

Multilayer Content Services in Heterogeneous Mobile Opportunistic Networks

Vu San Ha Huynh

Thesis submitted to the University of Nottingham

for the degree of Doctor of Philosophy

February 2021

Abstract

Today we are living in the world where networks are becoming significantly more complex and dynamic while the applications are increasingly diverse as the results from the emerging of smart, ubiquitous devices embedded in our daily activities. Applications and services hosted in these smart mobile edges (e.g., augmented reality, self-driving, and various cognitive applications) may suffer from limited network coverage and localized congestion due to dynamic mobility of users and surge of traffic demand. This thesis is concerned with supporting the development of new distributed technologies which are able to support dynamic peer-to-peer content sharing, decrease high operating costs, and handle intermittent disconnections.

This thesis proposes CafRepCache, a multilayer collaborative adaptive and predictive caching framework in mobile opportunistic networks. CafRepCache is built on the principle of predictive analytics and heuristics to answer the fundamental multi-attribute decision problem: selecting which nodes represent the best carriers for given contents and deciding which contents to be cached to bring contents as close as possible to the subscribers in order to minimise the end-to-end latency while trying to balance the trade-offs between maximising content delivery while avoiding network congestion and keeping high cache efficiency. Our framework is a fully-distributed and adaptive algorithm that can capture, analyse and predict in real time the spatial-temporal locality of content requests in the dynamic changing connectivity patterns, dynamic mobility of publishers/subscribers and dynamic resources.

We extensively evaluate our work using realistic and pseudo-realistic applications and connectivity traces showing that, independent of the network connectivity/mobility and content traffic patterns, our framework outperforms a number of benchmarking and state-of-the-art caching algorithms. Our results show that CafRepCache is able to choose the best suitable set of contents to be cached in the best suitable set of nodes. CafRepCache brings contents closer to the subscribers, minimize the latency of content discovery and content retrieval, keep high cache efficiency and maximise success ratio of content retrieval while avoiding network congestion in heterogeneous mobile opportunistic networks.

Publications

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

- Radenkovic, M. and Huynh, V.S.H. “Low-cost mobile personal clouds”. in 2016 International Wireless Communications and Mobile Computing Conference (IWCMC). 2016. IEEE.
- Radenkovic, M. and Huynh, V.S.H. “Collaborative cognitive content dissemination and query in heterogeneous mobile opportunistic networks”. in Proceedings of the 3rd Workshop on Experiences with the Design and Implementation of Smart Objects. 2017. ACM.
- Huynh, V.S.H. and Radenkovic, M. “A novel cross-layer framework for large scale emergency communications”. in 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC). 2017. IEEE.
- Radenkovic, M., Huynh, V.S.H, and Manzoni, P. “Adaptive real-time predictive collaborative content discovery and retrieval in mobile disconnection prone networks”. IEEE Access, 2018. 6: p. 32188-32206. IEEE.
- Huynh, V.S.H., Radenkovic, M. and John, R. “Understanding information-centric layer of adaptive collaborative caching framework in mobile disconnection-prone networks”. in 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC). 2018. IEEE.
- Huynh, V.S.H. and Radenkovic, M. “Understanding complementary multi-layer collaborative heuristics for adaptive caching in heterogeneous mobile opportunistic networks”. in 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC). 2018. IEEE.
- Huynh, V.S.H. and Radenkovic, M. “Interdependent Multi-layer Spatial Temporal-based Caching in Heterogeneous Mobile Edge and Fog Networks”. in Proceedings of the 9th International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS). p. 34 – 45. 2019.
- Radenkovic, M., Huynh, V.S.H, John, R. and Manzoni, P. “Enabling Real-time Communications and Services in Heterogeneous Networks of Drones

and Vehicles”. in The 15th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). 2019.

- Radenkovic, M. and Huynh, V.S.H. “Energy-Aware Opportunistic Charging and Energy Distribution for Sustainable Vehicular Edge and Fog Networks”. in Proceedings of the 5th International Conference on Fog and Mobile Edge Computing (FMEC) p. 5-12. 2020. IEEE.
- Huynh, V.S.H. and Radenkovic, M. “Leveraging Social Behaviour of Users Mobility and Interests for Improving QoS and Energy Efficiency of Content Services in Mobile Community Edge-clouds”. in Proceedings of the 10th International Conference on Cloud Computing and Services Science (CLOSER). p. 382-389 2020.
- Radenkovic, M. and Huynh, V.S.H. “Cognitive Caching at the Edges for Mobile Social Community Networks: A Multi-Agent Deep Reinforcement Learning Approach”. IEEE Access. p. 179561 – 179574, 2020. IEEE.

Acknowledgement

I would like to deeply thank my supervisors, Dr. Milena Radenkovic, Prof. Robert John, Prof. Natasa Milic-Frayling for all their support and kindness during the course of my PhD. I would especially like to thank Dr. Milena Radenkovic for her invaluable input and feedback, which challenged me and encouraged me from the early stages of my PhD. Her patient guidance, encouragement and advice help me so much to accomplish my PhD thesis.

Many thanks to my friends and colleagues for their help, support, kindness and friendship over the years.

And, I would like to express my sincere love, gratitude and appreciation to my family for their loves, caring, understanding and sacrifices for me. They are always there to support, motivate and give me invaluable advice on everything. To my family, thank you for being a part of my wonderful journey.

Contents

1. Introduction.....	1
1.1 Background and Motivation	1
1.2 The Specific Focus of This Thesis.....	3
1.3 Envisaged Networks & Applications.....	5
1.4 Thesis Contributions.....	6
1.5 Thesis Structure	9
2. Literature Review	10
2.1 Criteria.....	10
2.1.1 Support for fully-distributed predictive decision making in real time with limited localised knowledge	11
2.1.2 Support for adaptability and responsiveness to disconnections due to dynamic mobility in dense and sparse topologies	12
2.1.3 Support for dynamic traffic demands from users acting as both mobile publishers and subscribers.....	12
2.1.4 Support for limited resources, network congestion controls and avoidance	13
2.1.5 Support for efficient and fair distribution of cache space across trusted collaborators.....	13
2.1.6 Support for multidimensional interdependences between multilayers of mobile networks: mobility, traffic patterns and resources	14
2.1.7 Support for maintaining acceptable success rate and low end-to-end latency	14
2.2 Techniques for efficient peer-to-peer content sharing.....	14
2.2.1 Web caching	15
2.2.2 Content distribution networks.....	15
2.2.3 Mobile offloading	16
2.2.4 Content caching in Information-Centric Networks (ICN) and Content- Centric Networks (CCN).....	17
2.2.5 Opportunistic Caching in Mobile Opportunistic Networks.....	18
2.2.6 Summary.....	20
2.3 Forwarding techniques to deal with disconnections and dynamic mobility	22
2.3.1 Non-Social based forwarding.....	22
2.3.2 Social based forwarding	23
2.3.3 Summary.....	24
2.4 Techniques to support network congestion and resources awareness	24
2.4.1 Congestion Control.....	25

2.4.2 Multi-path Routing	25
2.4.3 Peer-to-Peer Replication	26
2.4.4 Clustering techniques	26
2.4.5 Network Coding	27
2.4.6 Summary.....	27
2.5 In-network opportunistic caching techniques in mobile opportunistic networks.....	28
2.5.1 Utility-based Cache.....	29
2.5.2 Social Cache.....	29
2.5.3 Location Cache.....	30
2.5.4 Popularity-based Cache.....	31
2.5.5 Probabilistic Cache	32
2.5.6 Label Cache	32
2.5.7 Single Cache & Collaborative Cache	33
2.5.8 Summary.....	34
2.6 Techniques to support capturing spatial-temporal locality and multilayer interdependences in mobile opportunistic networks	37
2.7 Summary.....	38
3. Collaborative Cognitive Predictive Content Caching Framework	40
3.1 CafRepCache Foundation	42
3.2 CafRepCache's Multi-layer Multi-dimensional Design Space	43
3.3 CafRepCache's Multi-layer Framework	44
3.4 CafRepCache Analytical Model	48
3.4.1 CafRepCache collaborative caching as bargaining game	48
3.4.2 Complex graph theory for network and mobility	49
3.4.3 Contents	50
3.4.4 Mobile publishers and subscribers in mobile opportunistic networks.....	51
3.5 CafRepCache's Content Heuristics: Predicting Content Popularity By Leveraging The Spatial-Temporal Locality of Content Requests	52
3.5.1 Dynamic content request temporal heuristics	53
3.5.2 Dynamic content request spatial clustering heuristics	55
3.5.3 Collaborative relative utilities.....	57
3.5.4 Aggregated historical & present signals	58
3.5.5 Reputation-based weighting mechanism for collaborations	59
3.6 The CafRepCache Pseudocode	63
3.7 Summary.....	66
4. Implementation and Evaluation	67

4.1 Heterogeneous Network Topology Traces	71
4.1.1 Random Network Connectivity Traces.....	72
4.1.2 Real-world Social Network Connectivity Traces	73
4.1.3 Random vs. Scale-free Network Connectivity Traces	75
4.2 Heterogeneous Content Publish-Subscribe Traces	77
4.2.1 Pseudo-realistic Content Request Traces	77
4.2.2 Real-world Content Request traces.....	78
4.3 Forwarding Performance Evaluation	83
4.3.1 Success Ratio In Content Discovery And Retrieval.....	84
4.3.2 Delay In Content Discovery And Retrieval	93
4.3.3 Packet Loss In Content Discovery And Retrieval.....	100
4.4 Caching Performance Evaluation	106
4.4.1 Cache Hit Ratio.....	107
4.4.2 Eviction Rate	110
4.4.3 Number of Hops from Caching Points to Subscribers	112
4.4.4 Footprint Reduction.....	113
4.5 Network Resource Utilisation Evaluation	114
4.5.1 Number of Caching Points	114
4.5.2 Computational Costs (CPU Usage)	115
4.5.3 Communication Costs (Number of packets forwarded).....	117
4.6 Summary	120
5. Discussion	121
5.1 Network Coding and Energy Efficiency	121
5.2 Early Deployment of MoDiToNeS Platform in Heterogeneous Mobile Opportunistic Networks	126
5.3 CafRepCache Large-Scale Real-World Application Domains	134
5.4 Summary.....	138
6. Conclusion	139
6.1 Summary of Contributions	139
6.2 Future Work.....	143
6.2.1 Artificial Intelligence and Machine Learning for Offline Analysis...	143
6.2.2 Fairness and Incentiveness.....	143
6.2.3 Dynamic Scalability with Software-Defined Networking (SDN) and Network Functions Virtualization (NFV)	144
6.2.4 Security and Privacy.....	144
Bibliography	146

List of Figures

Figure 3.1 Content Retrieval: Content Discovery and Content Delivery.	41
Figure 3.2 CafRepCache's theory foundation.....	43
Figure 3.3 CafRepCache's design space.....	44
Figure 3.4 Multi-layer CafRepCache framework	45
Figure 3.5 Input and output of CafRepCache caching framework.	46
Figure 3.6 CafRepCache's integral components.....	47
Figure 3.7 Reputation system model to deal with weighting problem in collaboration process.	60
Figure 4.1 Experimental Methodology.	70
Figure 4.2 Random Networks vs. Scale-free Networks.....	72
Figure 4.3a Contact Duration.....	74
Figure 4.3b Isolation Duration.....	74
Figure 4.3c Number of Contacts.....	75
Figure 4.4a Success ratio vs. number of subscribers.	76
Figure 4.4b Average hops count between caching points and subscribers.	76
Figure 4.5a Content distribution.	78
Figure 4.5b Content size proportion.	78
Figure 4.6 Publisher/Caching Point (Green) – Subscriber (Red) relationship.....	79
Figure 4.7a Spatial-temporal correlation of content requests in Foursquare dataset. .	80
Figure 4.7b Temporal locality of content requests in Foursquare dataset.	81
Figure 4.7c Spatial locality of content requests in Foursquare dataset.	82
Figure 4.8a Content retrieval success ratio vs. Number of subscribers (SF Cab).....	86
Figure 4.8b Content retrieval success ratio vs. Number of subscribers (Infocom).	87
Figure 4.8c Content retrieval success ratio vs. Number of subscribers (RollerNet)...	87
Figure 4.9a Content retrieval success ratio vs. Content popularity skewness (SF Cab).	89
Figure 4.9b Content retrieval success ratio vs. Content popularity skewness (Infocom).	90
Figure 4.9c Content retrieval success ratio vs. Content popularity skewness (RollerNet).	90
Figure 4.10a Content discovery success ratio vs. Content popularity skewness (SF Cab).....	91
Figure 4.10b Content discovery success ratio vs. Content popularity skewness (Infocom).	92
Figure 4.10c Content discovery success ratio vs. Content popularity skewness (RollerNet).	92
Figure 4.11a Content retrieval delay vs. Number of subscribers (SF Cab).	94
Figure 4.11b Content retrieval delay vs. Number of subscribers (Infocom).	95
Figure 4.11c Content retrieval delay vs. Number of subscribers (RollerNet).	95
Figure 4.12a Content retrieval delay vs. Content popularity skewness (SF Cab).....	96
Figure 4.12b Content retrieval delay vs. Content popularity skewness (Infocom).	96
Figure 4.12c Content retrieval delay vs. Content popularity skewness (RollerNet)...	97

Figure 4.13a Content discovery delay vs. Content popularity skewness (SF Cab). ...	98
Figure 4.13b Content discovery delay vs. Content popularity skewness (Infocom)...	99
Figure 4.13c Content discovery delay vs. Content popularity skewness (RollerNet).	99
Figure 4.14 Difference in delay in two scenarios: cache hit and cache miss.	100
Figure 4.15a Content retrieval packet loss vs. Number of subscribers (SF Cab).	101
Figure 4.15b Content retrieval packet loss vs. Number of subscribers (Infocom)....	102
Figure 4.15c Content retrieval packet loss vs. Number of subscribers (RollerNet)..	102
Figure 4.16a Content retrieval packet loss vs. Content popularity skewness (SF Cab).	103
Figure 4.16b Content retrieval packet loss vs. Content popularity skewness (Infocom).	103
Figure 4.16c Content retrieval packet loss vs. Content popularity skewness (RollerNet).	104
Figure 4.17a Content discovery packet loss vs. Content popularity skewness (SF Cab).....	104
Figure 4.17b Content discovery packet loss vs. Content popularity skewness (Infocom).	105
Figure 4.17c Content discovery packet loss vs. Content popularity skewness (RollerNet).	105
Figure 4.18 Cache hit ratio vs. Content popularity skewness α	108
Figure 4.19 Cache hit ratio vs. Spatial localisation factor β	109
Figure 4.20 Cache hit ratio vs. Popularity variation degree.....	110
Figure 4.21 Eviction ratio vs. Relative cache size.	112
Figure 4.22 Average latency (by hops) vs. Subscriber-caching point connectivity..	113
Figure 4.23 Footprint reduction vs. Number of subscribers.	114
Figure 4.24 Relative number of caching points vs. Number of subscribers in random and scale-free topologies.....	115
Figure 4.25a Average CPU usage vs. Number of subscribers in SF Cab.	116
Figure 4.25b Average CPU usage vs. Number of subscribers in Infocom.	116
Figure 4.25c Average CPU usage vs. Number of subscribers in RollerNet.	117
Figure 4.26a Average communication costs vs. Number of subscribers in SF Cab.	118
Figure 4.26b Average communication costs vs. Number of subscribers in Infocom.	119
Figure 4.26c Average communication costs vs. Number of subscribers in RollerNet.	119
Figure 5.1 Packet Loss vs. Number of Subscribers.	124
Figure 5.2 % of Node Coding vs. Number of Subscribers.	125
Figure 5.3 % of Time Coding vs. Number of Subscribers.....	125
Figure 5.4 Energy Consumption (J) vs. Number of Subscribers	126
Figure 5.5 Deployment of MoDiToNeS platform running CafRepCache in Nottingham agricultural field with autonomous drones and vehicle.	129
Figure 5.6a User interfaces on the vehicle taking images of the drone while gathering sensor readings.....	129
Figure 5.6b. User interfaces on the vehicles taking images of each other while gathering sensor readings.....	130
Figure 5.7a Single hop communication in suburb area of Valencia for 2.4 Ghz and 5 Ghz respectively.	131

Figure 5.7b Single hop communication in dense city centre area of Valencia for 2.4 Ghz and 5 Ghz respectively.	131
Figure 5.8 Edge cloud to edge cloud (E2E) latency in multi-hop communications via drone and vehicle intermediaries for agricultural applications.	132
Figure 5.9 Edge cloud to edge cloud (E2E) latency in multi-hop drone and vehicle communications in urban scenarios.	133
Figure 5.10 Edge cloud to edge cloud (E2E) latency in multi-hop communications for different drone's altitude in suburb scenario.	133
Figure 5.11 Example large-scale real-world scenario of CafRepCache deployment.	135

List of Tables

Table 2.1 Overview of techniques that deal with content sharing problem.....	22
Table 2.2 Overview of forwarding techniques.....	24
Table 2.3 Overview of techniques that support congestion control and resource awareness.....	28
Table 2.4 Overview of in-network opportunistic caching techniques in mobile opportunistic networks.....	37
Table 2.5 Overview of the techniques that deal with spatial-temporal locality and multilayer interdependences in mobile opportunistic networks.....	38
Table 2.6 Overview of the techniques that deal with efficient content dissemination and query in mobile opportunistic networks.....	39
Table 3.1 Table of notations	52
Table 3.2 CafRepCache’s pseudo code when a content is arrived.	63
Table 3.3 CafRepCache’s pseudo code when an interest is arrived.	65

Chapter 1

1. Introduction

1.1 Background and Motivation

Over recent years, mobile devices and connections are growing faster than both the global population and Internet users. Current large-scale networking systems thus have been evolved to adapt to the increasing complexity and dynamics in both underlying networking infrastructures and its applications.

Applications are growing at an accelerated pace, becoming more diverse and heterogeneous both in breadth and depth. Global network traffic will increase threefold over the next 5 years [1, 2]. The number of devices connected to IP networks will be more than three times the global population in which over half of global traffic is from wireless and mobile devices [1, 2]. While a majority of Internet traffic will still be dominated by streamed video content, new applications which involve distributed mobile decision making in real time (e.g. remote health care, critical applications for public safety communications, augmented/virtual reality apps and automatic driving/traffic control) have emerged as the increasing number of IoT devices are introduced and adopted annually in different forms with increased capabilities and intelligence.

At the same time, the underlying network has been the focus of huge transformation by emerging novel technologies such as network virtualisation or edge cloud and fog computing as the current Internet approaches are not well suited to support such applications. Smart and cloud-based infrastructures [3-5], which are large-scale distributed network systems built on multi-layer architecture, provide a large range of services and virtually unlimited available resources for users. However, the growing heterogeneous applications involving real-time mechanism have demands that cannot be satisfied by the current approaches [6, 7]. First, these centralized resources can create delays and performance issues for devices and data that are far from a centralized public cloud [8]. Second, with the rapid increase of powerful mobile devices, an individual mobile device can be both a cloud user and a service provider. The increasing number of applications and services are being hosted in the mobile

devices or mobile edges at low cost are becoming significant content providers. Traffic demands for these mobile providers are growing and can lead to localised surges of traffic [9-11]. Third, connectivity may be intermittent due to the nature of mobile users or simply because of limited resources and energy capacity of mobile devices which results in the fragmentation of networks and islands of connectivity ranging in size, from sparse to dense [9, 10]. Therefore, we argue that moving from the core of the networks (e.g. centralised infrastructure/cloud services) to the edges of the network (and thus closer to the users such as people, vehicles, base stations, access points) is essential to address these challenges. Edge computing [8, 12] is a concept to bring processing close to mobile users. By eliminating the distance and time it takes to send data to the centralized cloud, we can improve the delay and performance of data transport, as well as devices and applications on the edge. Fog computing [13, 14] standardises how edge computing should work and facilitates the operation between end devices and cloud data centres. Efficiency, low-cost, timely manner and local relevance are the advantages of Mobile Edge/Fog Computing over conventional Internet Cloud.

This thesis aims to design, build and validate a reliable and efficient latency-aware content caching framework that extends the current state of the art Content-Centric Network (CCN) [15]/Information-Centric Network (ICN) [16-18] research for the Mobile Opportunistic Edge and Fog Computing in order to address these challenges. CCN/ICN departs from the current Internet and adopts the paradigm shift where data objects, rather than host end-points, are named in order to address the requirements of the tremendous growth of demand of the Internet today such as scalable content distribution, mobility, security, etc. Caching in CCN/ICN architectures allows the network layer to identify cached content without the application layer.

Although caching content closer to users is a common solution to reduce the network traffic and improve the latency, caching mechanisms employed by present Internet (e.g. Web caching [19], Peer-to-Peer (P2P) caching [16, 18], Content-delivery network (CDN) caching [20, 21]) are unable to be directly ported to content-aware caching in Mobile Edge Computing. Traditional approaches to content caching through websites, mobile apps and services

which rely on the remote centralized clouds, gather remote data and process them centrally, do not address sufficiently limited network coverage, users dynamic mobility and network congestion. In addition, dynamic mobile topology, the volatility of cached content and the ubiquity of in-network caches add difficulties and challenges to the mathematical modelling and analysis of the caching system in MEC. Extensive research in caching is being done but with significant differences in terms of context and priorities, knowledge assumed, mobility patterns, workload patterns, scale, cost, etc. Even though there are approaches, such as mobile offloading [22, 23] and content distribution networks [20] in which large commercial providers (e.g. Netflix, BT, GTT, etc.) use data centres to hold their data across a large geo area while delegating content distribution to servers located as close as possible to the users that aim to address problems of fast-increasing data in mobile networks and traffic surges, these models often assume good connectivity among the nodes, global knowledge of the users across large geo areas and do not address high topology dynamics with intermittent disconnections and dynamically changing publishers and subscribers workload patterns.

This thesis on caching framework extends and complements work done by commercial providers as well as research literature by focusing on scenarios which have significantly higher topology dynamics including potential disconnections and more dynamic workload patterns such that there is unknown in advance publishers and subscribers usage patterns and their spatial-temporal distributions.

1.2 The Specific Focus of This Thesis

To support different types of applications, including heavy real-time multimedia applications at the network edge, we propose new mobile edge predictive caching which leverages information on the dynamic content traffic workload, mobility of publishers/subscribers and local resource availability (energy, storage, computation). We investigate complex challenges of opportunistic discovery of content and services stored in remote (mobile) edges and opportunistic delivery to the requesting (mobile) edges. Due to the nature of mobile users, we do not assume end-to-end connections for transmission of data but rather to be able to opportunistically discover and retrieve requested

contents from mobile edges in heterogeneous mobile networks, with or without the assistance of network infrastructures. Compared to the conventional approaches, opportunistic content discovery and retrieval are nontrivial and increasingly complex due to lack of a central knowledge (e.g. to store any content index) and time-varying network topology.

The intuitive initiation that has been universally agreed is that a node prefers fetching from the closest source to reduce delay and network traffic, thus we aim to move the cache closer to the subscriber which then becomes very challenging and less clear in fragmented and dynamic topology with both mobile subscribers and publishers. In addition to this, existing opportunistic caching policies in mobile edge/fog networks typically cannot capture, predict and adapt to the spatial-temporal locality of content requests needed for more accurate content popularity-based caching decisions because they rely on assumptions that content interests occur independently of users' mobility and resources [24, 25]. More specifically, when content caching predictions are not sufficiently fine-grained, the content request needs to traverse from the subscriber to the publishers because the caching points are unable to fulfil the content requests. The state when the content requested is not found in caching point's memory is considered as cache miss which results in longer content retrieval delay and higher resource consumption. As opposed to cache miss, a caching mechanism aims to maximise the number of content requests being filled successfully by the caching points, which is also called cache hit. Note that due to the nature of forwarding and replication mechanism in mobile opportunistic networks with no centralised controllers, we consider cache hit as the first request that matches with the content in a caching point without being forwarded to publishers and indicates the efficiency of caching decisions and locations. To achieve the ultimate goal that maximize the end-to-end content retrieval success ratio, cache hit ratio and reduce latency, there are three essential issues for any caching system:

1. How to determine what contents to cache in the caching points to increase the probability that future content requests will be fulfilled successfully based on historical spatial-temporal content traffic patterns?

2. How to choose where to cache the contents that are as close as possible to the subscribers and are sufficiently highly central for overall content forwarding while avoiding overloading local and regional resources?
3. How to manage the cache to not remove long-term useful contents because of one-timer contents while still being responsive to the dynamic variation of content traffic patterns?

We propose new collaborative adaptive predictive caching approach suitable for content dissemination and query in heterogeneous (non-uniform) opportunistic mobile networks and with dynamically changing (non-uniform) workloads. In addition, in-network caching and replication mechanisms in Mobile Edges could (be) affected by routing/forwarding and searching decisions, thus suggests a distributed multi-layer framework involving jointly optimizing in-network cache management, routing, forwarding with congestion avoidance and energy efficiency.

Our content caching approach aims to capture more accurately the dynamic spatial-temporal correlation of mobility and traffic patterns as well as the mobility-content traffic interplay needed to support more reliable, adaptive caching algorithms. Previous research [26, 27] has shown that the centralised solution is not scalable in mobile complex heterogeneous network topologies due to its high complexity and single point of failure problem. Moreover, distributed solution technique [26, 27] may still cause high connectivity overheads although it provides a cheaper computable lower bound compared to the centralised solution. Besides, previous research has also shown that collaborative caching usually outperforms both locally and centrally optimized algorithms [28]. Therefore, our caching solution is based on fully-distributed predictive edge analytic which build multi-layer complementary real-time distributed predictive heuristics to maintain the best possible trade-off between caching performance and resource consumption.

1.3 Envisaged Networks & Applications

Consider a scenario such as urban and cultural events that may have a range of official websites and mobile services provided for them (e.g. Festivals held in Edinburgh, Avignon, Glastonbury attracting over 200,000 downloads before the

festival [29]) but do not perform well (or not at all) at the events. These problems are mainly due to network connectivity problems that may be caused by i) the limited network coverage (consider the festivals that take place in sparse/rural area (e.g. Glastonbury Festival) or a city (e.g. Avignon Festival) with many high building blocks, physical obstacles and network interferences), ii) congestion (consider the surge of traffic demand from over 100000 visitors, tourists, etc. in Avignon or Glastonbury Festival in the same period of time and location). Mobile users, tourists, artists, etc. are now not just consumers of content (subscribers), but also prime producers (publishers) of complex contents. In such environments, this massive “mobile” contents cannot easily be shared or requested by users. Instead of a content having to be requested and retrieved from a remote mobile publisher every time a member user of a group requests it, the content should be stored or cached nearby the users so that the subsequent requests will not need to be answered directly by the publishers or infrastructures which are far away (or even disconnected) from users but are being served by the nearby intermediate (their friends, family, group partners, etc.) as a localized ad-hoc service, thus helps to reduce latency and improve the reliability of mobile services.

Another scenario that motivates content caching in mobile opportunistic networks is when drivers travelling along the streets may benefit from receiving information from the other drivers about various conditions on the road e.g. drivers who are able to capture the video or a picture of the event may be able to share it with cars around it and other cars which may be on the way to the affected location [6, 30]. More specifically, publishing car, instead of uploading the accident video or images to cloud-based infrastructures to notify others about a particular event or problem, could place the contents to the following cars or RSUs so that subsequent same requests for the accident images from the middle cars could be able to be served in a localized and more responsive manner and with lower delays (compared to the infrastructure).

1.4 Thesis Contributions

This thesis contributes to the field of mobile heterogeneous opportunistic networks by providing a framework for enabling mobile edge and fog devices

to perform content-aware latency-aware and resource-aware caching operations. This framework is the result of the following core contributions.

- A multilayer content caching framework that predictively adapts to the dynamics and complexity of multiple network layers ranging from network underlying infrastructure to its heterogeneous applications. By this way, the content services are more robust, scalable and adaptive to very different time-varying dynamically changing: network topologies, mobility patterns, resources and workloads of content publishers and subscribers in heterogeneous mobile opportunistic networks.
- Multidimensional predictive edge analytics that build multi-attribute complementary predictive heuristics and utilities. Each predictive analytic and heuristic is able to produce complex trend analytics of different network layer: dynamic content traffic workload as well as dynamic network topologies and resources. When combined together, our predictive analytics and heuristics help to deal with multiple complex inherent trade-offs between maximising cache hit ratio, content delivery success ratio while minimising the end-to-end latency, resource consumption and enabling congestion avoidance.
- Temporal content-request predictive analytics and heuristics that combine request frequency, recency and betweenness to choose the best suitable contents to cache and when to cache. It helps to predict in real time the trend of content request patterns over time, more responsive to the dynamic variation of content popularity, avoids losing valuable contents by reducing the caches for one-timers contents.
- Spatial content-request predictive analytics and heuristics that combine clustering coefficient, similarity, closeness, tie strengths between different subscribers for the same content in order to choose the best caching point for the content and the group of subscribers. It enables to answer the question of what and where to cache a content better in the dynamic changing content workload as well as mobility of subscribers and publishers.
- Spatial-temporal predictive analytics and heuristics help to perform a precise spatial-temporal behaviour characterisation of both individual and localised regions of subscribers in order to cache the most suitable set of contents in

the most suitable set of nodes. It allows improved congruency with both the underlying network and users demands, more accurately capture the temporal and spatial locality of users' mobility and content requests as well as users' mobility-content traffic patterns interplay.

- A content replacement policy and offload mechanism that avoid losing low-popular contents by offloading them to nodes with higher available resources while still keeping them as close as possible to the subscribers.
- A principle of dynamic predictive relative utility to propose a collaborative algorithm which allows individual nodes to achieve greater utility than if they do not collaborate.
- A new reputation-based popularity weighing mechanism to adaptively weight different exchanged values gathered from different nodes in the ego network based on node's reputation and connectivity, in order to improve the accuracy of collaborative forwarding and caching decisions in a fully-distributed manner.
- An extensive evaluation emulating real network conditions, incorporating device connectivity traces from a number of possible deployment scenarios with different heterogeneous content publish-subscribe patterns, both are real-world and pseudo real-world models and experimentally evaluating a range of criteria.
- A real deployment in two different real-world contexts of mobile heterogeneous sensing and query in mobile opportunistic environments as well as performance evaluation across a range of criteria.

We offer a novel adaptive real-time predictive content forwarding and caching approach which is multi-dimensional (i.e. mobile, digital and physical), multi-layer (i.e. comprise intelligent routing and storage, user data analytics and processing), multi natured (i.e. social, vehicular, computer networks), heterogeneous (both in terms of end resources, communication networks and traffic demand) and dynamically changing (both in terms of time and geographical locations) to address the complex challenges of content discovery and retrieval in unreliable network environments. Our approach supports high reliability, latency-sensitive and efficiency with resource and congestion awareness in the face of unknown-in-advance dynamically changing social,

interest and network structures. We envisage that our forwarding and caching solution will be integrated with the future mobile apps, services, smart transports, smart infrastructures to enable more robust network support and allow more reliable rich mobile multimedia services in a variety of application areas where users suffer from limited network coverage, traffic demand and network congestion.

1.5 Thesis Structure

The remainder of this thesis is structured as follows.

Chapter 2 first introduce a set of criteria we used to review existing work, then presents detailed description and discussion of the most relevant literature surrounding the focus of this thesis and according to this set of criteria. More specifically, we give an overview of peer-to-peer content sharing techniques, ICN/CCN, content forwarding and caching in mobile opportunistic networks.

Chapter 3 explores the design space and introduces the overview of our proposed framework, followed by a functional model of the approach, a detailed description of our proposed solution and a description of our algorithm.

Chapter 4 describes our experimental methodology, the implementation and performance evaluation of our proposed solution. The experimental methodology comprises of a detailed description of different realistic and pseudo-realistic underlying network topologies as well as application traffic models, our competitive protocol, performance metrics and our emulation configuration.

Chapter 5 discusses the possible extensions of our framework, such as using collaborative joint adaptive network coding with caching to deliver improved QoS and support more efficiently dynamic workload. We discuss the dynamic scalability of our framework via the emergence of network softwarisation which open up new ways to architect and deploy our framework in different application domains.

Chapter 6 concludes the work and main contributions presented in this thesis and discusses future directions.

Chapter 2

2. Literature Review

The dynamic variations of mobility, users' content traffic patterns and resource restrictions in mobile opportunistic networks pose many challenges (i.e. topology changes, disconnections, lack of global knowledge, etc.) for the design and implementation of future content caching services with an aim to bring contents as close as possible to the users. Even though no work has explicitly addressed this problem, there is a wide range of relevant existing techniques that may be useful to building such a service that improves the reliability and efficiency of mobile services support for users.

This chapter is organised as follows. We first introduce, define and motivate a set of criteria used for reviewing existing work. In the second part, we move to detailed descriptions and discussions of the most relevant approaches according to this set of criteria. In Section 2.2 we give an overview of peer-to-peer content sharing techniques, which are categorised into Web Caching (Section 2.2.1), Content Distribution Networks (Section 2.2.2), Mobile Offloading (Section 2.2.3), Content Caching in Information-centric Networks (Section 2.2.4) and Opportunistic Caching in Mobile Opportunistic Networks (Section 2.2.5). In Section 2.3 & 2.4, we review interest and content forwarding methods as well as congestion control methods in Mobile Opportunistic Networks. In Section 2.5, we focus on the literature regarding opportunistic caching in Mobile Opportunistic Networks showing that none of the existing work resolves our research problem. In section 2.6, we review different techniques that aim to capture the multidimensional interdependences between multiple layers of networks: mobility, content requests and resources in both time and location which has been shown to be significantly important for the design of future mobile caching algorithms.

2.1 Criteria

We have done an extensive literature review that helps us to identify a set of criteria to be applied when reviewing related existing work (and to evaluate the work presented in this thesis). For each criterion, we will discuss its importance

and necessity to support building smart content caching services in mobile opportunistic networks. The list of our criteria are:

1. Support for fully-distributed predictive decision making in real time with limited localised knowledge.
2. Support for adaptability and responsiveness to potentially fast-changing local conditions, disconnections due to dynamic mobility in dense and sparse topologies.
3. Support for dynamic traffic demands from users acting as both mobile publishers and subscribers, be responsive to dynamic variation of content popularity while mitigating flash crowd effects.
4. Support for network congestion controls and avoidance.
5. Support for efficient and fair distribution of cache space across trusted incentivized collaborators.
6. Support for capturing the spatial-temporal locality of users' mobility and content requests and adapting to multidimensional interdependences between the multilayers of mobile networks: mobility, traffic patterns and resources.
7. Support for low end-to-end latency in the content retrieval process while maintaining acceptable success rate and minimising packet loss and resource consumptions.

2.1.1 Support for fully-distributed predictive decision making in real time with limited localised knowledge

The limited connectivity, lack of infrastructure and fragmented nature of opportunistic networks mean that it is neither efficient to obtain and maintain a global level of knowledge of the network nor to rely on closed-loop (acknowledgement based) techniques for any decisions.

Therefore, it is important to provide a framework that allows every node in the network to make fully-localised predictive decisions in real time based on its limited localised knowledge. Moreover, it is undoubtedly important for each node in the network to be able to collaborate and exchange its local observations with other neighbour nodes when two nodes are in contact in order to capture

and detect various events (e.g. user connectivity patterns, request patterns) in a wider, more accurate and responsive manner.

2.1.2 Support for adaptability and responsiveness to disconnections due to dynamic mobility in dense and sparse topologies

Intermittent node connectivity forms islands of network coverage ranging in size, from sparse to dense. While sparse networks suffer from limited network connectivity, dense networks are prone to transmission collisions (due to wireless interference) and network congestions. Because of the nature of mobile users, the network may change significantly over time, popular nodes may become extremely less popular, and emerging new nodes may become extremely popular in a very short time.

Due to the dynamic mobility and connectivity patterns, the paths between publishers and subscribers are usually intermittent which requires the caching services to adapt and respond quickly to the fast-changing local conditions in both dense and sparse environments.

2.1.3 Support for dynamic traffic demands from users acting as both mobile publishers and subscribers

Traffic demands for mobile edge-hosted providers are increasingly challenging as the published data is massive (i.e. the majority of Internet traffic is now dominated by streamed video content) and can lead to surges of traffic if the content becomes popular. Any node can start to publish, subscribe or help to cache contents at any time and from any location while being mobile. Moreover, the content request workload changes dynamically due to not only users' behaviours but also the dynamic mobility of users and sporadic connectivity of nodes in mobile opportunistic networks.

Emerging research [25, 31, 32] states that content requests often exhibit both temporal locality and spatial locality. The temporal locality of content captures the effect that, if the content is requested at a certain period in time, more likely it will be requested again in near future. The spatial locality of content requests is based on the fact that content requests of the same content are more likely to be issued by geographically close areas. More precisely, the requests coming from a specific region in space are more likely to be similar than those collected

over regions far apart. Thus, caching services should be able to capture the spatial-temporal locality of content, more responsiveness to the rising trend of newly high popular contents and fading out of older contents.

Furthermore, caching decisions based on the popularity of contents help to solve the cache pollution or flash crowd problem which is caused due to one-timer contents or fake news [33] and has been shown to affect the performance of a caching policy by increasing both the content retrieval latency and the network traffic. One-timer contents are contents that have been requested once, while cache pollution occurs when one-timer contents are cached. Since one-timer contents have been estimated to even up to 70% of the content requests [33] and content caching is expected to serve a higher number of contents under more severe constraints such as limited cache size, cache pollution becomes an important complication that a variety of caching policies without intelligent content popularity prediction such as the fixed-probabilistic caching policies and the static local-based caching policies fail to address.

2.1.4 Support for limited resources, network congestion controls and avoidance

Mobile devices have limited resources, battery capacity, buffer availability and transmission bandwidth. Due to the limited resource capacity, highly connected nodes in the network may quickly become overloaded and unusable, causing even more disconnections and consequently lower throughput.

Forwarding and caching algorithms in mobile opportunistic networks identify a subset of highly connected nodes to which they forward and cache the majority of the traffic. As these nodes become congested, forwarding or caching to lower connected but higher resource availability nodes may lead to more spread of the network load in the network and lower congestion rates.

2.1.5 Support for efficient and fair distribution of cache space across trusted collaborators

Whilst we assume that nodes in the network follow a specific algorithm, we still need to guarantee that the algorithm itself is not greedy. In order to avoid selfish behaviour that leads to decreased network-wide performance, we aim to design an underlying algorithm that can adaptively share distributed cache space across

trusted collaborators when both network topology and workloads are dynamically varying.

2.1.6 Support for multidimensional interdependences between multilayers of mobile networks: mobility, traffic patterns and resources

Emerging research has shown that understanding and leveraging the multidimensional interdependent multilayers of mobile networks: mobility, traffic patterns and resources are significantly important for the design of future mobile caching algorithms.

In earlier studies, this interdependence has not been widely considered, and models for both mobility and network traffic planes have been developed and evaluated largely in isolation [24]. For example, when evaluating mobile systems' performance, traffic generation generally follows regular patterns, drawn from common distributions (e.g., exponential or uniform), while assuming neither transmission nor reception of data impacts mobility. However, simply observing people walking while staring at (or reacting to) their smartphones suggests that such interdependencies need to be captured properly. Therefore, capturing and predicting the dynamic spatial-temporal correlations of mobility and traffic patterns are profoundly open complex questions that affect the design of future mobile algorithms.

2.1.7 Support for maintaining acceptable success rate and low end-to-end latency

The delay-sensitive and critical applications face the problem of high latency when a potentially large number of devices and users need to upload information to or search for information from the clouds and centralized infrastructure.

In mobile opportunistic networks forwarding and caching algorithms, content retrieval success ratio and latency are the key concerns that are always challenged by the disconnections and congestions in the networks.

2.2 Techniques for efficient peer-to-peer content sharing

The following section gives a brief overview of recent advances in peer-to-peer content sharing techniques focusing on improving the performance of content discovery and retrieval in heterogeneous network scenarios.

2.2.1 Web caching

Web caching has been recognized widely as one of the effective techniques which keep data that are expected to be frequently requested easier to be retrieved in order to alleviate the service bottleneck and reduce the network traffic, thus minimize the user access latency. Many efficient web caching solutions have been proposed and employed in the web system architecture, typically deployed in the application level.

Authors in [19] proposed a peer-to-peer cache-oriented approach for Web applications based on the principles of Behavioural Locality inspired by collaborative filtering.

Authors in [34] showed evidence that content requests in static web follow the Zipf's law distribution. The authors introduced a simple model for web request and suggested that this simple model may be sufficient to understand certain asymptotic properties of cache performance.

Both [19] and [34] apply centralised decision making and require global information, thus do not support fully-localised distributed predictive decisions (criterion 1) which we identify as one of the most important criteria in this research proposal. Moreover, these approaches are built for static web caching in fixed networks which does not support the mobility of publishers and subscribers. Today's social networks have also questioned the validity of Zipf's law model in [34], showing that content requests often exhibit both temporal and spatial locality with respect to the mobility behaviour of users.

2.2.2 Content distribution networks

A content distribution network or content delivery network which is composed of a large number of edge servers (also called surrogates) distributed in different regions has become a significant service in the current Internet dealing with the explosive growth of Internet traffic. Content caching and server load balancing, etc. techniques enable the CDN to push or pull the contents to the edge servers located close to users, thus the users can obtain the desired content nearby.

Authors in [20] proposed a regional caching approach of video content that captures the overlap between inter-regional and intra-regional preferences to provide better cache performance.

Authors in [35] investigated the conflict between caching decisions and recommendation systems in CDN that to balance the trade-off between satisfying user individual preference and optimising aggregated demand over all users served by the cache. The authors showed that by carefully choosing between the individual user interest and preferences from many users, the recommender system can result in improved QoE for end-users.

The caching algorithms in [20, 35] rely on centralised infrastructures and require global information, thus do not support fully-localised distributed decision making (criterion 1). Furthermore, while typical CDN approaches address problems of traffic surges, they do not support users' mobility and disconnection tolerance (criterion 2), dynamically changing workloads and network topologies (criterion 6) which are inherent in the scenarios of mobile publishers and subscribers and can be leveraged to optimize the distribution performance.

2.2.3 Mobile offloading

Mobile data traffic is growing exponentially leading the backhaul data rate requirements to become the major bottleneck to reducing costs and raising revenue for network operators. Mobile offloading has attracted significant attention since they can effectively reduce the backhaul traffic by eliminating duplicate data transmission that carries popular content. However, current mobile offloading approaches rely on static infrastructure with global knowledge on the network topology (criterion 1) and do not support the regular intermittent in network connectivity between mobile dynamic changing publishers and subscribers (criterion 2).

Authors in [22] proposed a social-aware hybrid offloading strategy for load balancing and computation sharing based on node's stability which is measured by contact frequency and duration in order to improve the availability of offloading support for mobile users.

Research in [36, 37] focuses on the problem of offloading whole content and streaming videos from vehicles acting as mobile caches (vehicular cloud) respectively. Intuitively, the rough trade-off is the following: the higher the number of vehicles caching a content, the higher the chance that this content (or

chunk) will be downloaded from such vehicles “on time” (rather than from expensive macro cell links).

However, [22, 36, 37] does not support predicting network congestion nor end-to-end latency awareness during the offloading process. Moreover, while typical mobile offloading techniques address problems of fast-increasing data in mobile networks, they do not address high topology dynamics with intermittent disconnections and dynamically changing publishers and subscribers workload patterns.

2.2.4 Content caching in Information-Centric Networks (ICN) and Content-Centric Networks (CCN)

Information-centric networks (ICN) and Content-centric networks (CCN) have attracted many research efforts as a foundation for the future Internet. Current network architecture follows host-centric end-to-end communication paradigm and thus, was not designed to address the requirements of the tremendous growth of demand of the Internet today such as scalable content distribution, mobility, security, etc. ICN/CCN departs from the current Internet and adopts the paradigm shift where information objects, rather than host end-points, are named. Caching is a fundamental feature of ICN/CCN architectures, allowing the network layer to identify cached content without the application layer. ICN/CCN caching is often separated into two categories: on-path caching and off-path caching. Typical ICN/CCN approaches for traditional Internet architecture assume global resolution system or Domain Name System (DNS) and have limited support for dynamic mobility of both publishers and subscribers (criterion 2).

