This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2616871, IEEE Transactions on Mobile Computing

A Proxy-based Collaboration System to Minimize Content Download Time and Energy Consumption

Insun Jang, Gwangwoo Park, Dongeun Suh, Sangheon Pack, and György Dán

Abstract—Mobile collaborative community (MCC) is an emerging technology that allows multiple mobile nodes (MNs) to perform a resource intensive task, such as large content download, in a cooperative manner. In this paper, we introduce a proxy-based collaboration system for the MCC where a content proxy (CProxy) determines the amount of chunks and the sharing order scheduled to each MN, and the received chunks are shared among MNs via Wi-Fi Direct. We formulate a multi-objective optimization problem to minimize both the collaborative content download time and the energy consumption in an MCC, and propose a heuristic algorithm for solving the optimization problem. Extensive simulations are carried out to evaluate the effects of the number of MNs, the wireless bandwidth, the content size, and dynamic channel conditions on the content download time and the energy consumption. Our results demonstrate that the proposed algorithm can achieve near-optimal performance and significantly reduce the content download time and has an energy consumption comparable to that of other algorithms.

Index Terms—Mobile collaborative community, content download, proxy-based collaboration system, α-local search of sharing order (α-LSSO).

1 INTRODUCTION

Although the peak bit rate of wireless access technologies is continuously increasing, it is still insufficient for bandwidth-intensive applications, such as large content downloads (e.g., multimedia service [2]) and real-time 3D video streaming. Moreover, since link spectral efficiency has fundamental limits, collaborative bandwidth aggregation techniques at the data link and network layers have been recently considered, including systems for collaborative content download [3]–[13]. In these systems, multiple mobile nodes (MNs) within proximity of each other form a collaboration group, called mobile collaborative community (MCC), to improve content download performance. Each MN in the MCC downloads a part of the content, often referred to as a chunk, and shares the received chunk with other MNs in the MCC via unicast or multicast transmission. During such collaborative download, the MNs use multiple interfaces, a wireless wide area network (WWAN) for downloading content chunks from the origin server and a wireless local area network (WLAN) for sharing the content chunks within the MCC.

Collaborative content download in an MCC can lead to reduced content download time because the WLAN typically provides a much higher data rate than the WWAN [3]–[7]. Moreover, each MN can reduce its use of the WWAN, which may lead to lower communication cost [8], [9], and may also reduce its energy consumption [10], [11]. At the system level, content download via an MCC can reduce the traffic load of the WWAN [12], [13], thus providing benefits for the mobile operator as well. Minimizing the content download time and the energy consumption in an MCC is, however, challenging as the download time and the energy consumption depend both on the chunk sizes downloaded by the individual MNs and on the sharing order among the MNs within the MCC. Furthermore, the optimal choice of these parameters is a function of the WWAN and WLAN channel conditions, i.e., the achievable data rates of the MNs.

Previous works have explored distributed and centralized solutions for forming and managing MCCs [3]–[13]. In the distributed solutions [3], [4], [6], [9], MNs spontaneously form and manage the MCC by exchanging control messages for collaboration with each other. The distributed approach can incur high control overhead due to frequent exchanges of control information for collaboration. Moreover, it is not easy to obtain up-to-date information about the status of neighboring MNs through fast changing wireless channels, and thus the resulting performance can be far from the optimal. In the centralized solutions [5], [8], [10], [12], [13], collaboration is mediated by a central entity such as a base station (BS) and a content server (CServer). Even though the centralized approach can allow scheduling to coordinate the MNs, the central entity can be a bottleneck and communication with the central entity can incur a significant

- *A preliminary version of this paper was presented at IEEE ICC 2014, Sydney, Australia, June 2014 [1].*

- *I. Jang, G. Park, D. Suh, and S. Pack are with the School of Electrical Engineering, Korea University, Seoul, Korea. (e-mail:{zerantoss, kwoo17, fever1989, shpack}@korea.ac.kr, corresponding author: S. Pack).*
- *G. Dán is with the School of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden (e-mail: gyuri@ee.kth.se).*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2616871, IEEE Transactions on Mobile Computing

latency depending on the distance between the central entity and the MCC. In addition, both centralized and distributed solutions are discussed in [7], [11].

In this paper, we propose a proxy-based collaboration system that combines the advantages of the distributed and the centralized approaches. In the proposed system, the MCC formation and the chunk sharing are performed using Wi-Fi Direct [14] in a distributed manner while a content proxy (CProxy) performs the scheduling and the MCC management, including the collection of MN information, in a centralized manner. For scheduling at the CProxy, we formulate the problem of minimizing both the collaborative content download time and the energy consumption in an MCC as a multi-objective optimization problem, by jointly considering the chunk size and the sharing order. Then, the multi-objective optimization problem is transformed into a single-objective mixed integer nonlinear programming (MINLP) problem by forming the weighted sum of the objectives [15].

Since the MINLP problem is known to be NP-hard, we propose a heuristic algorithm called α-local search of the sharing order (α-LSSO), which is inspired by the 2-opt algorithm [16]. We show that α-LSSO runs in polynomial time and thus can be executed at the CProxy. Simulation results demonstrate that α-LSSO achieves near-optimal performance and can significantly reduce the content download time and the achieved energy consumption is comparable to that of other algorithms depending on the value of the parameter α, which allows to balance between reduction of content download time and reduction of energy consumption.

The contribution of this paper is three-fold. First, it jointly considers the chunk size and the sharing order for the minimization of download time and energy consumption. Second, the proposed proxy-based approach combines the advantages of the distributed and the centralized approaches, and can be realized by means of emerging technologies such as software-defined networking (SDN) [17] and network function virtualization (NFV) [18]. Third, extensive simulation results show that the performance of the proposed algorithm is close to that of the optimal solution while it can be easily implemented and operates in polynomial time.

The remainder of this paper is organized as follows. Section 2 discusses the related work. The proxy-based collaboration system is introduced in Section 3 and the optimal scheduling problem is described in Section 4. The proposed α-LSSO algorithm is described in Section 5 and extensive simulation results are presented in Section 6. Section 7 concludes the paper.

2 Related Work

Previous works in the area of collaborative content download follow a distributed approach [3], [4], [6], [9], a centralized approach [5], [8], [10], [12], [13], or both of them [7], [11].

Following the distributed approach, MNs spontaneously form and manage the MCC by exchanging control messages among each other. C5 [3] exploits MAC layer multicast for one-hop neighbors to increase the efficiency in downloading a content. In C5, the chunk size is fixed and the sharing order is randomly coordinated by the MNs within the MCC assuming that the content size is infinite, which makes that the optimal content download time cannot be achieved. In COMBINE [4], the MNs within the WLAN range pool their WWAN bandwidths and support the WWAN download for a target MN. COMBINE finds the optimal chunk sizes of the MNs to reduce the WWAN download time of the target MN under constraints on the communication cost and the energy consumption. Nonetheless, the sharing order over the WLAN is not taken into account. MicroCast [6] leverages WiFi overhearing and network coding to increase the average download rate in an MCC. Although exploiting the overhearing feature in the wireless medium is novel, the issues of determining the chunk size and the sharing order for MNs are not thoroughly investigated. COSMOS [9] considers peer-to-peer collaboration for video streaming, where peers pull chunks of video content from a server and exchange the chunks and related meta-information with neighbors through broadcasting. COSMOS uses a collaborative download technique through multi-hop wireless links, but does not aim to optimize the content download time or the energy consumption.

Following the centralized approach, a central entity controls the collaboration through interaction with the MCC. In [5], a framework to maximize the total throughput in an MCC is introduced in which helper MNs overhear the data that a specific MN receives from the BS and then forward it to the MN. The problem of selecting the optimal transmission modes of the MNs is formulated as an integer linear programming (ILP) problem, and a heuristic algorithm that executes in the BS is proposed. CHUM [8] reduces the communication cost of distributing multimedia content in a WWAN. In CHUM, a server determines the data download order of the MNs, and the MNs share the downloaded data through multi-hop wireless links, but the collaboration between MNs is not optimized. The authors in [12] considered cellular offloading through an MCC. In the considered problem, the BS aims to minimize the number of cellular channels used for downloading contents while ensuring fair energy consumption among the MNs, and a solution is provided by using dynamic programming. Likewise, [13] investigates the reduction of cellular network usage through the MCC with the objective of reducing the energy consumption of the BS and the MNs. With these two objectives, a joint optimization problem is formulated in order to find the optimal transmission rate of content and the relay durations of MNs. The problem is decomposed into two subproblems and then an optimal solution is derived analytically. [10] aims at reducing the energy consumption of the MNs in an MCC by selecting a specific MN that receives a chunk from a BS and then shares it to other MNs. The BS chooses the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2616871, IEEE Transactions on Mobile Computing

TABLE 1
Comparison of collaborative content download schemes (Dist: Distributed, Cent: Centralized, Alg: Algorithm, Opt: Optimization, Max: Maximize, Min: Minimize).

