# Content-Aware D2D Caching for Reducing Visiting Latency in Virtualized Cellular Networks

**Guolin Sun[1*], Hisham Al-Ward[1], Gordon Owusu Boateng[1] and Wei Jiang[2,3]**
[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China Chengdu-China
[2] German Research Center for Artificial Intelligence (DFKI GmbH), Kaiserslautern, Germany
[3] Department of Electrical and Information Technology (EIT), Technische University (TU) Kaiserslautern, Germany
[e-mail: guolin.sun@uestc.edu.cn]
*Corresponding author: Guolin Sun

*Received August 7, 2018; revised September 17, 2018; accepted September 22, 2018; published February 28, 2019*

## Abstract

Information-centric networks operate under the assumption that all network components have built-in caching capabilities. Integrating the caching strategies of information centric networking (ICN) with wireless virtualization improves the gain of virtual infrastructure content caching. In this paper, we propose a framework for software-defined information centric virtualized wireless device-to-device (D2D) networks. Enabling D2D communications in virtualized ICN increases the spectral efficiency due to reuse and proximity gains while the software-defined network (SDN) as a platform also simplifies the computational overhead. In this framework, we propose a joint virtual resource and cache allocation solution for latency-sensitive applications in the next-generation cellular networks. As the formulated problem is NP-hard, we design low-complexity heuristic algorithms which are intuitive and efficient. In our proposed framework, different services can share a pool of infrastructure items. We evaluate our proposed framework and algorithm through extensive simulations. The results demonstrate significant improvements in terms of visiting latency, end user QoE, InP resource utilization and MVNO utility gain.

***Keywords:*** resource allocation; caching strategy; latency reduction; software-defined networking; information-centric networking; wireless D2D communication.

## 1. Introduction

The world today is witnessing a rapid growth of mobile traffic. This unprecedented development is accompanied with an ever-increasing demand from the end user for an improved quality of experience (QoE) and faster data delivery. The latest Cisco visual networking index reports that mobile data traffic has grown 18-fold between 2011 and 2016 and predicts that the global mobile data traffic will increase sevenfold and reach 49.0 exabytes per month by 2021 [1]. This dramatic rise in traffic inevitably leads to congestion of backhaul networks, which in turn results in a higher cost of operation and maintenance, a lower quality of service and a low-speed data delivery. The gaps between the expected and the feasible performances are too huge for the current 4G technologies and the traditional IP networks to fill on their own, and thus new solutions need to be sought.

One innovative proposal to overcoming these challenges is content caching. When the content is placed somewhere closer to the end user, the amount of data traveling through the network is reduced, and the visiting time is proportionally decreased as well. The problem of optimal cache placement has been thoroughly studied in academia [2][3], and the practical implementations of distributed web proxy caches have been in use successfully for the last two decades. The success of the caching solutions in delivering services with better delivery performance and less in-network load have led to the creation of a completely new networking paradigm, information centric networking (ICN). ICN places more value on the name of a certain content rather than the actual location of the content and incorporates inherent in-network caching capabilities for network components. That is, when a request is made for a content, the ICN network searches for the content using its unique name and connects the requesting user to the intermediate node with a cached copy of this content [4][5]. ICN-based air caching is thought of as a promising technique in the context of 5G wireless networks [6].

Network function virtualization (NFV) and software defined networking (SDN) can further improve the performance of ICN. NFV is the decoupling of the network infrastructure from the services it provides [7]. By means of NFV, the physical infrastructure can be divided and abstracted into isolated substrate resources called virtual network slices [8]. This enables the infrastructure providers (InPs) to maximize their revenue and utilization of the infrastructure by leasing these virtual network slices out to many mobile virtual network operators (MVNOs). It also helps the service providers by reducing their capital and operational investments since they are sharing the same infrastructure. Similarly, if the InP owns a wireless network infrastructure, the scheme can be referred to as wireless network virtualization [9]. SDN, on the other hand, is the decoupling of the control plane of the network (i.e. the decision mechanism) from its data plane (i.e. the forwarding mechanism) [10]. Enabling a dynamic reconfiguration of the network, along with the separation of control and data planes, makes it a promising platform for many emerging technologies, including information-centric virtualization [11][12].

Device-to-device (D2D) communication which is a core concept of 5G can increase the spectral efficiency by spectrum reuse in an overlay or underlay manner in a cellular network. D2D communication has attracted a lot of interests in smart cyber-physical systems from academia [13] and various standardization bodies [14] because of its numerous advantages. D2D communication can be defined as the direct data exchange between any two mobile users without the assistance of the base station (BS). D2D communication has the advantage of introducing both reuse and proximity gains [15]. Though popular as an independent field of study, including D2D communications within the ICN scheme is yet to gain momentum.

In this paper, not only do we aim at achieving the best possible reduction for the end user's content visiting time, but also we seek to allow the infrastructure providers and mobile

operators to achieve satisfactory gains. Our contributions can be summarized as follows:

- We propose a SD-enabled information-centric virtual wireless D2D network framework for latency-sensitive applications.
- We formulate a joint optimization problem, which comprises virtual resource slicing and cache allocation, and by means of wireless virtualization, we allow mobile users of different MVNOs to communicate and exchange content.
- In order to avoid the overhead complexity of distributed algorithms like the alternating direction method of multipliers (ADMM) [16], we designate the SDN controller to solve the formulated problem, and in order to reduce the in-network travel time, we delegate the resulting caching tables along with the mapping responsibilities to the BS.
- We formulate four algorithms, which are the combination of two known caching approaches and two caching strategies.
- Extensive simulations are conducted to verify the significance of the proposed work. It is demonstrated that the content visiting time, end user satisfaction, InP resource utilization, and MVNO utility gain are greatly improved under the proposed scheme.

The rest of the paper is organized as follows: Section 2 outlines the related work, while Section 3 demonstrates the proposed system model. Section 4 focuses on problem formulation and algorithms. Section 5 discusses the experimental results of the proposed solutions. We finally conclude the paper in Section 6

## 2. Related Work

In-network caching has been identified as a promising enabling technology for the next generation cellular networks [17]. The idea of caching the popular content at the small BSs in order to reduce congestion in backhaul networks was first introduced in [18]. In [19], a content placement framework was also studied. The authors in [20] estimated content popularity with the multi-armed bandit theory and developed a collaborative framework in which content is shared among small cell BSs. These three works share the same assumption of static users and did not account for the users' own caches. The work in [21] explored the concept of dynamic femto-caching, which takes into account the mobility of users. Recently, many works have focused on developing strategies for caching at end users and allowing D2D communications. In [22], for instance, an optimal distance for the D2D communication was defined in a simple network with one BS, while [23] extended it to consider virtually grouping users into clusters that allow intra-cluster sharing. Furthermore, the work in [24] took the proposed work a step further and assumed the co-existence of both cache-assisted BSs and users.

Integrating virtualization with ICN was proposed in [6] to enhance the end-to-end performance. The authors proposed an information-centric wireless virtualization scheme that enables the sharing of both infrastructure and content among different providers. In [25], the authors advanced the scope of [6] to incorporate D2D communications. Their system was virtualized and allowed for multiple MVNOs to share the content, and by developing a distributed algorithm with ADMM, they showed an improved MVNO utility gain and reduced backhaul usage. This work was extended in [26], where the authors added SDN functionalities to the proposed network. They formulated the virtual resource allocation and caching optimization as a discrete stochastic optimization problem and evaluated the system in terms of total MVNO utility and reduced backhaul usage.

The reduction of visiting latency in D2D networks was only implied in the related works but was not tested for. In this paper, we are largely motivated to exhibit how the different QoS

latency requirements for different MVNOs are met under our proposed schemes. The solutions we provide in this paper are much simpler and more intuitive than that of [26], but they nevertheless perform superbly and satisfy the different QoS latency requirements. Additionally, unlike the works in [25][26], we evaluate our performance results against those from non-caching as well as caching at the BS scenarios.

It is common in literature to categorize the caching action as taking place either after a user initiates a request [3] or before it, assuming the user's traffic is predictable [3] [19]. We refer to the first approach as the pull caching approach and the second as the push caching approach, or pre-caching. Instead of choosing one approach over the other, we incorporate both frameworks into the building of our algorithms and evaluate their performances in a D2D context. Additionally, we also try two different cache allocation strategies; a greedy and a newly designed load-aware balanced allocation (LABA). By combining these two strategies with the two approaches, we obtain four algorithms for caching content to the D2D network that will be tested.

