Collaborative Content Distribution with Device-to-Device Communications

by

Jianguo Xie

Bachelor of Computer Science and Technology,
University of Science and Technology of China, China, 2015

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Computer Science

In the Graduate Academic Unit of Computer Science

Supervisor(s):	Wei Song, Ph.D., Faculty of Computer Science
Examining Board:	Huajie Zhang, Ph.D., Faculty of Computer Science
	John M. DeDourek, M.S., Faculty of Computer Science
	Julian Meng, Ph.D., Dept. Electrical & Computer Engineering

This thesis is accepted

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

May, 2017

Abstract

With the increasing penetration of smart devices, device-to-device (D2D) communications offer a promising paradigm to accommodate the ever-growing mobile traffic and unremitting demands. The redundant storage and communication capacities of smart devices can be exploited for collaborative content caching and distribution. In this thesis, we first studied the D2D pairing problem, which appropriately pairs a device requesting a content item with a nearby device which caches the requested item. We formulated the D2D pairing problem as an integer linear program (ILP) and developed a heuristic channel-aware algorithm to solve it. Computer simulations were conducted to compare the channel-aware algorithm with the optimal solution, as well as a minimum distance-based algorithm and a random algorithm. The results show that the channel-aware algorithm outperforms the random algorithm in terms of total number of served D2D pairs and average latency of served pairs.

Then, we studied the message allocation problem, which allocates the message requests to be served by the cache devices via D2D multicast. Aiming to minimize the total transmission cost or maximize the gain in cost saving for the base station (BS), this message allocation problem can be formulated from different perspectives, as a weighted set cover problem (WSCP), a hypergraph matching problem, or a multiple-choice knapsack problem (MCKP). We compared three

algorithms for the formulated problems, including a greedy algorithm, a heuristic algorithm based on Lagrangian relaxation, and a fully polynomial-time approximation scheme (FPTAS), respectively. The simulations results show that the WSCP based algorithm outperforms the other two in static scenarios in terms of D2D offload ratio and total cost. In dynamic scenarios, the MCKP based algorithm performs the best because the approximation guarantee of the FPTAS results in solutions closest to the optimum.

Acknowledgements

I would like to gratefully acknowledge the following people for their help and support during my study at UNB.

Firstly, I would like to give the sincere gratitude to my supervisor Dr. Wei Song. She spent a large amount of time in guiding me and helping with my study and my research. Beyond that, she taught me to treat the science with a rigorous attitude and carry my responsibility without the excuses.

Secondly, I want to thank my friends Xi Tao and Li Ji for helping me enhance my writing skills. And I want to thank my fellow Yiming Zhao for discussing the questions with me. Also, I am grateful to all staff of the Faculty of Computer Science for building such a nice environment and offering relevant resources for my study and research.

Thirdly, I want to express my appreciation for my thesis examining committee members, Professor John DeDourek, Professor Huajie Zhang, and Professor Julian Meng. Their comments are important for my thesis.

In the end, I want to thank my friends and families for supporting me with company or love in my daily life so that I could finish my study at UNB successfully.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

3GPP	The Third Generation Partnership Project
5G	The five generation cellular networks
AP	Access point
BS	Base station
CCN	Content-centric networking
CDF	Cumulative probability distribution
CDN	Content delivery network
D2D	Device-to-device
D2D LAN	D2D local area network
DC-DC	Direct D2D communications with device controlled link establishment
DC-OC	Direct D2D communications with operator controlled link establishment
DR-DC	Device relaying with device controlled link establishment
DR-OC	Device relaying with operator controlled link establishment
EPC	Evolved packet core
FPTAS	Fully polynomial-time approximation scheme
GO	Group owner
ILP	Integer linear program
SINR	Signal-to-interference-plus-noise ratio
LCFS	Last-come first-served

MCKP	Multi-choice knapsack problem
P2P	Peer to peer
QoS	Quality-of-service
RAN	Radio access network
WLAN	Wireless local area network
WSCP	Weighted set cover problem

List of Symbols

D	Set of requesting devices
S	Set of caching devices
M	Set of content items (files)
R	Set of requests
m	Number of files in M
Φ	Radius of cell coverage region
L	Maximum D2D transmission range
P_d	Transmit power of D2D transmitters
P_c	Transmit power of cellular uplink user
P_{BS}	Transmit power of BS
$P_{\min}$	Target transmission success possibility
$d_{j,k}$	Distance between transmitter j and receiver k
$h_{j,k}$	Channel gain between transmitter j and receiver k
$d_{c,k}$	Distance between cellular uplink user and receiver k
$h_{c,k}$	Channel gain between cellular uplink user and receiver k
d_c	Distance between cellular uplink user and D2D receiver
d_k	Distance between D2D transmitter and D2D receiver
ξ_k	Received SINR at requesting device k
r_k	Achievable data rate at requesting device k

α	Path-loss exponent
β	Decoding SINR threshold
ϵ	Accuracy parameter for MCKP
B	Channel bandwidth
C	Cost budget of cache devices
F	File size of each message
$r_{\min}$	Minimum date rate required for D2D links
σ^2	Additive noise power
γ_c	Exponent of Zipf distribution for caching
γ_r	Exponent of Zipf distribution for requesting
λ_r	Mean arrival rate of requesting devices
λ_c	Mean arrival rate of caching devices
λ_q	Mean arrival rate of requests
μ_r	Mean departure rate of requesting devices
μ_c	Mean departure rate of caching devices

Chapter 1

Introduction

1.1 Motivation

The past decade has witnessed an explosive growth of mobile traffic, smart devices, and assorted applications. To accommodate the surging demands, the traditional network architecture is experiencing evolutionary changes. For example, network densification through small cells and device-to-device (D2D) communications are dominant themes for the fifth-generation (5G) mobile networks. In particular, D2D enables direct communications between mobile devices in a peer-to-peer (P2P) fashion bypassing the base station (BS). This creates another "cell tier" that underpins small cells and offers various benefits such as expanding coverage, offloading traffic, improving energy efficiency, and facilitating proximity-based services.

There are many promising D2D applications for 5G mobile networks, such as proximity services and emergency communications. In proximity services, users can discover nearby users and share data with each other. D2D communications also find broad applications in disaster response, when the network infrastructure

is damaged by extreme weather, natural disasters, biohazards, terrorist threats and other life-threatening events. For example, D2D communications can be used as emergency communications to help people get connected and stay informed. Moreover, D2D communications can be applied in content distribution and traffic offloading, which can expedite the distribution process in a cost-effective manner and alleviate congestion of the cellular network.

Collaborative content distribution highlights the collaboration between users who share content with each other. Some users would serve as the content providers to fulfill the content requests of other users with their cached content. D2D communications can facilitate collaborative content distribution among users in close proximity. The short transmission range can potentially achieve high data rate and short latency with low energy consumption. This is particularly favourable for video services that demand large bandwidth and real-time service requirement. In addition, with the collaborative content distribution over D2D communications, much video traffic can be offloaded from the cellular network, thus benefiting other non-D2D cellular users as well.

1.2 Research Objectives

This thesis focuses on two research problems for D2D-assisted content distribution. The first one is the D2D pairing problem that appropriately matches a requesting device with a cache device to fulfill the content request. The D2D paring should take into account the content availability, devices' spatial distribution, and D2D interference and link conditions, so that we can match as many requesting and cache devices as possible to deliver the content successfully. In the second problem, we further consider D2D multicast to reduce energy consumption in content

distribution. When a cache device stores multiple messages (i.e., content items), it is critical to properly select the multicast message so that it can satisfy as many requesting devices as possible. In this part of work, we evaluate different message allocation strategies, which aim at minimizing the content distribution cost or maximizing the gain by diverting content requests to cache devices.

1.2.1 D2D Pairing

To fulfill the goal of offloading content distribution traffic from the cellular network via D2D links, we need to appropriately match a requesting device with a cache device which can deliver the requested content via D2D communications. We call this the D2D pairing problem. On one hand, the cache device should satisfy the request with a minimum transmission rate requirement to avoid unacceptable latency. On the other hand, we must limit the D2D interference among the D2D links so that more D2D traffic can be accommodated.

In this work, we consider both the static scenario and dynamic scenario. In the static scenario, there are a fixed set of requesting devices and a fixed set of cache devices. Each requesting device aims to download one message from a given library, while each cache device only stores one message from the library. The D2D pairing needs to match as many requesting devices to their feasible cache devices as possible. Here, it is assumed that all the D2D links share the same cellular uplink channel. For one thing, the uplink resources are often less utilized than are the downlink resources. For another, the D2D interference can be better handled at the more powerful BS [1]. The goal of D2D pairing is to maximize the number of successful pairs which can fulfill the content requests at a data rate not less than the minimum threshold.

In the dynamic scenario, we consider the dynamic arrivals and departures of

requesting and cache devices into the cell. The content requests also occur in a random pattern. Specifically, we consider a time-slotted scenario where the content requests are processed periodically. In each period, the requesting and caching information is collected to conduct a round of D2D pairing. It is possible that not all requests are fulfilled in the current round. Hence, the unpaired requesting devices will be placed in a waiting queue to be considered for the next round. Hence, we need to consider the queueing mechanism that deals with the backlogged requests.

1.2.2 D2D Multicast Message Allocation

Even though a good pairing strategy can offload traffic from the cellular network, it is important to consider the energy consumption during this offloading process. The energy consumption of battery-powered D2D users should be limited to really harvest the benefits of D2D communications. Multicasting the requested content can potentially save more energy than unicasting the requested items one by one. Therefore, we further consider multicast for the content distribution scenario with D2D communications to improve energy efficiency.

Here, we consider a more general content distribution scenario, in which each cache device possesses a subset of messages from a library, while a requesting device is allowed to demand multiple messages. Hence, it is beneficial that a cache device broadcasts one or more messages to multiple requesting devices that stay in the close proximity (transmission range). Thereby, certain traffic can be offloaded from the BS and the content is distributed in a more cost-effective manner. The key is how to allocate multicast messages for the cache devices to achieve certain performance goal.

In this work, we study different strategies that allocate messages to cache devices

which can deliver the allocated messages by multicast. From the perspective of the whole system, it is preferable that the total cost of distributing the requested messages is minimized. In this case, the message allocation problem can be formulated as a weighted set cover problem, which is an NP-hard problem but can be solved by a greedy algorithm with an approximation guarantee.

In a time-slotted dynamic scenario, the content requests are processed periodically. To limit the energy consumption of the cache devices, we can restrict that at most one message is selected for one cache device, and the total cost for all cache devices is upper bounded by a budget. From the perspective of the BS, it would be preferable that the BS maximizes its gain from diverting content requests to cache devices. This gain can be quantified as the difference between the multicast cost for multiple messages and the total unicast cost for each individual message. In this case, the message allocation problem can be formulated as the multiple-choice knapsack problem (MCKP). MCKP is also NP-hard but has a fully polynomial-time approximation scheme (FPTAS) [2].

In [3], the authors consider a similar message allocation problem based on a simplified channel model that neglects D2D interference. Different from the above, they aim at minimizing the total content distribution cost, although they also limit the number of multicast messages to one for each cache device. In addition, they do not restrict the cost that cache devices can afford. It is proved that this problem is also NP-hard. In [3], the authors apply a heuristic algorithm and the subgradient method to the Lagrangian dual problem to obtain a feasible solution. In this work, we also adapt their solution to our content distribution scenario with a more realistic channel model. This approach is compared with the above message allocation strategies in the static scenario and dynamic scenario.

1.3 Thesis Organization

This thesis contains five chapters, and the remaining of this thesis is organized as follows. Chapter 2 introduces the background knowledge and the related works we have reviewed. Chapter 3 presents a channel-aware D2D pairing algorithm to offload content distribution traffic via D2D links and the corresponding evaluation results in both static scenarios and dynamic scenarios. Chapter 4 introduces three different algorithms for multicast message allocation from three different perspectives and presents the results that compare them in the static and dynamic scenarios. Chapter 5 concludes this thesis and gives the directions of future work.

Chapter 2

Background and Related Work

This chapter describes the background knowledge associated with D2D communications and content distribution. It involves a broad range of fields including WiFi-Direct and LTE-Direct. It also introduces some related works about content distribution using D2D communications.

2.1 D2D Communications

D2D communications enable devices to directly communicate with each other without routing data through the BS, thereby offering various benefits such as expanding coverage, offloading traffic, improving energy efficiency, and facilitating proximity-based services. First, due to the short transmission range between proximate devices, D2D communications can achieve high data rates and low latency while saving energy. Besides, the direct transmission bypassing the BS may increase radio resource utilization efficiency and possibly reduce routing through the core network [1]. Moreover, the direct paths may offload cellular traffic and alleviate congestion.

In [4], the authors classify D2D communications into two categories, namely, D2D

direct and D2D local area networks (D2D LAN). D2D direct refers to one-hop communications, while D2D LAN refers to multi-hop communications. However, according to [1], D2D networking mainly consists of local, opportunistic, and single-hop communications. More importantly, the D2D technology typical relies on assistance from the network infrastructure for control functions such as session setup, resource allocation, and group formation. This is different from mobile ad hoc networks (MANETs), which have been extensively studied in the past decade. MANETs often involve multiple long hops and have no infrastructure support.

In [5], D2D communications are categorized based on the spectrum on which D2D communications operated. Inband D2D communications refer to the D2D communications operate on licensed spectrum, while outband D2D communications operate on unlicensed spectrum. Furthermore, inband D2D communications can share the spectrum in two different manners, *i.e.*, overlay spectrum sharing and underlay spectrum sharing. In overlay spectrum sharing, the licensed spectrum is divided into two parts and assigned to D2D and cellular users, respectively. Intra-cell interference between D2D and cellular users can thus be completely avoided through orthogonal channel assignment. Underlay spectrum sharing allows D2D and cellular users to access the same spectrum simultaneously, which can enhance spectrum efficiency but result in non-negligible co-channel interference to the cellular network users. Therefore, effective resource allocation and interference management are required to address the co-channel interference and to provide high performance for both D2D users and cellular users.

In recent years, there has been a rich literature on resource allocation for D2D communications with underlay spectrum sharing. Interestingly, many solutions apply a game-theoretical approach or an auction-based design, considering the distributed and autonomous nature of D2D users. The main challenge is to limit

the co-channel interference while achieving high spectral efficiency. In [6], a noncooperative game is formulated for D2D underlay resource allocation. A distributed interference-aware algorithm is proposed to derive a Nash equilibrium while maximizing energy efficiency. In [7], Xiao *et al.* present a Bayesian overlapping coalition formation game to analyze the spectrum sharing problem. In [8], a coalition formation game is proposed to address the uplink resource allocation problem for D2D and cellular users, which is proved to converge to a *Nash stable* coalition structure with geometric rate and enhances the system sum rate.

When incorporating D2D communications into cellular networks, the operators can have different levels of control over D2D communications. From this perspective, D2D communications can be divided into four main types [9]:

- Device relaying with operator controlled link establishment (DR-OC): Devices communicate with base stations through other devices. The operator communicates with the relaying devices for partial or full control link establishment.

- Direct D2D communications with operator controlled link establishment (DC-OC): Devices communicate with each other directly. Base stations will only help them to build a link.

- Device relaying with device controlled link establishment (DR-DC): Devices communicate with other devices through other devices. The link between the source and destination is controlled by devices.

- Direct D2D communications with device controlled link establishment (DC-DC): Devices communicate with each other directly without participation of the base station.

In DR-OC and DC-OC, resource allocation is executed by base stations. On the other hand, in DR-DC and DC-DC, there is no centralized entity to supervise the resource allocation between devices. In addition to resource allocation and interference management, security is an important issue for D2D communications, since data can be routed through other devices. One possible solution is to use a closed access mechanism. In a closed access, each device has a list of trusted devices, while other devices off the list can only communicate with the device through the base stations. The other possible solution is to use a proper encryption for the information exchanged between devices.

2.2 Wi-Fi Direct and LTE Direct

2.2.0.1 Wi-Fi Direct

As mentioned in Section 2.1, inband D2D communications operating under licensed spectrum have to deal with challenging issues such as resource allocation and interference management. On the other hand, outband D2D communications over the unlicensed spectrum are another alternative to offload traffic from the cellular network. Wi-Fi Direct is such an example which can be used for D2D communications in the wireless local area networks (WLANs).

The authors in [10] provide a thorough overview of the Wi-Fi Direct technology. As the authors describe, Wi-Fi Direct devices communicate by establishing groups. The device in a group that has the functionality of the access point (AP) in the regular Wi-Fi network is called group owner (GO), while the other devices in the group are called group clients. Once a group is established, other devices can join the group as a client. There are several ways in which two devices can establish a group.

- Standard: In this case, the devices need to find each other, and then negotiate which one will be the GO. Firstly, the devices scan around themselves to find existing groups. Secondly, a discovery algorithm is executed. In this algorithm, the devices alternate between two states: Listen and search. In the search state, the devices send requests for grouping. In the listen state, the devices accept the request and respond. The time spent in each state is randomly distributed between 100 ms and 300 ms. Thirdly, the devices negotiate with each other using three-way handshake to decide their roles in the group. For example, suppose that A and B are the devices that are negotiating with each other. This negotiation process consists of a request from A to B, a response from B to A, and a confirmation from A to B. Lastly, the devices establish a secure connection using Wi-Fi Protected Setup (WPS) and configure the address.

- Autonomous: In this case, a device can form a group autonomously where it immediately becomes the GO. Other devices can find this group using the scan mechanism and join the group. Then the devices establish a secure connection using WPS and configure the address.

- Persistent: This case only differs from the standard approach in one apsect. For two devices that have formed a group, the negotiation between the devices only includes two steps, namely, invitation and invitation response. The invitation can be sent from either of the devices.

Energy efficiency is important in D2D communications. In Wi-Fi Direct, there are two power saving mechanisms: Opportunistic power save and notice of absence.

- Opportunistic power save: After a time window, if the GO determines that all the clients in the group are in the power saving state, then the GO will

go into power saving state until it is woken up by clients.

- Notice of absence: In this case, the GO can announce time intervals, during which none of the clients is allowed to communicate with the GO.

According to the Wi-Fi Direct specification, the GO cannot be changed for a formed group. In [11], the authors propose a protocol for D2D communications using Wi-Fi Direct and allow GO transfer when necessary. When the base station (BS) detects that another device in the group has better channel quality than the current GO, then GO transfer occurs. Specifically, after a group is established, the GO will notify the BS that this group has been established. The notification information includes all identities of the group clients. After the group notification, the BS will send a message to all group clients through the GO over Wi-Fi.
Then all the clients will respond to this message. This process can verify that all the clients are already the members of the group over Wi-Fi. When a new device joins the group, the GO of the group will notify the BS and the BS will send a message to verify that this device is already in the group over Wi-Fi. When a device leaves the group, the GO will notify the BS of this fact and the BS doesn't need to verify that it is gone. In this protocol, all the communications between a group and the BS work as follows. The clients communicate with the GO over Wi-Fi and the GO communicate with the BS over cellular spectrum.

2.2.0.2 LTE Direct

As discussed in Section 2.1, inband D2D communications operate over licensed spectrum, over which the cellular network operator can have tight control. LTE Direct is an inband D2D technology and has been incorporated into the Release 12 by the 3rd Generation Partnership Project (3GPP) for Long Term Evolution

Advanced (LTE-Advanced). Although LTE Direct targets at public safety networks in 3GPP Release 12, it has many other possible commercial applications in social networking, large file transfer, location-based services, online advertising, and so on.

Different from the asynchronous message-based device discovery in Wi-Fi Direct, LTE Direct allows devices to broadcast their services at the physical layer, and devices wake up synchronously on a periodical basis to discover all devices within range. LTE Direct allows for the discovery of thousands of devices and their services in the proximity of around 500m. The discovery occurs autonomously with little impact on the device battery life while protecting users' privacy. According to the comparison study in [12], LTE Direct enables discovery of approximately 16 times more devices than Wi-Fi Direct. It is also pointed out that low duty cycle for discovery in LTE Direct leads to significant power improvement over Wi-Fi Direct. Furthermore, LTE Direct can leverage the LTE infrastructure for network assisted connection setup. Low priority connections can even yield to high priority connections in proximity.

Proximal discovery technology empowers mobile devices and applications to continuously discover friends, offers, and other relevant value in the close proximity. In addition to LTE Direct, existing proximal discovery approaches include location-based technologies or Bluetooth Low Energy (BT-LE) proximity beacons [13]. BT-LE proximity beacons operate in the unlicensed spectrum and only broadcast discovery services through an associated mobile app to nearby users within a range of approximately 50m. In contrast, the location-based proximal discovery approaches continuously track users' locations and determine what is in the vicinity by comparing against a centralized, cloud-based database. Privacy and battery drain are main concerns of these location-based approaches due to

perpetual location tracking and constant network pings to access the cloud.

LTE Direct runs on the licensed spectrum and provides service layer discovery to mobile applications. Mobile applications can rely on LTE Direct to monitor specific mobile application services on other devices, and can also announce their own services. This allows mobile applications to be closed while LTE Direct does the work. LTE Direct would notify the mobile applications when it detects a match to the monitor it sets. Using only less than 1% of the uplink spectrum resources, LTE Direct-enabled devices can periodically (*e.g.*, every 10 seconds) broadcast their needs and services via a beacon called "expression" toward all devices in proximity. Expressions are 128-bit service-layer identifiers that represent many different things, such an identity, a service, an interest, or a location [13]. Nonetheless, LTE Direct can leverage the tightly controlled LTE network for timing, resource allocation, as well as user authentication. For example, all devices can be synchronously woken up for efficient discovery, thereby prolonging battery lifetime. As a distributed discovery approach, LTE Direct enables devices to autonomously determine relevance by filtering expressions. Therefore, power consumption is further reduced by avoiding frequent costly access to the cloud-based database.

2.3 Content Distribution with D2D Communications

Traditionally, content providers often deliver their content via a content delivery network (CDN), which is a globally distributed network of proxy servers deployed in multiple data centers. The pervasive smart devices introduce abundant resources at the network edge. Their redundant storage and communication capaci-

ties can be exploited for collaborative content distribution, which can complement traditional CDNs with lower delivery cost and higher performance. On the other hand, D2D-assisted content distribution can help offload traffic from the cellular network.

In [14], the authors propose an approach for offloading cellular traffic onto D2D communications. In their proposed approach, when the clients are in proximity, they can exchange data with each other in D2D communications rather than in cellular communications. In this way, when the devices in proximity have the content wanted by each other, they can access it directly from nearby devices and thus offload traffic from the cellular network. In this approach, a trusted third party server is expected to implement device discovery and content discovery. Users can publish contents at the server. When a device requests to access certain contents that have been published at the server, the server can list all the devices that provide these contents. The server will track the locations of the publishing devices and the devices that request access to the contents. When these devices are in proximity, a D2D communication session can be established to transfer the contents between them. If the server cannot find a device in proximity that holds the requested content, regular cellular communications will be started for the transfer.

In [15], the authors compare Wi-Fi cooperation with D2D-based multicast for content distribution. With Wi-Fi cooperation, when a set of users S download the same files, each user downloads different parts of the files over the cellular network using unicast transmission. Then, the users exchange their downloaded parts with others over Wi-Fi using multicast transmission. In the D2D-based multicast approach, only users in S that have the best channel states are selected to download the files over the cellular network using multicast transmission. Then

these users will distribute the files to the rest of users in S over D2D communications. The results in [15] show that both approaches can significantly improve the throughput and energy consumption. In addition, the D2D-based multicast approach over the unlicensed spectrum performs better than that over the licensed spectrum in terms of energy consumption.

The authors in [16] propose an approach that pre-fetches video data in caches located at Wi-Fi hotspots and exploits D2D communications to obtain parts of video data from nearby devices. The key of this approach is the mobility and throughput prediction. Mobility prediction will provide how many hotspots the user will encounter, when they will be encountered, and how long the user will stay within the range of each hotspot. The device will instruct the caches in the hotspots to pre-fetch certain parts of the video data according to the predictions. The time spent on pre-fetching data for the caches and obtaining data in D2D communications can be adjusted according to the throughput prediction. In [17], the authors discuss the techniques related to caching in current mobile networks, focusing on three key questions: Where to cache, what to cache, and how to cache? Based on trace-driven evaluation, it compares between evolved packet core (EPC) caching, radio access network (RAN) caching, and content-centric networking (CCN) based caching. The CCN-based caching is found to excel in reducing content access delay and traffic load.

In [3], a randomized auction mechanism is proposed to offload message requests from the BS to nearby devices, so that message requests are fulfilled via D2D communications with lower costs. In [18], we consider a similar content distribution scenario but further restrict the cost budget that the BS can afford. The cost profile takes into account the wireless channel fading to quantify the resource cost that a source device takes to fulfill a content request while meeting a mini-

mum quality-of-service (QoS) requirement. The proposed reverse auction mechanism maintains an approximation guarantee for the worst case while subject to a polynomial computational time. Meanwhile, our mechanism ensures desirable economic properties including truthfulness in expectation and individual rationality. In [19], Jiang *et al.* formulate the local caching problem for D2D content distribution as a Knapsack problem to address the limited storage capacities of devices, and develop a distributed caching algorithm based on message popularity estimates. In addition, they explore the sender-receiver pairing problem as a maximum weighted bipartite matching problem, and design a matching algorithm as an auction. However, the work in [3, 19] neglects D2D interference but only respects the classic half-duplex constraint of wireless communications.

The work in [20] focuses on a D2D content distribution scenario, where a helper device can send a file to a requesting device within a *collaboration distance*. The authors study a probabilistic caching policy in which each device caches a file according to a probability distribution. The optimal caching distribution is obtained to maximize the offloading ratio. It is shown that the maximal offloading ratio increases with the collaboration distance. Nonetheless, this conclusion is conditioned on the assumption that the interference among D2D links is treated as noise. If this assumption is relaxed, the collaboration distance actually results in two conflicting effects. On one hand, a larger collaboration distance increases the request hit ratio over helper devices. On the other hand, more D2D pairs are introduced within a larger collaboration distance, which obviously leads to higher interference among the D2D links and thus decreases the decoding success probability or the transmission rate. In [21], Golrezaei *et al.* considers such conflicting effects and analyze the optimal collaboration distance. Unfortunately, their work is based on a simplified physical-layer communication model, where a

cell is divided into equal-area square clusters and only one D2D link is allowed per cluster. This simplification is reasonable since they mainly examine a deterministic caching policy and a random caching policy. However, a more realistic interference-aware D2D communication model is essential for D2D pairing since the interference directly affects the receiving QoS and therefore the pairing feasibility.

Chapter 3

Channel-Aware D2D Pairing for Collaborative Content Distribution

In this chapter, we study the D2D pairing problem, which appropriately pairs a device requesting a content file with a nearby device which caches the requested file. First, we formulate the D2D pairing problem as an integer linear program (ILP). Due to the complexity of the problem, we further develop a heuristic channel-aware D2D pairing algorithm. Computer simulations are conducted to compare the channel-aware algorithm with the optimal solution, as well as a minimum distance-based algorithm and a random algorithm. We consider both the static scenario and dynamic scenario with arrivals and departures of requesting and cache devices. The simulation results demonstrate that the channel-aware algorithm outperforms the random algorithm in terms of total number of served D2D pairs and average latency of served pairs.

