a subset of better connected nodes to which they forward the majority of the traffic. As these nodes become overloaded, forwarding to a lower ranked node may lead to more even spread of the network load in the network and lower congestion rates, but also to increased number of intermediaries for the forwarded messages. We aim to provide a mechanism that manages to dynamically balance more even utilisation of network nodes while keeping the traffic delays and number of forwarded packets low.

Adaptation: DTN nodes typically have self organised fully distributed behaviour which means that they base their decisions on the knowledge gathered in their local environment. As we aim to optimise network wide behaviour based on nodes localised decisions, the question of how the individual nodes can get the feedback about the remote network state and affect it becomes highly challenging. We aim to provide an

adaptive mechanism that achieves network-wide optimisations by only fully relying on localised node aggregate heuristics and multi-dimensional metrics.

Robustness: We define robustness in terms of providing increased packet resiliency and collaboration between the nodes. More specifically, due to nodes limited battery resources and high mobility, our aim is to investigate strategies for ensuring that the effect of dropped packets is minimal on the final delivery rates. As the closed-loop acknowledgment-based loss recovery is ineffective in DTN environments, we look into packet replication techniques that can help with this. Similarly, in order to avoid greedy localised node-only behaviour that leads to decreased intermediary resources and lower network-wide performance, we aim to provide a mechanism that carefully balances opportunistic usage of intermediary resources and cooperative behaviour that leads to improved end to end delivery rates.

3.3 Analytical Model

We model the network as a temporal graph $G=(V,E)$ because the connectivity of the network E and the state of the nodes V change over time. We model each of these as time series and depict the vertices as $V=\{V_t: t \in T\}$ and the edges as $E=\{E_t: t \in T\}$, where t is a member of the time series T . We assume that connectivity is bidirectional and therefore the edges of the graph are undirected, the edge connecting nodes $u,v \in V_t$ we denote as $\{u,v\} \in E_t$.

The traditional representation of a path in a graph that is commonly used to depict the route that a message is transmitted along is an alternating sequence of vertices and edges. In this work we model a path as a sequence of resource locations a message occupies, where each index represents a particular time interval e.g. $P=(x_0,x_0,e_1,x_1,\dots,e_k,x_k)$ where $e_i=\{x_{i-1},x_i\}$. Intuitively the best route is the path with the smallest resource cost, i.e. the path with the minimum number of transfers and shortest storage time. Each second a message occupies storage or transfer bandwidth it adds cost to a network resource, and it can be more efficient for messages to travel via a greater number of hops with smaller in-network delays in order to arrive at the destination, rather than to be kept in storage waiting for a high demand resource to become available adding to the messages latency. We aim to carefully manage the trade-off between increasing the storage occupation and increasing the hop count. Using alternative paths is particularly suitable for social DTNs due to the path explosion phenomenon [9].

The demand of a resource is dependent on the set of forwarding demands. The amount of demand for a resource X at a time t is given by:

$$D_X^t = \sum_{S \neq X \in V} \sum_{D \neq X \in V} F_{SD}^t(X)$$

where $F_{SD}^t(X)$ denotes the number of paths between source node S and destination D that contains the resource X at time t . Which paths contain X is influenced by the effects of the forwarding strategy.

Each resource X in the network can have a different stress level at any given time t as a result we denote stress as $ST_x^t = \frac{D_x^t}{C_x}$ which is a measure of demand D_x^t of a resource X at a given time t against the capacity for that resource C_x . Packet loss occurs when $D_x^t > C_x$ i.e. the level of demand D_x^t of a resource X at a given time t is greater than the resource capacity C_x . The total cost of delivering a single copy of a message is the sum of all storage and transmission occurrences in the lifetime of a given message along a path between the source and the destination. As messages can be replicated, one or more copies of the message are transmitted, each following an independent path. The number of paths is limited by a replication limit (M) and the cost of delivery for a replicated message is the sum of all path costs in the replication path set.

In order to impartially evaluate the inherent load distribution of a network we observe the stress effects of a uniformly non-biased forwarding strategy that selects the next hop randomly. The result of selecting the next hop randomly is congruent with a random walk and therefore nodes that are better connected are more likely to receive messages. Forwarding based on a heuristic which favours a node because of its connectivity puts well connected nodes in even greater demand than when simply randomly selected. Connectivity observations such as: how recently a node has encountered the destination, the duration of connectivity a node has experienced with the destination, how frequently a node encounters the destination and a nodes betweenness or degree centrality are each used as heuristics within forwarding strategies. The better the node is connected the greater the probability it has of fulfilling the criteria of this type of forwarding strategy. Forwarding based on connectivity is a method of seeking the shortest path. Shortest path routing is greedy and it can be the best solution in a network with no opposing traffic. However, this is not realistic because it assumes no delay. Delay is relative to the level of congestion experienced by a resource X at a given time t , and is a measurement of the current demand D and buffered demand B over the number of available outlets for the resource X at a given time divided by the degree centrality of the node X at time t $d^t(X)$ which can be denoted as $delay^t(X) = \frac{D_x^t + \sum m \in B^t}{d^t(X)}$.

We define resource consumption as subgraph $G'=(V',E')$ where the set of vertices are defined as the set of vertices that have a demand greater than 0 $V'=\{\forall v_t \in V : D_v^t > 0\}$ and the set of edges is defined as the set of edges that have a demand greater than 0 $E'=\{\forall e_t \in E : D_e^t > 0\}$. Network utilization can be measured as difference between the available resources G and the consumed resources G' depicted as $U = \frac{|G'|+\|G'\|}{|G|+\|G\|}$ where $|G|$ and $|G'|$ denote the size of the set of vertexes for G and G' , and $\|G\|$ and $\|G'\|$ denote the size of the set of edges for G and G' . The relationship between F and G is constrained such that $0 \leq U \leq C_G$ Where C_G is the total capacity of the graph. The forwarding criteria H influences the size of F (the number of paths that wish to use these resources) based on its utilization of G . Given two forwarding criteria H and H' each utilizing the network to the value of U and U' respectively, H has a greater capacity than H' if $U > U'$. H is therefore more effective as F can utilize more of the elements of G .

Existing DTN forwarding algorithms direct traffic towards the most desirable next-hop nodes as this is the optimal solution when the network is free. When the traffic demands increase, these nodes become inundated and this is made even more challenging as the flow of traffic is unpredictable and has a tendency to accumulate in some regions of the network. [2] shows that the price of anarchy is unbounded if the optimal forwarding solution is based purely on minimizing delay in a flow-independent model, but that this can be improved by forwarding based on how congested the resource is in addition to the delay cost, resulting in the price of anarchy being at most 4/3 provided the cost functions are all linear or $d/\log d$ if the cost functions are polynomials.

3.4 Design Space and Multi-layer Overview of CafRep

We briefly describe the design space for CafRep and show different dimensions of possible approaches that can be useful for congestion aware dissemination in DTNs in Figure 1. The vertical axis represents the Direction Influence that refers to the connectivity dimension such as centrality, similarity, interaction strength and other social contexts that identify affiliation such as clique identification and delivery probability used to influence the social forwarding decisions. The two horizontal axes represent the resource dimensions. The multi-path transport approach, such as Caf  [14, 31, 15, 30], introduces methods of avoiding congested regions of the network. The replication restriction technique, such as Retiring Replicants [38], helps to increase network capacity by reducing the in-network occupancy of redundant messages. Our CafRep's design philosophy is to consider the combination of route optimisation with multi-path transport, sending rate restriction and social connectivity.

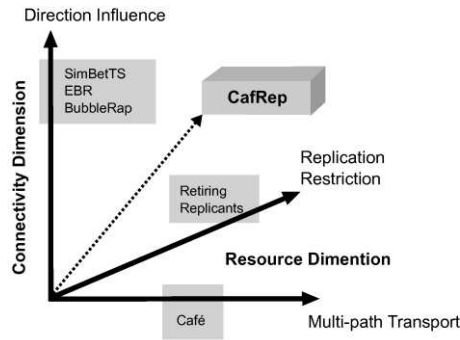


Figure 1. Design Space

CafRep takes a multi-layered approach that is illustrated in Figure 2. The edges in the graph on the Network Layer illustrate connectivity and the vertices represent the nodes. In reality the connections go up and down over a period of time and the network can get disconnected for the majority of time, but for simplicity the time series information has been flattened. We assume the following: source node and destination node belong to the same interest group, multiple paths exist between the two nodes and the socially optimal route is also the most congested path. The Interest Layer, a part of the Application Layer, maps users into areas of interest as each application can have its own topics and interest groups. CafRep Layer combines Congestion Layer and Social Layer in order to broaden the next hop selection criteria allowing nodes with capacity, or less direct routes, to receive messages by monitoring both social and congestion signals and dynamically balancing between them. The Network Layer, on the Physical Layer, illustrates the actual route a message would take through this example network, given the trade-off between shortest path and resource-driven routing, in order to redistribute load to avoid congestion as identified in our criteria.

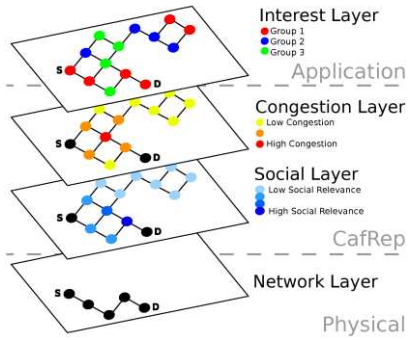


Figure 2. Multi-Layer View

3.5 Congestion Aware Forwarding and Replication: CafRep

3.5.1 CafRep Design Overview

We describe unified adaptive forwarding and adaptive replication management approach into a common congestion control framework for DTN routing, CafRep (Congestion Aware Forwarding and Replication). CafRep works as a local adaptive forwarding and replication 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. It dynamically combines three types of implicit congestion heuristics: **social (contact) driven heuristics** that exploits contact relationships among nodes to allow optimal directionality and delivery probability of a node; **node resource driven heuristics** that aim to adopt to the nodes' buffer availability, delays and congesting rates; and **ego network (regional) driven heuristics** that aim to detect and adapt buffer availability, delays and congesting rates of different parts of the network. Selecting the node that represents the best carrier for the message and deciding on the optimal number of replicas to forward are both multiple attribute decision problems across multiple measures, where the aim is to select the node and number of messages that provide the maximum utility for carrying a certain number of messages. To achieve this we propose $CafRepUtil_D$ heuristic (Formula 1) that is responsible for determining the overall improvement an encountered node represents when compared to the sending node, and deciding on which nodes will be selected as next hop and how many copies of the message are to be disseminated. Formula 1 shows the $CafRepUtil_D$ utility calculation node (X) uses when forwarding a message towards a destination (D) in order to evaluate each encountered node (i) within its contact set (C) as the sum of the set of carefully selected equally weighted utilities of different congestion heuristics:

$$CafRepUtil_D(X) = \sum_{h \in H} Util_h(X) \quad (1)$$

CafRepUtil gets calculated for all contacts of the sending node (node X) and selects the best next hop as the node with the highest CafRepUtil value if its CafRepUtil is higher than the sending node's. The number of messages to be sent to it is decided by Formula 2. If node X has more than one copy of the message ReplRate shows how the number of copies can be divided between the two nodes. This division is dependent on the CafRepUtil utility value of each node; therefore, the division of the replication rate for

the destination d between node X and node $C(X)$ is given by Formula 2 where M is the number of replicas of a message m at node X .