## 3. System Model

In this section, we address the business model before elaborating on the network model. A discussion on the delay model concludes this section.

### 3.1. Business model

In this paper, we adopt the three-layer business model, which includes service providers (SPs), MVNOs, and InPs [6]. In our scenario, as illustrated in **Fig. 1**, the SP owns only the original content and needs it to reach a target audience, while the InP owns the underlying infrastructure (e.g. mobile devices with caching capabilities, BS, core network entities (CNEs) and radio access network (RAN) but has no interest in creating or managing the content The MVNO in this sense is the party that takes the content from the SP and utilizes the InP's resources in order to deliver a service to the clients. The three parties need one another for the successful delivery of a service and, naturally, the making of financial profit. In our scenario, one MVNO can manage the original content of one or more SPs as long as they provide the same unique type of content (e.g. video). Similarly, one MVNO can contract one or more InPs to improve its service reachability if it deems fit. The MVNO pays the InP for using the network and spectrum infrastructure on a charge-by-use basis and makes a profit by charging end users on a QoE-conditioned charge-by-use basis. The InP makes its profit by leasing out its infrastructure to MVNOs. However, as the charging is on a charge-by-use basis, not a fixed quota, it is very important for the InP to ensure that its resources are fully utilized at all times since idle resources means a loss of profit. The success of this virtualization scheme is judged by the ability to deliver the negotiated service level agreements (SLAs).

### 3.2. Network model

We consider a network with $S$ MVNOs in a given geographical area. We also have a set of mobile devices $M = \{1,2,\dots,|M|\}$, $m \in M$, and a set of base stations $K = \{1,2,\dots,|K|\}$, $k \in K$, which is the set of the super users. The mobile devices and the super users together make a set of nodes, $N = \{1,2,\dots,|N|\}$, $n \in N$ and maintain $M, K \in N$. In addition, the set of MVNOs, in terms of slices, $S = \{1,2,\dots,|S|\}$, $s \in S$, and the set of mobile users exclusively belonging to one slice is denoted as $U_s = \{1,2,\dots,|U|\}$, $u \in U_s$. All the slices share this common infrastructure. In this paper, we consider the single-cell scenario consisting of an area with the entire infrastructure owned by a single InP; however, the principles can be extended and applied to multiple InPs environments and scenarios. In our scenario, the InP owns the

bandwidth spectrum, some core network infrastructure equipment, and cache-enabled infrastructure devices. As the focus of this paper is on D2D communications, we assume there is only one BS, and the set of BSs is reduced to having one element, $k$. It is of highest importance in our paper to note the difference between a mobile device and mobile user. A *mobile device* refers to the caching capabilities available at the D2D level and owned by the InP, which is a potential transmitter. On the other hand, a *mobile user* is the requesting user that belongs to a MVNO. In this sense, the D2D communication denotes the data transfer from a mobile device $m$, to a mobile user $u$.

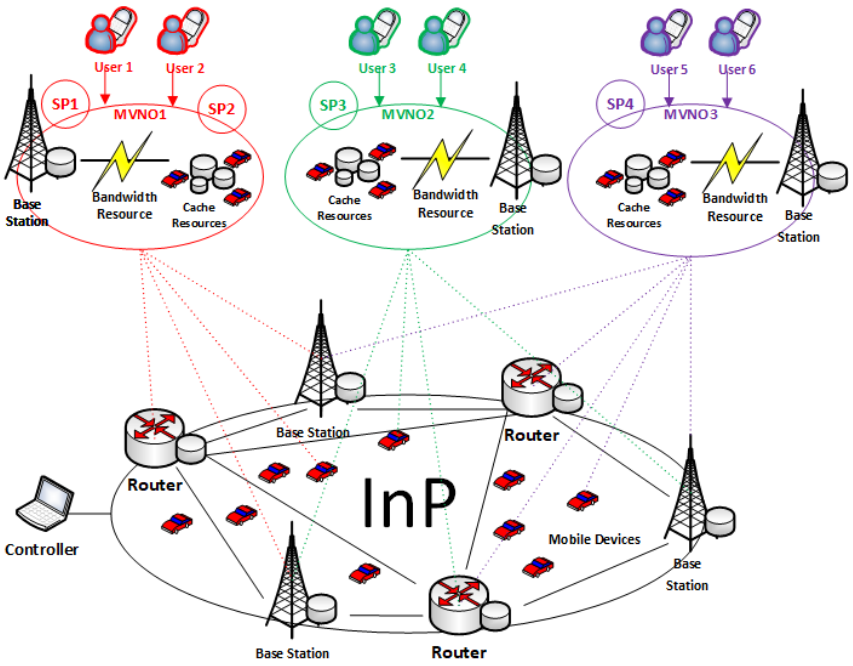



**Fig. 1**. System model

Let the content library, which is all the contents available by all the slices be $F = \{f_1, f_2, \dots, f_s\}$, where $f_i$ is the $i^{th}$ content file which was provided by the $i^{th}$ slice. The content file is further divided into equal and fix-sized segments called objects $o$ such that $f_i = \{o_1, o_2, \dots, o_j\}$, where $o_i$ is the last object in the $i^{th}$ content file. Correspondingly, the caching capacity at the node is represented with $C_n$. This caching capacity is divided into equal and fix-sized segments called slots. All mobile devices have the same total caching capacity, and the size of a slot is identical to that of an object $o$. Cache allocation occurs when an object occupies a caching slot. We assume $Z_{on}(s)$ to be a caching binary variable; such that $Z_{on}(s) = 1$ indicates that object $o$ of slice $s$ is cached on a node $n$ and 0 otherwise.

In this paper, an orthogonal spectrum scheme is considered to decrease the complexity of analyzing mutual interference between the cellular and D2D transmission modes [25]. As such, the bandwidth for a general node $n$ is $BW_n$, which can be divided into two orthogonal parts, $BW_c$ for the cellular communications and $BW_{d2d}$ for the D2D communications.

Let $X_{un} \in \{0,1\}$ be a binary association variable, where $X_{un} = 1$ denotes that the user $u$ is associated with the node $n$, be it a mobile or a super user, and $X_{un} = 0$ otherwise. The D2D communications are all assumed to be half-duplex in our system [27] and are expressed mathematically as;

$$\sum_{u \in U_s, s \in S} X_{um} \leq 1, \forall m \in M \,. \quad (1)$$

Let $Y_{un} \in [0,1]$ be the fraction of system bandwidth allocated to the requesting user $u$, which could be used to formulate the following constraints;

$$\sum_{u \in U_s, s \in S} X_{uk} Y_{uk} \leq 1, \quad (2)$$

$$\sum_{u \in U_s, s \in S, m \in M} X_{um} Y_{um} \leq 1 . \quad (3)$$

The constraint in equation (2) ensures that the sum of fractions of system bandwidth occupied by all of the mobile users associated with the BS $k$ does not exceed 1. Similarly, the constraint in equation (3) safeguards the sum of fractions of D2D communication bandwidth used between the mobile devices from exceeding 1.

In order to measure the performance in our simulation, we first calculate the effective transmission rate of a user before converting it into visiting latency in time unit. A user's effective transmission rate $R_{un}$ can be expressed as;

$$R_{un} = \sum_{n \in N} X_{un} Y_{un} BW_n r_{un} , \quad (4)$$

where $r_{un}$ is Shannon's achievable spectrum efficiency, which can be calculated from the expression $r_{un} = \log_2(1 + \gamma_{un})$. $\gamma_{un}$ is the SINR model and can be obtained from the expression

$$\gamma_{un} = {g_{un} P_n} \Big/ {\left(\sum_{k,k \neq n} g_{un} P_k + \sigma\right)} , \quad (5)$$

where $g_{un}$ is the large-scale channel gain, $\sigma$ is the power spectrum density of AWGN and $P_n$ is the received signal power of user $u$ from transmission node $n$. The channel gain can be obtained by considering the path loss as;

$$PL = 20\ log_{10}(D) + 20\ log_{10}(F) + 32.4 , \quad (6)$$

where $D$ is the distance between the reciever and transmitter in $km$ and $F$ is the frequency band in $MHz$. The shadowing small scale fading effect is assumed to be a Gaussian random variable with zero mean and a standard deviation $\delta$ of 8dB.