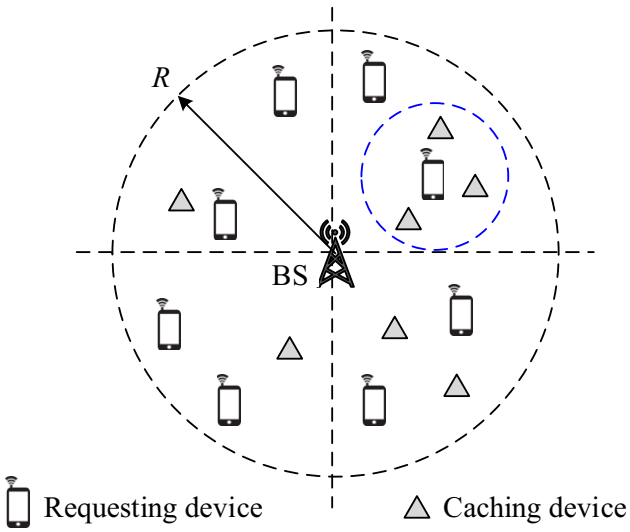


Fig. 3.1: D2D content distribution scenario.

3.1 System Model

3.1.1 Content Distribution Model

In this work, we consider a content distribution scenario depicted in Fig. 3.1. A set of requesting devices, D, are requesting content items (referred to as a "file" henceforth for simplicity) from a "library" of size m, denoted by M. The circular disk $\mathcal{C}$ denotes the coverage region of a base station (BS) centered at the origin. The requesting devices enter region $\mathcal{C}$ following a Poisson process of mean rate λ_r, and the requesting devices are uniformly distributed within $\mathcal{C}$. It is further assumed that the staying time of the requesting devices follows an exponential distribution with an average $1/\mu_r$. Each time a device $k \in D$ only requests one file from the library M independently according to a Zipf popularity distribution, which has been shown to be a good model that captures the popularity of video

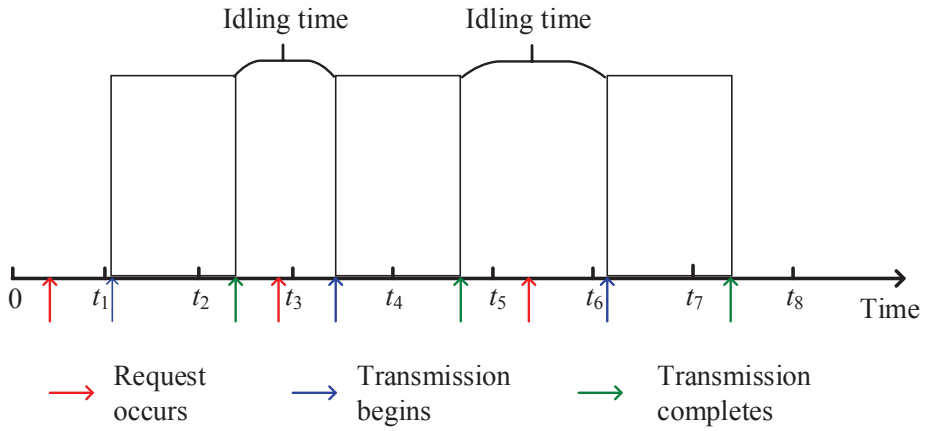


Fig. 3.2: On-off dynamics of content requests.

clips [22]. Specifically, a device in D requests file i with a probability

$$q_i = \frac{\frac{1}{i^{\gamma_r}}}{\sum_{j=1}^{m} \frac{1}{j^{\gamma_r}}}, \qquad 1 \leq i \leq m, \gamma_r > 0$$

where the exponent γ_r characterizes the relative popularity of files. A larger value of γ_r implies that more requests are concentrated on fewer files. Let m_k^{r} denote the file requested by device k. The set of requests from the devices in D is denoted by Q. For the same requesting device, the idling time between a completed request and the next one follows an exponential distribution of mean $1/\lambda_q$. After downloading a content item, the user often spends some watching time before requesting the next item. The interactive service nature is modelled by the alternating "on" phase (in transmission) and "off" phase (being idle), as illustrated in Fig. 3.2.

In addition, there are another set of devices, S, which enters the BS's coverage region $\mathcal{C}$ according to a Poisson process of mean rate λ_c and stays therein for an exponentially distributed staying time with mean $1/\mu_c$. Considering the random caching policy studied in [21], we assume that each device $j \in S$ caches one file

in the library M according to a Zipf distribution with exponent γ_c. That is, a device in S caches file i with a probability

$$f_i = \frac{\frac{1}{i^{\gamma_c}}}{\sum_{j=1}^{m} \frac{1}{j^{\gamma_c}}}, \qquad 1 \leq i \leq m, \gamma_c > 0.$$

Here, we denote the file cached at device j by m_j^{t}. The cache devices can be pooled together and form the fog nodes in fog computing. Then, instead of fulfilling each content request by the BS, it is potentially beneficial to serve some requesting devices in D via D2D communications with nearby cache devices in S. Considering that each device is equipped with a single omnidirectional antenna, each cache device can serve at most one requesting device. This D2D content distribution not only offloads traffic from the BS but also is more cost-effective in view of the close proximity.

3.1.2 Channel Model

Here, we consider a D2D underlaid cellular network, where the D2D links share the uplink spectrum of regular cellular users. There are several good reasons for favoring the use of uplink resources, such that the uplink resources are often less utilized, and the BS is more powerful in interference mitigation [1]. Assume that all potential D2D transmitters of the cache devices in S all share the same cellular uplink channel. This cellular channel is preferably unused, or it is allocated to a cellular user that is uniformly located within the coverage region of the cell, $\mathcal{C}$. Supposing that the request from device $k \in D$ is fulfilled by device $j \in S$, the

received signal at D2D receiver k is written as

$$\begin{aligned} y_k =& \sqrt{P_d} d_{j,k}^{-\frac{\alpha}{2}} h_{j,k} x_k + \sqrt{P_c} d_{c,k}^{-\frac{\alpha}{2}} h_{c,k} x_c \\ &+ \sum_{j' \in S, j' \neq j} \theta_{j'} \sqrt{P_d} d_{j',k}^{-\frac{\alpha}{2}} h_{j',k} x_{\varphi(j')} + n_k. \end{aligned} \tag{3.1}$$

Here, α is the path-loss exponent, n_k is the additive noise at D2D receiver k distributed as $\mathcal{CN}(0, \sigma^2)$. Besides, $\theta_{j'}$ is a binary variable indicating whether transmitter $j' \in S$ is selected to serve a requesting device. For function $\varphi : S \mapsto D$, $\varphi(j')$ gives the receiver device in D that transmitter j' is matched to if there is one. For D2D transmitter j and the cellular user using the same uplink channel, x_k and x_c are their sent signals, respectively, P_d and P_c are their respective transmit power, $d_{j,k}$ and $d_{c,k}$ are their respective distance to D2D receiver k, and $h_{j,k}$ and $h_{c,k}$ are the corresponding distance-independent channel gain that captures the fading effect. Considering Rayleigh fading channels, $|h_{j,k}|^2$ and $|h_{c,k}|^2$ follow an exponential distribution of unit mean. As seen in (3.1), the received signal at D2D receiver k includes the expected signal, the interference from the cellular user, the integrated interference from all other active D2D transmitters in S, and the additive noise. The signal-to-interference-plus-noise ratio (SINR) at D2D receiver k is then given by

$$\xi_k = \frac{P_d d_{j,k}^{-\alpha} |h_{j,k}|^2}{P_c d_{c,k}^{-\alpha} |h_{c,k}|^2 + \sum_{j' \in S, j' \neq j} \theta_{j'} P_d d_{j',k}^{-\alpha} |h_{j',k}|^2 + \sigma^2}. \tag{3.2}$$

Then, the achievable data rate at receiver device k can be obtained as

$$r_k = B \cdot \log_2(1 + \xi_k) \tag{3.3}$$

where B is the carrier bandwidth. Given a minimum rate requirement $r_{\min}$, it implies that the received SINR cannot be less than a *decoding threshold* β, where $r_{\min} = B \cdot \log_2(1 + \beta)$.

3.2 D2D Paring Problem Analysis and Solutions

3.2.1 Research Problem

To fulfill the goal of offloading content distribution traffic from the cellular network via D2D links, we need to appropriately pair each requesting device with a cache device. On one hand, the cache device should satisfy the request with a transmission rate no less than $r_{\min}$ to avoid unacceptable latency. On the other hand, we must limit the D2D interference among the D2D links so that more D2D traffic can be accommodated. The pairing of requesting and cache devices is the research focus of this work.

Specifically, we consider a time-slotted scenario where the content requests are processed periodically, as depicted in Fig. 3.3. At the beginning of each period, the requesting and caching information collected in the previous period is used to conduct a round of D2D pairing. It is possible that not all requests are fulfilled in the current round. Hence, the unpaired requesting devices will be placed in a waiting queue to be considered for the next round. As seen, the pairing result not only affects the offloading efficiency in the current round but also influences the pairing subsequently. Hence, we need to consider the queueing mechanism that deals with the backlogged requests.

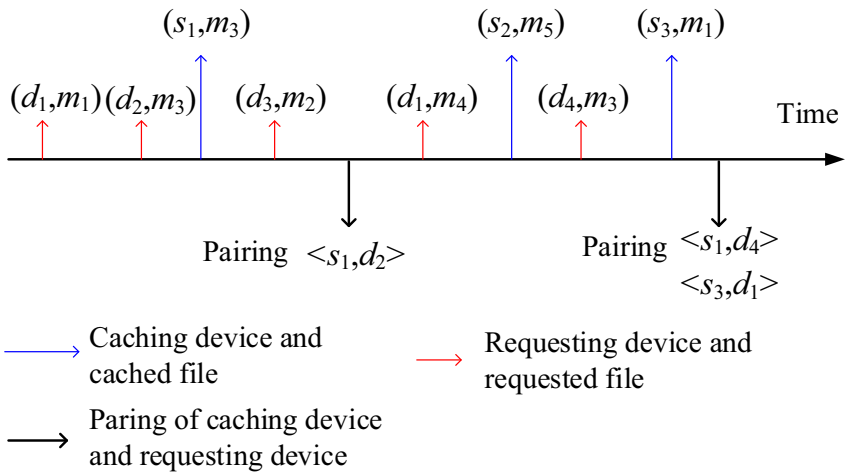


Fig. 3.3: Periodical processing of D2D pairing.

3.2.2 Problem Analysis

Given a particular pairing period in the content distribution scenario, we have set D for requesting devices with the corresponding content requests, and set S for cache devices with their cached files. Considering a requesting device is only paired with a feasible cache device, we model the feasibility relationship between S and D by a bipartite graph, for which Fig. 3.4(a) shows an example. An edge in Fig. 3.4(a) implies that a cache device contains the requested file of the corresponding receiver and satisfies the minimum data rate. In other words, the existence of edges only indicates the feasibility condition. If we want to relate a matching of the bipartite graph to the D2D pairing problem, we should further specify the edge weights to capture the D2D interference and SINR. When an edge is included in a matching for D2D pairing, the interference to be experienced by other potential pairs will be affected accordingly. As a result, the edge weights are not independent as in regular bipartite graphs but inter-dependent in our case. Hence, we cannot obtain an optimal D2D pairing by finding a maximum or minimum weighted bipartite matching, which is solvable in polynomial time.

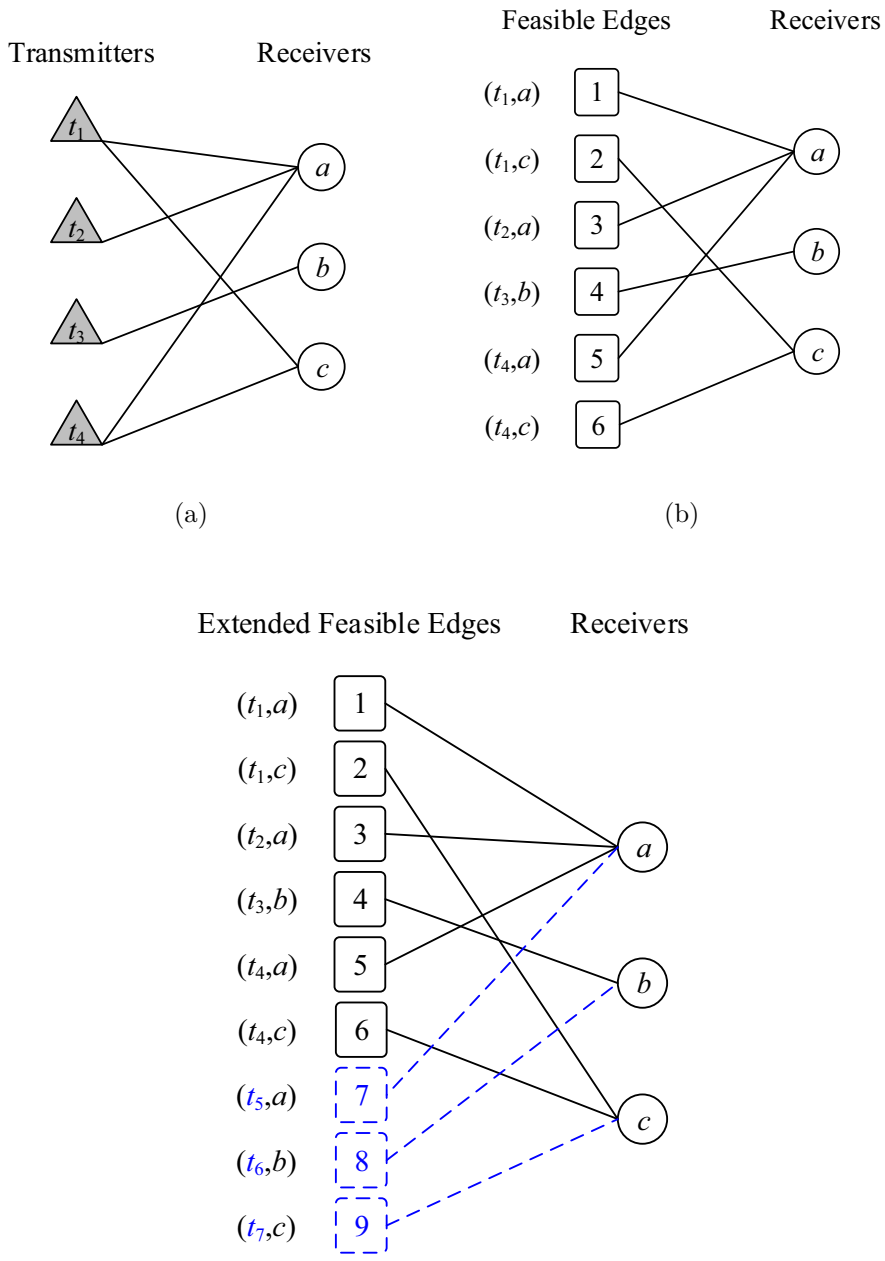


Fig. 3.4: Bipartite graph modeling for D2D pairing.

To incorporate the interference effect, we modify the bipartite graph by replacing the vertices on the left side with the edges. For example, Fig. 3.4(a) is restructured to Fig. 3.4(b). Based on the modified bipartite graph, the D2D pairing problem can be reformulated in detail. Referring to the left-side vertices as set G, we use a binary variable $x_{\ell k}$ to indicate whether edge (ℓ, k), $\ell \in G$ and $k \in D$, is selected for the D2D pairing. For each edge (ℓ, k), we define two related vector variables, $p_\ell = \{p_{\ell,k'} : \forall k' \in D\}$ and $w_\ell = \{w_{\ell,k'} : \forall k' \in D\}$, to represent the desired signals and unwanted corresponding interferences with respect to each potential receiver in D, respectively. For instance, selecting edge $(3, a)$ in Fig. 3.4(b) indicates that transmitter t_2 is paired to receiver a. Accordingly, we have $p_1 = \{s_{t_2,a}, 0, 0\}$ and $w_1 = \{0, s_{t_2,b}, s_{t_2,c}\}$, where $s_{t_2,k} = P_d d_{t_2,k}^{-\alpha} |h_{t_2,k}|^2, k \in \{a, b, c\}$, gives the received power at device k due to the transmission by device t_2. In addition, let $w_{c,k}$ denote the interference from the cellular user to receiver k. Then, an optimal D2D pairing means a maximum one-to-one matching that satisfies certain additional constraints. Specifically, the D2D pairing problem can be written as

$$\max. \quad \sum_{k \in D} \sum_{l \in G} x_{\ell,k} \tag{3.4a}$$

$$\text{s.t.} \quad \sum_{k \in D} x_{\ell,k} \leq 1, \forall \ell \in G \tag{3.4b}$$

$$\sum_{\ell \in G} x_{\ell,k} \leq 1, \forall k \in D \tag{3.4c}$$

$$\frac{\sum_{\ell \in G} x_{\ell,k} p_{\ell,k}}{\sum_{\ell \in G} \left(\sum_{k' \in D} x_{\ell,k'} \right) w_{\ell,k} + w_{c,k} + \sigma^2} \geq \beta, \tag{3.4d}$$

$$\forall \sum_{\ell \in G} x_{\ell,k} = 1, k \in D.$$

Here, constraint (3.4d) ensures that a selected pairing for receiver k satisfies the decoding condition for successful transmission. Overall, problem (3.4) aims to

maximize the total number of such successful pairs. The ratio of the objective value over the number of requesting devices (denoted by $N_r = |D|$) is actually the *satisfaction ratio* for the current period.

As seen in (3.4), constraint (3.4d) is defined only for the receivers that are successfully paired. To remove this condition and simplify the solution, we add N_r virtual transmitters, where each is dedicated to serve one distinct receiver in D. This extends the bipartite graph in Fig. 3.4(b) to Fig. 3.4(c), in which the set of vertices on the left side becomes $G' = G \cup H$ and H denotes N_r edges between the virtual transmitters and all receivers. For any edge $(\ell, k) \in H$ between a virtual transmitter j and receiver k, we have $w_\ell = \{0, ..., 0\}$ and $p_\ell = \{0, ..., s_{j,k}, ..., 0\}$, where $s_{j,k} = \beta \cdot (\sum_{\ell \in G} w_{\ell,k} + w_{c,k} + \sigma^2)$. This implies that a virtual transmitter causes zero interference to others, and the received power at its dedicated receiver is always high enough to achieve an SINR no less than the decoding threshold even if all potential real transmitters in S are selected for transmission. This extension ensures that there always exists a feasible solution that successfully pairs every receiver in D, *i.e.*, the matching between every virtual transmitter to its dedicated receiver. The objective value for problem (3.4) then becomes N_r. To prioritize the pairing of receivers with real feasible transmitters, we define a cost variable c_ℓ for each $(\ell, k) \in G'$ such that $c_\ell = \{c_{\ell,k'} : k' \in D\}$, where $c_{\ell,k'} = 0$ if $k' = k$ and $\ell \in G$, and $c_{\ell,k'} = \infty$ otherwise. Accordingly, problem (3.4) can be

revised to

$$\text{min.} \quad \sum_{k\in D}\sum_{l\in G'} x_{\ell,k}c_{\ell,k} \tag{3.5a}$$

$$\text{s.t.} \quad \sum_{k\in D} x_{\ell,k} \leq 1, \forall \ell \in G' \tag{3.5b}$$

$$\sum_{\ell\in G'} x_{\ell,k} = 1, \forall k \in D \tag{3.5c}$$

$$\frac{\sum_{\ell\in G'} x_{\ell,k}p_{\ell,k}}{\sum_{\ell\in G'}\Big(\sum_{k'\in D} x_{\ell,k'}\Big)w_{\ell,k} + w_{c,k} + \sigma^2} \geq \beta, \forall k \in D. \tag{3.5d}$$

As seen in problem (3.5), the objective function (3.5a), constraint (3.5c), and constraint (3.5d) extend the corresponding objective and constraints in problem (3.4). Defining the inner summation in the denominator of (3.5d) as an additional variable y_ℓ, we can reformulate (3.5) as an integer linear program (ILP) in (3.6):

$$\text{min.} \quad \sum_{k\in D}\sum_{l\in G'} x_{\ell,k}c_{\ell,k} \tag{3.6a}$$

$$\text{s.t.} \quad y_\ell \leq 1, \forall \ell \in G' \tag{3.6b}$$

$$y_\ell = \sum_{k\in D} x_{\ell,k}, \forall \ell \in G' \tag{3.6c}$$

$$\sum_{\ell\in G'} x_{\ell,k} = 1, \forall k \in D \tag{3.6d}$$

$$\sum_{\ell\in G'}\Big(y_\ell\beta w_{\ell,k} - x_{\ell,k}p_{\ell,k}\Big) \leq -\beta(w_{c,k} + \sigma^2), \forall k \in D. \tag{3.6e}$$

3.2.3 Problem Solutions

In the static scenario, we do not need to consider the arrivals and departures of devices and the arrivals of requests. Therefore, an optimal solution can be obtained by solving the ILP formulated in Section 3.2.2. However, when there is a large problem size, it is very time consuming to obtain an optimal solution even with

the problem reformulation in (3.6). On the other hand, in the dynamic scenario, the D2D pairing is conducted at the beginning of each period. In case that not all requests are fulfilled in the current round, the unpaired requesting devices will be queued to be considered for the next round. As seen, the pairing result not only affects the offloading efficiency in the current round but also influences the pairing subsequently. Since the arrivals and departures of devices are unknown in advance, it is impossible to derive an optimal pairing solution for the entire service process. Based on the above considerations, we would develop some lightweight D2D pairing fast algorithms that run period by period.

Intuitively, we could use a random algorithm, which chooses a requesting device randomly and pairs it with an arbitrary feasible cache device. In the random algorithm, there is only feasibility check. To improve the pairing efficiency, more information should be taken into account. Particularly, the distance and channel state of each potential pair of D2D devices are two important factors to be considered in the pairing. The channel state between two D2D devices, *i.e.*, the requesting device and the cache device, directly decides the maximum achievable data rate at the current period. Though the distance between the paired devices does not directly determine the data rate, a shorter distance generally results in a higher data rate on average. Therefore, distance can also be a useful factor if instantaneous channel state information is not available or hard to collect. In contrast, it is easier to derive the distance by obtaining each device's location via a localization technique based on signal strength, time-of-arrival or angle-of-arrival measurements with nearby nodes [23, 24], or through a GPS receiver which becomes increasingly ubiquitous in mobile devices. Furthermore, the distribution of the paired D2D devices will affect the remaining pairing progress, which makes the distance some important guiding information for the D2D pairing. In addi-

tion, the arrival time of each content request also needs to be taken into account since the queueing order of requests affects the fulfilling latency. For instance, we can deal with the arriving requests in different manners, *e.g.*, first-come first-served (FCFS) or last-come first-served (LCFS). Here, we consider LCFS ordering due to its advantage in terms of the latency performance, which will be shown in Section 3.3.

Based on the above discussions, we focus on two D2D pairing algorithms, *i.e.*, a heuristic channel-aware algorithm given in Alg. 1 and a minimum distance-based algorithm given in Alg. 2. Either algorithm is invoked at the beginning of each period to conduct D2D pairing for buffered and newly arrived requests.The general procedure of the two algorithms is similar, except that Alg. 1 is based on the channel state information whereas Alg. 2 uses the distance information. Taking Alg. 1 as an example, we explain the basic steps in the following. First, the requesting device with the most recently arrived request is chosen. Then, the algorithm identifies the set of candidate cache devices, which store the requested file and meet the minimum data rate requirement $r_{\min}$. After that, each candidate cache device is checked to see whether adding the new D2D pair will severely interfere with the existing D2D transmission such that some D2D pair may fail to satisfy the rate requirement $r_{\min}$. Finally, the current requesting device is paired to the candidate cache device which achieves the highest data rate and does not cause the above negative impact to existing D2D transmission.

3.3 Simulation Results and Discussions

In this section, we present simulation results to evaluate the performance of different algorithms for the D2D pair problem. Here, we focus on three metrics,

Algorithm 1 A heuristic channel-aware D2D pairing algorithm.

Input: S (set of cache devices), D (set of requesting devices),
Q (set of requests), φ (current pairing), $r_{\min}$

Output: φ

$\mathbb{D} \leftarrow$ Reordered requesting devices D in a descending order of arrival time for requests in Q
for $d_k \in \mathbb{D}$ **do**
    $S_k \leftarrow \varnothing$ // Feasible cache devices as candidates
    $R_k \leftarrow \varnothing$ // Achievable data rates from feasible caches
    for $s_j \in S$ **do**
        if $r_{j,k} = B\log(1+\xi_k) \geq r_{\min}$ and $m_k^t = m_j^r$ **then**
            $S_k \leftarrow S_k \cup \{s_j\}$
            $R_k \leftarrow R_k \cup \{r_{j,k}\}$
        end if
    end for
    $\mathbb{S}_k \leftarrow$ Reordered cache devices in an ascending order of R_k
    for $s_j \in \mathbb{S}_k$ **do**
        $invalid \leftarrow false$
        for $\{s_{j'}, d_{k'}\} \in \varphi$ **do**
            // Adding the new pair will disable an existing
            // D2D pair in transmission
            if $r_{k'}(\varphi \cup \{s_j, d_k\}) < r_{\min}$ **then**
                $invalid \leftarrow true$
                break
            end if
        end for
        if $invalid = false$ **then**
            $\varphi \leftarrow \varphi \cup \{s_j, d_k\}$ // Add a new D2D pair
        end if
    end for
end for
return φ

Algorithm 2 A minimum distance-based D2D pairing algorithm.

Input: S (set of cache devices), D (set of requesting devices),
Q (set of requests), φ (current pairing), $r_{\min}$

Output: φ

$\mathbb{D} \leftarrow$ Reordered requesting devices D in a descending order of arrival time for requests in Q
for $d_k \in \mathbb{D}$ **do**
    $S_k \leftarrow \varnothing$ // Feasible cache devices as candidates
    $L_k \leftarrow \varnothing$ // Distance between d_k and feasible caches
    for $s_j \in S$ **do**
        if $r_{j,k} = B\log(1+\xi_k) \geq r_{\min}$ and $m_k^t = m_j^r$ **then**
            $S_k \leftarrow S_k \cup \{s_j\}$
            $L_k \leftarrow L_k \cup \{d_{j,k}\}$
        end if
    end for
    $\mathbb{S}_k \leftarrow$ Reordered cache devices in a descending order of L_k
    for $s_j \in \mathbb{S}_k$ **do**
        $invalid \leftarrow false$
        for $\{s_{j'}, d_{k'}\} \in \varphi$ **do**
            // Adding the new pair will disable an existing
            // D2D pair in transmission
            if $r_{k'}(\varphi \cup \{s_j, d_k\}) < r_{\min}$ **then**
                $invalid \leftarrow true$
                break
            end if
        end for
        if $invalid = false$ **then**
            $\varphi \leftarrow \varphi \cup \{s_j, d_k\}$ // Add a new D2D pair
        end if
    end for
end for
return φ

i.e., total number of served D2D pairs and their average latency and distance. We consider the heuristic channel-aware algorithm in Alg. 1 and the minimum distance-based pairing algorithm in Alg. 2 (referred to as the min-distance algorithm for convenience), as well as the optimal pairing solution and a random pairing algorithm. The random algorithm chooses a requesting device randomly and pairs it with an arbitrary feasible cache device. In the following, we first show the results in the static scenario, where the cache devices, requesting devices and content requests are fixed. Then, we study the pairing performance in the dynamic scenario as defined in Section 3.1, with arrivals and departures of requesting and cache devices and content requests.