$$ReplRate_D(X) = M \cdot \frac{CafRepUtil_D(C(X))}{CafRepUtil_D(X) + CafRepUtil_D(C(X))} \quad (2)$$

There are seven non-trivial heuristics $h(X)$ that cover different dimensions of the problem and when combined they allow managing a number of trade-offs between different challenges we identified in section 3. For each of the heuristics we define their respective utilities as measurements of their relative gain, loss or equality, calculated as pair-wise comparison between the node's own congestion heuristics and that of the encountered contacts. More specifically, we use a pair-wise comparison matrix on the normalized relative weights of the social and resources heuristics of nodes and their ego networks in the following way.

$$Util_h(X) = \frac{h(C(X))}{h(X) + h(C(X))} \quad (3)$$

The set of our selected congestion heuristics includes: node retentiveness (Ret), node receptiveness (Rec), node congesting rate (CR), and their weighted ego-network counterparts (WEN_{Ret} , WEN_{Rec} , WEN_{CR}), along with a node social heuristic ($Social$), and is given in Formula 4. In this paper, we use SimBetTS as our social heuristic that was introduced in [8].

$$h \in H = \{Ret, Rec, CR, Social, WEN_{Ret}, WEN_{Rec}, WEN_{CR}\} \quad (4)$$

Before we move to describing CafRep algorithm in a greater detail we briefly describe the congestion heuristics used in CafRep for completeness purposes. More detailed description of these heuristics is given in [31][48].

Retentiveness (Ret) refers to the node's available storage for the new packets that are sent to them. Retentiveness is an important attribute to consider because of the store and forward nature of opportunistic DTN networks. Nodes with limited storage, either due to popularity or simply due to more limited hardware constraints, are more susceptible to packet loss. Retentiveness is calculated as an exponentially weighted moving average of a node's remaining storage. Formula 5 shows retentiveness of X is calculated as the sum of all message occupancy subtracted from the node's buffer capacity ($B_c(X)$).

$$Ret(X) = B_c(X) - \sum_{i=1}^N m_{size}^i(X) \quad (5)$$

Receptiveness (Rec) refers to the node's ability to receive packets and forward them on. This is an important observation as increasing in-network delays is an indication that the volume of traffic a node or region is receiving is greater than the bandwidth available to it for offloading. The total current message delay is calculated as the sum of the difference between the time each message in a node's buffer was received and the current time. The delay between receiving a message and forwarding a message is

constrained by the size of the buffer and the bandwidth available for a node to offload the messages. Nodes with large amounts of storage are more susceptible to receiving more messages than they are capable of offloading. Formula 6 shows receptiveness is the total current message delay, calculated as the sum of differences between the current time (T_{now}) and the time each message was received ($M_{received}$).

$$Rec(X) = \sum_{i=1}^N (T_{now} - m_{received}^i(X)) \quad (6)$$

Congesting Rate (CR) refers to a measure of how fast a node is likely to congest, an important observation as it indicates stability. Congesting rate is calculated as the percentage of time a node or region has been congested divided by the average time between fullness periods for the node or region, given in Formula 7. This congestion signal indicates the likelihood of traffic spikes that could cause the message to be dropped. Each node keeps track of the percentage of time it has been full ($T_{\%Full}(X)$) in Formula 7a, and the average time between its fullness periods ($T_{AT}(X)$) in Formula 7b .

$$CR(X) = \frac{T_{\%Full}(X)}{T_{AT}(X)} \quad (7)$$

$$T_{\%Full}(X) = 100 \cdot \frac{T_{FullBuffer}(X)}{T_{TotalTime}(X)} \quad (7a) \quad T_{AT}(X) = \frac{1}{N} \cdot \sum_{i=1}^N T_{iend}(X) - T_{istart}(X) \quad (7b)$$

Note that $T_{FullBuffer}(X)$ is the time the buffer has been and $T_{TotalTime}(X)$ is the total time duration. $T_{istart}(X)$ and $T_{iend}(X)$ are the start and end of between full buffer periods respectively.

Ego Network retentiveness, receptiveness and congesting rate (WEN_{ret} WEN_{rec} WEN_{cr}) refer to congestion heuristics of the node's ego network. Ego network (EN) is defined here as a network consisting of a single node together with the nodes they have encountered and gives each node their own perspective of the network. CafRep allows nodes to aggregate resource observations disseminated by encountered nodes in order to form an ego-network perspective of the network. Ego-network information can be aggregated in many different ways and we have explored a number of models for weighting the contacts within a nodes ego-network in order to improve the accuracy of prediction of the EN congestion levels. This is highly important as it leads to better performance for both forwarding and replication optimizations and making the nodes less selfish. More specifically, we have considered techniques such as simple average, weighted moving average (EWMA) and social weighting of the nodes ego network congestion heuristics. Our experiments have shown that EWMA gives better performance than the simple weighting and the social

weighting across diverse network topologies. Formula 8 shows how we use EWMA to aggregate congestion heuristic information in order to allow the short-term fluctuations to be smoothed out and longer-term trends to be highlighted making it suitable for forecasting. This is updated at each new encounter for each congestion heuristic (h).

$$WEN_h(X, C_i(X)) = (1 - \lambda) \cdot WEN_h(X) + \lambda \cdot h(C_i(X)) \quad (8)$$

λ is a fraction that represents the responsiveness of the smoothing, this is typically around 0.8 [38, 26]. Related state of the art work on congestion control in DTNs [38, 26] that uses implicit local congestion signals to estimate global congestion levels uses EWMA with values of 0.85 or 0.9 for the most recent value a node observes. In [38], each node independently calculates a local approximation of the current congestion level as *the* ratio of drops and replications collected in the last time interval. [38] calculate new congestion value using an estimated weighted moving average (EWMA) with the new congestion view being weighted 0.9 to keep node's congestion view fresh. Similarly, in [26], the locally calculated encounter value (EV) used to track node's rate of encounter uses the exponentially weighted moving average that places an emphasis proportional to 0.85 on the most recent complete Current Window counter. The authors argue that their experiments showed that λ of 0.85 and update interval of around 30 seconds allow for reasonable results in a variety of networks. In our experiments we have chosen a value of 0.8 even though results were similar for 0.85 and 0.9.

CafRep Algorithm: Our CafRep algorithm functions as follows. When a forwarding node (X), meets contacts on its way, it exchanged relevant heuristics and calculates the CafRepUtil of each contact. The CafRepUtil allows the node X to detect how well connected its contact Y is and how available Y and Y 's ego network are in terms of buffer, delay and congesting rate parameters. If there are multiple encounters, node X sorts them in the reverse order (from the highest to the lowest) in terms of their CafRepUtil, calculates respective sending and replication rates for each node and sends the appropriate ReplRate copies of the messages to each one of them invoking the pseudo code for interest-driven content transmission over CafRep given in Figure 4. If the receiving node still has available resources after receiving all the topics of interest, node X will send additional messages it has in its buffer (in a FIFO manner) as long as they are not duplicates. The reason for this is that the node X should offload the messages to nodes with better resources even if they are not directly or indirectly interested in it as they will have better opportunities to forward these messages on. If two contacts are equally suited to forward a message, they each receive half

of the available copies. A contact receives more or fewer messages depending whether it has greater or smaller utility respectfully. The replication rate is rounded to the nearest integer so that in the single copy case messages are propagated provided a minimum of equivalent utility is met. In this way, CafRep adapts the initial number of copies (M) in order to carefully manage the trade-off between the network size and traffic demands.

As CafRepUtil of a node moves up and down, the replication limit grows to take advantage of all available resources, but backs-off when congestion increases, similar to how TCP updates its congestion window [22]. As a result, our protocol is able to replicate at adaptively lower rates in the parts of the network that have low buffer availability, increased node delay and are likely to congest faster. As CafRep node discovers parts of the network with higher buffer availability, lower node delays and slower congesting rates, it replicates at higher rates. Using social utilities together with resource utilities as part of CafRepUtil enables little (or no) replication at high rates on nodes and network regions that available but not on the direct path to the destination. At the same time, using Resources Utilities allows CafRep to replicate at nodes that do not have high social utilities but have high available resources. With the use of combined adaptive forwarding and replication, we allow the sender to stop sending until it finds the right node that it can redirect the traffic to without incurring additional packet loss. Using ego network resource utilities in addition to social utilities allows CafRep to account for a wider view of the network resources and connectivity patterns while allowing differing local conditions. This is very suitable for the DTNs where the

```

Algorithm CafRep
List CafeRepUtils = {}
For each Contact in scan do:
    Contact.NodeHeuristics = exchangeNodeHeuristicsInfo(Contact)
    Contact.CalculateNodeUtilities(Contact.NodeHeuristics)
    Contact.CafeRepHeuristics = exchangeCafeRepHeuristicsInfo(Contact)
    Contact.CalculateCafeRepUtility(Contact.CafeRepHeuristics)
    CafeRepUtils.Insert(Contact)
End For

For each Contact in reverse sorted CafeRepUtils do:
    Contact.SendingRate = CafeRep.ComputeSendingRate(Contact)
    ...Contact.ReplRate = CafeRep.ComputeReplRate(Contact)
    ...Contact.Sent = TransmitContent(Contact, Contact.SendingRate, Contact.ReplRate)
End For

For each Contact in reverse sorted CafeRepUtils do:
If Contact.Sent < Contact.SendingRate then:
    TransmitFifoTo(Contact, Contact.SendingRate - Contact.Sent, Contact.ReplRate)
End For

```

Figure 3 CafRep pseudo code

nodes and parts of the networks can be highly heterogeneous in terms of connectivity and resource

parameters. We argue that CafRepUtil congestion metric is more efficient for heterogeneous DTNs than the metrics that estimates global congestion parameters in DTNs proposed in EBR and RR as it conveys information about which parts of the network are more congested than the others, and can opportunistically use parts of the network that are available while the others are busy. In this way, CafRep enables replication at different rates at different parts of the network that are dissimilar from each other and thus have different social and resource characteristics and patterns.

Content Transmission over CafRep: We describe an example of an interest driven overlay for content dissemination on the top of CafREP in Figure 4. We assume that the content contains topics and each topic has chunks that can be exchanged when the two nodes meet via summary vector. Each chunk has a unique ID. For easier bootstrapping, we assume that some subscribers that are known to the publishers, while others are discovered in an ad-hoc manner. When propagating resource and centrality information, our nodes also propagate their interests by exchanging their interest profiles (topics they are interested in) and summary vectors in order to avoid sending chunks to nodes that are already carrying the same chunks. The Summary Vector contains a list of chunk IDs per topic that a node currently carries per topic.