Reference	MCC Control Solutions		Objectives	Decision Variables
	Formation	Scheduling		
[1]	-	Cent. Opt	Min. download time	Chunk size, sharing order
[3]	Dist. Alg		Max. download rate	Chunk selection
[4]	Dist. Alg		Min. download time	Chunk size
[5]	-	Cent. Opt.	Max. throughput	Transmission mode
[6]	Dist. Alg		Max. download rate	Chunk selection
[7]	Cent. Alg or Dist. Alg		Max. download rate	Chunk size
[8]	Cent. Alg		Min. communication cost	MN selection
[9]	Dist. Alg		Min. communication cost	Sharing range
[10]	-	Cent. Opt	Min. energy consumption	MN selection
[11]	Cent. Opt or Dist. Alg		Min. energy consumption	MN selection, multicast rate
[12]	Cent. Opt		Min. cellular usage	MN selection, multicast rate
[13]	-	Cent. Opt	Min. energy consumption, cellular usage	Data rate, sharing duration
Proposed	Dist. Alg	Cent. Opt	Min. download time, energy consumption	Chunk size, sharing order

MN from a list of candidate nodes that can reduce the energy consumption, and then selects a particular MN that leads to a fair allocation of the communication cost.

A few recent works consider both centralized and distributed approaches [7], [11]. In [7], the authors propose Unity, which is running on each MN and aims at accelerating content download by utilizing WiFi and Bluetooth for local sharing. [7] introduces an architecture consisting of user interface, local networking, downloader, and controller, and evaluates two heuristic collaboration policies, equal chunk division and bandwidth-proportional chunk division. [7] also presents Unity cloud which is a centralized solution to support collaborative downloading through intermittently connected MNs. In [11], centralized and distributed algorithms are provided for forming disjoint MCCs with the selection for content distributors and determining the multicast data rate in each MCC, while minimizing the energy consumption.

These previous works focus on various aspects of content download in an MCC, but they do not consider the joint optimization of the chunk size and the sharing order, and they do not allow to explore the trade-off between download time and energy consumption. Most related work to ours is [1], where the problem of jointly finding the chunk sizes and the sharing order that minimize the content download time is formulated as an MINLP problem, and a heuristic algorithm is proposed to approximate the optimal solution. Compared to [1], in this work, we consider the joint optimization of the chunk size and the sharing order as a multi-objective optimization problem, which allows simultaneous optimization of the content download time and the energy consumption. Moreover, the optimization is performed in the proposed proxy-based collaboration system. These considerations set our work apart from our own previous work [1]. Table 1 summarizes the comparison of collaborative content download schemes including the proposed scheme.

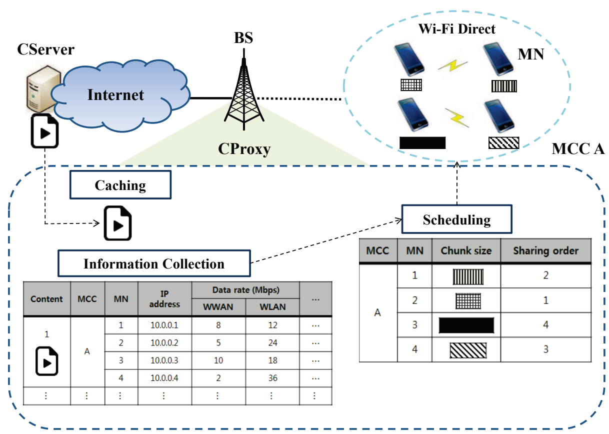


Fig. 1. **Proxy-based collaboration system.**

3 Proxy-based Collaboration System

In this section, we present the proxy-based collaboration system and explain the operation of collaborative content download.

3.1 System Overview

The proxy-based collaborative download system consists of a CServer, a CProxy, and an MCC as shown in Figure 1. The MCC is composed of several MNs that want to download the same content, which is stored in the CServer. The CProxy performs the optimization and management of the MCC in a centralized manner, while possibly pre-fetching the content from the CServer. In

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2616871, IEEE Transactions on Mobile Computing

addition, the MNs use Wi-Fi Direct for distributed MCC formation and chunk sharing.

The proposed MCC formation scheme leverages device discovery and group formation mechanisms defined in Wi-Fi Direct [14]. MNs first recognize each other by alternating between *search* and *listen* states on so called *social* channels (i.e., channels 1, 6, or 11 in the 2.4GHz band). Subsequently, MNs negotiate to create the MCC and elect an MN as a group owner (GO) that performs access point (AP)-like functionalities. The GO in the MCC selects an operating channel and assigns IP addresses to the MNs by means of the dynamic host configuration protocol (DHCP). As Wi-Fi Direct leverages the IEEE 802.11 standard infrastructure mode, the MNs can use typical Wi-Fi data rates and ranges (up to IEEE 802.11n) for chunk sharing.

The CProxy is a central entity that supports collaborative content download by implementing three key functions: centralized collection of the MCC information, optimal scheduling, and content caching and prefetching. As illustrated in Figure 1, the CProxy can be colocated with the BS between the CServer and the MCC[1], and thus it can monitor and collect status information (e.g., wireless channel conditions and connectivity disruption due to mobility) of the MNs in the MCC. Therefore, the CProxy can have access to the necessary channel quality information for optimizing the chunk size and the sharing order of the MNs. In addition, the CProxy fetches the contents from the CServer and caches them, thereby accelerating the service of requests for the content. Consequently, the MNs within the MCC can receive the content faster than if receiving it from the CServer.

3.2 Operation of Collaborative Content Download

Figure 2 shows the operation of collaborative content download in the proxy-based collaboration system. We make the reasonable assumption that each content has a unique content identifier (ID), e.g., a uniform resource identifier (URI). An MN can request a content from the CProxy by specifying the content ID. After receiving the request, the CProxy sends a response message to the MN, which includes the service set identifier (SSID) of the corresponding MCC for the content (Step 1). In this way the same SSID is assigned to MNs that request the same content. Subsequently, the CProxy fetches the content from the CServer and caches it (Step 2). Meanwhile, the MNs that have received a response message turn on Wi-Fi Direct and recognize each other by means of the discovery procedure of Wi-Fi Direct. **After that, the MNs engage in negotiating a GO for the MCC. Considering two MNs initially (denoted by A and B), the GO negotiation follows a three-way handshake by means of three messages: 1) *request*, 2) *response*, and 3) *confirmation* [14]. First, MN A sends a *request* message, which includes a numerical GO intent value and a list of operating channels for the MCC, to MN B. MN B itself chooses an intent value, and recognizes itself as a GO if its intent value is larger than that of MN A. After that, the GO (i.e., MN B) selects its operating channel from the channel list and sends a *response* message to MN A. In sequel, MN A sends a *confirmation* message to the GO to complete GO negotiation. Once the GO is chosen, it invites the remaining MNs that were found during the device discovery procedure, assigns a unique MCC ID, and chooses a multicast IP address.** Finally, the GO advertises the MCC information including the multicast IP address and the MCC ID to the MNs within the MCC (Step 3). **After completing the MCC formation, the MNs inform the CProxy of their network information (i.e., WWAN/WLAN data rates) and of the MCC information[2] (Step 4).** Based on these information, the CProxy determines the chunk size and the sharing order in the MCC by executing the α-LSSO algorithm presented in Section 5 (Step 5). The CProxy divides the content into chunks (Step 6), and it then distributes the chunks and the scheduling information (i.e., the sharing order and the chunk size) to the MNs, who then share the chunks via multicast using Wi-Fi Direct (Steps 7–8).

