Congestion Control Framework For Delay-Tolerant Communications

Andrew Michael Grundy

Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy

July 2012

To my parents

Abstract

Detecting and dealing with congestion in delay tolerant networks is an important and challenging problem. Current DTN forwarding algorithms typically direct traffic towards particular nodes in order to maximise delivery ratios and minimise delays, but as traffic demands increase these nodes may become unusable.

This thesis proposes Café, an adaptive congestion aware framework that reduces traffic entering congesting network regions by using alternative paths and dynamically adjusting sending rates, and CafRep, a replication scheme that considers the level of congestion and the forwarding utility of an encounter when dynamically deciding the number of message copies to forward. Our framework is a fully distributed, localised, adaptive algorithm that evaluates a contact's next-hop potential by means of a utility comparison of a number of congestion signals, in addition to that contact's forwarding utility, both from a local and regional perspective. We extensively evaluate our work using two different applications and three real connectivity traces showing that, independent of the network inter-connectivity and mobility patterns, our framework outperforms a number of major DTN routing protocols.

Our results show that both Café and CafRep consistently outperform the state-of-the-art algorithms, in the face of increasing traffic demands. Additionally, with fewer replicated messages, our framework increases success ratio and the number of delivered packets, and reduces the message delay and the number of dropped packets, while keeping node buffer availability high and congesting at a substantially lower rate, demonstrating our framework's more efficient use of network resources.

Publications

Here is a list of refereed publications produced as part of this thesis.

- Milena Radenkovic and Andrew Grundy, Framework for Utility Driven Congestion Control in Delay Tolerant Opportunistic Networks, in: The 7th International Wireless Communications and Mobile Computing Conference (IWCMC 2011), Istanbul.
- Milena Radenkovic and Andrew Grundy, Congestion Aware Forwarding in Delay Tolerant and Social Opportunistic Networks, in: The Eighth International Conference on Wireless On-Demand Network Systems and Services (WONS 2011), Bardonecchia, Italy, Pages 60-67, 26th January 2011.
- Andrew Grundy and Milena Radenkovic, Promoting Congestion Control in Opportunistic Networks, in: IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2010), Niagara Falls, Canada, 11th October 2010.
- Milena Radenkovic and Andrew Grundy, MobiCom 2010 Poster: Congestion Aware Data Dissemination in Social Opportunistic Networks, in: SIGMOBILE Mobile Computing and Communications Review (MC2R) Volume 14, Number 3, Pages 31-33, July 2010.
- Andrew Grundy and Milena Radenkovic, Decongesting Opportunistic Social-based Forwarding, in: Seventh Annual Conference on Wireless On demand Network Systems and Services (WONS 2010), Kranjska Gora, Slovenia, Pages 82-85, 3rd February 2010.

- Andrew Grundy and Milena Radenkovic, Routing in Wireless Networks of Varying Connectivity, in: International Conference on Wireless and Mobile Communications (ICWMC 2009), Cannes, France, Pages 18-23, 24th August 2009.

Acknowledgements

I would like to offer my deepest gratitude to Dr. Milena Radenkovic and Prof. Uwe Aickelin for all their help, support and kindness during the course of my PhD. Our collaboration and discussions were very important for me and for my work.

I would like to thank the Engineering and Physical Science Research Council (Grant number EP/D062659/1) and the University of Nottingham for sponsoring my PhD.

Many thanks also to my colleagues in the School of Computer Science for their help, support and friendship over the years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over 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. [31] showed that in real world traces of different university campus wireless networks, node encounters are sufficient to build a connected relationship graph.

Mobile Ad-Hoc Networks (MANETs) [16] provide mobile, multi-hop, wireless networking in the face of dynamic topologies and bandwidth-constrained, variable capacity links, but require contemporaneous end-to-end paths between the source and destination nodes in order to successfully transmit messages and as such are better suited to small networks such as an office environment.

Delay Tolerant Networks (DTNs) [24] operate a store-carry-forward method of message delivery, which can be successfully used for data transmission despite the absence of contemporaneous end-to-end paths, as show in deployments such as DakNet [65] and ZebraNet [39]. DTN deployments can be categorised as Pure Opportunistic, Schedule Based and Social. Pure Opportunistic DTNs assume that contact reoccurrence may never happen and as such nodes disseminate messages as prolifically as possible, such as in disaster recovery scenarios. Schedule Based DTNs adaptively predict periods of connectivity, these networks often exhibit a fixed topology and are challenged by environmental issues, such as long range radio communication [20] or Interplanetary Networking [12]. Social

DTNs, often referred to as Pocket Switch Networks (PSNs) [33], exhibit reoccurring contact encounters and as such forwarding may be achieved using elements of network analysis [11].

Transferring messages across these networks is a challenging problem, as such algorithm development has been primarily concerned with providing maximum throughput and minimal delays while typically assuming unlimited storage and transfer bandwidth. DTN forwarding algorithms can be grouped into two categories: replication-based forwarding and single copy forwarding.

Replication-based forwarding disseminates copies of a message (replicates) throughout the network in order to increase the accessibility of a message, therefore increasing the probability it will encounter the destination node. The most primitive replication-based forwarding algorithm is referred to as epidemic forwarding [83], which replicates copies of a message to all nodes in the network - this method is excessive and therefore not scaleable. In order to address the number of redundant copies produced by epidemic forwarding, algorithms have emerged that forward a fixed number of copies to a subset of encountered nodes [77, 78, 60].

Single copy forwarding functions by selecting a custodian for a message from the contacts encountered by a node, with the intention that the message will propagated towards the destination. Forwarding to only one node raises the question of which node should receive the message, this has been observed by [22] as an instance of the optimal stopping problem [75]. The simplest example of single copy forwarding is Direct Delivery [76], a scheme which only forwards a message to another node if it is the messages destination. More complex approaches involve network analysis, such as calculating transitive delivery probability and social network parameters regarding the nodes connectivity in order to determine the suitability of a node for being a messages next-hop [54, 35, 18].

Mobile devices have limited resources, therefore it is crucial that the routine assumption of unlimited storage, transfer bandwidth and battery power is not in place. By limiting resources, better connected nodes in the network quickly become congested and unusable, causing even more disconnections and consequently even lower delivery rates [29, 66].

Traditional congestion control, such as the mechanisms of TCP [4], are integral to the stability of the traditional Internet, but are not usable in the non-Internet like scenarios that are tolerant to delays, as they are built around the assumption of contemporaneous end-to-end connectivity and follow a closed loop procedure (relying on the fast turnaround of acknowledgments). TCP is also designed for data to be transferred as small data segments, Instead DTNs are designed for conveying potentially very large units of data.

It is clear that forwarding algorithms need to be concerned with congestion control in order to be robust. Initial research in congestion control for DTNs is concerned with buffer management [13, 50, 56, 74] schemes for selectively dropping packets in order to improve delivery probability. Buffer management is important, but it does not address the issue of finite capacity links and should be a last resort. The state of the art work on congestion control in DTNs [82, 81, 60, 66] propose methods for adaptive replication management and adaptive replication placement.

1.1 Research Problem

In this Thesis I investigate congestion control mechanisms for delay tolerant networks. More specifically I investigate the benefits and costs of using and combining congestion avoidance, multi-path forwarding and rate limiting techniques.

This section is concerned with describing the research problem tackled in this Thesis. We were motivated by the observation that social forwarding strategies that use networking theory in order to identify next-hop nodes quickly suffered from congestion when forwarding bandwidth and buffer sizes were restricted. Social forwarding strategies typically evaluate the Centrality of an node, the simplest of these is Degree Centrality (the number of connections), in order to select a next-hop node.

Our initial analysis was configured such that the nodes in the network had unlimited storage and transfer resources and followed a basic forwarding strategy, which simply states that nodes must only forward to nodes with equal or higher values of Centrality. Using this forwarding strategy we investigated the maximum number of connections a node could be faced with, assuming that every node in the network will have traffic to disseminate (Demand). Figure 1.1 illustrates the key finding from our our preliminary evaluation, the correlation between Degree Centrality and Demand. We observe that nodes with higher Centrality values suffer from much higher levels of Demand than nodes with lower centrality values. In the majority of cases nodes are not elected as custodians, and the level of demand is significantly increased for the top 10% of nodes to between 30 and 50 times above the average level of demand.

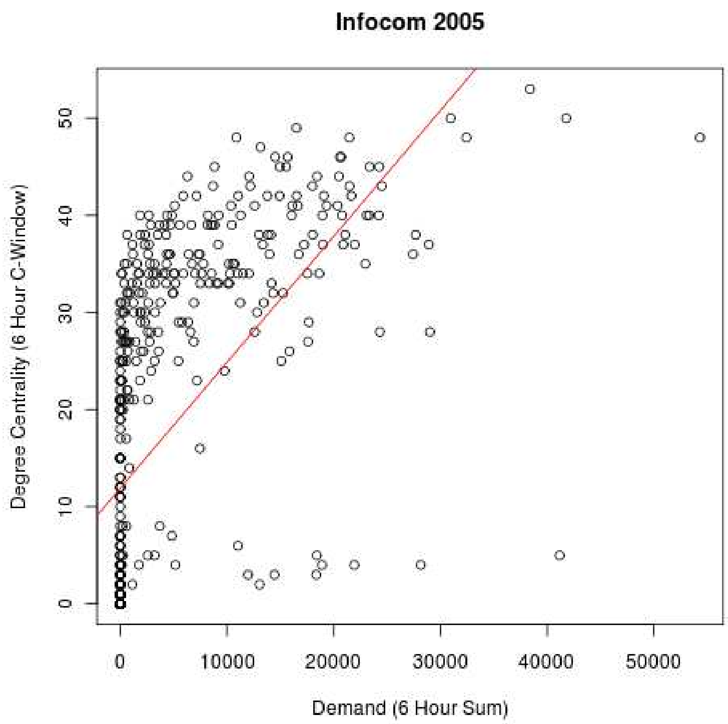


Figure 1.1: Correlation Between Degree Centrality and Demand

The Demand in Figure 1.1 correlates to a message transferal and as we have identified that a small subset of nodes are faced with 30 to 50 times more traffic than the average node in the network, it is clear that by removing the reoccurring assumption of unlimited resources, congestion is a prominent problem that needs to be addressed.

1.2 Thesis Contributions

This thesis contributes to the field of Delay Tolerant Networking by providing a framework for enabling devices to perform congestion aware operations. This framework is the result of the following core contributions:

- A congestion aware method of forwarding that manipulates traffic across multiple paths in order to avoid sending traffic towards congested or congesting nodes, which was published in the Seventh Annual Conference on Wireless On demand Network Systems and Services (WONS 2010) [29].

- An implicit clustering mechanism that enables nodes to make regionally aware forwarding decisions, allowing nodes to avoid congested or congesting regions, resulting in a less selfish forwarding strategy, which was published in IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2010) [30].

- A replication placement mechanism that ensures copies of a message are adaptively allocated to contacts, which ensures that more copies go towards nodes that are predicted to be better suited for delivering a message to its destination, provided they are not congested and the number of packets is adaptively reduced if the node is congesting, which was published in the Eighth Annual Conference on Wireless On demand Network Systems and Services (WONS 2011)[69].

- A replication copy management mechanism that allows nodes to adaptively calculate the number of copies of message to be disseminated, which allows a node to make use of available resources and back off in oder to preserve resources when they are limited, which has been published in the 7th International Wireless Communications and Mobile Computing Conference (IWCMC 2011) [70].

- A Peer-2-Peer application model, which simulates a real world application in order to achieve realistic traffic demands within the simulation environment, which has been published in SIGMOBILE Mobile Computing and Communications Review (MC2R) [68].

- An extensive evaluation by means of simulation over a number of real device connectivity datasets, with both a variety of traffic demands and buffer size.

- A social application model, which is derived from real world data, which has been collected from 100 Facebook users, in order to achieve realistic traffic demands within the simulation environment.

1.3 Thesis Structure

The remainder of this thesis is structured as follows.
Chapter 2 reports the related literature surrounding the focus of this Thesis, specifically: delay tolerant forwarding techniques (social and non-social), congestion control methods for traditional networks, MANETs and DTNs and literature regarding the effective use of network resources in order to increase productivity.
Chapter 3 identifies the challenges this Thesis is presented with, the research criteria for this Thesis, an analytical model of the problem domain, an overview of our framework, a detailed description of our proposed solution and a description of our algorithm.
Chapter 4 is concerned with the procedure used in order to achieve a rigorous analysis of our algorithm, which comprises of: a description of the three real world connectivity datasets used to encapsulate real device encounter patterns, a detailed explanation of the application traffic models and their utilisation within the simulation environment and a detailed description of our emulation configuration.
Chapter 5 is concerned with the rigorous evaluation of our work, such that performance can be evaluated independently of inter-contact times and contact

durations, which result from the underlying network topology. Specifically, we describe the performance characteristics of our algorithm evaluated across three vastly different connectivity datasets, with increasing traffic demands, generated by our distributed file casting application [68].

Chapter 6 is concerned with the effects that different traffic models have on the performance of our work, more specifically this Chapter evaluates the impact that real world social networking usage patterns have on both benchmark DTN forwarding algorithms and state-of-the-art DTN congestion control algorithms in comparison to our algorithm.

Chapter 7 discusses the possible additional applications of our framework, such as using our congestion indicator to adaptively apply network coding or to incorporate additional or alternative signals, such as transmission energy costs.

Chapter 8 contains our concluding remarks and proposes future directions in which our work can be evaluated and enhanced.

Chapter 2

Literature Review

In this chapter we focus on a wide range of state-of-the-art techniques that address message forwarding, congestion control and enhancing productivity. We consider literature from a wide range of research areas within the networking community, including: DTN, MANET, Peer-to-peer and Internet networking. We discuss what we have learnt from the existing work and outline why each of these techniques do not answer our research problem.

The rest of this chapter is structured as follows: in Section 2.1 we give an overview of message forwarding techniques, which are categorised into Social (Section 2.1.2) and Non-Social (Section 2.1.1) methods. In Section 2.2 we review congestion control methods for the Internet (Section 2.2.1), MANETs (Section 2.2.2) and DTNs (Section 2.2.3), showing that non of the existing work resolves our research problem. In Section 2.3 we focus on the literature regarding the effective use of network resources in order to increase productivity, more specifically: Resource Pooling (Section 2.3.1), Multi-path Routing (Section 2.3.2), Peer-to-peer Replication Strategies (Section 2.3.3), Clustering (Section 2.3.4) and Network Coding (Section 2.3.5).

2.1 Forwarding

A forwarding strategy is an algorithm that aims to forward a message to a given destination. This differs from routing because end-to-end connectivity is not assumed and as such the emphasis is more on selecting who to transfer the message to next as a node encounter other nodes, rather than calculating the shortest path from a map of the entire network.

2.1.1 Non-Social based forwarding

Direct Delivery [76]

Direct Delivery routing operates simply by the source node keeping its messages in its buffer until it encounter the destination nodes. This model implies either that nodes are mobile and as such they will eventually encounter the recipient or that connectivity for some reason is periodic and the node simply has to wait until the connection to the recipient is live again.

Epidemic Routing [39]

Messages are flooded through the network using flooding, high priority messages may have a higher hop count in order to improve delivery rate and reduce latency. Each host stores a list of messages that it is sending or has received (and subsequently is forwarding), this list of messages is sent to nodes that appear in this nodes neighbourhood, any previously unseen messages are then transmitted.

Spray And Wait [77]

Spray And Wait makes use of replication, predetermining the number of copies in a static non-adaptive way. This algorithm has two phases: the spray phase, which involves the sender distributing the copies to encountered nodes; and the wait phase, in which the nodes that are carrying the message copies follow the Direct Delivery [76] method of routing on behalf of the sender.

Spray And Focus [78]

Spray And Focus similarly to Spray And Wait disseminates a predetermined number of copies of a message to encountered nodes. Spray And Focus has a more complex second phase than Spray And Wait, as the intermediary nodes also forward the message replicas. In the second phase an intermediary node follows the single-copy utility-based scheme from [79], which states that a message is forwarded if an encountered node has a smaller average time between encounters with the destination node.

2.1.2 Social-based forwarding

LABEL [34]

LABEL is a middle ground between Direct Delivery and Epidemic that uses a social label to reduce the number of packets that are disseminated, a source node waits until it finds a node in the same community (i.e. a node that has the same label) as the destination before forwarding.

Probabilistic routing (PRoPHET) [54]

When a source node sends a message PRoPHET selects a subset of nodes in the neighbourhood to forward the message to, this selection is achieved by ranking nodes based upon their predicted probability to successfully deliver the message. PRoPHET is effective but requires a period of training in order to calculate the probabilities. Furthermore, because of the amount of network information that is required to make the forwarding decisions the size of PRoPHETs routing tables can grow rapidly.

Bubble Rap [35]

The main concept in this work is that each node calculates a windowed degree centrality value, which is used to decide how effective the node will be in the forwarding process. The centrality value is used in order to decide which neighbouring nodes a message is forwarded to, only nodes with a higher centrality than the current sending node is sent the message. A node's centrality is measured by the number, and quality of contacts encountered by the node, the value is windowed by splitting up a day into 6 hour periods (Morning, Afternoon, Evening and Night), the authors show that a nodes connectivity patterns can become more meaningful when they are grouped this way.

Encounter Based Routing (EBR) [60]

EBR is a quota based replication protocol that aims to obtain high message delivery ratios and good latency performance, while maintaining low overheads. EBR aims to lower the overhead required to deliver a message by reducing the total number of messages exchanged, EBR facilitates this by forwarding more copies of a message to nodes that are better connected. The connectivity of a node is calculated as an exponentially weighted moving average of a node's windowed degree centrality.

SimBetTS [18]

SimBetTS is a complex utility-based social forwarding strategy, which incorporates three significant concepts: Betweenness Centrality, Node Similarity and Tie Strength. Betweenness Centrality is calculated using a locally calculated (ego network) representation of the number of nodes that are indirectly connected through a given node. Node Similarity is calculated as the number of common neighbours between a node and the destination. Tie Strength is a combination of contact Frequency, Closeness and Recency. Frequency is defined as the total number of times two nodes have encountered each other, in comparison to their total number of node encounters. Closeness is defined as the total amount of time two nodes have been connected, in comparison to their total duration of connectivity. Recency is defined as the duration of time since two nodes last encountered, in comparison with the duration each has been active.

2.1.3 Discussion

A common vulnerability of the Forwarding algorithms featured in this Chapter is the inability to achieve the high performance they promote when network resources are not unlimited and traffic levels are challenging. [66] argues that in a simplified network scenario where nodes forward randomly, better connected nodes have a higher probability of receiving a packet, which indicates that load distribution is naturally skewed towards better connected nodes. Additionally to this [66] argue that by making informed forwarding decisions based on a heuristic that favours connectivity, load distribution becomes even more unbalanced.

The forwarding algorithms in this Chapter greedily select a next-hop, as they each define a heuristic that is used to select the shortest possible path, with no consideration regarding other nodes in the network. This Thesis is motivated by the observation that, when following a forwarding strategy, all the data in the network is traveling towards an ever decreasing subset of nodes, which have a greater heuristic than the current sender, in order to maximise delivery ratios and minimise delays, but as traffic demands increase, these nodes become congested and consequently unusable. The concepts of selfish flow and selfish routing in static (time-invariant) networks have been thoroughly explored over the last few years [71, 85]. Work such as [46] assesses the loss of efficiency caused by selfishness of individual participants.

2.2 Congestion Control

This Section reviews a wide range of existing congestion control techniques categorised into Internet congestion control, MANET congestion control and DTN congestion control. The Internet congestion control section reviews the congestion control mechanisms commonly deployed throughout the Internet. The MANET congestion control section evaluates the clean slate approaches to congestion control, which are not TCP centric. The DTN congestion control section inspects the state-of-the-art congestion control mechanisms specifically developed for our problem domain. We outline what we have learnt from the existing congestion control mechanisms and outline the reasons why this work does not solve our research problem.

2.2.1 Congestion Control In The Internet

This Section discusses existing congestion control mechanisms as developed for the Internet, more specifically the Internet. Congestion control was not an original component within the Internet architecture. In 1986 the Internet began to suffer from congestion-collapses, resulting in TCP (the main bandwidth consumer at the time) being retrofitted with a congestion control mechanisms, which have since been a critical factor in the robustness of the Internet. Recent years have seen the rapid emergence in time critical applications, such as media streaming. Time critical applications prefer timeliness to reliability resulting in developers turning to UDP, a protocol without a congestion control mechanism, this has led to the development of Datagram Congestion Control Protocol. The remainder of this section discuss both TCP congestion control and Datagram Congestion Control Protocol.

TCP Congestion Control [4]

TCP uses four algorithms in order to avoid congestion collapse, as specified in [36]: slow start, congestion avoidance, fast retransmit and fast recovery.

The core concept is that a TCP flow maintains a congestion window (cwnd), which is a sender-side limit on the amount of data the sender can transmit into the network before receiving an acknowledgment. The receiver also advertises a window size (rwnd) in order to inform the sender of its limitations, the minimum of cwnd and rwnd is used to regulate the traffic rate.

The slow start algorithm is used at the beginning of a transfer, or after detecting loss. During slow start, the cwnd is incremented each time an acknowledgment is received that acknowledges new data. When the slow start algorithm increases the cwnd past the slow start threshold TCP switches to the congestion avoidance algorithm. During congestion avoidance, the cwnd is additively increased per round-trip time (RTT). Congestion avoidance continues until congestion is detected.

If a TCP receiver has a message arrive out-of-order it sends an acknowledgment back to the sender specifying the sequence number that was expected. An out-of-order acknowledgment can be caused by dropped messages, in-network re-ordering of messages, by replication of an acknowledgment or message by the network, or when a message fills a gap in the ordering (this is for the benefit of loss recovery algorithms, such as NewReno [27]).

The fast retransmit algorithm interprets 3 duplicate acknowledgments as indication that a message has been dropped, which triggers the message specified as the next in the sequence in the repeated acknowledgments to be sent. After the fast retransmit algorithm has retransmitted the message that is assumed to have been dropped, the fast recovery algorithm manages the transmission of messages. The fast recovery algorithm is used in place of the slow start algorithm as it artificially inflates the cwnd in order to utilise the resources intended for the message that is assumed to have left the network.

Datagram Congestion Control Protocol (DCCP) [44]

TCP Congestion Control is crucial for the Internet to maintain stability. [25] identifies the negative impact of non-congestion-controlled best-effort traffic as ranging from extreme unfairness against competing TCP traffic to the potential for congestion collapse.

TCP offers reliability and congestion control and as such is predominantly used to transfer data within the Internet, but for some applications, such as streaming media, transfer speed outweighs reliability. DCCP has been developed in order to provide congestion control to fast, but less reliable traffic transfer protocols such as UDP.

DCCP offers the functionality for congestion control mechanisms to function, but it allows the application the choice of which algorithm to use. The two congestion control mechanisms that have currently been developed are TCP-like Congestion Control and TCP-Friendly Rate Control (TFRC) [26].

TFRC is an end-to-end equation-based congestion control mechanism for best effort traffic, which adapts to congestion without responding in an unnecessarily severe way upon receiving a single congestion indication. The main differences in design principals between TCP Congestion Control and TFRC is that sending rate is not aggressively increased, it is increased gradually in response to a decreased loss rate, and sending rate is not halved on receipt of a single loss acknowledgement, instead it is halved in response to multiple successive loss acknowledgements.

2.2.2 MANET Congestion Control

This section discusses the related field of MANET congestion control. [57], a survey on congestion control for MANETs, identifies that the majority of work in this field is concerned with improving the performance of TCP congestion control in MANET scenarios. EXplicit rAte-based flow ConTrol (EXACT) [14] and Ad-hoc Transport Protocol (ATP) [80] are in-network, rate based flow control. They are clean slate approaches that are not concerned with TCP congestion control optimisation.

EXplicit rAte-based flow ConTrol (EXACT) [14]

EXACT nodes determine a current bandwidth that they can supply to neighbouring nodes, calculating fair bandwidth shares for each flow. EXACT senders set the two variables in each packet header, a maximum sending rate field that is based on the sender's link capacity and a current sending rate. EXACT intermediary nodes keep track of flows and assigned sending rates, which involves measuring the current bandwidth of the outgoing wireless links, computing the available rates for the current flows and updating the header of each passing data packet to inform end nodes of any changes in resource availability. EXACT receivers send the available line capacity information back to the sender in an acknowledgement.

Ad-hoc Transport Protocol (ATP) [80]

ATP does not require any flow-specific state variables in the intermediate nodes, instead nodes calculate an exponential average of the delay of all packets passing through them in order to calculate the available bandwidth. Each node in the network maintains two parameters, an exponential average of the queuing delay and an exponential average of the transmission delay. The two parameters monitor same flow and opposing flow contention. Each node, including the destination node, update the packets that pass through their buffer with the sum of these two delay signals, provided they have a higher delay observation than all preceding recipients. The sender obtains this information from the receiver in the form of an acknowledgement, based on its current rate, and the rate specified in the acknowledgement determines whether the sender increase, decrease, or maintain its rate.

2.2.3 DTN 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 [83, 54, 35, 18, 77, 78]. The majority of research in congestion control for DTNs is concerned with buffer management [6, 13, 56, 74]. Recent developments have been concerned with replication management [74, 82, 68] and distribution [60, 66].

Queueing Strategies

[56] evaluates a number of queueing strategies, more specifically: First in first out (FIFO), Evict most forwarded first (MOFO), Evict most favourably forwarded first (MOPR), Evict shortest life time first (SHLI) and Evict least probable first (LEPR).

FIFO drops messages based upon the order in which they enter the buffer, such that the first message that entered the queue is the first to be dropped.

MOFO attempts to maximise the dissemination of messages through the network by dropping the message that has been forwarded the largest number of time, enabling messages that have a lower hop count to travel further within the network.

MOPR keeps a value for each message in its queue and each time a message is replicated the message value is increased based on the predictability of the message being delivered, the message with the highest value is dropped first.

SHLI uses the message timeout value, which specifies when it is no longer useful, such that the message with the shortest remaining life time is dropped first.

LEPR functions by a node ranking the messages within its buffer based on the predicted probability of delivery, the message with the lowest probability is dropped first.

Resource Allocation Protocol (RAPID) [6]

RAPID models DTN forwarding as a utility-driven resource allocation problem. Routing is achieved by prioritising messages to be forwarded and messages to be dropped based upon a utility function. The utility metric is dependant on the goal of the network, RAPID defines 3 metrics: Minimising Average Delay, Minimising Missed Deadlines and Minimising Maximum Delay. When using the Minimising Average Delay Metric a node attempts to greedily replicate the message that reduces the average delay among all packets in its buffer. The Minimising Missed Deadlines Metric replicates the message that has the highest probability of being delivered within its deadline. The Minimising Maximum Delay Metric replicates the packet with the earliest creation time in order to minimising the maximum delay for each message.

Storage Routing (SR) [74]

SR avoids congested nodes dropping packets by sending a set of messages out to neighbours with available storage, when buffer capacity is free the previously expelled messages are retrieved. SR comprises of two algorithms: a node selection algorithm and a message selection algorithm. The node selection algorithm targets alternative custodians needed to deliver the message either by forwarding towards the destination or back through the congested node, by allowing the message to loop. The node search is implemented as a form of expanding ring search in which all nodes up to k hops away are considered for migration. The message selection algorithm chooses either the the first message in the buffer, the last message in the buffer or the oldest message in the buffer.

Autonomous Congestion Control (ACC) [13]

ACC implements congestion control by applying a financial model to buffer space management, in order to propagate buffer utilisation stress backwards through the network to the source nodes. Unoccupied buffer space is modelled as money, network traffic as the daily financial activities of an investment banker. A router has limited buffer space similarly to the amount of capital a banker has to invest. If a router manages to forward traffic it receives commission, as the remaining balance of a nodes spending becomes low the node has to economise, thus becoming unwilling to accepting high risk messages.

FairRoute [66]

FairRoute argue that considering only contact histories when selecting a next-hop node cannot achieve balanced traffic distribution. FairRoute proposes that in addition to contact histories, a node's queue length is evaluated, such that a node with a larger number of messages in it's queue is nominated more often as a next-hop. FairRoute also argue that more messages in a node's queue indicates that the node is more popular and therefore more capable of forwarding the messages it receives. The result of a node only being able to forward to nodes with larger queue sizes in order to achieve load balancing holds similarities to back pressure congestion control.

Density-Aware Spray-and-Wait (DA-SW) [82]

DA-SW is a adaptive replication variant of the spray-and-wait [77] algorithm. The way in which DA-SW selects the number of messages to replicate is by consulting a precomputed abacus with the measurements obtained from previous deployments. This abacus is the result of exhaustive offline training, making the end product bespoke for a given environment, therefore unless the network exhibits the exact same characteristics again DA-SW will not function effectively. The DA-SW abacus correlates the optimal number of copies to disseminate with the trailing window degree centrality (the number of encounters in the past 30 seconds).

Retiring Replicas (RR) [81]

RR is an adaptive replication algorithm that adjusts a node's initial replication limit based on a locally perceived global level of congestion. RR monitors the level of congestion (CV) by calculating an exponentially weighted moving average of the ratio of dropped and replicated packets during a window of time. When a node notices either equivalence or an improvement in CV the replication limit is additively increased, when the CV value worsens the replication limit is multiplicatively decreased. RR introduces an online adaptive replication technique that treats the network as if it has a globally uniform level of congestion, but this is not typical in such fragmented networks, as different islands of connectivity each have their own traffic demands and as the topology of the network changes burst of traffic spread across the network producing congestion hotspots.

2.2.4 Discussion

We have presented Internet, MANET and DTN congestion control methods.

Internet congestion control methods, TCP Congestion Control and DCCP, both assume end-to-end connectivity and as such use a closed loop acknowledgement-based system, which relies on a feedback loop in order to control the dynamic behaviour of the system. Internet congestion control systems are sender or receiver driven and thus intermediary nodes have little or no involvement.

MANET Congestion Control mechanisms EXACT and ATP monitor in-network congestion signals in order to allow for more dynamic rate control, allowing traffic to change path instantaneously, but they still operate with a closed loop acknowledgement driven methodology.

Existing DTN congestion control methods each has its merit, but none successfully answers our research question, as they do not consider the problems associated with social forwarding, limited resources and multi-copy dissemination. DTN Queueing Strategies, including RAPID, are needed in order for nodes to cause the minimal amount of damage to the network when forced to discard messages, but do not address the means of preventing the network reaching such a critical state. SR avoids the need to drop messages by temporarily sending messages out to surrounding nodes, but this is not sufficient as at best SR only manages to temporarily alleviates congestion as it relies on contacts being present and not suffering from congestion themselves.

ACC and FairRoute achieve congestion control similarly to back pressure congestion control. ACC applies a financial model to manage buffer space allocation, while FairRoute imitates a social hierarchy based upon popularity, which is gauged by the number of messages received by a node. The problem with these solutions is that the network has to become severely congested before the sender is restricted and key nodes in the network will still be the first to congest.

DA-SW and RR are congestion control methods centred around the concept of adapting the message replication limit in response to network conditions. Similarly to Queueing Strategies, adapting the message replication limit is necessary, but does not answer our research question, as they assume that congestion is uniform and therefore does not address the issue of individual nodes or regions congesting.

2.3 Additional Mechanisms

This Section discusses the state-of-the-art methods in Internet, peer-to-peer, MANET and DTN literature for efficiently utilising network resources. This Section also reviews publications which identify valuable untapped resources, typically unused due to the greedy characteristic of existing methods.

2.3.1 Resource Pooling

Mechanisms such as load balancing, failure resistance, multiplexing and network coding are all methods of resource pooling as they achieve a network state that behaves more like a single pooled resource. [87] state that load-dependent routing has proved elusive to conventional Internet routing and remains the holy grail of routing systems. Wischik et al. [87] believe that the natural evolution of the Internet should be to harness multi-path-capable end systems in order to achieve resource pooling, with the aim to increase reliability, flexibility and efficiency. [87] outline problems with current Internet mechanisms, such as slowness to recover from failures and bad interactions between the application and network level.

2.3.2 Multi-path Routing

Multi-path Routing of TCP in MANETs

[51] is an experimental study of TCP performance over multi-path routing in MANETs that identifies mobility as one of the major factors degrading TCP performance. [51] builds on top of Split Multi-path Routing (SMR) [48], which is an extension of Dynamic Source Routing (DSR) [38]. [51] identify that using TCP concurrently over SMR's multiple paths shuffles delivery order, consequently causing TCP to initiate its congestion control mechanisms, which results in drastically reduced throughput. [51] also investigates TCP over one path at a time and keeping backup routes, which are ordered by shortest hop and shortest delay that a sender can quickly switch to when the current path breaks.