As described in Figure 3, when two nodes meet, the sending node scans for neighbouring nodes' and calculates their respective relative CafRep Utilities. Each neighbouring nodes' CafRep Utility is weighted and sorted in a list so that the node with the largest weight appears first in the list. When the sending and replication rates are determined for a node, pseudo code in Figure 4 is invoked per node to transmit the

Algorithm: TransmitContent over CafRep

```

Function TransmitContent(Contact, Rate, Repl)
  Vector ContactTI = Contact.obtainTopicInterestList(Contact)
  ...Buffer SendBuffer = {}
  For each Topic in Node.TI  $\cap$  ContactTI do:
    ...Vector ContactTopicSV = Contact.obtainSummaryVector(Contact, Topic)

    ...For Chunk in Node.Topic.SV complement ContactTopicSV do:
      Chunk.ReplCount = Repl
      SendBuffer.append(Chunk)
    .....If SendBuffer.Length  $\geq$  Rate then:
      .....Break
    .....End If
    End For
  End For
  ChunksSent = Transmit(SendBuffer)
Return ChunksSent

```

Figure 4 Content over CafRep

content to each node. Before the sending node starts sending the chunks to its contact, the receiving node is queried about its topics and chunks that it already has per topic. When the queried node responds, the

sending node finds the intersection of the topics between the two nodes and then it calculates for each topic the complement of the two summary vectors the nodes have to determine the list of topics and associated chunks to be sent without duplication.

3.5.2 Exploring Different Weighting for Congestion Heuristics

CafFRep adaptively moves the traffic away from the more overloaded parts of the network to freer parts of the network while avoiding greedy choices of choosing the currently more available nodes that may congest later on. The decisions of how cautious (versus how greedy) a node should be when choosing the next hop and deciding on the number of packets are not trivial to make and could depend on the state of the network and relative weights of the CafRep utilities. At times of low congestion, social utility is the primary utility for forwarding and replication policy but as the congestion increases the social utility plays a smaller role while resource considerations become increasingly important. More specifically, social utility allows the nodes to find the most direct route to the destinations and gives more copies to the nodes with higher social utility value but can congest them and cause them to be unusable. When congestion happens, the combined utility allows CafRep to de-cluster individual nodes and parts of the network by leveraging social metric with resource constraints. In Section 3.5.1 we defined CafRep utility as the sum of the equally contributing utility values. Computing optimal weights that adaptively favour different utilities differently at different times is a very difficult problem even with the complete knowledge about the environment [47]. In the case of inherent uncertainty of the DTN environments, extreme heterogeneity of the connectivity patterns and no feedback, related research [47] has ruled out provably efficient online routing algorithms. In [46], the authors analyzed the impact of storage and transmission limitations on DTN message routing by and aimed to provide a comprehensive formalization of this problem based on the first principle of Wardrop, but their based their work on the Oracle based routing algorithm. This is not applicable to our work as we are interested in fully distributed opportunistic approach.

Even though we do not aim to theoretically analyse different utility weighting models, we have, in our earlier work [31], considered assigning different weights to each of the utilities for Café in order to empirically explore their relative importance on the adaptive forwarding. Our extensive experiments over RollerNet trace[21] revealed a number of interesting findings. We showed that utilising only a single forwarding strategy based on one node utility leads to suboptimal next hop choices. For example, in [15]

retentiveness was weighted significantly higher than receptiveness and our results showed increased delays and decreased success ratios over Infocom 2006 trace[33] compared to when they were equally weighted [31]. Similarly, when we weighted the receptiveness significantly higher than the retentiveness, Café performance was lower due to the increased packet loss rates [31] as a result of greedy storage utilisation. Another interesting finding was that using ego network retentiveness only was highly valuable for forwarding in social opportunistic networks such as Infocom trace[33]. In particular, even a simple ego network retentiveness utility for the majority of time performed better than more sophisticated analysis of the node only resources statistics. This was expected as Infocom is a social trace with high degree of reoccurring contacts. Both of our empirical results with Café over RollerNet and Infocom 2006 traces [21,33] revealed that combined metric of similar weights allows the forwarding protocol to be sufficiently dynamic and flexible to operate as mainly social protocol at times of low congestion and as a mainly resource driven protocol at times of high congestion.

In this section, we describe our experiments with different CafRep weightings across three topologically different traces: one highly social (Infocom 2006[33]), one sparse social (Sassy[44]) and one highly sparse vehicular (SF Cabs[45]). These traces are described in Section 4.1. As with Café evaluation, extreme differences between CafRep utilities do not result in performance gain, so here we look at the finer differences between the CafRep utilities' weightings. As per each individual trace, we could not detect dramatic differences in the performance results between different CafRep weightings. Figure 5 shows no more than 5% difference in success ratios between the best and worst performer per trace. However, we could clearly detect that some utilities were significantly more dominant than the others across different traces.

More specifically, in order to understand the impact of the SocialUtil, we looked into High SocialUtil (0.6 of the CafrepUtil), Low SocialUtil (0.3 of CafRepUtil) and Equal SocialUtil (0.5 of the CafRepUtil) while the node utilities and ego network utilities were ranked equally. Figure 5 and Figure 6 show that CafRep with lowest social weighting (0.3) has the highest success ratio (46% to 48%) and the lowest delay (29 to 45 hours) for San Francisco Cab trace [45]. This is expected as this trace has the sparsest topology with least repetitive connectivity patterns. CafRep with high SocialUtil achieved highest success ratios (90% to 81%) and lowest delays (67 to 132 sec) for Infocom 2006 and Sassy (73% to 64% and 44 to 83 minutes). CafRep with equal SocialUtil and low SocialUtil closely follow the high Social utility for the respective traces. This is expected as both of these traces are social and have more repeating contact patterns so SocialUtil is able to utilise the complex graphs statistic in order to make better predictions about the most direct route to

the destination. Even though both include human mobility patterns, compared to Infocom 2006 trace, Sassy trace [44] has much less regular and sparser connectivity, no highly central points (as the ones deployed by Infocom 2006). This results in lower performance results for all weightings of SocailUtils for Sassy compared to Infocom.2006

In order to understand the relative importance of ego network utilities and node utilities, we looked into CafRep with low SocialUtil (0.2 of CafREpUtil) and unequal weightings between ego network utilities and node utilities: first we weighted ego network utilities 0.6 and node utilities as 0.2 of CafREpUtil (we refer to it as HighEN Util) and then we weighted ego network utilities 0.2 and node utility as 0.6 of CafRepUtil (we refer to it LowEN Util). It is interesting to see that ego network resources play less important role than node resources in San Francisco Cab trace[45] compared to Sassy[44] and Infocom 2006[33] traces. More specifically, CafRep with HighEN Util achieves lower success ratio than CafRep with LowEN Util for the SF Cabs trace[45] (and vice versa for SASSY and Infocom 2006 traces). Figure 5 shows CafRep with HighEN Utils success ratio ranging from around 47% to 44% for SF Cab trace, followed by 68%- 60% for Sassy trace and 86%-76% for Infocom 2006 trace. This lowest performance of SF Cab trace [45] is due to it having the least repeating mobility patterns and thus not benefiting significantly from the nodes' ego network resources predictions for future forwarding. So the most greedy approach of low EN Util and low Social Util works the best for SF Cab trace [45] while its performance it the worst for the other two social traces. In the case of both social traces, HighEN Util performs better than LowEN Util but worse than higher SocialUtil version of CafRep. This is expected as ego networks play a more important role in social traces.

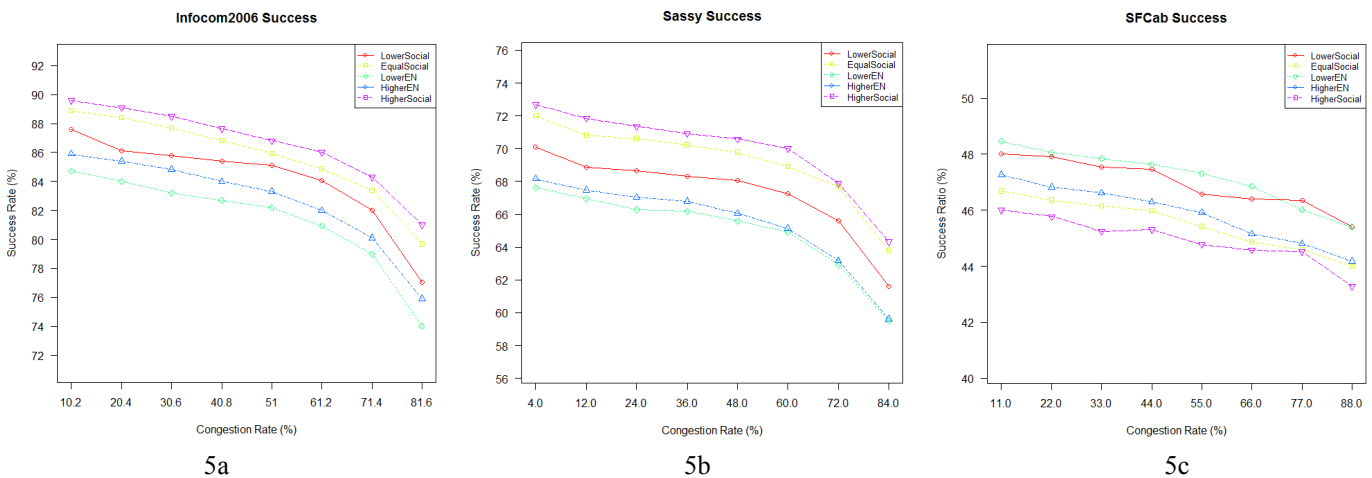


Figure 5 Success ratios of varying CafRep weighting models

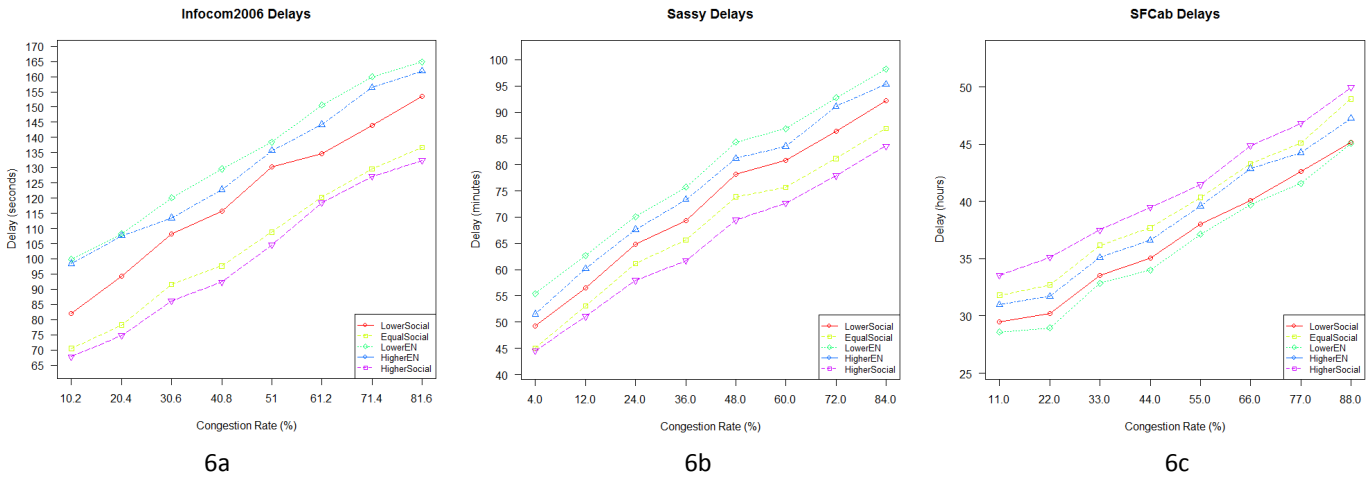


Figure 6. Delays of varying CAFERP weighting models

4 Evaluation Methodology

We perform extensive evaluations of CafRep against state of the art DTN protocols across a range of metrics, three realistic topologies, and two application scenarios. As mobility and connectivity patterns of nodes have major impact on the performance of communication protocols in DTNs, we choose three real-life connectivity and GPS traces from CRAWDAD [1] to ensure sensible transmission ranges and realistic movement patterns of mobile users and vehicles. Our selected traces exhibit vastly different connectivity patterns and we describe them in Section 4.1 We use two different application scenarios that have different traffic patterns: publish subscribe podcasting application where publishers publish messages with fixed sizes at constant bit rate, and Facebook social networking application that has heterogeneous users that generate content at a variety of different rates and sizes. We induce varying levels of congestion by increasing the percentage of randomly chosen publishers for the podcasting application and by decreasing buffer sizes for the Facebook application. Increasing the number of randomly chosen publishers allows us to have non-uniform temporal and spatial congestion rates across the network topology. This example podcasting application is appropriate for the realistic photo/video uploading application scenarios that were shown to have a larger number of publishers than subscribers [41].

