IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 30, 2018, accepted May 17, 2018, date of publication May 23, 2018, date of current version June 29, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2840040

Adaptive Real-Time Predictive Collaborative Content Discovery and Retrieval in Mobile Disconnection Prone Networks

MILENA RADENKOVIC[1], VU SAN HA HUYNH[1], AND PIETRO MANZONI[2]

[1]School of Computer Science, The University of Nottingham, Nottingham NG8 1BB, U.K.

[2]Departamento de Informática de Sistemas y Computadores, Universitat Politècnica de València, 46022 València, Spain

Corresponding author: Milena Radenkovic (milena.radenkovic@nottingham.ac.uk)

ABSTRACT Emerging mobile environments motivate the need for 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. In this paper, we investigate complex challenges related to the mobile disconnection tolerant discovery of content that may be stored in mobile devices and its delivery to the requesting nodes in mobile resource-constrained heterogeneous environments. We propose a new adaptive real-time predictive multi-layer caching and forwarding approach, CafRepCache, which is collaborative, resource, latency, and content aware. CafRepCache comprises multiple multi-layer complementary real-time distributed predictive heuristics which allow it to respond and adapt to time-varying network topology, dynamically changing resources, and workloads while managing complex dynamic tradeoffs between them in real time. We extensively evaluate our work against three competitive protocols across a range of metrics over three heterogeneous real-world mobility traces in the face of vastly different workloads and content popularity patterns. We show that CafRepCache consistently maintains higher cache availability, efficiency and success ratios while keeping lower delays, packet loss rates, and caching footprint compared to the three competing protocols across three traces when dynamically varying content popularity and dynamic mobility of content publishers and subscribers. We also show that the computational cost and network overheads of CafRepCache are only marginally increased compared with the other competing protocols.

INDEX TERMS Mobile disconnection tolerant networks, content discovery and retrieval, content caching, latency awareness, congestion awareness.

I. INTRODUCTION

We live in the era when smart, ubiquitous devices are embedded in our day-to-day activities and allow us to form diverse dynamic communities in which we are able to share rich and complex data. Majority of the applications and services which are hosted in the mobile edges today suffer from two related problems: they do not handle well limited network coverage and they may generate dynamically changing volumes of traffic which can cause localized congestion. Both of these cause delays and packet losses in the network and so seriously impair end-user service quality [4], [19], [39], [40]. There is currently limited work that combines support for rich collaboration of dynamically varying numbers of simultaneous distributed mobile users with predictive localized responsiveness to congestion in the face of lack of global network information and unknown-in-advance user publishers and subscribers activity patterns. While mobile offloading approaches [35] and content distribution network approaches [19] address problems of fast increasing data in mobile networks and traffic surges respectively, they do not address high topology dynamics with intermittent disconnections and dynamically changing publishers and subscribers workload patterns which we focus on in this paper.

We propose to integrate distributed predictive adaptive localized ad-hoc real-time decision making for content retrieval across multiple layers to include predictive congestion avoidance in mobile disconnection tolerant networks and predictive content-centric layer in the face of unknown-in-advance dynamically changing social, interest and network structures. We investigate complex challenges of mobile latency aware and disconnection tolerant discovery of content stored in remote mobile devices and its delivery to the requesting nodes in heterogeneous mobile disconnection prone environments.

At the heart of our approach is a distributed edge based collaborative caching which uses several multidimensional predictive analytics that builds multi-attribute complementary predictive heuristics and utilities. We build on the principle of dynamic predictive relative utilities [14], [18] and propose a new intelligent collaborative caching algorithm which allows individual nodes to achieve greater caching utility compared to the case where no collaboration is used in making decisions in distributed mobile disconnection tolerant caching. The questions we specifically focus on are: where to cache, what to cache and how to manage the cache. Previous research has shown that collaborative caching usually outperforms both locally and centrally optimized algorithms [22]. Note that our focus is not to build a protocol that forces nodes to collaborate or provide protection against malicious behavior, but rather to design an underlying algorithm that can adaptively share distributed cache space across trusted collaborators when both network topology and workloads are dynamically varying. We extend the idea of behavioral locality to *exploit similarities between the content interests and users' connectivity patterns*. We expand the idea of *content popularity* with *popularity stability* in order to minimize the negative impact of flash crowds [4]. We tackle the challenge of maximizing the number of data chunks retrieval with as low delay as possible even in sparse fragmented topologies. We propose that distributed nodes use CafRepCache algorithm to form dynamic transient interest and data dissemination topologies based on predictive analysis and commonalities between their interests, caches and retrieval histories as well as connectivity histories. This provides each node with a set of overlay neighbors whose browsing history most closely resembles 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*. We argue that *adaptive replication* and *caching* are *both necessary* to address multi-user data communications in dynamic *fragmented and sparse topologies*. We envisage that CafRepCache will be an integral part of a robust network support that will allow reliable operation of future rich mobile multimedia services in a variety of application areas (such as smart transportations [31], [42] or smart festivals mass events [33], [34]) where users often suffer from limited network coverage and congestion.

The rest of the paper is organized as follows. Section II provides a systematic review of related work across a range of specific multilayer challenges in peer-to-peer content sharing in mobile disconnection prone networks and outline ten criteria which we use to evaluate the related work and guide our proposal. In Section III, we describe key features of our proposal to address the identified criteria and propose congestion, latency and content aware adaptive predictive and collaborative protocol, CafRepCache. More specifically, we propose a novel distributed multi-layer real-time predictive and adaptive CafRepCache design, provide its analytical model and identify multiple novel complementary real-time predictive CafRepCache's heuristics that enable CafRepCache to manage adaptive real-time complex tradeoffs among time-varying networks, dynamically changing resources, dynamically changing workloads, and changing content popularity. Section IV describes CafRepCache architecture and design space, and provides pseudo-code which uses multiple complementary multi-layer adaptive fully distributed real-time predictive analytics to allow CafRepCache nodes to dynamically adapt to dynamic underlying topologies, dynamic resources, workloads and content popularity. Section V provides a detailed description of multiple sets of experiments which we perform to evaluate CafRepCache performance against three state-of-the-art intelligent caching algorithms in heterogeneous real-world mobile disconnection prone networks. More specifically, we compare CafRepCache against SocialCache [9], HyMobi [10] and Least Recently Used (LRU) across a range of metrics over three heterogeneous real-world mobile social and vehicular traces for dynamically varying workload patterns and content popularity. We show that CafRepCache fully localized *single node and ego network* adaptive real-time collaborative predictive heuristics can better predict and adapt to dynamically varying regional and temporal resources, forwarding possibilities and data interests. CafRepCache consistently manages to achieve higher content discovery and delivery rates and better caching efficiency while keeping lower delays and packet loss rates compared to three competing protocols for very different real-world dynamically changing mobile topologies. We also show that CafRepCache computational and network overhead costs are only marginally increased compared to the other competing protocols. Section V gives a conclusion and outlines future work.

II. RELATED WORK

In this section, we provide systematic review of related work across a range of specific challenges for peer to peer content sharing in mobile disconnection prone networks with dynamically changing workloads. We begin with reviewing useful state of the art content centric and content distribution techniques for peer to peer content sharing and identify what prevents them from being used in the dynamically changing and disconnection prone network environments and workloads. We then move to reviewing state of the art approaches for data forwarding and congestion control in mobile disconnection tolerant networks on which we build our proposal. Subsequently, we review emerging intelligent caching mechanisms which are most relevant for dynamic content and network scenario and identify their limitations. We present a table which summarizes ten core criteria which we used to evaluate the related work and which guide our proposal.

A. P2P CONTENT SHARING

Recent advances in peer-to-peer content sharing and content-centric networks focus on improving the performance of content discovery and retrieval by increasing success ratio and reducing the delivery latency and cost in fixed network

TABLE 1. Overview of the techniques that deal with efficient content dissemination and query in mobile disconnection prone networks.

	Behave	Social Forward LRU	Social Cast	Region Cache	Chunk Cache	Social Cache	Focal	HyMobi	CafRep	CafRepCache
Adaptive Forwarding	No	Yes	Yes	No	Limited	Yes	Yes	Yes	Yes	Yes
Adaptive Replication	No	No	No	No	Yes	No	No	No	Yes	Yes
Fully-Localised Collaborative Predictive Caching	No	No	No	No	Yes	Yes	Yes	No	No	Yes
Congestion Aware	No	No	No	No	No	No	No	No	Yes	Yes
Social/Contact Aware	No	Yes	Yes	No	No	Yes	No	Yes	Yes	Yes
Resource Aware	No	No	Yes	No	No	No	Yes	Yes	Yes	Yes
Fully Localised	No	Yes	Yes	No	Yes	Yes	No	Yes	Yes	Yes
Disconnection Tolerant	No	Yes	Yes	No	Yes	Yes	No	Yes	Yes	Yes
End-to-End Latency Aware	No	No	No	No	No	No	Yes	No	No	Yes
Dynamic Changing Topologies	No	Yes	Yes	No	Yes	Yes	No	Yes	Yes	Yes

scenarios [19], [20], [25]. Dernbach *et al.* [19] propose a regional caching approach of video content that captures the overlap between inter-regional and intra-regional preferences to provide better cache performance. Frey *et al.* [20] propose a peer-to-peer cache-oriented approach for Web applications based on the principles of Behavioral Locality inspired by collaborative filtering. Dernbach *et al.* [19] and Frey *et al.* [20] apply centralized decision making and require global information, thus do not support fully-localized distributed decision making which we aim to do in our proposal (see Table 1). Carofiglio *et al.* [25] propose an approach which combines latency-aware fully distributed caching and load-aware dynamic forwarding strategies in order to improve the content-centric network delivery performance. To optimize performance of popularity-aware caches which requires optimal split of the forwarded traffic, Carofiglio *et al.* [25] route more popular content requests through a single path without considering congestion control and avoidance (Table 1). Carofiglio *et al.* [25] also do not support fully-localized (i.e. open loop) decision makings. None of the above approaches support disconnection tolerance and dynamically changing workloads and network topologies which are inherent in the scenarios of mobile publishers and subscribers [5].

The problem of content sharing among mobile publishers and subscribers is still not understood sufficiently clearly [7], [38], [45] and leads to lower quality of service. In the rest of this section, 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. Even though these protocols support fully-localized distributed decision making for mobile subscribers and publishers, they do not address all of the criteria we identify and tackle in our proposal.

B. CONTENT FORWARDING, REPLICATION, CONGESTION CONTROL AND AVOIDANCE IN MOBILE DISCONNECTION PRONE NETWORKS

Daly and Haahr [18] propose SimBetTS which combines betweenness, similarity and tie strength for social routing metric to direct the traffic to more central nodes and increase the probability of finding the optimal relay for delivering packets to the destination. Costa *et al.* [38] propose SocialCast which is a routing framework for publish-subscribe that utilizes social metrics (i.e. colocation and mobility patterns among communities) to identify the best relay nodes that support delay-tolerant communication in human networks. Zhou *et al.* [37] propose an efficient data forwarding in opportunistic networks which uses predictive nodes' social contact patterns from the temporal perspective. None of the above approaches supports latency awareness, resource awareness, congestion control and avoidance (Table 1) which is necessary for improving network reliability of social routing and forwarding that may congest at the points with higher social centrality.