Finally, in the SD-enabled system, the controller runs the algorithms based on the data collected and creates the cache allocation table. The mobile users do not have a global view of the D2D network and will have to consistently consult the BS on which communication channels to establish. As each mobile device has two communication interfaces; one for cellular and the other for D2D communications, reaching the cellular network for instructions does not conflict with the D2D communications. In order to reduce visiting latency, we minimize the in-network communication time between the infrastructure devices and the controller by delegating the mapping tasks to the super user. In addition to its duties of matching requests to objects, the super user can also be used for data transfer only as a last resort (i.e. if and only if no D2D interface is available or if the content object is not cached in any mobile user within the range of the requesting user).

### 3.3. Delay model

In this paper, we break down the time needed for the content to reach the requesting user into three phases, as shown in **Fig. 2**: 1) *radio access delay:* the delay caused by crossing the wireless channel and queueing to access the highly-in-demand BS, 2) *in-network visiting and downloading delay:* the time it takes for the data to travel back and forth through the backhaul network before reaching the BS closest to the requesting user, and 3) *wireless content delivery delay:* the time needed to offload the content from the closest BS to the requesting user. We tackle each type of delay and finally formulate a universal delay expression.

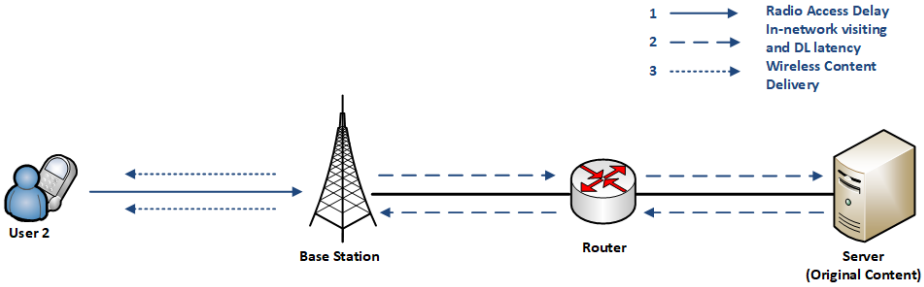



**Fig. 2.** The delay model

First, we study the delay caused by the mobile users attempting to connect to the BS. We assume the population of $U$ users randomly selceting $\mu$ preambles. The author in [28] assumed the average time between random access opportunities (RAOs) as $\tau$ and the maximum rate of successful departures from the channel as $\varepsilon = \frac{\mu}{\tau e}$, which is referred to as the channel capacity and $e$ is Euler's constant. From this, we can consider the expected delay for simultaneous arrival of nodes as the radio access delay as;

$$t_R = \frac{U}{2\varepsilon}. \quad (7)$$

Secondly, the in-network visiting and downloading delay is a function proportionally related to the number of (wired backhaul network) hops the data travels through. By assuming the number of hops to be $h$ and the delay time for each hop to be fixed as $T$, we obtain the in-network delay as;

$$t_I = (T * h_{uo}), \quad (8)$$

where $h_{uo}$ is the number of hops between the user $u$ and the original object $o$.

Finally, we discuss the means of calculating the transmission delay of a wireless link in 3.2 knowing that we first have to obtain the effective transmission rate. We use equation (4) and substitute node $n$ with BS $k$ to get the following effective transmission rate;

$$R_{uk} = X_{uk} Y_{uk} BW_k r_{uk}. \quad (9)$$

By considering equation (9), we can apply the methods outlined in Section 2 and the object size to find the wireless content delivery delay, denoted as $t_{uk}$. Combining the three delay components above, the total visiting time for a user $u$ to obtain an object $o$ from its original content provider can be expressed as;

$$t_{uo} = t_R + (T * h_{uo}) + t_{uk} \quad (10)$$

This view of the delay model is intentionally universal as it does not account for the effect of caching. The effects of caching and the modifications it incurs on the delay model are examined in Section 4 when the two caching approaches we are adopting in this paper are introduced. That said, this delay model in its current form is sufficient enough to develop the latency-sensitive QoE measurement metric that we will use in this study. We assume the content objects have equal size of $b_o$, and the user transmission rate for each slice can be translated into an equivelant downloading time metric. As such, we first express the satisfaction $\xi$ of a user $u$ with a user transmission rate $R_u$ with the following sigmoid function;

$$\xi(R_u) = \frac{1}{1+e^{-\beta(R_u - R_S^{min})}}, \quad (11)$$

where $\beta$ is a constant to determine the steepness of the curve and $R_s^{min}$ is the minimum transmission rate for all users in slice $s$ and $R_u$ is the average transmission rate of user $u$. Then, by considering the latency constraint $t_{uo} \leq q_s, \forall u \in U, \forall o \in O, \forall s \in S$, we create a corresponding latency-sensitive utility function of $t_{uo}$ as our primary QoE metric, which can be expressed as the user satisfication $\xi(t_{uo})$ for a user $u$ of slice $s$ as;

$$\xi(t_{uo}) = \frac{1}{1+e^{-\beta(q_s - t_{uo})}}, \quad (12)$$

where $\beta$ is the stepness constant, $d_{uo}$ is the minimum visiting time for user $u$ of slice $s$ for obtaining object $o$, and $q_s$ is the time delay requirement by slice $s$. The overall satisfaction by all users belonging to all slices, which is also our objective function, is expressed as;

$$\sum_{s \in S} \sum_{u \in U} \sum_{o \in f_s} \xi(t_{uo}) . \quad (13)$$

The utility function for the MVNO is the net difference between revenue generated from the end users and expenses paid to the InP. Let $Sat_s$ be the number of satsified end users for a given slice $s$, $SF_s$ the subscription fee that users pay, and $\alpha$ the cost of renting one $b_0$ slot of caching storage from the InP. The utility function for a given slice is then calculated as;

$$Ut(s) = (Sat_s \cdot f_s \cdot SF_s) - (\lambda_s \cdot \alpha), \quad (14)$$

where $\lambda_s$ denotes the total number of a slice's objects cached in the node and is calculated by $\lambda_s = \sum_{n \epsilon N} Z_{on}(s), \forall o \epsilon f_s$.

## 4. Problem Formulation

### 4.1. Problem Optimization

In this section, we formulate an optimization problem for multiple slice resource allocation in virtualized wireless D2D networks, and discuss the resource allocation constraints. We uphold the assumption that each slice has its own QoS requirement in terms of visiting time latency, denoted as $q_s$. For a lower visiting latency time, we attempt to cache the content primarily on the D2D network, with the cellular network only used as a last resort to maintain continuity of service. In equation (15), we aim at maximizing the satisfaction of any user $u \epsilon U_s$ of any slice $s \in S$ requesting any object $o \in f_s$ from either a neighboring mobile device $m \in M$ or the super user $k$ as;

$$max \sum_{s \epsilon S} \sum_{m \epsilon M} \sum_{u \epsilon U_s} \sum_{o \epsilon f_s} \zeta(t_{um}) . Z_{om}(s) . p_{uo} . A_m + \sum_{s \epsilon S} \sum_{u \epsilon U_s} \sum_{o \epsilon f_s} \zeta(t_{uk}) . Z_{ok}(s) . p_{uo} . \vartheta_k , \quad (15)$$

$$t_{un} . Z_{on}(s) . p_{uo} \leq q_s , \forall u, n, o, s \epsilon U_s , N, f_s , \quad (16)$$

$$\sum_{o \epsilon f_s} \sum_{n \epsilon N} b_o Z_{on}(s) \leq l_s, \forall s \epsilon S , \quad (17)$$

$$\sum_{s \epsilon S} \sum_{o \epsilon f_s} b_o Z_{on}(s) \leq C_n, \forall n \epsilon N , \quad (18)$$

$$\sum_{n \epsilon N} Z_{on}(s) \leq \omega, \quad (19)$$

where $\xi(t_{un})$ is the satisfaction function of visiting time $t_{un}$, and $Z_{on}(s)$ is the content caching binary variable. $p_{uo}$ is the probability that a user $u$ requests an object $o$. The contents provided by the slices are diverse and their populatrity follows the Zipf popularity distribution; for any object $o \epsilon f_i$, $p_{uo} = \frac{1}{o^\delta} / \sum_{z \epsilon f_i} \frac{1}{z^\delta}$, where $\delta$ is the Zipf parameter. $A_m$ is the availability of the wireless D2D interface on the mobile device $m$, where $A_m = 1$ indicates that the mobile

device's interface is available at the moment, and $A_m = 0$ indicates that the mobile device is currently engaged in another connection (either as a transmittor or reciever), and thus its interface is currently unavailable. $\vartheta_k$ is a Boolean variable allowing or prohibiting the requesting of content objects from the super user, $k$. The variable is at a default value zero, i.e. $\vartheta_k = 0$, which means the mobile users cannot contact the super user for content delivery purposes. Only when no mobile device has the desired object ($\sum_{m \in M} Z_{om} = 0$), or when there are no available mobile devices ($\sum_{m \in M} A_m = 0$), can this Boolean variable be set to $\vartheta_k = 1$ and content be cached at and requested from the super user.

