AN ABSTRACT OF THE THESIS OF

Haya Alorayj for the degree of Master of Science in Computer Science presented on March 15, 2019.

Title: Collaborative Filtering-Based In-Network Content Placement and Caching for 5G Networks

Abstract approved: ______________________________

Bechir Hamdaoui

As the number of wireless devices, the demand for high data rates, and the need for always-on connectivity are growing and becoming more stringent with the evolvement and emergence of 5G systems, network engineers and researchers are being faced with new unique challenges that need to be addressed. Among many challenges, traffic congestion bottleneck at back-haul links arising from the massive connections emerges as one key challenge that 5G systems need to tackle. One solution approach that has been investigated as a key enabler for addressing such traffic bottlenecks is in-network content caching, where frequently-accessed content is placed closer to end users at the network edges so that the amounts of traffic that need to traverse core network and back-haul links are reduced. In this thesis, we propose a content placement and caching technique that leverages collaborative filtering and k-means clustering to make efficient content placement decisions, thereby reducing downloading time and back-haul traffic. We simulate the proposed technique and compare it with two other existing caching techniques, and show that the proposed approach outperforms existing ones by achieving higher hit ratios, reducing backhaul traffic, and decreasing download times. We therefore show that the proposed technique improves the users' quality of experience by minimizing network latency and the overall network performance by alleviating backhaul traffic congestions.

Collaborative Filtering-Based In-Network Content Placement and Caching for 5G Networks

by

Haya Alorayj

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented March 15, 2019
Commencement June 2019

Master of Science thesis of Haya Alorayj presented on March 15, 2019.

APPROVED:

Major Professor, representing Computer Science

Head of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Haya Alorayj, Author

ACKNOWLEDGEMENTS

In the beginning, I thank Allah for all his graces. I would also thank my family, especially my mother who offers unconditional love and prayers through all my life. I am gratefully for my husband support and encouragement all the way through. I would also thank my supportive friends, and my colleagues who offered their consultation and feedback. Finally, I would like to thank my academic advisor: Professor Bechir Hamdaoui, for his guidance

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

Chapter 1: Introduction

As the number of wireless users and the data rates of wireless applications increase, the task of providing enough network resources that can cope with these demands becomes challenging. Particularly, downloading (e.g., multimedia) content from different Internet locations exposes networks to traffic congestion bottlenecks, consequently, causing delay, waste of bandwidth, and even network failure. There are various approaches successfully implemented in literature to overcome these defects. For example, using distributed small datacenters instead of central cloud in large zones [2, 3, 4, 5], and prefetching content in edge nodes are promising solutions for alleviating and controlling back-haul congestions [6, 7, 8, 9, 10, 11]. Large streaming services, online businesses, and websites are seizing the opportunity provided by the concept of Collaborative Filtering (CF), aka recommendation systems, to improve performance. CF refers to utilizing the history of users' behavior for a pattern to predict their future behaviors. CF is divided into two types: User-Based CF and Item-Based CF, which will be discussed further in chapter 2.

In [11], Niu et al. proposed the idea of applying the concept of CF in networking and communication. The framework contains a number of base stations (BSs), which were grouped based on their requests history. In other words, BSs with similar requests' preferences are grouped in to the same cluster, and the goal of clustering is to sort history of requests for all base stations within a cluster to find file popularity, where the number of requests for each file is used as weight. In addition to file popularity, their content placement uses User-Based CF. After finding the degree of similarity between BSs within the cluster, they calculate the probability of request to each BS for all files. Even though their approach successfully improves the hit ratio, more could be done. The CF implementation objective was mainly to increase content placement; however, CF concept can be leveraged to enable more concrete improvement to other network issues such as congestion bottlenecks. In addition, the purpose of clustering is to obtain the file popularity, meaning: first, part of caches memory for all BSs within the cluster is identical, and second, disregarding the opportunity of in-network content caching. Moreover, file prediction utilization has several vulnerabilities such as neglecting the files' features and similarities, Item-based CF. Other papers such as [12, 13, 14, 15, 13, 16], discussed the use of CF to improve the user quality of experience in websites, e-commerce, applications, media-services provider, and primary care services. However, none of them used it in networking and communication.

Despite of all its capabilities, CF concept, especially item-based CF, in networking and communication is not fully used [11].

1.1 Thesis Contributions

The thesis raises three questions, which are:

- Will the implementation of hybrid cache prefetching, file popularity and content's prediction increase users' successful attempts to find requested files in their local caches?
- Will clustering cloudlets based on their history preferences increase users' successful attempts to find requested files within the cluster?
- Will cache prefetching and in-network caching reduce network back-haul link cost?

By adapting hybrid CF approaches, this thesis aims to maximize the probability that users requested files are in the nearest datacenter (or cloudlet), which will help improve the hit ratio and decrease back-haul traffic. The focus of this work is on: first, using file popularity and item-based CF prefetching to bring content to local cache; Second, clustering cloudlets using user-based CF to enable efficient in-network caching.

The main contributions of this thesis are:

- Evaluate the performance of an existing CF approach proposed in [11], along with a base-line Zipf-distribution-based system model [17].
- Present a Collaborative Filtering for content placement and in-network caching strategy that promises to enhance the latency and network traffic congestion.
- Simulate the proposed framework and evaluate its performance.

1.2 Organization of the Thesis

The remaining of the thesis will be organized as follows. Chapter 2 provides a brief background and literature review on the subject. The system model is described in Chapter 3. In Chapter 4, we illustrate the proposed Collaborative Filtering-based content placement and in-network caching scheme. The simulation-based performance evaluation and results are presented in Chapter 4. Finally, in Chapter 5, we conclude the thesis and present some tasks for future investigation.

Chapter 2: Background and Literature Review

2.1 System Architecture and Requirements

Central cloud computing has been an important part of Internet of Things (IoT) environment by providing higher computation, control, and maintenance. However, these advantages come with a cost. With the rapid increase and growth of IoT environments and users (smart city, IoT, ..etc.), the demand of network resources is escalating causing link overloading. This means a cloud computing is risking from connection dropping, losing bandwidth, unnecessary resources consumption, latency, and most importantly, a single point of failure. As a result, edge computing approach floated to cover some of these limitations by promising to improve IoT with better latency handling, mobility, locations awareness, IoT greening, and higher streaming. This section is illustrating a literature review of how the use of edge computing (with the mobile device) can overcome three main challenges, namely connection disturbance, resources wasting, and latency, to increase the overall network performance [18, 19, 20, 21, 22, 23, 24, 3, 25].

Different systems and architectures in edge computing overcome the main common issues in cloud computing and offer more mobility, energy consumption reduction, and speediness. Next,a few related terminologies will be explained.

Mobile Edge Networks: Over the past few years, mobile devices have obtained more capacities and computation capabilities. Many researchers show their interest to make use of these mobile devices' capacities to increase the IoT performance by engaging them in tasks execution. This simple approach proved to be a very powerful tool to move the execution of the tasks closer to the edge [26].

Cloudlet: Cloudlet (or edge cloud) approach represents a smaller cloud near to the edge with all control of power and storage. Despite all its possibilities, cloudlet can only work in small or limited geographic areas [26, 2].

2.1.1 Connection Interruption

One of the main drawbacks that cloud computing faces is connection interruption. Due to its static nature, central cloud computing limits the IoT movements, especially in the smart city and Internet of Vehicles (IoV) context. Frequent connection dropping impacts the IoT environments in many aspects, including wasted bandwidth, increased delay, and increased energy consumption. However, edge computing offers a location awareness which provides more mobility by only collecting the data itself rather than the physical location [2, 3].

Since the connection dropping increases latency and wastes bandwidth, Wang et al. aimed to reduce the connection dropping via edge computing in IoV. Their approach is based on using Information Centric Network (ICN) which, unlike an IP address, is able to easily adapt to a dynamic environment and guarantee packages delivery. They used Named data Networking (NCN) as ICN implementation to reduce the delay in package catching. Unlike IP address, NCN approach simply gives the packages unique names with their source and destination. This mechanism is basically about classifying data to only keep and store the useful ones. They kept what they consider as shareable data such as car accident warnings and road congestion notifications, and discard all social interaction services [26, 27, 28].

On the other hand, some approaches such as in [2, 4] use the available mobile device resources to minimize the connections interruption and increase mobility.

Shi et al. present a system called Serendipity to handles cases where mobility is extremely important. The system contains only mobile devices, which can be either an initiator or a remote computation resource. The basic idea of the system is that mobile devices share their unused resources to overcome latency and limited resources challenges [2, 4].

Habak et al. present a system called: Femto clouds. This system goal is to benefit from the unused mobile device capacities by involving them into the network as a edge node to overcome the latency problem. The system environment contains several mobile devices connected to a control device through TCP. The control device works as a hotspot to mobile devices [4].

Lastly, to measure the effect of resource allocation scheduling strategies with mobility, Bittencourt et al. test three scheduling strategies with edge computing infrastructure to support mobility. The goal of the study is to measure and compare the performance of these three strategies based on how they handled two types of applications (delay sensitive and delay tolerant applications) [3].

2.1.2 Multimedia Access

Multimedia content access and streaming requires mechanisms that allow content to played on the fly while being downloaded [29, 30]. In addition, due to its store nature, content streaming mechanisms allow users to download and play content at any time, not necessarily live. Multimedia content represents a significant fraction of today's network traffic and the trend is only increasing with the emergence of the different wireless networking technologies (e.g., vehicular networks [31], sensor/IoT networks [32, 33, 34, 35, 36, 37]). With such an increase in network resource demands, network service providers are being faced with several key challenges crucial to fulfilling the upcoming data rate needs, so that high quality of user experience is maintained to the users over wireless networks, thereby prompting the development of new wireless technologies (e.g. MIMO [38, 39], DSA [40, 41, 42] and many others) to address challenges like shortage in spectrum supply, user mobility, and interference mitigation.

As a result, quality of user's experience for mobile content streaming has received considerable attention from the research community. For instance, [43] proposes rate adaptation and scheduling solutions for video multicasting when some users in a given region are interested in viewing the same content at the same time. [44] discusses video streaming quality when streaming over wireless networks. [45, 46, 47] study video streaming over vehicular networks. For instance, [46] proposes EUDP, which unlike UDP, uses Sub-Packet Forward Error Correction, and adopts the unequal protection of video frame types to enhance video quality.