Radenkovic and Grundy [14] propose fully localized adaptive forwarding, replication and congestion control protocols, Café and CafREP, to enable congestion-aware mobile social framework for predictively adaptive data forwarding and rate adaptation over heterogeneous disconnection tolerant networks. Flores *et al.* [10] propose a social-aware hybrid offloading strategy 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. However, Flores *et al.* [10] do not consider predictive congestion and latency awareness (Table 1). The above approaches mainly focus on adaptive content forwarding, replication and congestion control in mobile disconnection prone networks, but have not focused on in-network fully-localized predictive caching which is important to addressing latency awareness which we aim to do in this paper (see Table 1). In the next section, we review the state of the art content caching algorithms in mobile disconnection prone networks.

C. CONTENT CACHING IN MOBILE DISCONNECTION PRONE NETWORKS

Le *et al.* [9] propose a forwarding and cache replacement policy for SocialCache based on content popularity driven by frequency and freshness of content requests. As part of its replacement, SocialCache 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 [9] is not resource and congestion aware (Table 1).

Vigneri *et al.* [32], [41] show that vehicles and people acting as mobile relays can store more replicas of popular content and thus can increase the "effective" storage capacity a user has access to since a user can meet a large number of vehicles and people. Vigneri *et al.* [32], [41] focus on the problem of offloading data traffic from main infrastructure to vehicles acting as mobile caches. However, the above approaches neither address dynamic changing topologies and workload pattern, nor take into account resource awareness, network overhead cost and social awareness (e.g. between vehicles and people) which have been shown in Table 1 as important criteria in mobile disconnection prone networks.

Wang *et al.* [2] propose a low-complexity heuristic FairCache algorithm that models collaborative caching as a bargaining game to address the fairness problems in fixed environments. Wang *et al.* [2] suffer from multiple limitations when applied to the mobile dynamic heterogeneous context: i) Wang *et al.* [2] do not support dynamic content request patterns which is significant for fully-localized collaborative predictive caching (Table 1). ii) Wang *et al.* [2] consider cache hit ratio to define utility function but does not take into account end-to-end latency (Table 1) [3] which is important for our mobile disconnection tolerant scenarios [4]. iii) Wang *et al.* [2] do not support congestion control, resource awareness (Table 1) 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).

Wang *et al.* [2] and Bertsimas *et al.* [13] have shown that collaborative caching usually outperforms locally optimized algorithms [22] and attaining a global optimum often disadvantages some parties e.g. nodes may be unfairly exploited by other caches redirects (at the cost of their own performance), thus we argue that our work does not aim to achieve the global optimum as in [24] and [26], but to propose a distributed fully-localized predictive caching algorithm in collaborative manner which allows every individual node to achieve greater utility.

Our previous work [4] describes our early proof of concept CafRepCache that combines multi-path content and interest forwarding and replication with latency aware adaptive collaborative cognitive caching in heterogeneous opportunistic mobile networks which utilizes fully localized and ego networks multi-layer predictive heuristics about dynamically changing topology, resources and popularity content. CafRepCache [4] does not require the global knowledge of network topology and content, and it aims to enable the high-performance efficiency of individual caches while avoiding draining the resources of other nodes and decreasing their performances. In this work, we formalize the model and provide comprehensive performance analysis of CafRepCache in a wide range of different context scenarios and against more competitive and benchmark protocols.

D. SUMMARY

In Table 1 we systematically outline core criteria that we used to review the related work and guide our proposal for a distributed multi-layer real-time predictive and adaptive forwarding and caching framework in mobile disconnection prone networks (CafRepCache).

Table 1 presents a summary of the techniques discussed in this section in terms of the ten criteria. Note that, apart from CafRepCache, none of the existing approaches provides support for all criteria: adaptive forwarding, adaptive replication, fully-localized collaborative predictive caching, congestion awareness, social contact awareness, resource awareness, full localisation, disconnection tolerance, end-to-end latency awareness and dynamic changing topologies. In our section V, we compare our CafRepCache algorithm against state-of-the-art intelligent forwarding and caching algorithms in mobile disconnection tolerant networks: SocialCache [9] and HyMobi [10], and also use Least Recently Used (LRU) algorithm as a benchmarking cache replacement algorithm.

III. CAFREPCACHE PROPOSAL

CafRepCache framework has a distributed multilayer structure as shown in Fig. 1. CafRepCache is able to perform distributed predictive analytics of multivariate mixed data (e.g. content and mobility) and manage dynamic trade-offs between minimizing the end-to-end latency and maximizing content delivery while enabling resource efficiency and congestion avoidance. Fig. 1 shows CafRepCache cross-layer architecture which consists of multi-layers: Physical Layer, Mobile Disconnection Tolerant Network (Mobile DTN) Layer, Content-Centric Network (CCN) Layer and Application Layer. Nodes in Physical Layer may include IoT mobile devices (e.g. car, smartphone, etc.) or static devices (e.g. Road Side Units) with different dynamic physical constraints (e.g. storage, computational resources, energy, etc.). In the Mobile DTN Layer, nodes are coupled with users whose connectivity forms scale-free graph that has power-law connectivity patterns (such as complex social graphs have). Nodes in the Mobile DTN Layer scan and discover neighbors, exchange information via multi-dimensional vectors which may include predictive analytics of node connectivity patterns, predictive analytics of resource availability of nodes and their ego networks. In the Content-Centric Network Layer, contents are generated by publishers and requested by subscribers; publishers and subscribers may be mobile or static devices in the network. We consider that each content has a list of attributes (e.g. name, topic, type, etc.) but the knowledge on content popularity, content locality, content distribution and content availability *are not known in advance*. Each content in the network consists of multiple chunks which have their own characteristics such as chunk

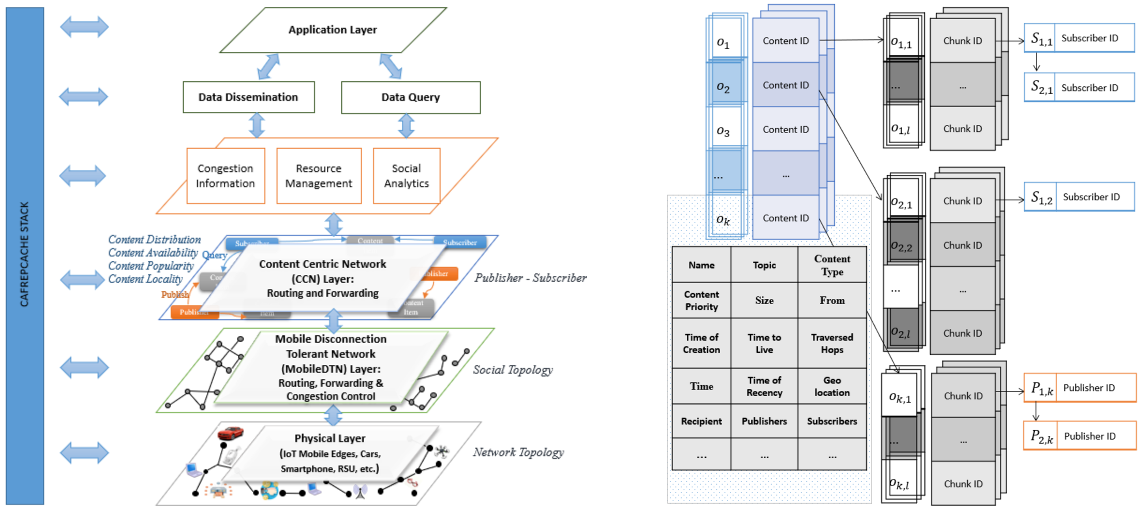


FIGURE 1. Multi-layer CafRepCache architecture.

size, chunk popularity, etc. Based on both node's and content's characteristics, we propose three integrated modules: Resource Management, Social Analytics and Congestion Awareness. Resource Management module allows real-time predictive buffer storage, intelligently schedules and prioritizes packets; Social Analytics module examines social metrics (e.g. similarity, betweenness, tie strength) of neighbors and continuously evaluates its social heuristic utility value; Congestion Awareness module resolves the retentiveness, receptiveness and congesting rate of nodes to avoid overload and congestion. On the top of our framework, the Application Layer offers two types of services: content dissemination and multi-attribute content query.

A. CAFEREPACHE FRAMEWORK AND SYSTEM MODEL

We model CafRepCache system as a network G that consists of a set N of nodes 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 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. At each node $n_i \in N$, $q_{i,k}^t$ is the normalized request rate of the content o_k (i.e. content popularity) observed locally from n_i at time t, $\sum_k q_{i,k}^t = 1$; $z_{i,k}^t$ is the normalized aggregated request rate of the content o_k observed from all the neighbors of n_i at time t, $\sum_k z_{i,k}^t = 1$. When two CafRepCache nodes are in contact, they exchange their local content popularity observations. Each node continuously resolves the value of dynamically changing content popularity based on its local observation and the collaborative observations it gets from others.

$W(q_{i,k}^t, z_{i,k}^t) = \alpha q_{i,k}^t + \beta z_{i,k}^t$ denotes the function to weight the value of collaborative observations over local observation.

At time t, each node $n_i \in N$ may act as either a subscriber of content o_k, denoted by $S_{i,k}^t$ or a publisher $P_{i,k}^t$ or a caching point $C_{i,k}^t$ with caching capability. Thus for any content o_k, a set of subscribers who are interested in o_k is denoted as $S_k^t = \{S_{i,k}^t | n_i \in N\}$ and so on.

We integrate "ego network" of each node n_i: EN_i as a dynamic network consisting of the node n_i and contacts it meets most *frequently* or most *recently*. In this way, ego network allows each node to give its own regional or temporal perspective of the network (or both are included).

To model our caching strategy, we denote $x_{i,k}^t \in \{0, 1\}$ whether node n_i decides to cache or not content o_k at time t, $y_{i,j,k}^t \in \{0, 1\}$ whether node n_i forwards or not content o_k's request to neighbor $n_j \in EN_i$ at time t, $\omega_{i,j,k}^t \in \{0, 1\}$ whether node n_i decides to offload or not content o_k to n_j and $\varepsilon_{i,k} \in \{0, 1\}$ whether node n_i decides to drop or not content o_k at time t.

Table 2 sums up the main notations used in the paper.

In Game Theory each node aims to optimize its personal "utility". We model our collaborative cognitive caching as a "bargaining game" inspired by heuristic algorithm FairCache [2] and extend it to address real-world challenges about the lack of support for *dynamic demand matrix*, *dynamic node availability* and *congestion* identified in [2], [4], and [10]. CafRepCache does this by enabling responsiveness to dynamically changing network topology, congestion avoidance and varying patterns of content publishers/subscribers while allowing low latency content retrieval, high cache efficiency and efficient use of resources. Our cognitive caching utility aims to serve subscribers with the lowest possible delay and without saturating available resources, thus either using its local cache or by redirecting

TABLE 2. Table of notions used in the paper.