Constraint (16) guarantees that the visiting latency requirement of a slice $q_s$, is satisfied. Constraint (17) ensures that the total number of allocated objects of size $b_0$ does not exceed the pre-negotiated and leased capacity $l_s$. Constraint (18) ensures that a single node $n$ does not obtain more cache objects than its actual physical storage capacity, $C_n$. Constraint (19) allows a copy of content object $o$ to be cached on one or more nodes. The formulated problem is NP-hard; hence we develop some heuristic algorithms to solve such a problem.

## 4.2. Algorithms

In this paper, we utilize two different caching approaches and two different caching strategies to develop the cache allocation solutions. The approach can either be a pull or push, while the caching strategies can be either greedy or load-aware balanced allocation (LABA). The two caching approaches differ in the type of pre-calculation to be done at the beginning of the procedure and when to run the allocation algorithms. The two caching strategies, on the other hand, discuss the means of generating a caching table and allocating the content objects. The approaches and caching strategies are discussed further in 4.2.1 and 4.2.2, respectively. In combining these approaches and strategies, we have a total of 4 unique algorithms to test as illustrated in **Table 1**.

**Table 1.** The four proposed algorithms

| | **Cache allocation frameworks** | | |
|---|---|---|---|
| Cache Allocation Strategies | | *Pull* | *Push* |
| | *Greedy* | (1) Greedy Pull Algorithm | (3) Greedy Push Algorithm |
| | *LABA* | (2) LABA Pull Algorithm | (4) LABA Push Algorithm |

Though varied, these algorithms share the same concept of calculating and sorting out neighbor lists in their initialization stages. **Algorithm 1** demonstrates how this principal calculation is done. As illustrated in **Algorithm 1**, neighboring lists are created for each mobile user in each slice. The algorithm starts with initializing *userDist*, which is a dissimilarity matrix that records the distances between any two nodes within the coverage area of one super user. If the mobile device is within the D2D communication range (lines 5-6), the effective transmission rate and equivalent delay between this device and the mobile user are calculated. In lines 7~8, this rate and delay are compared to the slice's minimum required transmission rate and maximum time delay. If both satisfy the conditions, the $L$ value for this mobile user is increased. $L$ is the number of eligible users specified for mobile user. In lines 11~15, we place high values on the mobile devices that are outside the D2D range. When all the mobile users in one slice are evaluated, the list is sorted out in ascending order based on its $t_{um}$ values, meaning the mobile device with the lowest time delay tops the list and the device that is outside the range falls last.

**Algorithm 1: Creating the neighbor lists**

**Initialize** $userDist$ from the network status;
**for** $s \in \mathcal{S}$
  **for** $u \in \mathcal{U}_s$
    **for** $m \in M$
      **if** $0 < userDist(u,m) \leq Range_m$
        Calculate its $R_{um}$ and $t_{um}$ values and record the pair;
        **if** $R_{um} \geq R_s^{min}$ && $t_{um} \leq q_s$
          $Slice(s).Subscriber(u).L = ++1;$
        **end**
      **else**
        Set $t_{ud} = H;$
      **end**
    **end**
  **end**
  Create the neighbor struct $User(u).Neighbor(m)$ and sort it out based on the $t_{ud}$ values;
**end**

### 4.2.1. Caching Approaches

### A. The Pull Approach:

In this approach, the content object is cached on the InP network only after a user requests the said content. That is, the user will pull the content towards it. The caching is done around, but not directly to, the requesting users. **Approach 1** defines how the pull mechanism works for a given slice, while **Fig. 3** provides a visual illustration of it.

First, the initialization starts with the status of the network being collected by the super user and reported to the controller, which then calculates the eligibility metric and creates user and neighbor lists. In this framework, we require the pull approach-assisted algorithms to determine $Ft$, the number of the first users around which the content shall be cached. We allow for one copy object to be cached for each requesting user to allow the content to reach as many mobile users as possible. If this is not done, mobile devices with a high capacity close to the first users could consume all the purchased capacity and create hotspots that are out of the wireless D2D communication range for many mobile users. After initialization, the system waits for requests from mobile users. When the super user receives a request, it first checks the order of the requesting user for that slice. If the user ordinal number is higher than $\boldsymbol{\varepsilon}$, the super user consults its caching table and instructs the mobile user on how to get the desired content. On the other hand, if the requesting user is one of the first requesting users ($Ft$), the super user reports back to the controller. The controller then updates the caching table and allows for more cache copies to be allocated to the infrastructure devices. Once the caching table is updated, the super user is then able to instruct the mobile user on where to get the content.

The delay model for the pull approach differs from the universal delay model in (10) and is reformulated as;

$$\begin{matrix} First\ requesting\ users \\ Subsequent\ user \end{matrix} \quad \begin{matrix} t_{uo} = t_R + (T * h_{uo}) + t_{cache} + t_{um} \\ t_R + t_{um} \end{matrix} \tag{20}$$

where $t_{cache}$ is the delay time for caching the content from the BS to the D2D network devices, and $t_{um}$ is the time delay beween user $u$ and mobile device $m$.

**Approach 1: The pull approach**

1: **Initialization:**
   a) The controller collects network status and slice data
   b) The controller prepares the user and neighbor lists
   c) Based on the purchased capacity and number of content object, the controller determines the number of first users, *Ft*

2: **Content Visiting:**
First, the super user has to consider the order of the requesting user
**If** (new user is one of the first *Ft users)*
   a) Send the request to the controller
   b) *Cache Table Generation:* Based on the chosen cache strategy, the caching table is calculated for the infrastructure devices around the requesting user by the controller
   c) *Cache Allocation:* The physical resources are allocated to infrastructure devices.
   d) *Mapping:* The super user instructs the mobile users on how to reach the desired objects.
**Else if** (new user is not one of the first *ε users)*
   *Mapping:* The super user instructs the mobile users on how to reach the desired objects

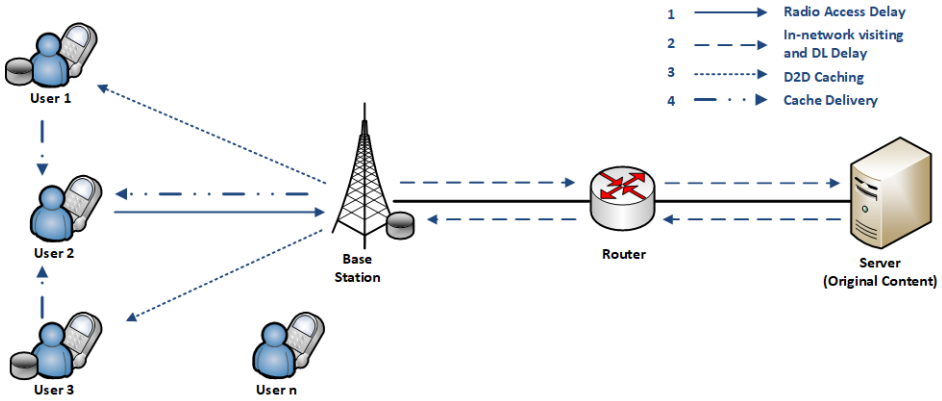



**Fig. 3.** The pull approach

## B. The Push Approach

In this approach, the content is cached on the infrastructure devices before any user makes a request (i.e. pre-caching). That is, the content is pushed closer to the end users. Since there are no requesting users at the time of caching, the content objects are cached around some *pivotal* users. The controller chooses some pivotal users from the set of mobile devices, preferably at optimal distances from one another, and then generates the caching table. The number of pivotal users is determined by considering the purchased capacity and the number of object content. Unlike the pull approach where the cache generation table occurs incrementally (one user at a time), the cache table generation and content cache allocation happen categorically once in the push approach. **Fig. 4** shows the latency model for the push approach.

