Computer Communications 83 (2016) 45–55

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Joint storage assignment for D2D offloading systems

Wei Wang*, Xiaobing Wu, Lei Xie, Sanglu Lu

State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China

ARTICLE INFO

Article history:
Received 3 September 2014
Revised 18 December 2015
Accepted 22 February 2016
Available online 3 March 2016

Keywords:
Device-to-Device communication
DTN
Opportunistic networks

ABSTRACT

D2D offloading reduces the load of cellular network by asking mobile nodes to download content directly from storage of neighboring helpers via short range links. In this paper, we introduce a novel storage assignment scheme that can enhance storage utilization for D2D networks that have different types of storage nodes. Unlike traditional D2D systems that only use storage as static content cache, our scheme uses on-demand relaying to enhance storage utilization. Our on-demand relaying scheme replicates rare content when it is requested. Therefore, the proposed scheme can greatly increase the amount of content supported by the offloading system. We develop a convex optimization based algorithm to find the optimal storage assignment tradeoff between static caching and on-demand relaying. Numerical results and real-world trace-driven simulations show that our algorithm can achieve 30% reduction in offloading failure rate compared to static schemes.

1. Introduction

Mobile traffic has increased at a compound annual growth rate of more than 70% in recent years [1]. Meeting such a large surge in traffic demand is a great challenge for both cellular network designers and operators. One way to relieve the pressure on current cellular networks is to offload part of the mobile traffic through Device-to-Device (D2D) communication [2]. Because most mobile traffic comes from content downloading [1], one of the important offloading targets is mobile content, which includes application updates and video clips. In D2D offloading systems for content downloading, mobile *helpers* serve as content caches for other nodes. These helpers directly deliver content to neighboring devices through short-range and low-cost wireless links (e.g., Bluetooth, WiFi Direct or LTE D2D links [3]). In this way, the downloading traffic does not pass through base stations and core networks so that the limited resource of cellular networks can be saved [4].

Existing D2D offloading systems "statically" assign content to mobile helpers based on content popularity [5,6]. *Subscribers* need to directly contact mobile helpers that have the content of interest in their cache. Such static caching schemes have limited scalability with respect to network size. Consider the case where a subscriber called Alice wishes to download a movie through the D2D offloading system. Due to limited mobility, Alice can only contact a small number of mobile helpers before she becomes impatient and starts downloading the movie directly through the cellular network [5]. To make offloading successful, the movie must be cached in at least one of the helpers in a small virtual region that Alice can contact before she becomes impatient. The size of the "region" is determined by the user mobility pattern and the time that Alice can wait. Because the total storage contributed by helpers in this virtual region is limited, the offloading system can only provide a limited amount of content to Alice. Even if the offloading system covers a large area and has a huge number of helpers, it can only offload a constant amount of content due to the locality of the content in the static caching scheme.

In this paper, we improve the scalability of D2D systems by introducing different roles to mobile helpers. We assign some of the mobile helpers as on-demand *relays* instead of static caches, which are referred to as *seeds* in this paper. Due to the inherent non-uniformity in the contact pattern, relays may have a higher contact probability to the subscriber than seeds. For example, Alice may request her friends to act as mobile relays to download the movie. Although it might be difficult for Alice to directly contact a seed, there will be a higher chance that one of her friends can successfully download the movie. Afterwards, Alice can download the movie from her friends when she intentionally or accidentally meets them. In this way, the number of helpers that Alice can contact is effectively enlarged with the help of her friends.

In addition to using friends as relays, mobile nodes mounted on buses or static nodes deployed at public places (e.g., entrance of subways) can also act as relay nodes. These types of nodes have regular contact patterns to a given subset of subscribers. For example, a subscriber may pass by a given subway station every day. If

* Corresponding author. Tel.: +86 13851658076.
E-mail addresses: ww@nju.edu.cn, wangwei.ww@gmail.com (W. Wang), wuxb@nju.edu.cn (X. Wu), lxie@nju.edu.cn (L. Xie), sanglu@nju.edu.cn (S. Lu).

http://dx.doi.org/10.1016/j.comcom.2016.02.012
0140-3664/

there is a relay deployed at the subway station, this relay can actively search for the content requested by the subscriber and make sure the content can be delivered when the user passes by. Our analysis shows that using one relay with regular contact patterns can be as effective as increasing the number of seeds by a constant ratio.

An important issue in a relaying system is to allocate storage for content. Because users can choose to statically cache the content or dynamically download new content for their friends, mobile relays and seeds are actually different roles played by the same group of helpers. In this way, relays and seeds need to share the limited storage resources in mobile helpers and storage allocation becomes critical to the performance of the offloading system. When too many helpers are acting as seeds, there will be an insufficient amount of relays to "ferry" content between seeds and subscribers. Conversely, if there were too few seeds storing the requested content, it would be difficult for relays to find a seed, making relaying useless. To find the optimal tradeoff point, we introduce the concept of *offloading efficiency* to incorporate mobile relays and seeds in the same optimization framework. We prove that the storage optimization problem is convex for various types of contact patterns and can be efficiently solved in a distributed manner. Our numerical and trace-driven results show that relaying can reduce the offloading failure rate by more than 30% in D2D systems.

The main contributions of this paper are as follows:

1. We propose a new content download pattern that uses relays in addition to traditional seeds to improve the scalability of D2D offloading systems.
2. We introduce a storage optimization framework that can be applied to different types of helpers in D2D offloading systems.
3. Assuming constant helper density and Poisson contact processes, we show that offloading systems with n helpers can support $\Theta(\sqrt{n})$ pieces of content when using both relays and seeds, while the amount of content supported by a static caching system is only $O(1)$.
4. We show that the joint storage assignment problem for relays and seeds is convex and can be efficiently solved. Furthermore, our model can be extended to other types of helpers with concave offloading efficiency functions. This leads to a better understanding of storage allocation when the network has multiple types of helpers.

The rest of this paper is organized as follows: Section 2 reviews related works in D2D content dissemination and Section 3 introduces the system model used in this paper. Section 4 analyzes the D2D offloading efficiency for relays and shows that on-demand relays can asymptotically increase the amount of content supported by the system. The storage assignment problem is described in Section 5 and the benefits of our scheme are shown by numerical results and trace-driven simulations in Section 6. Finally, Section 7 presents conclusions of this work.

2. Related work

Message delivery in opportunistic networks: Opportunistic networks utilize the intermittent connection between mobile devices to transfer data from the source to the destination [7]. For message delivery applications, the source first replicates the message to several relays using short-range links. Relays then deliver the message to the destination when they contact the destination [8,9]. Most research works in message delivery applications consider how to reduce the end-to-end delay under storage constraints in relays [10–12]. The content offloading systems studied in this paper are different to message delivery applications, since the initial request comes from the destination rather than the source.

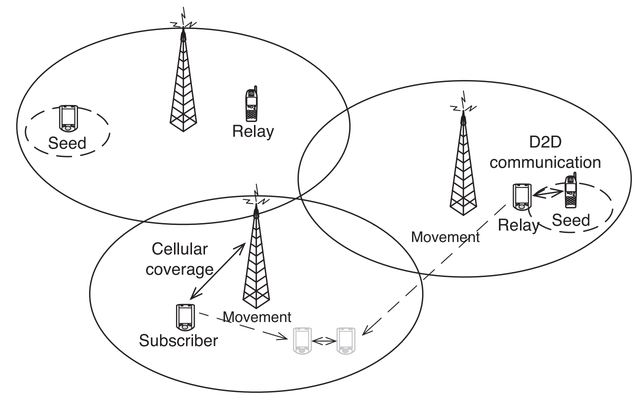


Fig. 1. Illustration of the communication model.

Information searching in opportunistic networks: Content offloading systems focus on reducing the data delivery cost for a given set of content [5,6,13–16]. As contents are cached in a distributed way, one important issue in content offloading systems is to search information in a network with dynamic topology and intermittent connections. User interests and network structures can be used to help the content searching process. SPOON utilizes interest based searching to improve the performance of P2P file sharing [17]. Social contact information is utilized in [18] to find the optimal caching point based on user interests. Seeker-Assisted Searching (SAS) use *seekers* to help node to search information, while assuming users with the same interest meets more frequently [19]. Optimal searching strategy in linear opportunistic networks is studied in [20].

Replication optimization in opportunistic networks: The goal of optimal content replication is to maximize the social welfare, which is defined as the expected utility gain from offloading. Delay-utility models are introduced in [5] to describe the impatience of subscribers. The utility optimization problem is shown to be NP-hard [5,6]. There are several ways to obtain approximate solutions, including greedy heuristic algorithms [6], convex relaxation [5] and a distributed algorithm based on voting [13]. Distributed caching in Femtocells is considered in [21]. The storage allocation problem under this scenario is similar to D2D networks with fixed contact patterns. This problem can be formulated as a set cover problem that is NP-hard [21]. Utility based replication algorithm is used in [22] to distribute mobile videos via human mobility between fixed venues. In [23], Chen et al. propose a new file replica optimization algorithm that considers both the node storage and contact frequency.