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 drop 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
US	social utility
UR	resource utility
UE	energy utility

a request to a nearby collaborative cache, rather than forwarding to the original publisher.

B. BACKGROUND OF BARGAINING GAMES

The bargaining game is a model [1] 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 cooperating with others.

Definition 1: An in-network caching problem is a pair $< U, \left(u_1^0, u_2^0, u_3^0, \ldots, u_i^0\right) >$ 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 $(u_1^*, u_2^*, \ldots u_i^*) = f(U, u_1^0, \ldots, u_i^0)$.

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* $f\left(U, u_i^0\right) \neq u_i$
- 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$(u_1^*, u_2^*, \ldots u_i^*) \in U' \subset U$ *then* $= f(U', u_1^0, \ldots, u_i^0) = f(U, u_1^0, \ldots, u_i^0) = (u_1^*, u_i^*)$
- Independence of linear transformation: Let $u_1' = c_1 u_1 + c_2, u_2' = c_3 u_2 + c_4, \ldots$ *then* $f(U', c_1 u_1^0 + c_2, \ldots, c_x u_i^0 + c_{x+1}) = (c_1 u_1^* + c_2, \ldots, c_x u_i^* + c_{x+1})$

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

$$arg_{(U_1,U_2,\ldots)\in U, U_i > u_i^0} \max \prod_{n_i \in N} (U_i - u_i^0)$$

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:

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

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

C. CAFREPCACHE

Core CafRepCache utility aims to provide higher delivery success and lower delay when serving dynamically changing users' demand patterns in dynamic mobile networks. The node's utility is improved by a caching node using its local cache, or redirecting the request to nearby collaborative cache. In order to improve cache performance, we integrate the following hybrid and complementary the criteria:

- *Content utility*: Content utility is dynamically resolved by performing predictive analytics of local cache hit rates and collaborating with other nodes – this increases the cache's utility since the relay nodes serve contents on behalf of publishers.
- *Social utility*: Social utility is collaboratively analyzed and resolved based on complex network metrics [18], [4]: similarity, betweenness and tie strength that is used to find the higher centrality nodes for caching contents or forwarding requests in order to increase the higher chances of requested contents being successfully delivered to subscribers (i.e. the success ratio). The intuition behind is that the caching nodes not only need to predict and cache the right contents but also have to deliver successfully that requested contents to subscribers.
- *Resource utility*: Resource utility [14] is based on dynamic real-time predictive analytics of available dynamic in-network storage, dynamic in-network delays that node n_i may add to content o_k before cache hit happens and the content are sent towards subscriber and by how likely caching (or forwarding a request of) content o_k will result in a congestion.
- *Energy utility:* Energy utility [8] is based on real-time predictive analytics of nodes' energy levels and energy cost of multi-path forwarding.

- *End-user utility*: End-user utility monitors and analyses numbers of different subscribers that adds value to the cache's utility in order to avoid serving content requests to only a small number of subscribers in specific local region.

More specifically, multiple, multi-layer and complementary CafRepCache utility metrics can be defined separately as below:

1) CONTENT UTILITY

Content utility is resolved dynamically by performing caching decisions on whether to cache, forward or delegate content chunks based on local and collaborative content popularity predictions. Intuitively, node n_i's content utility will be maximized when it caches the highest popularity contents as it will improve the cache hit ratio. Node n_i will also gain content utility by helping to redirect the request to a collaborative cache and a hit is attained there. The utility gained is proportional to the size of the cached content.

$$\sum_{o_k \in O} \delta_k W(q^t_{i,k}, z^t_{i,k}) x^t_{i,k} + \sum_{o_k \in O} \sum_{n_j \in EN_i} \delta_k W(q^t_{j,k}, z^t_{j,k}) y^t_{i,j,k} \quad (2)$$

Our content popularity analytics is defined as:

$$P(Ti) = \frac{ObservedTimePeriod(Ti)*}{Total - time * Betweeness\,(Ti) * Recency(Ti)}$$

$P\,(Ti)$ measures probability caching decision over a certain period (i.e. temporal locality) in which P is the weight that identifies the content popularity. Betweeness(Ti) is the temporal function that measures the time gap between continuous requests and Recency(Ti) denotes the most recent interest request. P(Ti) aims to provide a trade-off between current observed content popularity versus long-term interest in it in order to balance between potentially fake news and long-term useful content. 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).

2) SOCIAL UTILITY

Social utility is based on complex temporal social metrics (betweenness, similarity and tie strength [18], [4]) that favors higher centrality nodes to cache contents as they may have a high probability to deliver that content to their subscribers successfully. The intuition behind this is that in order to gain utility, node n_i not only needs to cache the right content but also need to ensure that the cached content will be delivered successfully to the subscriber. In mobile heterogeneous environments, only cache hit ratio does not guarantee for a subscriber to retrieve its requested content. The same logic is applied when node n_i decides to forward the request to its ego network that is shown in the second term of (3).

$$\sum_{o_k \in O} \sum_{s \in S^t_k} US_{i,s} \cdot x^t_{i,k} + \sum_{o_k \in O} \sum_{n_j \in EN_i} \sum_{s \in S^t_k} US_{j,s} \cdot y^t_{i,j,k}$$
$$US_{i,s} = \frac{1}{|F|} * \sum_{f \in F} \alpha US^f_{i,s}$$
$$f \in F = \{Social, EN_{Soc} | Social = \{Betweenness, Similarity, TieStrength\}\} \quad (3)$$

Social utility of node n_i is measured by the Betweenness value of n_i as well as the Similarity and Tie Strength between n_i and subscriber $s \in S^t_k$.

$US^{Bet}_{i,s} = \sum_{a=1}^{N} \sum_{b=1}^{a-1} \frac{g_{ab}(n_i)}{g_{ab}}$, $g_{ab}(n_i)$ is the number of paths linking node a and node b that includes n_i

$US^{Sim}_{i,s} = |N_i \cap N_s|$ is the similarity of contact between node n_i and subscriber s.

$US^{TS}_{i,s} = \frac{f(s)}{F(n)-f(s)} + \frac{rec(s)}{T(n)-rec(s)} + \frac{d(s)}{D(n)-d(s)}$ *combine both frequency and recency of contacts between cache node n_i and subscribers.*

$US^{EN}_{i,s} = \frac{1}{EN} \sum_{n_j \neq n_i \in EN} US^{Bet}_{j,s} + US^{Sim}_{j,s} + US^{TS}_{j,s}$ is the utility value of n_i's ego networks.

3) RESOURCE UTILITY

Resource utility is analyzed based on the remaining storage, the delay that n_i may add to content o_k before a cache hit happens and the content is sent towards subscriber. Furthermore, resource utility is also measured by how likely caching (or forwarding a request of) content o_k will result in a congestion.

$$\sum_{o_k \in O} \sum_{s \in S^t_k} UR_{i,s} \cdot x^t_{i,k} + \sum_{o_k \in O} \sum_{n_j \in EN_i} \sum_{s \in S^t_k} UR_{j,s} \cdot y^t_{i,j,k}$$
$$UR_{i,s} = \frac{1}{|F|} * \sum_{f \in F} \alpha UR^f_{i,s}$$
$$f \in F = \{Resource, EN_{Res} | Resource = \{Retentiveness, Receptiveness, CongestingRate\}\} \quad (4)$$

$UR^{Ret}_{i,s} = \theta_i - \sum_{o_k \in O} s_k$ as the node's available storage at time t, measured by the sum of all cached content occupancy subtracted from the node's cache buffer capacity.

$UR^{Rep}_{i,s} = \sum_{o_k \in O} t_{now} - t^i_{received}$ as the delay node n_i adds to content o_k, measured by the sum of differences between the current time and the time each content was received.

$$UR^{CR}_{i,s} = \left(\frac{t^{congestion}_{end} - t^{congestion}_{start}}{t^{congestion+1}_{start} - t^{congestion}_{end}}\right)\frac{f}{e^f} \quad \text{where } t^{congestion}_{start}$$

or $t^{congestion}_{end}$ denotes the time when congestion occurs or finishes, $t^{congestion}_{start}$ and $t^{congestion+1}_{start}$ are two adjacent congestions occurred, f is the observed congestion frequency. For simplicity, we assume $f = 1$.

$$UR^{EN}_{i,s} = \frac{1}{EN} \sum_{n_j \neq n_i \in EN} UR^{Ret}_{j,s} - UR^{Rep}_{j,s} - UR^{CR}_{j,s}$$

4) ENERGY UTILITY

$$\sum_{o_k \in O} UE_{i,k} x_{i,k}^t + \sum_{o_k \in O} \sum_{n_j \in EN_i} UE_{j,k} y_{i,j,k}^t \tag{5}$$

We denote E_i as a remaining energy of node n_i. Energy utility $UE_{i,k}$ is measured as the remaining lifetime RT of node n_i which is defined as $RT_{i,k} = \frac{\gamma E_i}{cost_k}$ in which $cost_k$ is the energy cost to transmit the content o_k. γ will be set low if node n_i is an important node which has to be protected from being battery drainage.

5) NUMBER OF SERVED SUBSCRIBERS

The utility value of node n_i increases as it serves to as many different subscribers as possible. The intuition behind is that a caching point will be encouraged to fairly serve not only highly connectivity subscribers requesting high popularity contents but also those who have lower connectivity and requests lower popularity contents.

$$\sum_{o_k \in O} S_k^t x_{i,k}^t + \sum_{o_k \in O} \sum_{n_j \in EN_j} S_k^t y_{i,j,k}^t \tag{6}$$

As the result, we define the total cache's utility function U_i^t (or U_i for simplicity) for node n_i as:

$$\begin{aligned} U_i &= \mu_1\,(2) + \mu_2\,(3) + \mu_3\,(4) + \mu_4\,(5) + \mu_5\,(6) \\ &= \Big(\textstyle\sum_{o_k \in O} \big(\mu_1 \delta_k W(q_{i,k}^t, z_{i,k}^t) + \mu_4 UE_{i,k} + \mu_5 S_k^t\big) \\ &\quad + \textstyle\sum_{o_k \in O} \sum_{s \in S_k^t} \big(\mu_2 US_{i,s} + \mu_3 UR_{i,s}\big)\Big)\, x_{i,k}^t \\ &\quad + \Big(\textstyle\sum_{o_k \in O} \sum_{n_j \in EN_i} \Big(\mu_1 \delta_k W(q_{j,k}^t, z_{j,k}^t) \\ &\qquad\qquad + \mu_4 UE_{j,k} + \mu_5 S_k^t\Big) \\ &\quad + \textstyle\sum_{o_k \in O} \sum_{n_j \in EN_i} \sum_{s \in S_k^t} \big(\mu_2 US_{j,s} + \mu_3 UR_{j,s}\big)\Big)\, y_{i,j,k}^t \end{aligned} \tag{7}$$

In [14] and [44], we show that assigning different weights μ to each of the CafRep's and CafRepCache's utilities results in different performance for real-world traces. We argue that there is a need to study and evaluate both empirically and theoretically the performance of different utility weighting models for multiple complementary heuristics in order to understand its impact on every layer of our caching framework. In this paper's experiments, we assume equal weight between each heuristic utility of CafRepCache as studying weighting models are out of scope of this paper and we plan to investigate adaptive weighting across utilities in the future work.