Authors in [38] proposed a novel network architecture built on IP’s engineering principles, but using named content rather than host identifiers as its central abstraction. The authors demonstrated an in-network content caching approach to improve the reliability and scalability of content-named networks, deployed in fixed network topologies. [38] does not support dynamically changing mobility and content traffic workloads of publishers and subscribers.

Authors in [39] utilised stateful SDN to deploy Information-centric Networks in current TCP/IP Infrastructure. The architecture comprises stateful SDN

switches, each of which is attached to a local cache. This combination of stateful switch and cache can successfully replicate the behaviour of an ICN node. As the authors implemented their solution in fixed network topologies, they do not consider the intermittent paths between nodes in the network and the dynamically changing network topologies.

Research in [40] proposed a novel approach that combines latency-aware fully distributed caching and load-aware dynamic forwarding strategies to improve the content-centric network delivery performance. However, to optimise the performance of popularity-aware caches which requires an optimal split of the forwarded traffic, it routes more popular content requests through a single path without considering congestion control and avoidance. It also does not support fully-localised (i.e. open loop) decision makings

Authors in [41] proposed a decentralised, real-time distribution of content in router caches, reduce caching redundancy and in turn, make more efficient utilisation of available cache resources along a content delivery path in ICNs. [41] does not support the dynamic mobility of users. In addition, the proposed solution relies on an assumption that the popularity distribution of the network contents approximately follows Zipf's law and the characterisation of content requests have been based on Independent Reference Model (IRM) which assumes that content requests occur independently, thus cannot capture the spatial-locality of content traffic patterns.

2.2.5 Opportunistic Caching in Mobile Opportunistic Networks

Traditional ICN/CCN architecture and key functionality described above assuming that network connectivity is continuous (end-to-end communication), and not intermittent. This section focuses on content caching for mobile and opportunistic networks. In contrast to content caching approaches for ICN architecture, content caching applied in mobile opportunistic networks does not assume a global resolution system or Domain Name System (DNS).

Authors in [26] proposed a low-complexity heuristic FairCache algorithm that models collaborative caching as a bargaining game to address the fairness problems in fixed environments. [26] suffers from multiple limitations when applied to the mobile dynamic heterogeneous context: i) [26] does not support

dynamic content request patterns which is significant for fully-localised collaborative predictive caching. ii) [26] considers cache hit ratio to define utility function but does not take into account end-to-end latency which is important for mobile disconnection tolerant scenarios. iii) [26] does not support congestion control, resource awareness nor considers the trade-off between caching performance and overheads which is necessary for more realistic scenarios (i.e. note that optimal solution for the bargaining game requires either global knowledge which is infeasible in dynamic topology or massive messages exchanged between individuals).

Research in [27] proposed a model to understand the spread of content discovery messages in an opportunistic network under a given hop-limit to ensure efficiency without being resource-wasteful. The authors integrated interest scoped-flooding with simple Least Recently Used caching strategy and showed that search scope can be limited to a few hops as after several hops the extra gain diminishes as a function of content availability and tolerated delay. [27] only study the efficiency of forwarding algorithm and thus do not consider the cache efficiency.

Research in [28] argued that as non-cooperative caching is close to the current Internet architecture, it generally neglects all the content networks benefits. On the other hand, explicit cooperation is hard to achieve and the attempts had negative experiences. Thus, the authors suggested a novel collaborative caching and showed that collaborative caching usually outperforms both locally and centrally optimized algorithms. [28] focus on the fixed network topologies, thus do not consider the dynamic mobility, connectivity of content publishers and subscribers.

Authors in [42] proposed utility-driven caching, where each content is associated with a utility, which is a function of the corresponding content hit probability, then formulate optimization problems where the objectives are to maximize the sum of utilities over all contents. [43] introduced a novel non-cooperative game model to characterize the caching problem among selfish nodes without any central coordination. However, both [42] and [43] solutions

achieved the global optimum which is shown as being outperformed by collaborative caching [28].

Authors in [44] proposed a forwarding and cache replacement policy for Social Cache based on content popularity driven by frequency and freshness of content requests. As part of its replacement, Social Cache may remove a cached content from the network, thus reduce the cache hit ratio and increase delays. This problem is exacerbated when the resource is limited and the replacement rate is high as is not resource and congestion aware.

Research in [45] proposed an approach to find an optimal search path (in hops) of content discovery where the query takes actions based on its a priori information and the observations made during the search. However, since a node has multiple observations from other neighbours, it applies simple exponential moving average to update aggregated estimated values. The authors also applied single-path search query which means the request is not replicated and assume the delivery path is equally long to the search path. In addition, the authors do not consider resource awareness and congestion control. [45] also assumed all content requests between different users are independent as well as there is a uniform request rate between different users. The authors do not address caching problem in opportunistic networks and assume that content popularity measurements are all local.

Authors in [46] designed an algorithm to improve the efficiency of content sharing by assessing what information is relevant for the users and what content needs to be exchanged during contacts to satisfy the users' interests. [46] consider only node contact durations to disseminate contents. The content popularity is based on a simple counter upon arrival of content requests (frequency) thus cannot capture the locality of content requests. Furthermore, only the local observation of interests are considered, thus cannot capture more regional and wider network states. The proposed algorithm also do not support congestion control while disseminating contents in networks.

2.2.6 Summary

Table 2.1 shows an overview of techniques that deal with efficient peer-to-peer content sharing problem.

Domain		Key Contributions	Cons
Behave [19]	Web Caching	• Peer-to-peer cache based on behavioural locality • Collaborative filtering in decentralised caches • Exploits overlapping communities of peers with similar content interests.	• No mobility support (criterion 2) • No spatial-temporal locality support (criterion 6) • Rely on global knowledge (criterion 1)
RegionCache [20]	CDNs	• Decentralised caching • Global content popularity vs. regional content polarity • Capture overlapping between inter-regional & intra-regional content requests.	• Limited support for high topology dynamics with intermittent disconnections (criterion 2) • Rely on global knowledge (criterion 1) • No spatial-temporal locality prediction of content requests (criterion 6) • No support for dynamic changing publishers & subscribers (criterion 2)
HyMobi [22]	Mobile Offloading	• Load balancing & computation sharing between nodes in offload communities • Credit-based & reputation-based incentive for strong social catalyst	• No latency awareness for content sharing (criterion 7) • No congestion avoidance (criterion 4) • Limited support for high topology dynamics with intermittent disconnections (criterion 2) • Assume uniform content workload patterns from publishers & subscribers (criterion 6).

FairCache [26]	ICN/CCN	• Collaborative caching as a bargaining game • Low-complexity heuristic • Fairness considered between collaborators	• No dynamic content requests support (criterion 6) • No congestion control (criterion 4) • No mobility support (criterion 2) • Only cache hit considered, no latency awareness (criterion 7)
SocialCache [44]	Mobile Opportunistic Networks	• Forwarding and cache replacement policy • Content popularity driven by frequency and freshness of requests • Node centrality in cluster	• No congestion control & avoidance (criterion 4,5) • No spatial-temporal locality support (criterion 6) • Isolated layers between mobility and content traffic (criterion 6)

Table 2.1 Overview of techniques that deal with content sharing problem.

In the rest of this chapter, we review the state-of-the-art forwarding, replication, congestion control and caching protocols which have been proposed for the context of mobile heterogeneous environments.

2.3 Forwarding techniques to deal with disconnections and dynamic mobility

Forwarding algorithm aims to forward a message to a given destination which is different from routing techniques because end-to-end connectivity is not assumed. Moreover, forwarding techniques focus on selecting the next node(s) to transfer the message with respect to nodes encountering other nodes, rather than calculating the shortest path in the entire networks.

2.3.1 Non-Social based forwarding

In Epidemic routing [47], messages are flooded through the network using flooding. Each node continuously replicates and transmits messages that it is sending or has received (and subsequently is forwarding) to newly discovered neighbour nodes that have not seen the messages. While Epidemic routing is

flooding-based in nature, more intelligent techniques can be used to limit the number of message transfers.

Spray And Focus routing protocol [48] and Spray And Wait routing protocol [49] share the same characteristic that it 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 that in the second phase, a relay node takes advantage of potential opportunities to forward a message closer to its destination by using the single-copy utility-based scheme, which states that a message is forwarded if an encountered node has a smaller average time between encounters with the destination node.

2.3.2 Social based forwarding

SimBetTS [50] is a complex utility-based social forwarding strategy, which incorporates three significant concepts: Betweenness Centrality, Node Similarity and Tie Strength to define the highest central nodes in the networks for forwarding messages.

MaxProp [51] is a flooding-based routing protocol in which if contact is discovered, any previously unseen messages by the contact will attempt to be replicated and transferred. MaxProp maintains a priority queue based on the destination of each message, ordered by the estimated likelihood of a future path to that destination so that it can determine the priority/order of messages being transmitted or dropped.

In Bubble Rap routing protocol [52], each node calculates a degree centrality value which evaluates the effectiveness of the node in the forwarding process. Based on the centrality value, a node can decide which neighbouring nodes a message should be forwarded to.

Authors in [53] proposed a hybrid Swarm Intelligence-based approach designed to address the routing problem in mobile environments based on the analysis of mobility patterns and operational metrics that characterize the opportunistic social connectivity between wireless users. The most promising message forwarders are selected via a greedy transition rule which is based mainly on local information captured from the disconnection-tolerant networks

environment. Whenever global information is available, it can also be used to support decisions.

2.3.3 Summary

Table 2.2 shows an overview of forwarding techniques that deal with disconnections and dynamic mobility of nodes in the network.

Category		Key Contributions	Cons
Epidemic [47]	Non-social forwarding	• Messages are flooded through the networks.	• No congestion control (Criterion 4) • No resource awareness (Criterion 5) • No latency awareness (Criterion 7)
Spray and Focus [48]		• Disseminates a predetermined number of copies of a message to encountered nodes. • Utility-based scheme	
SimbetTS [50]	Social forwarding	• Betweenness centrality heuristic • Node Similarity heuristic • Tie Strength heuristic	
Bubble Rap [51]		• Human interaction is heterogeneous both in terms of hubs and groups or communities. • Windowed degree centrality heuristic	

Table 2.2 Overview of forwarding techniques.

2.4 Techniques to support network congestion and resources awareness

This section reviews a wide range of existing congestion control and resource-aware techniques categorised into Congestion control, Multipath routing, Replication, Clustering and Network Coding. The Congestion control section reviews the forwarding techniques integrated with congestion control mechanisms deployed in mobile heterogeneous opportunistic networks. The Multipath routing section and P2P Replication section inspect the state-of-the-art multipath routing and forwarding protocols which have shown multi-path network system is crucial to achieve resource pooling, increase reliability, scalability, flexibility and efficiency [54]. Clustering section describes mechanisms which divide nodes into virtual groups (also called "cluster") based

on some predefined metrics such as density, velocity, location, etc. The network coding section inspects the state-of-the-art network coding mechanisms specifically developed for our network domain, exploiting the power-law behaviour that these networks exhibit (i.e. the existence of more centralised nodes that can collectively connect to the rest of the nodes) and use highly central nodes as message relays that employ random linear coding addressed to the same destination.

2.4.1 Congestion Control

Authors in [55] proposed a fully localised adaptive forwarding, replication and congestion control protocols, Café, to enable a congestion-aware mobile social framework for predictively adaptive data forwarding and rate adaptation over heterogeneous disconnection tolerant networks. The authors proposed novel heuristics retentiveness, receptiveness, congesting rate to manage in-network delay, resource availability and congesting rate in the network to predict and avoid the network congestions.

Authors in [56] proposed RAPID utility-driven resource allocation models for disconnection-tolerant networks (DTNs) forwarding where the decisions to forward or drop messages are based on a utility function. The utility metrics in RAPID are: average delay, missed deadlines and maximum delay.

Research in [57] proposed a fully distributed, low complexity, and responsive load controller to avoid potential computation congestions when executing in-network services.

2.4.2 Multi-path Routing

Authors in [58] proposed SocialCast which is a routing framework for publish-subscribe that utilises social metrics (i.e. colocation and mobility patterns among communities) to identify the best relay nodes that support delay-tolerant communication in human networks.

Authors in [59] conducted an insightful analysis of the fairness achieved in bandwidth allocation by coordinated and uncoordinated rate control over multiple links in peer-to-peer networks.

2.4.3 Peer-to-Peer Replication

Study in [60] proposed CafRep, a forwarding and replication framework based on utility function of nodes to control the message replication rate such that the replication limit grows to take advantage of all available resources, but backs-off when congestion increases, similar to how TCP updates its congestion window. As a result, the protocol is able to replicate at adaptively lower rates in the parts of the network that have low buffer availability, increased node delay and are likely to congest faster. When CafRep node discovers parts of the network with higher buffer availability, lower node delays and slower congesting rates, it replicates at a higher rate.

Authors in [61] studied the replica placement in a P2P system to optimise the number of replicas. It shows that centralised control of resource placement is NP-hard problem and that if the control is fully decentralised, selfishness can be harmful to the system performance that leads to ultimately unusable.

2.4.4 Clustering techniques

In the last recent years, researchers have proposed a various number of clustering techniques with an aim to provide a stable virtual communication backbone for efficient content delivery in mobile networks and increase its scalability. In cluster structure, each node may be assigned in different status or roles: i) cluster head is a local coordinator in its cluster, has responsibility for intra-cluster communication, data forwarding, etc. ii) cluster gateway is similar with conventional gateway's meaning, a cluster gateway is in charge of inter-cluster communication and forwarding information between clusters; iii) cluster member.

Authors in [62] focus on geo-based content sharing where content items are cached at the anchor location and making it available to other nodes within a maximum distance from this location. However, it takes into account only a subset of mobile devices actively using GPS. Furthermore, error-prone GPS measurements reduce the performance of content sharing in many scenarios. Moreover, there is no guarantee that content posted to an anchor zone will remain available due to the mobility of users.

Authors in [63] argued that mobile edges do not move randomly in realistic, thus explored cluster approach to group mobile edges based on mobility, speed, direction, social interest, etc. in order to get better caching decisions amongst nearby nodes.

2.4.5 Network Coding

Research in [64] proposed CafNC algorithm where each node learns the status of its neighbours, and their ego-networks in order to detect coding opportunities, and codes as long as the recipients can decode. The design allows CafNC to efficiently support multiple unicast flows, with dynamic traffic demands and dynamic publishers and subscribers.

Authors in [65] showed that network coding is applicable for the ICN/CCN architecture due to its proven benefits for multicasting scenarios and distributed storage networks. The authors proposed a practical implementation of linear coding scheme for the ICN/CCN and showed that it helps to improve the performance of the ICN/CCN and significantly reduces the network traffic and average download delay.

2.4.6 Summary

Table 2.3 shows an overview of techniques that deal with network congestion and resource availability problem.

Category		Key Contributions	Cons
Cafe [55]	Congestion Control	• Retentiveness heuristic: manage in-network delay. • Receptiveness: manage resource availability • Congesting rate • Ego network resources	• No collaborative cognitive caching (criterion 3) • Limited spatial-temporal locality support (criterion 6)
SocialCast [58]	Multipath Routing	• Publish-subscribe model that utilises social metrics (i.e. colocation and mobility patterns among communities) to identify multiple best relay nodes.	• No latency aware (criterion 7) • No collaborative cognitive caching (criterion 3) • No congestion control (criterion 4) • No spatial-temporal locality support (criterion 5)

CafRep [55]	P2P Replication	• Utility function of nodes to control the message replication rate such that the replication limit grows to take advantage of all available resources, but backs-off when congestion increases. • Ego network resources	• No collaborative cognitive caching (criterion 3) • Limited spatial-temporal locality support (criterion 6)
GeoCluster [62]	Clustering	• Contents are adaptively cached at the anchor location and making it available to other nodes within a maximum distance from this location.	• No mobility support (criterion 2) • No congestion control (criterion 4)
CafNC [64]	Network Coding	• Each node uses combined local heuristics on social, resource and content metric to detect and exploit coding opportunities. • Estimate of congestion levels and content distribution for coding decisions.	• No collaborative cognitive caching (criterion 3) • Limited spatial-temporal locality support (criterion 6)

Table 2.3 Overview of techniques that support congestion control and resource awareness.

2.5 In-network opportunistic caching techniques in mobile opportunistic networks

Traditional cache mechanisms (e.g. Web, P2P, CDN caching) are dependent on the application layer and designed for one traffic class. Typical in-network content caching [16] leverages in-network storage to make the caching more transparent to applications in which content's name are unified and consistent in order to achieve cross-application caching systems. The naming of content objects is either hierarchical or self-certifiable [16-18].

In this section, we describe the state-of-the-art in-network content caching specifically developed for our problem domain - mobile opportunistic disconnection-tolerant networks. We discuss what we have learnt from the

existing caching mechanisms and outline the reasons why these work does not solve our research problem.

2.5.1 Utility-based Cache

Utility-based cache focuses on how to improve a particular utility function measured by nodes in the network. The utility function can be defined based on different network characteristics (e.g. latency, resource availability, etc.) assuming that higher utility implies higher network performance for a node itself and for global networks.

Authors in [26] proposed a low-complexity heuristic FairCache algorithm that models collaborative caching as a bargaining game to address the fairness problems in fixed environments. [26] suffers from multiple limitations when applied to the mobile dynamic heterogeneous context such as it does not support dynamic content request patterns as well as congestion control, latency awareness, resource awareness nor considers the trade-off between caching performance and overheads which is necessary for more realistic scenarios (i.e. note that optimal solution for the bargaining game requires either global knowledge which is infeasible in dynamic topology or massive messages exchanged between individuals).

Research in [42] proposed utility-driven caching, where each content is associated with a utility, which is a function of the corresponding content hit probability, then formulate optimization problems where the objectives are to maximize the sum of utilities over all contents. Authors in [43] introduced a novel non-cooperative game model to characterize the caching problem among selfish nodes without any central coordination. However, both [42] and [43] solutions achieved the global optimum which is shown as being outperformed by collaborative caching [28].

2.5.2 Social Cache

Previous research [18] have studied the effect of cache centrality, cache space and cache resource allocation to the network performance. The authors argued that cache space allocation affects the caching performance, showing that cache space should be allocated based on nodes' social centrality. On the other hand, allocating more resource to the edge rather than the core (central) nodes will

benefit the caching performance. The authors also show that the cache hit ratio is proportional to the size of cache allowing more content to be cache. However, as in-network caching mechanisms should operate at line speed, large cache space adds lookup computational overhead.

Authors in [44] proposed a forwarding and cache replacement policy for Social Cache based on the nodes' centrality in the networks and the content popularity driven by frequency and freshness of content requests. As part of its replacement, Social Cache may remove a cached content from the network, thus reduce the cache hit ratio and increase delays. This problem is exacerbated when the resource is limited and the replacement rate is high as it is not resource nor congestion aware.

Authors [66] developed a model for detecting circles that combine network structure as well as user profile information. For each circle, the authors learn its members and the circle-specific user profile similarity metric. Modelling node membership to multiple circles allows detecting overlapping as well as hierarchically nested circles. However, as the proposed solution is social heavy-based, it does not support the potential network congestion in highly central nodes or region of nodes in ego (neighbouring) networks.

2.5.3 Location Cache

In location-based cache, the contents are cached at some specified node(s) based on a particular function. In traditional caching systems, cache locations are often predetermined. Traditional Web cache usually locates the content copies along the path from the source to the request, while CDN cache proactively copies the content to the edge based on prior knowledge of workload. In content networks, content caches are ubiquitous as there are no fixed caching points [67]. Intuitively, content cache in ICN for fixed topology aims to move the cache closer to the subscriber which reduces delay and traffic load. However, in mobile and dynamic topology, the cache location becomes very challenging and less clear with both mobile subscribers and publishers as mobility adds difficulties to the mathematical model of the caching coordination. Furthermore, caching techniques are moving from file level to chunk level that

makes it possible to retrieve different parts of the same file from different caching locations (e.g. similar to P2P caching technique).

Research in [62] focused on geo-based content sharing where content items are cached at the anchor location and making it available to other nodes within a maximum distance from this location. However, it takes into account only a subset of mobile devices actively using GPS. Besides, error-prone GPS measurements reduce the performance of content sharing in many scenarios. Moreover, there is no guarantee that content “posted” to an anchor zone will remain available due to the mobility of users.

Authors in [68] utilised entropy in their caching decisions to measure mobility prediction uncertainty and locate the best prefetching node, thus eliminating redundancy and improving cache resource efficiency. The authors showed that the increase in latency does not negate the performance gains of proactive caching. Moreover, the gains are amplified by the reduction in server load and cache redundancy achieved, thus improve the cache efficiency. However, [68] assumed the assist of infrastructure and does not support the disconnection tolerant and dynamic changing network topologies due to the mobility of content publishers and subscribers.

2.5.4 Popularity-based Cache

Previous studies on in-network caching technologies, such as Web caching have highlighted the dependency of caching policies on the basis of access time and distribution of content requests which requires a need for an intelligent adaptive predictive content popularity resolving mechanism. Intuitively, more popular contents will satisfy a higher number of content requests, thus should be cached more frequently (i.e. have more replicas) in order to improve the overall performance of caching solutions.

Authors in [69] proposed a network model to study cooperative caching in a setting with two characteristics: 1. delay-tolerant; 2. lack of infrastructure. The difference in popularity of different web content induces a non-uniform request distribution, thus derive the optimal caching scheme using convex optimization techniques. However, the proposed algorithm in [69] could not be able to capture in real time the multilayer interdependences of traffic patterns with

mobility and resources as they rely on an assumption that the content distribution in the networks approximately follows Zipf's law.

Research in [70] showed the effects of one-timer contents on the performance of a caching policy by increasing both the content retrieval latency and network traffic. The authors assumed fixed network topologies, thus do not consider the mobility support for their proposed caching algorithms.

Authors in [71] designed a simple approach for estimating the hit rates of a cache operating the least recently used (LRU) replacement policy. As the authors rely on an assumption that the content distribution in the networks approximately follows Zipf's law, the proposed caching solutions could not capture in real time the multilayer interdependences of content traffic patterns with mobility and resources.

2.5.5 Probabilistic Cache

Probabilistic caching policies rely on probability function to make their decision on creating a replica on a node. Depending on the structure of this probability, whether it is a fixed value or a mathematical formula consists of a number of individual components.

Authors in [72] designed an approach to computing the hop-based caching probability with a small amount of overhead. Then providing a caching weight based on the mean residence time of the cached contents, which guarantees that the cached contents remain in a router for an expected duration.

Authors in [73] proposed an on-path in-network caching scheme ProbPD, based on two variables, the contents popularity rates and the distance ratio of each node from the source, using the performance metrics of cache hit rates, cache eviction rates, hop counts and content delivery times.

Both [72] and [73] rely on centralised infrastructure with global network knowledge but do not support the dynamic mobility, connectivity of mobile publishers and subscribers.

2.5.6 Label Cache

In label cache, the nodes are able to cache a specific range of contents. Each node is pre-assigned a range of labels as well as aware of the range of labels of

the remaining nodes within a network. Thus, label-based caching constitutes a form of cooperative caching with a centralised control unit.

[74] proposed Hybrid Symmetric-Multicast Hash-routing (HR Hybrid SM) and Hybrid Asymmetric-Multicast Hash-routing (HR Hybrid AM) protocols that require edge-domain routers to calculate a hash function which maps content identifiers to cache nodes. Then, the schemes select the most efficient content caching and forwarding strategy based on the location of source, cache and receiver.

Research in [75] designed a cooperative caching strategy for the treatment of large video streams with on-demand access.

2.5.7 Single Cache & Collaborative Cache

A novel non-cooperative game theory model is introduced in [43] to characterize the caching problem among selfish nodes without any central coordination. However, the caching algorithm achieves the global optimum which is shown as being outperformed by collaborative caching algorithms.

Fairness [26] explored the topic from a utility-based game theory perspective. Intuitively, in a bargaining game theory, caches would wish to engage in a collaborative algorithm if they attain greater utility than if they were not to engage. This observation mandates some concept of fairness, where benefits are spread fairly across caches, and individuals are not expected to sacrifice personal performance by collaborating.

Regarding the incentives and weighting process between collaborator, authors in [76] proposed a reputation framework based on mobility and centrality to improve security in highly mobile, sparse and disconnected environments. Authors integrated centrality, local observations, selective deviation test and adaptive timer techniques in the reputation model to improve efficiently node monitoring and trust resolution. [77] proposed a reputation aware adaptive obfuscation in mobile opportunistic networks based on implicit reputation building, complex graph connectivity analytics and obfuscation history analyses. The authors presented a testing scheme and hierarchical reputation states to enable an accurate, intelligent and efficient fully distributed reputation building based on the nodes' obfuscation behaviour. [78] proposed an

intelligent cooperative sensing framework that enhances the reliability of the manufacturing plant in the disconnection prone heterogeneous sensor networks. Each node has its own trust value which is resolved based on the weighted sum of multiple nodes' observations of the node. The framework identifies and uses nodes that have both high centrality and reputation to improve the reliability and accelerate reputation convergence while being able to detect and isolate faulty nodes that have low trust value.

An ego network of a node is defined in [50] as its neighbours consisting of that node together with the nodes they are connected most recently (i.e. k-hops neighbours) or nodes it meets at regular interval (most frequently) over the time duration and all the links among those nodes. Consequently, ego network analysis can be performed locally by individual nodes without complete knowledge of the entire network, allows each node to give its own regional or temporal perspective of the network (or both are included).

Collaborative caching requires a significant amount of exchanged messages to gather useful information about a localised region in the network. [79] proposed a cognitive-based data dissemination algorithm with an aim to reduce the number of content replicas between caching nodes without mitigating the loss of performance in terms of content diffusions. The algorithm prefers to replicate new contents which are only defined as contents a node has never seen before, thus results in lack of granularity. In addition, the authors do not consider scenarios of potentially losing useful contents due to one-timer contents or flash crowd problem. The proposed algorithm defines how popular a node and a content is based on the counter of frequency (either contact frequency or request frequency) that could not be able to capture other characteristics of node centrality and content popularity. Moreover, an aggregate measure together with the stochastic approach in recognition process results in a loss of granularity with respect to the information about the single content diffusion

2.5.8 Summary

In general, caching content in heterogeneous mobile opportunistic networks can be performed in two different approaches: operations on the unit of file and operations on the unit of chunks [80, 81]. Although most existing caching

proposals work on the file-unit, recent caching policies perform on the level of chunks. Traditional file-based caches are typically based on the assumption of the Independent Reference Model (IRM) [82] that the probability of requesting an object is determined only by its popularity, and not dependent on the previous requests. The object popularity has been received extensive studies (e.g. access frequencies of Web, P2P objects follow Zipf distribution) that many caching proposals assume Zipf distribution model for the content workload. However, both traditional file-based caches and object popularity model cannot be applied to chunk-based cache and recent object popularity. Firstly, requests of different chunks of the same file are correlated and dependant to each other (e.g. users often decide to whether watch a whole video based on its beginning). Secondly, recent studies have shown that content popularity has changed during the past few years, thus file-level content popularity model cannot be easily extended to chunk level.

In addition, cache transparency poses a requirement on the line-rate operation speed of caches. Moreover, different traffic types of different applications require different cache objectives (e.g. text vs. video) and cache management. Cache transparency to applications results in the sharing between different types of traffics and applications. Differentiate caching services has been studied by previous research, ranging from fixed sharing to dynamic sharing. [83] proposed an adaptive fixed partitioning of cache space for different traffic classes. The authors suggest that cache sharing based on fixed partition not only relies on dynamic feedback of workload controlled by a node but also on the position of that node (e.g. dedicating cache closest to users to VoD applications). However, fixed partitioning techniques are hard to achieve the optimal solution, such that some traffic classes will run out their cache space, whilst others will have unused cache space at all time.

Table 2.4 shows an overview of in-network opportunistic caching techniques in heterogeneous mobile opportunistic networks.

	Considered Factors	Fully-Localised Predictive in Real-Time	Mobility & Connectivity Awareness	Content Awareness	Resources Awareness
Social Cache [44]	• Node social-tie relationship • K-means node cluster • Content request frequency & recency	Yes	Yes	Yes	No
Fair Cache [26]	• Fairness in cache game • Utility function • Heuristics: cutting out unpopular contents, distant nodes.	Yes	Limited	Yes	Limited
Cluster-based Cache [68]	• CC-GA Cluster in social networks • Influential nodes and popular content by eigenvector centrality	Limited	Yes	Yes	No
HR Hybrid SM [74]	• On-path caching • Hash-routing • Path stretch	No	No	Limited	No
Chunk Cache [80]	• Chunk level • Different popularity between early chunks and later chunks of same content. • Link bandwidth to offload content to vehicles.	No	Limited	Yes	Yes
Cognitive-based	• Recognition heuristics: frequency and	Yes	Limited	Yes	No

Cache [79]	recency counter • Modified Take The Best: popularity of content depends on popularity of nodes holding it.
---------------	---

Table 2.4 Overview of in-network opportunistic caching techniques in mobile opportunistic networks.

2.6 Techniques to support capturing spatial-temporal locality and multilayer interdependences in mobile opportunistic networks

Existing content caching policies rely on an assumption that the popularity distribution of the network contents approximately follows Zipf's law and the characterisation of content requests have been based on Independent Reference Model (IRM) which assumes that content requests occur independently. However, recent researches on today's social networks [24, 25, 31, 32, 84, 85] have questioned the validity of the IRM model and Zipf's law model, argue that content requests often exhibit both temporal and spatial locality, thus requires next-generation content popularity resolving approach that is able to adaptively predict and capture the spatial-temporal correlation in content requests to support more reliable, adaptive, cognitive caching algorithms.

However, the above approaches either i) rely on empirical analysis (offline data analysis) to capture mobility and traffic spatial-temporal patterns, thus not able to support fully-localised predictive decisions in real time or ii) could not adapt with the dynamic changing mobility and intermittent paths between publishers and subscribers as their analysis on spatial-temporal correlation are based on archon location.

Table 2.5 shows an overview of the techniques that deal with spatial-temporal locality and multilayer interdependences in mobile opportunistic networks.

	Theoretical or Empirical	Fully- Localised Predictive in Real-Time	Dynamic Mobility & Locations	Dynamic Content Request Patterns	Dynamic Resources
[25]	Theoretical	No	Limited	Yes	No
[31]	Empirical	No	Yes	Yes	No
[84]	Empirical	No	Limited	Yes	No
[24]	Empirical	No	Yes	Yes	No
[32]	Empirical	No	Limited	No	No
[85]	Theoretical	No	Limited	No	No

Table 2.5 Overview of the techniques that deal with spatial-temporal locality and multilayer interdependences in mobile opportunistic networks.

2.7 Summary

In Table 2.6, we systematically outline core criteria that we used to review the related work and guide my research for a distributed multi-layer real-time predictive and adaptive forwarding and caching framework in mobile opportunistic (disconnection-prone) networks. Table 2.6 presents a summary of the techniques discussed in this chapter in terms of the ten criteria, showing that none of the existing approaches provides support for all criteria: fully-localised predictive decision making (1), adaptive forwarding (2), adaptive replication (3), collaborative caching (4), congestion awareness (5), social contact awareness (6), resource awareness (7), disconnection tolerance due to dynamic mobility (8), latency awareness (9) and spatial-temporal correlation awareness (10).

	Fully Local Pred (1)	Adpt Forwd (2)	Adpt Repl (3)	Collab Cache (4)	Cogst Awr (5)	Soc. Awr (6)	Res. Awr (7)	Dis/ Dyn Topo (8)	Delay Awr (9)	S-T Corr Awr (10)
Behave [19]	No	No	No	No	No	No	No	No	No	No
CGr Ant [53]	Yes	Yes	No	No	No	Yes	No	Yes	No	Yes
Social Cast [58]	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No	Yes
Region Cache [20]	No	No	No	No	No	No	No	No	No	No

Chunk Cache [80]	Yes	Limited	Yes	Yes	No	No	No	Yes	No	No
Social Cache [44]	Yes	Yes	No	Yes	No	Yes	No	Yes	No	No
Focal [40]	No	Yes	No	Yes	No	No	Yes	No	Yes	No
Hy Mobi [22]	Yes	Yes	No	No	No	Yes	Yes	Yes	No	No
CafRep [60]	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No

Table 2.6 Overview of the techniques that deal with efficient content dissemination and query in mobile opportunistic networks.

This chapter has reviewed some of the most relevant techniques and approaches according to the criteria given in section 2.1. We argue that there is no single approach that satisfies all the criteria and we need some kind of combined approach that would provide the best support for all the criteria.

The model proposed in the next chapter attempts to satisfy all of the ten criteria and to reconcile the challenges and conflicts in some criteria. More specifically, the model proposes a way of answering the following questions: Where and when to cache contents in dynamic challenged environments? What contents to be cached? and How to manage the caches efficiently? As the networks are becoming more dynamic, centralised solutions will not be scalable in mobile heterogeneous networks due to high complexity and a single point of failure problem [86, 87]. Distributed solution techniques may still cause high overheads and cannot adapt sufficiently to the time and space dynamics of mobile networks compared to real-time fully-distributed predictive edge analytics and heuristics based solutions [26, 27]. In addition to this, non-cooperative caching which is close to the current Internet architecture generally neglects all the content networks benefits [28, 42]. On the other hand, explicit cooperation is hard to achieve and the attempts had negative experiences. Thus, collaborative caching has been shown to usually outperforms both locally and centrally optimized algorithms [28, 42]. The next chapter proposes a novel fully-distributed collaborative adaptive real-time heuristic-based multilayer caching framework.

Chapter 3

3. Collaborative Cognitive Predictive Content Caching Framework

This chapter presents CafRepCache, an intelligent predictive collaborative adaptive caching framework in mobile opportunistic networks. We propose a novel spatial-temporal predictive analytics and heuristics of users' content requests to answer the fundamental multi-attribute decision problems: how to decide what contents to be cached and how to identify the most appropriate carriers for different contents to bring contents as close as possible to their subscribers. Our novel caching service is built on top of state-of-the-art CafRep (Congestion-Aware Forwarding and Replication) [55, 60, 88] which, in turn, is built on SimBetTS (Similarity, Betweenness, Tie Strength) social forwarding [50, 60] in order to balance the trade-offs between maximising content retrieval success ratio and minimising content retrieval latency in dynamic mobility, topology changes while enabling resource-aware congestion control and avoidance.

Fig. 3.1 illustrates the scenarios of the end-to-end opportunistic content retrieval between mobile content subscribers, publishers and caching points in heterogeneous mobile opportunistic networks. The end-to-end *content retrieval* process starts from when content subscribers sending one or many content requests to the networks where the relevant content publishers or temporary caching points should be reached possibly and next, contents must be retrieved from these nodes to the subscribers. The multimedia requested contents may vary from text, pictures to web pages or videos. More specifically, when a subscriber initiates a search query which is carried physically by the mobile nodes and forwarded to other nodes upon contact, we refer to this as *content discovery* as time taken between sending interest packets to the networks and discovering a requested content [10]. Compared to the conventional web search, opportunistic content discovery is nontrivial and increasingly complex due to two major challenges: lack of a central database to store any content index and time-varying network topology [45]. After the query reaches one of the

publishers or caching points, a response generated is routed towards the subscriber in the second step. We refer to this as *content delivery* as time taken for the content to be delivered successfully to the interested subscribers from the time content was discovered in the network [10]. Regarding the amount of data carried in the query and response, the latter is expected to be (much) larger in size [45]. Considering the predominance of video traffic, response messages carrying the requested content would be much larger compared to the search query with few keywords. In contrast, for other queries, e.g., a train schedule, responses may have a comparable size to that of queries. Content services to discover and retrieve the contents in heterogeneous mobile opportunistic networks are paramount for a wide range of applications [60, 89, 90] such as content retrieval in a dense network, e.g., search for a festival’s program during the event, or in a sparse network, e.g., for the shuttle hours to a city in a summer cottage. In the challenging inherent uncertainty of opportunistic networks, CafRepCache aims to support: i) finding relevant contents with high probability and ii) delivering contents quickly to subscribers before the user loses interest or the content becomes stale.

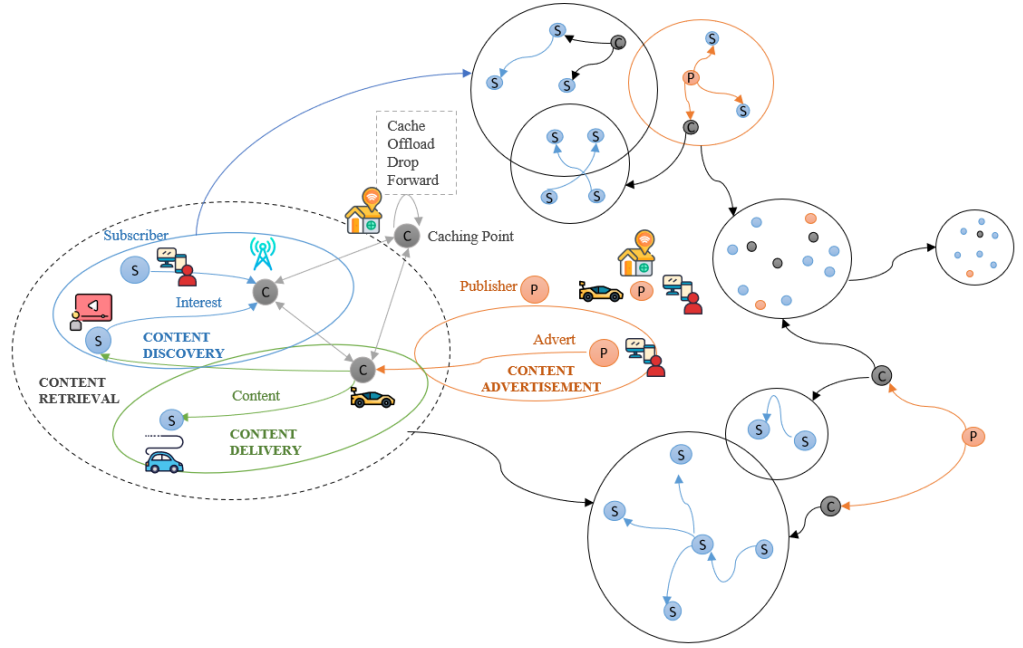


Figure 3.1 Content Retrieval: Content Discovery and Content Delivery.

The remaining of this chapter is as follow: section 1 discusses the foundation theory of CafRepCache, section 2 and 3 explore the design space and introduces

the overview of the proposed framework. Section 4 provides the functional model of the approach. Section 5 discusses real-time predictive content heuristics of CafRepCache. Section 6 describes the CafRepCache pseudo code in details while the section 7 gives conclusions and a short summary of the chapter

3.1 CafRepCache Foundation

In CafRepCache, we build up our novel content caching layer over two underlying networking layers: social connectivity/mobility [50, 60] and resources [55, 60, 88]. More specifically, we use state-of-the-art social heuristics: similarity, betweenness and tie strength proposed in SimBetTS [50, 60] to drive CafRepCache social forwarding decisions. They are shown as the most appropriate metrics for message transfer to select best next-hop nodes [50], [91] because they allow ranking known contacts and predicting future contacts (similarity [50, 60]), analysing and predicting information flow (betweenness centrality [50, 60]), identifying and forecasting future paths in the networks (tie strength [50, 60]) to find the higher centrality nodes for caching and forwarding contents in order to increase the higher chances of requested contents being successfully delivered to subscribers [50, 60]. In addition to this, we use state-of-the-art resource heuristics of network's buffer availability (i.e. retentiveness [55, 60, 88]), in-network delay (i.e. receptiveness [55, 60, 88]) and rate of congestion proposed in CafRep [55, 60, 88] in order to make better congestion-aware resource-aware forwarding and caching decisions as they build a strong foundation to provide a trade-off between shortest path and resource-driven path in order to redistribute load to avoid congestion [55, 60, 88]. The definition of social [50, 60] and resource [55, 60, 88] analytics and heuristics that we use in CafRepCache framework is not limited to a static combination but rather our caching layer can be built on any other underlying social and resource heuristics as well.

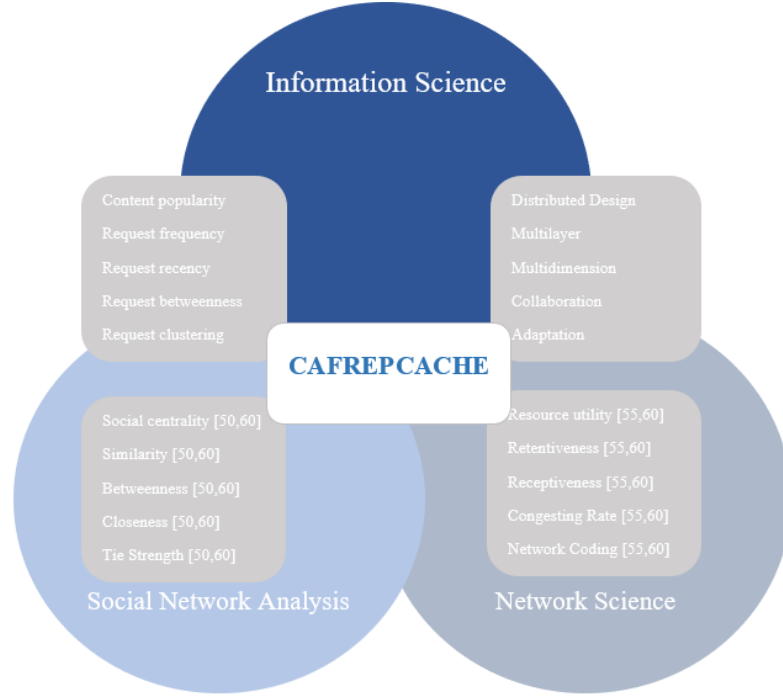


Figure 3.2 CafRepCache's theory foundation.

As shown in Fig. 3.2, CafRepCache caching proposal combines three theory foundation: network science [92], social network analysis [50, 52] and information science [92]. Our proposal is built on network science [92] that utilises graph theory to describe different states and processes on our networks as complex temporal network graphs [93]. Social network analysis [50, 52] helps us to understand the network characteristics (e.g. load distribution, shortest social-driven resource-driven path) and network evolution where human, vehicles, etc. are involved [6, 94]. In addition, while the networks are non-uniformed and heterogeneous, information and contents in the networks are also different from each other and not uniformly distributed. We rely on information science [9, 92] to analyse the characteristics of content, content requests, publisher/subscribers and study the interdependency between contents and underlying networks [11, 25, 31].

3.2 CafRepCache's Multi-layer Multi-dimensional Design Space

We propose a novel multilayer multidimensional design space for CafRepCache based on its multilayer complex theory foundation and describe all of its dimensions to enable adaptive real-time predictive collaborative caching in mobile opportunistic networks (Fig. 3.3). The core dimension of our design

space is the novel content dimension which captures and predicts the spatial-temporal locality of content traffics as the content is not requested scattered randomly and independently over time; but rather particular contents are interested at a certain time interval and in geographically close areas. Our CafRepCache's design philosophy is to extend the idea of behavioural locality to exploit similarities between the content interests and users mobility. We propose that edge nodes use CafRepCache algorithm to form dynamic transient interest-based and data dissemination topologies based on predictive analysis and commonalities between their interest, cache and retrieval histories as well as connectivity histories. This provides each node with a set of overlay neighbours whose browsing history most closely resemble their own. CafRepCache emerges from this topology as the federation of the local caches of a node's ego network and the closest available nodes.

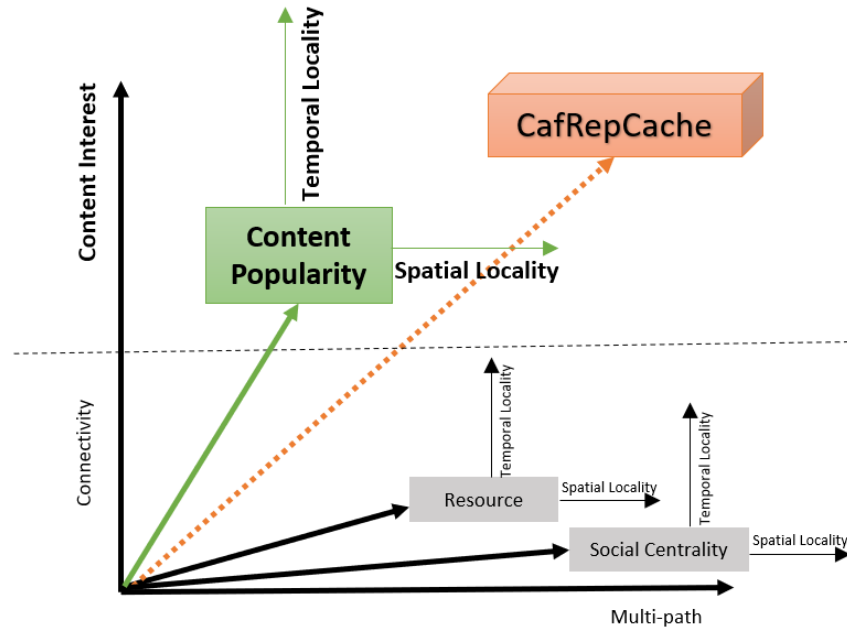


Figure 3.3 CafRepCache's design space.

