Université du Québec
Institut national de la recherche scientifique
Centre Énergie, Matériaux et Télécommunications

EDGE CACHING AND NETWORK SLICING FOR WIRELESS CELLULAR NETWORKS

Par

DUY THINH TRAN

Thèse présentée pour l'obtention du grade de
Doctorat en philosophie, Ph.D.
en télécommunications

Jury d'évaluation

Examinateur externe	Prof. Thi-Mai-Trang Nguyen *Sorbonne Université, Paris, France*
	Prof. Huan Nguyen *Middlesex University, London, United Kingdom*
Examinateur interne	Prof. André Girard *INRS-ÉMT*
Directeur de recherche	Prof. Long Bao Le *INRS-ÉMT*

To my parents Trần Hiệp Thành & Đặng Thị Kim Vân
To my wife Trần Thị Nhung
To my son Trần Duy Phát Andy

Acknowledgments

I would like to gratefully acknowledge and express a sincere thank you to my supervisor, Professor Long Bao Le for giving me the opportunity to pursue doctoral study at INRS-ÉMT, University of Québec. I am truly privileged to have learned from his remarkable technical knowledge and research enthusiasm. Since the very beginning, he has always pointed me in good research directions and encouraged me to pursue them to concrete results. His invaluable support and guidance during my study have certainly helped me complete this Ph.D. dissertation. I would like to express my gratitude to other members of my Ph.D. committee – Professor André Girard of INRS-ÉMT, University of Québec who has regularly reviewed and constructively commented on the progress of my doctoral study. I would also like to thank Professor Thi-Mai-Trang Nguyen of Sorbonne Université (France) and Professor Huan Nguyen of Middlesex University (UK) for serving as the external examiner to my Ph.D. dissertation.

I would like to express gratitude to all my colleagues for the wonderful and memorable time at the Networks and Cyber Physical Systems Lab (NECPHY-Lab), INRS-ÉMT, University of Québec: Vu Ha, Tuong Hoang, Tan Le, Hieu Nguyen, Dai Nguyen, Tam Tran, Ti Ti Nguyen, Thanh-Dung Le, Tri Nguyen, Hoang Vu, Tung Phan, Duy Nguyen, Dat Nguyen, Thong Vo, and Miao Wang.

Finally, my deepest love and gratitude are devoted to all of my family members: Mom and Dad, beloved wife and son, Parents-in-laws, and my sister Tram Ngoc Tran, who always support me in each and every endeavor in my life. My Ph.D. study would not be finished without the constant and unconditional support from my family. I thank you all and hope that I made you proud of my accomplishments.

Abstract

The fifth-generation (5G) wireless cellular system is expected to provide huge improvement in comparison to the fourth-generation (4G) system in supporting more stringent and versatile technical requirements. Particularly, the 5G system should be capable of providing a 1000-fold of network throughput, supporting ultra reliable and low latency communications, and handling massive connectivity. Novel techniques must be devised and well integrated to enable the 5G wireless cellular system fulfill such stringent key performance requirements of diverse wireless applications and to significantly reduce the capital expenditure (CAPEX) and operational expenditure (OPEX) for 5G cellular network operators. Wireless network virtualization (WNV), or network slicing, has been considered as a promising networking approach for addressing this problem. Efficient techniques for advanced resource management of network slices must be developed to achieve high resource utilization efficiency and flexibility while satisfying each slice's quality of service (QoS) constraints.

Harnessing new kind of resources (i.e., new resource dimensions) is essential to help the future 5G wireless cellular system satisfy these stringent technical requirements in addition to exploiting new spectrum bands (e.g., millimeter wave (mmWave)) and techniques for improving spectrum and energy efficiency. Moreover, content caching at the network edge, i.e, placing popular contents or files at places closer to end users can potentially help significantly reduce network traffic and access delay considering the rapid increase of mobile video data in the wireless cellular network. The 5G network performance would be further improved if different types of network resources such as frequency spectrum, transmission power, and storage resources were efficiently utilized and managed. As a result, it is crucial to design frameworks for network resource management and content placement. Motivated by these promising key directions, the general objective of this Ph.D. research is to develop efficient resource management techniques enabling wireless edge caching and network virtualization in the wireless cellular network. Our research has resulted in three major research contributions, which are presented in three corresponding chapters of this dissertation.

First, we study the caching problem for heterogeneous small-cell networks with bandwidth allocation and caching-aware base station (BS) association. The caching control and bandwidth allocation problem aims at minimizing the request miss rate for one network operator who has a limited bandwidth and storage capacity in serving its end users (UEs). To solve this problem, we propose a Line-Search-based-Iterative (LSBI) algorithm which determines the solution by combining the line-search algorithm to obtain the optimal bandwidth allocation with the iterative caching algorithm to acquire a caching solution. Numerical results demonstrate that the LSBI algorithm significantly outperforms existing caching algorithms, and is on a par with a performance bound.

Second, we investigate the joint resource allocation and content caching problem which aims to efficiently utilize the radio and content storage resources in multi-cell virtualized wireless network with highly congested backhaul links. In this design, we minimize the maximum content request rejection rate experienced by users of different mobile virtual network operators (MVNO) who share a common resource pool of subcarriers and storage repositories owned by an infrastructure provider (InP). We solve the resulting mixed-integer non-linear programming (MINLP) problem by proposing a bisection-search based algorithm that iteratively optimizes the resource allocation and content caching placement. We further propose a low-complexity heuristic algorithm which achieves moderate performance loss compared to the bisection-search based algorithm. Extensive numerical results confirm the efficacy of our proposed framework which significantly reduces the maximum request outage probability compared to other benchmark algorithms.

Third, we study the resource allocation and pricing problem in the virtualized wireless network that captures the multilateral interactions among access/backhaul service providers and their UEs by using the multi-leader-multi-follower (MLMF) Stackelberg game approach. Toward this end, we show how to formulate such a Stackelberg game and prove the existence of a unique game equilibrium. Then, we develop a distributed algorithm based on updating underlying best-response functions, which is proved to converge to the game equilibrium. Numerical results are presented to provide important insights into the interactions among the involved stakeholders and demonstrate the economical efficacy of the proposed design with respect to existing benchmarks.

In summary, different efficient resource management algorithms have been developed considering several enabling 5G wireless technologies. Moreover, extensive numerical results are presented in each contribution to gain further insights and to evaluate the performance of our proposed designs. The solid results achieved in this dissertation would form good foundations for our future studies where research issues such as mobility management, security and privacy, and applications of machine learning techniques for more effective network management can be addressed.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

2.s.s	Two-sided scalability
4G	Fourth-Generation
5G	Fifth-Generation
AR	Augmented reality
ASP	Access service provider
BA	Bandwidth allocation
BS	Base station
BSP	Backhaul service provider
C-RAN	Cloud radio access networks
CAPEX	Capital expenditure
CN	Core network
D2D	Device-to-device
eMBB	Enhanced mobile broadband
HetNets	Heterorgeneous networks
InP	Infrastructure provider
LSBI	Line-search-based-iterative
MBS	Macro-cell base station
MIMO	Multi input multi output
MINLP	Mixed-integer non-linear program
MLMF	Multi-leader-multi-follower
mMTC	Massive machine type communications
multi-RAT	Multi radio access technologies
MVNO	Mobile virtual network operator

NFV	Network functions virtualization
OFDMA	Orthogonal frequency-division multiple access
OPEX	Operational expenditure
OS	Operating system
QoS	Quality of service
RA	Resource allocation
RAN	Radio access network
RRH	Remote radio head
SBS	Small-cell base station
SDN	Software-defined networking
SDR	Software defined radio
SE	Stackelberg equilibrium
SLA	Service level agreement
SLMF	Single-leader-multiple-follower
SNR	Signal-to-noise ratio
UDN	Ultra dense networks
UE	User equipment
uRLLC	Ultra reliable and low latency communications
VM	Virtual machine
VNF	Virtual network functions
VWN	Virtualized wireless networks
WBN	Wireless backhaul networks
WNV	Wireless network virtualization

Chapter 1

Extended Summary

1.1 Background and Motivation

The fifth-generation (5G) wireless cellular system, which has started rolling out by 2020, is expected to provide a huge network performance improvement and to support new services and applications, compared to those enabled by the current fourth-generation (4G) system [2–4]. Specifically, 1000-fold increase of network throughput compared to that of 4G systems is the target spec promised by the 5G systems [5]. This significant network capacity increment is for coping with the ever-increasing mobile traffic generated from enhanced mobile broadband (eMBB) services such as mobile video streaming [6–9]. In addition to support the eMBB service type, the future 5G wireless cellular system also supports the other two key service types, namely ultra-reliable low-latency communications (uRLLC) and massive machine type communications (mMTC) for respectively serving mission-critical applications and a massive number of simultaneous connections from wireless devices [8, 10, 11]. Accordingly, a hefty burden of network traffic as well as stringent requirements are put on both the radio access network (RAN) and the backhaul network, which establish end-to-end connections between user equipments (UEs) and core network (CN) via base stations (BSs). New techniques and novel network architectures must be devised and well incorporated together to enable the 5G wireless cellular system to fulfill such stringent and versatile requirements [3, 4, 7, 8].

1.1.1 Advanced Resource Utilization and Management

Exploiting new radio spectrum bands [12] and enhancing spectrum efficiency are two necessary and complementary approaches for network throughput and network quality improvement. Moreover, leveraging other kind of resources, especially storage repository, is a promising approach to reduce communication delay and relieve traffic congestion [3]. In fact, by deploying storage devices at BSs in the network and pre-fetching popular content/files, which is also referred as *content caching*, [1] to these storage repositories, one can bring popular contents in closer proximity to UEs. As a result, the traffic in the backhaul links induced by accessing these contents, which are usually stored in the CN if they are not cached at the BSs, is also relieved significantly [13]. By doing so, the end-to-end access latency to these contents is reduced, thus improving users' quality of service (QoS) [14–16].

Innovations in enhancing the spectrum efficiency and leveraging emerging resource dimensions, typically the content caching, are most beneficial if they are engineered jointly with other resource management frameworks [3, 15, 16]. Yet designing advanced resource management frameworks that can utilize the advantages of both the spectrum efficiency enhancement and content caching is challenging and requires much more further research.

1.1.2 Network Slicing - Wireless Network Virtualization

To help the 5G network meet stringent requirements of its diverse service types, it is crucial to design innovative network architectures (e.g., ultra dense networks (UDN) [17, 18], cloud radio access network (C-RAN) [19]) that not only integrate advanced technologies but also make them work together in a seamless way. However, deploying and operating these novel network architectures for 5G systems as well as integrating innovative technologies into these systems require a massive overhaul in network infrastructure, both in the air interface and in the backhaul network. Such requirements can incur a surcharge of capital expenditure (CAPEX) and operation costs (OPEX), as well as slowing down the deployment time of new technologies and network services [7, 8]. Wireless network virtualization (WNV), also known as *network slicing*, has been considered as a promising networking paradigm for addressing this problem [20].

[1] We use the term content and file interchangeably in this doctoral dissertation.

In fact, wireless virtualization allows multiple mobile virtual network operators (MVNOs), also known as service providers (SPs), to share the same network infrastructure and a common resource pool owned and managed by one (or several) infrastructure provider(s) (InP). On this common network infrastructure, the InP is in charge of flexibly and efficiently allocating network resources to MVNOs based on their contracts. Each MVNO in turn uses the rented resources and infrastructure to provide its own services [21] to its clients including UE or other MVNOs with committed quality of service (QoS). Accordingly, network slicing helps network operators and SPs reduce CAPEX and OPEX by utilizing network resource in a flexible and efficient manner while better meeting the required QoS [20]. Thanks to scalable and flexible characteristics, network slicing also expedites technology implementation and integration into 5G wireless cellular networks [20].

1.1.3 Economic Aspects of Resource Sharing Among 5G Network Tenants

Network slicing is an important technology for which the monolithic network can be virtually sliced into multiple network slices to support specialized wireless services. Appropriately designed network slices, for instance, could be designated for the high-speed streaming services such as YouTube and Netflix, or the uRLLC services for the factory control applications [22]. Network slicing also provides a paradigm shift toward multi-tenancy in the next-generation wireless network [23] where individual tenants (e.g., MVNOs, SPs) own and manage corresponding network slices. By enabling service trading among tenants, this paradigm shift offers greater business opportunities and greater savings in CAPEX and OPEX [22].

Accordingly, there exists multilateral interactions between SPs, InPs, and UEs regarding the economics aspect. Here, the interaction can be an economic competition between the SPs providing same service type to a market, or it can be a buy-and-sell interaction between the SPs and their customers such as UEs. These multilateral interactions among the SPs and their customers constitute to a resource trading market. Designing an appropriate framework for operating such market is crucial for achieving efficient network serviceability and high profits. Game theory is an appropriate tool for modeling such market [24].

1.2 Research Contributions

This Ph.D. research focuses on three main objectives. First, we develop a joint radio resource allocation and content caching framework under small-cell heterogeneous network (HetNet) setting, where we consider the resource allocation and content caching problem for a single network operator with its own resource pool. Second, we study the joint resource allocation and content caching in the virtualized multi-cell network environment where multiple network operators sharing a common resource pool of wireless channels and storage repositories under the coordination of a centralized controller. Third, we consider resource allocation problem concerning the multilateral interactions between SPs as well as between the SPs and their customers in a network slicing setting by using the Stackelberg game theory. All of the objectives aim to directly address important technical issues of future network scenario (HetNets) and emerging network paradigms (network slicing). The main contributions of this Ph.D. dissertation are as follows:

1.2.1 Caching for Heterogeneous Small-Cell Networks with Bandwidth Allocation and Caching-Aware BS Association

In this contribution, we study the caching problem for heterogeneous small-cell networks with bandwidth allocation and caching-aware BS association. There are some existing works that study the caching problem for small-cell networks [15, 16] where they investigate the joint caching, routing, and channel assignment. However, all these works do not consider the stochastic behavior of content request and service processes. Meanwhile, authors in [25, 26] study the joint caching and resource allocation design based on the time-varying signal-to-noise ratio (SNR). This design would require frequent cache updates, which is not cost efficient because the SNR usually varies quickly over time. Accordingly, a general caching design for HetNets where mobile users can be associated with either a small-cell base station (SBS) or macro-cell base station (MBS) and allocated radio resource to download their desired contents should be considered. BS associations in such the heterogeneous network should take into caching decisions (i.e., being caching-aware) where users should associate with BSs which have favorable channel conditions and store their requested contents.

Motivated by the aforementioned issues, we design a joint content caching and bandwidth allocation problem for HetNets where we make the following key contributions.

- We design a joint content caching and bandwidth allocation framework for minimizing the request miss ratio.
- We propose a Line-Search-based-Iterative (LSBI) algorithm which determines the solution by combining the line-search algorithm to obtain the optimal bandwidth allocation with the iterative caching algorithm to acquire a caching solution.
- Numerical results demonstrate that the LSBI algorithm significantly outperforms existing caching algorithms, and is on a par with a performance bound.

1.2.1.1 System Model

We consider a heterogeneous small-cell caching system consisting of one MBS denoted as BS 0 and S non-overlapping SBSs in the set $\mathcal{M}_s = \{1, \cdots, S\}$ deployed within the coverage area of the MBS. Let $\mathcal{M} = \{0\} \cup \mathcal{M}_s$ denote the set of all BSs. We assume that the system bandwidth B is assigned orthogonally to the MBS and SBSs, and all SBSs reuse the same bandwidth. Let B_0 and B_s respectively denote the bandwidth assigned to the MBS and all SBSs, where $B_0 + B_s \leq B$ and we denote $\boldsymbol{B} = [B_0, B_s]$.

Let w_m be the bandwidth required to serve a user in BS $m \in \mathcal{M}$. Let $\boldsymbol{K} = [K_0, \cdots, K_m, \cdots, K_M]$ be the service capacity of the system, where K_m represents the maximum number of users that can be served simultaneously by BS $m \in \mathcal{M}$. We also denote $\boldsymbol{K} = [\boldsymbol{K}_0, \bar{\boldsymbol{K}}_0]$, where $\bar{\boldsymbol{K}}_0 = [K_1, \cdots, K_M]$. Then, to maintain the required users' QoS in cell m, K_m should satisfy $K_m \leq B_s/w_m \ \forall m \in \mathcal{M}_s$, $K_0 \leq B_0/w_0$, and $K_m \in Z^+$, where Z^+ denotes the set of non-negative integers.

We consider the following adaptive caching-aware BS association strategy. As user k in the coverage area of SBS m requests a file, the SBS will serve the user (i.e., user k will be associated with SBS m) if it is serving less than K_m users and the file is currently cached at the SBS. Otherwise, the request is redirected to the MBS. At the MBS, if the requested file is available in its cache and the MBS is serving less than K_0 users, the request will be served (i.e., user k will switch its association to the MBS). Otherwise, the request is missed.

We assume that users request files in set $\mathcal{F} = \{f_1, \cdots, f_F\}$. These files are assumed to have the same size and can be stored in the caches of the BSs for future downloads. We assume that the popularity distributions of the files in $\mathcal{F}$ depend on the service area where users in different areas

can have different file preferences. Let $\boldsymbol{p}_m = [p_{m1}, \cdots, p_{mF}]$ denote the file request probabilities of users in the coverage area of BS $m \in \mathcal{M}$ where p_{mf} denotes the probability that file f is requested by some user in the coverage area of BS m and $||\boldsymbol{p}_m||_1 = 1 \ \forall m \in \mathcal{M}$. We assume that content requests in BS $m \in \mathcal{M}$ follow the Poisson process with average rate λ_m(requests/s). We assume that $\boldsymbol{p}_m$ and λ_m are known. Finally, we assume that it takes T_m seconds for BS m to serve one request (i.e., the file download time).

Let $\boldsymbol{x}_m = [x_{m1}, \cdots, x_{mF}]$ and $\boldsymbol{x} = [\boldsymbol{x}_0, \cdots, \boldsymbol{x}_S]$ represent the caching decisions of BS m and all BSs, respectively. Specifically, $x_{mf} \in \{0, 1\}$ denotes the caching status of file f at SBS m, where $x_{mf} = 1$ means that file f is cached at BS m, $x_{mf} = 0$, otherwise. We also denote the caching vector of all BSs in the system as $\boldsymbol{x} = (\boldsymbol{x}_s, \bar{\boldsymbol{x}}_s)$ where $\boldsymbol{x}_s$ and $\bar{\boldsymbol{x}}_s$ are the caching vectors of BS $s \in \mathcal{M}$ and other BSs, respectively.

1.2.1.2 Problem Formulation

We first analyze the caching performance of a particular SBS, which is used in the problem formulation. Since the request rate associated with SBS $m \in \mathcal{M}_s$ is λ_m, the request rate for file f at SBS m is $\lambda_m p_{mf}$. If file f is not cached at SBS m, the request is redirected to the MBS. Denote $\lambda_{mf}^{\mathsf{redb}}(\boldsymbol{x}_m)$ as the redirected rate to the MBS from SBS m for file f due to the unavailability of file f in the cache. Then, $\lambda_{mf}^{\mathsf{redb}}(\boldsymbol{x}_m)$ can be expressed as $\lambda_{mf}^{\mathsf{redb}}(\boldsymbol{x}_m)$=$\lambda_m p_{mf}(1 - x_{mf})$. Consequently, the average request rate for all files to SBS m can be calculated as

$$\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m) = \sum_{f \in \mathcal{F}} \lambda_m p_{mf} x_{mf}. \tag{1.1}$$

Note that the aggregate request follows the Poisson process because all individual request processes are Poisson [27]. Recall that SBS m can serve at most K_m users simultaneously and it takes T_m (s) to serve one request. Therefore, we can model the request/service at SBS m as an $M/D/K_m/K_m$ queue, which has Poisson arrivals, deterministic service time, K_m servers, and zero-length buffer. Hence, the probability a request being blocked [27] at SBS m can be expressed as

$$P_m^r(\boldsymbol{x}_m, K_m) = \frac{(\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)T_m)^{K_m}}{K_m!} \left(\sum_{i=0}^{K_m} \frac{(\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)T_m)^i}{i!} \right)^{-1}. \tag{1.2}$$

Note that if the request for file f is blocked by SBS m due to its limited service capability, the request is redirected to the MBS. Denote $\lambda_{mf}^{\mathsf{reda}}(\boldsymbol{x}_m, K_m)$ as the redirected request rate to the MBS from SBS m due to its limited service capability then this parameter can be calculated as $\lambda_{mf}^{\mathsf{reda}}(\boldsymbol{x}_m, K_m)=\lambda_{mf}^{\mathsf{req}}(\boldsymbol{x}_m)P_m^r(\boldsymbol{x}_m, K_m)$. As all requests which are rejected by SBSs due to either the limited service capability or the unavailability of requested files at the SBSs' caches are redirected to the MBS, we can calculate the total request rate of file f redirected to the MBS as $\lambda_f^{\mathsf{red}}(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{K}}_0)=\sum\limits_{m\in\mathcal{M}_s}\left(\lambda_{mf}^{\mathsf{redb}}(\boldsymbol{x}_m)+\lambda_{mf}^{\mathsf{reda}}(\boldsymbol{x}_m, K_m)\right)$. Therefore, the total request rate of file f to the MBS including original and redirected requests can be expressed as $\lambda_{Mf}^{\mathsf{req}}(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{K}}_0)=\lambda_0 p_{0f}+\lambda_f^{\mathsf{red}}(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{K}}_0)$.