2.1.3 Smart City

Key challenges have arisen that need to be overcome to be able to support various new smart city applications that are unique in their requirements and characteristics, in terms of numbers, amounts of needed bandwidth, network connectivity, etc. These challenges have called for the development of innovative techniques and technologies across the board, ranging from wireless access (e.g., cognitive radios [48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72], MIMO [73, 74, 39], and spectrum access [75, 76, 77, 78, 79, 79, 80]) to edge cloud computing and networking (e.g., fog computing [81, 82, 83, 25, 84, 85] and in-network caching [6, 9, 8]), with the ultimate aim of improving end-to-end latency, enhancing spectrum resource efficiency, and reducing traffic congestion. City resident populations have also been increasing at a rate of nearly 60 million per year. By 2050, it is projected that more than 2/3 of the population worldwide will be in urban cities [1]. As urban cities are getting densely populated, providing network speeds that can meet the unprecedented demands is becoming more challenging. Sending and receiving data between end-users and central cloud computing faces the threat of delay, especially with the rapidly growing number of end-users [2, 86]. Shi et al. introduce a comparison between three cloudlet-assisted edge computing architectures to measure their performance in the matter of reducing delay and latency in a real-time application [2]. Mtibaa et al. create a hyper system where the mobile device can offload tasks to traditional cloud computing, Cloudlets, or locally. The execution of the tasks depends on the tasks requirements and the available resources [86].

Therefore, it is important to leverage new emerging networking technologies such as in-network caching and edge cloud computing to enable fast and efficient access to network services to improve the overall experience of city users.

2.1.4 Energy Consumption

In central cloud computing IoT, the senses communicate with the server in order to request resources such as storage or power. The server then sends these requests to the cloud datacenter to process and send the requested resources to the senses. Going back and forth costs the network bandwidth and increases the delay. In addition, part of the energy is wasted during transmission. Using a small power source (Microgrid) close to the edge computing promise to enhance the power consumption [87, 2, 88].

Serendipity system [2] has many goals, and one of them is energy awareness. The system aiming to decrease the energy consumption since it basically relays on mobile devices which has limited battery power. The idea of the system is that mobile devices balance between accomplishing their assigned tasks and keeping their energy as long as

[1] World population data sheet: `http://www.prb.org/`

possible. In order to do so, the system establishes a utility function illustrating the energy consumption for all nodes in the range, and how much residual energy these nodes have, therefore, enhancing tasks assignment decision.

Jalali et al. propose power consuming reduction technique that combines edge computing and microgrid system. The paper describes how minimizing the need to request resources from the central grid decreases the energy consumption in IoT application. The paper conducted a literature survey comparing between cloud and edge computing energy consuming to underline the main factors which impact the energy consumption [88].

Energy consumption is the main scope in [87], where the authors addressed the problem of connection dropping in mobile ad hoc networks (MANET), which leads to wasted energy. Their proposed framework uses the fog computing principles and considers MANET with D2D connection and cloudlet as a datacenter. This framework increases MANET mobility, where end-users can move and restore their needed services instantly, at the same time, minimizing the energy spending [87].

2.2 Collaborative Filtering

Collaborative Filtering (CF) has been adapted by many large corporations such as Netflix, Amazon, YouTube etc. Anticipating what users might like and offer it before they ask for it is proven to be successful in terms of advertising, user satisfaction, and most importantly, minimizing latency [27]. CF can be defined as the technique of predicting users' behaviours based on deducing a pattern in their previous behavior using different data sources. CF relies on two methods to predict users' interest: User-Based and Item-Based Collaborative Filtering [89, 90, 91, 92, 93].

- **User-Based Collaborative Filtering(UB-CF):** In this type of CF, prediction is based on finding users with similar preferences, and the process consists mainly of grouping users with similar preference, see Table 2.1. After that, the items which are more frequently requested or highly rated by the users in that group will be suggested to the users in the same group who did not request it yet. The similarity here is observed and measured by users' responses and actions to a set of items. The rating or, in some cases, number of requests will be treated as a weight in the similarity calculations [94, 95, 96, 97, 98].

- **Item-Based Collaborative Filtering(IB-CF):** Contrary to UB-CF, Item-Based Collaborative Filtering compares between items instead of users, see Table 2.2. These items might be files that users rated or requested. The similarity in IB-CF is mainly based upon measuring how many features these items share. Item Features or attributes capture the properties of the items; for e.g, Genre could be used to capture movies' attributes. In this type, unlike UB-CF, each user will be considered individually using their own datasets [94, 95, 96, 97, 98],

  Both approaches have their strengthens and weaknesses; however, a hybrid approach of Collaborative Filtering will overcome the weakness of each type.

Similarity Measure: Various algorithms and mathematical formulations have been introduced for applying Collaborative Filtering. The choice of the method or mathematical formulations depends on the nature of the available dataset [1]. Based on [1], Table 2.3 compares different popular similarity measures by illustrating their strength and weakness, Where r_{A_i} denotes the number of requests/ratings-value user A made for item i, and I is the total number of co-rated/co-requested items.

Table 2.1: User-Based Collaborative Filtering.

Users / Files	F_1	F_2	$F_{..}$	F_i	$F_{..}$	F_I
A	..	..	..	..	..	..
B	..	..	..	..	..	..
C	..	..	..	..	..	..
D	..	..	..	..	..	..
E	..	..	..	..	..	..

Table 2.2: Item-Based Collaborative Filtering.

Users / Files	F_1	F_2	$F_{..}$	F_i	$F_{..}$	F_I
A	..	..	..	..	..	..
B	..	..	..	..	..	..
C	..	..	..	..	..	..
D	..	..	..	..	..	..
E	..	..	..	..	..	..

Table 2.3: Similarity Measure [1].

Similarity Measure	Equation	Strengths	Weaknesses
Cosine Coefficient	$sim(A,B) = \frac{\sum_{i=1}^{I} A_i B_i}{\sqrt{\sum_{i=1}^{I} A_i^2}\sqrt{\sum_{i=1}^{I} B_i^2}}$	Consider the absolute value of the index	Neglect the user preference
Pearson Correlation Coefficient	$sim(A,B) = \frac{\sum_{i=1}^{I}(A_i-\overline{A})(B_i-\overline{B})}{\sqrt{\sum_{i=1}^{I}(A_i-\overline{A})^2(B_i-\overline{B})^2}}$	Consider the user preference	product a misleading results
Mean Squared Difference	$sim(A,B) = 1 - \frac{\sum_{i\in I}(r_{A_i}-r_{B_i})^2}{\lvert I\rvert}$	Consider the absolute value of the index	Neglect the item feature “
Jaccard	$sim(A,B) = \frac{I_A \cap I_B}{I_A \cup I_B}$	Consider the item feature	Neglect the absolute value of the index 4

2.3 In-Network Caching

2.3.1 Proactive Caching

In last few years, many papers discussed prefetching contents as a way to improve network performance. Prefetching is a promising solution to improve the cache hit ratio, minimizing network traffic congestion, decreasing bandwidth consumption etc. Some approaches purpose pre-fetching popular items. For instance, in [8], Abuhadra et al. used hybrid approaches combining proactive caching and prefetching contents, where proactive caching proposes fetching the most popular contents in the local base station in a proactive fashion, and the prefetching is fetching content (e.g. video), in advance, to the next base station that users are expected to be connected to in their path. This approach is built on high mobility networks where the user connection time to one base station is very low [8, 11, 99], leading to frequent cell handovers. In [9], Sinky et al. also used hybrid approaches of popularity and cooperative content caching where the decision of caching a content is shared within the cluster cloudlets. The analysis of cloudlets states, such as the available resources, and the content popularity have been taken into consideration for the caching decision [9].

2.3.2 Clustering

Clustering refers to grouping 'similar' data points in the same cluster. Determining the correct clustering method depends on the goal of the clustering and the nature of the dataset. There are several clustering algorithms that each having its weaknesses and strengths [100, 101].

2.3.2.1 Hierarchical Clustering

Hierarchy Clustering initially assumes that each data point is a cluster by itself, then, depending on the goal of clustering, groups numbers of data points into one cluster. In this method, the number of clusters can not be controlled. Sinky et al. [9] used Hierarchical Clustering to form multiple communities of cloudlet datacenters. The paper clusters cloudlets based on distance between the cloudlets and the server. The main purpose of the cluster is to allow in-network caching where the cloudlets can access other cache content within the cluster; therefore, it maximizes the possibility of finding the content in a Neighbors cloudlet rather than fetching it from the core network, consciously, improving number of cache hits and back-haul delay [9].

2.3.2.2 Partitional Clustering

On the other hand, K-means clustering starts with a certain number of clusters and adds data points to the cluster with the closest centroid value. That means the optimal number of cluster should be determined before applying the clustering. Niu et al. in [11] established K-means clustering to find the similarity between base stations based on the number of requests to certain set of files. The goals of the paper are: find the file popularity within the cluster, and predict the BS future requests by using the history of requests of other base stations within the cluster, user-based CF [11, 102].

Chapter 3: System Model

As shown in the Figure 3.1, a three-tier network architecture covering and serving a highly dense urban area is considered in this work, where the top layer of the architecture is the core network where the content/database/library server resides. This server also keeps tracks of all access history from the various cloudlets that it serves. The middle (or distribution) layer is composed of a number of cloudlet data centers (CD), with each CD being associated with a number of end-users/devices. The set of end users constitutes the access (bottom) layer of this architecture. Throughout, we use R to denote the number of CDs, with the set of CDs being CD=$\{CD_1, CD_2, .., CD_R\}$, and N to denote the number of users, with the set of users being U=$\{U_1, U_2, .., U_N\}$, with users having diverse preferences and interests. The library/database contains a set F of Q files, F=$\{F_1, F_2, .., F_Q\}$, where all files in the library have the same size of L Mbits. We assume that each file has a set A of I attributes, A=$\{A_1, A_2, .., A_I\}$, with each attribute representing/capturing some feature. For e.g., if the files represent movies, then attributes could represent the Genre of the movies, i.e., horror, drama, comedy, etc., with each moving having an index value (between 0 and 1) for each Genre. For instance, a horror movie with lots of scary action scenes may be assigned larger index values for its Horror and Action attributes.