We compare CafRep to two benchmark DTN routing protocols and three competitive (adaptive) routing DTN protocols. All our experiments are done in the ONE simulator [20] and our performance metrics include: success ratio, end to end delay, node buffer availability, number of forwarded packets, packet loss rates, number of delivered packets and number of replicated packets. We aim to show that CafRep adapts well to the topologically different networks and two applications while being efficient in terms of network resources.

4.1 Real-life Connectivity and GPS Trace Datasets

To evaluate CafRep in topologically different networks, we choose to use three real world connectivity and GPS traces from CRAWDAD: Infocom 2006 [33], Sassy 2011 [44] and San Francisco Taxi Cabs 2011[45] that have different mobility and connectivity patterns. We show that CafRep is successful independently of the connectivity patterns of the underlying networks. The connectivity and GPS traces we use are briefly described below:

Infocom 2006 trace [33] consists of a 4-day long trace that is based on a human mobility experiment conducted at Infocom 2006. A total of 78 volunteers joined the experiment and each was given an iMote device capable of connecting to other Bluetooth-capable devices. In addition 20 static long-range iMote devices were placed at various locations of the conference venue; three of these were semi-static as they were placed in the building lifts. This dataset has been shown to exhibit strong community structure [16]

Sassy trace [44] consists of a 79 day long trace that is based on a human mobility experiment conducted at St. Andrews in 2011. A total of 27 volunteers joined the experiment and each was given a TMote Invent sensor mote and encounters were tracked in their day-to-day activities for a period of 79 days. The range of these devices was about 10m and encounters were uploaded to a base station regularly (but the base station did not have any role in forwarding). Similarly to Infocom 2006, this dataset has some social structure but unlike Infocom 2006, it has no infrastructure-like nodes, is much sparser and results in much longer disconnection periods.

San Francisco Cab Trace [45] are live traces that record the GPS coordinates of 550 cabs, logged approximately every 10 seconds, over a period of 30 days, in the San Francisco Bay Area. We have downloaded the most recent at the time of writing traces for the period of September 20th 2011 to October 20th 2011 via the Cabspotting.org API. These traces are part of the Cabspotting project [45] that aimed to infer and visualise Taxi Cab collocation information from GPS coordinates in the San Francisco Bay Area. We have assumed that two cabs are collocated if their physical distance is less than 50 meters, furthermore, as cab clocks are not synchronized we have assumed a 60 second interval during which, if the distance between two cabs is less than 50 meters, those cabs are assumed to be collocated. We use 100 taxis that were logging their location data most frequently for higher confidence and accuracy of the GPS traces. This trace has shown to exhibit long periods of disconnections, short periods of connectivity and islands of connectivity that are rarely populated by more than two nodes.

Figure 7 below draws a comparison between the three described mobility traces Infocom 2006 [33], Sassy [44], and SF Cabs[45], highlighting their topological differences.

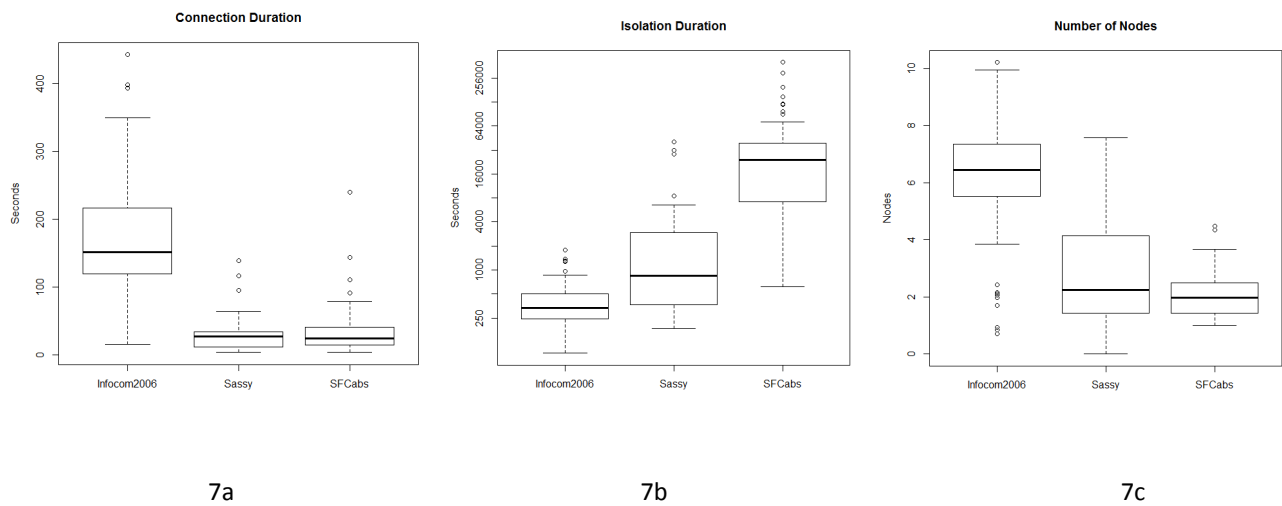


Figure 7 Data traces Characteristics

Figure 7 shows that SF Cab trace [45] is the most challenging trace with very short connectivity durations, very high disconnections and low number of connected nodes during connected times. Both Sassy [44] and SF Cabs [45] are significantly more challenging compared to Infocom 2006 [33] both due to the smaller numbers of neighbours during connectivity times (connectivity sets), and short connections combined with long disconnections.

Figure 7(a) shows that both Sassy[44] and SF Cab [45] traces exhibit predominantly short contact durations (a mean of 33sec and 31sec, a median of 27sec and 24sec and a maximum of 2.3min and 4min respectively) while Infocom 2006 displays substantially longer contact durations (a mean of 3min, a median of 2.5min and a maximum value of 7min).

Figure 7(b) illustrates three different trends regarding node isolation periods. SF Cabs trace suffers the longest periods of isolation (a mean of 10hours, a median of 6.5hours and a maximum value of 4.5 days); while Sassy is significantly more disconnected than Infocom 2006 but much more connected than SF with mean 1.5 hours, median 14 minutes and maximum 11 hours; and Infocom 2006 experiences substantially lower periods of isolation (a mean of 5.5 minutes, a median of 6.4 minutes and a maximum value of 1.5 hours)

Figure 7 (c) displays the difference in node connectivity between the three datasets. This is calculated as the number of active connections a node has at the time a new connection is established. SF Cabs trace has the lowest observed node connectivity (a mean of 2, a median of 1.9 and a maximum value of 4.47 connections), Infocom 2006 has the highest observed node connectivity (a mean of 6.19, a median of 6.44 and a maximum value of 10 connections) and Sassy experiences a mean of 2.95, a median of 2.23 and a maximum value of 7.57 connections)

Each of these datasets provides a different and challenging environment for the algorithms to perform in: SF Cabs due to the very long periods of isolation, short periods of connectivity and small connectivity sets; Infocom 2006 due to the longer periods of connectivity, moderate isolation periods and larger connectivity sets. Sassy due to short connectivity periods, longer isolation periods and small connectivity sets. In section 5 we show that CafRep adapts well to the dynamics of all three datasets as it keeps high success ratio and availability, low delay and packet loss rates, and outperforms the other protocols across all datasets. We compare the performance of CafRep against benchmark protocols: Prophet [23], Spray and Focus (SF) [37] and competitive adaptive protocols Café[15], Encounter Based Routing (EBR) [26] and Retiring Replicas (RR) [38]. Prophet has become a prevalent benchmark forwarding algorithm, SF as it is a prominent fixed replication forwarding algorithm, EBR due to replication placement being variable and RR because of its distinct adaptive replication capping.

4.2 Application Models

In order to evaluate CafRep in the presence of congestion at different rates and locations, we have designed and built a fully distributed, interest driven overlay file casting application as described in Section 3. We randomly assign topic interest and choose varying number of publishers and subscribers. The data is published by the publishers at the constant bit rate (5Mb/sec) and messages of uniform sizes (1MB) are sent to the neighbours. We assume fixed, limited buffer 1GB for this set of experiments. We run eight increments of congestion levels induced by increasing number of publishers ranging from 1/9 to 8/9 of total number of nodes in that connectivity dataset. All simulations are repeated a ten times with different random subscribers and publishers. Related work on publish-subscribe data dissemination in DTNs in [43] explicitly relies on detecting communities and does not consider congestion control. [19] proposes content-based forwarding and buffer management based on content popularity, adding explicit application hints to messages that are visible to each intermediary node, allowing them to cache content, act as distributed storage, or perform application-specific forwarding, but they do not consider congestion awareness or multiple sources. [27] allows generic functions such as bundle routing to be performed differently per application, operation, or resource, but particularly enables application support by means of caching or distributed storage, but does not consider congestion aware forwarding.

In order to consider the impact of the real world social application on the behaviour of CafRep, we designed our Facebook Application that allows us to model the traffic typical of communication in social networking applications. We extracted the friendship graphs and statistical data regarding the size and frequency of posts for every user from the Facebook application and used it to drive the publish and

subscribe application on the top of CafRep. We have sampled the usage patterns of 95 Facebook users and extracted list of friends associated with each user, their 20 most recent wall post messages and their list of interests. The size of a wall post is calculated as the total download cost (e.g. if a message contains a http link or a photo then this is measured and appended to the message size).