The request miss rate associated with the MBS due to the unavailability of the files can be expressed as $\lambda_M^{\mathsf{rb}}(\boldsymbol{x}, \bar{\boldsymbol{K}}_0) = \sum_{f\in\mathcal{F}}\lambda_{Mf}^{\mathsf{req}}(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{K}}_0)(1-x_{0f})$. Consequently, the request rate for all files at the MBS can be calculated as

$$\lambda_M^{\mathsf{reqa}}(\boldsymbol{x}, \bar{\boldsymbol{K}}_0) = \sum_{f\in\mathcal{F}}\lambda_{Mf}^{\mathsf{req}}(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{K}}_0)x_{0f}. \tag{1.3}$$

Note that the requests (with the rate $\lambda_{mf}^{\mathsf{reda}}(\boldsymbol{x}_m, K_m)$) redirected from the SBS m to the MBS due to the limited service capability at this SBS form an overflow traffic, which is a non-Poisson process. One can replace this overflow process by an Poisson approximation using several techniques such as *Hayward's approximation* and *equivalent random method* [28]. By doing so, we can model content request/service at the MBS as an $M/D/K_0/K_0$ queue with an input Poisson process. Consequently, the request miss probability due to limited serving capability of the MBS can be calculated similar to (1.2), i.e., $P_0^r(\lambda_M^{\mathsf{reqa}}((\boldsymbol{x}, \bar{\boldsymbol{K}}_0), \boldsymbol{K})) = \frac{(\lambda_M^{\mathsf{reqa}}(\boldsymbol{x}, \bar{\boldsymbol{K}}_0)T_0)^{K_0}}{K_0!}\left(\sum\limits_{i=0}^{K_0}\frac{(\lambda_M^{\mathsf{reqa}}(\boldsymbol{x}, \bar{\boldsymbol{K}}_0)T_0)^i}{i!}\right)^{-1}$. As a result, the request miss rate due to the limited serving capability of the MBS can be expressed as $\lambda_M^{\mathsf{ra}}(\boldsymbol{x}, \boldsymbol{K}) = \lambda_M^{\mathsf{reqa}}(\boldsymbol{x}, \bar{\boldsymbol{K}}_0)P_0^r(\lambda_M^{\mathsf{reqa}}((\boldsymbol{x}, \bar{\boldsymbol{K}}_0), \boldsymbol{K}))$. Finally, the total request miss rate of the system can be calculated as $\lambda^{\mathsf{miss}}(\boldsymbol{x}, \boldsymbol{K})=\lambda_M^{\mathsf{ra}}(\boldsymbol{x}, \boldsymbol{K})+\lambda_M^{\mathsf{rb}}\left(\boldsymbol{x}, \bar{\boldsymbol{K}}_0\right)$ [2]. Our design problem which minimizes

[2] We omit steps approximating the overflow process to Poisson process in this work due to high computation cost. Thus the results obtained $\lambda_M^{\mathsf{ra}}(\boldsymbol{x}, \boldsymbol{K})$ give optimistic result.

the request miss rate can be formulated as

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{B},\boldsymbol{K}} \quad & \lambda^{\text{miss}}(\boldsymbol{x},\boldsymbol{K}) \\ s.t. \quad & \sum_{f\in\mathcal{F}} x_{mf} \leq C_m,\ x_{mf} \in \{0,1\} \\ & K_m \leq B_s/w_m\ \forall m \in \mathcal{M}_s, K_0 \leq B_0/w_0 \\ & B_0 + B_s \leq B, K_m \in Z^+\ \forall m \in \mathcal{M}. \end{aligned} \tag{1.4}$$

Here, C_m denotes the maximum number of files that can be cached at BS $m \in \mathcal{M}$. By minimizing the request miss rate, we can indirectly reduce the backhaul traffic load and high service delay due to content download from content servers.

1.2.1.3 Proposed Algorithms

Note that K_0 is an integer variable, and it is limited by the system bandwidth, $K_0 \leq B/w_0$. Therefore, we can perform line search for all possible solutions of K_0. For a given optimal value K_0^*, to obtain the optimal solution of problem (1.4), the optimal bandwidth allocation and service capacity of the SBS can be determined as follows: **(i)** $B_0^* = K_0^* w_0$, **(ii)** $B_s^* = B - K_0^* w_0$, and **(iii)** $K_m^* = \lfloor (B - K_0^* w_0)/w_m \rfloor,\ \forall m \in \mathcal{M}_s$. Substituting $\boldsymbol{B}^*$ and $\boldsymbol{K}^*$ to problem (1.4) yields the following caching optimization problem

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \lambda^{\text{miss}}(\boldsymbol{x},\boldsymbol{K}^*) \\ s.t. \quad & \sum_{f\in\mathcal{F}} x_{mf} \leq C_m,\ x_{mf} \in \{0,1\}. \end{aligned} \tag{1.5}$$

Based on these observations, we propose Algorithm 1.1 to solve problem (1.4). In particular, it solves problem (1.4) by line-searching over possible values of K_0. For a given value of K_0^*, Algorithm 1.1 calculates the bandwidth allocation and serving capacity vectors $\boldsymbol{B}^*$ and $\boldsymbol{K}^*$. Then, it solves the caching problem (1.5) using Algorithm 1.2 explained in the next section. The request miss rates obtained from solving problem (1.5) for different values of $\boldsymbol{K}^*$ are compared to determine the optimal solution which achieves the lowest request miss rate.

Algorithm 1.1. JOINT BANDWIDTH ALLOCATION AND CACHING ALGORITHM (LSBI)

Initialization: $K_0^* = 0$, $K_0^{\mathsf{max}} = \lfloor B/w_0 \rfloor$, $\lambda_{\mathsf{opt}}^{\mathsf{miss}} = \infty$.
repeat
  $K_0^* = K_0^* + 1$
  Calculate $\boldsymbol{B}^*, \boldsymbol{K}^*$ according to **(i)**, **(ii)**, and **(iii)**.
  Solve problem (1.5) by using Algorithm 1.2 to obtain $\lambda^{\mathsf{miss}}(\boldsymbol{K}^*)$ and $\boldsymbol{x}^*$.
  if $\lambda_{\mathsf{opt}}^{\mathsf{miss}} > \lambda^{\mathsf{miss}}(\boldsymbol{K})$ **then**
    Set $\lambda_{\mathsf{opt}}^{\mathsf{miss}} \leftarrow \lambda^{\mathsf{miss}}(\boldsymbol{K}^*)$.
    Set $\boldsymbol{x}_{\mathsf{opt}} \leftarrow \boldsymbol{x}^*$, and $\boldsymbol{B}_{\mathsf{opt}} \leftarrow \boldsymbol{B}^*$
  end if
until $K_0^* > K_0^{\mathsf{max}}$.
Output λ^{miss}, $\boldsymbol{B}_{\mathsf{opt}}$ and $\boldsymbol{x}_{\mathsf{opt}}$.

In the following, we present an algorithm to solve the caching optimization problem (1.5) for a given $\boldsymbol{K}^*$. We omit $\boldsymbol{K}^*$ in all related notations in the following for brevity. The objective function of (1.5) can be re-expressed as

$$\begin{aligned}\lambda^{\mathrm{miss}}(\boldsymbol{x}) &= \lambda_M^{\mathsf{reqa}}(\boldsymbol{x}) P_0^r(\lambda_M^{\mathsf{reqa}}(\boldsymbol{x})) - \lambda_M^{\mathsf{reqa}}(\boldsymbol{x}) + \sum_{f\in\mathcal{F}} \lambda_{Mf}^{\mathsf{req}}(\boldsymbol{x}) \\ &= \lambda_{\mathsf{c}} + g_0(\lambda_M^{\mathsf{reqa}}(\boldsymbol{x})) + \sum_{m\in\mathcal{M}_s} g_m(\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)) \end{aligned} \tag{1.6}$$

where $\lambda_{\mathsf{c}} = \sum_{m\in\mathcal{M}} \lambda_m$. $g_0(\lambda) \triangleq \lambda P_0^r(\lambda) - \lambda$ and $g_m(\lambda) \triangleq \lambda P_m^r(\lambda) - \lambda$, which correspond to the MBS and SBSs, respectively. $\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)$ and $\lambda_M^{\mathsf{reqa}}(\boldsymbol{x})$ are given in equations (1.1) and (1.3), respectively.

Proposition 1.1. *For each $m \in \mathcal{M}$, $g_m(\lambda)$ decreases with λ.*

The decreasing property of $g_m(\lambda)$ with respect to λ is leveraged to design the caching algorithm. Specifically, $\lambda_M^{\mathsf{reqa}}(\boldsymbol{x})$ and $\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)$ are increasing functions of $\boldsymbol{x}$ and $\boldsymbol{x}_m$ for all $m \in \mathcal{M}$ in (1.6). Hence, to minimize the request miss ratio for a given $\boldsymbol{K}^*$, each BS has to cache to its full storage capacity to attain higher λ. Since solving the caching problem (1.5) optimally requires an extensive computation due to binary caching vector $\boldsymbol{x}$, we propose an iterative algorithm to solve problem (1.5) by sequentially solving the caching problem of each BS for a given caching solutions of other BSs until convergence.

Caching decision for the SBSs: Let $\boldsymbol{x}^t$ denote the caching solution in iteration t. Moreover, we denote $\mathcal{F}_0^t$ and $\bar{\mathcal{F}}_0^t$ as the sets of cached and un-cached files in the MBS in iteration t, respectively.

Then, the caching decision sub-problem for SBS m in iteration $t+1$ can be stated as

$$\begin{aligned} \min_{\boldsymbol{x}_m} \ & \lambda^{\mathsf{miss}}(\boldsymbol{x}_m) = \lambda_{\mathsf{c}} + g_0(\lambda_M^{\mathsf{reqa}}(\boldsymbol{x}_m)) + g_m(\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)) \\ s.t. \quad & \sum_{f\in\mathcal{F}} x_{mf} \leq C_m, x_{mf} \in \{0,1\}, \ \forall f \in \mathcal{F}. \end{aligned} \tag{1.7}$$

Problem (1.7) is still a mixed integer program which is challenging to solve. Since SBS m should cache C_m files, we propose a caching scheme in which SBS m caches $C_m - \bar{C}_m$ and $\bar{C}_m$ most popular files in sets $\mathcal{F}_0^t$ and $\bar{\mathcal{F}}_0^t$, respectively. Denote $\bar{C}_m^*$ as the optimal value of $\bar{C}_m$, which can be determined by a line-search algorithm since $\bar{C}_m^* \in [0, C_m]$.

Caching decisions for the MBS: The caching problem of the MBS can be stated as

$$\begin{aligned} \min_{\boldsymbol{x}_0} \quad & \lambda_{\mathsf{c}} + g_0(\lambda_M^{\mathsf{reqa}}(\boldsymbol{x}_0)) + \sum_{m\in\mathcal{M}_s} g_m(\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)) \\ s.t. \quad & \sum_{f\in\mathcal{F}} x_{0f} \leq C_0, x_{0f} \in \{0,1\} \ \forall f \in \mathcal{F}. \end{aligned} \tag{1.8}$$

The objective function of problem (1.8) can be written as $\lambda^{\mathsf{miss}}(\boldsymbol{x}_0) = \lambda_{\mathsf{c}}^M + g_0(\sum_{f\in\mathcal{F}} \lambda_{Mf}^{\mathsf{req}}(\bar{\boldsymbol{x}}_0) x_{0f})$. As $g_0(\lambda)$ is a decreasing function, the optimal solution of problem (1.8) is obtained when $\sum_{f\in\mathcal{F}} \lambda_{Mf}^{\mathsf{req}}(\bar{\boldsymbol{x}}_0)$ is maximized. Denote $\mathcal{C}_0^*$ as the set of C_0 highest values of $\lambda_{Mf}^{\mathsf{req}}(\bar{\boldsymbol{x}}_0)$. Then, to obtain the optimal solution of problem (1.8), MBS should cache all files in $\mathcal{C}_0^*$.

Final caching algorithm: From the caching design described in Algorithm 1.2, we can see that it creates a sequence of feasible solutions for problem (1.5) where the value of its objective function monotonically decreases over iterations. Therefore, Algorithm 1.2 converges to a feasible solution.

Algorithm 1.2 finds caching solutions for the MBS and SBSs. The caching solution for the MBS has complexity $\mathcal{O}(MF)$. The caching solution for the SBSs has complexity $\mathcal{O}(\sum_{m\in\mathcal{M}} C_m) \approx \mathcal{O}(F)$. Algorithm 1.1 has the complexity of $\mathcal{O}(K_0^{\mathsf{max}}(MF+F)) \approx \mathcal{O}(K_0^{\mathsf{max}} MF)$, which is linear with key system parameters.

1.2.1.4 Numerical Results

We consider a simulation setting with a single MBS and $|\mathcal{M}_s| = 9$ SBSs, each with coverage radius $d = 50m$, deployed within the coverage area of the MBS with coverage radius $R = 500m$. We set

Algorithm 1.2. CACHING ALGORITHM

Initialization MBS caches its C_0 most popular files and SBS m caches its C_m most popular files. Set max iteration N and tolerance ϵ.
$t = 0$
repeat
  $t = t + 1$
  $m = t$ modulo M
  if $m = 0$ **then**
    Perform caching for MBS to obtain $\boldsymbol{x}_0^{t+1^*}$
  else
    Perform caching for SBS m to obtain $\boldsymbol{x}_m^{t+1}$
  end if
  $\boldsymbol{x}_m^{t+1^*} = \text{argmin}_{\boldsymbol{x}_m^{t^*}, \boldsymbol{x}_m^{t+1}} \{\lambda^{\mathsf{miss}}(\boldsymbol{x}_m^{t^*}), \lambda^{\mathsf{miss}}(\boldsymbol{x}_m^{t+1})\}$
until $|\lambda^{\mathsf{miss}}(\boldsymbol{x}_m^{t+1}) - \lambda^{\mathsf{miss}}(\boldsymbol{x}_m^{t})| < \epsilon$ or $t > N$
Output $\boldsymbol{x}^*$

$B = 20$ and $w_m = 1,\ \forall m \in \mathcal{M}$ bandwidth units. The number of files is set $F = 100$ and the Zipf skew parameter $\gamma = 0.8$. The storage capacities of the MBS and SBSs are assumed to be $C_0 = 20$ and $C_m = 5$, respectively. The content request processes at the MBS and SBSs are Poisson processes with the normalized rates of 10^{-5} $\mathsf{requests/s/m^2}$ and 10^{-4} $\mathsf{requests/s/m^2}$, respectively. The service times of one request for the MBS and SBS are set to $10s$ and $5s$, respectively.

Figs. 1.1a and 1.1b respectively demonstrate the request miss rate versus the system bandwidth B, the caching capacity of each SBS C_m. In both figures, the LSBI algorithm outperforms the three baseline algorithms. Figs. 1.1a and 1.1b also show the small performance gap between our proposed algorithm and the performance bound, which confirms the efficacy of our proposed framework. Fig. 1.1c illustrates the request miss rates of the LSBI algorithm versus the MBS's coverage radius for different values of γ. The request miss rate of the LSBI algorithm is smaller as γ becomes larger. Moreover, the higher value of R results in increasing request miss rate since larger R leads to the higher request rate. Finally, Fig. 1.1d shows the cached files at the MBS and 2 SBSs in one particular system realization where the x-axis indicates the file indices and the y-axis shows the request probabilities of different files. We can see that SBSs tend to cache their most popular files while the caching solution of the MBS contains files ranging from low to high request probabilities. This is because each SBS would attempt to minimize the redirected request rate to the MBS by caching its most popular files. Moreover, the MBS accommodates redirected requests from all SBSs; therefore, its caching solution contains files spreading out from low to high preferences.

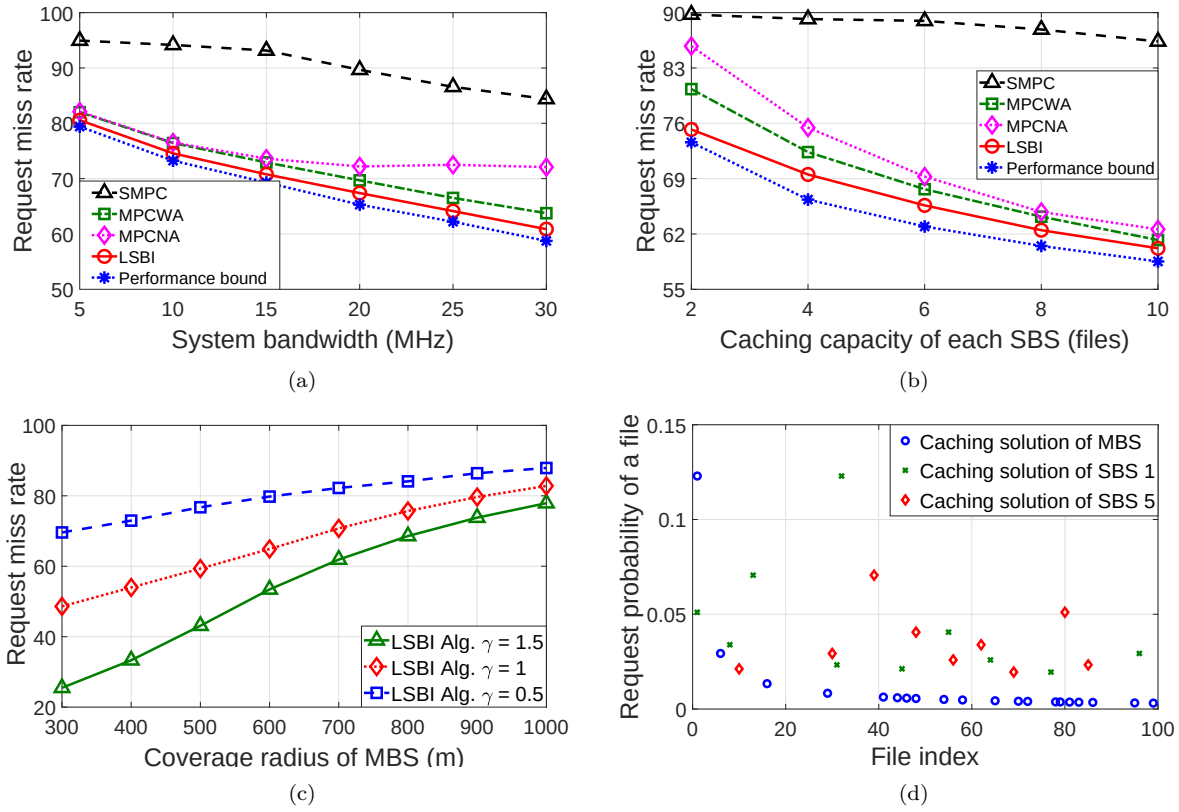


Figure 1.1 – Request miss ratio vs (a) system bandwidth B, (b) Caching capacity of each SBS, (c) Coverage radius of MBS. (d) Caching solution of MBS and SBSs.

1.2.2 Joint Resource Allocation and Content Caching in Virtualized Wireless Networks

In this contribution, we study the joint resource allocation and content caching problem for virtualized content-centric wireless networks. Recently, different content caching frameworks [29–31] have been introduced to leverage the evolution of network architectures such as femtocells and C-RAN based networks. Most of the existing works on content caching, however, do not consider the highly congested network scenario due to the lack of radio resource and bandwidth in the wireless access and backhaul links [32]. Furthermore, in the virtualized wireless environment where multiple MVNOs operate on the shared infrastructure with limited storage capacity, content caching for network performance improvement could be less significant since the InP likely partitions its limited storage capacity at BSs to MVNOs. Therefore, efficient and shareable content caching among MVNOs and optimization of radio resource allocation can effectively boost the network performance

[3]. Motivated by the aforementioned issues, we study the joint radio resource allocation and content caching design for the virtualized wireless networks (VWN) where we make the following key contributions.