In comparison, we study tradeoffs between using storage for relaying and static caching under the D2D scenario. Our model combines static caching systems [5,6] and message relaying methods in DTN [24]. Unlike the traditional DTN routing problems, which only consider cache size in the asymptotical sense [8], we study cache utility with finite storage in the offloading system. In contrast to traditional DTN systems, where the source tries to find a best way to replicate the message, our system adopts a destination centric model, where the destination actively searches for the content.

3. Network model

3.1. System overview

The D2D offloading system studied in this paper combines two communication networks: a cellular network and an opportunistic network of D2D communication, as shown in Fig 1. The cellular network provides pervasive network coverage and a global communication channel for all mobiles. Due to the high cost of cellular communication, it is mainly used for control message exchange and content delivery traffic is offloaded through D2D communication. D2D communication leverages opportunistic short-range links formed between nearby mobiles to reduce content delivery cost. The links used for D2D communication can be WiFi Direct, Bluetooth, LTE/LTE-A, *etc*. Compared to directly communicating with the base station, these short-range links are usually faster, consumes less energy and free of mobile data charges. For example, LTE only provides 300 Mbps shared transmission rate, while the 802.11ac protocol used by WiFi provides transmission rate higher than 500 Mbps [25]. Real-world measurements also show that WiFi is 14.5 times and 23 times more energy efficient than 3G and LTE, respectively [26]. However, these D2D links have short communication ranges. For example, WiFi only has communication range up to 100 m. Therefore, short-range links only exist when mobiles move close to each other, and subscribers may become impatient before they can find a D2D transmission opportunity [5].

One important issue for D2D systems is to encourage nodes to act as *helpers*. Helpers contribute their storage as content caches and allow other nodes to download content from them through D2D links. Therefore, users should have motivations to contribute storage and battery power of their devices. Helpers may come from two sources: mobile devices provided by users and special devices deployed by the network operator. The incentive for mobile helpers may come from discounts or other benefits provided by the network operator. Such mechanisms have been well studied in existing work, e.g., [27]. In D2D system, it is also possible to utilize social connections to provide incentives for helpers. For example, users may ask their friends to help them in content downloading. Note that incentive schemes developed for traditional D2D offloading systems can be directly applied to our system, as our system do not incur higher storage and energy cost to helpers than traditional offloading schemes.

Network operators are also willing to deploy helpers to offload the traffic of cellular systems so that they do not need to continuously deploy more base stations to handle the increasing traffic. These helpers can be mobile nodes mounted on buses [28] or fixed caching points on Femtocells [21]. As D2D helpers do not need high-throughput backhaul, deployment of D2D helpers is much easier than deploying small base stations, *e.g.*, microcells or picocells. The advantage of these specialized helpers is that they can provide heterogeneous mobility patterns, and it is easier for network operators to optimize the content distribution process on specialized devices.

Different types of helpers can coexist in D2D offloading systems. Helpers with different mobility patterns, storage sizes or delivery strategies may lead to different performance tradeoffs. In this paper, we mainly focus on two types of helpers: Firstly, *seeds* are stable content caches that serve for all requests to a given content. The system allocates a certain number of seeds for each piece of content to ensure the availability of content. Secondly, *relays* are temporary storage allocated for a specific request. They are selected according to the information of the given request to enhance storage utility. The content cached in relays may dynamically change due to arrival or completion of relaying requests. Note that a single node may act as different roles at the same time: it may have part of its storage used as seeds and part of its storage used as relays.

Relaying procedure:

The content downloading process for a system with both seeds and relays is described as follows:

1. A subscriber sends a content request to the cellular operator. Either the operator or the subscriber can select some relays among all available relays to help the subscriber downloading the content. Detailed relay selection procedure is described in Section 4.2. Relays are notified through the cellular network when they are selected. Each relay then reserves relay storage segments for this downloading session and starts searching for the given content. As the notification is sent through the cellular network, the delay of the notification process can be ignored compared to the delay of D2D offloading.
2. Relays try to replicate the requested content in the reserved relay storage when contact a seed. The content is replicated to the relay via short range D2D links.
3. When the subscriber contacts a seed or a relay that has already replicated the content, the subscriber can download the content through D2D links. We call this as an *offloading success*.
4. If the subscriber becomes impatient before reaching any seed or relay with the requested content, he/she will directly download the content through the cellular network. We call this as an *offloading failure*.
5. Relays receive notifications when the offloading process succeeds or fails. The storage reserved for the downloading session is then released.

The advantage of the above on-demand relaying scheme is that it only uses the cellular network to deliver a small amount of control information, while the content replications are delivered by D2D links between seeds and relays, so the overall relaying cost is negligible. Mobile helpers acting as relays also do not consume significantly more energy or bandwidth than those acting as seeds, because the only extra cost for relays is to receive the control message. Note that our relaying scheme is different from relaying schemes in traditional DTN networks [8,10–12], which mainly focus on point-to-point message delivery rather than content distribution.

3.2. System model

We assume that content in the system is divided into segments with lengths of M. Such a segmentation mechanism is commonly used in streaming and content delivery systems [29]. Typical values for segment lengths are 1 to 2 M Bytes. Segments of the same file are treated as if they are independent pieces of content, so we use the term “content” instead of segments in the rest of this paper. Therefore, it is possible that the user downloads some of the segments of a given file using the offloading system, while other segments of the same file are directly downloaded from the cellular network.

We assume that there are N pieces of content in the system and that each content i has a mean request rate of R_i. We further assume that the contact duration between nodes is long enough so that at least one segment can be downloaded during the contact period. As short-range links normally have high transmission rate, *e.g.*, WiFi has transmission rates from 54–500 Mbps and Bluetooth has rate of 24 Mbps, it takes less than one second to transmit a single segment. Thus, the duration for content transmission is negligible compared to the durations of the contacts, which have average length of 370 s [30]. Systems with limited contact durations can be further optimized using schemes described in [31].

Contacts between nodes are assumed to be independent non-lattice renewal processes. Our model covers most of the existing mobility models, which assume that the contact patterns are Poisson processes or have power-law inter-contact times [5,32]. In the later part of this paper, we may further limit the contact processes to be only Poisson processes, which have been verified by real-world traces [33] as well as theoretical analysis [32]. To verify our model in real systems, we also use real-world traces in our experiments, see Section 6.2. In real-world, contact processes between different types of nodes can have different inter-contact time distributions. For example, the contact process between relays and subscribers may be different from that between seeds and subscribers.

We assume that there are n helpers willing to contribute their storage. We assume that the storage contributed by each helper is able to hold I pieces of content. This assumption can be relaxed and will be discussed in Section 5. The storage in helpers can be used as either seeds or relays. We use $n_{s,\,i}$ and $n_{r,\,i}$ to represent the number of seeds and relays used by offloading sessions for content i.

We define the offloading failure probability as the probability that the subscriber *cannot* discover the content through D2D communication within a given time period of T. To evaluate the efficiency of a single helper, we define the offloading efficiency E_s of a single seed as follows:

$$E_s = -\ln P_s, \tag{1}$$

where P_s is the probability that the subscriber *cannot* receive the content from that particular seed. A higher offloading efficiency means that the seed is more efficient in transmitting the content through the D2D link. Similarly, we define the efficiency of a relay E_r as $-\ln P_r$, where P_r is the probability that the subscriber cannot receive the content from the given relay. Table 1 summarizes the symbols used in this paper.

As the offloading failure probability depends on the size of the cache allocated for the given content, it is a function of the number of relays and seeds for content i. We denote the offloading failure probability for content i as $F(n_{s,\,i}, n_{r,\,i})$. Therefore, the total amount of traffic offloaded by D2D communication can be written as:

$$D = \sum_{i=1}^{N} MR_i(1 - F(n_{s,i}, n_{r,i})). \tag{2}$$

For example, suppose that the contact process between a seed and a subscriber is a Poisson process with a rate of λ_s. Then, the probability that the seed cannot meet the subscriber during the time period of T is given by $e^{-\lambda_s T}$. Consequently, the offloading efficiency of a single seed is $E_s = \lambda_s T$.

Offloading fails when all the $n_{s,\,i}$ seeds and $n_{r,\,i}$ relays are unable to deliver the content to the subscriber. When contacts between different pairs of nodes are independent, the event that the given seed or relay can successfully deliver the content is also independent of other relays/seeds. Therefore, we can multiply the failure probability of a single seed/relay to get:

$$F(n_{s,i}, n_{r,i}) = e^{-n_{s,i}E_s} e^{-n_{r,i}E_r} = P_s^{n_{s,i}} P_r^{n_{r,i}}. \tag{3}$$

Eq. (3) allows us to directly compare the efficiency of relays and seeds. When there are more than two types of helpers, the offloading failure probability can be written in a similar way. For details, please refer to Section 5.3.