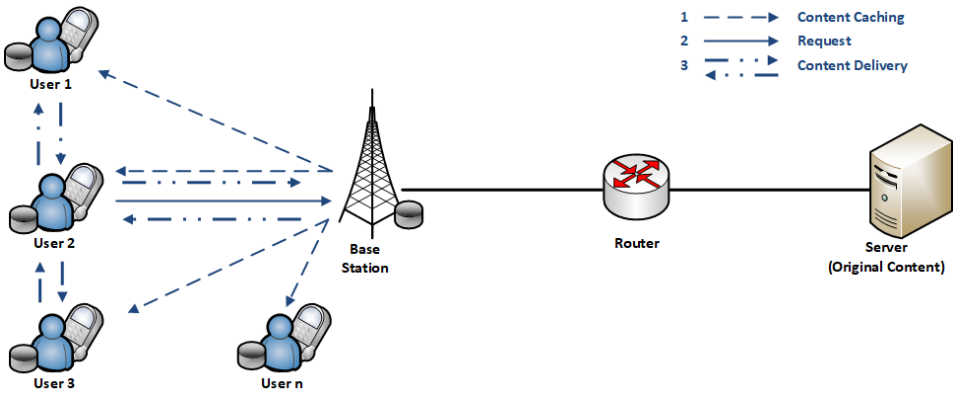



**Fig. 4.** The push approach

The push approach has a clear advantage of reduced visiting latency. **Approach 2** defines the framework for this approach for a given slice, which is more sequential and organized than that of the pull approach. Here, there are three distinct modules: initialization, cache allocation, and content visiting. Initialization starts with collecting the network statuses and creating the user and neighbor lists, which are calculated in the same way as in the pull approach. Additionally, the controller has to choose pivotal users at this stage around which the content objects will be cached. Cache allocation immediately follows initialization as the controller generates the complete and final caching table, moves it down to the super user and allows for resources to be cached to the physical infrastructure resources. Finally, the super user waits for mobile users to make their requests in order for it to map them to the pre-cached content objects.

**Approach 2: The push approach**

1: **Initialization:**
   a) The controller collects network status and slice data
   b) The controller prepares the User and Neighbor lists and surveys the available D2D slot capacity
   c) Based on the purchased capacity and number of content object, the controller determines the number of and selects the pivotal users (Pt)

2: **Cache Allocation:**
   a) *Cache Table Generation:* The controller, based on the chosen cache strategy, creates the caching table such that content objects are cached around pivotal users.
   b) *Cache Allocation:* The physical resources are allocated to infrastructure devices.

3: **Content Visiting:**
   *Mapping:* The super user instructs the mobile users on how to reach the desired objects

It is evident that this model will have better latency-related performance as it eliminates the in-network visiting and downloading delay. The super user has the caching table and is responsible for mapping the requesting users to their desired slice objects. The corresponding delay model can be expressed as follows:

$$t_{uo} = t_R + t_{um} . \quad (21)$$

### 4.2.2. The Caching Strategies:

In this sub-section, we examine the cache allocation strategies used in this paper. We have a total of two strategies: greedy and LABA.

## A. Greedy

Basically, the greedy-enabled algorithms cache the content objects as close as possible to the targeted mobile user, be it a requesting user or a pivotal user. These neighboring mobile devices, on which the content objects are cached, take as many content objects as possible. If local capacity allows it, all content objects could possibly be cached on one mobile user closest to the targeted user with this strategy. **Strategy 1** illustrates the greedy cache allocation strategy.

As illustrated in **Strategy 1**, when attempting to cache an object, the list of eligible neighboring devices of the requesting user *M* is scanned in line 3.

**Strategy 1: Greedy cache allocation**

```
1   Initialize counter
2   for o ∈ f_s
3     for m ∈ M
4       if Slice(s).cachingTable (m, o) == 0 ...
5         && Slice(s).copy_al_obj (o) < counter ...
6         && Slice(s).total_al_obj < leased ...
7         && User (m).C > 0
8                 Slice(s).cachingTable(m, o) = 1;
9                 Slice(s).copy_al_obj(o) = copy_al_obj(o) + 1;
10                Slice(s).total_al_obj = total_al_obj + 1;
11                User(m).C = User(m).C − 1;
12      end
13    end
14  end
```

Lines 4 to 7 are conditions that need to be satisfied before making the caching decision. The first condition prevents duplication by ensuring that a particular object is not currently cached on a device. The second condition guarantees that the number of object copies does not exceed the number of requesting or pivotal users. This condition is imposed to prevent the first copies from monopolizing the entire leased capacity, thus ensuring that all object copies are going to be cached. The third condition makes sure that there is enough leased capacity before allowing the object to be cached, while the fourth condition checks whether the mobile device has the physical capacity to receive the object copy. Lines 8~11 are the caching decision stage and ensuing updates. Line 8 updates the caching table, and it reads as: for slice $s$, the object $o$ is cached on mobile device $d$. Lines 9~10 update the number of copies cached for this object and the total allocated objects for this slice, respectively. Line 11 updates the remaining caching capacity at the mobile device.

## B. Load-Aware Balanced Allocation (LABA)

The LABA strategy aims at caching the content in a way that guarantees the best possible *even* distribution of content objects in the network. It is observed that when using the greedy strategy with mobile devices of considerable caching storage capacity, the content is cached on only a handful of mobile devices. Due to the disparity of mobile users and their limited D2D communication range, this leaves them being unable to find a transmitting mobile device. The LABA strategy considers all the neighboring mobile devices of a targeted user and strives to cache objects on all them if possible, in an orderly fashion. As demonstrated in **Strategy 2,** the LABA cache allocation strategy starts by initializing a counter and setting a round robin (RR) variable to 1. The RR value is used to indicate the starting point of the neighbor user row

in the cache table. This starting point in the greedy strategy for every object is always 1 (i.e. the closest neighbor). However, as the aim is to cache every two consecutive objects on two different devices, the starting points need to be changed accordingly. We examine the neighboring list of the targeted user starting with 1 as was set by the RR. In line 5, we obtain the true identity (ID) of this mobile device and proceed to check if it meets the conditions in lines 6~10.

**Strategy 2: LABA cache allocation**

```
1   Initialize counter
2   RR=1;
3   for o ∈ f_s
4     for j = RR: Slice(s).Subscriber(Pt).L
5       val = User4(Pt).Neighbor(j).ID
6       if Slice(s).cachingTable (val, o) == 0 ...
7         && copy_al_obj(o) < counter ...
8         && User(val).C > 0 ...
9         && total_allocation < Slice (s).leased ...
10        && Slice(s).User(val).C < Slice (s).limit
11              Slice(s).cachingTable (val, o) = 1;
12              copy_al_obj (o) = copy_al_obj (o) + 1;
13              User (val).C = User (val).C − 1;
14              total_allocation = total_allocation + 1;
15              Slice(s).User(val).C = Slice (s).User(val).C + 1;
16              if val == User (Pt).Neighbor(RR).ID
17                RR = RR + 1;
18              end
19              if RR == Slice(s).Subscriber(Pt).L4 + 1
20                RR = 1;
21              end
22              Break
23        end
24    end
25  end
```

The first condition is to prevent duplication, while the second is to ensure that we do not have more object copies than pivotal users at that specific moment. The third condition checks the physical availability of storage capacity at the mobile device, while the fourth checks the logical upper capacity bound imposed for a slice to use in a single mobile device. The final condition is to make sure there is enough leased capacity for caching. In lines 11~15, the caching table and the system are updated accordingly. Line 16 is used to ensure that the pointer RR is moved one step at a time, only if the previous RR value was used for caching. In line 19, we reset the pointer back to one when there are more objects than mobile neighbors. Finally, the break function at line 22 ensures that only one object is cached at one device at a time.

## 5. Performance Evaluaton

In this section, we conduct two experiments, using the following metrics to evaluate the performances of a given algorithm: 1) average visiting time, 2) end user's QoE, 3) InP resource utilization, and 4) MVNO gain.