We now introduce two tables: CD features Table 3.1 and File features Table 3.2.

Table 3.1: CD features.

Cloudlets/Features	A_1	A_2	$A_{..}$	A_I
CD_1	..	..	..	..
CD_2	..	..	..	..
$CD_{..}$	..	..	..	..
CD_i	..	..	..	..
$CD_{..}$	..	..	..	..
CD_R	..	..	..	..

Table 3.2: File features.

Features/Files	F_1	F_2	$F_{..}$	F_j	$F_{..}$	F_Q
A_1	..	..	..	..	..	..
A_2	..	..	..	..	..	..
$A_{..}$	..	..	..	..	..	..
A_I	..	..	..	..	..	..

For each file and each CD, the sum of all values of the attributes is equal to one. That is, $\sum_{j=1}^{I} A_j = 1$ for every file and for every CD.

To obtain the number of times cloudlet CD_i requests file F_j, a history of request Table 3.3, REQ, is created, with the $i \times j$ entry being calculated as:

$$REQ_{i,j} = \frac{CD_i \cdot F_j}{I} \leq 1 \tag{3.1}$$

where CD_i represents a row in Table 3.1 and F_j represents a column in Table 3.2. Alternatively, from a practical viewpoint, REQ can also be updated through history by setting $\mathrm{REQ}_{i,j}$ to the fraction of times file j is requested by CD i among all file requests. Note that the history of requests, REQ, will be used later to decide for content placement.

Table 3.3: REQ: History of Request

CD/file	F_1	F_2	$F_{..}$	F_j	$F_{..}$	F_Q
CD_1	..	..	..	..	..	..
CD_2	..	..	..	..	..	..
$CD_{..}$	..	..	..	..	..	..
CD_i	..	..	..	..	..	..
$CD_{..}$	..	..	..	..	..	..
CD_R	..	..	..	..	..	..

We assume that each CD is equipped with a cache of size S_m with

$$S_m = Library \times \eta \tag{3.2}$$

where η is a tunable parameter representing a fraction between 0 and 1, and *Library* is the total capacity of the database. We consider that the cache size is divided into two halves: One half, $S_{rank} = S_m/2$, used for caching content based on CDs' local popularities, P for each CD.

S_{rank} contains a set of the M most popular files, $\mathrm{F_1,\ldots,F_M}$, with $\mathrm{F_1}$ being the most requested file, $\mathrm{F_2}$ being the second most requested file, and so on. The ranking of these files depends on the P value, where P_{CD_i,F_m} represents the popularity of $\mathrm{F_m}$ in cloudlet $\mathrm{CD_i}$. The value of P is obtained from the history of requests. The second half of the cache, $S_{sim} = (1 - S_{rank})$, is used for caching H files based on item-based collaborative filtering (CF), which will be further explained in Chapter 4. In these two cache spaces, cache placement is assumed to be done during the off-peak hours.

All CDs are clustered into C clusters, $\mathrm{CLS}=\{\mathrm{CLS_1,CLS_2,..\ CLS_C}\}$. How to determine the optimal number of clusters, C, as well as which clustering algorithm to use will be explained in Chapter 4.

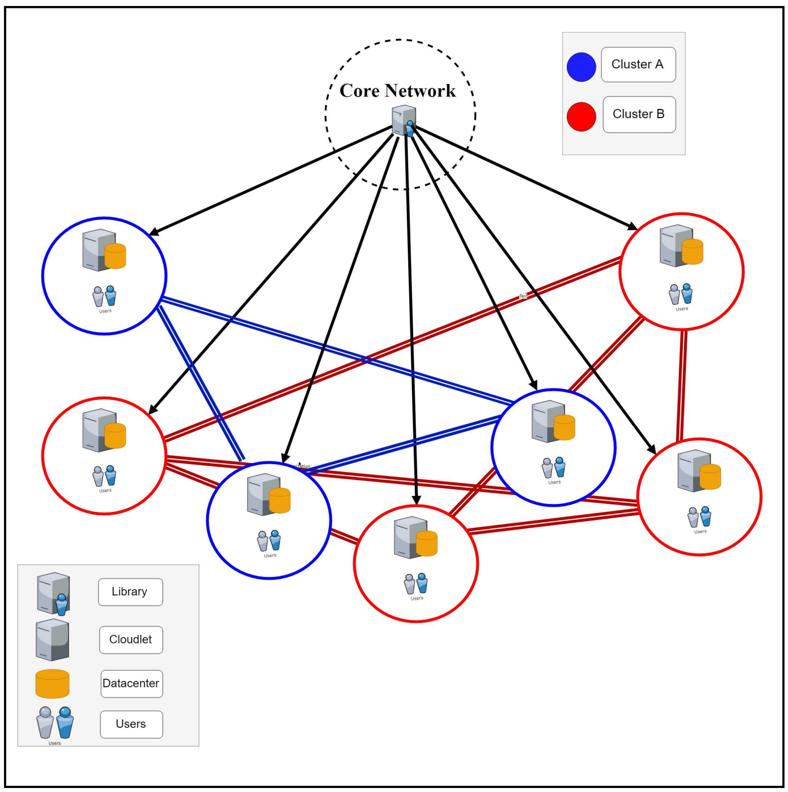

Figure 3.1: System model

Chapter 4: Hybrid Collaborative Filtering Based In-Network Content Placement and Caching

4.1 CD Clustering

Using the entry of history of request Table 3.3, REQ, whose entries are given in Eq. 3.1, K-means clustering is applied to group the cloudlets into C clusters. We use the Cosine distance metric, as defined in Eq. 4.1, to model the distance between any pair of cloudlets, CD_i and CD_j. Cloudlets within the same cluster tend to have distance values that are close to 0, whereas distance values of those dissimilar cloudlets will be close to 1. Finally, a zero value means that the cloudlet is located exactly in the middle of two clusters. In the following Eq.4.1, $REQ_{i,m}$ denotes the number of request that CD_i made for F_m.

$$dist(CD_i, CD_j) = 1 - \frac{\sum_{m=1}^{Q} REQ_{i,m} REQ_{j,m}}{\sqrt{\sum_{m=1}^{Q} (REQ_{i,m})^2} \sqrt{\sum_{m=1}^{Q} (REQ_{j,m})^2}} \tag{4.1}$$

In this section, we illustrate how the optimal number of clusters was achieved. To determine the optimal number of clusters, there are multiple existing methods, such as elbow and Silhouette methods, that can be used. In this thesis, we used Silhouette coefficient [103, 104] as the metric for deciding how well the clustering method is. The Silhouette coefficient measures how much a certain observation fits to its cluster. As the average silhouette coefficient value gets closer to one, it implies how well these data points lie within its current cluster. To obtain the optimal number of clusters, the average silhouette coefficient is calculated for every possible number of clusters, and the one with the highest average silhouette coefficient value is chosen. Figure 4.1 illustrates the different silhouette coefficients for 2 to 8 clusters, and Figure 4.2 shows the observations within the cluster [103, 104].

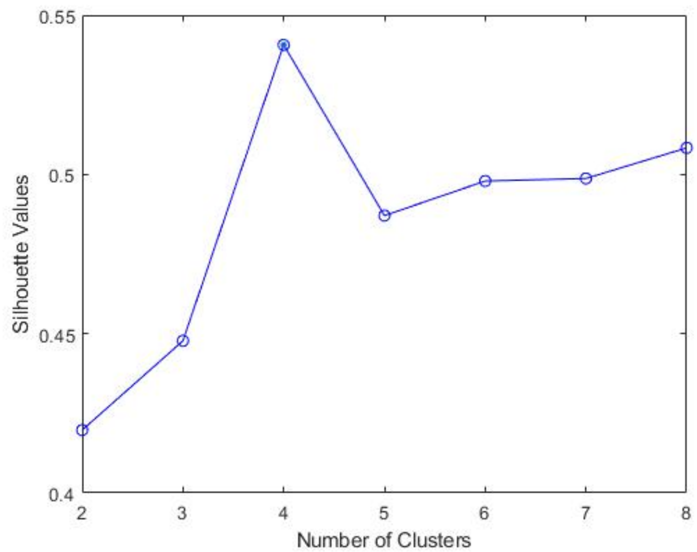

Figure 4.1: Optimal number of clusters

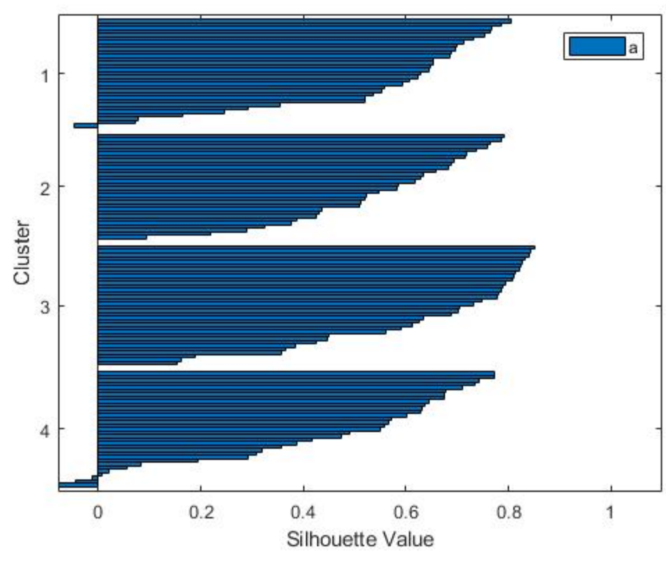

Figure 4.2: Silhouette Value

4.2 CD Caching

Each cloudlet CD_i contains its own cache space S_m. Recall that content placement in Part-I of S_m is based on the local popularity of the files. But part-II of S_m contains the files with the highest probability to be requested in the future. In this part, item-based CF is implemented and works as follows. First, using Cosine Coefficient as expressed in Eq. 4.2, a similarity index between any pair of files, F_i and F_j, is calculated [94].

$$sim(F_i, F_j) = \frac{F_i.F_j}{||F_i||.||F_j||} = \frac{\sum_{a=1}^{I} F_{a,i}F_{a,j}}{\sqrt{\sum_{a=1}^{I}(F_{a,i})^2}\sqrt{\sum_{a=1}^{I}(F_{a,j})^2}} \tag{4.2}$$

where again F_i and F_j are the ith and jth columns File feature Table 3.2, and a is the ath feature of I total number of features. Unlike Cosine distance metric, Cosine Coefficient has the value 1 when the two files are identical, and the value zero when they are not sharing any properties. As shown in Table 4.1, we assume that the similarity index between the same file is zero; that is, $sim(F_i, F_i) = 0$ for all i.