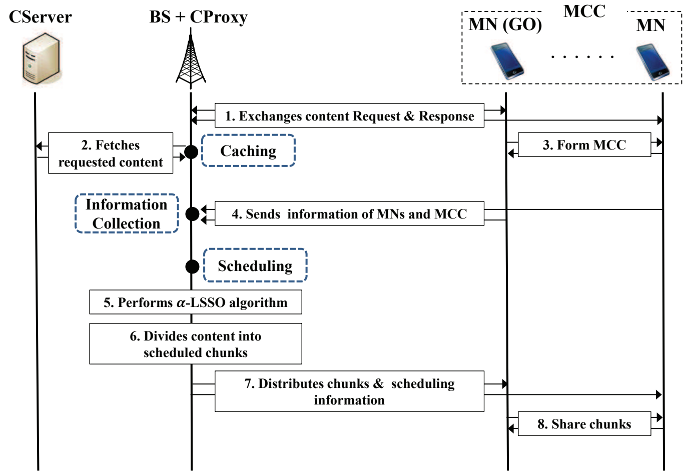


Fig. 2. **The operation of collaborative content download.**

Collaborative content download is clearly susceptible to disruptions in wireless connectivity and to potential MN failures. **If an MN looses WWAN connectivity to the CProxy, the CProxy informs the other MNs about the chunk assigned to the MN that lost connectivity. This is because the CProxy cannot know whether the**

1. Recent advances in SDN/NFV technologies allow a flexible and programmable BS platform [17], [19] and a mobile edge cloud (MEC) platform [20], which are 5G enablers for efficient wireless content delivery. Various functionalities and services, including a CProxy, can be run on these platforms.

2. **The transmission delay for information collection can be assumed to be negligible since the signaling message for the status information is very small compared to the size of the content and the CProxy can be deployed on a MEC platform. As an example, in Long Term Evolution (LTE) the channel quality indicator (CQI) information is expressed as a 4-bit integer [29].**

MN will transmit the downloaded data to the other MNs. It is thus safe to simply consider that the chunk assigned to the failed MN has to be reassigned. On the contrary, if an MN cannot transmit a portion of its assigned chunk to the other MNs in the MCC owing to WLAN disconnection, the affected MNs, except the disconnected MN, report the undelivered data to the CProxy. When the CProxy detects such failure events for affected MNs in the MCC, it creates a new download schedule by executing the α-LSSO algorithm for the failed portion of the content, as done in [3], [4], [6], [7]. **After that, the CProxy re-distributes the failed portion of the content with the corresponding scheduling information to the remaining MNs in the MCC[3].**

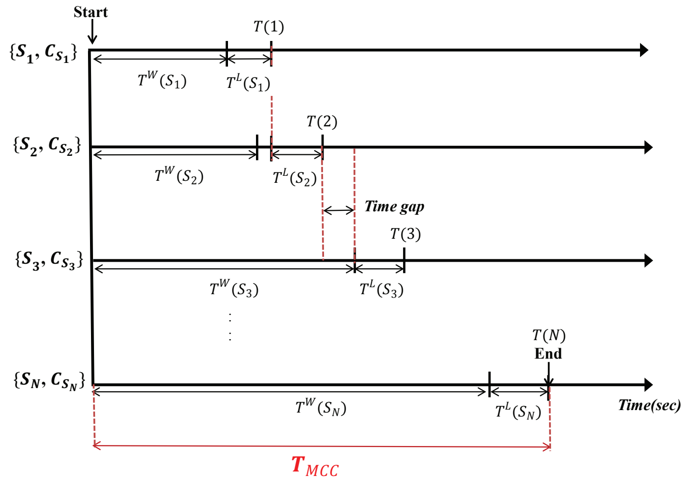


Fig. 3. Timing diagram for problem formulation.

4 System Model and Problem Formulation

In this section, we formulate the problem of minimizing the content download time and the energy consumption in an MCC by jointly considering the chunk size and the sharing order.

4.1 System Model

We consider a set $\mathcal{N} = \{1, \ldots, N\}$ of MNs that form an MCC. We denote the WWAN data rate of MN n by w_n and its WLAN data rate by l_n. Note that the WLAN data rate refers to the maximum data rate for multicasting a chunk to all other MNs in the MCC. The content has size $C \in \mathbb{N}$, is divided into N chunks, and chunk n is downloaded via the WWAN to MN n. We denote by C_n the size of chunk n, and define the chunk size vector $\mathbf{C} = \{C_1, \ldots, C_N\}$. By definition $\mathbf{C} \in \mathcal{C} = \{\mathbf{c} \in \mathbb{R}_+^N | \sum_{n=1}^N c_n = C\}$. Recall that only one MN at a time can share its chunk using the WLAN, thus there need to be N sharing rounds. We denote by $S_i \in \mathcal{N}$ the MN that shares its chunk at the ith sharing round, and we define the sharing order vector $\bar{S} = (S_1, \ldots, S_N)$. To simplify notation in the problem formulation, for $1 \le i, n \le N$ we define the sharing assignment $S_{(i,n)} = \delta_{S_i,n}$, where δ is the Kronecker delta, and the $N \times N$ sharing assignment matrix $\bar{\bar{\mathbf{S}}} = (S_{(i,n)})$. With this definition $S_{(i,n)} = 1$ if and only if MN n shares its chunk at sharing round i, and $S_{(i,n)} = 0$ otherwise. We denote by $\mathcal{S}$ the set of possible sharing assignment matrices, which is the set of $N \times N$ permutation matrices.

4.2 Download time

The total download time consists of the WWAN download times and of the WLAN sharing times, as illustrated in Figure 3. We denote by $T^W(S_i)$ the time when MN S_i finishes the download of its chunk over the WWAN, which is given by

$$T^W(S_i) = \sum_{n=1}^{N} \frac{C_n S_{(i,n)}}{w_n}.$$

Similarly, we denote by $T^L(S_i)$ the WLAN sharing time of MN S_i, i.e.,

$$T^L(S_i) = \sum_{n=1}^{N} \frac{C_n S_{(i,n)}}{l_n}.$$

We denote by $T(i)$ the elapsed content download time until the end of the WLAN sharing round i. Finally, we define the total collaborative download time T_{MCC} as the elapsed time from the beginning of the WWAN content download until the end of the last WLAN sharing round.

Let us start with deriving an expression for $T(i)$. It is easy to see that $T(1) = T^W(S_1) + T^L(S_1)$. To derive $T(i)$ for $1 < i \le N$, we need to consider two cases. In the first case, the WWAN download of MN S_i finishes before the WLAN sharing of MN S_{i-1}, i.e., $T^W(S_i) \le T(i-1)$ (e.g., between S_1 and S_2 in Figure 3). In this case, $T(i) = T(i-1) + T^L(S_i)$. In the second case, the WLAN sharing of MN S_{i-1} finishes before the WWAN download of MN S_i, i.e., $T^W(S_i) > T(i-1)$ (e.g., between S_2 and S_3 in Figure 3). Thus, there exists a gap of $T^W(S_i) - T(i-1)$ between the end of sharing round $i-1$ and the start of sharing round i, and $T(i) = T(i-1) + (T^W(S_i) - T(i-1)) + T^L(S_i)$. By defining $T(0) = 0$, we can express $T(i)$ using the following recursion

$$\begin{aligned} T(i) = \ & T(i-1) + \max\left(T^W(S_i) - T(i-1),\ 0\right) \\ & + T^L(S_i). \end{aligned} \tag{1}$$

Observe that by definition $T_{MCC} = T(N)$.

4.3 Energy consumption model

We model the energy consumption of each MN as a function of the chunk transmission/reception time and

3. **Additional time and energy consumption for re-distribution of the CProxy are needed. The effect of MN failure on the content download time is analyzed in Appendix B.**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2616871, IEEE Transactions on Mobile Computing

TABLE 2
Summary of notations.

Parameter	Description
N	Number of MNs in MCC
w_n	WWAN data rate of MN n
l_n	WLAN data rate of MN n
C	Content size
C_n	Chunk size assigned to MN n
S_i	ID of the MN that shares its chunk at sharing round i
$S_{(i,n)}$	Indicator function of whether MN n shares its chunk at sharing round i
$T(i)$	Elapsed content download time up to sharing round i
$T^W(n)$	WWAN download time of MN n
$T^L(n)$	WLAN sharing time of MN n
T_{MCC}	Total collaborative content download time in MCC
E_n^W	Reception energy consumption of MN n in WWAN
$E_n^{L,tx}$	Transmission energy consumption of MN n in WLAN
$E_n^{L,rx}$	Reception energy consumption of MN n in WLAN
E_{MCC}	Energy consumption in MCC

of the consumed power for the given data rate [12], [21]. We make the reasonable assumption that the WWAN receive power P^W is the same for all MNs, and we denote by $P^{L,tx}$ and $P^{L,rx}$ the WLAN transmit and receive powers of the MNs. Using this notation, we can express the energy consumed by MN S_i while receiving its chunk C_{S_i} via the WWAN as

$$E_{S_i}^W = \sum_{n=1}^{N} \frac{C_n S_{(i,n)}}{w_n} P^W. \tag{2}$$

Similarly, we can express the energy consumed by MN S_i while transmitting its chunk C_{S_i} to other MNs in the MCC via the WLAN as

$$E_{S_i}^{L,tx} = \sum_{n=1}^{N} \frac{C_n S_{(i,n)}}{l_n} P^{L,tx}. \tag{3}$$

To compute the energy consumed for receiving data via the WLAN, recall that MN S_i receives chunks from $(N-1)$ MNs in the MCC. Thus, the energy consumed by MN S_i while receiving chunks from other MNs via the WLAN is

$$E_{S_i}^{L,rx} = \sum_{u=1,u\neq n}^{N} \sum_{n=1}^{N} \frac{C_u S_{(i,n)}}{l_u} P^{L,rx}. \tag{4}$$

The total energy consumption of MN S_i is then given by

$$E_{S_i} = E_{S_i}^W + E_{S_i}^{L,tx} + E_{S_i}^{L,rx}. \tag{5}$$

Finally, we can express the average energy consumption of an MN in the MCC as

$$E_{MCC} = \frac{1}{N} \sum_{i=1}^{N} E_{S_i}. \tag{6}$$

Table 2 summarizes the most frequent notations.