3.3 CafRepCache's Multi-layer Framework

Figure 3.4 shows a multi-layer view of CafRepCache conceptual model which consists of: Physical Layer, Mobile Disconnection Tolerant Network (Mobile DTN) Layer, Content-Centric Network (CCN) Layer and Application Layer [10] where each layer is able to perform simultaneously multiple complementary predictive analytics and heuristics. Each CafRepCache's layer

addresses different optimization problems to produce complex trend analytics without the expense of other layers by capturing the interdependences between multiple layers and dealing with multiple complex inherent trade-offs in order to achieve cross-layer multi objectives in different contexts. At the heart of CafRepCache is our novel content caching layer with predictive content analytics and heuristics built on top of state-of-the-art resource [55, 60, 88] and social forwarding layer [50, 60]. CafRepCache’s multilayer multidimensional predictive analytics and heuristics fill up, complete each other and when combined together, they enable better caching decisions, provide higher delivery success, lower delay and resource consumption when serving dynamically changing users’ demand patterns in dynamic mobile networks.

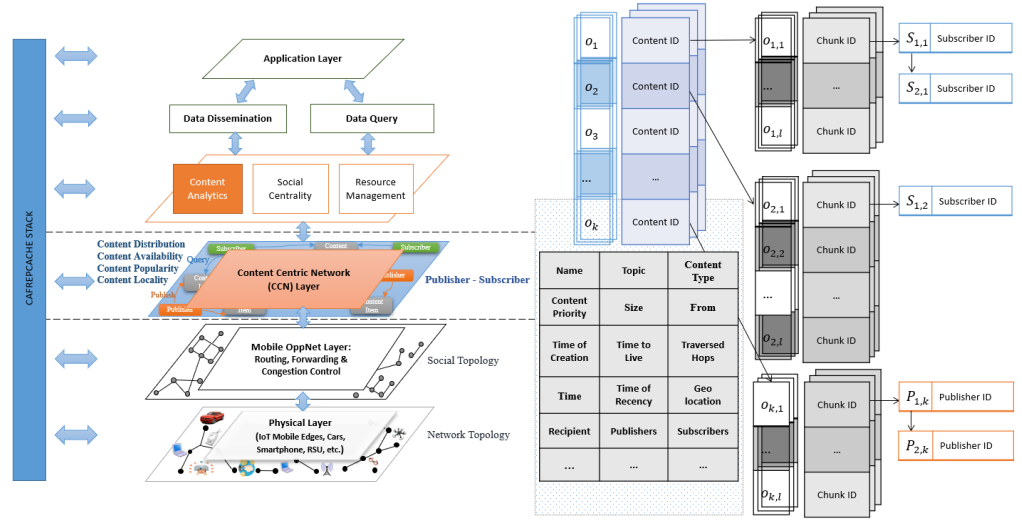


Figure 3.4 Multi-layer CafRepCache framework

More specifically, in physical layer, CafRepCache nodes are represented as mobile edge-cloud devices (e.g. IoT, smartphone, vehicles, drones, etc.) or static devices (e.g. Road Side Units) with different dynamic physical constraints (e.g. storage, computational resources, energy, etc.). In the network layer, we consider scenarios where nodes are coupled with users whose connectivity forms either random [95] or scale-free networks [11, 50, 52]. Typical scenarios of random networks are natural disaster events or mobile crowdsourcing systems while that of complex temporal scale-free networks are urban cultural events or public transport systems which represent different degrees of human social behaviour following power-law connectivity patterns. In the content layer, information are generated by mobile publishers and requested by mobile

subscribers; publishers and subscribers may be mobile or static devices in the network. We consider that each content has a list of static and dynamic attributes (e.g. name, topic, type, geo/location, time, etc.) in which the knowledge on content popularity, content locality, content distribution and content availability are not known in advance.

As shown in Fig. 3.5, when two CafRepCache nodes are in contact, they exchange a list of novel temporal-spatial predictive analytics of content popularity (together with node connectivity [50, 60] and resource availability [55, 60, 88]) they observed locally. As CafRepCache nodes have fully-distributed behaviour with no global knowledge, they base their caching decisions on the knowledge gathered in their local and regional environment. Each node continuously monitor, analyse and predict the value of dynamically changing predictive analytics in real time while collaborating and exchanging their observations, analysis and predictions with others. Our proposed reputation-aware weighting mechanism help CafRepCache caching points efficiently utilize the exchanged values aggregated from other nodes through collaborative process, thus enhance the accuracy of its predictive analytics.

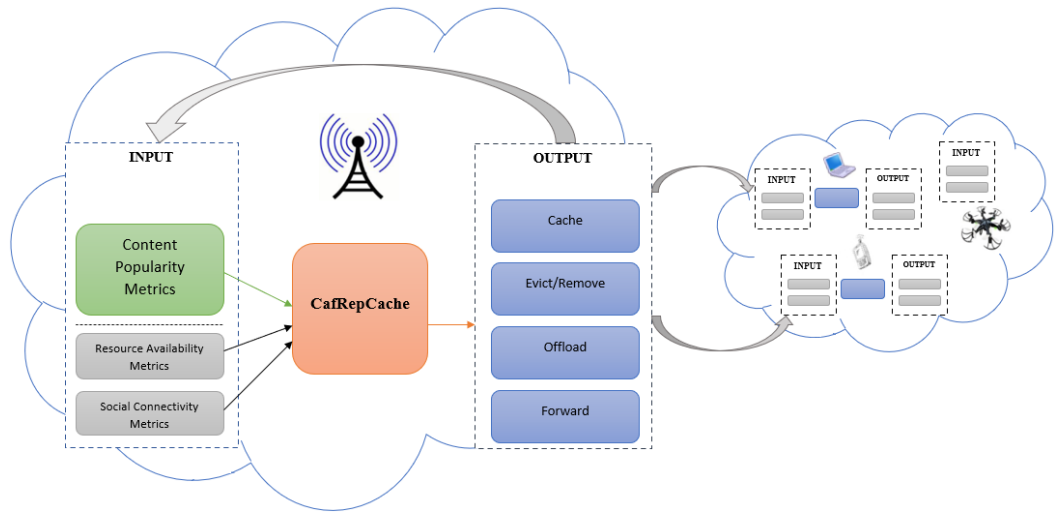


Figure 3.5 Input and output of CafRepCache caching framework.

Fig. 3.6 shows the internal modules of CafRepCache. Various input data such as content interest, content data, content request metrics are fed to the dynamic caching modules where CafRepCache node then calculates the combined spatial-temporal predictive popularity for each content based on the request temporal heuristics and spatial heuristics. The temporal heuristic values help to

determine the popularity of content over time based on content request frequency, recency and inter-event request betweenness while the spatial heuristic values helps to determine how close the current caching node is to the subscribers of requested content. We assume temporal heuristics and spatial heuristics have equal weight. Based on the sum value of temporal heuristics and spatial heuristics, CafRepCache node then decides whether it will be a caching candidate for specific content: content is cached if it relatively has higher predicted popularity (i.e. high popularity over time and its subscribers are relatively close to the current caching node) compared to other cached contents in CafRepCache's caching buffer. When the caching buffer is full, CafRepCache node decides which content to be evicted/removed from its buffer, then simply drop the least popular content or offload it to the neighbouring nodes. The social and resource modules make sure that the content will be offload or forwarded to neighbours that have high caching buffer availability while still being highly central and relatively close to its subscribers. While CafRepCache continuously adapts and responds to the changing local conditions, its future state is also affected by its environment and the state attained in its own previous decisions. CafRepCache nodes are able to adapt quickly when the information is updated as well as modify their own behaviour when getting feedback from remote network state.

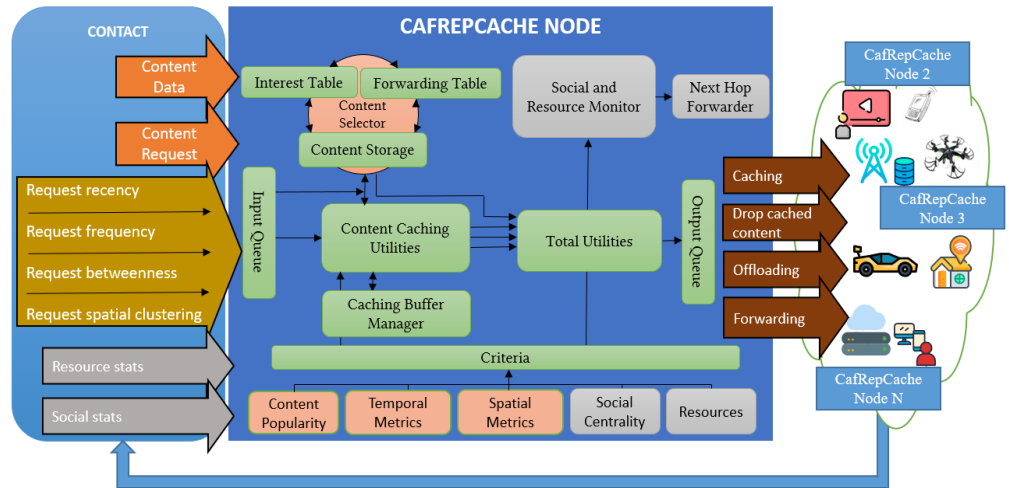


Figure 3.6 CafRepCache's integral components.

3.4 CafRepCache Analytical Model

This section describes the formal analytical model of our collaborative cognitive content caching in mobile disconnection-tolerant networks which relies on the identified design space and architecture.

3.4.1 CafRepCache collaborative caching as bargaining game

In game theory, the bargaining game is a model [26] aims to understand better how individuals collaborate when they target to reach a mutually beneficial agreement. We model our CafRepCache algorithm as a bargaining game where nodes reach an agreement on how to efficiently share the utility gained by collaborating with others.

Definition 1. An in-network caching problem is a pair $\langle U, (u_1^0, u_2^0, u_3^0, \dots, u_i^0) \rangle$ where $U \subset R^N$ is a compact and convex set, containing all values gained via collaboration and u_i^0 is an initial disagreement value defined as the worst utility value payoff a node would accept for collaboration.

Definition 2. A fair caching solution is a function $f: U^e \rightarrow \varphi, U^e \subset U$ is the Pareto frontier of set U such that $(u_1^*, u_2^*, \dots, u_i^*) = f(U, u_1^0, \dots, u_i^0)$.

A cache solution is fair if it satisfies the axioms imposed:

- Individual Rationality: $u_i^* > u_i^0$
- Feasibility: $u_i^* \in U$
- Pareto efficiency: If $u_i, u_i' \in U, u_i < u_i'$ then $f(U, u_i^0) \neq u_i$
- Symmetry: If $(u_1, u_2) \in S \leftrightarrow (u_2, u_1) \in S, u_1^0 = u_2^0$ then $u_1^* = u_2^*$
- Independence of irrelevant alternatives: If $(u_1^*, u_2^*, \dots, u_i^*) \in U' \subset U$ then $f(U', u_1^0, \dots, u_i^0) = f(U, u_1^0, \dots, u_i^0) = (u_1^*, u_i^*)$
- Independence of linear transformation: Let $u_1' = c_1 u_1 + c_2, u_2' = c_3 u_2 + c_4, \dots$ then $f(U', c_1 u_1^0 + c_2, \dots, c_x u_i^0 + c_{x+1}) = (c_1 u_1^* + c_2, \dots, c_x u_i^* + c_{x+1})$

Theorem 1. There is a unique solution called Nash Bargaining Solution (NBS) that satisfies all the axioms above:

$$\arg_{(u_1, u_2, \dots) \in U, u_i > u_i^0} \max \prod_{n_i \in N} (U_i - u_i^0)$$

The symmetry axiom implies that all players are equal in the bargaining game. However, to be more realistic, we argue that some nodes have priority over others, thus we modify the objective function to:

$$\arg_{(U_1, U_2, \dots) \in U, U_i > u_i^0} \max \prod_{n_i \in N} (U_i - u_i^0)^{w'_i} \quad (1)$$

in which w'_i is a weighing parameter that values each node's priority.

Note that optimal solution for the bargaining game requires either global knowledge which is infeasible in dynamic topology or massive messages exchanged between individuals [9, 10, 26].

3.4.2 Complex graph theory for network and mobility

We model CafRepCache system as a network G that consists of a set N of nodes n_i ($n_i \in N$) and a set E of edges, $G = (N, E)$. As the connectivity of the network and the state of the nodes change over time, we model each of these sets as time series, thus $N = \{N^t : t \in T\}$ and $E = \{E^t : t \in T\}$.

We define “ego network” of each node n_i : EN_i as a network consisting of n_i together with the nodes they are connected most recently (i.e. k -hops neighbours) or nodes it meets at regular interval (most frequently) over the time duration ΔT and all the links among those nodes. In this way, ego network allows each node to give its own regional or temporal perspective of the network (or both are included) [9, 10, 60].

We briefly describe two types of very different network topologies characteristics: complex temporal random networks and complex temporal scale-free networks. While fixed and wired networks are mostly scale-free [96], mobile and opportunistic networks can be either random [97] or scale-free [98]. We analyse its challenges by studying how likely that the interest request of a content is not satisfied (cache miss) in each topology that helps us to understand the worst case and best case scenarios of our caching proposal in the dynamic heterogeneity of mobile networks. In our proposal, we explore the encounter rate and the frequency that nodes contact each other in each network topology. The actual path along which a mobile node moves is out of our focus that simplifies our model.

In random networks, the majority of previous studies in mobile disconnection-prone networks assume *Poisson* contact processes and model user encounters as independent *Poisson processes* with rate λ due to the time between two consecutive contacts of a pair of nodes follows exponential distribution [95]. The probability of some node $n_i \in N$ connecting with some other node n_j [95] is: $P(n_i) = \frac{1}{N} \int P(v_{ij}) dn_i$. The probability that node $n_i \in N$ connects with exactly n other nodes [95] within time T : $P(n \text{ encounters}) \approx \frac{e^{-\lambda T} (\lambda T)^n}{n!}$. Let p'_k be the probability that content o_k is stored in the cache of an arbitrary nodes, then the probability of the cache miss in random network is equal to: Prob(content o_k is not in the cache) * Prob(interest request never reaches correct caching points within time T)

$$= (1 - p'_k) * \sum_n P(n \text{ encounters}) * P(\text{none of the } n \text{ encounters has requested contents})$$

$$= (1 - p'_k) * \sum_n \frac{e^{-\lambda T} (\lambda T)^n}{n!} (1 - p'_k)^n$$

Complex temporal scale-free networks are characterized by a highly heterogeneous degree distribution, which follow a power-law distribution [98]. Although the network may change significantly over time, the degrees of its nodes obey the power-law model at any time [98]. The probability $P(n \text{ encounters})$ of a node in the network goes for large values of n as: $f(n) \sim n^{-\gamma}$ where γ is the shape parameter of the power-law distribution and represents the degree of the power-tail. Then the probability of a cache miss of content o_k in scale-free networks is $(1 - p'_k) * \sum_n n^{-\gamma} (1 - p'_k)^n$.

3.4.3 Contents

We assume that each CafRepCache node in the network $n_i \in N$ has a cache of size θ_i . We denote with O a set of content files that can be requested by the network. Each content $o_k^t \in O$ (or o_k for simplicity) is published at time t and has the size δ_k .

Content o_k consists of an array of chunks $o_{k,l}$. For simplicity, we assume all chunks $o_{k,l}$ of a single content o_k will have the same chunk size $\delta_{k,l}$ without losing generality. We also denote r_k^t as the interest about content o_k at time t .

We denote $P(n_i, o_k, \Delta t)$ as the popularity of content o_k observed by node n_i during the interval time Δt . Content popularity implies the probability of how likely a content will be requested in a period of time. Contents are globally popular if they have been requested by a high number of subscribers coming from different regions in the networks while localised popular contents are those which have been requested by subscribers from the same location. In mobile opportunistic networks where the locations of subscribers and publishers are dynamically changing while they are moving, we measure how similar the connectivity and mobility is between subscribers to justify whether they are in the same region or not. We assume a heterogeneous environment where not every subscribers are interested in the same content and not all publishers produce the same content.

3.4.4 Mobile publishers and subscribers in mobile opportunistic networks

At time t , a node $n_i \in N$ may act as either a subscriber of a content o_k (denoted by $S_{i,k}^t$) or a publisher ($P_{i,k}^t$) or a caching point ($C_{i,k}^t$). Thus, for any content o_k , we denote a set of subscribers who are interested in o_k as $S_k^t = \{S_{i,k}^t, n_i \in N\}$ and so on. A node can become a subscriber of a content and a publisher of another content at the same time. Similarly, a node can be a subscriber of a content which later becomes a caching point serving other nodes in the network.

Each node $n_i \in N$ in the network receives certain requests of content o_k at time t , denoted as local content request rate $q_{i,k}^t$. In addition to this, $z_{i,k}^t$ is denoted as the aggregated request rate of the content o_k observed from all the collaborative neighbours of n_i at time t .

When a node n_i receives a content o_k or a request/interest message of o_k at time t , there are four possible real-time decisions (i.e. actions) that the node can make according to the characteristics of the topology and the content:

1. cache the content, denoted as $x_{i,k}^t \in \{0,1\}$ referred to whether to cache or not.
2. forward content and interest message to its neighbours, denoted as $y_{i,k}^t \in \{0,1\}$.
3. offload/delegate the content, denoted as $\omega_{i,k}^t \in \{0,1\}$.
4. evict/remove the content, denoted as $\varepsilon_{i,k}^t \in \{0,1\}$.

Table 3.1 sums up the main notations.

Notation	Meaning
$G = (N, E)$	network G that consists of a set N of nodes and a set E of edges
θ_i	cache size of node n_i
O	a set of content objects that can be requested by the network
$o_k^t, o_{k,l}$	content o_k published at time t and consisted of multiple chunks $o_{k,l}$
$\delta_k, \delta_{k,l}$	size of content o_k and size of the chunks of o_k .
r_k^t	the interest of content o_k at time t
S_k^t	a set of subscribers who are interested in content o_k at time t
$q_{i,k}^t$	normalised request rate of the content o_k (i.e. content popularity) observed locally from n_i at time t
$z_{i,k}^t$	normalised aggregated request rate of the content o_k observed from all the neighbours of n_i at time t
$W(q_{i,k}^t, z_{i,k}^t)$	weights the value of collaborative observation over local observations
EN_i	the ego network of each node n_i
$x_{i,k}^t$	decision whether to cache content o_k in node n_i at time t
$y_{i,j,k}^t$	decision whether to forwards content o_k 's request to neighbour n_j at time t
$\omega_{i,j,k}^t$	decision whether to offload content o_k to neighbour n_j at time t
$\varepsilon_{i,k}$	decision whether to evict/remove content o_k at time t
U_i	aggregated utility value of node n_i
u_i^0	disagreement utility value of node n_i , defined as the value achieved by not collaborating with others

Table 3.1 Table of notations

3.5 CafRepCache's Content Heuristics: Predicting Content Popularity By Leveraging The Spatial-Temporal Locality of Content Requests

Existing opportunistic caching policies in mobile edge/fog networks such as [69] typically cannot capture and predict adaptively the spatial-temporal locality

of content requests needed for more accurate content popularity-based caching decisions because they rely on assumptions that content interests occur independently of users' mobility and resources [25, 85]. Higher popular contents receive higher number of content requests, thus should be cached more frequently (i.e. have more replicas) in order to improve the overall performance of caching solutions. We are motivated by the observation that if content caching predictions are not sufficiently fine-grained for different contents in different locations over time, they result in increased cache miss, delay and resource consumption [9, 10, 25, 85]. We propose a novel set of predictive analytics based on the characteristics of contents (i.e. Request Frequency, Recency and Betweenness) as well as the characteristics of network topologies (i.e. Request Clustering) [9, 10]. CafRepCache observes, captures, analyses and predicts multilayer multidimensional predictive analytics to make complementary content-based (i.e. choose higher popular content) and node-based (i.e. choose more important nodes) caching decisions. This allows CafRepCache to capture in real time the spatial-temporal locality in all layers (contents, networks, resources), proactively predict and bring high popular contents as close as possible to the subscribers, reduce the delay of high demanding traffic while still be responsive to less popular contents. CafRepCache's caching decisions are based on both local and collaborative predictive analytics and heuristics. Collaboration within ego networks [50] allows each node to share its own regional and temporal perspective of the network with the others. We consider the situation where a node gathering predictive analytic information observed by its neighbours differentiate between them according to their reputation and credit-based feedback [9, 10, 99].

3.5.1 Dynamic content request temporal heuristics

Each node in the network resolves the request frequency, recency and betweenness in fully-localised distributed manner. When two nodes are in contact, they exchange their local observations and continuously resolve the value of dynamically changing predictive heuristic based on both its local observation and the collaborative observations it gets from others. We introduce our method to measure the temporal heuristics based on time-series request frequency, recency and betweenness as below.

Request frequency counters are additively increased upon the arrival of a content request and decreased through time. This is to ensure that if the number of requests for a content has been reduced, its popularity counter will be reduced accordingly and the content will be subject to eviction or offload to other nodes. Given a caching point observes average f interests of content k during the interval Δt , the request frequency is measured as:

$RequestFrequency = f^{-\lambda \Delta t}$ (1) where λ is a control parameter. Request frequency implies that the more content requests have been observed by a caching point during a short interval time Δt , the more likely that content will be requested in the same interval, thus capture the temporal locality of content requests

Predictive analytic of request recency enables our caching design to capture the content popularity trend in responsive manner based on the recorded time stamp of recent requests in different locations for each caching point, thus allow to predict adaptively the emerging contents that may become highly popular in near future and contents that are currently considered as high popular but will be less popular soon. Given a caching point n observes the most recent interest of content k at $t_{recentReq}$, then we denote $t_{recentReq-1}$, $t_{recentReq-2}$, $t_{recentReq-3}$, etc. as the time that previous interests have been recorded by the caching point. The request recency is measured as how likely a recent content request will trigger a subsequent request at the current time.

$$RequestRecency = \frac{2}{1 + e^{\frac{\lambda(t_{currentTime} - t_{recentReq}) + \dots + (t_{currentTime} - t_{recentReq-f})}{f}}} \quad (2)$$

where λ is a control parameter. Predictive analytics of request recency shows that the smaller gap between current time and recent requests observed in the past of content k , the higher chance that content k will be requested soon in the future.

Request betweenness provides the trade-off between current observed content popularity versus long terms interest in it in order to balance between potentially one-timer contents or fake news [33] and long-term useful content. One-timer

contents are contents that have been requested once leading to possible cache pollution when one-timer contents are cached instantly that downgrades the overall performance of caching system. The time gap between continuous requests in the period of time Δt is measured as: $t_{currentTime} - t_{recentReq}, t_{recentReq} - t_{recentReq-1}, t_{recentReq-1} - t_{recentReq-2}, \dots$

Then the request betweenness heuristic is measured as:

$$RequestBetweenness = \pm \sqrt{\frac{1}{f} \sum_{i=1}^f (t_{recentReq-i} - t_{recentReq-i-1} - T_{averageTimeGap})^2} \quad (3)$$

Note that the term “betweenness” (centrality) has been widely used in OppNet [50] to measure the information flow in the network graph. In this thesis, the request betweenness refers to the time gaps between continuous inter-event requests.

The novel temporal heuristics are the combination of predictive analytics of *request frequency*, *recency* and *betweenness* which allow CafRepCache to choose the best suitable contents to cache and when to cache by predicting in real time the locality trend of content request patterns over time in different locations and avoid losing valuable contents by reducing the caches for one-timers contents [33].

$$TemporalHeuristics = Freq(o_k) + Rec(o_k) + Bet(k_k) \quad (4)$$

3.5.2 Dynamic content request spatial clustering heuristics

As the requests are more likely to be similar in a specific region [9, 10, 25, 85], we propose the fully-localised distributed spatial heuristics that aim to classify, recognize and predict content interests coming from dynamic changing clusters of subscribers. The spatial heuristic shows that the higher request rate coming from the same localised region or dynamic cluster of different subscribers, the higher likely that content will be requested again by other subscribers within that location [9, 10, 25, 85]. Given a caching point observes interest requests of content k from a set of subscribers $s \in S_k$ within a time interval Δt , the spatial heuristic is measured by the clustering coefficient [91], similarity, closeness and tie strength [50, 60] between different subscribers of the same content as below:

$$SpatialHeuristic = CCoef(S_k) + Sim(s_1, s_2) + Close(s_1, S_k) + StrTieStr(s_1, s_2) \quad (5)$$

$$\forall s_1, s_2 \in S_k$$

The temporal clustering coefficient [91] measures the degree to which nodes in a graph tend to cluster together by resolving the topological overlap of the neighbourhood (Ego Networks) of all nodes $s_k \in S_k$ in two adjacent snapshots t and $t+1$ of a time-varying graph as:

$$CCoef(S_k) = \frac{1}{|S_k|} \sum \frac{\sum_{EN} a_t(k, j) a_{t+1}(k, j)}{\sqrt{\sum_{EN} a_t(k, j) \sum_{EN} a_{t+1}(k, j)}} \quad (6)$$

where $s_k, s_j \in S_k$ are nodes in the network and $a_t(k, j) = 1$ if at least there is one contact (k, j) in time T between s_k and s_j .

The similarity value between $s_1, s_2 \in S_k$ within a time interval Δt is [14,19]: $\frac{\sum_t |N_{s1} \cap N_{s2}|}{\Delta t}$ (7) where $|N_{s1} \cap N_{s2}|$ is the similarity in contacts between two subscriber s_1, s_2 of the content k .

The closeness centrality of s_1 is a measure of how close s_1 is to any other node in S_k . It is measured as the inverse of the average distance from s_1 to any other node in S_k [14,19]: $\frac{s-1}{\sum_j d_{1j}}$ (8) where d_{1j} is the distance between s_1 and s_j in the set of subscribers $s \in S_k$.

The node tie strength value between of subscribers $s_1, s_2 \in S_k$ within a time interval Δt is [2,3,14]:

$$\sum_t \frac{f(s_1)}{F(s_2) - f(s_1)} + \frac{rec(s_1)}{T(s_2) - rec(s_1)} + \frac{d(s_1)}{D(s_2) - d(s_1)} \quad (9)$$

where $\frac{f(s_1)}{F(s_2) - f(s_1)}$ measures the frequency of contacts between s_1, s_2 ;

$\frac{rec(s_1)}{T(s_2) - rec(s_1)}$ measures the recency of contacts between s_1, s_2 and $\frac{d(s_1)}{D(s_2) - d(s_1)}$

indicate the relative distance by hops between s_1, s_2 . The complex temporal graph metrics of contact frequency, recency and topology distance allow congruency with the underlying dynamic changing network topology and connectivity. In order to balance the trade-off between serving contents requested from a specific local region of highly connected subscribers and from

multiple less-connected subscribers, we favour the weak tie strength which helps to give a wider and broader long-term predictive content popularity instead of only favouring and serving the contents requested from a local region of highly connected subscribers.

Spatial heuristics allow CafRepCache to choose the most suitable caching points to cache popular contents based on its relative location with the subscribers and between the subscribers themselves.

3.5.3 Collaborative relative utilities

For each of the heuristics h , we define their respective utilities $Util_h(n, C_i(n))$ as measurements of their relative gain, loss or equality, calculated as pair-wise comparison between the node's own heuristics and that of the encountered contacts in the following way:

$$Util_h(n, C_i(n)) = \frac{h(n)}{h(n) + h(C_i(n))} \quad (10)$$

Ego-network driven predictive analytics allows collaboration with asynchronous neighbours. Specifically, ego-network metric for heuristic h of node n is calculated as the aggregated values of all the encountered nodes for node n .

$$EN_h(n) = \frac{1}{N} \sum_{i \neq n \in N}^N \alpha_i h(i) \quad (11)$$

We combine the above relative utility driven as below.

$$Util(n, C_i(n)) = \frac{1}{|H|} * \sum_{h \in H} Util_h(n, C_i(n)) \quad (12)$$

$$h \in H = \{Content, Social, Resource, EN_{Cont}, EN_{Soc}, EN_{Res}\}$$

Our work does not aim to achieve the global optimum, but to assume there are no malicious nor selfish nodes in the network and use principle of dynamic predictive relative utilities to propose a collaborative algorithm which allows individual nodes to achieve greater utility than if they do not collaborate.

In our earlier work [60, 100], we show that assigning different weights μ to each CafRepCache's utilities results in different performance for real-world traces.

We currently assume equal weight between each heuristic utility of CafRepCache.

3.5.4 Aggregated historical & present signals

Each node monitors a number of signals such as content popularity that it stores and disseminates to encountered nodes over time. We have explored a number of models for aggregating historical and present information in order to detect signal changes in data and predict an estimation of aggregated value.

Exponential (smoothing) Moving Average (EMA) is a technique used to detect significant changes in data by considering the most recent data [101]. Instead of maintaining a large array of time series data for each signal, it represents the information as a single value where short-term fluctuations are smoothed out and longer-term trends are highlighted making it suitable for forecasting. EMA value is defined as:

$$EMA(x) = \alpha EMA(x)_t + (1-\alpha)EMA(x)_{t-1}$$

where α is a fraction that represents the responsiveness of the smoothing, typically has value of 0.8 as EMA consider the most recent data rather than historical data.

EMA [101] is easy to learn and apply, memory-saving (i.e. only three pieces of data are required instead of large array of time-series data) and is able to produce accurate prediction in scenarios where the recent observations are more significant importance compared to historical observations. However, EMA is not sufficiently responsive to the real-time dynamic fast-changing trend analytic as EMA has smoothing variable which slowly adapts to the actual trend. Moreover, EMA ignores a lot of historical state information and thus, cannot capture the temporal and spatial locality of signals over time [101].

Instead of representing the information as a single value using EMA that places a greater weight and significance on the most recent data points which may lose the information of spatial-temporal locality in the past, we maintain an array of time series data to capture and predict adaptively the spatial-temporal locality of content requests using historical information. Due to the continuous nature of users' mobility and content requests, we propose to apply the concept of high-

order Markov chain [102] on our complex real-time analytics to predict the content traffic trends based on the historical information of content traffics related users' mobility. The K-order is defined as:

$$P(C_t = i_t | C_{t-1} = i_{t-1}, \dots, C_0 = i_0) =$$

$$P(C_t = i_t | C_{t-1} = i_{t-1}, \dots, C_{t-k} = i_{t-k})$$

Nevertheless, there is no single-bullet solution to address the complex problem of aggregating historical & present signals. While Markov model is the most flexible generalisation of sequence profiles, it is expensive in terms of memory and computational time. Furthermore, when $K = 1$, a lot of historical state information is still being ignored and only the information of the current moment is used [102]. Such limitations make the practical application of 1-order Markov chain prediction method lack of prediction accuracy. Compared with other existing time-order-based prediction methods, the K-order Markov chain performs much better when the order number $K = 2$ [102].

We argue that while high-order Markov chain helps us to leverage historical information of temporal and spatial locality of content requests related subscribers' mobility to predict the traffic demand in different locations over long period of time, its accuracy is not better than EMA while the resource consumption is higher in random networks where the variation in connectivity, resource and content popularity is dynamically random. On the other hand, EMA is best used for forecasts that are short-term and in the absence of seasonal, cyclical or regional variations [101].

3.5.5 Reputation-based weighting mechanism for collaborations

We have explored a number of models for weighting the contacts' signal within a node's ego-network in order to improve the accuracy of caching prediction of its regional networks. More specifically, we have considered techniques such as simple average or exponentially weighted moving average. In our proposal, we propose a reputation-based weighting mechanism built in CafRepCache.

Our novel distributed reputation aware mechanism integrates four main features (Fig. 3.6): centrality, credit-based feedback, reputation discounting and secondary response (as in line with [76, 77, 103]) to weight the value of

exchanged network information among ego network nodes in order to improve the collaborative caching decisions in highly mobile and disconnection-prone scenarios. Centrality feature means that nodes with higher centrality have higher connectivity or higher probability of getting in contact with many other nodes than nodes with lower centrality, thus its observation on popularity of a particular content is preferred. Credit-based feedback helps evaluate the accuracy and usefulness the predictive analytic value reported from a node in ego networks to a caching point over time. Reputation discounting increases and decreases proportionally the reputation or weighting of a node based on the accuracy of its exchanging value. The secondary response avoids reputation discounting backfires by increasing sensitivity to nodes that have been deemed predictively inaccurate in the past [10, 76, 77, 103].

Fig. 3.7 shows the interaction between the key components of our reputation model in order to provide automatic and autonomous weighting collaborative signals based on the available neighbours' reputations.

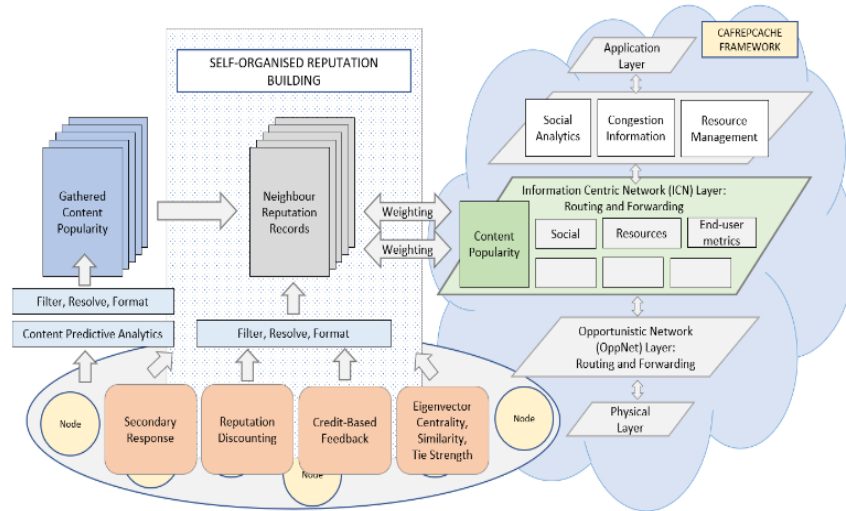


Figure 3.7 Reputation system model to deal with weighting problem in collaboration process.

Assuming that a caching point n_i gets a demand to resolve popularity of a certain content o_k in order to make a caching decision (e.g. whether to cache that content or not, whether to choose to offload or evict the content, etc.). Caching point n_i aggregates different popularity values $p_{j,k}$ of content o_k observed from each neighbour node $n_j \in EN_i$ in their ego network EN_i through

the collaborative caching process. We argue that every node's perspective of the network may have a different weight and level of accuracy, thus we weigh the value of content popularity observed from a node based on its reputation. We resolve a node's reputation based on its *social centrality* (i.e. centrality, similarity and tie strength [50]), thus the aggregated content popularity of the content o_k observed from all the neighbours of n_i is: $z_{i,k} = \frac{1}{|EN_i|} \sum_{n_j \in EN_i} SocialDegree_{i,j} \cdot p_{j,k}$

$$SocialDegree_{i,j} = \alpha U_i^{centrality} + \beta U_{i,j}^{similarity} + \gamma U_{i,j}^{TS} \quad (13)$$

Let $A_{i,j}$ be the adjacency matrix, i.e. $A_{i,j} = 1$ if n_i is linked to n_j and $A_{i,j} = 0$ otherwise. The relative eigenvector centrality [50] of n_i is defined as: $x_i = \frac{1}{\lambda} \sum_{n_j \in EN_i} x_j = \frac{1}{\lambda} \sum_{j=1}^N A_{i,j} x_j$ in which λ is a constant.

We argue that nodes with higher similarity value [50] with the caching point may help to give short-term predictive content popularity due to the spatial and temporal locality of content requests. The similarity degree between two nodes n_i and n_j is the number of common contacts between them measured by $|N_i \cap N_j|$.

We also consider nodes which have weak tie strength [50] with the caching point as we argue that these nodes may have completely different connectivity patterns compared to the caching point, thus may give a wider and broader long-term predictive content popularity [9, 10, 99]. The tie strength value between node n_i and n_j is defined as: $TS = \frac{f(j)}{F(i) - f(j)} + \frac{rec(j)}{T(i) - rec(j)} + \frac{d(j)}{D(i) - d(j)}$

We observed that centrality and similarity have a major contribution to the reputation building due to the spatial and temporal locality of content request whiles nodes with weak tie strength does not add significant value to the accuracy of aggregated content popularity. Thus, we propose a *credit-based feedback* algorithm in order to evaluate the accuracy and usefulness the content popularity value reported from a node in ego networks to a caching point.

At time $t = 0$, the reputation of all nodes in the network is equal to their social degree as described in (13). After the first exchanging information, node n_i gets

the collaborative content popularity from others and weight them based on its reputation. After the first cache hit, caching point n_i weights up the reputation value of nodes who reported correctly and weights down the reputation value of those who reported incorrectly.

$$\begin{cases} R_{i,j}(t) = \text{SocialDegree}_{i,j} \text{ at } t = 0 \\ R_{i,j}(t) = R_{i,j}(t - 1) \pm \partial \end{cases} \quad (14)$$

where ∂ is the value added or deducted based on the success or failure of content popularity prediction of a collaborative node. ∂ value is proportional to the gap between the content popularity value it reported to n_i and the actual content popularity n_i experiences after a certain time window ΔT . The reputation of a node decreases when it reported to n_i that a certain content is high popular but n_i does not encounter any request for that content, and vice versa. In this case, ∂ value added or deducted is proportional to the time duration ΔT .

In addition to using historical observations, we also use adaptive expiration timer, reputation discounting and secondary response [10, 76, 77, 103] specifically tailored for our highly challenging environment in order to increase the accuracy and reliability of the reputation resolution. Each node holds N neighbour reputation records where N can be determined by the node's resource capacity. Nodes with higher reputation and centrality should hold enough reputation records about other nodes in order to provide sufficient coverage of the nodes in its own area. Node recycles these records using expiration time (i.e. expiration timer) to balance the trade-off between different overheads and the need to have enough reputation about different neighbours. Our protocol uses reputation discounting to ensure that old reputations will age giving more chance for nodes to reclaim their reputation by consistently behaving in a cooperative manner and contributing to the aggregated content popularity. The reputation value of a node ages along time: $R_{i,j}(t) = \alpha * R_{i,j}(t - 1)$ where α is aging factor.

The secondary response avoids reputation discounting backfires by increasing sensitivity to nodes that have been deemed predictively inaccurate in the past.

Assuming that n_i take reputation into account for every node, the aggregated content popularity of the content o_k observed from all the neighbours of n_i is

$$\text{now rewritten as: } \begin{cases} z_{i,k} = \frac{1}{|EN_i|} \sum_{n_j \in EN_i} R_{i,j} p_{j,k} \\ R_{i,j}(t) = \text{SocialDegree}_{i,j} \text{ at } t = 0 \\ R_{i,j}(t) = \alpha R_{i,j}(t-1) \pm \partial \end{cases} \quad (15)$$

3.6 The CafRepCache Pseudocode

Each node has a unique ID and every routed message has an associated key and state information which may contain static and dynamic content topics and content data, publishers' IDs, subscribers IDs, timestamps, location, IDs of other encountered nodes, times stamps of these meetings etc. Contents are tagged with a set of attributes which are hashed and stored in DHT-like overlay that effectively matches the hash value of interest with attributes representing the content.

Handle Content Packet Arrival
When the <i>Content</i> is received at a <i>Node</i>: if <i>Node</i> is a cache candidate of <i>Content</i> cache <i>Content</i> at <i>Node</i> end if while <i>Node</i> is congesting: for each content in Content Store for each Contact in scan do U = Contact.calculateNodeUtility(Contact.nodeHeuristics) if $U > U_max$ $U_max = U$ end if end for if $U_max > \text{Node.Utility}$ delegate the <i>content</i> from <i>Node</i> to <i>Contact</i> else evict/remove the <i>content</i> from <i>Node</i> to free up the space end while forward the <i>Content</i> toward its subscribers

Table 3.2 CafRepCache's pseudo code when a content is arrived.

Each publisher proactively advertises its list of content name to neighbours who have high social utility and availability utility using CafRepCache forwarding protocol. Intermediate nodes keep track of the content name associated with the publisher ID and forward the advertisement to next-hops. Eventually, the knowledge of which node owns which content in the network will be disseminated throughout the network.

Each subscriber sends its interest to neighbours who also have high sociality and availability. Intermediate node first checks if it has the requested content, then forward the content to the subscriber. Otherwise, it checks whether if it knows who owns the content (results from content advertisement process), then forward the request to next hops. When interest packet reaches the nearest cached content or the publisher, the node forwards the actual content data back to the subscriber using CafRepCache forwarding scheme.

During content retrieval process, using interest forwarding table, relay node matches the content topic and summary vector of the subscriber with the information it has about the published content, and forwards it to the subscriber. Along with forwarding the content or queries to next hops that have high social centrality and resources, intermediate nodes decide whether to cache the content, forward it or delegate it in case of resources limitations. When a caching node detects it is likely to start congesting, it ranks the content in terms of its popularity and delegates the least popular content to a suitable node. Nodes suitability is ranked in terms of the same multi-criteria metric we described (social, resources and workload). Each caching point tries to maximise its utility value or in other words, minimise its total cost. During the operation, the node adapts with changing environment and adjusts its local caching strategy that may decide whether to cache the content if it improves the utility value, or stop retrieving a content from another node due to high cost.

Handle Content Request Arrival
When the <i>Interest</i> (or request) of a <i>Content</i> is received at a <i>Node</i>: if <i>Node</i> has already cached <i>Content</i>: send <i>Content</i> to <i>Interest</i> 's subscriber else

```

listInterest = {}
if listInterest[] not contains Interest
    listInterest.add(Interest)
end if
if listSubscribers[] not contains Interest's subscriber
    listSubscribers.add(Interest 's subscriber)
end if
for each interest in listInterest[] do:
    if interest is expired:
        listInterest.remove(interest)
    end if
end for
ListUtils = {}
for each Contact in scan do:
    Contact.nodeHeuristics = exchNodeHeurInfo(Contact)
    Contact.calculcateNodeUtilities(Contact.NodeHeuristics)
    Contact.heuristics = exchCafRepCacheHeurInfo(Contact)
    U = Contact.calculateUtility(Contact.CafRepCacheHeuristics)
    ListUtils.insert(Contact.Utility)
    calculateContactReputation(credit,ΔT)
    Contact.contentPopularity =
exchContentPopularityInfo(Contact.Reputation)
Contact.calculateContentPopularity(Contact.ContentPopularity)
end for
(x,y) = in-networkCaching(ListUtils)
if x == 1
    Node is set to be a cache candidate of the Content
end if
if y == 1
    forward Interest from Node to Contact who has highest utility value
end if

```

Table 3.3 CafRepCache's pseudo code when an interest is arrived.

3.7 Summary

This chapter has introduced CafRepCache, a multi-layer predictive caching framework that combines real-time adaptive multipath content and interest forwarding, with adaptive cognitive collaborative caching. In the first part, we explored the multidimensional design space and multilayer architecture of CafRepCache framework. The second part of the chapter described the functional model of collaborative cognitive caching and identified the parameters and factors to be considered for each of its functional stages. In the third part, we described in detail CafRepCache complementary multi-layer real-time predictive heuristics resolved by both single node and ego networks in order to predict and adapt with dynamically changing topology, dynamic resources and varying content traffic patterns as well as manage the complex dynamic trade-offs between them. The fourth part describes CafRepCache pseudocode in detail.

The next chapter describes the implementation of CafRepCache, evaluates CafRepCache against state-of-the-art caching solutions in heterogeneous environments to understand its performance characteristics and demonstrates its effectiveness in terms of cache hit ratio, content retrieval success ratio, latency, packet loss, resource consumptions and energy.