Path Explosion Phenomenon [21]

Path Explosion Phenomenon is the occurrence of multiple different paths to a destination node that arrive within a short space in time, showing that there are many near optimal paths in a DTN. [21] identify that typically iMote (small bluetooth connectivity logging devices) connectivity traces exhibit the Path Explosion Phenomenon, such that the majority of messages (97%) within a short space of time (150 seconds) have the opportunity to arrive at the destination node by following one of a large number of paths (2000).

2.3.3 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 others data in order to increase data availability [72], also resulting in the pooling of the upload capacity of many network nodes [87].

BlueTorrent [41] is an example of a peer-to-peer application over a Bluetooth ad hoc network, the simulation results show that P2P file sharing improves download times in comparison to typical AP only dissemination. [72] studied the problem of replica placement in a P2P system intending to optimise availability and/or the number of replicas. [72] 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 [72, 67]. The most common approach to P2P replication is the random distribution of copies [17, 10]. [9] analyses how many randomly placed replicas are required to achieve a desired level of availability. [72] argue that replication should not be random, but be based on cliques of peers replicating each others data, limiting the selfishness of the participants.

2.3.4 Clustering Schemes

In order for forwarding decisions to be made in a less greedy way, nodes can organise themselves into groups (clusters) - these clusters can then act as a collective in order to achieve the best solution for the group rather than greedily for a given individual. In this section we are going to discuss the clustering schemes developed for MANETs and DTNs, which divide nodes into different virtual groups that are used in order to synthesise a hierarchical structure similarly to that exhibited by the Internet. In order for these schemes to function nodes are categorised as cluster-head, cluster-gateway, or as cluster-member. The cluster-head is central and is in charge of organising the virtual group. The cluster-gateway is a node who is elected as a communication path into other clusters. The remaining nodes are cluster-members who relay messages to the cluster head.

MANET Clustering Schemes

The main objective for MANET clustering techniques is to overcome scalability issues and provide throughput and delay performance guarantees. MANET clustering schemes are split into dominating set based, low maintenance, mobility-aware, energy-efficient, load-balancing and combined-metrics-based. The majority of clustering schemes are high-maintenance and are not suitable for environments that typically suffer from a large number of disconnections. Passive Clustering (PC) [28] is a low-maintenance scheme designed to function within environments that observe frequent topology changes, which could result in the generation of an excessive number of control overheads. PC elects a cluster-head with the first declaration wins rule, which simply elects the first node to broadcast a message within a network region as the cluster-head. A cluster-head only resigns if no cluster-gateway nodes exist for a predefined period of time.

DTN Clustering Schemes

Existing work regarding DTN clustering [19, 32, 1] attempts to establish and maintain hierarchical cluster meta-information for every contact encountered, in order to route traffic using a clustered map of the network.

Bottom-up Hierarchical Clustering [1] produces overlapping clusters, which are formed based upon node encounter frequency. Initially each node is considered the cluster-head of it's own single node cluster, as nodes meet the two closest nodes (nodes that have the highest encounter frequency) form a new cluster. The forwarding approach only forwards a message to a node if it shares a cluster with the destination, the receiving node then epidemically forwards to all other cluster members.

Controlled Routing Based on Hierarchy Forwarding and Cluster Control [32] is a mobility aware clustering technique which selects the least mobile nodes (Steady Nodes) to become dominating nodes, which form a Steady Network that is intended to provide a communication backbone. Nodes whose mobility behaviour is sporadic and cannot be predicted are then able to forward messages via the Steady Network in order to communicate.

Contact Probability Clustering [19] triggers a synchronisation process when two nodes meet, which checks to see if they should be in the same cluster, membership is based upon contact probability. This synchronisation is also triggered by a timeout if the two nodes are in the same cluster and have not encountered each other for a long period of time, because they may have encountered other nodes with greater shared contact probability. The routing strategy involves the sender consulting its clusters routing table to see if a cluster-gateway exists with the cluster ID for the cluster containing the destination, if so the node forwards the message to this cluster-gateway, which employs the Direct Delivery forwarding strategy.

2.3.5 Network Coding

Internet networking achieves the upper bound of unicast throughput by using max-flow/min-cut algorithms. In more complex network scenarios where messages are destined for more than one receiver information theory techniques can be applied such that packets can be combined without increasing the message size and later split, thus allowing for better utilisation of transfer bandwidth.

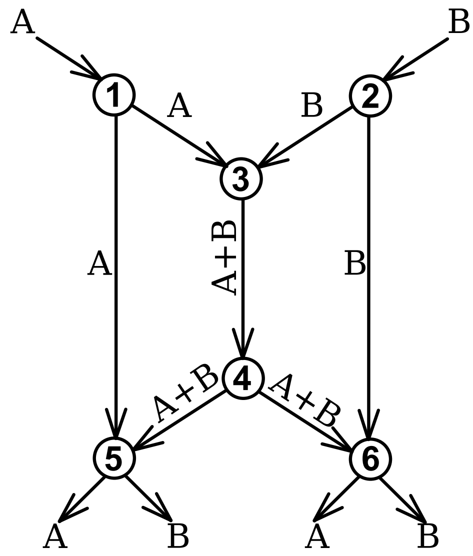


Figure 2.1: Butterfly Network

Figure 2.1 illustrates a network scenario where two sources (nodes 1 and 2) are forwarding one message each (A and B respectively) to two destinations (5 and 6). If routing is used in order to transfer these messages, only one message at a time can be transmitted between nodes 3 and 4, thus either destination 5 will receive A twice or destination 6 would receive B twice. Figure 2.1 shows that by combining message A and B at node 3, at nodes 5 and 6 the message can be split by subtracting the distinct message they have received from the combined message in order to obtain the other distinct message.

Delay Tolerant Network Coding

Network coding in delay tolerant networks has been explored by extending Epidemic [53], Binary Spray (E-NCP) [52] and Probabilistic [86] forwarding algorithms, such that all messages received by an intermediary node are combined before forwarding. HubCode [2] and FairMix [42] differ to these early works, as they investigate methods of selecting which nodes should mix and which messages should be mixed respectively.

HubCode [2] identify that coding at every node is computationally expensive and results in a large control overhead, as nodes have to exchange the coefficient matricies of the encoded messages. HubCode suggests similarly to many cluster-head selection algorithms that the best connected nodes in the network region should be the nodes which have additional responsibilities. HubCode has two variants HubCodeV1 and HubCodeV2. HubCodeV1 utilises traditional network coding [89] and exchanges the large control messages typical of this method, but with the benefit of restricting this to between highly central nodes (hubs). HubCodeV2 requires intermediary nodes to decode the messages they receive and re-code outgoing messages, as a result nodes only need exchange message ID information, which results in more focused message combinations, lower control overheads, but at the cost of higher computational overheads.

FairMix [42] identify that existing work, which mixes a blocks of packets indiscriminately, leads to large unfair decoding delays. The decoding delays are a result of destination nodes having to wait until they have received a potentially large amount of data before they can decode the data and the retrieve the individual messages. FairMix alleviates this unfairness by restricting message encoding to messages which share the same destination. This is achieved by nodes maintaining virtual queues which segregates messages destined for different destinations.

2.3.6 Discussion

Resource pooling shows that nodes could benefit from pooling capacity of the many and varied connections with intermittently connected nodes, which could move traffic away from more congested regions provided the next-hop selection process utilises the correct congestion signals. Resource pooling is supported by the fact these networks exhibit the path explosion phenomenon, a consequence of this phenomenon is that, in the majority of cases, nodes loose little performance by forwarding to a next-hop node that is not optimal (greedily selected).

Peer-to-peer replication shows the benefits of replicating information within a network which suffers form disconnections, in order to increase availability. This work also identifies the difficulty of optimising this, even with a global level of knowledge. This gives us a good understanding of the limiting factors and requirements to consider when designing our replication placement strategy.

MANET Clustering techniques are typically too high maintenance for DTNs. Passive Clustering offers a low cost clustering solution, which could be used in order for nodes to structure dense pockets of connectivity, in order to reduce the number of packet collisions, but offers little aid to the structuring of large fragmented networks. Existing DTN Clustering algorithms attempt to maintain routing information for every pair of network nodes by actively propagating cluster updates, similarly to route update propagation techniques employed by proactive MANET routing protocols, but it is well understood that DTN forwarding needs to be more dynamic than this.

It is clear that network capacity can potentially be increased by adopting network coding, but is computationally expensive and can result in large control overheads and increased delays due to potentially large amount of data required before a node can decode its messages.

Chapter 3

Model Formulation

In this Chapter we identify the challenges (Section 3.1) this Thesis is presented with, outline our research criteria (Section 3.2) in response to these challenges and analytically describe the problem domain (Section 3.3). We give an overview of our framework (Section 3.4), followed by a detailed description of our proposal (Section 3.5) and end this Chapter by describing our algorithm, identifying the possible outcomes of our message forwarding decision process (Section 3.6).

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.

3.1.1 Distributed decisions

The limited connectivity and fragmented nature of opportunistic networks mean that it is neither possible to obtain and maintain a global level of knowledge of the network or to rely on closed-loop (acknowledgment based) techniques for any decisions. Rather, each device in the DTN network must act independently and base on the limited localised knowledge with the aim to enable better performance of the whole network.

3.1.2 Limited resources

Buffer capacity, transfer bandwidth and battery life are limited. As traffic demands increase, messages can only be transferred successfully if devices consider the availability of these resources when selecting a next hop.

3.1.3 Network density and Localised surges in traffic

Node connectivity is sporadic and islands of connectivity 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, with limited resources, communicate in an efficient, adaptive and robust manner when there are a large number of traffic demands, with different data sources and destinations?

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

3.2.1 Efficiency

We define efficiency in terms of providing support for minimising the traffic latency and optimising utilisation of network resources.

Minimise traffic latency

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.

Optimise utilisation of network resources

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.

3.2.2 Adaptation

DTN nodes typically have 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, nodes have to adapt quickly when information is updated, which raises the question of how the individual nodes can get feedback about the remote network state and how they can modify their own behaviour in order to affect it.

We intend to provide an adaptive mechanism that achieves network-wide optimisations by completely relying on localised aggregated node heuristics and multi-dimensional metrics.

3.2.3 Robustness

We define robustness in terms of providing increased message resiliency, via fail-safe mechanisms which prevent packet loss causing failed message delivery, and collaboration between the nodes.

Increase packet resiliency

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.

Collaboration

Whilst we assume that nodes in the network follow the prescribed algorithm, we still have to ensure that the algorithm itself is not greedy. 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 a time series and depict the vertices as $V = \{V_t : t \in T\}$ and the edges as $E = \{E_t : t \in T\}$. 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, commonly used to depict the route that a message is transmitted along, is an alternating sequence of vertices and edges. In this work the sequence index represents the time interval and the sequence element represents the location of the message at a given time. Intuitively the best route is the path with the smallest resource cost, which is 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 tradeoff between increasing the storage occupation and increasing the hop count. Using alternative paths is particularly suitable for opportunistic networks, due to the path explosion phenomenon [21].

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 in the demand set F that contain 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 a given time t as a result we denote stress as $S_x^t = \frac{D_x^t}{C_x}$ which is a measure of demand D_x^t of a resource x at a given time t against the capacity of the 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 between the source and the destination. As message can be replicated, in this case one or more copies of the message are transmitted, each following an independent path. The number of paths is limited by a replication limit and the cost of delivery for a replicated message is the sum of the costs associated with all paths 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 a 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 although it can be the best solution in a network with no opposing traffic, but this is not realistic as 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 demand D and buffered demand B over the number of available outlets (degree centrality d) at a given time t.

We define resource consumption as subgraph $G' = (V', E')$ where the set of vertices are defined as $V' = \{\forall x_t \in V : D^t_x > 0\}$ and the set of edges is defined as $E' = \{\forall e_t \in E : D^t_e > 0\}$. Network utilisation 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 capacity of the graph. The forwarding criteria H influences the size of F (the number of messages that can be forwarded) based on its utilisation of G. Given two forwarding criteria H and H' each utilising the network to the value of U and U' respectively, H has a greater capacity than H' if $U > U'$.

Existing DTN forwarding algorithms direct traffic towards the most desirable next-hop nodes, this is the optimal solution when the network is free, but as traffic demands increase these nodes become inundated, this is made more challenging as the flow of traffic is unpredictable and has a tendency to accumulate in regions of the network.

3.4 The Overview

This Section gives an overview of the design space for our Congestion Aware Forwarding (Café) and Replication (CafRep) Framework.

3.4.1 Café

Café provides a congestion aware method for single copy message forwarding. Nodes following the Café algorithm observe buffer availability, message delay and rate of congestion, in addition to social metrics, in order to make forwarding decisions. Because forwarding decisions are a combined heuristic, which consider not only the directionality a next-hop provides for the traffic, but also the level of traffic demand the next-hop is experiencing, messages are forwarded to a larger variety of nodes than competing algorithms. By forwarding to a larger set of neighbouring nodes the level of congestion in the network is kept low, this means that queue sizes are lower and message transfer can be achieved at a faster rate with a lower risk of packet loss.

3.4.2 CafRep

CafRep is an extension of Café that introduces the concepts of implicit clustering for regionally aware forwarding decisions, multi-copy forwarding (Replication) for increasing message transfer robustness and methods for sending rate adaptation.

Figure 3.1 shows the design space for our framework, which illustrates the different dimensions of possible approaches that can be useful for congestion aware dissemination. The vertical axis represents the Direction Influence. This, the connectivity dimension, is the characteristics such as centrality, similarity, interaction strength and other social context that identify affiliation such as clique identification and delivery probability that could be used to influence the forwarding direction. The two horizontal axes represent the resource dimension. The multi-path transport approach, such as Café [29, 69, 30, 68], introduces methods of avoiding congested regions of the network. Replication restriction

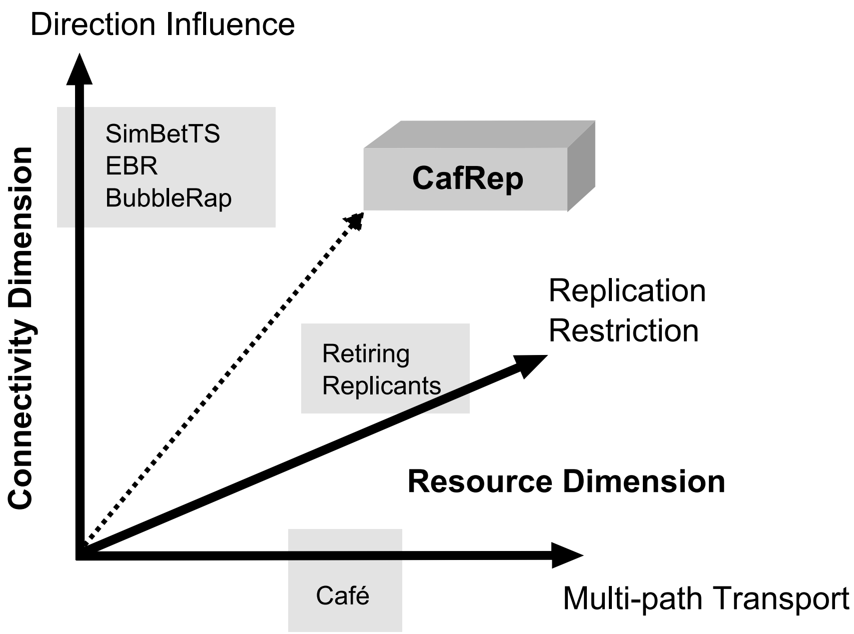


Figure 3.1: Design Space

techniques, such as Retiring Replicants, help to increase network capacity by reducing the in-network occupancy of redundant messages. Our congestion control algorithm design philosophy is to consider the combination of route optimisation with multi-path transport and sending rate restriction.

Figure 3.2 shows a multi-layer view of our conceptual model. The edges in this graph illustrate connectivity and the vertices represent the nodes. In reality the connections would go up and down over a period of time, the networks this work is concerned with are disconnected for the majority of time, but for simplicity the time series information has been flattened. In this example we have assumed the following: nodes S and D are source and destination respectively and 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, this is in the application layer as each application will have its own topics and interest groups. Both the Congestion Layer and Social Layer are within the CafRep Layer, this

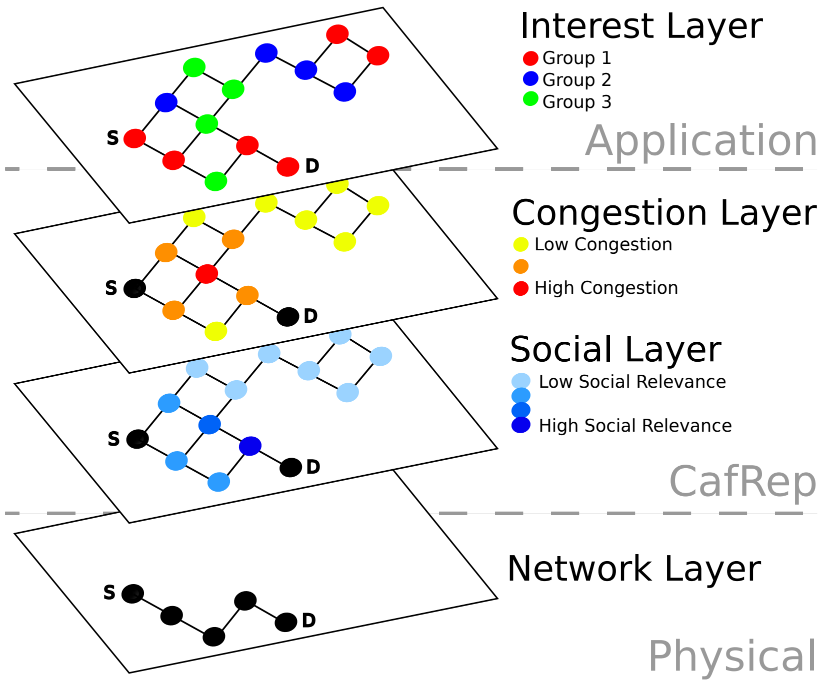


Figure 3.2: Multi-Layer View

is because we propose to broaden the next hop selection criteria allowing nodes with capacity, on perhaps less direct routes, to receive messages by monitoring both social and congestion signals. The Network Layer, on the Physical Layer, illustrates the actual route a message would take through this example network, given the tradeoff between shortest path and resource-driven routing, in order to redistribute load to avoid congestion as identified in our criteria.

Figure 3.3 illustrates the architectural overview of our conceptual model. The inputs into the system comprise of social, interest and and congestion signals for both the node itself and the contacts it encounters, forwarded and generated messages and the node's current contact set. The various signals are observed and updated by the systems social, interest and congestion monitors, each monitor then feeds its data to the dynamic replication module, along with the current contact set and the messages the node has generated and acquired. The dynamic replication module then calculates for each message if it is to be replicated, dropped or re-queued, this decision is complex and multi-

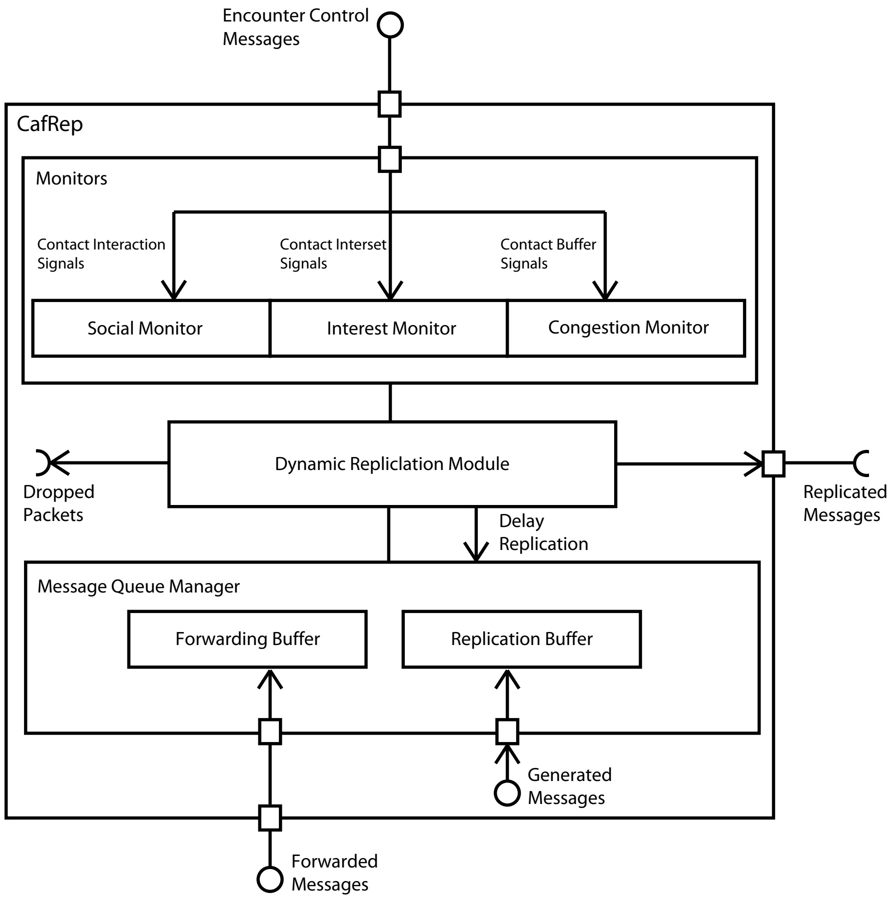


Figure 3.3: Architectural View

dimensional.

3.5 CafRep Proposal

This Section describes CafRep, a unified congestion control framework for DTN routing, which comprises of a combined adaptive forwarding and adaptive replication management.

3.5.1 Combined Utility Heuristic

Selecting which node represents the best carrier for a given message and deciding on the optimal number of replicas to forward to the selected node while trying to balance keeping latency low, chance of packet loss low and node availability high, is a multiple attribute decision problem. Our proposal aims to adaptively select both the node and the number of messages by dynamically combining different types of locally and regionally observed social and resource signals in order to provide the maximum utility for successful delivery while avoiding congested regions and keeping latency low to match our criteria given in Section 3.2. We achieve this by proposing a utility function (Formula 3.1) that is used as a measurement of relative gain, loss or equality, calculated as pair-wise comparison between the node's own congestion signals and that of encountered contacts.

$$Util_{signal}(X, C_i(X)) = \frac{signal(C_i(X))}{signal(X) + signal(C_i(X))} \tag{3.1}$$

The signals we select and monitor are important and non-trivial as they cover different dimensions of the problem in order to manage a number of tradeoffs between different challenges, as introduced in Section 3.1. At the core of a CafRep node is the *CafRepUtil*$_D$ heuristic (Formula 3.2) which is responsible for determining the overall improvement an encountered node represents when compared to a sending node and is used for choosing the next hop as well as deciding the number of message copies that are to be disseminated.

$$\mathit{CaféUtil}_D(X, C_i(X)) = \frac{1}{|U|} \cdot \sum_{u \in U} Util_u(X, C_i(X)) \tag{3.2}$$

Formula 3.2 shows the $CaféUtil_D$ heuristic, which is the sum of the set of utilities: $U = \{Ret, Rec, CR, Fwd_D, WEN_{Ret}, WEN_{Rec}, WEN_{CR}\}$. U comprises of the following comparable attributes: retentiveness (Ret), receptiveness (Rec), congesting rate (CR), and their weighted ego-network counterparts (WEN_{Ret}, WEN_{Rec}, WEN_{CR}), along with a forwarding heuristic (Fwd_D). In this Thesis, Fwd_D is the utility of the social forwarding predictor introduced in [18].

In Formula 3.2 each heuristic is weighted equally, as $CaféUtil_D$ is calculated as a simple average. We chose to make a nodes congestion level be equally important as a nodes forwarding potential, but these values could be waited allowing a node to be more responsive to either the level of congestion or the directionality towards the destination each next-hop offers.

3.5.2 Observation Histories

Each node monitors a number of congestion signals that it stores and disseminates to encountered nodes. Rather than maintaining a large array of time series data for each signal, we apply exponential smoothing (given in Formula 3.3) to the data in order to represent the information as a single value where short-term fluctuations are smoothed out and longer-term trends are highlighted making it suitable for forecasting. λ is a fraction that represents the responsiveness of the smoothing, this is typically 0.8.

$$EWMA^t(X) = \lambda \cdot EWMA^{t-1}(X) + (1 - \lambda) \cdot NEW(X) \tag{3.3}$$

3.5.3 Congestion Signals

Each of the congestion signals described in this section were discovered experimentally. By removing the assumption of unlimited resources and transfer bandwidth our experiments showed that storage was the first resource to be exploited [29]. We noticed that by adapting forwarding protocols to consider buffer availability we managed to increased success ratios, but this also increased delay [30]. When responding to both storage availability and delay we noticed that some nodes were congested for long periods of time and as soon as they became available nodes were exploiting them again, this led to us observing and reacting to a node's rate of congestion [69].

Retentiveness

This subsection defines retentiveness as the nodes or regions ability to keep possession of the packets that are sent to them. Retentiveness is an important attribute to consider because of the store and forward nature of opportunistic 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.

$$Ret(X) = B_c(X) - \sum_{i=1}^{N} M^i_{size}(X) \tag{3.4}$$

Formula 3.4 shows that individual retentiveness values are calculated as the sum of all message occupancy subtracted from the node's buffer capacity. This can be heavily influenced by the buffer policy, in this work we have used the first in first out buffer policy.

Receptiveness

This subsection defines receptiveness as the nodes or regions ability to receive packets and forward them on. This is an important observation, as increasing delay 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.

$$Rec(X) = \sum_{i=1}^{N} (T_{now} - M^i_{recieved}(X)) \tag{3.5}$$

Formula 3.5 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 ($M_{received}$).

Congesting Rate

This subsection defines congesting rate as the speed in which a node or a region is likely to congest, an important observation as it indicates if the resource offers long term stability. This congestion signal indicates the likelihood of traffic spikes that could cause the message to be dropped.

$$CR(X) = \frac{T_{FullBuffer}(X)/T_{TotalTime}(X)}{1/N \cdot \sum_{i=1}^{N}(T_i end(X) - T_i start(X))} \tag{3.6}$$

Formula 3.6 shows that congesting rate is calculated as the proportion of time a buffer is full divided by the duration between periods of congestion. A node is said to be congested when the amount of available buffer is below a given watermark. $T_{FullBuffer}(X)$ is the proportion of time a node's buffer is full, $T_{TotalTime}(X)$ is the proportion of time a node has been monitoring its buffer, and $T_i start(X)$ and $T_i end(X)$ are the start and end times of periods where a node has not been congested respectively.

3.5.4 Collaboration

In this Section we describe how CafRep provides collaboration by implicitly clustering nodes. We use the term clustering in this section to describe the unification of nodes, such that regionalised decisions can be made. By aggregating observations disseminated by encountered nodes, such as Retentiveness, Receptiveness and Congesting Rate, a CafRep node forms a ego-network perspective of the level of congestion within the network. Nodes share this aggregated regional information in order to resolve routing decisions, therefore messages are not only forwarded towards one node, but also an implicit cluster of nodes, resulting in a less selfish forwarding strategy.

An ego-network can be defined as a network consisting of a single node (ego) together with the nodes they are connected to (alters) and all the links connecting the alters, each node therefore has their own perspective of the network in the form of their own ego-network. Each node as an ego collects congestion signal information when it encounters alters. The way in which a node aggregates it's ego-network information affects the quality of the representation of the ego-region and therefore the effectiveness of the routing decisions made based on these aggregated observations.

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 ego-network congestion levels in order to better performance for both forwarding and replication allocation: More specifically, we have considered techniques such as simple average (Formula 3.7), exponentially weighted moving average (Formula 3.8) and relative social weighting (Formula 3.9) of the nodes ego network congestion signals.

$$EN_s(C(X)) = \frac{1}{N} \cdot \sum_{i=1}^{N} s(C_i(X)) \tag{3.7}$$

Formula 3.7 shows how the simple average of ego-network congestion signals is calculated. In Formula 3.7 the congestion signal (s) is being aggregated at an equal weight for every contact ($C_i(X)$) in the contact set ($C(X)$) of the ego-node (X).

$$WEN_s^e(C_i(X)) = \lambda \cdot WEN_s^{e-1} + (1 - \lambda) \cdot s(C_i(X)) \tag{3.8}$$

Formula 3.8 shows how the exponentially weighted moving average aggregation of ego-network congestion signals is calculated. In Formula 3.8 the congestion signal (s) is being aggregated at every contact encounter (e), where λ represents the degree of influence a new observation has, we typically set this to 0.8. The benefit of this method over the simple average is that newer congestion signals are more prominent, as a result routing decisions can be made with more relevant observations.

$$SWEN_s(X, C(X)) = \sum_{i=1}^{N} (SW_{C_i(X)}^{X} \cdot s(C_i(X))) \tag{3.9}$$

Formula 3.9 shows how the relative social weighted aggregation of ego-network congestion signals is calculated. In Formula 3.9 the congestion signal (s) is aggregated by weighting each encounters signal ($s(C_i(X))$) based on the relative social proximity ($SW^X_{C_i(X)}$) of the contacts in the ego-node's (X) encounter set ($C(X)$).

$$SW^X_{C_i(X)} = \frac{1}{4} \cdot (S^X_{C_i(X)} + R^X_{C_i(X)} + D^X_{C_i(X)} + F^X_{C_i(X)}) \tag{3.10}$$

Formula 3.10 shows that the relative social weight is calculated as the average of a number of relative social closeness observations, more specifically: contact set similarity (Formula 3.11), recency of connectivity (Formula 3.12), connectivity duration (Formula 3.13) and connectivity frequency (Formula 3.14). The social closeness observations are relative as they are normalised in proportion to the connectivity experience by the ego-node.

$$S^X_{C_i(X)} = \frac{|C(X) \cap C(C_i(X))|}{\sum_{i=1}^{N} (|C(X) \cap C(C_i(X))|)} \tag{3.11}$$

Formula 3.11 shows the contact set similarity observation is calculated as the cardinality of the intersect of the ego-node's contact set and a given encountered node's contact set, which is normalised as a fraction of the sum of all contact set similarity values.

$$R^X_{C_i(X)} = 1 - \frac{T_{now} - T_{last\_seen}(C_i(X))}{\sum_{i=1}^{N} (T_{now} - T_{last\_seen}(C_i(X)))} \tag{3.12}$$

Formula 3.12 shows the recency of connectivity observation is calculated as the difference in time between now and the time in which a given contact was last encountered, which is normalised as a fraction of all recency values.

$$D^X_{C_i(X)} = \frac{\sum_{e=1}^{N} d_e(C_i(X))}{\sum_{i=1}^{N} \sum_{e=1}^{N} d_e(C_i(X))} \tag{3.13}$$

Formula 3.13 shows the connectivity duration observation is calculated as the sum of all encounter duration with a given contact, which is normalised as a fraction of all connectivity duration values.

$$F^X_{C_i(X)} = \frac{f(C_i(X))}{\sum_{i=1}^{N} f(C_i(X))} \tag{3.14}$$

Formula 3.14 shows the connectivity frequency observation is calculated as the number of encounters with a given contact, which is normalised as a fraction of all connectivity frequency values.

	Success Ratio	Total Delivered	Delay	Transfer Cost
EN	0.842	209.772	2221.331	462.628
WEN	0.842	209.583	2377.969	463.343
SEN	0.844	210.302	2165.965	476.386

Table 3.1: A Result Summary Detailing The Average Performance of Our Contact Observation Aggregation Techniques Over The Infocom 2006 Dataset

Table 3.1 shows the average of our results, which were obtained from numerous simulations over the Infocom dataset (described in Section 4.1), using our social network traffic model (described in Section 4.2.2) to provide a realistic network load. Our experiments show that relative social weighting gives better performance than the simple weighting and the exponentially weighted moving average, as success ratio and the number of delivered messages are higher and the message delays are lower - this is at the cost of a slightly elevated transfer cost as messages are forwarded around better defined regions of congestion. Our results obtained from numerous simulations over the DieselNet and RollerNet network topologies are consistent with Table 3.1.

3.5.5 Replication Management

CafRep sends more than one copy of a message through the network, in order to maximise the delivery speed, success rate and robustness. This raises two key challenges: what should the copy limit (M) be and how should these M copies be distributed? We address these below.

Copy Limit

The initial number of copies (M) needs to be adaptive in order to carefully manage the tradeoff between the network size and traffic demands. If M was fixed it could be too large, causing the network to be prone to congesting, or too small, causing both increased delays and failure rate. CafRep adaptively adjusts M as follows. If all M copies of a message are sent M is additively increased. If messages need to be dropped from the sending buffer CafRep selects those with the highest number of sent copies and M is multiplicatively decreased.