4.4 Download Time-Energy Consumption Minimization Problem

In the following, we formulate the problem of minimizing T_{MCC} and E_{MCC} by jointly determining the chunk size vector $\mathbf{C}$ and the sharing assignment matrix $\bar{\bar{\mathbf{S}}}$. We first introduce constraints, followed by the objective function. The first constraint ensures that the sum of the chunk sizes downloaded by the individual MNs is the same as the original content size,

$$\sum_{n=1}^{N} C_n = C. \tag{7}$$

Second, we require that all MNs share their chunks only once, i.e., one sharing round is assigned to only one MN,

$$\sum_{n=1}^{N} S_{(i,n)} = 1, \qquad 1 \le i \le N, \tag{8}$$

$$\sum_{i=1}^{N} S_{(i,n)} = 1, \qquad \forall n \in \mathcal{N}. \tag{9}$$

Recall that $S_{(i,n)}$ takes binary values and $C_n \in \mathbb{R}^+$,

$$S_{(i,n)} \in \{0,1\},\ C_n \ge 0, \quad 1 \le i, n \le N. \tag{10}$$

We can now formulate the problem of minimizing the content download time and the energy consumption in an MCC as the following multi-objective optimization problem

$$\min_{\mathbf{C}\in\mathcal{C},\bar{\bar{\mathbf{S}}}\in\mathcal{S}} (T_{MCC}, E_{MCC}) \quad \text{subject to} \quad (7)-(10). \tag{11}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2616871, IEEE Transactions on Mobile Computing

We use the weighted sum method [15] with weight α to transform the above multi-objective problem into a single objective problem. In order to obtain a dimensionless objective function with an upper bound of one [15], we normalize the download time and the energy consumption by the maximum download time T_{max} and the maximum energy consumption E_{max}, respectively. T_{max} and E_{max} can be computed by observing that the worst case scenario is when the MN with the worst channel condition downloads and shares the full content. The ease of computing these values motivates our choice for using these as normalizing factors, instead of, e.g., the average values which would need the solution of the optimization problem itself. Consequently, we obtain the following optimization problem

$$\min_{\mathbf{C}\in\mathcal{C},\bar{\mathbf{S}}\in\mathcal{S}} \alpha\frac{T_{MCC}}{T_{max}} + (1-\alpha)\frac{E_{MCC}}{E_{max}} \quad \text{subject to} \quad (7)-(10). \tag{12}$$

Note that the real variable C_n and the binary variable $S_{(i,n)}$ are multiplied in the problem (12). Thus, the optimization problem (12) is an MINLP problem. Since MINLP problems are known to be NP-hard [22], in what follows we propose a polynomial-time approximation algorithm.

5 α-Local Search of Sharing Order Algorithm

The proposed α-Local Search of Sharing Order (α-LSSO) Algorithm evaluates sharing order vectors in an iterative manner. α-LSSO is based on the 2-opt algorithm, which is a local search algorithm used for approximating the traveling salesman problem (TSP) [16]. 2-opt is known to provide good results in terms of approximation ratio and running time on Euclidean instances of the TSP problem, but its worst case computational complexity is exponential [23]. To ensure polynomial computational complexity, in α-LSSO the search space is pruned compared to 2-opt.

5.1 Algorithm Description

The proposed α-LSSO algorithm reduces the content download time and the energy consumption by changing the sharing order vector repeatedly. The condition for accepting a new sharing order vector depends on the weight α, which allows to trade-off between reduction of T_{MCC} and reduction of E_{MCC}. Algorithm 1 shows the operation of α-LSSO.

The rationale for iteratively changing the sharing order vector is as follows. For a particular sharing order vector $\bar{S}$, the optimization problem (12) corresponds to a linear programming (LP) problem for the chunk size vector $\mathbf{C}$, which can be solved in polynomial time via well-known algorithms (e.g., interior point method [24]) implemented in various LP solvers (e.g., IBM ILOG CPLEX), and provides the optimal chunk size vector for given $\bar{S}$. The alternative would be to repeatedly change the chunk size vector $\mathbf{C}$, but for a particular chunk size vector $\mathbf{C}$, the optimization problem (12) corresponds to an ILP problem for the sharing order vector $\bar{S}$, which is NP-hard. Furthermore, there is a continuum of chunk size vectors and thus enumerating all factors is infeasible.

Algorithm 1 α-LSSO algorithm.

1: Choose initial sharing order vector $\bar{S}\in\mathcal{S}$ uniform at random
2: Let $\mathbf{C} = \arg\min_{\mathbf{C}\in\mathcal{C}} LP_{MCC}(\bar{S},\mathbf{C})$, compute T_{MCC} and E_{MCC}
3: Set $T^{min} = T_{MCC}$, $E^{min} = E_{MCC}$, $\bar{S}^{min} = \bar{S}$, $\mathbf{C}^{min} = \mathbf{C}$, $i = 1$, $j = 1$
4: **for** $i = 1$ **to** N **do**
5:   **for** $j = 1$ **to** N **do**
6:     **if** $i \neq j$ **then**
7:       Set $S'_i = S^{min}_j$ and $S'_j = S^{min}_i$
8:       Let $\mathbf{C}' = \arg\min_{\mathbf{C}\in\mathcal{C}} LP_{MCC}(\bar{S}',\mathbf{C})$, compute T'_{MCC} and E'_{MCC}
9:       Let $T_r = (T^{min} - T'_{MCC})/T^{min}$
10:       Let $E_r = (E^{min} - E'_{MCC})/E^{min}$
11:       Let $G_r = \alpha T_r + (1-\alpha)E_r$
12:       **if** $G_r \geq 0$ **then**
13:         Set $T^{min} = T'_{MCC}$, $E^{min} = E'_{MCC}$, $\bar{S}^{min} = \bar{S}'$, $\mathbf{C}^{min} = \mathbf{C}'$
14:       **end if**
15:     **end if**
16:   **end for**
17: **end for**

The algorithm starts with a randomly chosen initial sharing order vector $\bar{S}\in\mathcal{S}$ (line 1). As mentioned above, the chunk size vector $\mathbf{C}$ for given $\bar{S}$ can be obtained by solving an LP problem (denoted by LP_{MCC}). Using the computed $\mathbf{C}$ and $\bar{S}$, the content download time T_{MCC} and the energy consumption E_{MCC} can be computed (line 2). Then, T^{min}, E^{min}, $\bar{S}^{min}$, and $\mathbf{C}^{min}$ are set to T_{MCC}, E_{MCC}, $\bar{S}$, and $\mathbf{C}$, respectively (line 3). After the initialization, two MNs are swapped in the sharing order upon every iteration of the algorithm; an MN at a sharing round in $\bar{S}^{min}$ is swapped with another MN at another sharing round in $\bar{S}^{min}$. The swapping procedure is repeated once for every pair of sharing rounds (i.e., $N\times(N-1)$) (lines 4-17), which guarantees that the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2616871, IEEE Transactions on Mobile Computing

running time of the algorithm is bounded.

For each swapping, the content download time T'_{MCC} and the energy consumption E'_{MCC} are recomputed (lines 7-8), and are used to compute the relative change T_r and E_r for the content download time and for the energy consumption, respectively (lines 9-10). Following the multi-objective problem formulation with α, we define the ratio $G_r = \alpha T_r + (1-\alpha) E_r$ (line 11). If $G_r \geq 0$, the modified sharing order $\bar{S}'$ replaces the so far best sharing order $\bar{S}^{min}$, and $\mathbf{C}^{min}$, T^{min} and E^{min} are accordingly updated (lines 12-14).

