IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 24, 2020, accepted December 30, 2020, date of publication January 5, 2021, date of current version January 14, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3049442

Cache Partitioning and Caching Strategies for Device-to-Device Caching Systems

MINJOONG RIM[1] AND CHUNG G. KANG[2], (Senior Member, IEEE)
[1]Department of Information and Communication Engineering, Dongguk University, Seoul 04620, South Korea
[2]School of Electrical Engineering, Korea University, Seoul 02841, South Korea

Corresponding author: Minjoong Rim (minjoong@dongguk.edu)

This work was supported in part by the National Research Foundation of Korea, Ministry of Science and ICT (MSIT), Korea Government, under Grant 2019R1F1A1056963, and in part by the Samsung Research Funding and Incubation Center for Future Technology under Grant SRFC-IT1702-13.

ABSTRACT The amount of traffic in wireless networks is increasing exponentially, and this problem can be mitigated using device-to-device (D2D) caching technology, which installs a cache on a mobile end device. Devices can reduce the cell load through self-offloading via content in their own cache and D2D offloading using content in others' caches. However, especially in the early stage of D2D caching systems, a limited number of devices with a small storage might be used, and it is required to develop a caching scheme with excellent performance despite the small cache size. Regarding content popularity, which is common to most users, the preference probability values are not concentrated on some pieces of content, making it difficult to achieve satisfactory performance using a small cache. On the other hand, when considering individual users, content preferences may contain large values for specific content based on individual characteristics. In addition, the performance can be improved by considering short-term content preferences that reflect changes in content preferences over time or newly created content during peak hours. In this article, the hit ratio is divided into six parts considering self- and D2D offloading, common and individual user preferences, and little and large temporal changes in content preferences during peak hours. We also conceptually divide the cache of a helper into six areas in relation to the six parts of the hit ratio, and discuss cache partitioning and proactive caching strategies according to the environment.

INDEX TERMS D2D caching, wireless caching, mobile caching, content preference, cache partitioning, data offloading.

I. INTRODUCTION

Due to the increase in content-based services such as video streaming, the amount of traffic in wireless networks is exponentially growing [1]–[4]. In order to solve the data explosion in wireless networks, small cell technology, which reduces the cell size, is being used, in addition to developing mmWave technology, which utilizes high-frequency carriers with a wider bandwidth [5]–[7]. As the cells become smaller, the averaging effect disappears, resulting in significantly different data requirements over time and location, and this will worsen with a further decrease in cell size [8].

As wireless communication technology using high carrier frequencies is developed and high-frequency bands are allocated for cellular communication, wireless bandwidth is dramatically increasing [5]–[7]. However, high-frequency radios have coverage issues, and many areas can remain in the shadow areas of high frequency carriers. In addition, the data transmission rate of wireless broadband can exceed the existing wire transfer rate, which can result in a bottleneck at the backhaul unless a new backhaul is installed [9]–[13]. Therefore, there may be areas where the backhaul becomes a bottleneck or where the base station does not fully utilize the broadband radio allocated to the operator, and the amount of data supply may vary depending on the region.

Another way to cope with the data explosion in wireless networks is through caching [14]. To reduce the burden of core networks, a technology for placing cached versions of content in multiple geographical locations is being employed, and evolving into mobile edge caching (MEC) technology, which installs caches at the edge of the network close to the user [15]–[19]. As an extreme form of MEC for wireless networks, femto caching technology, which installs caches in base stations, or device-to-device (D2D) caching technology,

The associate editor coordinating the review of this manuscript and approving it for publication was Qilian Liang.

which installs caches on mobile end devices, is also being considered [20]–[24].

Installing caches on end devices has several advantages over installing caches on base stations or wired networks [16], [25]–[27]. First, installing a cache on a base station may not be suitable to solve the data explosion in the cell if the radio section is a bottleneck, whereas D2D communication can be executed using bands that are owned by the operator but not fully utilized by the base station. D2D caching can solve both wireless bottleneck and backhaul bottleneck problems. Second, if the number of devices increases in some area, the overall cache size of the devices increases, making it easy to cope with the traffic congestion. Third, the technology and storage capacity of devices can be developed at a faster rate due to shorter device replacement cycles and easier installation compared to base stations. Fourth, each device can more accurately identify and use the content preferences, traffic characteristics, and mobility tendencies of individual devices, through installation of application programs on the device. Fifth, cache installation is possible without depending on the operator, and a device with a cache can support multiple operators' devices via D2D communication. Sixth, the mobility of devices can be used to solve the imbalance in data demand and supply across regions. Seventh, self-offloading of a device without a separate delivery process is possible by storing content for itself.

On the other hand, mobile end devices may have smaller cache sizes and narrower coverage than wired networks. In addition, the neighboring environment of a device may change rapidly due to the mobility of devices. In network caching systems, since a large-sized cache can be used for a large number of users, global content popularity can play an important role. In D2D caching systems, a small-sized cache is used for a small number of neighboring users, excellent performance may not be achieved with global content popularity alone, and it may be necessary for a device to utilize short-term and local information such as the short-term mobility pattern of the device or the personalized tendencies of neighboring devices [28]. This article discusses caching strategies that can take advantage of the characteristics of D2D caching.

If all or most of devices can store and supply content with large caches, then there is a high possibility that excellent performance can be achieved. However, for general devices, it is difficult to secure enough storage size and to provide content to other devices due to selfish behavior and power consumption problems [29]–[31]. In addition, security, copyright and access rights issues may arise from storing and supplying data [32]–[34]. In this article, a certified specialized device that can serve content to other devices is referred to as a *helper* [35]–[37]. Particularly, early in the D2D caching systems, there may not be many helpers due to the aforementioned issues and cost implications. Thus, a caching method that can achieve satisfactory performance with a limited number of helpers is required.

In this article, we assume that high frequency bands with sufficiently large bandwidths are allocated to the operator. When a base station uses a broadband radio but the backhaul is a bottleneck, a helper utilizes the base station as a relay to transmit data. If a base station uses a narrowband radio, a helper in the cell can transmit data using single-hop or multi-hop D2D communication using the frequency bands owned by the operator but not used by the base station [23], [38]–[42]. When a device requests content that is available from a nearby helper's cache, the helper can serve the content; this process is referred to as *D2D offloading* in this article [43], [44]. Since a helper has a cache of its own, before attempting D2D offloading, it first checks its own cache for content; this process is referred to as *self-offloading* in this article [45]–[47]. In order to increase self- or D2D offloading using a small cache, it is necessary to store content with high user preferences.

Helpers can proactively store content with high content preferences before peak hours and supply the content to themselves and other users during peak hours. However, global content preferences that consider all users are unlikely to obtain sufficient offloading with a small cache size, because the probability values can be spread over a large number of pieces of content. On the other hand, if individual users are considered, preference probability values may be concentrated on some specific content depending on individual characteristics, which presents a possibility of better performance [48]–[52]. In this article, global content preferences common to all users are referred to as *common preferences*, and content preferences different based on individual users are referred to as *personal preferences*. The devices neighboring a helper may vary due to the device mobility, and it is not always possible to exploit the personal preferences of neighboring devices.

Content preferences predicted in off-peak hours may vary during peak hours. New content may be available during peak hours, such as news, while there may be significant change of content preferences over time for some content [28], [53]–[56]. In this article, content preferences with little temporal change during peak hours are referred to as *static preferences*, and content preferences of newly generated content or with considerable temporal changes during peak hours are referred to as *dynamic preferences*. For consideration of content with dynamic preferences, the cache of a helper should be updated frequently during peak hours [28]. The amount of data available may vary based on the region due to high frequency carriers, and the amount of data required may vary contingent on time and place due to the small cell technology. Hence, not all cells may be overloaded during peak hours. If some cells with a relatively low average load intermittently enter an underload state, then the cache of the helpers in the cells can be updated to enhance the performance [28], [57].

Considering personal or dynamic preferences, content preferences may be concentrated on some specific pieces of content, potentially providing decent performance even with a small cache. Considering both at the same time, in other words, considering dynamic preferences of individual users, there is a potential to improve the performance further. Dynamic personal preferences can be further supported by content prediction and content recommendation technologies [57]. When users view videos, they tend to watch videos sequentially or view one of the recommended pieces of content, and the next content to watch can be predicted based on the content currently being viewed [58]–[63]. For instance, if a user is watching an episode of a series, the next video to view is likely to be the next episode. Since content prediction will suggest a few pieces of content, preference probability values may be concentrated on specific content, and satisfactory performance may be achieved even with a small cache size.

In this article, considering common and personal, and static and dynamic characteristics, content preferences are divided into four parts, namely, *static common*, *static personal*, *dynamic common* and *dynamic personal* preferences, and then we discuss how they should be considered.

Common preferences are not dependent on the location or mobility of devices. Each helper's personal preferences are associated with self-offloading, and thus they are independent of the location or mobility of the helper. However, when considering the personal preferences of a helper's neighbors, the mobility characteristics of the helper and the neighboring devices should be considered. When discussing hit ratio or caching, each of the personal parts can be divided into two parts: self and neighbor. In other words, *static common*, *static self*, *static neighbor*, *dynamic common*, *dynamic self*, and *dynamic neighbor* parts can be discussed in regards to the hit ratio. Furthermore, each of the content in the cache can be classified as one of the six categories depending on which of the six parts contributes the most to the hit ratio. Based on the category of content, the cache of a helper can be conceptually divided into six areas so that we can determine which categories of content primarily need to be stored.

This article discusses proactive caching strategies considering the six parts of the hit ratio. In addition, we discuss which categories of content should be mainly considered for proactive caching according to the environment. In particular, we discuss cache partitioning and caching strategies depending on the cell load, the number of devices in the cell, and the mobility of devices. Here, the partitioning of a cache does not mean that the content is stored after clearly partitioning the cache's physical space, but it is done to check the properties of the content stored in the cache and to describe the caching strategy of the helper.

The contributions of this article are as follows.

(a) While most of the literature related to caching only considers common preferences or additionally considers individual user preferences [14]–[27], [29]–[37], this article considers content preferences of the four parts: static common, static personal, dynamic common, and dynamic personal preferences.

(b) While many literatures have not fully stated the characteristics of D2D caching systems [14]–[27], [29]–[37], this article considers self-offloading and D2D offloading in addition to the four parts of content preferences. The hit ratio of a helper is then divided into the six parts: static common, static self, static neighbor, dynamic common, dynamic self, and dynamic neighbor parts. Based on the hit ratio of the six parts, we discuss which parts should be mainly considered depending on the environment. We also categorize each piece of content, and discuss which categories of content should be primarily stored depending on the environment.

(c) While most of the literature assumes homogeneous environments in regards to the cell load or device mobility, or tries to solve an optimization problem in a given heterogeneous environment [14]–[27], [29]–[47], this article discusses cache partitioning and proactive caching strategies in consideration of environments where cell loads vary based on the region and time, and where there exist both mobile and non-mobile devices.

(d) While most of the literature provides solutions to optimization problems with specific environments, and does not describe how different helpers in various environments can cooperate in a distributed manner [14]–[27], [29]–[47], this article considers diverse environments through a simplified but unified model, and discuss how helpers in different environments should cooperate.

(e) In addition, this article discusses self-offloading and D2D offloading, considering devices that can store and deliver content to other users and devices that store content but only use it for themselves. This article describes the caching strategy for self-caching devices and demonstrates the effectiveness of self-offloading.

The organization of the rest of this article is as follows. Section II describes the system model. Specifically, it covers the content storage and delivery, cell load, device mobility, and content preference models used in this article, in addition to discussing the hit ratio and cache partitioning. Section III presents cache partitioning and caching strategies based on the degree of the cell load and the device mobility. Section IV illustrates cache partitioning and caching strategies with various numerical results, and Section V presents the concluding remarks.

II. SYSTEM MODEL

A. CONTENT STORAGE AND DELIVERY MODEL

Due to cost and other issues, only a limited number of devices will be able to store and deliver content, especially in the early days of D2D caching systems. In this article, a device that can store and supply content to other devices is referred to as a *helper* and the other devices are called *user equipment* (UE). UEs do not supply content to other devices, although some of them may have a self-cache for self-offloading.

Using a self-cache without delivering content to other devices can alleviate many problems such as security, copyright, and access rights issues, and reduce power consumption resulting from D2D discovery and delivery. In this article, a UE that does not have a cache is called a *type-I UE*, and a UE that has a self-cache is called a *type-II UE*. It is assumed that most of the UEs are of type-I, and unless otherwise specified, a UE means a type-I UE. A caching strategy is also required for a type-II UE, but this is similar as in the case of a helper in the absence of other devices, thus this article does not focus on this, and only discusses the cases when necessary.

Suppose that the operator for a considered system has a sufficiently wide bandwidth at high frequencies. A sufficient amount of data can be provided if the cell uses all bandwidth held by the operator and a high capacity backhaul is provided. However, not all cells may use broadband radio due to the coverage problem, and high-capacity backhauls may not be provided for all cells due to cost implications. If a cell does not use all the bandwidth available to the operator, then a helper in the cell can use the unused bandwidth to supply data through D2D communication. If a cell uses all the bandwidth but the backhaul is a bottleneck, a helper in the cell can utilize the base station as a relay to deliver data. For simplicity of discussion, this article assumes that a helper can provide data to any device in the same cell using multi-hop D2D communication or utilizing a base station as a relay.

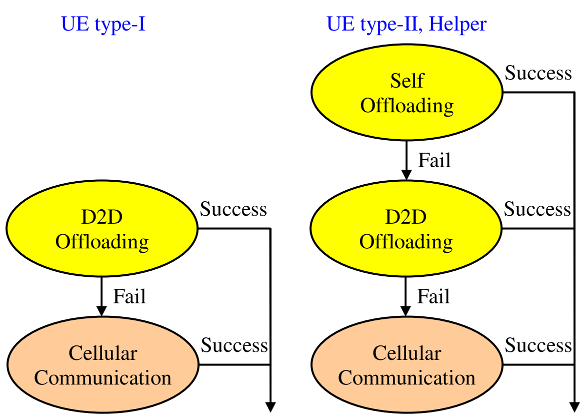

FIGURE 1. Offloading.