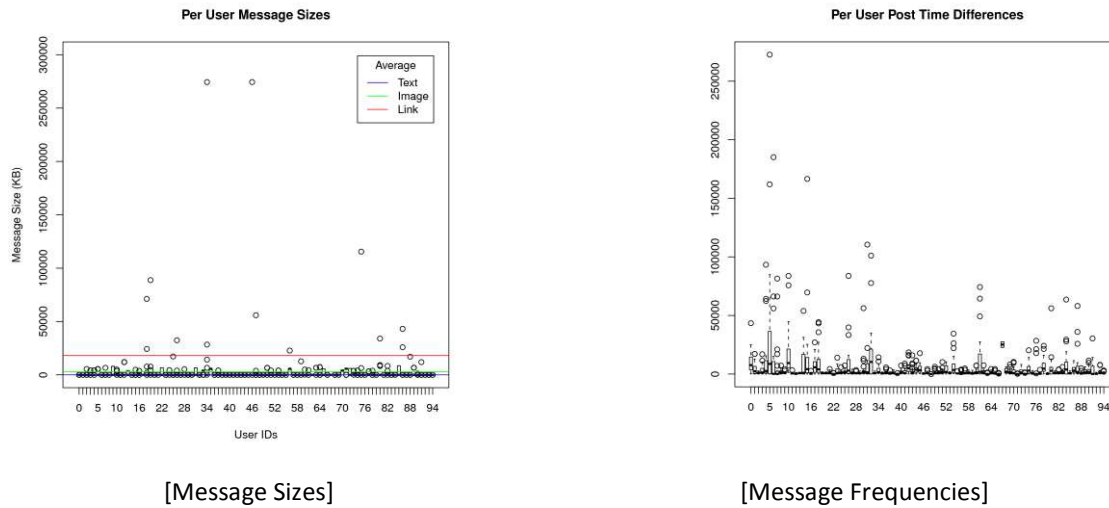


Figure 8: Facebook Traffic Characteristics

Figure 8a shows the non-uniform sizes of messages generated for each user with the Mean size of 1MB, a Median size of 82KB and a maximum size of 268MB. We observe 3 types of message, text, picture and link that are posted at different frequencies: text messages are the most commonly posted (78%), followed by posts containing pictures (19%), with only 3% of posts encompassing links. We found that the most frequent message types (text) formed a small proportion of total traffic (153KB in total and 82B on average out of total 2.3 GB observed data) while the remainder of the data was almost equally split between picture messages (1.2GB in total and 3MB on average) and links (1.1GB in total and 18MB on average). Figure 8(b) illustrates the duration of time between posts for each users. The most prolific participant posted every 3 minutes on average, whilst the most inactive profile posted content once every 25 days on average. We observed that nodes are connected to 3 friends on average and at most 2% of all other nodes.

We define “user profile” that contains statistical information such as: the ratio of text, picture and link messages, the average and standard deviation of message sizes for each message type and the average and standard deviation of message frequency information. Formula 9 illustrates how we use the average and standard deviation values in order to generate a new and meaningful value for the next message generation time and message size. λ is the average of x , σ is the standard deviation of x and W is a normally distributed random number between -1 and 1.

$$P(x) = [0.5 + \lambda(x) + (\sigma(x) \cdot W)] \quad (9)$$

When a message is generated (or as the application starts) the next message is scheduled for generation. When a new message is generated a message type is assigned, as per the observed message type distribution. When the message type has been selected the message size can be assigned based on the relevant statistics. We run ten increments of congestion levels, decreasing the size of buffers from 100MB down to 10MB at 10MB increments. We randomly select traffic profiles for the nodes to follow and simulate multiple runs with different random traffic profiles. Each experiment is emulated over 10 runs, with each run having a different random seed.

5 Evaluation

In this section we first discuss our findings from the interest driven file casting application experiments across each of the three CRAWDAD datasets described in 4.1. We then briefly report on the observations from our Facebook application over RollerNet trace [21] that has highly challenging connectivity patterns due to the Accordion Effect described in [39]. For more consistent comparisons between the datasets, we use CafRepUtil for all of our experiments as defined in section 3 with SocialUtil, node resource utilities and ego network resource utilities being equally weighted. We show that CafRep adapts well to different mobility and connectivity patterns and outperforms five major competitive and benchmark protocols.

Compared to the traces we used in our earlier work [15,30,31,48], in this work, we use two new, very different data traces (social and vehicular) that are much sparser and have much less frequently recurring connectivity patterns (Sassy[44] and SF Cabs[45]). It is interesting to see that CafRep performance is slightly worse for the new social trace (Sassy traces[44]) compared to the old social trace (Infocom traces [33]). Similarly, CafRep also performs on average worse for the new vehicular trace (SF Cab traces[45]) than for the old vehicular (DieselNet) traces. This is due to Sassy[44] and SF Cab traces[45] having much sparser topology and no infrastructure-like nodes that are taking part in the forwarding process. Both Infocom 2006[33] and DieselNet[3] traces had the motes and APs that were taking part in the forwarding decisions and were highly central nodes. Without such nodes, CafRep's performance is lower but still better than for other comparative protocols. As expected, the non-adaptive benchmark Spay and Focus [37] and Prophet[23] protocols perform worse compared to other adaptive protocols across all the measures. Both of these protocols start dropping the packets at the congested nodes early into the simulation that results in higher delays and lower success ratios. Prophet[23] performs worse than SnF[37]

because it is a non-adaptive single copy protocol.

5.1 Success Ratio and Delay

Across all three data connectivity traces CafRep achieves higher success ratio than all other protocols as it more efficiently detects new parts of the networks that have more resources and avoids the parts of the network that are congesting. Figure 9 shows comparative success ratios across five protocols and three traces in the presence of increasing number of publishers. Figure 9 shows that CafRep has on average 10-15% higher success ratio than RR[38] and EBR[26] and more than two times higher success ratios than for SnF[37] and Prophet[23] across all traces. This is due to the combined CafRep utility, its ego network utilities in particular, that manage to predict resource availability in different regions of the network with good accuracy. Contrary to this, RR[38] is concerned with global congestion measures do not sufficiently account for varying regional behaviours and have negative impact on the success ratio when regional congestions arise. More specifically, RR[38] reacts to detected packet loss in a uniform way so that nodes that are congesting still receive packets (even though fewer copies) that they may drop and available nodes do not receive sufficient number of packets. This overutilization and underutilisation of resources increases the delays and results in lower success ratios for RR[38]. Figure 9 shows success ratios ranging from 77% to 60% for Infocom2006, from 67% to 50% for Sassy, and from 46% to 39% for SFCab. In EBR[26], the respective rates of encounter between two nodes determine the appropriate fraction of message replicas the nodes should exchange. Because EBR[26] relies only on the prediction of the future rate of encounters for each node to decide on the probability of successful message delivery, EBR[26] does not detect congestion and results in highly central nodes congesting at even faster rate which causes increased delays (Figure 10) and lower success ratios (Figure 9). Figure 9 shows success ratios ranging from 72% to 58% for Infocom2006, from 63.6% to 50% for Sassy, and from 44% to 37% for SFCab. Non adaptive protocols do not check for resource constrains but aggressively keep the most direct routes to the destination that leads to the nodes quickly getting congested, increasing delays and having low success ratios (Figures 9 and 10). Figure 9 shows that Prophet's success ratios ranges from 36% to 24% for Infocom2006, from 29% to 19% for Sassy, and from 33% to 13% for SFCab. Similarly, SnF ranges from 55% to 26% for Infocom2006, from 44% to 20% for Sassy, and from 38% to 21% for SFCab. SnF[37] has higher success ratios than Prophet[23] because it is a replication based protocol. It is interesting to see that Café has higher success ratios than SnF and Prophet – more that 50% higher that SnF and more than 200% higher than Prophet across all data traces. The only exception to this is at times of low levels of congestion in SF Cab Trace where the differences are down to 25%.

Between the three connectivity data traces, CafRep achieves lowest success ratio in the SF Cab traces[45] (48% to 44%) due to three reasons: first due to the SF Cab[45] topology described in section 5.1, the social utility in CafRep cannot identify nodes that are significantly more central; second as the isolation periods are very high, the nodes cannot offload their content for a very long time; and third as the connectivity periods are very short, the publishing nodes that generate data at a constant rate cause increased dropped packets. It is interesting to see that CafRep achieves around 20% lower success ratio over SF Cab trace[45] than over the DieselNet[3] vehicular trace (that we used in our earlier work [48] and showed 70%-65% success ratio). This is due to DieselNet trace[3] having a few access points that take part in the forwarding of packets and also having significantly lower isolation periods than SF Cab trace [45] and longer connectivity durations. SF Cab trace[45] that was generated for the purpose of tracking cabs in the SF Bay Area and has much sparser topology that covers a significantly wider area (400 and 1600 square miles) without using any access points (APs) or road-side units (RSUs). We believe that is important that even with such challenging and sparse trace, CafRep manages to outperform other protocols. CafRep achieves higher success ratio over Sassy[44] data trace than with SF Cab trace[45] but much lower than when compared with Infocom data trace. Sassy[44] trace is more challenging than Infocom 2006 because it does not include any infrastructure-like nodes (such as semi-static and static nodes such as those deployed during Infocom2006[33]) that participate in forwarding, has significantly shorter connectivity durations and sparser network. Because Sassy trace[44] does not contain nodes that are significantly better connected and has much less repetitive pattern than Infocom 2006(where the participants used the lift or often came to the registration desk where the semi-static and static nodes were deployed) SocialUtility (as part of CafRep) could not be as helpful as with this trace as with Infocom 2006 trace in identifying the most direct routes to destinations. However, Sassy is significantly more connected than SF Cab trace[45] and thus has much higher success ratios.

Across all the three connectivity traces, CafRep manages to keep lower delays compared to the other protocols due to the following two reasons. First, CafRep is able to predict regional in-network delays with good accuracy because of checking for the in-network delays of both the nodes and their ego networks. In this way CafRep is able to more quickly identify (potentially) longer but less congested paths with lower delays than the other protocols. Neither RR [38] nor EBR[26] are able to efficiently discover longer routes with smaller queue sizes but instead only decrease their replication rate and remain in the congested regions for longer periods of times. Figure 10 shows delays for RR ranging from 100-220sec, 70-120minutes and 45-75hours for CafRep across Infocom, Sassy[44] and SF Cab traces[45] respectively. For the majority of time this is about two times as high delay as for CafRep across the three traces. Non adaptive protocols

(Prophet[23] and SnF[37]) have aggressive forwarding strategy that quickly leads into congesting the en-route nodes that increase delays. Figure 10 shows delays for SnF and Prohpet ranging from 180-320 sec, 80-160minutes and 60-100 hours for the three traces respectively. This is between 1.5 to 4 times higher than the delays for CafRep. Second, using Social utility as part of the CafRep utility, CafRep ensures best prediction of the most direct route to the destination especially for Sassy and Infocom 2006 traces. Neither of the RR[38], EBR[26], Prophet[23] and SnF[38] use social metrics and resource metrics together and are thus not able to adjust to the dynamics in both of these dimensions.

Infocom 2006 delays are significantly lower than for the other two datasets across all protocols (seconds versus minutes and hours). This is due to topology characteristics such as short disconnection times, long connection times in Infocom 2006[33], moderate connectivity sets and few infrastructure-like nodes that allow for more efficient data dissemination. For similar reasons, delays across all protocols are significantly higher for SF Cab Traces [45] than for delays for Sassy[44] and Infocom 2006 traces. Still, because CafRep considers the relative in-network delays, it is able to maintain lower delays than the rest of the protocols.