5.2 Time Complexity Analysis

In Algorithm 1, the **for** loop in lines 4-17 iterates N times, and the number of iterations of the **for** loop at lines 5-16 is $N-1$. Therefore, the number of iterations in the nested **for** loop is $N \times (N-1)$. Since solving the LP problem for $\mathbf{C}$ can be done in polynomial time, the complexity of α-LSSO is polynomial in N. In particular, if an interior point method with the complexity $O(N^{3.5})$ is used for solving the LP problem, the complexity of α-LSSO becomes $O(N^{5.5})$. Since the complexity of α-LSSO is polynomial, the CProxy is able to execute the algorithm in real-time without the need for offline computations. Note that the complexity does not include the time and signaling needed for MCC formation since the time for MCC formation is relatively small compared to the content download time, e.g., the time for MCC formation in Wi-Fi Direct can be less than 5sec as in [25].

6 Simulation Results

In what follows we provide simulation results to give insight into the trade-off between download time and energy consumption for collaborative content download. Our evaluation does not consider the resource and time requirement of MCC formation using Wi-Fi Direct, as a careful evaluation of MCC formation would require a real implementation. An implementation of MCC formation combined with collaborative content download is subject of our future work, and we refer to [25], [26] for a discussion of implementation issues for Wi-Fi Direct. In what follows we thus present simulation results concerning content download in an MCC.

To evaluate the performance of α-LSSO, we conducted extensive simulations, in which MNs in an MCC are randomly placed over the unit disc and have a communication range of 2 units. Consequently, all MNs in the MCC can communicate with each other directly, and we simulate multicast WLAN transmissions. The WLAN multicast data rate of each MN is determined depending on its maximum distance to the other MNs [27] and we consider the adaptive modulation and coding (AMC) scheme of the IEEE 802.11n standard [28] with a maximum data rate of 600Mbps. In terms of WWAN data rates, we assume the AMC scheme used in Long Term Evolution (LTE) [29]; the corresponding WWAN data rates of MNs are randomly selected with a maximum data rate of 300Mbps. The WWAN reception power is set to 1.8W and the WLAN transmission and reception powers are 0.925W and 0.425W, respectively [12]. We consider a content size of 5GB, which could correspond to a full high-definition (HD) or ultra HD (UHD) video encoded at a bitrate of 10Mbps or 14Mbps, and a length of approximately 1 hour [30]. The default number of MNs is 10, and the figures show the averages of 200 simulation runs. As a baseline for comparison, we consider three algorithms.

- **Equal chunk size (ECS) algorithm**: The chunk sizes are equal (i.e., C/N) and the sharing order follows an ascending order of the WWAN download time in the MCC. This algorithm does not optimize either the chunk size vector or the sharing order vector for minimizing the content download time or the energy consumption.
- **WWAN chunk size (WCS) algorithm**: The algorithm finds the optimal chunk size vector that minimizes the WWAN content download time (denoted by WCS_T) or the WWAN energy consumption (denoted by WCS_E), instead of the total download time or energy consumption. The order of sharing in the WLAN is chosen uniform at random.
- **Random sharing order (RSO) algorithm**: The algorithm chooses a sharing order vector uniform at random, it then finds the optimal chunk size vector by solving LP_{MCC}. We denote the solution for minimizing the content download time by RSO_T (i.e., $\alpha = 1$) and for minimizing the energy consumption by RSO_E (i.e., $\alpha = 0$).

We implemented the algorithms and computed the Pareto-optimal solution (i.e., the solution to the MINLP problem) using C++ in conjunction with a well-known LP solver, IBM ILOG CPLEX 12.6.

6.1 α-LSSO vs. Pareto optimal Solution

We first compare the optimal solution obtained by solving the MINLP problem (12) to the solution obtained by α-LSSO for the same value of α. In order to enable computing the optimal solution, we consider scenarios with up to $N = 10$ MNs forming an MCC, which is likely reasonable for most MCCs in practice. Table 3 shows the relative difference between the optimal solution and the α-LSSO solution. The results in Table 3 show that α-LSSO achieves near-optimal performance, as the relative differences are less than 3% for all values of N. In particular, for $\alpha = 0$ the solution achieved by 0-LSSO is optimal, because the energy consumption is rarely affected by the sharing order, which results in comparable LP solutions for chunk size vectors regardless of the sharing order vectors.

6.2 Effect of α

Figure 4 shows the content download time (left axis) and the energy consumption (right axis) as a function of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2616871, IEEE Transactions on Mobile Computing

TABLE 3
Relative difference between α-LSSO and Pareto optimal solution (unit : %).

α	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
$N = 5$	0.0	0.82	1.23	2.25	1.37	1.14	1.80	1.29	0.78	0.90	1.16
$N = 6$	0.0	1.75	0.88	1.20	1.83	1.97	0.66	1.71	1.43	0.51	0.95
$N = 7$	0.0	1.58	2.11	1.75	1.24	1.23	1.28	1.68	0.98	0.73	1.23
$N = 8$	0.0	0.63	0.89	1.57	1.22	1.80	2.42	2.13	1.29	1.33	1.51
$N = 9$	0.0	1.39	1.87	0.71	1.80	0.96	1.28	1.81	1.59	1.77	1.47
$N = 10$	0.0	2.28	1.76	1.43	1.28	1.29	2.13	1.83	1.72	1.48	1.56

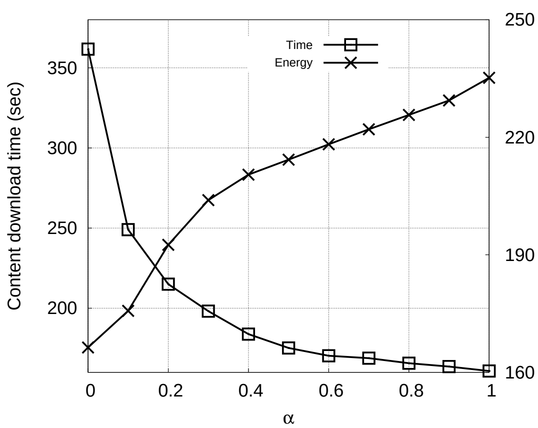


Fig. 4. Effect of the weighting factor α.

weighting factor α. The figure shows that α-LSSO can explore the trade-off between the content download time and the energy consumption in an MCC by choosing an appropriate α. Recall that as α approaches 1, the emphasis is given to minimizing the content download time, while for α close to 0, the energy consumption minimization becomes more important. We observe that the content download time decreases fast for small α, with a decreasing marginal gain, while the energy consumption increases almost linearly as a function of α. The reason why α affects the two quantities differently is that while the content download time is sensitive to the changes in the sharing order, the energy consumption is fairly insensitive.

Figure 4 also demonstrates that the proposed algorithm provides flexibility in terms of reduction of the content download time and reduction of the energy consumption depending on the choice of α. In particular, the content download time and the energy consumption can be well-balanced around $\alpha = 0.2$.

6.3 Effect of the number of MNs (N)

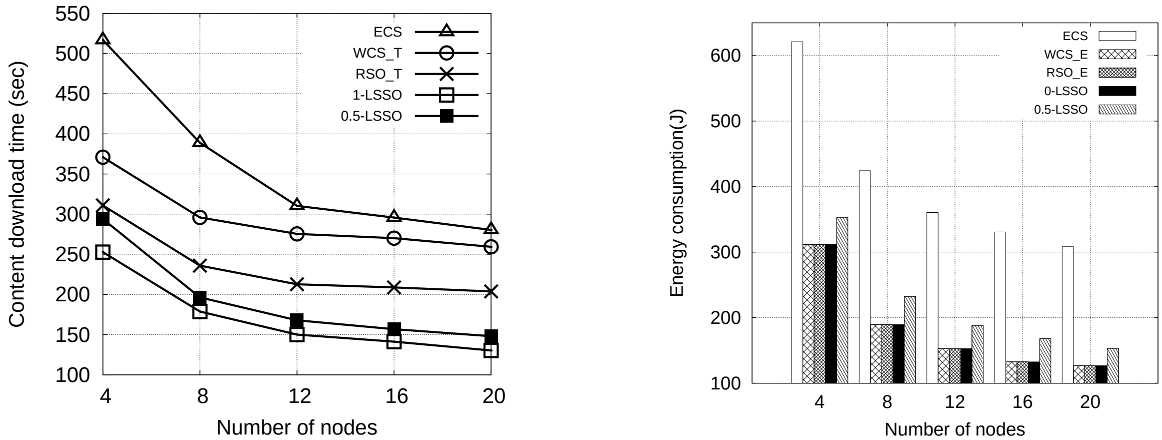
Figure 5(a) shows the content download time as a function of the number of nodes N, for $\alpha = 0.5$ and $\alpha = 1$. The figure shows that the content download time decreases in N with a decreasing marginal gain for all five considered algorithms. The reason is that as the number N of MNs increases, each MN in the MCC downloads a smaller chunk via the WWAN, which it then shares through the WLAN. Comparing 1-LSSO to the other algorithms, we observe a significant gain for all MCC sizes; 1-LSSO achieves a content download time 18.7–36.1% lower than RSO_T, 31.9–50.0% lower than WCS_T, and 51.1–53.6% lower compared to ECS.