A helper stores content in an underload cell and supplies content in an overload cell. Before the start of peak hours, all cells are assumed to be underloaded, thus helpers can store content in any cells. A helper can update the cache if it is in a cell that goes into an underload condition during peak hours. When a device requests content in an overloaded cell, if it is a type-I UE, as shown in Fig. 1, it first checks whether there is a helper in the cell who has the requested content in the cache. If the content is found, then the device retrieves the content from the helper; this process is referred to as *D2D offloading*. If the content is not found in the caches of the helpers in the cell, then the content is fetched from the base station. If the device that requests content is a helper or a UE of type-II, it first checks if the content is available in its own cache. If the content is found in its cache, the device can use the content without any delivery process; this process is referred to as *self-offloading*. If self-offloading fails, then D2D offloading can be attempted.

B. CELL LOAD MODEL

In this article, we assume that the operator has a very wide bandwidth at high frequencies. Cells with broadband radios and high-capacity backhauls may provide a large data supply, while cells using low-frequency carriers or without high-capacity backhauls may only provide a small amount of data. In addition, as cells become smaller, the averaging effect disappears, and the data requirements of cells may vary over time. Therefore, cells with a large data supply may be intermittently underloaded even during peak hours.

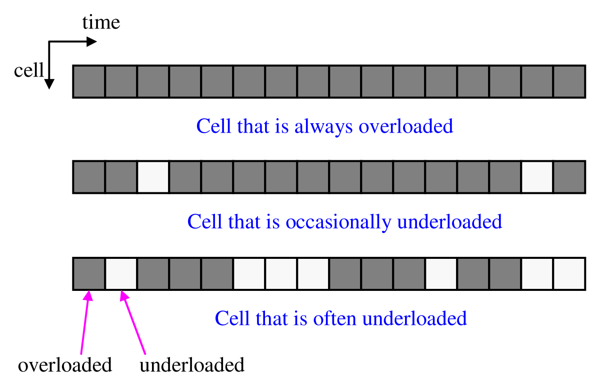

FIGURE 2. Illustration of cell load model.

In this article, we consider three types of overload cells, as shown in Fig. 2. The first case involves a cell that is always overloaded during peak hours. In this case, a helper in the cell stores content prior to peak hours, and the cache of the helper is not updated during peak hours. The second case concerns a cell that is mostly overloaded during peak hours, but can occasionally be in an underload condition. In this case, a helper updates its cache whenever the cell is in an underload state. However, it may not be easy for a helper to consider dynamic preferences since the content updates may not be sufficiently frequent. The third case involves a cell with a high data supply, which can often go to an underload state even during peak hours. In this case, a helper can frequently update the cache during peak hours. In other words, the content update period is very short.

C. DEVICE MOBILITY MODEL

We assume that there are N_{device} devices per cell and the number of cells considered in this article, denoted as N_{cell}, is very large. Hence, the total number of devices considered in this article, denoted as N_{total}, is $N_{cell}N_{device}$. We consider nomadic devices with little mobility and mobile devices such as cars and buses. We assume that among N_{device} devices in a cell, $N_{nomadic}$ devices are nomadic and the other $N_{device} - N_{nomadic}$ devices are mobile. In order to briefly discuss the mobility of devices, we assume that a device remains in one cell for a period of T seconds, and a time duration is considered as a unit of time T.

Throughout this article, we assume that device 1 is the helper considered for caching. Assuming that the content update time is 0, the probability that device 1 (the helper) and device n are in the same cell at time m is denoted as $\delta(n, m)$. Since it is assumed that there are N_{device} devices per cell, the following equations can be established:

$$\delta(1, m) = 1 \qquad \sum_{n=2}^{N_{total}} \delta(n, m) = N_{device} - 1 \tag{1}$$

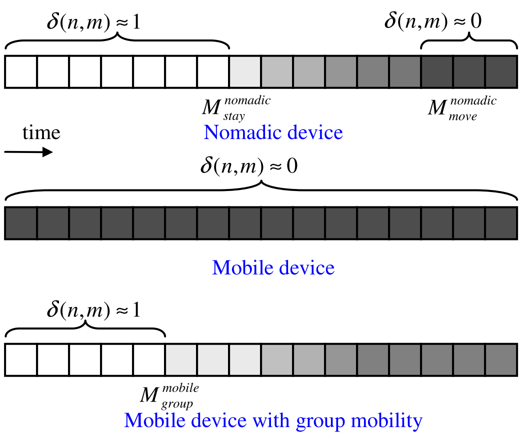

FIGURE 3. Device mobility model.

Fig. 3 briefly shows the mobility model considered in this article. In the case of a nomadic device, it is assumed that the probability of movement is not high when considering a short time. Suppose that device 1 is a nomadic helper and devices 2 through $N_{nomadic}$ are the other nomadic devices in the same cell. In this article, we assume that they are likely to remain in the same cell for a short time $M^{nomadic}_{stay}$. Let $S_{nomadic}$ denote the set of the nomadic devices in the cells other than the target cell, and S_{mobile} denote the set of the mobile devices in all cells. We also assume that the helper and a device in $S_{nomadic}$ will remain in different cells for a short time $M^{nomadic}_{stay}$, and the probability that the helper and a device S_{mobile} stay in the same cell is very low. Specifically, we assume that following equations can be established:

$$\frac{1}{M}\sum_{m=1}^{M} \delta(n, m) \approx 1 \quad (2 \le n \le N_{nomadic}, M \le M^{nomadic}_{stay})$$
$$\frac{1}{M}\sum_{m=1}^{M} \delta(n, m) \approx 0 \quad (n \in S_{nomadic}, M \le M^{nomadic}_{stay})$$
$$\frac{1}{M}\sum_{m=1}^{M} \delta(n, m) \approx \frac{1}{N_{cell}} \quad (n \in S_{mobile}) \tag{2}$$

Considering a very long time, it is assumed that even a nomadic device will eventually move to another cell. If device 1 is a nomadic helper and device n are also nomadic, then the following equation is assumed to hold true for a very long time $M^{nomadic}_{move}$:

$$\frac{1}{M}\sum_{m=1}^{M} \delta(n, m) \approx \frac{N_{device} - 1}{N_{total} - 1} \quad (n \neq 1, M \ge M^{nomadic}_{move}) \tag{3}$$

In other words, if the number of cells is very large, they will not remain in the same cell eventually.

Suppose that device 1 is a mobile helper with an unknown path. If there is no other device that is socially-connected or moving together with the helper, the following equation is assumed to hold true:

$$\frac{1}{M}\sum_{m=1}^{M} \delta(n, m) \approx \frac{N_{device} - 1}{N_{total} - 1} \quad (n \neq 1, M \ge 1) \tag{4}$$

Suppose that N_{social} devices including the helper are socially-connected, so that they will meet each other and stay or move together for most of the time when considering a very long time over M^{mobile}_{social}, regardless of their location at time 0 [64]–[66]. Suppose that device 1 is a helper and devices 2 through N_{social} are socially connected with the helper. Then, the following equations are assumed to hold:

$$\frac{1}{M}\sum_{m=1}^{M} \delta(n, m) \approx 1 \quad (2 \le n \le N_{social}, M \ge M^{mobile}_{social})$$
$$\frac{1}{M}\sum_{m=1}^{M} \delta(n, m) \approx \frac{N_{device} - N_{social}}{N_{total} - N_{social}}$$
$$(n > N_{social}, M \ge M^{mobile}_{social}) \tag{5}$$

Suppose that N_{group} devices are in the same cell at time 0, and they move together due to social-tie or by group mobility, for instance, since they are in the same car or bus [64]–[66]. In that case, we assume that they are in the same cell within a short time M^{mobile}_{group}. Suppose that device 1 is a helper and devices 2 through N_{group} are moving together with the helper. Then, we assume that the following equations hold:

$$\frac{1}{M}\sum_{m=1}^{M} \delta(n, m) \approx 1 \quad (2 \le n \le N_{group}, M \le M^{mobile}_{group})$$
$$\frac{1}{M}\sum_{m=1}^{M} \delta(n, m) \approx \frac{N_{device} - N_{group}}{N_{total} - N_{group}}$$
$$(n > N_{group}, M \le M^{mobile}_{group}) \tag{6}$$

In the case of a bus or an autonomous vehicle, it may be possible to predict its travel path for the short term. Suppose that device 1 is a mobile helper that can accurately predict its movement path within a short time $M^{mobile}_{predict}$, and the mobile helper visits the cell of device n, which is nomadic, from time $M_{in}(n)$ to time $M_{out}(n) - 1$. If the mobile helper does not go through the cell to which device n belongs, then we let $M_{in}(n) = M_{out}(n) = 0$. In this case, we assume that the following equation holds:

$$\frac{1}{M}\sum_{m=1}^{M}\delta(n,m) \approx \frac{M_{out}(n)-M_{in}(n)}{M} \quad (M_{out}(n)-1 \le M \le M_{predict}^{mobile}, M_{stay}^{nomadic}) \tag{7}$$

Suppose that device 1 is a mobile helper that can accurately predict the movement path within a short period $M_{predict}^{mobile}$, and the mobile helper visits a target cell from time M_{in}^{cell} to time $M_{out}^{cell}-1$. Also, suppose that each cell has $N_{nomadic}$ nomadic devices, and devices 2 through $N_{nomadic}+1$ are the nomadic devices in the target cell. Let $S_{nomadic}$ denote the set of the nomadic devices in the cells other than the target cell, and S_{mobile} denote the set of the mobile devices other than the mobile helper. In this case, the following equations are assumed to hold:

$$\begin{aligned}
&\frac{1}{M_{out}^{cell}-M_{in}^{cell}}\sum_{m=M_{in}^{cell}}^{M_{out}^{cell}-1}\delta(n,m)\\
&\approx 1 \quad (2 \le n \le N_{nomadic}+1, M_{out}^{cell} \le M_{predict}^{mobile}, M_{stay}^{nomadic})\\
&\frac{1}{M_{out}^{cell}-M_{in}^{cell}}\sum_{m=M_{in}^{cell}}^{M_{out}^{cell}-1}\delta(n,m)\\
&\approx 0 \quad (n \in S_{nomadic}, M_{out}^{cell} \le M_{predict}^{mobile}, M_{stay}^{nomadic})\\
&\frac{1}{M_{out}^{cell}-M_{in}^{cell}}\sum_{m=M_{in}^{cell}}^{M_{out}^{cell}-1}\delta(n,m)\\
&\approx \frac{N_{device}-N_{nomadic}-1}{(N_{device}-N_{nomadic})N_{cell}-1} \quad (n \in S_{mobile})
\end{aligned} \tag{8}$$

Using the device mobility model described in this subsection, we discuss the caching strategies of nomadic or mobile helpers.

D. CONTENT PREFERENCE MODEL

While the number of pieces of content to be considered may be very large, the sum of the helpers' cache sizes in a cell may be limited, especially in the early days of D2D caching systems. Therefore, a meaningful hit ratio can be obtained only when the preference probability values are concentrated on specific content such that a small number of preference values are large compared to the others, as shown in Fig. 4.

Although the number of pieces of video content is almost infinitely large, the number of pieces having meaningful preference probabilities to be considered for caching, denoted as K_{total}, may be limited. For simplicity of discussion, we assume that the size of a piece of content is all the same. When the content preferences of device n is predicted at time 0, the *content preference* of content k of device n at time m, denoted as $P(n,k,m)$, is defined as the probability when the device requests content, the content is content k.

In this article, content preference $P(n,k,m)$ is assumed to comprise four parts, as shown in Fig. 5. The content preferences created by content popularity measured for a large number of users are common to all users and may not change rapidly over time. These preferences are referred to as *static common preferences* in this article. Since static common preferences are constructed in consideration of many users, the probability distribution may be spread over a large number of pieces of content, and an excellent performance may not be obtained with a small cache size, as shown in Fig. 4.

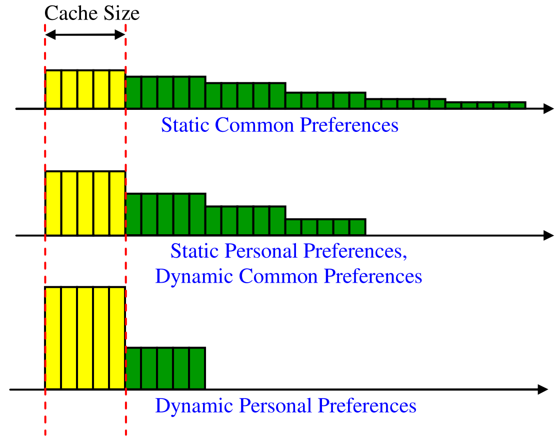

FIGURE 4. Illustration of probability distributions of content preferences.

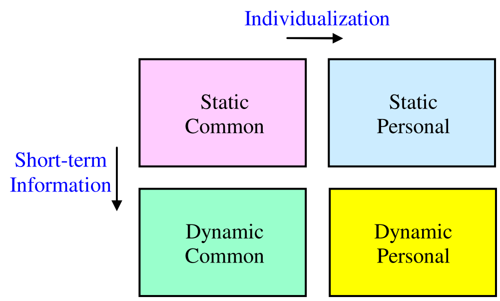

FIGURE 5. Four parts of content preferences.

On the other hand, content preferences generated considering individual tastes and propensity can show different characteristics depending on the user and may not change rapidly over time. These are referred to as *static personal preferences* in this article. Static personal preferences may have skewed probabilities for some content compared to static common preferences, and may enhance the hit ratio for a small cache size, as shown in Fig. 4 [48]–[52].

News, sports, and other video content that occur during peak hours may have high user preferences and varied preferences over time [28], [53]–[56]. Content preferences with a large temporal change can also be stored in a small cache since the probability values may be concentrated on some pieces of content. These content preferences are referred to as *dynamic common preferences* in this article.

When users watch video content, they tend to watch sequentially or view one of the recommended pieces of content, and the next content to be viewed can be predicted based on the current content being watched by each user [57]–[63]. Content preferences by content prediction can vary rapidly over time since the prediction is based on the video content the user is watching. In general, different users are likely to watch different videos, and thus they have a strong personal tendency. These content preferences are called *dynamic personal preferences*.