Copy Distribution

By limiting the number of replicas raises the question of which nodes should receive the traffic this has been observed by [22] as an instance of an optimal stopping problem [75]. CafRep makes use of $\textit{CaféUtil}_D$, its principal utility function, to deduce the number of copies of a message a particular contact should receive, as shown in Formula 3.15. If two contacts are equally suited to forward a message, they each receive half of the available copies. Similarly, a contact receives more or fewer message if 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.

$$\textit{RepRate} = \lfloor M \cdot \textit{CaféUtil}_D(X, C_i(X)) + 0.5 \rfloor \tag{3.15}$$

The more copies of a message a node has to disseminate the longer a node is likely to have to retain a message. The availability of the contacts ego-network also affects the duration of time a contact is likely to have to store a message, because if a contact encounters nodes with little or no available storage the contact will not be able to offload. By using the $\textit{CaféUtil}_D$ function we ensure that nodes with higher forwarding potential, lower levels of congestion and lower ego-congestion levels receive a greater volume of traffic than nodes that do not offer forwarding opportunities or available resources.

3.6 The Algorithm

Various events trigger CafRep to respond, such as Encountering a new contact (Algorithm 1), and receiving a message summary (Algorithm 2), other than these triggers, provided a node has messages to send and contacts to send them to, CafRep periodically updates its forwarding list (Algorithm 3).

Algorithm 1 Encounter Node

```
// Update encounter history
if peer ∈ EncounterHistory then
    EncounterHistory[peer].update(time)
else
    EncounterHistory.add(peer)
end if
// Send summary message to peer
newMsg ← new Message()
newMsg.addProperty(EncounterHistory)
newMsg.addProperty(CongestionSummary)
newMsg.addProperty(EgoCongestionSummary)
sendMsg(peer, newMsg)
```

Algorithm 2 Receive Summary Message

```
// Get properties from transferred message
msg ← messageTransferred()
EH ← msg.getProperty(EncounterHistory)
CS ← msg.getProperty(CongestionSummary)
ECS ← msg.getProperty(EgoCongestionSummary)
// Update social and congestion observation histories
EncounterHistory[peer].update(EH, CS, ECS)
EgoNetworkUpdate(CS)
SocialUpdate()
```

Algorithm 1 shows that on receiving a beacon message announcing a contacts existence CafRep can update the social metrics about the contact, such as the time it was last observed, the observation frequency and observation duration, the beaconed message is then deleted. CafRep then constructs and sends a summary message to the new contact, this includes social information about observed nodes, congestion observations and ego-network congestion observations.

Algorithm 2 shows that when a CafRep node receives a message summary it updates its encounter history, congestion summary and ego-network congestion summary for the contact. Because the node has made changes to its observed network state, it must update its ego-network congestion observation and its social network information such as betweenness centrality.

Algorithm 3 shows the routine a CafRep node periodically iterates, provided it has messages to send and contacts to send them to. The first objective is to transfer any messages a node may have in its buffer destined for nodes in the current set of contacts, following this CafRep seeks to find the best available contact for each message in the buffer to be forwarded to. When the best next-hop node has been identified, its ID, utility and the message ID are added to the list of messages to forward. The sendMsgs function iterates through these forwarding demands and calculates the number of copies of each message to forward based on the nodes utility as shown in Formula 3.15.

Algorithm 3 Message Transfer Update

```
// Forward message to destinations
for all {m ∈ B : m_dest ∈ C} do
   sendMsg(m_dest, m)
end for
// update forwarding list
forwardingList = []
for all m ∈ B do
   betsUtil ← 0.0
   bestNode ← x
   for all c ∈ C do
      cUtil ← CafRepUtil_D(x, c)
      if cUtil ≥ bestUtil then
         bestUtil ← cUtil
         bestNode ← c
      end if
   end for
   if bestNode != x then
      fowardingList.append((bestNode, bestUtil, m))
   end if
end for
sendMsgs(fowardingList)
```

3.6.1 Transfer Decision Boundary Outcomes

In this section we identify the possible outcomes of the message forwarding decision process, given the combination of signals monitored and received by our system at each of the possible extremes.

X \ Y		High CL		Low CL	
	X \ Y	High Fwd	Low Fwd	High Fwd	Low Fwd
High CL	High Fwd	0 0	1 0	0 1	1 1
	Low Fwd	0 1	0 0	0 1	0 1
Low CL	High Fwd	1 0	1 0	1 1	1 0
	Low Fwd	1 1	1 0	0 1	1 1

Table 3.2: A Decision Matrix Illustrating Transfer Decision Outcomes

Table 3.2 illustrates 16 different combinations of Congestion Level (CL) and Forwarding Ability (Fwd) between nodes X and Y. In Table 3.2 if a cell contains the value 1 it indicates that the node would transfer a message, and if a cell contains the value 0 it denotes that the node would not forward a message.

At times when both nodes are suffering high levels of congestion, a packet will only be forwarded if the potential next-hop increases delivery probability and the level of congestion is proportionate in comparison to the gain in delivery probability. This means that a contact that is within range of the destination may receive a message from a node who is not connected to the destination, even when the contact has less available resources than the sender.

When one of the nodes is congested and the other is not, messages will always be forwarded away from the congested region, and messages will only be forwarded towards the congested node if doing so relatively increases delivery probability. A result of allowing traffic to move away from the overloaded regions (or nodes) is that we make it possible for messages to be delivered to nodes that would otherwise be unreachable due to the level of competing traffic demands.

If both nodes are not suffering from congestion messages will be forwarded, provided the next hop does not decrease delivery probability. The reason for deciding to forward to a node regardless of the fact it offers no major improvement, either due to having equally high of low forwarding ability, is because of the low probability of the encountered node dropping the message, allowing us to propagate the message out into the network without increasing the probability of it being dropped.

This table offers a view of the outcome of our forwarding algorithm when the observed parameters are strikingly different, but does not address when the values are indifferent, which is a commonly observed state when bootstrapping. In the event of indifference our algorithm takes advantage of the forwarding opportunity in order to propagate the message as soon as possible towards the destination.

Chapter 4

Evaluation Methodology

This Chapter is concerned with the procedure used in order to achieve a rigorous analysis of CafRep. Figure 4.1 illustrates the process employed in order to evaluate CafRep against both the state-of-the-art and benchmark algorithms. We evaluate multiple runs of each of the forwarding protocols, across three connectivity datasets and two application traffic models, using the Opportunistic Network Environment (ONE) simulator [43].

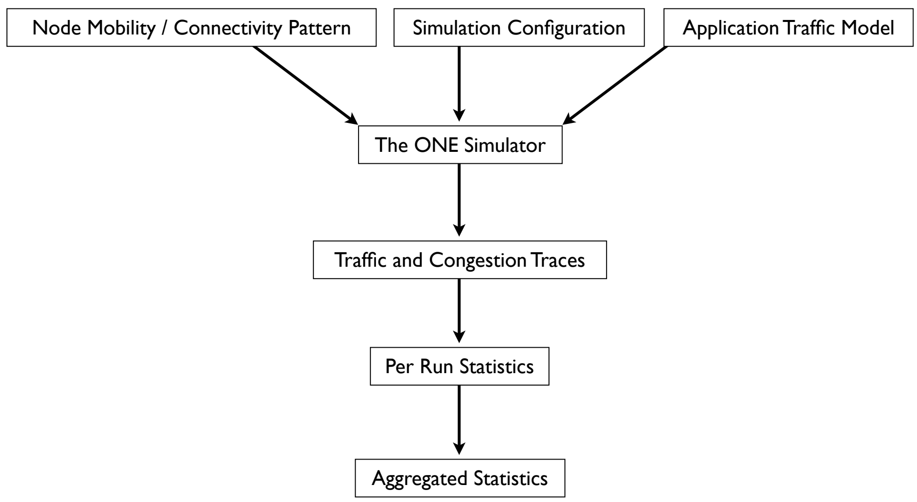


Figure 4.1: Emulation Process

In DTNs, the mobility of the nodes has a major impact on the performance of communication protocols. In order to represent sensible transmission range and realistic movement patterns of the mobile users and vehicles, we use three different connectivity traces from CRAWDAD [45]. Our selected datasets exhibit vastly different connectivity patterns derived from the different mobility patterns of each real world scenario.

In order to evaluate our proposal, we have integrated two different traffic models, a social networking application and an interest driven application. Our two applications offer contrasting traffic patterns, the social networking application has heterogeneous users generating content at a variety of different rates and sizes, while the interest driven application users send messages of a fixed size at a constant data rate.

4.1 Connectivity Trace Datasets

Our evaluation employs three real world connectivity traces from CRAWDAD: Infocom [73], RollerNet [49] and Dieselnet [8] that have different mobility and connectivity patterns. We aim to evaluate our protocol in different network contexts and show that CafRep is successful independently of the inter contact times of the underlying topology.

Infocom trace [73] 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 [34].

RollerNet [49] dataset represents a class of DTNs that follows a pipelined shape, it has extreme dynamics in the mobility pattern of a large number of nodes. During three hours, roller-bladers travel about 20 miles, covering a large portion of Paris. Contact loggers were deployed on 62 volunteers of three different types: friends of the authors, members of rollerblading associations, and staff operators. Both loggers and other participants had Bluetooth on their cell phones.

DieselNet trace [8] consists of 20 days of traces of 40 UMass transit buses covering approximately 150 square miles. This trace contains connection events between buses as well as between buses and Access Points. DieselNet buses were subject to the schedule of the UMass campus. This trace has been shown to exhibit long periods of disconnectivity, short periods of connectivity and islands of connectivity that are rarely populated by more than two nodes.

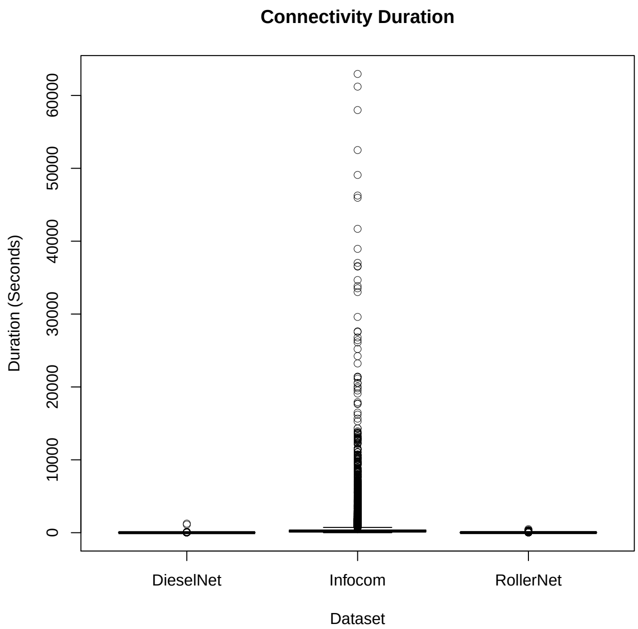


Figure 4.2: Contact Duration

Figure 4.2 shows that both DieselNet and RollerNet datasets exhibit predominantly short contact durations (a mean of 11.42 and 21.75, a median of 5 and 16 and a maximum of 1254 and 488 seconds respectively) while Infocom displays substantially longer contact durations (a mean of 373, a median of 220 and a maximum value of 62953 seconds).

Figure 4.3 illustrates three different trends regarding node isolation periods.

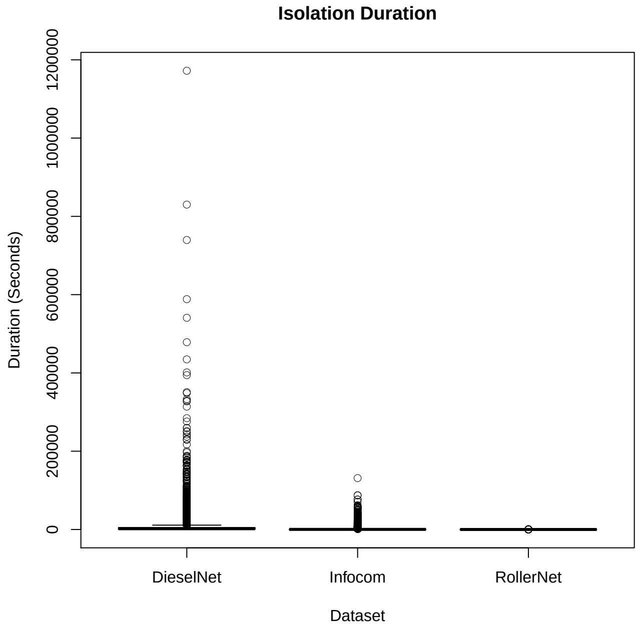


Figure 4.3: Isolation Duration

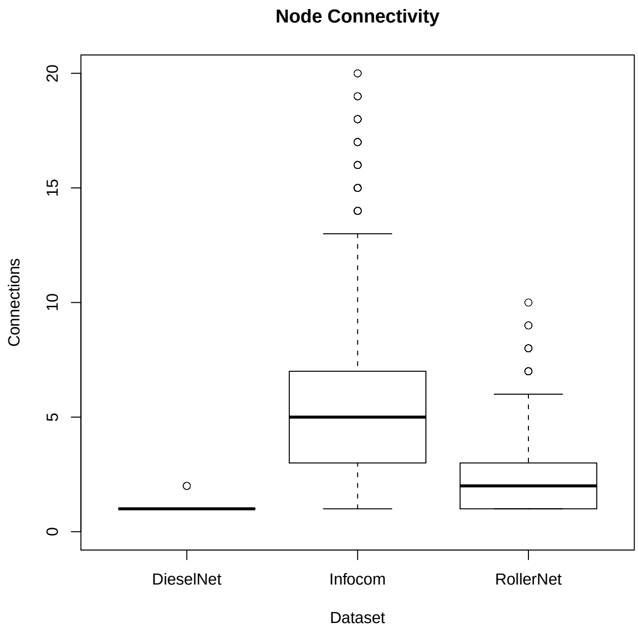


Figure 4.4: Node Conectivity

DieselNet suffers the longest periods of isolation (a mean of 11583, a median of 1991 and a maximum value of 1172129 seconds), Infocom experiences substantially lower periods of isolation (a mean of 1475, a median of 243 and a maximum value of 131280 seconds) and RollerNet has the shortest periods of isolation (a mean of 26.34, a median of 16 and a maximum value of 948 seconds).

Figure 4.4 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. DieselNet has the lowest observed node connectivity (a mean of 1, a median of 1 and a maximum value of 2 connections), Infocom has the highest observed node connectivity (a mean of 5.42, a median of 5 and a maximum value of 21 connections) and RollerNet experiences approximately half the node connectivity observed from the Infocom dataset (a mean of 2.31, a median of 2 and a maximum value of 10 connections).

4.2 Application Models

This Section presents the two application traffic models used to evaluate CafRep. The first, a more simplistic model, is an application model that operates by a set of publisher nodes in the network generating content for a set of subscriber nodes, which is detailed in greater depth in Section 4.2.1. The second application model is driven by real world data that we have collected from a number of Facebook users who had agreed to participate in our experiment. Our real world application model and its utilisation within the simulation environment is described in detail in Section 4.2.2.

4.2.1 Distributed File Casting Application

In order to evaluate CafRep we have designed and built a fully distributed, interest driven overlay file casting application. Each node publishing content (publishers) forward its content to nodes that are interested in it. The publishers also forward their content to nodes that frequently encounter other nodes that are interested in the content, provided they have sufficient resource availability.

If the publishing node does not meet any node interested in its topic for a while, it may cache its content in the highly connected and available nodes. In this way, we minimise the usage of non-interested intermediaries and thus provide more efficient forwarding. Both caching and forwarding policies are driven by the topic interest, resource availability and social closeness of a contact, along with the number of encounters a contact has had with nodes that are following the topic. Our content is organised as in previous file casting work [59] where each publisher generates content for a given topic and each topic has chunks that can be exchanged when two nodes meet. Each chunk has a unique ID and each topic consists of a number of chunks. We randomly assign topic interests and choose a varying number of publishers and subscribers in order to induce congestion at different rates and locations. Related work on publish-subscribe data dissemination in DTNs in [88] explicitly relies on detecting communities and does not consider congestion control. [37] 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. [62] 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.

4.2.2 Real World Social Network Application

We also consider the impact that real world social network traffic requirements have on the behaviour of Café, CafRep and other benchmark and state-of-the-art DTN forwarding protocols. We designed the Facebook Application Traffic experiment in order to allow us to model the traffic typical of communication in social networking applications. More specifically, this refers to extracting a friendship graph and statistical data regarding the size and frequency of posts from Facebook and using this information to drive a publish and subscribe application on the top of the forwarding protocols. We have collected the usage patterns of 95 Facebook users, by creating a Facebook application and disseminating it through friends (by posting a link) we collected a connected subset of Facebook users. The information collected from a participant included: A list of friends associated to the participants, the 20 most recent wall post messages for each candidate, and the users list of selected interests.

Figure 4.5 illustrates the relationship between each of the participants. We can see that Figure 4.5 is Spoke-hub distribution, with the balance of friends per node being skewed towards those participants who managed to encourage the most additional candidates to participate. In the interest of conducting realistic experiments we adjusted the way in which we selected nodes for the friendship graph. Our initial method simply acknowledged a contacts friends if they were also participants (Figure 4.5) We decided to expand this friendship graph to include non-participant IDs, provided they existed in more than one participants friend list (Figure 4.6).

Figure 4.6 shows the friendship graph of collectively associated friends, which consists of a total of 1950 nodes. We observe that Figure 4.6 is more of a mesh network, as it exhibits a smaller variance in the number of observed contacts per node, than the previous participant only friendship graph (Figure 4.5). Figure 4.7 shows that nodes in the collectively associated friendship graph are connected to 3 friends on average and at most 2% of all other nodes.

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 4.8 shows the size of messages generated for each user. We observe that message sizes are not uniform, as they have a Mean size of 1MB, a Median size of 82KB and a maximum size of 268MB.

Figure 4.9 (a) shows 3 types of message, text, picture and link. Text messages are the most commonly posted (78%), followed by posts containing pictures (19%), with only 3% of posts encompassing links.

Figure 4.9 (b) shows that in contrast to message frequency, where text messages are most common, they only formed a small proportion (153KB in total

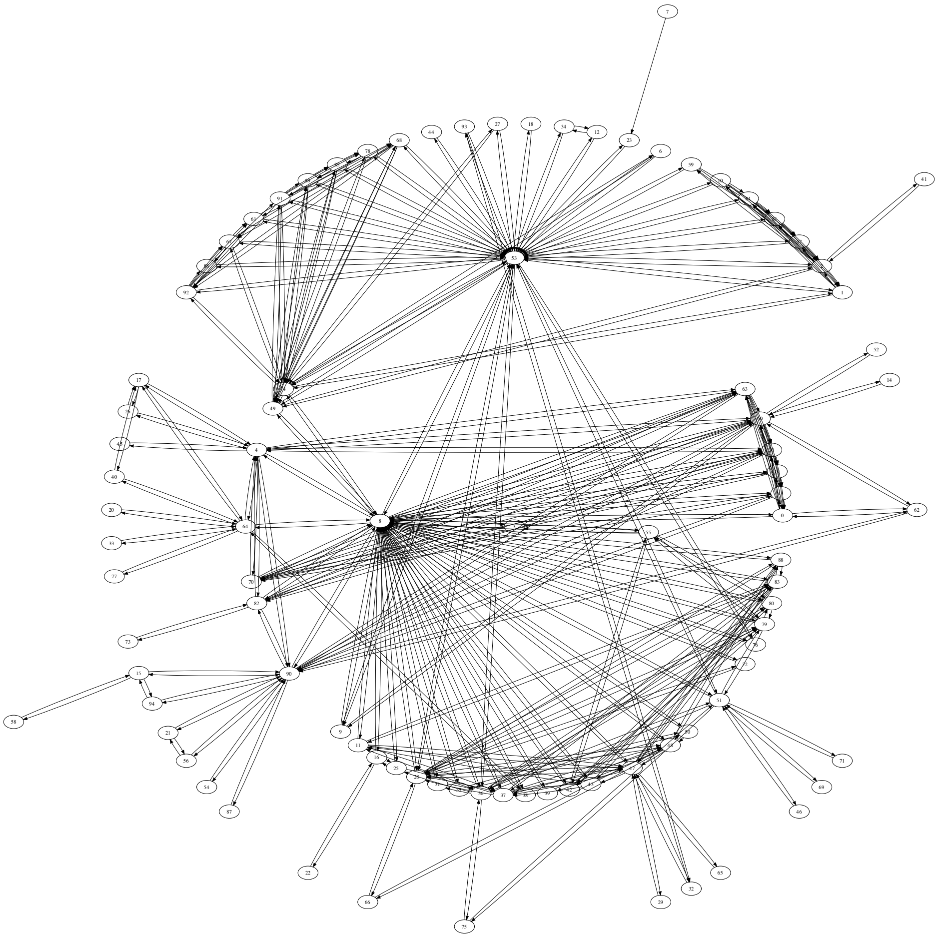


Figure 4.5: Relationships between participant

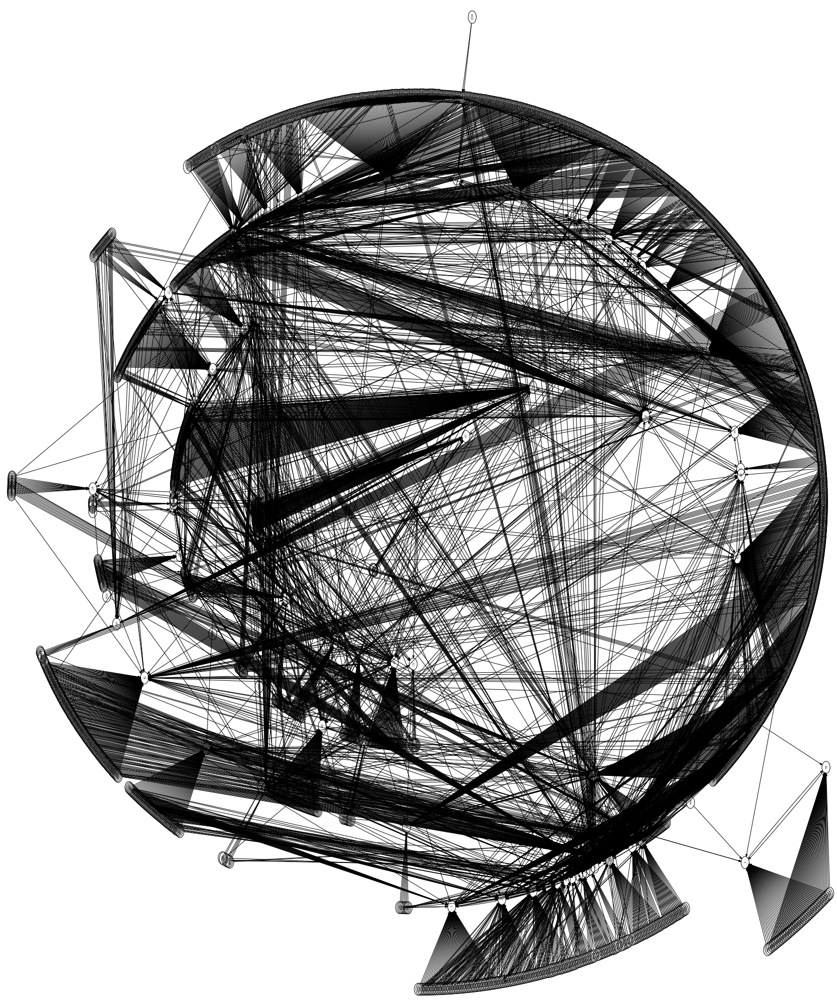


Figure 4.6: Relationships between participants and collective friends

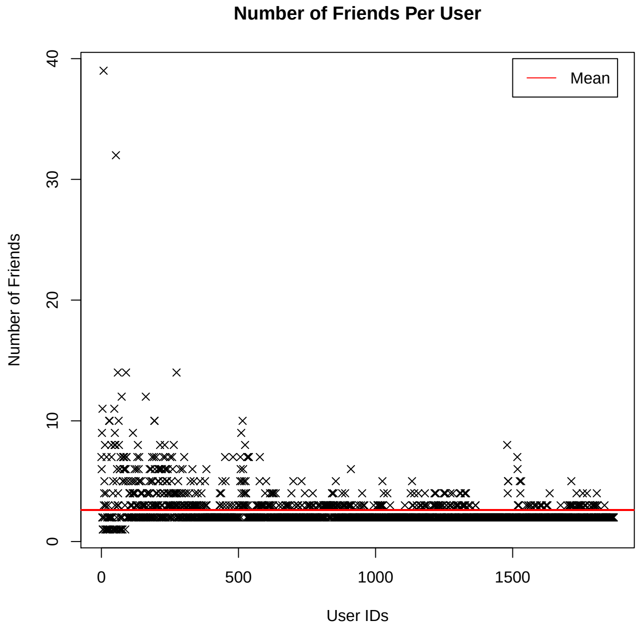


Figure 4.7: Number of Friends per Facebook User

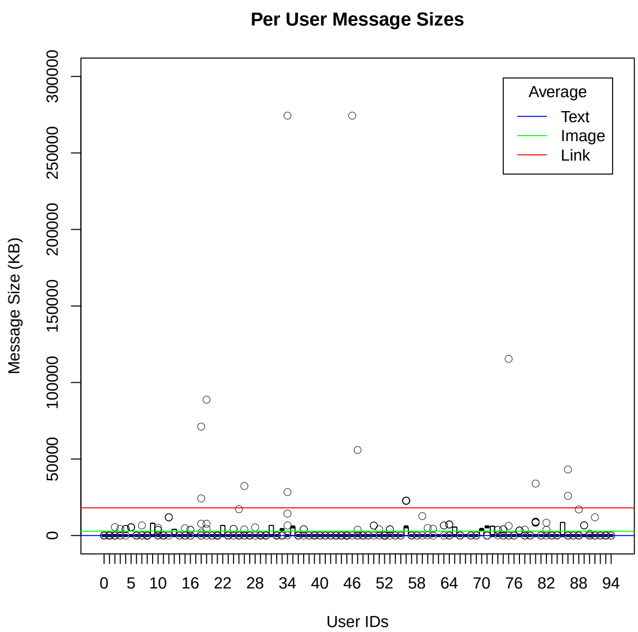
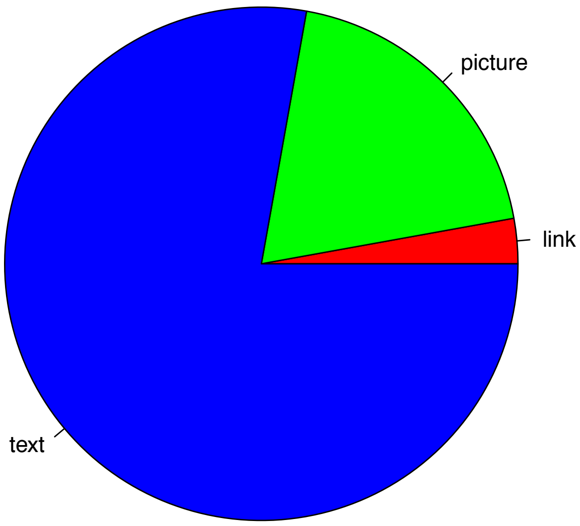
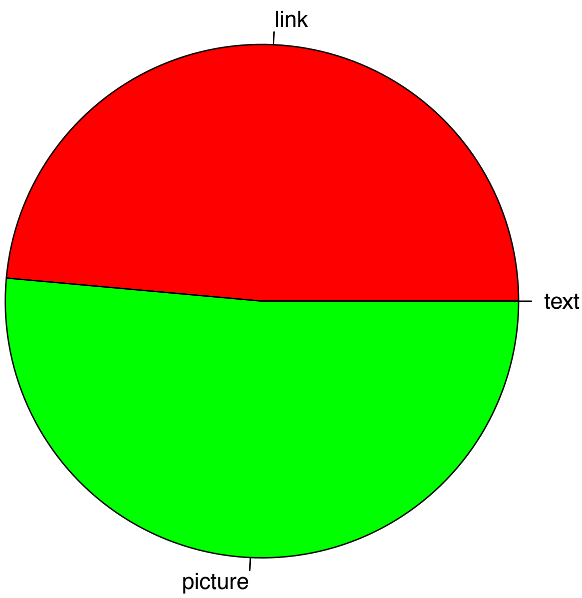


Figure 4.8: Facebook Traffic Message Sizes



(a) Message Frequency Proportions



(b) Message Size Proportions

Figure 4.9: Message Size and Frequency Proportions

and 82B on average) of the of the total observed data (2.3GB).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).

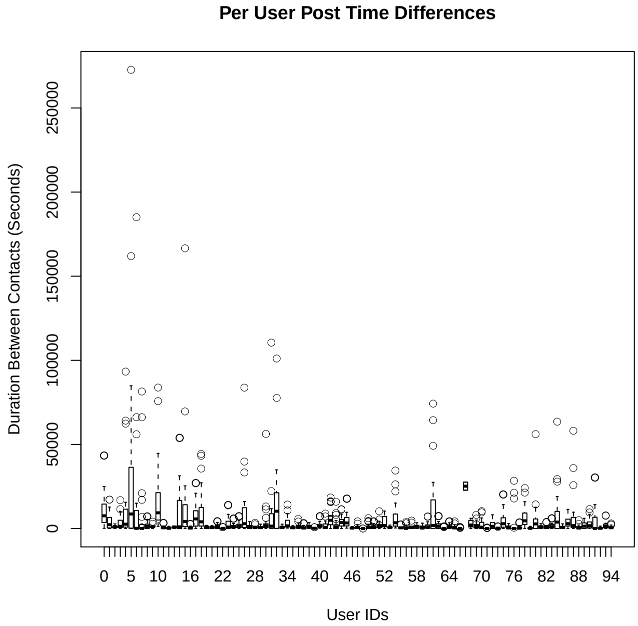


Figure 4.10: Facebook Traffic Post Frequencies

Figure 4.10 illustrates the duration of time between posts for each user. The most prolific participant posted every 3 minutes on average, whilst the most inactive profile posted content once every 25 days on average. We observe that in the extreme cases users posted 20 messages in as short a period as 83 minutes and as long a duration as 602 days.

Our implementation comprises of 95 actor profiles, each based on a different participant. A profile is made up of 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 4.1 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.

$$P(x) = \lfloor 0.5 + \lambda(x) + (\sigma(x) \cdot W) \rfloor \tag{4.1}$$

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.

4.3 Emulation Configuration

This Section details how we configure the traffic models detailed in Section 4.2 in order to produce a range of different traffic levels, such that we gradually induce a state of congestion, which we can then evaluate.

4.3.1 Distributed File Casting Application

We simulate congestion by increasing the number of senders and topic popularity with our interest driven application. Increasing the number of publishers and subscribers allows us to evaluate very different congestion rates originating in different network locations at different times, similarly to the diversity of usage typical of mobile devices [23]. [84] show that in social online system analysis there is typically a smaller number of publishers with large audiences, while in the photo uploading application scenario it is more usual to have a larger number of publishers. We report findings from our interest driven file casting application experiments across each of the three CRAWDAD datasets. We have run eight increments of congestion levels induced by increasing number of publishers first and then subscribers ranging from 1/9 to 8/9 of the total number of nodes in that connectivity dataset, we report on experiments with increasing number of publishers, but note that the results for increasing number of subscribers are similar to the ones presented here. All simulations are repeated 10 times, each with different random subscribers and publishers.

4.3.2 Facebook Application

We simulate 3 different experiments with the Facebook application model: low traffic profiles, randomly selected traffic profiles and high traffic profiles. The low and high traffic profiles use a Pareto distribution function to select a traffic profile from an array of traffic profiles that are ordered on a statistic of their expected traffic generation, as detailed in Formula 4.2.

$$T(P) = \frac{MSI(P)}{P_{freq}^{avg} - P_{freq}^{\sigma}} \tag{4.2}$$

Formula 4.2 shows that traffic profiles are sorted by calculating a message size indicator and dividing this by a message frequency indicator. The message frequency indicator is a statistic based on the distance between each message generation, which is calculated as the average message frequency minus the standard deviation of message frequencies.

$$MSI(P) = \sum_{t \in \{text, picture, link\}} (P_s^{avg}(t) + P_s^{\sigma}(t)) * P_r(t) \tag{4.3}$$

Formula 4.3 shows that the message size indicator is calculated as the sum of the average message size ($P_s^{avg}(t)$), which is added to the message size standard deviation ($P_s^{\sigma}(t)$), and then multiplied by the ratio at which each message type was observed ($P_r(t)$).

We simulate congestion by decreasing buffer size, each experiment has been simulated with buffer sizes ranging from 10MB to 100MB, increasing at 10MB intervals. Each buffer size is evaluated over 10 runs, each with different random seed values. We also simulate each experiment across each of the three datasets outlined in Section 4.1.

Chapter 5

Distributed File Casting Application Evaluation

This Chapter is concerned with the rigorous evaluation of our work, such that performance can be evaluated independently of inter-contact times and contact durations that result from the underlying network topology. We assess performance across three vastly different connectivity datasets: DieselNet, Infocom and RollerNet (as outlined in Section 4.1).