Recall that WCS_T optimizes the chunk sizes only based on the WWAN data rates, regardless of the channel conditions in the WLAN, unlike 1-LSSO and RSO_T. Conversely, RSO_T only optimizes the chunk sizes but considering both the WWAN download time and the WLAN sharing time. While 1-LSSO approaches the optimal solution by repeatedly adapting the sharing order vector, RSO_T uses a randomly chosen initial sharing order vector, which explains its inferior performance. The importance of optimizing the sharing order is highlighted by the result that even 0.5-LSSO achieves better performance than RSO_T in spite of it catering to the conflicting requirements of minimizing the energy consumption and the download time. We can thus conclude that joint optimization of the chunk size and the sharing order is important for minimizing the content download time.

Figure 5(b) shows the energy consumption as a function of the number of MNs. The figure shows that the energy consumption decreases in N with a decreasing marginal gain, for all five algorithms. The decrease of the energy consumption is due to that as N increases, smaller chunks are assigned to each MN. It is interesting to note that the energy consumption of 0-LSSO is comparable with that of RSO_E and WCS_E regardless of N, which can be explained by that the total (and thus the average) energy consumption in the MCC is not significantly affected by the sharing order. Nonetheless, 0.5-LSSO results in a higher energy consumption than RSO_E and WCS_E, which shows that the objective of minimizing the content download time and that of minimizing the energy consumption are conflicting in general.

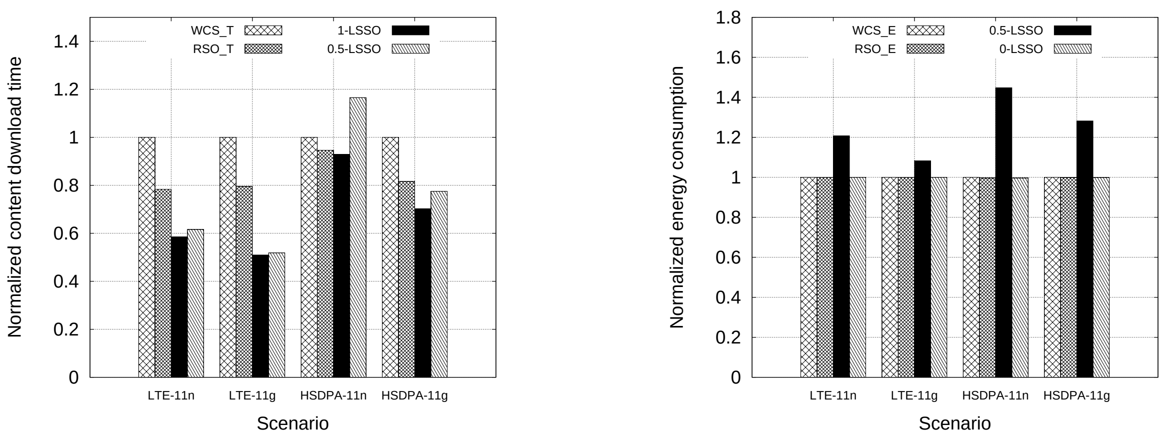
Interestingly, the results in Figure 5 show that the performance improvement becomes marginal as N increases. This can be explained as follows. As N increases, each MN downloads smaller chunks and thus the individual WWAN download times decrease. Nonetheless, the number of sharing rounds increases with N, which compensates the reduction the individual WWAN download times. As a result, the performance in an MCC

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2616871, IEEE Transactions on Mobile Computing



(a) Content download time (b) Energy consumption

Fig. 5. Effect of the number N of MNs.



(a) Content download time (b) Energy consumption

Fig. 6. Effect of the wireless bandwidth and technology.

becomes saturated as the number of MNs increases. We support this observation using analytical results shown in Appendix A.

6.4 Effect of Wireless Bandwidth

In order to analyze the performance of the algorithms with respect to the wireless bandwidth, we now consider two alternative wireless communication technologies to LTE and 802.11n considered so far. For the WWAN, we consider high speed downlink packet access (HSDPA), and assume AMC as in [31] with a maximum data rate of 14.4Mbps, which is less than that of LTE. For the WLAN, we consider IEEE 802.11g, with data rates determined depending on the distance [32] and using AMC as in [33], with a maximum data rate of 54Mbps, again less than that of 802.11n. Based on the 2 WWAN and 2 WLAN technologies, we define four scenarios: 1) LTE-IEEE 802.11n (denoted by LTE-11n), 2) LTE-IEEE 802.11g (denoted by LTE-11g), 3) HSDPA-11n, and 4) HSDPA-11g.

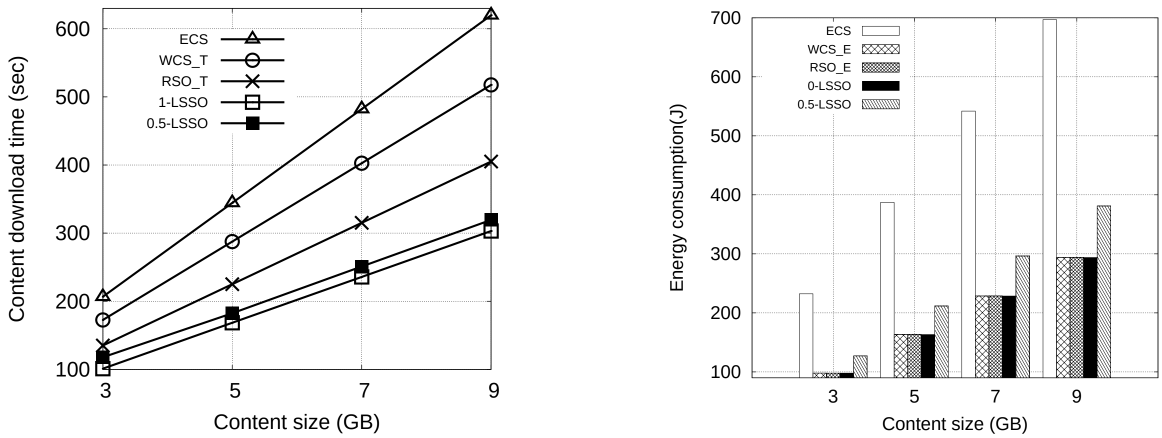
Figure 6 shows the content download time and the energy consumption for the four scenarios. All results are normalized by the content download time of WCS and the results of ECS are not shown due to extremely long content download time and high energy consumption.

Figure 6(a) shows that 1-LSSO achieves the best performance for all scenarios. It is important to note that the gain of 1-LSSO is highest for the scenario LTE-11g, when the WWAN data rates are high and the WLAN data rates are low. The reason is that if WLAN data rates are lower (e.g., 11g), the impact of the sharing order is more significant owing to the increased WLAN sharing time. Thus, the gain of the local search of the sharing order in 1-LSSO is significant.

When lower WWAN data rates are used (HSDPA), it is the chunk size vector (instead of the sharing order) that significantly affects the content download time. Since the chunk size vector is considered in all algorithms, the increased content download time due to lower WWAN data rates reduces the content download time differences between the algorithms.

Figure 6(b) shows the corresponding results for the energy consumption. The figure shows that the energy consumption of 0-LSSO is almost the same as that of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2616871, IEEE Transactions on Mobile Computing



(a) Content download time (b) Energy consumption

Fig. 7. Effect of the content size C.

WCS_E and RSO_E, for all scenarios. Recall that the energy consumption in the MCC is mostly affected by the chunk size vector, which is optimized by WCS_E and by RSO_E, which explains why the energy consumption results are similar for these algorithms and 0-LSSO. Meanwhile, 0.5-LSSO has the worst performance for all scenarios due to the consideration of content download time.