In this article, we assume that the content preference of content k of device n at time m can be divided into four parts and written as

$$P(n,k,m) = P_{SC}(k) + P_{SP}(n,k) + P_{DC}(k,m) + P_{DP}(n,k,m) \quad (9)$$

where $P_{SC}(k)$ is the static common preference, $P_{SP}(n,k)$ is the static personal preference for device n, $P_{DC}(k,m)$ is the dynamic common preference at time m, and $P_{DP}(n,k,m)$ is the dynamic personal preference of device n at time m. We assume that the dynamic preferences $P_{DC}(k,m)$ and $P_{DP}(n,k,m)$ decrease with time m. The weights of the four parts can be defined as follows:

$$\begin{aligned} P_{SC}^{sum} &\equiv \sum_{k=1}^{K_{total}} P_{SC}(k) \\ P_{SP}^{sum}(n) &\equiv \sum_{k=1}^{K_{total}} P_{SP}(n,k) \\ P_{DC}^{sum}(m) &\equiv \sum_{k=1}^{K_{total}} P_{DC}(k,m) \\ P_{DP}^{sum}(n,m) &\equiv \sum_{k=1}^{K_{total}} P_{DP}(n,k,m) \\ P_{SC}^{sum} + P_{SP}^{sum}(n) &+ P_{DC}^{sum}(m) + P_{DP}^{sum}(n,m) \le 1 \end{aligned} \quad (10)$$

$P_{SC}(k)$, $P_{SP}(n,k)$, $P_{DC}(k,0)$, and $P_{DP}(n,k,0)$ may not have meaningfully large values for all content, and the number of non-zero values can be less than K_{total}, as shown in Fig. 4. Let $K_{SC}^{Nonzero}$, $K_{SP}^{Nonzero}(n)$, $K_{DC}^{Nonzero}$, and $K_{DP}^{Nonzero}(n)$ be the number of non-zero values of preferences for static common, static personal, dynamic common, and dynamic personal preferences, respectively. While $K_{SC}^{Nonzero}$ can be large, $K_{SP}^{Nonzero}(n)$, $K_{DC}^{Nonzero}$, and $K_{DP}^{Nonzero}(n)$ may be smaller. In particular, content prediction may produce a small number of pieces of content, and $K_{DP}^{Nonzero}(n)$ can be very small.

We assume that personal preference values are independent of the device, and thus the average of personal preferences for all devices cannot be large enough to consider the storage of content, which is written as follows:

$$\begin{aligned} \frac{1}{N_{total}} \sum_{n=1}^{N_{total}} P_{SP}(n,k) &\approx 0 \\ \frac{1}{N_{total}} \sum_{n=1}^{N_{total}} P_{DP}(n,k,0) &\approx 0 \end{aligned} \quad (11)$$

Also, we assume that dynamic preference values can be reduced to meaninglessly small values after a long time over M_{static}, which is written as follows:

$$\begin{aligned} &\frac{1}{M} \sum_{m=1}^{M} P_{DC}(k,m) \approx 0 \\ &\frac{1}{M} \sum_{m=1}^{M} P_{DP}^{\max}(k,m) \approx 0 \\ &\text{where } P_{DP}^{\max}(k,m) \equiv \max_{n} \{P_{DP}(n,k,m)\} \quad (M \ge M_{static}) \end{aligned} \quad (12)$$

On the other hand, we assume that the amount of decrease in dynamic preferences is insignificant within a short time $M_{dynamic}$ ($\ll M_{static}$), and it is not necessary to consider the decrease in the values when updating the cache.

$$\begin{aligned} \frac{1}{M} \sum_{m=1}^{M} P_{DC}(k,m) &\approx P_{DC}(k,0) \\ \frac{1}{M} \sum_{m=1}^{M} P_{DP}(n,k,m) &\approx P_{DP}(n,k,0) \quad (M \le M_{dynamic}) \end{aligned} \quad (13)$$

Considering the four parts of content preferences, this article discusses the caching strategies of helpers.

E. HIT RATIO

In this article, the *hit ratio* for a specific period is defined as the amount of data self-offloaded or D2D offloaded compared to the total amount of data requested during that specific period. If cooperation between helpers is not considered, each helper performs caching in order to maximize offloading by the helper, and the final performance can be evaluated based on each cell. However, if there exists only one helper in a considered cell, there may be no significant difference between the hit ratios measured based on either the helper or the cell. This article discusses each helper's caching strategy. To this end, this article mainly discusses the cases of one helper per cell, and then describes the change in the caching strategy based on the number of helpers. For simplicity of discussion, we assume that data demand of each device, denoted as R_{device}, is constant regardless of the device or time, when the cell is in an overload condition. Assuming that the number of devices per cell is N_{device}, the amount of data required by one cell until time M before content update, denoted as $R_{request}$, can be expressed as follows:

$$R_{request} = N_{device} R_{device} M \quad (14)$$

Suppose that device 1 is a helper and there is no other helper in the cell. If only content k is stored in the cache of the helper, the amount of offloaded data by the helper, denoted as $R_{offloading}(k)$, can be written as

$$R_{offloading}(k) = \sum_{m=1}^{M} \sum_{n=1}^{N_{total}} \delta(n,m) R_{device} P(n,k,m) \quad (15)$$

and the *hit ratio* for content k, denoted as $\rho_{content}(k)$, can be defined as follows:

$$\rho_{content}(k) \equiv \frac{R_{offloading}(k)}{R_{request}} = \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=1}^{N_{total}}\delta(n,m)\,(P_{SC}(k)+P_{SP}(n,k) + P_{DC}(k,m)+P_{DP}(n,k,m)) \quad (16)$$

Using Eq. (1), the common parts in Eq. (16) can be rewritten as follows:

$$\frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=1}^{N_{total}}\delta(n,m)P_{SC}(k) = P_{SC}(k)$$
$$\frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=1}^{N_{total}}\delta(n,m)P_{DC}(k,m) = \frac{1}{M}\sum_{m=1}^{M}P_{DC}(k,m) \quad (17)$$

While self-offloading is always feasible if the content is stored in the helper's cache, the mobility characteristics of devices should be taken care of when we utilize the personal preferences of the helper's neighbors. Using Eq. (1), each of the personal preference parts of Eq. (16) can be divided into two parts as follows:

$$\frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=1}^{N_{total}}\delta(n,m)P_{SP}(n,k) = \frac{1}{N_{device}}P_{SP}(1,k) + \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=2}^{N_{total}}\delta(n,m)P_{SP}(n,k)$$
$$\frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=1}^{N_{total}}\delta(n,m)P_{DP}(n,k,m) = \frac{1}{N_{device}M}\sum_{m=1}^{M}P_{DP}(1,k,m) + \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=2}^{N_{total}}\delta(n,m)P_{DP}(n,k,m) \quad (18)$$

In Eq. (18), the first part in each equation is related to self-offloading, and the second part has to do with D2D offloading using the personal preferences of neighbors. From Eqs. (17) and (18), the hit ratio in Eq. (16) can be divided into six parts as shown in Fig. 6, and can be written as

$$\rho_{content}(k) = \rho_{SC}^{content}(k) + \rho_{SS}^{content}(k) + \rho_{SN}^{content}(k) + \rho_{DC}^{content}(k) + \rho_{DS}^{content}(k) + \rho_{DN}^{content}(k) \quad (19)$$

where

$$\rho_{SC}^{content}(k) \equiv P_{SC}(k)$$
$$\rho_{SS}^{content}(k) \equiv \frac{1}{N_{device}}P_{SP}(1,k)$$
$$\rho_{SN}^{content}(k) \equiv \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=2}^{N_{total}}\delta(n,m)P_{SP}(n,k)$$

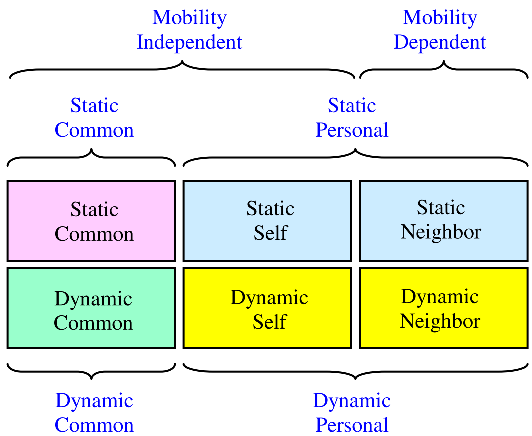

FIGURE 6. **Six parts of hit ratio.**

$$\rho_{DC}^{content}(k) \equiv \frac{1}{M}\sum_{m=1}^{M}P_{DC}(k,m)$$
$$\rho_{DS}^{content}(k) \equiv \frac{1}{N_{device}M}\sum_{m=1}^{M}P_{DP}(1,k,m)$$
$$\rho_{DN}^{content}(k) \equiv \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=2}^{N_{total}}\delta(n,m)P_{DP}(n,k,m). \quad (20)$$

In Eqs. (19) and (20), SC stands for *static common*, SS for *static self*, SN for *static neighbor*, DC for *dynamic common*, DS for *dynamic self* and DN for *dynamic neighbor*. While the common and self parts of the hit ratios are not associated to the device mobility, the neighbor parts can be affected by the mobility of the helper and the neighboring devices.

The helper sorts $\rho_{content}(k)$ in descending order and then selects the K_{cache} largest elements to construct a set of content S_{cache} to be stored in its cache.

If there are multiple helpers in a cell, the content set needs to be determined by considering the content overlapping with the content sets of other helpers. It may be difficult to find an optimal solution in a general environment having multiple helpers. In this article, the objective is not to find the optimal caching method in a specific environment, but to understand which parts of the hit ratio are important in various environments. Hence, this article mainly describes the cases where there is only one helper in a cell, and discusses how the caching strategy may change when there are multiple helpers in the cell.

Given a set of content stored in the cache, S_{cache}, the *hit ratio* based on the content stored in the cache, denoted as ρ_{cache}, can be defined as

$$\rho_{cache} \equiv \frac{\sum_{k\in S_{cache}} R_{offloading}(k)}{R_{request}} = \rho_{SC}^{cache} + \rho_{SS}^{cache} + \rho_{SN}^{cache} + \rho_{DC}^{cache} + \rho_{DS}^{cache} + \rho_{DN}^{cache} \quad (21)$$

where

$$\begin{aligned}
\rho_{SC}^{cache} &\equiv \sum_{k \in S_{cache}} P_{SC}(k) \\
\rho_{SS}^{cache} &\equiv \frac{1}{N_{device}} \sum_{k \in S_{cache}} P_{SS}(1,k) \\
\rho_{SN}^{cache} &\equiv \frac{1}{N_{device}M} \sum_{m=1}^{M} \sum_{n=2}^{N_{total}} \sum_{k \in S_{cache}} \delta(n,m) P_{SN}(n,k) \\
\rho_{DC}^{cache} &\equiv \frac{1}{M} \sum_{m=1}^{M} \sum_{k \in S_{cache}} P_{DC}(k,m) \\
\rho_{DS}^{cache} &\equiv \frac{1}{N_{device}M} \sum_{m=1}^{M} \sum_{k \in S_{cache}} P_{DS}(1,k,m) \\
\rho_{DN}^{cache} &\equiv \frac{1}{N_{device}M} \sum_{m=1}^{M} \sum_{n=2}^{N_{total}} \sum_{k \in S_{cache}} \delta(n,m) P_{DN}(n,k,m). \quad (22)
\end{aligned}$$

In this article, we describe how much each of the six parts of the hit ratio, ρ_{SC}^{cache}, ρ_{SS}^{cache}, ρ_{SN}^{cache}, ρ_{DC}^{cache}, ρ_{DS}^{cache}, and ρ_{DN}^{cache}, can contribute to the hit ratio ρ_{cache}, which could explain what caching strategy each helper should utilize in a given environment.

F. CACHE PARTITIONING

In this article, the hit ratio is divided into the six parts, as shown in Fig. 6. Similarly, we can define the six categories of content corresponding to the six parts. Certainly, one piece of content can contribute to both self-offloading and D2D offloading, and content for neighbors can benefit all users as well. However, each piece of content can map to the part that has the greatest impact on the hit ratio. Based on the categories of content in the cache, we can see which categories are important for a given environment. The part with a large hit-ratio contribution may not always match the main category of content stored in the cache. In many cases, a content category that occupies a large portion of the cache area may have a large contribution to the hit ratio. However, if the preference values are concentrated only on a few pieces of content for a certain content category, only a small number of pieces of content can be stored for the category, while the part associated to the content category can greatly contribute to the hit ratio. Conversely, if a category has similar preference values that are not very large, then a large portion of the cache may need to be allocated for the category, while the hit ratio contribution may not be large.

To establish how much portion of the cache is primarily associated to each part of the hit ratio, the cache is divided into six areas by assigning each piece of content in the cache to the category corresponding to the part that contributes the most to the hit ratio, which can be written as follows:

$$\begin{aligned}
k \in S_{\alpha}^{cache} \quad & \text{where } \alpha = \arg\max_{\beta} \rho_{\beta}^{content}(k) \\
& k \in S_{cache} \\
& \alpha, \beta \in \{SC, SS, SN, DC, DS, DN\} \quad (23)
\end{aligned}$$

Using Eq. (23), S_{cache} with size K_{cache} can be divided into 6 sets of S_{SC}^{cache}, S_{SS}^{cache}, S_{SN}^{cache}, S_{DC}^{cache}, S_{DS}^{cache}, and S_{DN}^{cache} with size K_{SC}^{cache}, K_{SS}^{cache}, K_{SN}^{cache}, K_{DC}^{cache}, K_{DS}^{cache}, and K_{DN}^{cache}, respectively. Therefore, the following equation holds:

$$K_{SC} + K_{SS} + K_{SN} + K_{DC} + K_{DS} + K_{DN} = K_{total} \quad (24)$$

From these values, we can then check which categories of content should be primarily considered for storage. Cache partitioning is conceptual to access how the six categories affect caching. The cache area is not physically separated, pre-allocated, or used exclusively.