**Table 2.** Common simulation parameters

| | |
|---|---|
| BS and D2D Tx Powers | 46 and 24 dBm |
| BS and D2D bandwidths | 20 and 10 MHz |
| D2D communication range, $Range_m$ | 65 m |
| Path loss | $20\ \log_{10}(D(km)) + 100$ |
| Shadowing | Log-normal distribution with standard deviation of 8dB |
| Noise spectral density | -174 dBm |
| The size of content file $f_i$ | 150 objects |
| Size of object $b_o$ | 0.5 Mbyte |
| Transmission rate minimum requirement, $R_s$ | 4 Mbps |

**Table 3.** Scenario-specific Parameters

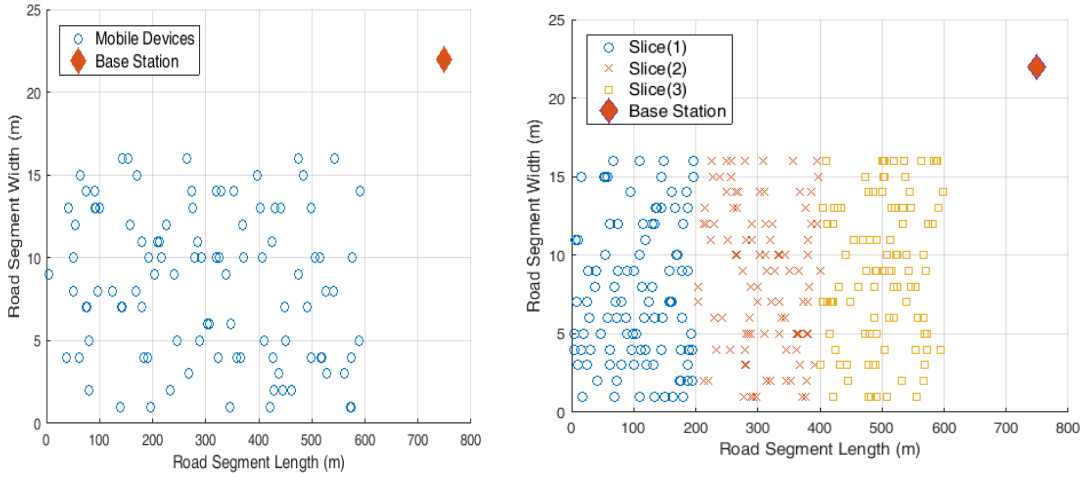
| | Scenario 1 | Scenario 2 |
|---|---|---|
| Number of slices, $S$ | 1 | 3 |
| Visiting latency demand, $q_s$ | 12 ms | 10, 12, 14 ms |
| Number of mobile users, $u_s$ | 100 | 100 per slice |
| Leased capacity per slice, $l_s$ | 1500 | 1050, 900, 750 |
| Mobile device caching capacity per slice, $C_m$ | 30 | 90 |
| Static limit on cache allowed on a mobile device per slice, $C_m(s)$ | NA | 30 |
| BS caching capacity $C_k$ | 150 | 450 |

## 5.1. Scenario configuration

We use a MATLAB simulator to show the performance of the proposed solutions. We run the simulations on an X64 VAIO® laptop with a dual-core CPU (Intel i5-2430M), 4GB RAM and a Microsoft Windows 7 SP1 operating system. We assume a fixed BS and randomly distributed end users on a geographical area. For four algorithms, we run each algorithm for 100 times and take the averages into account for randomness effect of the simulation.

In our simulation,we assume a population of 100 mobile devices uniformly distributed on a 640mx16m-long road segment coverage area of one BS that acts as a super user $k$. The distance between the mobile users and the macro cell vary from 168m to 572m. In this setting, a mobile device averages 18 neighboring mobile devices, though the actual number of neighbors can reduce to as low as 4 neighbors and increases to as high as 25 depending on the actual location of the device. All the infrastructure devices (macro cell and mobile users) are equipped with in-network caching capabilities. The cache capability is defined in terms of slots, and each mobile device has $C_d$ slots, while the BS has $C_k$ slots. The mobile users existing in one slice can only contact the MVNO it belongs to. The average delay for each hop $T$ in the backhaul network is assumed as 2ms, and the remote visiting latency from the BS to the original server is assumed as 35ms. The rest of the simulation parameters are summarized in **Table 2**.

In the first scenario, there is only one MVNO, while in the second scenario there are three MVNOs. The different parameters for the two scenarios are summarized in **Table 3**, whereas **Fig. 5** showcases the topologies for the two scenarios i.e. **Fig. 5a** for a single MVNO and **Fig. 5b** for three MVNOs.




a) Scenario 1 topology

b) Scenario 2 topology

**Fig. 5.** Scenario configuration topologies

### 5.2. Scenario 1: A single MVNO (One Slice → Many Users)

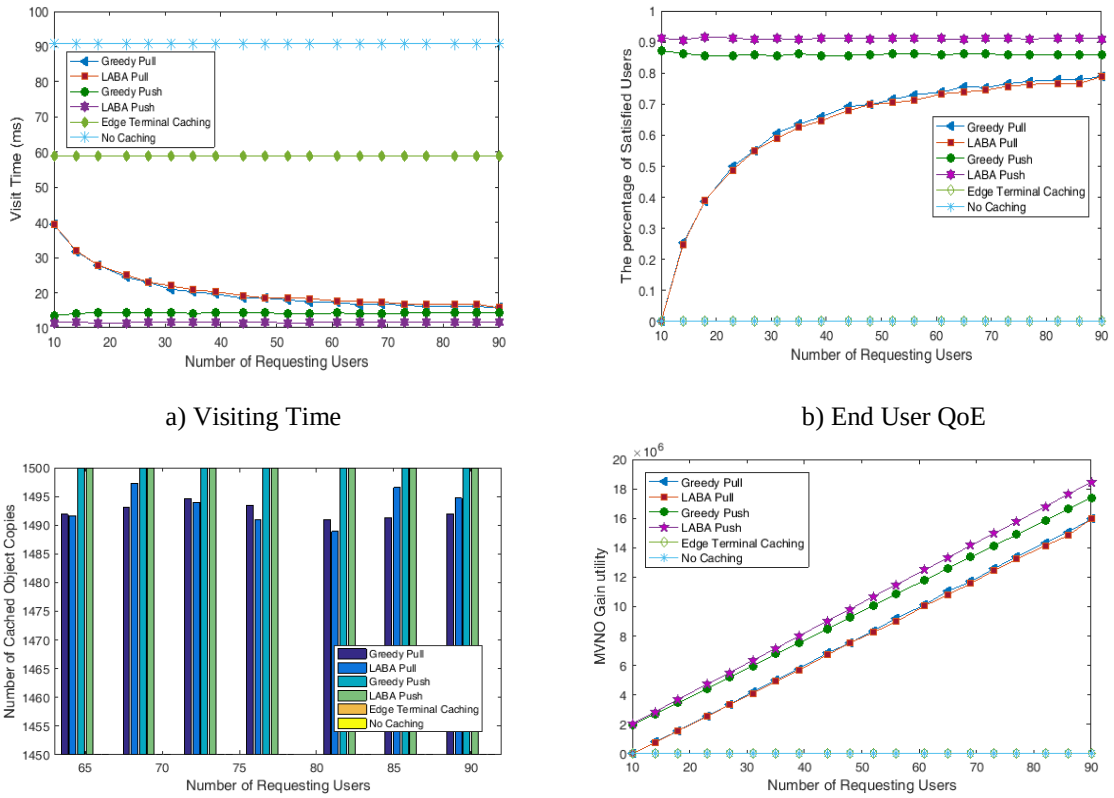
In this experiment, we assume a single operator to be providing content to the entire population of mobile users. One aim of this experiment is to compare the four algorithms introduced and find which one is best suited for latency reduction purposes. In addition, we analyze the performance advantages of caching at the D2D network, caching at the BS, and no caching at all.

The results in **Fig. 6a** illustrates the performance advantages of caching content in the D2D network in contrast to caching at the BS or no caching at all. In using any of the four algorithms, the visiting latency is significantly reduced. This improvement, however, varies in level and is dictated by the type of algorithm implemented. The pull-based algorithms registered high visiting latency times for the first users, due to the fact that the content was being pulled towards them from the original SP every time a request was initiated. The push-based algorithms, on the other hand, realized a more balanced and lower visiting latency.