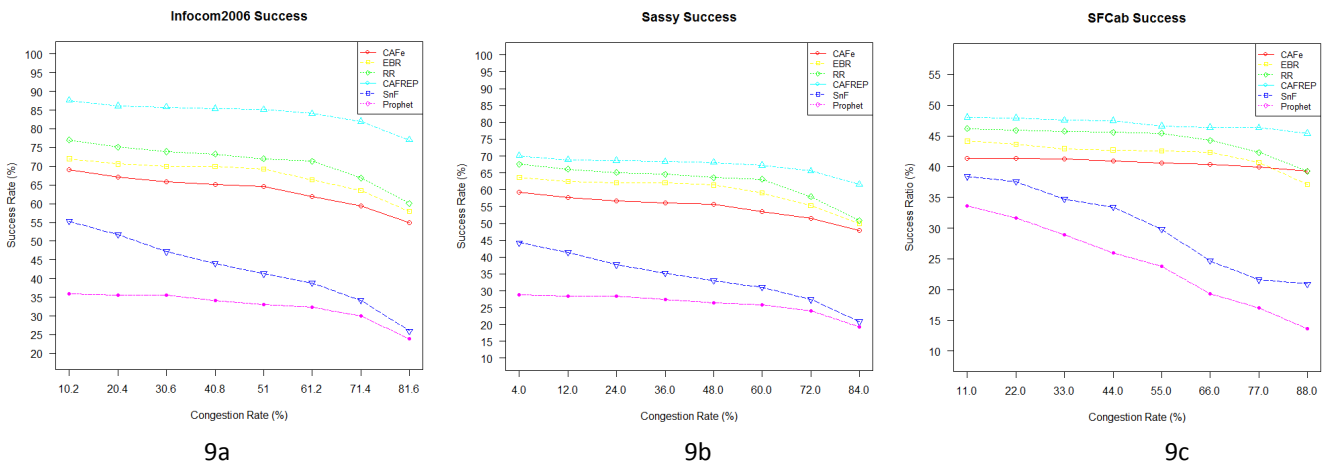


Figure 9. Comparative success ratios across three traces

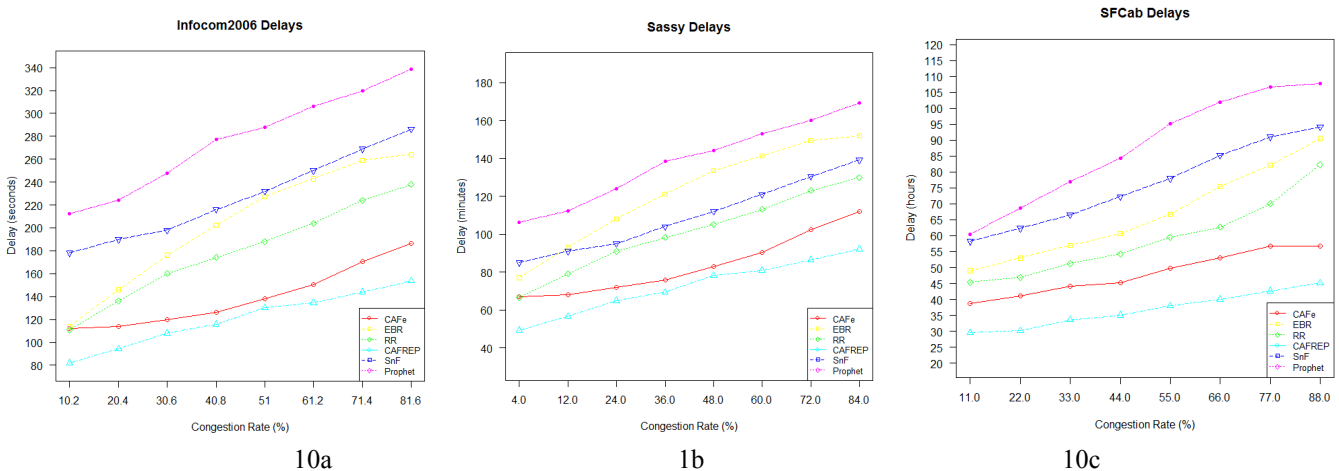


Figure 10. Comparative delays across three traces

5.2 Availability and Packet Loss

Across all three traces, CafRep sustains higher availability than other protocols. This is because CafRep is able to make good predictions of node and ego network availability which avoid depleting the storage resources of frequently used nodes and regions in the network that drop packets. Figure 11 shows CafRep availability ranging from 88%-70% for Infocom 2006, 70%-60% for Sassy[44] and 60% to 35% for Sf Cabs[45]. EBR[26] results in lower availability as it congests the regions that are highly central and where the nodes cannot offload the traffic faster than the traffic is generated (that is the example application scenario we are considering). Figure 11 shows that EBR[26] manages only 70%-20% availability over Infocom[33] and Sassy[44] traces and 40% to 20% for SFCabs trace[45]. It is interesting to see in Figure 12 that RR[38] has higher packet loss rates than CafRep despite RR[38] tracking and adapting to the global packet loss congestion signals. We believe that this is due to RR[38] not being fast enough to detect sudden opportunities of longer but more available paths as it awaits for the global signs of increased packet loss to act locally. RR[38] manages to keep availability levels at 80% to 20% for Infocom 2006[33] and Sassy traces[44] and only 45% to 30% for SF Cabs[45]. RR[38] maintains packet loss rates 33%-63% , 49%-83% and 75%-95% for Infocom 2006[33], Sassy[44] and SF Cabs traces[45] (given in Figure 12). RR is more suitable for even traffic distribution and connectivity patterns, such as constant bit rate traffic with a random way point mobility model, as the nodes in such networks become congested at a uniform rate. In heterogeneous DTNs that we consider, uniformity is not common, so it is more likely that traffic hot spots trigger messages to be dropped while other areas of the network remain highly available. For such scenarios, when RR [38] uniformly reduces the number of replicas entering the network, it results in underutilisation of existing network resources and overutilization of some network parts.

The low availability of CafRep over SF Cabs traces[45] comes primarily from the very long node isolation periods combined with intense traffic generation during short connectivity durations. This means that, despite discovering more paths than other protocols can discover, with the increase of publishing nodes (above 70% congestion rate where most nodes are publishing), publishing nodes are forced into decreased availability and dropping the packets that they cannot offload. With increased congestion levels (above 50% of publishing nodes), Figure 11 shows that CafRep manages to keep two times higher availability than EBR[26], and above 50% higher than RR[38]. It is interesting to see that the differences in availability between the three adaptive protocols are smallest for low congesting rates in Sassy[44] and Infocom 2006 data traces[33], but as the congestion levels increase, CafRep starts to dominate over RR and EBR starts to fall behind more dramatically. For SF traces[45], CafRep is significantly better than RR[38] and EBR[26]

across all congestion levels. This is because EBR[26] and RR[38] are slower to utilise already very infrequent opportunities to divert the traffic along diverse paths.

All protocols manage highest availability over the Infocom 2006 trace [33] because this trace is highly social (allows repeating patterns of connectivity) that allows for accurate social/connectivity predictions that CafRep, EBR[26], RR[38] and SnF[37] use. In addition to this, this trace deploys infrastructure-like nodes that facilitate interaction among nodes, has longer contact times, shorter smaller isolation periods and bigger connectivity islands. Availability for both non-adaptive protocols, Prophet and SnF, is low across all the three traces and it ranges from 0.6-0.13, 0.6-0.14 and 0.3-0.1 across Infocom 2006, Sassy and SF Cab traces respectively. With the constant bit rate data transmission of the publishing nodes, these two protocols quickly saturate the en-route nodes that increase packet drop rates ranging from 0.59-98%. 0.71-96% and 0.62-95% across Infocom 2006, Sassy and SF Cabs trace respectively. Cafe, even though being a single packet protocol, maintains significantly better availability: up to twice the availability of and around three times lower packet loss rates than non-adaptive protocols. This is due to Cafe's adaptive forwarding that predicts identifies highly congesting nodes and regions, and avoids them.