III. CACHING STRATEGIES

A. NOMADIC HELPER IN A CELL THAT IS ALWAYS OVERLOADED

In the section, we describe caching strategies for various environments. Throughout this article, we assume that the number of devices per cell is N_{device}. Consider a nomadic helper in a cell that is always overloaded during peak hours. If the helper remains in the cell during peak hours, then the cache cannot be updated. Suppose that device 1 is the nomadic helper considered for caching, and the peak-hour period M is very long, thus satisfying the conditions $M \geq M_{static}$ and $M \geq M_{move}^{nomadic}$. Using Eqs. (3) and (11), the static neighbor part of the hit ratio for content k in Eq. (20) can be approximated as follows:

$$\begin{aligned}
\rho_{SN}^{content}(k) &= \frac{1}{N_{device}M} \sum_{m=1}^{M} \sum_{n=2}^{N_{total}} \delta(n,m) P_{SP}(n,k) \\
&= \frac{1}{N_{device}} \sum_{n=2}^{N_{total}} P_{SP}(n,k) \frac{1}{M} \sum_{m=1}^{M} \delta(n,m) \\
&\approx \frac{1}{N_{device}} \sum_{n=2}^{N_{total}} P_{SP}(n,k) \frac{N_{device}-1}{N_{total}-1} \\
&\leq \frac{1}{N_{device}} \sum_{n=2}^{N_{total}} P_{SP}(n,k) \frac{N_{device}}{N_{total}} \\
&\leq \frac{1}{N_{total}} \sum_{n=1}^{N_{total}} P_{SP}(n,k) \approx 0 \quad (25)
\end{aligned}$$

From Eq. (12), the dynamic common part in Eq. (20) can be represented as

$$\rho_{DC}^{content}(k) = \frac{1}{M} \sum_{m=1}^{M} P_{DC}(k,m) \approx 0 \quad (26)$$

and the dynamic self part can be written as follows:

$$\begin{aligned}\rho_{DS}^{content}(k) &= \frac{1}{N_{device}M}\sum_{m=1}^{M} P_{DP}(1,k,m)\\ &\leq \frac{1}{N_{device}}\frac{1}{M}\sum_{m=1}^{M} P_{DP}^{\max}(k,m)\\ &< \frac{1}{M}\sum_{m=1}^{M} P_{DP}^{\max}(k,m) \approx 0 \end{aligned}\tag{27}$$

From Eqs. (1) and (12), the dynamic neighbor part can be approximated as follows:

$$\begin{aligned}\rho_{DN}^{content}(k) &= \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=2}^{N_{total}} \delta(n,m)P_{DP}(n,k,m)\\ &\leq \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=2}^{N_{total}} \delta(n,m)P_{DP}^{\max}(k,m)\\ &= \frac{N_{device}-1}{N_{device}}\frac{1}{M}\sum_{m=1}^{M} P_{DP}^{\max}(k,m)\\ &\leq \frac{1}{M}\sum_{m=1}^{M} P_{DP}^{\max}(k,m) \approx 0 \end{aligned}\tag{28}$$

Hence, the hit ratio for content k can be expressed as follows:

$$\begin{aligned}\rho_{content}(k) &\approx \rho_{SC}^{content}(k) + \rho_{SS}^{content}(k)\\ &\approx P_{SC}(k) + \frac{1}{N_{device}}P_{SP}(1,k) \end{aligned}\tag{29}$$

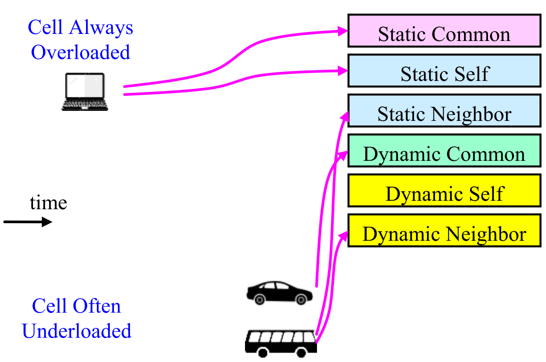

FIGURE 7. **Caching of a nomadic helper in a cell that is always overloaded.**

As shown in Fig. 7, the helper needs to store content based on the static common preferences and its own static personal preferences. It is difficult to achieve excellent performance especially when the cache size is small, since the helper cannot exploit the static neighbor and dynamic preferences. When the number of devices per cell is small, the contribution of the static self part to the hit ratio increases. As the number of devices per cell increases, the contribution of the static common part to the hit ratio increases. In a cell that is always overloaded, it is difficult to obtain excellent performance with only a nomadic helper, and the assistance of mobile helpers moving from other cells may be required, as shown in Fig. 7.

B. SOCIALLY CONNECTED DEVICES IN A CELL THAT IS ALWAYS OVERLOADED

Suppose that N_{social} devices including the helper are socially-connected, so that they will meet each other and remain in the same cell for most of the time when considering a very long time over M_{social}^{mobile} regardless of their location at time 0 [64]–[66]. Suppose that device 1 is a nomadic helper in a cell that is always overloaded, and devices 2 through N_{social} are socially connected with the helper. Assuming that the peak-hour period M is very long and satisfies the condition $M \geq M_{social}^{mobile}$, then the static neighbor part of the hit ratio for content k can be expressed using Eq. (5) as follows:

$$\begin{aligned}\rho_{SN}^{content}(k) &= \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=2}^{N_{total}} \delta(n,m)P_{SP}(n,k)\\ &= \frac{1}{N_{device}}\sum_{n=2}^{N_{social}} P_{SP}(n,k)\frac{1}{M}\sum_{m=1}^{M}\delta(n,m)\\ &\quad + \frac{1}{N_{device}}\sum_{n=N_{social}+1}^{N_{total}} P_{SP}(n,k)\frac{1}{M}\sum_{m=1}^{M}\delta(n,m)\\ &\approx \frac{1}{N_{device}}\sum_{n=2}^{N_{social}} P_{SP}(n,k) + \frac{1}{N_{device}}\\ &\quad \times \sum_{n=N_{social}+1}^{N_{total}} P_{SP}(n,k)\frac{N_{device}-N_{social}}{N_{total}-N_{social}} \end{aligned}\tag{30}$$

From Eq. (11), the second part of Eq. (30) can be approximated as follows:

$$\begin{aligned}&\frac{1}{N_{device}}\sum_{n=N_{social}+1}^{N_{total}} P_{SP}(n,k)\frac{N_{device}-N_{social}}{N_{total}-N_{social}}\\ &\leq \frac{1}{N_{device}}\sum_{n=N_{social}+1}^{N_{total}} P_{SP}(n,k)\frac{N_{device}}{N_{total}}\\ &\leq \frac{1}{N_{total}}\sum_{n=1}^{N_{total}} P_{SP}(n,k) \approx 0 \end{aligned}\tag{31}$$

From Eq. (31), Eq. (30) can be rewritten as

$$\rho_{SN}^{content}(k) \approx \frac{1}{N_{device}}\sum_{n=2}^{N_{social}} P_{SP}(n,k) \tag{32}$$

and the hit ratio for content k can be expressed as follows:

$$\begin{aligned}\rho_{content}(k) &\approx \rho_{SC}^{content}(k) + \rho_{SS}^{content}(k) + \rho_{SN}^{content}(k)\\ &\approx P_{SC}(k) + \frac{1}{N_{device}}\sum_{n=1}^{N_{social}} P_{SP}(n,k) \end{aligned}\tag{33}$$

Considering devices that are socially-connected, although cells are always overloaded, some neighbors can be taken care of, and the performance can be improved [62]–[64].

C. NOMADIC HELPER IN A CELL THAT IS OCCASIONALLY UNDERLOADED

Consider a nomadic helper in a cell that is usually overloaded but intermittently underloaded. We assume that the content update period M of the helper is not very short satisfying the condition $M \geq M_{static}$, and there are $N_{nomadic}$ nomadic devices per cell having little mobility satisfying the condition $M \leq M_{stay}^{nomadic}$. Suppose that device 1 is the nomadic helper and devices 2 through $N_{nomadic}$ are the other nomadic devices in the cell. Let $S_{nomadic}$ denote the set of the nomadic devices in the cells other than the target cell, and S_{mobile} denote the set of the mobile devices in all cells. Since $M \geq M_{static}$, we do not take care of the dynamic preferences from Eqs. (26), (27), and (28). Using Eq. (2), the static neighbor part of the hit ratio for content k can be expressed as follows:

$$\begin{aligned}\rho_{SN}^{content}(k) &= \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=2}^{N_{total}}\delta(n,m)P_{SP}(n,k)\\ &= \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=2}^{N_{nomadic}}\delta(n,m)P_{SP}(n,k)\\ &\quad+\frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n\in S_{nomadic}}\delta(n,m)P_{SP}(n,k)\\ &\quad+\frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n\in S_{mobile}}\delta(n,m)P_{SP}(n,k)\\ &\approx \frac{1}{N_{device}}\sum_{m=1}^{M}P_{SP}(n,k)\\ &\quad+\frac{1}{N_{device}N_{cell}}\sum_{m=1}^{M}\sum_{n\in S_{mobile}}P_{SP}(n,k) \qquad (34)\end{aligned}$$

From Eq. (11), the second part of Eq. (34) can be approximated as

$$\begin{aligned}\frac{1}{N_{device}N_{cell}}\sum_{n\in S_{mobile}}P_{SP}(n,k) &= \frac{1}{N_{total}}\sum_{n\in S_{mobile}}P_{SP}(n,k)\\ &\leq \frac{1}{N_{total}}\sum_{n=1}^{N_{total}}P_{SP}(n,k)\approx 0\end{aligned} \qquad (35)$$

and Eq. (34) can be rewritten as follows:

$$\rho_{SN}^{content}(k) \approx \frac{1}{N_{device}}\sum_{n=2}^{N_{nomadic}}P_{SP}(n,k) \qquad (36)$$

Hence, the hit ratio for content k can be represented as follows:

$$\begin{aligned}\rho_{content}(k) &\approx \rho_{SC}^{content}(k)+\rho_{SS}^{content}(k)+\rho_{SN}^{content}(k)\\ &\approx P_{SC}(k)+\frac{1}{N_{device}}\sum_{n=1}^{N_{nomadic}}P_{SP}(n,k) \qquad (37)\end{aligned}$$

As shown in Fig. 8, the helper stores content considering the static common and static personal preferences of the nomadic devices in the cell. Compared with Eq. (29), we can observe that the larger the number of nomadic devices, the larger the hit ratio, meaning that the performance can be enhanced if a helper can support many neighbors. In addition, if the number of nomadic devices becomes close to the number of devices per cell, the neighbor preferences may become paramount, especially when the number of devices per cell is small.

If the number of helpers in $N_{nomadic}$ nomadic devices in a cell is not one but N_{helper} ($\leq N_{nomadic}$), it can be considered that the cache size has increased by N_{helper} times in the cell. In this case, Eq. (37) can be rewritten as follows:

$$\begin{aligned}\rho_{content}(k) &\approx P_{SC}(k)+\frac{1}{N_{device}}\sum_{n=1}^{N_{helper}}P_{SP}(n,k)\\ &\quad+\frac{1}{N_{device}}\sum_{n=N_{helper}+1}^{N_{nomadic}}P_{SP}(n,k) \qquad (38)\end{aligned}$$

Assuming that each helper stores the content of static self category in its own cache, as the number of helpers increases, the importance of the self part increases while the importance of the neighbor part decreases.

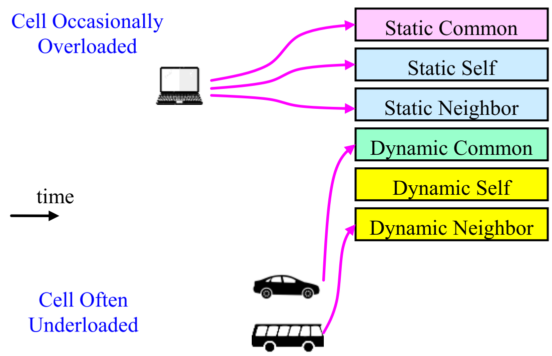

FIGURE 8. Caching of a nomadic helper in a cell that is occasionally underloaded.

Considering a cell that is usually overloaded, it is difficult to obtain good performance with only nomadic helpers since dynamic preferences are not taken care of. As shown in Fig. 8, it may require the assistance of mobile helpers from other cells having a low load.

D. NOMADIC HELPER IN A CELL THAT IS OFTEN UNDERLOADED

Consider a nomadic helper in a cell that is often in a low load state. We assume that the content update period M of the helper is very short satisfying the condition $M \leq M_{dynamic}$, and there are $N_{nomadic}$ nomadic devices per cell satisfying the condition $M \leq M_{stay}^{nomadic}$. Suppose that device 1 is the considered nomadic helper and devices 2 through $N_{nomadic}$ are the other nomadic devices in the cell. Let $S_{nomadic}$ denote the set of the nomadic devices in the cells other than the target cell, and S_{mobile} denote the set of the mobile devices in all cells. From Eq. (13), the dynamic common and dynamic self parts of the hit ratio for content k can be approximated as follows:

$$\rho_{DC}^{content}(k) = \frac{1}{M}\sum_{m=1}^{M} P_{DC}(k,m) \approx P_{DC}(k,0)$$
$$\rho_{DS}^{content}(k) = \frac{1}{N_{device}M}\sum_{m=1}^{M} P_{DP}(1,k,m) \approx \frac{1}{N_{device}} P_{DP}(1,k,0) \quad (39)$$

From Eqs. (2) and (13), the dynamic neighbor part of the hit ratio for content k can be expressed as follows:

$$\begin{aligned}\rho_{DN}^{content}(k) &= \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=2}^{N_{total}} \delta(n,m)P_{DP}(n,k,m) \\ &\approx \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=2}^{N_{total}} \delta(n,m)P_{DP}(n,k,0) \\ &= \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=2}^{N_{nomadic}} \delta(n,m)P_{DP}(n,k,0) \\ &\quad + \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n\in S_{nomadic}} \delta(n,m)P_{DP}(n,k,0) \\ &\quad + \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n\in S_{mobile}} \delta(n,m)P_{DP}(n,k,0) \\ &\approx \frac{1}{N_{device}}\sum_{n=2}^{N_{nomadic}} P_{DP}(n,k,0) \\ &\quad + s\frac{1}{N_{device}N_{cell}}\sum_{n\in S_{mobile}} P_{DP}(n,k,0) \end{aligned} \quad (40)$$

Using Eq. (11), the second part of Eq. (40) can be approximated as

$$\begin{aligned}\frac{1}{N_{device}N_{cell}}\sum_{n\in S_{mobile}} P_{DP}(n,k,0) &= \frac{1}{N_{total}}\sum_{n\in S_{mobile}} P_{DP}(n,k,0) \\ &\le \frac{1}{N_{total}}\sum_{n=1}^{N_{total}} P_{DP}(n,k,0) \approx 0 \end{aligned} \quad (41)$$

and Eq. (40) can be rewritten as follows:

$$\rho_{DN}^{content}(k) \approx \frac{1}{N_{device}}\sum_{n=2}^{N_{nomadic}} P_{DP}(n,k,0) \quad (42)$$

Hence, the hit ratio for content k can be expressed as follows:

$$\begin{aligned}\rho_{content}(k) &= \rho_{SC}^{content}(k) + \rho_{SS}^{content}(k) + \rho_{SN}^{content}(k) \\ &\quad + \rho_{DC}^{content}(k) + \rho_{DS}^{content}(k) + \rho_{DN}^{content}(k) \\ &\approx P_{SC}(k) + P_{DC}(k,0) \\ &\quad + \frac{1}{N_{device}}\sum_{n=1}^{N_{nomadic}} (P_{SP}(n,k) + P_{DP}(n,k,0)) \end{aligned} \quad (43)$$

The hit ratio includes all six parts, as shown in Fig. 9. Due to frequent content updates and low device mobility, the dynamic preferences can be considered and the hit ratio can be increased.