3.3.1 Static Scenario

We fist evaluate the performance of the pairing algorithms in the static scenario. The Zipf exponent for caching (γ_c) is set to 0.7, and the Zipf exponent for requesting (γ_r) is set to 0.9. Table 3.1 lists other important parameters, which are selected by referring to existing works such as [25]. The results are based on the average of 10^4 runs.

Fig. 3.5 shows the satisfaction ratio of different D2D pairing algorithms with a varying number of cache devices. Here, we vary the number of cache devices between 3 and 13 and fix the number of requesting devices to 10, since it is time-consuming to obtain the optimal solution to (3.6) when the problem size is large. As seen, the ratio of satisfied requesting devices increases fast with the number of cache devices. This is because the cache devices provide more resources. In addition, it is found that the performance of the channel-aware algorithm is fairly close to that of the optimal solution, and slightly better than that of the min-distance algorithm when there are more cache devices available. According to

Table 3.1: Simulation parameters.

Symbol	Definition	Values
m	Library size	10
Φ	Radius of cell	300 m
P_d	Transmit power of D2D device	100 mW
B	Carrier bandwidth	10^6 MHz
α	Path loss exponent	4
β	SINR decoding threshold	-0.6 dB
σ^2	Noise variance	-147 dBm
γ_c	Zipf exponent for cached messages	0.7
γ_r	Zipf exponent for requested mesages	0.9
λ_r	Arrival rate of requesting devices	5
λ_c	Arrival rate of cache devices	1
λ_q	Arrival rate of requests	0.5
μ_r	Departure rate of requesting devices	$\frac{1}{60}$
μ_c	Departure rate of cache devices	$\frac{1}{60}$

the channel model in (3.1), the received signal is subject to log-distance path loss and Rayleigh fading. As a result, the received signal strength on average is higher with a shorter distance. Hence, the channel-aware algorithm and the min-distance algorithm present close performance. Moreover, it is seen that the difference between the optimal solution and the channel-aware algorithm becomes larger when the number of cache devices increases. That indicates that the optimal strategy is able to explore better pairing choices when there are more options. In contrast, the channel-aware algorithm or the distance-based algorithm make pairing decisions in an incrementally greedy manner. We will further compare the channel-aware algorithm and the minimum distance-based algorithm with the random algorithm in the dynamic scenario with a more reasonable number of cache devices.

Fig. 3.6 shows the variation of the satisfaction ratio of different D2D pairing algo-

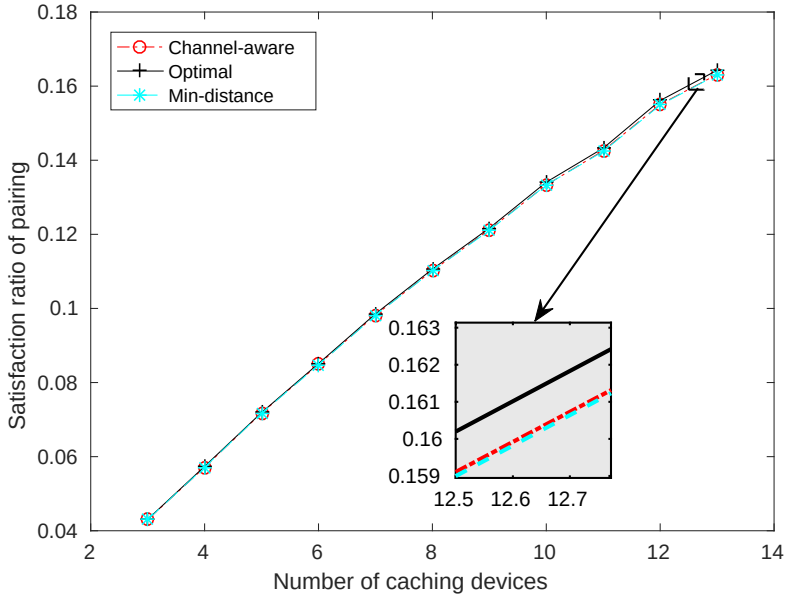


Fig. 3.5: Satisfaction ratio of pairing with a varying number of cache devices.

rithms with the number of requesting devices. We can observe a trend opposite to that in Fig. 3.5. That is, a smaller proportion of requesting devices can be successfully paired to feasible cache devices since the resource competition becomes intense. Besides, the strength of the optimal solution is more evident when there are more requesting devices, which result in a higher volume of content requests.

3.3.2 Dynamic Scenario with Fixed Setting

After evaluating the performance of the D2D pairing algorithms in the static scenario, we further compare their performance in the dynamic scenarios in this section. It is mentioned earlier that we are unable to derive an optimal pairing solution for the dynamic scenario since it is impossible for the BS to know the arrivals and departures of devices and requests a priori. The BS only knows the locations of current devices and the channel states between them. Since it is not possible to predict the arrivals and departures of the devices, it is hard to

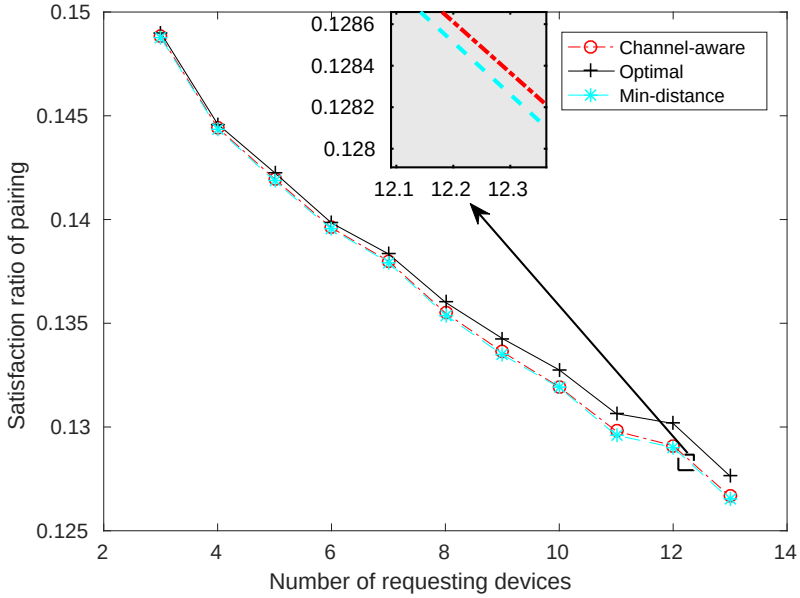


Fig. 3.6: Satisfaction ratio of pairing with a varying number of requesting devices.

determine an optimal pairing strategy that maximizes the overall pairing efficiency instead of focusing on the current period. Hence, when the devices move into and leave the cell dynamically, the D2D pairing is conducted in a periodical manner as illustrated in Fig. 3.3. Here, we only compare the channel-aware algorithm in Alg. 1 and the minimum distance-based algorithm in Alg. 2 with the random algorithm, which is explained in Section 3.2. The random algorithm first randomly chooses a requesting device from the waiting queue, and assigns an arbitrary feasible cache device that is still available (unpaired yet). It continues this process until all requesting devices are satisfied or no feasible cache device is available.

3.3.2.1 Number of served D2D pairs and latency

In this section, we consider that each single run consists of 300 pairing periods. Similar to the static scenario, the Zipf exponents for caching and requesting are set to 0.7 and 0.9, respectively. In addition, we fix the arrival rate of requesting

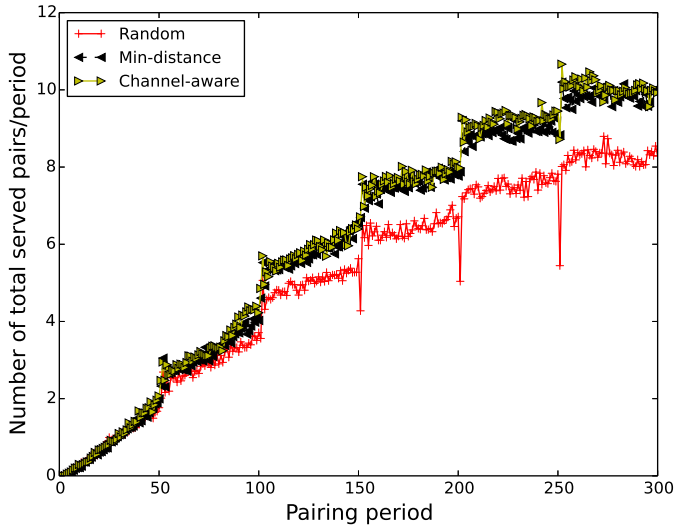


Fig. 3.7: Average number of served pairs per time slot.

devices at $\lambda_r = 5/s$, and the arrival rate of cache devices at $\lambda_c = 1/s$. The arrival rate of requests from each requesting device is set to $\lambda_q = 0.5/s$, which means that a requesting device waits for $2s$ on average after a completed transmission before sending a new request. The average staying time of both cache devices and requesting devices is set to $60s$. Besides, we set the file size to 323399 bits such that the transmission of each file can finish within two time slots. Other parameters are the same as those listed in Table 3.1.

Fig. 3.7 shows the evolution of the number of served pairs in a single run. As we can see in Fig. 3.7, there are very few pairs served at the beginning since the system starts with zero requesting device and zero cache device. Then, the number of D2D pairs increases with time when more caching and requesting devices come into the cell, which form more D2D pairs. As the simulation updates the D2D channel states every 50 time slots, some ongoing D2D links may fail to satisfy the transmission rate requirement and thus are removed from the system. Also, it is observed that the channel-aware algorithm achieves the best performance

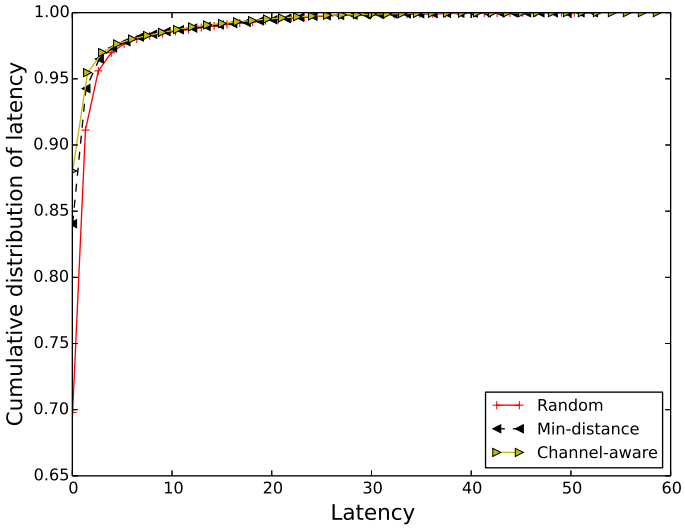
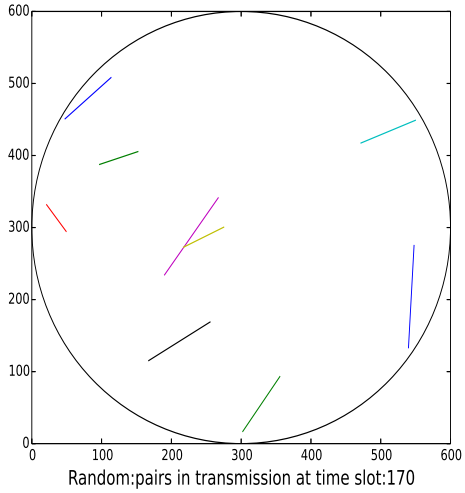


Fig. 3.8: CDF of latencies of served pairs in a single run.

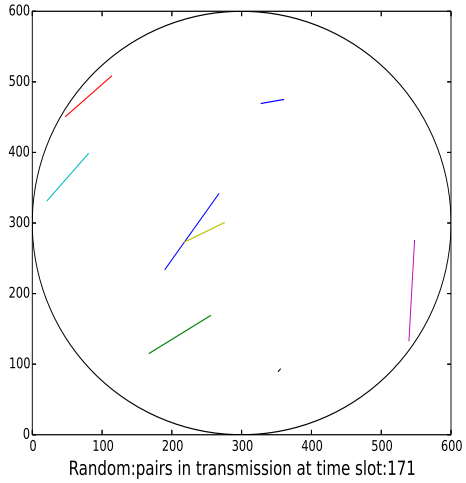
in terms of the number of served D2D pairs per time slot. Further, we run the simulation with the parameters given above for 200 times. Fig. 3.8 shows that the cumulative probability distribution (CDF) of the latencies of the served D2D pairs. It is found that the largest latency of both the channel-aware algorithm and the min-distance algorithm is around 30 time slots.

3.3.2.2 Spatial distribution of D2D pairs

To further examine the differences among these D2D pairing algorithms, we show the spatial distribution of the D2D pairs resulting from different pairing algorithms at certain time. Fig. 3.9 , Fig. 3.10, and Fig. 3.11 show the spatial distribution of the D2D pairs at the 170th and 171th time slots, which result from the random algorithm, the min-distance algorithm, and the channel-aware algorithm, respectively. For comparison fairness, the three sets of results are based on the same set of simulation data regarding the arrivals and departures of devices and the requests. That means that the same sets of cache devices and requesting devices

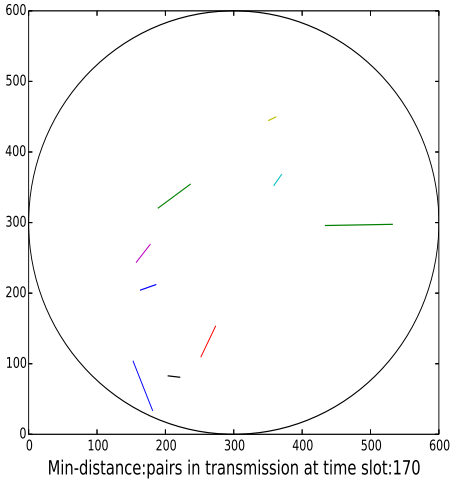


(a) At the 170th time slot.

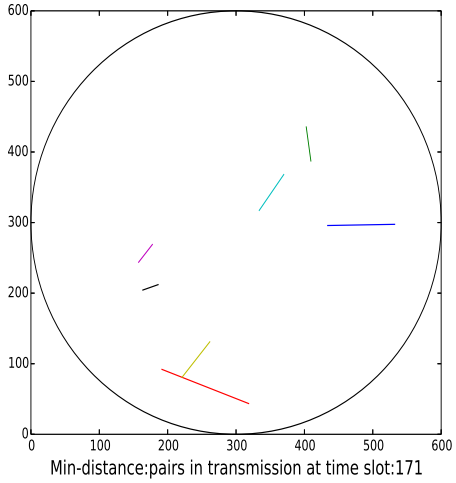


(b) At the 171th time slot.

Fig. 3.9: Spatial distribution of D2D pairs in transmission in the cell with the random algorithm in two slots.

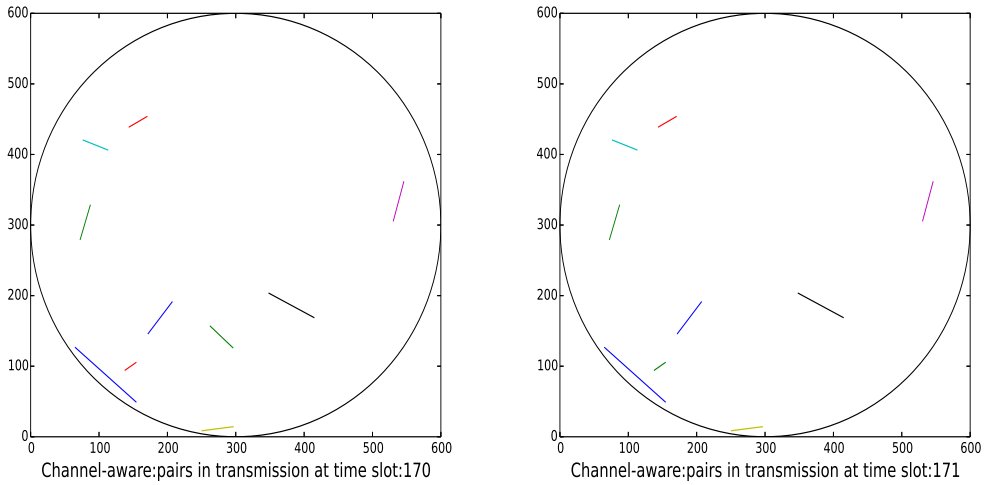


(a) At the 170th time slot.



(b) At the 171th time slot.

Fig. 3.10: Spatial distribution of D2D pairs in transmission in the cell with the min-distance algorithm in two slots.



(a) At the 170th time slot. (b) At the 171th time slot.

Fig. 3.11: Spatial distribution of D2D pairs in transmission in the cell with the channel-aware algorithm in two slots.

appear in the same positions of the cell in each time slot. Meanwhile, these cache devices and requesting devices leave the cell at certain fixed time according to the simulation data set. The results are based on a single run for comparison clarity. By comparing the result in Fig. 3.9 and that in Fig. 3.10, it is obvious that the min-distance algorithm indeed intends to choose the pairs with the closest distance, while the random algorithm exhibits a rather random trend. It is easy to understand this since the min-distance is designed to first consider the feasible cache device closest to a requesting device. On the other hand, comparing Fig. 3.9 with Fig. 3.11, we can see that the channel-aware algorithm also leads to D2D pairs that are closer, which looks similar to the result of the random algorithm. This is because the channel-aware algorithm tends to pair a requesting device with the cache devices of the best channel state, which generally implies a shorter distance.

Referring to Fig. 3.10 and Fig. 3.11, we cannot see clearly the advantages of

either algorithm just by comparing the spatial distribution of the D2D pairs in transmission. However, we believe that channel-aware algorithm performs better than the min-distance algorithm. The channel-aware algorithm uses the channel state as the main factor for pairing, while the min-distance algorithm is based on distance. A shorter distance cannot always ensure a higher transmission rate due to the randomness of the wireless channel. Indeed, the spatial distributions of the transmitting pairs look similar with the channel-aware algorithm and the min-distance algorithm, and both allow more D2D pairs compacted within the cell than with the random algorithm. However, the average transmission rate of the D2D pairs with the channel-aware algorithm might be higher than that of the min-distance algorithm, which will be further investigated in the following experiments.

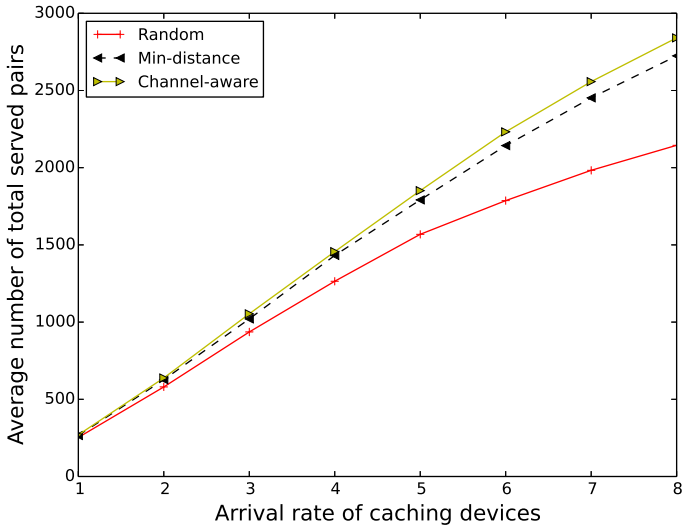
3.3.3 Dynamic Scenario with Different Settings of Cache Devices

As shown in Fig. 3.5 and Fig. 3.6 in Section 3.3, the difference between D2D pairing algorithms is potentially larger when the system resources are more abundant. Hence, in this section, we further evaluate the performance of the three pairing algorithms in the system with varying resources in a wider range. Due to the computational complexity for obtaining the optimal solution to the D2D pairing problem in (3.6), we focus on the three heuristic pairing algorithms, *i.e.*, the random algorithm, the min-distance algorithm, and the channel-aware algorithm. Similar to the experiments discussed in Section 3.3.2, we focus on a cell coverage of a radius $\Phi = 300m$. In this section, we fix the settings for the requesting devices and vary those of the cache devices to examine the effects of these varying settings with different pairing algorithms. In particular, we set the Zipf exponent

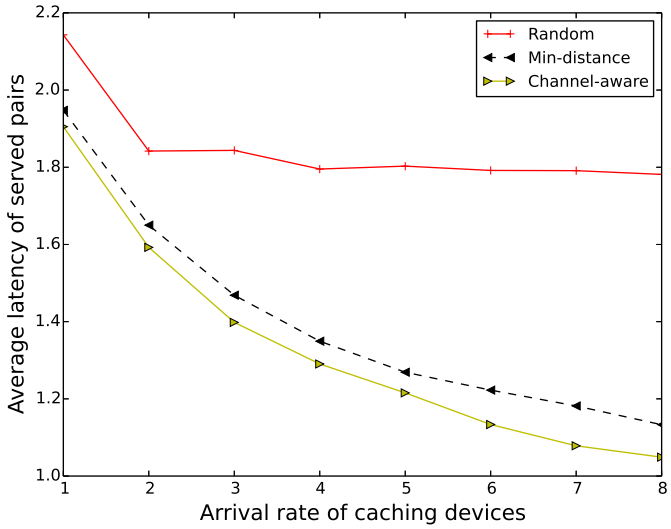
of requesting devices to 0.9, and their mean arrival rate to $\lambda_r = 1/s$. The average idling time before a new request is set to $2s$, which corresponds to the setting of $\lambda_q = 0.5/s$. Besides, the average staying time of requesting devices in the cell is set to $60s$. That is the duration that a moving user takes to drive at a speed 36 km/h from the center of the cell to the cell boundary. In the following, we vary the settings for the cache devices and show the performance results in terms of the total number of served D2D pairs, the average latency of served pairs, and the average distance of served pairs.

3.3.3.1 Varying arrival rate of cache devices

First, we vary the mean arrival rate of cache devices λ_c between 1 and 8 per second and show the performance of different pairing algorithms in Fig. 3.12. As seen in Fig. 3.12(a), more D2D pairs are successfully matched when the arrival rate of cache devices increases. Obviously, this is because more resources become available at the cache devices to fulfill more requests. In addition, it is observed that the channel-aware algorithm outperforms both the random algorithm and the min-distance algorithm. On the other hand, Fig. 3.12(b) demonstrates that the channel-aware algorithm achieves a lower latency to serve the content requests. Here, the latency of each served pair includes the time that the requesting device spends in the waiting queue as well as the file transmission time with the paired cache device. Clearly, the distance between the D2D pair has an important impact on the data rate and consequently the transmission time. As seen in Fig. 3.12(c), the average distance between D2D pairs is generally decreasing when more cache devices become available. This shorter distance potentially reduces the transmission time. Meanwhile, since there are more candidate cache devices, the waiting time of the requesting devices is also shorter on average.

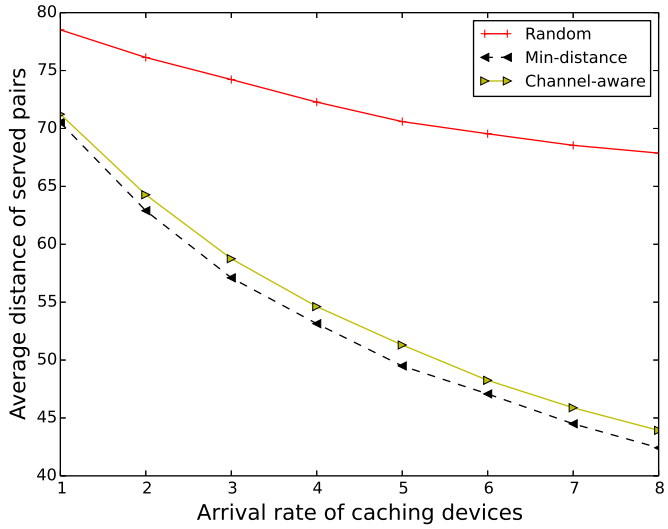


(a)



(b)

Fig. 3.12: Pairing performance with a varying arrival rate of cache devices.



(c)

Fig. 3.12: Pairing performance with a varying arrival rate of cache devices.

Comparing the three D2D pairing algorithms in Fig. 3.12, we can see that the channel-aware algorithm achieves the highest performance with all different arrival rates of cache devices. The performance of the min-distance algorithm is fairly close to that the channel-aware algorithm, while the random algorithm is the worst among the three options. In addition, it is observed that the performance gap among the three algorithms increases with a larger arrival rate of cache devices. This can be explained intuitively as follows. When the arrival rate of cache devices is very small, there are only a few cache devices in the cell to satisfy the content requests of requesting devices. The pairing results of different algorithms can be very similar since there are not many options. As a result, many unserved requests have to be queued at the end of the pairing period. On the other hand, when there are more pairing options with the increased arrival rate of cache devices, different pairing algorithms result in quite distinct D2D pairs for each period. The differences are accumulated period by period and thereby the performance gap

becomes more evident eventually.

Examining Fig. 3.12(c) more closely, we can see that the min-distance algorithm indeed results in the lowest average distance of served D2D pairs as opposed to the channel-aware algorithm and the random algorithm. This validates our implementation since the min-distance algorithm preferably pairs the caching and requesting devices of the closest distance. In addition, it is observed in Fig. 3.12(c) that the random algorithm pairs D2D devices of the largest distance, which generally leads to the lowest transmission rate between the D2D devices. As we know, the latency of each served pair includes the time that the requesting device spends in the waiting queue as well as the transmission time for the requesting device to receive the content file. As a consequence, the random algorithm exhibits the worst latency performance as seen in Fig. 3.12(b).

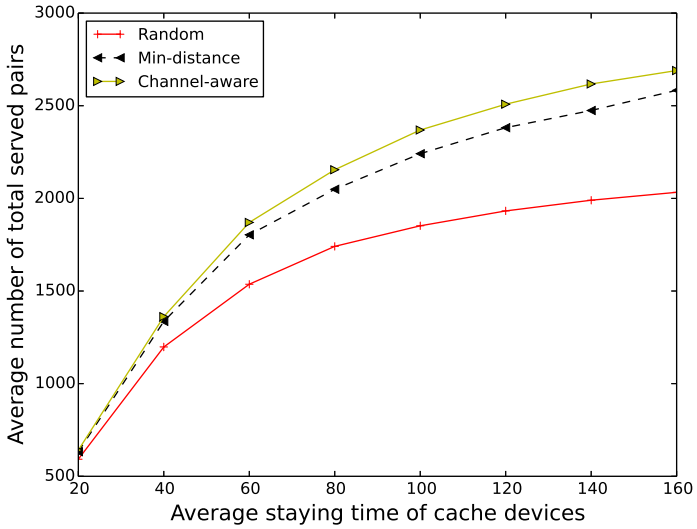
Another clear trend can be found in Fig. 3.12 regarding the channel-aware algorithm and the min-distance algorithm. That is, they perform closely in terms of the total number of served pairs and the average latency of served pairs, although the channel-aware algorithm outperforms the min-distance algorithm in a strict sense. Fig. 3.12 shows the average performance of 200 runs with each runs consisting of 300 pairing periods. As we know, the channel state between two devices is generally better for a shorter distance due to the averaging effect. In other words, since the channel-aware algorithm pairs the requesting device with the cache device of the best channel state, the distance between the requesting device and the cache device also tends to be minimal on average. This is confirmed by the result in Fig. 3.12(c) with respect to the average distance between served pairs. Therefore, the channel-aware algorithm and the min-distance algorithm achieve similar transmission rate and latency performance in an average sense. However, the difference between the two algorithms can be quite large for

an arbitrary running case.