Existing studies that use the static caching model are special cases of our system model in Eq. (3). When there are only seeds, we have $F(n_{s,i}, 0) = P_s^{n_{s,i}}$. For a Poisson process with a rate of λ_s, we have $F(n_{s,i}, 0) = e^{-n_{s,i}\lambda_s T}$, which is the same as the results of static caching systems [5,6].

4. On-demand relaying in D2D systems

4.1. Offloading efficiency for helpers

We study the offloading efficiency of our relaying scheme in this section. In the relaying scheme, the identity of the content that the subscriber is searching for is delivered to relays through the cellular network. Therefore, only two contacts are needed in our relaying scheme. In contrast, traditional relaying scheme requires three contacts with an extra contact between the subscriber and the relay to deliver the identity of the content. Both the seed and relays can deliver contents to the subscriber via short range links. The efficiency of the seed only depends on the contact process between subscribers and seeds. However, the efficiency of the relay is dependent on the number of seeds, in addition to the contact process between subscribers and relays. This is because relays need to first download the content from seeds before they can transmit it to the subscribers.

We first consider the offloading efficiency of a single relay node. When the contact pattern and number of seeds are known, the efficiency of the relays can be calculated as follows:

Theorem 1. *Suppose there are $n_{s,\,i}$ seeds for content i, and the inter-contact time between seeds and relays follows a Cumulative Distribution Function (CDF) of $Y(t)$ with a mean of μ_y. If the inter-contact time between relay and subscriber follows CDF of $X(t)$ with a mean of μ_x, then the offloading efficiency of the relay node for content i is given by:*

$$E_r = -\ln\left(\hat{X}(T) - \int_0^T \left(\hat{Y}(T-t)\right)^{n_{s,i}} d\hat{X}(t)\right), \tag{4}$$

where:

$$\hat{X}(t) = 1 - \int_0^t (1 - X(\tau))d\tau/\mu_x \tag{5}$$

$$\hat{Y}(t) = 1 - \int_0^t (1 - Y(\tau))d\tau/\mu_y. \tag{6}$$

Proof. There are two cases that the given relay will fail to deliver the content.

(i). The relay never meets the subscriber within time T. For non-lattice renewal processes, the CDF of the time between a randomly picked time point t and the next contact is given by $\int_0^t (1 - X(\tau))d\tau/\mu_x$ [34]. Therefore, the probability that the relay never meets the subscriber within time T is given by

$$\hat{X}(T) = 1 - \int_0^T (1 - X(\tau))d\tau/\mu_x. \tag{7}$$

(ii). The relay meets the subscriber within time T, but it never meets any seed before the last contact with the subscriber. Suppose that last contact between the relay and the subscriber happens in $(T-t,\ T-t+dt)$, with $0 \le t \le T$. The CDF of the time duration from last contact to the point of request expiration (at time T) is the age distribution of the renewal process. Therefore, the probability that the last contact happens in $(T-t,\ T-t+dt)$ (with age of t at the point of request expire) can be given by $-d\hat{X}(t)$. Because the probability that the relay does not meet any of the $n_{s,\,i}$ seeds before $T-t$ is given by $\left(\hat{Y}(T-t)\right)^{n_{s,i}}$, the probability for the second failure case is given by:

$$-\int_0^T \left(\hat{Y}(T-t)\right)^{n_{s,i}} d\hat{X}(t). \tag{8}$$

Taking the sum of the failure probability of case (i) and (ii) and translating it into offloading efficiency, we get the result of Theorem 1. □

Similarly, the offloading efficiency of seeds can be written as:

$$E_s = -\ln P_s = -\ln \hat{Y}(T). \tag{9}$$

From Theorem 1, we can see that the ratio of E_r/E_s is bounded above by the number of seeds:

Corollary 1. *The efficiency ratio of E_r/E_s is bounded above by $n_{s,\,i}$.*

Proof. Because both $\hat{X}(t)$ and $\hat{Y}(t)$ are Complementary Cumulative Distribution Function (CCDF) for age distribution of a renewal

Table 1
Notations used in this paper.

Symbol	Description
N	The number of contents
M	The segment length for contents
n	The total number of helpers
L	The side length for the network region
T	The time before the subscriber becomes impatient
I	The storage size for helpers
R_i	Request rate for content i
$n_{s,\,i}$	The number of seeds caching content i
$n_{r,\,i}$	The number of relays for content i
E_s, E_r	Offloading efficiency for one seed or one relay
P_s, P_r	The probability that subscriber *cannot* receive content from a seed or relay
λ_s, λ_r	The subscriber contact rate for seeds and relays
$X(t)$	CDF of the inter-contact time between relays and subscribers
$Y(t)$	CDF of the inter-contact time between seeds and relays

process, which are decreasing functions within range of [0, 1], we have:

$$\begin{aligned}
&\hat{X}(T) - \int_0^T \left(\hat{Y}(T-t)\right)^{n_{s,i}} d\hat{X}(t) \\
&\geq -\int_0^T \left(\hat{Y}(T-t)\right)^{n_{s,i}} d\hat{X}(t) \\
&= \mathbb{E}\left\{\left(\hat{Y}(T-t)\right)^{n_{s,i}} | 0 \leq t \leq T\right\} \\
&\geq \min\left\{\left(\hat{Y}(T-t)\right)^{n_{s,i}} | 0 \leq t \leq T\right\} \\
&= \left(\hat{Y}(T)\right)^{n_{s,i}}.
\end{aligned} \tag{10}$$

Therefore,

$$\begin{aligned}
E_r &= -\ln\left(\hat{X}(T) - \int_0^T \left(\hat{Y}(T-t)\right)^{n_{s,i}} d\hat{X}(t)\right) \\
&\leq -n_{s,i} \ln \hat{Y}(T).
\end{aligned} \tag{11}$$

As $E_s = -\ln \hat{Y}(T)$, we have $E_r \leq n_{s,\,i} E_s$. □

Example: poisson contact process

Consider the case where both contact processes are Poisson process. We have $X(t) = 1 - e^{-\lambda_r t}$ and $\mu_x = 1/\lambda_r$, where λ_r is the rate for the contact process between a relay and the subscriber. Similarly, we have $Y(t) = 1 - e^{-\lambda_s t}$ and $\mu_y = 1/\lambda_s$. Using Eqs. (5) and (6), we have $\hat{X}(t) = e^{-\lambda_r t}$ and $\hat{Y}(t) = e^{-\lambda_s t}$. We can obtain:

$$E_r = \begin{cases} -\ln\left(\dfrac{n_{s,i}\lambda_s e^{-\lambda_r T} - \lambda_r e^{-n_{s,i}\lambda_s T}}{n_{s,i}\lambda_s - \lambda_r}\right) & \lambda_r \neq n_{s,i}\lambda_s \\ \lambda_r T - \ln(1 + \lambda_r T) & \lambda_r = n_{s,i}\lambda_s \end{cases}. \tag{12}$$

Note that the sum of two exponentially distributed inter-contact times actually follows a hypoexponential distribution as expected.

Eq. (12) shows some characteristics of the offloading efficiency for relay nodes.

(i) When $\lambda_r \gg n_{s,\,i}\lambda_s$, E_r approaches $n_{s,\,i}\lambda_s T$. Compared to the offloading efficiency of the seed $E_s = \lambda_s T$, we see that a *single* relay has the same efficiency as $n_{s,\,i}$ seeds, where the upper bound in Corollary 1 is achieved. Therefore, for rare content with low $n_{s,\,i}$, using $n_{r,\,i}$ frequently contacting relays to download the content can have effects similar to multiplying $n_{s,\,i}$ by $n_{r,\,i}$.

(ii) When a relay has the same contact pattern to the subscriber as seeds, i.e., $\lambda_r = \lambda_s = \lambda$, we have a relay offloading efficiency of:

$$E_r = \lambda T + \ln\left(\frac{n_{s,i} - 1}{n_{s,i} - e^{-(n_{s,i}-1)\lambda T}}\right). \tag{13}$$

Note that we have $n_{s,i} - 1 \leq n_{s,i} - e^{-(n_{s,i}-1)\lambda T}$ when $n_{s,\,i} \geq 1$. Thus the second term in Eq. (13) is always negative and E_r can never be larger than λT, which means a relay is always less effective than a seed when $\lambda_s = \lambda_r$. This is reasonable because a relay must download the content from the seed before it can deliver it to the subscriber.