Chapter 4

4. Implementation and Evaluation

This chapter describes our experimental methodology, the implementation and performance evaluation of CafRepCache. We compare the performance of CafRepCache with three forwarding and caching benchmark protocols: LRU-SocialForwarding [71, 104], Time Aware Least Recent Used (TLRU) [105, 106] and Optimal Cache Placement Based Content Popularity (OCPCP) [107] and three state-of-the-art comparative protocols SocialCache [44], HyMobi [22] and LocationCache [108].

We analyse CafRepCache and competitive protocols performance across different underlying network topologies. In mobile opportunistic networks, nodes' mobility and connectivity have a major impact on the performance of communication protocols. Therefore, we use different set of heterogeneous connectivity traces which have different degree of mobility, connectivity patterns ranging from completely random connectivity traces to realistic social connectivity traces collected from CRAWDAD community resource [109]. While random traces give us random distribution in network connectivity, our chosen real-world connectivity traces follow non-uniform scale-free connectivity distribution with spatial and temporal heterogeneity. The random traces are generated by Random Waypoint model while our real-world connectivity traces vary from San Francisco Cab traces [110] to Infocom [111, 112], RollerNet [113]. San Francisco Cab traces [110] contains the mobility traces (i.e. GPS) of 550 cabs over a period of 30 days in the San Francisco Bay Area, USA. Infocom [111, 112] is a 4-day trace of Bluetooth sightings by groups of 78 users carrying small devices (iMotes) and additional 20 static long-range devices placed at various semi-static and static locations of the conference venue. RollerNet [113] includes traces of opportunistic sighting of Bluetooth devices carried by 62 roller-bladders during three hours when they travel about 20 miles in Paris and utilize Bluetooth on their cell phones for communication. Note that there are other real-world connectivity traces which can be used to simulate underlying network topologies for CafRepCache evaluation such as Oviedo [114], DieselNet [115], etc but in this thesis, we use San Francisco Cab

traces [110], Infocom [111, 112] and RollerNet [113] because each of these real-world dataset exhibits vastly different connectivity degrees derived from different mobility patterns of different real-world scenarios. Each trace represents different degree of network topology characteristics in terms of connectivity duration, isolation duration, number of contacts in a period of time (i.e. highly dynamic, short connectivity in SF Cab; medium dynamic and moderate connectivity in RollerNet; less dynamic, longer connectivity in Infocom). This gives us statistically sufficient diversity to evaluate CafRepCache and competitive caching protocols performance in different contexts.

Overlaying on top of aforementioned dynamic network topologies, we evaluate CafRepCache and competitive protocols performance with different heterogeneous content publish-subscribe patterns. The reason for this is that while different node mobility models help us to ground heterogeneous network topologies for forwarding contents and interest messages, it alone cannot verify the effectiveness of CafRepCache content caching decisions without content request model. Our publish-subscribe patterns diversify vastly from pseudo-realistic content request traces to real-world content request traces. While pseudo-realistic traces allow us to vary dynamic publish-subscribe patterns ranging from completely random content requests to scenarios where the networks are dominated extremely by a small set of contents (e.g. Zipf-distribution), real-world content request traces help to represent the realistic distribution of content requests over time and location. More specifically, in our pseudo-realistic traces, publishers and subscribers are randomly chosen and the data is generated with the constant bit rate; the content requests occur either uniformly and independently from each other (i.e. Independent Reference Model - IRM) or follow Zipf's distribution where there are some contents more popular than the others. On the other hand, our real-world content request traces do not necessarily follow IRM or Zipf's distribution, they are collected from realistic publish-subscribe traffic and are able to capture the temporal and spatial locality of content traffic in different locations over time that pseudo-realistic traces cannot capture. Our chosen real-world content requests traces are Youtube traces [116, 117] and Foursquare traces [118, 119]. YouTube dataset

[116, 117] is a real trace for content requests which contains total realistic 2135798 video requests. Foursquare traces [118, 119] contain 227,428 subscriptions of 18,201 users in different locations of New York city during the period of 10 months.

Our performance metrics are categorised into three groups: 1) forwarding-based criteria including success ratio, delay, packet loss; 2) caching-based criteria including cache hit ratio, eviction rate, footprint reduction, hops distance between caching points and subscribers; 3) resource utilisation criteria including computational costs, communication costs and number of caching points. This helps us to analyse thoroughly and understand deeply the effectiveness of each predictive analytic and heuristic on the performance of CafRepCache and other competitive forwarding and caching protocols.

We customise and use ONE simulator [120] to represent heterogeneous opportunistic network environments with different node mobility, connectivity models as well as various content traffic patterns. As ONE simulator [120] is proposed to support routing messages between nodes with various mobile opportunistic networks routing algorithms, it naturally does not support sufficiently content caching functionality. Therefore, we customise and extend the original ONE simulator for our experiments. All experiments are repeated ten times and averaged to give us statistically meaningful results.

We show that CafRepCache consistently maintains higher success ratios and cache availability while keeping lower delays, packet loss rates, and caching footprint compared to the competing benchmarking and state-of-the-art protocols across a wide range of topologies and user traffic. We also show that the computational cost and network overheads of CafRepCache are only marginally increased compared with the others.

The rest of chapter 4 is organized as follows. Section 4.1 describes in more details the connectivity characteristics of heterogeneous network mobility models in terms of contact durations, isolation duration and number of contacts. Section 4.2 describes the characteristic of different content request traces in term of the content distribution degree. Section 4.3 discusses CafRepCache performance across three forwarding criteria: success ratio, delay and packet

loss. In this section, we extensively evaluate CafRepCache against three competitive protocols (LRU [71, 104], SocialCache [44], HyMobi [22]) over three heterogeneous real-world mobility traces: San Francisco Cab [110], Infocom [111, 112] and RollerNet [113] in the face of vastly different workload and content popularity patterns from Youtube dataset [116, 117]. Section 4.4 evaluate CafRepCache performance against four state-of-the-art caching protocols: SocialCache [44], LocationCache [108], TLRU [105, 106] and OCPCP [107] across four caching criteria: cache hit ratio, eviction rate, footprint reduction, hops distance between caching points and subscribers. In this section, we choose Infocom as the underlying network topology and vary the content publish-subscribe patterns by dynamically varying the control parameter of content popularity, popularity variation, localisation factor, cache size and number of subscribers over Foursquare datasets [118, 119]. Section 4.5 discusses the network resource utilisation of CafRepCache in terms of computational costs, communication costs, number of caching points while varying the number of subscribers in three real-world mobility traces: San Francisco Cab, Infocom and RollerNet. Section 4.6 gives conclusions.

The Fig. 4.1 below shows the overview of our experimental methodology.

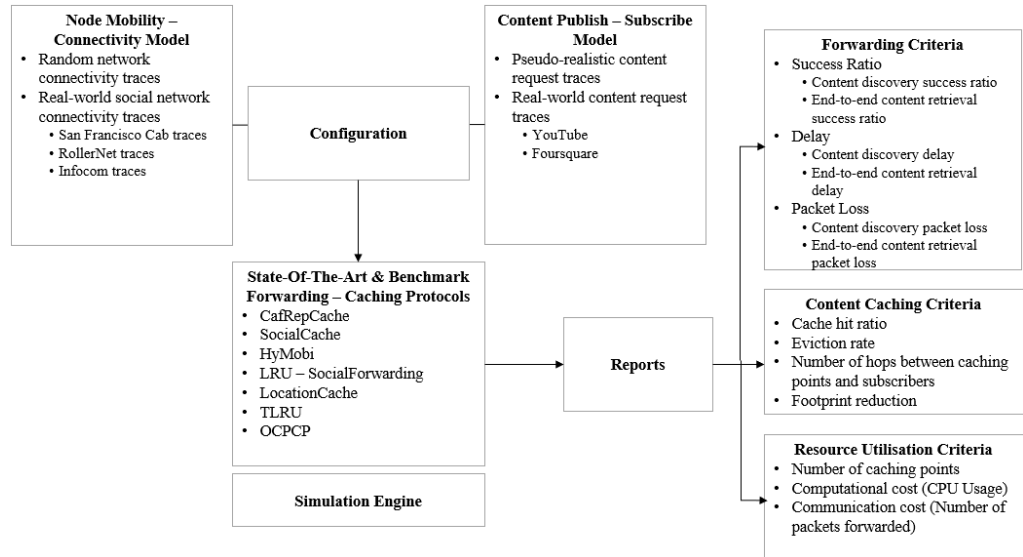


Figure 4.1 Experimental Methodology.

4.1 Heterogeneous Network Topology Traces

We analyse CafRepCache performance in different underlying network topologies with different degree of mobility, connectivity patterns ranging from completely random traces to realistic social traces collected from CRAWDAD [109]. The random traces are generated using random waypoint model while realistic social traces are collected from three real-world traces: San Francisco Cab [110], Infocom [111, 112] and RollerNet [113].

In the perspective of complex graph theory, our network topology traces represent the heterogeneity of node degree distribution which can be categorised into random and scale-free networks as shown in Fig 4.2. Most nodes in random networks have the same number of connections and the degree distribution will be a Gaussian bell-shaped curve [121, 122]. Random graphs have short average paths and low clustering [121, 122]. Example scenarios of random networks are natural disaster events or mobile crowdsourcing systems. On the other hand, scale-free networks are characterized by a highly heterogeneous degree distribution, which follows a power-law distribution [98, 123]. They are called scale-free because zooming in on any part of the distribution doesn't change its shape: there is a few, but significant number of nodes with a lot of connections and there's a trailing tail of nodes with a very few connections at each level of magnification. In the temporal dynamics scale-free networks, popular nodes may become extremely less popular, and emerging new nodes may become extremely popular, in a very short time. Then, although the network may change significantly over time, the degrees of its nodes obey the power-law model at any time [98, 123]. Example scenarios of complex temporal scale-free networks are urban cultural events or public transport systems which represent different degrees of human social behaviour. Empirical evidence [27, 96] shows fixed and wired networks are typically scale-free, whereas mobile and opportunistic disconnection-tolerant networks (DTNs) can be either random [97] or scale-free [27, 98].

The remaining of this section describes in more details the connectivity characteristics of heterogeneous network mobility models in terms of contact durations, isolation duration and number of contacts. This gives us statistically sufficient diversity to evaluate CafRepCache performance in different publish-

subscribe contexts as well as to understand fundamental performance limitations of CafRepCache in different heterogeneous network topologies.

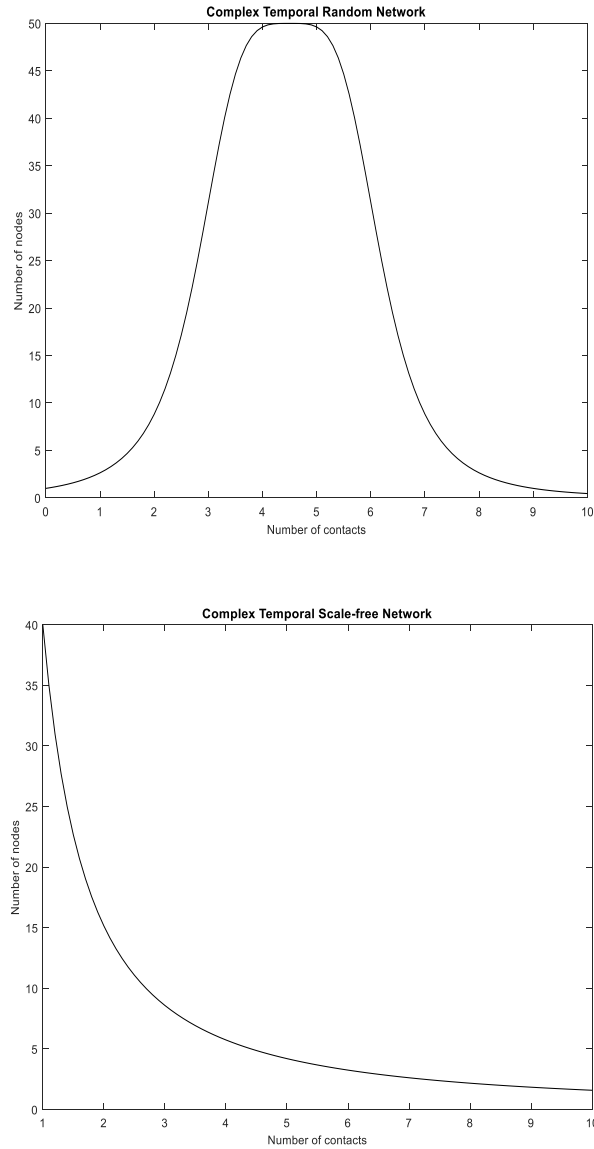


Figure 4.2 Random Networks vs. Scale-free Networks

4.1.1 Random Network Connectivity Traces

We generate random network connectivity traces by using random waypoint mobility model in ONE simulation as the underlying node mobility. Random waypoint model [124, 125] is a synthetic random model for the movement of mobile users and is commonly used to evaluate opportunistic ad-hoc networks. The movement of nodes is governed in the following manner [124, 125]: each node pauses for a fixed time duration before starting to move from one waypoint to the next waypoint. The waypoints are uniformly distributed over the given

convex area and a random velocity of node is drawn from the velocity distribution. This behaviour is repeated for the length of the simulation.

Random network connectivity traces allow us to evaluate CafRepCache performance in highly dynamic changing network environments as well as to analyse CafRepCache in heterogeneous terrestrial applications where the network connectivity is random and unpredictable.

4.1.2 Real-world Social Network Connectivity Traces

We use San Francisco Cab [110], Infocom [111, 112] and RollerNet [113] traces in ONE [120] simulator. Fig 4.3. a, b and c show that connectivity characteristics of SF Cab, Infocom and RollerNet in terms of contact duration, isolation duration and number of contacts. San Francisco trace [110] is the most challenging trace compared with RollerNet [113] and Infocom [111, 112] due to very short connectivity durations, very high disconnections and low number of contacts during connected times. We observe that both San Francisco and RollerNet traces exhibit short contact durations (a mean of 45 s and 33 s, a median of 11 s and 24 s and a maximum of 73 s and 42s respectively) while Infocom has substantially longer contact durations (a mean of 2.5 min, a median of 2 min and a maximum value of 4 min). San Francisco trace suffers from the longest isolation periods (a mean of 0.5 h, a median of 1.7 h and a maximum value of 3 h) compared to RollerNet and Infocom (with a mean of 1.5 min and 4 min, a median of 1 min and 6 min, and a maximum of 4 min and 10 min respectively). In addition, RollerNet trace has the highest observed number of contacts compared to San Francisco Cab and Infocom traces.

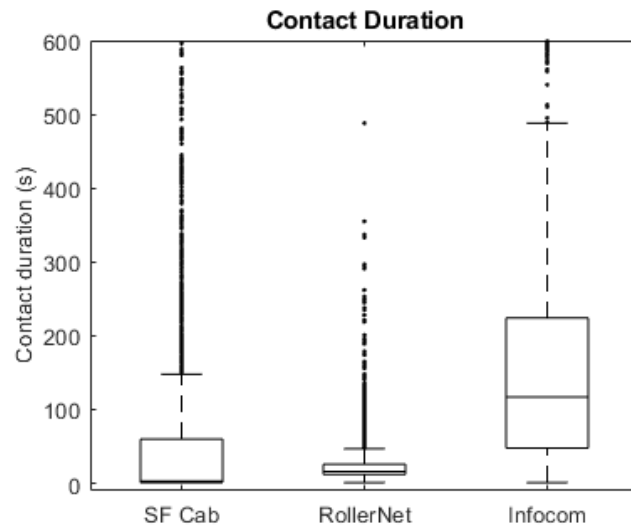


Figure 4.3a Contact Duration.

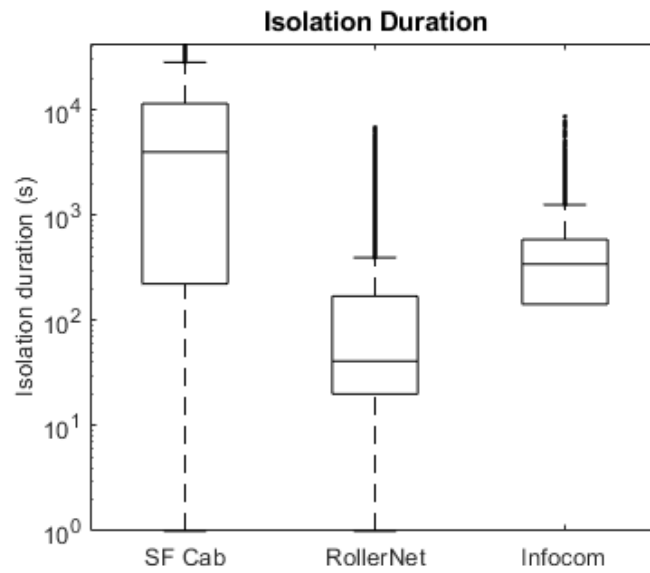


Figure 4.3b Isolation Duration.

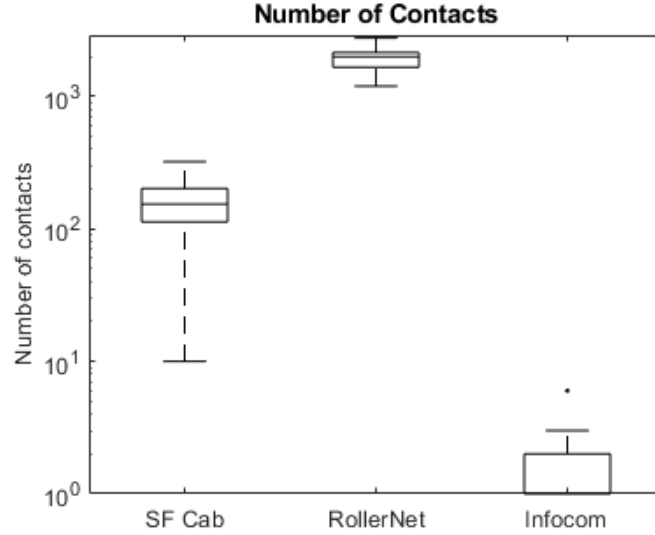


Figure 4.3c Number of Contacts

4.1.3 Random vs. Scale-free Network Connectivity Traces

We analyse and compare CafRepCache performance in two aforementioned underlying network topologies: completely random network topologies and social scale-free network topologies while varying the number of subscribers in the networks and using Foursquare dataset [118, 119] as real-world content request patterns. We use real-world Infocom [111, 112] dataset as a representative of scale-free network topologies. Fig. 4.4 shows that CafRepCache achieves 79.4% and 92.4% success ratio while the hop count average from the caching points to subscribers are 3.01 and 2.7 in random network and scale-free network respectively. We argue that in random network, CafRepCache benefits from its cache redundancy mechanism that select highly suitable locations for caching and replication when needed as adaptive replication and caching are both necessary to address multi-user data communications in dynamic fragmented and sparse topologies. In scale-free network, CafRepCache utilises its multidimensional predictive analytics and complex temporal graph metrics to make caching decisions in a predictive manner and congruent with the underlying network mobility, connectivity and content interest.

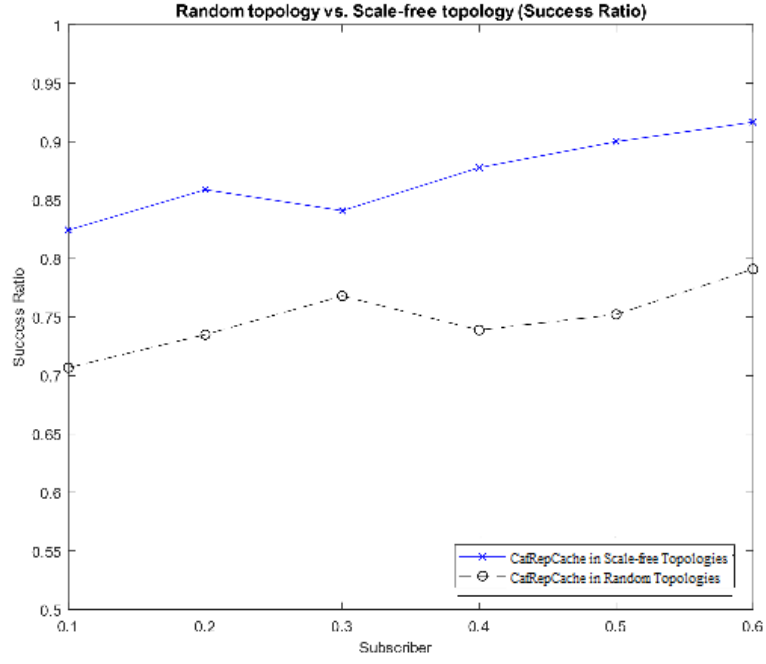


Figure 4.4a Success ratio vs. number of subscribers.

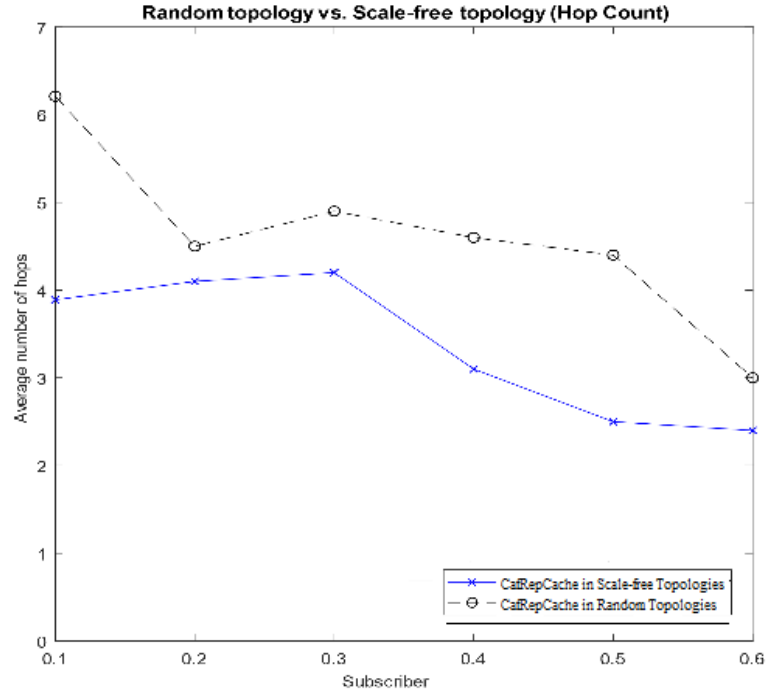


Figure 4.4b Average hops count between caching points and subscribers.

While we evaluate CafRepCache in random network topologies as worst-case scenarios where the network connectivity is totally random and unpredictable which are not always realistic [27, 98], for the rest of our experiments, we will focus on conducting CafRepCache performance evaluation against different

caching strategies in real-world scale-free network topologies where human, vehicles, etc. are involved [27, 98].

4.2 Heterogeneous Content Publish-Subscribe Traces

4.2.1 Pseudo-realistic Content Request Traces

We generate pseudo-realistic content request traces which allow us to have a deeper understanding of CafRepCache caching performance in various dynamic publish-subscribe patterns ranging from completely random content requests to scenarios where a small set of contents are requested more often than the others.

The pseudo-realistic content request traces follow Zipf-distribution such that the probability for a request of the k^{th} most popular content is $P(k, \alpha, K) = \frac{1/k^\alpha}{\sum_{q=1}^K 1/q^\alpha}$ with α refers to the popularity skewness.

We vary the parameter α to control the publish-subscribe patterns. The smaller α leads to a more uniform popularity distribution, which $\alpha = 0$ meaning that contents are randomly requested. While α is getting bigger (e.g. 1), the network is dominated by a subset of contents, resulting in a highly skewed degree distribution.

The pseudo-realistic content request traces represent the heterogeneity in content publish-subscribe patterns which allows us to achieve statistical diversity in evaluating CafRepCache caching performance in different content traffic patterns in addition to the diversity in underlying network mobility and connectivity we discussed in the previous section.

Figure 4.5 shows the distribution of three types of contents: text, picture and video as well as its content size proportion in our pseudo-realistic traces. In our evaluation with pseudo-realistic traces, text messages and videos are the most common (42% and 40% respectively) content traffic, followed by pictures (18%). However, texts only form a small proportion in size (537.6 MB in total, 128 KB average) compared to videos (33.6 GB in total, 8.4 MB average) and pictures (4.5 GB in total, 2.5 MB average). We generate dummy data to build up different size of contents.

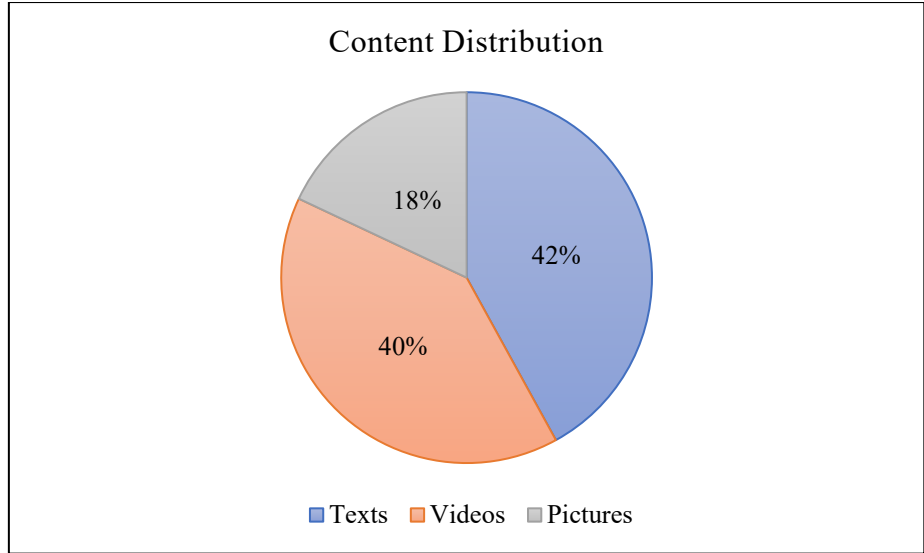


Figure 4.5a Content distribution.

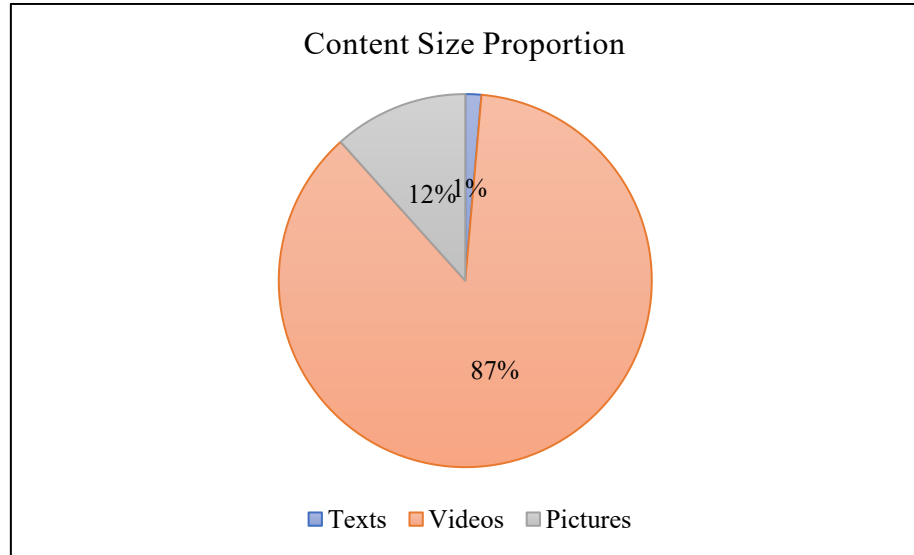


Figure 4.5b Content size proportion.

4.2.2 Real-world Content Request traces

We use real-world Youtube traces [116, 117] and Foursquare [118, 119] datasets to represent the realistic distribution of content requests. Youtube and Foursquare traces are collected from realistic publish-subscribe traffic and are able to capture the temporal and spatial locality of content traffic in different locations over time that pseudo-realistic traces cannot capture.

YouTube dataset [116, 117] is a real trace for content requests which contains total 2135798 video requests, each has video id, length, views and other attributes. Existing research [116, 117] suggests that content requests in

Youtube dataset generally follows power-law distribution with truncated tails. However, Youtube datasets do not capture the content traffic patterns in different locations over time. Fig 4.6 shows the relationship between mobile subscribers (red) and mobile publishers (green) presented as a graph as well as their relative location in a timestamped snapshot.

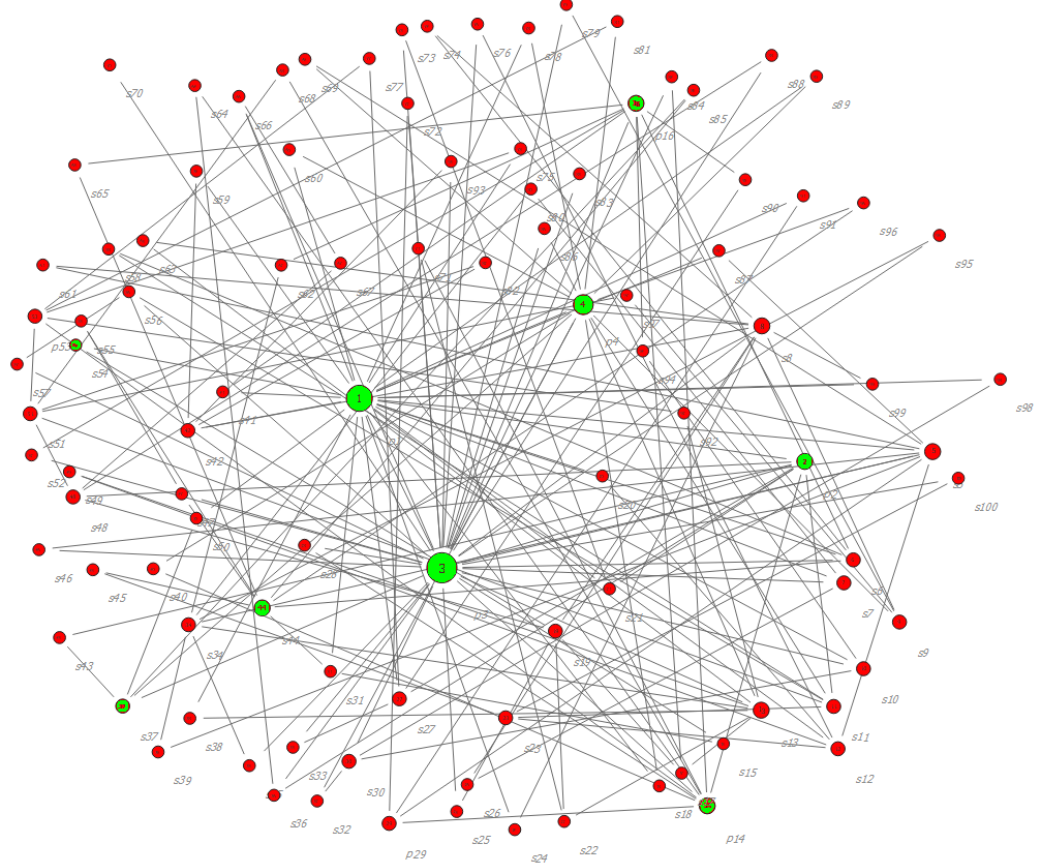


Figure 4.6 Publisher/Caching Point (Green) – Subscriber (Red) relationship.

Real-world Foursquare dataset [118, 119] models realistic publish-subscribe in the finest granularity in terms of overall distribution as well as content traffic patterns in different locations over time. Foursquare New York dataset [118, 119] is a real trace for content requests which is collected through location-based service Foursquare API (<https://developer.foursquare.com/>) describing the spatial-temporal locality of content requests in terms of user interests at public venues. It contains 227,428 subscriptions of 18,201 users in different locations of New York city during the period of 10 months. Each check-in is associated with its timestamp, its GPS coordinates and its semantic meaning (represented by fine-grained venue-categories). We assume the requested

content itself in Foursquare dataset have the same content types and distribution as we described in our pseudo-realistic datasets. We provide a detailed analysis of subscriber's behaviour across different days of the week in different locations from both network and social perspective, which brought us to the decision of modelling one week of traffic, that verified to be sufficient to profile subscriber behaviour.

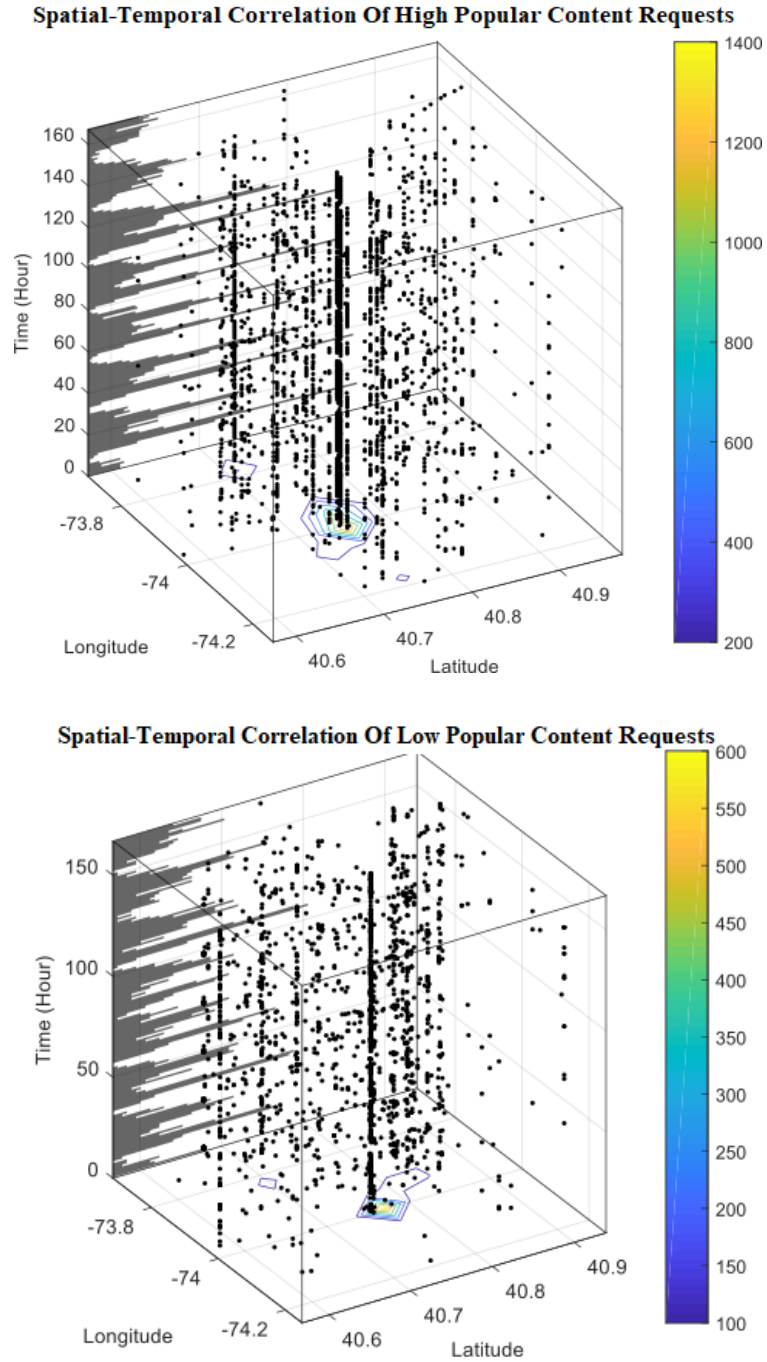


Figure 4.7a Spatial-temporal correlation of content requests in Foursquare dataset.

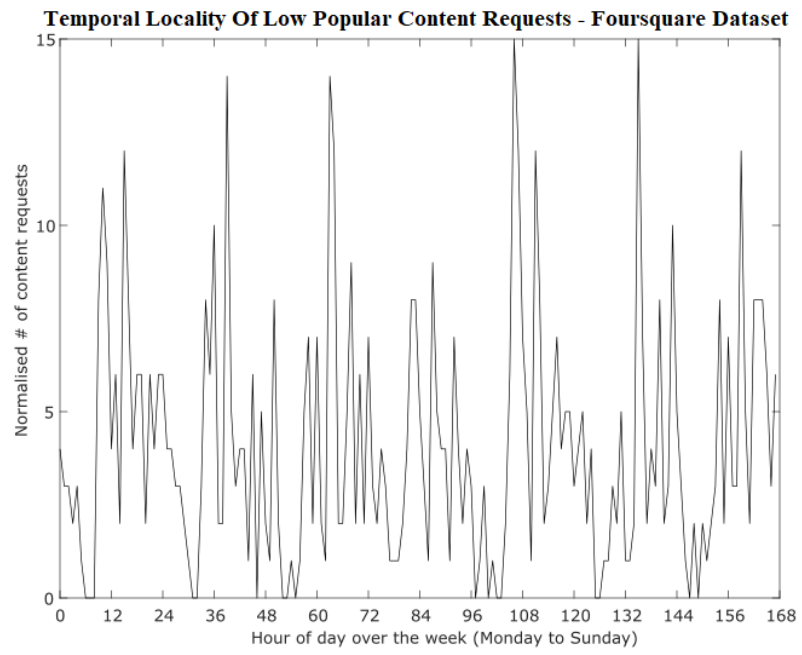
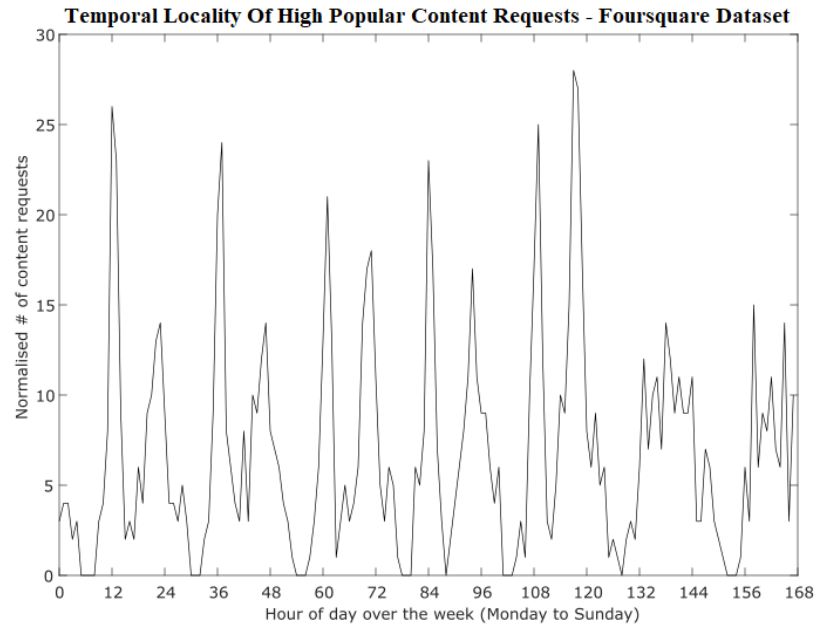


Figure 4.7b Temporal locality of content requests in Foursquare dataset.

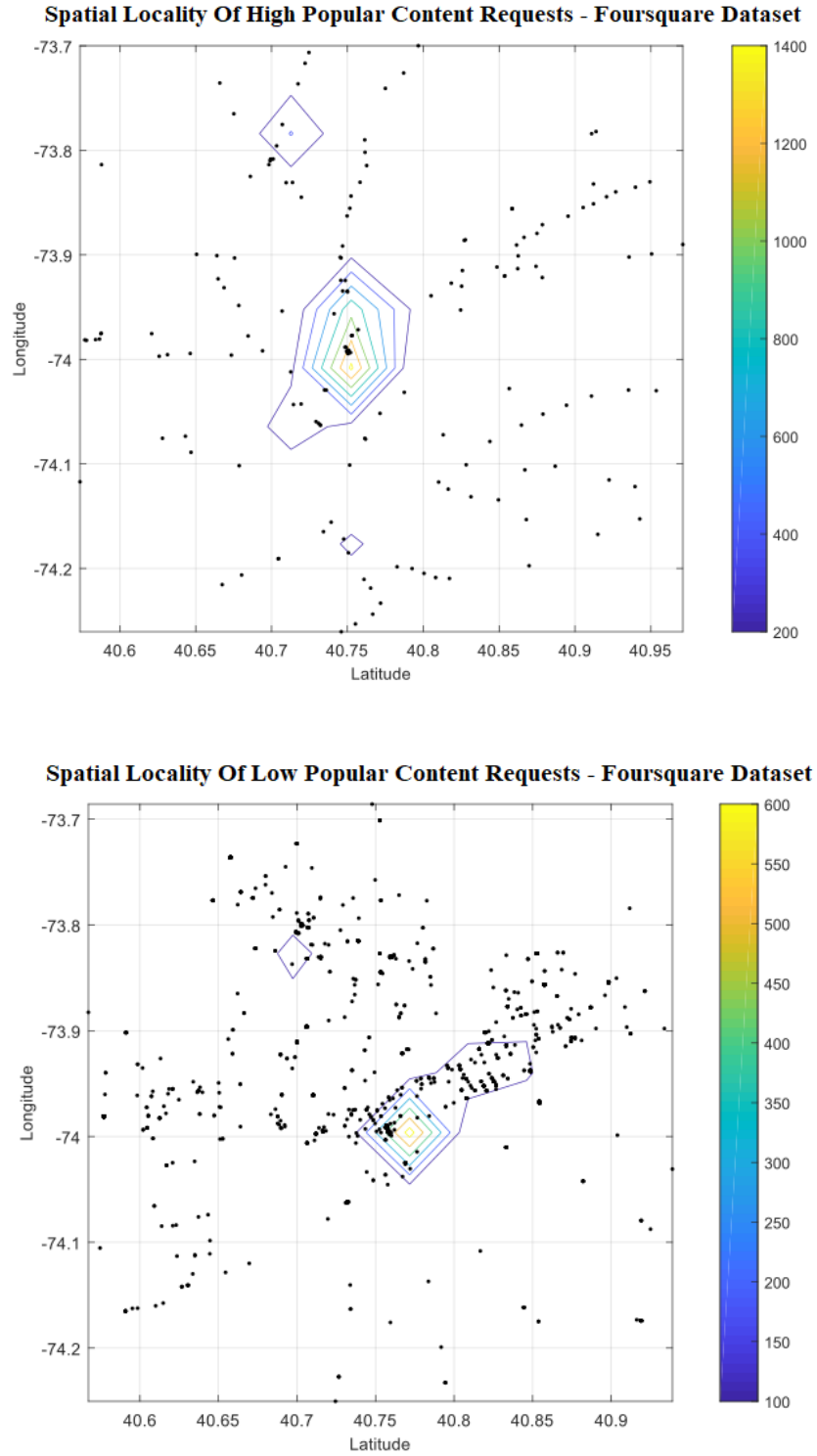


Figure 4.7c Spatial locality of content requests in Foursquare dataset.

Fig. 4.7 shows the spatial and temporal correlation of content traffic (i.e. spatial-temporal requests pattern of mobile subscribers) in Foursquare dataset for a content in different locations over time. It shows the temporal patterns (similarity) of content traffic during weekdays and weekend: if a content is

requested at a certain point in time, more likely it will be requested again in near future. In fact, nor are the content references scattered randomly and independently over time; rather, a content might be of particular interest at a certain time interval, while its popularity gradually fades out. The locations of mobile subscribers feature different degrees of similarity in content request such that the location 1 and 2 which are relatively close to each other have similar request patterns compared to that of location 3 which are far away from others. This captures the impact that the geographical diversity of the users has on the observed trace of requested contents by them. More precisely, the requests coming from a specific region in space are more likely to be similar than those collected over regions far apart.

As opposed to pseudo-realistic content traces, realistic content publish-subscribe traces do not necessarily follow any constant distribution such as Zipf [34] or Weibull distribution [126], thus represent the heterogeneity in real-world content traffic patterns. When evaluating CafRepCache and other competing protocols performance, we use the request distribution directly extracted from the real-world data traces while varying other parameters such as the number of subscribers. In addition to this, we also extract different request distribution from real-world dataset following different skewness parameter α in order to vary dynamically changing publish-subscribe content traffic patterns to have a deeper insight into caching performance. We consider the impact that real-world network traffic have on the behaviour of CafRepCache and other benchmark and state-of-the-art forwarding and caching protocols and how our proposed heuristics help to make more accurate predictive caching decisions for different publish-subscribe traces in different locations over different period of time.