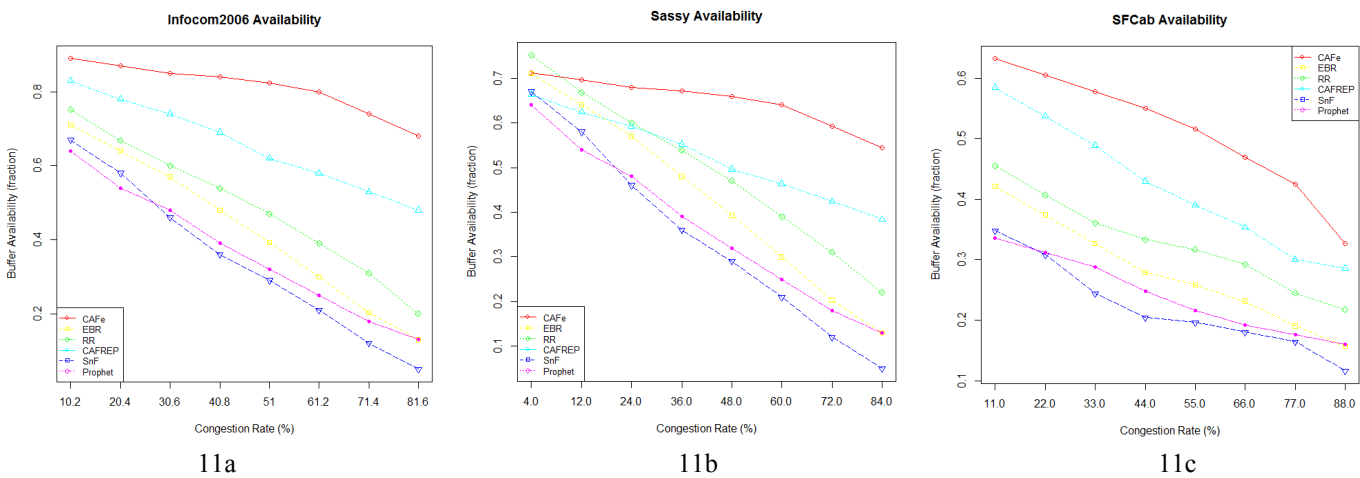


Figure 11 Comparative Availability across three data traces

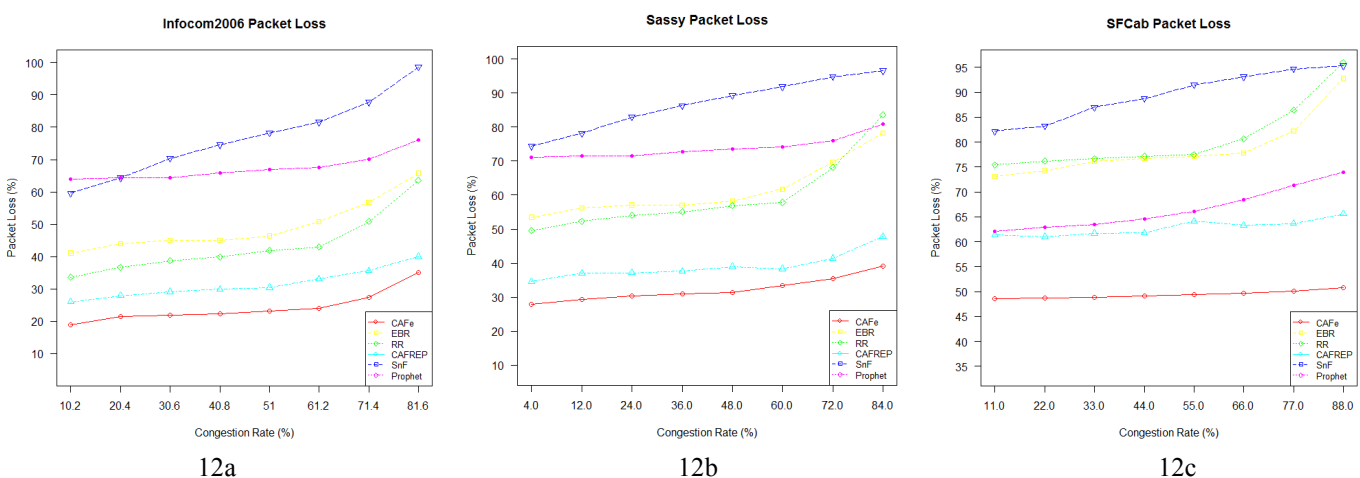


Figure 12 Comparative Packet Loss across three data traces

5.3 Performance costs: Forwarded, Replicated Packets and Delivery Cost

We assess performance costs in terms of total average number of delivered, forwarded and replicated packets. Figure 13a shows that even though CafRep may longer paths but less congested paths, it does not generate significantly more forwarded packets compared to the two adaptive replication protocols RR[38] and EBR[26]. For example, we observe similar average numbers of forwarded packets for CafRep, EBR[26] and RR[38] across Infocom 2006[33] and Sassy[44] but higher for CafRep over SF Cabs[45]. This is due to the following two reasons. First, even though CafRep uses greater diversity of available paths, its Social Util allows it to keep as direct route to the destination as possible over the social traces. EBR and RR do not exploit more complex connectivity relationships and may thus result in long paths especially in the presence of increasing congestion. Since SF Can Trace is not a social trace and CafRep benefits less from the sophisticated contact relationship analysis, it results in marginally higher number of forwarded packets. Second, even though EBR, RR and CafRep forward similar rates of packets, EBR and RR drop more packets as the forwarding is not appropriately distributed to match the non-uniform varying resources in the network. The number of forwarded packets for Café[15] and Prophet[23] is smaller than for the other protocols as they do not use replication. As expected, SnF[37] forwards on average highest number of packets as it is non-adaptive replication-based protocol.

More interestingly, CafRep and RR replicate similar number of packets across Sassy trace[44] while CafRep replicates 15-30% more than RR[38] and EBR[26] for SF Cab trace[45] (Figure 13b). We believe that higher replication rates for CafRep in SF Cab trace [45] are due to CafRep discovering more forwarding opportunities that the other two adaptive replication protocols do not discover. Compared to CafRep, EBR[26] replicates 15%-20% more packets over Infocom 2006 trace[33] because that trace contains highly central nodes that EBR[26] uses heavily for the replication control and as EBR does not check for resource constraints this replication results in increased packet loss rates and lower availability.

Figure 13c shows the cost of delivery that we define as the number of messages forwarded over the number of messages delivered across six protocols and over the three traces. This is a significant metric as it relates to the average number of hops required to deliver a message. As expected, Café[15] has the lowest cost of delivery across the three traces as it adaptively forwards to offload network hotspots in a delay-aware manner and does not include replication. Out of the replication-based protocols, CafRep outperforms EBR[26], Prophet and SnF[37] across all the three data traces and is only marginally outperformed by RR[38] in the SF Cab dataset[45]. As discussed in section 3, we believe that this higher cost for CafRep in SF Cab trace[45] comes from high weighting of ego network resources and social utility

in a trace that does not have node connectivity with good pattern of regularity. We expect better results for CafRep over this trace if we deployed low weighting of social utilities and ego network resources so that the results do not suffer from the poor quality for the ego networks formed in unstructured environments. Our results also show that SnF is consistently the worst performer with approximately double the cost of all the other algorithms, this is due to the static nature of SnF's[37] replication scheme, which under populates the network with traffic when demands are low and does not reduce the volume of traffic when contention is high. Across all the protocols, the cost of message delivery is the lowest for Infocom 2006 trace [33], followed by Sassy trace[44] and then SF Cab trace[45]. This is due to significantly better connected Infocom 2006 trace than the other two traces, and most isolated and more random connectivity patterns for SF Cab trace[45].

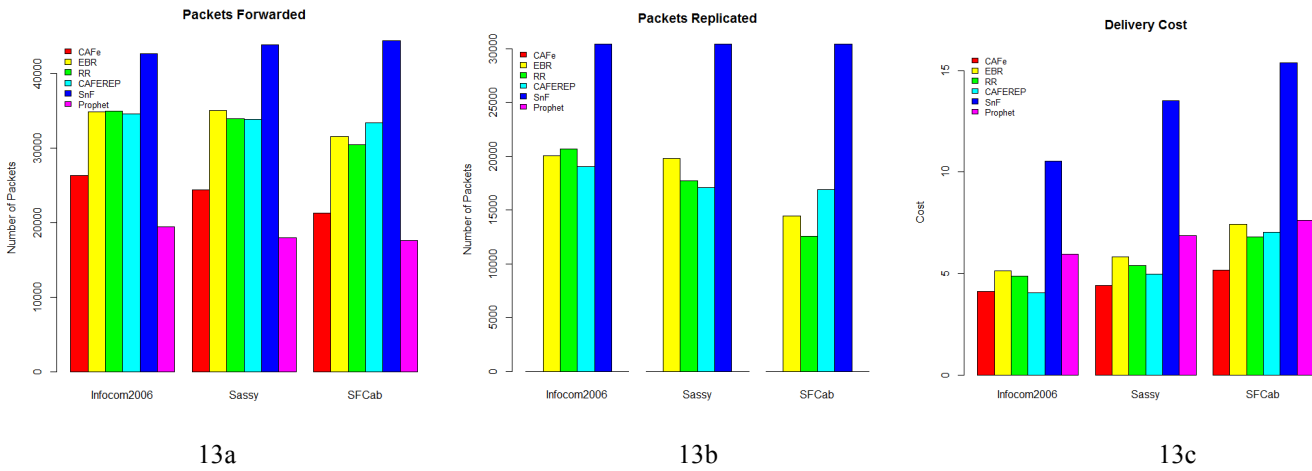


Figure 13. Comparative performance costs across three data traces

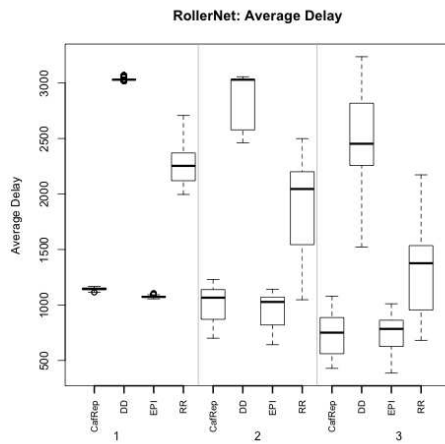
5.4 Facebook Application

This section explores the effects that real world social networking traffic usage patterns have on the performance of CafRep. Figure 14 illustrates three experiments with different traffic profiles: low traffic profiles, randomly selected traffic profiles and high traffic profiles, which are indicated by 1, 2 and 3 respectively. We compare CafRep with benchmark DTN protocols Direct Delivery (DD)[49] and Epidemic (EPI)[50], and the state-of-the-art competing congestion control algorithm Retiring Replicas (RR)[38].

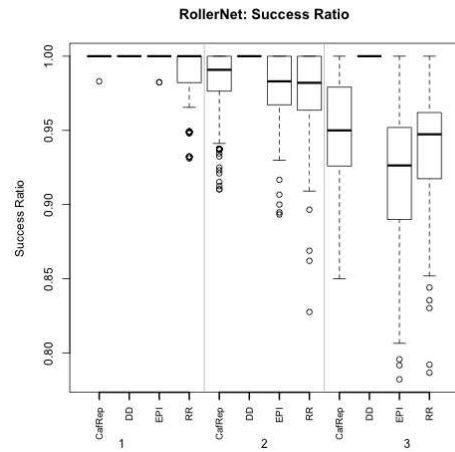
In Figure 14a we evaluate the average number of seconds delay each algorithm experienced across the three different traffic profiles. We observe higher delays for less active profiles (that generate smaller messages at low frequencies) as delivery opportunities are low and message generation is infrequent. This in turn causes nodes to store their messages within their buffer for extended periods of time. These user profiles cause lower variance for the majority of algorithms which reflects the decreased contention in the network. We observe that CafRep has lower delays than all other algorithms excluding epidemic for these

profiles. RR[38] is the exception to the reduced variance, as it displays high delay variance and much larger delays than other algorithms. This is because RR[38] assumes that congestion is uniformly spread and do not identify congesting regions in the network, which occur more frequently when traffic patterns are less uniform. As the message sending rates intensify the probability of a node having generated a message for an encountered destination increases, as a result message delays are consistently lower in comparison to the less active profiles. The increased traffic demands from these profiles lead to a higher level of contention in the network, which results in a higher variance in delays. This is best illustrated by the Direct Delivery[49] results that show significantly reduced delays and increased contention. We also observe that in the high traffic experiments CafRep even outperforms Epidemic[50]. This best illustrates the benefits CafRep offers by alleviating contention in the network through the use of multi-path forwarding. Our results show that in the mixed traffic profile experiments delays experienced are between the low and high traffic profiles, but the degree of variance is the highest of all 3 experiments. This is due to the heterogeneity of the user activity, which leads to volatile traffic demands. Despite the challenges presented by these mixed demands CafRep continues to outperform the other algorithms, while RR[38] experiences increased contention due to its assumption of homogeneous user patterns.

In Figure 14b we show the success ratio of CafRep in comparison to DD[49], EPI[50] and RR[38]. We observe that CafRep supports real application traffic, as our results for real application traffic are consistent with our previous experiments. CafRep has the highest success ratio in the less active profile experiments, in comparison with mixed and high traffic profiles as the levels of congestion are lower. CafRep's success ratio decreases as the levels of contention increases and success ratio for low traffic profiles is higher than mixed, which is higher than the high traffic profile experiments. However CafRep always has higher success ratio than all other protocols excluding direct delivery[49], which has 100% success ratio as it only forwards messages directly to the destination nodes.