In summary, the offloading efficiency of relays can reach the upper bound of $n_{s,\,i}$ times the number of seeds when relays have high contact rates to subscribers. However, relays are less efficient than seeds when they have same contact rates as seeds.

4.2. Relay selection

According to the above analysis, relaying through a node with the same contact pattern as seeds is not helpful. However, D2D contact patterns are inherently non-uniform. The contact processes between different groups of mobile nodes have different contact patterns. As seeds serve for all the subscribers in the network, the contact process between a seed and a subscriber should follow the typical contact pattern between random pairs of nodes. However, relays may have different contact patterns to a given subscriber. We can intentionally select relays for a given offloading request based on previous knowledge of their contact patterns. For example, a device belonging to a friend of the user or a device mounted at a subway station that the user always passes by, can be selected as the relay in order to improve the efficiency of content distribution. As discussed in Section 4.1, the offloading efficiency mainly depends on the contact rates of relays. Therefore, we propose to select relays based on their contact rates to reduce the relay selection complexity.

The relay selection process can be carried out either by the network operator or by the subscriber. When the relay is selected by the network operator, the network operator needs to record the top-k nodes with the highest contact rates for each subscriber. Such records can also be generated by the user devices and submitted to the operator along with the offloading request. These nodes will have higher contact rate to the subscriber compared to randomly selected nodes.

To verify the existence of nodes with higher contact rates, we studied the MIT reality trace [35], which contains Bluetooth traces from 100 mobiles for more than 9 months. Fig. 2 shows the CDF of the contact rates for the top-k frequently contacted neighbors for each mobile, normalized by the average contact rate. We see that more than 90% of the subscribers have at least 5 neighbors with a contact rate 10 times higher than average. More than 80% subscribers have 10 neighbors with contact rates 10 times higher than average. Similar distributions of frequently contacted neighbors are also observed in recently collected traces as in [36].

Relays can also be selected by the subscribers. A subscriber can request friends or colleagues to help with downloading. As

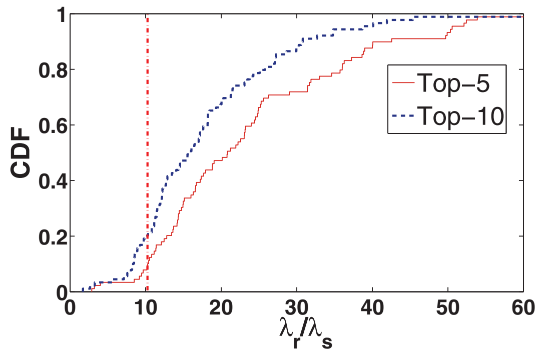


Fig. 2. CDF of λ_r/λ_s for top frequent contacting friends in MIT reality trace.

subscribers are familiar with their friends, they will know the exact contact patterns of these relays. Such relays may exhibit different contact patterns compared to relays selected by the network operator. For example, the subscriber may contact colleagues regularly on workdays [35]. Suppose that inter-contact time between relay and subscriber is bounded above by T_0, with $T_0 < T$. By the reversibility of the contact process [34], the efficiency of a single relay is at least as good as $n_{s,\,i}$ seeds that duplicated the content at time T_0 after the subscriber requested the content. By Theorem 1, we have the efficiency E_r bounded below by $-n_{s,i}\ln(\hat{Y}(T-T_0))$. Considering the case that the contact process between the seed and relay is a Poisson process, E_r is bounded below by $n_{s,i}(T-T_0)\lambda_s$. In this case, a single relay has the same efficiency as $n_{s,\,i}/2$ seeds when $T_0 = T/2$.

The above analysis shows the potential benefits of using nodes with regular contact patterns as relays. For example, nodes deployed at subway stations have high contact rates to many users. If we use these nodes as seeds to cache popular content, their efficiency is equivalent to λ_r/λ_s regular seeds. However, if we use them as relays, their efficiency is multiplied by $n_{s,\,i}/2$ times, which can be higher than λ_r/λ_s. Therefore, it is preferable to use these nodes as relays instead of as seeds.

Note that our relay selection process only considers the contact rate of a given relay. Relays can improve the offloading efficiency as long as they have higher contact rates than a randomly selected node. In the case that two relays always have the same mobility pattern, e.g., two mobile phones belonging to the same person, the offloading efficiency will be worse than relays have independent mobility patterns. How to select relays under correlated mobility patterns is leaved as future works.

4.3. Asymptotic analysis for Poisson process

Relays with higher contact rates increase the amount of content that the system can support when the network size increases. Consider a network deployed on a square region with side length of L. Suppose that the density of helpers in the network is ρ. The number of helpers n then increases as ρL^2 when the network size L increases. As the network size increases, the contact rate between a particular pair of nodes decreases at a speed of $\Theta(L^{-2})$ for various types of mobility models [32].

We first show that the total amount of content that the offloading system can support is $O(1)$ under static caching systems. In static caching systems with Poisson contact process, we have $F(n_{s,i}, 0) = e^{-n_{s,i}\lambda_s T}$, where $\lambda_s = \Theta(L^{-2})$. Suppose that $n_{s,i} = o(L^2)$, i.e., $0 \le n_{s,\,i} < c_1 L^2$, for*arbitrary* small constant c_1 when $L \to \infty$. We can make $F(n_{s,\,i}, 0)$ arbitrarily close to 1 when L approaches infinity by taking the constant c_1 small enough. Therefore, to achieve an acceptable offloading success probability, $n_{s,\,i}$ at least should be $\Omega(L^2)$, i.e., $n_{s,\,i} \ge c_2 L^2$ for *some* constant c_2 when $L \to \infty$. The total number of available storage segments is $\rho L^2 I$ and we have $\sum_{i=1}^{N} n_{s,i} \le \rho L^2 I$. Therefore, the total amount of content that the system can support is $N = O(1)$.

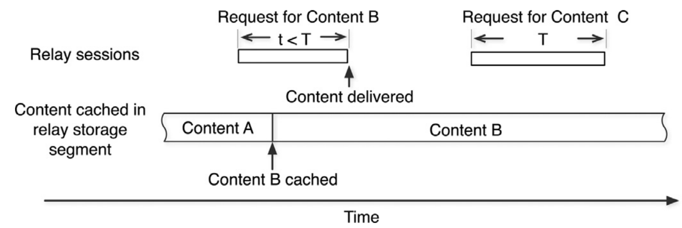


Fig. 3. Content replacement process for a single relay storage segment.

Now consider system with relays. As friends contact regularly irrespective of the network size, it is reasonable to assume that the contact rate between relays and subscribers, λ_r, remains constant when the network size increases. As $\lambda_s = \Theta(L^{-2})$, we have $\lambda_s \ge c_\lambda L^{-2}$ for some constant c_λ.

To ensure that $F(n_{s,\,i}, n_{r,\,i}) \le F_0$ for some constant $F_0 \le 1$, we can choose $n_{s,i} = n_{r,i} = L\sqrt{-\ln F_0/(c_\lambda T)}$. For sufficiently large L, we have $\lambda_r \gg n_{s,\,i}\lambda_s$ due to $n_{s,\,i}\lambda_s$ is $O(L^{-1})$ in this case. Using the analysis in Section 4.1, we have $E_r \ge (n_{s,i} - \varepsilon)\lambda_s T$ for arbitrarily small ε. Therefore,

$$\begin{aligned} F(n_{s,i}, n_{r,i}) &\le e^{-(n_{s,i}+n_{r,i}n_{s,i}-n_{r,i}\varepsilon)\lambda_s T} \\ &\le e^{-(L\sqrt{-\ln F_0/(c_\lambda T)})^2 c_\lambda L^{-2} T} \\ &= F_0. \end{aligned} \tag{14}$$

Thus, we can ensure that there are at least N pieces of content with bounded failure probability, while $\sum_{i=1}^{N}(n_{s,i} + n_{r,i}) \le \rho L^2 I$. It is easy to see that $N = \Theta(L)$ in this case. Consequently, the amount of content increase as $N = \Theta(\sqrt{n})$ when the network size L increases.

Note that the relaying scheme needs $\Theta(\sqrt{n})$ relays, which requires more friends with constant contact rates when the network size increases. In cases where subscribers only have constant number of friends, multi-hop relaying can be used. In multi-hop relaying, friends of friends are requested to act as relays. It can be shown that offloading efficiency of $(n_{s,i} - \varepsilon)\lambda_s T$ can be achieved by using relays that are h hop away from the subscriber when each hop satisfies $\lambda_r \gg n_{s,\,i}\lambda_s$ and h is $O(\log n)$. Therefore, we can still find $\Theta(\sqrt{n})$ relays via multi-hop relaying in the case where each subscriber/relay only has a constant number of friends.