We propose CafRepCache as a low-complexity distributed and predictive heuristic algorithm that does not require the global knowledge of network topology and content, combines multi-path content and interest forwarding with adaptive collaborative cognitive caching and replication that allow individual nodes to achieve greater utility compared to the case when there is no collaboration in making decisions.

IV. CAFREPCACHE WITH COMPLEMENTARY MULTI-LAYER REAL-TIME PREDICTIVE HEURISTICS

We describe CafRepCache which comprises of multiple mobile edge predictive heuristics that leverage information on the local available resources, connectivity patterns, mobility of publishers/subscribers and dynamic content popularity. Fig. 2 shows CafRepCache' architecture and design space.

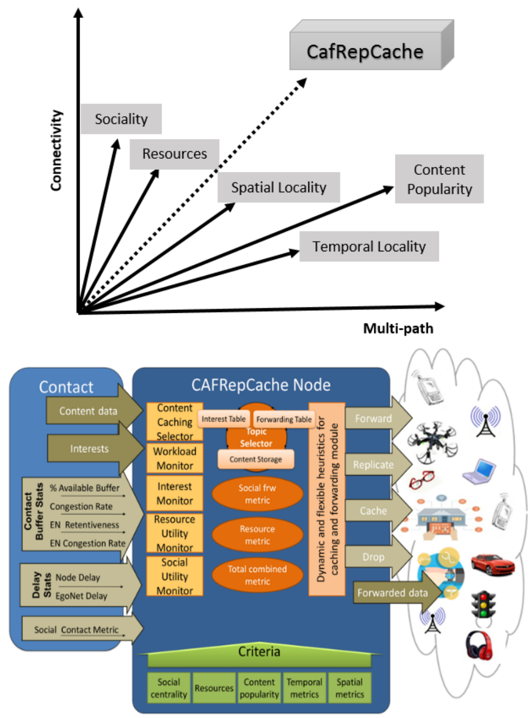


FIGURE 2. CafRepCache's design space and CafRepCache node architecture.

We provide CafRepCache pseudo code in Table 3.

Each node has a unique ID and every routed message has an associated key and state information which may contain 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.

Each publisher proactively advertises its list of content name to neighbors 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 neighbors who also have high sociality and availability. Intermediate node first checks if it has the requested content, then forward the content

TABLE 3. CafRepCache's pseudo code.

```
Handle Content Packet Arrival
When the Content is received at a Node:
if Node is a cache candidate of Content
    cache Content at Node
end if
while Node 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 > Node.Utility
            delegate the content from Node to Contact
        else
            remove the content from Node to free up the space
end while
forward the Content toward its subscribers

Handle Content Request Arrival
When the Interest (or request) of a Content is received at a Node:
if Node has already cached Content:
     send Content to Interest '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)
         Contact.contentPopularity =
exchContentPopularityInfo(Contact)

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
```

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 maximize its utility value or in other words, minimize its total cost. During the operation, the node adapts with changing environment and adjusts its local caching strategy ($x^t_{n,k}, y^t_{n,i,k}$) 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.

V. EVALUATION

A. EVALUATION METHODOLOGY

In order to better understand performance characterizes of the multi-dimensional and multi-layer design of CafRepCache, we perform two major groups of experiments each aiming to compare CafRepCache with multiple state-of-the-art and benchmark proposals across a range of criteria and in different contexts. As mobility and connectivity of the nodes have a major impact on the performance of any opportunistic communication protocol, it is fundamental to evaluate our caching algorithm over multiple heterogeneous real-world mobile data sets. We use San Francisco Cab [21], RollerNet [29] and Infocom [30] traces in ONE [17]. San Francisco Cab Trace [21] includes GPS traces of 550 cabs over a period of 30 days in the San Francisco Bay Area. RollerNet [29] spans three hours during which 62 rollerbladders travel about 20 miles in Paris and utilize Bluetooth on their cell phones for communication. Infocom [30] is a 4-day trace that consists of 78 volunteers equipped with Bluetooth devices and additional 20 static long-range devices placed at various semi-static and static locations of the conference venue. Fig. 3 shows analysis of the distribution of contact duration, isolation duration and number of contacts in these different mobile heterogeneous datasets.

San Francisco trace [21] is the most challenging trace compared with RollerNet [29] and Infocom [30] due to very short connectivity durations, very high disconnections and low number of contacts during connected times. We observe that both San Francisco [21] and RollerNet [29] 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 [30] 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.

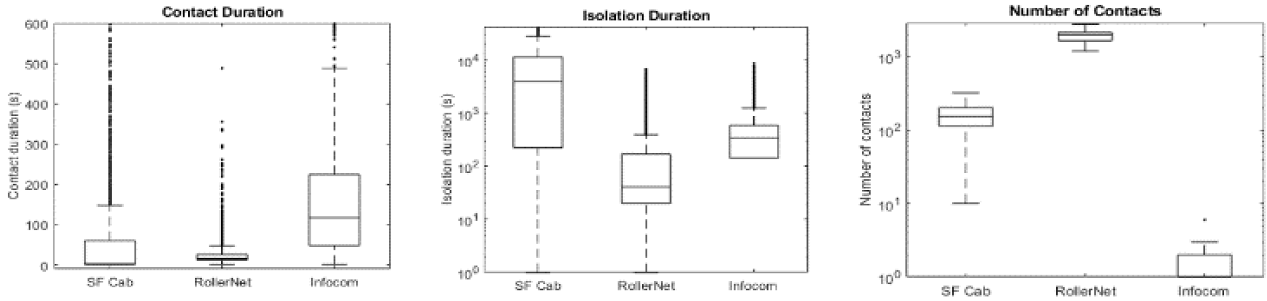


FIGURE 3. Distribution of contact duration, isolations and number of contacts.

In the first set of experiments, we compare CafRepCache against two state-of-the-art intelligent caching algorithms in opportunistic networks, SocialCache [9], HyMobi [10] and a benchmark protocol Least Recently Used (LRU) over different criteria: cache hit ratio, success ratio, delay and packet loss in the face of dynamically varying content popularity skewness in order to evaluate our forwarding and caching algorithm in the presence of dynamic content request patterns. We analyze two parts of *end-to-end content retrieval* (shown in Fig. 4): *content discovery*(i.e. time taken between sending interest packets to the network and discovering a requested content) and *content delivery* (i.e. time taken for the content to be delivered successfully to the interested subscribers from the time content was discovered in the network) in order to provide more insightful analysis of CafRepCache compared with other state of the state of the art and benchmarks protocols.

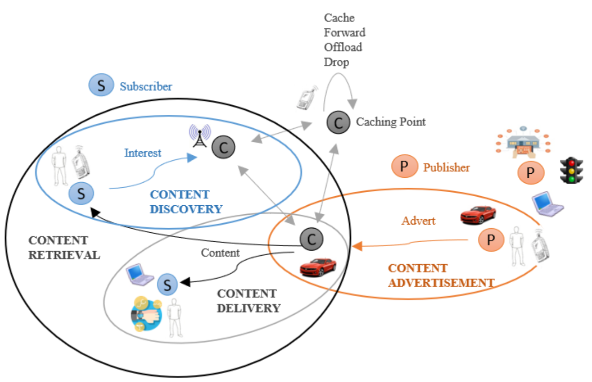


FIGURE 4. Content Retrieval: Content Discovery and Content Delivery.

In the second set of experiments, we conduct performance analysis of CafRepCache against two state-of-the-arts caching algorithms (SocialCache [9], HyMobi [10]) and a benchmark algorithm (LRU) while dynamically varying patterns of subscribers (in terms of their geographical and connectivity properties as well as their numbers). We show that CafRepCache outperforms other competing protocols in terms of success ratio, delay, packet loss and relative footprint reduction for content discovery, content delivery and the end-to-end content retrieval process in the presence of heterogeneous network connectivity and dynamic workloads.

In terms of forwarding interests, we assume nodes with higher social utility (i.e. betweenness and similarity centrality) and resource utility are preferred. Note that in the experiments, we assume interest packet size are relatively small compared to the content packet. The general simulation parameters details are shown in Table 4.

TABLE 4. Values of the simulation parameters.

	Parameter	Value
Mobility Traces	San Francisco/ RollerNet/Infocom Area	$\approx$ 120 km^2
Network	SF/RollerNet/Infocom # of nodes	100/62/100
	SF/RollerNet/Infocom Simulation time	12/3/3 hours
	Request generation rate	Variable
	Number of contents	1000
	File size	1 MB
	Chunk size	128 kB
	Interest packet size	8 kB
	Advert packet size	1 MB
	Cache size Total content population	0.5%

B. EVALUATION IN THE PRESENCE OF DYNAMIC CONTENT POPULARITY SKEWNESS

We generate content requests to follow Zipf-distribution using Hawkes process [28] in which the probability for a request of the k$^{\text{th}}$ 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. The smaller α leads to a more uniform popularity distribution, meaning contents are randomly requested. We vary content popularity skewness α while constantly setting localisation factor $\beta = 0.8$ to evaluate caching performance in the presence of dynamic content popularity over multiple criteria: cache hit probability, success ratio, latency and packet loss. We run six increments with content popularity skewness α ranging from 0.2 to 1.2 as we observe that our caching performance converges when α gets larger than >1.2. For each experiment with α, we assume that a random 50% of node population are subscribers and a random 25% of nodes