6.5 Effect of Content Size

Figure 7(a) shows the content download time as a function of the content size for five algorithms. The figure shows that the gain of 1-LSSO is significant and increases with the content size. Figure 7(b) shows the corresponding results for the energy consumption. We observe that 0-LSSO, RSO_E, and WCS_E have comparable energy consumption regardless of the content size, again due to the optimization of the chunk size vector.

From Figure 7, we can observe that the download time and the energy consumption scale linearly with respect to the content size. Recall that the download time and the energy consumption are a linear function of the content size for a given sharing order and relative chunk sizes. Thus, the optimal sharing order and chunk size vector would not depend on the content size. This observation could allow one to pre-compute the optimal sharing order and (normalized) chunk size vector for typical MCC sizes and channel conditions, and apply the best pre-computed solution depending on the actual scenario. The evaluation of the impact of such an approximate solution on the performance gains of MCC could be an interesting subject of future work.

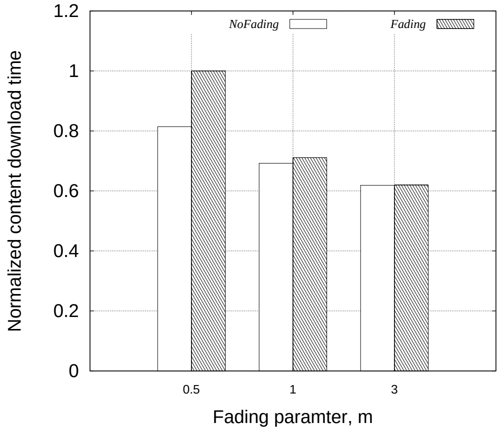
6.6 Effect of Dynamic Channel Conditions

In α-LSSO, the wireless channel conditions are assumed to be static during the collaborative download of the entire content, although in practice they are likely to change due to fading. In what follows, we investigate the performance of α-LSSO under dynamic channel conditions. To model dynamic channel conditions, we adopt the Nakagami-m model [34], which allows to capture different intensities of fading through the fading parameter m, and can be represented by a finite state Markov chain (FSMC) [35]. Each state in the FSMC corresponds to one data rate (i.e., a transmission mode) in multi-rate wireless networks. To compute the state transition probabilities defined in [35], we consider the signal-to-noise ratio (SNR) requirements on IEEE 802.16 (i.e., WiMAX) [37] for the WWAN and the requirements on 802.11a [36] for the WLAN. For a given Nakagami-m model, we use α-LSSO to determine the chunk size vector and the sharing order vector based on the average date rates. The simulations are then performed using traces of fading channels generated from the Nakagami-m model (denoted by $Fading$ in the figures). As a comparison, we show simulation results obtained using non-fading channels with the same average data rates (denoted by $NoFading$ in the figures).

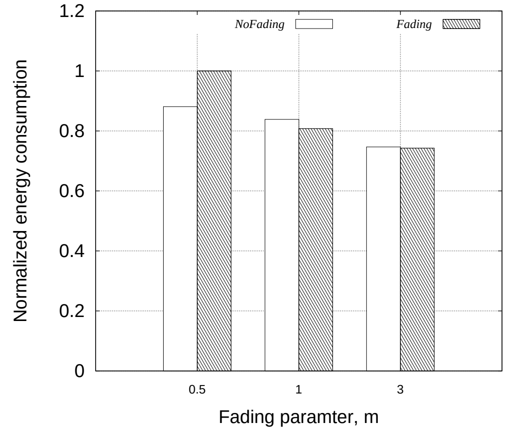
Figure 8 shows the content download time and the energy consumption as a function of m for 0.5-LSSO with 5 MNs[4]. All results are normalized by the content download time and the energy consumption of $Fading$ when $m = 0.5$. The results show that the relative differences of the content download time and of the energy consumption due to channel fading are highest for small values of m. Nonetheless, even for $m = 0.5$, the differences are within 25% and 15% for the download time and the energy consumption, respectively, which is still smaller than the difference between the results using different algorithms (c.f. Figs 5 to 7).

In the case of significant channel dynamics, α-LSSO could be executed in multiple stages. As an example, the CProxy can divide the content into multiple sub-contents and can then execute α-LSSO for each sub-content based

4. The average received SNRs of the 5 MNs are set to $\{5.0, 9.0, 13.0, 17.0, 21.0\}\,dB$ in the WWAN and $\{8.0, 10.0, 12.0, 14.0, 16.0\}\,dB$ in the WLAN.



(a) Content download time



(b) Energy consumption

Fig. 8. Effect of m ($\alpha = 0.5$).

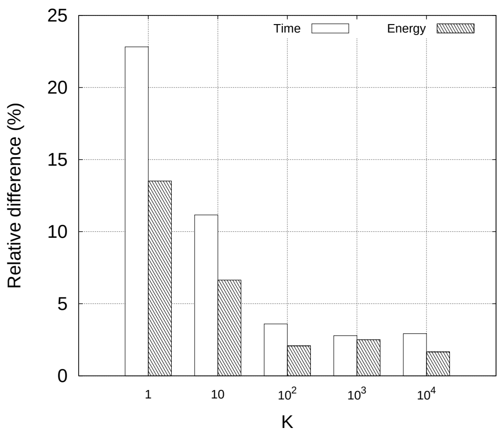


Fig. 9. Effect of K ($m = 0.5$).

on the predicted average channel conditions. Intuitively, as the number of stages increases, the channel conditions used for the computations can become more accurate, but the overhead for α-LSSO increases owing to frequent computations. Figure 9 shows the relative differences of content download time and energy consumption between $NoFading$ and $Fading$ as a function of the number of stages (denoted by K) when $m = 0.5$. The results confirm that the impact of channel fading decreases as K increases. Thus, if frequent channel state predictions are available, then it may be beneficial to execute α-LSSO in stages for the download of large contents. The optimal the number of stages depends on the characteristics of fading, e.g., due to mobility, and its optimization could be an interesting topic of future research.

7 CONCLUSION

In this paper, we introduced a proxy-based collaboration system where Wi-Fi Direct is used for the distributed MCC formation with chunk sharing and a CProxy performs the scheduling and the management for the MCC in the centralized manner with chunk distribution. The system combines the advantages of the distributed and of the centralized approaches as a hybrid approach, and can be realized by means of emerging technologies such as SDN and NFV. We formulated the scheduling problem at the CProxy as a multi-objective optimization problem to minimize the content download time and the energy consumption in an MCC by choosing the optimal chunk size and sharing order. We transformed the multi-objective optimization problem into an MINLP problem with a single-objective, and proposed a heuristic algorithm, α-LSSO, with low computational complexity. Simulation results demonstrate that α-LSSO achieves near-optimal performance and can significantly reduce the content download time and has comparable energy consumption compared with other algorithms depending on α while allowing to explore the trade-off between download time and energy consumption. In our future work, we will consider advanced MCC services accounting for the MNs' mobility and will extend MCC to vehicular environments.

ACKNOWLEDGEMENT

This work was supported in part by Korea Evaluation Institute of Industrial Technology (KEIT, Grant Number: 10051306), in part by National Research Foundation (NRF)/Swedish Foundation for Strategic Research (SSF) through the the Modane project (NRF-2014R1A2A1A12066986).

REFERENCES

[1] I. Jang, D. Suh, and S. Pack, "Minimizing Content Download Time in Mobile Collaborative Community," in *Proc. IEEE ICC*, June 2014.
[2] L. Zhou, M. Chen, Y. Qian, and H.-H. Chen, "Fairness Resource Allocation in Blind Wireless Multimedia Communications," *IEEE Transactions on Multimedia*, vol. 15, no. 4, pp. 946–956, June 2013.
[3] L. Tu and C. M. Huang, "Collaborative Content Fetching Using MAC Layer Multicast in Wireless Mobile Networks," *IEEE Transactions on Broadcasting*, vol. 57, no. 3, pp. 695–706, September 2010.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2616871, IEEE Transactions on Mobile Computing

[4] G. Ananthanarayanan, V. N. Padmanabhan, L. Ravindranath, and C. A. Thekkath, "COMBINE: Leveraging the Power of Wireless Peers through Collaborative Downloading," in *Proc. ACM Mobisys*, June 2007.

[5] J. Lee, J. Choi, and S. Bahk, "Opportunistic Downlink Data Delivery for Mobile Collaborative Communities," *Elsevier Computer Networks*, vol. 57, no. 7, pp. 1644–1655, May 2013.

[6] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, and A. Markopoulou, "Microcast : Cooperative Video Streaming on Smartphones," in *Proc. ACM Mobisys*, June 2012.