- We present the problem formulation that minimizes the maximum request outage probability for all MVNOs at different BSs while avoiding content caching redundancy at the storage locations.
- To solve the obtained optimization problem, which is a mixed-integer non-linear program (MINLP), we propose a bisection-search based algorithm that iteratively optimizes the resource allocation and content caching placement.
- Extensive numerical results confirm the efficacy of our proposed framework which significantly reduces the maximum request outage probability compared to other benchmark algorithms.

1.2.2.1 System Model

We consider a downlink virtualized orthogonal frequency-division multiple access (OFDMA) multi-cell wireless network with caching repository deployed at each BS. The system consists of a set $\mathcal{K} = \{1, ..., K\}$ of BSs, which are connected to the CN via highly congested backhaul links. It is assumed that the network has W^{max} orthogonal wireless channels of equal bandwidth serving all the UEs associated with these BSs. This network infrastructure including all BSs, the backhaul and core networks, radio and storage resources are assumed to be owned and managed by an InP.

The InP serves a set $\mathcal{M} = \{1, ..., M\}$ of MVNOs, who rent resources and network infrastructure to serve their UEs. For convenience, we use MVNO (m, k) to denote MVNO m associated with BS k. For the channel allocation, we denote $\boldsymbol{w} = \{w_{11}, ..., w_{km}, ..., w_{KM}\}$ as the channel allocation vector, whose elements w_{km} represent the number of wireless channels allocated to MVNO (m, k). The UEs of each MVNO m are interested in accessing contents in a common content set $\mathcal{F} = \{f_1, ..., f_F\}$ of F files or contents, whose size of each content is normalized by 1 [13, 25]. Content requests from UEs of MVNO m in the coverage of BS k are assumed to follow the Poisson process with an average rate λ_{km} (requests/s).

[3]We use the terms file and content exchangeably in this doctoral dissertation.

Let C_k denote the capacity of the storage repository installed at BS k, which can cache up to C_k files where $C_k \in \mathbb{Z}_+$. Moreover, $\mathcal{Q}_{km} = \{q_{km1}, ..., q_{kmF}\}$ denotes the content request probability distribution where q_{kmf} represents the probability that UEs of MVNO (m, k) requests file f. $\boldsymbol{x}_{km} = \{x_{km1}, ..., x_{kmF}\}$ is the caching decision vectors for BS $k \in \mathcal{K}$ and MVNO $m \in \mathcal{M}$. $\boldsymbol{x} = \{\boldsymbol{x}_{11}, ..., \boldsymbol{x}_{km}, ..., \boldsymbol{x}_{KM}\}$ to denote the content caching decision vector for all MVNOs (m, k). Here, $x_{kmf} \in \{0, 1\}$ and $x_{kmf} = 1$ if file f is cached at BS k to serve requests from MVNO m, and $x_{kmf} = 0$, otherwise. Moreover, to ensure some minimum QoS requirement, we assume that one channel (if available) must be allocated to download a requested file from the associated BS for any UE.

1.2.2.2 Problem Formulation

We now study the file rejections due to lack of radio resources (i.e., there is no available channel) for a given caching solution $\boldsymbol{x}$. The total request rate from MVNO m for all files in $\mathcal{F}$, if they are cached at BS k, is

$$h_{km}(\boldsymbol{x}) = \sum_{f \in \mathcal{F}} \lambda_{km} q_{kmf} \left(\sum_{i \in \mathcal{M}} x_{kif} \right). \tag{1.9}$$

Otherwise, the total cache-missed file request rate from MVNO m to all files in $\mathcal{F}$ at BS k is

$$\bar{h}_{km}(\boldsymbol{x}) = \sum_{f \in \mathcal{F}} \lambda_{km} q_{kmf} \left(1 - \sum_{i \in \mathcal{M}} x_{kif} \right) = \lambda_{km} - h_{km}(\boldsymbol{x}). \tag{1.10}$$

Assume that it takes T_{km} (s) for BS k to serve a cache-hit file request from MVNO m. T_{km} represents the download time from the content cache to the UE of MVNO (m, k). With w_{km} channels allocated by the InP to MVNO (m, k) to serve requests of UEs, at most w_{km} file requests from MVNO m can be simultaneously served by its associated BS. The file requests from MVNO m at BS k can be modeled as an $M/D/w_{km}/w_{km}$ queue with Poisson arrivals, deterministic service time, w_{km} servers, and no waiting buffer [27].

We assume that all cache-missed file requests are rejected due to high delay for downloading content from the CN. Additionally, any cache-hit file request from MVNO m at BS k is only rejected if all w_{km} channels are used to service other ongoing w_{km} requests. From [27], the probability that there are w_{km} ongoing cache-hit file requests from MVNO m being served by BS k can be calculated

as

$$P_{km}(\boldsymbol{x},\boldsymbol{w}) = \frac{(h_{km}(\boldsymbol{x})T_{km})^{w_{km}}}{w_{km}!}\left(\sum_{i=0}^{w_{km}}\frac{(h_{km}(\boldsymbol{x})T_{km})^{i}}{i!}\right)^{-1}. \tag{1.11}$$

Consequently, the rejection rate for the cache-hit request from MVNO m at BS k due to channel unavailability can be expressed as

$$\mu_{km}(\boldsymbol{x},\boldsymbol{w}) = h_{km}(\boldsymbol{x})P_{km}(\boldsymbol{x},\boldsymbol{w}). \tag{1.12}$$

From (1.10) and (1.12), the total file request outage probability from MVNO m at BS k can be calculated as

$$\Phi_{km}(\boldsymbol{x},\boldsymbol{w}) = \frac{\mu_{km}(\boldsymbol{x},\boldsymbol{w}) + \bar{h}_{km}(\boldsymbol{x})}{\lambda_{km}}. \tag{1.13}$$

To avoid poor QoS and unfair treatment in serving file requests from different MVNOs at different BSs, we consider the joint channel allocation and content caching optimization problem which minimizes the highest outage probability among MVNOs at all BSs while accounting for the file caching redundancy avoidance and other system constraints. This problem can be formulated as follows:

$$\min_{\boldsymbol{x},\boldsymbol{w}} \quad \max_{k\in\mathcal{K},m\in\mathcal{M}} \Phi_{km}(\boldsymbol{x},\boldsymbol{w}) \tag{1.14a}$$

$$\text{s.t.} \quad \sum_{m\in\mathcal{M}} x_{kmf} \le 1, \forall k\in\mathcal{K}, \forall f\in\mathcal{F} \tag{1.14b}$$

$$\sum_{m\in\mathcal{M}}\sum_{f\in\mathcal{F}} x_{kmf} \le C_k, \forall k\in\mathcal{K} \tag{1.14c}$$

$$w_{km} \ge W_{km}^{\mathsf{min}}, \forall k\in\mathcal{K}, \forall m\in\mathcal{M} \tag{1.14d}$$

$$\sum_{k\in\mathcal{K}}\sum_{m\in\mathcal{M}} w_{km} \le W^{\mathsf{max}} \tag{1.14e}$$

$$x_{kmf} \in \{0,1\}\ \forall k\in\mathcal{K}, \forall m\in\mathcal{M}, \forall f\in\mathcal{F}, \tag{1.14f}$$

where (1.14b) and (1.14c) capture the file redundancy avoidance and storage capacity constraints, respectively; (1.14d) represents the service-level-agreement (SLA) constraints for MVNO m at BS k, which guarantees certain minimum number of allocated channels for each MVNO; (1.14e) denotes the bandwidth constraint; and (1.14f) denotes the integer caching decision variables at BSs.

Algorithm 1.3. Channel Allocation for a Given Caching Solution

1: **allocate** $W_{km}^{\min}$ channels to MVNO m at BS k to satisfy (1.14d).
2: **calculate** $W^{\text{free}} = W^{\max} - \sum_{k\in\mathcal{K}} \sum_{m\in\mathcal{M}} W_{km}^{\min}$
3: **while** $W^{\text{free}} > 0$ **do**
4: **find** $(k^*, m^*) = \underset{k,m}{\operatorname{argmax}}\, \Phi_{km}(\boldsymbol{x}, \boldsymbol{w})$
5: $w_{k^*m^*} = w_{k^*m^*} + 1$
6: $W^{\text{free}} = W^{\text{free}} - 1$
7: **end while**
8: **obtain** optimal $\boldsymbol{w}^*$

1.2.2.3 Proposed Algorithms

The main objective of this contribution is to solve problem (1.14), which is a MINLP due to the integer variables $\boldsymbol{x}$ and $\boldsymbol{w}$ and the nonlinear function $\Phi_{km}(\boldsymbol{x}, \boldsymbol{w})$. We propose a two-step iterative algorithm for finding the channel allocation and content caching placement in each iteration. The overall procedure can be illustrated as

$$\underbrace{\boldsymbol{x}^*_{(\mathbf{0})} \rightarrow \boldsymbol{w}^*_{(\mathbf{0})}}_{\text{Initialization, } \varphi_{(0)}} \rightarrow \ldots \rightarrow \underbrace{\boldsymbol{x}^*_{(i)} \rightarrow \boldsymbol{w}^*_{(i)}}_{\text{Iteration } i, \varphi_{(0)}} \rightarrow \ldots \rightarrow \underbrace{\boldsymbol{x}^* \rightarrow \boldsymbol{w}^*}_{\text{Optimal } \varphi^*},$$

where the stopping condition is $|\varphi_{(i)} - \varphi_{(i-1)}| < \varepsilon$ with $0 < \varepsilon << 1$. Based on the caching solution ${\boldsymbol{x}^*}_{(i-1)}$ obtained in the previous iteration $(i-1)$., we propose Algorithm 1.3, which is based on the property of $P_{km}(\boldsymbol{x}^*, \boldsymbol{w})$ stated in Proposition 1.2, to find the optimal channel allocation ${\boldsymbol{w}^*}_{(i)}$ in iteration i.

Proposition 1.2. *(i) For a given $\boldsymbol{x}^*$, $P_{km}(\boldsymbol{x}^*, \boldsymbol{w})$ in (1.11) is a decreasing function of $\boldsymbol{w}$. (ii) For a given $\boldsymbol{w}^*$, $P_{km}(\boldsymbol{x}, \boldsymbol{w}^*)$ is an increasing function of $\boldsymbol{x}$.*

Lemma 1.1. *For a given caching strategy $\boldsymbol{x}^*$, Algorithm 1.3 optimally allocates channels to individual MVNOs at all BSs to minimize the largest request outage probability in the network.*

After obtaining $\boldsymbol{w}^*{}_{(i)}$, we proceed to find the content caching solution $\boldsymbol{x}^*{}_{(i)}$ in iteration i through solving the following problem

$$\min_{\boldsymbol{x}} \quad \max_{k\in\mathcal{K},m\in\mathcal{M}} \Phi_{km}\left(\boldsymbol{x},\boldsymbol{w}^*\right) \tag{1.15a}$$

$$\text{s.t.} \quad \sum_{m\in\mathcal{M}} x_{kmf} \leq 1, \forall k \in \mathcal{K}, \forall f \in \mathcal{F} \tag{1.15b}$$

$$\sum_{m\in\mathcal{M}} \sum_{f\in\mathcal{F}} x_{kmf} \leq C_k, \forall k \in \mathcal{K} \tag{1.15c}$$

$$x_{kmf} \in [0,1]\ \forall k \in \mathcal{K}, \forall m \in \mathcal{M}, \forall f \in \mathcal{F}. \tag{1.15d}$$

Here, $\boldsymbol{x}$ is relaxed to continuous variable as shown in (1.15d). We then rewrite $\Phi_{km}\left(\boldsymbol{x},\boldsymbol{w}^*\right)$ as function of $h_{km}(\boldsymbol{x})$, i.e., $\Phi_{km}\left(h_{km}(\boldsymbol{x}),\boldsymbol{w}^*\right)$.

Proposition 1.3. *$\Phi_{km}\left(h_{km}(\boldsymbol{x}),\boldsymbol{w}^*\right)$ is a convex function of $h_{km}(\boldsymbol{x})$ for a given $\boldsymbol{w}^*$. Moreover, it is a decreasing function of $h_{km}(\boldsymbol{x})$.*

Consequently, $\max_{k\in\mathcal{K},m\in\mathcal{M}} \Phi_{km}\left(h_{km}\left(\boldsymbol{x}\right),\boldsymbol{w}\right)$ can be considered as the pointwise maximum function over $h_{km}(\boldsymbol{x})$, which is convex [33]. This allows us to transform problem (1.15) to the following convex optimization problem over $\boldsymbol{h}$, where $\boldsymbol{h} = \{h_{km}\left(\boldsymbol{x}\right)\}, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}$.

$$\min_{\boldsymbol{h},\varphi} \quad \varphi \tag{1.16a}$$

$$\text{s.t.} \quad \Phi_{km}\left(h_{km}(\boldsymbol{x}),\boldsymbol{w}^*\right) \leq \varphi, \forall k \in \mathcal{K}, \forall m \in \mathcal{M} \tag{1.16b}$$

$$h_{km}\left(\boldsymbol{x}\right) \in \mathcal{H}, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}. \tag{1.16c}$$

In (1.16), $\mathcal{H}$ denotes the set of all feasible values of $h_{km}\left(\boldsymbol{x}\right)$, which is dependent on the feasible set of $\boldsymbol{x}$ according to the constraints of problem 1.15.

$$\max_{\boldsymbol{x}} \quad \sum_{k\in\mathcal{K}}\sum_{m\in\mathcal{M}} h_{km}\left(\boldsymbol{x}\right) \tag{1.17a}$$

$$\text{s.t.} \quad h_{km}\left(\boldsymbol{x}\right) \geq h_{km}^{\mathsf{low}}, \forall m \in \mathcal{M}, \forall k \in \mathcal{K} \tag{1.17b}$$

$$\sum_{m\in\mathcal{M}} x_{kmf} \leq 1, \forall k \in \mathcal{K}, \forall f \in \mathcal{F} \tag{1.17c}$$

$$\sum_{m\in\mathcal{M}}\sum_{f\in\mathcal{F}} x_{kmf} \leq C_k, \forall k \in \mathcal{K} \tag{1.17d}$$

$$x_{kmf} \in [0,1] \, \forall m \in \mathcal{M}, \forall f \in \mathcal{F}. \tag{1.17e}$$

To solve 1.16, we propose the bisection-search based Algorithm 1.4 to determine the caching decision solution $\boldsymbol{x}^*{}_{(i)}$. With the newly obtained $\boldsymbol{w}^*{}_{(i)}$ and $\boldsymbol{x}^*{}_{(i)}$, we compute the maximum request outage probability $\varphi_{(i)}$ for iteration i. The Newton's search method for calculating the hit rate h_{km} of MVNO m at BS k, given the request outage probability φ_{km} and and channel allocation w_{km}. After that, we solve problem (1.17) to find the relaxed solution $\boldsymbol{x}^*{}_i$, i.e., $\boldsymbol{x}^*{}_i \in [0,1]$. Here, h_{km}^{low} is the output of the inverse function Φ_{km}^{-1} taking φ as the input. After executing Algorithm 1.4, we proceed to round the obtained caching decision $\boldsymbol{x}^*$ into integer values by using Algorithm 1.5.

1.2.2.4 Numerical Results

In this section, we evaluate the performance of our proposed algorithms through computer simulation under the following setting. We consider the network with 5 BSs serving 3 MVNOs, which access a list of 100 files, i.e., $K = 5, M = 3$ and $F = 100$. The average request rates for each MVNO are randomly chosen in the range of $[1, 15]$, which results in the total of request rates from tens to hundreds requests arriving to the considered network in one second. We assume that all BSs share $W^{\mathsf{max}} = 90$ wireless channels in the orthogonal manner to serve file requests from MVNOs. Each SLA requirement is set with $W_{km}^{\mathsf{min}} = 2, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}$.

Fig. 1.2a shows that the proposed bisection-search based algorithm (Algorithm 1.4) with cache sharing consistently achieves the smallest maximum request outage probability. Moreover, the proposed rounding operation for caching decision variables result in negligible performance loss compared to the achieved performance before rounding, which confirms the efficacy of our design

Algorithm 1.4. Iterative Channel Allocation and Content Caching Placement

set $i = 1$ and tolerance $\varepsilon > 0$.
initialize $\boldsymbol{x}^*_{(i)}$ according to most popular caching strategy with equal storage partition.
initialize channel allocation $\mathbf{w}_{(i)}$ using Algorithm 1.3 given $\boldsymbol{x}^*_{(i)}$.
calculate $\Phi_{km}, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}$.
find largest outage probability $\varphi_{(i)} = \max_{k,m} \Phi_{km}$.
set $\Delta_\varphi = 1$
while $\Delta_\varphi > \varepsilon$ **do**
  $i = i + 1$
  set $\phi^{\mathsf{up}} = 1$
  set $\phi^{\mathsf{low}} = 0$
  while $\phi^{\mathsf{up}} - \phi^{\mathsf{low}} > \varepsilon$ **do**
    $\phi_{(i)} = \left(\phi^{\mathsf{up}} + \phi^{\mathsf{low}}\right)/2$
    find h^{low}_{km} from $\phi_{(i)}$ by using Newton' search method for all $m \in \mathcal{M}$ and $k \in \mathcal{K}$.
    solve problem (1.17) to find $\boldsymbol{x}^*$.
    if $\boldsymbol{x}^*$ is feasible **then**
      $\phi^{\mathsf{up}} = \phi_{(i)}$
      $\boldsymbol{x}^*_{(i)} = \boldsymbol{x}^*$
    else
      $\phi^{\mathsf{low}} = \phi_{(i)}$
    end if
  end while
  find optimal $\boldsymbol{w}^*_{(i)}$ by using Algorithm 1.3.
  calculate $\Phi^{(i)}_{km}, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}$.
  find $\varphi_{(i)} = \max_{k,m} \Phi^{(i)}_{km}$
  calculate $\Delta_\varphi = |\varphi^*_{(i)} - \varphi^*_{(i-1)}|$
end while
obtain final $\boldsymbol{w}^*$ and $\boldsymbol{x}^*$ from Algorithm 1.3 given $\boldsymbol{x}^*_{(i)}$.

Algorithm 1.5. Rounding Caching Decision Variables

initialize small $\varepsilon > 0$
obtain the optimal request outage probability value φ from Algorithm 1.4.
repeat
  obtain $h^{\mathsf{low}}_{km}, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}$ using Newton's search method.
  solve problem (1.17) with integral constraint.
  if integral solution $\boldsymbol{x}^*_{\mathsf{INT}}$ is not found **then**
    $\varphi = \varphi + \varepsilon$
  end if
until integral solution $\boldsymbol{x}^*_{\mathsf{INT}}$ is found.

(the request outage probability obtained under relaxation from Algorithm 3 is the lower bound of the optimum value). Moreover, the proposed heuristic algorithm achieves performance very close to the proposed bisection-search based algorithm in the different-order popularity case, and both algorithms result in the same solution in the same-order file popularity case.