Table 4.1: File Similarity.

Similarity	F_1	F_2	$F_{..}$	F_j	F_i	$F_{..}$	F_Q
F_1	0	..	..	..	..	..	..
F_2	..	0	..	..	..	..	..
$F_{..}$	..	..	0	..	..	..	..
F_j	..	..	..	0	..	..	..
F_i	..	..	..	..	0	..	..
$F_{..}$	..	..	..	..	..	0	..
F_Q	..	..	..	..	..	..	0

Next, using Table3.3 and Table4.1, we construct a file prediction Table 4.2, PRE, whose entry $PRE_{i,j}$ is calculated as [94, 95]:

$$PRE_{i,j} = \sum_{k=1,k\neq j}^{Q'} sim(F_k, F_j)REQ_{i,k} \tag{4.3}$$

where Q' represents the set of all the files requested by CD_i. Here, the higher the value of $PRE_{i,j}$, the more likely that CD i will request file j in the future. We want to mention that in our simulation evaluation presented in Chapter 5, the prediction value provided by Eq. 4.3 was normalized by dividing it by the sum of the similarities between the considered file and all the other files.

Table 4.2: PRE

CD/file	F_1	F_2	$F_{..}$	F_j	$F_{..}$	F_Q
CD_1	..	..	..	..	..	..
CD_2	..	..	..	..	..	..
$CD_{..}$	..	..	..	..	..	..
CD_i	..	..	..	..	..	..
$CD_{..}$	..	..	..	..	..	..
CD_R	..	..	..	..	..	..

4.3 File Downloading Scheme

Each cloudlet CD_i will have its own cache content, and we assume that the number of cloudlets is large enough to cover the residential area, and the communication time between the users and the cloudlet is large enough to deliver the requested contents to the users. Assume that each user U_n will be associated with one cloudlet CD_i, which is typically the nearest one to it. Now considering that U_n (associated with CD_i) is requesting file F_x, first, CD_i will be checked whether it contains F_x in its local cache. If F_x is available, it will be delivered to U_n. Otherwise, CD_i will search for the file within all CDs belonging to its cluster, CLS_D. If F_x is founded within the cluster, it will be delivered to U_n without fetching it to the local cache. Otherwise, if F_x is neither in CD_i nor or in any CD in CLS_D, the file will be requested and downloaded from the core network library, and the fetched file will replace the least requested file in CD_i's local cache. The searching algorithm is illustrated in Algorithm 1.

In this work, the likelihood of requesting files by users generated based on the history of request Table 3.3. Each user is assumed to stay connected to the same cloudlet for a duration that is long enough to allow full content delivery [105, 95].

Algorithm 1 Searching for F_x

Input: CD_i, x
Output: $\mathbf{F}_x$

```
if CD_i request F_x then
  (S_rank, S_sim) ⇐ Check
  if Check ≠ 1 then
    (Cls_z) ⇐ Check, CD_i ∈ CLS_D
    if Check ≠ 1 then
      CN_Library ⇐ Check
      return F_x
    else
      return F_x
    end if
  else
    return F_x
  end if
end if
```

Chapter 5: Performance Evaluation

5.1 Simulation Setup and Methodology

In this section, we evaluate the proposed Collaborative filtering-based in-network content placement and caching by comparing it to the existing CF and the base-line approaches. We used MATLAB for the simulation. The system contains different numbers of cloudlet with different cache capacities. It has 2000 users in low mobility network, where we assume to have connection time that is long enough to transmit one file before the users change their locations. We also assume that all cloudlets are connected to one server where it is located outside the area and the distance between the server and all cloudlets is 100 kilometers, and the searching time within the cluster is neglectable. The core network contains a library with 10000 files, which will be cached based on a certain method explained in Chapter 4. It also contains the history of requests for all cloudlets. The system parameters are described in Table 5.1.

Table 5.1: System Parameters

Parameter	Value
Number of users	2000
Number of cloudlets	16;32;64;128;256
Number of files	10000
Number of attributes	4
Local cache size	20%,30%,40%,50%
Back-haul link capacity	10 Gbit/s
Front-haul link capacity	1 Gbit/s
File size	30 blocks= 15 Gbits/s
Block size	0.5 Gbit/s
Number of Clusters	4

5.2 Performance Metrics

In order to evaluate the performance, we investigate how much the proposed approach impacts the network performance in three different aspects: Hit ratio, overall delay and Back-haul congestion. The three performance metrics are described as follows:

1. Hit Ratio (HR): In this thesis, three different hit ratios are calculated: (1) In-Cache hit ratio: calculates number of times a requested file is successfully found in the local cloudlet cache. (2) In-Cluster hit ratio: number of times a requested file is successfully found within the cluster. (3) Overall Hit ratio: the summation of In-Cache and In-Cluster hit ratio [105].

$$Hit\,ratio = \frac{Number\,of\,hits}{Total\,number\,of\,requests} \tag{5.1}$$

   The improvement will be measured as:

$$Total_{HR}\,Improvement = \frac{Existing\,approach_{HR} - Proposed\,approach_{HR}}{max\{Proposed\,approach_{HR}, Existing\,approach_{HR}\}} \tag{5.2}$$

   The improvement value ranges from -1 to 1, where value closer to one means positive improvement. A zero value means no improvement at all, and as the number gets lower than zero, it means negative performance.

2. Back-haul network congestion (BHC): To avoid a bottleneck, the back-haul link usage must be minimized. In this thesis, we are interested in doing that by prefetching content in the local cache. To measure how successful is the proposed approach in avoiding a bottleneck, we will measure the Back-haul delay. The Back-haul delay is calculated every time the a requested file is not found locally, in local cache or within a cluster, and fetched from the core network library. This will allow us to see the impact of the proposed approach on network traffic and contribution to the bottleneck problem.

$$Total_{BHC}\,Improvement = \frac{Existing\,approach_{BHC} - Proposed\,approach_{BHC}}{max(Proposed\,approach_{BHC}, Existing\,approach_{BHC})} \tag{5.3}$$

Again, the value of improvement will range from -1 to 1 Eq.5.2.

3. Total delay (Delay): The total delay measures how long it takes from initiating a request until the requested data is delivered. This delay includes the time used to find out whether the file is found locally, within cluster or fetched from the core network using Back-haul link. So:

$$TotalDelay = Fronthauldelay + Backhauldelay \tag{5.4}$$

Front-haul delay calculates the time required from requesting a file to check whether it is available locally, in the local cache or within the cluster. If the file is found locally, the Back-haul delay will be equal to 0, otherwise, the Back-haul delay will be calculated accordingly.

The improvement will be measured:

$$Total_{Delay}\, Improvement = \frac{Existing\ approach_{Delay} - Proposed\ approach_{Delay}}{max Proposed\ approach_{Delay}, Existing\ approach_{Delay}} \tag{5.5}$$

Again, the value of improvement will range from -1 to 1 Eq.5.2.

5.3 Simulation Results

We compared the performance of the proposed approach with other current approaches. For the base-line approach, the clustering is based on the physical distance between the cloudlets, where the cloudlets that are near to each other are most probably to be grouped in the same cluster. Content placement in the local cache follows zipf distribution, which means all cloudlets have the same content. The second exiting approach is the CF-based approach [11], in which clustering the cloudlets is based on their history of requests, REQ. Within each cluster, the history of requests will be in descendant order. The content with the highest number of requests will be at the top of the rank. In this approach, cloudlets within a cluster will have identical content in the first part of the cache. The second part will contain the files with highest prediction values, user-based CF. The file prediction value will be measured using similarity coefficients, between cloudlets [11], and the decision of caching is dependent on the value of P; for more details, see [11]. In both systems, K-means clustering is applied.

The simulation yielded very positive outcomes. The results of each performance metric will be presented and analysed separately.

5.3.1 Hit ratio results

In case of overall hit ratio, the following three Figures 5.1,5.2, and 5.3 illustrate that the proposed approach outperforms the other approaches across different numbers of cloudlets. It can be seen clearly that the increment of cloudlets number has a positive impact on the proposed and the existing CF approaches. However, with different cache sizes, the overall hit ratio for all approaches is affected positively.

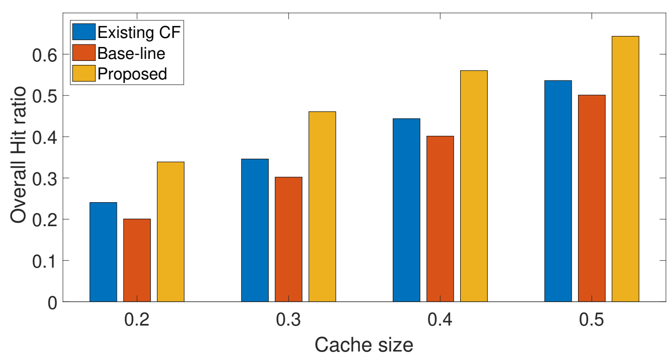

Figure 5.1: Overall hit ratio with 32 cloudlets

From Figure 5.1, for all different cache size values, the proposed CF is higher than both the base-line and the existing CF approaches. A steady progression is observed. Through all cache capacities, the hit ratio increases in the same rate for all approaches.

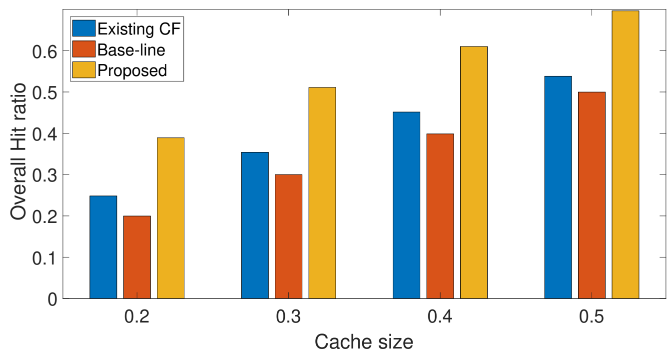

Figure 5.2: Overall hit ratio with 128 cloudlets

When comparing the 32 cloudlet scenario with the 128 cloudlets, the proposed CF approach accomplishes better hit ratio by expanding the performance gap. Likewise, the progress through the cache capacities is stable.