This Chapter presents the extensive evaluations of CafRep under various levels of traffic demand, which is generated by our distributed file casting application (as described in Section 4.2.1), across a wide range of metrics and different network scenarios in the ONE [43] simulator.

This Chapter evaluates success ratio, delivered packets, transfer cost and message delay in order to assess forwarding performance; and congesting rate and buffer availability to assess network resource utilisation.

We compare the performance of CafRep against Probabilistic routing (Prophet) [54], Spray and Focus (SF) [78], Encounter based routing (EBR) [60] and SimBetTS [18] with Retiring Replicas (RR) [81] replication management. We evaluate against Prophet as it 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. We implement RR on top of SimBetTS because SimBetTS is the social forwarding algorithm used in order to evaluate CafRep, as such we can gauge a better comparison between the two congestion control algorithms. The range is not plotted in these results as the results observed accross our simulation runs were similar.

5.1 Forwarding Performance Evaluation

This Section is concerned with the performance metrics commonly evaluated within the DTN forwarding literature. We evaluate Success Ratio, Delivered Messages, Dropped Messages, Transfer Cost and Message Delay in order to show that forwarding performance is not lost through the addition of congestion control.

5.1.1 Success Ratio

The success ratio observations across all three environments shows that CafRep outperforms the other five major protocols and has the best performance ranging between 70% and 90% while the competing algorithms achieve between 10% and 70%.

Figure 5.1 shows that in DieselNet CafRep achieves the best performance with close to 70% of success ratio for all congestion levels, followed by Café at 65% and RR and EBR at 60%. Prophet has low success ratio from 40% to 10% due to lack of interest points that it depends on. Spray and Focus has approximately 40% success ratio.

Figure 5.2 shows that in Infocom CafRep has the highest success ratio ranging between 80% and 90% for all congestion levels. CafRep success ratio is 25% higher than Café and 30%-40% higher than RRs and EBRs respectively. Spray and Focus has over 40% lower success ratio than CafRep. Prophet has the lowest success ratio ranging from 35% to only 10% which is more than three times lower than CafRep for all congestion levels.

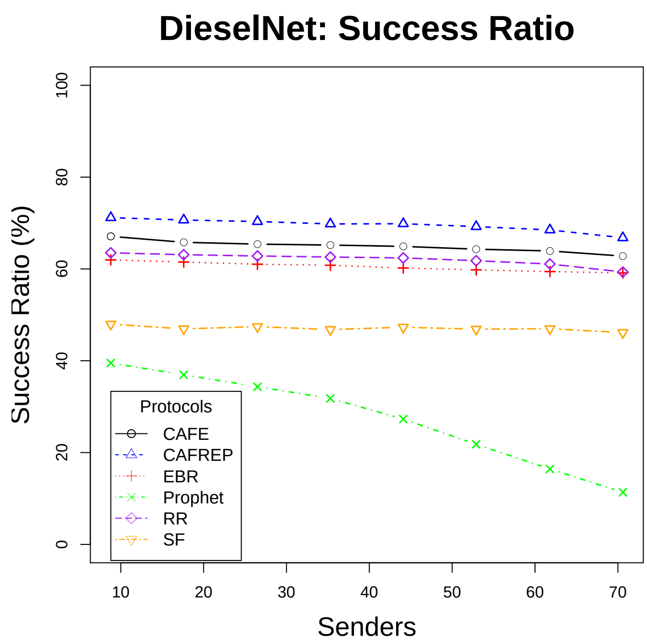


Figure 5.1: Success Ratio for DieselNet

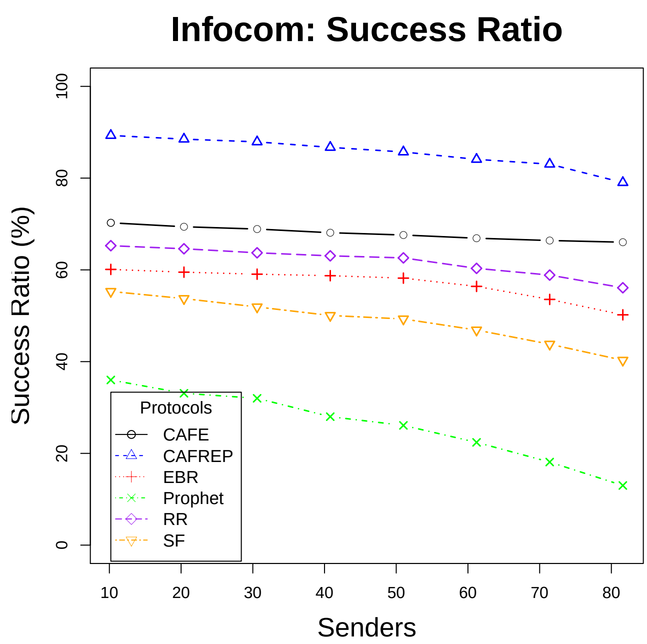


Figure 5.2: Success Ratio for Infocom 2006

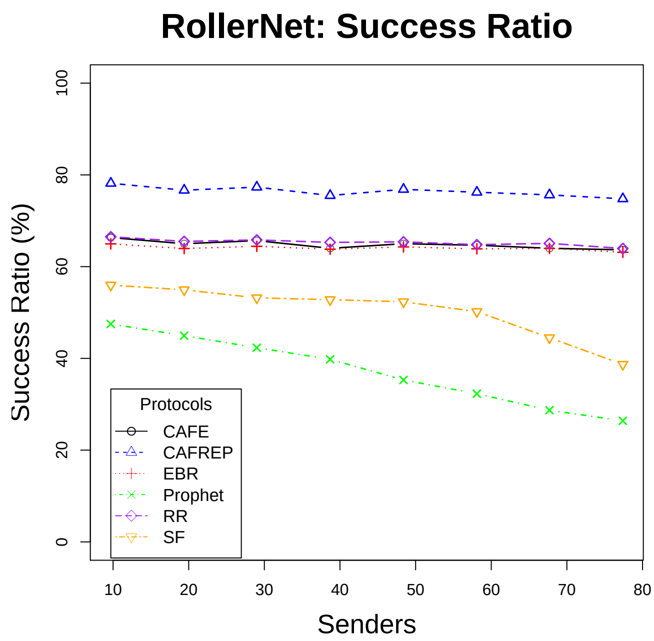


Figure 5.3: Success Ratio for RollerNet

Figure 5.3 shows that in RollerNet CafRep has the highest success ratio that is close to 80% for all congestion levels. It is interesting to see that RR, EBR and Café have similar success ratios in the area of 65%, which is around 23% lower than CafRep's success ratio. SF and Prophet have much lower success ratio compared to all adaptive protocols, ranging from 58% to 40% and 48% to 25% respectively, which is close to two times less than what CafRep achieves.

We observe slightly lower success ratio levels for CafRep in the simulations over the DieselNet connectivity traces, this is due to the connectivity pattern of nodes being less suited to the social forwarding algorithm used in $CaféUtil_D$, but despite this slight drop in performance CafRep still outperforms all other algorithms. CafRep performs best over the Infocom 2006 dataset, which has the densest islands of nodes and the longest periods of connectivity between nodes out of the three datasets used. CafRep performance over the infocom 2006 dataset illustrates how CafRep can effectively manipulate traffic demands across heterogeneous encounters, where other algorithms such as Prophet forward to a greedily selected minority subset of nodes, which results in poor success ratios that deteriorate rapidly as traffic levels increase.

5.1.2 Delivered Messages

Across all three datasets we observe that CafRep delivers more packets than any of the other algorithms, 14.6% more than Café and between 18.27% and 60.83% more than the other algorithms.

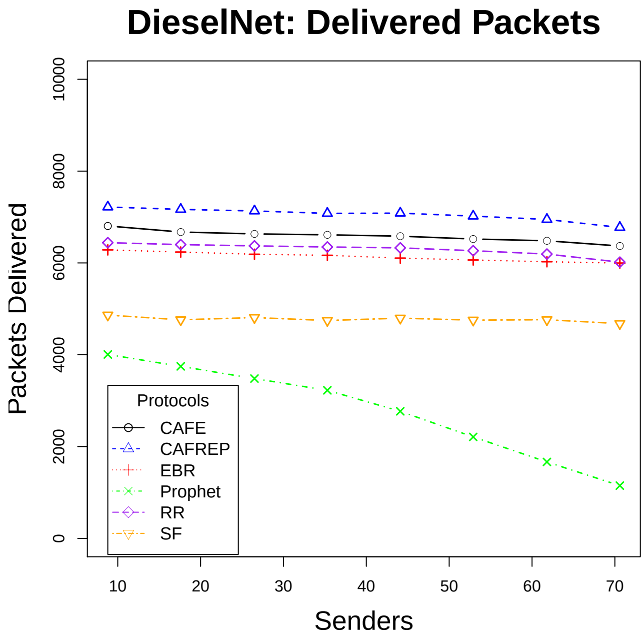


Figure 5.4: Delivered Messages for DieselNet

Figure 5.4 shows that in DieselNet CafRep delivers on average 7% more packets than Café, 12% more than RR, 15% more than EBR, 48% more than SF and 154% more than Prophet. Figure 5.5 shows that in Infocom CafRep delivers on average 26% more packets than Café, 38% more than RR, 50% more than EBR, 75% more than SF and 228% more than Prophet. Figure 5.6 shows that in RollerNet CafRep delivers on average 18% more packets than Café, 17% more than RR, 19% more than EBR, 52% more than SF and 106% more than Prophet.

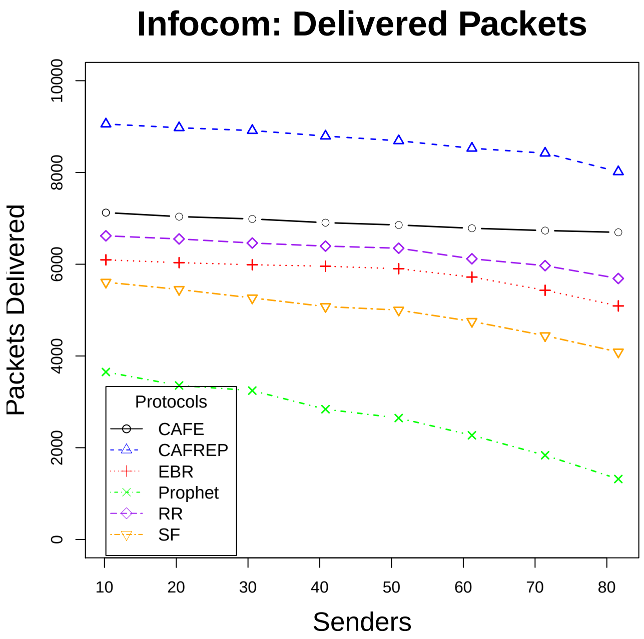


Figure 5.5: Delivered Messages for Infocom 2006

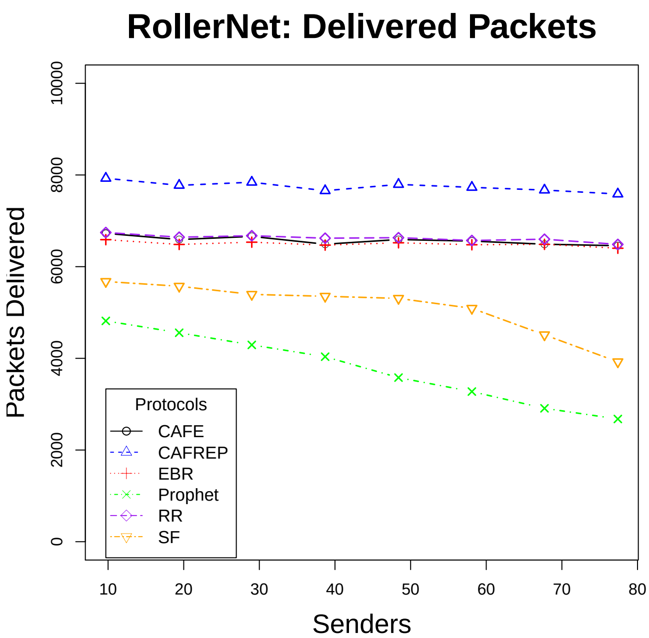


Figure 5.6: Delivered Messages for RollerNet

5.1.3 Dropped Messages

CafRep drops from 33.03% to 67.26% less on average than the other algorithms, across the three datasets, which equates to a difference of between 1689.17 and 4759.67 packets over the duration of the experiments.

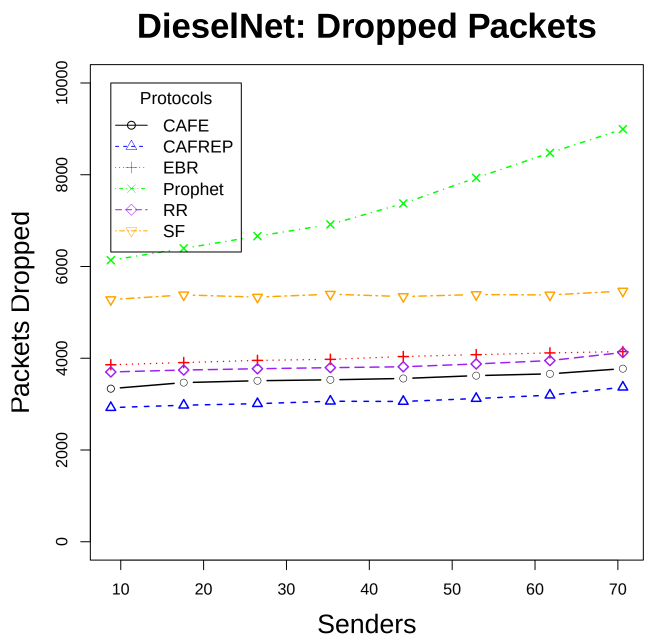


Figure 5.7: Dropped Messages for DieselNet

Figure 5.7 shows that in DieselNet CafRep drops on average 15% less than Café, 25% less than RR, 30% less than EBR, 74% less than SF and 138% less than Prophet. Figure 5.8 shows that in Infocom CafRep drops on average 122% less than Café, 164% less than RR, 197% less than EBR, 253% less than SF and 411% less than Prophet. Figure **??** shows that in RollerNet CafRep drops on average 49% less than Café, 47% less than RR, 52% less than EBR, 110% less than SF and 166% less than Prophet.

5.1.4 Transfer Cost

CafRep forwards between 8% and 35% more packets than the other algorithms, but due to the high success ratio this means the average cost of delivering a

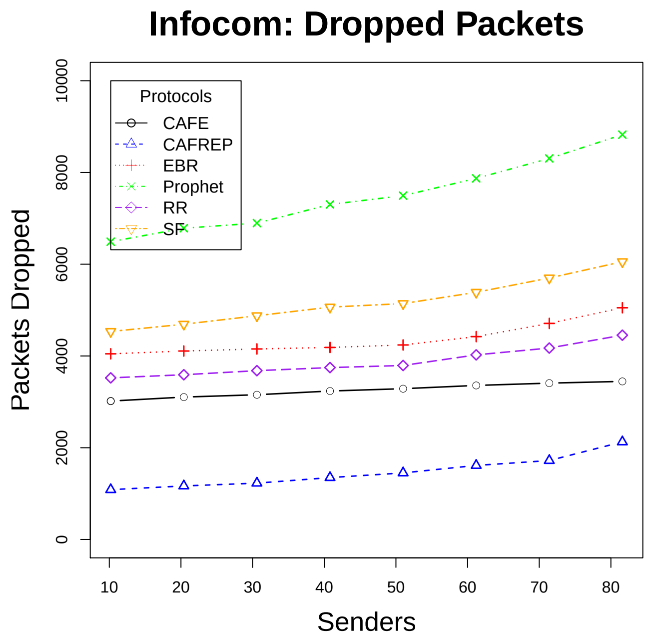


Figure 5.8: Dropped Messages for Infocom 2006

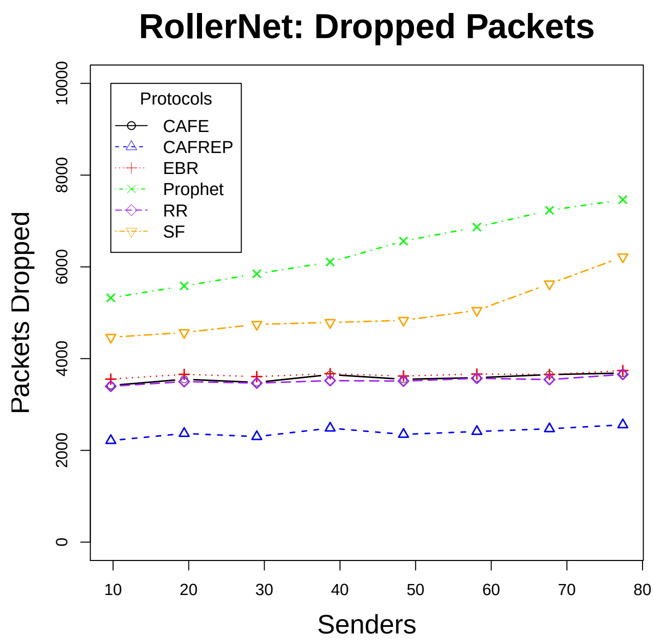


Figure 5.9: Dropped Messages for RollerNet

message is 3.76 hops, followed by Café taking on average 4.02 hops. RR, EBR and SF forward between 10% and 21% more taking 4.10, 4.18 and 4.66 hops respectively on average to deliver a packet. Prophet forwards substantially less packets but takes 6.17 hops to deliver a message.

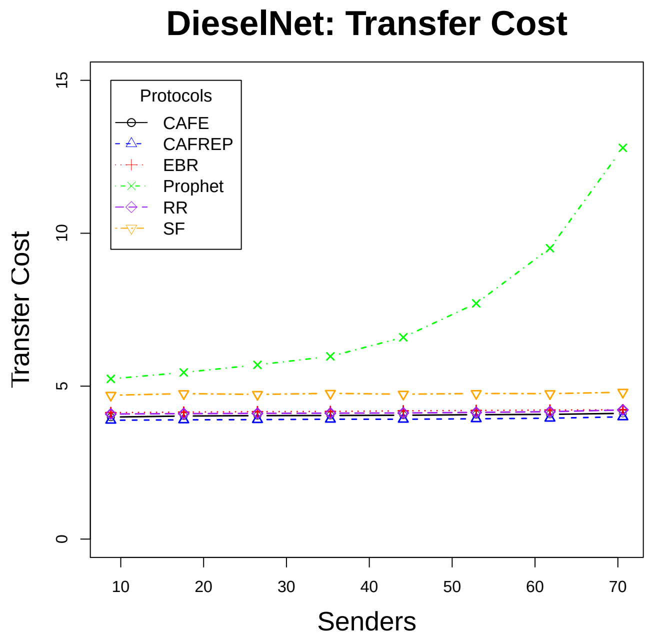


Figure 5.10: Transfer Cost for DieselNet

Figure 5.10 shows that in DieselNet CafRep delivers a packet in 0.12 fewer hops than Café, 0.21 fewer than RR, 0.26 fewer than EBR, 0.82 fewer than SF and 2.65 fewer than Prophet. This equates to CafRep forwarding on average 4% more than Café, 6% more than RR, 7% more than EBR, 18% more than SF and 34% more than Prophet.

Figure 5.11 shows that in Infocom CafRep delivers a packet in 0.36 fewer hops than Café, 0.54 fewer than RR, 0.70 fewer than EBR, 1.05 fewer than SF and 3.20 fewer than Prophet. This equates to CafRep forwarding on average 13% more than Café, 17% more than RR, 20% more than EBR, 26% more than SF and 42% more than Prophet.

Figure 5.12 shows that in RollerNet CafRep delivers a packet in 0.28 fewer

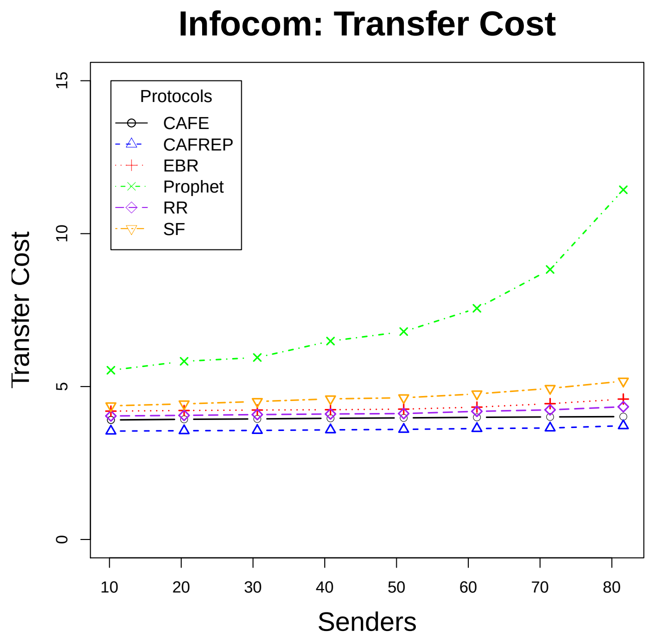


Figure 5.11: Transfer Cost for Infocom 2006

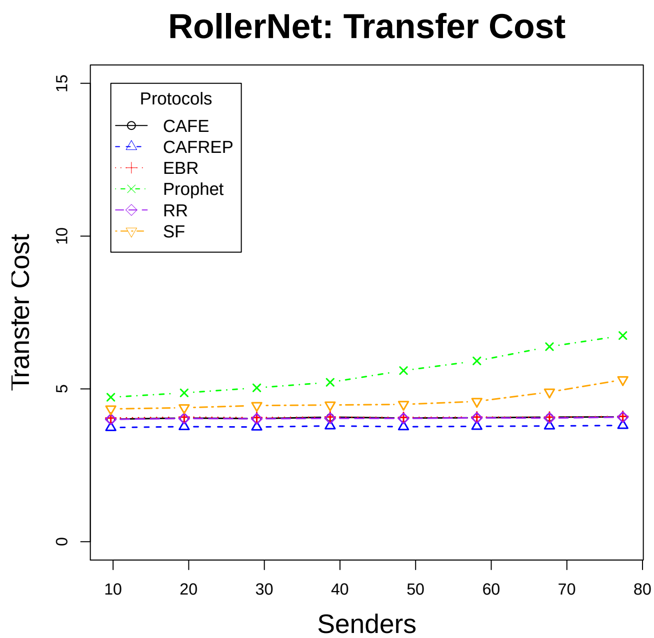


Figure 5.12: Transfer Cost for RollerNet

hops than Café, 0.27 fewer than RR, 0.30 fewer than EBR, 0.81 fewer than SF and 1.66 fewer than Prophet. This equates to CafRep forwarding on average 9% more than Café, 8% more than RR, 9% more than EBR, 20% more than SF and 30% more than Prophet.

Despite the fact Café and CafRep avoid and reduce congestion by forwarding along suboptimal paths at times of increased traffic demands, the cost associated with delivering a message is lower than all other algorithms, which is compelling as it identifies that although Café and CafRep forward messages via a greater number of intermediaries than the greedily selected optimal routes, the overall cost of delivery is improved.

5.1.5 Message Delay

The delay observed across all the three traces show that CafRep achieves lower delays compared to the other five protocols. On average for highly diverse network connectivity, CafRep takes 53.85 seconds to deliver a packet and Café, RR, EBR, SF and Prophet take 63, 81, 90, 102 and 134 seconds respectively.

Figure 5.13 shows that in DieselNet CafRep achieves the lowest delays in the range of 55 to 80 minutes for increasing congestion levels, followed by Café that has delays ranging from 80 minutes to 110 minutes. RR and EBR have higher delays from both CafRep and Café ranging from 90-160minutes and 100-180 minutes respectively. This means that EBR and RR are taking approximately two or more times longer to deliver packets than CafRep for all congestion levels. Spray and Focus and Prophet have at least three time larger delays than CafRep.

Figure 5.14 shows that in Infocom CafRep has more than two times lower delays for low congestion rates than all other protocols. For medium to high congestion levels, CafRep increases delays so that it has 10% higher delays than Café. Across all congestion levels CafRep has up to two times lower delays than RR and Spray and Focus, and up to three times lower delays than EBR for all congestion levels. Prophet has highest delays that are persistently twice as high in comparison with CafRep.

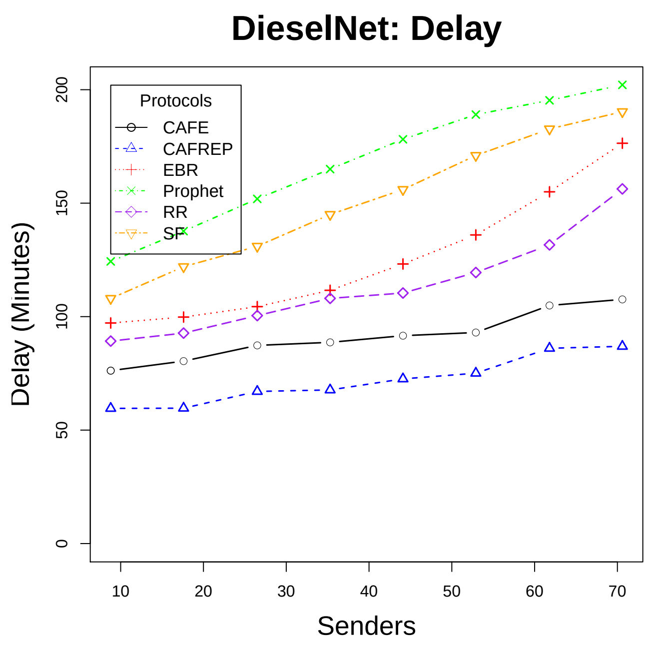


Figure 5.13: Message Delay for DieselNet

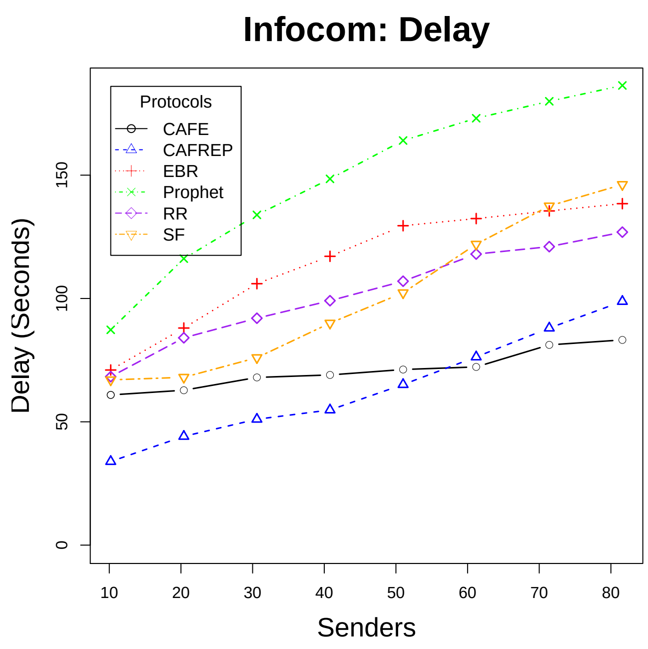


Figure 5.14: Message Delay for Infocom 2006

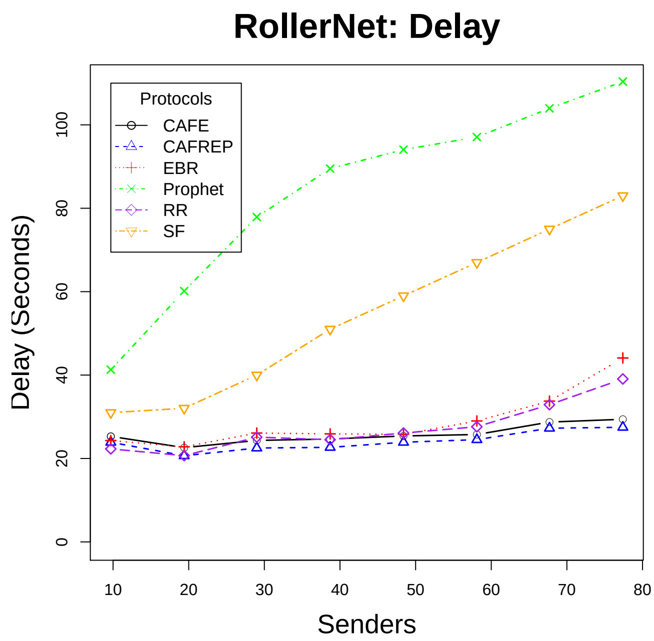


Figure 5.15: Message Delay for RollerNet

Figure 5.15 shows that in RollerNet all adaptive protocols manage to keep 20 seconds delay for low and medium congestion levels. For high congestion levels CafRep and Café have the lowest delay (around 28 sec) while RR and EBR increase their delays to up to two times higher than CafRep. Delays for Spray and Focus and Prophet are considerably higher than any of the adaptive protocols ranging from 30sec to 80sec (50% to 400% slower than CafRep) and 40 to 110sec (200% to 500% slower than CafRep) respectively.

CafRep out performs all other algorithms, and for the majority of our experiments CafRep outperforms Café. Café only outperforms CafRep in the most congested traffic scenarios over the Infocom 2006 dataset, this is due to the number of replicas that CafRep disseminates (M) being fixed in these experiments and the preset value of M being too high for the density and traffic demands experienced. We evaluate our adaptive replication limit, as described in Section 3.5.5, in Chapter 6.

5.2 Network Resource Utilisation Evaluation

This Section is concerned with evaluation metrics that identify the effectiveness of the congestion control mechanisms of CafRep and the state-of-the-art algorithms, whilst illustrating the shortcomings of benchmark algorithms, which greedily select next-hop nodes and do not address congestion control.

5.2.1 Congesting Rate

The observed congesting rate levels across all three datasets and all levels of congestion show that CafRep congests at the lowest rate in comparison with other replication techniques, 54% lower than SF, 40% less than RR, 48% less than EBR and 52% less than Prophet. Café congests at the lowest rate 42% less then CafRep.

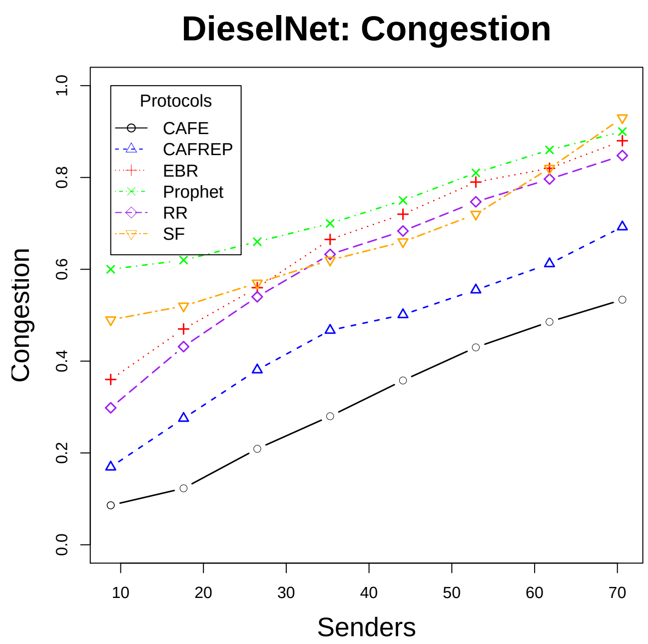


Figure 5.16: Congesting Rate for DieselNet

Figure 5.16 show that CafRep congests at the lowest rate in comparison with other replication techniques, 46% lower than SF, 36% less than RR, 44% less than EBR and 62% less than Prophet. Café congests at the lowest rate 53%

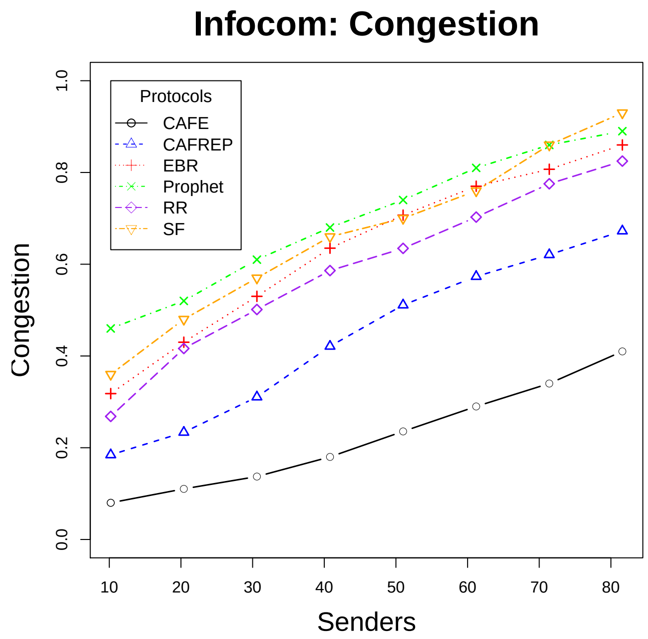


Figure 5.17: Congesting Rate for Infocom 2006

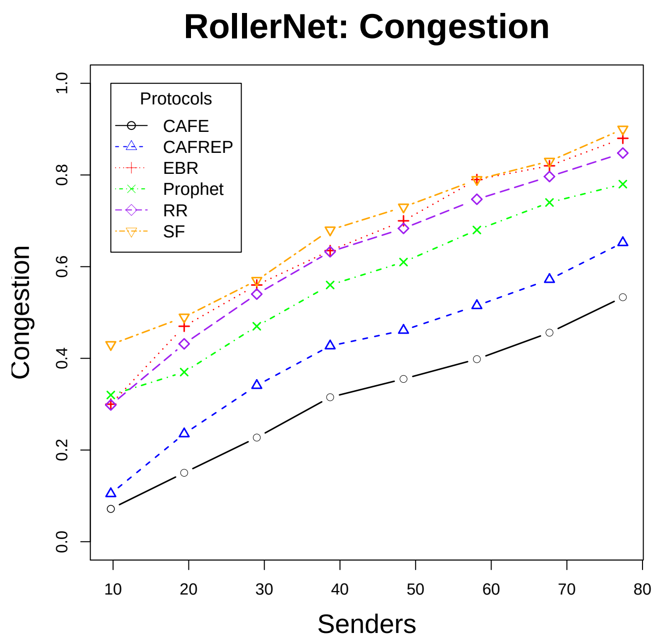


Figure 5.18: Congesting Rate for RollerNet

less then CafRep. Figure 5.17 show that CafRep congests at the lowest rate in comparison with other replication techniques, 51% lower than SF, 33% less than RR, 43% less than EBR and 57% less than Prophet. Café congests at the lowest rate 49% less then CafRep. Figure 5.18 show that CafRep congests at the lowest rate in comparison with other replication techniques, 64% lower than SF, 50% less than RR, 56% less than EBR and 37% less than Prophet. Café congests at the lowest rate 24% less then CafRep.