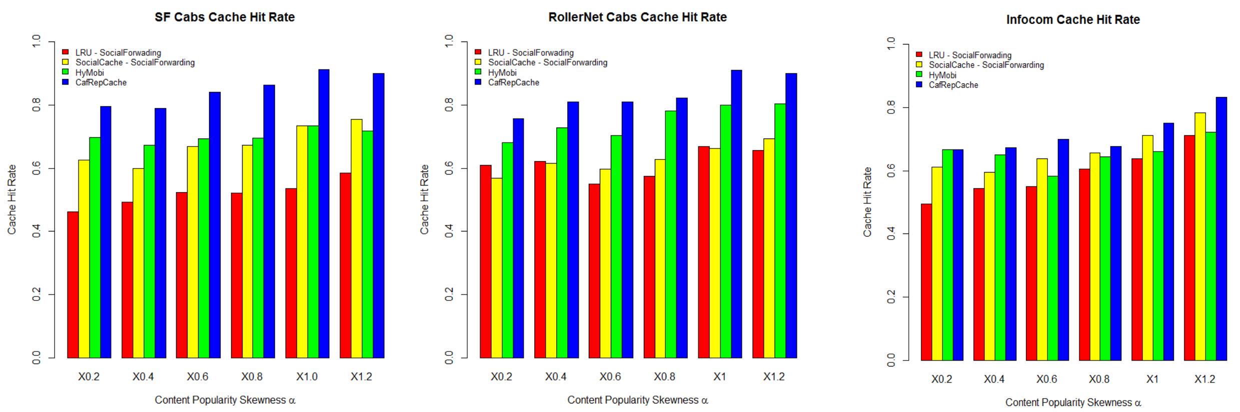


FIGURE 5. **Cache hit rate vs. Content popularity skewness.**

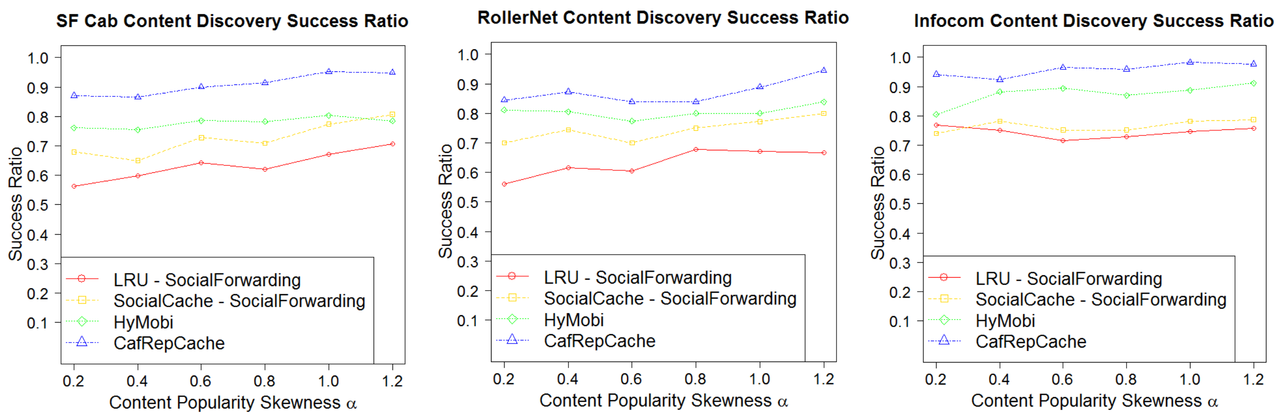


FIGURE 6. **Content discovery success ratio vs. Content popularity skewness.**

are publishers. All experiments are repeated ten times and averaged. We assume that contents are uniformly distributed among publishers. We compare CafRepCache performance against benchmark protocol LRU and state-of-the-art SocialCache [9] and HyMobi [10] over three heterogeneous traces ranging from vehicular to social: SF Cab [21], RollerNet [29] and Infocom [30]. In order to enable fair analysis, we implement LRU over social forwarding as other algorithms have different social forwarding algorithms and all three traces have been shown to have social character [4], [14].

We begin by analyzing the performance of *cache hit ratio* which refers to *how many* interest packets are *matched with the contents* in caching points *without* being forwarded to publishers and indicates the efficiency of caching decisions and locations. In Fig. 5, we show that CafRepCache achieves the highest cache hit ratio (typically above 88% for all three traces and in the face of dynamic content popularity skewness α) compared to state of the art algorithms SocialCache and HyMobi and benchmark protocol (LRU). CafRepCache is followed by HyMobi, which manages up to 70% for all three traces, SocialCache ranges from 56%-66% for San Francisco and RollerNet traces and up to 77% for Infocom trace. LRU has the worst performance manging around 44%-48% for San Francisco, 58%-64% for RollerNet and up to 70% for Infocom. We observe that higher α leads to bigger performance gap between CafRepCache and other competing protocols. This is because CafRepCache profits from its *adaptive caching* and *smart partial replication* which allow it to minimize packet loss rates and better predict and cope with fragmentation more efficiently. CafRepCache is also able to take advantage of highly skewed content popularity and content request locality to efficiently predict the incoming content requests.

Similarly to the improved cache hit ratios for CafRepCache, content discovery success ratios in CafRepCache outperforms other state-of-the-art and benchmark protocols across all three traces and for dynamically changing content popularity (as shown in Fig 6). 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

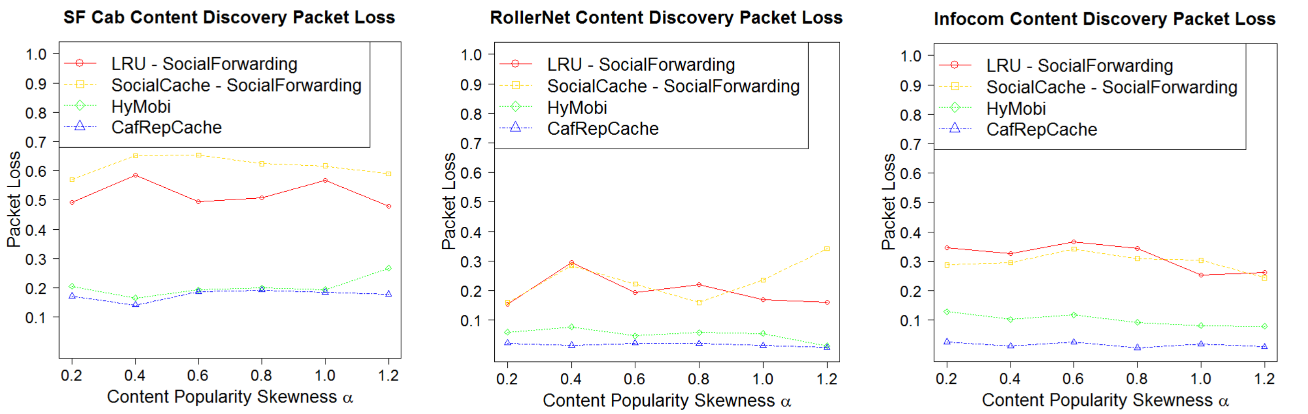


FIGURE 7. Content discovery packet loss vs. Content popularity skewness.

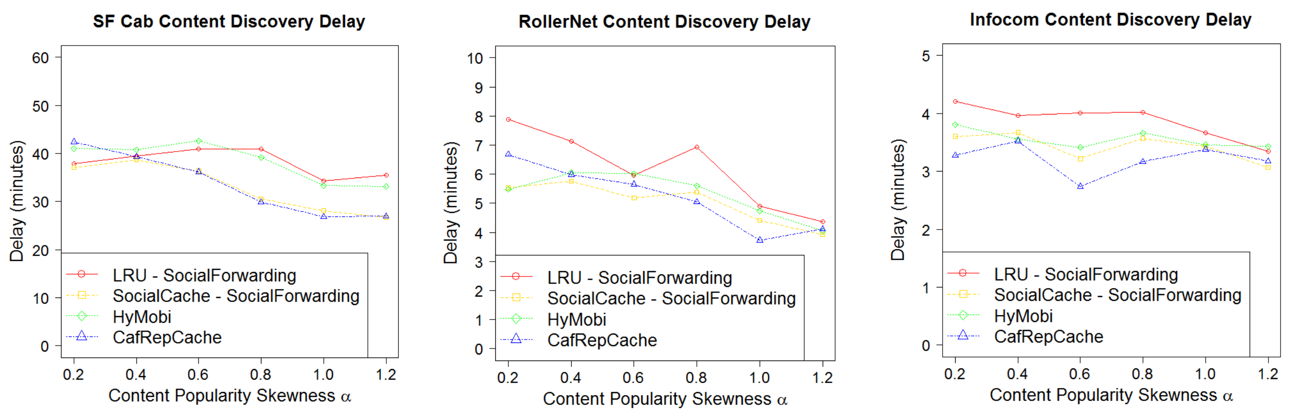


FIGURE 8. Content discovery delay vs. Content popularity skewness.

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.

In Fig. 7, CafRepCache shows lowest packet loss rates compared to HyMobi, SocialCache and LRU because it uses 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 currently in order 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 as shown in Fig 7. Because CafRepCache performs adaptive and predictive analytics on the content, connectivity and resource levels, fRepCache 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 node*s leading to the lowest packet loss compared to the other algorithms. Fig. 8 shows performance analysis of content discovery delays between CafRepCache, two state of the art and one benchmark algorithms 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 minimizes 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

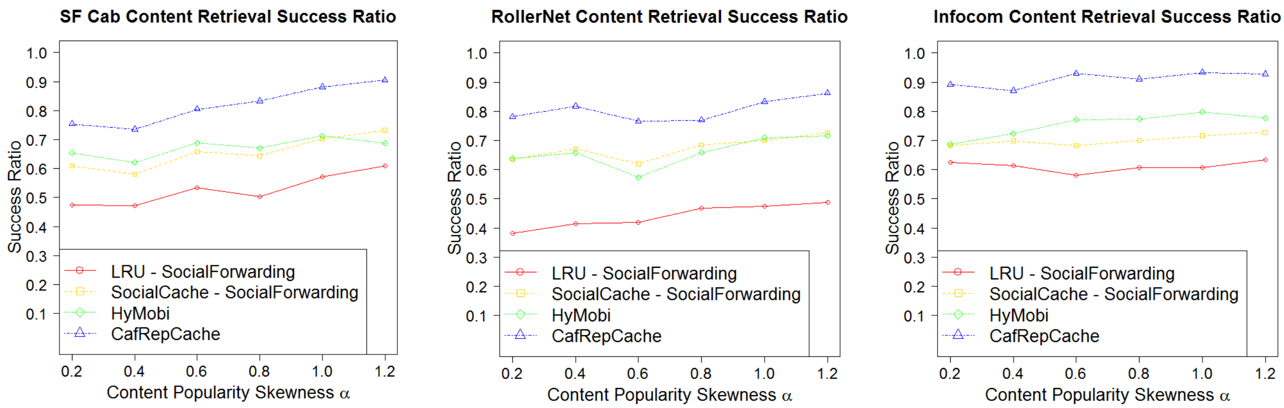


FIGURE 9. End-to-end content retrieval success ratio vs. Content popularity skewness.

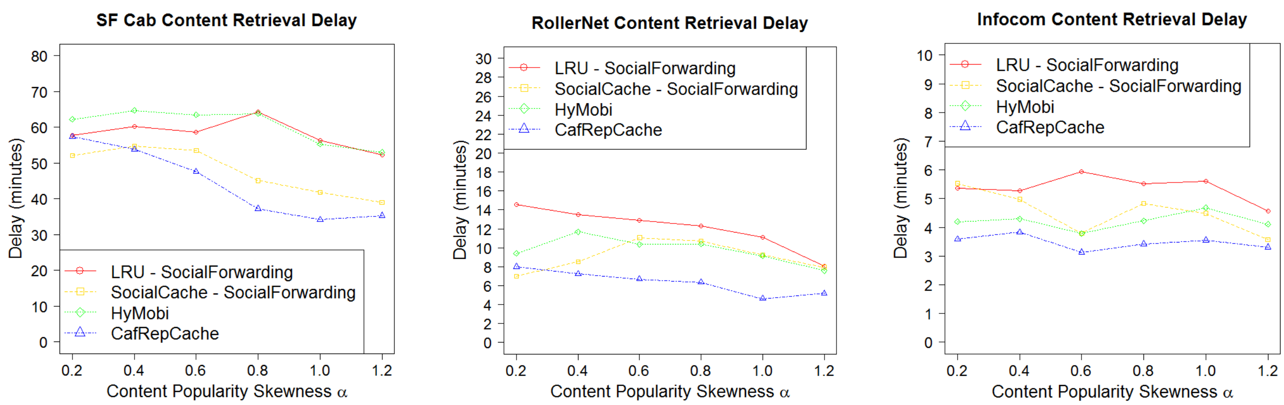


FIGURE 10. End-to-end content retrieval delay vs. Content popularity skewness.