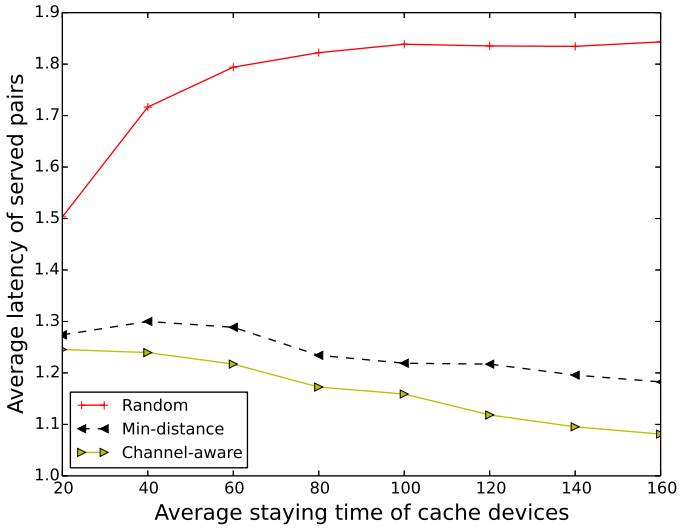
3.3.3.2 Varying staying time of cache devices

As shown in Fig. 3.12, when the arrival rate of cache devices increases, the D2D pairing algorithms can serve more requesting/cache device pairs and achieve lower service latency. In fact, in addition to the arrival rate of cache devices, the average staying time of cache devices in the cell is another important factor that affects the abundance of resources. Basically, the longer the average staying time of cache devices, the more resources available in the cell. In this section, we further examine the performance of the three pairing algorithms with varying staying time of cache devices. Similar to Section 3.3.3.1, we fix the parameters of the requesting devices, but vary the average staying time of cache devices between $20s$ and $120s$. That is, μ_c ranges between $\frac{1}{120}/s$ and $\frac{1}{20}/s$. Fig. 3.13 presents the pairing performance of the three pairing algorithms in terms of the total number of served D2D pairs, their average latency and average distance.

Similar to Fig. 3.12, Fig. 3.13 shows that the channel-aware pairing algorithm achieves the best performance in terms of the number of served pairs and the average latency of served pairs. The channel-aware pairing algorithm significantly outperforms the random algorithm, but performs closely to the min-distance algorithm in some cases. Although a good channel often exists between a pair of close devices and provides a high transmission rate, it is not always as such due to the randomness over the wireless channel. As a result, the channel-aware algorithm offers better performance in terms of average transmission rate and service latency. Correspondingly, the higher fulfilling rate toward requesting devices in each pairing period further improves the system throughput and thus results in more served pairs in total.

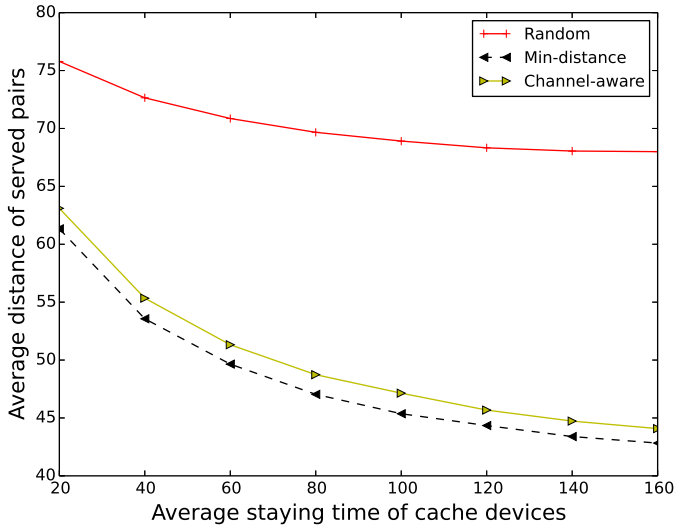


(a)



(b)

Fig. 3.13: Pairing performance with a varying staying time of cache devices in the cell.



(c)

Fig. 3.13: Pairing performance with a varying staying time of cache devices in the cell.

Looking at Fig. 3.13(a) more closely, we can see that more D2D pairs are served with the increase of the average staying time of cache devices. Intuitively, this makes sense since a larger average staying time implies that each cache device stays within the cell for a longer duration. Thereby, more caching resources are available to serve the content requests arising from the cell. Moreover, the performance gap among the three pairing algorithms becomes larger when the average staying time of cache devices increases. Similar to Fig. 3.12(a), this is because the existence of more feasible cache devices provides more pairing options to requesting devices. As the pairing algorithms explore the larger manipulation space in different ways, the differences in their pairing devices are reflected more evidently in the result. Regarding the average service latency of served pairs, Fig. 3.13(b) shows that the three pairing algorithms exhibit different trends with the varying staying time of cache devices. First, the average latency of the channel-aware algorithm decreases almost linearly with the average staying time. This decrease can be attributed

to the fact that more abundant caching options are available in the cell with a longer staying time of cache devices. As a result, it is more likely that a requesting device is paired with a cache device of a better channel state. This is confirmed by the observation in Fig. 3.13(c) that the average distance of served pairs becomes shorter with the increase of staying time. A shorter distance often implies a better channel state on average, and thus a shorter average latency. Moreover, comparing Fig. 3.12(b) and Fig. 3.13(b), we can see that, regarding the service latency, the channel-aware algorithm is more sensitive to the mean arrival rate of cache devices than the average staying time.

On the other hand, the min-distance algorithm experiences a slight increase in latency when the average staying time of cache devices increases from $20s$ to $40s$. Beyond that, the latency of the min-distance algorithm decreases with the increase of staying time as the channel-aware algorithm does. For this range, the reason is similar to the above explanation. That is, when the staying time of cache devices is increasing, the average distance of the paired requesting and cache devices becomes shorter, which leads to a generally better channel state and higher transmission rate. As a result, the average latency is reduced accordingly. By contrast, we can interpret the slight increase of latency at the beginning as follows. When the average staying time of cache devices is less than $40s$, which is less than the average staying time of requesting devices $60s$, it is more likely that some paired cache devices may leave the cell before the file transmission completes. Such interrupted transmissions result is lower service latency on average.

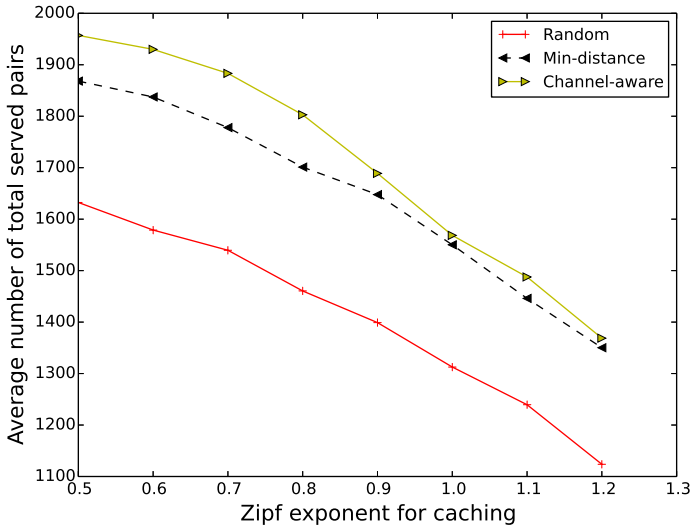
In addition, Fig. 3.13(b) shows that the average latency of the random algorithm exhibits an opposite trend in terms of the average latency. As seen in Fig. 3.13(c), for the random algorithm, the average distance of served pairs also decreases with a longer staying time of cache devices. Nonetheless, the increase of transmission

rate with the shorter distance is quite minor. When the average staying time of cache devices is less than $60s$, there is a higher likelihood of transmission interruption by the early departures of cache devices. As a result, the average latency increases in this range. When the average staying time of cache devices exceeds $60s$, we can see that the average latency has little variation. This is because the slight increase in transmission rate is not sufficient enough to offset the increasing transmission demands since there are much fewer premature departures with interrupted transmissions.

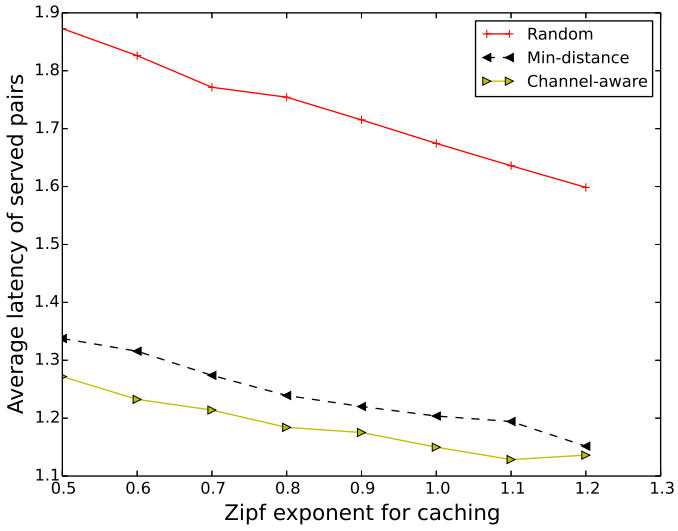
3.3.3.3 Different Zipf exponents of cache devices

As described in Section 3.1, all cache devices store the content files from the library according to a Zipf distribution. In the Zipf distribution, the exponent γ_c is very important. A larger value of γ_c implies that the files cached at the devices are concentrated on fewer files. We can control this parameter to simulate different cases with respect to the popularity of the files. Hence, in this section, we vary the Zipf exponent for caching between 0.5 and 1.2. The mean arrival rate of cache devices is set to $\lambda_c = 5/s$, while the average staying time of cache devices is set to $1/\mu_c = 60s$. The parameters for the requesting devices are also fixed as given at the beginning of Section 3.3.3. In particular, the Zipf exponent γ_r for requesting is set to 0.9. As before, we evaluate the performance of three D2D pairing algorithms under such a setting.

Fig. 3.14 shows the variations of the three metrics with the Zipf exponent for caching. As seen in Fig. 3.14(a), for all the three pairing algorithms, the total number of served pairs decreases with the Zipf exponent for caching. Similarly, the channel-aware algorithm achieves the best performance, while the random algorithm performs the worst. When the Zipf exponent for caching increases, the

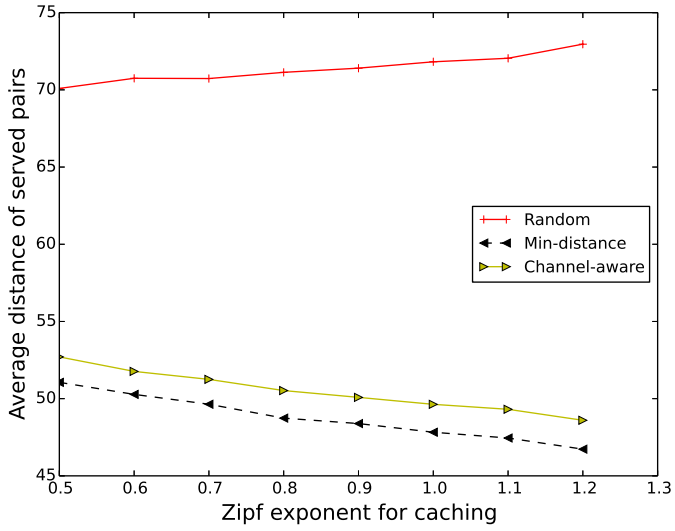


(a)



(b)

Fig. 3.14: Pairing performance with a varying Zipf exponent for caching.



(c)

Fig. 3.14: Pairing performance with a varying Zipf exponent for caching.

files cached at the devices are more overlapped. In the mean time, the popularity of the requested files remains the same. As a consequence, more requesting devices may demand some content files that are not cached at the cache devices, which makes it even harder to satisfy these requests. Therefore, the total number of successful pairs decreases accordingly. Another important observation in Fig. 3.14(a) is that the gap between the channel-aware algorithm and the min-distance algorithm becomes larger with a smaller Zipf exponent for caching. This observation can be explained as follows. When the cached files are concentrated on more files, potentially, there are more candidate cache devices that can fulfill the requests. As a result, there is a larger search space for the pairing decisions. The channel-aware algorithm determines the pairing result that can serve the request device in a higher transmission rate. Thus, the resources at the corresponding cache device can be released faster and serve other requests. Therefore, the channel-aware algorithm approaches a larger performance gain over the min-distance algorithm.

Fig. 3.14(b) shows the average latency decreases with a larger Zipf exponent for caching. This is mainly because there are fewer served pairs when the Zipf exponent for caching increases, which can be seen in Fig. 3.14(a). When fewer D2D pairs are transmitting at the same time, the interference among them is generally more tolerable and higher data rates are achievable. This can be further confirmed by the smaller D2D distances shown in Fig. 3.14(c) for the channel-aware algorithm and the min-distance algorithm, although the average distance slightly increases with the random algorithm. Consequently, the average latency is shorter when fewer D2D pairs are served simultaneously.

It is also worth noticing in Fig. 3.14(b) that there exists a slight increase of average latency with the channel-aware algorithm when the Zipf exponent for caching is changed from 1.1 to 1.2. Recall that the Zipf exponent for requests is 0.9. That means, when the Zipf exponent for caching exceeds 0.9, it is more likely that the cache devices do not store some unpopular files that may be requested by the requesting devices. In other words, it becomes even harder for a requesting device to be matched to a feasible cache device. If a larger number of such requests have to be accommodated, it will inevitably lead to a longer latency. As seen in Fig. 3.14(a), the channel-aware algorithm pairs a largest number of D2D devices and thereby incorporates more pairs that experience longer latencies. Therefore, the average latency of the channel-aware algorithm is slightly higher when the Zipf exponent for caching increases from 1.1 to 1.2.

The result in Fig. 3.14 also demonstrates that a good caching model is essential such that the advantages of a good pairing strategy can be taken into full play. For instance, if all cache devices in the cell kept the same file that is unwanted most, then most of the requests could not be satisfied. On the contrary, if all cache devices stored the file that is most popular for the requesting devices, a majority

of the requests could be fulfilled. Though these are two extreme situations, we can learn from the result that an effective caching mechanism should adapt the selection of cached files according to the popularity of these files in the requests. A good pairing algorithm can improve its performance when working with an effective caching mechanism.

3.3.4 Dynamic Scenario with Different Settings of Requesting Devices

As found in Section 3.3.3, the channel-aware algorithm always achieves the best performance in terms of the total number of served pairs and the average latency with the different settings for the cache devices in the dynamic scenario. The min-distance algorithm also achieves reasonable performance that is fairly close to that of the channel-aware algorithm in many cases. While the channel-aware algorithm requires the channel state information for each potential D2D link, the min-distance algorithm needs less information and thus involves a smaller overhead. Therefore, the min-distance algorithm can be considered as a good candidate when there is resource limitation for gathering the channel state information.

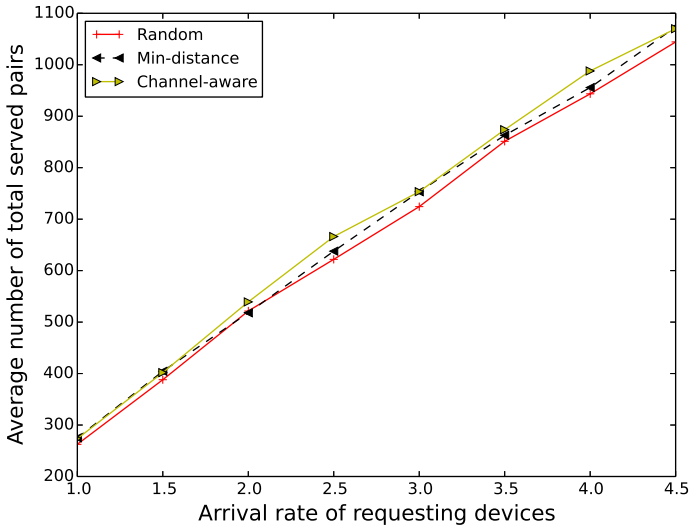
In this section, we examine the performance of different D2D pairing algorithms with the varying parameters of the requesting devices. Here, we fix the mean arrival rate of cache devices to $\lambda_c = 1/s$, the average staying time of cache devices to $1/\mu_c = 60s$, and the Zipf exponent for caching to $\gamma_c = 0.7$. In the following, we run simulations with different settings for the mean arrival rate of requesting devices, their average staying time, and the Zipf exponent for requesting.

3.3.4.1 Varying arrival rate of requesting devices

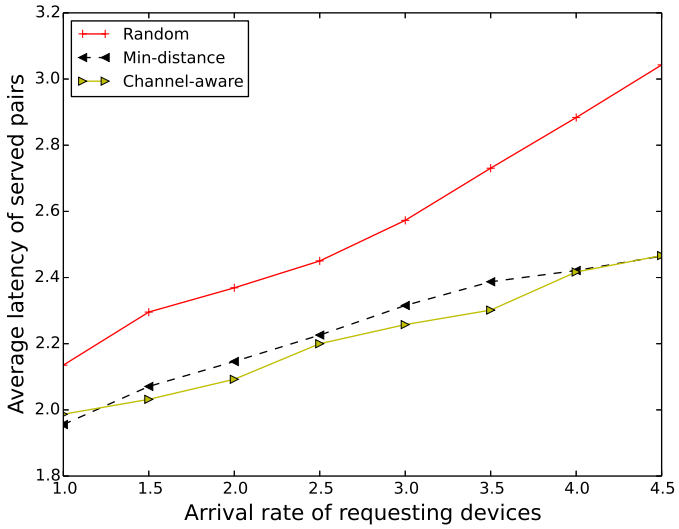
Fig. 3.15 shows the performance of the three D2D pairing algorithms on different metrics when the mean arrival rate of requesting devices λ_r is increased from $1/s$ to $4.5/s$. As seen in Fig. 3.15(a), when the arrival rate is increasing, more D2D pairs are served in total. This is intuitive since more requesting devices generally start more content requests to be served by feasible cache devices. In addition, similar to Fig. 3.12, the channel-aware algorithm achieves the best performance although the results of the three pairing algorithms are much closer.

Fig. 3.15(b) shows that the average latency of served pairs also increases when the mean arrival rate of requesting devices is larger. This is mainly because of the rising demands from the requesting devices that have to be accommodated by the set of cache devices. Since more cache devices have to transmit files at the same time to serve a larger number of requests, the average transmission rate will be lower to tolerate the higher interference among the D2D links. Meanwhile, the increased transmission time further results in more requests buffered in the queue and a generally longer waiting time. Both the increasing transmission time and waiting time lead to the higher service latency. Besides, it is found the latency performance of the channel-aware algorithm and the min-distance algorithm overlaps in some points but their performance is very close.

Fig. 3.15(c) shows the average distance of served pairs. As seen, the average distance with the random algorithm decreases with the increasing of the arrival rate of requesting devices. The random algorithm considers the requesting devices in a random order. Hence, when there are more requesting devices, the distance between the paired requesting device and the cache device would be closer since there are more pairing alternatives. In contrast, the average distance with the channel-aware algorithm and the min-distance algorithm only fluctuates slightly.

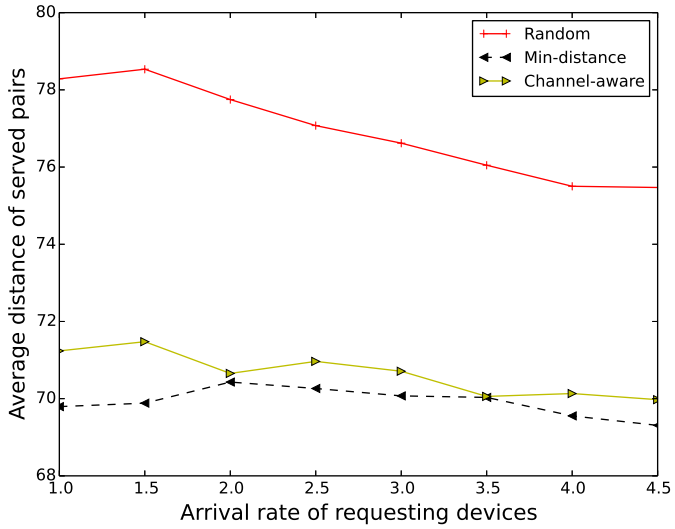


(a)



(b)

Fig. 3.15: Pairing performance with a varying arrival rate of requesting devices.



(c)

Fig. 3.15: Pairing performance with a varying arrival rate of requesting devices.

This is because these two algorithms determine the pairing results by ordering the requesting devices according the arrival time of their requests. Therefore, the results are not directly affected by the locations and channel states of the requesting devices that newly arrive into the cell.

3.3.4.2 Varying staying time of requesting devices

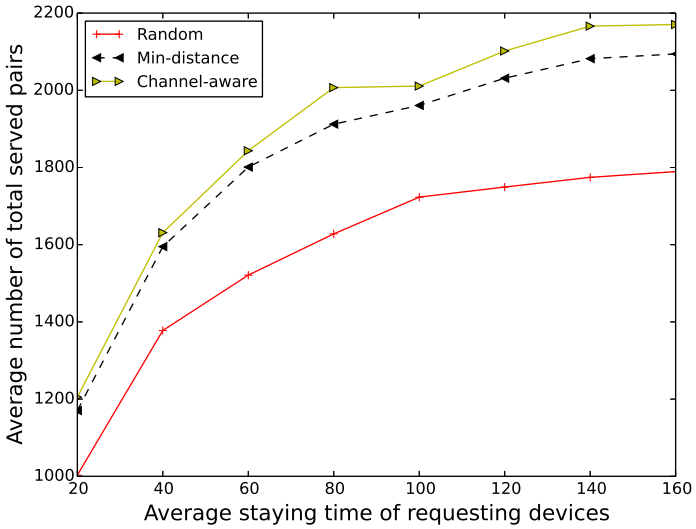
In addition to the arrival rate of requesting devices, their staying time in the cell also influences the amount of demands that arise from these devices. According to the system model described in Section 3.1, as long as a requesting device stays within the cell coverage, it generates a sequence of requests with a random idling time before each new request. Therefore, a longer staying time implies a potentially higher volume of traffic demands. In this section, we set the mean arrival rate of requesting devices to $\lambda_r = 1/s$, and set the Zipf exponent for requesting to $\gamma_r = 0.9$, but change their average staying time from $1/\mu_r = 20s$ to

$1/\mu_r = 120s$. Recall that the average staying time of cache devices is $1/\mu_c = 60s$. Fig. 3.16 shows the performance of three D2D pairing algorithms in terms of the total number of served pairs, their average latency and average distance. Fig. 3.16(a) shows that more D2D pairs are served when the average staying time of requesting devices increases. This is obvious since a longer staying time implies that more content requests are started from the requesting devices. As seen in previous cases, the channel-aware algorithm supports a largest number of D2D pairs, and its gap to other algorithms enlarges when more pairing options exist for more requests.

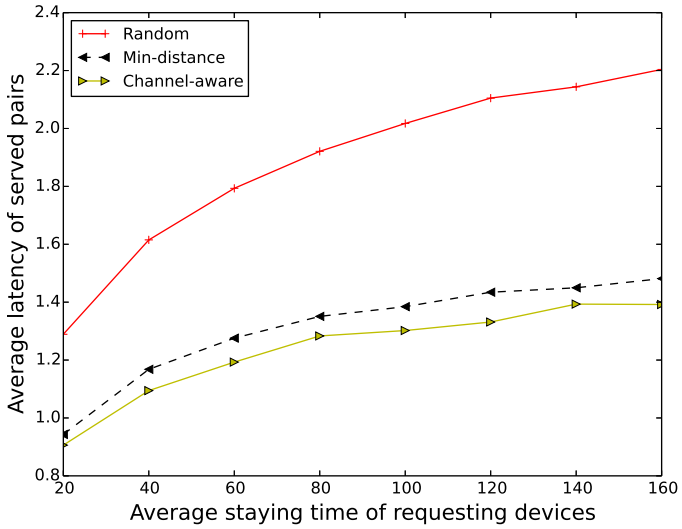
Fig. 3.16(b) shows the increase of average latency when more requests have to be accommodated by the cache devices. As observed in Fig. 3.16(c), the average distance of served pairs is rather stable when the staying time of requesting devices is varying. This observation implies that the average transmission rate of these served pairs does not vary greatly. The increase of average latency is mainly due to the longer waiting time in the queue. When the requesting devices stay in the cell for a longer duration, their competition for the caching resources becomes more intense. As a result, more requests have to be placed in the waiting queue before being served, which thus leads to a longer average latency. Even so, the channel-aware algorithm still achieves the best latency performance.

3.3.4.3 Different Zipf exponents of requesting devices

In the end, we examine the performance of different pairing algorithms when the Zipf exponent for requesting is changing. Here, we fix the settings for the cache devices, as well as most parameters of requesting devices, including $\lambda_r = 1/s$ and $1/\mu_r = 60s$. Then, we vary the Zipf exponent for requesting γ_r from 0.5 to 1.2. Note that the Zipf exponent for caching is set to $\gamma_c = 0.7$.

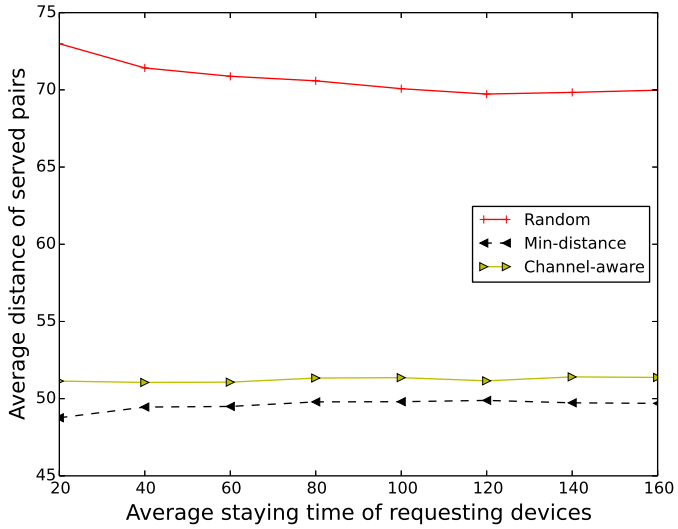


(a)



(b)

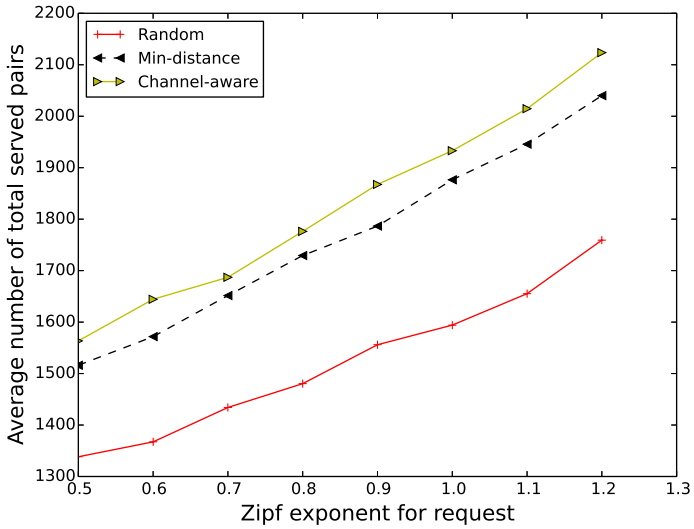
Fig. 3.16: Pairing performance with a varying staying time of requesting devices in the cell.