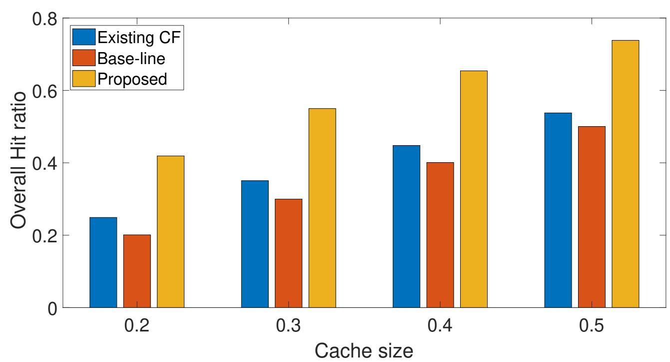

Figure 5.3: Overall hit ratio with 256 cloudlets

Similar to the 32 and the 128 cloudlets, Figure 5.3 shows that the 256 cloudlets keeps a steady distance with the other approaches across all cache sizes. However, the performance gap is larger between the proposed approach and the other approaches. That means the increment of the numbers of cloudlet increases the overall hit ratio. For all the figures, the base-line hit ratio is not affected by the number of cloudlets, since cloudlets content in this approach is identical, thereby eliminating the possibility of the In-Cluster caching. Another observation is the hit ratio improvement between the base-line and the existing CF approach is narrow, especially with 50% cache size.

Next, the Figures, 5.4 and 5.5, illustrate the overall hit ratio gain over the current existing approaches with 20% and 50% cache size.

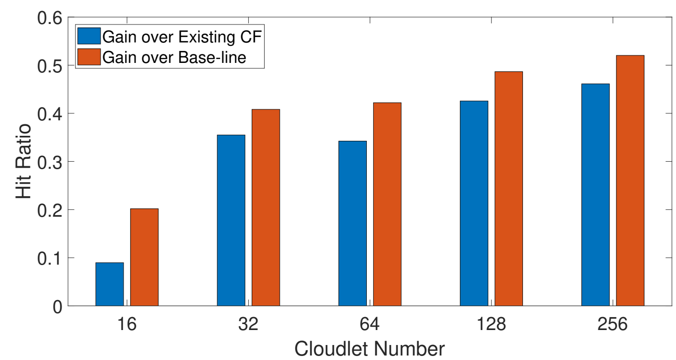

Figure 5.4: Overall Hit ratio gain for 20% cache size

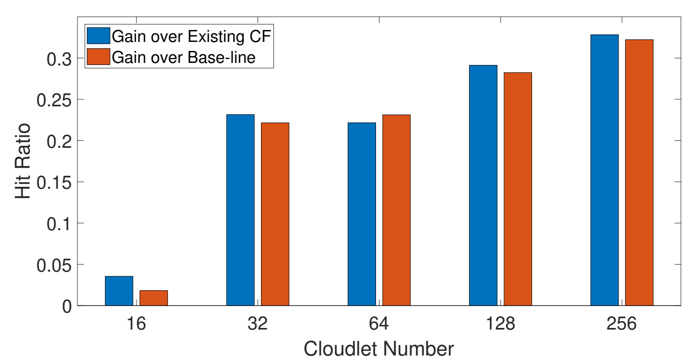

Figure 5.5: Overall Hit ratio gain for 50% cache size

Generally, the increment of the numbers of cloudlets affects the hit ratio gain positively. In the Figure 5.4, the gain over the base-line approach to the proposed one is noted. On the other side, the gain over the existing CF is less compared to the base-line approach. However, with the 32 and the 64 cloudlets, the gain over both approaches is almost the same. In the same way, the number of cloudlets has a beneficial impact. We can see that with the 64 cloudlets, the gain from the existing CF approach declined slightly. With the 256 cloudlets, the proposed approach achieved the highest gain with 50% improvement from the base-line approach and 45% from the existing CF approach. The

Figure 5.5 shows a surprisingly higher gain over the existing CF compared to the base-line approach, except with the 64 cloudlets. This indicates that increment of the cache capacity has negative influence on the existing CF approach. Thus, the 256 cloudlets is the best numbers of cloudlet for our proposed CF approach, and 20% cache size had the highest gain in hit ratio.

To answer the first and second research questions, the In-Cache and the In-Cluster hit ratio results will be presented next:

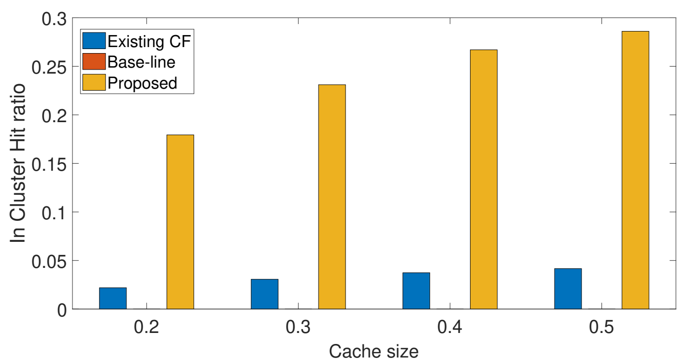

Figure 5.6: In-Cluster Hit ratio

The Figure 5.6 proves that the proposed in-network caching is successfully implemented. As expected, the base-line has 0% In-Cluster hit ratio, due to the uniform cache content. Moreover, the existing CF hit ratio is less than 20% compared to the proposed content placement method, since within the cluster cloudlets, in the existing CF approach, share 50% identical content.

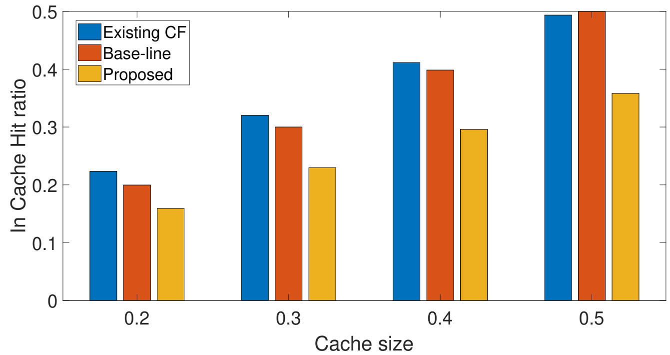

Figure 5.7: In-Cache Hit ratio

Figure 5.7 illustrates that our approach achieves lesser In-Cache hit ratios compared to the other two cases. Through all cache capacities, the existing CF outperforms the remaining approaches, yet with 50% cache space, the base-line approach has the highest hit ratio.

5.3.2 Back-haul results

The impact of CF-based content placement and in-network caching on back-haul congestion is illustrated next. The Figures 5.8, 5.9, and 5.10 display how the number of cloudlets reflects on the back-haul delay congestion:

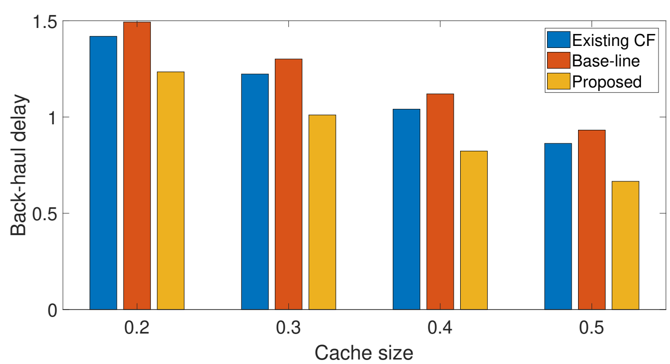

Figure 5.8: Back-haul delay with 32 cloudlets

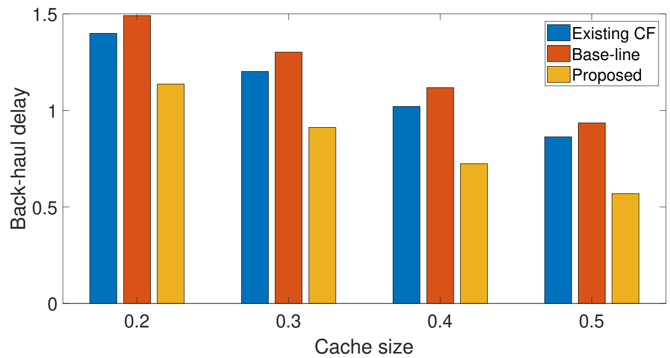

Figure 5.9: Back-haul delay with 128 cloudlets

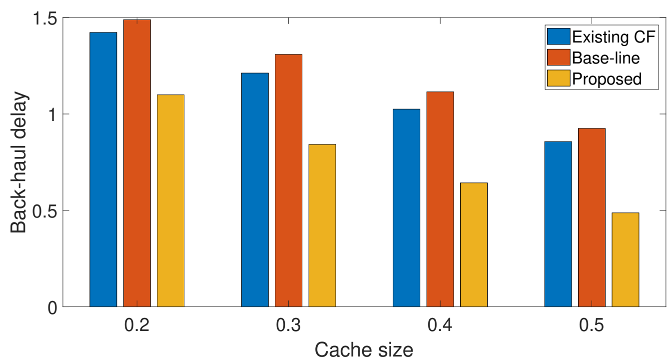

Figure 5.10: Back-haul delay with 256 cloudlets

Throughout all three numbers of cloudlets, a pattern is observed: expanding the capacity of the local cache is decreasing the back-haul delay. Also, for the base-line approach, back-haul delay is relatively the same for all the numbers of cloudlet. For the existing CF approach, level of delay improvement is negligible, however, a slight enhancement is noted with the 128 cloudlets. Surely, the proposed approach outperforms all other approaches with different numbers of cloudlets and cache space values. The progress of back-haul congestion is rapid with larger cache space and more number of cloudlets as a result of gaining access to larger In-Cluster contents.

The following Figures present the impact of storage capacity on back-haul congestion.