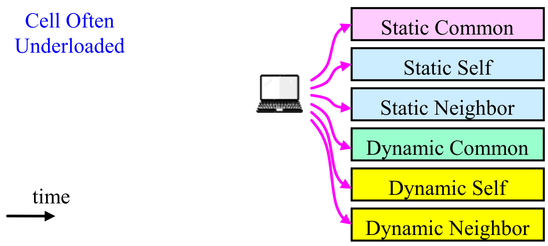

FIGURE 9. Caching of a nomadic helper in a cell that is often underloaded.

If the number of helpers among the $N_{nomadic}$ nomadic devices is increased to N_{helper} ($\le N_{nomadic}$), then it can be said that the cache size is increased by N_{helper} times in the cell, and $N_{helper}K_{cache}$ pieces of content can be stored. In this case, Eq. (43) can be rewritten as follows:

$$\begin{aligned}\rho_{content}(k) &\approx P_{SC}(k) + P_{DC}(k,0) \\ &\quad + \frac{1}{N_{device}}\sum_{n=1}^{N_{helper}} (P_{SP}(n,k) + P_{DP}(n,k,0)) \\ &\quad + \frac{1}{N_{device}}\sum_{n=N_{helper}+1}^{N_{nomadic}} (P_{SP}(n,k) + P_{DP}(n,k,0)) \end{aligned} \quad (44)$$

Compared to Eq. (43), the self parts becomes larger and the neighbor parts becomes smaller.

In the extreme case that all devices in the cell are nomadic helpers, if each helper stores content based on its own personal preferences, then the neighbor parts in Eq. (44) may disappear. Eq. (44) can be rewritten as

$$\begin{aligned}\rho_{content}(k) &\approx P_{SC}(k) + P_{DC}(k,0) \\ &\quad + \frac{1}{N_{device}}\sum_{n=1}^{N_{device}} (P_{SP}(n,k)+P_{DP}(n,k,0)) \end{aligned} \quad (45)$$

and the helpers do not consider the personal preferences of neighbors.

For simplicity of discussion, it is assumed that all users have the same amount of data request and the cache sizes of helpers are equal. In reality, the amount of data requested varies based on the device and the cache size may vary depending on the helper [47], [51], [67]. In this case, helpers with large cache sizes may be required to support neighbors with high data demands.

Consider a situation where there is no helper in a cell but all devices in the cell have a self-cache, in other words, all devices are UEs of type-II. Considering device n in the cell, the hit ratio for content k can be expressed as follows.

$$\begin{aligned}\rho_{content}(k) &\approx P_{SC}(k) + P_{DC}(k,0) \\ &\quad + P_{SP}(n,k) + P_{DP}(n,k,0) \end{aligned} \quad (46)$$

Compared to Eq. (45), there is no difference in that the personal preferences of neighbors are not considered. However, in the case of type-II UEs, the content stored in the cache of each device is not shared with other devices. Therefore, each device needs to take full care of the common category content.

E. MOBILE HELPER

If a sufficiently large percentage of the cells are often underloaded, then a mobile helper update the cache whenever it enters a cell in a low-load state, and content updates can occur frequently. Suppose that the content update period M is small, thus satisfying the condition $M \leq M_{dynamic}$. Then, Eq. (39) holds. From Eqs. (4) and (11), the static neighbor part of the hit ratio for content k can be approximated as follows:

$$\begin{aligned}\rho_{SN}^{content}(k) &= \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=2}^{N_{total}}\delta(n,m)P_{SP}(n,k)\\ &= \frac{1}{N_{device}}\sum_{n=2}^{N_{total}}P_{SP}(n,k)\frac{1}{M}\sum_{m=1}^{M}\delta(n,m)\\ &\approx \frac{1}{N_{device}}\sum_{n=2}^{N_{total}}P_{SP}(n,k)\frac{N_{device}-1}{N_{total}-1}\\ &\leq \frac{1}{N_{device}}\sum_{n=2}^{N_{total}}P_{SP}(n,k)\frac{N_{device}}{N_{total}}\\ &\leq \frac{1}{N_{total}}\sum_{n=1}^{N_{total}}P_{SP}(n,k)\approx 0 \end{aligned} \tag{47}$$

From Eqs. (4), (11), and (13), the dynamic neighbor part of the hit ratio for content k can be expressed as

$$\begin{aligned}\rho_{DN}^{content}(k) &= \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=2}^{N_{total}}\delta(n,m)P_{DP}(n,k,m)\\ &\approx \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=2}^{N_{total}}\delta(n,m)P_{DP}(n,k,0)\\ &= \frac{1}{N_{device}}\sum_{n=2}^{N_{total}}P_{DP}(n,k,0)\frac{1}{M}\sum_{m=1}^{M}\delta(n,m)\\ &= \frac{1}{N_{device}}\sum_{n=2}^{N_{total}}P_{DP}(n,k,0)\frac{N_{device}-1}{N_{total}-1}\\ &\leq \frac{1}{N_{device}}\sum_{n=2}^{N_{total}}P_{DP}(n,k,0)\frac{N_{device}}{N_{total}}\\ &\leq \frac{1}{N_{total}}\sum_{n=0}^{N_{total}}P_{DP}(n,k,0)\approx 0 \end{aligned} \tag{48}$$

and the hit ratio for content k can be represented as follows:

$$\begin{aligned}\rho_{content}(k) &\approx \rho_{SC}^{content}(k)+\rho_{SS}^{content}(k)\\ &\quad+\rho_{DC}^{content}(k)+\rho_{DS}^{content}(k)\\ &\approx P_{SC}(k)+P_{DC}(k,0)\\ &\quad+\frac{1}{N_{device}}(P_{SP}(1,k)+P_{DP}(1,k,0)) \end{aligned} \tag{49}$$

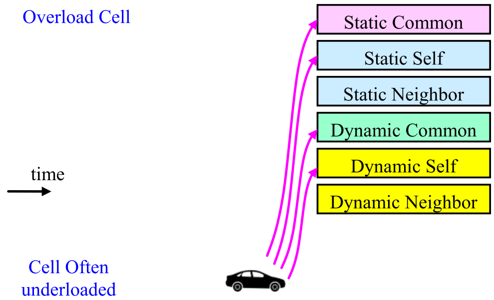

FIGURE 10. Caching of a mobile helper.

Since a mobile helper is constantly changing its neighbors, it is difficult to store content based on the neighbors' personal preferences. However, if a mobile helper frequently visits cells in a low-load state, the performance can be enhanced by considering the dynamic preferences, as shown in Fig. 10.

As shown in Figs. 7 and 8, a mobile helper can move after storing content in a cell that is in an underload condition and then supply content with high dynamic common preferences when entering a cell that is overloaded.

If there are multiple mobile helpers per cell, each of them can perform self-caching and may reduce performance degradation caused by not caring for neighbors. In addition, if there is a mobile UE of type-II, it can improve performance through self-caching considering the dynamic preferences, which are not taken care of by a nomadic type-II UE in a cell that is overloaded most of the time.

F. MOBILE HELPER WITH GROUP MOBILITY

Mobile devices do not necessarily move independently. If mobile devices are in the same place at time 0 and they are socially connected or in the same vehicle, it may be assumed that they move together within a short period of time M_{group}^{mobile} [64]–[66]. If a mobile helper often visits cells in a low-load state and updates the cache frequently, the content update period M may satisfy the conditions $M \leq M_{dynamic}$ and $M \leq M_{group}^{mobile}$. Suppose that device 1 is a mobile helper and devices 2 through N_{group} are UEs moving together with the helper. Using Eq. (6), the static neighbor part of the hit ratio for content k can be approximated as follows:

$$\begin{aligned}\rho_{SN}^{content}(k) &= \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=2}^{N_{total}}\delta(n,m)P_{SP}(n,k)\\ &= \frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=2}^{N_{group}}\delta(n,m)P_{SP}(n,k)\\ &\quad+\frac{1}{N_{device}M}\sum_{m=1}^{M}\sum_{n=N_{group}+1}^{N_{total}}\delta(n,m)P_{SP}(n,k)\\ &\approx \frac{1}{N_{device}}\sum_{n=2}^{N_{group}}P_{SP}(n,k)\\ &\quad+\frac{1}{N_{device}}\sum_{n=N_{group}+1}^{N_{total}}P_{SP}(n,k)\frac{N_{device}-N_{group}}{N_{total}-N_{group}} \end{aligned} \tag{50}$$

From Eq. (11), the second part of Eq. (50) can be expressed as

$$\begin{aligned}
&\frac{1}{N_{device}} \sum_{n=N_{group}+1}^{N_{total}} P_{SP}(n,k) \frac{N_{device}-N_{group}}{N_{total}-N_{group}} \\
&\leq \frac{1}{N_{device}} \sum_{n=N_{group}+1}^{N_{total}} P_{SP}(n,k) \frac{N_{device}}{N_{total}} \\
&\leq \frac{1}{N_{total}} \sum_{n=1}^{N_{total}} P_{SP}(n,k) \approx 0
\end{aligned} \tag{51}$$

and Eq. (50) can be rewritten as follows:

$$\rho_{SN}^{content}(k) \approx \frac{1}{N_{device}} \sum_{n=2}^{N_{group}} P_{SP}(n,k) \tag{52}$$

Similarly, the dynamic neighbor part of the hit ratio for content k can be written as

$$\begin{aligned}
\rho_{DN}^{content}(k) &= \frac{1}{N_{device}M} \sum_{m=1}^{M} \sum_{n=2}^{N_{total}} \delta(n,m) P_{DP}(n,k,m) \\
&\approx \frac{1}{N_{device}M} \sum_{m=1}^{M} \sum_{n=2}^{N_{total}} \delta(n,m) P_{DP}(n,k,0) \\
&\approx \frac{1}{N_{device}} \sum_{n=2}^{N_{group}} P_{DP}(n,k,0)
\end{aligned} \tag{53}$$

and the hit ratio for content k can be represented as

$$\begin{aligned}
\rho_{content}(k) &= \rho_{SC}^{content}(k) + \rho_{SS}^{content}(k) + \rho_{SN}^{content}(k) \\
&\quad + \rho_{DC}^{content}(k) + \rho_{DS}^{content}(k) + \rho_{DN}^{content}(k) \\
&\approx P_{SC}(k) + P_{DC}(k,0) \\
&\quad + \frac{1}{N_{device}} \sum_{n=1}^{N_{group}} (P_{SP}(n,k) + P_{DP}(n,k,0))
\end{aligned} \tag{54}$$

meaning that all six parts can be considered, as shown in Fig. 11. If devices move together, the performance might be improved through considering the personal preferences of the devices in the group.

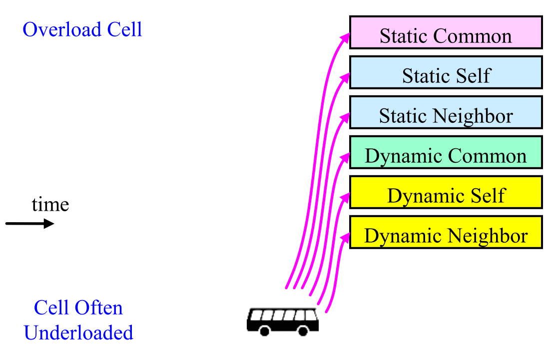

FIGURE 11. Caching of a mobile helper with group mobility or predicting the travel path.

G. MOBILE HELPER WITH PREDICTING THE TRAVEL PATH

Consider a vehicle that can predict the travel route, such as a bus or an autonomous vehicle. Suppose that the movement path of a mobile helper can be predicted for a short time $M_{predict}^{mobile}$, and the content update period M satisfies the conditions $M \leq M_{dynamic}$ and $M \leq M_{predict}^{mobile}$. Using Eqs. (7) and (13), the static and dynamic neighbor parts of the hit ratio for content k can be written as

$$\begin{aligned}
\rho_{SN}^{content}(k) &= \frac{1}{N_{device}M} \sum_{m=1}^{M} \sum_{n=2}^{N_{total}} \delta(n,m) P_{SP}(n,k) \\
&\approx \frac{1}{N_{device}} \sum_{n=2}^{N_{total}} \frac{M_{out}(n) - M_{in}(n)}{M} P_{SP}(n,k) \\
\rho_{DN}^{content}(k) &= \frac{1}{N_{device}M} \sum_{m=1}^{M} \sum_{n=2}^{N_{total}} \delta(n,m) P_{DP}(n,k,m) \\
&\approx \frac{1}{N_{device}M} \sum_{m=1}^{M} \sum_{n=2}^{N_{total}} \delta(n,m) P_{DP}(n,k,0) \\
&\approx \frac{1}{N_{device}} \sum_{n=2}^{N_{total}} \frac{M_{out}(n) - M_{in}(n)}{M} P_{DP}(n,k,0)
\end{aligned} \tag{55}$$

assuming that $M_{out}(n) - 1 \leq M$. The hit ratio for content k can be represented as follows:

$$\begin{aligned}
\rho_{content}(k) &= \rho_{SC}^{content}(k) + \rho_{SS}^{content}(k) + \rho_{SN}^{content}(k) \\
&\quad + \rho_{DC}^{content}(k) + \rho_{DS}^{content}(k) + \rho_{DN}^{content}(k) \\
&\approx P_{SC}(k) + P_{DC}(k,0) \\
&\quad + \frac{1}{N_{device}} (P_{SP}(1,k) + P_{DP}(1,k,0)) \\
&\quad + \sum_{n=2}^{N_{total}} \frac{M_{out}(n) - M_{in}(n)}{N_{device}M} \\
&\quad \times (P_{SP}(n,k) + P_{DP}(n,k,0))
\end{aligned} \tag{56}$$

If a mobile helper can predict its travel path, then it can take into account the personal preferences of the nomadic devices of the cells in the predicted path [28], [57].