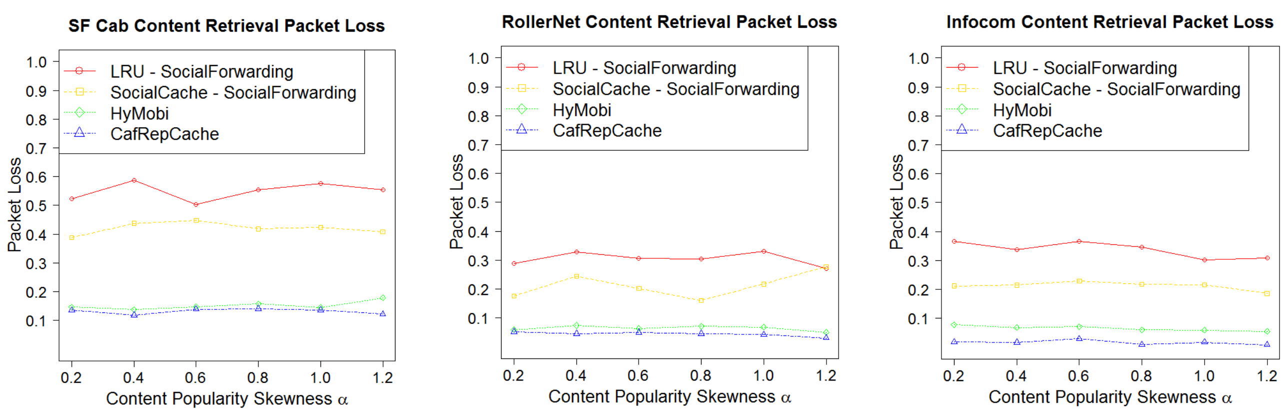


FIGURE 11. End-to-end content retrieval packet loss vs. content popularity skewness.

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. Therefore we move to analyzing content retrieval performance for the rest of the paper.

Figures 9-11 show the end-to-end success ratio, delay and packet loss for content retrieval over three traces (San Francisco, RollerNet and Infocom) and dynamically varying content popularity. We observe that CafRepCache achieves highest content retrieval success ratio (82% for San Francisco, 88% for RollerNet and 92% for Infocom), lowest delay (below 40min for San Francisco, 6min for RollerNet and 3min for Infocom) and lowest packet loss (below 13% for San Francisco, 4% for RollerNet and 1% 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 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 [4] 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.

Fig. 12 shows the importance of CafRepCache caching by comparing delays between requesting content with and without caching points. We show that caching helps to significantly reduce the delays compared to when there are no caches and interests have to traverse the network in order to be matched the content at the publishers.

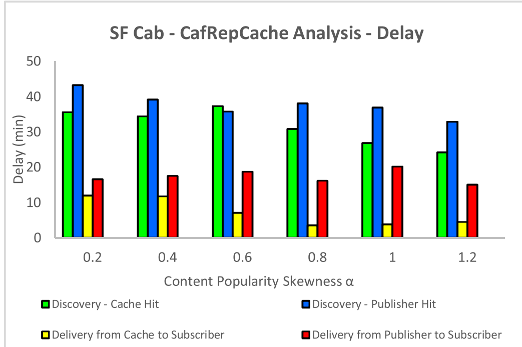


FIGURE 12. Difference in delay in two scenarios: cache hit and cache miss.

C. EVALUATION IN THE PRESENCE OF DYNAMIC PATTERNS OF SUBSCRIBERS

In order to understand the influence of variable workloads, in these experiments, we vary the patterns of subscribers requesting contents from a fixed number of publishers (e.g. 20%) and evaluate success ratio, delay and packet loss of different caching algorithms on different content retrieval processes. We assume that subscribers are mobile and not uniformly distributed as well as that they can have different connectivity patterns. Note that 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). In Figures 13-15, we 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%). This allows us to show how CafRepCache performs in a wide range of different topologies ranging from dense to sparse. We compare CafRepCache against two state of the art (HyMobi and SocialCache) and one benchmark (LRU) algorithms over multiple criteria and over three heterogeneous mobility traces.

Fig.13 shows that, in terms of content retrieval success ratios, CafRepCache outperforms the other competing algorithms SocialCache and HyMobi [9], [10] 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

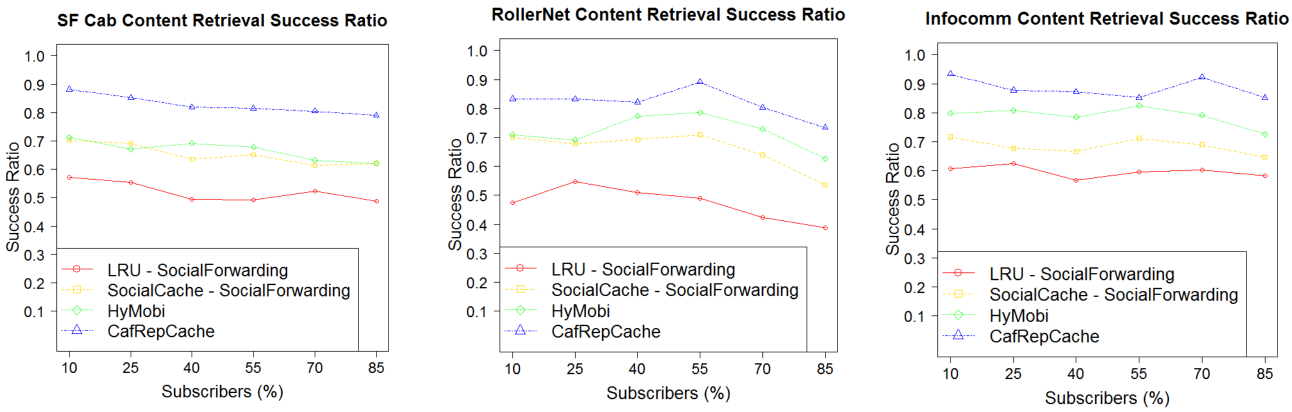


FIGURE 13. Content retrieval success ratio vs. Number of subscribers.

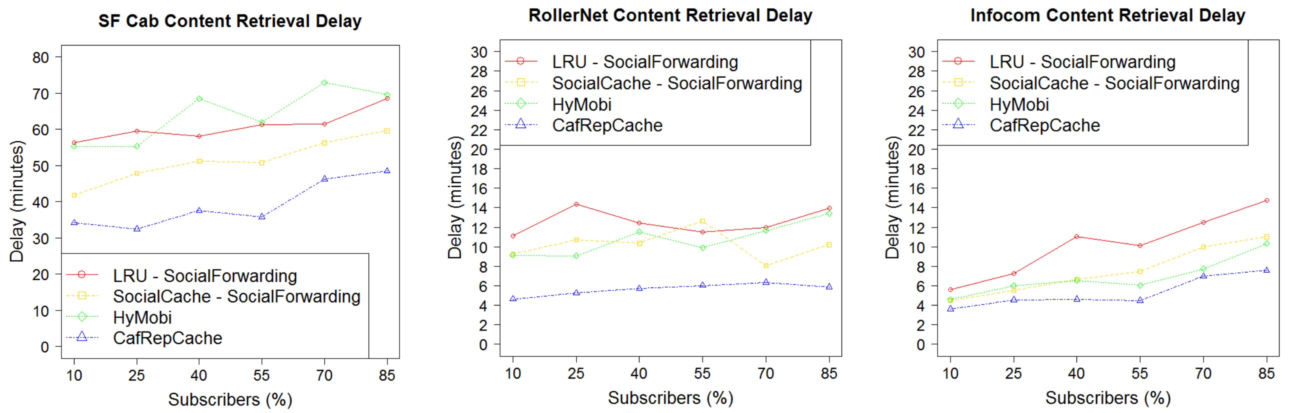


FIGURE 14. Content retrieval delay vs. Number of subscribers.

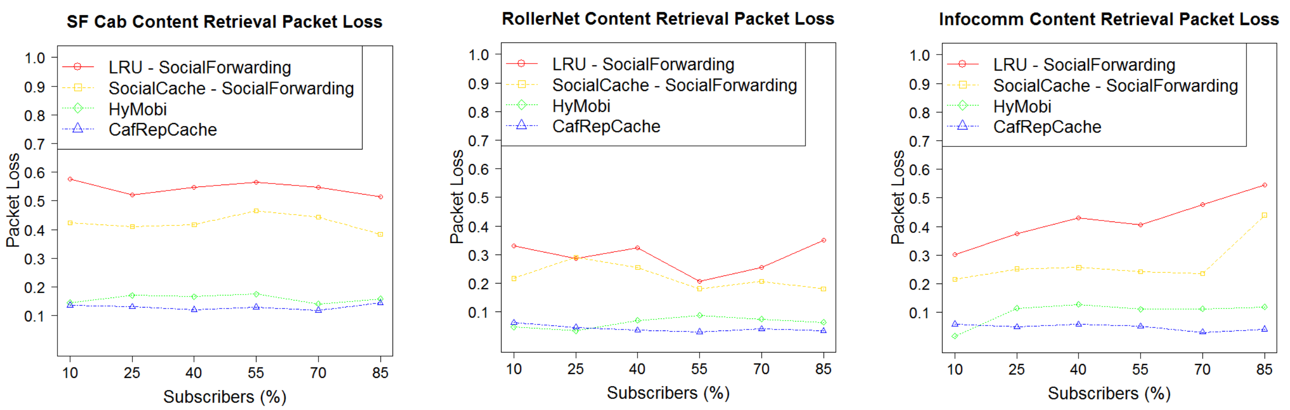


FIGURE 15. Content retrieval packet loss vs. Number of subscribers.

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.

In Fig.14 and Fig.15, CafRepCache achieves the lowest overall delay and packet loss compared to the other state of the art caching algorithms [9], [10] and LRU. We observe that the delays are slightly increased for all protocols when the diversity and number of subscribers increase but this is due to increase in subscribers with lower connectivity degree. CafRepCache reduces delays 20-30% compared with SocialCache, LRU or HyMobi. CafRepCache manages to keep packet loss to less than 10% when serving an increasing number of subscribers with the dynamic content workload. 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.