4.4. Reuse of relays

Relays can be reused for offloading different pieces of content. This is especially useful for offloading rare contents. Define a relay session as the period which starts at the time when the request reaches the relay, and ends when either the content is delivered to the subscriber or the request expires. As shown in Fig. 3, content stored in a single relay segment changes over time while the storage segment always retain some content even after the session is finished. Moreover, relays who do not contact a seed will do nothing during the relay session, as shown in the relay session for content C in Fig. 3. Among the $\Theta(\sqrt{n})$ relays reserving resources for the session, only a small fraction $F(n_{s,\,i}, 0)$ of them actually download the content.

As described in Section 3.1, the relay storage segment is committed to caching the requested content during the relay session. Since the number of relay segments used for a session for content i is $n_{r,\,i}$, which can be obtained through methods described in 5.1, the average number of relay segments committed to relaying content i can be derived by Little's Theorem:

$$\widetilde{n}_{r,i} = R_i n_{r,i} \widetilde{T}_i, \tag{15}$$

where R_i is the request rate of content i, $n_{r,\,i}$ is the number of relays for each request and $\widetilde{T_i}$ is the average duration of a relay session. The fraction of offloading session with length longer than t can be written as $F(n_{s,\,i}, n_{r,\,i}, t)$ by substituting the time limit T with a variable t in the offloading failure probability. Therefore, we have $\widetilde{T_i} = \int_0^T F(n_{s,i}, n_{r,i}, t)dt$ as subscribers waiting at most for time T.

From Eq. (15), we see that $\widetilde{n}_{r,i}$ can be smaller than $n_{r,\,i}$ when $R_i\widetilde{T_i} < 1$. This shows rare content do not actually use $n_{r,\,i}$ storage segments on average. Thus, it would be more efficient to allocate more relays for rare content, since a relay storage segment can dynamically switch between different content to get better storage utilization. For "hot" contents that are always simultaneously requested by multiple subscribers, i.e., $R_i\widetilde{T_i} > 1$, a better strategy is to statically cache the content in seeds, since spreading the "hot" content among different relays is less efficient when λ_r is comparable to $n_{s,\,i}\lambda_s$ as shown in Section 4.1.

Relays stay idle between relay sessions. The average length of idle period can be derived using the G/G/m/m queueing model. The utilization factor for relay segments depends on both the request arrival process and the node contact patterns, which do not have a general closed form formula. To simplify our model, we use the relay reuse factor $k_{u,i} = R_iT$ to approximate the utility of relays for content i in Section 5.

4.5. Dynamical loads

There are two cases where the requesting rate of R_i can change. The first case is a long-term evolution of content popularity, where R_i changes slowly. The second case is short-term load surges, where R_i can temporally exceed its average rate.

To handle long-term load changes, the cellular network keeps track of the requests of each content. As all offloading requests are sent through the cellular network, we can estimate the long-term request rates for a given content i as follows. Suppose that there are $r_i(t)$ requests for content i in a given time slot t, we use Exponential Moving Average (EMA) to estimate the long-term request rate as:

$$R_i(t) = \alpha R_i(t-1) + (1-\alpha) r_i(t) \tag{16}$$

where $R_i(t)$ is the long-term request rate at time t and α is a constant smoothing factor with value between 0.8 and 0.95. The size of the time slot can be chosen as one hour and the smoothing factor α determines the smoothness of the estimation. After estimated the requesting rate, we execute the optimization algorithm described in Section 5 to calculate the optimal values for $n_{s,\,i}$ and $n_{r,\,i}$.

To handle short-term surges in content request, we apply a rate limiting algorithm for relay requests. Note that the number of seeds, $n_{s,\,i}$, is independent of short-term requesting rate changes. We only need to modify the number of relays $n_{r,\,i}$ for the given request. In case that the instantaneous request rate $r_i(t)$ is larger than long-term average $R_i(t)$, we set the number of relays allocated for a given request to $\frac{\beta R_i(t)}{r_i(t)} n_{r,i}$ so that the total number of relays allocated for content i will not exceed $\beta R_i(t) n_{r,\,i}$, where β is an overload factor which allows a single content to use more caches than the long-term optimal value. In case of load surge, the number of relays allocated for the content will be limited and the offloading efficiency for the given content will decrease temporally.

5. Storage assignment optimization

5.1. Problem formulation

The analysis in Section 4 shows that the efficiency of relays is determined by various factors, including the number of seeds, relay contact patterns and content request rates. Therefore, we should carefully set the number of relays used for each piece of content based on these parameters. Seeds need to statically cache a piece of content to ensure that there are minimal copies of the given content in the network. This leaves limited storage for relays. Thus, we need to dynamically adjust the number of relays allocated for each piece of content, depending on its popularity and the number of seeds currently caching the content.

In this section, we jointly optimize the storage assigned to seeds and relays so that the overall offloading failure probability is minimized for content with different request rates.

The storage assignment problem for a system with both relays and seeds can be formulated as follows:

$$Minimize \ \sum_{i=1}^{N} MR_i F(nf_{s,i}, nf_{r,i}) \tag{17}$$

$$s.t. \ \sum_{i=1}^{N} (f_{s,i} + k_{u,i} f_{r,i}) \le I \tag{18}$$

$$f_{s,i} + f_{r,i} \le 1 \ \forall i \tag{19}$$

$$f_{s,i} \ge 0, f_{r,i} \ge 0 \ \forall i \tag{20}$$

where $f_{s,i} = \frac{n_{s,i}}{n}$ and $f_{r,i} = \frac{n_{r,i}}{n}$ are the fractions of nodes that serve as seeds and relays for content i. Eq. (18) limits the sum of storage segments allocated for seeds and relays to be smaller than nI, which is the total number of storage segments in the system. The fraction of relay segments is multiplied by $k_{u,\,i}$, which reflects the impact of relay reuse as described in Section 4.4. Eq. (19) requires that the number of seeds plus the number of the relays should not be larger than n for any content.

In the above formulation, we relax the storage optimization problem from an integer programming problem, which is NP-hard [5], to a convex optimization problem that can be efficiently solved. This is because we use real variables $f_{s,\,i}$ and $f_{r,\,i}$ instead of integer variables $n_{s,\,i}$ and $n_{r,\,i}$. We can show that the approximation ratio of the above relaxation approaches 1 when n goes to infinity, using similar arguments as in [5].

Theorem 2. *The objective function Eq. (17) in the storage assignment problem is convex, when $E_r(f_{s,\,i})$ is concave and twice differentiable.*

Proof. The convexity of the objective function in Eq. (17) can be derived as follows:

As the nonnegative weighted sum of convex functions is convex, it is enough to show that $F(nf_{s,\,i}, nf_{r,\,i})$ is convex respect to $f_{s,\,i}$ and $f_{r,\,i}$. We have:

$$F(nf_{s,i}, nf_{r,i}) = e^{-nf_{s,i}E_s - nf_{r,i}E_r}. \tag{21}$$

Define $g(f_{s,\,i}, f_{r,\,i})$ as $-nf_{s,i}E_s - nf_{r,i}E_r$. Since $F(nf_{s,i}, nf_{r,i}) = e^{g(f_{s,i}, f_{r,i})}$, if $g(f_{s,\,i}, f_{r,\,i})$ is convex then $F(nf_{s,\,i}, nf_{r,\,i})$ is also convex [37].

As we have shown in Section 4.1, E_r is normally a function of $f_{s,\,i}$ and E_s is a constant with respect to $f_{s,\,i}$ and $f_{r,\,i}$. Therefore, when $g(f_{s,\,i}, f_{r,\,i})$ is twice differentiable, we have:

$$\nabla^2 g(f_{s,i}, f_{r,i}) = -nf_{r,i} \frac{\partial^2 E_r(f_{s,i})}{\partial f_{s,i}^2}. \tag{22}$$

Since $nf_{r,\,i}$ is always positive, $g(f_{s,\,i}, f_{r,\,i})$ is convex when $E_r(f_{s,\,i})$ is concave. □

Corollary 2. *The storage assignment problem in Eqs. (17)–(20) is convex for relays with nonlattice renewal contact process and twice differentiable offloading efficiency function.*

Proof. To show $E_r(f_{s,i}) = -\ln P_r(f_{s,i})$ is concave, we have:

$$P_r(f_{s,i}) = \hat{X}(T) - \int_0^T \left(\hat{Y}(T-t)\right)^{nf_{s,i}} d\hat{X}(t). \tag{23}$$

Note that $\hat{X}(t)$ is a decreasing function, therefore we have $-d\hat{X}(t)/dt \geq 0$. Since exponential functions are log-convex, $\left(-d\hat{X}(t)/dt\right)\left(\hat{Y}(T-t)\right)^{nf_{s,i}}$ is a log-convex function of $f_{s,\,i}$ on every point $t \in [0, T]$. Taking integral over t preserves log-convexity [37], so $-\int_0^T \left(\hat{Y}(T-t)\right)^{nf_{s,i}} d\hat{X}(t)$ is also log-convex. Adding a positive constant $\hat{X}(T)$ preserves log-convexity, thus $P_r(f_{s,\,i})$ is log-convex and $E_r(f_{s,\,i})$ is concave. This directly leads to the convexity of the objective function as discussed above. It is clear that all the constraints in the storage assignment optimization problem in Eqs. (18)–(20) are linear, so the storage assignment problem is convex. □

Corollary 2 shows the convexity of the storage assignment problem. Therefore, the optimal solution for this problem can be found by efficient algorithms such as gradient projection methods.