We present the maximum request outage probability among MVNOs at all BSs versus the total number of channels, the Zipf parameter γ, and the maximum request rate in Figs. 1.2b, 1.2c,

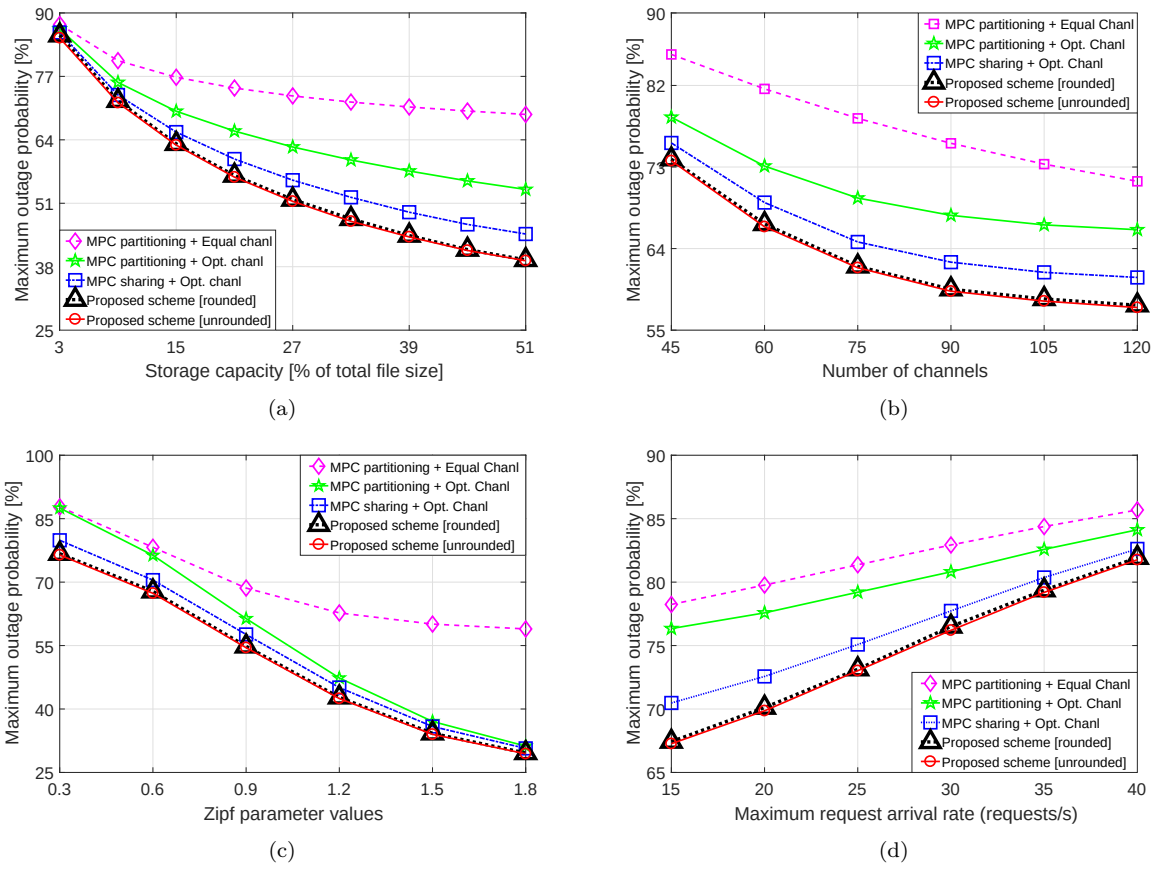


Figure 1.2 – Maximum outage probability vs (a) storage capacity, (b) number of channels, (c) Zipf parameter, and (d) maximum request arrival rate.

and 1.2d, respectively. Similar to Figure 1.2a, Fig. 1.2b confirms the greatest performance of our proposed bisection-search based algorithm as it achieves the lowest request outage probability compared with the remaining baselines. Figures 1.2a and 1.2b also imply that instead of partitioning the available storage space to individual MVNOs, it is better to share it among MVNOs co-located at the same BS.

1.2.3 Resource Allocation for Multi-Tenant Network Slicing: A Multi-Leader Multi-Follower Stackelberg Game Approach

Network slicing also provides the paradigm shift toward multi-tenancy in the next-generation wireless network [23] where individual tenants (e.g., MVNOs, SPs) own and manage corresponding network slices. Various game theoretic approaches have been applied to tackle different resource allocation problems in the network slicing context [34–37]. The multilateral interactions among SPs

and their customers such as UEs, which constitute to a service trading market, can be modeled by using the Stackelberg game theory. Numerous works in resource allocation [38–43] and other fields of industry [44, 45] have applied single-leader-multiple-follower (SLMF) Stackelberg game. The multi-leader-multi-follower (MLMF) Stackelberg game has been employed in some recent works [46–50]. Nevertheless, studying the interactions among the peer access service providers (ASPs) and between ASPs and their UEs in the network slicing-based wireless network has not been considered. The related works above only consider the single-source service selection between the stakeholders, i.e., a UE can only select one SP for purchasing service. In our work, we thus study the resource allocation and pricing problem for network slicing that captures interactions among access/backhaul service providers and their UEs by using the MLMF Stackelberg game approach. Furthermore, we allow any UE and ASP to be able to lease services from different ASPs and backhaul service providers (BSPs) at the same time, respectively, thereby enabling the multiple slice connectivity. Our contributions in this study are as follows.

- We formulate the interactions among UEs, ASPs as a MLMF Stackelberg game [51].
- We derive the price best-response functions for the ASPs and the throughput best-response function for the UEs in the access layer. We prove the existence of a unique Stackelberg game equilibrium. We further prove that these best-response functions belong to the class of standard functions [52] and they satisfy the so-called two-sided scalability (2.s.s) property [53].
- These results above are leveraged in developing a distributed algorithm that converges to the game equilibrium.
- We evaluate the efficacy of our proposed framework and investigate the achievable performance and strategies of different network stakeholders via extensive numerical studies.

1.2.3.1 System Model

We consider the downlink of a cellular network with both wireless backhaul and access communications. We assume that the infrastructures and wireless resources of the wireless backhaul and access layers are owned and manged by a set $\mathcal{I} = \{1, ..., i, ..., I\}$ of backhaul service providers (BSPs) and a set $\mathcal{J} = \{1, ..., j, ..., J\}$ ASPs, respectively. We assume that BSP $i \in \mathcal{K}$ owns one corresponding

wireless backhaul hub (WBH). Each BSP i has a sufficiently large and dedicated bandwidth non-overlapped with the spectra of other BSPs. Each ASP j possesses one BS and the spectrum band of W_j^{A} Hz non-overlapped with the bands of other ASPs. We assume that these BSs serve a set $\mathcal{K} = \{1, ..., k, ..., K\}$ of UEs in a particular service area (i.e., a cell) and the spectrum bands used by these ASPs are non-overlapped. Consequently, there is no co-channel interference among ASPs in this considered network section. In practice, the ASPs can reuse their spectrum in other areas which are sufficiently far away, thus introducing certain inter-cell interference. However, one can greatly mitigate such interference with careful cell planning. Furthermore, the average co-channel interference can be estimated and accumulated in the background noise power for users in the considered service area [48].

We further assume that the ASPs must purchase backhaul communication resources from BSPs to support end-to-end communications between UEs and the core network (CN). Interactions among different network stakeholders for resource trading occur in fixed-size time intervals where the number of active users remains the same in each time interval. Let denote w_{ij} as the amount of bandwidth (Hz) that ASP j acquires from the BSP i, and r_{ij} (bps/Hz) is the average spectrum efficiency achieved by the corresponding backhaul link. We assume that UE k purchases a fraction $s_{jk} \in [0, 1]$ of the ASP j's spectrum resource for data transmission in the access layer; then, the average spectrum efficiency achieved by the access link between UE k and ASP j is denoted as r_{jk} (bps/Hz).

We assume that each ASP can transfer data over multiple backhaul links through multiple connections with different BSPs simultaneously. Similarly, each UE can receive data over multiple access links associated with different ASPs simultaneously. In practice, this can be achieved by multi-connection technology [54–57], multi-band data aggregation techniques [58, 59], traffic steering techniques [60–64], and supporting protocols [65, 66]. As a result, ASPs (UEs) can purchase backhaul (access) bandwidth from different BSPs (ASPs) at the same time, thereby enabling multiple slice connectivity [67] for both the UEs and ASPs. The interactions among these network stakeholders and end users constitute a trading market including the backhaul and access resource markets.

1.2.3.2 MLMF Stackelberg Game Formulation

In this game formulation, the ASPs and UEs act as the leaders and the followers, respectively. The leaders play first in Stage I by imposing access prices taking into account the potential budgets and demands of the UEs, and the ASPs decide the amount of bandwidth resource acquired from the BSPs for cost minimization. In the second stage (Stage II), each UE optimally purchases data from the ASPs to maximize its utility given the prices imposed by the ASPs. Each player selfishly maximizes its own payoff function in this MLMF Stackelberg game.

Stage I: Noncooperative Access Pricing Game among ASPs and Backhaul Resource Acquisition The payoff of ASP j can be defined as the revenue earned from providing access services to UEs minus the backhaul expenditure. Specifically, the ASP $j \in \mathcal{J}$ is interested in maximizing the following payoff function:

$$P_j(p_j, \boldsymbol{s}_j, \boldsymbol{w}_j) = \delta_j U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j) - \eta_j C_j^{\mathsf{A}}(\boldsymbol{w}_j) \tag{1.18}$$

where δ_j and η_j are respectively the coefficients associated with the revenue function $U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j)$ and cost function $C_j^{\mathsf{A}}(\boldsymbol{w}_j)$. $\boldsymbol{w}_j = (w_{ij})_{i\in\mathcal{I}}$ denotes the vector of backhaul bandwidth purchased by the ASP j from the BSPs. $\boldsymbol{s}_j = (s_{jk})_{k\in\mathcal{K}}$ represents the vector of access bandwidth fractions that ASP j allocates to the associated UEs to meet their throughput demands $(d_{jk})_{k\in\mathcal{K}}$. The revenue function $U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j)$ is defined as

$$U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j) = \sum_{k=1}^{K} p_j d_{jk} - \theta_j \sum_{k=1}^{K} W_j^{\mathsf{A}} s_{jk}. \tag{1.19}$$

where θ_j is the associated coefficient. Because each ASP must purchase the backhaul bandwidth to serve its UEs, the ASP j has to pay the backhaul cost $C_j^{\mathsf{A}}(\boldsymbol{w}_j)$, which is defined as follows:

$$C_j^{\mathsf{A}}(\boldsymbol{w}_j) = \sum_{i=1}^{I} q_i w_{ij} + \frac{1}{2}\sum_{i=1}^{I} w_{ij}^2 + \nu_j \sum_{i'\neq i} w_{ij} w_{i'j} \tag{1.20}$$

where q_i denotes the price per backhaul bandwidth unit (\$/Hz) offered by the BSP i. In this paper, we set $\nu_j \in (0, 1)$ so that the ASP j can transmit data in different spectra according to the above substitutability property, thus ensuring the multi-band transmission ability for each ASP.

For simplicity, we consider the optimization of the two functions $U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j)$ and $C_j^{\mathsf{A}}(\boldsymbol{w}_j)$ independently. Specifically, the ASP j has to set the competitive price p_j to attract more demand from UEs considering the available spectrum resource W_j^{A} and the prices offered by other ASPs. The revenue maximization of individual ASPs taking into account the prices of other ASPs thus constitute to a game called *Stage-I* AG *game*, which is defined later.

The Stage-I AG *Game - The Access Resource Pricing Game*

The utility function (1.19) of the ASP j can be expressed as $U_j^{\mathsf{A}}(p_j, \boldsymbol{p}_{-j}, \boldsymbol{s}_j)$ to describe the interactions with other ASPs. Here, $\boldsymbol{p}_{-j} = (p_{j'})_{j' \in \mathcal{J}, j' \neq j}$ denote the strategies of other ASPs except ASP j. Now, the AG game can be defined as follows:

1. *Players:* The ASPs in the set $\mathcal{J}$.
2. *Strategy:* $p_j^{\mathsf{L}} \leq p_j \leq p_j^{\mathsf{U}},\ \forall j \in \mathcal{J}$ such that

$$\sum_{k=1}^{K} s_{jk} \leq 1, \forall j \in \mathcal{J}. \tag{1.21}$$

3. *Utility function:* $U_j^{\mathsf{A}}(p_j, \boldsymbol{p}_{-j}, \boldsymbol{s}_j),\ \forall j \in \mathcal{J}$.

In the AG game, p_j^{L} and p_j^{U} are the positive lower and upper bounds on the access price, which are imposed by the market regulations for the ASP j. Note that (1.21) represents the access bandwidth (fraction) allocation constraint for each ASP j.

Stage II: Traffic Demand Optimization of Each UE: The payoff function of each UE $k \in \mathcal{K}$ is defined as follows:

$$U_k^{\mathsf{E}}(\boldsymbol{d}_k) = e_k \sum_{j=1}^{J} \log\left(1 + d_{jk}\right) \tag{1.22}$$

where $\boldsymbol{d}_k = (d_{jk})_{j \in \mathcal{J}}$ denotes the throughput demand vector of UE k, $d_{jk} = s_{jk} W_j^{\mathsf{A}} r_{jk}$ is the throughput of UE k supported by ASP j, and e_k denotes the utility coefficient of UE k.

Given the price imposed by the leaders (ASPs) $\boldsymbol{p}^*$, UE k is interested in maximizing its payoff by acquiring appropriate throughput for different ASPs considering its maximum budget B_k.

Mathematically, this throughput demand optimization problem can be formulated as

$$\max_{\boldsymbol{d}_k \geq \mathbf{0}} U_k^{\mathsf{E}}(\boldsymbol{d}_k) \tag{1.23a}$$

$$\text{s.t. } \sum_{j=1}^{J} p_j d_{jk} \leq B_k \tag{1.23b}$$

where (1.23b) captures the maximum budget constraint of UE k.

1.2.3.3 Analysis of the MLMF Stackelberg Game Equilibrium

We derive the Stackelberg equlibrium (SE) of the considered Stackelberg game by using the *backward induction* method. Specifically, we first derive the optimal throughput demand of UEs in Stage II of the Stackelberg game. Then, we use this result to derive the NE among the ASPs in Stage I of the game.

i. Optimal Throughput Demand of UEs in Stage II: We first state the optimal throughput demand of UEs in Stage II of the Stackelberg game in the following lemma.

Lemma 1.2. *For given prices $(p_j)_{j\in\mathcal{J}}$ and budget B_k, the optimal throughput demand of UE k obtained by solving problem* (1.23) *can be expressed as follows:*

$$d_{jk}^* = \left[\frac{B_k + \sum_{j'=1}^{J} p_{j'}}{J p_j} - 1\right]^+ = \left[\frac{B_k + \sum_{j'\neq j} p_{j'}}{J p_j} + \frac{1}{J} - 1\right]^+, \ \forall j \in \mathcal{J} \tag{1.24}$$

where $[x]^+ = \max\{0, x\}$.

ii. Pricing and Resource Allocation Solution for the Access Layer in Stage-I AG *Subgame:* The bandwidth portion allocated by ASP j to UE k can be expressed as

$$s_{jk}^* = \frac{B_k + \sum_{j'\neq j} p_{j'}}{J D_{jk} p_j} + \frac{1}{D_{jk}}\left(\frac{1}{J} - 1\right) = \frac{\alpha_{jk}(\boldsymbol{p}_{-j})}{p_j} + \beta_{jk} \tag{1.25}$$

where $\alpha_{jk}(\boldsymbol{p}_{-j}) \triangleq \frac{B_k + \sum_{j'\neq j} p_{j'}}{J D_{jk}}$ and $\beta_{jk} \triangleq \frac{1}{D_{jk}}\left(\frac{1}{J} - 1\right)$. By substituting the results in (1.24) and (1.25) into (1.19), the utility achieved by ASP j becomes

$$U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j, \boldsymbol{p}_{-j}) = K(\frac{1}{J} - 1)p_j + \sum_{k=1}^{K} \frac{B_k + \sum_{j'\neq j} p_{j'}}{J} - \theta_j W_j^{\mathsf{A}} \left(\frac{\alpha_j(\boldsymbol{p}_{-j})}{p_j} + \beta_j\right). \tag{1.26}$$

where $\alpha_j(\boldsymbol{p}_{-j}) \triangleq \sum_{k=1}^{K} \alpha_{jk}(\boldsymbol{p}_{-j})$ and $\beta_j \triangleq \sum_{k=1}^{K} \beta_{jk}$. We use the notation $U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j, \boldsymbol{p}_{-j})$ to present the impact of other ASPs' strategies $\boldsymbol{p}_{-j}$ to the utility of ASP j. Now, we substitute the result for s_{jk}^* in (1.25) to (1.21) and exploit the fact that $p_j > 0$, this constraint becomes

$$\begin{aligned} &\sum_{k=1}^{K} \left(\frac{\alpha_{jk}(\boldsymbol{p}_{-j})}{p_j} + \beta_{jk} \right) \leq 1 \\ &\Leftrightarrow (1-\beta_j)p_j - \alpha_j(\boldsymbol{p}_{-j}) \geq 0. \end{aligned} \tag{1.27}$$

Given the strategies $\boldsymbol{p}_{-j}$ of other ASPs, the optimal strategy of ASP $j \in \mathcal{J}$ for the Stage-I AG subgame, i.e., its best response, is the solution of the following optimization problem:

$$\begin{aligned} F_j(\boldsymbol{p}_{-j}) = &\underset{p_j \in \mathcal{P}_j}{\operatorname{argmax}} \; U_j^{\mathsf{A}}(p_j, \boldsymbol{p}_{-j}) \\ &\text{s.t. constraint (1.27).} \end{aligned} \tag{1.28}$$

We state the convexity of this problem in the following lemma.

Lemma 1.3. *Problem* (1.28) *is convex.*

Theorem 1.1. *There exists a NE for the Stage-I* AG *subgame.*

The KKT optimality conditions of problem (1.28) are given by

$$\frac{\partial \mathcal{L}_j^{\mathsf{A}}}{\partial p_j} = 0 \tag{1.29a}$$

$$\text{constraint (1.27)} \tag{1.29b}$$

$$\lambda_j \geq 0 \tag{1.29c}$$

$$\lambda_j \left[(1-\beta_j)p_j - \alpha_j \right] = 0. \tag{1.29d}$$

where $\mathcal{L}_j^{\mathsf{A}}(p_j, \lambda_j) = U_j^{\mathsf{A}}(p_j, \boldsymbol{p}_{-j}) + \lambda_j \left[(1-\beta_j)\, p_j - \alpha_j(\boldsymbol{p}_{-j}) \right]$ and λ_j is the Lagrange multiplier. After solving (1.29), we obtain the best response function of the price offered to the UEs of an ASP $j \in \mathcal{J}$

as follow:

$$p_j^{(t+1)} = \begin{cases} F_j^{\mathsf{h}}(\boldsymbol{p}_{-j}^{(t)}) = \left[\sqrt{\frac{\alpha_j(\boldsymbol{p}_{-j}^{(t)}) \sqrt{\alpha_j(\boldsymbol{p}_{-j}^{(t)})}}{(1-\beta_j)^2}} \right]_{\mathcal{P}_j} & \text{if } \theta_j < \frac{K\left(1-\frac{1}{J}\right)\sqrt{\alpha_j(\boldsymbol{p}_{-j})}}{W_j^{\mathsf{A}}(1-\beta_j)^2} \\ F_j^{\mathsf{n}}(\boldsymbol{p}_{-j}^{(t)}) = \left[\sqrt{\frac{\theta_j}{K(J-1)} \sum_{k=1}^{K} \frac{B_k + \sum_{j' \neq j} p_{j'}}{r_{jk}}} \right]_{\mathcal{P}_j} & \text{otherwise} \end{cases} \tag{1.30}$$

where t is the iteration index. Based on the results in Lemma 1.6, we propose a distributed algorithm, which is summarized in Algorithm 1.6, to find the NE among the ASPs with respect to the prices $\boldsymbol{p}$ and $\boldsymbol{s}^*$.

Definition 1.1. *A function $\boldsymbol{F}(\boldsymbol{p})$ is called standard and satisfies the 2.s.s property if the following conditions hold [53]:*

- *Positivity:* $\boldsymbol{F}(\boldsymbol{p}) > \boldsymbol{0}$
- *Monotonicity: if* $\boldsymbol{p} \geq \boldsymbol{p'}$ *then* $\boldsymbol{F}(\boldsymbol{p}) \geq \boldsymbol{F}(\boldsymbol{p'})$
- *Scalability:* $\forall \mu > 1$, $\mu \boldsymbol{F}(\boldsymbol{p}) \geq \boldsymbol{F}(\mu \boldsymbol{p})$
- *Two-sided scalability (2.s.s): For all* $\mu > 1$, $\frac{1}{\mu}\boldsymbol{p} \leq \boldsymbol{p'} \leq \mu \boldsymbol{p}$ *implies* $\frac{1}{\mu}\boldsymbol{F}(\boldsymbol{p}) < \boldsymbol{F}(\boldsymbol{p'}) < \mu \boldsymbol{F}(\boldsymbol{p})$.