We observe that our single copy congestion aware forwarding algorithm Café observes a significantly lower level of congestion than all other methods of forwarding and that multi copy congestion aware forwarding also produces comparatively low levels of congestion. In DieselNet and RollerNet we observe an almost constant cost of replication between Café and CafRep, but in Infocom we observe that in highly congested scenarios the cost of replicating doubles in comparison with the low congestion scenarios.

5.2.2 Buffer Availability

The availability across all three datasets for all six protocols shows CafRep outperforms EBR, RR, SF and Prophet. On average CafRep has 57% available storage, Café improves on this with 72% available storage due to it being a single copy strategy. EBR, RR, SF and Prophet all congest more with between 39% and 33% available storage on average.

Figure 5.19 shows that in DieselNet Café has highest availability ranging from 90% to close to 60% for all congestion levels as it does not have replication. CafRep maintains high node availability ranging from 80% to 30% for increasing congestion levels. CafRep consistently outperforms EBR, RR, Spray and Focus and Prophet for medium to high congestion levels. For low congestion levels CafRep has two times higher availability than Prophet and Spray and Focus and up to 20% higher than RR and EBR.

Figure 5.20 shows that in Infocom Café maintains the highest availability ranging from 90% to 60% for increasing congestion levels. CafReps availability ranges from 80% to 35% as congestion increases. When compared to the other

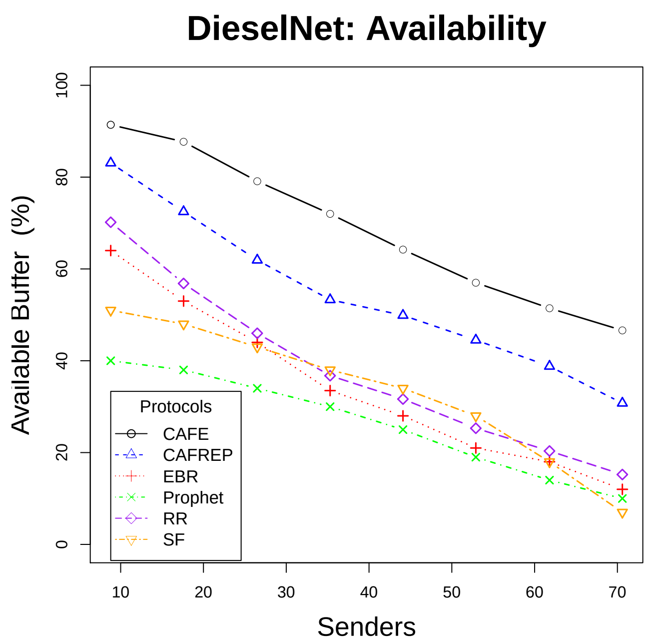


Figure 5.19: Buffer Availability for DieselNet

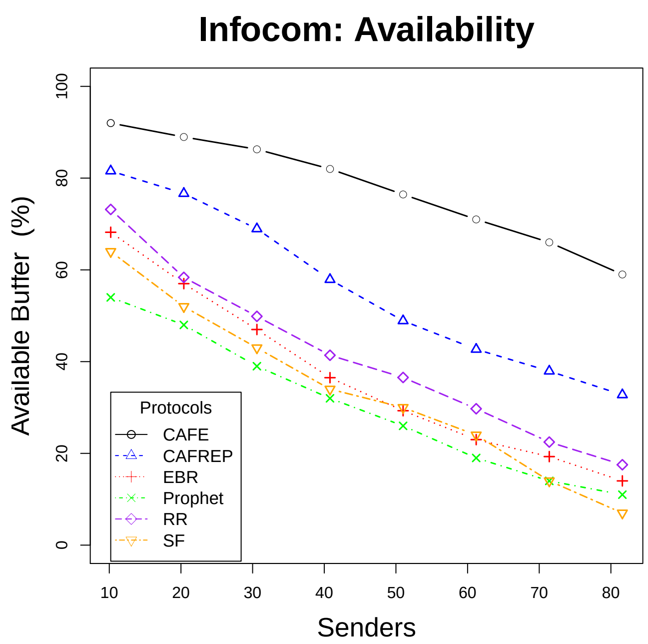


Figure 5.20: Buffer Availability for Infocom 2006

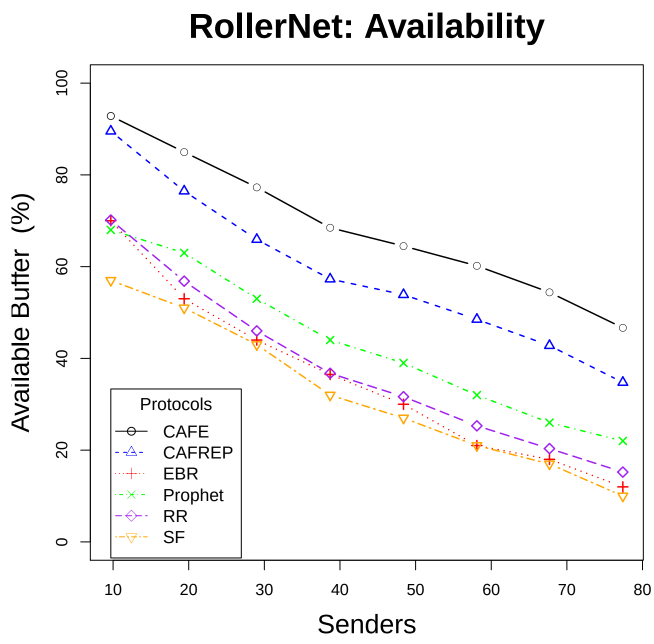


Figure 5.21: Buffer Availability for RollerNet

replication-based protocols, availability levels of CafRep are up to two times higher than availability levels of RR, EBR, Spray and Focus and Prophet for all congestion levels.

Figure 5.21 shows that in RollerNet Café maintains the highest availability for all congestion levels ranging from 90% to 50%. It is interesting to see that CafRep maintains high availability ranging from 80% to 35% for increasing congestion levels. CafRep availability is persistently higher than EBR, RR, Spray and Focus and Prophet. For low congestion levels, CafRep is 20% better than EBR and RR and more than two times better than Spray and Focus and Prophet. For medium to high congestion levels, CafRep is more than two times better than all the other replication-based protocols.

Similarly to the observed congesting rate, the lowest observed buffer consumption is achieved by Café, followed by CafRep, again this indicates the cost of replicating data, but CafRep's level of available resources is significantly higher than benchmark and state-of-the-art algorithms and considering the reduction in delay and transfer cost, and the significant increase in delivered messages, both in comparison to other algorithms and our Café algorithm, therefore CafRep's slightly reduced buffer availability is a proportional cost.

5.3 Evaluation Summary

Our results show that for the majority of our experiments across the three dataset CafRep replicates fewer packets than the other algorithms. This is compelling as CafRep, with fewer replicated messages, increases success ratio and the number of delivered packets, and reduces the message delay and the number of dropped packets, while keeping node buffer availability high and congesting at a substantially lower rate. CafRep's performance improvements are a direct result of being sensitive to resource availability. By moving traffic into the network where possible and preventing key nodes from being exploited CafRep is able to benefit from the path explosion phenomenon. Because when key nodes are congested CafRep forwards to nodes that have slightly lower forwarding potential, but more resources availability, the network experiences shorter buffer queues, shorter delays and reduced packet loss.

Chapter 6

Real World Social Network Application Evaluation

This Chapter is concerned with the effects that different traffic models have on the performance of our work, more specifically this Chapter evaluates the impact that real world social networking usage patterns have on both benchmark DTN forwarding algorithms and state-of-the-art DTN congestion control algorithms in comparison to CafRep. The graphs in this Chapter illustrate 3 experiments: low traffic profiles, randomly selected traffic profiles and high traffic profiles, as outlined in Section 4.3.2. Each traffic profile is evaluated with buffer sizes ranging from 10MB to 100MB, increasing at 10MB intervals. Each experiment is emulated over 10 runs, with each run having a different random seed.

This Chapter presents an extensive evaluation of CafRep across three vastly different connectivity datasets (as described in Section 4.1), in the face of heterogeneous traffic demands, with regards to a wide range of performance metrics in the ONE [43] simulator.

This Chapter evaluates success ratio, delivered packets, transfer cost and message delay in order to assess forwarding performance; and transmitted messages, buffer availability, buffer delay and congesting rate to assess network resource utilisation.

In this Chapter we extensively compare the performance of CafRep against both benchmark DTN forwarding algorithms and state-of-the-art DTN congestion control algorithms. The benchmark algorithms we evaluate against are Direct Delivery (DD), Epidemic (EPI), Prophet (PRO) and SimBetTS (SBTS). The state-of-the-art algorithms we compare CafRep to are Encounter Based Routing (EBR), FairRoute (FR), Retiring Replicas over SimBetTS (RR) and Spray-and-focus (SF).

6.1 Forwarding Performance Evaluation

This Section is concerned with the performance metrics commonly evaluated within the DTN forwarding literature. We evaluate Success Ratio, Delivered Messages, Dropped Messages, Transfer Cost and Message Delay in order to show the effectiveness of both benchmark forwarding algorithms and state-of-the-art congestion control algorithms in the face of a variety of real world application traffic patterns, identifying that CafRep's forwarding performance is not compromised through the addition of congestion control.

6.1.1 Success Ratio

We observe in the box plots in Figures 6.1, 6.2 and 6.3 that of all 3 datasets the RollerNet experiments exhibit the highest success ratio, with all results above 80% for the high load traffic profiles, above 90% for the randomly selected traffic profiles and above 95% for low load traffic profiles. Infocom achieves above 40% in the majority of cases, with few outliers. DieselNet has outliers as low as 20% success ratio for the majority of algorithms, but on contrast exhibits higher median values than Infocom.

For the majority of experiments, regardless of traffic pattern, CafRep is observed to have a higher median success ratio than both benchmark and state-of-the-art algorithms, excluding DD. DD achieves 100% success ratio consistently as it does not forward messages to intermediaries. CafRep is also observed to have a higher 1st Quartile value than the Median values of EPI, PRO, SBTS, RR and SF in a number of experiments. CafRep's success ratio maintains a median success ratio above 80% in all experiments and achieves a median success ratio above 98% in over half of the experiments.

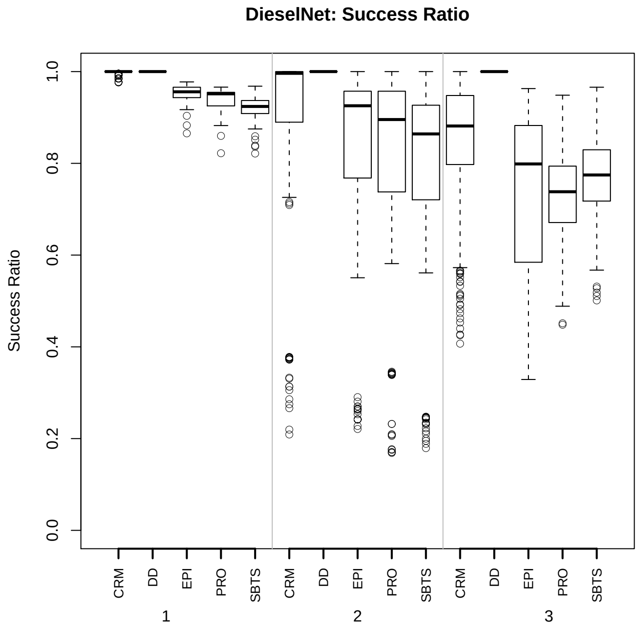
These results show that CafRep is resilient to heterogeneous traffic demands over diverse deployment scenarios. We observe that algorithms which were evaluated under the assumption of unlimited resources, which select next hop nodes based on a greedy locally calculated heuristic, perform inadequately, especially under the pressures of elevated traffic demands. We also observe that in simulations with low traffic demands CafRep outperforms the greedy forwarding approaches, this is compelling as it shows that even in low levels of traffic it is less effective to employ a greedy algorithm than an algorithm that selects less optimal routes in favour of reducing in-network contention.

6.1.2 Delivered Messages

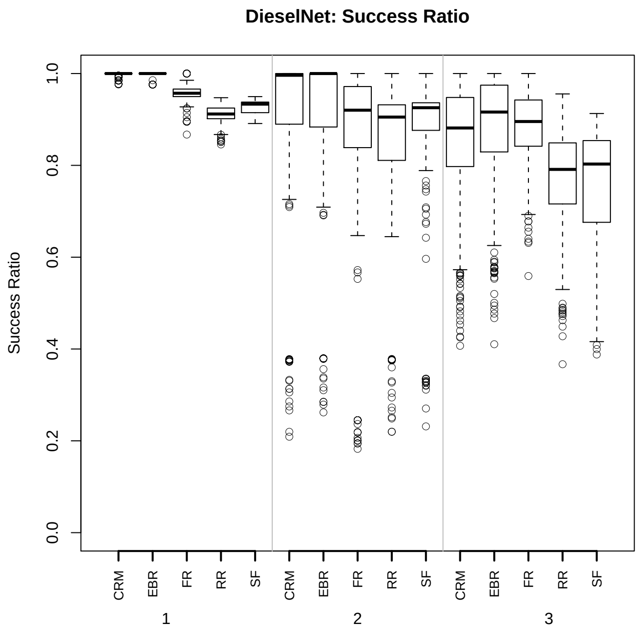
We observe in the box plots in Figures 6.4, 6.5 and 6.6 that of all 3 datasets the DieselNet experiments show the highest number of delivered packets. Infocom has lower maximum and upper quartile values, but a similar median to DieselNet. RollerNet exhibits a much lower number of delivered packets, around a quarter of the values observed in the Infocom experiments. We observe little difference between the algorithms in DieselNet and RollerNet in comparison to Infocom, which illustrates that CafRep delivers substantially more than the benchmark and state-of-the-art algorithms.

For the majority of experiments, regardless of traffic pattern, CafRep is observed to have higher total number of delivered messages than both benchmark and state-of-the-art algorithms. In Infocom CafRep's lower quartile is greater than the median of the majority of benchmark algorithm's, as well as being above and almost above the upper quartile of EBR and FR respectively.

Epidemic forwarding is the best benchmark for messages delivery, as nodes persistently forward as much data through the network as possible, which is why it is compelling to observe that CafRep delivers more messages than Epidemic over both the Infocom and RollerNet datasets. We observe that greedy

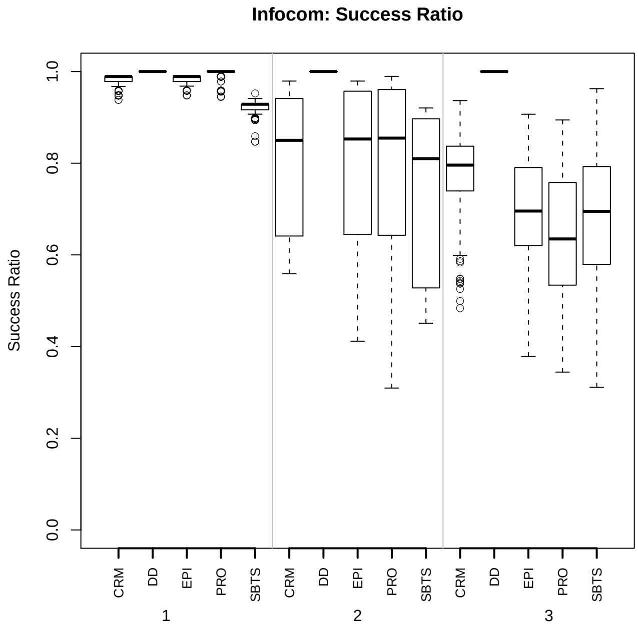


(a) Benchmark / DieselNet

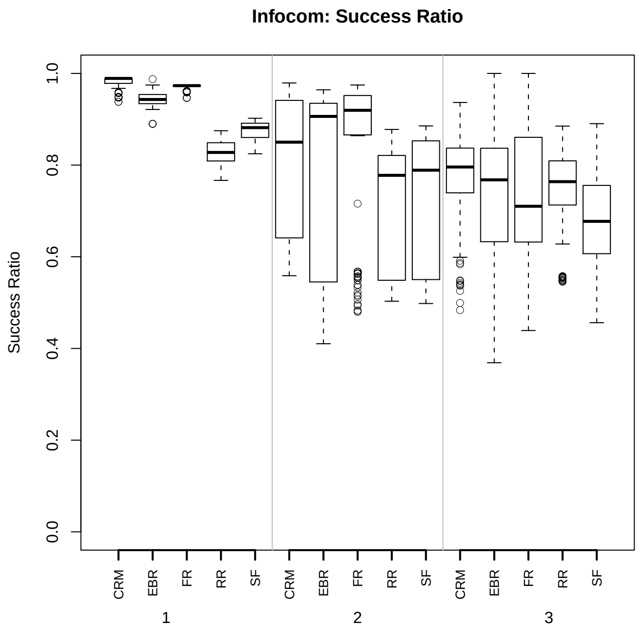


(b) State-of-the-art / DieselNet

Figure 6.1: Success Ratio for DieselNet

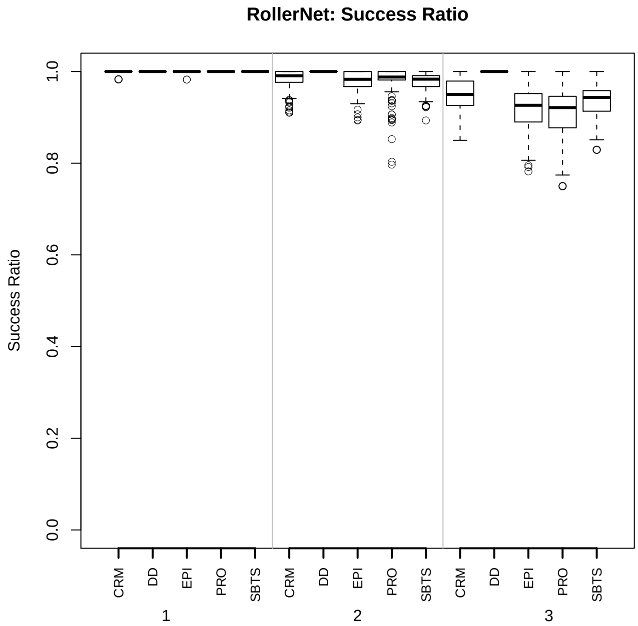


(a) Benchmark / Infocom 2006

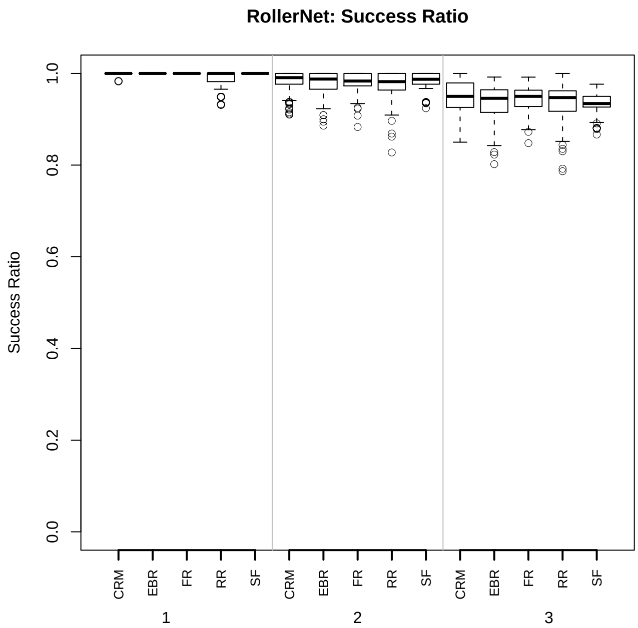


(b) State-of-the-art / Infocom 2006

Figure 6.2: Success Ratio for Infocom 2006



(a) Benchmark / RollerNet



(b) State-of-the-art / RollerNet

Figure 6.3: Success Ratio for RollerNet

algorithms, despite selecting the optimal route through the network, continually deliver an inadequate number of messages in comparison to CafRep.

6.1.3 Dropped Messages

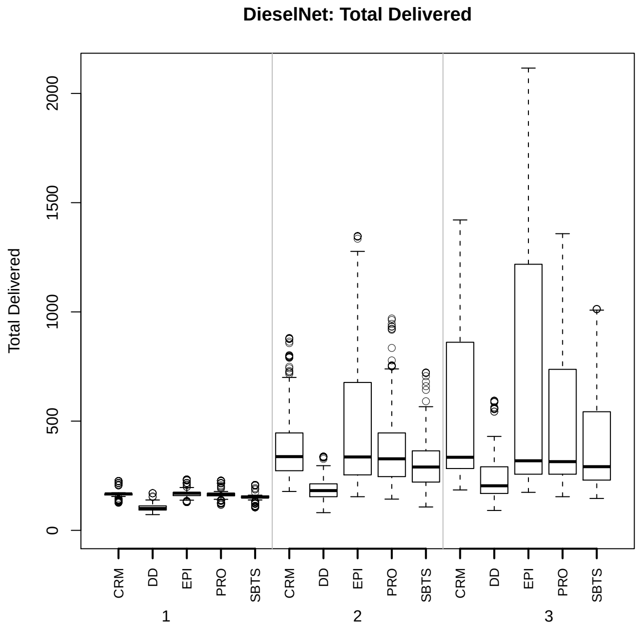
DieselNet has typically higher levels of dropped messages than exhibited in Infocom and RollerNet. Infocom has typically low median values, apart from EPI, PRO and EBR which have similarly high levels of dropped packets in comparison with the DieselNet results. The RollerNet experiments produce half of the level of dropped messages compared to the DieselNet emulations.

We observe in the box plots in Figures 6.7, 6.8 and 6.9 that CafRep has low levels of dropped messages in comparison to benchmark algorithms EPI, PRO and SBTS. DD shows little sign of dropping messages, but this does not represent the messages dropped by DD at the source nodes, when the buffer fills, which is a result of DD not forwarding to intermediaries. In comparison to the state-of-the-art algorithms CafRep drops fewer than EBR and FR in the majority of experiments and marginally more than RR and SF. Message dropping in CafRep is expected, as packet loss is used as a congestion signal, which results in a reduction in the message replication limit, which reduces the number of redundant messages produced in times of increased congestion.

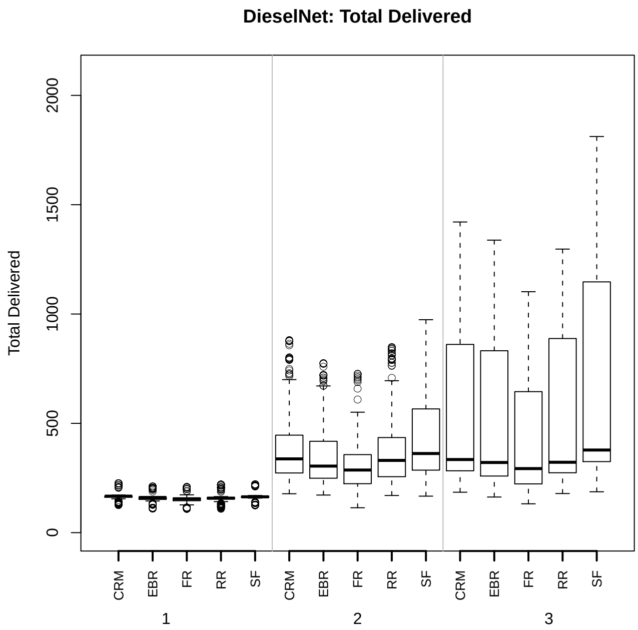
6.1.4 Transfer Cost

The largest transfer cost is observed across Figures 6.10, 6.11 and 6.12 are for Infocom, but the results exhibit relatively low median costs in comparison to the maximum values. RollerNet experiments exhibit similar median transfer costs to Infocom, but with lower maximum transfer costs. Emulations using the DieselNet traces have substantially lower transfer costs, with median values around 10% of the medians of the other two datasets, which is expected due to the small number of transfer opportunities due to the scarcity of node encounters.

In comparison to the benchmark algorithms CafRep's cost for transferring messages is high, but as the traffic demands increase CafRep becomes twice as