5.2. Distributed solution

The Lagrangian of the storage assignment optimization problem can be written as:

$$\begin{aligned} &L(f_{s,r}, f_{r,i}, \nu) \\ &= \sum_{i=1}^{N} MR_i F(nf_{s,i}, nf_{r,i}) + \sum_{i=1}^{N} \nu_i (f_{s,i} + f_{r,i} - 1) \\ &\quad + \nu_0 \left(\sum_{i=1}^{N} (f_{s,i} + k_{u,i} f_{r,i}) - I \right) \end{aligned} \tag{24}$$

where ν_i, $i \in \{0, 1, \ldots, N\}$ are Lagrangian multipliers for the constraints.

The optimal solutions $f^*_{s,i}$, $f^*_{r,i}$, ν^*_i must satisfy the KKT conditions [37], which require

$$f^*_{s,i} + f^*_{r,i} - 1 \leq 0, \quad i = 1, \ldots, N \tag{25}$$

$$\sum_{i=1}^{N} \left(f^*_{s,i} + k_{u,i} f^*_{r,i}\right) - I \leq 0 \tag{26}$$

$$\nu^*_i \geq 0, \quad i = 0, \ldots, N \tag{27}$$

$$\nu^*_i \left(f^*_{s,i} + f^*_{r,i} - 1\right) = 0, \quad i = 1, \ldots, N \tag{28}$$

$$\nu^*_0 \left(\sum_{i=1}^{N} \left(f^*_{s,i} + k_{u,i} f^*_{r,i}\right) - I \right) = 0 \tag{29}$$

$$MR_i \varphi_s(f^*_{s,i}, f^*_{r,i}) + \nu^*_i + \nu^*_0 = 0, \quad i = 1, \ldots, N \tag{30}$$

$$MR_i \varphi_r(f^*_{s,i}, f^*_{r,i}) + \nu^*_i + k_{u,i}\nu^*_0 = 0, \quad i = 1, \ldots, N \tag{31}$$

where $\varphi_s(f_{s,i}, f_{r,i}) = \frac{\partial F(nf_{s,i}, nf_{r,i})}{\partial f_{s,i}}$ and $\varphi_r(f_{s,i}, f_{r,i}) = \frac{\partial F(nf_{s,i}, nf_{r,i})}{\partial f_{r,i}}$. By condition Eq. (28), we see that either $\nu^*_i = 0$ or $f^*_{s,i} + f^*_{r,i} = 1$ must be satisfied. Therefore, if we do not allocate all nodes as seeds or relays for a specific content i (which means $f^*_{s,i} + f^*_{r,i}$ is smaller than 1), we should have $\nu^*_i = 0$. Thus except for extremely popular content with $f^*_{s,i} + f^*_{r,i} = 1$, KKT conditions require:

$$R_i \varphi_s(f^*_{s,i}, f^*_{r,i}) = -\nu^*_0 / M \tag{32}$$

$$R_i \varphi_r(f^*_{s,i}, f^*_{r,i}) / k_{u,i} = -\nu^*_0 / M. \tag{33}$$

Note that $-\nu^*_0/M$ is a constant for all content i. When ν^*_0 is given, we can separately solve Eqs. (32) and (33) for each content to get the optimal allocation. If the solution has $f^*_{s,i} + f^*_{r,i} \geq 1$, it is evident that content i is so popular that we can simply set $f^*_{s,i} + f^*_{r,i} = 1$ and $\nu^*_i > 0$ for content i. Therefore, one way to solve the above problems is to use ν_0 as a global pricing signal [38] to coordinate the distributed storage allocation $f_{s,\,i}$ and $f_{r,\,i}$ for each content i.

5.3. Extension of the optimization framework

In the above discussion, we mainly focus on systems with two types of helpers, namely, seeds and relays. The optimization framework in Section 5.1 can be extended to cases where there are more than two types of helpers.

Consider the case where there are WiFi APs or Femtocells acting as helpers in the D2D offloading system. These APs or Femtocells can provide low cost Internet access to nearby nodes. Thus, they can be treated as if they were seeds for all content. Suppose that there are n_A APs in the system. Then, we can change Eq. (3) to $F(n_{s,i}, n_{r,i}, n_A) = e^{-n_{s,i}E_s} e^{-n_{r,i}E_r} e^{-n_A E_A}$, where E_A is the offloading efficiency of APs which can be calculated through the contact patterns between nodes and APs. The offloading efficiency of E_r is a function of both $n_{s,\,i}$ and n_A in this scenario, because relays can also download from APs or Femtocells. However, the general structure of the problem does not change and the framework of Section 5.1 can also be applied. Using our optimization framework, tradeoffs between deploying APs and mobile helpers can be easily characterized.

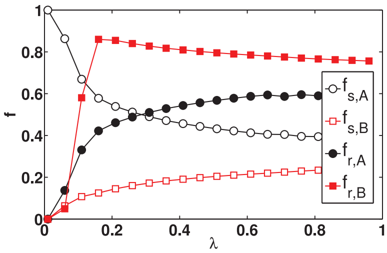
The systems may contain different types of helpers that have different contact patterns or different storage sizes. In these cases, each type of helpers will have its own fraction parameter $f_{h,\,i}$ for content i in the optimization problem and each type of helper may have a different offloading efficiency E_h. By Theorem 2, the general tradeoff point for storage assignment in different types of helpers can be solved when the offloading efficiency functions are concave.

5.4. Practical issues

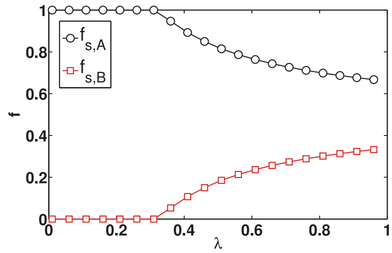
The above optimization framework provides a way to allocate the storage in an offloading system. Although centralized optimizations in cellular networks are possible, there are several practical issues to be considered in implementation.

Number of contents: First, the number of content pieces may be very large in real systems. Therefore, solving $f_{s,\,i}$ and $f_{r,\,i}$ for each content may incur high computational costs. To address this problem, we propose to group contents into a fixed number of content categories based on the popularity of the content. We observe that the solutions for different contents only depend on the value of R_i. Therefore, we can aggregate contents with similar request rates into one content category and optimize over content categories to obtain approximate solutions. In practice, content can be grouped into bins with geometric sequences of request rates, e.g., the first bin contains contents with request rate smaller than 0.1 requests per day and the rest bins covers content request rates of 0.1–0.2, 0.2–0.4, 0.4–0.8, …, respectively. Consider the case there are l categories of contents, each with m_i contents of request rate R_i. We can change the objective function Eq. (17) to $\sum_{i=1}^{l} Mm_i R_i F(nf_{s,i}, nf_{r,i})$, the constraint in Eq. (18) to $\sum_{i=1}^{l} m_i\left(f_{s,i} + k_{u,i} f_{r,i}\right) \leq I$ and keep all other constraints unchanged to reduce the problem size.

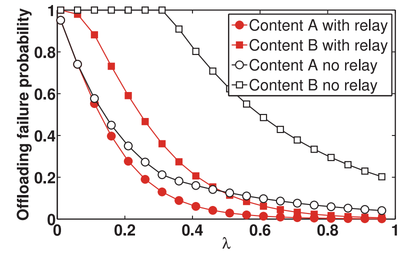
Limited number of friends: Second, when the optimal solution has large $f_{r,\,i}$ for some pieces of content, the subscriber may not have enough friends with high contact rates. To address this limitation, we add constraints, such as $nf_{r,\,i} \leq 5$, $\forall i$, to limit the number



(a) Storage allocation for relaying scheme



(b) Storage allocation for static scheme



(c) Failure probability for relay and static scenario

Fig. 4. Convex optimization result ($n = 5000$, $k_r = 10$, $T = 1$).

of relays for each session to be smaller than five, *i.e.*, less than five relays are used in each session. Our numerical results in Section 6 show that such limitations do not significantly change the system performance.