Lemma 1.4. $\boldsymbol{F}^{\mathsf{h}}(\boldsymbol{p}) = \left(F_j^{\mathsf{h}}(\boldsymbol{p}_{-j})\right)_{j \in \mathcal{J}}$, *where $F_j^{\mathsf{h}}(\boldsymbol{p}_{-j})$ is the upper-part equation of* (1.30)*, is a standard function with 2.s.s property in domain $\mathcal{P}$.*

Lemma 1.5. $\boldsymbol{F}^{\mathsf{n}}(\boldsymbol{p}) = \left(F_j^{\mathsf{n}}(\boldsymbol{p}_{-j})\right)_{j \in \mathcal{J}}$, *where $F_j^{\mathsf{n}}(\boldsymbol{p}_{-j})$ is the lower-part equation of* (1.30)*, is a standard function with 2.s.s property in domain $\mathcal{P}$.*

Lemma 1.6. *The iterative updates in* (1.30) *for $F_j^{\mathsf{h}}(\boldsymbol{p}_{-j}^{(t)})$ and $F_j^{\mathsf{n}}(\boldsymbol{p}_{-j}^{(t)})$ for all $j \in \mathcal{J}$ converge to the corresponding fixed point $\boldsymbol{p}^{\mathsf{h}*}$ and $\boldsymbol{p}^{\mathsf{n}*}$, respectively.*

1.2.3.4 The Backhaul Expenditure Minimization Problem

At the equilibrium point of the Stage-I AG subgame, each ASP obtains the total throughput demand from associated UEs, which is given by

$$R_j^{\mathsf{A}}(\boldsymbol{s}_j^*) \triangleq \sum_{k=1}^{K} s_{jk}^* W_j^{\mathsf{A}} r_{jk}, \ \forall j \in \mathcal{J} \tag{1.31}$$

Algorithm 1.6. Distributed Algorithm for Pricing and Access Bandwidth Allocation

Initialize $t = 0$ and ε.
Each ASP set $p_j^{(t)} = p_j^{\mathsf{L}}, \forall j \in \mathcal{J}$.
Each ASP announces its prices $p_j^{(t)}$ to the access market.
// Price adjustment
while True **do**
  $t = t + 1$
  for $j = 1$ to J **do**
    if $\theta_j < \frac{K\left(1-\frac{1}{J}\right)\sqrt{\alpha_j(\boldsymbol{p}_{-j})}}{W_j^{\mathsf{A}}(1-\beta_j)^2}$ **then**
      ASP j updates its prices $p_j^{(t)}$ according to F_j^{h} in (1.30)
    else
      ASP j updates its prices $p_j^{(t)}$ according to F_j^{n} in (1.30)
    end if
  end for
  Each ASP announces its prices $p_j^{(t)}$ to other ASPs.
  if $||\boldsymbol{p}^{(t+1)} - \boldsymbol{p}^{(t)}|| < \varepsilon$ **then**
    Exit the While loop
  end if
end while
All the ASPs announce their prices to the UEs.
// Backhaul bandwidth acquisition
for Each ASP $j \in \mathcal{J}$ **do**
  Calculate $s_{jk},\ \forall j \in \mathcal{J}, k \in \mathcal{K}$ according to (1.25).
  Calculate $\boldsymbol{w}_j = (w_{ij})_{i\in\mathcal{I}}$ using Algorithm 1.7.
  Calculate the payoff according to (1.18).
end for

where $(s_{jk}^*)_{j\in\mathcal{J},k\in\mathcal{K}}$ is the bandwidth demand of UEs at the Stage-I AG subgame's NE point. Accordingly, the objective of each ASP is to minimize the total payment to all the BSPs given their backhaul prices while avoiding the traffic congestion at its BS. This backhaul constrained cost minimization problem for each ASP j can be stated as follows:

$$\min_{\boldsymbol{w}_j \geq \mathbf{0}} C_j^{\mathsf{A}}(\boldsymbol{w}_j) \tag{1.32a}$$

$$\text{s.t.} \sum_{i=1}^{I} w_{ij} r_{ij} \geq R_j^{\mathsf{A}}(\boldsymbol{s}_j^*) \tag{1.32b}$$

where (1.32b) is the backhaul constraint and R_j^{A} is defined in (1.31). This is a convex optimization problem due to the convex objective function and linear constraint (1.32b). Accordingly, we propose Algorithm 1.7 to solve the equivalent Karush–Kuhn–Tucker (KKT) conditions of problem 1.32.

Lemma 1.7. *By setting* $\nu_j \leq \min_{i\in\mathcal{I}} \left\{ \frac{r_{ij}}{R_j} \right\}$*, Algorithm 1.7 will converge.*

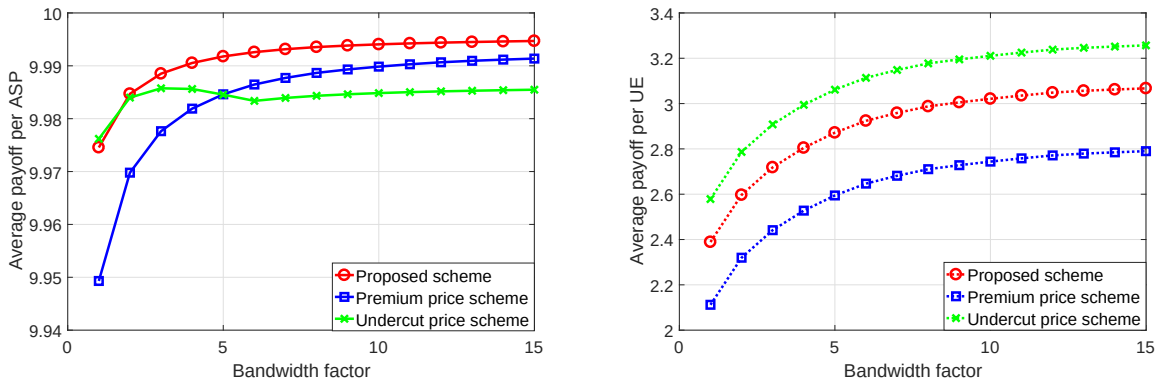
Algorithm 1.7. BACKHAUL BANDWIDTH ACQUISITION OF AN ASP

Init γ_j^{L}, γ_j^{U}, $\forall j \in \mathcal{J}$, and ε.
while $||\boldsymbol{\gamma}^{\mathsf{U}} - \boldsymbol{\gamma}^{\mathsf{L}}|| > \varepsilon$ **do**
  Set $\gamma_j = (\gamma_j^{\mathsf{L}} + \gamma_j^{\mathsf{U}})/2$.
  Calculate $w_{ij} = \xi_j \left[\gamma_j \kappa_{ij} - \rho_{ij}\right]^+$, $\forall i \in \mathcal{I}, j \in \mathcal{J}$.
  Calculate $R_j = \sum_{i=1}^{I} w_{ij} r_{ij}$, $\forall j \in \mathcal{J}$.
  for $j = 1$ to J **do**
    if $R_j < R_j^{\mathsf{A}}(\boldsymbol{s}_j^*)$ **then**
      Update $\gamma_j^{\mathsf{L}} = \gamma_j$
    else
      Update $\gamma_j^{\mathsf{U}} = \gamma_j$
    end if
  end for
end while
Return γ_j, w_{ij}, $\forall i \in \mathcal{I}$.

1.2.3.5 Numerical Results

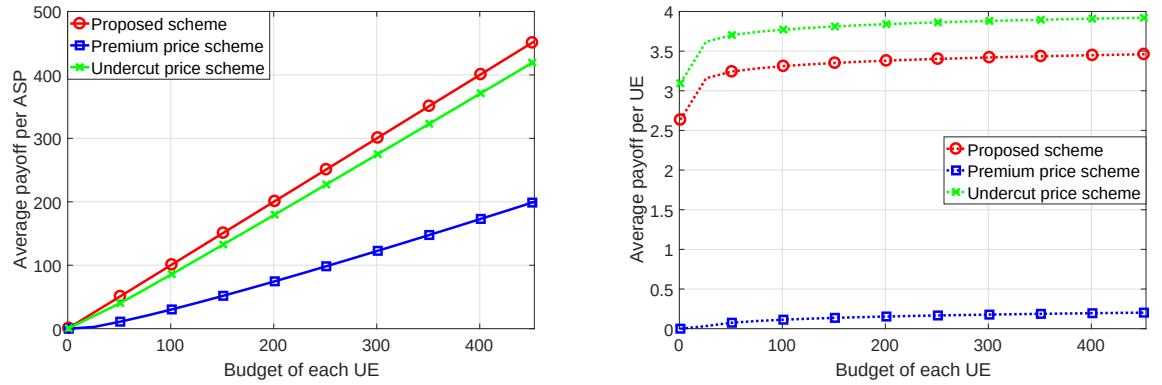
We evaluate the performance achieved by the proposed game theoretic framework for a wireless network with 3 BSPs ($I = 3$), 4 ASPs ($J = 4$) and 20 UEs ($K = 20$). The BSs of ASPs and UEs are uniformly distributed in a circular area with the radius of 200 meters, while the wireless backhaul hubs are located with distances up to 400 meters from the origin. The average spectrum efficiency of each access link between a BS of an ASP and a UE (i.e., $(r_{jk})_{j\in\mathcal{J},k\in\mathcal{K}}$) ranges from 0.35 to 12 bps/Hz, depending on the communication distance. Also, the average spectrum efficiency of each backhaul link between a BS of an ASP and a WBH of a BSP (i.e., $(r_{ij})_{i\in\mathcal{I},j\in\mathcal{J}}$) ranges from 1 to 5 bps/Hz.

In Figs. 1.3 and 1.4, we compare the average payoffs of the ASPs and UEs resulted from the proposed framework and two other baselines schemes. In Figs. 1.3a and 1.3b, we show the average payoffs of each ASP and each UE, respectively due to the proposed framework and the considered baseline schemes as the access bandwidth factor varies. The average payoffs of each APS and each UE are respectively shown in Figs. 1.4a and 1.4b for different values of each UE's budget. It can be observed that the proposed framework enables the ASPs to achieve highest average payoff (Figs. 1.3a and 1.4a) in comparison with other baseline schemes for all considered values of the bandwidth factor and UE's budget. Due to imposing significantly high prices under the premium price scheme, the ASPs discourage throughput demands from the UEs (even though when they have large budgets as shown in Fig. 1.3b). This premium scheme thus results in inferior average payoffs for both the ASPs and UEs compared to those due to the proposed scheme, as shown in Figs. 1.3 and 1.4.



(a) Average payoff of ASPs. (b) Average payoff of UEs

Figure 1.3 – Average payoff of ASPs and UEs in comparison with baseline pricing schemes when the access bandwidth factor varies.



(a) Average payoff of ASPs. (b) Average payoff of UEs

Figure 1.4 – Average payoff of APSs and UEs in comparison with baseline pricing schemes when the UE's budget varies.

Under the undercut price scheme, the UEs are able to earn a slightly higher average payoff than that due to the proposed framework. The average payoff of each ASP under this scheme, however, cannot surpass the one due to the proposed framework. This is because a player (the ASP 4 in this circumstance) lowers its price below the SE to attract more throughput demands from the UEs while deteriorating the sale force of other rival ASPs. The increment in throughput demands from the UEs gained by the ASP 4, however, cannot compensate the lost due to the low-imposing price of this player. Therefore, the ASPs, who are the market leaders, are better off adopting the proposed framework to achieve highest average payoff for each of them while allowing a decent average payoff for each UE (the follower).

1.3 Concluding Remarks

In this doctoral dissertation, we have proposed joint resource management techniques and algorithms as well as game theory based resource trading mechanism for various future network environments. Specifically, we have made three important research contributions. First, we have propose a resource management solution for heterogeneous small-cell networks with bandwidth allocation and caching-aware BS association. Here, we combine the low-complexity line-search algorithm for bandwidth allocation with the iterative caching algorithm for content placement to minimize the content request miss ratio of users.

Second, we have designed a joint resource allocation and content caching framework for VWN whereby we minimize the maximum content request rejection rate experienced by users of different MVNOs in different BSs in a highly congestion backhaul scenario. We further proposed an efficient bisection-search based algorithm to optimized the resource allocation and content placement at each BS, which outperforms other benchmark algorithms.

Finally, we have proposed a MLMF Stackelberg game based framework for resource pricing and allocation that captures interactions between SPs as well as between SPs and UEs. We have showed that the proposed game based framework achieves a unique game equilibrium. The, we have developed a distributed algorithm based on updating the underlying best-response functions, which was proved to converge to the game equilibrium. The research in our doctoral study has resulted in three journal papers [68–70] and six papers on prestigious conferences [42, 71–75].

Chapter 2

Résumé Long

2.1 Contexte et motivation

Le système cellulaire sans fil de cinquième génération (5G), qui a commencé à être déployé en 2020, devrait apporter une amélioration considérable des performances du réseau et prendre en charge de nouveaux services et applications, par rapport à ceux activés par le système actuel de quatrième génération (4G) [2–4]. Plus précisément, une augmentation de 1000 fois du débit du réseau par rapport à celle des systèmes 4G est la spécification cible promise par les systèmes 5G [5]. Cette augmentation significative de la capacité du réseau permet de faire face au trafic mobile toujours croissant généré par les services améliorés à large bande mobile (eMBB - enhanced Mobile Broadband) tels que le streaming vidéo mobile [6–9]. En plus de prendre en charge le type de service eMBB, le futur système cellulaire sans fil 5G prend également en charge les deux autres types de services clés, à savoir les communications ultra-fiables à faible latence (uRLLC - ultra Reliable Low Latency Communications) et les communications massives de type machine (mMTC - massive Machine Type Communications) pour servir respectivement les applications critiques et un grand nombre de connexions simultanées à partir d'appareils sans fil [8, 10, 11]. En conséquence, une lourde charge de trafic réseau ainsi que des exigences strictes sont imposées à la fois au réseau d'accès radio (RAN - Radio Access Network) et au réseau de liaison, qui établissent des connexions de bout en bout entre les équipements utilisateur (UE - User Equipment) et le réseau central (CN - Core Network) via stations de base (BS - Base Station). De nouvelles techniques et de nouvelles

architectures de réseau doivent être conçues et bien intégrées pour permettre au système cellulaire sans fil 5G de répondre à des exigences aussi strictes et polyvalentes [3, 4, 7, 8].

2.1.1 Utilisation et gestion avancées des ressources

Exploiter de nouvelles bandes de spectre radioélectrique [12] et améliorer l'efficacité du spectre sont deux approches nécessaires et complémentaires pour améliorer le débit et la qualité du réseau. En outre, tirer parti d'autres types de ressources, en particulier le référentiel de stockage, est une approche prometteuse pour réduire les délais de communication et soulager la congestion du trafic [3]. En fait, en déployant des périphériques de stockage au niveau des BS dans le réseau et en récupérant à l'avance du contenu / des fichiers populaires, également appelé *mise en cache de contenu*, [1] dans ces référentiels de stockage, on peut rapprocher les contenus populaires la proximité des UE. Par conséquent, le trafic dans les liaisons de raccordement induit par l'accès à ces contenus, qui sont généralement stockés dans le CN s'ils ne sont pas mis en cache au niveau des BS, est également considérablement réduit [13]. Ce faisant, la latence d'accès de bout en bout à ces contenus est réduite, améliorant ainsi la qualité de service des utilisateurs (QoS - Quality of Service) [14–16].

Les innovations visant à améliorer l'efficacité du spectre et à exploiter les dimensions émergentes des ressources, généralement la mise en cache du contenu, sont plus avantageuses si elles sont conçues conjointement avec d'autres cadres de gestion des ressources. Pourtant, la conception de cadres de gestion des ressources avancés qui peuvent utiliser les avantages de l'amélioration de l'efficacité du spectre et de la mise en cache du contenu est difficile et nécessite des recherches beaucoup plus poussées.

2.1.2 Tranchage réseau - Virtualisation de réseau sans fil

Pour aider le réseau 5G à répondre aux exigences strictes de ses divers types de services, il est essentiel de concevoir des architectures de réseau innovantes (par exemple, réseaux ultra denses (UDN - Ultra Dense Network) [17, 18], réseau d'accès radio cloud (C-RAN - Cloud RAN) [19]) qui non seulement intègrent des technologies avancées mais les font également fonctionner ensemble de manière transparente. Cependant, le déploiement et l'exploitation de ces nouvelles architectures de

[1] Nous utilisons le terme contenu et fichier de manière interchangeable dans cette thèse de doctorat.

réseau pour les systèmes 5G ainsi que l'intégration de technologies innovantes dans ces systèmes nécessitent une refonte massive de l'infrastructure réseau, à la fois dans l'interface radio et dans le réseau de liaison. Ces exigences peuvent entraîner une surcharge des dépenses en capital (CAPEX - Capital Expenditure) et des coûts d'exploitation (OPEX - Operational Expenditure), ainsi que ralentir le temps de déploiement des nouvelles technologies et des services de réseau [7, 8]. La virtualisation de réseau sans fil (WNV - Wireless Network Virtualization), également connue sous le nom de *tranchage réseau*, a été considérée comme un paradigme de réseau prometteur pour résoudre ce problème [20].

En fait, la virtualisation sans fil permet à plusieurs opérateurs de réseaux virtuels mobiles (MVNO - Mobile Virtual Network Operator), également appelés fournisseurs de services (SP - Service Provider), de partager la même infrastructure réseau et un pool de ressources commun détenu et géré par un (ou plusieurs) fournisseur (s) d'infrastructure (InP - Infrastructure Provider). Sur cette infrastructure réseau commune, l'InP est en charge de l'allocation flexible et efficace des ressources réseau aux MVNO en fonction de leurs contrats. Chaque MVNO utilise à son tour les ressources et l'infrastructure louées pour fournir ses propres services [21] à ses clients, y compris l'UE ou d'autres MVNO avec une qualité de service (QoS) engagée. En conséquence, le découpage du réseau aide les opérateurs de réseau et les SP à réduire les CAPEX et les OPEX en utilisant les ressources du réseau de manière flexible et efficace tout en répondant mieux à la QoS requise [20]. Grâce à des caractéristiques évolutives et flexibles, le découpage de réseau accélère également la mise en œuvre et l'intégration de la technologie dans les réseaux cellulaires sans fil 5G [20].

2.1.3 Aspects économiques du partage des ressources entre les locataires du réseau 5G

Le découpage de réseau est une technologie importante pour laquelle le réseau monolithique peut être virtuellement découpé en plusieurs tranches de réseau pour prendre en charge des services sans fil spécialisés. Des tranches de réseau correctement conçues, par exemple, pourraient être désignées pour les services de streaming à haute vitesse tels que YouTube et Netflix, ou les services uRLLC pour les applications de contrôle d'usine [22]. Le découpage de réseau fournit également un changement de paradigme vers la multi-location dans le réseau sans fil de prochaine génération [23] où les locataires individuels (e.g., MVNO, SP) possèdent et gèrent les tranches de réseau correspondantes.

En permettant l'échange de services entre les locataires, ce changement de paradigme offre de plus grandes opportunités commerciales et de plus grandes économies en CAPEX et OPEX [22].

En conséquence, il existe des interactions multilatérales entre les SP, les InP et les UE concernant l'aspect économique. Ici, l'interaction peut être une concurrence économique entre les SP fournissant le même type de service à un marché, ou il peut s'agir d'une interaction d'achat et de vente entre les SP et leurs clients tels que les UE. Ces interactions multilatérales entre les SP et leurs clients constituent un marché d'échange de ressources. La conception d'un cadre approprié pour faire fonctionner un tel marché est cruciale pour atteindre une facilité d'entretien du réseau efficace et des profits élevés. La théorie des jeux est un outil approprié pour modéliser un tel marché [24].

2.2 Contributions à la recherche

Dans le cadre de cette thèse de doctorat, les contributions de recherche ont pour trois objectifs principaux. Tout d'abord, nous développons un cadre commun d'allocation de ressources radio et de mise en cache de contenu dans un cadre de réseau hétérogène à petites cellules (HetNet - Heterogeneous Network), où nous considérons le problème d'allocation de ressources et de mise en cache de contenu pour un opérateur de réseau unique avec son propre bassin de ressources. Deuxièmement, nous étudions l'allocation conjointe des ressources et la mise en cache de contenu dans l'environnement de réseau multicellulaire virtualisé où plusieurs opérateurs de réseau partagent un bassin de ressources commun de canaux sans fil et de référentiels de stockage sous la coordination d'un contrôleur centralisé. Troisièmement, nous considérons le problème d'allocation des ressources concernant les interactions multilatérales entre les SP ainsi qu'entre les SP et leurs clients dans un cadre de découpage de réseau en utilisant la théorie des jeux de Stackelberg. Tous les objectifs visent à répondre directement aux problèmes techniques importants du scénario de réseau futur (HetNets) et des paradigmes de réseau émergents (découpage de réseau). Les principales contributions de ce doctorat la thèse est la suivante:

2.2.1 Mise en cache pour les réseaux hétérogènes à petites cellules avec allocation de bande passante et association bs compatible avec la mise en cache

Dans cette contribution, nous étudions le problème de la mise en cache pour les réseaux hétérogènes à petites cellules avec allocation de bande passante et association BS sensible à la mise en cache. Il existe des travaux existants qui étudient le problème de mise en cache pour les réseaux à petites cellules [15, 16] où ils étudient la mise en cache conjointe, le routage et l'attribution des canaux. Cependant, tous ces travaux ne prennent pas en compte le comportement stochastique des processus de demande de contenu et de service. Pendant ce temps, les auteurs de [25, 26] étudient la mise en cache conjointe et la conception d'allocation des ressources sur la base du rapport signal sur bruit (SNR - Signal-to-noise Ratio) variant dans le temps. Cette conception nécessiterait des mises à jour fréquentes du cache, ce qui n'est pas rentable car le SNR varie généralement rapidement dans le temps. En conséquence, une conception générale de mise en cache pour les HetNets où les utilisateurs mobiles peuvent être associés à une station de base à petites cellules (SBS - Small-cell BS) ou à une station de base à macrocellules (MBS - Macro-cell BS) et à une ressource radio allouée pour télécharger le contenu souhaité doit être envisagée. Les associations BS dans un tel réseau hétérogène devraient prendre des décisions de mise en cache (i.e., être sensibles à la mise en cache) où les utilisateurs devraient s'associer à des BS qui ont des conditions de canal favorables et stocker leurs contenus demandés.