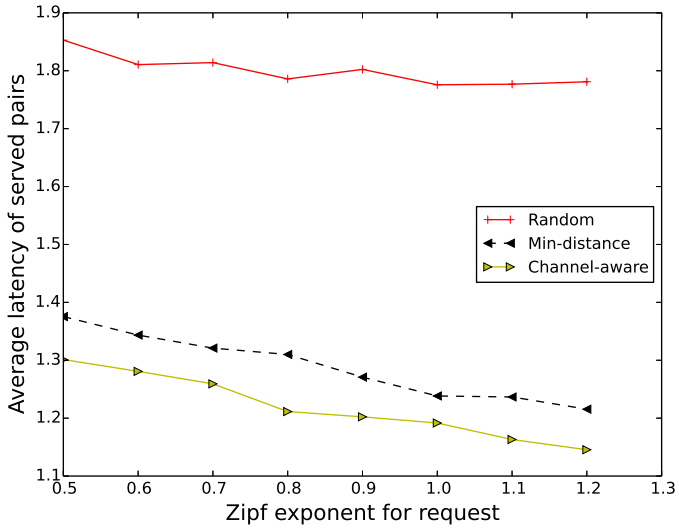
(c)

Fig. 3.16: Pairing performance with a varying staying time of requesting devices in the cell.

Fig. 3.17 shows the pairing results of the channel-aware algorithm, the min-distance algorithm, and the random algorithm. As seen in Fig. 3.17(a), the total number of served pairs increases almost linearly with the Zipf exponent for requesting. This observation can be easily interpreted as follows. While the content files are stored at the cache devices in the same manner, a higher Zipf exponent for requesting implies that the demands from the requesting devices actually target at a smaller number of most popular files. In other words, more requesting devices ask for the files that are cached in most of the cache devices, which will make these requests easier to satisfy. As a result, more requests are successfully served. Based on the findings in Fig. 3.14(a) and Fig. 3.17(a), we can conclude that it is important to relate the Zipf exponent for caching to that of the requests. For sure, it is undesirable that the requested files are only cached in few cache devices. The performance of the pairing algorithms can perform in a best degree only if both Zipf exponents are properly aligned.

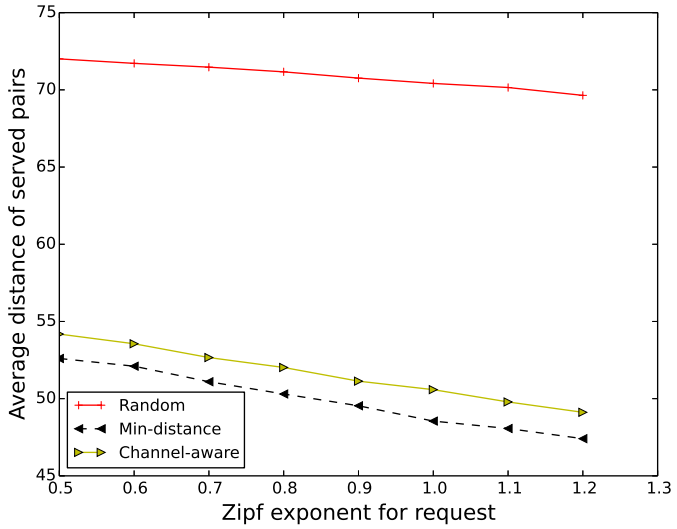


(a)



(b)

Fig. 3.17: Pairing performance with a varying Zipf exponent for requesting.



(c)

Fig. 3.17: Pairing performance with a varying Zipf exponent for requesting.

Fig. 3.17(b) shows that the average latency of the channel-aware algorithm and the min-distance algorithm decreases when the Zipf exponent for requesting is increasing. Though the random algorithm exhibits some slight fluctuations due to the randomness effect, the average latency also goes down in the long run. This decrease in latency is mainly attributed to better pairing results which lead to shorter D2D distances and higher transmission rates. When the Zipf exponent for requesting is larger, more requests are concentrated on a smaller number of most popular files. In such a case, it is likely that more cache devices have stored these files. The availability of more candidate cache devices enable the pairing algorithms to end up with better options with shorter D2D distances. This can be confirmed by the result in Fig. 3.17(c), which shows that the average distance of served pairs decreases with the Zipf exponent for requesting in all three pairing algorithms. The shorter distance generally leads to a higher transmission rate and thus shorter average latency.

Chapter 4

Multicast Message Allocation for D2D Content Distribution

In Chapter 3, we focus on offloading the content distribution traffic from the cellular network to D2D communications. It has been shown that our proposed channel-aware pairing algorithm outperforms a random pairing strategy. To further reduce the energy consumed at cache devices, this chapter studies another important problem for D2D-assisted content distribution, which allocates the message requests to be served by the cache devices via D2D multicast. Aiming to minimize the total transmission cost or maximize the gain in cost saving for the BS, this message allocation problem can be formulated from different perspectives. Here, we introduce three different formulations with corresponding solutions and evaluate them via simulations.

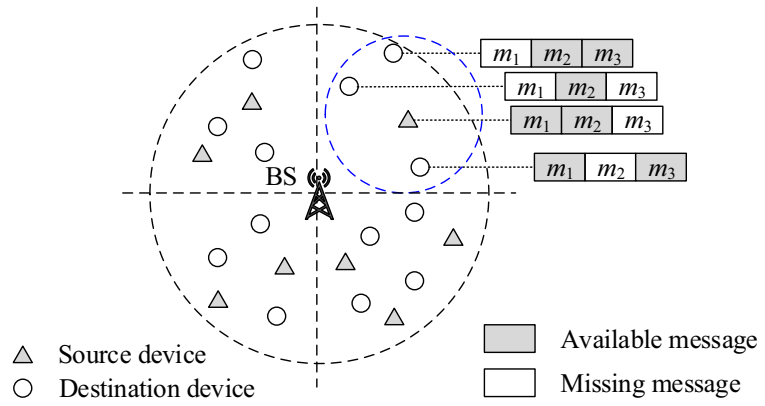


Fig. 4.1: Content distribution scenario with D2D multicast.

4.1 System Model

We consider a content distribution scenario within a cell of radius Φ as depicted in Fig. 4.1. Here, a set of requesting devices, D, are requesting content items (referred to as a "messages") from a "library" of size m, denoted by M. The requesting devices enter cell coverage following a Poisson process of mean rate λ_r, and the requesting devices are uniformly distributed within the cell. It is further assumed that the staying time of the requesting devices follows an exponential distribution with mean $1/\mu_r$. For the same requesting device, the interval between adjacent requests follow an exponential distribution of mean $1/\lambda_q$. In addition, there are another set of devices, S, $|S| = n$, which enters the BS's coverage according to a Poisson process of mean rate λ_c and stays therein for an exponentially distributed time with mean $1/\mu_c$. Each $s_i \in S$ already possesses a subset of messages $M_i \subseteq M$, $|M_i| = m_i$.

If all requests are to be fulfilled by the BS, the BS has to unicast the requested messages and introduce a high cost. Hence, we allow the cache devices to multicast some of their possessed messages to multiple requesting devices that fall within

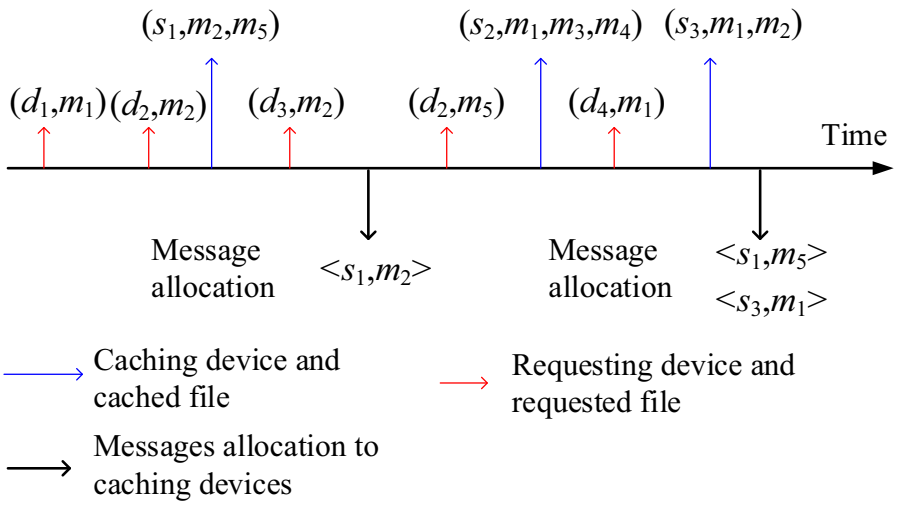


Fig. 4.2: Periodical message allocation.

the transmission range, denoted by L. Then, the goal of the multicast scheme is to satisfy as many requests as possible. As shown in Fig. 4.2, we consider a time-slotted scenario where the content requests are processed periodically. At the end of each period, the requesting and caching information is collected to allocate multicast messages for cache devices at the beginning of the next period. In case that not all requests are fulfilled in the current round, the remaining requests are served by the BS.

4.1.1 Channel Model

As in Section 3.1, we consider a D2D underlaid cellular network, where the D2D links share the uplink spectrum of regular cellular users. Assume that all potential D2D transmitters of the cache devices in S all share the same cellular uplink channel. This cellular channel is preferably unused or allocated to a cellular user that is uniformly located within the cell. Supposing that the request from device $d_j \in D$ is fulfilled by device $s_i \in S$, the received signal at D2D receiver d_j is

written as

$$\begin{aligned} y_j =& \sqrt{P_d} d_{i,j}^{-\frac{\alpha}{2}} h_{i,j} x_i + \sqrt{P_c} d_{c,j}^{-\frac{\alpha}{2}} h_{c,j} x_c \\ &+ \sum_{i' \in S, i' \neq i} \theta_{i'} \sqrt{P_d} d_{i',j}^{-\frac{\alpha}{2}} h_{i',j} x_{i'} + n_j. \end{aligned} \tag{4.1}$$

Here, α is the path-loss exponent, n_j is the additive noise at D2D receiver d_j distributed as $\mathcal{CN}(0, \sigma^2)$. Besides, $\theta_{i'}$ is a binary variable indicating whether transmitter $s'_i \in S$ is selected to multicast to certain requesting devices. For D2D transmitter s_i and the cellular user using the same uplink channel, x_i and x_c are their sent signals, respectively, P_d and P_c are their respective transmit power, $d_{i,j}$ and $d_{c,j}$ are their respective distance to D2D receiver d_j, and $h_{i,j}$ and $h_{c,j}$ are the corresponding distance-independent channel gain that captures the fading effect. Considering Rayleigh fading channels, $|h_{i,j}|^2$ and $|h_{c,j}|^2$ follow an exponential distribution of unit mean. As seen in (4.1), the received signal at D2D receiver d_j includes the expected signal, the interference from the cellular user, the integrated interference from all other active D2D transmitters in S, and the additive noise. The signal-to-interference-plus-noise ratio (SINR) is given by

$$\xi_j = \frac{P_d d_{i,j}^{-\alpha} |h_{i,j}|^2}{P_c d_{c,j}^{-\alpha} |h_{c,j}|^2 + \sum_{i' \in S, i' \neq i} \theta_{i'} P_d d_{i',j}^{-\alpha} |h_{i',j}|^2 + \sigma^2}. \tag{4.2}$$

The achievable data rate at receiver device d_j is obtained as

$$r_j = B \cdot \log_2(1 + \xi_j) \tag{4.3}$$

where B is the carrier bandwidth.

4.1.2 Cost Model

Due to the periodical processing nature, we cannot use the instantaneous channel conditions in multicast message allocation. Instead, we use the average channel conditions to estimate resource costs. Accordingly, at the beginning of each period, the BS allocates the messages that the cache devices need to multicast for surrounding requesting devices.

Given cache device $s_i \in S$ that possesses message $m_k \in M_i$, let D_{ik} denote the set of destination devices that request message m_k and fall within the transmission range of s_i. Assume that the co-channel interference between D2D and cellular users can be approximated by a Gaussian process similar to additive noise with mean I and N, respectively. As the cost of multicast transmission is constrained by that of the *worst* receiver, we estimate the SINR ξ_j at the worst receiver $d_j \in D_{ik}$ by an exponential distribution with mean

$$\overline{\xi}_j = \frac{P_d}{N+I} d_{i,j}^{-\alpha}. \tag{4.4}$$

As in [26], a receiver can successfully decode the received message only when the local SINR is not less than a threshold β. Then, the probability that message m_k is received successfully by all requesting devices in D_{ik} is given by

$$P_{ik} = \mathrm{P}\left[\xi_j \geq \beta\right] = e^{-\beta \frac{(N+I)}{P_d} d_{i,j}^{\alpha}}. \tag{4.5}$$

Assume that the resource cost of cache device s_i is proportional to the required SINR to achieve a minimum transmission success probability $P_{\min}$, *i.e.*,

$$c_{ik} = z\left(-\frac{\beta \cdot d_{i,j}^{\alpha}}{\ln\left(P_{\min}\right)}\right) \tag{4.6}$$

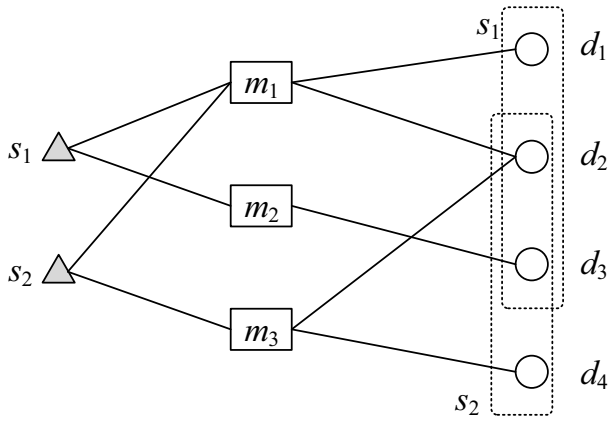
where $z(\cdot)$ is a monotonic non-decreasing function which maps the required resource to a comparable cost.

4.2 Multicast Message Allocation Problem and Solutions

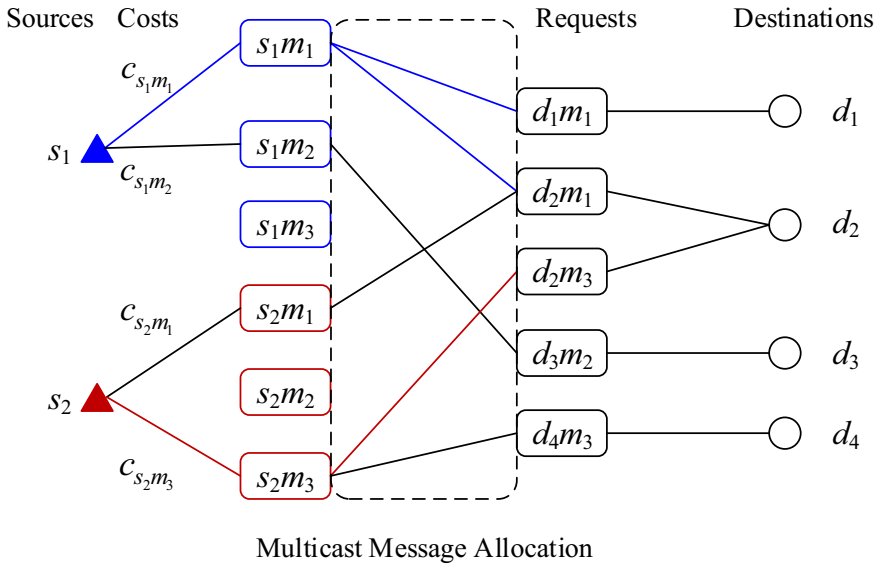
The multicast message allocation problem described in Section 4.1 can be formulated from different perspectives. Consider a concrete example which is modelled as a graph in Fig. 4.3(a). Here, there are two source devices in S, three messages in M, and four destination devices in D. Three destination devices fall within the transmission range of each source device. To present different formulations for the message allocation problem, we extend the graph in Fig. 4.3(a) to a new graph in Fig. 4.3(b). Assume that the resource cost for a source device to broadcast a message is specified according to the model in Section 4.1. In addition, each destination device can obtain its missing message from certain source devices within its transmission range. Given each cache device is subject to a multicast cost as described in Section 4.1.2, the cache devices should be allocated multicast messages appropriately to achieve certain design goal.

4.2.1 Weighted Set Cover Problem

First, each source device can be assigned to multicast certain messages so that all service requests are fulfilled with a minimum total transmission cost. From this



(a) A graph model.



(b) An extended graph model.

Fig. 4.3: A concrete content distribution example, where s_1 can transmit to $\{d_1, d_2, d_3\}$ while s_2 can reach $\{d_2, d_3, d_4\}$.

perspective, the multicast message allocation problem can be formulated as

$$\text{minimize} \quad \sum_{i=1}^{n'} \sum_{m_k \in M_i} c_{ik} x_{ik} \tag{4.7a}$$

$$\sum_{i=1}^{n'} x_{ik} t_{ij} \geq 1, \forall r_{jk} = 1, m_k \in M_i, d_j \in D \tag{4.7b}$$

$$x_{ik} \in \{0, 1\}, \forall s_i \in S \cup S_{BS}, \forall m_k \in M_i. \tag{4.7c}$$

Here, the binary parameter t_{ij} represents whether requesting device d_j falls within the transmission range of s_i, and r_{jk} is also binary indicating whether d_j requests message m_k. The set of requests is denoted by R, in which each element corresponds to a request (j, k) from d_j for message m_k. To ensure that there is always a feasible solution, we consider the BS as virtual cache devices and add a set of $|R|$ duplicated nodes, denoted by S_{BS}, to S. Each virtual device in S_{BS} only serves one request with a cost equal to the BS's unicast cost for the request. Then, the new set of cache devices is denoted by $S' = S \cup S_{BS}$, where $n' = |S'|$. Accordingly, x_{ik} is a binary variable indicating whether cache device $s_i \in S'$ sends message $m_k \in M_i$ to the reachable requesting devices.

In this problem formulation, the requests served by cache device s_i when sending message m_k is a subset of the requests in R. Problem (4.7) aims to find certain subsets whose union is the universe R and whose total cost is minimal. As seen, this is an NP-hard weighted set cover problem (WSCP). A greedy algorithm [27] can solve this problem with an approximation ratio $[1 + \ln(d')]$, where $d' = \max\{|D_{ik}|, \forall s_i \in S', m_k \in M_i\}$, is the maximum cardinality of the subsets.

As seen in Alg. 3, the message allocation solution x_{ik} is initialized to be all zeros. The corresponding set of satisfied requests Q is empty. Then we start a loop to find a subset of requests that can be satisfied with the minimum average cost per

Algorithm 3 Multicast message allocation based on WSCP formulation.

Input: S' (set of cache devices and BS-like virtual devices), M (set of messages), R (set of requests)
Output: $x = \{x_{ik} : s_i \in S', m_k \in M\}$

Initialize $x_{ik} \leftarrow 0, \forall s_i \in S', m_k \in M_i$
Initialize $Q \leftarrow \varnothing$
while $Q \neq R$ **do**
    $\{s_{i'}, k'\} \leftarrow \underset{\forall s_i \in S', m_k \in M_i}{\operatorname{argmin}} \frac{c_{ik}}{|R_{ik}|}$
    $Q \leftarrow Q \cup R_{i'k'}$
    $x_{i'k'} \leftarrow 1$
end while
return $\{x_{ik}\}$

request. Let R_{ik} denote the requests that can be satisfied by cache device $s_i \in S'$ (including the virtual cache devices simulating the BS) by broadcasting message $m_k \in M_i$. For any cache device s_i, its average cost per request for broadcasting message m_k is evaluated by $\frac{c_{ik}}{|R_{ik}|}$. The cache device and message that give the minimum average cost are considered in each iteration. The corresponding subset of requests $R_{i'k'}$ is added into the set of satisfied requests Q. The loop continues until all requests are satisfied.

4.2.2 Lagrangian Relaxation with Subgradient Method

Nonetheless, a main drawback of the WSCP based algorithm is that the good-quality cache devices may be allocated multiple messages for multicast. This may quickly deplete the energy of these devices. In [3], it further limits the number of messages that can be multicast by each cache device by one. Here, we slightly modify their problem formulation. Similar to the formulation in (4.7), we add a number of duplicated nodes, S_{BS}, as virtual cache devices. Then, the message

allocation problem with the extra constraint is formulated as follows:

$$\text{minimize} \quad \sum_{i=1}^{n'} \sum_{m_k \in M_i} c_{ik} x_{ik} \tag{4.8a}$$

$$\text{subject to} \sum_{m_k \in M_i} x_{ik} \leq 1, \forall s_i \in S \cup S_{BS} \tag{4.8b}$$

$$\sum_{i=1}^{n'} x_{ik} t_{ij} \geq 1, \forall r_{jk} = 1, m_k \in M_i, d_j \in D \tag{4.8c}$$

$$x_{ik} \in \{0, 1\}, \forall s_i \in S \cup S_{BS}, \forall m_k \in M_i. \tag{4.8d}$$

As seen, (4.8b) is the extra constraint that limits the number of multicast messages allocated to each cache device. We note that this problem is a special hypergraph matching problem, *i.e.*, the three-dimensional matching problem, which is NP-hard. It is unknown whether constant-factor approximation algorithms exist. In [3], the authors first use the subgradient method for a Lagrangian dual to obtain a lower bound closest to the optimum and then apply a heuristics method to derive a feasible solution. Unfortunately, the approximation ratio of this algorithm is uncertain and not guaranteed. The algorithm is rewritten and presented in Alg. 4. According to [3], Lagrangian relaxation is first applied to problem (4.8). Relaxing constraint (4.8c), we have the *Lagrangian dual*:

$$\max_{\lambda} \quad Z(\lambda) = \min_{x} \left\{ \sum_{i=1}^{n'} \sum_{m_k \in M_i} c_{ik} x_{ik} + \sum_{r=(j,k)=1}^{\eta} \lambda_r \Big(1 - \sum_{i=1}^{n'} x_{ik} t_{ij}\Big) \right\} \tag{4.9a}$$

$$\text{subject to} \quad \sum_{m_k \in M_i} x_{ik} \leq 1, \forall s_i \in S' \tag{4.9b}$$

$$x_{ik} \in \{0, 1\}, \forall s_i \in S', m_k \in M_i \tag{4.9c}$$

where η is the number of requests from all destination devices, $\lambda = \{\lambda_1, ..., \lambda_\eta\} \geq 0$

Algorithm 4 Multicast message allocation based on Lagrangian relaxation and subgradient method.

Input: S' (set of cache devices and BS-like virtual devices), M (set of messages), R (set of requests), ϵ

Output: $x = \{x_{ik} : s_i \in S', m_k \in M\}$

Initialize $x_{ik} \leftarrow 0, \forall s_i \in S, m_k \in M_i$
Initialize $\lambda_r^0 \leftarrow \min \frac{c_{ik}}{R_{ik}}, \forall s_i \in S', m_k \in M_i, r \in R_{ik}$
$\tau \leftarrow 0$, $\zeta^\tau \leftarrow 1$, $Z^* \leftarrow -\infty$
repeat
    $Z(\lambda^\tau) \leftarrow 0$
    for $\forall s_i \in S'$ **do**
        $k' \leftarrow \text{argmin}\left(c_{ik} - \sum_{r \in R_{ik}} \lambda_r^\tau\right), \forall m_k \in M_i$
        $x_{ik'}^\tau \leftarrow 1$
        $Z(\lambda^\tau) \leftarrow Z(\lambda^\tau) + c_{ik'} x_{ik'}^\tau$
    end for
    for $r = (j, k) \in R$ **do**
        $\gamma_r^\tau \leftarrow 1 - \sum_{i=1}^{n'} x_{ik}^\tau t_{ij}$
        $Z(\lambda^\tau) \leftarrow Z(\lambda^\tau) + \gamma_r^\tau \lambda_r^\tau$
    end for
    if $Z(\lambda^\tau) > Z^*$ **then**
        $x_{ik} \leftarrow x_{ik}^\tau$
        $Z^* \leftarrow Z(\lambda^\tau)$
    end if
    for $1 \leq r \leq \eta$ **do**
        $\lambda_r^{\tau+1} \leftarrow \max\{0, \lambda_r^\tau + \zeta^\tau \gamma_r^\tau\}$
    end for
    $\zeta^{\tau+1} \leftarrow \frac{\zeta^\tau}{2}$
    $\tau \leftarrow \tau + 1$
until $|Z(\lambda^\tau) - Z(\lambda^{\tau-1})| < \epsilon$
$R' \leftarrow$ Requests that are not satisfied by $\{x_{ik}\}$
for $\forall s_i \in S', \sum_{m_k \in M_i} x_{ik} = 0$ **do**
    $k' \leftarrow \text{argmin}_{m_k \in M_i}\{c_{ik}\}$
    // s_i fulfills request(s) in R' by sending message $m_{k'}$
    if $R_{ik'} \cap R' \neq \varnothing$ **then**
        $x_{ik'} \leftarrow 1$
    end if
end for
return $\{x_{ik}\}$

are the *Lagrange multipliers*, and $Z(\lambda)$ is the *Lagrangian subproblem*, given by

$$Z(\lambda) = \min_{x} \left\{ \sum_{i=1}^{n'} \sum_{m_k \in M_i} c_{ik} x_{ik} - \sum_{i=1}^{n'} \sum_{r=(j,k)=1}^{\eta} \lambda_r x_{ik} t_{ij} + \sum_{r=1}^{\eta} \lambda_r \right\} \tag{4.10a}$$

$$\text{subject to} \quad \sum_{m_k \in M_i} x_{ik} \leq 1, \forall s_i \in S' \tag{4.10b}$$

$$x_{ik} \in \{0,1\}, \forall s_i \in S', m_k \in M_i. \tag{4.10c}$$

Given fixed Lagrange multipliers, the Lagrangian subproblem can be easily solved by selecting at most one message (if any is available) with the minimum $(c_{ik} - \sum_{r \in R_{ik}} \lambda_r)$, where R_{ik} is the subset of requests that can be fulfilled by source s_i sending message $m_k \in M_i$.

To solve the Lagrangian dual in (4.9), the subgradient method can be used to obtain a lower bound closest to the optimum. Let λ^τ denote the tentative solution to the Lagrangian dual in iteration τ. First, λ^0 is initialized such that $\lambda_r^0 = \min \frac{c_{ik}}{R_{ik}}$, $\forall s_i \in S', m_k \in M_i$, and $r \in R_{ik}$. That is, λ_r^0 is set to the minimum average cost to fulfill request r. According to the subgradient method, λ^τ in iteration τ is updated iteratively by

$$\lambda^{\tau+1} = \max\{0, \lambda^\tau + \zeta_\tau \gamma^\tau\} \tag{4.11}$$

where ζ_τ is the step size, and γ_r^τ is the *subgradient* in iteration τ. Let x^τ denote the solution to the Lagrangian subproblem in iteration τ. The subgradient of $Z(\lambda)$, *i.e.*, γ_r^τ, is given by

$$\gamma_r^\tau = 1 - \sum_{i=1}^{n'} x_{ik}^\tau t_{ij}, \quad r = (j,k),\ 1 \leq r \leq \eta. \tag{4.12}$$

The iterations continue until the gap between $Z(\lambda^{\tau})$ and $Z(\lambda^{\tau-1})$ is sufficiently small. Then the Lagrangian dual is solved. The solution x_{ik}^{τ} is further modified with a heuristic algorithm so as to derive a feasible solution when the relaxed constraint is added back. It just scans all sources without allocated messages, starting with the message at the minimum cost, and checks if they can serve more unfulfilled requests. This procedure stops until no further message can be allocated to improve the fulfilled requests by sources. The remaining unfulfilled requests are rolled back to be served by the BS.