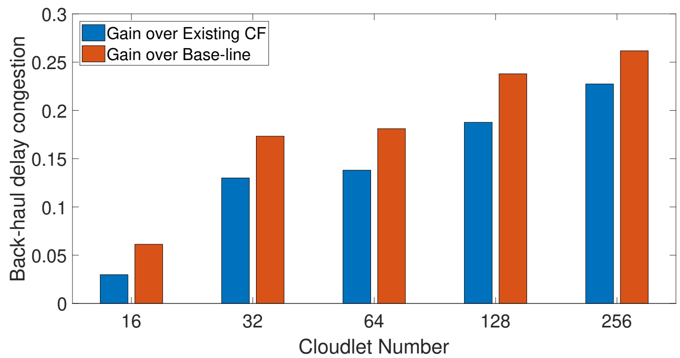

Figure 5.11: Back-haul gain for 20% cache size

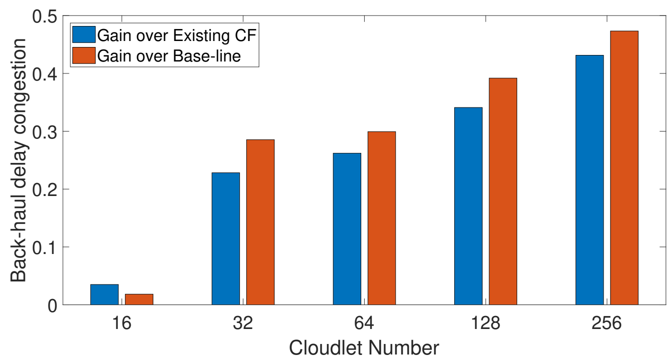

Figure 5.12: Back-haul gain for 50% cache size

For both cache capacities,20% and 50%, the proposed CF approach back-haul congestion is bigger with various cloudlet numbers. From the Figure 5.11, the gain over the base-line always outsizes the existing CF system. With the 32 and the 64 cloudlets, the improvement is limited. For the existing CF, the improvement is sudden from the 16 to the 32 cloudlets. In the Figure 5.12, the gain is greater with all storage sizes. For all numbers of cloudlets, the improvement from the base-line is greater compared to the existing CF approach, except with the 16 cloudlets. Thus implies the base-line approach is performing better with the 16 cloudlets than the existing CF approach. In total, back-haul congestion enhances with larger cache memory space.

5.3.3 Overall delay results

In case of overall delay, the Figures 5.13, 5.14, and 5.15 illustrate the delay behaviours based upon number of cloudlets.

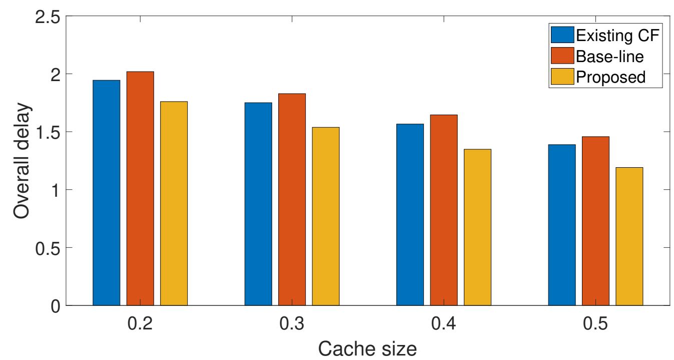

Figure 5.13: Overall delay with 32 cloudlets

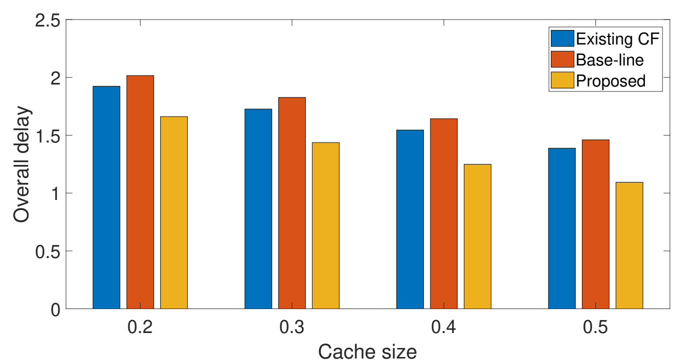

Figure 5.14: Overall delay with 128 cloudlets

Comparing between the 32 and the 128 cloudlets, with cache size 30% and 40%, the base-line delay remain the same. Also noted that with the 50% cache size, the existing CF approach delay did not change. One the other side, our proposed CF approach shows rapt delay reduction with the increment of cache size and cloudlet number.

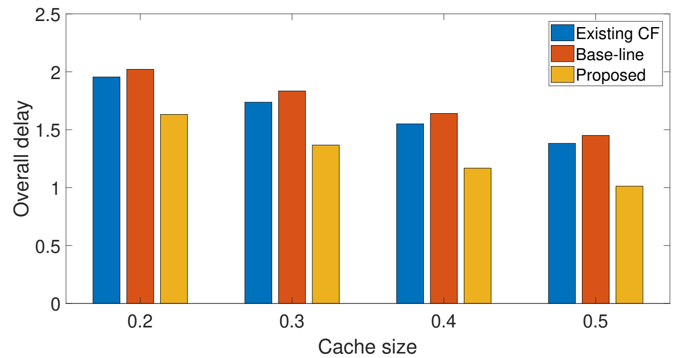

Figure 5.15: Overall delay with 256 cloudlets

The scenario with 256 cloudlets shows further delay reduction for the proposed approach compared to the other approaches. For the existing CF approaches, the delay is better with the 32 cloudlets than with the 128 and the 256 cloudlets. For the base-line approach, the cache capacity influences the performance, in terms of delay, positively. As shown, the proposed approach is scorning less overall delay compared to the other schemes.

Next, the Figure 5.16 and 5.17illustrate the results of cache capacity on the percentage of gain.

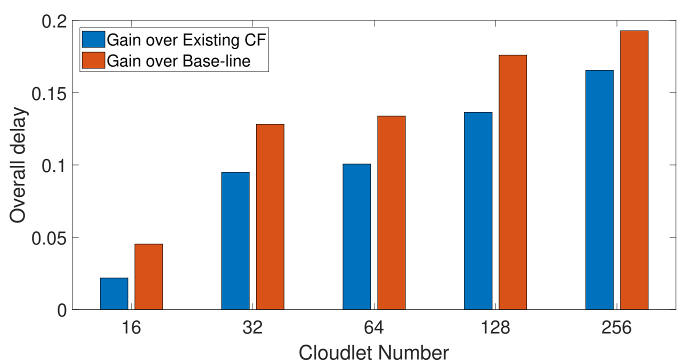

Figure 5.16: Overall delay gain for 20% cache size

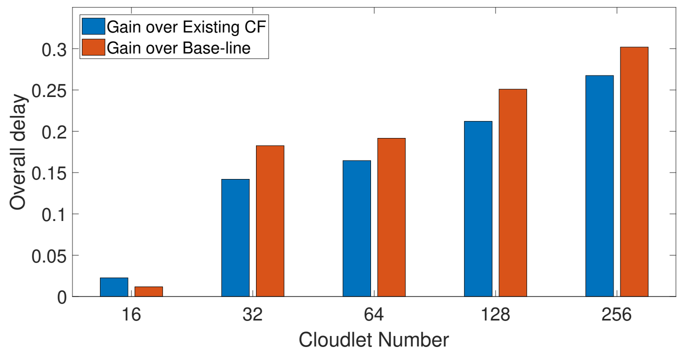

Figure 5.17: Overall delay gain for 50% cache size

From both approaches, the gain is always positive with all the ranges of the numbers of cloudlet and cache capacities. It can be clearly stated that, based on the two Figures, larger cache space increases the gain. Similar to Figure 5.12, the Figure 5.17, with the 16 cloudlets, the base-line approach performs better than the existing CF approach. Except with the 16 cloudlets, the proposed approach performs better with higher cache space.

In all performance metrics, the proposed approach evaluation results are better. It has been observed that the increment in the numbers of cloudlets impacts the CF-based in-network content placement and caching positively. Nonetheless, In-Cache hit ratio for the proposed approach is less compared to the other two approaches. This means that item-based CF was not as successful as anticipated. Moreover, it means that the positive overall hit ratio results are due to the In-Cluster hit ratio. Hence a larger number of cloudlets within a cluster means a larger accessible cache content. That will maximize the possibility of a hit; therefore, the overall performance will respectively progress.

For the existing CF approach, the In-Cache content was accurately placed; however, this will serve only the local cloudlet users and not the users within the cluster. Yet, the proposed approach optimized the network performance by increasing the overall hit ratio, minimizing the back-haul traffic and overall delay.

Chapter 6: Conclusion and Future Work

In this thesis, we explore and examine the effect of Collaborative Filtering-Based in-network content placement and caching on network performance. The framework used multiple techniques, such as clustering, user-based CF, and item-based CF, to increase the accuracy of content prediction and prefetching. The proposed approach yielded very promising results in reducing overall network delay, increasing the hit ratio, in addition to alleviating congestion bottlenecks at backhaul links. Moreover, we found out that item-based CF did not give an accurate prediction, thus requiring further investigation and study.

For future work, there are several considerable vulnerabilities. First, finding a better way to implement the concept of item-based CF, and identifying possible application's misconduct. In addition, our findings could possibly be improved by implementing machine learning with the proposed Collaborative Filtering-Based in-network content placement and caching. We believe that Machine learning can enhance item-based CF accuracy. Finally, mobility needs also to be addressed to cope with the era of mobile devices, such as smart phones and other hand-held wireless devices.

Bibliography

[1] J. Deng, Y. Wang, J. Guo, Y. Deng, J. Gao, and Y. Park, "A similarity measure based on Kullback–Leibler divergence for collaborative filtering in sparse data," *Journal of Information Science*, p. 016555151880818, Oct. 2018.

[2] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura, "Serendipity: enabling remote computing among intermittently connected mobile devices," in *Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing - MobiHoc '12*, (Hilton Head, South Carolina, USA), p. 145, ACM Press, 2012.

[3] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, "Mobility-aware application scheduling in fog computing," *IEEE Cloud Computing*, vol. 4, no. 2, pp. 26–35, 2017.