4.3 Forwarding Performance Evaluation

In this section, we evaluate CafRepCache performance across three forwarding criteria: success ratio, delay and packet loss. We compare CafRepCache against three competitive protocols (LRU [71, 104], SocialCache [44], HyMobi [22]) over three heterogeneous real-world mobility traces: San Francisco Cab [110], Infocom [111, 112] and RollerNet [113] in the presence of heterogeneous network connectivity and dynamic workloads (by varying number of

subscribers) and in the presence of dynamic content request patterns from Youtube dataset [116, 117]. In addition to end-to-end content retrieval performance, we also evaluate CafRepCache and other competitive caching performance in content discovery process (i.e. time taken between sending interest packets to the network and discovering a requested content) in order to provide more insightful analysis of CafRepCache compared with other state of the art and benchmarks protocols

The total number of nodes in San Francisco Cab, Infocom and RollerNet data traces are 100, 62 and 100 nodes respectively. To evaluate the impact of heterogeneous network connectivity and dynamic workloads on caching performance, we have run six increments of the number of subscribers and publisher ranging from 10% to 85% of the total number of nodes in that dataset. We report on experiments with the increasing number of subscribers, but note that the results for the increasing number of publishers are similar to the ones presented here. All simulations are repeated 10 times, each with different random subscribers and publishers. The content workload is increased due to the increasing number of subscribers.

When we vary the number of subscribers, we utilise the real-world content request distribution extracted from Youtube dataset [116, 117]. To evaluate the impact of dynamic request patterns on caching performance, we also extract different request distributions from Youtube dataset [116, 117] following different skewness parameter α ranging from 0.2 to 1.2 to vary the dynamically changing publish-subscribe patterns. We keep the content request rate per second following real-world Youtube dataset [116, 117]. Varying skewness parameter α allows us to evaluate CafRepCache and other competitive protocols in heterogeneous publish-subscribe patterns ranging from completely random to power-law distribution.

4.3.1 Success Ratio In Content Discovery And Retrieval

The first metric we used to evaluate CafRepCache and other competitive caching performance is success ratio. We measure success ratio in end-to-end content retrieval process which indicates the probability that subscribers sending interest messages and successfully receiving requested contents from

caching points or publishers. In addition to end-to-end success ratio, we evaluate success ratio specified for interest message in content discovery process in order to measure the delivery rate of interest packets from subscribers toward caching points. This allows us to analyse thoroughly the successful communications for both interest packets and requested contents between subscribers, publishers and caching points.

Connectivity patterns (i.e. how central subscribers are in the network) and their locations (how close to the publishers and to each other subscribers may be) strongly influences content retrieval characteristics. We first rank the nodes in terms of their connectivity (more specifically degree centrality), then increase the numbers of subscribers starting from top 10% best connected random nodes, followed by increments of 15% (i.e. 25, 40%, 55, 70 and 85%) of the total number of nodes in SF, Infocom and RollerNet datasets. We assume that subscribers are mobile and not uniformly distributed as well as that they can have different connectivity patterns. This allows us to show how CafRepCache performs in a wide range of different topologies ranging from dense to sparse (Fig. 4.6 a, b and c). We compare CafRepCache against two state of the art (HyMobi and SocialCache) and one benchmark (LRU) algorithms over three heterogeneous mobility traces: SF, Infocom and RollerNet across three state-of-the-art criteria: success ratio, delay, packet loss.

Fig.4.8 shows that, in terms of content retrieval success ratios, CafRepCache outperforms the other competing algorithms SocialCache and HyMobi as well as LRU. CafRepCache starts off with 85% to 95% success ratio in end-to-end content retrieval process when there are 10% of subscribers with high connectivity requesting the contents while other algorithms are 10-50% lower across all three traces. When we increase numbers and diversity of subscribers requesting hybrid content popularity, CafRepCache algorithm manages to keep success ratio above 80% for all three traces.

CafRepCache is followed by HyMobi, SocialCache, and LRU whose performance is lower and ranges between 50-70% due to the lack of multilayer adaptation mechanisms. More specifically, predictive analytics of resources and connectivity enables CafRepCache to manage topologies with very different

density/sparseness patterns while predictive content analytics enables both smart caching and replication of suitable content at the suitable nodes. Note that sparse networks present limited forwarding and caching options at any given time, while dense networks are prone to suffering from transmission collisions due to wireless interference and high content workload.

CafRepCache not only manages to cognitively cache and replace the right contents due to its predictive content analytics but also be able to place the interest and content packets in the most suitable locations due to its adaptive forwarding and replication that diverts the content workload from its conventional social aware path at times of congestion and directs it via a different path that decreases the load of caching points, thus reduces end-to-end delays while keeping high success ratios. HyMobi, SocialCache and LRU either underutilise or overutilise the forwarding, caching opportunities and the resources, thus results in lower success ratios and higher delays.

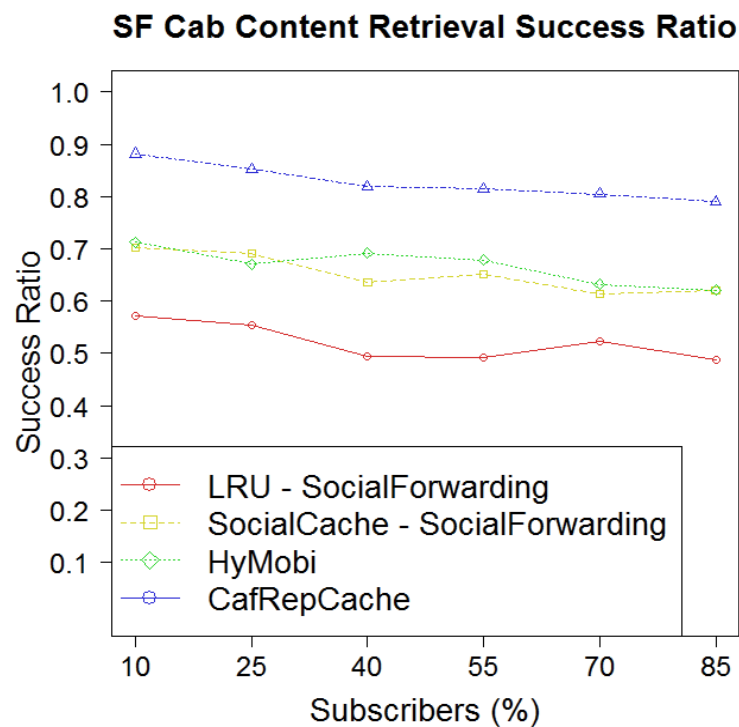


Figure 4.8a Content retrieval success ratio vs. Number of subscribers (SF Cab).

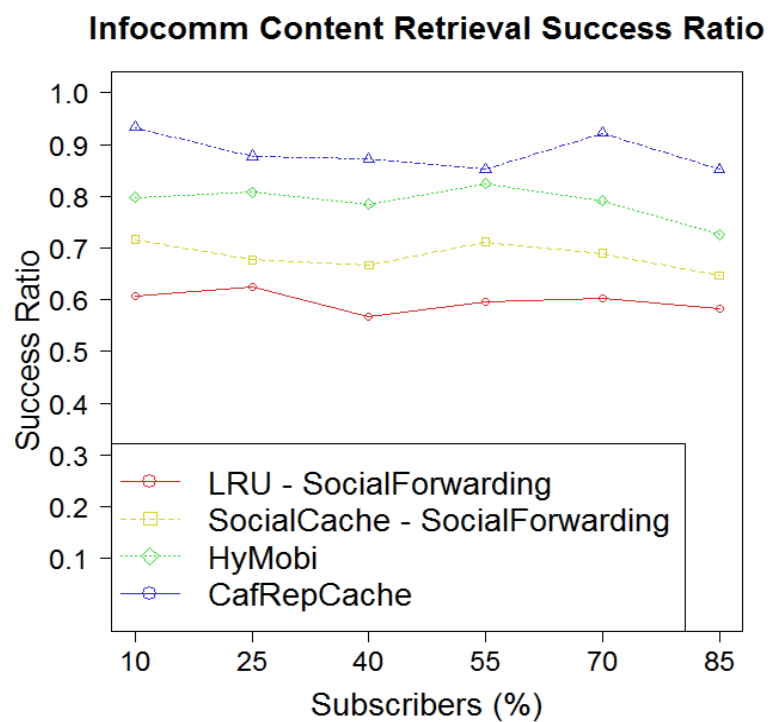


Figure 4.8b Content retrieval success ratio vs. Number of subscribers (Infocom).

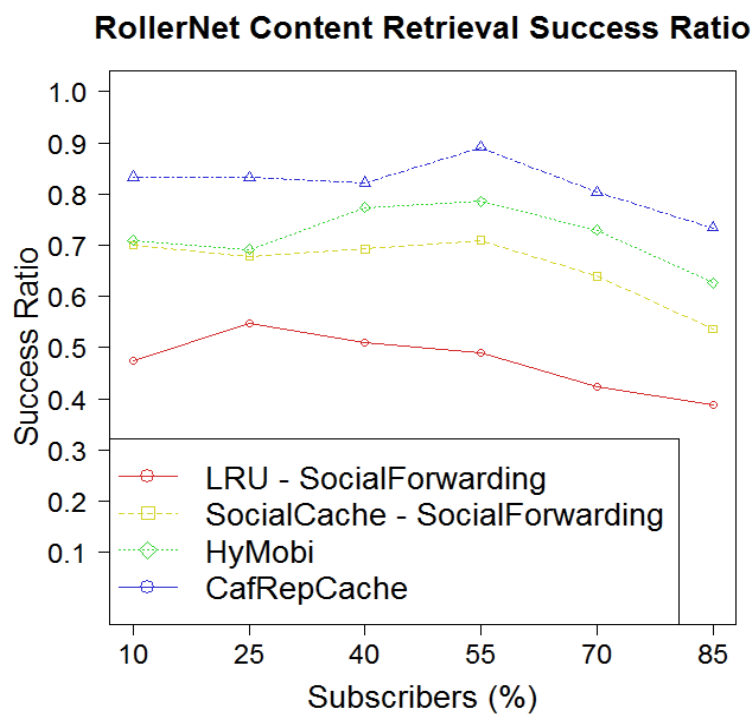


Figure 4.8c Content retrieval success ratio vs. Number of subscribers (RollerNet).

Figure 4.9 shows the end-to-end success ratio for content retrieval over three traces (San Francisco, RollerNet and Infocom) and in dynamically varying content traffic patterns. We observe that CafRepCache achieves highest content retrieval success ratio (82% for San Francisco, 88% for RollerNet and 92% for Infocom). HyMobi outperforms LRU and SocialCache but underperforms CafRepCache. LRU, in overall, has the lowest performance. We observe that higher α leads to the more significant gap in performance between CafRepCache and state of the art (HyMobi and SocialCache) and benchmark protocols (LRU).

Across the three traces, CafRepCache manages to keep higher success ratio and lower delays compared to the other protocols due to the following two reasons: First, CafRepCache is able to predict regional in-network delays with good accuracy because of checking for the in-network delays of both the nodes and their ego networks. In this way, CafRepCache is able to more quickly identify (potentially) longer but less congested paths with lower delays than the other protocols. Neither SocialCache, LRU nor HyMobi are able to adaptively forward interest and content nor adaptively manage the content chunks. Second, using Social utility as part of the CafRepCache utility, CafRepCache ensures the best prediction of the most direct route to the destination. Neither of the HyMobi, LRU, SocialCache uses predictive social metrics and resource metrics together and are thus not able to adjust to the dynamics in both of these dimensions.

Similarly, across all three traces, CafRepCache sustains higher node and region availability than other protocols. This is because CafRepCache is able to make good predictions of node and ego network availability which avoids depleting the caching resources of frequently used caching nodes and regions in the network that may drop packets. HyMobi, SocialCache and LRU protocols result in lower node caching availability as they congest the regions that are highly central and where the nodes cannot offload the traffic faster than the traffic is generated (that is the example application scenario we are considering). SocialCache and LRU heuristics perform well because they allow congruency with distributed mobile data queries and dynamic interactions while depicting dynamics of the underlying topology (all three topologies have social character

[60, 88] with social metrics being applicable). HyMobi is more successful compared to SocialCache and LRU as it performs in-network predictive resource analytics and rebalances the caching nodes locations so that it avoids congestion while keeping high social metrics to drive caching closer to the subscribers. CafRepCache is most successful as it includes both social and resource metrics, but when the caching node predicts that it is likely to get congested, it delegates caching of the least popular content from its local cache to another node that has most appropriate contact it meets.

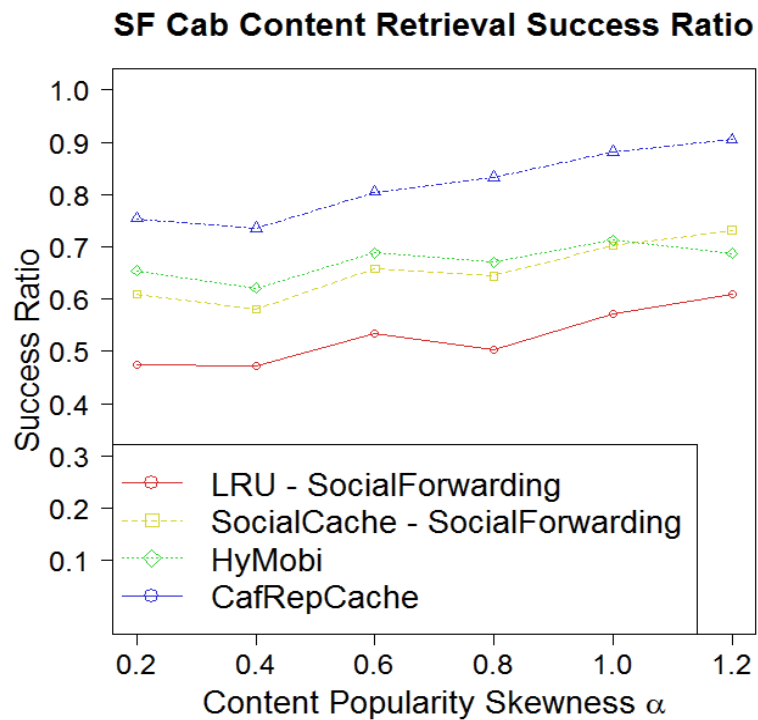


Figure 4.9a Content retrieval success ratio vs. Content popularity skewness (SF Cab).

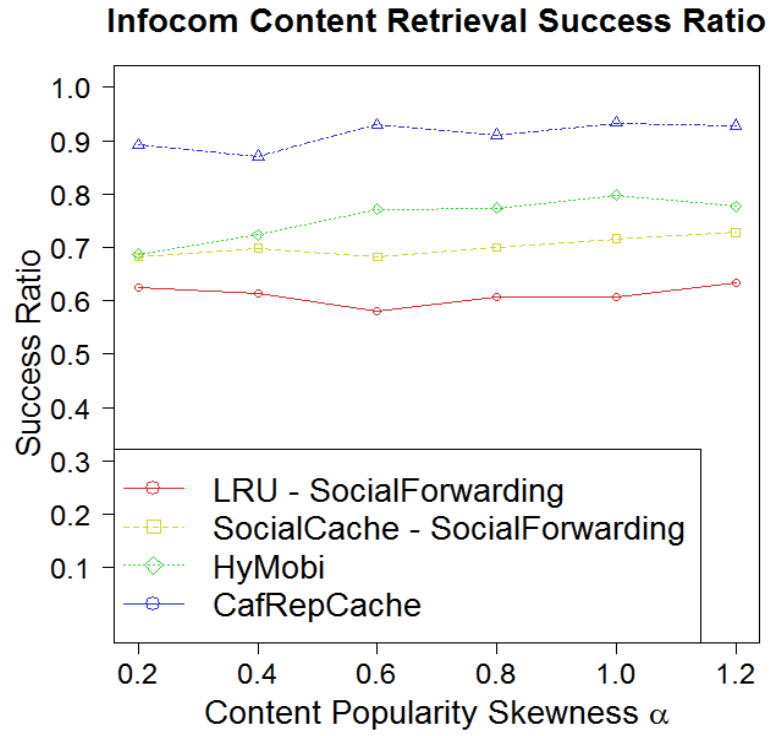


Figure 4.9b Content retrieval success ratio vs. Content popularity skewness (Infocom).

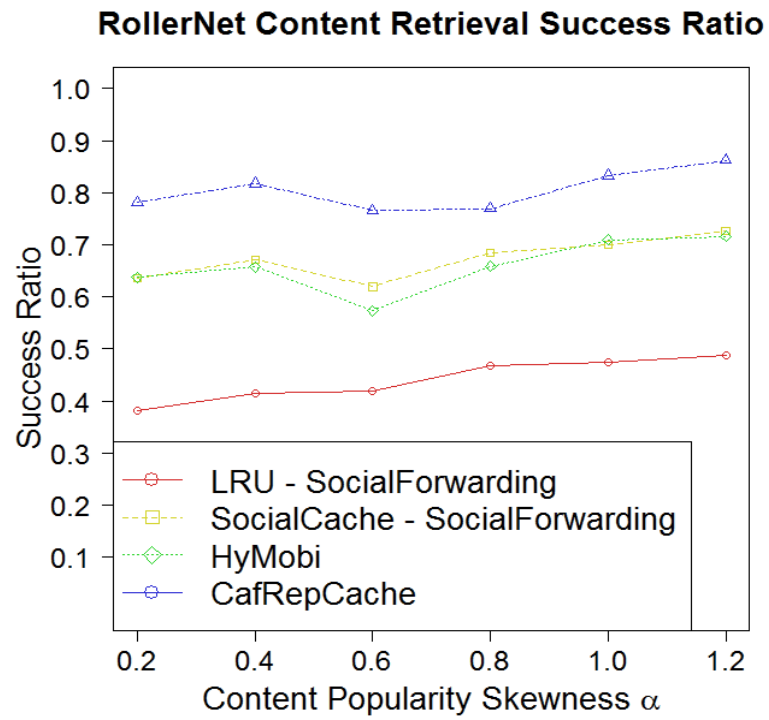


Figure 4.9c Content retrieval success ratio vs. Content popularity skewness (RollerNet).

Similarly to the improved content retrieval success ratio for CafRepCache, content discovery success ratios in CafRepCache outperforms other state-of-the-art (i.e. SocialCache and HyMobi) and benchmark protocols (LRU) across all three traces in dynamically changing content traffic patterns. As we vary the content popularity in each trace, CafRepCache manages to keep above 95% content discovery success ratio while HyMobi and SocialCache managed up to 88% and 76% respectively. HyMobi shows an improvement of 12% compared to SocialCache (without offloading) due to its offloading policy and no content replacement. However, when the cache spaces become limited and the content replacement rate is high, both only removing strategy and offloading strategy result in decreased content discovery. LRU has the worst performance of 75% as it suffers from resource congestion and non-adaptive content management.

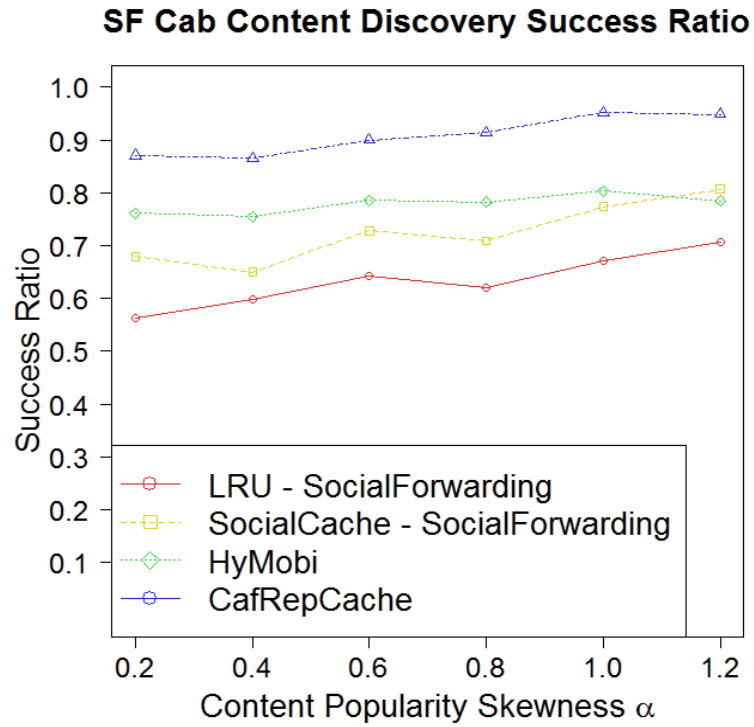


Figure 4.10a Content discovery success ratio vs. Content popularity skewness (SF Cab).

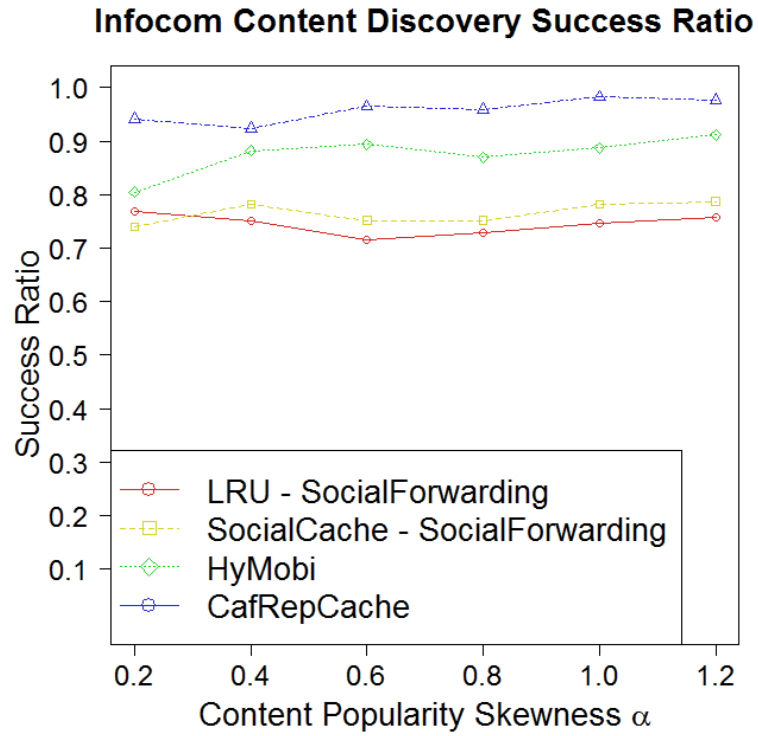


Figure 4.10b Content discovery success ratio vs. Content popularity skewness (Infocom).

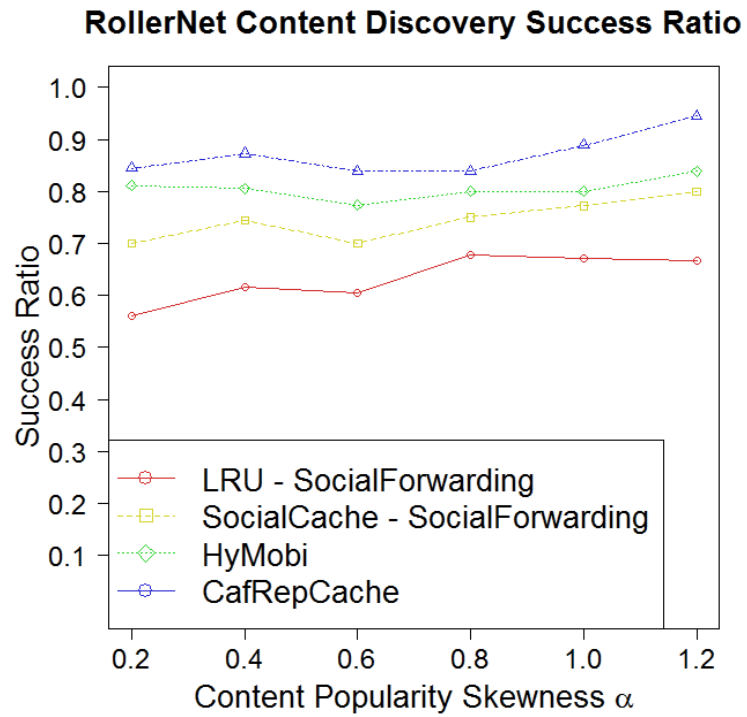


Figure 4.10c Content discovery success ratio vs. Content popularity skewness (RollerNet).

4.3.2 Delay In Content Discovery And Retrieval

We measure the end-to-end delay in content retrieval process as the latency between subscribers sending interest and retrieving the requested contents. In addition to end-to-end delay, we analyse caching performance in deeper detail by measuring the delay to discover contents which denotes the latency time taken between sending interest packets to the network and discovering a requested content in caching points or publishers. Fig. 4.11 and 4.12 show that CafRepCache achieves the lowest overall delay compared to the state of the art caching algorithms SocialCache and HyMobi as well as benchmarking LRU protocol in the heterogeneous network connectivity and publish-subscribe patterns.

We observe that the delays are slightly increased for all protocols when the number of subscribers increases due to the increase in subscribers with lower connectivity degree. CafRepCache reduces delays 20-30% compared with SocialCache, LRU or HyMobi. This is due to CafRepCache managing dynamic real-time trade-off between predictive in-network node and ego network delay analytics and predictive content analytics, and thus enables collaborative adaptive and predictive forwarding, caching and replication necessary for content retrieval in heterogeneous mobile environments.

More specifically, SocialCache uses social driven heuristics for both contacts and contents that exploits contact relationships among nodes and social-based popularity of contents in order to allow optimal directionality, caching opportunity and delivery probability of a node, thus helps to reduce the end-to-end delays. HyMobi uses resource-driven heuristics to offload the traffic from high central nodes and avoid congestion which helps to reduce the packet loss compared to SocialCache but may increase the delays as it does not adapt to regional congestions and may push the contents far away from interests packets.

CafRepCache dynamically combines implicit content, social and resource (buffer availability, in-network delays, congesting rate) heuristics in addition to ego network (regional) driven heuristics that aim to detect and adapt buffer availability, delays and congesting rates of different parts of the network, thus keep the delay and packet loss as low as possible. Selecting the node that

represents the best carrier for the right contents and deciding on the optimal number of replicas to forward and cache are multiple attribute decision problems across multiple measures that CafRepCache is able to deal with, where the aim is to select the best contents to cache, the most suitable nodes to carry the contents and the number of cached contents that provide the maximum utility in the presence of dynamic content workload, connectivity and resources in heterogeneous mobile environment.

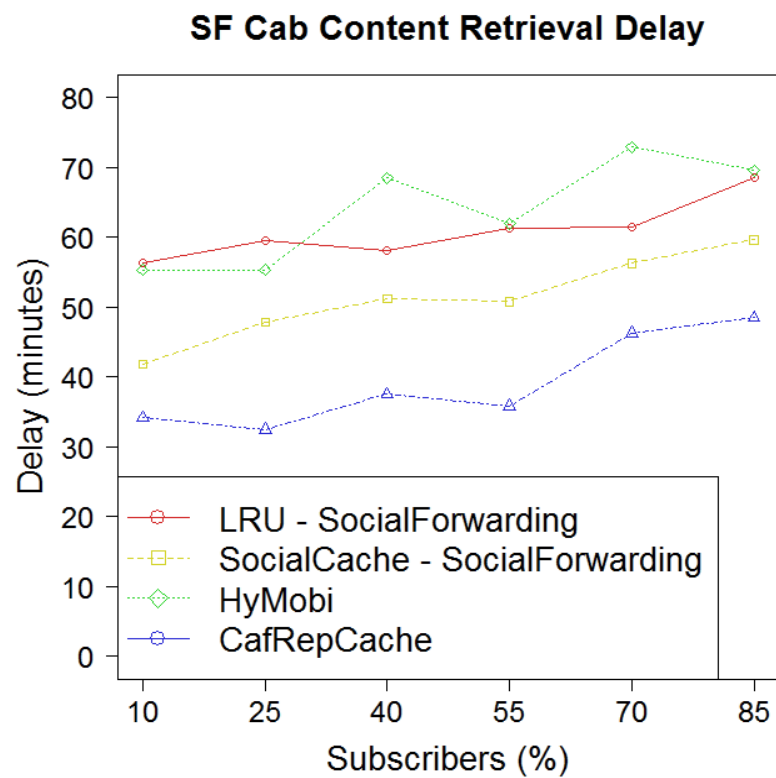


Figure 4.11a Content retrieval delay vs. Number of subscribers (SF Cab).

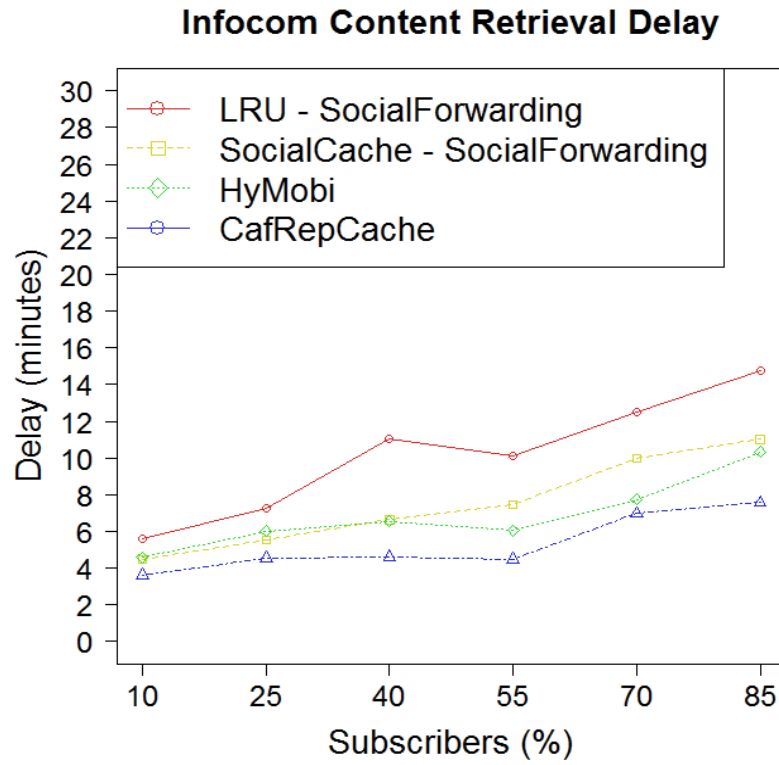


Figure 4.11b Content retrieval delay vs. Number of subscribers (Infocom).

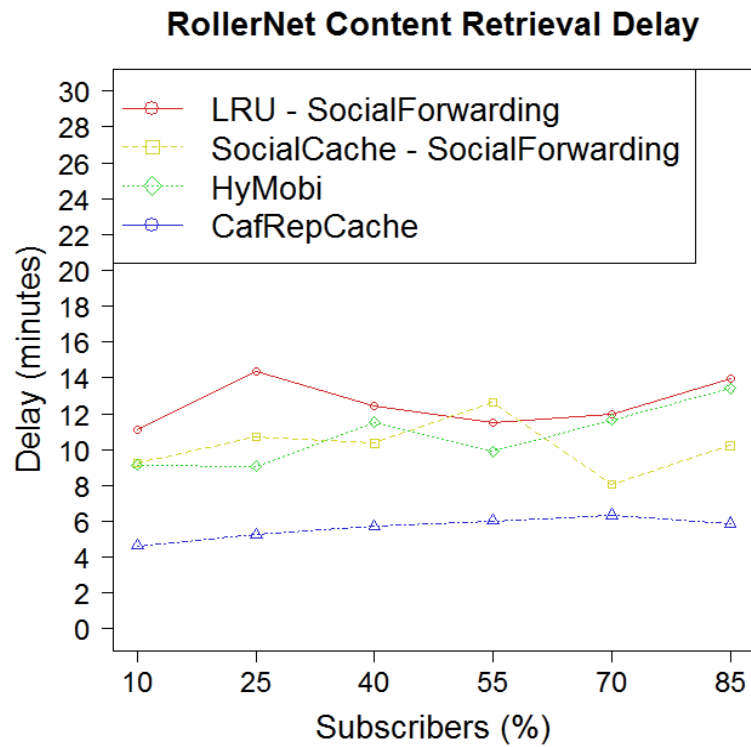


Figure 4.11c Content retrieval delay vs. Number of subscribers (RollerNet).

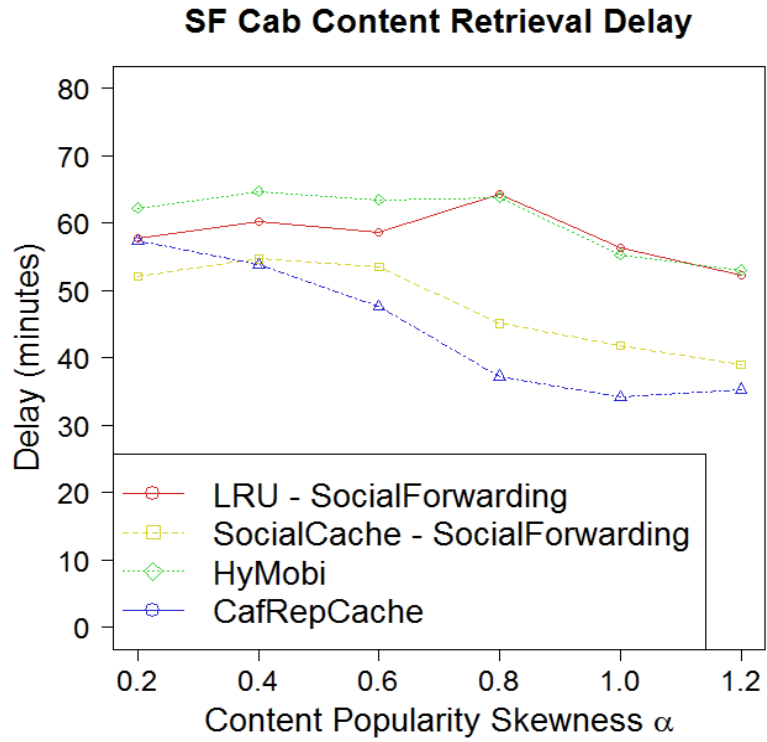


Figure 4.12a Content retrieval delay vs. Content popularity skewness (SF Cab).

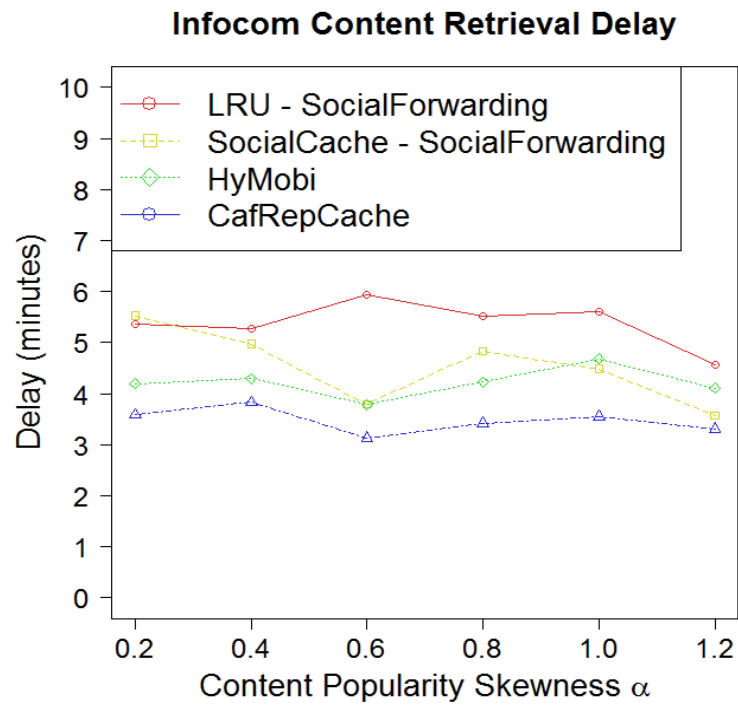


Figure 4.12b Content retrieval delay vs. Content popularity skewness (Infocom).

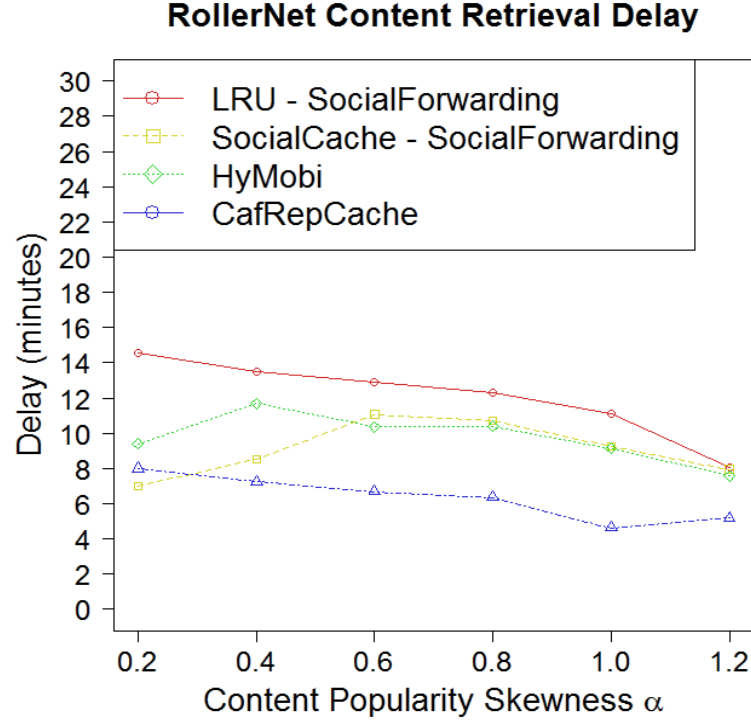


Figure 4.12c Content retrieval delay vs. Content popularity skewness (RollerNet).

Figure 4.13 shows performance analysis of content discovery delays between CafRepCache, two state of the art (i.e. SocialCache, HyMobi) and one benchmark algorithms (LRU) over three traces and varying content popularity skewness. CafRepCache achieves the lowest delay across all three traces and in the face of all very different content popularity patterns (i.e. it manages below 33 min for San Francisco, below 5 min for RollerNet, and below 3 min for Infocom). CafRepCache is followed by HyMobi which manages on average 38min, 5min and 4sec, LRU manages 38 min, 6 min, 6 min and SocialCache (33 min, 5 min and 4 min) for San Francisco, RollerNet and Infocom traces respectively). CafRepCache minimises delays through its predictive analytics of in-network delays and predictive node and ego network congesting rates. This in turn also enables the highest cache hit ratio and allows the requested contents to be mainly found in the caching points in CafRepCache (thus avoiding the need for getting the content from the publishers directly). Although HyMobi has slightly higher discovery success ratio than SocialCache, it has higher latency when finding the requested content because the content may be

pushed farther away from the subscribers during the offloading process. Our extensive experiments have shown that the performance of content delivery from caches to subscribers is significantly higher than that of content discovery (e.g. the delay in content delivery is much lower than that in content discovery). The overall content retrieval performance (success ratio, delay, packet loss) is thus significantly affected by the content discovery process, which in turn, is affected by the cache hit ratio for all three traces.

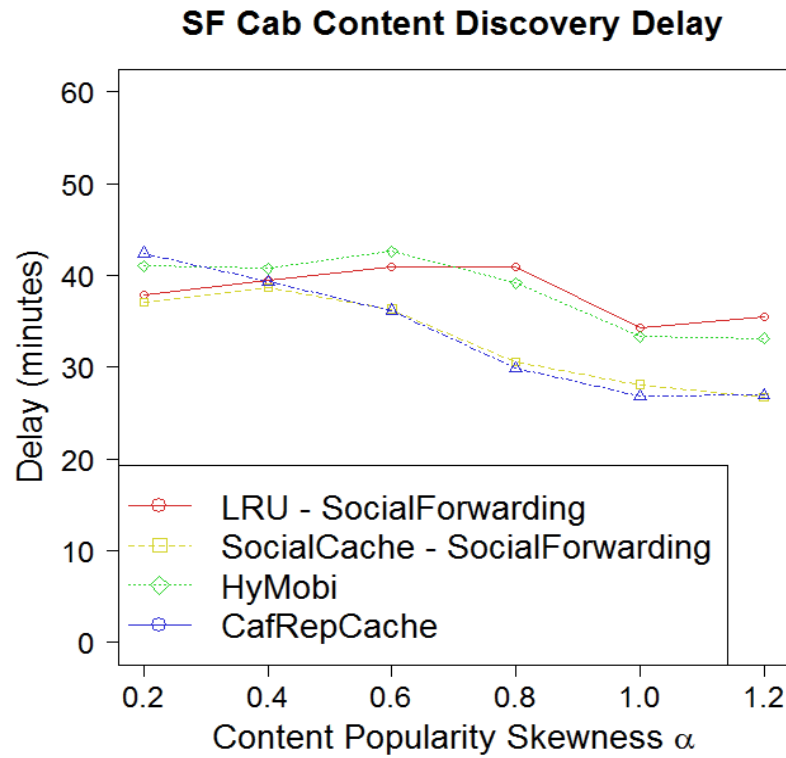


Figure 4.13a Content discovery delay vs. Content popularity skewness (SF Cab).

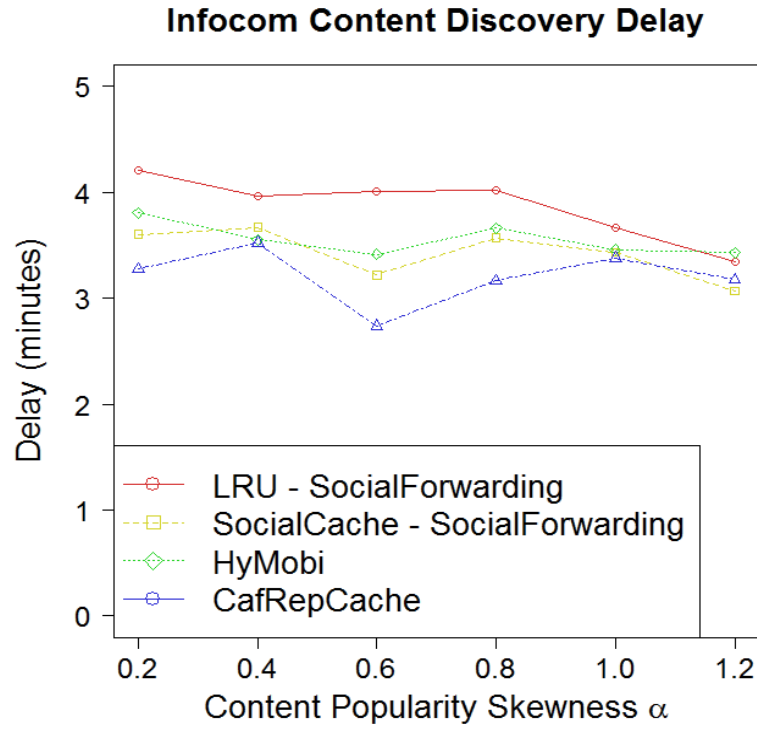


Figure 4.13b Content discovery delay vs. Content popularity skewness (Infocom).

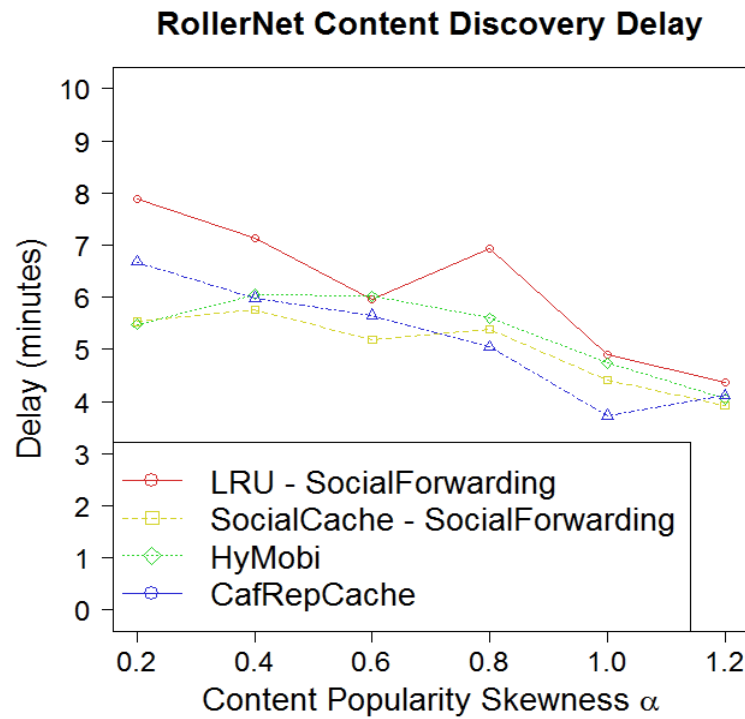


Figure 4.13c Content discovery delay vs. Content popularity skewness (RollerNet).