If a mobile helper capable of predicting the movement path stores content by considering only a specific overload cell in the movement path, then it can only consider the data request at that particular time in the cell. Suppose that each cell has $N_{nomadic}$ nomadic devices and the mobile helper, which is device 1, remains in the target cell from M_{in}^{cell} to $M_{out}^{cell} - 1$. Let the nomadic helpers in the cell under consideration be devices 2 through $N_{nomadic} + 1$, the set of the nomadic helpers in the cells other than the target cell be $S_{nomadic}$, and the set of mobile devices other than the mobile helper be S_{mobile}. Using Eq. (8), the static neighbor part of the hit ratio for content k

can be written as follows:

$$
\begin{aligned}
&\rho_{SN}^{content}(k) \\
&= \frac{1}{N_{device}(M_{out}^{cell} - M_{in}^{cell})} \sum_{m=M_{in}^{cell}}^{M_{out}^{cell}-1} \sum_{n=2}^{N_{total}} \delta(n,m) P_{SP}(n,k) \\
&= \frac{1}{N_{device}(M_{out}^{cell} - M_{in}^{cell})} \sum_{m=M_{in}^{cell}}^{M_{out}^{cell}-1} \sum_{n=2}^{N_{nomadic}+1} \delta(n,m) P_{SP}(n,k) \\
&\quad + \frac{1}{N_{device}(M_{out}^{cell} - M_{in}^{cell})} \sum_{m=M_{in}^{cell}}^{M_{out}^{cell}-1} \sum_{n \in N_{nomadic}} \delta(n,m) P_{SP}(n,k) \\
&\quad + \frac{1}{N_{device}(M_{out}^{cell} - M_{in}^{cell})} \sum_{m=M_{in}^{cell}}^{M_{out}^{cell}-1} \sum_{n \in N_{mobile}} \delta(n,m) P_{SP}(n,k) \\
&\approx \frac{1}{N_{device}} \sum_{n=2}^{N_{nomadic}+1} P_{SP}(n,k) \\
&\quad + \frac{1}{N_{device}} \sum_{n \in S_{mobile}} P_{SP}(n,k) \frac{N_{device} - N_{nomadic} - 1}{(N_{device} - N_{nomadic})N_{cell} - 1}
\end{aligned} \tag{57}
$$

From Eq. (11), the second part of Eq. (57) can be approximated as

$$
\begin{aligned}
&\frac{1}{N_{device}} \sum_{n \in S_{mobile}} P_{SP}(n,k) \frac{N_{device} - N_{nomadic} - 1}{(N_{device} - N_{nomadic})N_{cell} - 1} \\
&\le \frac{1}{N_{device}} \sum_{n \in S_{mobile}} P_{SP}(n,k) \frac{N_{device} - N_{nomadic}}{(N_{device} - N_{nomadic})N_{cell}} \\
&= \frac{1}{N_{device}} \sum_{n \in S_{mobile}} P_{SP}(n,k) \frac{N_{device}}{N_{total}} \\
&\le \frac{1}{N_{total}} \sum_{n=1}^{N_{total}} P_{SP}(n,k) \approx 0
\end{aligned} \tag{58}
$$

and Eq. (57) can be rewritten as follows:

$$
\rho_{SN}^{content}(k) \approx \frac{1}{N_{device}} \sum_{n=2}^{N_{nomadic}+1} P_{SP}(n,k) \tag{59}
$$

Similarly, the dynamic neighbor part of the hit ratio for content k can be expressed as

$$
\begin{aligned}
&\rho_{DN}^{content}(k) \\
&= \frac{1}{N_{device}(M_{out}^{cell} - M_{in}^{cell})} \sum_{m=M_{in}^{cell}}^{M_{out}^{cell}-1} \sum_{n=2}^{N_{total}} \delta(n,m) P_{DP}(n,k,m) \\
&\approx \frac{1}{N_{device}(M_{out}^{cell} - M_{in}^{cell})} \sum_{m=M_{in}^{cell}}^{M_{out}^{cell}-1} \sum_{n=2}^{N_{total}} \delta(n,m) P_{DP}(n,k,0) \\
&\approx \frac{1}{N_{device}} \sum_{n=2}^{N_{nomadic}+1} P_{DP}(n,k,0)
\end{aligned} \tag{60}
$$

and the hit ratio for content k can be represented as follows:

$$
\begin{aligned}
\rho_{content}(k) &= \rho_{SC}^{content}(k) + \rho_{SS}^{content}(k) + \rho_{SN}^{content}(k) \\
&\quad + \rho_{DC}^{content}(k) + \rho_{DS}^{content}(k) + \rho_{DN}^{content}(k) \\
&\approx P_{SC}(k) + P_{DC}(k,0) \\
&\quad + \frac{1}{N_{device}} \sum_{n=1}^{N_{nomadic}+1} (P_{SP}(n,k) + P_{DP}(n,k,0))
\end{aligned} \tag{61}
$$

As shown in Figs. 7 and 8, a mobile helper that is capable of predicting the travel path can move and enter a cell that is overloaded in order to offload. For a cell that is always overloaded, the mobile helper can supply content with large static personal preferences or with large dynamic personal preferences. For a cell that is mostly overloaded but occasionally in an underload condition, the mobile helper can supply content with large dynamic preferences, which could be difficult for nomadic helpers in the cell to provide. If the degree of the cell load varies based on the region, the mobility of devices may be a crucial factor for performance improvement, and it may be desirable to determine whether some devices are socially connected or in the same vehicle, or whether mobile helpers can predict their movement path.

IV. NUMERICAL RESULTS

A. SIMULATION PARAMETERS

This section presents numerical results for the six parts of hit ratio and cache partitioning through simulation. In the following simulation results, the weights of the four parts of content preferences at time 0, P_{SC}^{sum}, $P_{SP}^{sum}(n)$, $P_{DC}^{sum}(0)$, and $P_{DP}^{sum}(n,0)$ are all 0.25, and the number of pieces of content considered for caching, K_{total} is 100,000. The four parts of content preferences of device n at time 0, $P_{SC}(k)$, $P_{SP}(n,k)$, $P_{DC}(k,0)$, and $P_{DP}(n,k,0)$ are all Zipf distributions with Zipf parameter of 0.8, but the numbers of non-zero elements, $K_{SP}^{Nonzero}(n)$, $K_{DC}^{Nonzero}$, and $K_{DP}^{Nonzero}(n)$ are limited to 500, 1000, and 20, respectively, while $K_{SC}^{Nonzero}$ is 100,000. This means that dynamic and/or personal preferences are assumed to have larger values on some specific content, while static common preference values are spread over a large number of pieces of content. The orders of the personal preferences are independent of the device. The cache size of a helper, K_{cache} is 100, and the number of devices per cell, N_{device} is limited to 15 or less.

B. NOMADIC HELPER IN A CELL THAT IS ALWAYS OVERLOADED

Figs. 12 and 13 present the hit ratio and cache partitioning, respectively, for the case of a nomadic helper in a cell that is always overloaded. The number of helpers is one, and the number of devices in the cell including the helper varies under the assumption that all devices are nomadic. The cache cannot be updated during peak hours, and the cache update period is assumed to be very long. Since the neighboring devices of the helper may vary during peak hours as a result

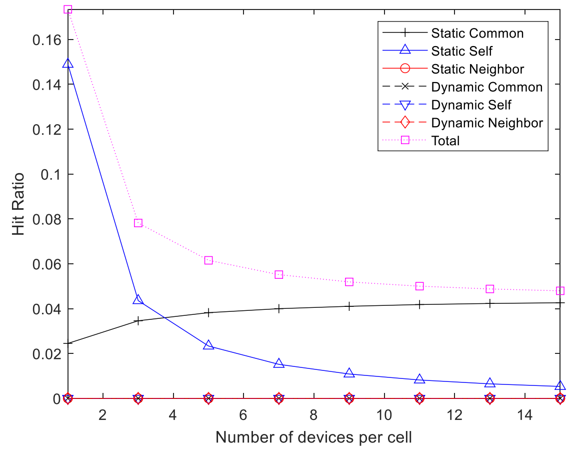

FIGURE 12. **Hit ratio of a nomadic helper in a cell that is always overloaded.**

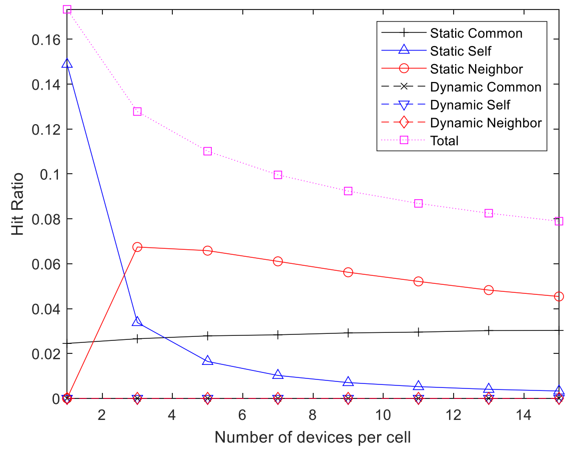

FIGURE 14. **Hit ratio of a nomadic helper in a cell that is occasionally underloaded.**

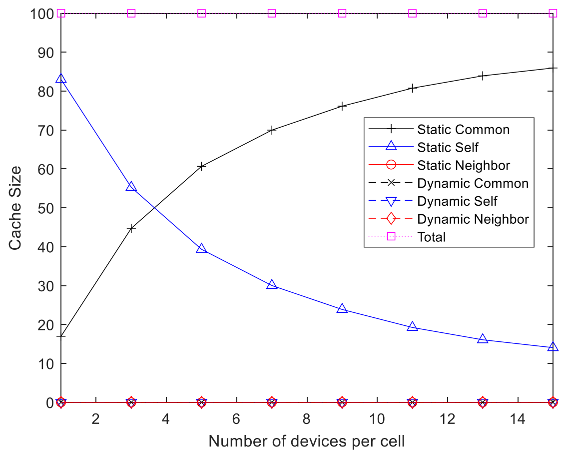

FIGURE 13. **Cache partitioning of a nomadic helper in a cell that is always overloaded.**

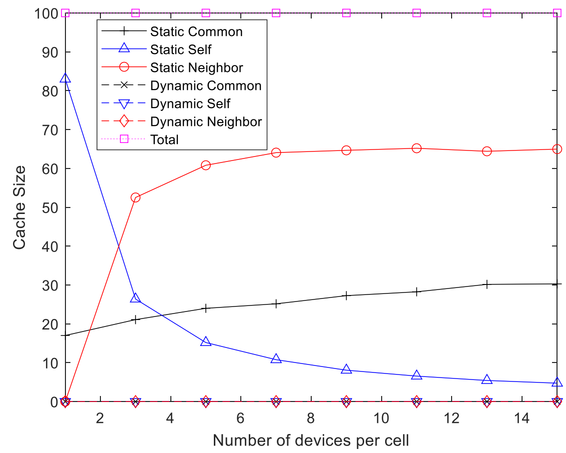

FIGURE 15. **Cache partitioning of a nomadic helper in a cell that is occasionally underloaded.**

of the device mobility, only the static common and static self parts can be considered. In Fig. 12, it can be seen that as the number of devices in the cell increases, the portion of self-requests among the total data requests decreases, thus causing a decrease in the static self part of the hit ratio. In Fig. 13, we observe that as the number of devices in the cell increases, the content should be stored considering the static common part, rather than the static self part.

C. NOMADIC HELPER IN A CELL THAT IS OCCASIONALLY UNDERLOADED

Figs. 14 and 15 illustrate the hit ratio and cache partitioning, respectively, for the case of a nomadic helper in a cell that occasionally goes into a low-load state. The number of nomadic helpers is one, and the number of devices in the cell including the helper varies under the assumption that all devices are nomadic. Since we assume that the content updates do not occur frequently enough to consider the dynamic preferences, content is stored in consideration of only the static preferences. In particular, it can be seen that content with static neighbor category is primarily stored in the cache, and the contribution of the static neighbor part to the hit ratio is high. As the number of devices in the cell increases, the contribution of self-caching decreases and the static self area of the cache also decreases. As the number of devices increases, the number of neighbors that need to be considered for storage increases, but the cache size is limited, thus the neighbor part of the hit ratio decreases.

D. NOMADIC HELPER IN A CELL THAT IS OFTEN UNDERLOADED

Figs. 16 and 17 show the hit ratio and cache partitioning, respectively, for the case of a nomadic helper in a cell that is frequently underloaded. The number of helpers is one, and the number of devices in the cell including the helper varies based on the assumption that all devices are nomadic. Fig. 16 shows that if the cache can be updated frequently and thus dynamic preferences can be considered for storage, then the hit ratio can be greatly improved. As the number of devices increases, the number of neighbors also increases, and thus the importance of the self parts decreases. Since the size of the cache is limited, the neighbor parts of the hit ratio are not improved with an increase in the number of devices. When performing cache partitioning, the cache size is divided into small integers, and the curves do not appear smooth regardless of the number of simulations. However,

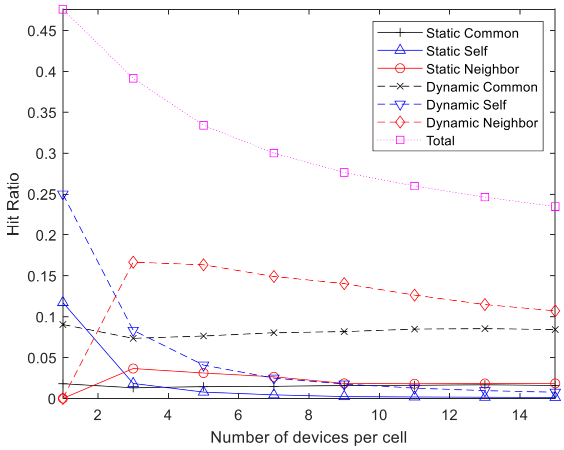

FIGURE 16. Hit ratio of a nomadic helper in a cell that is often underloaded.