14a



14b

Figure 14. Average Delivery Delay (a) and Success Ratio (b)

6 Conclusion and Future Work

We proposed CafRep that uses a combined local social, buffer and delay metrics for congestion aware message forwarding and replication that maximizes message delivery ratio and availability of nodes while minimizing latency and packet loss rates at times of increasing congestion levels. At the core of CafRep is a combined relative utility driven heuristic that allows highly adaptive forwarding and replication policies by managing to detect and offload congested parts of the network and adapting the sending/forwarding rates based on resource and contact predictions. We empirically investigated a number of weighting models for analysing the impact of different utilities on the CafRep performance. We have done extensive performance analysis of CafRep in three CRAWDAD real connectivity and GPS traces with different mobility and connectivity patterns: Infocom 2006, Sassy and San Francisco Cabs. We show that CafRep outperforms five other state of the art DTN adaptive and non-adaptive routing protocols across the majority of metrics across all the traces. We also show that CafRep maintains performance levels in two application scenarios: publish subscribe constant bit rate podcasting and Facebook traffic application. We believe that CafRep provides a useful generic and highly adaptive congestion control framework suitable for different types of resource constraint DTN application scenarios. We also plan to investigate the efficiency of CafRep in the context of more realistic anycast and multicast applications.

7 Acknowledgments

The work was supported by the Engineering and Physical Sciences Research Council UK (EPSRC) Grant number EP/D062659/1

References

- [1] Community resource for archiving wireless data at dartmouth (crawdad).
<http://crawdad.cs.dartmouth.edu>.
- [2] E. Anshelevich and S. Ukkusuri. Equilibria in dynamic selfish routing. *Algorithmic Game Theory*, pages 171–182, 2009.
- [3] N. Banerjee, M. Corner, D. Towsley, and B.N. Levine. CRAWDAD data set umass/diesel/transfer/fall2007 (v. 2008-09-14). Downloaded from <http://crawdad.cs.dartmouth.edu/umass/diesel/transfer/fall2007>, 2008.
- [4] R. Bhagwan, D. Moore, S. Savage, and G. Voelker. Replication strategies for highly available peer-to-peer storage. *Future directions in distributed computing*, pages 153–158, 2003.
- [5] R. Bhagwan, K. Tati, Y.C. Cheng, S. Savage, and G.M. Voelker. otal recall: System support for automated availability management. In *NSDI*, 2004.
- [6] S. Burleigh, E. Jennings, and J. Schoolcraft. Autonomous congestion control in delay-tolerant networks. *American Institute of Aeronautics and Astronautics*, 2007.
- [7] L.P. Cox, C.D. Murray, and B.D. Noble. Pastiche: Making backup cheap and easy. *ACM SIGOPS Operating Systems Review*, 36(SI):285–298, 2002.
- [8] E.M. Daly and M. Haahr. Social network analysis for information flow in disconnected Delay-Tolerant MANETs. *IEEE Transactions on Mobile Computing*, pages 606–621, 2008.
- [9] 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*, pages 161–174. ACM, 2007.
- [10] 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*, pages 251–260. ACM, 2008.
- [11] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin. Diversity in smartphone usage. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 179–194. ACM, 2010.

- [12] A. Ferreira. Building a reference combinatorial model for MANETs. *Network, IEEE*, 18(5):24–29, 2004.
- [13] L. Ford and D. Fulkerson. *Flows in networks*. 1962.
- [14] 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, 2010.
- [15] A. Grundy and M. Radenkovic. Promoting congestion control in opportunistic networks. In *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'2010)*, Niagara Falls, Canada, Canada, 10 2010.
- [16] P. Hui and J. Crowcroft. How small labels create big improvements. In *Proceedings of the 2006 ACM CoNEXT conference*. ACM New York, NY, USA, 2006.
- [17] 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*, pages 241–250. ACM New York, NY, USA, 2008.
- [18] Wei jen Hsu and Ahmed Helmy. Impact: Investigation of mobile-user patterns across university campuses using wlan trace analysis. Technical report, IEEE INFOCOM, 2005.
- [19] Y. Jiao, Z. Jin, and Y. Shu. Data Dissemination in Delay and Disruption Tolerant Networks Based on Content Classification. In *Mobile Ad-hoc and Sensor Networks, 2009. MSN'09. 5th International Conference on*, pages 366–370. IEEE, 2010.
- [20] A. Keranen, J. Ott, and T. Karkkainen. The ONE simulator for DTN protocol evaluation. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, pages 1–10. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.
- [21] J. Leguay and F. Benbadis. CRAWDAD data set upmc/rollernet (v. 2009-02-02). Downloaded from <http://crawdad.cs.dartmouth.edu/upmc/rollernet>, February 2009.
- [22] 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*, pages 457–460. ACM, 2009.
- [23] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. *Lecture Notes in Computer Science*, pages 239–254, 2004.
- [24] A. Lindgren and K.S. Phanse. Evaluation of queuing policies and forwarding strategies for routing in intermittently connected networks. *Proc. of IEEE COMSWARE*, 2006.

- [25] 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*, pages 75–82. ACM, 2007.
- [26] S.C. Nelson, M. Bakht, R. Kravets, and A.F. Harris. Encounter: based routing in DTNs. *ACM SIGMOBILE Mobile Computing and Communications Review*, 13(1):56–59, 2009.
- [27] J. Ott and M.J. Pitkanen. Dtn-based content storage and retrieval. In *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, pages 1–7. IEEE, 2007.
- [28] J.M. Pujol, A.L. Toledo, and P. Rodriguez. Fair Routing in Delay Tolerant Networks. In *Proceedings of IEEE INFOCOM*, 2009.
- [29] M. Radenkovic and C. Greenhalgh. Multi-party distributed audio service with TCP fairness. In *The tenth ACM international conference on Multimedia*, pages 11–20. ACM, 2002.
- [30] M. Radenkovic and A. Grundy. Poster: Congestion Aware Data Dissemination in Social Opportunistic Networks. *SIGMOBILE Mobile Computing and Communications Review (MC2R)*, 14(3):31–33, 2010.
- [31] M. Radenkovic and A. Grundy. Congestion aware forwarding in delay tolerant and social opportunistic networks. In *Eighth Annual Conference on Wireless On demand Network Systems and Services (WONS)*, 2011.
- [32] K. Rzadca, A. Datta, and S. Buchegger. Replica placement in p2p storage: Complexity and game theoretic analyses. In *2010 International Conference on Distributed Computing Systems*, pages 599–609. IEEE, 2010.
- [33] 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.
- [34] M. Seligman, K. Fall, and P. Mundur. Storage routing for dtn congestion control. *Wireless Communications and Mobile Computing*, 7(10):1183–1196, 2007.
- [35] A.N. Shiryaev, B. Rozovskii, and G. Grimmett. Optimal stopping rules. 2008.
- [36] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259. ACM, 2005.
- [37] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops' 07. Fifth Annual IEEE International Conference on*, pages 79–85. IEEE, 2007.

- [38] N. Thompson, S. Nelson, M. Bakht, T. Abdelzaher, and R. Kravets. Retiring replicants: Congestion control for intermittently-connected networks. In *Proceedings of Infocom*, 2010.
- [39] P.U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M.D. De Amorim, J. Whitbeck, et al. The accordion phenomenon: Analysis, characterization, and impact on dtn routing. In *Proc. IEEE Infocom*. Citeseer, 2009.
- [40] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical report, Citeseer, 2000.
- [41] M. Valafar, R. Rejaie, and W. Willinger. Beyond friendship graphs: a study of user interactions in Flickr. In *Proceedings of the 2nd ACM workshop on Online social networks*, pages 25–30. ACM, 2009.
- [42] D. Wischik, M. Handley, and M.B. Braun. The resource pooling principle. *ACM SIGCOMM Computer Communication Review*, 38(5):47–52, 2008.
- [43] E. Yoneki, P. Hui, S.Y. 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*, page 234. ACM, 2007.
- [44] G. Bigwood, D. Rehunathan, M. Bateman, T. Henderson and Saleem Bhatti, CRAWDAD data set st_andrews/sassy (v. 2011-06-03), Downloaded from http://crawdad.cs.dartmouth.edu/st_andrews/sassy, 2011.
- [45] San Francisco cabs data traces, downloaded from the Cab Spotting project, <http://cabspotting.org/api> for the period 20 Sep 2011– 20 Oct 2011
- [46] P.U. Tournoux, V. Conan, J. Crowcroft, J. Leguay, M.D. de Amorim and F. Benbadis. Wardrop Equilibrium Formulation of Resource-Constrained DTN Routing in Public Safety Networks. In *Proceedings of the 8th IEEE International Conference on Mobile Adhoc and Sensor Systems, MASS 2011, Valencia, Spain, October 17-22, 2011*.
- [47] A. Balasubramanian, B. Levine and A. Venkataramani. DTN routing as a resource allocation problem. In *ACM SIGCOMM Computer Communication Review*. 2007.
- [48] M. Radenkovic and A. Grundy. Framework for utility driven congestion control in delay tolerant opportunistic networks. In *Proceedings of the 7th International Wireless Communications and Mobile Computing Conference (IWCMC)*. 2011.
- [49] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Single-copy routing in intermittently connected mobile networks. In *Sensor and Ad Hoc Com- munications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 235–244. IEEE, 2004.

[50] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical report, Citeseer, 2000.

[51] A.J. Mashhadi, S.B. Mokhtar and L. Capra. Fair content dissemination in participatory DTNs. Ad Hoc Networks. 2011. Elsevier.

8 Vitae



Dr. Milena Radenkovic has received her PhD Degree in 2002 from the University of Nottingham, UK and her Dipl Ing from the University of Nis, Serbia in 1998. Her research spans the areas of mobile and delay tolerant networking, P2P systems, and their application to pervasive gaming, social networking, and environmental monitoring. She has been an Investigator of four EPSRC grants. She has organised and chaired multiple conferences, served on many program committees and editorial boards and published in premium venues including ACM MC2R, IEEE WONS, IEEE Multimedia, MIT Press PRESENCE, ACM Multimedia, ACM VRST, ACM CCGRID.



Mr. Andrew Grundy is currently a final year Ph.D. researcher at the University of Nottingham. He received his BSc degree in Computer Science from the University of Leicester in 2008. He has published work in venues including ACM MC2R, IEEE WiMob and IEEE WONS. His research interests include delay tolerant, self-organising, distributed and mobile systems, social network analysis and network theory.

Dr. Milena Radenkovic has received her PhD Degree in 2002 from the University of Nottingham, UK and her Dipl Ing from the University of Nis, Serbia in 1998. Her research spans the areas of mobile and delay tolerant networking, P2P systems, and their application to pervasive gaming, social networking, and environmental monitoring. She has been an Investigator of four EPSRC grants. She has organised and chaired multiple conferences, served on many program committees and editorial boards and published in premium venues including ACM MC2R, IEEE WONS, IEEE Multimedia, MIT Press PRESENCE, ACM Multimedia, ACM VRST, ACM CCGRID.

Mr. Andrew Grundy is currently a final year Ph.D. researcher at the University of Nottingham. He received his BSc degree in Computer Science from the University of Leicester in 2008. He has published work in venues including ACM MC2R, IEEE WiMob and IEEE WONS. His research interests include delay tolerant, self-organising, distributed and mobile systems, social network analysis and network theory.

*Author Photo

[Click here to download high resolution image](#)



*Author Photo

[Click here to download high resolution image](#)