[4] K. Habak, M. Ammar, K. A. Harras, and E. Zegura, "Femto Clouds: Leveraging Mobile Devices to Provide Cloud Service at the Edge," in *2015 IEEE 8th International Conference on Cloud Computing*, (New York City, NY, USA), pp. 9–16, IEEE, June 2015.

[5] M. Wang, J. Wu, G. Li, J. Li, Q. Li, and S. Wang, "Toward mobility support for information-centric IoV in smart city using fog computing," in *2017 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, (Oshawa, ON, Canada), pp. 357–361, IEEE, Aug. 2017.

[6] H. Sinky and B. Hamdaoui, "Cloudlet-aware mobile content delivery in wireless urban communication networks," in *Global Communications Conference (GLOBECOM), 2016 IEEE*, pp. 1–7, IEEE, 2016.

[7] H. Sinky, B. Khalfi, B. Hamdaoui, and A. Rayes, "Adaptive edge-centric cloud content placement for responsive smart cities," *IEEE Network*, 2019.

[8] R. Abuhadra and B. Hamdaoui, "Proactive in-network caching for mobile on-demand video streaming," in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2018.

[9] H. Sinky, B. Khalfi, B. Hamdaoui, and A. Rayes, "Responsive content-centric delivery in large urban communication networks: A linknyc use-case," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1688–1699, 2018.

[10] P. Blasco and D. Gunduz, "Learning-based optimization of cache content in a small cell base station," in *2014 IEEE International Conference on Communications (ICC)*, (Sydney, NSW), pp. 1897–1903, IEEE, June 2014.

[11] Y. Niu, S. Gao, N. Liu, Z. Pan, and X. You, "Clustered small base stations for cache-enabled wireless networks," in *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, (Nanjing), pp. 1–6, IEEE, Oct. 2017.

[12] A. Pujahari and V. Padmanabhan, "Group recommender systems: Combining user-user and item-item collaborative filtering techniques," in *2015 International Conference on Information Technology (ICIT)*, pp. 148–152, IEEE, 2015.

[13] Q. Han, Í. M. d. R. de Troya, M. Ji, M. Gaur, and L. Zejnilovic, "A collaborative filtering recommender system in primary care: Towards a trusting patient-doctor relationship," in *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 377–379, IEEE, 2018.

[14] S. G. Moghaddam and A. Selamat, "A scalable collaborative recommender algorithm based on user density-based clustering," in *The 3rd International Conference on Data Mining and Intelligent Information Technology Applications*, pp. 246–249, IEEE, 2011.

[15] C. Yu, Q. Tang, Z. Liu, B. Dong, and Z. Wei, "A recommender system for ordering platform based on an improved collaborative filtering algorithm," in *2018 International Conference on Audio, Language and Image Processing (ICALIP)*, pp. 298–302, IEEE, 2018.

[16] E. Shakirova, "Collaborative filtering for music recommender system," in *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pp. 548–550, IEEE, 2017.

[17] T. Tao, "Benfords law, zipfs law, and the pareto distribution," *Retrieved from*, 2009.

[18] S. Abdelwahab, S. Zhang, A. Greenacre, K. Ovesen, K. Bergman, and B. Hamdaoui, "When clones flock near the fog," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1914–1923, 2018.

[19] S. Abdelwahab and B. Hamdaoui, "Flocking virtual machines in quest for responsive iot cloud services," pp. 1–6, 2017.

[20] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Replisom: Disciplined tiny memory replication for massive iot devices in lte edge cloud," *IEEE Internet of Things Journal*, vol. 3, no. 3, pp. 327–338, 2016.

[21] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Cloud of things for sensing as a service: sensing resource discovery and virtualization," in *Global Communications Conference (GLOBECOM), 2015 IEEE*, pp. 1–7, IEEE, 2015.

[22] S. Abdelwahab and B. Hamdaoui, "Fogmq: A message broker system for enabling distributed, internet-scale iot applications over heterogeneous cloud platforms," *arXiv preprint arXiv:1610.00620*, 2016.

[23] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 416–464, 2018.

[24] E. Rapti, C. Houstis, E. Houstis, and A. Karageorgos, "A Bio-Inspired Service Discovery and Selection Approach for IoT Applications," in *2016 IEEE International Conference on Services Computing (SCC)*, (San Francisco, CA), pp. 868–871, IEEE, June 2016.

[25] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13–16, ACM, 2012.

[26] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[27] B. N. T. Duy, Q. A. Nguyen, P. L. Vo, and T.-A. Le, "Optimal content placement for adaptive bit-rate streaming in cache networks," in *2015 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS)*, pp. 243–247, IEEE, 2015.

[28] S. Guizani, "Internet-of-things (IoT) feasibility applications in information Centric Networking System," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, (Valencia, Spain), pp. 2192–2197, IEEE, June 2017.

[29] S. Misra, M. Reisslein, and G. Xue, "A survey of multimedia streaming in wireless sensor networks," *IEEE communications surveys & tutorials*, vol. 10, no. 4, 2008.

[30] A. Majumda, D. G. Sachs, I. V. Kozintsev, K. Ramchandran, and M. M. Yeung, "Multicast and unicast real-time video streaming over wireless lans," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 524–534, 2002.

[31] F. Soldo, C. Casetti, C.-F. Chiasserini, and P. Chaparro, "Streaming media distribution in VANETs," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pp. 1–6, IEEE, 2008.

[32] S. Ehsan and B. Hamdaoui, "A survey on energy-efficient routing techniques with qos assurances for wireless multimedia sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 265–278, 2012.

[33] I. F. Akyildiz, T. Melodia, and K. R. Chowdury, "Wireless multimedia sensor networks: A survey," *IEEE Wireless Communications*, vol. 14, no. 6, 2007.

[34] B. Hamdaoui and P. Ramanathan, "Energy-efficient and mac-aware routing for data aggregation in sensor networks," *Proc. of Sensor Network Operations*, 2004.

[35] B. Hamdaoui and P. Ramanathan, "Cross-layer optimized conditions for qos support in multi-hop wireless networks with mimo links," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 4, 2007.

[36] T.-W. Chen, J. Tsai, and M. Gerla, "Qos routing performance in multihop, multimedia, wireless networks," in *Universal Personal Communications Record, 1997. Conference Record., 1997 IEEE 6th International Conference on*, vol. 2, pp. 557–561, IEEE, 1997.

[37] B. Hamdaoui and P. Ramanathan, "Sufficient conditions for flow admission control in wireless ad-hoc networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 4, pp. 15–24, 2005.

[38] X. Ge, X. Huang, Y. Wang, M. Chen, Q. Li, T. Han, and C.-X. Wang, "Energy-efficiency optimization for mimo-ofdm mobile multimedia communication systems with qos constraints," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 5, pp. 2127–2138, 2014.

[39] B. Hamdaoui and P. Ramanathan, "A cross-layer admission control framework for wireless ad-hoc networks using multiple antennas," *IEEE Transactions on Wireless Communications*, vol. 6, no. 11, 2007.

[40] B. Hamdaoui and K. G. Shin, "OS-MAC: An efficient mac protocol for spectrum-agile wireless networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 8, pp. 915–930, 2008.

[41] B. Hamdaoui, "Adaptive spectrum assessment for opportunistic access in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 2, pp. 922–930, 2009.

[42] M. NoroozOliaee and B. Hamdaoui, "Analysis of guard-band-aware spectrum bonding and aggregation in multi-channel access cognitive radio networks," in *2016 IEEE International Conference on Communications (ICC)*, pp. 1–7, IEEE, 2016.

[43] R. K. Panta, "Mobile video delivery: Challenges and opportunities," *IEEE Internet Computing*, vol. 19, pp. 64–67, May 2015.

[44] K. Piamrat, C. Viho, J. M. Bonnin, and A. Ksentini, "Quality of experience measurements for video streaming over wireless networks," in *2009 Sixth International Conference on Information Technology: New Generations*, pp. 1184–1189, April 2009.

[45] T. A. Q. Pham, K. Piamrat, and C. Viho, "Qoe-aware routing for video streaming over vanets," in *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, pp. 1–5, Sept 2014.

[46] S. Zaidi, S. Bitam, and A. Mellouk, "Enhanced user datagram protocol for video streaming in vanet," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, May 2017.

[47] C. Xu, F. Zhao, J. Guan, H. Zhang, and G. M. Muntean, "Qoe-driven user-centric vod services in urban multihomed p2p-based vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 62, pp. 2273–2289, Jun 2013.

[48] M. NoroozOliaee, B. Hamdaoui, X. Cheng, T. Znati, and M. Guizani, "Analyzing cognitive network access efficiency under limited spectrum handoff agility," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 3, pp. 1402–1407, 2014.

[49] M. NoroozOliaee, B. Hamdaoui, T. Znati, and M. Guizani, "Forced spectrum access termination probability analysis under restricted channel handoff.," in *WASA*, pp. 358–365, Springer, 2012.

[50] M. NoroozOliaee, B. Hamdaoui, and K. Tumer, "Efficient objective functions for coordinated learning in large-scale distributed osa systems," *Mobile Computing, IEEE Transactions on*, vol. 12, no. 5, pp. 931–944, 2013.

[51] M. Maiya and B. Hamdaoui, "imac: improved medium access control for multichannel multi-hop wireless networks," *Wireless Communications and Mobile Computing*, vol. 13, no. 11, pp. 1060–1071, 2013.

[52] M. NoroozOliaee, B. Hamdaoui, and M. Guizani, "Maximizing secondary-user satisfaction in large-scale dsa systems through distributed team cooperation," *Wireless Communications, IEEE Transactions on*, vol. 11, no. 10, pp. 3588–3597, 2012.

[53] B. Hamdaoui, M. NoroozOliaee, K. Tumer, and A. Rayes, "Coordinating secondary-user behaviors for inelastic traffic reward maximization in large-scale osa networks," *Network and Service Management, IEEE Transactions on*, vol. 9, no. 4, pp. 501–513, 2012.

[54] P. Venkatraman, B. Hamdaoui, and M. Guizani, "Opportunistic bandwidth sharing through reinforcement learning," *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 6, pp. 3148–3153, 2010.

[55] A. Sivanantha, B. Hamdaoui, M. Guizani, X. Cheng, and T. Znati, "Em-mac: An energy-aware multi-channel mac protocol for multi-hop wireless networks," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International*, pp. 1159–1164, IEEE, 2012.