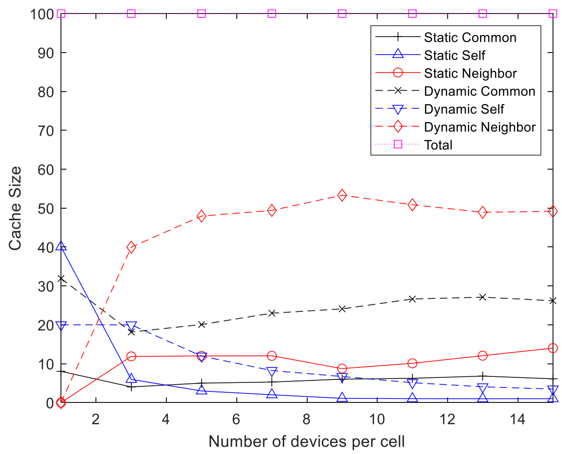

FIGURE 17. Cache partitioning of a nomadic helper in a cell that is often underloaded.

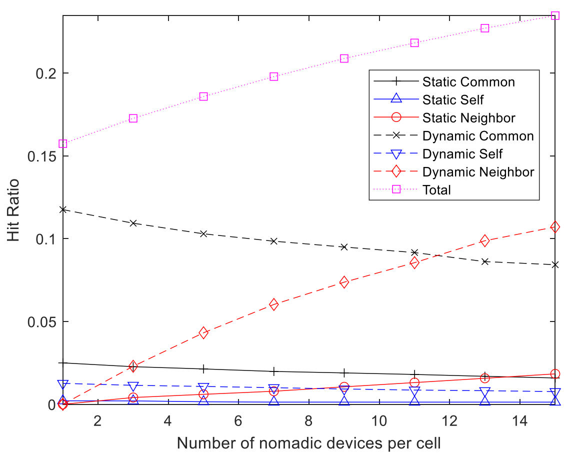

FIGURE 18. Hit ratio based on the number of nomadic devices.

it is possible to understand the characteristics of the caching system from the figures.

In Figs. 18 and 19, the number of nomadic helpers is one, the number of devices in the cell is 15, and the number of nomadic devices in the cell including the helper varies. The other devices in the cell are mobile. Since the helper can utilize the personal preferences of the nomadic devices, the hit

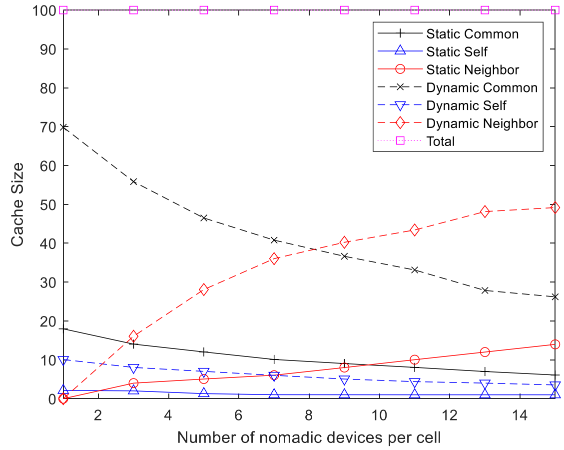

FIGURE 19. Cache partitioning based on the number of nomadic devices.

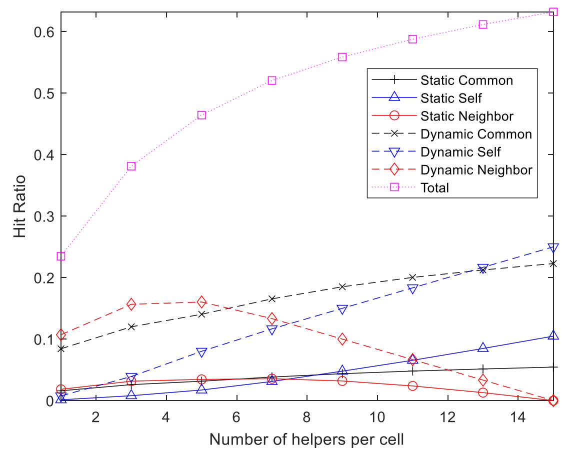

FIGURE 20. Hit ratio based on the number of nomadic helpers.

ratio increases with the number of nomadic devices in the cell. When the number of devices per cell is fixed, increasing the number of nomadic devices emphasizes the importance of the neighbor parts, since the proportion of neighbors supported by the helper among all devices increases.

Figs. 20 and 21 show the hit ratio and cache partitioning, respectively, when the number of devices in the cell is 15, and all are nomadic, while the number of helpers varies. If the content updates occur frequently and the devices have little mobility, then the effective size of the cache in the cell increases with the number of helpers, thus improving the hit ratio. If all the devices in the cell are helpers, then each device can consider its own personal preferences, eliminating the need to consider the neighbors' personal preferences. Therefore, as the number of helpers increases, the importance of the neighbor parts decreases, while the importance of the self and common parts increases. When all the devices are helpers, the neighbor parts diminish.

E. MOBILE HELPER

Figs. 22 and 23 show the hit ratio and cache partitioning, respectively, for a mobile helper. If the percentage of cells that frequently go under a low-load condition is large enough, then

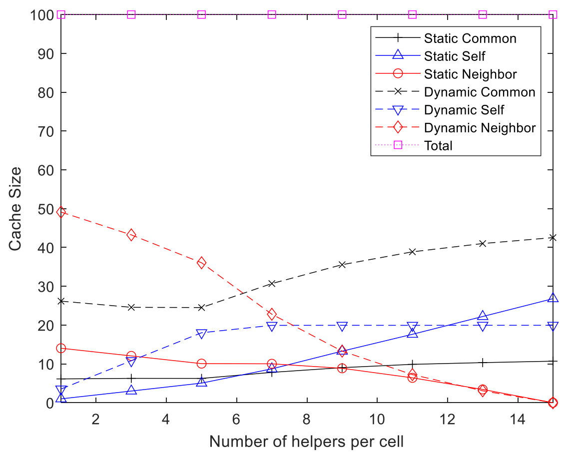

FIGURE 21. Cache partitioning based on the number of nomadic helpers.

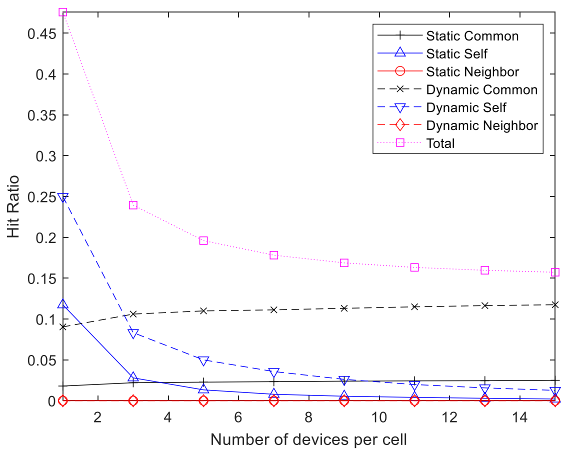

FIGURE 22. Hit ratio of a mobile helper.

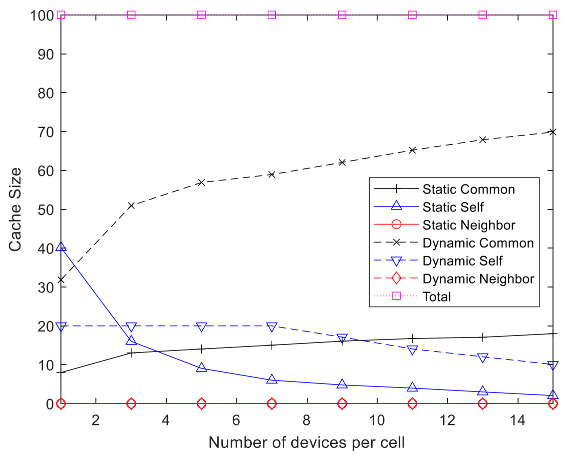

FIGURE 23. Cache partitioning of a mobile helper.

a mobile helper can frequently enter a cell in an underload state. In that case, the content update period is very short, and the dynamic preferences can be considered. However, if the mobile devices cannot predict the movement path and there is no other device moving together with the helper, it is difficult to identify the helper's neighbors, and the neighbor preferences cannot be considered. In Figs. 22 and 23, the number of helpers is one, and the number of devices per cell including the helper varies. As the number of devices increases, the importance of the common parts also increases, while the importance of the self parts decreases. Compared to Figs. 12 through 17, the importance of the dynamic common part is significant, and the mobile helper can store content mainly considering the dynamic common preferences.

If a mobile helper can move with other devices forming a group or if the mobile helper can predict the movement path, then the results similar to those in the previous section will be produced, and thus separate simulation results are not included.

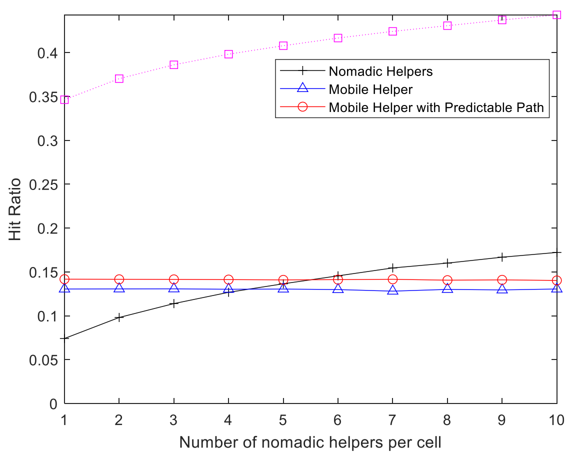

FIGURE 24. Hit ratio when nomadic and mobile helpers cooperate.

F. COOPERATION BETWEEN NOMADIC AND MOBILE HELPERS

In cells that cannot go into an underload state frequently, nomadic helpers alone may not achieve satisfactory results, since they cannot take into consideration the dynamic preferences for caching. In this case, nomadic and mobile helpers may divide their roles by considering different parts of content preferences. Fig. 24 shows the hit ratio of a cell that occasionally enters an underload state. The cell has 15 devices. Among them, 10 devices are nomadic and 5 devices are mobile. Two of the five mobile devices are helpers, with one that can predict the movement path while the other cannot. In the figure, the number of helpers out of 10 nomadic devices is changing, and the total number of helpers in the cell is the number of nomadic helpers plus two. While the nomadic helpers mainly consider the static preferences, the mobile helper with an unknown path provides content with large dynamic common preferences to all devices, and the mobile helper with a predictable path can handle the dynamic personal preferences of the nomadic devices in the target cell. As shown in Fig. 24, it is difficult to achieve excellent performance with only the nomadic helpers, and the hit ratio can be enhanced when cooperating with the mobile helpers.

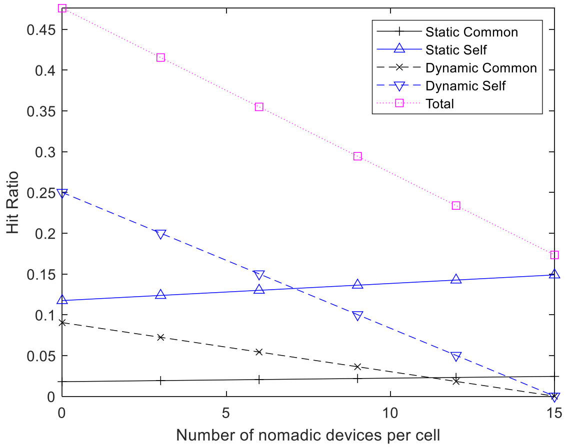

FIGURE 25. Hit ratio with self-caching.

G. DEVICE WITH A SELF-CACHE

When a device provides content to other devices using D2D communication, there can be various problems such as power consumption, security, copyright, etc., and the number of helpers may be limited at the onset of D2D caching systems. However, if there are many devices capable of self-caching, then the performance can be improved although they do not deliver content to other devices. Fig. 25 shows the performance when there are 15 type-II UEs in a cell that is always overloaded. Since the devices do not deliver content to others, the neighbor parts of the hit ratio are not considered. Among 15 devices, the number of nomadic devices varies as in the figure, while the others are mobile. We assume that the nomadic devices consider the static common and static self parts because it is not easy to update the cache. It is also assumed that the mobile devices consider the static common, static self, dynamic common, and dynamic self parts owing to frequent content updates. Since the mobile devices can consider the dynamic preferences, when the number of nomadic devices is small, in other words, when there are many mobile devices, the hit ratio is high and the contribution of the dynamic preferences is significant.

V. CONCLUSION

In this article, we examined how a helper should store content in its cache according to the environment. To this end, content preferences are divided into four parts: static common, static personal, dynamic common and dynamic personal preferences. In consideration of self-offloading and D2D offloading in addition to the four parts of content preferences, the hit ratio is divided into six parts: static common, static self, static neighbor, dynamic common, dynamic self and dynamic neighbor parts. We focused on the aspects of caching and the contribution of the six parts to the hit ratio based on the cell load and the mobility of devices. As a result, we can see that the caching strategy of a helper should be determined based on many factors such as the cell load, the number of devices in the cell, the number of nomadic devices in the cell, the number of helpers in the cell, the mobility of the helper, and the mobility of other devices. When compared to nomadic helpers, mobile helpers can play different roles in regards to the hit ratio and performance can be improved through their cooperation. The effectiveness of self-caching devices was also confirmed.