Similarly to [3], we consider caching footprint and aim to keep it a low as possible. We analyze a relative footprint reduction metric rather than footprint reduction metric proposed [3] as it is more suitable for Content Centric Networks in mobile heterogeneous environments.

$$\begin{aligned} &\textit{Relative footprint reduction} \\ &\quad = \textit{success ratio} \cdot \left(1 - \frac{\textit{hop count}}{\textit{hop count when no cache}}\right). \end{aligned}$$

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. Fig. 16 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.

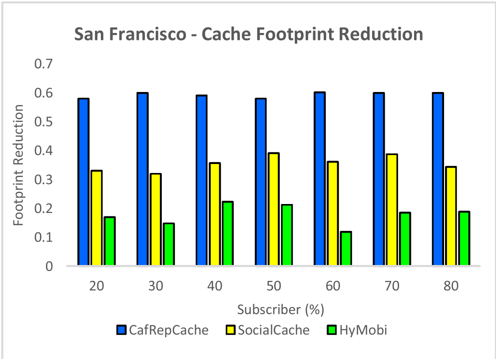


FIGURE 16. **Content Retrieval – Footprint Reduction.**

TABLE 5. **Location efficiency.**

Location Efficiency	Infocom 2006	San Francisco	RollerNet
Replication	90%	70%	84%
Caching	93%	68%	87%

In Table 5, we evaluate the efficiency of CafRepCache individual caches in terms of how much of the cached content they store is delivered to the subscribers. We show that the caching efficiency for each traces is very high: 90% (Infocom), 70% (San Francisco) and 84% (RollerNet). Similarly, we investigate individual partial replication efficiency and show in Table 5 that its efficiency is very high too across all thee traces: 93% (Infocom), 68% (San Francisco) and 67% (RollerNet) (Table 5). In this way, we show that our collaborative and adaptive cognitive caching manages to select highly suitable locations for caching and replication as well as suitable content chunks to cache and replicate when needed for very heterogeneous mobility and connectivity trances.

Figures 17 and 18 show a comparative analysis of resource consumptions of CafRepCache, SocialCache, HyMobi and

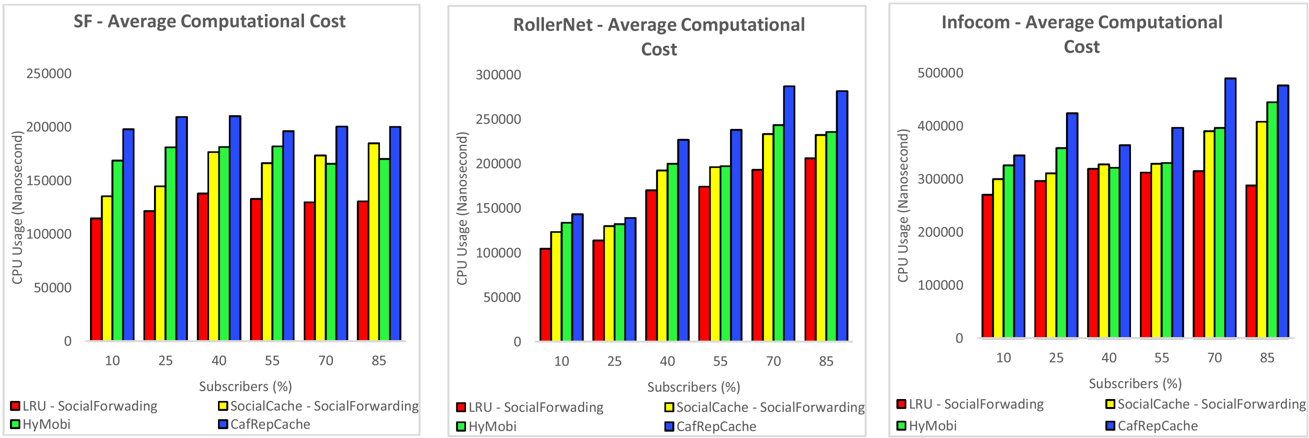


FIGURE 17. Average Computational Cost vs. Number of subscribers.

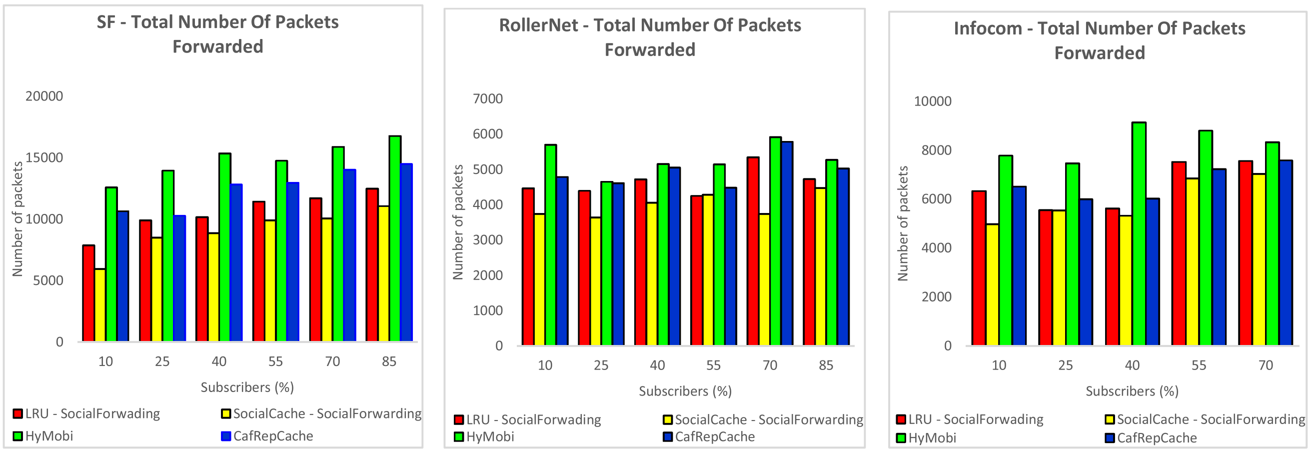


FIGURE 18. Total number of packets forwarded vs. Number of subscribers.

LRU-SocialForwarding in terms of average computational cost (CPU usage) and routing cost (total number of packets forwarded while routing). In Fig. 17, 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. Regarding the total number of packets forwarded, Fig. 18 shows that CafRepCache has up to 18.4% increase in the number of forwarded packets compared to SocialCache and LRU-SocialForwarding. Note that this increase 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.

In [44], we showed that different weights of CafRepCache's utilities may result in significantly different performances for dynamic workloads and network connectivity. Computing optimal weights that adaptively favor different utilities differently at different times is a very challenging problem even with the complete knowledge about the environment [14], thus we plan to extensively study the performance of each complementary heuristic and investigate different utility weighting models in order to understand the impact of each one on every layer of our caching framework across heterogeneous real-world mobility, connectivity traces for different workload and content popularity patterns. More specifically, we plan to evaluate and compare different content forwarding protocols in mobile disconnection prone networks, then conduct performance analysis of different content forwarding protocols built with resources awareness algorithm in order to evaluate the influences of our resources aware heuristics on the performance of content dissemination and query. After that, we plan to evaluate the performance of the forwarding protocols with resource awareness and our cognitive collaborative caching algorithms built on the content layer in order to analyze the effect of our adaptive content-aware caching protocol in the presence of dynamic content request patterns, dynamic mobility and connectivity.

Additionally, in [43], we argue that the question on how to adaptively weight and combine the value of local observation and different collaborative observations is an important and challenging problem that needs to be addressed in order to utilize efficiently the exchanged information, thus improve caching performance. We integrated ''ego network'' of each node as a dynamic network consisting of that node and contacts it meets most frequently or most recently. Thus, ego network allows each node to give its own regional and temporal perspective of the network. However, as not every node's perspective of the network may have the same importance and level of accuracy, we plan to investigate if fuzzy approaches or fully distributed collaborative reputation approaches may be helpful towards proposing new *adaptive weighting mechanisms* which can evaluate and weight different exchanged observation derived from different nodes in the network in a predictive manner and congruent with the underlying network mobility, connectivity and content interest in order to improve the overall performance of our framework. Note that we assume there are no malicious nor selfish nodes in the network in this paper, but for our future work we plan to investigate different incentive mechanisms (such as proposed in [36]) as well as to explore extending CafRepCache to adaptively utilize the exchanged information from trusted collaborators. More specifically, we plan to focus on the content-centric layer of CafRepCache and investigate complex challenges related to *content popularity weighting* process in CafRepCache suitable for heterogeneous mobile disconnection prone environments. We aim to propose a new intelligent popularity weighting mechanism that will allow CafRepCache to adapt to realistic cases where caching points gathering content popularity observed by others may not take into account equally all nodes but differentiate between them according to nodes' dynamic reputation values or fuzzy logic which will enhance the accuracy of CafRepCache predictive content analytics and improve its cache hit ratio.

VI. CONCLUSION

This paper proposed CafRepCache, a multi-layer predictive caching framework that combines real-time adaptive multi-path content and interest forwarding, with adaptive cognitive collaborative caching. We showed that CafRepCache significantly improves the performance of content discovery and retrieval over very different time-varying real-world network topologies and mobility patterns for wide range of dynamic changing workload of content publishers and subscribers against the three competing protocols across a range of metrics. CafRepCache utilizes *both single node and ego networks* multi-layer real-time predictive heuristics to manage complex dynamic trade-offs between dynamically changing topology, dynamic resources and varying content popularity and interest in order to achieve high cache availability, cache efficiency and success ratio while keeping low delays, packet loss rates and cache footprint.

In our future work, we plan to investigate energy efficient data sharing approaches that will make CafRepCache smart data dissemination and query more usable, reliable and scalable in opportunistic disconnection tolerant networks. In partnership with Nottingham City Council, we will build and deploy a real-world testbed which integrates CafRepCache with a lightweight publish/subscribe messaging transport protocol designed for connecting constrained devices, people, vehicles and infrastructure to contribute towards integrative smart city planning. We envisage that CafRepCache will be an integral part of a more robust and reliable network support that allows the scalable operation of rich mobile services with and without the infrastructure.

REFERENCES

[1] J. F. Nash, Jr., ''The bargaining problem,'' *Econometrica*, vol. 18, no. 2, pp. 155–162, 1950.

[2] L. Wang, G. Tyson, J. Kangasharju, and J. Crowcroft, ''Milking the cache cow with fairness in mind,'' *ACM/IEEE Trans. Netw.*, vol. 25, no. 5, pp. 2686–2700, Oct. 2017.

[3] L. Wang, J. Kangasharju, and J. Crowcroft. (2016). ''You really need a good ruler to measure caching performance in information-centric networks.'' [Online]. Available: https://arxiv.org/abs/1603.05963

[4] M. Radenkovic and V. S. H. Huynh, ''Collaborative cognitive content dissemination and query in heterogeneous mobile opportunistic networks,'' in *Proc. 3rd Workshop Experiences Des. Implement. Smart Objects*, 2017, pp. 7–12.