Motivés par les problèmes susmentionnés, nous concevons un problème conjoint de mise en cache de contenu et d'allocation de bande passante pour les HetNets où nous apportons les contributions clés suivantes.

- Nous concevons un cadre commun de mise en cache de contenu et d'allocation de bande passante pour minimiser le taux de demandes manquées.
- Nous proposons un algorithme Line-Search-based-Iterative (LSBI) qui détermine la solution en combinant l'algorithme de recherche de ligne pour obtenir l'allocation optimale de la bande passante avec l'algorithme de mise en cache itérative pour acquérir une solution de mise en cache.

- Les résultats numériques démontrent que l'algorithme LSBI surpasse considérablement les algorithmes de mise en cache existants et qu'il est à la hauteur d'une limite de performances.

2.2.1.1 Modèle de système

Nous considérons un système de mise en cache hétérogène à petites cellules consistant en un MBS noté BS 0 et S sans chevauchement SBS dans l'ensemble $\mathcal{M}_s = \{1, \cdots, S\}$ déployé dans la zone de couverture du MBS. Soit $\mathcal{M} = \{0\} \cup \mathcal{M}_s$ l'ensemble de tous les BS. Nous supposons que la bande passante du système B est attribuée orthogonalement aux MBS et SBS, et tous les SBS réutilisent la même bande passante. Soit B_0 et B_s respectivement la bande passante attribuée au MBS et à tous les SBS, où $B_0 + B_s \leq B$ et nous désignons $\boldsymbol{B} = [B_0, B_s]$.

Soit w_m la bande passante requise pour servir un utilisateur dans BS $m \in \mathcal{M}$. Soit $\boldsymbol{K} = [K_0, \cdots, K_m, \cdots, K_M]$ la capacité de service du système, où K_m représente le nombre maximal d'utilisateurs pouvant être servis simultanément par BS $m \in \mathcal{M}$. Nous désignons également $\boldsymbol{K} = [\boldsymbol{K}_0, \bar{\boldsymbol{K}}_0]$, où $\bar{\boldsymbol{K}}_0 = [K_1, \cdots, K_M]$. Ensuite, pour maintenir la QoS des utilisateurs requis dans la cellule m, K_m doit satisfaire $K_m \leq B_s/w_m \ \forall m \in \mathcal{M}_s$, $K_0 \leq B_0/w_0$, et $K_m \in Z^+$, où Z^+ désigne l'ensemble des entiers non négatifs.

Nous considérons la stratégie d'association BS adaptée à la mise en cache adaptative suivante. Étant donné que l'utilisateur k dans la zone de couverture de SBS m demande un fichier, le SBS servira l'utilisateur (i.e., que l'utilisateur k sera associé à SBS m) s'il dessert moins de K_m utilisateurs. et le fichier est actuellement mis en cache sur le SBS. Sinon, la demande est redirigée vers le MBS. Au niveau du MBS, si le fichier demandé est disponible dans son cache et que le MBS dessert moins de K_0 utilisateurs, la demande sera servie (c'est-à-dire que l'utilisateur k changera son association au MBS). Sinon, la demande est manquée.

Nous supposons que les utilisateurs demandent des fichiers dans l'ensemble $\mathcal{F} = \{f_1, \cdots, f_F\}$. Ces fichiers sont supposés avoir la même taille et peuvent être stockés dans les caches des BS pour de futurs téléchargements. Nous supposons que les distributions de popularité des fichiers dans $\mathcal{F}$ dépendent de la zone de service où les utilisateurs dans différentes zones peuvent avoir des préférences de fichier différentes. Soit $\boldsymbol{p}_m = [p_{m1}, \cdots, p_{mF}]$, les probabilités de demande de fichier des utilisateurs dans la zone de couverture BS $m \in \mathcal{M}$ où p_{mf} indique la probabilité que le fichier

f soit demandé par un utilisateur dans la zone de couverture de BS m et $||\boldsymbol{p}_m||_1 = 1\ \forall m \in \mathcal{M}$. Nous supposons que les demandes de contenu dans BS $m \in \mathcal{M}$ suivent le processus de Poisson avec un taux moyen λ_m(requests/s). Nous supposons que $\boldsymbol{p}_m$ et λ_m sont connus. Enfin, nous supposons qu'il faut T_m secondes à BS m pour répondre à une demande (i.e., le temps de téléchargement du fichier).

Soit $\boldsymbol{x}_m = [x_{m1}, \cdots, x_{mF}]$ and $\boldsymbol{x} = [\boldsymbol{x}_0, \cdots, \boldsymbol{x}_S]$ représentent les décisions de mise en cache de BS m et de toutes les BS, respectivement. Plus précisément, $x_{mf} \in \{0, 1\}$ indique l'état de mise en cache du fichier f à SBS m, où $x_{mf} = 1$ signifie que le fichier f est mis en cache à BS m, $x_{mf} = 0$, sinon. Nous désignons également le vecteur de mise en cache de tous les BS du système comme $\boldsymbol{x} = (\boldsymbol{x}_s, \bar{\boldsymbol{x}}_s)$ où $\boldsymbol{x}_s$ et $\bar{\boldsymbol{x}}_s$ sont les vecteurs de mise en cache de BS $s \in \mathcal{M}$ et d'autres BS, respectivement.

2.2.1.2 Formulation du problème

Nous analysons d'abord les performances de mise en cache d'un SBS particulier, qui est utilisé dans la formulation du problème. Étant donné que le taux de demande associé à SBS $m \in \mathcal{M}_s$ est λ_m, le taux de demande pour le fichier f à SBS m est $\lambda_m p_{mf}$. Si le fichier f n'est pas mis en cache sur SBS m, la demande est redirigée vers le MBS. Désignez $\lambda_{mf}^{\mathsf{redb}}(\boldsymbol{x}_m)$ comme taux redirigé vers le MBS de SBS m pour le fichier f en raison de l'indisponibilité du fichier f dans le cache. Ensuite, $\lambda_{mf}^{\mathsf{redb}}(\boldsymbol{x}_m)$ peut être exprimé comme $\lambda_{mf}^{\mathsf{redb}}(\boldsymbol{x}_m)$=$\lambda_m p_{mf}(1 - x_{mf})$. Par conséquent, le taux de demande moyen pour tous les fichiers vers SBS m peut être calculé comme suit:

$$\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m) = \sum_{f \in \mathcal{F}} \lambda_m p_{mf} x_{mf}. \tag{2.1}$$

Notez que la requête agrégée suit le processus de Poisson car tous les processus de requête individuels sont Poisson [27]. Rappelons que SBS m peut servir simultanément au plus K_m utilisateurs et qu'il faut T_m (s) pour répondre à une requête. Par conséquent, nous pouvons modéliser la demande / le service à SBS m comme une file d'attente $M/D/K_m/K_m$, qui a des arrivées de Poisson, un temps de service déterministe, des serveurs K_m et tampon de longueur nulle. Par conséquent,

la probabilité qu'une requête soit bloquée [27] à SBS m peut être exprimée comme

$$P_m^r(\boldsymbol{x}_m, K_m) = \frac{(\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)T_m)^{K_m}}{K_m!} \left(\sum_{i=0}^{K_m} \frac{(\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)T_m)^i}{i!} \right)^{-1}. \tag{2.2}$$

Notez que si la demande de fichier f est bloquée par SBS m en raison de sa capacité de service limitée, la demande est redirigée vers le MBS. Désignez $\lambda_{mf}^{\mathsf{reda}}(\boldsymbol{x}_m, K_m)$ comme le taux de demande redirigé vers le MBS de SBS m en raison de sa capacité de service limitée, ce paramètre peut être calculé comme $\lambda_{mf}^{\mathsf{reda}}(\boldsymbol{x}_m, K_m)$=$\lambda_{mf}^{\mathsf{req}}(\boldsymbol{x}_m)P_m^r(\boldsymbol{x}_m, K_m)$. Comme toutes les demandes rejetées par les SBS en raison de la capacité de service limitée ou de l'indisponibilité des fichiers demandés dans les caches des SBS sont redirigées vers le MBS, nous pouvons calculer le taux de demande total du fichier f redirigé vers le MBS comme $\lambda_f^{\mathsf{red}}(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{K}}_0)$= $\sum\limits_{m\in\mathcal{M}_s} \left(\lambda_{mf}^{\mathsf{redb}}(\boldsymbol{x}_m)+\lambda_{mf}^{\mathsf{reda}}(\boldsymbol{x}_m, K_m)\right)$. Par conséquent, le taux de demande total du fichier f vers le MBS, y compris les demandes originales et redirigées, peut être exprimé comme suit: $\lambda_{Mf}^{\mathsf{req}}(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{K}}_0)$=$\lambda_0 p_{0f}$+$\lambda_f^{\mathsf{red}}(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{K}}_0)$.

Le taux de demandes manquées associé au MBS en raison de l'indisponibilité des fichiers peut être exprimé comme $\lambda_M^{\mathsf{rb}}(\boldsymbol{x}, \bar{\boldsymbol{K}}_0) = \sum_{f\in\mathcal{F}} \lambda_{Mf}^{\mathsf{req}}(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{K}}_0)(1-x_{0f})$. Par conséquent, le taux de demande pour tous les fichiers du MBS peut être calculé comme suit:

$$\lambda_M^{\mathsf{reqa}}(\boldsymbol{x}, \bar{\boldsymbol{K}}_0) = \sum_{f\in\mathcal{F}} \lambda_{Mf}^{\mathsf{req}}(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{K}}_0)x_{0f}. \tag{2.3}$$

Notez que les requêtes (avec le taux $\lambda_{mf}^{\mathsf{reda}}(\boldsymbol{x}_m, K_m)$) redirigées du SBS m vers le MBS en raison de la capacité de service limitée à ce SBS forment un trafic de débordement, qui est un processus non-Poisson. On peut remplacer ce processus de débordement par une approximation de Poisson en utilisant plusieurs techniques telles que *approximation de Hayward* et *méthode aléatoire équivalente* [28]. Ce faisant, nous pouvons modéliser la demande / le service de contenu au niveau du MBS comme une file d'attente $M/D/K_0/K_0$ avec un processus de Poisson d'entrée. Par conséquent, la probabilité de manque de demande due à la capacité de desserte limitée du MBS peut être calculée de manière similaire à (2.2), i.e., $P_0^r(\lambda_M^{\mathsf{reqa}}((\boldsymbol{x}, \bar{\boldsymbol{K}}_0), \boldsymbol{K})) = \frac{(\lambda_M^{\mathsf{reqa}}(\boldsymbol{x}, \bar{\boldsymbol{K}}_0)T_0)^{K_0}}{K_0!} \left(\sum\limits_{i=0}^{K_0} \frac{(\lambda_M^{\mathsf{reqa}}(\boldsymbol{x}, \bar{\boldsymbol{K}}_0)T_0)^i}{i!} \right)^{-1}$. Par conséquent, le taux de demandes manquées en raison de la capacité de desserte limitée du MBS peut être exprimé comme suit: $\lambda_M^{\mathsf{ra}}(\boldsymbol{x}, \boldsymbol{K}) = \lambda_M^{\mathsf{reqa}}(\boldsymbol{x}, \bar{\boldsymbol{K}}_0)P_0^r(\lambda_M^{\mathsf{reqa}}((\boldsymbol{x}, \bar{\boldsymbol{K}}_0), \boldsymbol{K}))$. Enfin, le taux total de demandes manquées du système peut être calculé comme suit: $\lambda^{\mathsf{miss}}(\boldsymbol{x}, \boldsymbol{K})$=$\lambda_M^{\mathsf{ra}}(\boldsymbol{x}, \boldsymbol{K})$+$\lambda_M^{\mathsf{rb}}\left(\boldsymbol{x}, \bar{\boldsymbol{K}}_0\right)$

[2]. Notre problème de conception qui minimise le taux de demandes manquantes peut être formulé comme suit:

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{B},\boldsymbol{K}} \quad & \lambda^{\text{miss}}(\boldsymbol{x},\boldsymbol{K}) \\ s.t. \quad & \sum_{f\in\mathcal{F}} x_{mf} \leq C_m,\ x_{mf} \in \{0,1\} \\ & K_m \leq B_s/w_m\ \forall m \in \mathcal{M}_s, K_0 \leq B_0/w_0 \\ & B_0 + B_s \leq B, K_m \in Z^+\ \forall m \in \mathcal{M}. \end{aligned} \tag{2.4}$$

Ici, C_m indique le nombre maximal de fichiers pouvant être mis en cache dans BS $m \in \mathcal{M}$. En minimisant le taux de demandes manquées, nous pouvons indirectement réduire la charge de trafic de liaison et le retard de service élevé dû au téléchargement de contenu depuis les serveurs de contenu.

2.2.1.3 Algorithmes proposés

Notez que K_0 est une variable entière, et elle est limitée par la bande passante du système, $K_0 \leq B/w_0$. Par conséquent, nous pouvons effectuer une recherche en ligne pour toutes les solutions possibles de K_0. Pour une valeur optimale donnée K_0^*, afin d'obtenir la solution optimale du problème (2.4), l'allocation optimale de la bande passante et la capacité de service du SBS peuvent être déterminées comme suit: **(i)** $B_0^* = K_0^* w_0$, **(ii)** $B_s^* = B - K_0^* w_0$, and **(iii)** $K_m^* = \lfloor (B - K_0^* w_0)/w_m \rfloor\ ,\ \forall m \in \mathcal{M}_s$. La substitution de $\boldsymbol{B}^*$ et $\boldsymbol{K}^*$ au problème (2.4) génère le problème d'optimisation de la mise en cache suivant

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \lambda^{\text{miss}}(\boldsymbol{x},\boldsymbol{K}^*) \\ s.t. \quad & \sum_{f\in\mathcal{F}} x_{mf} \leq C_m,\ x_{mf} \in \{0,1\}. \end{aligned} \tag{2.5}$$

Sur la base de ces observations, nous proposons l'algorithme 2.1 pour résoudre le problème (2.4). En particulier, il résout le problème (2.4) en recherchant en ligne les valeurs possibles de K_0. Pour une valeur donnée de K_0^*, l'algorithm 2.1 calcule l'allocation de bande passante et les vecteurs de capacité de desserte $\boldsymbol{B}^*$ et $\boldsymbol{K}^*$. Ensuite, il résout le problème de mise en cache (2.5) en utilisant

[2]Nous omettons les étapes rapprochant le processus de débordement au processus de Poisson dans ce travail en raison du coût de calcul élevé. Ainsi, les résultats obtenus $\lambda_M^{\text{ra}}(\boldsymbol{x},\boldsymbol{K})$ donnent un résultat optimiste.

l'algorithme 2.2 expliqué dans la section suivante. Les taux de demandes manquées obtenues en résolvant le problème (2.5) pour différentes valeurs de $\boldsymbol{K}^*$ sont comparés pour déterminer la solution optimale qui atteint le taux de demandes le plus bas.

Algorithm 2.1. ALLOCATION CONJOINTE DE BANDE PASSANTE ET ALGORITHME DE MISE EN CACHE (LSBI)

1: Initialisation: $K_0^* = 0$, $K_0^{\mathsf{max}} = \lfloor B/w_0 \rfloor$, $\lambda_{\mathsf{opt}}^{\mathsf{miss}} = \infty$.
2: **repeat**
3: $K_0^* = K_0^* + 1$
4: Calculer $\boldsymbol{B}^*, \boldsymbol{K}^*$ selon **(i)**, **(ii)**, et **(iii)**.
5: Résoudre le problème (2.5) en utilisant l'algorithme 2.2 pour obtenir $\lambda^{\mathsf{miss}}(\boldsymbol{K}^*)$ et $\boldsymbol{x}^*$.
6: **if** $\lambda_{\mathsf{opt}}^{\mathsf{miss}} > \lambda^{\mathsf{miss}}(\boldsymbol{K})$ **then**
7: Fixer $\lambda_{\mathsf{opt}}^{\mathsf{miss}} \leftarrow \lambda^{\mathsf{miss}}(\boldsymbol{K}^*)$.
8: Fixer $\boldsymbol{x}_{\mathsf{opt}} \leftarrow \boldsymbol{x}^*$, et $\boldsymbol{B}_{\mathsf{opt}} \leftarrow \boldsymbol{B}^*$
9: **end if**
10: **until** $K_0^* > K_0^{\mathsf{max}}$.
11: Produire λ^{miss}, $\boldsymbol{B}_{\mathsf{opt}}$ et $\boldsymbol{x}_{\mathsf{opt}}$.

Dans ce qui suit, nous présentons un algorithme pour résoudre le problème d'optimisation de la mise en cache (2.5) pour un $\boldsymbol{K}^*$ donné. Nous omettons $\boldsymbol{K}^*$ dans toutes les notations connexes ci-dessous pour plus de brièveté. La fonction objective de (2.5) peut être ré-exprimée comme

$$\begin{aligned}\lambda^{\mathrm{miss}}(\boldsymbol{x}) &= \lambda_M^{\mathsf{reqa}}(\boldsymbol{x}) P_0^r(\lambda_M^{\mathsf{reqa}}(\boldsymbol{x})) - \lambda_M^{\mathsf{reqa}}(\boldsymbol{x}) + \sum_{f\in\mathcal{F}} \lambda_{Mf}^{\mathsf{req}}(\boldsymbol{x}) \\ &= \lambda_{\mathsf{c}} + g_0(\lambda_M^{\mathsf{reqa}}(\boldsymbol{x})) + \sum_{m\in\mathcal{M}_s} g_m(\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)) \end{aligned} \tag{2.6}$$

où $\lambda_{\mathsf{c}} = \sum_{m\in\mathcal{M}} \lambda_m$. $g_0(\lambda) \triangleq \lambda P_0^r(\lambda) - \lambda$ et $g_m(\lambda) \triangleq \lambda P_m^r(\lambda) - \lambda$, qui correspondent respectivement aux MBS et SBS. $\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)$ et $\lambda_M^{\mathsf{reqa}}(\boldsymbol{x})$ sont donnés dans les équations (2.1) et (2.3), respectivement.

Proposition 2.1. *Pour chaque $m \in \mathcal{M}$, $g_m(\lambda)$ diminue avec λ.*

La propriété décroissante de $g_m(\lambda)$ par rapport à λ est exploitée pour concevoir l'algorithme de mise en cache. Plus précisément,$\lambda_M^{\mathsf{reqa}}(\boldsymbol{x})$ et $\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)$ augmentent fonctions de $\boldsymbol{x}$ et $\boldsymbol{x}_m$ pour tous $m \in \mathcal{M}$ in (2.6). Par conséquent, pour minimiser le ratio de demandes manquées pour un $\boldsymbol{K}^*$ donné, chaque BS doit mettre en cache sa pleine capacité de stockage pour atteindre un λ plus élevé. Étant donné que la résolution du problème de mise en cache (2.5) nécessite de manière optimale un calcul étendu en raison du vecteur de mise en cache binaire $\boldsymbol{x}$, nous proposons un algorithme

itératif pour résoudre le problème (2.5) en résolvant séquentiellement le problème de mise en cache de chaque BS pour une solution de mise en cache donnée d'autres BS jusqu'à la convergence.

Décision de mise en cache pour les SBS: Soit $\boldsymbol{x}^t$ la solution de mise en cache dans l'itération t. De plus, nous désignons $\mathcal{F}_0^t$ et $\bar{\mathcal{F}}_0^t$ comme les ensembles de fichiers mis en cache et non mis en cache dans le MBS dans l'itération t, respectivement. Ensuite, le sous-problème de décision de mise en cache pour SBS m dans l'itération $t+1$ peut être déclaré comme

$$\begin{aligned} \min_{\boldsymbol{x}_m} \ & \lambda^{\mathsf{miss}}(\boldsymbol{x}_m) = \lambda_{\mathsf{c}} + g_0(\lambda_M^{\mathsf{reqa}}(\boldsymbol{x}_m)) + g_m(\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)) \\ s.t. \quad & \sum_{f\in\mathcal{F}} x_{mf} \leq C_m, x_{mf} \in \{0,1\}, \ \forall f \in \mathcal{F}. \end{aligned} \tag{2.7}$$

Le problème (2.7) est toujours un programme entier mixte difficile à résoudre. Étant donné que SBS m doit mettre en cache les fichiers C_m, nous proposons un schéma de mise en cache dans lequel SBS m met en cache $C_m - \bar{C}_m$ et $\bar{C}_m$ les fichiers les plus populaires en ensembles $\mathcal{F}_0^t$ et $\bar{\mathcal{F}}_0^t$, respectivement. Désignons $\bar{C}_m^*$ comme la valeur optimale de $\bar{C}_m$, qui peut être déterminée par un algorithme de recherche de ligne puisque $\bar{C}_m^* \in [0, C_m]$.