In the more detailed analysis, Fig. 4.14 shows the importance of CafRepCache caching by comparing delays between requesting content with and without caching points in San Francisco Cab dataset (note that the results are similar with RollerNet and Infocom). We show that caching helps to significantly reduce the delays compared to when there are no caches or cache miss scenarios meaning interests have to traverse the network in order to be matched with the content at the publishers. As shown in Fig. 4.14, the average delay to discover content when there is a cache hit is less than 25 minutes (green bar) compared to 37 minutes when there is no cache match happened (cache miss – blue bar). In addition, the average delay to deliver contents from the caching points is less than 9 minutes (yellow bar) compared to 16 minutes when delivering content from the publishers (red bar).

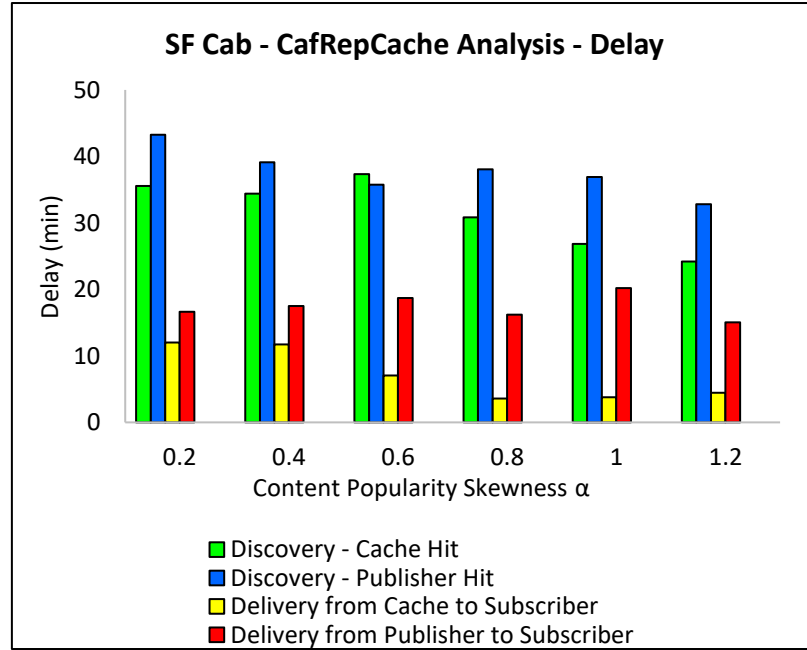


Figure 4.14 Difference in delay in two scenarios: cache hit and cache miss.

4.3.3 Packet Loss In Content Discovery And Retrieval

We measure the packet loss rate of CafRepCache and other competitive caching algorithms. In mobile opportunistic networks, packet loss may occur due to many reasons such as lack of buffer availability or nodes disconnection, etc. Fig 4.15, 4.16 and 4.17 show that CafRepCache achieves the lowest packet loss rates compared to HyMobi, SocialCache and LRU. CafRepCache manages to keep packet loss to less than 10% when serving an increasing number of

subscribers in the dynamic content workload. It is due to CafRepCache's distributed predictive analytics of how likely the nodes and their ego networks are to congest and predictive in-network delay analytics in order to be able to avoid offloading from one node to another node where congestion may happen. SocialCache and LRU have highest packet loss rates as they push the contents to the nodes with the highest social degree and may cause their congestion. HyMobi aims to offload the contents to other nodes which have more resource availability to avoid removing the cached content from the network, however it does not consider how fast these resources are likely to congest and thus may result in delayed congestion. LRU and SocialCache packet loss rates are significantly higher (up to 2 to 3 times higher) than CafRepCache and HyMobi. Because CafRepCache performs adaptive and predictive analytics on the content, connectivity and resource levels, CafRepCache is able to choose the most suitable set of nodes to offload the cached content and the most suitable set of contents to remove from nodes leading to the lowest packet loss compared to the other algorithms.

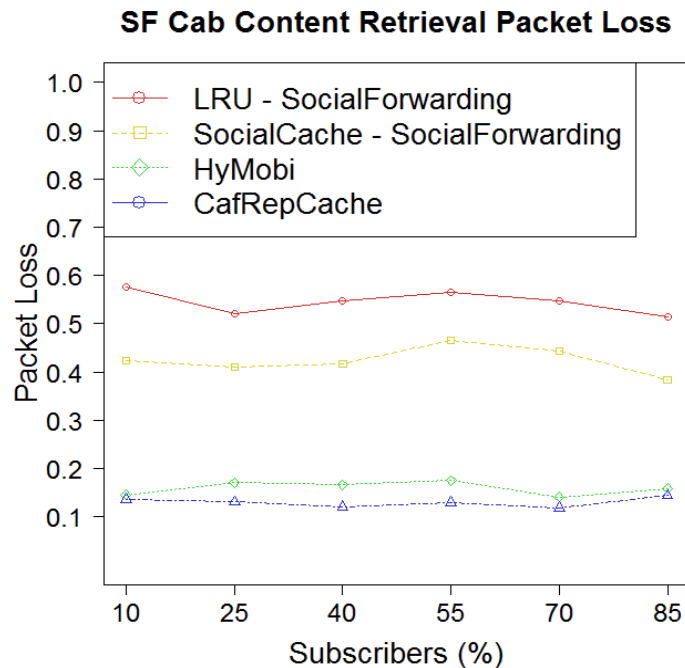


Figure 4.15a Content retrieval packet loss vs. Number of subscribers (SF Cab).

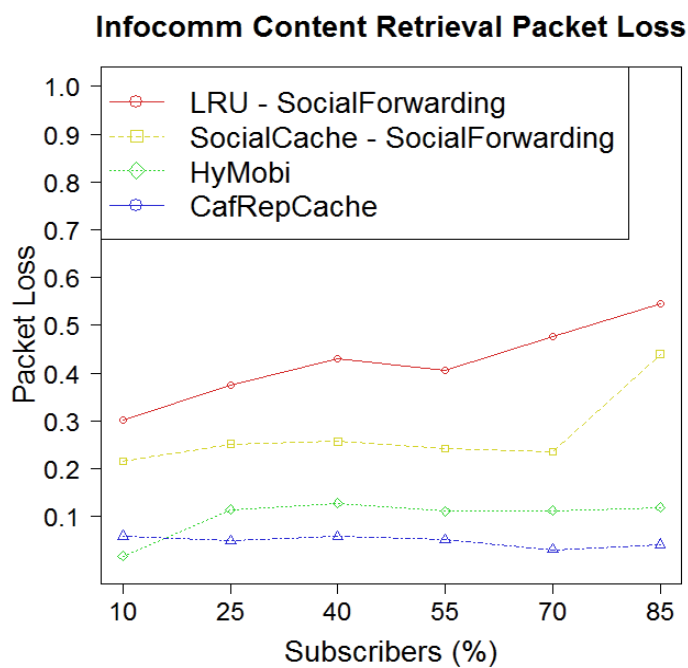


Figure 4.15b Content retrieval packet loss vs. Number of subscribers (Infocom).

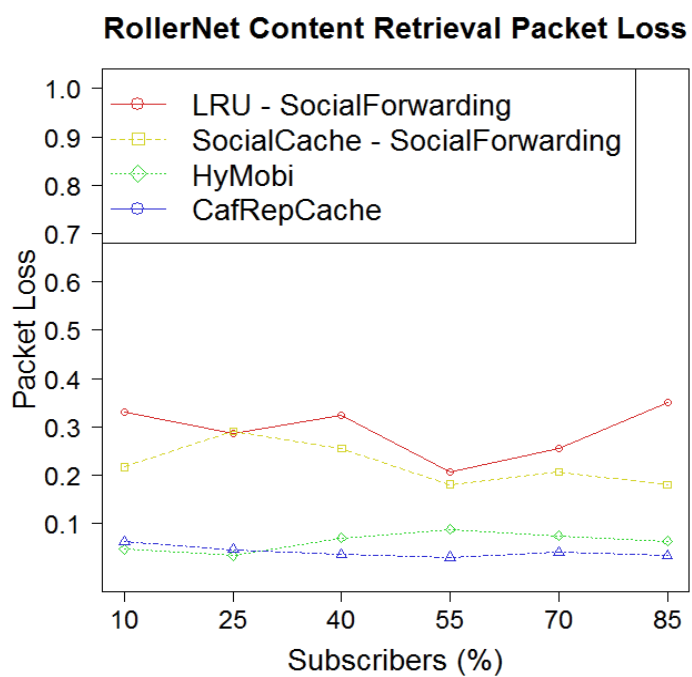


Figure 4.15c Content retrieval packet loss vs. Number of subscribers (RollerNet).

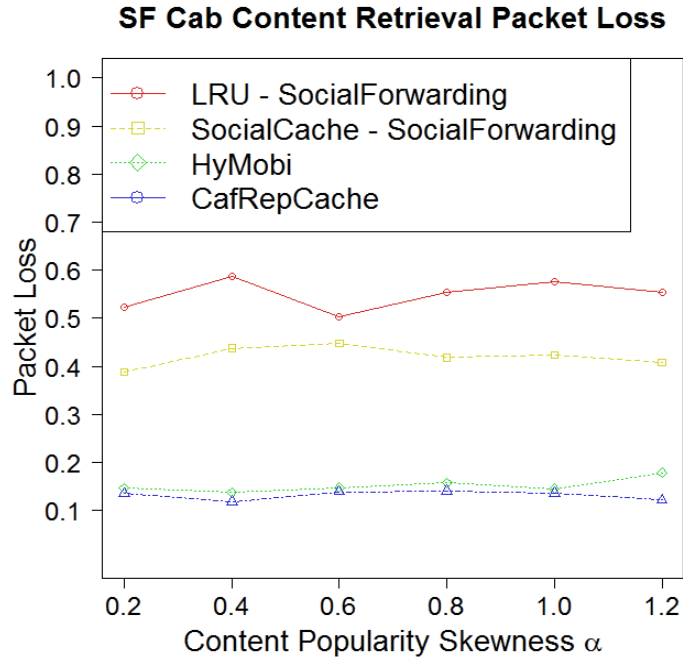


Figure 4.16a Content retrieval packet loss vs. Content popularity skewness (SF Cab).

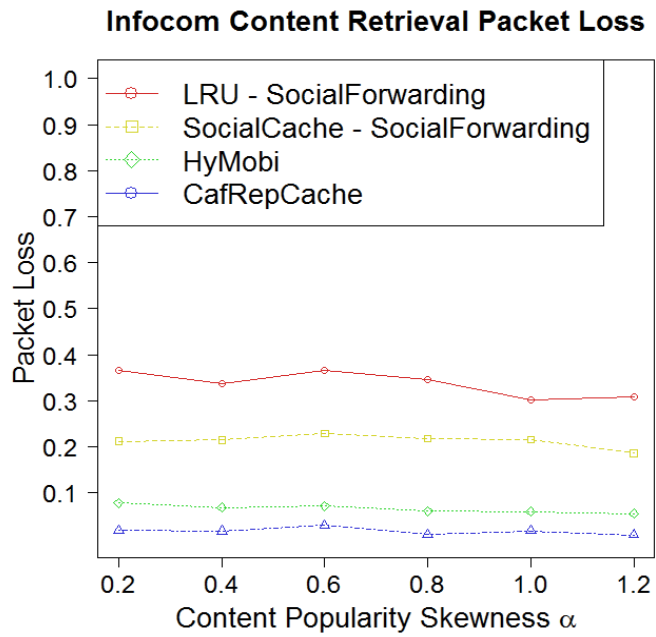


Figure 4.16b Content retrieval packet loss vs. Content popularity skewness (Infocom).

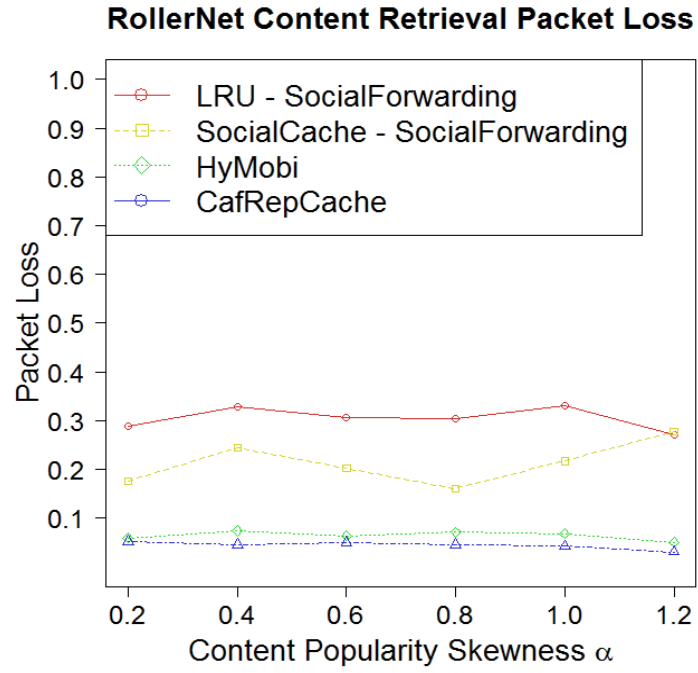


Figure 4.16c Content retrieval packet loss vs. Content popularity skewness (RollerNet).

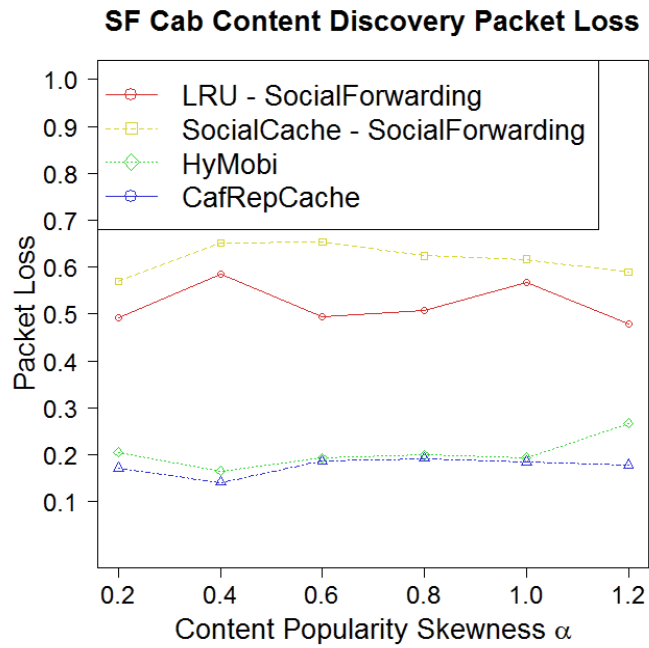


Figure 4.17a Content discovery packet loss vs. Content popularity skewness (SF Cab).

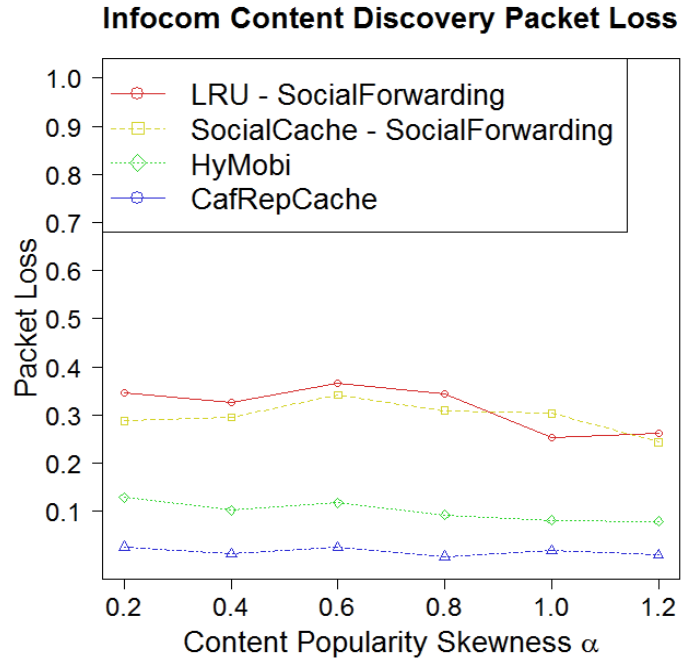


Figure 4.17b Content discovery packet loss vs. Content popularity skewness (Infocom).

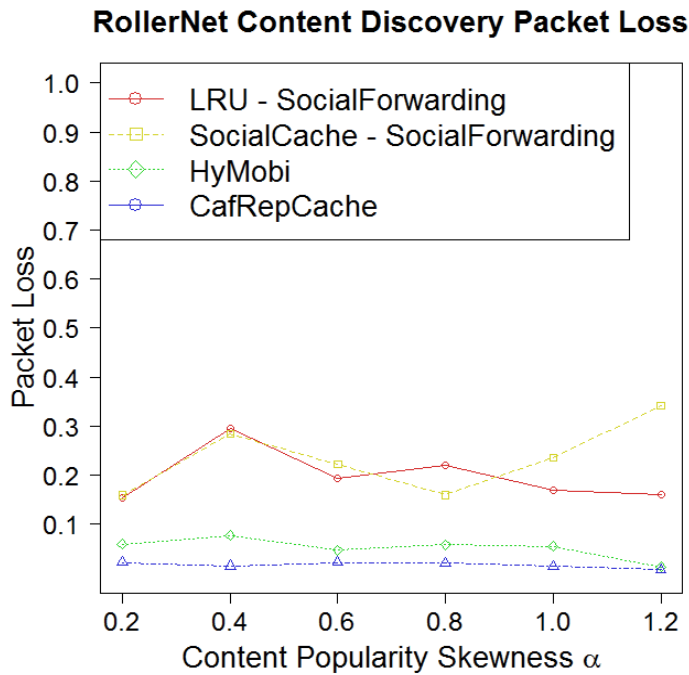


Figure 4.17c Content discovery packet loss vs. Content popularity skewness (RollerNet).

4.4 Caching Performance Evaluation

In the previous section, we analyse CafRepCache performance in different underlying network topologies with different degree of mobility across a range of forwarding-based criteria: success ratio, delay, packet loss. We compare CafRepCache against state-of-the-art caching algorithms with forwarding capability in mobile opportunistic networks. In order to provide more insight and verify the effectiveness of CafRepCache caching decisions, this section analyses CafRepCache performance while varying different heterogeneous content publish-subscribe patterns overlaid on top of dynamic network topologies. We evaluate CafRepCache performance against four state-of-the-art caching protocols: SocialCache [44], LocationCache [108], TLRU [105, 106] and OCPCP [107] across four state-of-the-art caching criteria: cache hit ratio, eviction rate, footprint reduction, hops distance between caching points and subscribers. Each chosen competitive protocol in this section employs different caching strategy whose effectiveness is measured by caching-based criteria (e.g. cache hit ratio) which the forwarding-based criteria cannot measure in deeper detail. We choose Infocom as the underlying network topology in this set of experiments and vary the content publish-subscribe patterns by dynamically varying the content popularity skewness α , popularity variation, localisation factor β , cache size and number of subscribers in Foursquare datasets [118, 119].

In the previous section, we have run six increments of the number of subscribers and publisher ranging from 10% to 85% of the total number of nodes to evaluate the impact of heterogeneous network connectivity and dynamic workloads on caching performance. In this section, to provide a deeper understanding of how content traffic patterns influence on CafRepCache caching decisions, we fix the constant number of subscribers (55% of nodes) and publishers (25% of nodes) and report on experiments with dynamically changing content request patterns, but note that the results for different number of subscribers or publishers are similar to the ones presented here. The total number of nodes in Infocom is 62 node. All simulations are repeated 10 times to give sufficient statistical results. In Foursquare datasets, we choose the total number of contents which are requested by mobile subscribers is 10^3 . We keep the number of content request

constant as the content workload is already controlled by other parameters such as popularity skewness, localisation factor, etc.

4.4.1 Cache Hit Ratio

Cache hit ratio refers to how many interest packets are matched with the contents in caching points without being forwarded to publishers. The higher cache hit ratio, the higher efficiency of caching decisions. We extract different request distributions from Foursquare dataset following different skewness parameter α [25] to vary the dynamically changing publish-subscribe patterns. We run six increments with content popularity skewness α ranging from 0.6 to 1.1. This allows us to evaluate CafRepCache caching performance in the presence of high degree heterogeneity in content publish-subscribe patterns. We keep the content request rate per second following real-world Foursquare dataset [118, 119].

The figure 4.18 shows that our cognitive caching achieves the highest cache hit ratio (ranging from 31% to 86.4%) compared to other competitive algorithms (OCPCP, TLRU, SocialCache, LocationCache) regarding the increase of popularity skewness α . We observe that higher α leads to bigger gap between CafRepCache and others. CafRepCache is followed by LocationCache and SocialCache that manage up to 76.6% cache hit ratio regarding the increase of content popularity skewness. This is because the CafRepCache request temporal heuristic takes advantage of highly skewed content popularity and content request temporal locality to predict efficiently the incoming content requests and adapts more accurately with the content requests while others neglect or could not adapt with the temporal locality of content. We observe that LTRU and OCPCP achieve the lowest cache hit ratio (ranging from 24,3% to 60%) as it relies only on simple request frequency or recency metrics which are not sufficiently fine-grained, thus could not be able to predict adaptively the temporal locality of content request patterns.

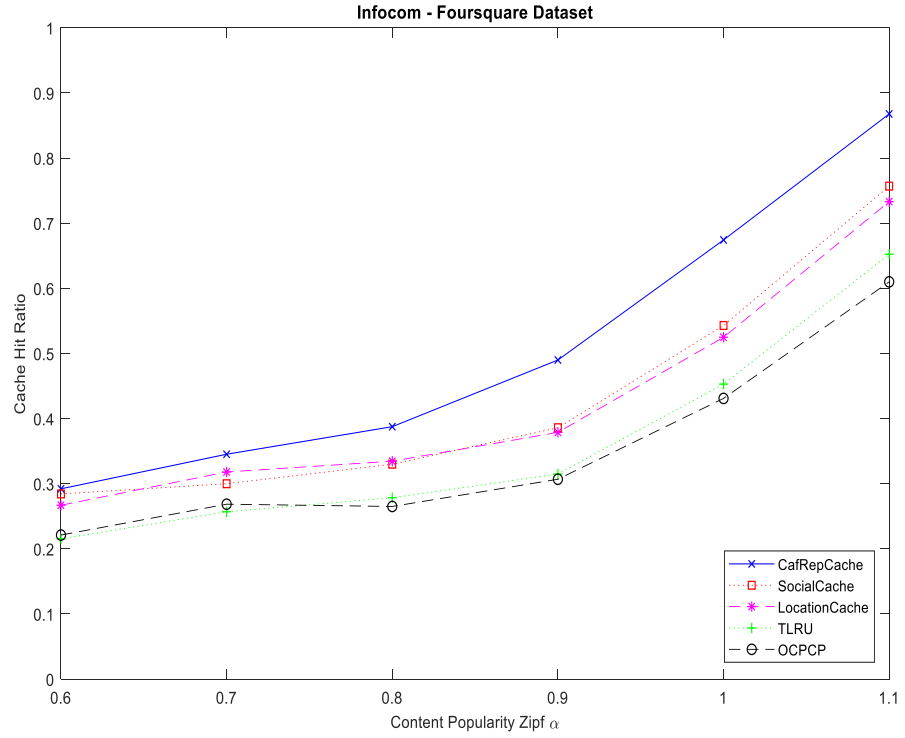


Figure 4.18 Cache hit ratio vs. Content popularity skewness α .

In addition to evaluate the impact of CafRepCache’s temporal heuristics by varying the content popularity skewness parameter α , we evaluate the request spatial-based heuristics integrated in our caching algorithm by varying the spatial localisation factor β (ranging from 0.1 to 0.85) utilized in Hawkes process [25] and measure the cache hit ratio. Low localization factor β means that content requests are generated independently with locations. With a localization factor of 0.85, other near nodes in a region more likely request the same content after a node requests it as similar to the original Foursquare dataset. Fig. 4.19 shows that the CafRepCache’s spatial clustering heuristic enables CafRepCache to adapt with the spatial locality of content requests. CafRepCache achieves the best performance (around 87% cache hit ratio) followed by LocationCache (80%) and SocialCache (72%). TLRU and OCPCP have no (or little) improvement to the accuracy of predicting content popularity, ranging from 52% to 63% cache hit ratio regarding the dynamically changing of spatial localization factor. We observe that higher β even leads to bigger gap between CafRepCache and other state-of-the-art content solutions. It is due to the spatial locality heuristic helps to classify and recognise content interests

coming from localised group of subscribers, then adaptively predict that the contents will be requested again by other subscribers within that location.

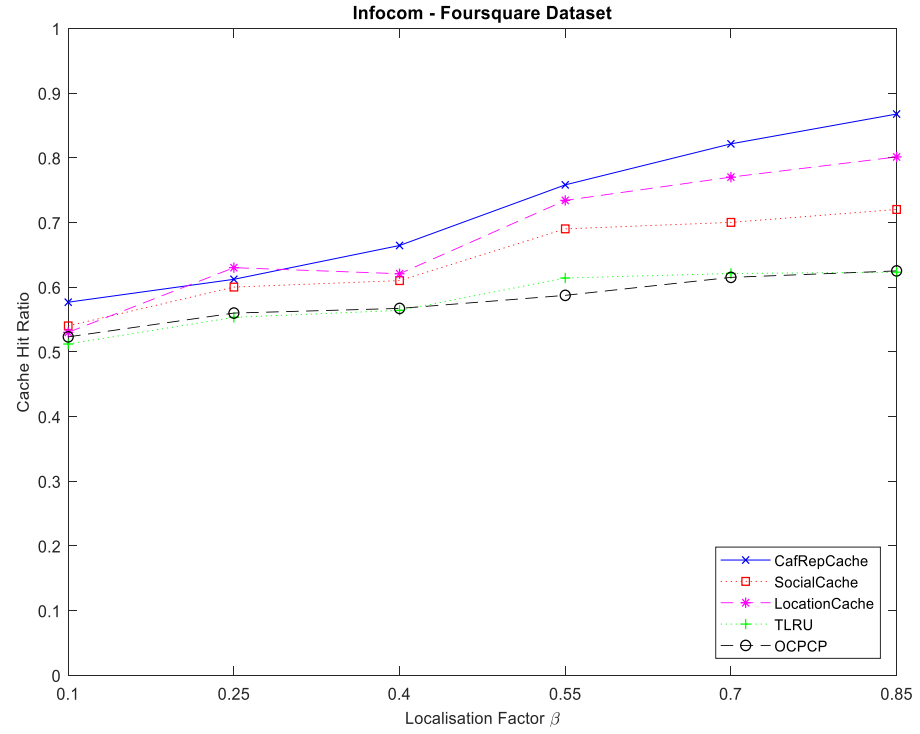


Figure 4.19 Cache hit ratio vs. Spatial localisation factor β .

We further study how quickly CafRepCache responds to the rising trend of newly high popular contents and fading out of older contents over time compared to other competitive protocols (i.e. SocialCache, LocationCache, TLRU and OCPCP). We argue that it is challenging for the caching algorithms to adapt quickly to the severe and quick popularity alteration as the content popularity changes dynamically in real time regarding the variation in user interests. In Fig. 4.120, we increase the popularity variation degree and evaluate CafRepCache performance against other competitive protocols. The higher popularity variation degree, the quicker it is for a content to become highly popular or stale.

As shown in Fig.4.20, CafRepCache's request recency metric keeps good stability regarding the popularity change (typically above 83.1% cache hit ratio), predict adaptively the emerging contents that may become highly popular in near future and contents that are currently considered as high popular but will be less popular soon. The sensitiveness of our proposed algorithm provides

better adaptation to rapidly changing content popularity and different network environment while all other competitive caching algorithms could not be able to adapt with sharp variation in popularity. As the results, CafRepCache is able to take advantage of highly skewed content popularity and content request locality to efficiently predict the incoming content requests, thus cache the right contents in suitable locations that helps to increase the overall success ratio and reduce the end-to-end latency.

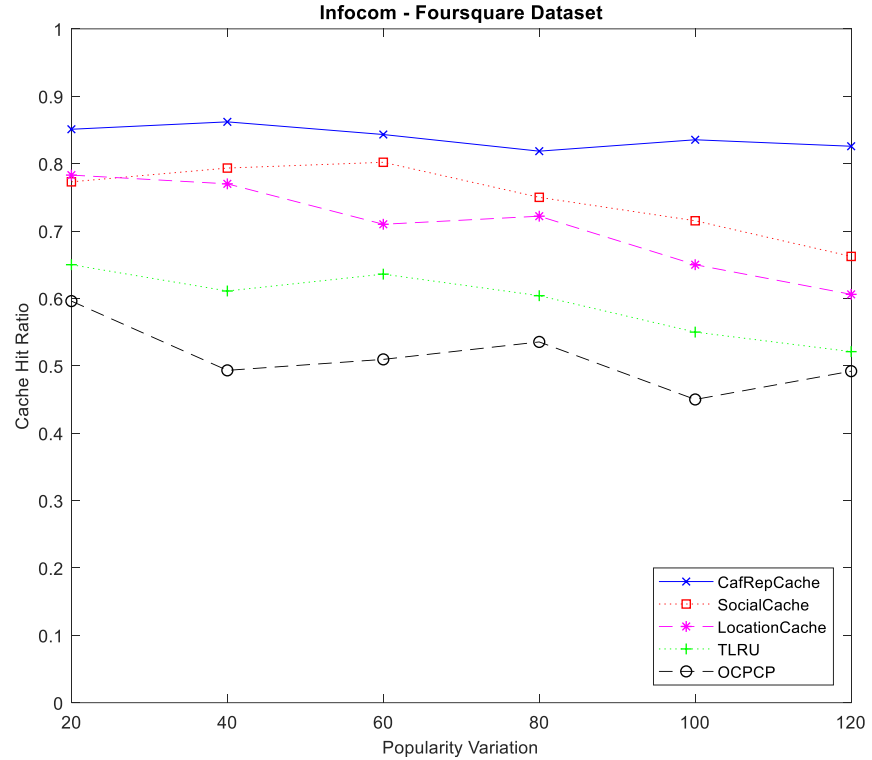


Figure 4.20 Cache hit ratio vs. Popularity variation degree.

4.4.2 Eviction Rate

Eviction is one of the popular metrics for the performance evaluation of content caching. When the cache space is full and caching point makes a decision to cache a new arrived content then one of the cached contents is evicted (i.e. removed) or offloaded to free up the buffer. If a stored content is evicted incorrectly and a new request arrives for it, then there is a cache miss and the content has to be retrieved from other nodes or publishers directly. Eviction rate helps us to measure caching performance in cache pollution conditions [33, 127] due to one-timer contents [33, 127]. One-timer contents are contents that have been requested once, while cache pollution occurs when one-timer contents are

cached. In other words, long-term popular contents will be evicted incorrectly. Since one-timer contents have been estimated to even up to 70% of the content requests [33, 127] and content caching is expected to serve a higher number of contents under more severe constraints such as limited cache size, cache pollution becomes an important complication that a variety of caching policies without intelligent content popularity prediction such as the fixed-probabilistic caching policies and the static local-based caching policies [16] fail to address.

High eviction rate due to one-timer contents affects the overall network throughput in terms of cache hit ratio and content retrieval latency. It increases the content retrieval latency and decreases the cache hit ratio of the caching services. Smaller cache buffer size offers more selective cached contents, thus requires more accurate content popularity prediction.

We vary the cache size and measure the cache eviction ratio to evaluate the impact of CafRepCache request betweenness heuristic in avoiding one-timer contents and mitigating flash crowd effect. The figure 4.21 shows that CafRepCache achieves the lowest eviction ratio regarding the dynamic changing in size of the caching points (decreasing from 0.62 to 0.34 eviction ratio when the cache size is increased) compared to the state-of-the-arts caching algorithms. CafRepCache is followed by SocialCache and LocationCache (ranging from 0.75 to 0.46 eviction ratio) while TLRU and OCPCP have the worst performance, especially when the cache space is relatively small. This is due to the request betweenness heuristic allows CafRepCache balance the trade-off between current observed content popularity versus long terms interest in order to avoid caching quickly potentially one-timer contents or fake news and losing the long-term useful contents. As the results, it helps CafRepCache to reduce the content retrieval latency and increase the overall success ratio.

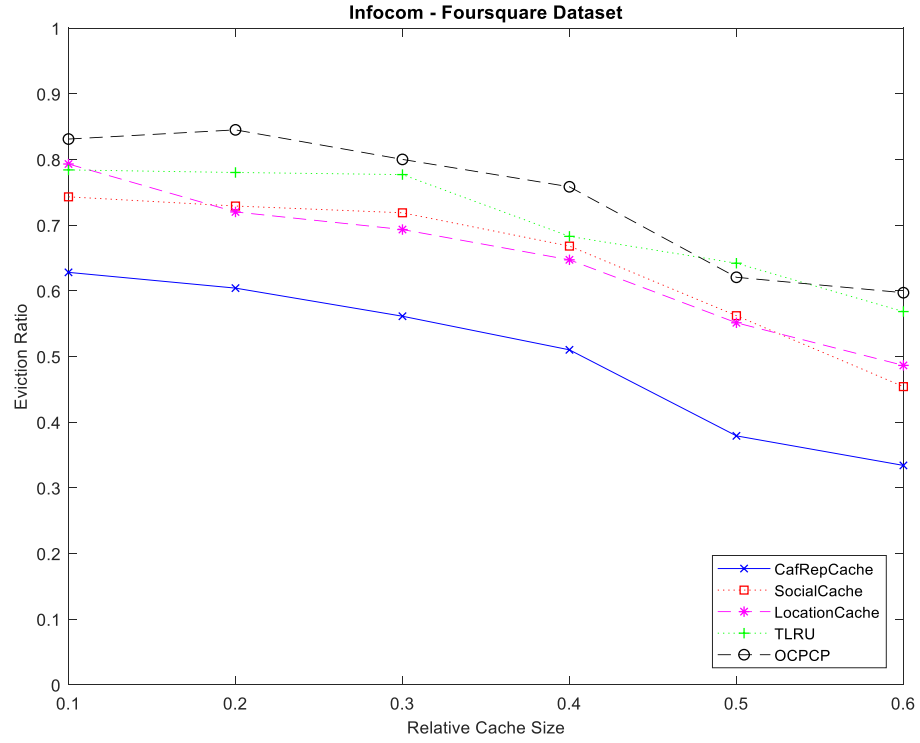


Figure 4.21 Eviction ratio vs. Relative cache size.

4.4.3 Number of Hops from Caching Points to Subscribers

One of the most important criteria in caching decisions is to cache the content as close as possible to the subscribers that requires less hop for the interests to reach the caching points and for the requested content to traverse back to the subscribers. Furthermore, the number of hop count between caching points and subscribers are strongly influenced by connectivity patterns (i.e. how central subscribers are in the network) and their locations (how close to the publishers and to each other subscribers are). Therefore, we evaluate the effect of subscriber-caching point connectivity on the performance of different state-of-the-art caching protocols in order to evaluate cache placement effectiveness and understand the nature of caching points regarding the dynamic connectivity of mobile subscribers.

Fig. 4.22 shows that CafRepCache outperforms other competitive caching protocols (i.e. SocialCache, LocationCache, TLRU, OCPCP) in terms of average delay measured by the number of hops. CafRepCache is able to bring the cached contents to only a few hops away from subscribers (2.7 hops) regarding the dynamic centrality of caching points. This is due to CafRepCache benefits from multidimensional multilayer analytics that allow it to place the

most suitable set of contents in the most suitable set of caching points which are not only highly central but also have similarity in contacts and interest requests with the subscribers. We also observe that the request spatial distance metric favours the contents requested from multiple subscribers which are far away from the caching points, thus help to avoid only serving subscribers from specific regions.

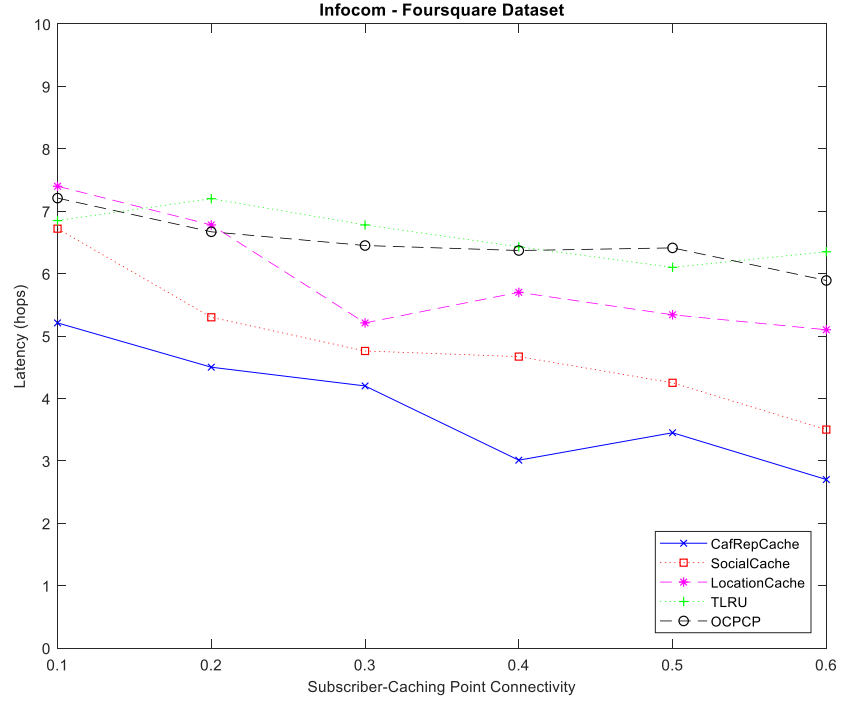


Figure 4.22 Average latency (by hops) vs. Subscriber-caching point connectivity.

4.4.4 Footprint Reduction

We analyse CafRepCache caching performance and other competitive protocol using footprint reduction metric. Note that we refer to relative footprint reduction metric rather than footprint reduction metric proposed in [128] as it is more suitable for mobile heterogeneous opportunistic network environments. Relative footprint reduction is measure as:

$$\text{Relative footprint reduction} = \text{success ratio} \cdot \left(1 - \frac{\text{hop count}}{\text{hop count when no cache}}\right).$$

where average hop count and footprint reduction are considered to be two key metrics for evaluating cache performance.

Higher footprint reduction value indicates higher cache hit ratio and traffic reduction. As shown in the formula above, a solution for improving the relative footprint reduction is to place the popular content as close to subscribers as possible that also means lower average hop count.

Figure 4.23 shows that CafRepCache outperforms HyMobi, SocialCache and LRU in terms of footprint reduction. This is because we not only get highest success ratio, but also cache the content as close as possible to the subscribers that requires less hop for the interests to reach the cache and for the requested content to traverse back to the subscribers.

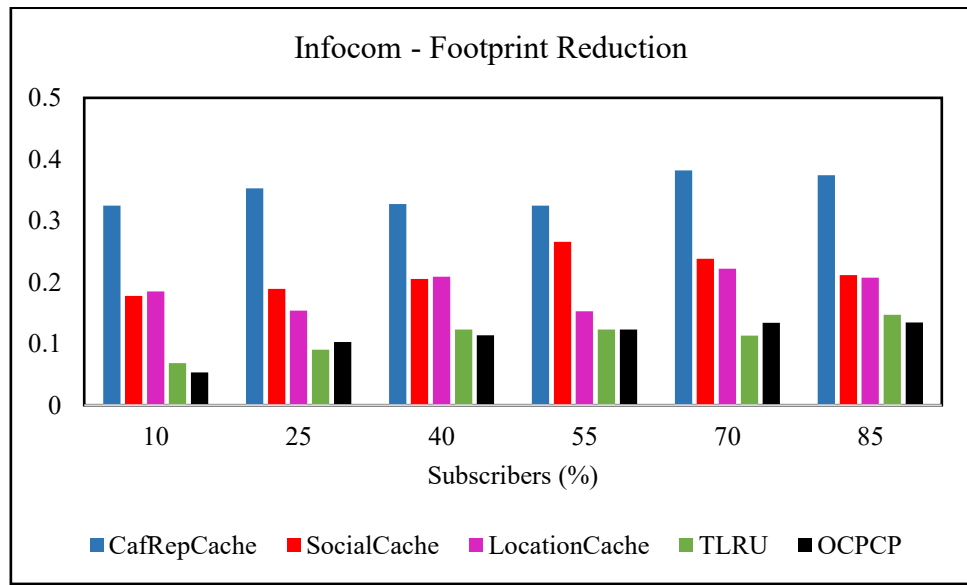


Figure 4.23 Footprint reduction vs. Number of subscribers.

4.5 Network Resource Utilisation Evaluation

4.5.1 Number of Caching Points

We analyse the number of CafRepCache caching points in both random and scale-free topologies to understand the nature of CafRepCache caching placement regarding the increase of content workload due to increasing number of subscribers. As we assume every node in the networks could become a caching service, measuring the number of caching points helps to detect potential universal cache (i.e. cache everything everywhere) [129] which causes unnecessarily redundant traffic [129] and reduces the cache resource utilisation.

Fig. 4.24 shows that the relative number of CafRepCache caching points increases from 0.12% to 0.23% in random networks and from 0.02 to 0.11% in

scale-free networks regarding the growth of subscribers. We argue that random networks with short average paths and low clustering require more number of caching points to serve the dynamic mobile subscribers while scale-free networks with high social character need less number of caching points which converge to 0.11% regarding the increase number of subscribers from 0.4% to 0.6%.

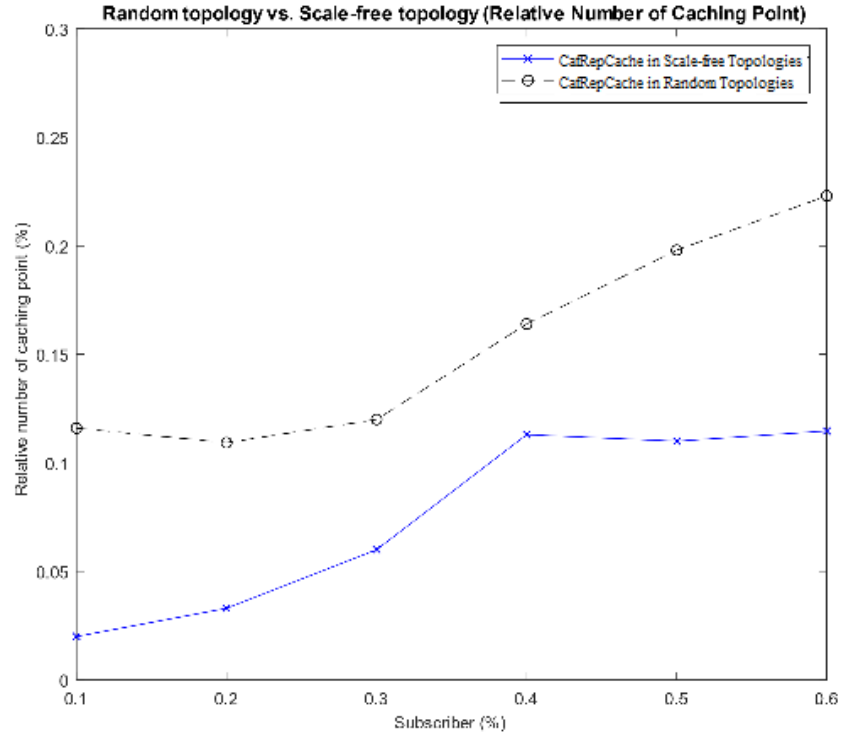


Figure 4.24 Relative number of caching points vs. Number of subscribers in random and scale-free topologies.

For the rest of this section, we discuss the network resource utilisation of CafRepCache in terms of computational costs and communication costs while varying the number of subscribers in three real-world mobility traces: San Francisco Cab, Infocom and RollerNet.

4.5.2 Computational Costs (CPU Usage)

We evaluate the computational cost as average CPU usage of CafRepCache and other competitive protocols. The CPU usage is measured in nanosecond which indicates time taken for each algorithm to make a caching or forwarding decision. The lower CPU usage, the less complexity and higher resource utilisation of a caching algorithm.

We observe that CafRepCache has a small computational increase (up to 11.2%, 20.7% and 27.9%) compared to HyMobi, SocialCache and LRU-SocialForwarding respectively while it has much higher success ratio, lower delay and packet loss compared to the competing protocols.