4.2.3 Multi-Choice Knapsack Problem

From the perspective of the whole system, it is preferable that the total transmission cost is minimized, while the BS may also aim to maximize its gain over unicast by diverting the requests to cache devices. Hence, we consider a different formulation that focuses on the gain achieved by diverting the requests from the BS to the cache devices. Moreover, our formulation takes into account a more flexible constraint on the cost budget of the cache devices similar to [18]. Then, the message allocation problem can be formulated as

$$\text{maximize} \quad \sum_{i=1}^{n} \sum_{m_k \in M_i'} (c_{Bk_i} - c_{ik}) x_{ik} \tag{4.13a}$$

$$\text{subject to} \quad \sum_{i=1}^{n} \sum_{m_k \in M_i'} c_{ik} x_{ik} \leq C \tag{4.13b}$$

$$\sum_{m_k \in M_i'} x_{ik} = 1, \forall s_i \in S \tag{4.13c}$$

$$x_{ik} \in \{0, 1\}, \forall s_i \in S, m_k \in M_i'. \tag{4.13d}$$

Here, for each cache device $s_i \in S$, we add a dummy message m_d with a zero cost and zero gain to the subset of available messages at s_i, which gives $M_i' = M_i \cup \{m_d\}$ and extends c_{ik} and c_{Bk_i} correspondingly. For simplicity, we use the same notation for c_{ik} and c_{Bk_i} without confusion.

As seen in (4.13b), the BS limits the total cost for all cache devices allocated with one multicast message by a cost budget C. This is to further address the varying costs of the cache devices for message multicast. In the objective function (4.13a), c_{Bk_i} is the BS's total cost to satisfy the requests that cache device s_i is able to serve by multicasting message m_k, so $(c_{Bk_i} - c_{ik})$ is the gain by diverting requests from the BS to s_i. Assume that $c_{ik} \leq c_{Bk_i}$ and $0 < c_{ik} \leq C, \forall s_i \in S, m_k \in M_i'$. For reference convenience, we denote $(c_{Bk_i} - c_{ik})$ by a gain parameter g_{ik} and represent the objective function in (4.13a) by $g(x)$. Hence, this formulation targets at the maximum gain achieved by involving D2D-assisted content distribution. Note that we have equality in constraint (4.13c), because there always exists a feasible solution that allocates the zero-cost dummy message to a cache device. Then, problem (4.13) is translated into the multiple-choice knapsack problem (MCKP) [28].

The classes for the MCKP is the cache devices in S. The messages in $M' = \sqcup M_i'$ are the items for allocation, where $\sqcup_i M_i'$ is the *disjoint* union of the subsets of messages of all cache devices, *i.e.*, the union of the elements in each M_i' by retaining the original set membership. For example, in Fig. 4.3(a), $M_1 = \{m_1, m_2\}$ and $M_2 = \{m_1, m_3\}$, which gives $M_1 \sqcup M_2 = \{s_1 m_1, s_1 m_2, s_2 m_1, s_2 m_3\}$. Let $\eta = |M'|$ denote the problem size. As the MCKP is NP-hard, we can use a fully polynomial-time approximation scheme (FPTAS) to obtain a suboptimal solution. Here, we consider the FPTAS in [2], which gives a $(1+\epsilon)$-approximation for any arbitrary $\epsilon > 0$ in running time polynomial in both the problem size η and $1/\epsilon$.

The details of this algorithm are given in Alg. 5. First, the algorithm proposed by Dyer and Zemel in [28] is used to obtain a fractional optimal solution x^* for the linear relaxation of problem (4.13). Let P^* denote the objective value of x^*, which contains at most two fractional variables. Consider a simple rounding algorithm, which returns an integer solution x_0 that maximizes the gain among three alternative integer solutions [2]: 1) x_{ak_1} that only keeps one fractional variable; 2) x_{ak_2} that keeps the other fractional variable; and 3) $x^*_{\bar{a}}$ that discards both fractional variables from x^*. Here, the first two solutions are feasible since any $c_{ik} \leq C$. The objective value of x_0 is denoted by P_0, which is further used in the scale factor K. A new instance of the MCKP in (4.13) is generated by scaling down $(c_{Bk_i} - c_{ik})$ in (4.13a) to $\lfloor \frac{c_{Bk_i} - c_{ik}}{K} \rfloor$ and solved by dynamic programming. Let $F\{i, p\}$ denote the minimum cost of only allocating messages to the first i cache devices so that the total gain would be p. Then, the solution x_s for the scaled problem is obtained so that the corresponding objective value $\hat{P}_s = \max\{p | F(n, p) \leq C\}$. The gain of x_s in the original problem (4.13) is obtained as P_s. Comparing P_0 and P_s, Alg. 5 returns the more profitable solution between x_0 and x_s.

4.3 Simulation Results and Discussions

In this section, we compare the performance of the algorithms that solve the three problems formulated in Section 3.2.3, including the WSCP in (4.7), hypergraph matching in (4.8), and MCKP in (4.13). We first consider the static scenario, in which all devices are distributed uniformly in the cell, the messages demanded by the requesting devices follow a Zipf distribution with exponent γ_r, and the messages stored at cache devices follow another Zipf distribution with exponent γ_c. Then, we evaluate the performance in a dynamic scenario where devices arrive

Algorithm 5 Multicast message allocation based on MCKP formulation.

Input: S (set of cache devices), D (set of requesting devices), R (set of requests), C (limit of cost), ϵ, $\{c_{Bk_i}, c_{ik} : s_i \in S, m_k \in M_i'\}$

Output: $x = \{x_{ik} : s_i \in S, m_k \in M\}$

Use Dyer-Zemel algorithm for (4.13) to obtain x^*
// Obtain first solution x_0 and scale factor K)
if x^* has no fractional variables **then**
    return $x \leftarrow x^*$
end if
// Rounding algorithm
$\{x_{ak_1}, x_{ak_2}\} \leftarrow$ two fractional variables of x^*
$p_1 \leftarrow g(x_{ak_1})$, $p_2 \leftarrow g(x_{ak_2})$, $p_3 \leftarrow g(x_{\bar{a}}^*)$
$P_0 \leftarrow \max\{p_1, p_2, p_3\}$, $x_0 \leftarrow$ solution to achieve P_0
$K \leftarrow \frac{\epsilon P_0}{n}$
// Obtain second candidate solution x_s by solving new MCKP
// instance scaled by K
for each $g_{ik}, \forall s_i \in S, m_k \in M_i'$ **do**
    $\hat{g}_{ik} = \lfloor \frac{c_{Bk_i} - c_{ik}}{K} \rfloor$
end for
Use Dyer-Zemel algorithm for scaled MCKP with $\hat{g}_{ik}$ and calculate $\hat{P}_0$ with the rounding algorithm
for each $p \in \{1, ..., 3\hat{P}_0\}$ **do**
    $F(0, p) \leftarrow \infty$, $x_0^p \leftarrow \emptyset$
end for
for each $p \in \{1, ..., 3\hat{P}_0\}$ **do**
    for each $i \in \{0, ..., n-1\}$ **do**
        Update $F(i+1, p)$ and corresponding x_{i+1}^p
    end for
end for
$\hat{P}_s \leftarrow \max\{p | F(n, p) \leq C\}$, $x_s \leftarrow x_n^{\hat{P}_s}$
$P_s \leftarrow \sum_{s_i \in S, m_k \in M_i'} (c_{Bk_i} - c_{ik}) x_s$
if $P_s > P_0$ **then**
    $x \leftarrow x_s$
else
    $x \leftarrow x_0$
end if
return x

Table 4.1: Simulation parameters.

Symbol	Definition	Values
Φ	Radius of cell coverage (m)	500
L	Maximum D2D transmission range (m)	200
α	Path-loss exponent	3
P_d	Power of D2D transmitters (mW)	100
P_c	Transmit power of cellular user (mW)	100
P_{BS}	Transmit power of BS (W)	10
B	Carrier bandwidth (MHz)	1
β	Decoding SINR threshold	2
$P_{\min}$	Target transmission success possibility	0.8
m	Number of messages	10
m_i	Cache capacity at cache device s_i	2
γ_c	Zipf exponent for cached messages	0.7
γ_r	Zipf exponent for requested messages	0.9
λ_c	Arrival rate of cache devices (per unit time)	1
λ_r	Arrival rate of requesting devices (per unit time)	2
μ_c	Average staying time of cache devices (unit time)	30
μ_r	Average staying time of requesting devices (unit time)	30
λ_q	Average rate of requests (per unit time)	1
ϵ	Accuracy parameter for MCKP	0.1
F	File size of each message	1000

at and leave the cell according to the system model described in Section 4.1. Table 4.1 lists the default values of the main simulation parameters if they are not specified elsewhere. In the following experiments, we vary the values of some parameters to examine their impact.

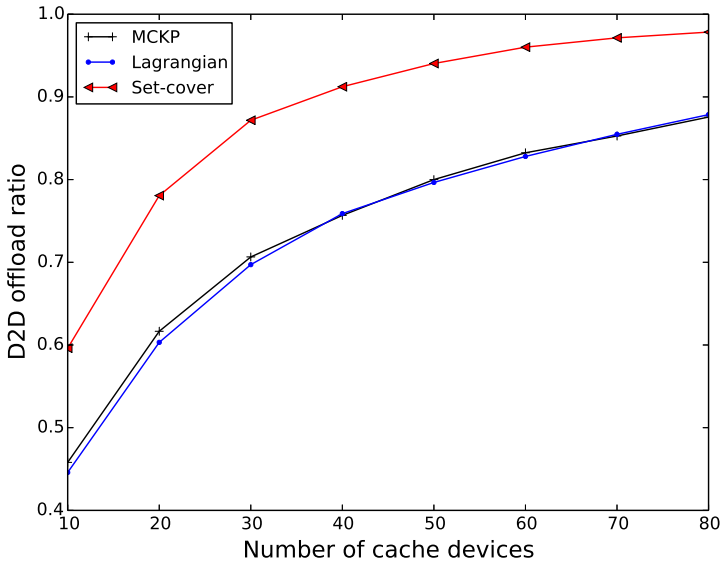
4.3.1 Static Scenario with Different Settings of Cache Devices

First, we consider the static scenario that the devices do not move and the parameters that generate the requests are fixed. Here, two important metrics are evaluated, *i.e.*, the total cost to fulfill all requests and the ratio of requests offloaded to D2D multicast. In this section, some parameters related to the cache devices are varied to evaluate their effects on performance.

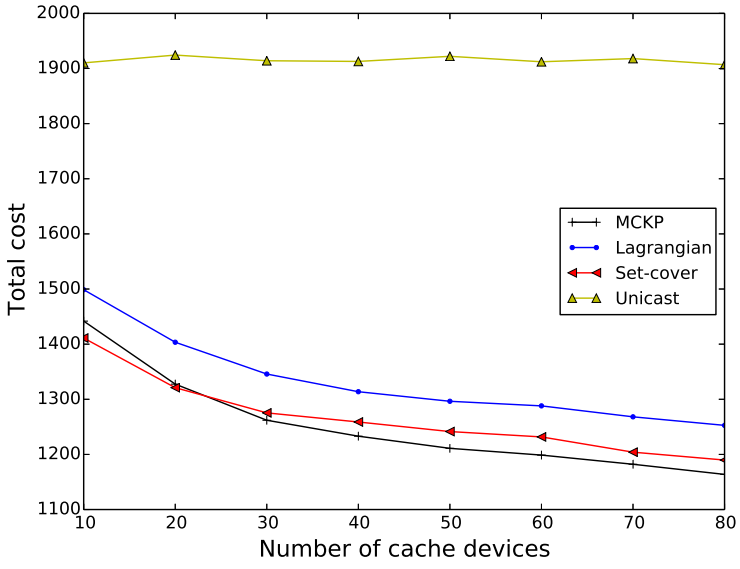
4.3.1.1 Different numbers of cache devices

Fig. 4.4 presents the result with a varying number of cache devices. The number of cache devices varies between 10 to 80 while the number of requesting devices is fixed at 50. In the mean while, the Zipf exponent for cached messages is fixed at 0.7 and the Zipf exponent for requested messages is fixed at 0.9. The cache capacity of each cache device is fixed at 2 messages. The result for each case is the average of 1000 random simulations.

Fig. 4.4(a) shows the D2D offload ratio, *i.e.*, the ratio of requests offloaded to D2D multicast, when the number of cache devices is different. As seen, the D2D offload ratio increases with the number of cache devices since more candidates become available to multicast messages. When the number of cache devices increases, all three algorithms can offload more requests to D2D multicast communications. The WSCP based algorithm achieves the highest offload ratio because there is no limit on the resource cost at each cache device. In contrast, the MCKP based algorithm and the Lagrangian relaxation based algorithm both offload fewer requests to accommodate the resource constraints of cache devices. In addition, the latter two algorithms perform very closely.



(a)



(b)

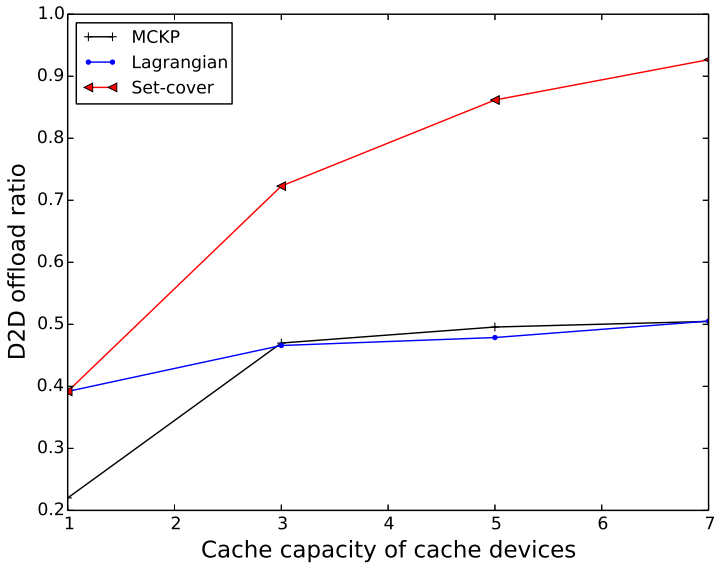
Fig. 4.4: Result when the number of cache devices varies.

As seen in Fig. 4.4(b), the total cost of three algorithms all decreases when the number of cache devices increases. This is because more requests can be offloaded to D2D multicast leading to lower total costs. Normally, D2D multicast can simultaneously satisfy multiple requests by sending one message and thus lead to lower costs than unicast by the BS for fulfilling the same set of requests. The Lagrangian relaxation based algorithm results in higher costs than the other two algorithms, which is partly due to the limit of one multicast message per cache device. In contrast, the WSCP based algorithm has no constraint on the number messages that each cache device is allocated. The MCKP based algorithm has offload ratio similar to that of the Lagrangian relaxation based algorithm, but the former one has lower total cost than the latter one. This is because the MCKP based algorithm aims at achieving more gain from diverting requests from the BS to D2D multicast. According to the definition of gain, a higher gain corresponds to a lower total cost.

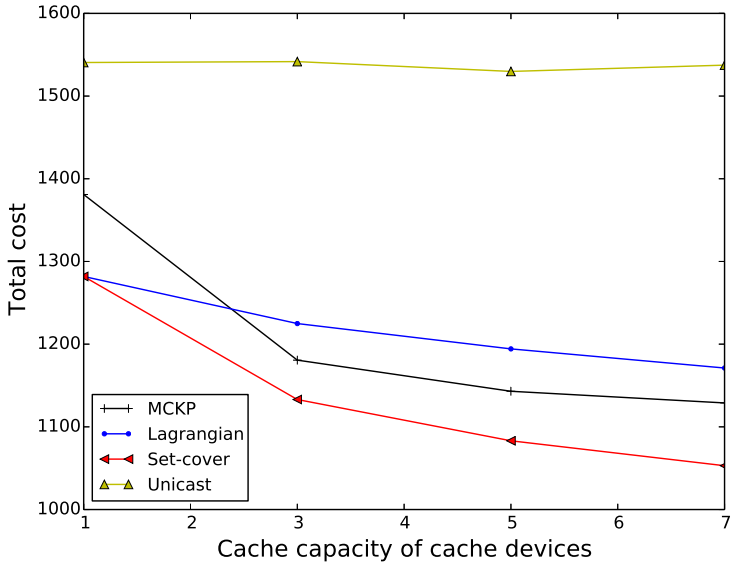
4.3.1.2 Varying cache capacity of cache devices

Fig. 4.5 shows the result when the cache capacity of cache devices varies. In this scenario, the number of cache devices is fixed at 10 and the number of requesting devices is fixed at 40. The Zipf exponent for cached messages γ_c is fixed at 0.7 and the Zipf exponent for requested messages γ_r is fixed at 0.9. The cache capacity of cache devices varies between 1 and 7.

In terms of D2D offload ratio, Fig. 4.5(a) shows that all of the three algorithms tend to have a higher D2D offload ratio as the cache capacity increases. This is because a larger cache capacity enables more message allocation options and leads to a higher D2D offload ratio. Similar as the result in Fig. 4.4, the WSCP based algorithm achieves the highest D2D offload ratio. The other two algorithms have



(a)



(b)

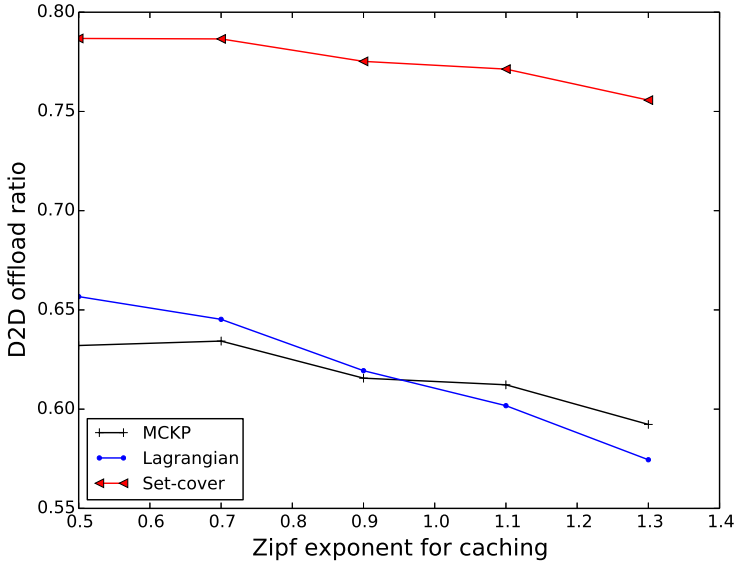
Fig. 4.5: Result when the cache capacity of cache devices varies.

close D2D offload ratio due to the constraint that one message at most is allocated to each cache device. Because of the same reason, the offload ratio of the MCKP based algorithm and the Lagrangian relaxation based algorithm only increases slightly when the cache capacity of cache devices is greater than 3. Nonetheless, the offload ratio of the WSCP based algorithm continues to increase substantially. Fig. 4.5(b) shows the total cost for fulfilling all the requests. As seen, all three algorithms tend to have a lower total cost with the increment of the cache capacity of cache devices. This trend can be explained by referring to the result in Fig. 4.5(a). Moreover, among these three algorithms, the WSCP based algorithm has the lowest total cost, while the Lagrangian relaxation based algorithm has the highest total cost. On one hand, the constraint on the number of allocated messages per device affects the performance of the Lagrangian relaxation based algorithm. On the other hand, the heuristic nature of this algorithm also restricts the exploration of better message allocation options. As a consequence, the message allocation result may not be satisfactory in some cases.

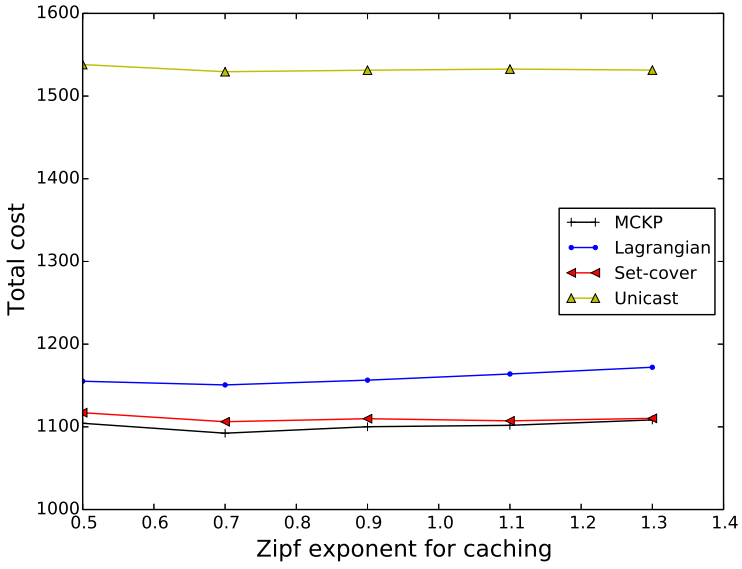
4.3.1.3 Different Zipf exponents for cached messages

In this scenario, we evaluate the performance of these three algorithms when the Zipf exponent for cached messages γ_c varies between 0.5 and 1.3. The number of cache devices is fixed at 20 and the number of requesting devices is fixed at 40. The cache capacity of cache devices is fixed at 2. The Zipf exponent for requested messages is fixed at 0.9.

Fig. 4.6(a) shows the D2D offload ratio versus the varying Zipf exponent for cached messages γ_c. As seen, for all three algorithms, the D2D offload ratio tends to decrease with a larger γ_c. While γ_c is increasing, it implies that all cache devices intend to cache the most popular messages. Consequently, many requests cannot



(a)



(b)

Fig. 4.6: Result when the Zipf exponent for cached messages γ_c varies.

be fulfilled by the cache devices because the requested messages are not available in any cache device within the maximum D2D transmission range. This explains why all three algorithms result in a lower D2D offload ratio when γ_c is increasing. As always, the WSCP based algorithm has the highest D2D offload ratio, while the other two algorithms have lower but close offload ratio due to the constraint on one multicast message per cache device.

On the other hand, Fig. 4.6(b) shows that the total cost to satisfy all the requests with varying γ_c. As seen in all three algorithms, the total cost does not present any evident variation with γ_c. Even though the D2D offload ratio decreases with γ_c, the decrease is too small to cause any significant change in the total cost. Despite this observation, it is still clearly shown that the Lagrangian relaxation based algorithm is subject to the highest total cost, while the total cost of the WSCP based algorithm is almost the lowest. This is consistent with the D2D offload ratio in Fig. 4.6(a). The total cost of the MCKP based algorithm is very close to that of the WSCP based algorithm. This demonstrates the good performance of the FPTAS that is used to solve the MCKP formulation.

4.3.2 Static Scenario with Different Settings of Requesting Devices

In this section, we show the result of the three message allocation algorithms in the static scenario when some parameters related to the requesting devices are varied. As above, we focus on the total cost and the D2D offload ratio.

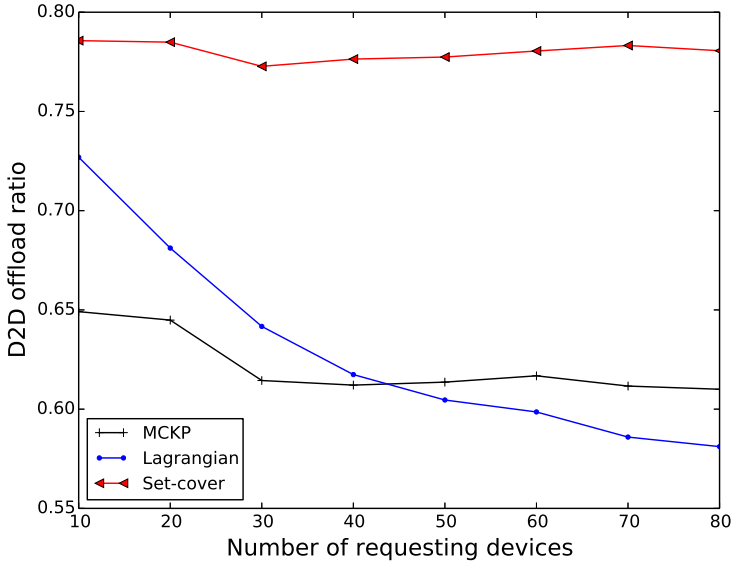
4.3.2.1 Different numbers of requesting devices

As in Section 4.3.1, we set the Zipf exponent for cached messages γ_c to 0.7 and the Zipf exponent for requested messages γ_r to 0.9. The number of cache devices

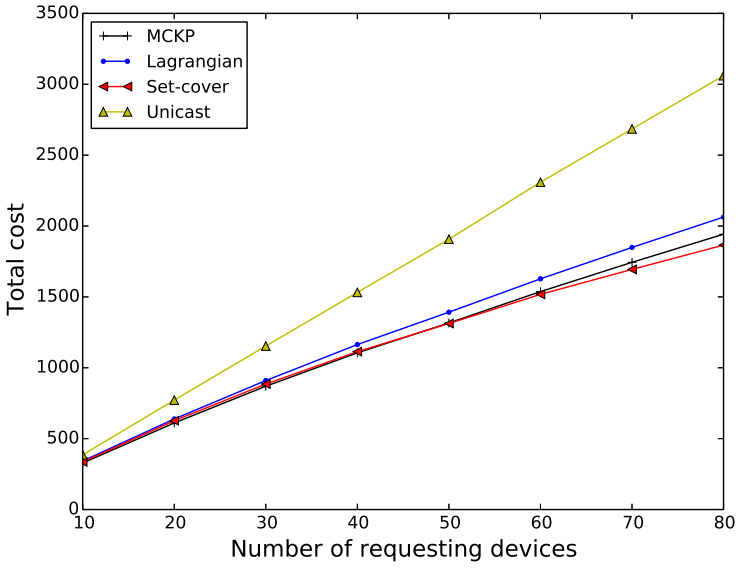
is fixed at 20 and the cache capacity of each cache device at 2.

Fig. 4.7 shows the total cost and D2D offload ratio of the three algorithms when the number of requesting devices varies between 10 and 80. First, it can be seen in Fig. 4.7(a) that the D2D offload ratio of the WSCP algorithm is not affected much by the number of requesting devices. This is because the WSCP based algorithm can exploit every cache device to multicast multiple messages. Hence, the algorithm can keep a high D2D offload ratio, as long as the number of cache devices is sufficient. On the other hand, the D2D offload ratio of the other two algorithms decreases with the number of requesting devices. This can be interpreted easily since more requesting devices start more requests. Since the Lagrangian relaxation based algorithm and the MCKP based algorithm limit one multicast message for each cache device, some requests may not be fulfilled by D2D multicast when the number of requesting devices is large.

Fig. 4.7(b) shows the corresponding total cost of the three algorithms. As seen, all the algorithms result in a higher cost when the number of requesting devices increases. More requesting devices can potentially start more requests and thus end up with higher total cost. Moreover, as the number of requesting devices becomes larger and larger, the differences among these three algorithms are more evident. The WSCP based algorithm achieves the best performance, which is reasonable since it has the highest D2D offload ratio. As always, the MCKP based algorithm outperforms the Lagrangian relaxation based algorithm in terms of the total cost because of the FPTAS for gain maximization. The gap between the total cost of the three algorithms and that of unicast by the BS also becomes greater with the increase of the number of requesting devices. This is because when a cache device multicasts a message, it can fulfill more requesting devices in the neighborhood with a higher density of requesting devices.