Décisions de mise en cache pour le MBS: Le problème de mise en cache du MBS peut être déclaré comme

$$\begin{aligned} \min_{\boldsymbol{x}_0} \quad & \lambda_{\mathsf{c}} + g_0(\lambda_M^{\mathsf{reqa}}(\boldsymbol{x}_0)) + \sum_{m\in\mathcal{M}_s} g_m(\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)) \\ s.t. \quad & \sum_{f\in\mathcal{F}} x_{0f} \leq C_0, x_{0f} \in \{0,1\} \ \forall f \in \mathcal{F}. \end{aligned} \tag{2.8}$$

La fonction objective du problème (2.8) peut être écrite comme $\lambda^{\mathsf{miss}}(\boldsymbol{x}_0) = \lambda_{\mathsf{c}}^M + g_0(\sum_{f\in\mathcal{F}} \lambda_{Mf}^{\mathsf{req}}(\bar{\boldsymbol{x}}_0) x_{0f})$. Comme $g_0(\lambda)$ est une fonction décroissante, la solution optimale du problème (2.8) est obtenue lorsque $\sum_{f\in\mathcal{F}} \lambda_{Mf}^{\mathsf{req}}(\bar{\boldsymbol{x}}_0)$ est maximisé. Notons $\mathcal{C}_0^*$ comme l'ensemble des C_0 valeurs les plus élevées de $\lambda_{Mf}^{\mathsf{req}}(\bar{\boldsymbol{x}}_0)$. Ensuite, pour obtenir la solution optimale du problème (2.8)), MBS doit mettre en cache tous les fichiers dans $\mathcal{C}_0^*$.

Algorithme de mise en cache final: D'après la conception de la mise en cache décrite dans l'algorithme 2.2, nous pouvons voir qu'il crée une séquence de solutions réalisables pour le problème (2.5)) où la valeur de sa fonction objective diminue de façon monotone au fil des itérations. Par conséquent, l'algorithme 2.2 converge vers une solution réalisable.

L'algorithme 2.2 trouve des solutions de mise en cache pour les MBS et les SBS. La solution de mise en cache pour le MBS a une complexité $\mathcal{O}(MF)$. La solution de mise en cache des SBS est complexe $\mathcal{O}(\sum_{m\in\mathcal{M}} C_m) \approx \mathcal{O}(F)$. L'algorithme 2.1 a la complexité de $\mathcal{O}(K_0^{\mathsf{max}}(MF+F)) \approx \mathcal{O}(K_0^{\mathsf{max}}MF)$, qui est linéaire avec les paramètres système clés.

Algorithm 2.2. ALGORITHME DE MISE EN CACHE

Initialisation: MBS met en cache ses fichiers les plus populaires C_0 et SBS m met en cache ses fichiers C_m les plus populaires. Définissez l'itération max N et la tolérance ϵ.
$t = 0$
repeat
  $t = t + 1$
  $m = t$ modulo M
  if $m = 0$ **then**
    Effectuer la mise en cache pour MBS pour obtenir $\boldsymbol{x}_0^{t+1^*}$
  else
    Effectuer la mise en cache pour SBS m pour obtenir $\boldsymbol{x}_m^{t+1}$
    $\boldsymbol{x}_m^{t+1^*} = \text{argmax}_{\boldsymbol{x}_m^{t^*}, \boldsymbol{x}_m^{t+1}}\{\lambda^{\mathsf{miss}}(\boldsymbol{x}_m^{t^*}), \lambda^{\mathsf{miss}}(\boldsymbol{x}_m^{t+1})\}$
  end if
until $|\lambda^{\mathsf{miss}}(\boldsymbol{x}_m^{t+1}) - \lambda^{\mathsf{miss}}(\boldsymbol{x}_m^{t})| < \epsilon$ or $t > N$
Produire $\boldsymbol{x}^*$

2.2.1.4 Résultats numériques

Nous considérons un paramètre de simulation avec un seul MBS et $|\mathcal{M}_s| = 9$ SBS, chacun avec un rayon de couverture $d = 50m$, déployés dans la zone de couverture du MBS avec un rayon de couverture $R = 500m$. Nous définissons $B = 20$ et $w_m = 1,\ \forall m \in \mathcal{M}$ unités de bande passante. Le nombre de fichiers est défini $F = 100$ et le paramètre d'inclinaison Zipf $\gamma = 0.8$. Les capacités de stockage des MBS et SBS sont supposées être respectivement $C_0 = 20$ et $C_m = 5$. Les processus de demande de contenu aux MBS et SBS sont des processus de Poisson avec des taux normalisés de 10^{-5} $\mathsf{requests/s/m^2}$ et 10^{-4} $\mathsf{requests/s/m^2}$, respectivement. Les temps de service d'une demande pour le MBS et le SBS sont respectivement fixés à $10s$ et $5s$.

Figues. 2.1a et 2.1b démontrent respectivement le taux de requêtes manquantes par rapport à la bande passante système B, la capacité de mise en cache de chaque SBS C_m. Dans les deux figures, l'algorithme LSBI surpasse les trois algorithmes de base. Figues. 2.1a et 2.1b montrent également le petit écart de performance entre notre algorithme proposé et la limite de performance, ce qui confirme l'efficacité de notre framework proposé. La figure 2.1c illustre les taux de demandes manquées de l'algorithme LSBI par rapport au rayon de couverture du MBS pour différentes valeurs de γ. Le taux de demandes manquées de l'algorithme LSBI est plus petit à mesure que γ devient

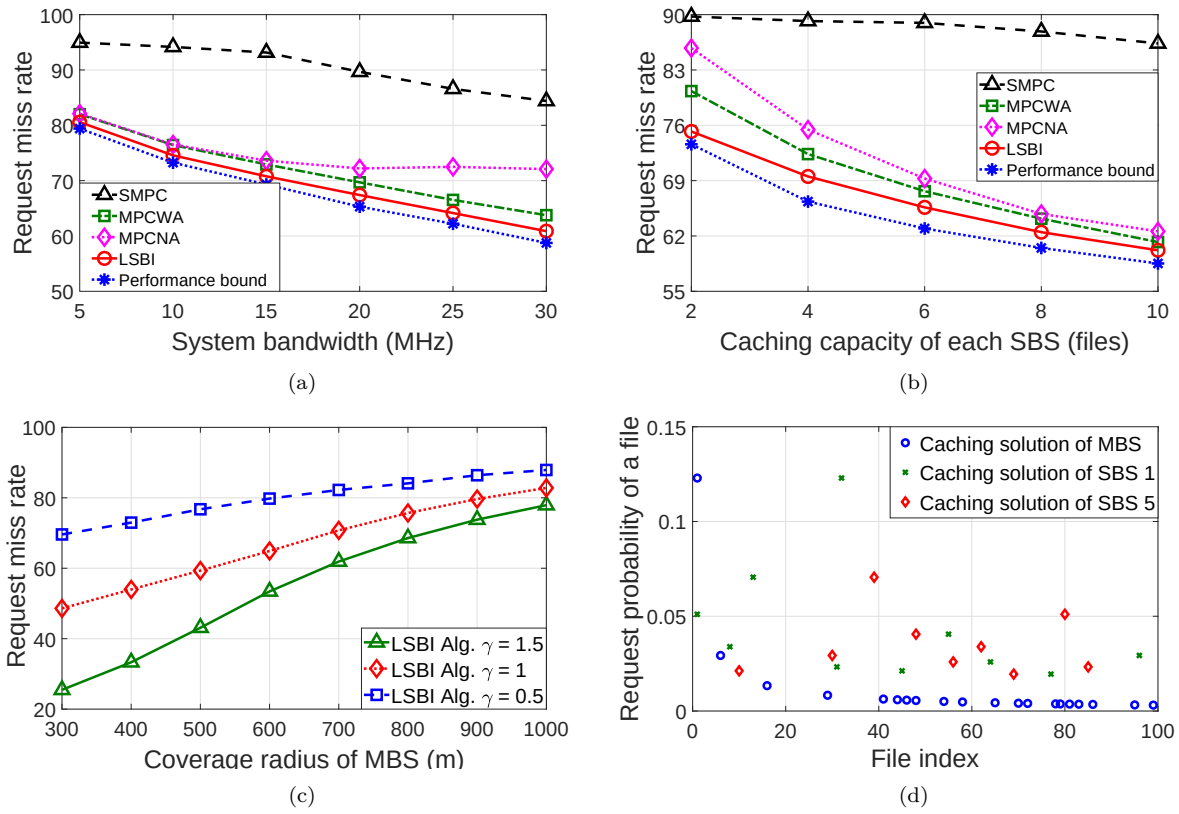


Figure 2.1 – Ratio de demandes manquées vs (a) bande passante du système B, (b) capacité de mise en cache de chaque SBS, (c) rayon de couverture du MBS. (d) Mise en cache d'une solution de MBS et SBS.

plus grand. De plus, la valeur plus élevée de R entraîne une augmentation du taux de demandes manquées, car une augmentation de R conduit à un taux de demande plus élevé. Enfin, la figure 2.1d montre les fichiers mis en cache au niveau du MBS et de 2 SBS dans une réalisation de système particulière où l'axe x indique les indices des fichiers et l'axe y montre les probabilités de demande de différents fichiers. Nous pouvons voir que les SBS ont tendance à mettre en cache leurs fichiers les plus populaires tandis que la solution de mise en cache du MBS contient des fichiers allant des probabilités de demande faibles à élevées. Cela est dû au fait que chaque SBS tenterait de minimiser le taux de demande redirigé vers le MBS en mettant en cache ses fichiers les plus populaires. De plus, le MBS accepte les demandes redirigées de tous les SBS; par conséquent, sa solution de mise en cache contient des fichiers s'étalant de préférences faibles à élevées.

2.2.2 Allocation conjointe de ressources et mise en cache de contenu dans les réseaux sans fil virtualisés

Dans cette contribution, nous étudions le problème conjoint d'allocation des ressources et de mise en cache de contenu pour les réseaux sans fil virtualisés centrés sur le contenu. Récemment, différentes solutions de mise en cache de contenu [29–31] ont été introduits pour tirer parti de l'évolution des architectures de réseau telles que les femtocellules et les réseaux basés sur C-RAN. Cependant, la plupart des travaux existants sur la mise en cache de contenu ne tiennent pas compte du scénario de réseau très encombré en raison du manque de ressources radio et de bande passante dans les liaisons d'accès sans fil et de liaison de retour [32]. En outre, dans l'environnement sans fil virtualisé où plusieurs MVNO opèrent sur l'infrastructure partagée avec une capacité de stockage limitée, la mise en cache de contenu pour l'amélioration des performances du réseau pourrait être moins importante car l'InP partitionne probablement sa capacité de stockage limitée aux BS aux MVNO. Par conséquent, la mise en cache de contenu efficace et partageable parmi les MVNO et l'optimisation de l'allocation des ressources radio peuvent effectivement améliorer les performances du réseau [3]. Motivés par les problèmes susmentionnés, nous étudions la conception conjointe d'allocation de ressources radio et de mise en cache de contenu pour les réseaux sans fil virtualisés (VWN - Virtualized Wireless Network), où nous apportons les contributions clés suivantes.

- Nous présentons la formulation du problème qui minimise la probabilité d'interruption de demande maximale pour tous les MVNO à différentes BS tout en évitant la redondance de la mise en cache de contenu aux emplacements de stockage.
- Pour résoudre le problème d'optimisation obtenu, qui est un programme non linéaire à nombres entiers mixtes (MINLP - Mixed Integer Nonlinear Program), nous proposons un algorithme basé sur la recherche en bissection qui optimise de manière itérative l'allocation des ressources et le placement de la mise en cache du contenu.
- De nombreux résultats numériques confirment l'efficacité de notre solution proposé qui réduit considérablement la probabilité d'interruption maximale des demandes par rapport à d'autres algorithmes de référence.

[3] Nous utilisons le fichier de termes et le contenu de manière interchangeable dans cette thèse de doctorat.

2.2.2.1 Modèle de système

Nous considérons un réseau sans fil multicellulaire à accès multiple par répartition en fréquence orthogonale (OFDMA - Orthogonal Frequency-Division Multiple Access) en liaison descendante avec un référentiel de mise en cache déployé à chaque BS. Le système se compose d'un ensemble $\mathcal{K} = \{1, ..., K\}$ de BS, qui sont connectés au CN via des liaisons de liaison très encombrées. On suppose que le réseau possède W^{max} canaux sans fil orthogonaux de largeur de bande égale desservant tous les UE associés à ces BS. Cette infrastructure de réseau comprenant toutes les BS, les réseaux de liaison et de base, la radio et les ressources de stockage est supposée appartenir et être gérée par un InP.

L'InP dessert un ensemble $\mathcal{M} = \{1, ..., M\}$ de MVNO, qui louent des ressources et une infrastructure réseau pour servir leurs UE. Pour plus de commodité, nous utilisons MVNO (m, k) pour désigner MVNO m associé à BS k. Pour l'allocation de canal, nous notons $\boldsymbol{w} = \{w_{11}, ..., w_{km}, ..., w_{KM}\}$ comme vecteur d'allocation de canal, dont les éléments w_{km} représente le nombre de canaux sans fil alloués au MVNO (m, k). Les UE de chaque MVNO m sont intéressés à accéder au contenu d'un ensemble de contenu commun $\mathcal{F} = \{f_1, ..., f_F\}$ des fichiers ou contenus F, dont la taille de chaque contenu est normalisé par 1 [13, 25]. Les demandes de contenu des UE de MVNO m dans la couverture de BS k sont supposées suivre le processus de Poisson avec un taux moyen λ_{km} (demandes / s).

Soit C_k la capacité du référentiel de stockage installé sur BS k, qui peut mettre en cache jusqu'à C_k fichiers où $C_k \in \mathbb{Z}_+$. De plus, $\mathcal{Q}_{km} = \{q_{km1}, ..., q_{kmF}\}$ représente la distribution de probabilité de demande de contenu où q_{kmf} représente la probabilité que les UE de MVNO (m, k) demande le fichier f. $\boldsymbol{x}_{km} = \{x_{km1}, ..., x_{kmF}\}$ est les vecteurs de décision de mise en cache pour BS $k \in \mathcal{K}$ et MVNO $m \in \mathcal{M}$. $\boldsymbol{x} = \{\boldsymbol{x}_{11}, ..., \boldsymbol{x}_{km}, ..., \boldsymbol{x}_{KM}\}$ à désigne le vecteur de décision de mise en cache de contenu pour tous les MVNO (m, k). Ici, $x_{kmf} \in \{0, 1\}$ et $x_{kmf} = 1$ si le fichier f est mis en cache à BS k pour répondre aux demandes de MVNO m et $x_{kmf} = 0$, sinon. De plus, pour garantir une exigence minimale de QoS, nous supposons qu'un canal (si disponible) doit être alloué pour télécharger un fichier demandé à partir de la BS associée pour tout UE.

2.2.2.2 Formulation du problème

Nous étudions maintenant les rejets de fichiers en raison du manque de ressources radio (c'est-à-dire qu'il n'y a pas de canal disponible) pour une solution de mise en cache donnée $\boldsymbol{x}$. Le taux de demande total de MVNO m pour tous les fichiers dans $\mathcal{F}$, s'ils sont mis en cache à BS k, est

$$h_{km}\left(\boldsymbol{x}\right)=\sum_{f\in\mathcal{F}}\lambda_{km}q_{kmf}\left(\sum_{i\in\mathcal{M}}x_{kif}\right). \tag{2.9}$$

Sinon, le taux total de demandes de fichiers manqués dans le cache de MVNO m à tous les fichiers dans $\mathcal{F}$ à BS k est

$$\bar{h}_{km}\left(\boldsymbol{x}\right)=\sum_{f\in\mathcal{F}}\lambda_{km}q_{kmf}\left(1-\sum_{i\in\mathcal{M}}x_{kif}\right)=\lambda_{km}-h_{km}\left(\boldsymbol{x}\right). \tag{2.10}$$

Notez que tous ces processus d'arrivée impliqués sont des processus de Poisson car le fractionnement ou la fusion de processus de Poisson crée des processus de Poisson. Supposons qu'il faut T_{km} (s) à BS k pour répondre à une demande de fichier avec accès en cache de MVNO m. T_{km} représente le temps de téléchargement du cache de contenu vers l'UE de MVNO (m,k). Avec w_{km} canaux alloués par l'InP à MVNO (m,k) pour répondre aux demandes des UE, au plus w_{km} les demandes de fichiers de MVNO m peuvent être servies simultanément par son BS associé . Les demandes de fichiers de MVNO m à BS k peuvent être modélisées comme une file d'attente $M/D/w_{km}/w_{km}$ avec arrivées de Poisson, temps de service déterministe, serveurs w_{km}, et aucun tampon d'attente [27].

Nous supposons que toutes les demandes de fichiers manqués dans le cache sont rejetées en raison du délai élevé de téléchargement du contenu à partir du CN. De plus, toute demande de fichier d'accès au cache de MVNO m à BS k n'est rejetée que si tous les canaux w_{km} sont utilisés pour traiter d'autres demandes w_{km} en cours. A partir de [27], la probabilité qu'il y ait w_{km} demandes de fichier en cache en cours de MVNO m servies par BS k peut être calculée comme

$$P_{km}(\boldsymbol{x},\boldsymbol{w})=\frac{\left(h_{km}(\boldsymbol{x})T_{km}\right)^{w_{km}}}{w_{km}!}\left(\sum_{i=0}^{w_{km}}\frac{\left(h_{km}(\boldsymbol{x})T_{km}\right)^{i}}{i!}\right)^{-1}. \tag{2.11}$$

Par conséquent, le taux de rejet pour la demande d'accès au cache de MVNO m à BS k en raison de l'indisponibilité du canal peut être exprimé comme

$$\mu_{km}\left(\boldsymbol{x},\boldsymbol{w}\right)=h_{km}(\boldsymbol{x})P_{km}\left(\boldsymbol{x},\boldsymbol{w}\right). \tag{2.12}$$

À partir de (2.10) et (2.12), la probabilité totale d'interruption de demande de fichier de MVNO m à BS k peut être calculée comme suit:

$$\Phi_{km}\left(\boldsymbol{x},\boldsymbol{w}\right)=\frac{\mu_{km}\left(\boldsymbol{x},\boldsymbol{w}\right)+\bar{h}_{km}\left(\boldsymbol{x}\right)}{\lambda_{km}}. \tag{2.13}$$

Pour éviter une mauvaise qualité de service et un traitement injuste dans le traitement des demandes de fichiers de différents MVNO à différentes BS, nous considérons le problème conjoint d'optimisation de l'allocation des canaux et de la mise en cache du contenu qui minimise la plus forte probabilité d'interruption parmi les MVNO à toutes les BS tout en tenant compte de l'évitement de la redondance de la mise en cache des fichiers et autres contraintes du système. Ce problème peut être formulé comme suit:

$$\min_{\boldsymbol{x},\boldsymbol{w}} \quad \max_{k\in\mathcal{K},m\in\mathcal{M}} \Phi_{km}\left(\boldsymbol{x},\boldsymbol{w}\right) \tag{2.14a}$$

$$\text{s.t.} \quad \sum_{m\in\mathcal{M}} x_{kmf}\leq 1,\forall k\in\mathcal{K},\forall f\in\mathcal{F} \tag{2.14b}$$

$$\sum_{m\in\mathcal{M}}\sum_{f\in\mathcal{F}} x_{kmf}\leq C_k,\forall k\in\mathcal{K} \tag{2.14c}$$

$$w_{km}\geq W_{km}^{\mathsf{min}},\forall k\in\mathcal{K},\forall m\in\mathcal{M} \tag{2.14d}$$

$$\sum_{k\in\mathcal{K}}\sum_{m\in\mathcal{M}} w_{km}\leq W^{\mathsf{max}} \tag{2.14e}$$

$$x_{kmf}\in\{0,1\}\ \forall k\in\mathcal{K},\forall m\in\mathcal{M},\forall f\in\mathcal{F}, \tag{2.14f}$$

où (2.14b) et (2.14c) capturent respectivement les contraintes d'évitement de redondance des fichiers et de capacité de stockage; (2.14d) représente les contraintes d'accord de niveau de service (SLA - Service Level Agreement) pour MVNO m à BS k, qui garantit un certain nombre minimum de canaux alloués pour chaque MVNO; (2.14e) dénote la contrainte de bande passante; et (2.14f) désigne le variables de décision de mise en cache d'entier au niveau des BS.