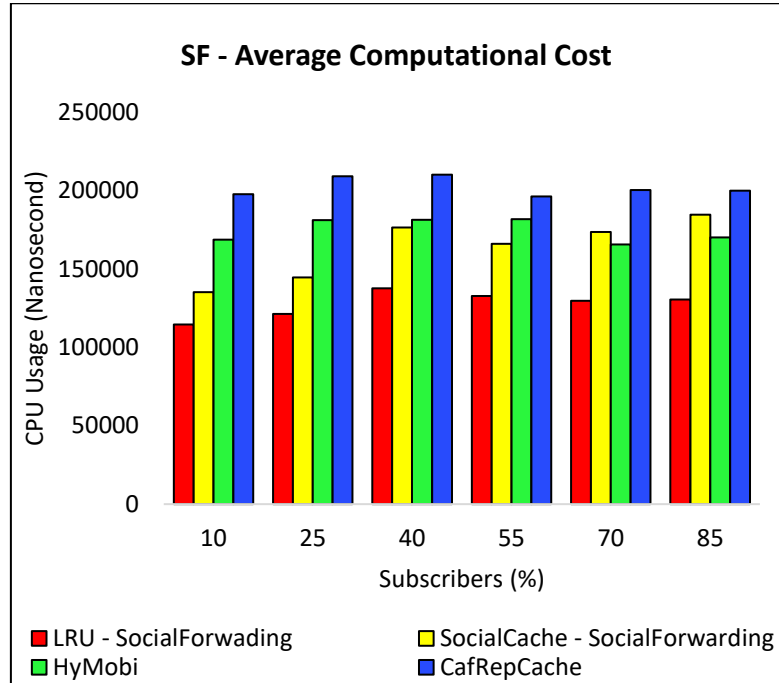


Figure 4.25a Average CPU usage vs. Number of subscribers in SF Cab.

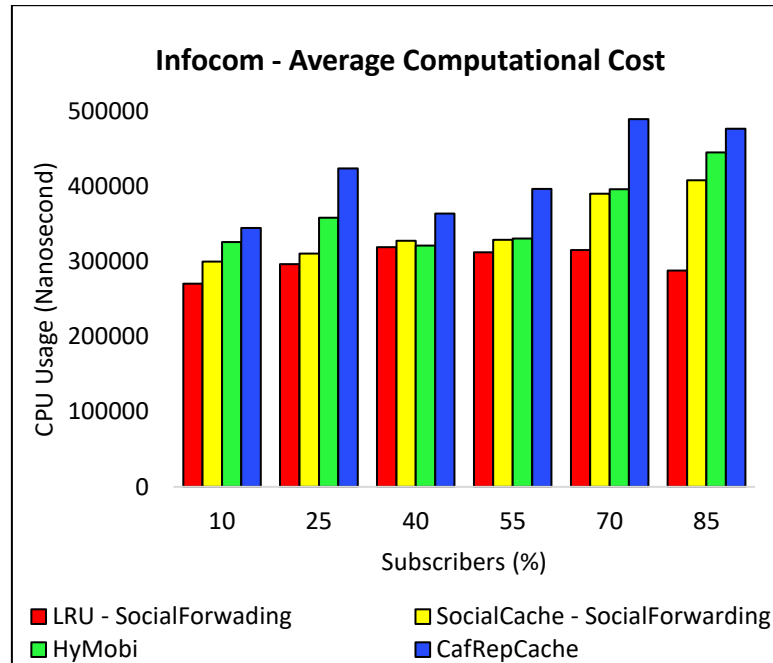


Figure 4.25b Average CPU usage vs. Number of subscribers in Infocom.

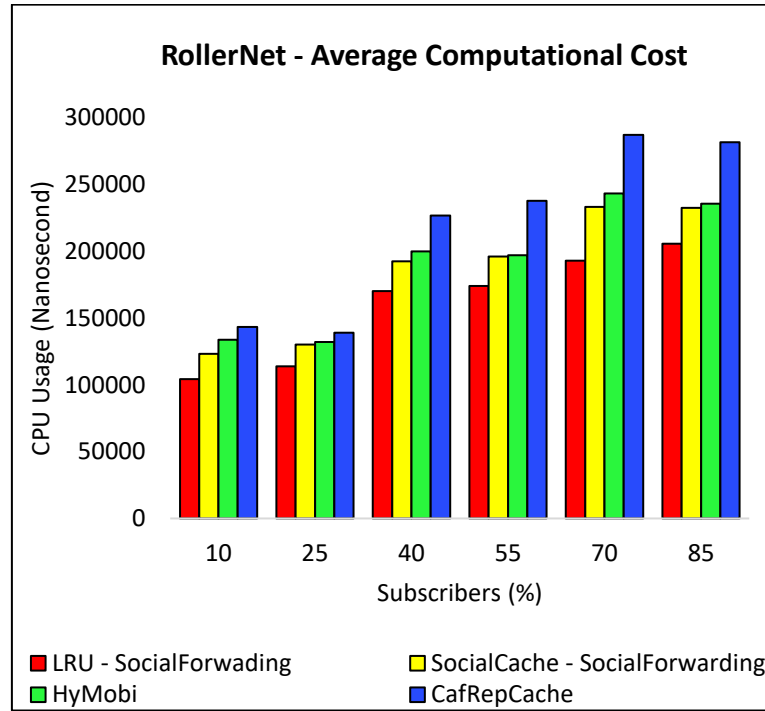


Figure 4.25c Average CPU usage vs. Number of subscribers in RollerNet.

The average computational cost of CafRepCache in Infocom scenario is higher than that in RollerNet and San Francisco Cab. This is due to Infocom traces have more compacted, denser network connectivity where every node have higher number of neighbours with long contact duration. As the result, every caching or forwarding decision in a node involves more neighbouring nodes to coordinate, collaborate and aggregate network information such as regional content popularity, resources, etc, as well as to replicate or offload content/content interests that eventually increases the average CPU usage.

4.5.3 Communication Costs (Number of packets forwarded)

In order to assess bandwidth efficiency, we measure communication costs between nodes in the networks as the number of packets forwarded. This can be interest packets or content packets in the end-to-end content retrieval process. We shows that CafRepCache has up to 18.4% increase in the number of forwarded packets compared to SocialCache and LRU-SocialForwarding. This is due to CafRepCache leverages adaptive replication of content packets and content interests in order to improve the delivery success ratio. In addition, when a cache buffer is full, a content may be offloaded to other neighbouring

nodes that requires higher communication costs to coordinate between the nodes. The number of forwarded packets in San Francisco Cab is higher than that in RollerNet and Infocom because of its highly dynamic topologies and short network connectivity that require higher number of replicated packets. Note that the increase of CafRepCache's communication costs compared to other protocols is only transient in nature when network congestion is predicted and the traffic is adaptively moved to less congested parts of the network (this is a feature which other protocols do not support). Interestingly, CafRepCache has fewer packets forwarded compared to HyMobi (above 10.9%) due to CafRepCache being able to both choose the most suitable set of nodes to forward the cached content to as well as the most suitable set of contents to remove from congested nodes, thus avoid unnecessarily forwarding, offloading packets and reduce the total cost of routing.

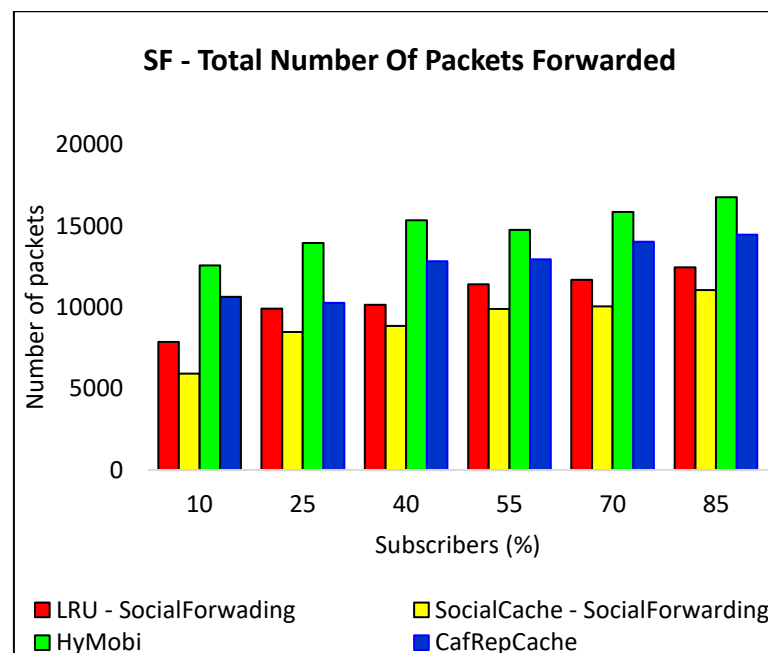


Figure 4.26a Average communication costs vs. Number of subscribers in SF Cab.

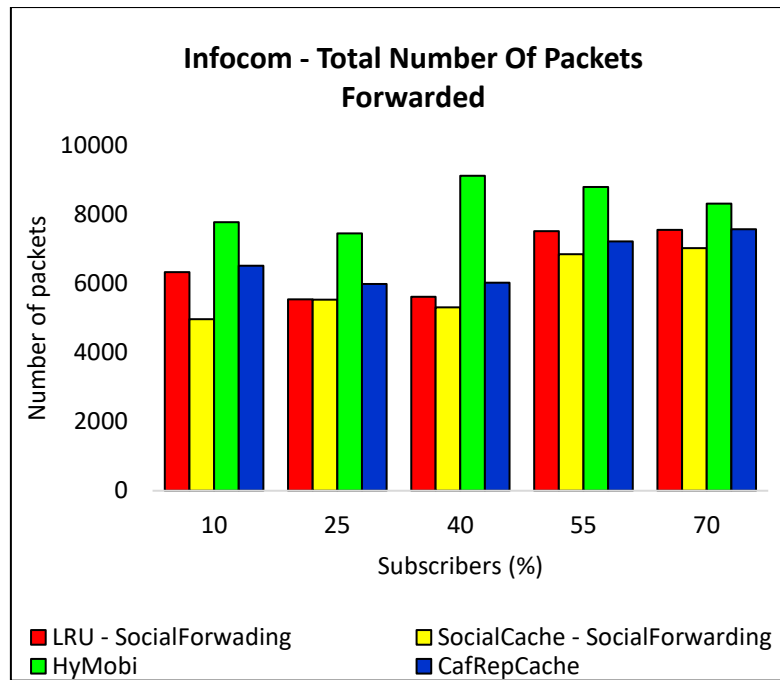


Figure 4.26b Average communication costs vs. Number of subscribers in Infocom.

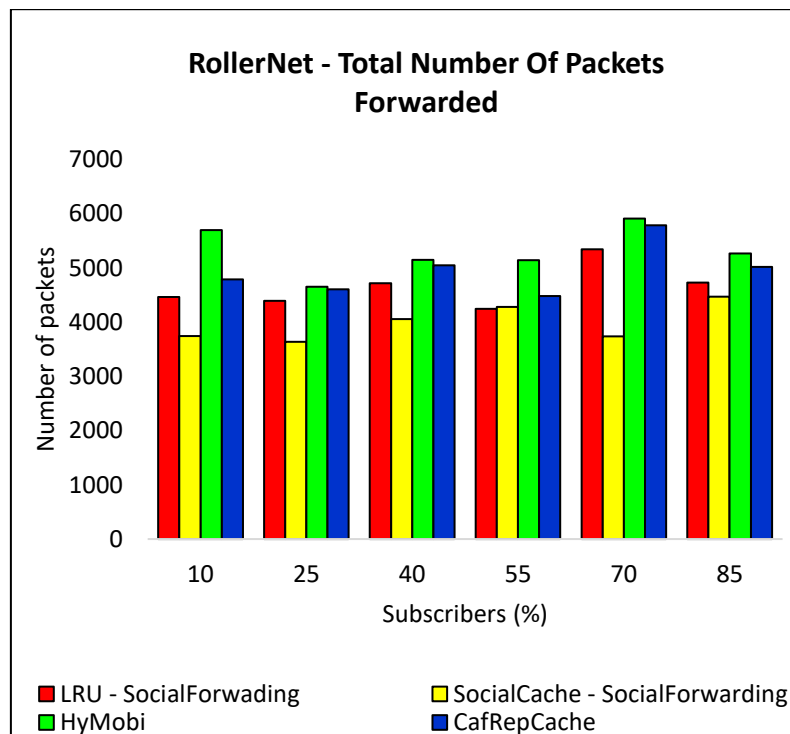


Figure 4.26c Average communication costs vs. Number of subscribers in RollerNet.

4.6 Summary

This chapter describes the implementation and performance evaluation of CafRepCache driven by three main criteria categories: 1) forwarding criteria 2) caching criteria and 3) resource utilisation criteria. We use customised ONE simulator to represent heterogeneous opportunistic network environments with different node mobility, connectivity models as well as various content traffic patterns. This gives us statistically sufficient diversity to evaluate CafRepCache performance in different contexts.

Our results show that CafRepCache significantly improves the performance of content discovery and retrieval over very different time-varying pseudo-realistic and real-world network topologies, mobility and content traffic patterns against state-of-the-art and benchmarking caching algorithms. CafRepCache consistently manages to achieve higher content discovery and delivery rates and better caching efficiency while keeping lower delays, packet loss and resource consumption. CafRepCache's performance improvements are a direct result of its fully localized single node and ego network adaptive real-time collaborative predictive heuristics which are able to better predict and adapt to dynamically varying regional and temporal forwarding possibilities, data interests and resources availability.

In the next chapter, we will discuss how we could use Network Coding to extend CafRepCache framework in order to improve the resource utilisation, particularly reduce CafRepCache energy consumption. The chapter will introduce our first CafRepCache's real-world deployment on MoDiToNeS [130] platform and discuss CafRepCache's applications in different large-scale scenarios.

Chapter 5

5. Discussion

This chapter discusses a possible extension to integrate into our framework, such that to apply adaptive network coding techniques to improve CafRepCache efficiency (Section 5.1), explores the initial deployment of CafRepCache on MoDiToNeS [130] platform (Section 5.2), envisages large-scale application domains of CafRepCache as well as discuss limitations and issues of CafRepCache deployments in these complex temporal network topologies (Section 5.3).

5.1 Network Coding and Energy Efficiency

Network coding for mobile opportunistic networks has attracted significant interest in recent years due to its potentials to improve data transmission efficiency. This section discusses our proposed network coding (NC) mechanism jointly integrated in the CafRepCache caching framework that leverages the spatial-temporal locality of content traffic [9, 10, 85, 118] and energy awareness to achieve fine-grained network coding rate and efficient network utilisation. We extend the idea of CafNC [64] which is the state-of-the-art adaptive congestion-aware forwarding and opportunistic network coding. CafNC [64] is able to exploit the social power-law behaviour of networks to select highly central nodes for coded message relays while utilising resource utilities for congestion control and avoidance. This allows CafNC [64] to achieve high content delivery success ratio while minimizing delay, packet loss, reducing redundancy and waste of network resources. We refine CafNC [64] with our novel spatial-temporal predictive content analytics and heuristics proposed in our chapter 3 as well as an energy-aware network coding rate to adaptively prevent network coding in the parts of the network that have little or no interest in the content and have low energy capacity, thus improve the energy efficiency of communications in mobile opportunistic networks.

Motivated by CafNC [64], CafRepCache is able to perform random linear network coding which encodes together contents that share the same topic or are sent to the same dynamic cluster of subscribers. As the requests are more

likely to be similar in a specific region, CafRepCache's content request spatial clustering heuristics classify, recognize and predict content interests coming from dynamic changing clusters of subscribers. The higher request rate coming from the same localised region of different subscribers, the higher chance that that content will be requested again by other subscribers within that location.

Selecting which node represents the best carrier for the set of content 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 contents. CafRepCache's utility value $Util_h(n, C_i(n))$ as presented in chapter 3 allows to determine the overall improvement an encountered node represents when compared to the sending node, and deciding on which nodes will be selected as next hop and which contents to be coded, cached and forwarded. In the network coding process, a node receives $c1$ (the encoded form of content $o_{k,1}$) directly from a caching point and linearly combined encoded symbol $(cn, \tau_1 c1 + \tau_n cn)$ from another node. All the encoded symbols are the linear combination of the segments of the requested content. A node can retrieve the data by solving a simple system of linear equations. They receive the encoded symbols simultaneously and each node is considered to have been allocated all the network resources. The set of coefficients associated with each coding point is called the coding vector. We define the transform matrices T :

$$T = \begin{bmatrix} 1 & 0 \\ \tau_1 & \tau_2 \end{bmatrix}$$

We consider Wl , as the symbol received by the caching point. The node can then recover all L encoded symbols by solving the following linear system of equation:

$$\begin{bmatrix} c1 \\ c2 \\ cl \end{bmatrix} = T^{-1} \begin{bmatrix} W1 \\ W2 \\ WL \end{bmatrix}$$

As energy efficiency plays a vital role in our solution due to the battery life of users' mobile devices is limited, we propose a $NCThreshold$ parameter to control the coding rate based on energy utility. $NCThreshold$ is a configurable parameter between 0 and 1 measured as the remaining lifetime of a node:

$$NCThreshold = \frac{\gamma E_i}{cost_k}$$

in which E_i is the remaining energy of node n_i , $cost_k$ is the energy cost to transmit and code the content o_k . γ is a control parameter which will be set low if node n_i is an important node and has to be protected from being battery drainage.

$$NCRate = Util_{i,k}^t * (1 - NCThreshold)$$

When $NCThreshold$ parameter is high, CafRepCache reduces the amount of coding a node carries out, and when it is low it relaxes the coding criteria. This results in CafRepCache detecting and exploiting more coding opportunities, on a wider range of nodes which are in reasonable level of energy in comparison with static coding techniques and adaptively codes less at times of extremely low available resources, so that packet loss rates are significantly reduced [9, 60, 64]. At a certain critical level of resources, nodes will not be involved in any decoding or encoding process even if they are interested in the content.

To sum up, for each content in the node buffer, the node periodically captures and predicts neighbouring nodes' interests and calculates their respective relative $Util_{i,k}^t$. Each neighbouring nodes $Util_{i,k}^t$ is weighted by the node reputation [77, 99] for the respective content 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 content is compared to the predefined network coding limit $NCThreshold$. If the number of packets for the content exceeds the threshold, then the node network codes the contents in sets of $NCThreshold$ and sends its coded contents to the corresponding neighbour node. If there are enough contents, CafRepCache forwards the contents without coding them. The caching node generates a coding vector and employs it to do a linear combination of the packets cached that are targeted to the same dynamic cluster of subscribers. The subscribers will collect packets which are linearly independent and as soon as the number of these packets reaches a certain number, the subscribers will decode them and deliver them to the upper layer.

We discuss the performance of CafRepCache (NC) across multiple metrics against state-of-the-art network coded forwarding and caching protocols including HubCode [131], CafNC [64] and CCCN [65]. Our initial experimental result in Fig. 5.1 shows the percentage of packet loss that protocols experience in the face of increased number of subscribers. CafRepCache shows lowest packet loss rates (below 11%) compared to CCCN, CafNC and HubCode. When NCRate is high, the level of energy depleting rate is relatively low, CafRepCache increases the probability of coding with low risk of packet loss. When NCRate is low, there may be a high level of energy depletion, CafRepCache stricts the coding criteria as the risk of high packet loss. In addition, CafRepCache leverages the spatial-temporal locality of content traffic-related users' mobility to choose the most suitable set of contents to be forwarded, cached and network coded. HubCode have highest packet loss rates (55.4%) as it constantly network codes the contents, even in the event of low resource availability which increases significantly the packet loss compared to only forwarding single packet.

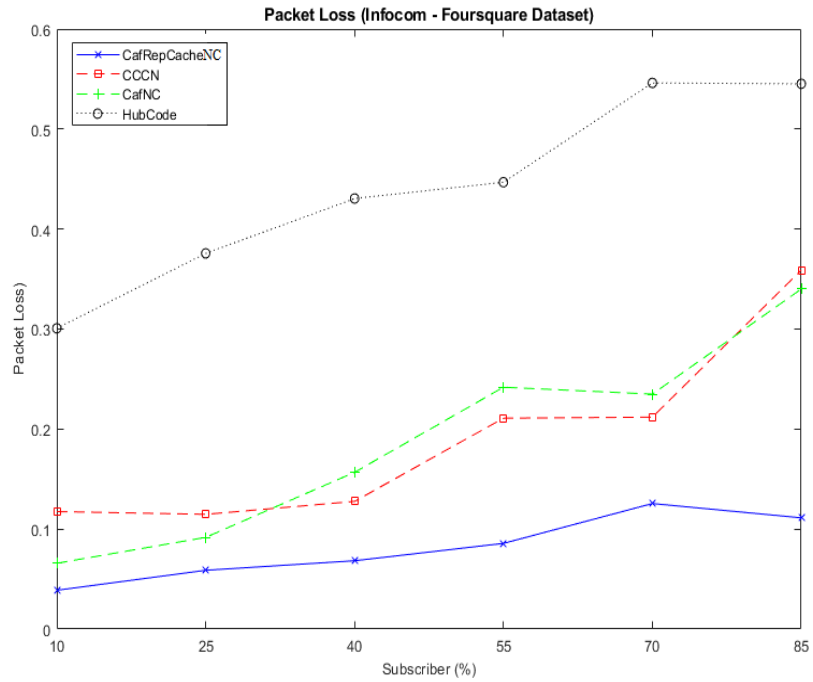


Figure 5.1 Packet Loss vs. Number of Subscribers.

Fig.5.2 and 5.3 show the percentage of time and percentage nodes that perform network coding in CafRepCache, CCCN, CafNC and HubCode in the face of increased number of subscribers. We observe that CafRepCache and CafNC use

18-21% of the nodes to encode traffic 29-35% of the time where HubCode uses 10% of the nodes to encode traffic all of the time (99%) which is 400% worse off than others. 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.

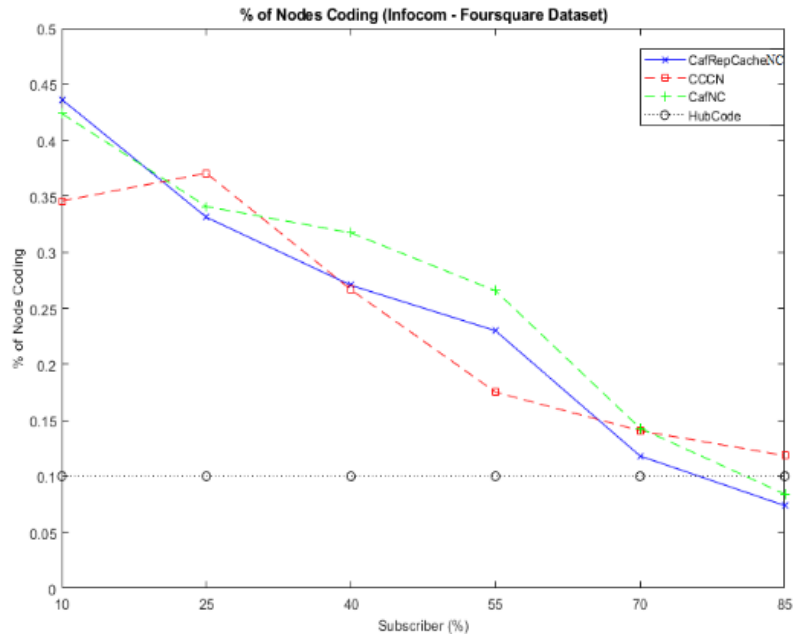


Figure 5.2 % of Node Coding vs. Number of Subscribers.

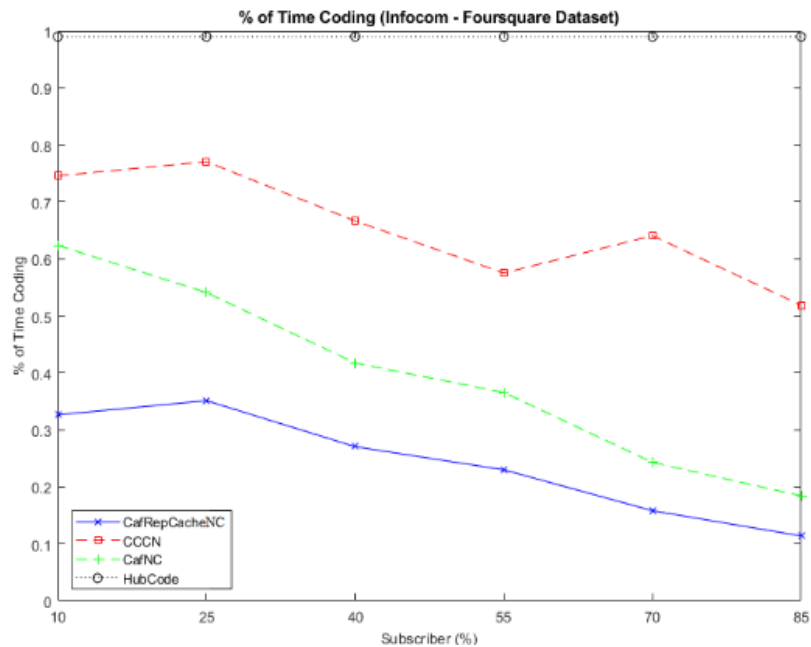


Figure 5.3 % of Time Coding vs. Number of Subscribers.

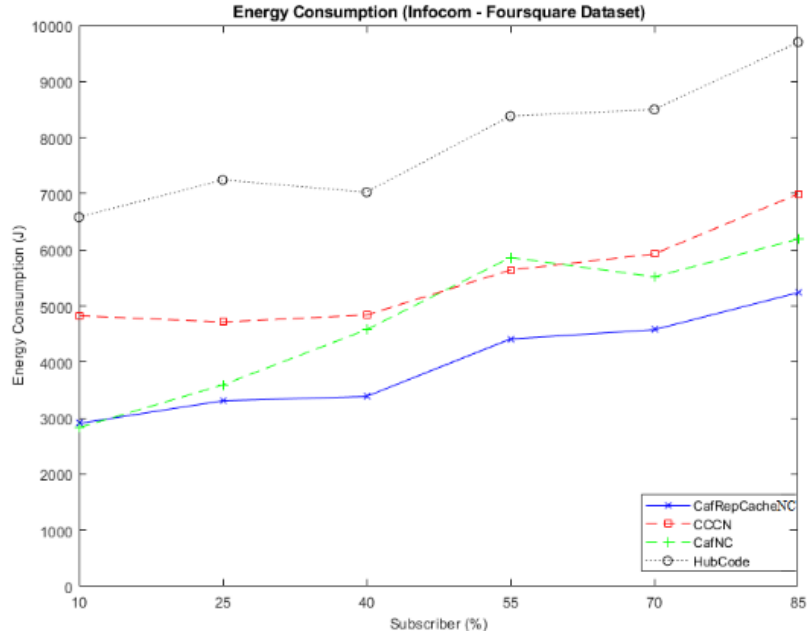


Figure 5.4 Energy Consumption (J) vs. Number of Subscribers

Fig 5.4 shows that CafRepCache consumed as half average energy (5095J) as HubCode (9824J) and less than 15-30% compared to CCCN (7012J) and CafNC (3239J) as shown in Fig. 5.1. Energy-aware network coding rate in CafRepCache captures and predicts energy level of neighboring nodes based on remaining energy capacity and total energy cost of transmitting and exchanging signals. It avoids forwarding, coding and caching packets to nodes or regions with high availability but has low battery capacity or about to be depleted. A dynamic threshold is specified for different types of nodes (e.g. more important nodes have to be more strictly protected from being battery drainage). In addition to this, CafRepCache leverages the spatial-temporal locality of content traffic-related users' mobility and resources to utilise efficiently coding opportunities. This allows CafRepCache to choose the most suitable set of contents to be forwarded, cached and network coded in the most appropriate relays in order to improve energy efficiency and reduce resource consumption.

5.2 Early Deployment of MoDiToNeS Platform in Heterogeneous Mobile Opportunistic Networks

MoDiToNeS [130] is a low-cost open-source platform which is highly suitable for fast prototyping and testing of fully-localised multi-layer real-time complex communication protocols in real-world scenarios. This section describes our early deployment of MoDiToNeS [130] in Raspberry Pis with a range of

protocols (CafRepCache, Epidemic, etc.) in two different real-world scenarios of agricultural monitoring and smart city applications. We show that CafRepCache can be successfully deployed and running in MoDiToNeS [130]. However, due to various real-world difficulties of the deployment such as limited number of nodes, short battery life, lack of dynamic content traffic generation, limited patterns of mobility, we have not investigated dynamic caching, dynamic publish-subscribe patterns, thus we cannot show rich graphs and analysis of CafRepCache caching performance as we did in chapter 4. Nevertheless, we aim to describe our multiple experiment set up scenarios for our early deployment and testing of MoDiToNeS platform [130] running CafRepCache to enable multihop communications with autonomous vehicles and drones.

In our experiments, MoDiToNeS [130] platform integrating with CafRepCache is deployed on a low-cost single-board computer Raspberry Pi B (ARM710, 700MHz, 512MB RAM, 8GB SD Card) which provides good processing power and flexible storage. It is integrated with a range of wired and wireless sensors using GPIO, I2C, RF modules, 802.11 Wifi, Bluetooth and USB. The sensors include directly attached camera sensor, MEMS Sensor Board with Low power 3D magnetometer, 3-Axis digital accelerometer, Humidity sensor, temperature / high precision pressure sensor, and a remote ANT+ protocols compatible sensor. The MoDiToNeS [130] sensor nodes are able to capture local and remote sensors' readings as well its real-time local resources utilisation (including CPU load, memory, disk usage, I/O) at configured time intervals. Nodes are equipped with a wireless interface to provide ad hoc network connectivity. Each node is able to store the sensor measurements locally and make them available for localised and remote queries as well as also generate simple format messages with sensor measurements which are sent to the subscriber nodes.

In the first deployment (Fig. 5.5), we focus on the agricultural context in the UK where the production of potatoes or livestock such as beef has always been a major constituent of agriculture [89]. There has always been a need for farmers to be able to observe their field crops or animals as often as possible so that they are informed quickly about deep-rooted problems in the fields which would cause tremendous damages. For example, frozen ground and frost event happens

when the temperature is so low that crops cannot recover their tissues or their internal structure. In the first set of experiments, we focus on supporting reliable communications between wireless sensor networks with autonomous drones and vehicle for agricultural monitoring applications. Fig 5.5 shows we deploy five nodes in an agricultural field near Nottingham city, UK (Latitude 52.948059, Longitude -1.202398) comprising of two drones, one vehicle and two static sensing nodes (publisher and subscriber). The two static sensing nodes are deployed in two different sides of the field: one is equipped with a range of field sensors acting as a publisher which gathers, processes and communicates various sensor readings about the physical environment such as air temperature, humidity, wind and soil temperature. The publishing node either regularly or on-demand sends the field's information to another node which acts as a subscriber and processor which can identify any variations in the soil. As the two nodes are deployed on different sides of the field, they are out of reach of each other and can be connected via multi-hop communications enabled by two drones and one vehicle. The two drones (Parrot Quadcopter AR Drone 2.0) and one vehicle acting relay nodes to gather, process and forward wireless sensor data from the field using multi-hop disconnection-tolerant communications. Drones and car's mobility is dynamically changing. The two drones and the car also capture regular camera data of the field and other sensor readings about physical conditions which are then sent to the subscriber.





Figure 5.5 Deployment of MoDiToNeS platform running CafRepCache in Nottingham agricultural field with autonomous drones and vehicle.

Fig. 5.6 shows the PNP4Nagios interfaces of vehicular node and drone nodes seeing each other with their cameras, exchanging and visualising other scalar sensor measurements.

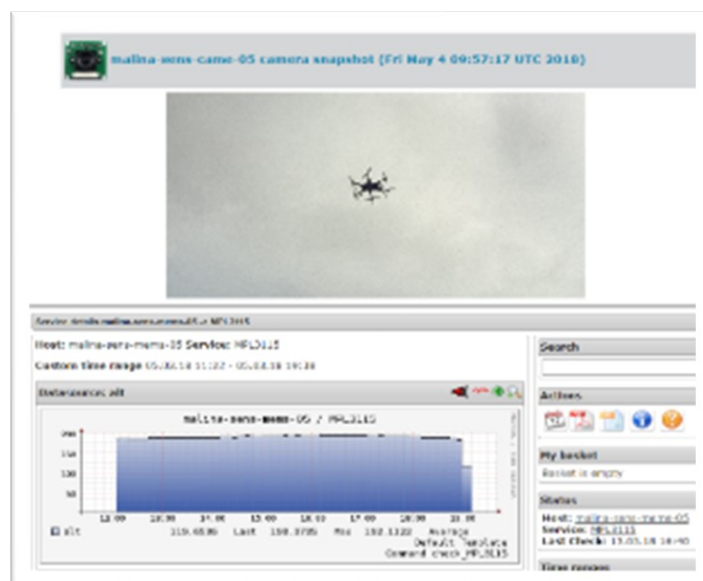


Figure 5.6a User interfaces on the vehicle taking images of the drone while gathering sensor readings.

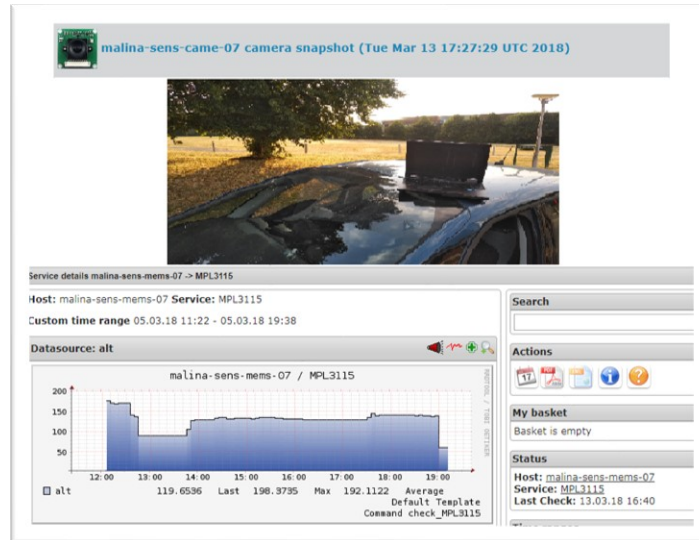


Figure 5.6b User interfaces on the vehicles taking images of each other while gathering sensor readings.

In the second deployment, we focus on the deployment of MoDiToNeS [130] platform integrating with CafRepCache in the urban context on autonomous vehicles (i.e. cars and drones) to support delay-tolerant vehicular networks when the connectivity is intermittent in Valencia, Spain. In this context, communications between vehicles and the infrastructures may play an important role in distributing various relevant information, for example, real-time traffic or multimedia information. In our deployment in Valencia, we have a limited number of vehicles and drones with short battery life and lack of dynamic content traffic. We considered both suburb area of Valencia with a lower populated area of the city of Valencia, Spain (Latitude 39.483920, Longitude -0.333793) with few signal blockages (Fig 5.7a), and a highly urban area which was crowded and dense area of the city (Latitude 39.473695, Longitude -0.332307) with residential buildings presenting obstructions and network interferences (Fig. 5.7b). We tested different wireless network frequency bands (2.4 GHz and 5GHz) on both single and multi-hop communications. The single-hop communication set up consists of two nodes: a static RSU/hovering drone acting as a content publisher and a moving vehicle acting as a content subscriber. The publisher continuously sends information such as sensors' readings, street images, traffic videos to the content subscriber who monitors road conditions.

Fig. 5.7 shows that single-hop wireless communications are significantly affected by the physical obstructions and network interferences, especially in the dense area of the city (Fig 5.7b). As expected, the content delivery ratio is below 50% at the distance of 120m in suburb area and 100m in highly urban area.

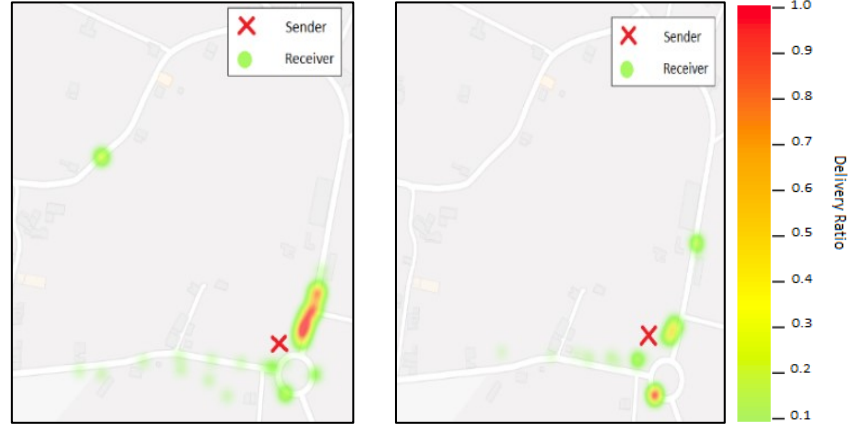


Figure 5.7a Single hop communication in suburb area of Valencia for 2.4 Ghz and 5 Ghz respectively.



Figure 5.7b Single hop communication in dense city centre area of Valencia for 2.4 Ghz and 5 Ghz respectively.

We also show the deployment with autonomous mobile flying vehicles (i.e. drones) to support delay-tolerant vehicular communications when the connectivity is limited. We then deploy MoDiToNeS [130] on two vehicular nodes and one drone in Valencia, Spain to evaluate the performance of multi-hop disconnection-tolerant networks for distributing in real time various relevant information such as traffic and multimedia data. In particular, a hovering drone acts as content publisher, one moving car as a content subscriber, an additional moving car as a relay node between the content subscriber and publisher.

Fig. 5.8 shows average edge cloud to edge cloud (E2E) latency of content query and dissemination in multi-hop delay-tolerant communications via one or two drones and one vehicle intermediaries in the face of increasing content size ranging from 128KB (i.e. scalar data) to 4MB (i.e. multimedia data) in Nottingham agricultural context. We observe that having two intermediary drones lowers latency compared to having only one intermediary drone. When the content size is small (128KB – i.e. temperature readings, etc), the average latency of sending and receiving content is 51.09s for two-node intermediaries and 29.12s for three-node intermediaries. When the content size increases (4MB – i.e. images), the average latency of sending and receiving content with two-node intermediaries is 66.1s while that with three-node intermediaries is 45.2s.

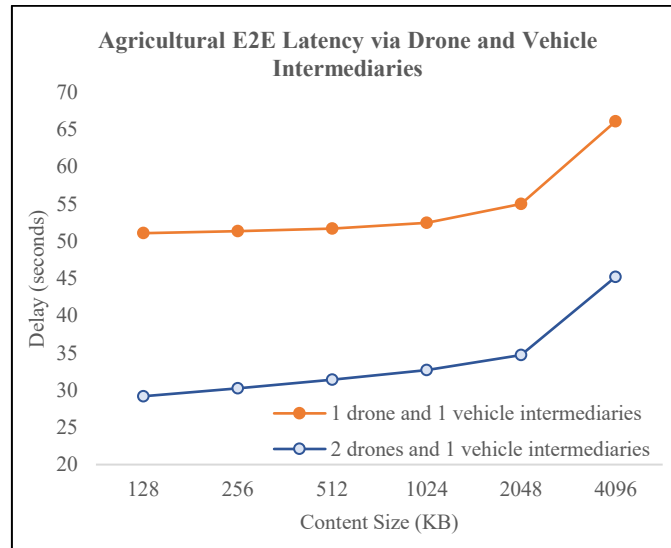


Figure 5.8 Edge cloud to edge cloud (E2E) latency in multi-hop communications via drone and vehicle intermediaries for agricultural applications.

Fig. 5.9 shows the average latency of multi-hop drone and vehicle communications using two frequency bands (2.4 GHz and 5 GHz) in the suburbs and city centre of Valencia while varying the content sizes from 128KB (e.g. sensor readings) to 4 MB (e.g. images). We show that the average edge cloud to edge cloud (E2E) latency in suburb area of Valencia is lower than in the Valencia city centre area: approximately 56.25s in the suburbs compared to 164.11s for the city centre for 2.4Ghz; and similarly 68.21s in the suburbs versus 122.87s in the city centre for 5Ghz respectively. Highly dense city area has higher isolation time between vehicles and the hovering drone, especially when using 2.4 GHz

frequency band in the presence of physical obstruction and network interferences that result in very short contact time between vehicles and the hovering drone. When the content size increases from 128 KB to 4MB, the average edge cloud to edge cloud latency in suburb area increases from 62.23s to 196.06s while that in urban area ranges from 143.49s to 374.96s.

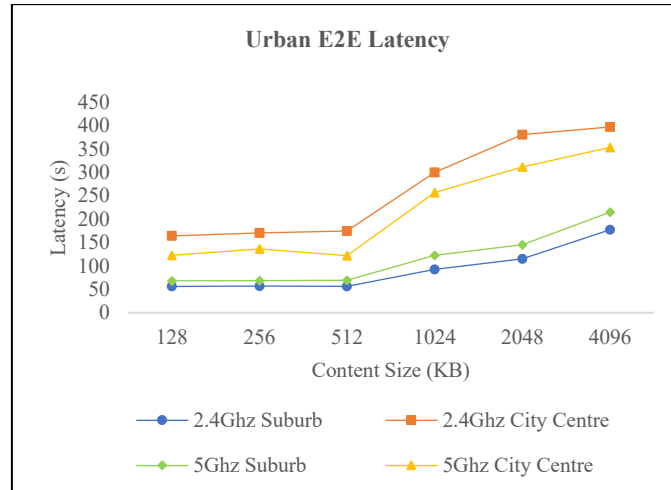


Figure 5.9 Edge cloud to edge cloud (E2E) latency in multi-hop drone and vehicle communications in urban scenarios.

We measure latency for different altitudes of hovering drones (5m, 20m and 40m) while cars' mobility is dynamically changing. Our drone-to-vehicle (D2V) communication band uses 5GHz rather than as 2.4 GHz frequency band as the latter is prone to providing poor performance due to interferences with drone controller. For safety reasons, we only conduct drone experiments in the suburb area of Valencia.

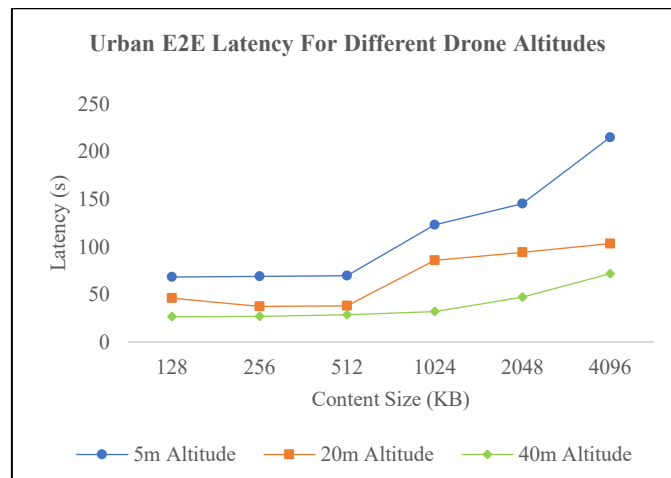


Figure 5.10 Edge cloud to edge cloud (E2E) latency in multi-hop communications for different drone's altitude in suburb scenario.

Fig. 5.10 shows that the multi-hop delay-tolerant V2V-D2V communications with autonomous drones keeps the average latency below 49.1s regarding the increasing content size. We also observe that the higher the drone is, the longer network coverage range and the lesser the probability of finding obstacles, thus increase the network connectivity and contact durations. When sending normal sensor reading packets (128KB), using 5Ghz frequency band in suburb area with the autonomous drone at 40 metres altitude achieve 26.52s delay, followed by that at 20 metres altitude (48.02s) and 5 metres altitude (68.21s). When we increase the content size to 4MB, the performance of multi-hop delay-tolerant V2V-D2V communications with the autonomous drone at 40 metres altitude keeps below 71.74s latency while that at 20 metres altitude is 103.36s respectively. We argue that the impact of increasing the drone's altitude more than 40m is negligible in our deployment in multi-hop disconnection-tolerant V2V-D2V communications with autonomous drones.