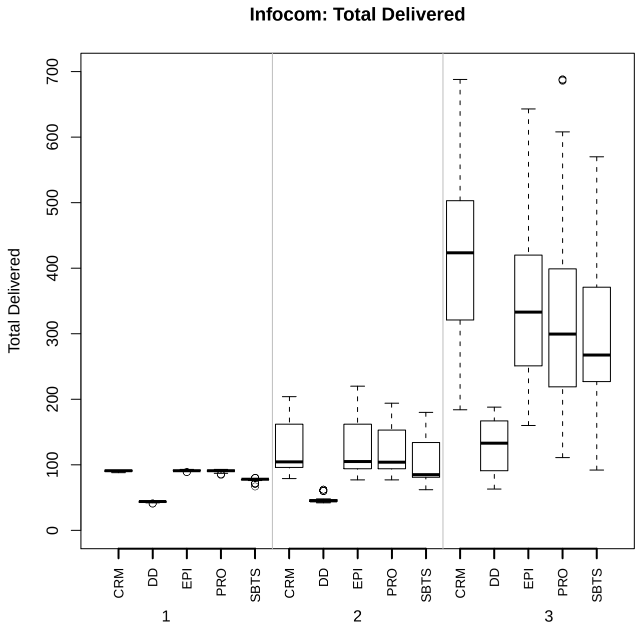


(a) Benchmark / DieselNet

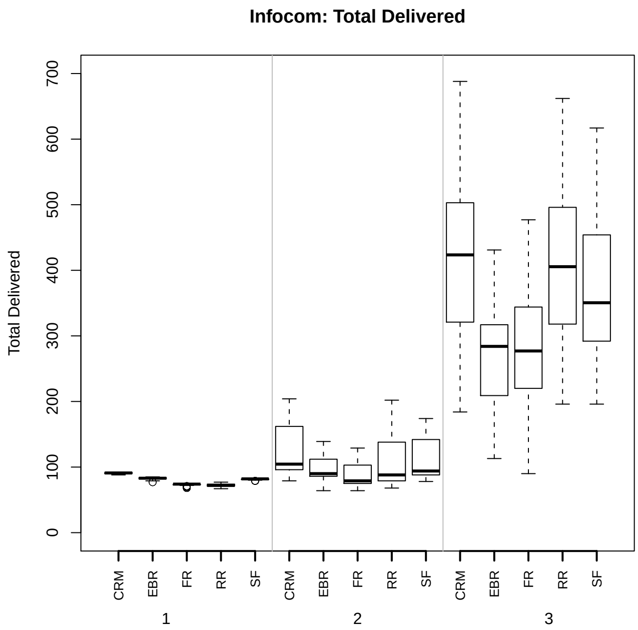


(b) State-of-the-art / DieselNet

Figure 6.4: Delivered Messages for DieselNet

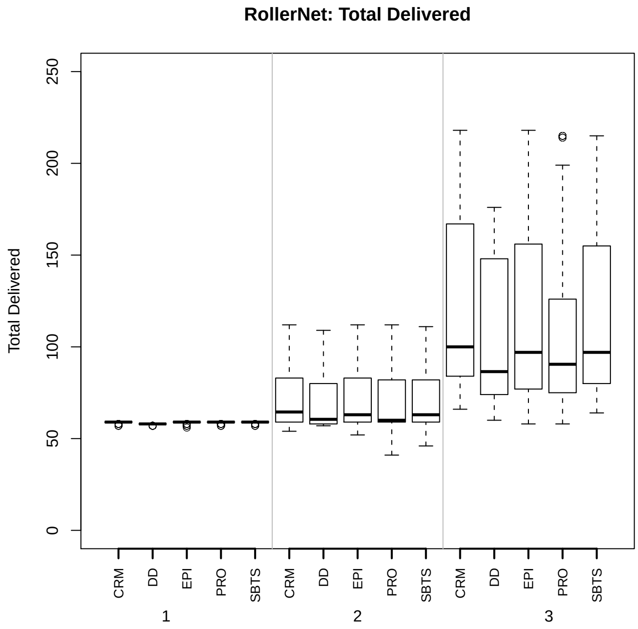


(a) Benchmark / Infocom 2006

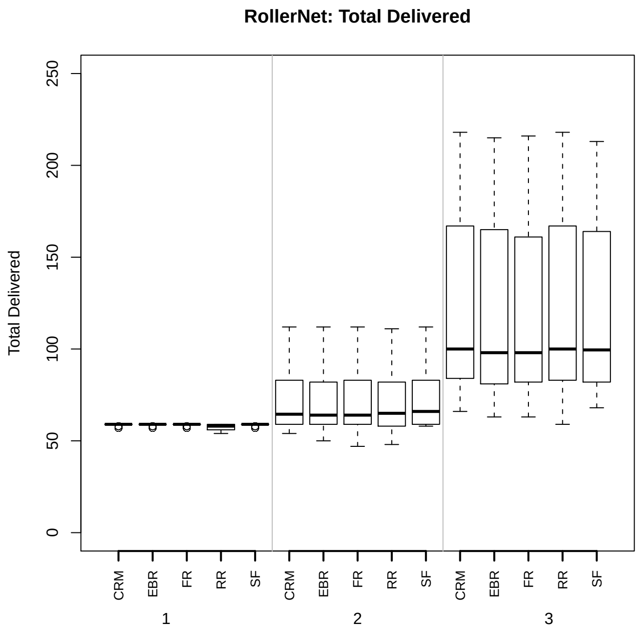


(b) State-of-the-art / Infocom 2006

Figure 6.5: Delivered Messages for Infocom 2006

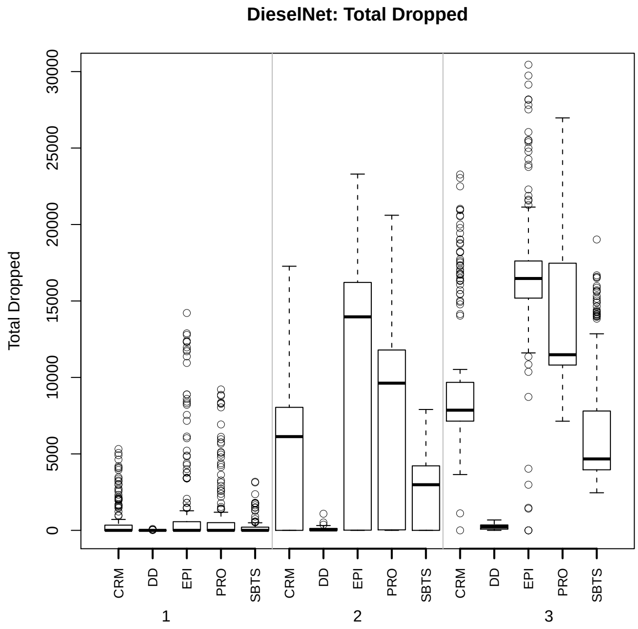


(a) Benchmark / RollerNet

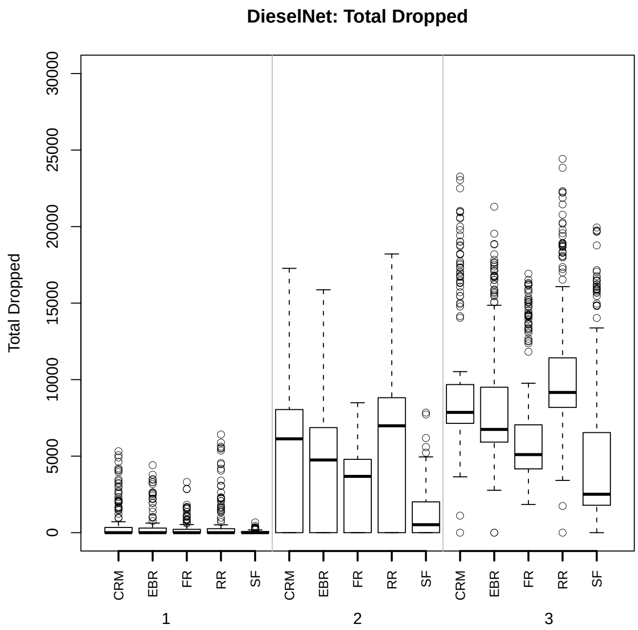


(b) State-of-the-art / RollerNet

Figure 6.6: Delivered Messages for RollerNet

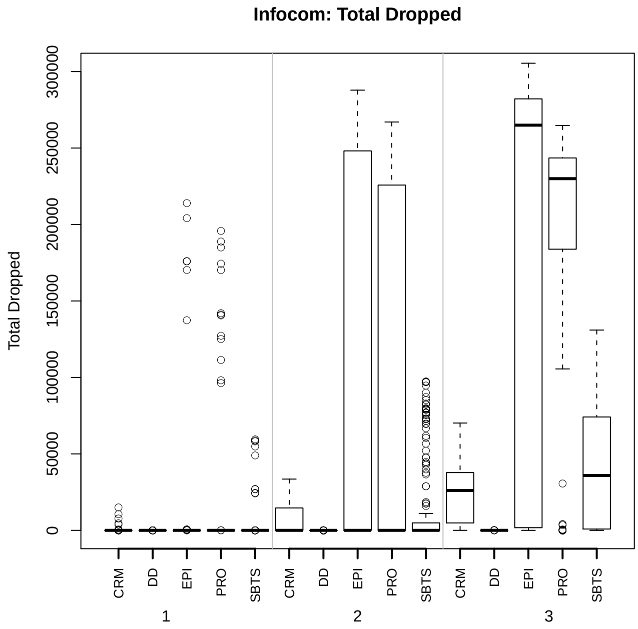


(a) Benchmark / DieselNet

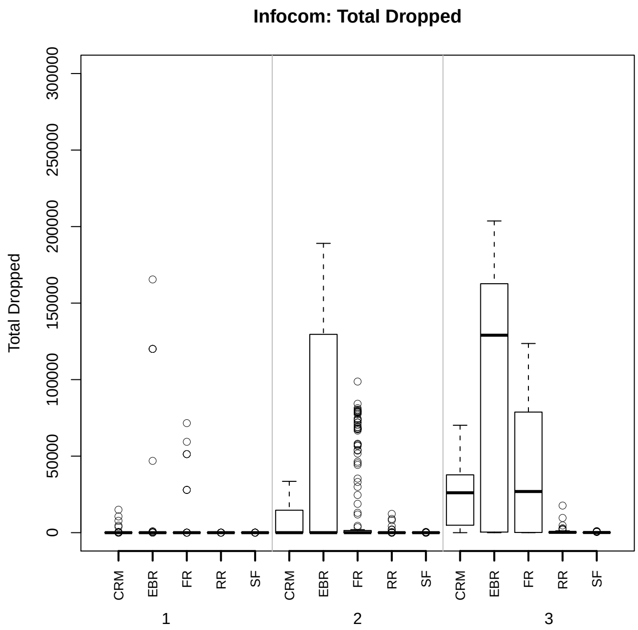


(b) State-of-the-art / DieselNet

Figure 6.7: Dropped Messages for DieselNet

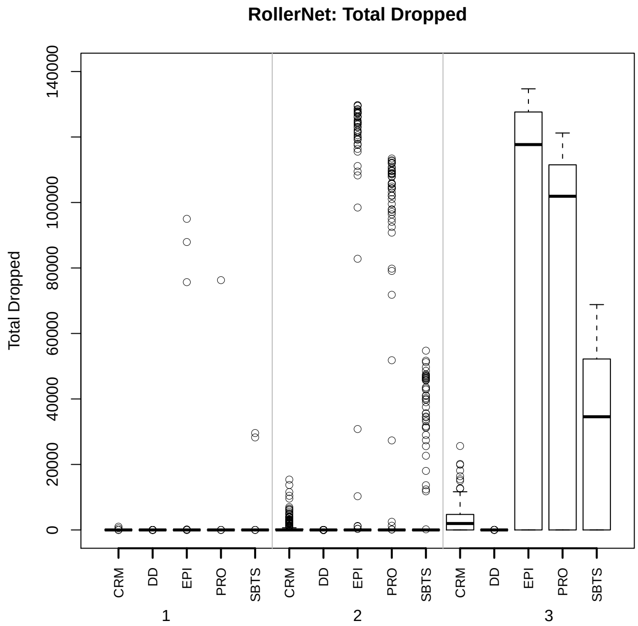


(a) Benchmark / Infocom 2006

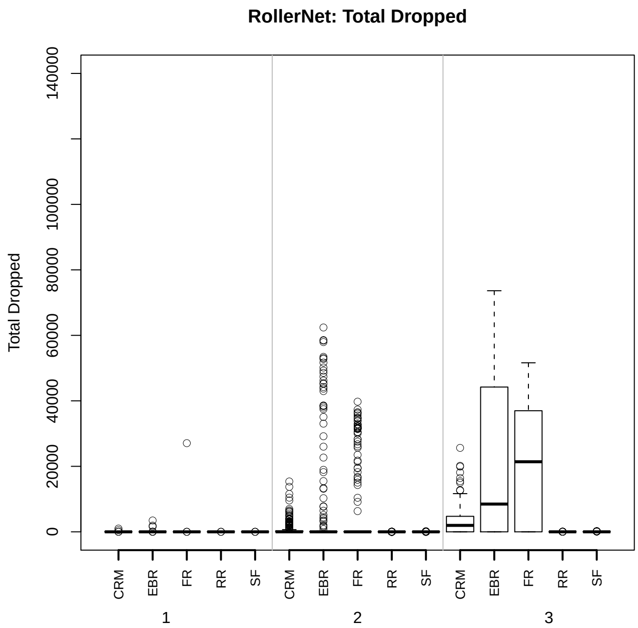


(b) State-of-the-art / Infocom 2006

Figure 6.8: Dropped Messages for Infocom 2006



(a) Benchmark / RollerNet



(b) State-of-the-art / RollerNet

Figure 6.9: Dropped Messages for RollerNet

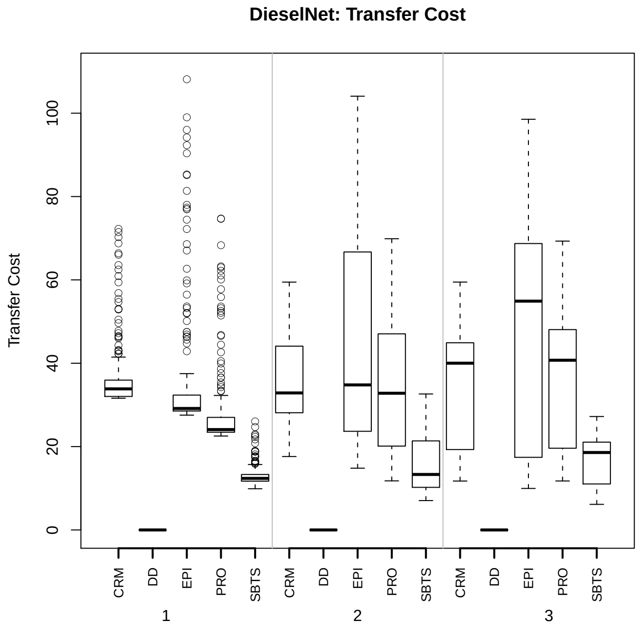
efficient as EPI and PRO and is only slightly more costly on average than SBTS, which is significant as SBTS is a single copy forwarding algorithm and as such does not forward redundant copies of messages - this shows that despite the fact CafRep replicates messages, it is able to maintain a transfer cost similarly to it's forwarding heuristic only counterpart, which is indicative of CafRep's efficient use of network resources. CafRep has a more consistent cost than the other algorithms, this is due to the adaptive copy limit mechanism and copy placement strategy, which aims to utilise the available network resources, without over utilising network resources and causing congestion.

6.1.5 Message Delay

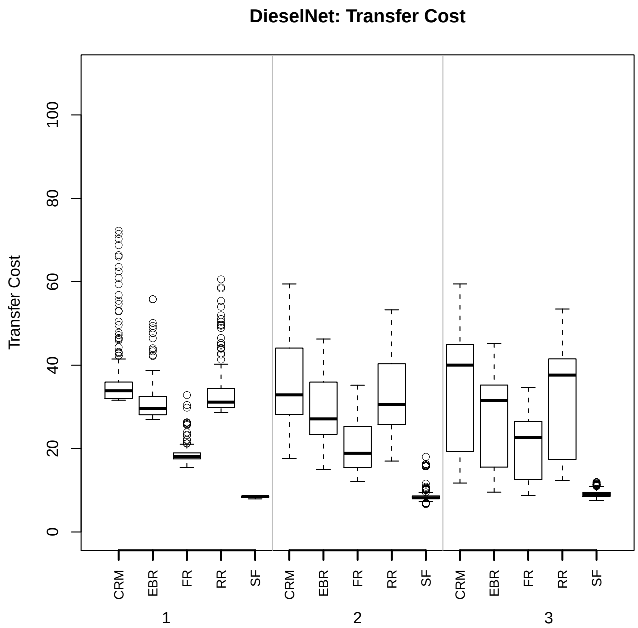
The message delay results observed across Figures 6.13, 6.14 and 6.15 show that DieselNet is substantially higher than the experiments over the other two connectivity traces, this is due to the large isolation periods, RollerNet has the shortest isolation durations and this is reflected in the comparatively short delays.

CafRep typically has a low level of delivery delay, this becomes more noticeable as traffic demands increase, which is compelling because it shows that when the network has a higher load CafRep moves traffic towards less greedily selected routes, routes that offer a less direct connection to the destination and towards nodes and regions of the network that have available resources. CafRep does not forward superfluously, when traffic levels are low CafRep's delivery delay is similar to or lower than both benchmark and state-of-the-art algorithms.

Our observations across all three datasets and traffic demands are compelling, as they show that although CafRep forwards to nodes that would be classified as suboptimal by most state of the forwarding algorithms. CafRep is able to reduce the delivery delay by avoiding congesting regions, allowing messages to traverse the network quickly, avoiding messages being added to the end of the queue of an already overwhelm node, which could potentially result in the message being dropped.

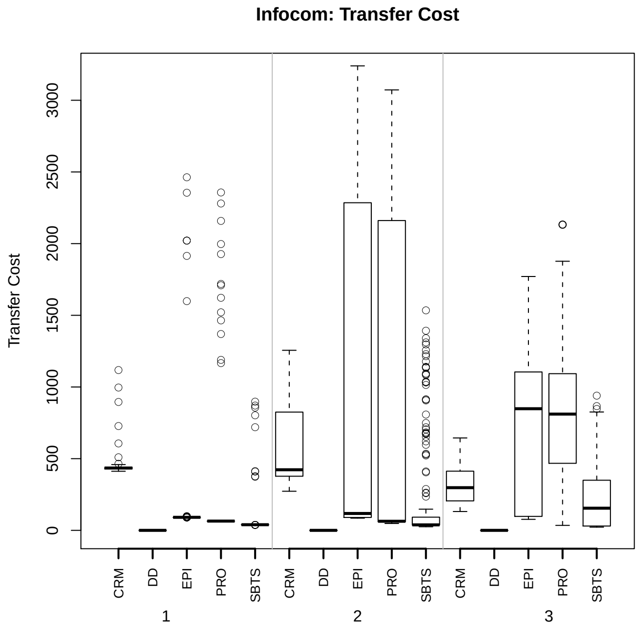


(a) Benchmark / DieselNet

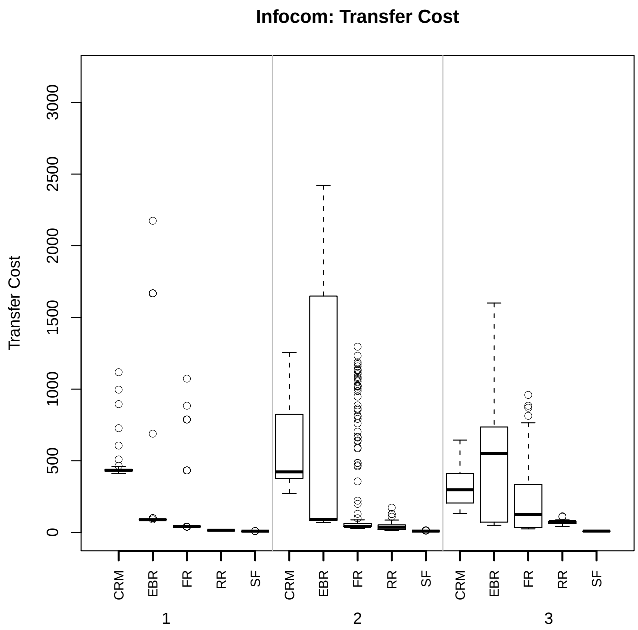


(b) State-of-the-art / DieselNet

Figure 6.10: Transfer Cost for DieselNet

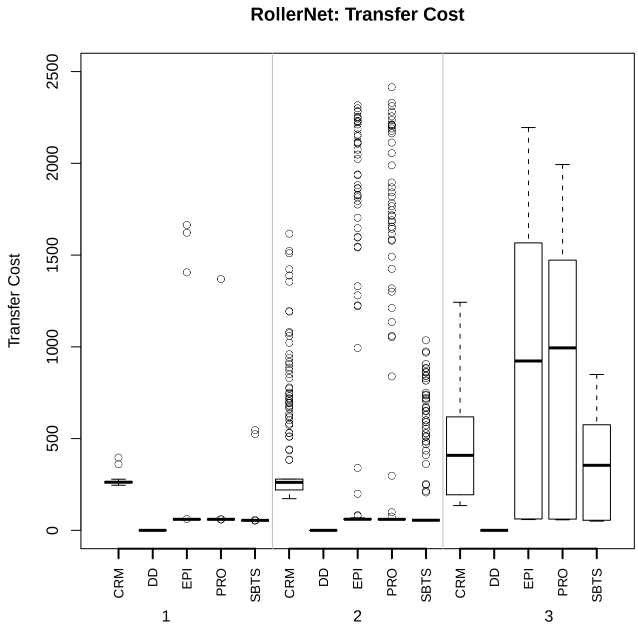


(a) Benchmark / Infocom 2006

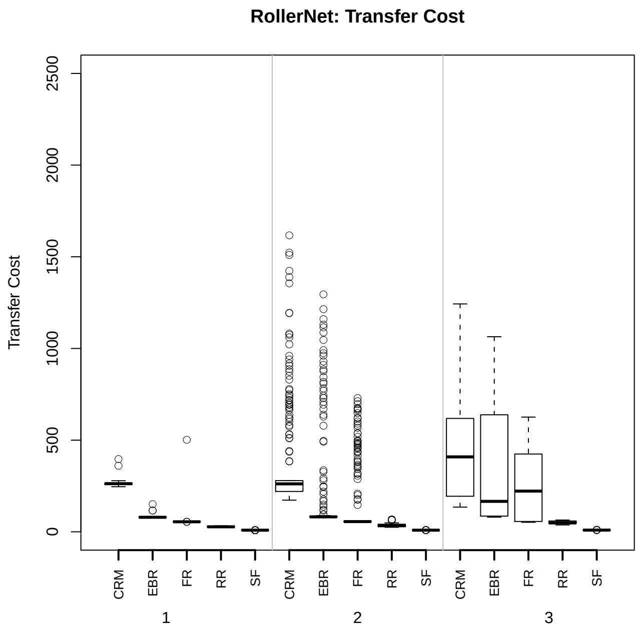


(b) State-of-the-art / Infocom 2006

Figure 6.11: Transfer Cost for Infocom 2006

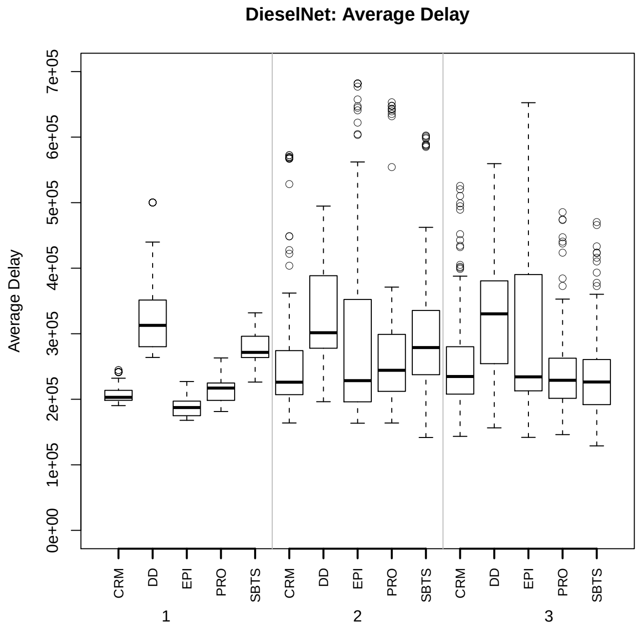


(a) Benchmark / RollerNet

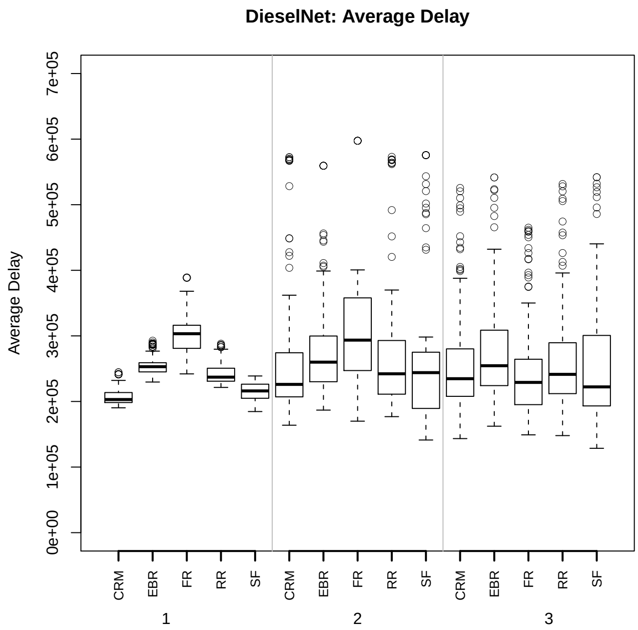


(b) State-of-the-art / RollerNet

Figure 6.12: Transfer Cost for RollerNet

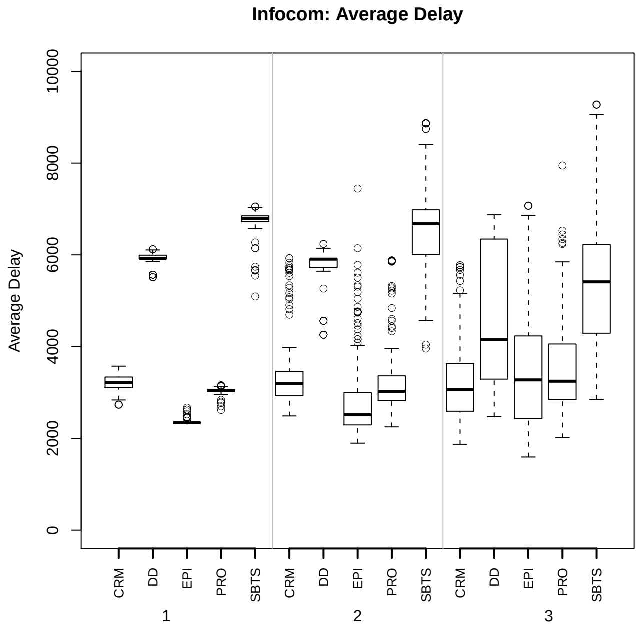


(a) Benchmark / DieselNet

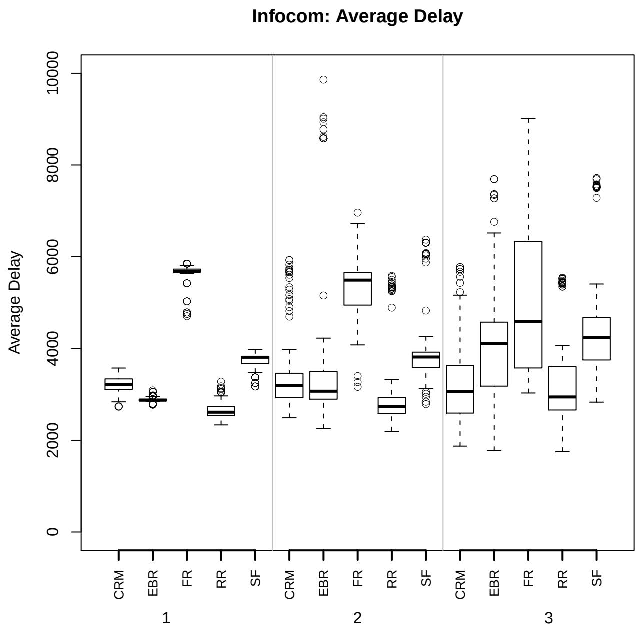


(b) State-of-the-art / DieselNet

Figure 6.13: Message Delay for DieselNet

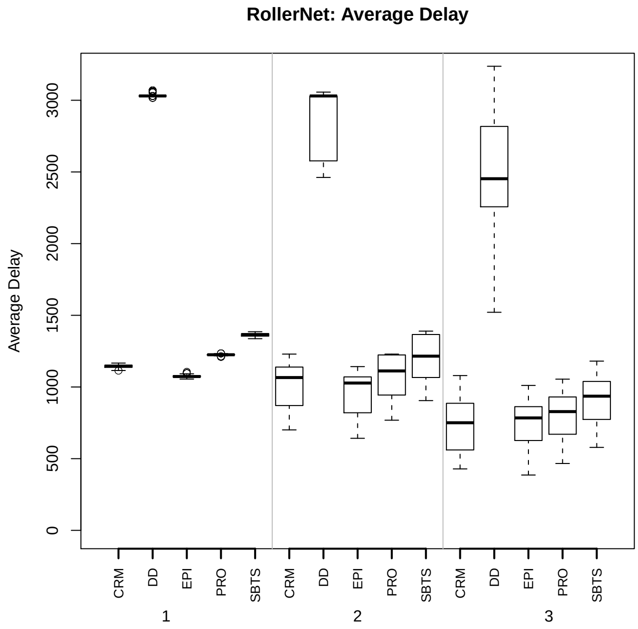


(a) Benchmark / Infocom 2006

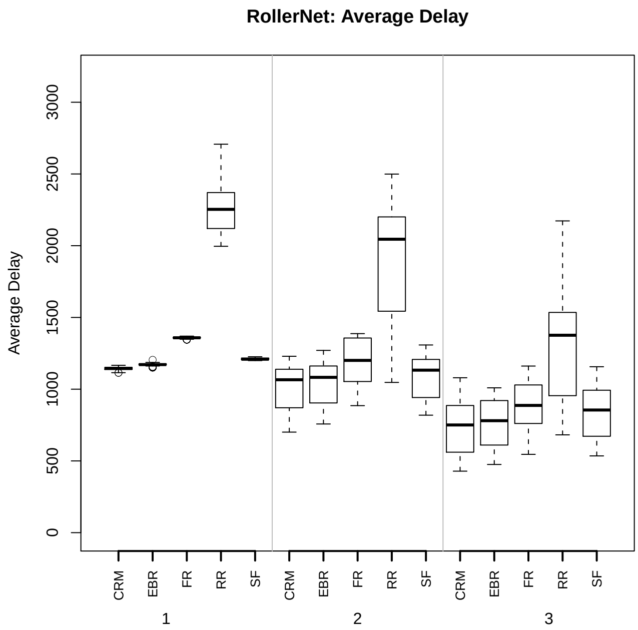


(b) State-of-the-art / Infocom 2006

Figure 6.14: Message Delay for Infocom 2006



(a) Benchmark / RollerNet



(b) State-of-the-art / RollerNet

Figure 6.15: Message Delay for RollerNet

6.2 Network Resource Utilisation Evaluation

This Section is concerned with the evaluation of metrics that identify the effectiveness of the congestion control mechanisms of CafRep and the other state-of-the-art algorithms, whilst illustrating the shortcomings of benchmark algorithms, which greedily select next-hop nodes and do not address congestion control. We evaluate the effectiveness of CafRep's use of the available network resources by comparing the number of messages that are transmitted into the network, the buffer availability of nodes and delay within the nodes buffers, against both benchmark and state-of-the-art algorithms.

6.2.1 Number of Transmitted Messages

This metric allows us to observe the quantity of unique messages forwarded into the network, this value can be low if the algorithm forwards too many message replicas into the network, thus not allowing for unique messages to begin transmitting or by the algorithm being overly selective when choosing the next-hop node.

We observe in Figures 6.16, 6.17 and 6.18 that DieselNet has the highest observed throughput, with maximum values of above 4000 messages. Infocom has lower maximum values, but similar median values to DieselNet, with around 500 messages sent on average in the high traffic simulations. RollerNet simulations exhibit much lower sending rates, with median values of around 100 messages sent.

Across all 3 experiments we observe that CafRep forwards a large number of messages into the network, sending more unique messages than all other algorithms over all traffic models for both Infocom and RollerNet scenarios. In times of high traffic demands CafRep disperses traffic forwarding towards nodes with marginally less forwarding potential than the optimal next-hop, in order to promote the use available resources and deter the over-subscription of key resources - this can be observed when CafRep is compared with SBTS (the forwarding algorithm used by CafRep to provide the forwarding heuristic in these evaluations), as SBTS can be observed to forward consistently fewer messages than CafRep.

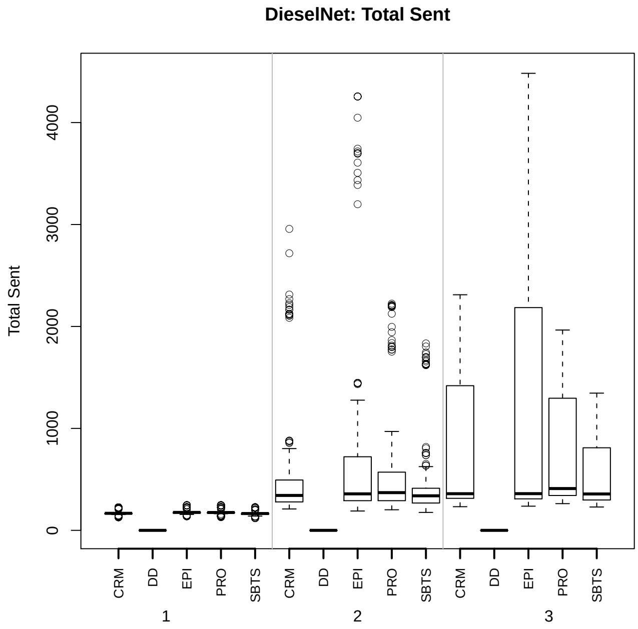
6.2.2 Retentiveness

We monitor this metric in order to indicate if a forwarding strategy detrimentally exploits network storage resources, as over retention can be the cause of high buffer consumption and excessive forwarding causes in-network congestion, both of which can lead to packet loss. A result of this paradox is that the evaluation of an algorithms performance based on buffer consumption is non-trivial, but by referring to the Figures in Section 6.1 we can deduce whether low buffer consumption is due to intelligent forwarding strategy or prolific message dropping.

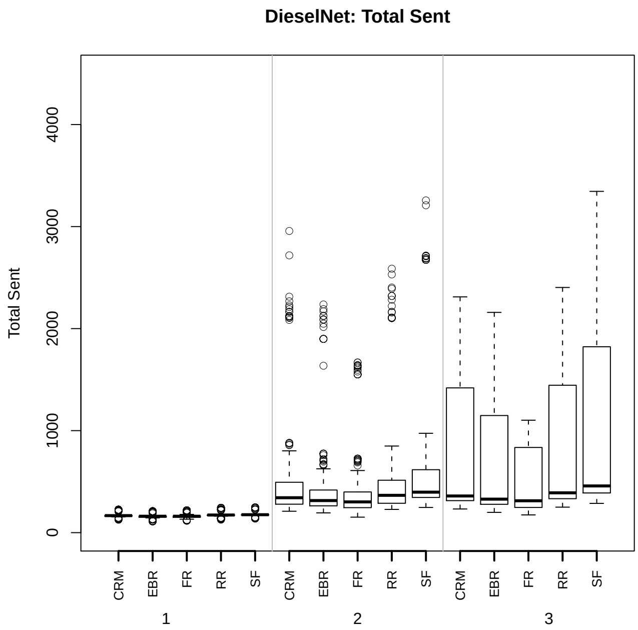
The results for DD are quintessential of over retention, as they display persistently high buffer levels due to excessive message retention - in DD the message buffer is filled by the source node and no messages are transferred unless the destination is met. The results for EPI are exemplary of prolific message dropping, as they show increasing availability as the traffic level increases, this is due to the excessively high level of messages dropped by EPI in order to make room for the persistent influx of new messages.

We observe across Figures 6.19, 6.20 and 6.21 that by avoiding nodes and regions of the network with high levels of observed congestion CafRep is able to maintain a greater level of buffer availability than both benchmark and state-of-the-art algorithms, excluding algorithms that prolifically drop messages. In order to confirm that CafRep does not have a high level of availability due to prolific message dropping, we refer to Figures 6.4, 6.5 and 6.6, which shows that CafRep has low levels of dropped messages in comparison with the other algorithms. We observe that across the 3 traffic patterns over all the datasets CafRep has a median value around 50%, with a tolerance of 20% for low and high traffic profiles.