In this article, the system model is simplified for ease of discussion. In order to derive a more accurate caching strategy, it is necessary to use a more generalized system model, in addition to considering system optimization in more diverse environments.

REFERENCES

[1] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Commun. Mag.*, vol. 51, no. 4, pp. 142–149, Apr. 2013.

[2] M. Gregori, J. Gomez-Vilardebo, J. Matamoros, and D. Gunduz, "Wireless content caching for small cell and D2D networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1222–1234, May 2016.

[3] M. Zhang, H. Luo, and H. Zhang, "A survey of caching mechanisms in information-centric networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1473–1499, 3rd Quart., 2015.

[4] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, Mar. 2017.

[5] S. A. Busari, S. Mumtaz, S. Al-Rubaye, and J. Rodriguez, "5G millimeter-wave mobile broadband: Performance and challenges," *IEEE Commun. Mag.*, vol. 56, no. 6, pp. 137–143, Jun. 2018.

[6] J. Jeon, "NR wide bandwidth operations," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 42–46, Mar. 2018.

[7] Z. Chen, X. Ma, B. Zhang, Y. X. Zhang, Z. Niu, N. Kuang, W. Chen, L. Li, and S. Li, "A survey on terahertz communications," *China Commun.*, vol. 16, no. 2, pp. 1–35, Feb. 2019.

[8] S. Andreev, V. Petrov, M. Dohler, and H. Yanikomeroglu, "Future of ultra-dense networks beyond 5G: Harnessing heterogeneous moving cells," *IEEE Commun. Mag.*, vol. 57, no. 6, pp. 86–92, Jun. 2019.

[9] M. Jaber, M. A. Imran, R. Tafazolli, and A. Tukmanov, "5G backhaul challenges and emerging research directions: A survey," *IEEE Access*, vol. 4, pp. 1743–1766, May 2016.

[10] M. Feng and S. Mao, "Dealing with limited backhaul capacity in millimeter-wave systems: A deep reinforcement learning approach," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 50–55, Mar. 2019.

[11] K. Zhu, W. Zhi, X. Chen, and L. Zhang, "Socially motivated data caching in ultra-dense small cell networks," *IEEE Netw.*, vol. 31, no. 4, pp. 42–48, Jul. 2017.

[12] Z. Yan, S. Chen, Y. Ou, and H. Liu, "Energy efficiency analysis of cache-enabled two-tier HetNets under different spectrum deployment strategies," *IEEE Access*, vol. 5, pp. 6791–6800, Jun. 2017.

[13] W. Wu, N. Zhang, N. Cheng, Y. Tang, K. Aldubaikhy, and X. Shen, "Beef up mmWave dense cellular networks with D2D-assisted cooperative edge caching," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3890–3904, Apr. 2019.

[14] L. Li, G. Zhao, and R. S. Blum, "A survey of caching techniques in cellular networks: Research issues and challenges in content placement and delivery strategies," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1710–1732, 3rd Quart., 2018.

[15] E. Zeydan, E. Bastug, M. Bennis, M. A. Kader, I. A. Karatepe, A. S. Er, and M. Debbah, "Big data caching for networking: Moving from cloud to edge," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 36–42, Sep. 2016.

[16] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: Design aspects, challenges, and future directions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 22–28, Sep. 2016.

[17] T. Zhang, X. Fang, Y. Liu, and A. Nallanathan, "Content-centric mobile edge caching," *IEEE Access*, vol. 8, pp. 11722–11731, Jan. 2020.

[18] J. Poderys, M. Artuso, C. M. Oest Lensbol, H. L. Christiansen, and J. Soler, "Caching at the mobile edge: A practical implementation," *IEEE Access*, vol. 6, pp. 8630–8637, Mar. 2018.

[19] Y. Chen, Y. Liu, J. Zhao, and Q. Zhu, "Mobile edge cache strategy based on neural collaborative filtering," *IEEE Access*, vol. 8, pp. 18475–18482, Jan. 2020.

[20] Y. Shen, C. Jiang, T. Q. S. Quek, and Y. Ren, "Device-to-device-assisted communications in cellular networks: An energy efficient approach in downlink video sharing scenario," *IEEE Trans. Wireless Commun.*, vol. 15, no. 2, pp. 1575–1587, Feb. 2016.

[21] L. Li, Y. Xu, J. Yin, W. Liang, X. Li, W. Chen, and Z. Han, "Deep reinforcement learning approaches for content caching in cache-enabled D2D networks," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 544–557, Jan. 2020.

[22] J. Song and W. Choi, "Mobility-aware content placement for device-to-device caching systems," *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, pp. 3658–3668, Jul. 2019.

[23] B. Chen, C. Yang, and Z. Xiong, "Optimal caching and scheduling for cache-enabled D2D communications," *IEEE Commun. Lett.*, vol. 21, no. 5, pp. 1155–1158, May 2017.

[24] Z. Chen, N. Pappas, and M. Kountouris, "Probabilistic caching in wireless D2D networks: Cache hit optimal versus throughput optimal," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 584–587, Mar. 2017.

[25] N. Golrezaei, P. Mansourifard, A. F. Molisch, and A. G. Dimakis, "Base-station assisted device-to-device communications for high-throughput wireless video networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 7, pp. 3665–3676, Jul. 2014.

[26] R. Wang, X. Peng, J. Zhang, and K. B. Letaief, "Mobility-aware caching for content-centric wireless networks: Modeling and methodology," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 77–83, Aug. 2016.

[27] J. Yao, T. Han, and N. Ansari, "On mobile edge caching," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2525–2553, 3rd Quart., 2019.

[28] M. Rim and C. G. Kang, "Peak-hour caching schemes of mobile devices for overload cells in wireless caching systems," *IEEE Access*, vol. 8, pp. 141331–141341, Aug. 2020.

[29] M. Mehrabi, D. You, V. Latzko, H. Salah, M. Reisslein, and F. H. P. Fitzek, "Device-enhanced MEC: Multi-access edge computing (MEC) aided by end device computation and caching: A survey," *IEEE Access*, vol. 7, pp. 166079–166108, Nov. 2019.

[30] L. Shi, L. Zhao, G. Zheng, Z. Han, and Y. Ye, "Incentive design for cache-enabled D2D underlaid cellular networks using stackelberg game," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 765–779, Jan. 2019.

[31] K. N. Doan, T. V. Nguyen, H. Shin, and T. Q. S. Quek, "Socially-aware caching in wireless networks with random D2D communications," *IEEE Access*, vol. 7, pp. 58394–58406, May 2019.

[32] J. Yan, D. Wu, and R. Wang, "Socially aware trust framework for multimedia delivery in D2D cooperative communication," *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 625–635, Mar. 2019.

[33] W. Jiang, G. Feng, S. Qin, T. S. P. Yum, and G. Cao, "Multi-agent reinforcement learning for efficient content caching in mobile D2D networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1610–1622, Mar. 2019.

[34] J. Xu, L. Chen, K. Liu, and C. Shen, "Designing security-aware incentives for computation offloading via device-to-device communication," *IEEE Trans. Wireless Commun.*, vol. 17, no. 9, pp. 6053–6066, Sep. 2018.

[35] X. Chen, X. Song, C. Jia, and T. Cao, "Device to device networks with cache-enabled and self-sustained mobile helpers," *IEEE Access*, vol. 7, pp. 93820–93829, Jul. 2019.

[36] X. Song, Y. Geng, X. Meng, J. Liu, W. Lei, and Y. Wen, "Cache-enabled device to device networks with contention-based multimedia delivery," *IEEE Access*, vol. 5, pp. 3228–3239, Mar. 2017.

[37] Y. Qu, C. Dong, H. Dai, Z. Wei, and Q. Wu, "Maximizing D2D-based offloading efficiency with throughput guarantee and buffer constraint," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 832–842, Jan. 2019.

[38] M. Rim, S. Chae, and C. G. Kang, "Interference mitigation and D2D parameter estimation for distributed-control D2D underlay systems," *Trans. Emerg. Telecommun. Technol.*, vol. 28, no. 1, pp. 1–10, Jan. 2017.

[39] M. Rim and C. G. Kang, "Carrier sensing for OFDMA-based D2D group-casting systems," *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2301–2310, Mar. 2017.

[40] J. Lee, S. H. Lee, M. Rim, and C. G. Kang, "System-level spatiotemporal offloading with inter-cell mobility model for device-to-device (D2D) communication-based mobile caching in cellular network," *IEEE Access*, vol. 8, pp. 51570–51581, Mar. 2020.

[41] S. Zhang, J. Liu, H. Guo, M. Qi, and N. Kato, "Envisioning device-to-device communications in 6G," *IEEE Netw.*, vol. 34, no. 3, pp. 86–91, May 2020.

[42] S. Zhang, H. Zhang, and L. Song, "Beyond D2D: Full dimension UAV-to-everything communications in 6G," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6592–6602, Jun. 2020.

[43] G. Zhao, S. Chen, L. Qi, L. Zhao, and L. Hanzo, "Mobile-traffic-aware offloading for energy- and spectral-efficient large-scale D2D-enabled cellular networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 6, pp. 3251–3264, Jun. 2019.

[44] Y. Qiao, Y. Li, and J. Li, "An economic incentive for D2D assisted offloading using stackelberg game," *IEEE Access*, vol. 8, pp. 136684–136696, Jul. 2020.

[45] J. Rao, H. Feng, C. Yang, Z. Chen, and B. Xia, "Optimal caching placement for D2D assisted wireless caching networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.

[46] Q. Li, Y. Zhang, A. Pandharipande, X. Ge, and J. Zhang, "D2D-assisted caching on truncated zipf distribution," *IEEE Access*, vol. 7, pp. 13411–13421, Feb. 2019.

[47] B. Chen, C. Yang, and A. F. Molisch, "Cache-enabled device-to-device communications: Offloading gain and energy cost," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4519–4536, Jul. 2017.

[48] M.-C. Lee, A. F. Molisch, N. Sastry, and A. Raman, "Individual preference probability modeling and parameterization for video content in wireless caching networks," *IEEE/ACM Trans. Netw.*, vol. 27, no. 2, p. 676 690, Apr. 2019.

[49] A. Tripathi, T. S. Ashwin, and R. M. R. Guddeti, "EmoWare: A context-aware framework for personalized video recommendation using affective video sequences," *IEEE Access*, vol. 7, pp. 51185–51200, Apr. 2019.

[50] P. Cheng, C. Ma, M. Ding, Y. Hu, Z. Lin, Y. Li, and B. Vucetic, "Localized small cell caching: A machine learning approach based on rating data," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1663–1676, Feb. 2019.

[51] D. Liu and C. Yang, "Caching at base stations with heterogeneous user demands and spatial locality," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1554–1569, Feb. 2019.

[52] B. Chen and C. Yang, "Caching policy for cache-enabled D2D communications by learning user preference," *IEEE Trans. Commun.*, vol. 66, no. 12, pp. 6586–6601, Dec. 2018.

[53] T. Zhang, H. Fan, J. Loo, and D. Liu, "User preference aware caching deployment for Device-to-Device caching networks," *IEEE Syst. J.*, vol. 13, no. 1, pp. 226–237, Mar. 2019.

[54] M. Ma and V. W. S. Wong, "An optimal peak hour content server cache update scheduling algorithm for 5G HetNets," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019.

[55] M.-C. Lee, H. Feng, and A. F. Molisch, "Dynamic caching content replacement in base station assisted wireless D2D caching networks," *IEEE Access*, vol. 8, pp. 33909–33925, Feb. 2020.

[56] Z. Ding, P. Fan, G. K. Karagiannidis, R. Schober, and H. V. Poor, "NOMA assisted wireless caching: Strategies and performance analysis," *IEEE Trans. Commun.*, vol. 66, no. 10, pp. 4854–4876, Oct. 2018.

[57] M. Rim and C. G. Kang, "Content prefetching of mobile caching devices in cooperative D2D communication systems," *IEEE Access*, vol. 8, pp. 141331–141341, Aug. 2020.

[58] Y. Fan, B. Yang, D. Hu, X. Yuan, and X. Xu, "Social- and content-aware prediction for video content delivery," *IEEE Access*, vol. 8, pp. 29219–29227, Feb. 2020.

[59] I. H. Mwinyi, H. S. Narman, K.-C. Fang, and W.-S. Yoo, "Predictive self-learning content recommendation system for multimedia contents," in *Proc. Wireless Telecommun. Symp. (WTS)*, Apr. 2018, pp. 1–6.

[60] P. Sermpezis, T. Giannakas, T. Spyropoulos, and L. Vigneri, "Soft cache hits: Improving performance through recommendation and delivery of related content," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 6, pp. 1300–1313, Jun. 2018.

[61] W. Hu, Y. Jin, Y. Wen, Z. Wang, and L. Sun, "Toward Wi-Fi AP-assisted content prefetching for an on-demand TV series: A learning-based approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 7, pp. 1665–1676, Jul. 2018.

[62] Z. Zhao, L. Guardalben, M. Karimzadeh, J. Silva, T. Braun, and S. Sargento, "Mobility prediction-assisted over-the-top edge prefetching for hierarchical VANETs," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1786–1801, Aug. 2018.

[63] M. Wang, C. Xu, X. Chen, H. Hao, L. Zhong, and S. Yu, "Differential privacy oriented distributed online learning for mobile social video prefetching," *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 636–651, Mar. 2019.

[64] R. Zhang, S. Jia, Y. Ma, and C. Xu, "Social-aware D2D video delivery method based on mobility similarity measurement in 5G ultra-dense network," *IEEE Access*, vol. 8, pp. 52413–52427, Mar. 2020.

[65] M. N. Soorki, W. Saad, M. H. Manshaei, and H. Saidi, "Social community-aware content placement in wireless device-to-device communication networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 8, pp. 1938–1950, Aug. 2019.

[66] M. Ahmed, Y. Li, M. Waqas, M. Sheraz, D. Jin, and Z. Han, "A survey on socially aware device-to-device communications," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2169–2197, 3rd Quart., 2018.

[67] Y. Lin, Z. Lin, P. Chen, Z. Chen, and L. Wu, "On consideration of content and memory sizes in 5G D2D-assisted caching networks," *IEEE Access*, vol. 8, pp. 52759–52773, Jan. 2020.