[7] P. Jassal, K. Yadav, A. Kumar, V. Naik, V. Narwal, and A. Singh, "Unity : Collaborative Downloading Content Using Co-located Socially Connected Peers," in *Proc. IEEE PERCOM*, March 2013.

[8] S. Kang and M. Mutka, "A Mobile Peer-to-Peer Approach for Multimedia Content Sharing Using 3G/WLAN Dual Mode Channels," *Wireless Communications and Mobile Computing*, vol. 5, pp. 633–645, September 2005.

[9] M.-F. Leung and S.-H. G. Chan, "Broadcast-Based Peer-to-Peer Collaborative Video Streaming Among Mobiles," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 350–361, March 2007.

[10] Z. Chang, T. Ristaniemi, and Z. Niu, "Energy Efficient User Grouping and Scheduling for Collaborative Mobile Cloud," in *Proc. IEEE ICC*, June 2014.

[11] L. Al-Kanj, Z. Dawy, W. Saad, and E. Kutanoglu, "Energy-Aware Cooperative Content Distribution Over Wireless Networks: Optimized and Distributed Approaches," *IEEE Transactions on Vehicular Technology*, vol.62, no. 8, pp. 3828–3847, October 2013.

[12] L. Al-Kanj, H. V. Poor, and Z. Dawy, "Optimal Cellular Offloading via Device-to-Device Communication Networks with Fairness Constraints," *IEEE Transactions on Wireless Communications*, vol. 13, no. 8, pp. 4628–4643, August 2014.

[13] Y. Wu, J. Chen, L. P. Qian, J. Huang, and X. Shen, "Energy-Aware Cooperative Traffic Offloading via Device-to-Device Cooperations: An Analytical Approach," *IEEE Transactions on Mobile Computing*, Early Access.

[14] WiFi Alliance, "P2P Technical Group, Wi-Fi Peer-to-Peer (P2P) Technical Specifications v1.5," August 2014.

[15] R. T. Marler and J. S. Arora, "Survey of Multi-objective Optimization Methods for Engineering," *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, April 2004.

[16] G. A. Croes, "A Method for Solving Traveling Salesman Problem," *Operations Res.6*, pp.791–812, 1958.

[17] Bruno Astuto A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *IEEE Communication Surveys & Tutorials*, vol. 16, no. 3, pp. 1617–1634, Third Quarter 2014.

[18] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network Function Virtualization: Challenges and Opportunities for Innovations," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 90–97, February 2015.

[19] ETSI ISG NFV, "Network Functions Virtualisation (NFV) - Network Operator Perspectives on Industry Progress," White Paper [Online]. Available:http://portal.etsi.org/NFV/NFV_White_Paper2.pdf.

[20] ETSI MEC Working Group, "Mobile Edge Computing (MEC) Technical Requirements," ETSI GS MEC 002, V1.1.1 (2016-03) [Online] Available:http://www.etsi.org/deliver/etsi_gs/MEC/001_099/002/01.01.01_60/gs_MEC002v010101p.pdf

[21] L. Al-Kanj, Z. Dawy, and E. Yaacoub, "Energy-Aware Cooperative Content Distribution over Wireless Networks: Design Alternatives and Implementation Aspects," *IEEE Communication Surveys & Tutorials*, vol. 15, no. 4, pp. 1736–1760, Fourth Quarter 2013.

[22] M. R. Garey and D. S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," New York: Freeman, 1979.

[23] M. Englert, H. Röglin, B. Vocking, "Worst Case and Probabilistic Analysis of the 2-Opt Algorithm for the TSP," *Algorithmica*, vol. 68, no. 1, pp. 190–264, January 2014.

[24] N. Karmarkar, "A New Polynomial-Time Algorithm for Linear Programming," *Combinatorica*, vol. 4, no. 4, pp. 373–395, 1984.

[25] D. Camps-mur, A. Garcia-saavedra, and P. Serrano, "Device-to-Device Communications with WiFi Direct: Overview and Experimentation," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 96–104, June 2013.

[26] H. Zhang, Y. Wang, and C. C. Tan, "WD2: An Improved WIFi-Direct Group Formation Protocol," in *Proc. ACM Mobicom Workshop on Challenged networks*, September, 2014.

[27] Juniper Networks White paper: Coverage or Capacity-Making the Best Use of 802.11n, 2011.

[28] IEEE 802.11n, Draft 2.0, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Enhancements for Higher Throughput, February 2007.

[29] "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Layer Procedures," ETSI TS 136 213 V8.8.0 (2009-10).

[30] G. J. Sullivan and J. R. Ohm, "Recent developments in standardization of high efficiency video coding (HEVC)," in *Proc. 33rd SPIE Applications of Digital Image Processing*, August 2010.

[31] "3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Physical layer procedures (FDD) (Release 8)," 3GPP TS 25.214 V8.9.0 (2010-03).

[32] Cisco Systems White Paper: Capacity, Coverage, and Deployment Condierations for IEEE 802.11g, 2005.

[33] IEEE 802.11g, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Further Higher Data Rate Extension in the 2.4 GHz Band, June 2003.

[34] G. L. Stüber, *Principles of Mobile Communication*, 2nd Edition, Norwell, Massachusetts: Kluwer, 2001.

[35] Q. Liu, S. Zhou, and G. B. Giannakis, "Queuing With Adaptive Modulation and Coding Over Wireless Links: Cross-Layer Analysis and Design," *IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 1142-1153, May 2005.

[36] Q. Liu, S. Zhou, and G. B. Giannakis, "Cross-Layer Combining of Adaptive Modulation and Coding With Truncated ARQ Over Wireless Links," *IEEE Transactions on Wireless Communications*, vol. 3, no. 5, pp. 1746–1755, September 2004.

[37] IEEE 802.16, Part 16: Air Interface for Broadband Wireless Access Systems, May 2009.

Insun Jang received the B.S. degree from Korea University, Seoul, Korea, in 2011. He is currently an M.S. and Ph.D. integrated course student in School of Electrical Engineering, Korea University, Seoul, Korea. His research interests include content delivery networks, mobile cloud networking, SDN/NFV, Future Internet, and vehicular networks.

Gwangwoo Park received the B.S., M.S., Ph.D degrees from Korea University, Seoul, Korea, in 2010, 2012, and 2016, respectively. He is working at Samsung Electronics. His research interests include content networking, multimedia streaming, information-centric networking, and Future Internet.

Dongeun Suh received the B.S. degrees from Korea University, Seoul, Korea, in 2012. He is currently an Ph.D. course student in School of Electrical Engineering, Korea University, Seoul, Korea. His research interests include HTTP-based adaptive multimedia streaming, delay-tolerant networking, and software-defined networking.

Sangheon Pack received the B.S. and Ph.D. degrees from Seoul National University, Seoul, Korea, in 2000 and 2005, respectively, both in computer engineering. In 2007, he joined the faculty of Korea University, Seoul, Korea, where he is currently a Professor in the School of Electrical Engineering. From 2005 to 2006, he was a Postdoctoral Fellow with the Broadband Communications Research Group, University of Waterloo, Waterloo, ON, Canada. He was the recipient of KICS (Korean Institute of Communications and Information Sciences) Haedong Young Scholar Award 2013, IEEE ComSoc APB Outstanding Young Researcher Award in 2009, and LG Yonam Foundation Overseas Research Professor Program in 2012. He was a publication co-chair of IEEE INFOCOM 2014/ACM MobiHoc 2015, a co-chair of IEEE VTC 2010-Fall transportation track, a co-chair of IEEE WCSP 2013 wireless networking symposium, a TPC vice-chair of ICOIN 2013, and a publicity co-chair of IEEE SECON 2012. He is an editor of Journal of Communications Networks (JCN) and a senior member of the IEEE. His research interests include Future Internet, SDN/ICN/DTN, mobility management, mobile cloud networking, multimedia networking, and vehicular networks.

György Dán is an associate professor at KTH Royal Institute of Technology, Stockholm, Sweden. He received the M.Sc. in computer engineering from the Budapest University of Technology and Economics, Hungary in 1999, the M.Sc. in business administration from the Corvinus University of Budapest, Hungary in 2003, and the Ph.D. in Telecommunications from KTH in 2006. He worked as a consultant in the field of access networks, streaming media and videoconferencing 1999-2001. He was a visiting researcher at the Swedish Institute of Computer Science in 2008, a Fulbright research scholar at University of Illinois at Urbana-Champaign in 2012-2013, and an invited professor at EPFL in 2014-2015. His research interests include the design and analysis of content management and computing systems, game theoretical models of networked systems, and cyber-physical system security in power systems.