In both real-world scenarios of agriculture and smart city, we have limited number of nodes (five nodes in Nottingham agricultural scenario and three nodes in Valencia urban scenario) with very short battery life, lack of dynamic content traffic generation and limited patterns of mobility due to safety reason. Thus, we have not investigated the caching performance of CafRepCache running in MoDiToNeS [130] platform. All nodes are in contact via IBR-DTN after some delay and we have sufficient buffers compared to the content sending rates, thus we observe that MoDiToNeS [130] nodes achieve 100% success ratio and 0% packet loss for delivering contents in our early experiments. In the next section, we envisage and discuss more large-scale real-world deployment scenarios of CafRepCache, its impacts in different application domains and examine the issues which may arise when deploying CafRepCache within these context of large-scale dynamic environments [132].

5.3 CafRepCache Large-Scale Real-World Application Domains

Large-scale real-world applications of CafRepCache are scenarios which have complex temporal network topologies and where the geo-temporal publish-subscribe or supply-demand needs to be better understanding and predicted such as smart energy [133], smart agriculture [89, 90], urban emergency [134], intelligent transport system [30], and smart manufacturing [78].

We discuss real-world smart city scenario such as EU FP7 ALMANAC [135] where we envisage CafRepCache caching system takes place either at the edges such as on femtocell, access points, mobile users, vehicles, etc. or at the dedicated cache servers to provide heterogeneous content services such as video streaming, file downloading, used by dynamic groups of mobile users who request contents in real time. In this context, multiple CafRepCache nodes are located in different areas providing computation and caching capacity (Fig 5.11). For example, a group of users in a city centre area acting as subscribers send their interests of information about nearby parking space, local coffee shops or interests of bus schedule to CafRepCache nodes. CafRepCache nodes can communicate with other neighbouring nodes and can collaboratively retrieve the requested contents from each other in a localized and more responsive manner and with lower delays compared to the infrastructure.

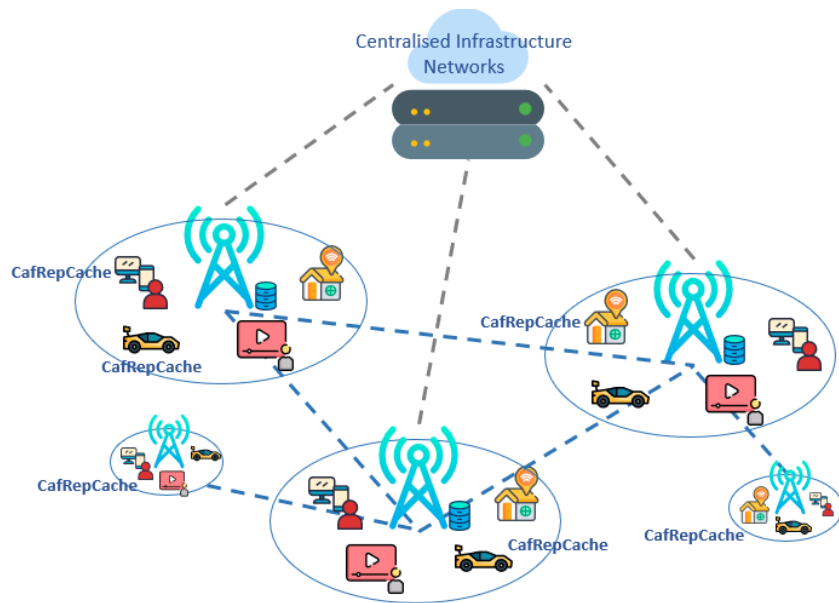


Figure 5.11 Example large-scale real-world scenario of CafRepCache deployment.

Another large-scale real-world application of CafRepCache is developing adaptive and personalized public transportation services within the smart city such as proposed in EU SmartSantander project [136]. To support such adaptive services, in addition to infrastructure-assisted network communications, we focus more on participatory sensing approach which utilise data from smartphones, vehicles, roadside units, etc. The infrastructure-based approach is usually comprehensive and accurate but unlike participatory mobile sensing, it

cannot observe an individual's experience and data query. For example, it cannot reveal how many people are waiting at a train platform which may go beyond the arriving train's maximum loading capacity, and queuing commuters have to wait for the subsequent train while watching YouTube or streaming live videos. We envisage CafRepCache combines these two data streams (infrastructure and mobile) to provide deeper insights into urban transportation as well as people behaviour and interests. CafRepCache utilises its spatial-temporal analytics on transportation data to detect abnormal transportation events (e.g. high interest demand at a specific taxi stand) while proactively bringing the predicted highly popular contents for users' future query (e.g. commuting-related interests) with high accuracy, low latency and low cost.

We discuss real-world smart agriculture scenarios such as the EU project RHEA [137] which focuses on the design, development, and testing of new generation of automatic and robotic systems such as robot fleets for highly effective agriculture and forestry management. The fleet of vehicles, robots and drones is not only a collection of sensors that capture field data to users and upload filtered sensor data to the communication infrastructure but also an ad hoc network of nodes that collaboratively exchange their sensor inputs with each other, observe, make decisions on crops in order to reduce the use of pesticides. We envisage CafRepCache can be deployed in these robot fleets to provides a collaborative framework between people, robots and infrastructure as well as enable more responsive and more accurate content sharing, discovery and retrieval between fleet members through multidimensional hybrid predictive analytics such as spatial-temporal analytics, social analytics, data of interest storage and query.

In all three examples of large-scale real-world scenarios, the requested contents are highly skewed such that a few number of contents are requested more frequently by the users/subscribers. This is because users may share some common interests such as bus schedule, traffic in smart transport, restaurants, parking spaces in smart city, fire incidents in urban emergency, crop conditions in smart agriculture etc, thus the content request will show a certain level of locality in each area as well as certain similarities over time. While CafRepCache nodes have limited caching storage and thus can only cache a

certain amount of contents, retrieving contents from the CafRepCache nodes and its neighbours has significant lower delay compared to retrieving contents from the server. When a request arrives, the local CafRepCache returns the cached content immediately to users if found in its local cache. If a local CafRepCache node does not cache a requested content, it then attempts to retrieve the content from its neighbour nodes which cause extra latency but is still faster and cost-effective than from the server. When the requested contents cannot be found from either local or neighbour nodes' cache storage, local CafRepCache node fetches the requested contents from the server as a lowest-priority back-up solution.

In order to deploy CafRepCache in either mobile devices or existing infrastructure, software-defined networking (SDN) or network function virtualization (NFV) [138] can provide a powerful tool for experimenting and deploying CafRepCache multilayer architecture and can greatly facilitate its functionality and management. SDN and NFV depend heavily on virtualization to enable network design and infrastructure to be abstracted in software and then implemented by underlying software across hardware platforms and devices. While NFV provides basic networking functions, SDN controls and orchestrates them for specific uses. SDN allows configuration and programmable network control. SDN and NFV differ in how they separate functions and abstract resources. SDN abstracts physical networking resources – switches, routers and so on – and moves decision making to a virtual network control plane. In this approach, the control plane decides where to send traffic, while the hardware continues to direct and handle the traffic. NFV aims to virtualize all physical network resources beneath a hypervisor, which allows the network to grow without the addition of more devices. While both SDN and NFV make networking architectures more flexible and dynamic, they perform different roles in defining those architectures and the infrastructure they support [139-142]. An existing implementation of the SDN paradigm is OpenFlow (OF) [143], which is a standard protocol enabling the communication between the SDN controller and the network devices. OpenState [144] is a recent extension of OF that enables a stateful SDN approach, leveraging a standard OF switching architecture.

We argue that different scenario has different degree of spatial-temporal locality in content interests and users' mobility which affects the performance of CafRepCache framework. In urban area, CafRepCache utilities its multidimensional predictive analytics and complex social temporal graph metrics to make caching decisions in a predictive manner and congruent with the underlying network mobility, connectivity and content interest. In rural area or in the scenarios of very dynamic random mobility and connectivity between nodes, our solution may be less effective compared to that in a more dense area. However, CafRepCache still manages to benefit from its cache redundancy mechanism that select highly suitable locations for caching and replication when needed as adaptive replication and caching are both necessary to address multi-user data communications in dynamic fragmented and sparse topologies. Nevertheless, we show that CafRepCache's predictive analytics achieve highly successful performance and it always opens up for the integration with other intelligent edge analytics for future large-scale heterogeneous scenarios.

5.4 Summary

This chapter begins with discussions on the collaborative joint adaptive network coding with CafRepCache caching framework in mobile disconnection-tolerant networks which is promising to deliver enhanced QoS such that to reduce data transmissions, energy consumption and support more efficiently dynamic workload. We then discuss the early deployment of MoDiToNeS platform prototyping CafRepCache in two different real-world scenarios of agricultural monitoring and smart city. We also explore the large-scale deployment challenges of CafRepCache framework and discuss CafRepCache's impacts in different application domains. In the next chapter, we summarise our thesis contributions and discuss future work.

Chapter 6

6. Conclusion

This chapter presents the main original contributions of this thesis and proposes future research directions of content discovery and retrieval services in heterogeneous mobile opportunistic networks.

6.1 Summary of Contributions

The thesis was concerned with proposing a multilayer fully-distributed predictive caching framework that combines real-time adaptive multipath content and interest forwarding, with adaptive cognitive collaborative caching.

- The work contributes to the philosophy of supporting content discovery and retrieval services in heterogeneous mobile opportunistic networks by recognising the importance of predictively adapting to the dynamics and complexity of multiple network layers ranging from network underlying infrastructure to its heterogeneous applications. By this way, the content services are more robust, scalable and adaptive to very different time-varying dynamically changing: network topologies, mobility patterns, resources and workloads of content publishers and subscribers in heterogeneous mobile opportunistic networks. This was published in 3rd Workshop on Experiences with the Design and Implementation of Smart Objects, MobiCom 2017.
- The work proposes CafRepCache, a fully-distributed collaborative predictive adaptive caching framework which consists of multidimensional multilayer predictive analytics and heuristics allowing CafRepCache to cache the most suitable set of contents in the most suitable set of nodes in dynamically changing mobility, content traffic patterns and resources. Each predictive analytic and heuristic is able to produce complex trend analytics of different network layer and when combined together, our predictive analytics and heuristics help to capture and predict the interdependences between multilayers in order to deal with multiple complex inherent trade-offs between maximising content delivery while minimising the end-to-end latency, resource consumption and enabling congestion avoidance. This was published in IEEE Access Journal, v. 6, p. 32188-32206, 2018 and in 3rd

Workshop on Experiences with the Design and Implementation of Smart Objects, MobiCom 2017.

- The work proposes temporal content request predictive analytics and heuristics that combine request frequency, recency and betweenness to chooses the best suitable contents to cache and when to cache. They allow CafRepCache to predict in real time the trend of content request patterns over time, more responsive to the dynamic variation of content popularity, avoids losing valuable contents by reducing the caches for one-timers contents and fake news. This work was published in IEEE Access Journal, v. 6, p. 32188-32206, 2018, in 3rd Workshop on Experiences with the Design and Implementation of Smart Objects, MobiCom 2017, and in 9th International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS) 2019.
- The work proposes spatial content request predictive analytics and heuristics that combine clustering coefficient, similarity, closeness, tie strengths between different subscribers for the same content in order to choose the best caching point for a content and the groups of subscribers. They allow CafRepCache to choose what content to cache and where to cache a content better in the dynamic changing content workload as well as dynamic mobility of subscribers and publishers. This work was published in IEEE Access Journal, v. 6, p. 32188-32206, 2018, in 3rd Workshop on Experiences with the Design and Implementation of Smart Objects, MobiCom 2017, and in 9th International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS) 2019.
- When combined together, spatial-temporal predictive analytics and heuristics help to extends the idea of behavioural locality to exploit similarities between the content interests and users connectivity as well as to perform a precise spatial-temporal behaviour characterisation of both individual and localised regions of subscribers in order to cache the most suitable set of contents in the most suitable set of nodes. It allows improved congruency with both the underlying network and users demands, more accurately capture the temporal and spatial locality of mobility and content requests as well as mobility-content traffic patterns interplay. This work was

published in IEEE Access Journal, v. 6, p. 32188-32206, 2018, in 3rd Workshop on Experiences with the Design and Implementation of Smart Objects, MobiCom 2017, and in 9th International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS) 2019.

- The work presents a content replacement policy and offloading mechanism that avoid losing low-popular contents by offloading them to nodes with higher available resources while still keeping them as close as possible to the subscribers. This allows CafRepCache to maintain high cache hit ratio, keep the eviction rate low when cache space is relatively small to balance the trade-off between current observed content popularity versus long terms interest in order to avoid caching too quickly potentially one-timer contents or fake news and losing the long-term useful contents. This work was published in IEEE Access Journal, v. 6, p. 32188-32206, 2018, and in 9th International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS) 2019.
- The work use principles of dynamic predictive relative utility to propose a collaborative algorithm which incentivises and allows individual nodes to achieve greater utility than if they do not collaborate. Where the global knowledge of networks is not assumed, this describes the unification of CafRepCache nodes to form a self-organised dynamic geo-temporal regional perspective view of the networks in order to obtain better predictive forwarding and caching decisions which result in less selfish strategy and more accurate, more responsive decisions for the overall networks. This was published in IEEE Access Journal, v. 6, p. 32188-32206, 2018 and in 3rd Workshop on Experiences with the Design and Implementation of Smart Objects, MobiCom 2017.
- The work introduces a new reputation-based popularity weighing mechanism to adaptively weight different exchanged values gathered from different nodes in the ego network based on node's reputation and connectivity, in order to improve the accuracy of collaborative forwarding and caching decisions in fully-distributed manner. This was published in 14th International Wireless Communications & Mobile Computing Conference (IWCMC) 2017.

- We have evaluated the contributions of this thesis by emulating real network conditions, incorporating device connectivity traces from a number of possible deployment scenarios with different heterogeneous content publish-subscribe patterns, both are real world and pseudo real-world models and experimentally evaluating a range of criteria.
- CafRepCache outperforms a number of state-of-the-art and benchmarking caching algorithms, independent of the network connectivity/mobility and content traffic patterns. CafRepCache achieves significantly higher cache hit ratio as well as end-to-end content retrieval success ratio while keeping lower delays, packet loss rates, cache footprint, cache eviction rate and resource consumption in the dynamic changing content workload, dynamic mobility patterns and dynamic resource availability.
- We successfully bridge theoretical protocols and simulations to the real world by deploying CafRepCache in different real-world contexts of mobile heterogeneous sensing and query in mobile opportunistic environments: 1) mobile personal cloud, 2) large-scale emergency applications, 3) networked drones and vehicles for agricultural monitoring applications and 4) vehicles and drone communications for smart city applications in urban environments. We discuss the performance characteristics of different mobile disconnection prone communication protocols and dynamic architectures across a range of criteria and show that CafRepCache's performance is generic enough to be deployed in any real-world contexts and applications. This work was published in 12th & 13th International Wireless Communications & Mobile Computing Conference (IWCMC) and in 15th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) 2019.
- We envisage that our philosophy of dynamic complex geo-temporal publish-subscribe or supply-demand will be integrated with the future mobile apps and services to enable more robust network support and allow more reliable rich mobile multimedia services in a variety of application areas such as smart energy [133], smart cloud, edge and fog computing [5, 11], smart transports and infrastructures [5, 6] where users may suffer from limited network coverage, traffic demand or network congestion. This work was

published in 12th & 13th International Wireless Communications & Mobile Computing Conference (IWCMC) and in 15th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) 2019.

6.2 Future Work

The results from the experiments presented in this thesis are encouraging, validating the effectiveness of CafRepCache in terms of improving the reliability and scalability of content services in mobile opportunistic networks. This section is concerned with the future directions that have not been fully addressed.

6.2.1 Artificial Intelligence and Machine Learning for Offline Analysis

Critical applications such as resource allocation [145], network monitoring[24, 146] or dynamic network management [147] require real-time mechanisms for online analysis as well as efficient techniques for offline deep analysis of massive historical data [147]. While CafRepCache predictive analytics and heuristics allow predictive adaptive response to changing local conditions in real time and at low cost, complex techniques and algorithms to analyse a huge amount of historical data offline is needed in order to gain deeper insight of network environments which open ups opportunities to innovate, redefine or improve the real-time predictive analytics and heuristics for better the accuracy and efficiency.

Artificial Intelligence (AI) and Machine Learning (ML) approaches have been recently re-gaining attraction in the networking domain. AI and ML represent a different paradigm to tackle these challenges due to its data-driven nature to automatically learn the complexity of the communications and networking environment and to dynamically adjust protocols without human interactions. However, applying AI/ML to networked systems remains inadequately understood and investigated [147]. Scalable approaches must be conceived to make them generally applicable in networking and communication systems.

6.2.2 Fairness and Incentiveness

In the proposed caching framework, we do not focus on building a protocol that forces nodes to collaborate, or provides protection against malicious behaviour,

but rather to design underlying algorithms that can adaptively share distributed cache space across trusted collaborators [26, 148].

In the future work, we plan to propose a novel incentive mechanism that incentivises all nodes in the network to collaborate, avoid the selfish and malicious behaviours of users in real world. However, we believe that the CafRepCache design philosophy will remain applicable and effective in a certain degree of fairness due to our relative utility model as well as to protect CafRepCache nodes against malicious or selfish behaviour due to our reputation and credit-based feedback mechanism.

6.2.3 Dynamic Scalability with Software-Defined Networking (SDN) and Network Functions Virtualization (NFV)

In the future work, we suggest possible solution to utilise Software-defined networking (SDN) [139, 140] and Network functions virtualization (NFV) [141, 142] to deploy scalable CafRepCache architecture more easily and more quickly, implement core features in distributed manner without centralised controller, involving more complex problems with mobility and disconnection. SDN/NFV can be the key to scalable and manageable deployment of CafRepCache as it allows network abstraction to decouple network forwarding functions from network control and other network functions from the hardware.

Distributed network control plan in SDN [139, 140] share aggregated local and network-wide information with the neighbour controllers. SDN controller can perform computation-intensive tasks to solve different optimization problem [139, 140]. By leveraging the regional view of network cache state, CafRepCache managed by SDN controller can easily obtain and maintain a consistent view of the underlying network cache state at the control layer, which can be used to determine where and when to cache content copies and compute optimal forwarding routes.

6.2.4 Security and Privacy

In the future work, we explore security and privacy concerns of content services in large-scale heterogeneous mobile opportunistic networks both in depth and breadth.

In the security section, there are multiple vulnerabilities unique to content-centric services, including cache pollution, content poisoning, and naming attacks [17, 76, 149-151]. Despite many similarities between a classical and the attack in content-centric services, the latter is novel and are still not well addressed [17, 76]. For example, cache pollution attack targets a caching point's content locality with the intention of altering its set of cached content resulting in an increase in the frequency of content retransmission, and reduced network throughput [17, 76].

In the privacy section, client privacy, content privacy, cache privacy, and name and signature privacy [17, 76, 149-151] will need to be studied. Due to the nature of pervasive caching, contents can be replicated throughout the network. Though this moves content close to the edge and helps reduce network load and content retrieval latency, it comes at a cost—publishers lose control over these cached copies and cannot arbitrate access. Thus, there is a need for efficient access control which allows the reuse of cached content and prevents unauthorized accesses.

Bibliography

1. Index, C.V.N., *Global mobile data traffic forecast update, 2016–2021 white paper*. Cisco: San Jose, CA, USA, 2017.
2. Index, C.V.N., *global mobile data traffic forecast update, 2017–2022*. Cisco White paper, 2019.
3. Hayes, B., *Cloud computing*. Communications of the ACM, 2008. **51**(7): p. 9-11.
4. Dinh, H.T., et al., *A survey of mobile cloud computing: architecture, applications, and approaches*. Wireless communications and mobile computing, 2013. **13**(18): p. 1587-1611.
5. Whaiduzzaman, M., et al., *A survey on vehicular cloud computing*. Journal of Network and Computer applications, 2014. **40**: p. 325-344.
6. Gerla, M., et al. *Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds*. in *2014 IEEE world forum on internet of things (WF-IoT)*. 2014. IEEE.
7. Radenkovic, M., *Cognitive privacy for personal clouds*. Mobile Information Systems, 2016. **2016**.
8. Shi, W. and S. Dustdar, *The promise of edge computing*. Computer, 2016. **49**(5): p. 78-81.
9. Radenkovic, M. and V.S.H. Huynh. *Collaborative cognitive content dissemination and query in heterogeneous mobile opportunistic networks*. in *Proceedings of the 3rd Workshop on Experiences with the Design and Implementation of Smart Objects*. 2017. ACM.
10. Radenkovic, M., V.S.H. Huynh, and P. Manzoni, *Adaptive real-time predictive collaborative content discovery and retrieval in mobile disconnection prone networks*. IEEE Access, 2018. **6**: p. 32188-32206.
11. Huynh, V.S.H. and M. Radenkovic. *Interdependent Multi-layer Spatial Temporal-based Caching in Heterogeneous Mobile Edge and Fog Networks*. in *Proceedings of the 9th International Conference on Pervasive and Embedded Computing and Communication Systems PECCS*. 2019.
12. Shi, W., et al., *Edge computing: Vision and challenges*. IEEE Internet of Things Journal, 2016. **3**(5): p. 637-646.
13. Dastjerdi, A.V. and R. Buyya, *Fog computing: Helping the Internet of Things realize its potential*. Computer, 2016. **49**(8): p. 112-116.
14. Liu, Z., et al., *DATS: Dispersive Stable Task Scheduling in Heterogeneous Fog Networks*. IEEE Internet of Things Journal, 2018.
15. Bari, M.F., et al., *A survey of naming and routing in information-centric networks*. IEEE Communications Magazine, 2012. **50**(12): p. 44-53.
16. Ioannou, A. and S. Weber, *A survey of caching policies and forwarding mechanisms in information-centric networking*. IEEE Communications Surveys & Tutorials, 2016. **18**(4): p. 2847-2886.
17. Tourani, R., et al., *Security, privacy, and access control in information-centric networking: A survey*. IEEE communications surveys & tutorials, 2017. **20**(1): p. 566-600.
18. Zhang, G., Y. Li, and T. Lin, *Caching in information centric networking: A survey*. Computer Networks, 2013. **57**(16): p. 3128-3141.
19. Frey, D., M. Goessens, and A.-M. Kermarrec. *Behave: Behavioral cache for web content*. in *IFIP International Conference on Distributed Applications and Interoperable Systems*. 2014. Springer.
20. Dernbach, S., et al. *Cache content-selection policies for streaming video services*. in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. 2016. IEEE.

21. Hachem, J., N. Karamchandani, and S. Diggavi. *Content caching and delivery over heterogeneous wireless networks*. in *2015 IEEE conference on computer communications (INFOCOM)*. 2015. IEEE.
22. Flores, H., et al., *Social-aware hybrid mobile offloading*. *Pervasive and Mobile Computing*, 2017. **36**: p. 25-43.
23. Lima, E., et al. *Impacts of human mobility in mobile data offloading*. in *ACM CHANTS*. 2018.
24. Alipour, B., et al., *Analyzing Mobility-Traffic Correlations in Large WLAN Traces: Flutes vs. Cellos*. arXiv preprint arXiv:1801.02705, 2018.
25. Dabirmoghaddam, A., M.M. Barijough, and J. Garcia-Luna-Aceves. *Understanding optimal caching and opportunistic caching at the edge of information-centric networks*. in *Proceedings of the 1st ACM conference on information-centric networking*. 2014. ACM.
26. Wang, L., et al., *Milking the cache cow with fairness in mind*. *IEEE/ACM Transactions on Networking (TON)*, 2017. **25**(5): p. 2686-2700.
27. Wang, L., et al., *Understanding scoped-flooding for content discovery and caching in content networks*. *IEEE Journal on Selected Areas in Communications*, 2018. **36**(8): p. 1887-1900.
28. Saha, S., A. Lukyanenko, and A. Ylä-Jääski. *Cooperative caching through routing control in information-centric networks*. in *2013 Proceedings IEEE INFOCOM*. 2013. IEEE.
29. Radenkovic, M., *Disconnection Aware Smart Festivals*. EPSRC RR0903 Impact Exploration Grant.
30. Loscri, V., et al. *A social internet of vehicles sharing SloT relationships*. 2019.
31. D'Silva, K., et al., *Predicting the temporal activity patterns of new venues*. *EPJ Data Science*, 2018. **7**(1): p. 13.
32. Hristova, D., et al., *A multilayer approach to multiplexity and link prediction in online geo-social networks*. *EPJ Data Science*, 2016. **5**(1): p. 24.
33. Mahanti, A., D. Eager, and C. Williamson, *Temporal locality and its impact on Web proxy cache performance*. *Performance Evaluation*, 2000. **42**(2-3): p. 187-203.
34. Breslau, L., et al. *Web caching and Zipf-like distributions: Evidence and implications*. in *Ieee Infocom*. 1999. INSTITUTE OF ELECTRICAL ENGINEERS INC (IEEE).
35. Chatzieleftheriou, L.E., M. Karaliopoulos, and I. Koutsopoulos. *Caching-aware recommendations: Nudging user preferences towards better caching performance*. in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. 2017. IEEE.
36. Vigneri, L., T. Spyropoulos, and C. Barakat. *Storage on wheels: Offloading popular contents through a vehicular cloud*. in *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2016. IEEE.
37. Vigneri, L., T. Spyropoulos, and C. Barakat. *Streaming content from a vehicular cloud*. in *Proceedings of the Eleventh ACM Workshop on Challenged Networks*. 2016. ACM.
38. Jacobson, V., et al. *Networking named content*. in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. 2009. ACM.
39. Mahmood, A., et al., *Efficient caching through stateful SDN in named data networking*. *Transactions on Emerging Telecommunications Technologies*, 2018. **29**(1): p. e3271.
40. Carofiglio, G., L. Mekinda, and L. Muscariello, *Joint forwarding and caching with latency awareness in information-centric networking*. *Computer Networks*, 2016. **110**: p. 133-153.

41. Psaras, I., W.K. Chai, and G. Pavlou. *Probabilistic in-network caching for information-centric networks*. in *Proceedings of the second edition of the ICN workshop on Information-centric networking*. 2012. ACM.
42. Dehghan, M., et al. *A utility optimization approach to network cache design*. in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. 2016. IEEE.
43. Chun, B.-G., et al. *Selfish caching in distributed systems: a game-theoretic analysis*. in *Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*. 2004. ACM.
44. Le, T., Y. Lu, and M. Gerla. *Social caching and content retrieval in disruption tolerant networks (DTNs)*. in *2015 International Conference on Computing, Networking and Communications (ICNC)*. 2015. IEEE.
45. Bayhan, S., et al., *Search in digital pockets: Retrieving remote content in mobile opportunistic networks*. IEEE Wireless Communications, 2016. **23**(3): p. 10-18.
46. Mordacchini, M., et al., *Design and evaluation of a cognitive approach for disseminating semantic knowledge and content in opportunistic networks*. Computer Communications, 2016. **81**: p. 12-30.
47. Vahdat, A. and D. Becker, *Epidemic routing for partially connected ad hoc networks*. 2000.
48. Spyropoulos, T., K. Psounis, and C.S. Raghavendra. *Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility*. in *Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07)*. 2007. IEEE.
49. Spyropoulos, T., K. Psounis, and C.S. Raghavendra. *Spray and wait: an efficient routing scheme for intermittently connected mobile networks*. in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. 2005. ACM.
50. Daly, E.M. and M. Haahr. *Social network analysis for routing in disconnected delay-tolerant manets*. in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*. 2007. ACM.
51. Burgess, J., et al. *MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks*. in *Infocom*. 2006. Barcelona, Spain.
52. Hui, P., J. Crowcroft, and E. Yoneki, *Bubble rap: Social-based forwarding in delay-tolerant networks*. IEEE Transactions on Mobile Computing, 2011. **10**(11): p. 1576-1589.
53. Vendramin, A.C.K., et al., *A social-aware routing protocol for opportunistic networks*. Expert Systems with Applications, 2016. **54**: p. 351-363.
54. Barré, S., C. Paasch, and O. Bonaventure. *Multipath TCP: from theory to practice*. in *International Conference on Research in Networking*. 2011. Springer.
55. Radenkovic, M. and A. Grundy. *Congestion aware forwarding in delay tolerant and social opportunistic networks*. in *2011 Eighth International Conference on Wireless On-Demand Network Systems and Services*. 2011. IEEE.
56. Balasubramanian, A., B. Levine, and A. Venkataramani. *DTN routing as a resource allocation problem*. in *ACM SIGCOMM Computer Communication Review*. 2007. ACM.
57. Wang, L., et al. *C3po: Computation congestion control (proactive)*. in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*. 2016. ACM.
58. Costa, P., et al., *Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks*. IEEE Journal on selected areas in communications, 2008. **26**(5): p. 748-760.

59. Key, P., L. Massoulié, and D. Towsley. *Path selection and multipath congestion control*. in *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*. 2007. IEEE.
60. Radenkovic, M. and A. Grundy, *Efficient and adaptive congestion control for heterogeneous delay-tolerant networks*. *Ad Hoc Networks*, 2012. **10**(7): p. 1322-1345.
61. Rzađca, K., A. Datta, and S. Buchegger. *Replica placement in p2p storage: Complexity and game theoretic analyses*. in *2010 IEEE 30th International Conference on Distributed Computing Systems*. 2010. IEEE.
62. Ott, J., et al. *On the sensitivity of geo-based content sharing to location errors*. in *2017 13th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*. 2017. IEEE.
63. Basu, P., N. Khan, and T.D. Little. *A mobility based metric for clustering in mobile ad hoc networks*. in *Proceedings 21st International Conference on Distributed Computing Systems Workshops*. 2001. IEEE.
64. Radenkovic, M. and S. Zakhary. *Flexible and dynamic network coding for adaptive data transmission in dtms*. in *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*. 2012. IEEE.
65. Bilal, M. and S.-G. Kang, *Network-coding approach for information-centric networking*. *IEEE Systems Journal*, 2018(99): p. 1-10.
66. Leskovec, J. and J.J. McAuley. *Learning to discover social circles in ego networks*. in *Advances in neural information processing systems*. 2012.
67. Ghodsi, A., et al. *Information-centric networking: seeing the forest for the trees*. in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*. 2011. ACM.
68. Said, A., et al., *Proactive Caching at the Edge Leveraging Influential User Detection in Cellular D2D Networks*. *Future Internet*, 2018. **10**(10): p. 93.
69. Wang, T., et al., *Cooperative caching based on file popularity ranking in delay tolerant networks*. arXiv preprint arXiv:1409.7047, 2014.
70. Li, J., et al. *Popularity-driven coordinated caching in named data networking*. in *Proceedings of the eighth ACM/IEEE symposium on Architectures for networking and communications systems*. 2012. ACM.
71. Fricker, C., P. Robert, and J. Roberts. *A versatile and accurate approximation for LRU cache performance*. in *2012 24th International Teletraffic Congress (ITC 24)*. 2012. IEEE.
72. Wang, Y., M. Xu, and Z. Feng. *Hop-based probabilistic caching for information-centric networks*. in *2013 IEEE Global Communications Conference (GLOBECOM)*. 2013. IEEE.
73. Ioannou, A. and S. Weber, *Information-centric networking: A thorough evaluation of popularity-based probabilistic on-path caching*. Dept. Comput. Sci. Statist., Trinity College Dublin, Dublin, Ireland, Tech. Rep, 2014. **1**.
74. Saino, L., I. Psaras, and G. Pavlou. *Hash-routing schemes for information centric networking*. in *ICN*. 2013. ACM.
75. Li, Z. and G. Simon. *Time-shifted tv in content centric networks: The case for cooperative in-network caching*. in *2011 IEEE international conference on communications (ICC)*. 2011. IEEE.
76. Zakhary, S.R. and M. Radenkovic. *Reputation-based security protocol for MANETs in highly mobile disconnection-prone environments*. in *2010 Seventh International Conference on Wireless On-demand Network Systems and Services (WONS)*. 2010. IEEE.
77. Radenkovic, M., A. Benslimane, and D. McAuley, *Reputation aware obfuscation for mobile opportunistic networks*. *IEEE Transactions on Parallel and Distributed Systems*, 2015. **26**(1): p. 230-240.

78. Radenkovic, M., I. Kostadinov, and B. Wietrzyk. *Increasing communication reliability in manufacturing environments*. in *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*. 2015. IEEE.
79. Valerio, L., et al. *Autonomic cognitive-based data dissemination in opportunistic networks*. in *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. 2013. IEEE.
80. Vigneri, L., et al. *Per-chunk caching for video streaming from a vehicular cloud*. in *Proceedings of the 12th Workshop on Challenged Networks*. 2017. ACM.
81. Arianfar, S., P. Nikander, and J. Ott. *Packet-level caching for information-centric networking*. in *ACM SIGCOMM, ReArch Workshop*. 2010.
82. Flajolet, P., D. Gardy, and L. Thimonier, *Birthday paradox, coupon collectors, caching algorithms and self-organizing search*. *Discrete Applied Mathematics*, 1992. **39**(3): p. 207-229.
83. Lu, Y., T.F. Abdelzaher, and A. Saxena, *Design, implementation, and evaluation of differentiated caching services*. *IEEE Transactions on Parallel and Distributed Systems*, 2004. **15**(5): p. 440-452.
84. Oliveira, E.M.R., et al., *Mobile data traffic modeling: Revealing temporal facets*. *Computer Networks*, 2017. **112**: p. 176-193.
85. Tang, J., et al., *Characterising temporal distance and reachability in mobile and online social networks*. *ACM SIGCOMM Computer Communication Review*, 2010. **40**(1): p. 118-124.
86. Robins, G., et al., *Recent developments in exponential random graph (p^*) models for social networks*. *Social networks*, 2007. **29**(2): p. 192-215.
87. Snijders, T.A., G.G. Van de Bunt, and C.E. Steglich, *Introduction to stochastic actor-based models for network dynamics*. *Social networks*, 2010. **32**(1): p. 44-60.
88. Grundy, A., *Congestion control framework for delay-tolerant communications*. 2012, University of Nottingham.
89. Wietrzyk, B. and M. Radenkovic, *Realistic large scale ad hoc animal monitoring*. *International Journal On Advances in Life Sciences*, 2010. **2**(1 and 2).
90. Brun-Laguna, K., et al. *(Not so) intuitive results from a smart agriculture low-power wireless mesh deployment*. in *Proceedings of the Eleventh ACM Workshop on Challenged Networks*. 2016. ACM.
91. Nicosia, V., et al., *Graph metrics for temporal networks*, in *Temporal networks*. 2013, Springer. p. 15-40.
92. Tiropanis, T., et al., *Network science, web science, and internet science*. *Communications of the ACM*, 2015. **58**(8): p. 76-82.
93. Easley, D. and J. Kleinberg, *Networks, crowds, and markets*. Vol. 8. 2010: Cambridge university press Cambridge.
94. Cunha, F.D., et al. *Is it possible to find social properties in vehicular networks?* in *2014 IEEE Symposium on Computers and Communications (ISCC)*. 2014. IEEE.
95. Bornholdt, S. and T. Röhl, *Self-organized critical neural networks*. *Physical Review E*, 2003. **67**(6): p. 066118.
96. Spring, N., R. Mahajan, and D. Wetherall. *Measuring ISP topologies with Rocketfuel*. in *ACM SIGCOMM Computer Communication Review*. 2002. ACM.
97. Hekmat, R. and P. Van Mieghem, *Connectivity in wireless ad-hoc networks with a log-normal radio model*. *Mobile networks and applications*, 2006. **11**(3): p. 351-360.

98. Yoneki, E., P. Hui, and J. Crowcroft. *Distinct types of hubs in human dynamic networks*. in *Proceedings of the 1st Workshop on social Network Systems*. 2008. ACM.
99. Huynh, V.S.H., M. Radenkovic, and R. John. *Understanding information centric layer of adaptive collaborative caching framework in mobile disconnection-prone networks*. in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. 2018. IEEE.
100. Huynh, V.S.H. and M. Radenkovic. *Understanding complementary multi-layer collaborative heuristics for adaptive caching in heterogeneous mobile opportunistic networks*. in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. 2018. IEEE.
101. Nelson, S.C., M. Bakht, and R. Kravets. *Encounter-based routing in DTNs*. in *IEEE INFOCOM 2009*. 2009. IEEE.
102. Ruan, M., X. Chen, and H. Zhou, *Centrality prediction based on K-order Markov chain in Mobile Social Networks*. Peer-to-Peer Networking and Applications, 2019: p. 1-11.
103. Buchegger, S., J. Mundinger, and J.-Y. Le Boudec, *Reputation systems for self-organized networks*. IEEE Technology and Society Magazine, 2008. **27**(1): p. 41-47.
104. Dan, A. and D. Towsley, *An approximate analysis of the LRU and FIFO buffer replacement schemes*. Vol. 18. 1990: ACM.
105. Bilal, M. and S.-G. Kang. *Time aware least recent used (TLRU) cache management policy in ICN*. in *16th International Conference on Advanced Communication Technology*. 2014. IEEE.
106. Bilal, M. and S.-G. Kang, *A cache management scheme for efficient content eviction and replication in cache networks*. IEEE Access, 2017. **5**: p. 1692-1701.
107. Zhang, G., et al., *An optimal cache placement strategy based on content popularity in content centric network*. JOURNAL OF INFORMATION & COMPUTATIONAL SCIENCE, 2014. **11**(8): p. 2759-2769.
108. Mardham, D., et al. *Opportunistic distributed caching for mission-oriented delay-tolerant networks*. in *2018 14th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*. 2018. IEEE.
109. Kotz, D. and T. Henderson, *Crawdad: A community resource for archiving wireless data at dartmouth*. IEEE Pervasive Computing, 2005. **4**(4): p. 12-14.
110. Piorkowski, M., N. Sarafijanovic-Djukic, and M. Grossglauser, *CRAWDAD data set epfl/mobility (v. 2009-02-24)*. 2009.
111. Scott, J., et al., *CRAWDAD dataset cambridge/haggle (v. 2009-05-29)*. CRAWDAD wireless network data archive, 2009.
112. Scott, J., et al., *CRAWDAD dataset cambridge/haggle (v. 2006-09-15)*. CRAWDAD wireless network data archive, 2006.
113. Leguay, J. and F. Benbadis, *CRAWDAD data set upmc/rollernet (v. 2009-02-02)*. 2009, Feb.
114. Cabrero, S., et al., *CRAWDAD dataset oviedo/asturies-er (v. 2016-08-08)*. Downloaded, 2016.
115. Banerjee, N., et al., *CRAWDAD data set umass/diesel/transfer/fall2007 (v. 2008-09-14)*. 2008.
116. Cha, M., et al. *I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system*. in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. 2007. ACM.
117. Cheng, X., C. Dale, and J. Liu. *Statistics and social network of youtube videos*. in *2008 16th International Workshop on Quality of Service*. 2008. IEEE.
118. Noulas, A., et al. *An empirical study of geographic user activity patterns in foursquare*. in *Fifth international AAAI conference on weblogs and social media*. 2011.

119. Yang, D., et al., *Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs*. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2014. **45**(1): p. 129-142.
120. Keränen, A., J. Ott, and T. Kärkkäinen. *The ONE simulator for DTN protocol evaluation*. in *Proceedings of the 2nd international conference on simulation tools and techniques*. 2009. ICST (Institute for Computer Sciences, Social-Informatics and
121. Bornholdt, S. and H.G. Schuster, *Handbook of graphs and networks: from the genome to the internet*. 2006: John Wiley & Sons.
122. Haight, F.A., *Handbook of the Poisson distribution*. 1967.
123. Vega, D., et al. *Topology patterns of a community network: Guifi. net*. in *2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2012. IEEE.
124. Bettstetter, C., G. Resta, and P. Santi, *The node distribution of the random waypoint mobility model for wireless ad hoc networks*. IEEE Transactions on mobile computing, 2003. **2**(3): p. 257-269.
125. Bettstetter, C., H. Hartenstein, and X. Pérez-Costa, *Stochastic properties of the random waypoint mobility model*. Wireless Networks, 2004. **10**(5): p. 555-567.
126. Choi, H.-K. and J.O. Limb. *A behavioral model of web traffic*. in *Proceedings. Seventh International Conference on Network Protocols*. 1999. IEEE.
127. Conti, M., P. Gasti, and M. Teoli, *A lightweight mechanism for detection of cache pollution attacks in named data networking*. Computer Networks, 2013. **57**(16): p. 3178-3191.
128. Wang, L., J. Kangasharju, and J. Crowcroft, *You Really Need A Good Ruler to Measure Caching Performance in Information-Centric Networks*. arXiv preprint arXiv:1603.05963, 2016.
129. Chai, W.K., et al. *Cache "less for more" in information-centric networks*. in *International Conference on Research in Networking*. 2012. Springer.
130. Radenkovic, M., J. Crowcroft, and M.H. Rehmani, *Towards low cost prototyping of mobile opportunistic disconnection tolerant networks and systems*. IEEE Access, 2016. **4**: p. 5309-5321.
131. Ahmed, S. and S.S. Kanhere, *HUBCODE: hub-based forwarding using network coding in delay tolerant networks*. Wireless Communications and Mobile Computing, 2013. **13**(9): p. 828-846.
132. Radenkovic, M., et al. *Enabling Real-time Communications and Services in Heterogeneous Networks of Drones and Vehicles*. in *The 15th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2019.
133. Radenkovic, M. and A. Walker. *CognitiveCharge: disconnection tolerant adaptive collaborative and predictive vehicular charging*. in *Proceedings of the 4th ACM MobiHoc Workshop on Experiences with the Design and Implementation of Smart Objects*. 2018. ACM.
134. Huynh, V.S.H. and M. Radenkovic. *A novel cross-layer framework for large scale emergency communications*. in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. 2017. IEEE.
135. Bonino, D., et al. *Almanac: Internet of things for smart cities*. in *2015 3rd International Conference on Future Internet of Things and Cloud*. 2015. IEEE.
136. Sanchez, L., et al., *SmartSantander: IoT experimentation over a smart city testbed*. Computer Networks, 2014. **61**: p. 217-238.
137. del Cerro, J., et al. *Aerial fleet in rhea project: A high vantage point contributions to robot 2013*. in *ROBOT2013: First Iberian Robotics Conference*. 2014. Springer.
138. Radenkovic, M. and V.S.H. Huynh. *Low-cost mobile personal clouds*. in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*. 2016. IEEE.

139. Kim, H. and N. Feamster, *Improving network management with software defined networking*. IEEE Communications Magazine, 2013. **51**(2): p. 114-119.
140. Kreutz, D., et al., *Software-defined networking: A comprehensive survey*. arXiv preprint arXiv:1406.0440, 2014.
141. Han, B., et al., *Network function virtualization: Challenges and opportunities for innovations*. IEEE Communications Magazine, 2015. **53**(2): p. 90-97.
142. Mijumbi, R., et al., *Network function virtualization: State-of-the-art and research challenges*. IEEE Communications Surveys & Tutorials, 2015. **18**(1): p. 236-262.
143. McKeown, N., et al., *OpenFlow: enabling innovation in campus networks*. ACM SIGCOMM Computer Communication Review, 2008. **38**(2): p. 69-74.
144. Bianchi, G., et al., *OpenState: programming platform-independent stateful openflow applications inside the switch*. ACM SIGCOMM Computer Communication Review, 2014. **44**(2): p. 44-51.
145. Jiang, N., et al., *Reinforcement learning for real-time optimization in NB-IoT networks*. IEEE Journal on Selected Areas in Communications, 2019. **37**(6): p. 1424-1440.
146. Chen, G., et al. *The spatiotemporal interplay of regularity and randomness in cellular data traffic*. in *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*. 2017. IEEE.
147. Chemouil, P., et al., *Special Issue on Artificial Intelligence and Machine Learning for Networking and Communications*. IEEE Journal on Selected Areas in Communications, 2019. **37**(6): p. 1185-1191.
148. Hargreaves, E., et al., *Fairness in online social network timelines: Measurements, models and mechanism design*. Performance Evaluation, 2019. **129**: p. 15-39.
149. Ding, A.Y., et al., *Software defined networking for security enhancement in wireless mobile networks*. Computer Networks, 2014. **66**: p. 94-101.
150. Lu, X., et al., *Privacy information security classification for internet of things based on internet data*. International Journal of Distributed Sensor Networks, 2015. **11**(8): p. 932941.
151. Radenkovic, M., et al. *Providing security for wireless community networks*. in *Proceedings of the 2013 ACM MobiCom workshop on Lowest cost denominator networking for universal access*. 2013. ACM.