For the end user QoE, **Fig. 6b** shows that the push-based algorithms demonstrated the highest average user satisfaction rate. User satisfaction $\xi(t_{uo})$ is the QoE metric we derived in sub-section 3.3. It is a latency-sensitive utility function of the total downloading delay and can be calculated with a sigmoid function as shown in equation (12). In **Fig. 6c**, we realize that though the leased capacity was the same for the four algorithms, the greedy-based algorithms were less efficient in utilizing this resource. This is the case because the first users around which the content is to be cached could be in close proximity with one another, thus the cache allocation was concentrated on a few devices. Moreover, due to the storage cache limitations at the D2D level, these devices in the concentrated area were full and as a result, some of the purchased caches could not be utilized. On the other hand, the push-based algorithms were able to utilize the full purchased capacity due to their choice of pivotal users and extended

reach. Finally, we can clearly see in **Fig. 6d** that the MVNO gain utility is best served when adopting the push-based algorithms.

Looking closely, the LABA push algorithm significantly outperforms the other algorithms in all aspects. Despite having the same resources, the LABA push algorithm provided the best and most consistent visiting latency reduction, and its users registered the highest level of satisfaction. It also ensured that the purchased capacity from the InP is fully utilized and that the MVNO achieved the highest possible gain. The performance results prove that the LABA push is the most suitable algorithm for caching content on the D2D network, and thus only this algorithm will be used in expanding our work to the multi-operator scenario.




a) Visiting Time

b) End User QoE

c) InP Resource Utilization

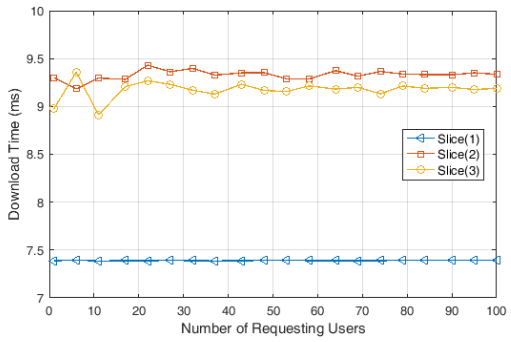
d) MVNO Gain

**Fig. 6.** Performance results for single MVNO scenario

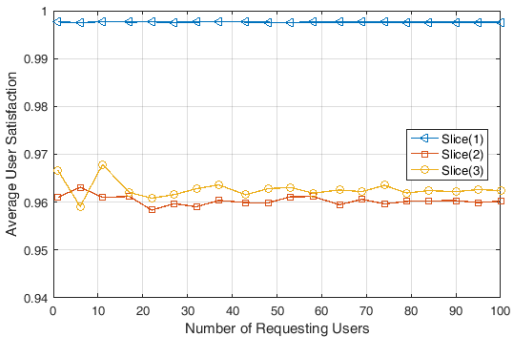
### 5.3. Scenario 2: Multiple MVNOs (Many Slices → Many Users)

In this simulation, we only use the LABA push algorithm. We assume the co-existence of multiple MVNOs sharing the same infrastructure, but each MVNO serves a unique content. The aim of this experiment is to extend our work and enable multiple MVNOs to share the same set of mobile infrastructure devices via means of virtualization while guaranteeing the latency-reduction advantages of implementing network slicing techniques on information-centric wireless D2D networks through our algorithm.

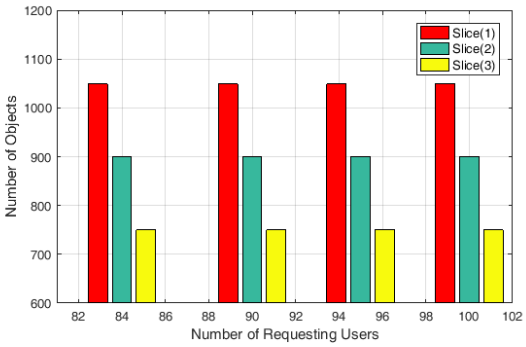
**Fig. 7** demonstrates the performance results for the same metrics we tested the first experiment with. The results prove the effectiveness of the proposed solution in serving multiple MVNOs by sharing the same infrastructure. Similar to the one MVNO case, all the slice operators satisfied their QoS requirements, the end users enjoyed service with satisfying QoE, the InP had its allocated resources fully utilized and the MVNOs realized considerable gains.



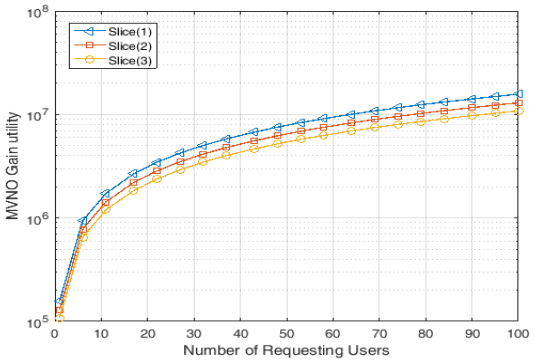

a) Downloading Time




b) End User QoE




c) InP Resource Utilization




d) MVNO Utility

**Fig. 7.** Performance results for multiple MVNOs scenario

**Fig. 8,** on the other hand, showcases the distribution of the cache objects belonging to different MVNOs on the InP's mobile devices. It can be observed that, the objects belonging to one operator were cached on mobile devices across the network. This means mobile users can communicate and exchange content from one another so long as they fall in range despite their MVNO association. This is a direct result of the virtualization adapted in our system and it allows for an extremely efficient utilization of the InP's resources and a reduced latency for the MVNO users.

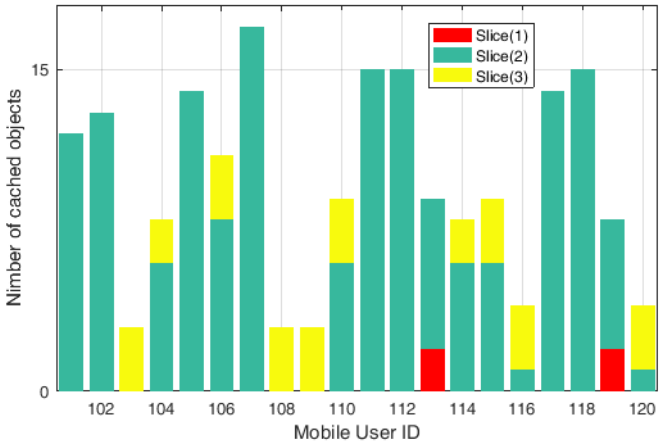



**Fig. 8.** Cached objects distribution

## 6. Conclusion

In this paper, we applied network cache slicing techniques on a designed virtualized wireless D2D network to reduce the visiting latency for end users. We combined two caching approaches and two caching strategies for caching content on the D2D network. We then conducted two experiments to test these algorithms and evaluate their performance in visiting latency reduction, among other criteria. In the first simulation, we compared the performances of our algorithms to that of others with different cache placement choices and we arrived at the conclusion that caching at the D2D network surpasses by far, other approaches when it comes to visiting latency reduction and end-user QoE. In the same experiment, we further inspected the four algorithms, analyzing their performances, and proved that the LABA push algorithm is unequivocally the most suited for D2D caching purposes. In the second simulation, we expanded our work to cover the case of multiple MVNOs co-existing and sharing the same InP. By means of virtualization, we proved the LABA push algorithms are able to deliver content with the same satisfactory performance across multiple MVNOs with different QoS requirements.

## Acknowledgment

This work is supported by National Natural Science Research Foundation of China, Grant no. 61771098, by the Fundamental Research Funds for the Central Universities under grant no. ZYGX2018J049, and the ZTE Innovation Research Fund for Universities Program 2016.