[56] B. Hamdaoui, M. NoroozOliaee, K. Tumer, and A. Rayes, "Aligning spectrum-user objectives for maximum inelastic-traffic reward," in *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on*, pp. 1–6, IEEE, 2011.

[57] M. NoroozOliaee, B. Hamdaoui, and K. Tumer, "Achieving optimal elastic traffic rewards in dynamic multichannel access," in *High Performance Computing and Simulation (HPCS), 2011 International Conference on*, pp. 155–161, IEEE, 2011.

[58] P. Venkatraman and B. Hamdaoui, "Cooperative q-learning for multiple secondary users in dynamic spectrum access," in *2011 7th International Wireless Communications and Mobile Computing Conference*, 2011.

[59] M. Maiya and B. Hamdaoui, "An improved ieee 802.11 mac protocol for wireless ad-hoc networks with multi-channel access capabilities," in *High Performance Computing and Simulation (HPCS), 2011 International Conference on*, pp. 162–168, IEEE, 2011.

[60] M. Grissa, A. A. Yavuz, and B. Hamdaoui, "TrustSAS: A trustworthy spectrum access system for the 3.5 ghz cbrs band," in *INFOCOM*, IEEE, 2019.

[61] M. Grissa, B. Hamdaoui, and A. A. Yavuz, "Unleashing the power of multi-server pir for enabling private access to spectrum databases," *IEEE Communications Magazine*, 2018.

[62] M. Grissa, B. Hamdaoui, and A. A. Yavuz, "Location privacy in cognitive radio networks: A survey," *IEEE Communications Surveys & Tutorials*, 2017.

[63] M. Grissa, A. A. Yavuz, and B. Hamdaoui, "Location privacy preservation in database-driven wireless cognitive networks through encrypted probabilistic data structures," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 2, pp. 255–266, 2017.

[64] M. Grissa, A. A. Yavuz, and B. Hamdaoui, "Preserving the location privacy of secondary users in cooperative spectrum sensing," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 2, pp. 418–431, 2017.

[65] M. Grissa, A. Yavuz, and B. Hamdaoui, "When the hammer meets the nail: Multi-server PIR for database-driven CRN with location privacy assurance," in *Communications and Network Security (CNS), 2017 IEEE Conference on*, IEEE, 2017.

[66] M. Grissa, A. A. Yavuz, and B. Hamdaoui, "Cuckoo filter-based location-privacy preservation in database-driven cognitive radio networks," in *Computer Networks and Information Security (WSCNIS), 2015 World Symposium on*, pp. 1–7, IEEE, 2015.

[67] M. Grissa, A. Yavuz, and B. Hamdaoui, "An efficient technique for protecting location privacy of cooperative spectrum sensing users," in *Computer Communications Workshops (INFOCOM WKSHPS), 2016 IEEE Conference on*, pp. 915–920, IEEE, 2016.

[68] N. Adem and B. Hamdaoui, "Jamming resiliency and mobility management in cognitive communication networks," in *Communications (ICC), 2017 IEEE International Conference on*, pp. 1–6, IEEE, 2017.

[69] N. Adem and B. Hamdaoui, "Delay performance modeling and analysis in clustered cognitive radio networks," in *2014 IEEE Global Communications Conference*, pp. 193–198, IEEE, 2014.

[70] N. Adem and B. Hamdaoui, "The impact of stochastic resource availability on cognitive network performance: modeling and analysis," *Wireless Communications and Mobile Computing*, 2015.

[71] N. Adem, B. Hamdaoui, and A. Yavuz, "Mitigating jamming attacks in mobile cognitive networks through time hopping," *Wireless Communications and Mobile Computing*, 2016.

[72] N. Adem, B. Hamdaoui, and A. Yavuz, "Pseudorandom time-hopping anti-jamming technique for mobile cognitive users," in *2015 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, IEEE, 2015.

[73] B. Hamdaoui and K. G. Shin, "Characterization and analysis of multi-hop wireless mimo network throughput," in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pp. 120–129, ACM, 2007.

[74] B. Hamdaoui and P. Ramanathan, "A cross-layer admission control framework for wireless ad-hoc networks using multiple antennas," *IEEE Transactions on Wireless Communications*, vol. 6, no. 11, 2007.

[75] B. Khalfi, B. Hamdaoui, and M. Guizani, "AirMAP: Scalable spectrum occupancy recovery using local low-rank matrix approximation," in *2018 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2018.

[76] B. Khalfi, A. Elmaghbub, and B. Hamdaoui, "Distributed wideband sensing for faded dynamic spectrum access with changing occupancy," in *2018 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2018.

[77] B. Khalfi, B. Hamdaoui, M. Guizani, and N. Zorba, "Efficient spectrum availability information recovery for wideband dsa networks: A weighted compressive sampling approach," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2162–2172, 2018.

[78] B. Hamdaoui, B. Khalfi, and M. Guizani, "Compressed wideband spectrum sensing: Concept, challenges, and enablers," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 136–141, 2018.

[79] B. Khalfi, A. Zaid, and B. Hamdaoui, "When machine learning meets compressive sampling for wideband spectrum sensing," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2017 13th International*, pp. 1120–1125, IEEE, 2017.

[80] B. Khalfi, B. Hamdaoui, and M. Guizani, "Extracting and exploiting inherent sparsity for efficient iot support in 5G: Challenges and potential solutions," *IEEE Wireless Communications Magazine*, vol. 24, no. 5, pp. 68–73, 2017.

[81] S. Abdelwahab, B. Hamdaoui, M. Guizani, and A. Rayes, "Enabling smart cloud services through remote sensing: An internet of everything enabler," *IEEE Internet of Things Journal*, vol. 1, no. 3, pp. 276–288, 2014.

[82] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "Maui: making smartphones last longer with code offload," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pp. 49–62, ACM, 2010.

[83] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for vm-based cloudlets in mobile computing,” *Pervasive Computing, IEEE*, vol. 8, no. 4, pp. 14–23, 2009.

[84] J. Flinn, “Cyber foraging: Bridging mobile and cloud computing,” *Synthesis Lectures on Mobile and Pervasive Computing*, vol. 7, no. 2, pp. 1–103, 2012.

[85] M. D. Kristensen, “Execution plans for cyber foraging,” in *Proceedings of the 1st workshop on Mobile middleware: embracing the personal communication device*, p. 2, ACM, 2008.

[86] A. Mtibaa, K. A. Harras, K. Habak, M. Ammar, and E. W. Zegura, “Towards mobile opportunistic computing,” in *2015 IEEE 8th International Conference on Cloud Computing*, pp. 1111–1114, IEEE, 2015.

[87] J. Li, X. Li, Y. Gao, Y. Gao, and R. Zhang, “Dynamic Cloudlet-Assisted Energy-Saving Routing Mechanism for Mobile Ad Hoc Networks,” *IEEE Access*, vol. 5, pp. 20908–20920, 2017.

[88] F. Jalali, S. Khodadustan, C. Gray, K. Hinton, and F. Suits, “Greening IoT with Fog: A Survey,” in *2017 IEEE International Conference on Edge Computing (EDGE)*, (Honolulu, HI, USA), pp. 25–31, IEEE, June 2017.

[89] A. Saric, M. Hadzikadic, and D. Wilson, “Alternative formulas for rating prediction using collaborative filtering,” in *International Symposium on Methodologies for Intelligent Systems*, pp. 301–310, Springer, 2009.

[90] P. Melville, R. J. Mooney, and R. Nagarajan, “Content-boosted collaborative filtering for improved recommendations,” *Aaai/iaai*, vol. 23, pp. 187–192, 2002.

[91] L. Terveen and W. Hill, “Beyond recommender systems: Helping people help each other,” *HCI in the New Millennium*, vol. 1, no. 2001, pp. 487–509, 2001.

[92] P. Melville and V. Sindhwani, “Recommender systems,” *Encyclopedia of Machine Learning and Data Mining*, pp. 1056–1066, 2017.

[93] Y. Koren, “Factor in the neighbors: Scalable and accurate collaborative filtering,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 4, no. 1, p. 1, 2010.

[94] G. Yao and L. Cai, “User-based and item-based collaborative filtering recommendation algorithms design,” *University of California, San Diego*, 2017.

[95] B. M. Sarwar, G. Karypis, J. A. Konstan, J. Riedl, *et al.*, “Item-based collaborative filtering recommendation algorithms.,” vol. 1, pp. 285–295, 2001.

[96] M. Venu Gopalachari and P. Sammulal, "Personalized collaborative filtering recommender system using domain knowledge," in *International Conference on Computing and Communication Technologies*, (Hyderabad, India), pp. 1–6, IEEE, Dec. 2014.

[97] G. Badaro, H. Hajj, W. El-Hajj, and L. Nachman, "A hybrid approach with collaborative filtering for recommender systems," in *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 349–354, IEEE, 2013.

[98] J. S. Breese and D. H. C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," p. 10.

[99] C. Bernardini, T. Silverston, and O. Festor, "Mpc: Popularity-based caching strategy for content centric networks," in *2013 IEEE international conference on communications (ICC)*, pp. 3619–3623, IEEE, 2013.

[100] A. Struyf, M. Hubert, P. Rousseeuw, *et al.*, "Clustering in an object-oriented environment," *Journal of Statistical Software*, vol. 1, no. 4, pp. 1–30, 1997.

[101] J. Leskovec and A. Rajaraman, "Clustering algorithms," *CS345a: Data Mining, Standford University, United State*, 2010.

[102] G. Seif, "The 5 clustering algorithms data scientists need to know," 2018.

[103] R. C. de Amorim and C. Hennig, "Recovering the number of clusters in data sets with noise features using feature rescaling factors," *Information Sciences*, vol. 324, pp. 126–145, 2015.

[104] T. Thinsungnoena, N. Kaoungkub, P. Durongdumronchaib, K. Kerdprasopb, and N. Kerdprasopb, "The clustering validity with silhouette and sum of squared errors," *learning*, vol. 3, p. 7, 2015.

[105] P. Melville and V. Sindhwani, "Recommender systems," *Encyclopedia of Machine Learning and Data Mining*, pp. 1056–1066, 2017.