(a)



(b)

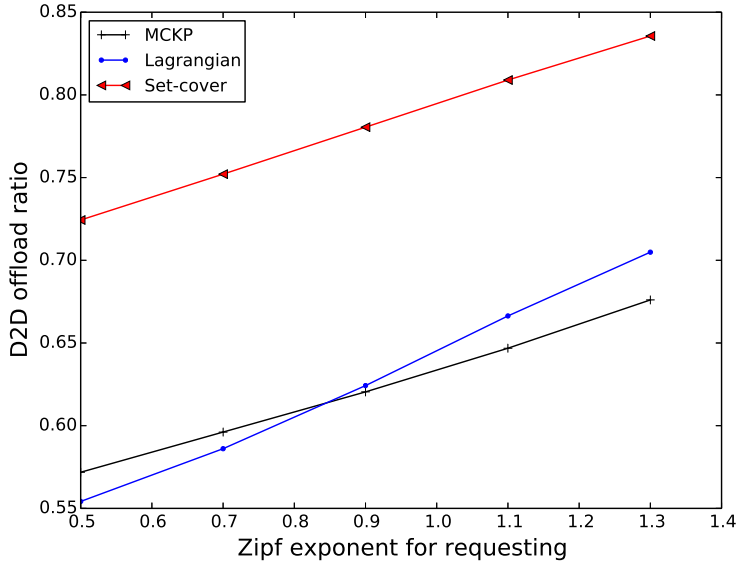
Fig. 4.7: Result when the number of requesting devices varies.

4.3.2.2 Different Zipf exponents for requested messages

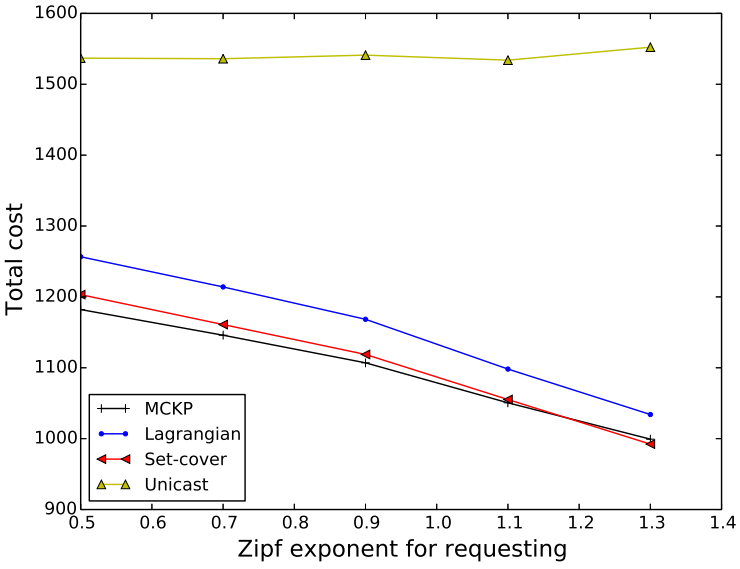
In this scenario, the number of cache devices is fixed at 20, the number of requesting devices is fixed at 40, the cache capacity of each cache device is set to 2, and the Zipf exponent for cached messages is fixed at 0.7.

Fig. 4.8 shows the result when the Zipf exponent for requested messages varies between 0.5 and 1.3. Fig. 4.8(a) shows the D2D offload ratio. Similar to the above results for the static scenario, the WSCP based algorithm achieves the best performance due to the unlimited number of multicast messages at each cache device. Moreover, it can be seen that the offload ratio of all three algorithms increases with the Zipf exponent for requested messages γ_r. When γ_r is increasing, all the requesting devices tend to request the most popular messages. This increases the possibility that the requested messages are available at the cache devices within the maximum transmission range, and thus leads to a higher D2D offload ratio. The MCKP based algorithm and Lagrangian relaxation based algorithm perform closely since both are subject to the same limit on the number of multicast messages per cache device.

Fig. 4.8(b) shows the total cost to fulfill all the requests. As seen, the costs of all three algorithms decrease when the Zipf exponent for requested messages is increasing. Referring to the result in terms of D2D offload ratio in Fig. 4.8(a), we have seen that the D2D offload ratio increases with the Zipf exponent for requested messages. When the total number of requesting devices is fixed, this means that the total number of requests is fixed since each requesting device only requests for two messages. That is why the total cost for unicast by the BS does not change. In this situation, a higher D2D offload ratio leads to more requests diverted from the BS to D2D multicast transmission. Hence, a higher offload ratio leads to a lower total cost. Among these three algorithms, the Lagrangian



(a)



(b)

Fig. 4.8: Result when the Zipf exponent for requested messages γ_r varies.

relaxation based algorithm is subject to the highest total cost, while the other two algorithms present lower total cost and are close to each other. The slight gain of the MCKP based algorithm over the Lagrangian relaxation based algorithm can be interpreted similarly as above.

4.3.3 Dynamic Scenario with Different Settings of Cache Devices

In the dynamic scenario, we consider arrivals and departures of devices and the dynamics of requests. Different from the static scenarios, the parameters used to control the density of devices are no longer the numbers of devices. Since the devices come and leave as time goes by, we use the arrival rate of devices and average staying time of devices to control the density of devices. In this section, we examine the impact of the parameters associated with the cache devices. Then, in Section 4.3.4, we vary the parameters of the requesting devices and evaluate the corresponding performance. In addition to that, we also use the arrival rate of requests for each requesting device to control the number of requests for each requesting device.

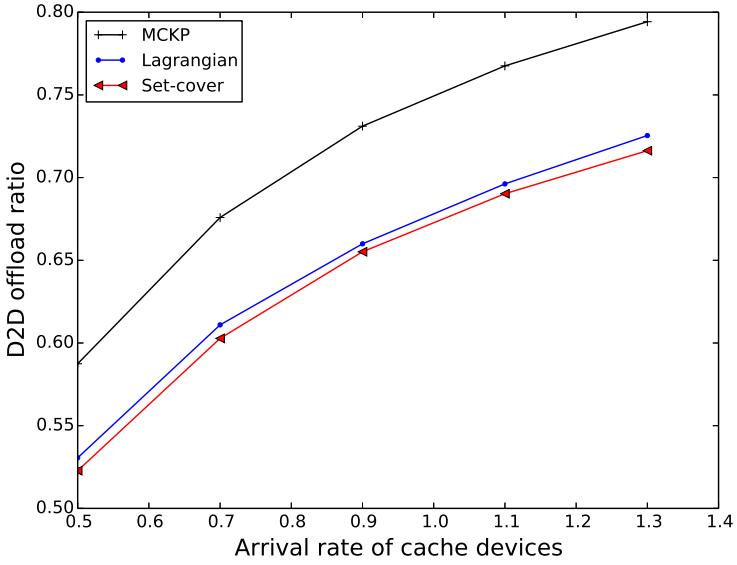
In the dynamic scenario, message allocation is conducted periodically as depicted in Fig. 4.2. At the end of each period, the requesting and caching information is collected to allocate multicast messages for cache devices at the beginning of the next period. In case that not all requests are fulfilled in the current round, the remaining requests are served by the BS. In each simulation, we consider 100 time periods. All the results presented in this section and Section 4.3.4 are the average of 1000 random cases. Four key performance indicators are considered in the dynamic scenario. In addition to the total cost and D2D offload ratio, we also evaluate the unit cost per message request and the service latency from the

moment that a request arrives to the moment that it is successfully served.

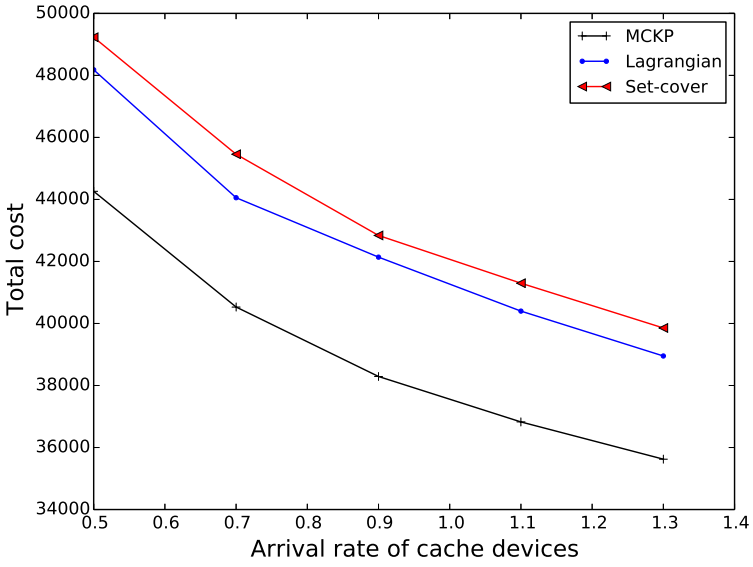
4.3.3.1 Varying arrival rate of cache devices

First, we consider a dynamic scenario where the arrival rate of cache devices varies between 0.5 and 1.3 and the other parameters are as specified in Table 4.1. Fig. 4.9 shows the result of the three algorithms when the arrival rate of cache devices varies between 0.5 and 1.3. As seen in Fig. 4.9(a), the MCKP based algorithm achieves the highest D2D offload ratio. Although the WSCP based algorithm allows to allocate multiple messages to a good-quality cache device, the cache device still needs to send the allocated messages one by one and hence restricts the offload ratio. The Lagrangian relaxation based algorithm essentially solves the cost minimization problem in a heuristic manner without an approximation guarantee. As a consequence, the message allocation result may not be satisfactory in some cases. On the other hand, the MCKP based algorithm uses the FPTAS with an approximation guarantee that is made sufficiently close to the optimum by setting the accuracy parameter ϵ to 0.1. Hence, this algorithm achieves the highest offload ratio.

Fig. 4.9(b) shows that the total costs of all three algorithms decrease with a higher arrival rate of cache devices, which increases the density of cache devices. In addition, it can be observed that the MCKP based algorithm achieves the lowest cost, while the greedy algorithm results in the highest cost. The different cost of the three algorithms can be explained by the corresponding offload ratio shown in Fig. 4.9(a). The consistency of the results in Fig. 4.9(b) and Fig. 4.9(a) also imply that almost all requests served by D2D multicast are successfully completed instead of rolling back to the BS. Consequently, Fig. 4.9(c) shows that the unit cost per served request exhibits a similar trend as the total cost in Fig. 4.9(b).

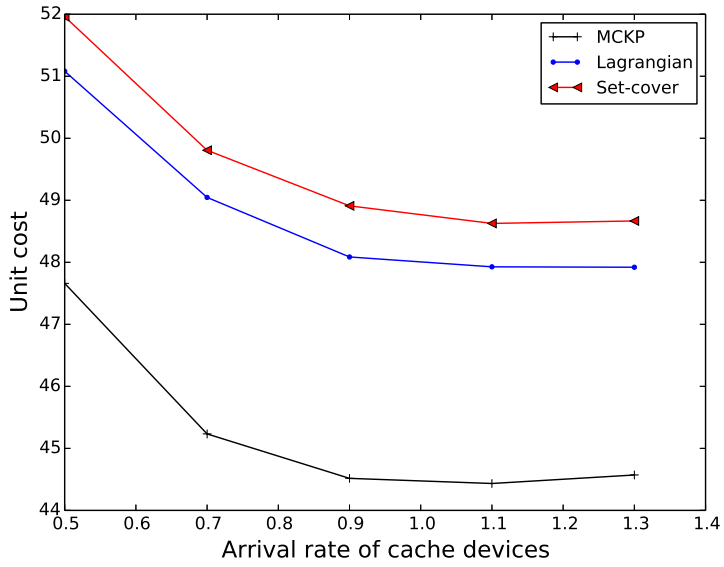


(a)

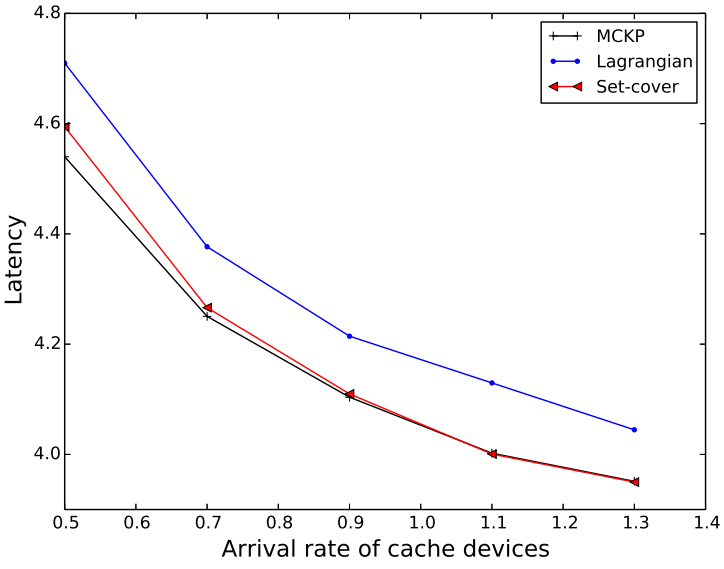


(b)

Fig. 4.9: Result when the arrival rate of cache devices varies.



(c)



(d)

Fig. 4.9: Result when the arrival rate of cache devices varies.

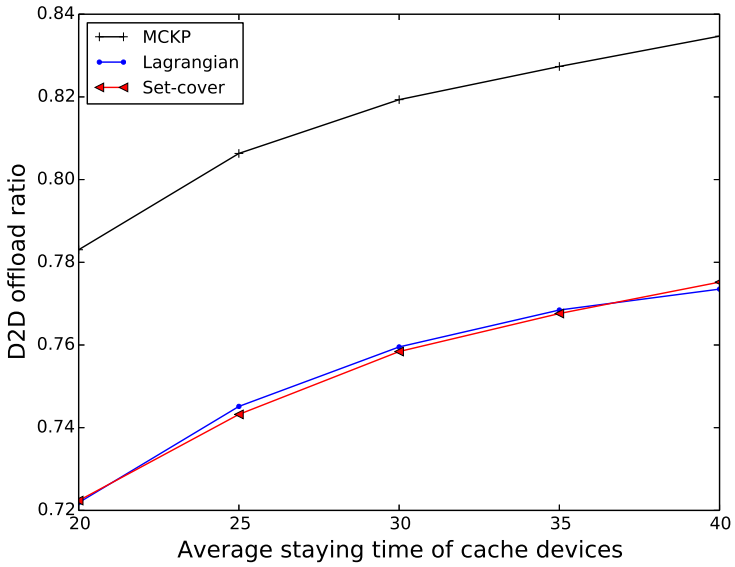
In terms of service latency, Fig. 4.9(d) shows that the FPTAS for the MCKP and the greedy algorithm for the WSCP achieve better performance than the Lagrangian relaxation based algorithm. As the resource cost is inversely proportional to mean SINR, a lower cost implies a higher transmission rate on average and thus leads to a lower latency. Due to the heuristic nature of the Lagrangian relaxation based algorithm, its latency performance is worse than the other two algorithms.

4.3.3.2 Different average staying time of cache devices

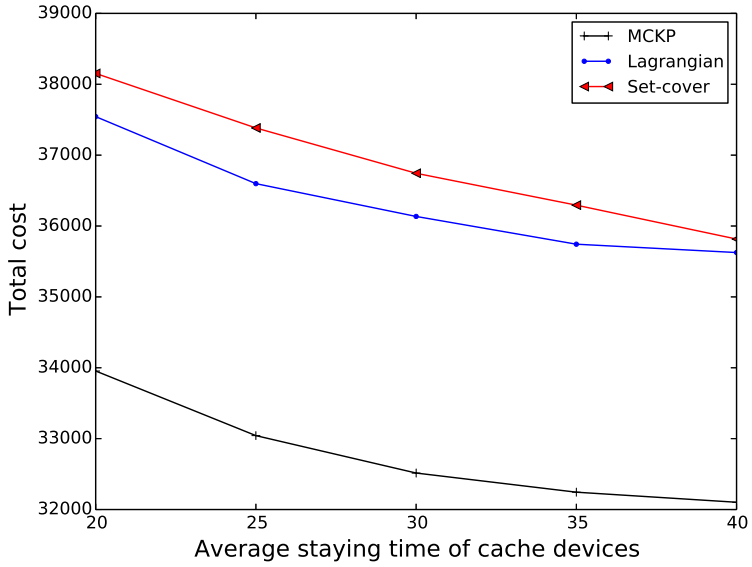
In this scenario, we change the average staying time of cache devices between 10 and 40, while keeping the values of other parameters the same as Table 4.1. Fig. 4.10 presents the result of this scenario. First, Fig. 4.10(a) shows that the D2D offload ratio of all three algorithms increases when the average staying time of cache devices is longer. Moreover, the MCKP based algorithm achieves the highest D2D offload ratio, the WSCP based algorithm achieves the lowest D2D offload ratio, and the Lagrangian relaxation based algorithm stays in the middle of them.

Fig. 4.10(b) shows that all three algorithms lead to lower total cost when the average staying time of cache devices increases. The MCKP based algorithm performs the best, while the WSCP based algorithm is subject to the highest total cost. This is consistent to the observation on D2D offload ratio in Fig. 4.10(a). The unit cost shown in Fig. 4.10(c) presents a similar trend among the three algorithms. In addition, there is only slight change with the unit cost because the decrease of the total cost is offset by more requests satisfied by D2D multicast.

Fig. 4.10(d) shows that the latency of all three algorithms decreases with the increase of the average staying time of cache devices. The MCKP based algorithm

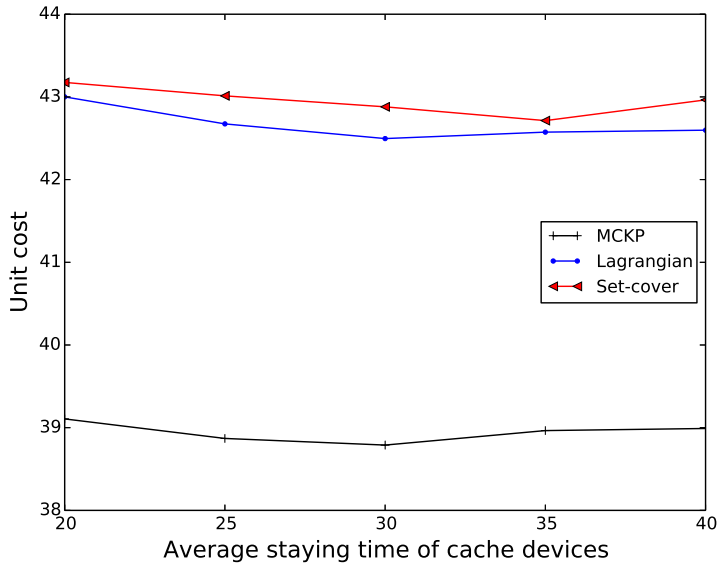


(a)

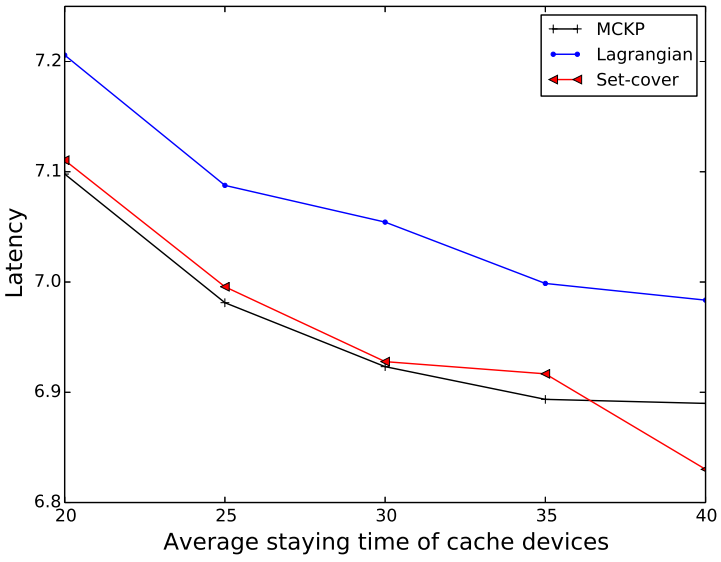


(b)

Fig. 4.10: Result when the average staying time of cache devices varies.



(c)



(d)

Fig. 4.10: Result when the average staying time of cache devices varies.

and WSCP based algorithm perform better than the Lagrangian relaxation based algorithm. The result in this scenario is very similar to the result in Fig. 4.9, because the increase of arrival rate or staying time of cache devices both increase the density of cache devices in the cell. The main difference is that the density increase with higher staying time of cache devices is not so evident as the direct increase with the larger arrival rate of cache devices. This causes the minor change in terms of total cost, which leads to almost invariant unit cost. However, the results for both scenarios in Fig. 4.9 and Fig. 4.10 demonstrate that with the increase of cache device density, the total cost decreases with a larger D2D offload ratio and a smaller average latency of all the finished transmissions.

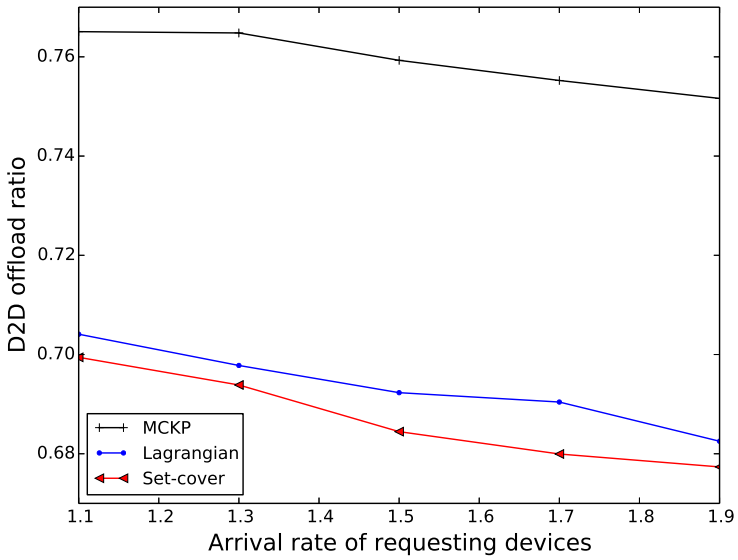
4.3.4 Dynamic Scenario with Different Settings of Requesting Devices

In this section, we examine the impact of the parameters associated with the requesting devices and their requests for messages. The rest of the parameters take the default values given in Table 4.1.

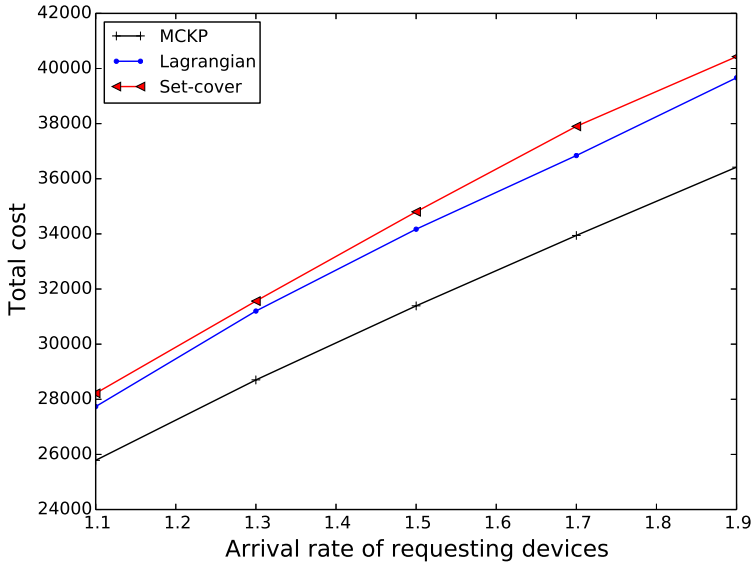
4.3.4.1 Different arrival rates of requesting devices

In this scenario, we change the arrival rate of requesting devices between 1.1 and 1.9. Fig. 4.11 shows the four performance metrics for this scenario. Fig. 4.11(a) shows that the D2D offload ratio of all three algorithms decreases with the increase of the arrival rate of requesting devices. This is because, when requesting devices arrive faster and generate more requests, the fixed number of cache devices cannot support all the requesting devices any longer. Thus, the D2D offload ratio decreases slowly.

Comparing the three message allocation algorithms, we can see that the D2D

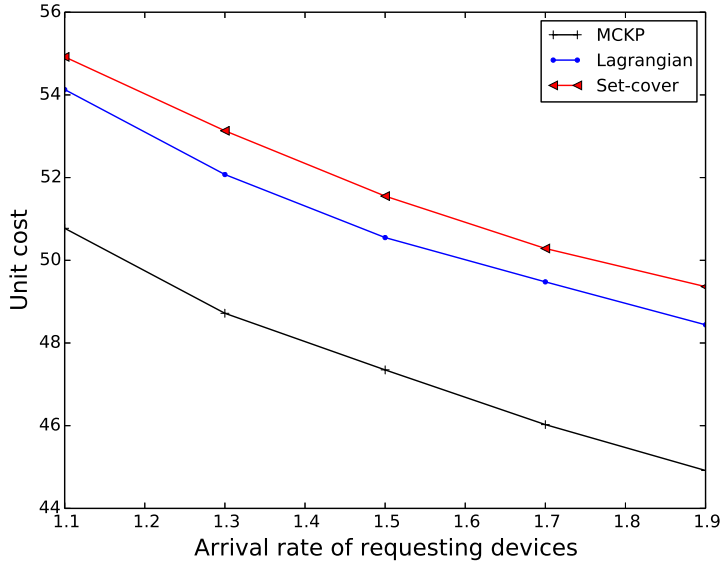


(a)

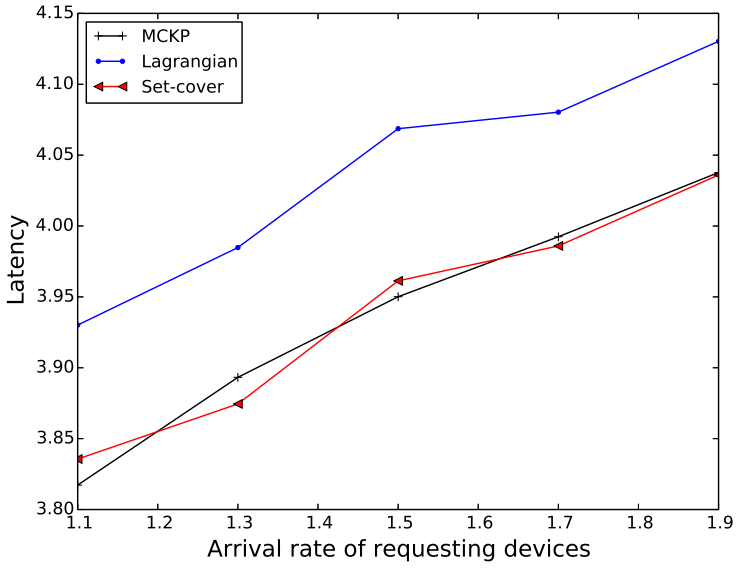


(b)

Fig. 4.11: Result when the arrival rate of requesting devices varies.



(c)



(d)

Fig. 4.11: Result when the arrival rate of requesting devices varies.

offload ratio of the MCKP based algorithm, the Lagrangian relaxation based algorithm, and the WSCP based algorithm presents an ascending order. The same order has been observed above in Fig. 4.9. As the arrival rates of cache devices and requesting devices basically changes the relative level between resource demands and supplies, the impact of increasing the arrival rate of cache devices is similar to that of decreasing the arrival rate of requesting devices.