Communication overhead: Relaying requires content requests to be sent to relays through the cellular network, which incurs additional control traffic above that of traditional static solutions [6]. However, content delivery systems normally have large segment sizes, such as 2M bytes in P2P-VoD systems [29]. It only takes several hundred bytes to send relay requests to 10 relays, which incurs less than 1% overhead. Compared to the 30% gain in offloading efficiency, the 1% communication overhead is small.

Unreliable contacts: In real networks, detection of the existence of D2D links might be unreliable, and data transfer over D2D links could be dropped. Moreover, some contacts may be too short to transmit even a single segment. Considering these cases, we use the effective contact rate in our system. Suppose that among all contacts, there are a portion γ of contacts that are unreliable or short contacts. The effective contact processes for seeds and relays can then be modeled as Poisson processes with intensity of $(1-\gamma)\lambda_s$ and $(1-\gamma)\lambda_r$, respectively, by the properties of Poisson process [34]. Our results and optimization schemes are still valid in this case.

6. Simulation results

6.1. Numerical results

We use the numerical results obtained from gradient projection algorithms to show the properties of the storage assignment problem.

(1) Example for two categories of content

We use an offloading system with only two content categories A and B to demonstrate the tradeoffs between relays and seeds. The requesting rates for the two categories of content are 0.5 and 0.1 respectively, and the amount of content in each category is 1000. We assume that there are $n = 5000$ helpers, each of which contributes one storage segment. We assume the contact processes are Poisson processes with $k_r = 10$ (i.e., relays have 10 times higher contact rates than seeds.). Fig. 4(a) and (b) shows the fractions of storage allocated for each *type* of contents in the relaying scheme and static cache scheme, respectively. From Fig. 4(b), we see that static cache assignment tends to allocate all storage to the popular content category A when λ is small. This implies that a static cache scheme only tries to offload a small number of contents. In the relaying scheme, the system allocates a high fraction of relays to the less popular content category B to help deliver content B even in the case that λ is quite low. The offloading failure probabilities are shown in Fig 4(c). Offloading failure probabilities of the relaying schemes for both content categories are always smaller than those of the static cases. The failure probability for the less popular content category B has been greatly improved in the relaying systems. This shows that relaying can support more content with low requesting rates. The numerical results under various different parameter settings also has similar trends that demonstrate relaying can improve the offloading of rare content.

(2) Offloading performance

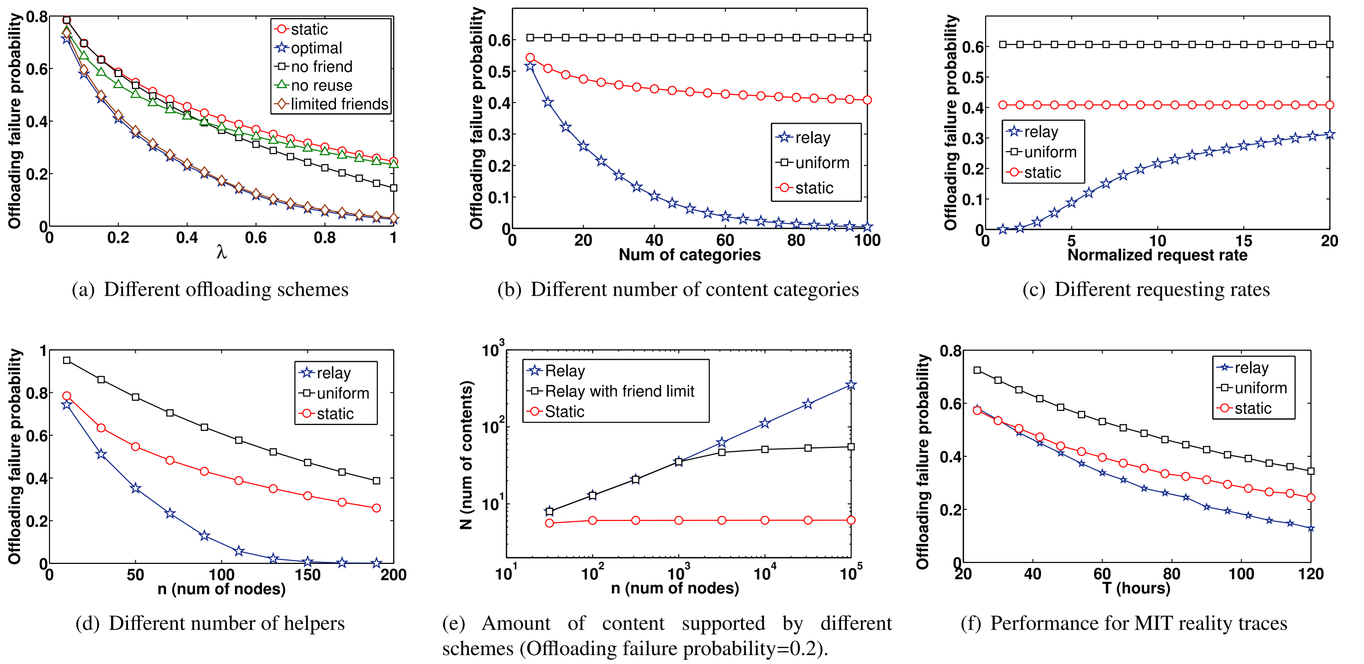
Fig. 5(a) illustrates systems with 100 content categories, where the content request rate follows a Zipf distribution with $\alpha = 0.8$ [39]. We observe that the overall failure probability in the *optimal* relaying case reduces much faster than in the static case. We further consider the impact of different factors on the offloading efficiency. For the curve labeled *no reuse*, we set $k_{u,i} = 1$ so that relays are never reused between contents. We observe that the performance of relaying is still better in this case than in the static case, because relays with higher contact rates to subscribers perform better than seeds. On the other hand, if relays do not have higher contact rates but can be reused, the offloading failure probability reduces faster when λ is higher, as shown in the *no friend* case. This is because relays with high reuse rates can be especially helpful for rare content, which has more chances to be offloaded when λ is high. We also observe that limiting the number of relays for each session has little impact on performance, because the curve for *limited friends*, where $nf_{r,\,i}$ is smaller than five, is almost identical to the optimal solution.

(3) Performance under different scenarios

Fig. 5(b) shows the relationship between offloading performance and the number of content categories in the system. In the experiments, we increase the number of content categories from 5 to 100. In the *uniform* case, we divide available storage slots in helpers evenly to these content categories without considering the request rates. Therefore, the offloading failure probability does not change when the number of content categories increases. The *static* scheme allocates storage based on content request rates, where popular content gets more replicas. The *relay* scheme uses part of the storage as seeds and allocates relays for content request based on our optimization algorithm. Both the *static* and *relay* scheme consider the request rates of contents, so that their offloading failure probabilities decrease as the number of content categories increases, *i.e.*, the request rates for content become different. The relay scheme has lower offloading failure probability than the static scheme.

Fig. 5(c) shows the performance of offloading schemes with different content request rates. Both the uniform and static scheme have constant offloading failure probability when the request rate increases, as they only use seeds for offloading. The relay scheme has better performance than the other two schemes. However, the gap reduces when the request rate increases. This is because relays reserve storage for each individual request. Thus, relays become congested when the request rates increase. However, the performance of relay scheme is always better than the static scheme, since the static solutions are in a subset of the relaying solutions.

Fig. 5(d) shows the offloading performance under different numbers of helpers. We observe that all schemes have better



(a) Different offloading schemes
(b) Different number of content categories
(c) Different requesting rates
(d) Different number of helpers
(e) Amount of content supported by different schemes (Offloading failure probability=0.2).
(f) Performance for MIT reality traces

Fig. 5. Offloading failure probability performances.

performance when the number of helpers increases. The offloading failure probability deceases in a much faster speed in the relay case than in the static and uniform case.

Fig. 5(e) considers the amount of content that an offloading system can support when the network size increases as discussed in Section 4.3. The number of helpers n increases from 100 to 100,000 and we use convex optimization to find the amount of content that the system can offload with a failure probability smaller than 20%. We observe that the static system can only offload a constant amount of content while in the relaying case, the amount of content increases with n. We also evaluate the case where subscribers have less than 10 friends. In this case, we observe that the relay system can support 10 times more content than the static caching system. However, the amount of content that the system can support converges after the number of helpers exceeds 5000. This implies that supporting larger amounts of content will likely require multi-hop relaying or common relays deployed by the network operator for all subscribers.

6.2. Trace-driven simulations

We use MIT reality traces [35] to verify the performance of our scheme in real networks. The trace is collected by 100 students and faculties who carries smart phones for more than 9 months to collect Bluetooth devices in proximity [35]. The Bluetooth trace are collected by 100 internal devices which have discovered approximately 20,000 external devices in total. Therefore, we only have a partial view of the entire contact process. The average contact rate for users is 31.44 per day, after removing empty days, *i.e.*, days with no contacts. Based on the estimation of average contact rate, we allocate storage to 100 Zipf distributed pieces of content using our optimization algorithm, while assuming that each node can store two pieces of contents. Only the 100 internal devices is selected as relays in the simulation, while all devices can be seeds.