[5] D. Grewe, M. Wagner, M. Arumaithurai, I. Psaras, and D. Kutscher, ''Information-centric mobile edge computing for connected vehicle environments: Challenges and research directions,'' in *Proc. Workshop Mobile Edge Commun. (MECOMM)*, Los Angeles, CA, USA, 2017, pp. 7–12.

[6] C. Fang, F. R. Yu, T. Huang, J. Liu, and J. Liu, ''A survey of green information-centric networking: Research issues and challenges,'' *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1455–1472, 3rd Quart., 2015.

[7] J. Augé, G. Caroglio, G. Grassi, L. Muscariello, G. Pau, and X. Zeng, ''MAP-Me: Managing anchor-less producer mobility in information-centric networks,'' *CoRR*, vol. abs/1611.06785, 2016.

[8] V. S. H. Huynh and M. Radenkovic, ''A novel cross-layer framework for large scale emergency communications,'' in *Proc. 13th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Valencia, Spain, Jun. 2017, pp. 2152–2157.

[9] T. Le, Y. Lu, and M. Gerla, ''Social caching and content retrieval in disruption tolerant networks (DTNs),'' in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2015, pp. 905–910.

[10] H. Flores *et al.*, ''Social-aware hybrid mobile offloading,'' *Pervasive Mobile Comput.*, vol. 36, pp. 25–43, Apr. 2017.

[11] S. Bayhan, E. Hyytia, J. Kangasharju, and J. Ott. ''Search in digital pockets: Retrieving remote content in mobile opportunistic networks,'' *IEEE Wireless Commun.*, vol. 23, no. 3, pp. 10–18, Jun. 2016.

[12] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[13] D. Bertsimas, V. F. Farias, and N. Trichakis, ''The price of fairness,'' *Oper. Res.*, vol. 59, no. 1, pp. 17–31, Feb. 2011.

[14] M. Radenkovic and A. Grundy, ''Efficient and adaptive congestion control for heterogeneous delay-tolerant networks,'' *Ad Hoc Netw.*, vol. 10, no. 7, pp. 1322–1345, 2012.

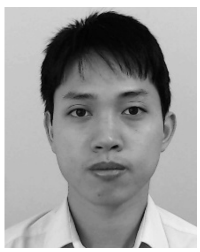
[15] M. V. Radenkovic and B. Wietrzyk, ''Wireless mobile ad-hoc sensor networks for very large scale cattle monitoring,'' in *Proc. ASWN*, 2006, pp. 1–10.

[16] M. Radenkovic, ''Cognitive privacy for personal clouds,'' *Mobile Inf. Syst.*, vol. 2016, 2016, Art. no. 7107103. [Online]. Available: https://doi.org/10.1155/2016/7107103

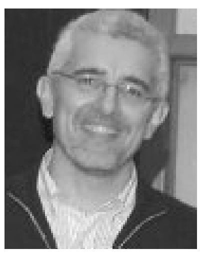
[17] A. Keränen, J. Ott, and T. Kärkkäinen, ''The ONE simulator for DTN protocol evaluation,'' in *Proc. 2nd Int. Conf. Simulation Tools Techn.*, Rome, Italy, 2009, Art. no. 55.

[18] E. M. Daly and M. Haahr, ''Social network analysis for routing in disconnected delay-tolerant MANETs,'' in *Proc. ACM MobiHoc*, 2007, pp. 32–40.
[19] S. Dernbach, N. Taft, J. Kurose, U. Weinsberg, C. Diot, and A. Ashkan, ''Cache content-selection policies for streaming video services,'' in *Proc. INFOCOM* Apr. 2016, pp. 1–9.
[20] D. Frey, M. Goessens, and A.-M. Kermarrec, ''Behave: Behavioral cache for Web content,'' in *Proc. DAIS*, 2014, pp. 89–103.
[21] (Sep. 2011). *Cab Mobility Traces, Exploratorium—The Museum of Science, the Cabspotting Project*. [Online]. Available: http://cabspotting.org/
[22] S. Saha, A. Lukyanenko, and A. Ylä-Jääski, ''Cooperative caching through routing control in information-centric networks,'' in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 100–104.
[23] B. Wietrzyk, M. Radenkovic, and I. Kostadinov, ''Practical MANETs for pervasive cattle monitoring,'' in *Proc. ICN*, Apr. 2008, pp. 14–23.
[24] M. Dehghan, L. Massoulie, D. Towsley, D. Menasche, and Y. C. Tay, ''A utility optimization approach to network cache design,'' in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.
[25] G. Carofiglio, L. Mekinda, and L. Muscariello, ''Joint forwarding and caching with latency awareness in information-centric networking,'' *Comput. Netw.*, vol. 110, pp. 133–153, Dec. 2016.
[26] B.-G. Chun *et al.*, ''Selfish caching in distributed systems: A game-theoretic analysis,'' in *Proc. 23rd Annu. ACM Symp. Princ. Distrib. Comput. (PODC)*, New York, NY, USA, 2004, pp. 21–30. [Online]. Available: http://doi.acm.org/10.1145/101>1767.1011771
[27] P. Key, L. Massoulié, and D. Towsley, ''Path selection and multipath congestion control,'' *Commun. ACM*, vol. 54, no. 1, pp. 109–116, Jan. 2011. [Online]. Available: http://doi.acm.org/10.1145/1866739.1866762
[28] A. Dabirmoghaddam, M. M. Barijough, and J. J. Garcia-Luna-Aceves, ''Understanding optimal caching and opportunistic caching at 'the edge' of information-centric networks,'' in *Proc. ICN*, New York, NY, USA, 2014, pp. 47–56. [Online]. Available: http://doi.acm.org/10.1145/2660129.2660143
[29] F. Benbadis and J. Leguay, ''CRAWDAD dataset UPMC/rollernet (v. 2009-02-02),'' Feb. 2009. [Online]. Available: https://crawdad.org/upmc/rollernet/20090202, doi: 10.15783/C7ZK53.
[30] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, ''CRAWDAD data set Cambridge/haggle (v. 2009-05-29),'' Tech. Rep., May 2009.
[31] Y. Lu, A. Misra, W. Sun, and H. Wu, ''Smartphone sensing meets transport data: A collaborative framework for transportation service analytics,'' *IEEE Trans. Mobile Comput.*, vol. 17, no. 4, pp. 945–960, Apr. 2018.
[32] L. Vigneri, S. Pecoraro, T. Spyropoulos, and C. Barakat, ''Per-chunk caching for video streaming from a vehicular cloud,'' in *Proc. 12th Workshop Challenged Netw.*, Oct. 2017, pp. 57–62.
[33] W.-Y. Kim, and S. Kim, ''The implementation of the smart festival management model using streaming service based on the complex recognition,'' *Int. J. Multimedia Ubiquitous Eng.*, vol. 8, no. 2, pp. 57–68, 2013.
[34] B. C. Lim, S. G. Kim, and B. C. Lee, ''Requirements analysis for smart festival management system,'' in *Proc. 37th Conf. KIPS*, 2012, vol. 19, no. 1, pp. 1004–1006.
[35] H. Zhou, H. Wang, X. Li, and V. C. M. Leung, ''A survey on mobile data offloading technologies,'' *IEEE Access*, vol. 6, pp. 5101–5111, 2018.
[36] H. Zhou, J. Wu, H. Zhao, S. Tang, C. Chen, and J. Chen, ''Incentive-driven and freshness-aware content dissemination in selfish opportunistic mobile networks,'' in *Proc. IEEE 10th Int. Conf. Mobile Ad-Hoc Sensor Syst.*, Oct. 2013, pp. 333–341.
[37] H. Zhou, C. M. V. Leung, C. Zhu, S. Xu, and J. Fan, ''Predicting temporal social contact patterns for data forwarding in opportunistic mobile networks,'' *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10372–10383, Nov. 2017.
[38] P. Costa, C. Mascolo, M. Musolesi, and G. P. Picco, ''Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks,'' *IEEE J. Sel. Areas Commun.*, vol. 26, no. 5, pp. 748–760, Jun. 2008.
[39] M. Radenkovic and N. Milic-Frayling, ''Demo: RasPiPCloud: A lightweight mobile personal cloud,'' in *Proc. 10th ACM MobiCom Workshop Challenged Netw.*, 2015, vol. 10, no. 1, pp. 57–58.
[40] M. Radenkovic and V. S. H. Huynh, ''Low-cost mobile personal clouds,'' in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Sep. 2016, pp. 836–841.
[41] L. Vigneri, T. Spyropoulos, and C. Barakat, ''Streaming content from a vehicular cloud,'' in *Proc. 11th ACM Workshop Challenged Netw. (CHANTS)*, New York, NY, USA, 2016, pp. 39–44.
[42] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, ''Urban computing with taxicabs,'' in *Proc. 13th Int. Conf. Ubiquitous Comput. (UbiComp)*, New York, NY, USA, 2011, pp. 89–98.
[43] V. S. H. Huynh, M. Radenkovic, and R. John, ''Understanding information centric layer of adaptive collaborative caching framework in mobile disconnection-prone networks,'' in *Proc. 14th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, 2018, pp. 25–29.
[44] V. S. H. Huynh and M. Radenkovic, ''Understanding complementary multi-layer collaborative heuristics for adaptive caching in heterogeneous mobile opportunistic networks,'' in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2018.
[45] M. Motani, V. Srinivasan, and P. S. Nuggehalli, ''PeopleNet: Engineering a wireless virtual social network,'' in *Proc. MobiCom*, 2005, pp. 243–257.

MILENA RADENKOVIC received the Dipl.-Ing. degree in electric and electronic engineering from the University of Nis, Serbia, and the Ph.D. degree in computer science from the University of Nottingham, U.K. She has authored over 80 papers in premium conference and journal venues. Her research interests span areas of intelligent mobile and disconnection tolerant networking, complex temporal graphs, self-organized security and distributed predictive analytics with applications to autonomous vehicles, mobile social networks, smart manufacturing, and predictive telemetry. She was a recipient of multiple EPSRC and EU grants for her research. She has organized and chaired multiple ACM and IEEE conferences and served on many program committees. She is an Editor of premium journals such as the *Ad Hoc Networks* (Elsevier), the IEEE Transactions on Parallel and Distributed Computing, and *ACM Multimedia*.



VU SAN HA HUYNH received the B.Sc. degree in computer science from the University of Nottingham in 2016. He is currently a Ph.D. Researcher with the Computer Science School, University of Nottingham. He has authored in venues including ACM MobiCom and IEEE IWCMC. His research interests include opportunistic, self-organizing, distributed, mobile networking and systems, Internet of Things, and network science.



PIETRO MANZONI received the M.S. degree in computer science from the Universita degli Studi of Milan, Italy, in 1989, and the Ph.D. degree in computer science from the Polytechnic University of Milan, Italy, in 1995. He is currently a Full Professor with the Department of Computer Engineering, UPV, Spain. His research activity is related to wireless networks protocol design, modeling, and implementation.