Corresponding to the decrease of D2D offload ratio in Fig. 4.11(a), Fig. 4.11(b) shows the increase of total cost with a larger arrival rate of requesting devices. This is consistent with the design goal to reduce cost by offloading requests to D2D multicast. With the highest D2D offload ratio, the MCKP based algorithm is subject to the lowest total cost. Similarly, the WSCP based algorithm has the highest total cost due to its lowest D2D offload ratio.

Fig. 4.11(c) shows that the relative level of the three algorithms in unit cost is the same as observed in Fig. 4.11(b) for the total cost. However, it is interesting to find that the unit cost decreases when the total cost increases with the arrival rate of requesting devices. This observation can be explained as follows. First, as seen in Fig. 4.11(a), the decrease of D2D offload ratio is very minor. Hence, with a higher density of requesting devices, the amount of requests offloaded to D2D multicast actually increases substantially. Eventally, although the total cost increases, the unit cost still decreases as much more requests are satisfied.

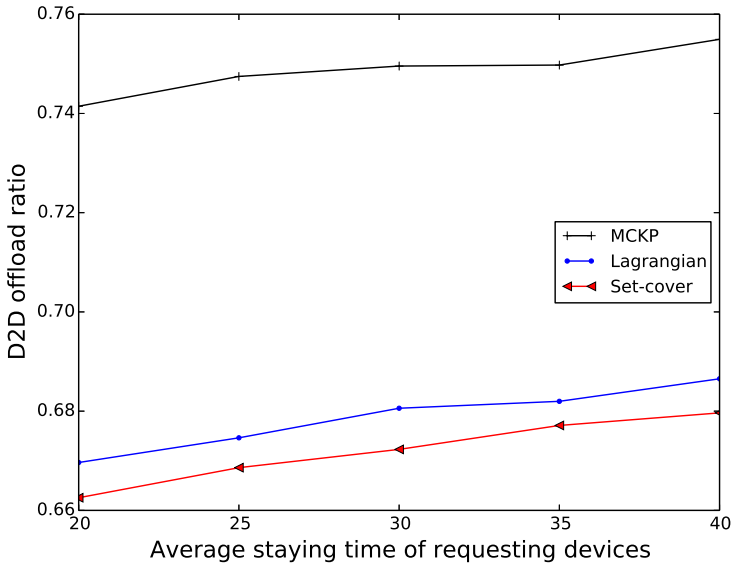
Fig. 4.11(d) shows that the latency of all three algorithms increases when the arrival of requesting devices is increasing. This totally makes sense since there are potentially more requests with the requesting devices. Due to more intense interference among the devices, the transmission rates of D2D multicast and BS unicast become lower. Nonetheless, the order of the three algorithms in terms of latency is the same as the order of them in Fig. 4.9(d).

4.3.4.2 Varying staying time of requesting devices

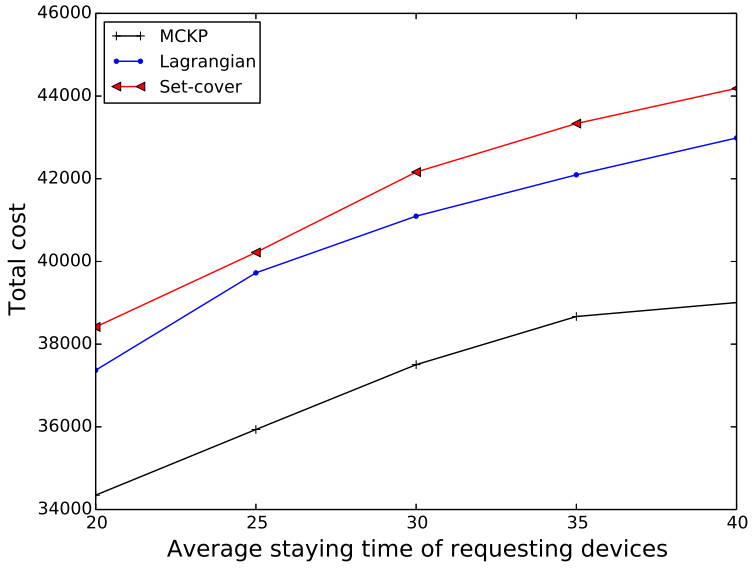
Fig. 4.12 shows the performance of the three message allocation algorithms when we change the average staying time of requesting devices between 20 and 40. First, Fig. 4.12(a) shows that the D2D offload ratio of all three algorithms increases when the average staying time of requesting devices is longer. Increasing the average staying time of requesting devices leads to higher density of requesting devices in the cell. However, the effect to the density is not so significant as the increase of the arrival rate of requesting devices. As a consequence, the density of requesting devices may not be high enough to saturate the cache devices. In addition, the longer staying time of requesting devices also enables more to successfully complete the message transmissions with D2D multicast while they stay within the cell. These factors together result in the minor increase of D2D offload ratio. As for the order of these three algorithms in terms of D2D offload ratio, it is the same as observed in Fig. 4.11(a).

Fig. 4.12(b) shows that the total cost of all three algorithms increases with the average staying time of requesting devices. The longer the requesting devices stay in the cell, the more requests they may generate. Hence, it is reasonable to observe the increasing total cost. In contrast, Fig. 4.12(c) shows that the unit cost decreases with the average staying time of requesting devices. This is due to the effect of two contradicting factors. On one hand, the longer staying time of requesting devices increases their density and results in more requests and higher total cost. On the other hand, the higher density also causes that each D2D multicast transmission can potentially satisfy more requests. The overall effect is the reduced unit cost with the increase of the average staying time of requesting devices.

Fig. 4.12(d) shows that the latency of all three algorithms is longer when the

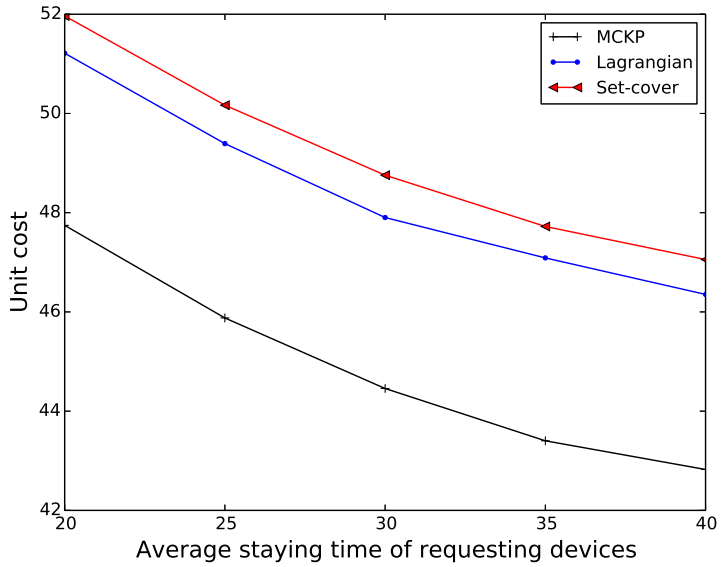


(a)

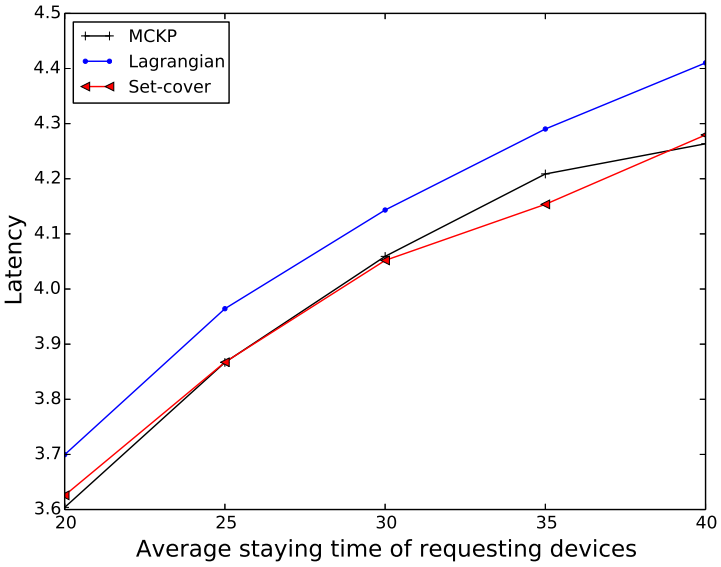


(b)

Fig. 4.12: Result when the staying time of requesting devices varies.



(c)



(d)

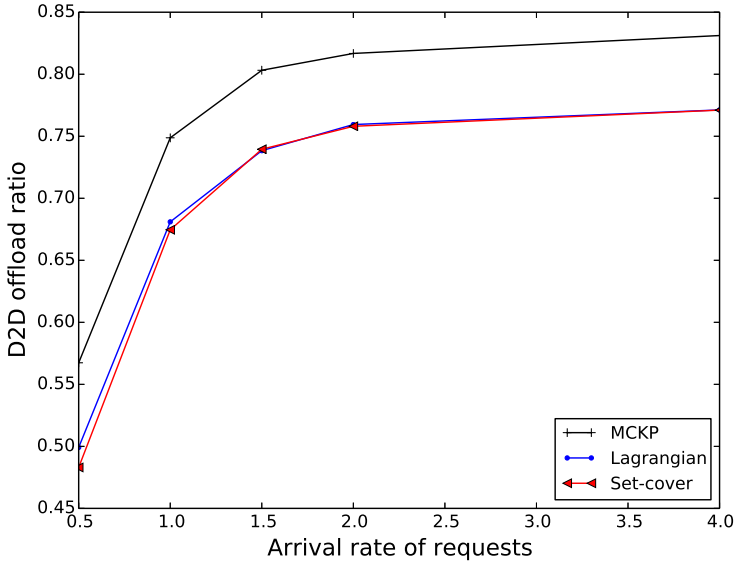
Fig. 4.12: Result when the staying time of requesting devices varies.

average staying time increases. The relative order of them is the same as the latency shown in Fig. 4.11(d). The increase of latency is due to more accumulated requests when the requesting devices stay in the cell for a longer time. Since each cache device can only send one message each time, more requests have to wait longer before receiving the messages.

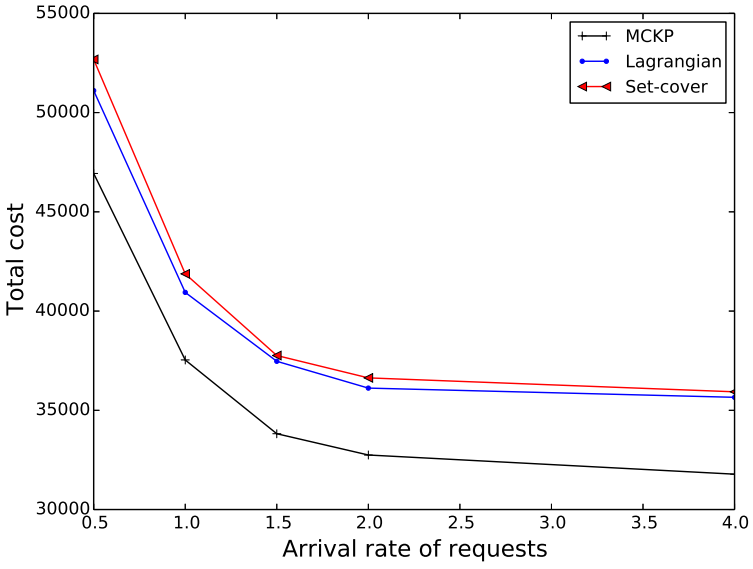
4.3.4.3 Different arrival rates of requests

In this scenario, we only change the arrival rate of requests for each requesting device between 0.5 and 4.0. The result of this scenario is presented in Fig. 4.13. Fig. 4.13(a) shows that the D2D offload ratio of the three algorithms increases with the the arrival rate of requests for each requesting device. The order of the three algorithms in terms of D2D offload ratio is the MCKP based algorithms as the highest, the Lagrangian relaxation based algorithm in the middle, and the WSCP based algorithm as the lowest. This is similar to the observation in Fig. 4.12(a). Moreover, when the arrival rate of requests is greater than 2, the increment of D2D offload ratio is almost invisible. It is because the number of requests has reached the capacity limit of the existing cache devices.

Fig. 4.13(b) shows that the total cost of all three algorithms decreases when the arrival rate of requests of each requesting device is increasing. This could be explained by referring to the results in terms of D2D offload ratio and latency. When the arrival rate of requests is increasing, the D2D offload ratio is increasing, which leads to the decrease of total cost. Meanwhile, although there are more requests, many of them do not finish successfully, which is confirmed by the rapidly rising latency when the arrival rate exceeds 1.0. As a consequence, the total cost even decreases.

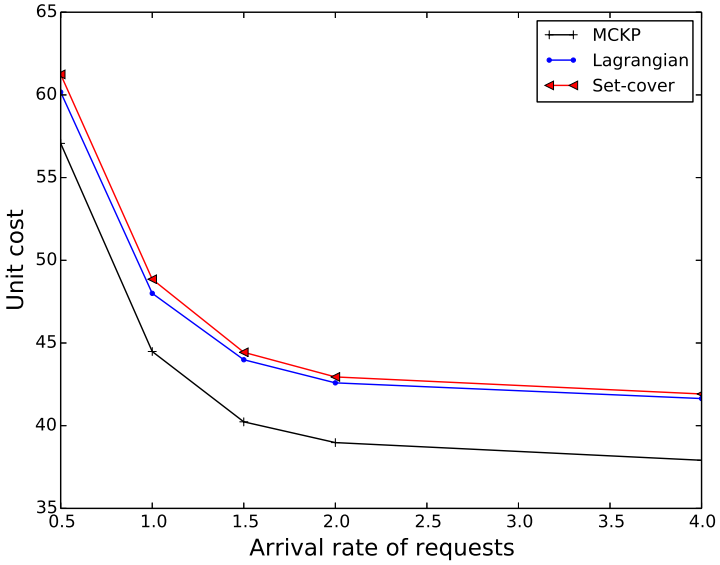


(a)

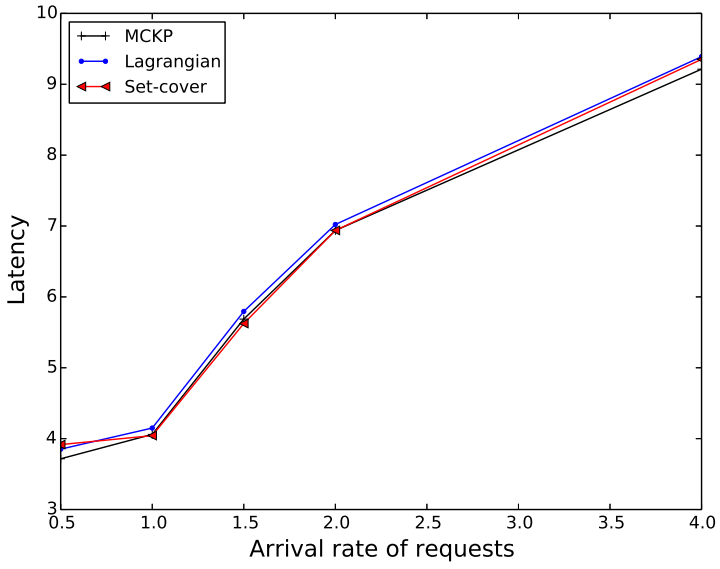


(b)

Fig. 4.13: Result when the arrival rate of requests varies.



(c)



(d)

Fig. 4.13: Result when the arrival rate of requests varies.

Fig. 4.13(c) shows that the unit cost decreases with the increase of arrival rate of requests. This is consistent with the result of D2D offload ratio in Fig. 4.12(a) and the result of total cost in Fig. 4.13(b). That is, higher D2D offload ratio leads to lower total cost since D2D multicast transmission is more cost-efficient. Moreover, it is seen that the decrease of unit cost almost stops at the point where the arrival rate of requests is 2. This is consistent with the observations in Fig. 4.12(a) and Fig. 4.13(b). Fig. 4.13(d) shows that a larger arrival rate of requests results in longer latency for all three algorithms. This increase is attributed to the accumulation effect of more requests as explained above for Fig. 4.12(d).

Chapter 5

Conclusion and Future Work

5.1 Conclusion

D2D communications can support many promising applications. Particulary, with D2D communications, certain devices can serve as the content providers to fulfill the content requests of other devices in close proximity with their cached content. The short transmission range can potentially achieve high data rates and low latency with low energy consumption. In addition, D2D-assisted content distribution can offload traffic from the cellular network, thus benefiting other non-D2D cellular users as well. In this thesis, we investigated two key problems for D2D-assisted collaborative content distribution, *i.e.*, the D2D pairing problem and the message allocation problem.

First, we studied the D2D pairing problem, which appropriately pairs a device requesting a content file with a nearby device which caches the requested file. The D2D pairing problem can be formulated as an integer linear program (ILP). Due to the complexity of the problem, we further developed a heuristic channel-aware D2D pairing algorithm. When the devices and requests arrive dynamically, a last-

come first-served waiting queue is used to keep the content requests unfulfilled by D2D communications so as to maximize the offloading traffic from the cellular network.

We conducted computer simulations to evaluate our proposed algorithm in both static and dynamic scenarios. In static scenarios, we compared our proposed algorithm with an optimal strategy for the ILP, which is based on exhaustive search but which is very time consuming. The results show that the proposed algorithm can achieve close performance in terms of satisfaction ratio as the optimal strategy when there are not a large number of devices. In addition, we compared the channel-aware algorithm with a minimum distance-based algorithm and a random algorithm in dynamic scenarios. The distance-based algorithm pairs a requesting device with its closest feasible cache device, while the random algorithm randomly selects a feasible cache device. The results show that our channel-aware algorithm outperforms the distance-based algorithm and the random algorithm in terms of total number of served D2D pairs and average latency of served pairs. Other than that, the cumulative distribution function of latency shows that most of the requests can be satisfied without high latency.

Second, we further investigated the message allocation problem, which determines the messages multicast by each cache device to serve the content requests via D2D communications. Multicast can save more energy for cache devices, but also raise more challenges to allocate the transmission messages for cache devices to fulfill the demands of requesting devices. On one hand, it is preferable that as much traffic as possible is relieved from the BS, so that the benefit of D2D communications is exploited to minimize the transmission cost. On the other hand, it is essential to mitigate the interference among D2D transmitters in order to simultaneously satisfy as many requests as possible.

The problem can be solved from different perspectives, aiming to minimize the total transmission cost or maximize the gain in cost saving for the BS. We evaluated three different algorithms that formulate the message allocation problem as a weighted set cover problem (WSCP), a hypergraph matching problem, or a multiple-choice knapsack problem (MCKP). The three problems can be solved by a greedy algorithm, a heuristic algorithm based on Lagrangian relaxation, and an FPTAS, respectively. The approximation ratio of the greedy algorithm is limited by the maximum cardinality of the subsets, while the FPTAS for the MCKP can be sufficiently close to the optimum by choosing the accuracy parameter. In contrast, the performance of the heuristic algorithm based on Lagrangian relaxation can vary with the datasets.

We conducted computer simulations to compare the three algorithms in static and dynamic scenarios. We focused on four performance metrics including D2D offload ratio, total cost, unit cost per message request, and service latency. The results show that in static scenarios the WSCP based algorithm outperforms the other two algorithms in terms of D2D offload ratio and total cost. This is because the WSCP formulation allows each cache device to be allocated multiple messages. However, even though each cache device can be allocated multiple messages, the messages have to be multicast one by one. As a result, in dynamic scenarios, the WSCP based algorithm becomes the worst in terms of all performance metrics. In contrast, the results reveal that the MCKP based algorithm achieves the best performance in terms of D2D offload ratio and total cost. This is because the FPTAS is used to solve the MCKP formulation and the approximation guarantee of the FPTAS leads to solutions closest to the optimum. Consequently, it can find better message allocation that diverts more requests to be successfully fulfilled by D2D multicast.

5.2 Future Work

There are some interesting research directions to extend this work. First, the latency of D2D transmissions can be further improved by considering some advanced queueing mechanism to reduce the waiting time of pending requests. One idea is to take into account estimated mobility of cache devices and requesting devices. For example, the D2D transmission between of a pair of high-mobility devices is less likely to complete successfully if the waiting time is too long. Hence, it is preferable to prioritize such requests in the waiting queue. However, a main challenge for this idea to obtain accurate mobility estimation for the devices according to their historical data.

In the D2D pairing problem, we focused on one-to-one matching between cache devices and requesting devices. When large-sized messages are considered, *e.g.*, video clips, it would be more beneficial if multiple cache devices can be selected to serve one message request. This can improve the transmission latency and even support real-time streaming during data transmission. However, this turns the D2D pairing problem from one-to-one matching to more complex one-to-many matching. To simplify the new problem, we can limit the possible combinations of cache devices that each requesting device can be assigned to. Even so, this problem can be reduced to an NP-hard set packing problem if we aim to maximize the number of satisfied requesting devices.

Another direction that deserves more in-depth exploration is to further consider a cost budget for each individual cache device for the multicast message allocation problem. In the MCKP formulation, we only restrict the total cost of all cache devices, which can limit the expense of the BS for utilizing external resources. For battery-powered devices, each device may have a maximum amount of energy that it is willing to devote into collaborative content distribution to protect its own

service requirements. This energy amount can be translated into an individual cost budget. Correspondingly, we need to add more constraints to the MCKP formulation. Then, each cache device should be allocated the multicast messages so as to maximize the utilization of its budget while satisfying the overall cost budget.

Bibliography

[1] X. Lin, J. G. Andrews, A. Ghosh, and R. Ratasuk, "An overview of 3gpp device-to-device proximity services," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 40–48, 2014.

[2] M. Bansal and V. Venkaiah, "Improved fully polynomial time approximation scheme for the 0-1 multiple-choice knapsack problem," *International Institute of Information Technology Tech Report*, 2004.

[3] Y. Zhu, J. Jiang, B. Li, and B. Li, "Rado: A randomized auction approach for data offloading via D2D communication," in *Proc. IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, October 2015.

[4] L. Song, D. Niyato, Z. Han, and E. Hossain, "Game-theoretic resource allocation methods for device-to-device (D2D) communication," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 136–144, 2014.

[5] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014.

[6] Z. Zhou, M. Dong, K. Ota, R. Shi, Z. Liu, and T. Sato, "Game-theoretic approach to energy-efficient resource allocation in device-to-device underlay

communications," *IET Communications*, vol. 9, no. 3, pp. 375–385, February 2015.

[7] Y. Xiao, K.-C. Chen, C. Yuen, Z. Han, and L. A. DaSilva, "A Bayesian overlapping coalition formation game for device-to-device spectrum sharing in cellular networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 7, pp. 4034–4051, July 2015.

[8] Y. Li, D. Jin, J. Yuan, and Z. Han, "Coalitional games for resource allocation in the device-to-device uplink underlaying cellular networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 7, pp. 3965–3977, July 2014.

[9] M. Tehrani, M. Uysal, and H. Yanikomeroglu, "Device-to-device communication in 5G cellular networks: Challenges, solutions, and future directions," *IEEE Communications Magazine*, pp. 86–92, 2014.

[10] D. Camps-Mur, A. Garcia-Saavedra, and P. Serrano, "Device-to-device communications with Wi-Fi Direct: Overview and experimentation," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 96–104, 2013.

[11] A. Asadi and V. Mancuso, "WiFi Direct and LTE D2D in action," in *Proc. IFIP Wireless Days (WD)*, 2013, pp. 1–8.

[12] S. Balraj, "LTE direct overview," Qualcomm Research, Tech. Rep.

[13] Qualcomm Technologies, Inc., "LTE Direct always-on device-to-device proximal discovery," 2014.

[14] S. Andreev, A. Pyattaev, K. Johnsson, O. Galinina, and Y. Koucheryavy, "Cellular traffic offloading onto network-assisted device-to-device connections," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 20–31, 2014.

[15] L. Militano, M. Condoluci, G. Araniti, A. Molinaro, A. Lera, and F. Fitzek, "Wi-Fi cooperation or D2D-based multicast content distribution in LTE-A: A comparative analysis," in *Proc. IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2015, pp. 1–6.

[16] V. Siris and D. Dimopolous, "Multi-source mobile video streaming with proactive caching and D2D communication," in *Proc. IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2015, pp. 1–6.

[17] X. Wang, M. Chen, T. Taleb, and A. Ksentini, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Communications Magazine*, vol. 52, pp. 131–139, February 2014.

[18] W. Song and Y. Zhao, "A randomized reverse auction for cost-constrained D2D content distribution," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, December 2016.

[19] J. Jiang, S. Zhang, B. Li, and B. Li, "Maximized cellular traffic offloading via device-to-device content sharing," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 1, pp. 82–91, January 2016.

[20] B. Chen and C. Yang, "Energy costs for traffic offloading by cache-enabled D2D communications," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, April 2016.

[21] 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 Transactions on Wireless Communications*, vol. 13, no. 7, pp. 3665–3676, July 2014.

[22] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I Tube, You Tube, Everybody Tubes: Analyzing the world's largest user generated content video system," in *Proc. ACM SIGCOMM Conference on Internet Measurement*, October 2007, pp. 1–14.

[23] G. Mao, B. Fidan, and B. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, July 2007.

[24] Z. C. G. Gokeda and Y. Yu, *Introduction to Direction-of-Arrival Estimation*. Artech House, 2010.

[25] N. Lee, X. Lin, J. Andrews, and R. Heath, "Power control for D2D underlaid cellular networks: Modeling, algorithms and analysis," *IEEE Journal on Selected Areas in Communications*, pp. 1–13, 2014.

[26] W. Song, P. Ju, A. Jin, and Y. Cheng, "Distributed opportunistic two-hop relaying with backoff-based contention among spatially random relays," *IEEE Trans. Veh. Technol.*, vol. 64, no. 5, pp. 2023–2036, 2015.

[27] N. E. Young, "Greedy set-cover algorithms," in *Encyclopedia of Algorithms*. Springer, 2008, pp. 379–381.

[28] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Springer, 2004, ch. The multiple-choice knapsack problem, pp. 317–347.

Vita

Candidate's full name: Jianguo Xie

University attended:

September 2011 – June 2015
Bachelor of Computer Science and Technology
School of Computer Science and Technology
University of Science and Technology of China
Hefei, Anhui, China

September 2015 – May 2017
Master of of Computer Science
Faculty of Computer Science
University of New Brunswick
Fredericton, New Brunswick, Canada

Publications:

1. J. Xie and W. Song, "Channel-aware device-to-device pairing for collaborative content distribution," in *Proc. IEEE Wireless Communications and Networking Conference* (WCNC17), San Francisco, California, April 2017.

2. J. Xie, W. Song, and X. Tao, "Study of multicast message allocation for content distribution with device-to-device communications," accepted to *IEEE Vehicular Technology Conference* (VTC17 Fall), Toronto, ON, Canada, September 2017.

3. J. Xie and W. Song, "Collaborative message distribution via device-to-device (D2D) communications," Poster in *Faculty of Computer Science Annual Research Expo*, University of New Brunswick, Fredericton, NB, Canada, April 2017.

4. J. Xie and W. Song, "Channel-aware device-to-device pairing for collaborative content distribution," Poster in *Canadian FloodNet AGM*, Toronto/Vaughan, ON, Canada, September 2016.