Fig. 5(f) depicts the offloading failure probabilities under different schemes. We observe that both the static and relay scheme outperform the uniform scheme. The offloading failure probability of the relay scheme is similar to the static scheme when T is small. However, the failure probability is reduced by more than 30% when $T > 80$. Comparing Fig. 5(a)–(f), we observe that the trace-driven results have trends similar to those in the numerical results, while the trace-driven result is worse than that predicted by theory. This may be caused by the limited choices of relays in the trace-driven simulation.

7. Conclusion

In this paper, we proposed a novel relay-based storage assignment framework for D2D systems. Instead of using storage as static content cache, our scheme uses on-demand relaying to enhance the storage utilization. We demonstrated the usefulness of this relaying framework through both theoretical analysis and trace driven simulations. To further improve our system, one important future research direction is to study the incentives of the helpers so that more users will participate in the offloading system.

Acknowledgment

This work is partially supported by NSFC Grants (No. 61373129, No. 91218302, No. 61321491, No. 61472185), JiangSu NSF (No. BK20151390), and EU FP7 IRSES MobileCloud Project Grant (No. 612212).

References

[1] Cisco, Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, Cisco White Paper, 2014, pp. 2013–2018.
[2] 3GPP, Feasibility study for proximity services, SA1 TR 28.803 (2012).
[3] K. Doppler, M. Rinne, C. Wijting, C. Ribeiro, K. Hugl, Device-to-device communication as an underlay to LTE-advanced networks, Commun. Mag., IEEE 47 (12) (2009) 42–49.
[4] B. Han, P. Hui, V.A. Kumar, M.V. Marathe, J. Shao, A. Srinivasan, Mobile data offloading through opportunistic communications and social participation, IEEE Trans. Mobile Comput. 11 (5) (2012) 821–834.
[5] J. Reich, A. Chaintreau, The age of impatience: optimal replication schemes for opportunistic networks, in: Proceedings of ACM CoNEXT, 2009.
[6] Y. Li, G. Su, P. Hui, D. Jin, L. Su, L. Zeng, Multiple mobile data offloading through delay tolerant networks, in: Proceedings of ACM CHANTS, 2011.
[7] M. Conti, M. Kumar, Opportunities in opportunistic computing, Computer 43 (1) (2010) 42–50.
[8] A. Balasubramanian, B. Levine, A. Venkataramani, DTN routing as a resource allocation problem, in: Proceedings of ACM SIGCOMM, 2007.

[9] T. Spyropoulos, K. Psounis, C.S. Raghavendra, Spray and wait: an efficient routing scheme for intermittently connected mobile networks, in: Proceedings of ACM SIGCOMM workshop on Delay-tolerant networking, 2005, pp. 252–259.
[10] J. Liu, X. Jiang, H. Nishiyama, N. Kato, Delay and capacity in ad hoc mobile networks with f-cast relay algorithms, IEEE Trans. Wirel. Commun. 10 (8) (2011) 2738–2751.
[11] C. Boldrini, M. Conti, A. Passarella, Performance modelling of opportunistic forwarding under heterogenous mobility, Comput. Commun. 48 (2014) 56–70.
[12] P. Sermpezis, T. Spyropoulos, Understanding the effects of social selfishness on the performance of heterogeneous opportunistic networks, Comput. Commun. 48 (2014) 71–83.
[13] S. Ioannidis, L. Massoulie, A. Chaintreau, Distributed caching over heterogeneous mobile networks, in: Proceedings of ACM SIGMETRICS, 2010.
[14] S. Ioannidis, A. Chaintreau, L. Massoulié, Optimal and scalable distribution of content updates over a mobile social network, in: Proceedings of IEEE INFOCOM, 2009, pp. 1422–1430.
[15] W. Gao, G. Cao, A. Iyengar, M. Srivatsa, Supporting cooperative caching in disruption tolerant networks, in: Proceedings of IEEE ICDCS, 2011.
[16] W. Peng, F. Li, X. Zou, J. Wu, The virtue of patience: offloading topical cellular content through opportunistic links, in: Proceedings of IEEE MASS, 2013, pp. 402–410.
[17] K. Chen, H. Shen, H. Zhang, Leveraging social networks for P2P content-based file sharing in disconnected manets, IEEE Trans. Mobile Comput. 13 (2) (2014) 235–249.
[18] W. Gao, G. Cao, User-centric data dissemination in disruption tolerant networks, in: Proceedings of IEEE INFOCOM, 2011.
[19] S. Bayhan, E. Hyytiä, J. Kangasharju, J. Ott, Seeker-assisted information search in mobile clouds, in: Proceedings of ACM SIGCOMM workshop on Mobile cloud computing, 2013, pp. 9–14.
[20] E. Hyytiä, S. Bayhan, J. Ott, J. Kangasharju, Searching a needle in (linear) opportunistic networks, in: Proceedings of ACM MSWiM, 2014, pp. 187–196.
[21] N. Golrezaei, K. Shanmugam, A. Dimakis, A. Molisch, G. Caire, Femtocaching: wireless video content delivery through distributed caching helpers, in: Proceedings of IEEE INFOCOM, 2012.
[22] G.M. Lee, S. Rallapalli, W. Dong, Y.-C. Chen, L. Qiu, Y. Zhang, Mobile video delivery via human movement, in: Proceedings of IEEE SECON, 2013.
[23] K. Chen, H. Shen, Maximizing P2P file access availability in mobile ad hoc networks though replication for efficient file sharing, IEEE Trans. Comput. 64 (4) (2015) 1029–1042.
[24] M. Grossglauser, D. Tse, Mobility increases the capacity of ad hoc wireless networks, IEEE/ACM Trans. Netw. 10 (4) (2002) 477–486.
[25] 2012, IEEE standard for information technology telecommunications and information exchange between systems local and metropolitan area networksspecific requirements part 11: Wireless lan medium access control (MAC) and physical layer (PHY) specifications, IEEE Std 802.11-2012.
[26] J. Huang, F. Qian, A. Gerber, Z.M. Mao, S. Sen, O. Spatscheck, A close examination of performance and power characteristics of 4G LTE networks, in: Proceedings of ACM MobiSys, 2012, pp. 225–238.
[27] X. Zhuo, W. Gao, G. Cao, Y. Dai, Win-coupon: an incentive framework for 3g traffic offloading, in: Proceedings of IEEE ICNP, 2011.
[28] J. Burgess, B. Gallagher, D. Jensen, B.N. Levine, Maxprop: routing for vehicle-based disruption-tolerant networks, in: Proceedings of IEEE INFOCOM, 6, 2006, pp. 1–11.
[29] Y. Huang, T.Z. Fu, D.-M. Chiu, J. Lui, C. Huang, Challenges, design and analysis of a large-scale P2P-VoD system, in: ACM SIGCOMM Computer Communication Review, 38, 2008, pp. 375–388.
[30] W. Wang, V. Srinivasan, M. Motani, Adaptive contact probing mechanisms for delay tolerant applications, in: Proceedings of ACM MobiCom, 2007, pp. 230–241.
[31] X. Zhuo, Q. Li, W. Gao, G. Cao, Y. Dai, Contact duration aware data replication in delay tolerant networks, in: Proceedings of IEEE ICNP, 2011.
[32] R. Groenevelt, Stochastic models in mobile ad hoc networks, University of Nice, 2010 Ph.d thesis.
[33] H. Zhu, L. Fu, G. Xue, Y. Zhu, M. Li, L.M. Ni, Recognizing exponential inter-contact time in VANETs, in: Proceedings of IEEE INFOCOM, 2010.
[34] S.M. Ross, Stochastic processes, Wiley, 1983.
[35] N. Eagle, A. Pentland, Reality mining: sensing complex social systems, Personal Ubiquitous Comput. 10 (4) (2006) 255–268.
[36] S. Liu, A.D. Striegel, Exploring the potential in practice for opportunistic networks amongst smart mobile devices, in: Proceedings of ACM MobiCom, 2013, pp. 315–326.
[37] S.P. Boyd, L. Vandenberghe, Convex Optimization, Cambridge university press, 2004.
[38] S.H. Low, D.E. Lapsley, Optimization flow control–I: basic algorithm and convergence, IEEE/ACM Trans. Netw. 7 (6) (1999) 861–874.
[39] L. Breslau, P. Cao, L. Fan, G. Phillips, S. Shenker, Web caching and zipf-like distributions: evidence and implications, in: Proceedings of IEEE INFOCOM, 1999.