Our observations for retentiveness are compelling as they show that CafRep not only reduces delay, increases the number of delivered messages and raises

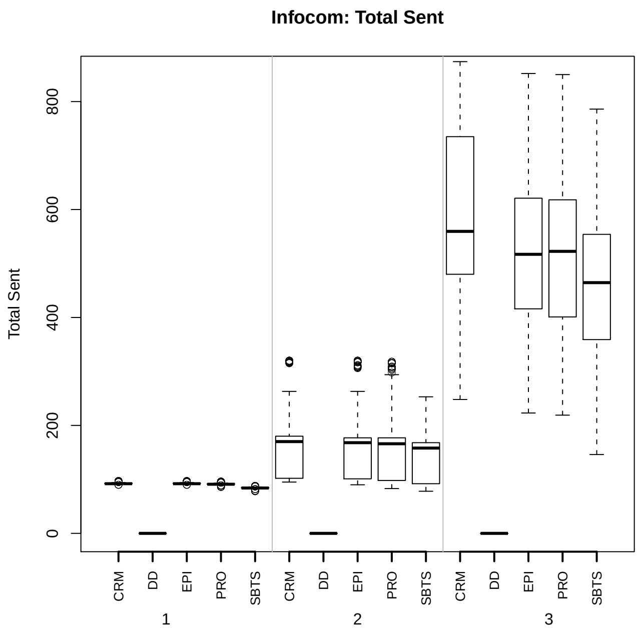


(a) Benchmark / DieselNet

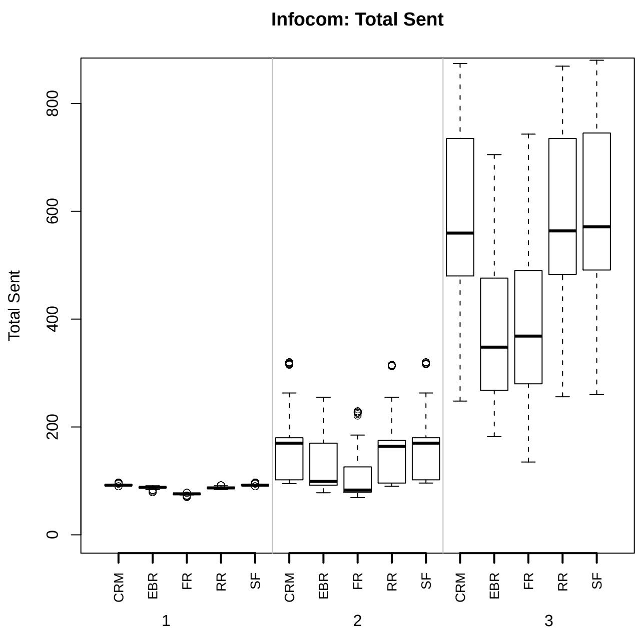


(b) State-of-the-art / DieselNet

Figure 6.16: Transmitted Messages for DieselNet

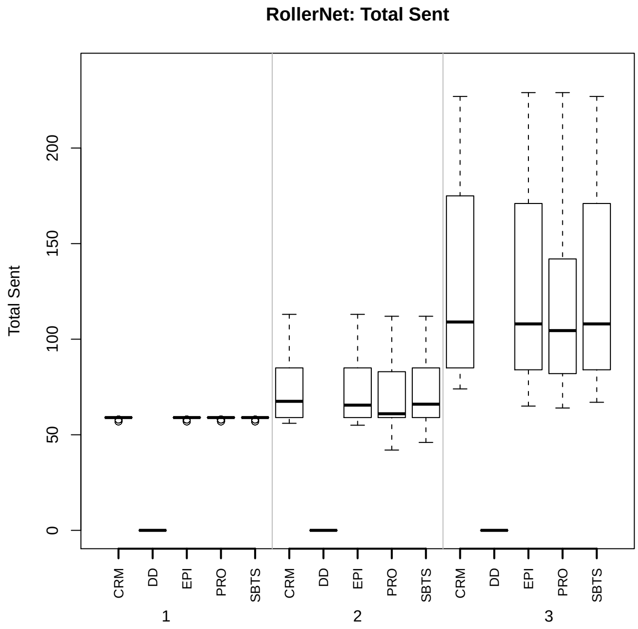


(a) Benchmark / Infocom 2006

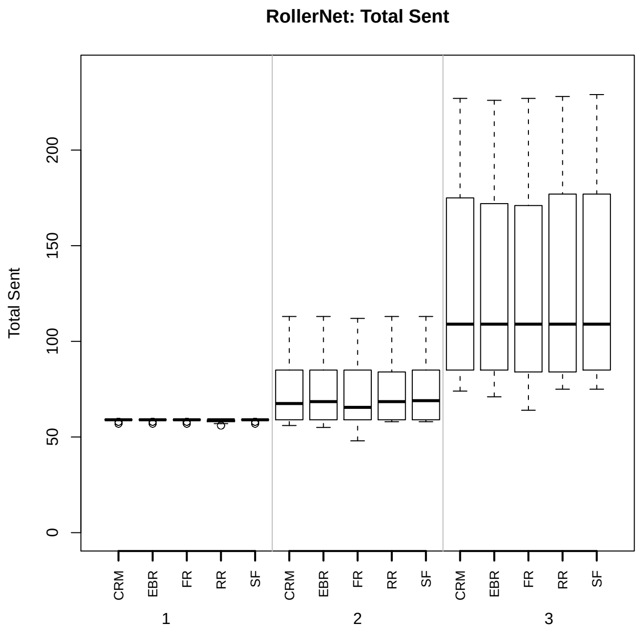


(b) State-of-the-art / Infocom 2006

Figure 6.17: Transmitted Messages for Infocom 2006



(a) Benchmark / RollerNet



(b) State-of-the-art / RollerNet

Figure 6.18: Transmitted Messages for RollerNet

the success rate, but it achieves this without exhausting a nodes resources. We also observe that the proportion of remaining space increases as traffic demands increase, this is due to the files sizes being commonly large, which results in more space becoming available when messages are forwarded and also this signifies CafRep's response to the additional delay's incurred by having to forward this cumbersome traffic.

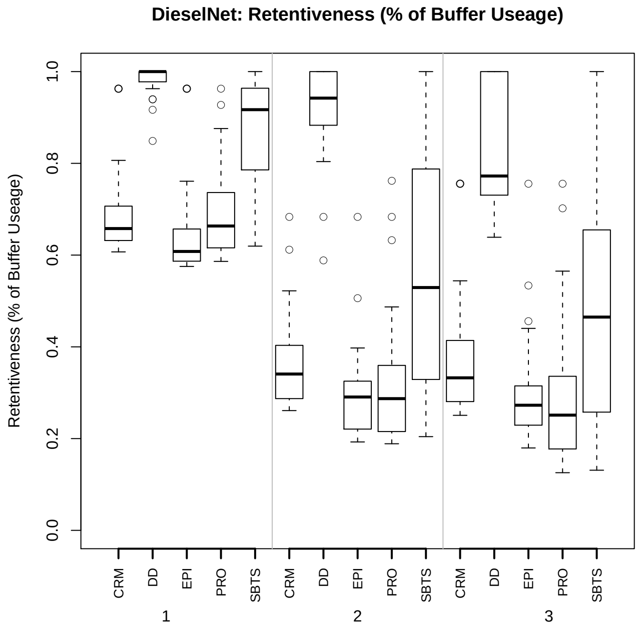
6.2.3 Receptiveness

This metric is a measurement of the level of delay within a nodes buffer, calculated as the difference between the time the sample is taken and the time in which each message was received by the node. This is an interesting metric as it illustrates the in-network delay, which is an indication of buffer queue length and the quality of the algorithm's next-hop selection. Typically a low receptiveness value would indicate an algorithm with good performance, but we observe that when a forwarding strategy leads nodes to prolifically drop messages the observed receptiveness value is low, an example of this is DD.

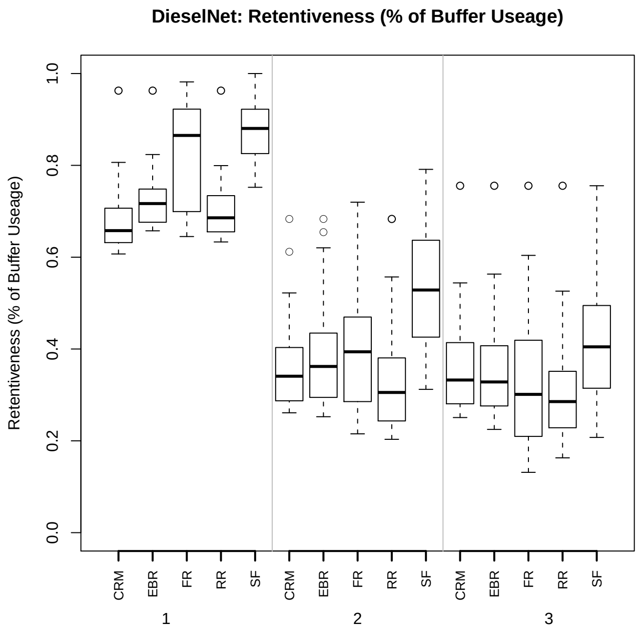
We observe across Figures 6.22, 6.23 and 6.24, similarly to message delivery delays in Section 6.1.5, that delays in DieselNet are substantially higher than the experiments over the other two connectivity traces, this is due to the DieselNet connectivity traces exhibiting large isolation periods. RollerNet has the shortest isolation durations and this is reflected in the comparatively short delays.

CafRep is observed to experience a level of delay which is consistent with the observed receptiveness of the other algorithms, which is compelling, as it shows that despite the fact CafRep forwards towards next-hop nodes that are not greedily selected, in-buffer delay is not greatly elevated. CafRep displays a consistent levels of delay for low and randomly selected traffic profiles across all 3 datasets, in Infocom we observe elevated levels of delay for the high load traffic profile simulations, which is consistent with the observed receptiveness of other algorithms.

As traffic demands increase and nodes begin to congest there is three logical actions a forwarding algorithm can employ: 1) the algorithms continues to

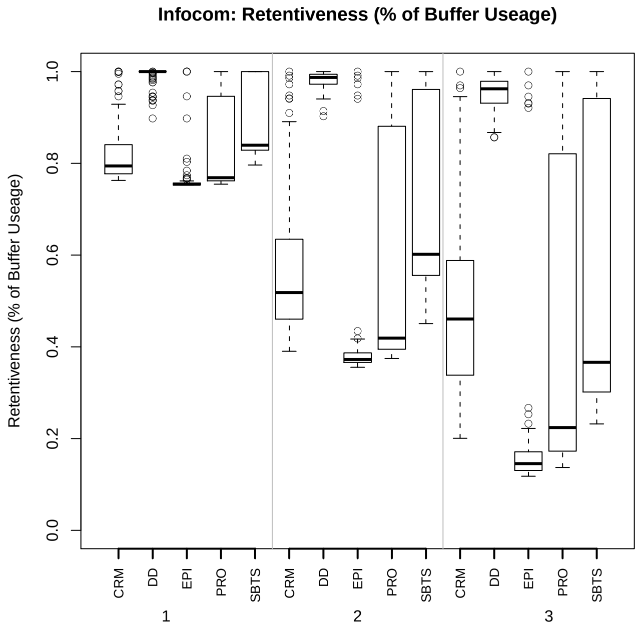


(a) Benchmark / DieselNet

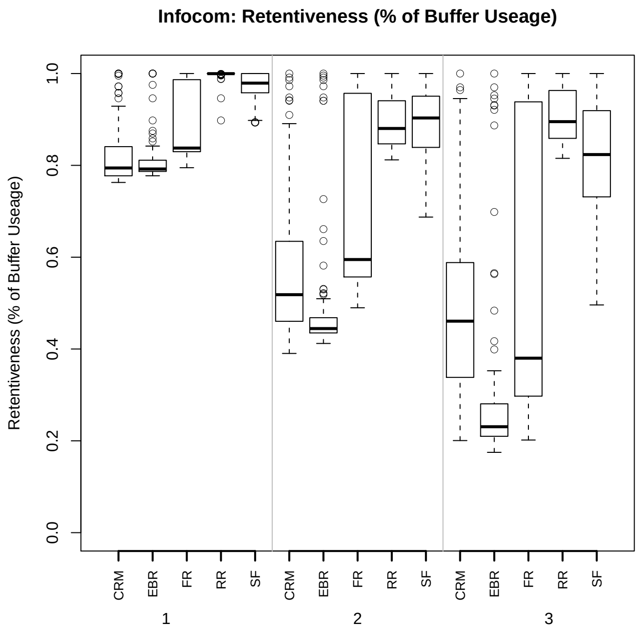


(b) State-of-the-art / DieselNet

Figure 6.19: Retentiveness for DieselNet

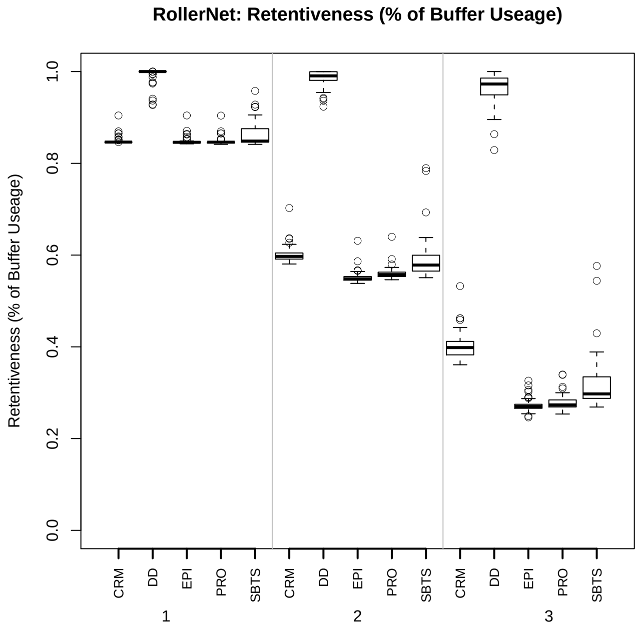


(a) Benchmark / Infocom 2006

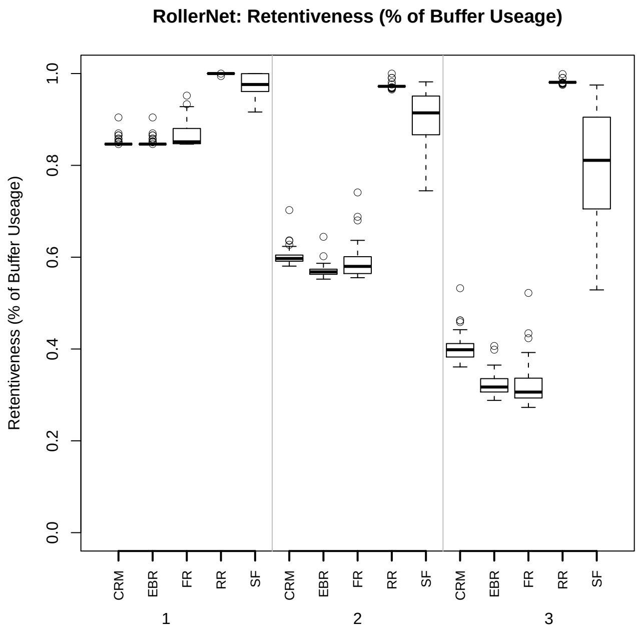


(b) State-of-the-art / Infocom 2006

Figure 6.20: Retentiveness for Infocom 2006



(a) Benchmark / RollerNet



(b) State-of-the-art / RollerNet

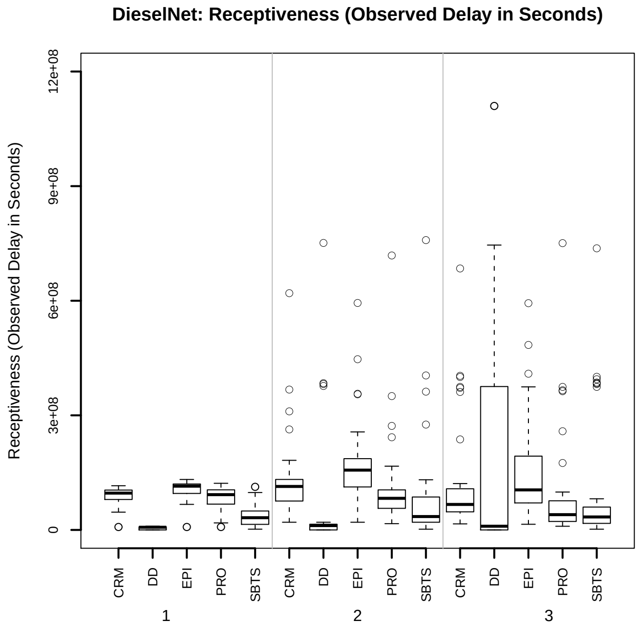
Figure 6.21: Retentiveness for RollerNet

forward to the optimal nodes and congestion collapse occurs; 2) the algorithm reduces the sending rate, which increases message delay and is likely to result in messages being dropped at the source; 3) the algorithm selects another route for the messages to follow, which increases the message delay; CafRep balances these three options, which is why, although it appears to deliver messages at a slower rate than other algorithms, this is a direct result of delivering more messages and dropping fewer messages, both at the source and within the network.

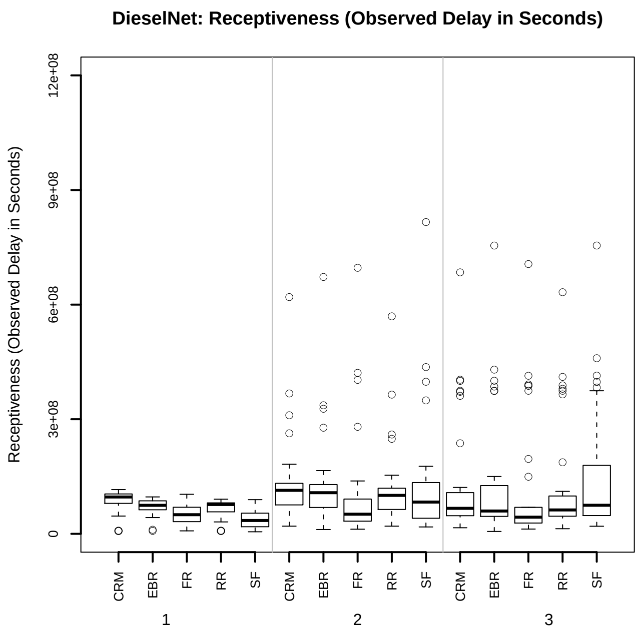
6.3 Evaluation Summary

In this Chapter we have observed that CafRep achieves a higher median success ratio, higher total number of delivered messages, a low levels of dropped messages and a low level of delivery delay than both benchmark and state-of-the-art algorithms, which identifies that CafReps forwarding performance despite the use of longer paths is not compromised through the addition of congestion control.

CafRep makes effective use of the available network resources by forwarding a large number of messages into the network, sending more unique messages than all other algorithms over all traffic models for both Infocom and RollerNet scenarios, maintaining a greater level of buffer availability than both benchmark and state-of-the-art algorithms, while experiencing a level of delay which is consistent with the observed receptiveness of the other algorithms, which is compelling, as it shows that despite the fact CafRep forwards towards next-hop nodes that are not greedily selected, rather than this being at the cost of performance we observe that CafRep improves forwarding performance across a number of metrics.

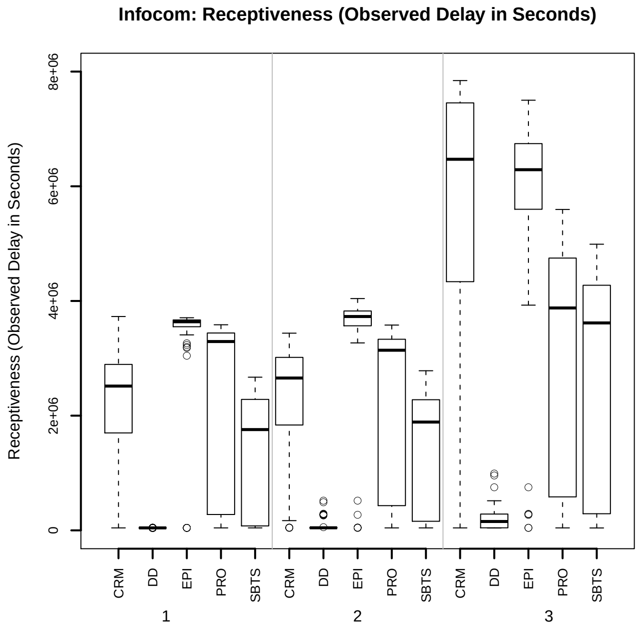


(a) Benchmark / DieselNet

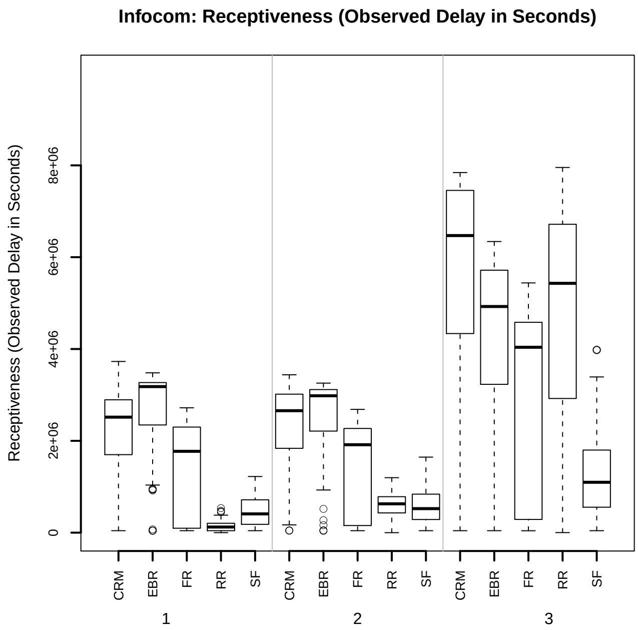


(b) State-of-the-art / DieselNet

Figure 6.22: Receptiveness for DieselNet

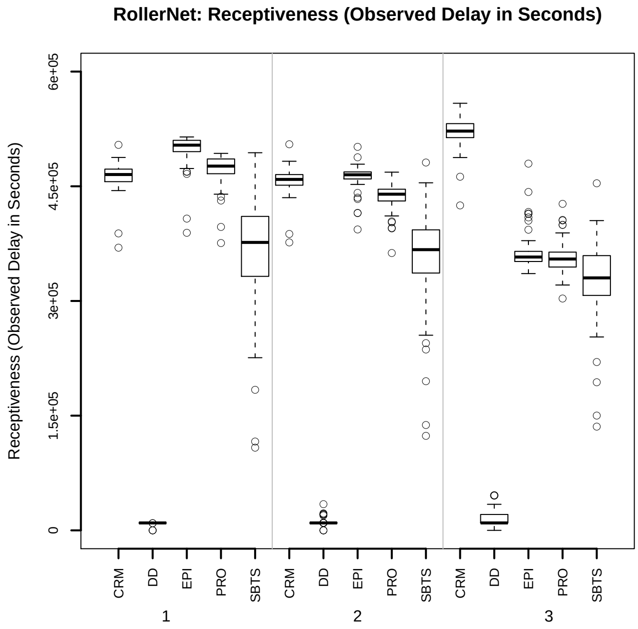


(a) Benchmark / Infocom 2006

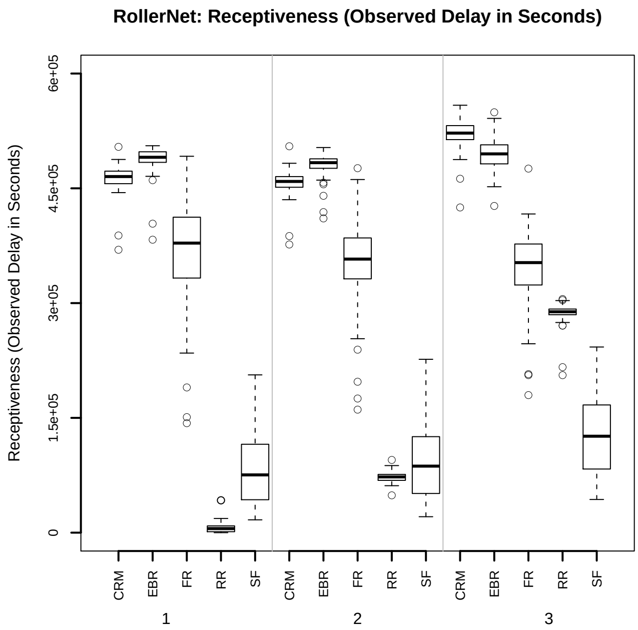


(b) State-of-the-art / Infocom 2006

Figure 6.23: Receptiveness for Infocom 2006



(a) Benchmark / RollerNet



(b) State-of-the-art / RollerNet

Figure 6.24: Receptiveness for RollerNet

Chapter 7

Discussion

This Chapter discusses the possible additional applications of our framework, such as using our congestion indicator to adaptively apply network coding techniques (Section 7.1), or to incorporate additional or alternative signals, such as transmission energy costs (Section 7.2), in order to ensure that forwarding addresses a different or extended criteria.

7.1 Network Coding

There has been a proliferation of interest in applying network coding to delay tolerant networks in order to improve data transmission efficiency, but existing network coding approaches for DTNs do not detect and adapt to congestion in the network. In Section 2.3.5 we review HubCode [2], E-NCP [52] and FairMix [42] as methods capable of increasing efficiency, but we also identify that packet loss could have a greater impact on the networks delivery rate if packets are coded [15], as all messages need to be received in order to extract every component message, which is significant due to the lack of reliability in disconnection prone networks.

Static network coding tequniques cannot support efficient communication in light of varying connectivity, mobility and application patterns. We propose CafNC, which extends the state of the art network coding in DTNs to be more dynamic and flexible in order to avoid dropping messages, particularly those that are network coded, as message dropping is especially harmful to coded packets [15]. Although, it is also essential that coding is not over restricted, as missing encoding opportunities may potentially cause huge delays and lower success ratios. In order to further decrease delays and to increase fairness, CafNC aims to code messages together that are on the same topic or sent to the same receiver.

Rather than choosing statically that network coding is performed on a pre-chosen set of highly central nodes or demanding that nodes perpetually perform network coding [2], we propose that network coding is performed on any node that determines that it has enough resources and enough packets to do the network coding. By allowing the network coding to be performed dynamically our network coding policy adaptively prevents network coding in the parts of the network that have low buffer availability, increased node delays and little or no interest in the content, and performs network coding in parts of the network with higher buffer availability, lower node delays, slower congesting rates and with a greater level of interest in the content.

Selecting which node represents the best carrier for the set of messages and deciding whether to network code them are both multiple attribute decision problems where the aim is to select the node that provides the maximum utility for carrying certain messages and only coding messages if the next hop is interested in the content and is capable of accepting a coded message without becoming overloaded. Similarly to Café and CafRep, we achieve this by employing a utility function that comprises of measurements of relative gain, loss or equality, calculated as pair-wise comparison between the node's own parameters and that of an encountered contact.

$$NCRate = \lfloor CaféUtil_D(X, C_i(X)) + 1 - CafNCThreshold \rfloor \tag{7.1}$$

Formula 7.1 shows how we dynamically calculate whether messages should be coded when forwarding by evaluating $CaféUtil_D$ against $CafNCThreshold$ a predetermined value between 0 and 1, which when high reduces the amount of coding a node carries out, and when low relaxes the coding criteria. This results in CafNC detecting and exploiting more coding opportunities, on a wider range of nodes, when congestion is low in comparison with static coding techniques and adaptively codes less at times of extreme congestion, so that packet loss rates are significantly reduced. 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.

Algorithm 1

```
for all Topic t ∈ Buffer do
  sortedContacts ← currentContacts sorted by CaféUtil_D ∗ length(Topic)
  for all Contact C_i ∈ sortedContacts do
    if NCRate(X, C_i) == 1 and length(Topic) ≥ NCLimit then
      send NCCode(Topic(C_i))
    else
      send Topic(C_i)
    end if
  end for
end for
```

Figure 7.1: CafNC Message Transfer Algorithm

Figure 7.1 gives an overview of the pseudo code for our CafNC that works as follows: For each topic in the node buffer, the node scans for neighbouring nodess interests and calculates their respective relative $CaféUtil_D$. Each neighbouring nodes $CaféUtil_D$ is weighted by the number of spaces for the respective topic that node has. The sending node then sorts the weighted neighbouring nodes utilities so that the node with the largest weight appears first in the list. The list is then traversed and for each member the number of packets for the respective topic is compared to the predefined network coding limit (*NCLimit*). If the number of packets for the topic exceeds the threshold, then the node network codes the messages in the topic in groups of *NCLimit* and sends its coded messages to the corresponding neighbouring node, if there are not enough messages in the topic set to code CafNC forwards the messages without coding them.

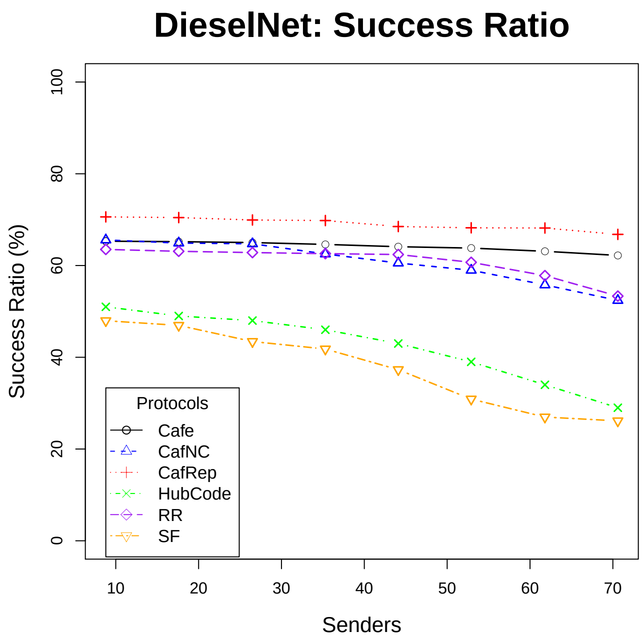
$$X_{n_i}(t) = \sum_{i=1}^{i=M} g_i(t)P_i + \sum_{j=1}^{j=N} h_j(t)Q_i \tag{7.2}$$

Formula 7.2, taken from [42], shows that messages are encoded at time t at node n_i by combining the packets received (P_1,...,P_M) and the messages within its buffer (Q_1,...,Q_N) with the coefficients $g_i(t)$ and $h_j(t)$ respectively. The sending node generates a random vector and employs it to do a linear combination of the packets cached that are targeted to the same destination. The destination nodes will collect packets which are linearly independent and as soon as the number of these packets reaches a certain number, the destination node will decode them and deliver them to the upper layer.

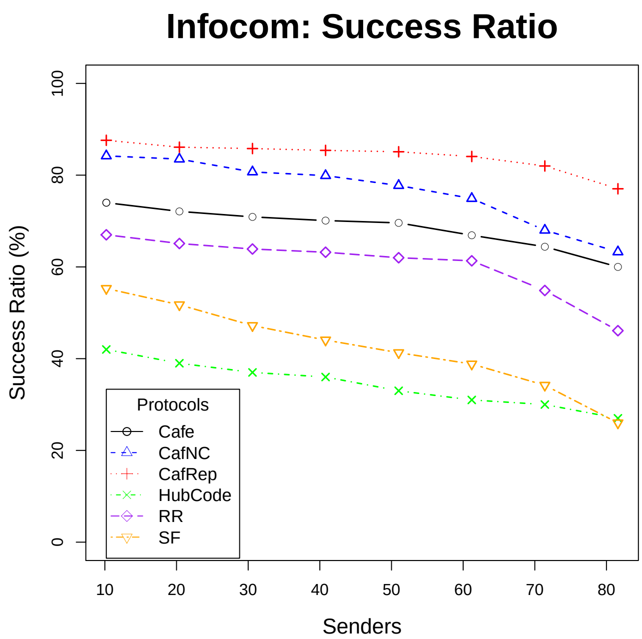
We perform an extensive evaluation of CafNC in comparison to five protocols an adaptive single copy forwarding algorithm (Café), adaptive multiple-copy forwarding algorithms (CafRep, Retiring Replicas [81]), a non-adaptive multi-copy forwarding algorithm (Spray and Focus [78]) and a static network coding algorithm (HubCode [2]), over multiple criteria using two vastly different connectivity datasets, Infocom 2006 [73] and DieselNet [8], from the CRAWDAD wireless data archive and we simulate traffic similarly to Chapter 5.

Figure 7.2 shows in both scenarios that CafNC can dramatically improve the success ratio of congested DTNs, in comparison to HubCode. When the congestion levels are low (the number of flows is small) the coding opportunities are few, and both coding schemes perform similarly to no coding, with HubCode performing worse than no coding when congestion levels are low over the infocom dataset.

We observe that in Figure 7.2 (a) SF and HubCode have the lowest success ratios, CafNC and RR perform similarly with around a 20% improvement in comparison with SF and HubCode. Café performs similarly to CafNC when congestion is low, but better maintains its success ratio as congestion increases. CafRep has the highest success ratio, with only between 7% and 10% difference to CafNC in low and high levels of congestion respectively.



(a) DieselNet



(b) Infocom 2006

Figure 7.2: Success Ratio

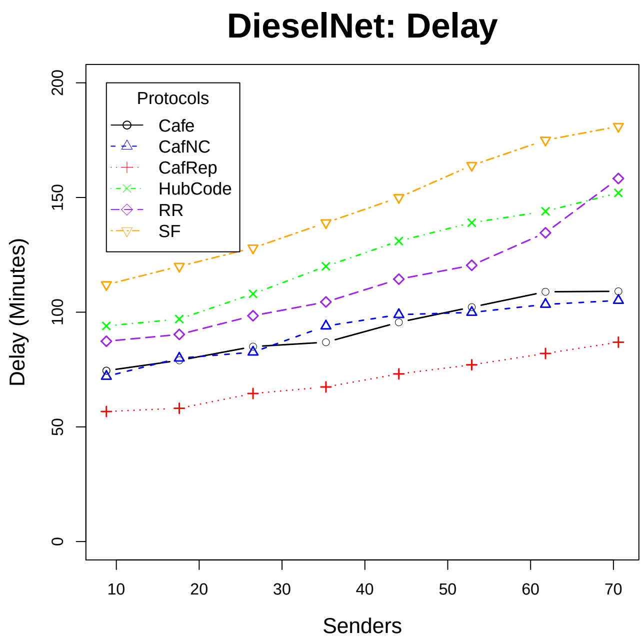
Figure 7.2 (b) shows that using Café to manage network coding, similarly to replication, has a positive impact on the success ratio of messages in social scenarios, with CafNC performing better than Café, RR, SF and HubCode over the Infocom dataset. In the social setting HubCode has the weakest performance.

As congestion increases, Café chooses more hops as it goes in a round-about way to the destination, avoiding congested nodes and regions. Figure 7.3 shows that despite the increase in path length, CafNC experiences substantially lower delays than no coding (Café) and state-of-the-art network coding (HubCode). In the social scenario there is little difference in delay between CafNC and CafRep, as they outperform all other algorithms.