## References

[1] Cisco, “Cisco visual networking index: global mobile data traffic forecast update, 2016–2021,” presented at the White Paper, San Jose, CA, USA, Feb. 2017. Article (CrossRef Link)

[2] P. Krishnan, Danny Raz and Yuval Shavitt, “The cache location problem,” *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 568-528, October 2000. Article (CrossRef Link)

[3] S. Acharya, M. Franklin, and S. Zdonik. “Balancing push and pull for data broadcast,” in *Proc. of the 1997 ACM SIGMOD International Conference on Management of Data*, pages 183–194, Tucson, AZ, May 1997. Article (CrossRef Link)

[4] G. Xylomenos et al., "A survey of information-centric networking research," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1024–1049, 2nd Quart. 2014. Article (CrossRef Link)
[5] C. Fang, F. R. Yu, T. Huang, J. Liu, and J. Liu, "A survey of green information-centric networking: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1455–1472, 3rd Quart. 2015. Article (CrossRef Link)
[6] C. Liang, F. R. Yu, and X. Zhang, "Information-centric network function virtualization over 5G mobile wireless networks," *IEEE Netw.*, vol. 29, no. 3, pp. 68–74, May 2015. Article (CrossRef Link)
[7] C. Liang and F. R. Yu, "Wireless network virtualization: a survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 358–380, 1st Quart. 2015. Article (CrossRef Link)
[8] T. Forde, I. Macaluso, L. Doyle, "Exclusive sharing & virtualization of the cellular network," *IEEE International Dynamic Spectrum Access Networks Symposium*, 3–6 May 2011. Article (CrossRef Link)
[9] H. Wen, P. K. Tiwary, and T. Le-Ngoc, "Wireless virtualization," *ser. SpringerBriefs in Computer Science*. Berlin, Germany: Springer-Verlag, Sep. 2013. Article (CrossRef Link)
[10] D. Kreutz, F. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky and S. Uhlig, "Software-defined networking: a comprehensive survey," in *Proc. of the IEEE*, vol. 103, no. 1, pp. 14 - 76, 2015. Article (CrossRef Link)
[11] B. A. A. Nunes, "A survey of software-defined networking: past, present and future of programmable networks," *IEEE Commun. Surveys and Tutorials*, vol. 99, pp. 1-18, Feb. 2014. Article (CrossRef Link)
[12] W. H. Chin, Z. Fan, R. Haines, "Emerging technologies and research challenges for 5G wireless networks," *IEEE Wireless Communications*, vol. 21, pp. 106-112, 2014. Article (CrossRef Link)
[13] Z. Cai and X. Zheng, "A Private and Efficient Mechanism for Data Uploading in Smart Cyber-Physical System," *IEEE Transactions on Network Science and Engineering (TNSE)*, vol. PP. no. 99, pp. 1-1, 2018. Article (CrossRef Link)
[14] "Proximity-Based Services (ProSe); Stage 2," 3rd Generation Partnership Project (3GPP), TS 23.303 V12.3.0, *Eur. Telecommun. Stand. Inst.*, Dec. 2014. Article (CrossRef Link)
[15] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1801–1819, 4th Quart. 2014. Article (CrossRef Link)
[16] S. P. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011. Article (CrossRef Link)
[17] X. Wang, "Cache in the Air: exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131-39, Feb. 2014. Article (CrossRef Link)
[18] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, G. Caire, "FemtoCaching: Wireless video content delivery through distributed caching helpers," *IEEE INFOCOM 2012*, pp. 1107–1115, March 2012. Article (CrossRef Link)
[19] P. Blasco, D. Gunduz, "Learning-based optimization of cache content in a small cell base station," *IEEE International Conference on Communications (ICC)*, pp. 1897–1903, June 2014. Article (CrossRef Link)
[20] F. Pantisano, M. Bennis, W. Saad, M. Debbah, "In-network caching and content placement in cooperative small cell networks," *1st International Conference on 5G for Ubiquitous Connectivity (5GU)*, pp. 128–133, Nov. 2014. Article (CrossRef Link)
[21] T. Wang, L. Song, Z. Han, "Dynamic femtocaching for mobile users," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 861–865, March 2015. Article (CrossRef Link)

[22] N. Golrezaei, A. G. Dimakis, A. F. Molisch, "Wireless device-to-device communications with distributed caching," *IEEE International Symposium on Information Theory Proceedings (ISIT)*, pp. 2781–2785, July 2012. Article (CrossRef Link)

[23] N. Golrezaei, P. Mansourifard, A. F. Molisch, A. G. Dimakis, "Base-Station Assisted Device-to-Device Communications for High- Throughput Wireless Video Networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 7, pp. 3665–3676, July 2014. Article (CrossRef Link)

[24] I. Pappalardo, G. Quer, B. D. Rao, M. Zorzi, "Caching strategies in heterogeneous networks with D2D small BS and macro BS communications," in *Proc. of IEEE Int. Conf. Commun. (ICC)*, pp. 1-6, May 2016. Article (CrossRef Link)

[25] K. Wang, F. R. Yu, H. Li, "Information-centric virtualized cellular networks with device-to-device (D2D) communications," *IEEE Trans. Veh. Technol.*, vol. 65, no. 11, pp. 9319-9329, Nov. 2016. Article (CrossRef Link)

[26] K. Wang, H. Li, F. R. Yu, W. Wei, "Virtual resource allocation in software-defined information-centric cellular networks with device-to-device communications and imperfect CSI", *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 10011-10021, Dec. 2016. Article (CrossRef Link)

[27] G. Fodor, E. Dahlman, S. Parkvall, G. Mildh, N. Reider, G. Miklos, Z. Turanyi, "Design aspects of cellular network assisted device-to-device communications," *IEEE Communication Magazine*, vol. 50, no. 3, 2012. Article (CrossRef Link)

[28] Mehmet. K, "Lower bounds on the LTE-A average random access delay under massive M2M arrivals," *IEEE Transactions on Communications*, vol.64, no.5, 2016, pp. 2104 - 2115. Article (CrossRef Link)

**Guolin Sun** received his B.S., M.S. and Ph.D. degrees all in Comm. and Info. System from the University of Electronic Sci.&Tech. of China (UESTC), Chengdu, China, in 2000, 2003 and 2005 respectively. After Ph.D. graduation in 2005, Dr. Guolin has got eight years industrial work experiences on wireless research and development for LTE, Wi-Fi, Internet of Things (ZIGBEE and RFID, etc.), Cognitive radio, Localization and navigation. Before he joins the School of Computer Science and Engineering, University of Electronic Sci.&Tech. of China, as an Associate Professor on Aug. 2012, he worked in Huawei Technologies Sweden. Dr. Guolin Sun has filed over 30 patents, and published over 30 scientific conference and journal papers, acts as TPC member of conferences. Currently, he serves as a vice-chair of the 5G oriented cognitive radio SIG of the IEEE (Technical Committee on Cognitive Networks (TCCN) of the IEEE Communication Society. His general research interest is software defined networks, network function virtualization, radio resource management.

**Hisham Al-Ward** received his Bachelor's degree in Electrical (Communications and Electronics) Engineering from Sana'a University, Yemen in 2011. He is currently pursuing a Master's degree in Computer Science at the University of Electronic Science and Technology of China (UESTC). A certified Cisco professional, he had a work experience as the network infrastructure engineer in YemenTrack, Sana'a, Yemen between 2013 and 2014. He is now a research assistant at the Mobile Cloud-Net Research Laboratory at UESTC. His current research interests include networking technologies, 5G enabling technologies, D2D communications, and IoT.

**Gordon Owusu Boateng** received his Bachelor in Telecommunications Engineering from the Kwame Nkrumah University of Science and Technology, Kumasi-Ghana, West Africa, in 2014. He is currently studying MSc. Computer Science and Technology in University of Electronic Science and Technology of China (UESTC). From 2014 to 2016, he worked under sub-contracts for Ericsson (Ghana) and TIGO (Ghana). He is also a member of the Mobile Cloud-Net Research Team – UESTC. His interests include Mobile/Cloud Computing, 5G Wireless Networks, Data Mining and SDN.

**Wei Jiang** received his Ph.D degree from Beijing University of Posts and Telecommunications (BUPT) in 2008. Since Mar. 2008, he has been worked 4 years in Central Research Institute of Huawei Technologies, in the field of wireless communications and 3GPP standardization. In Sept. 2012, he joined the Institute of Digital Signal Processing, University of Duisburg-Essen, Germany, where he was a Postdoctoral researcher and worked for EU FP7 ABSOLUTE project and H2020 5G-PPP COHERENT project. Since Oct. 2015, he joined the Intelligent Networking Group, German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, Germany, as a senior researcher and works for H2020 5G-PPP SELFNET project. Meanwhile, he also works for the Department of Electrical and Information Technology (EIT), Technische University (TU) Kaiserslautern, Germany, as a senior lecturer. He served as a vice Chair of IEEE TCCN special interest group (SIG) "Cognitive Radio in 5G". He is the author of more than 30 papers in top international journals and conference proceedings, and has 27 patent applications in wireless communications, most of which have already been authorized in China, Europe, United States or Japan. He wrote a chapter "From OFDM to FBMC: Principles and Comparisons" for the book "Signal Processing for 5G: Algorithms and Implementations" (Wiley, 2016). His present research interests are in digital signal processing, multi-antenna technology, cooperative communications, 5G, and machine learning.