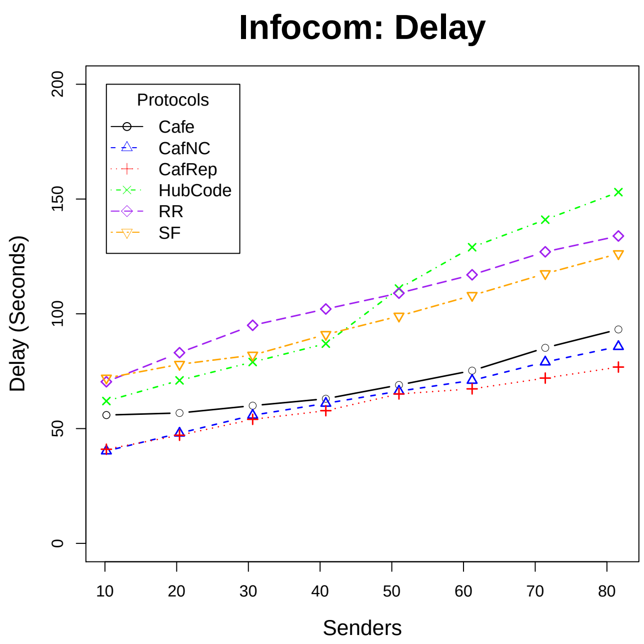
In Figure 7.4 it is interesting to see that CafNC has lower placket loss than HubCode, as this shows that as congestion increases predicting node and network region resource levels and content interest levels substantially decreases packet loss. In low levels of traffic, within the social connectivity setting we also observe that packet loss is lower for CafNC than any other method of forwarding and that as the level of congestion increases the packet loss is in line with the single copy uncoded approach.

Figure 7.5 shows the percentage of time and nodes that perform network coding for adaptive and hub-based static network coding. We observe that HubCode misses many coding opportunities when congestion is low and codes at a similar rate when the risk of packet loss is elevated. HubCode selects 10% of nodes to code all the time, but as a result HubCode misses the other opportunities to save on transfer bandwidth. Also, at times when the elected hub are congested, HubCode has significantly higher packet loss, which shows that persistently statically coding at a fixed set of nodes is a costly strategy.

Our CafNC outperforms adaptive single copy forwarding (Café), adaptive multiple-copy forwarding (RR), non-adaptive multi-copy forwarding (SF) and static hub-based network coding (HubCode) across a range of metrics, over two vastly different connectivity datasets. Our results show that as congestion increases (number of flows increases), without the amount of coding becoming adaptive, the performance deteriorates quickly because of the high level of

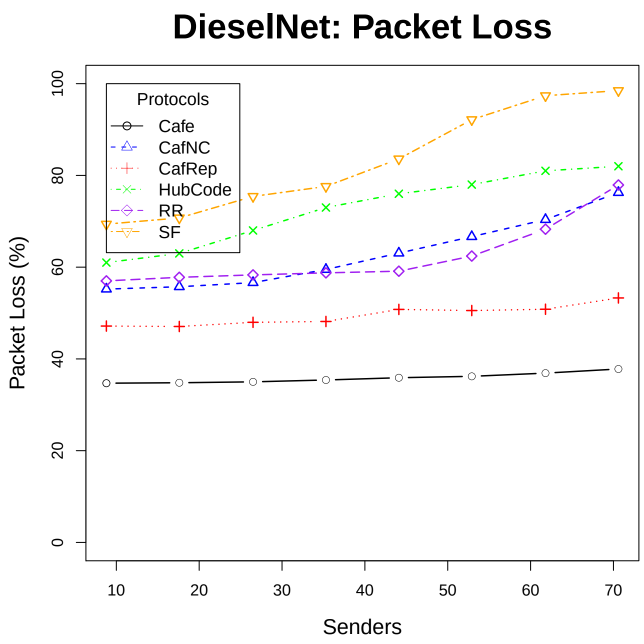


(a) DieselNet

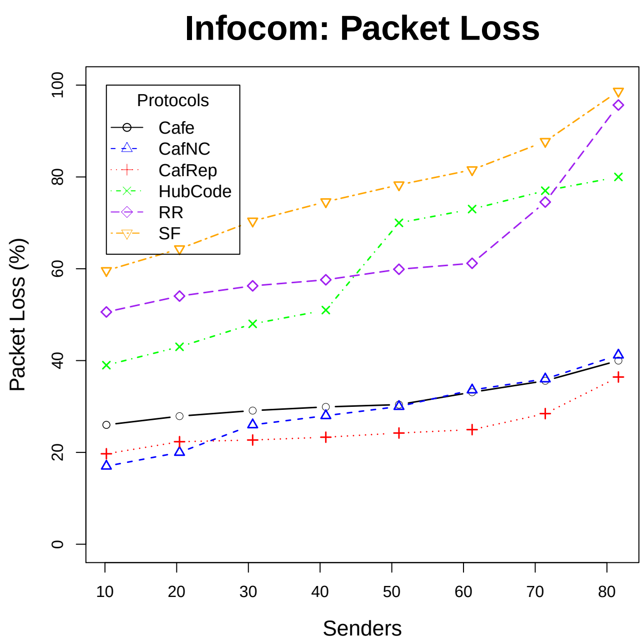


(b) Infocom 2006

Figure 7.3: Delay

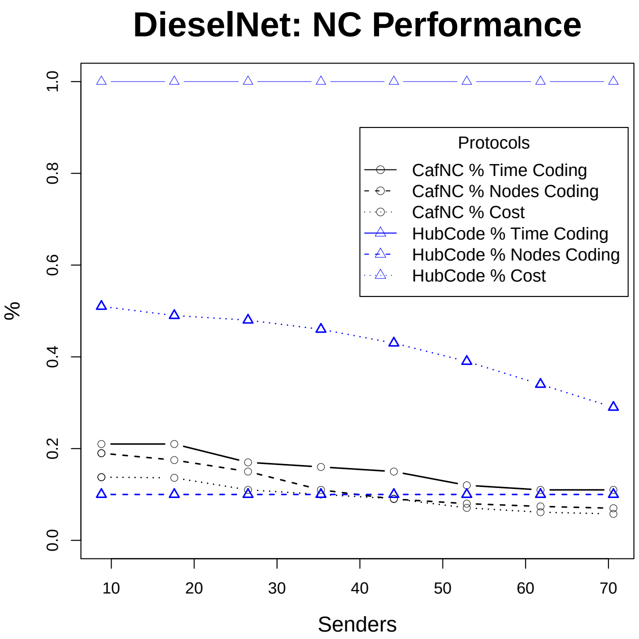


(a) DieselNet

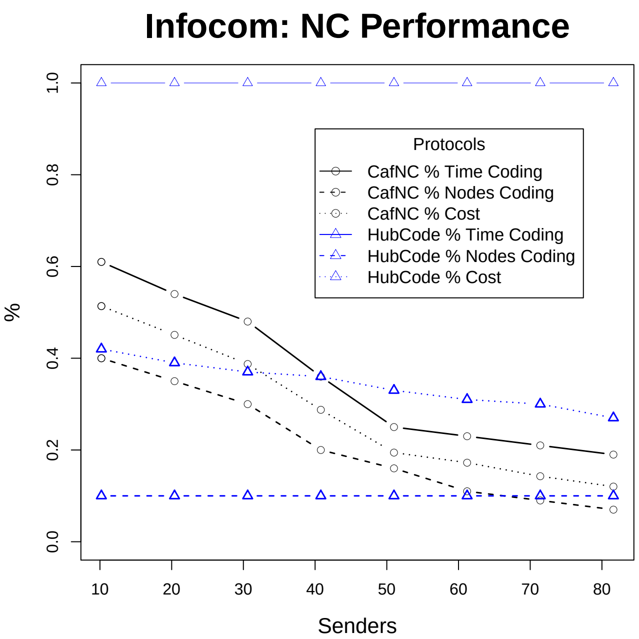


(b) Infocom 2006

Figure 7.4: Packet Loss



(a) DieselNet



(b) Infocom 2006

Figure 7.5: Network Coding Performance

contention in the network. In contrast, with the right amount of coding, the number of transmissions reduces for the same amount of data, resulting in lower congestion and consequently better performance.

7.2 Energy Efficiency

Unrealistic assumptions as regards energy constraints, similarly to the obsolete assumptions regarding unlimited bandwidth and storage capacity, need to be withdrawn. Recent work [7, 64] has shown that energy is the resource with the slowest rate of growth in improvements, and that the improvements mostly address form factor and not the longevity, also that local radio services such as WiFi consume much lower levels of power than services such as 3G. [64] show that the battery energy has had the lowest increases in performance in comparison with other mobile computing technology (as illustrated in Figure 7.6). [64] emphasise that as electronics have become smaller, more economical and less power consuming, batteries have also become smaller and more economical, but this has limited their capability, as energy density has progressed along flattening S-curves. [7] identifies that WiFi power consumption grows nearly three times slower compared to the cellular networks and for a 10K download, WiFi consumes one-sixth of 3Gs energy and one-third of GSMs energy, WiFi is still more energy efficient than 3G even when the cost of scanning and association is included, as illustrated in Figure 7.7.

Work such as [40, 55] are concerned with reducing the power used by the scanning and association process, which is required by radio devices in order to detect new encounters. Power is saved by sleeping the scanning process for periods of time, allowing the scanning to be performed at intervals rather than constantly. This method of periodical scanning is known as duty cycling. [40] is a study which explores the benefits of having two different radio interfaces, one a short range, low power interface and the other a longer ranger interface with a larger rate of power consumption. This allows the device to have a long duty cycle on the high power interface and a short duty cycle on the low

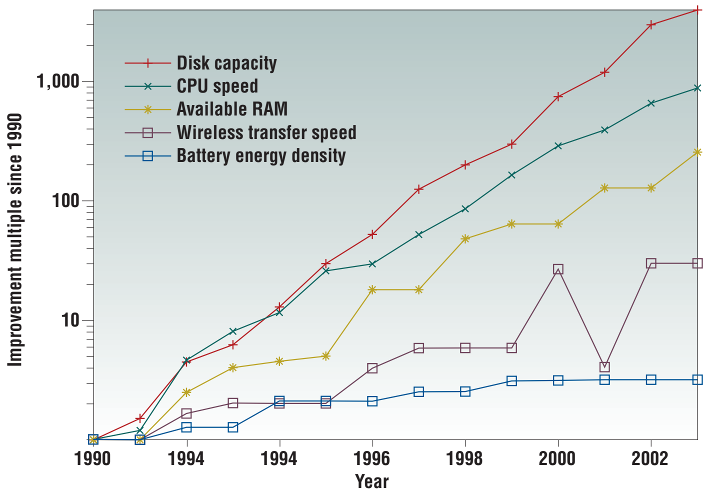


Figure 7.6: A graph, which featured in [64], that illustrates the rate of improvements in laptop computing technology from 1990 to 2003.

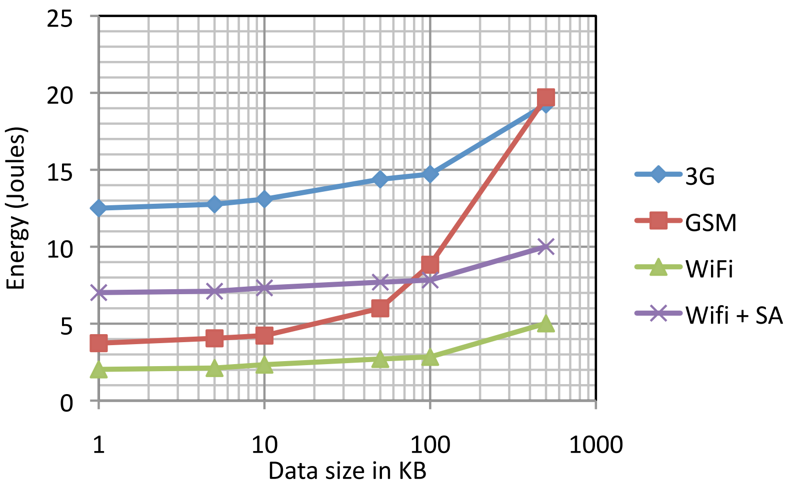


Figure 7.7: A graph, which featured in [7], showing the average energy consumption of WiFi, 3G and GSM for a range file sizes

power interface. The reason the longer range is able to have a longer distance between scans is because the additional signal range allows the nodes to be connected for a greater encounter duration. The results of the paper shows that by using the paired radio duty cycling energy saving can be substantial and the impact on contact discovery is substantially less than duty cycling with short range or long range independently. [55] is concerned with logging encounters between grey seals, they improve their contact detection mechanism by assuming a contact is a part of the same encounter unless the encountered node has not communicated for a period of time greater than 3 beacon intervals. The benefit of this is that it reduces the cost of reestablishing fragmented communication, as nodes are required to exchange summary messages upon a new encounter, by acknowledging that beacon messages can be lost and therefore not reacting to short breaks in communication the number of summary exchanges can be greatly reduced, which results in both a reduction in energy consumption and improved social statistics.

Existing work is concerned with the evaluation of available power resources and the cost associated with detecting and transmitting data from a low level perspective. Energy efficiency at the routing level is a complex criteria, as efficiency can be viewed from more than one perspective. Energy efficiency can be viewed as minimising the energy used by selecting one path over another, this can be due to the number of hops if the transmission media is homogeneous or the sum of transfer costs along the path if the transmission media is heterogeneous, for example one hop over WiFi costs less than transferring the data via 3G. The energy a node spends on behalf of the network in comparison with other nodes, is not only an issue of fairness, but of efficiency too, as the cost of routing through a disconnected network would become progressively more expensive, and therefore less efficient, if nodes in the network are exploited to the point where they have no remaining power, thus fragmenting the network further, which could raise the cost of a store-carry-and-forward transfer making it either undesirable or impossible.

Our work is concerned with mixed and varied networking environments, which cause path selection to be extemporaneous, which means comparing alternative paths is unrealistic. Predicting the amount of remaining battery would allow us to forward towards nodes that had greater available resources, but this solution is unsatisfactory as typically the predicted amount of remaining battery is often a poor approximation, and by forward towards nodes that have greater remaining battery life allows nodes to greedily spend their energy, allowing them to avoid collaborating. Energy fairness could be achieved, similarly to the way Café distributes storage and transfer resources, by monitoring and disseminating the amount of energy a node has consumed, and by aggregating these values to form an ego-network value, we would be able to evaluate a next hop based upon the amount of energy each node and region has spent on forwarding. It might be appropriate to segregate the energy consumption used for a nodes own gain and that spent on behalf of the network, as monitoring energy spent as a forwarding contribution rather than for self gain would prevent nodes that are not spending large amounts of energy forwarding their own traffic bearing an unfair proportion of the cost of forwarding as an intermediary. By forwarding messages towards nodes that have contributed less, as regards power resources, to the forwarding of messages, we are able to avoid depleting the power resources of highly central devices. We will also have to consider the fact that nodes will have different energy capacities and that some nodes may not be limited by energy at all (e.g. whilst charging or desktop machines).

Chapter 8

Conclusion

This Chapter provides a critical evaluation of contributions that have been presented throughout this Thesis (Section 8.1) and proposes future research directions for content delivery in DTNs (Section 8.2).

8.1 Summary of Contributions

This Thesis was motivated by the observation that the typical underlying assumptions regarding storage and transfer restraints used in the evaluation of DTN forwarding strategies was not representative of realistic mobile networks. More specifically, this Thesis was concerned with providing a framework for enabling devices to perform congestion aware operations that results in nodes: distributing traffic across multiple paths, make regionally aware forwarding decisions, allocating an appropriate number of copies to next-hop nodes, and limiting the total number of copies of a message disseminated, such that available resources are not over or under utilised.

We proposed Café, a congestion-aware framework for single copy forwarding, and CafRep, a congestion-aware framework for multi-copy forwarding. In order to distribute traffic across multiple paths we incorporated the monitoring and dissemination of congestion observations, which enables nodes to consider the level of congestion when evaluating the next-hop utility of encountered nodes, which results in nodes with slightly lower forwarding utility, but higher resource availability being selected when the optimal route is congesting. This work was published in the Seventh Annual Conference on Wireless On demand Network Systems and Services (WONS 2010) [29]. We introduce the concept of implicit clusters, which are formed by nodes aggregating the observations received from encountered nodes in a meaningful way, such that it reflects the network environment in which the node is typically situated. These aggregated observations are then disseminated in order to allow forwarding nodes to make regionally aware forwarding decisions. This work was published in IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2010) [30]. In order for CafRep to allocate an appropriate number of copies to an elected next-hop we have incorporated a replication placement mechanism that ensures copies of a message are adaptively allocated to contacts, ensuring that more copies go towards nodes with high forwarding utility and low congestion levels, and the number of packets is adaptively reduced if the node is congesting or has a lower forwarding utility. This work was published in the Eighth Annual Conference on Wireless On demand Network Systems and Services (WONS 2011)[69]. CafRep also incorporates our replication copy management mechanism, which adaptively adjusts the number of copies of a message a node can disseminate based upon the level of congestion, allowing a node to make use of available resources and back off in oder to preserve resources when they are limited. This work was published in the 7th International Wireless Communications and Mobile Computing Conference (IWCMC 2011) [70]. We have evaluated the contributions of this Thesis by emulating real network conditions, incorporating device connectivity traces from a number of possible deployment scenarios, both real world and pseudo real world [68] traffic models and experimentally evaluating a range of buffer levels. Our real world traffic model incorporates measurements of message size, post frequency, node interest similarities and friendship topology of actual social network application users. We have considered a number of performance metrics both in order to evaluate the benefits of congestion specific attributes, such as available buffer and delay, but also across forwarding performance metrics, such as success ratio and total delivered messages, which have shown that in addition to our framework providing a method of congestion aware routing, our solution provides improved forwarding performance as a result of increasing the utilisation of the available network resources.

8.2 Future Work

In Chapter 7 we introduce the preliminary proposals for controlling the rate of network coding (Section 7.1) and our views regarding Café's ability to provide a method for energy efficiency, by considering energy constraints as a component of the next-hop selection process (Section 7.2), but these supplementary applications are not concerned with enhancing Café itself. This Section is concerned with the future directions in which our work can be evaluated and enhanced.

8.2.1 Robustness

Our framework, along with the majority of other DTN protocols, assumes that nodes will benevolently follow the prescribed algorithm, but the reality is that in a competitive environment it is inevitably that attempts to deceive the network will occur if some form of personal gain can be achieved. The most apparent attack for a node resisting being elected as an intermediary when nodes are using our framework would be to disseminate unfavourable congestion observations. This is not simply a problem that is constrained to our framework, as disseminating information which would indicate poor forwarding ability would obstruct most forwarding algorithms. This form of selfish behaviour is often referred to as ”Free Riding” and has been observed within the Internet [3, 26], MANETs [58] and peer-to-peer systems [47].

In Algorithmic Game Theory [61] the price of stability [5] and the price of anarchy [63] are measurements of the performance loss caused by self-interest, calculated as the difference between a distributed system stable state (Nash equilibrium) and the optimal state achievable by a central authority or by all nodes unconditionally cooperating. The price of anarchy is the ratio between the worst case Nash equilibrium and the optimal state, while the price of stability is a measurement between the best Nash equilibrium and the optimal system state. The price of stability is best suited for measuring systems in which a degree of control can be enforced, while the price of anarchy gives the best performance evaluation when networking agents are completely unregulated. It is likely that delay tolerant systems are evaluated from the price of anarchy perspective, whereby DTN systems are developed to be resilient to loss and evaluated for worst case performance and developed to be as close to the system optimal as possible without regulation.

8.2.2 Explicit Clustering

In Section 2.3.4 we review clustering schemes developed for MANETs and DTNs, which divide nodes into different virtual groups that are used in order to synthesise a hierarchical structure similarly to that exhibited by the Internet, in order to structure communications, which allows the networks to scale to large numbers, reduces packet collision and, in MANETs, provide throughput and delay performance guarantees.

In Section 3.5.4 we describe how we provide collaboration by means of implicitly clustering nodes, which allows forwarding decisions to be less selfish through sharing aggregated ego-network observations. Our solution provides heuristics that describe network regions, but it does not explicitly define regions. In order to coordinate transmissions, prevent collisions and boost efficiency in dense regions, nodes must explicitly form clusters.

Low maintenance clustering techniques, such as passive clustering (PC) [28] in MANETs, only build clusters in the event of multiple nodes congregating in one area. PC does not solve the problem, as it is not tailored to DTNs, as such the algorithm makes decisions that are not necessarily best suited to DTNs. For example, PC elects the node that instigates communication as the cluster head, within a DTN it is probable that a node which is highly mobile would instigate a large number of connections, but these would only last for a short period of time. Because some nodes instigate more connections than others, they would have a high probability of becoming a cluster head. Because PC operates a first come first elected procedure (where the first node to send the clustering message becomes the cluster head), it is quite likely that this would result in clusters frequently electing ineffective cluster heads. This could be improved by nodes electing a cluster head based upon a local heuristic that identifies it as a stable node, better suited than other nodes for being the cluster head.

Bibliography

[1] S. Ahmed and S.S. Kanhere. Cluster-based forwarding in delay tolerant public transport networks. In *Local Computer Networks, 2007. LCN 2007. 32nd IEEE Conference on*, pages 625–634. IEEE, 2007.

[2] S. Ahmed and S.S. Kanhere. Hubcode: message forwarding using hub-based network coding in delay tolerant networks. In *the 12th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pages 288–296. ACM, 2009.

[3] A. Akella, S. Seshan, R. Karp, S. Shenker, and C. Papadimitriou. Selfish behavior and stability of the internet:: a game-theoretic analysis of TCP. In *the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 117–130. ACM, 2002.

[4] M. Allman, V. Paxson, and W. Stevens. Rfc2581: TCP congestion control. *Internet RFCs*, 1999.

[5] E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. In *the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 295–304. IEEE, 2004.

[6] A. Balasubramanian, B. Levine, and A. Venkataramani. DTN routing as a resource allocation problem. In *ACM SIGCOMM Computer Communication Review*, volume 37, pages 373–384. ACM, 2007.

[7] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani. Energy consumption in mobile phones: a measurement study and implications for network applications. In *the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 280–293. ACM, 2009.

[8] 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.

[9] 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.

[10] R. Bhagwan, K. Tati, Y.C. Cheng, S. Savage, and G.M. Voelker. otal recall: System support for automated availability management. In *NSDI*, 2004.

[11] U. Brandes and T. Erlebach. *Network analysis: methodological foundations*, volume 3418. Springer Verlag, 2005.

[12] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss. Delay-tolerant networking: an approach to interplanetary internet. *Communications Magazine, IEEE*, 41(6):128–136, 2003.

[13] S. Burleigh, E. Jennings, and J. Schoolcraft. Autonomous congestion control in delay-tolerant networks. *SpaceOps*, 2006.

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

[15] M. Chuah and P. Yang. Impact of selective dropping attacks on network coding performance in DTNs and a potential mitigation scheme. In *Computer Communications and Networks*, pages 1–6. IEEE, 2009.

[16] S. Corson and J. Macker. Rfc2501: Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations. *Internet RFCs*, 1999.

[17] 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.

[18] 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.

[19] H. Dang and H. Wu. Clustering and cluster-based routing protocol for delay-tolerant mobile networks. *IEEE Transactions on Wireless Communications*, 9(6):1874–1881, 2010.

[20] M. Demmer and K. Fall. DTLSR: delay tolerant routing for developing regions. In *the 2007 workshop on Networked systems for developing regions*, pages 1–6. ACM, 2007.

[21] V. Erramilli, A. Chaintreau, M. Crovella, and C. Diot. Diversity of forwarding paths in pocket switched networks. In *the 7th ACM SIGCOMM conference on Internet measurement*, pages 161–174. ACM, 2007.

[22] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot. Delegation forwarding. In *the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 251–260. ACM, 2008.

[23] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin. Diversity in smartphone usage. In *the 8th international conference on Mobile systems, applications, and services*, pages 179–194. ACM, 2010.

[24] K. Fall. A delay-tolerant network architecture for challenged internets. In *the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27–34. ACM, 2003.

[25] S. Floyd and K. Fall. Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Transactions on Networking (TON)*, 7(4):458–472, 1999.

[26] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-based congestion control for unicast applications. *ACM SIGCOMM Computer Communication Review*, 30(4):43–56, 2000.

[27] S. Floyd and T. Henderson. The NewReno modification to TCP's fast recovery algorithm. 1999.

[28] M. Gerla, T.J. Kwon, and G. Pei. On-demand routing in large ad hoc wireless networks with passive clustering. In *Wireless Communications and Networking Conference, 2000. WCNC. 2000 IEEE*, volume 1, pages 100–105. IEEE, 2002.

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

[30] 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.

[31] W. Hsu and A. Helmy. IMPACT: Investigation of mobile-user patterns across university campuses using wlan trace analysis. *CoRR*, abs/cs/0508009, 2005.

[32] D. Hua, X. Du, Y. Qian, and S. Yan. A DTN Routing Protocol Based on Hierarchy Forwarding and Cluster Control. In *2009 International Conference on Computational Intelligence and Security*, pages 397–401. IEEE, 2009.

[33] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot. Pocket switched networks and human mobility in conference environments. pages 244–251, 2005.

[34] P. Hui and J. Crowcroft. How small labels create big improvements. In *the 2006 ACM CoNEXT conference*. ACM New York, NY, USA, 2006.

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

[36] V. Jacobson. Congestion avoidance and control. In *ACM SIGCOMM Computer Communication Review*, volume 18, pages 314–329. ACM, 1988.

[37] Y. Jiao, Z. Jin, and Y. Shu. Data Dissemination in Delay and Disruption Tolerant Networks Based on Content Classification. In *the 5th International Conference on Mobile Ad-hoc and Sensor Networks (MSN'09).*, pages 366–370. IEEE, 2009.

[38] D.B. Johnson and D.A. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile computing*, pages 153–181, 1996.

[39] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.S. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet. *COMPUTER ARCHITECTURE NEWS*, 30(5):96–107, 2002.

[40] H. Jun, M.H. Ammar, M.D. Corner, and E.W. Zegura. Hierarchical power management in disruption tolerant networks with traffic-aware optimization. In *the 2006 SIGCOMM workshop on Challenged networks*, pages 245–252. ACM, 2006.

[41] S. Jung, U. Lee, A. Chang, D.K. Cho, and M. Gerla. Bluetorrent: Cooperative content sharing for Bluetooth users. *Pervasive and Mobile Computing*, 3(6):609–634, 2007.

[42] G. Karbaschi, A.C. Viana, S. Martin, and K. Al Agha. On using network coding in multi hop wireless networks. In *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*, pages 420–424. IEEE.

[43] A. Keranen, J. Ott, and T. Karkkainen. The ONE simulator for DTN protocol evaluation. In *the 2nd International Conference on Simulation*

Tools and Techniques, pages 1–10. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[44] E. Kohler, M. Handley, and S. Floyd. Designing dccp: Congestion control without reliability. *ACM SIGCOMM Computer Communication Review*, 36(4):27–38, 2006.

[45] D. Kotz and T. Henderson. Crawdad: A community resource for archiving wireless data at dartmouth. *IEEE Pervasive Computing*, pages 12–14, 2005.

[46] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. *Computer science review*, 3(2):65–69, 2009.

[47] K. Lai, M. Feldman, I. Stoica, and J. Chuang. Incentives for cooperation in peer-to-peer networks. In *Workshop on economics of peer-to-peer systems*. Citeseer, 2003.

[48] S.J. Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. In *IEEE International Conference on Communications (ICC)*, volume 10, pages 3201–3205. IEEE, 2001.

[49] 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.

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

[51] H. Lim, K. Xu, and M. Gerla. TCP performance over multipath routing in mobile ad hoc networks. In *Communications, 2003. ICC'03. IEEE International Conference on*, volume 2, pages 1064–1068. IEEE, 2003.

[52] Y. Lin, B. Li, and B. Liang. Efficient network coded data transmissions in disruption tolerant networks. In *IEEE Infocom*, pages 1508–1516. IEEE, 2008.

[53] Y. Lin, B. Liang, and B. Li. Performance modeling of network coding in epidemic routing. In *the 1st international MobiSys workshop on Mobile opportunistic networking*, pages 67–74. ACM, 2007.

[54] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. *Lecture Notes in Computer Science*, pages 239–254, 2004.

[55] A. Lindgren, C. Mascolo, M. Lonergan, and B. McConnell. Seal-2-seal: A delay-tolerant protocol for contact logging in wildlife monitoring sensor networks. In *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on*, pages 321–327. IEEE, 2008.

[56] A. Lindgren and K.S. Phanse. Evaluation of queuing policies and forwarding strategies for routing in intermittently connected networks. pages 1–10, 2006.

[57] C. Lochert, B. Scheuermann, and M. Mauve. A survey on congestion control for mobile ad hoc networks. *Wireless Communications and Mobile Computing*, 7(5):655–676, 2007.

[58] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Sustaining cooperation in multi-hop wireless networks. In *the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*, pages 231–244. USENIX Association, 2005.

[59] M. May, V. Lenders, G. Karlsson, and C. Wacha. Wireless opportunistic podcasting: implementation and design tradeoffs. In *the second ACM workshop on Challenged networks*, pages 75–82. ACM, 2007.

[60] 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.

[61] N. Nisan. *Algorithmic game theory.* Cambridge University Press, 2007.

[62] 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.

[63] C. Papadimitriou. Algorithms, games, and the internet. In *the thirty-third annual ACM symposium on Theory of computing*, pages 749–753. ACM, 2001.

[64] J.A. Paradiso and T. Starner. Energy scavenging for mobile and wireless electronics. *Pervasive Computing, IEEE*, 4(1):18–27, 2005.

[65] A. Pentland, R. Fletcher, and A. Hasson. Daknet: Rethinking connectivity in developing nations. *Computer*, 37(1):78–83, 2004.

[66] J.M. Pujol, A.L. Toledo, and P. Rodriguez. Fair routing in delay tolerant networks. In *IEEE Infocom*, pages 837–845. IEEE, 2009.

[67] 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.

[68] 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.

[69] 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.

[70] M. Radenkovic and A. Grundy. Framework for utility driven congestion control in delay tolerant opportunistic networks. In *the 7th International Wireless Communications and Mobile Computing Conference (IWCMC 2011)*, Istanbul, Turkey, July 2011.

[71] T. Roughgarden. *Selfish routing and the price of anarchy.* The MIT Press, 2005.

[72] 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.

[73] 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.

[74] M. Seligman, K. Fall, and P. Mundur. Storage routing for DTN congestion control. *Wireless Communications and Mobile Computing*, 7(10):1183–1196, 2007.

[75] A.N. Shiryaev, B. Rozovskii, and G. Grimmett. *Optimal Stopping Rules*. Springer-Verlag Berlin Heidelberg, 2008.

[76] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Single-copy routing in intermittently connected mobile networks. In *Sensor and Ad Hoc Communications and Networks*, pages 235–244. IEEE, 2004.

[77] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259. ACM, 2005.

[78] 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.

[79] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Efficient routing in intermittently connected mobile networks: the single-copy case. *IEEE/ACM Transactions on Networking*, 16(1):63–76, 2008.

[80] K. Sundaresan, V. Anantharaman, H.Y. Hsieh, and R. Sivakumar. ATP: a reliable transport protocol for ad-hoc networks. In *the 4th ACM interna-*

tional symposium on Mobile ad hoc networking & computing, pages 64–75. ACM New York, NY, USA, 2003.

[81] N. Thompson, S. Nelson, M. Bakht, T. Abdelzaher, and R. Kravets. Retiring replicants: Congestion control for intermittently-connected networks. In *IEEE Infocom*, 2010.

[82] 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 *IEEE Infocom*. Citeseer, 2009.

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

[84] M. Valafar, R. Rejaie, and W. Willinger. Beyond friendship graphs: a study of user interactions in Flickr. In *the 2nd ACM workshop on Online social networks*, pages 25–30. ACM, 2009.

[85] J.G. Wardrop. Some theoretical aspects of road traffic research. *OR*, 4(4):72–73, 1953.

[86] J. Widmer and J.Y. Le Boudec. Network coding for efficient communication in extreme networks. In *the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 284–291. ACM, 2005.

[87] D. Wischik, M. Handley, and M.B. Braun. The resource pooling principle. *ACM SIGCOMM Computer Communication Review*, 38(5):47–52, 2008.

[88] E. Yoneki, P. Hui, S.Y. Chan, and J. Crowcroft. A socio-aware overlay for publish/subscribe communication in delay tolerant networks. In *the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, page 234. ACM, 2007.

[89] X. Zhang, G. Neglia, J. Kurose, and D. Towsley. On the benefits of random linear coding for unicast applications in disruption tolerant networks. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*, pages 1–7. IEEE, 2006.