Algorithm 2.3. Allocation de canaux pour une solution de mise en cache donnée

1: **allouer** des canaux W_{km}^{min} à MVNO m à BS k pour satisfaire (2.14d).
2: **calculer** $W^{\mathsf{free}} = W^{\mathsf{max}} - \sum_{k\in\mathcal{K}} \sum_{m\in\mathcal{M}} W_{km}^{\mathsf{min}}$
3: **while** $W^{\mathsf{free}} > 0$ **do**
4: **trouver** $(k^*, m^*) = \underset{k,m}{\operatorname{argmax}}\, \Phi_{km}(\boldsymbol{x}, \boldsymbol{w})$
5: $w_{k^*m^*} = w_{k^*m^*} + 1$
6: $W^{\mathsf{free}} = W^{\mathsf{free}} - 1$
7: **end while**
8: **obtenir** $\boldsymbol{w}^*$ optimal

2.2.2.3 Algorithmes proposés

L'objectif principal de cette contribution est de résoudre le problème (2.14), qui est un MINLP en raison des variables entières $\boldsymbol{x}$ et $\boldsymbol{w}$ et de la fonction non linéaire $\Phi_{km}(\boldsymbol{x}, \boldsymbol{w})$. Nous proposons un algorithme itératif en deux étapes pour trouver l'allocation des canaux et le placement de la mise en cache du contenu à chaque itération. La procédure globale peut être illustrée comme suit:

$$\underbrace{\boldsymbol{x}^*_{(\mathbf{0})} \rightarrow \boldsymbol{w}^*_{(\mathbf{0})}}_{\text{Initialization, } \varphi_{(0)}} \rightarrow \ldots \rightarrow \underbrace{\boldsymbol{x}^*_{(i)} \rightarrow \boldsymbol{w}^*_{(i)}}_{\text{Iteration } i, \varphi_{(0)}} \rightarrow \ldots \rightarrow \underbrace{\boldsymbol{x}^* \rightarrow \boldsymbol{w}^*}_{\text{Optimal } \varphi^*},$$

où la condition d'arrêt est $|\varphi_{(i)} - \varphi_{(i-1)}| < \varepsilon$ with $0 < \varepsilon << 1$. Sur la base de la solution de mise en cache $\boldsymbol{x}^*{}_{(i-1)}$ obtenue lors de l'itération précédente $(i-1)$. Nous proposons l'algorithme 2.3, qui est basé sur sur la propriété de $P_{km}(\boldsymbol{x}^*, \boldsymbol{w})$ indiqué dans la proposition 2.2, pour trouver l'allocation optimale des canaux $\boldsymbol{w}^*{}_{(i)}$ dans l'itération i.

Proposition 2.2. *(i) Pour un $\boldsymbol{x}^*$, $P_{km}(\boldsymbol{x}^*, \boldsymbol{w})$ in (2.11) est une fonction décroissante de $\boldsymbol{w}$. (ii) Pour un $\boldsymbol{w}^*$ donné, $P_{km}(\boldsymbol{x}, \boldsymbol{w}^*)$ est une fonction croissante de $\boldsymbol{x}$.*

Lemma 2.1. *Pour une stratégie de mise en cache donnée $\boldsymbol{x}^*$, l'algorithme 2.3 alloue de manière optimale les canaux aux MVNO individuels à tous les BS pour minimiser la plus grande probabilité d'interruption de demande dans le réseau.*

Après avoir obtenu $\boldsymbol{w}^*{}_{(i)}$, nous procédons à la recherche de la solution de mise en cache de contenu $\boldsymbol{x}^*{}_{(i)}$ dans l'itération i en résolvant ce qui suit problème

$$\min_{\boldsymbol{x}} \quad \max_{k\in\mathcal{K},m\in\mathcal{M}} \Phi_{km}\left(\boldsymbol{x},\boldsymbol{w}^*\right) \tag{2.15a}$$

$$\text{s.t.} \quad \sum_{m\in\mathcal{M}} x_{kmf} \leq 1, \forall k \in \mathcal{K}, \forall f \in \mathcal{F} \tag{2.15b}$$

$$\sum_{m\in\mathcal{M}} \sum_{f\in\mathcal{F}} x_{kmf} \leq C_k, \forall k \in \mathcal{K} \tag{2.15c}$$

$$x_{kmf} \in [0,1]\ \forall k \in \mathcal{K}, \forall m \in \mathcal{M}, \forall f \in \mathcal{F}. \tag{2.15d}$$

Ici, $\boldsymbol{x}$ est relâché en variable continue comme indiqué dans (2.15d). On réécrit ensuite $\Phi_{km}\left(\boldsymbol{x},\boldsymbol{w}^*\right)$ en fonction de $h_{km}(\boldsymbol{x})$, i.e., $\Phi_{km}\left(h_{km}(\boldsymbol{x}),\boldsymbol{w}^*\right)$.

Proposition 2.3. *$\Phi_{km}\left(h_{km}(\boldsymbol{x}),\boldsymbol{w}^*\right)$ est une fonction convexe de $h_{km}(\boldsymbol{x})$ pour un $\boldsymbol{w}^*$ donné. De plus, c'est une fonction décroissante de $h_{km}(\boldsymbol{x})$.*

Par conséquent, $\max_{k\in\mathcal{K},m\in\mathcal{M}} \Phi_{km}\left(h_{km}\left(\boldsymbol{x}\right),\boldsymbol{w}\right)$ peut être considéré comme la fonction maximale ponctuelle sur $h_{km}(\boldsymbol{x})$, qui est convexe [33]. Cela nous permet de transformer le problème (2.15) en problème d'optimisation convexe suivant sur $\boldsymbol{h}$, où $\boldsymbol{h} = \{h_{km}\left(\boldsymbol{x}\right)\}, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}$.

$$\min_{\boldsymbol{h},\varphi} \quad \varphi \tag{2.16a}$$

$$\text{s.t.} \quad \Phi_{km}\left(h_{km}(\boldsymbol{x}),\boldsymbol{w}^*\right) \leq \varphi, \forall k \in \mathcal{K}, \forall m \in \mathcal{M} \tag{2.16b}$$

$$h_{km}\left(\boldsymbol{x}\right) \in \mathcal{H}, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}. \tag{2.16c}$$

Dans (2.16), $\mathcal{H}$ désigne l'ensemble de toutes les valeurs possibles de $h_{km}\left(\boldsymbol{x}\right)$, qui dépend de l'ensemble possible de $\boldsymbol{x}$ selon les contraintes du problème 2.15.

$$\max_{\boldsymbol{x}} \quad \sum_{k\in\mathcal{K}}\sum_{m\in\mathcal{M}} h_{km}\left(\boldsymbol{x}\right) \tag{2.17a}$$

$$\text{s.t.} \quad h_{km}\left(\boldsymbol{x}\right) \geq h_{km}^{\mathsf{low}}, \forall m \in \mathcal{M}, \forall k \in \mathcal{K} \tag{2.17b}$$

$$\sum_{m\in\mathcal{M}} x_{kmf} \leq 1, \forall k \in \mathcal{K}, \forall f \in \mathcal{F} \tag{2.17c}$$

$$\sum_{m\in\mathcal{M}}\sum_{f\in\mathcal{F}} x_{kmf} \leq C_k, \forall k \in \mathcal{K} \tag{2.17d}$$

$$x_{kmf} \in [0,1]\, \forall m \in \mathcal{M}, \forall f \in \mathcal{F}. \tag{2.17e}$$

Pour résoudre 2.16, nous proposons l'algorithme basé sur la recherche de bissection 2.4 pour déterminer la solution de décision de mise en cache $\boldsymbol{x}^*{}_{(i)}$. Avec les nouveaux $\boldsymbol{w}^*{}_{(i)}$ et $\boldsymbol{x}^*{}_{(i)}$, nous calculons la probabilité maximale d'interruption de la demande $\varphi_{(i)}$ pour l'itération i. La méthode de recherche de Newton pour calculer le taux de succès h_{km} de MVNO m à BS k, compte tenu de la probabilité d'interruption de la demande φ_{km} et de l'allocation de canal w_{km}. Après cela, nous résolvons le problème (2.17) pour trouver la solution détendue $\boldsymbol{x}^*{}_i$, i.e., $\boldsymbol{x}^*{}_i \in [0,1]$. Ici, h_{km}^{low} est la sortie de la fonction inverse Φ_{km}^{-1} prenant φ en entrée. Après avoir exécuté l'algorithme 2.4, nous procédons à l'arrondi de la décision de mise en cache obtenue $\boldsymbol{x}^*$ en valeurs entières en utilisant l'algorithme 2.5.

2.2.2.4 Résultats numériques

Dans cette section, nous évaluons les performances de nos algorithmes proposés par simulation informatique sous le paramètre suivant. Nous considérons le réseau avec 5 BS desservant 3 MVNO, qui accèdent à une liste de 100 fichiers, soit $K = 5, M = 3$ et $F = 100$. Les taux de demande moyens pour chaque MVNO sont choisis au hasard dans la fourchette de $[1, 15]$, ce qui se traduit par le total des taux de demande de dizaines à des centaines de demandes arrivant sur le réseau considéré en une seconde. Nous supposons que toutes les BS partagent W^{max} canaux sans fil de manière orthogonale pour répondre aux demandes de fichiers des MVNO. Chaque exigence SLA est définie avec $W_{km}^{\mathsf{min}} = 2, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}$.

Algorithm 2.4. Allocation itérative de canaux et placement de la mise en cache de contenu

fixer $i = 1$ et tolérance $\varepsilon > 0$.
initialiser $\boldsymbol{x}^*_{(i)}$ selon la stratégie de mise en cache la plus populaire avec une partition de stockage égale.
initialiser l'allocation des canaux $\mathbf{w}_{(i)}$ en utilisant l'algorithme 1.3 étant donné $\boldsymbol{x}^*_{(\boldsymbol{i})}$.
calculer $\Phi_{km}, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}$.
trouver la plus grande probabilité d'interruption $\varphi_{(i)} = \max_{k,m} \Phi_{km}$.
fixer $\Delta_\varphi = 1$
while $\Delta_\varphi > \varepsilon$ **do**
  $i = i + 1$
  fixer $\phi^{\mathsf{up}} = 1$
  fixer $\phi^{\mathsf{low}} = 0$
  while $\phi^{\mathsf{up}} - \phi^{\mathsf{low}} > \varepsilon$ **do**
    $\phi_{(i)} = \left(\phi^{\mathsf{up}} + \phi^{\mathsf{low}}\right)/2$
    trouver h_{km} à partir de $\phi_{(i)}$ en utilisant la méthode de recherche de Newton pour tous les $m \in \mathcal{M}$ et $k \in \mathcal{K}$.
    résoudre le problème (2.17) pour trouver $\boldsymbol{x}^*$.
    if $\boldsymbol{x}^*$ est faisable **then**
      $\phi^{\mathsf{up}} = \phi_{(i)}$
      $\boldsymbol{x}^*_{(\boldsymbol{i})} = \boldsymbol{x}^*$
    else
      $\phi^{\mathsf{low}} = \phi_{(i)}$
    end if
  end while
  trouver $\boldsymbol{w}^*_{(\boldsymbol{i})}$ optimal en utilisant l'algorithme 1.3.
  calculer $\Phi^{(i)}_{km}, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}$.
  trouver $\varphi_{(i)} = \max_{k,m} \Phi^{(i)}_{km}$
  calculer $\Delta_\varphi = |\varphi^*_{(i)} - \varphi^*_{(i-1)}|$
end while
obtenir $\boldsymbol{w}^*$ final et $\boldsymbol{x}^*$ de l'algorithme 1.3 donné $\boldsymbol{x}^*_{(\boldsymbol{i})}$.

Algorithm 2.5. Arrondir les variables de décision de mise en cache

initialiser petit $\varepsilon > 0$
obtenir la valeur de probabilité d'interruption de requête optimale φ à partir de l'algorithme 1.4.
repeat
  obtenir $h^{\mathsf{low}}_{km}, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}$ en utilisant la méthode de recherche de Newton.
  résoudre le problème (2.17) avec une contrainte intégrale.
  if la solution intégrale $\boldsymbol{x}^*_{\mathsf{INT}}$ est introuvable **then**
    $\varphi = \varphi + \varepsilon$
  end if
until la solution intégrale $\boldsymbol{x}^*_{\mathsf{INT}}$ est trouvée.

La Fig. 2.2a montre que la proposition Un algorithme basé sur la recherche de bissection (l'algorithme 2.4) avec partage de cache atteint systématiquement la plus petite probabilité d'interruption de requête maximale. De plus, l'opération d'arrondi proposée pour la mise en cache des variables de décision entraîne une perte de performances négligeable par rapport aux performances obtenues

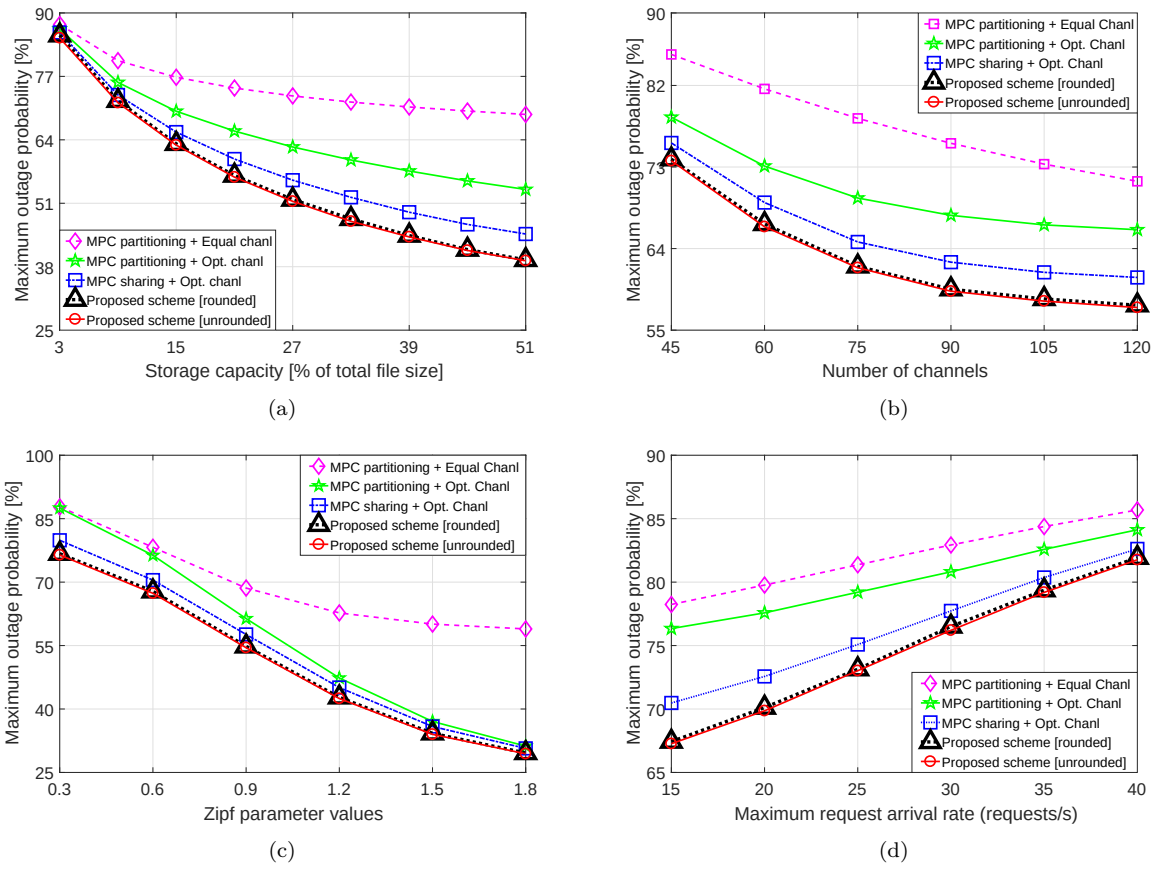


Figure 2.2 – Probabilité d'interruption maximale par rapport à (a) la capacité de stockage, (b) le nombre de canaux, (c) le paramètre Zipf et (d) le taux d'arrivée maximal des demandes.

avant arrondi, ce qui confirme l'efficacité de notre conception (la probabilité d'interruption de la demande obtenue sous relaxation de l'algorithme 3 est la borne inférieure de la valeur optimale) . De plus, l'algorithme heuristique proposé atteint des performances très proches de l'algorithme basé sur la recherche de bissection proposé dans le cas de popularité d'ordre différent, et les deux algorithmes aboutissent à la même solution dans le cas de popularité des fichiers du même ordre.

Nous présentons la probabilité d'interruption de demande maximale parmi les MVNO à tous les BS par rapport au nombre total de canaux, le paramètre Zipf γ et le taux de demande maximal sur les Figs. 2.2b, 2.2c, et 2.2d. Semblable à la figure 2.2a, Fig. 2.2b confirme les meilleures performances de notre algorithme basé sur la recherche de bissection proposée car il atteint la probabilité d'interruption de requête la plus faible par rapport aux lignes de base restantes. Les Figs. 2.2a et 2.2b impliquent également qu'au lieu de partitionner l'espace de stockage disponible en MVNO individuels, il vaut mieux le partager entre les MVNO co-localisés sur le même BS.

2.2.3 Allocation de ressources pour le découpage de réseaux multi-locataires: Une approche de jeu stackelberg multi-leaders multi-suiveurs

Le découpage de réseau fournit également le changement de paradigme vers la multi-location dans le réseau sans fil de nouvelle génération [23] où les locataires individuels (e.g., MVNOs, SPs) possèdent et gèrent les tranches de réseau correspondantes. Diverses approches théoriques du jeu ont été appliquées pour résoudre différents problèmes d'allocation de ressources dans le contexte de découpage de réseau [34–37]. Les interactions multilatérales entre les SPs et leurs clients tels que les UEs, qui constituent un marché d'échange de services, peuvent être modélisées en utilisant la théorie des jeux de Stackelberg. De nombreux travaux dans l'allocation des ressources [38–43] et d'autres domaines de l'industrie [44, 45] ont appliqué le jeu Stackelberg à un seul leader à plusieurs suiveurs (SLMF - Single-Leader-Multi-Follower). Le jeu Stackelberg multi-leader-multi-suiveur (MLMF - Multi-Leader-Multi-Follower) a été utilisé dans certains travaux récents [46–50]. Néanmoins, l'étude des interactions entre les fournisseurs de services d'accès homologues (ASP - Access Service Provider) et entre les ASP et leurs UE dans le réseau sans fil basé sur le découpage du réseau n'a pas été envisagée. Les travaux connexes ci-dessus ne prennent en compte que la sélection de services à source unique entre les parties prenantes, i.e., qu'un UE ne peut sélectionner qu'un seul SP pour l'achat de services. Dans notre travail, nous étudions donc le problème d'allocation des ressources et de tarification du découpage de réseau qui capture les interactions entre les fournisseurs de services d'accès / de liaison et leurs UE en utilisant l'approche du jeu MLMF Stackelberg. De plus, nous permettons à tout UE et ASP d'être en mesure de louer des services à différents ASP et fournisseurs de services de liaison (BSP - Backhaul Service Provider) en même temps, respectivement, permettant ainsi la connectivité à plusieurs tranches. Nos contributions dans cette étude sont les suivantes.

- Nous formulons les interactions entre les UE, les ASP comme un jeu MLMF Stackelberg [51]. Plus précisément, les ASP et les UE agissent respectivement en tant que leaders et suiveurs dans ce jeu. Chaque ASP lui propose un prix d'accès pour les UE en fonction de leurs demandes ainsi que des politiques de tarification des autres ASP. De plus, chaque ASP optimise la capacité de backhaul achetée auprès des BSP pour faire face à sa demande d'accès.
- Nous dérivons les fonctions de meilleure réponse de prix pour les ASP et la fonction de meilleure réponse de débit pour les UE dans la couche d'accès. Nous prouvons l'existence d'un équilibre de jeu Stackelberg unique. Nous prouvons en outre que ces fonctions de meilleure

réponse appartiennent à la classe des fonctions standard [52] et qu'elles satisfont la propriété dite d'évolutivité bilatérale (2.s.s - two-sided scalability) [53].

- Ces résultats ci-dessus sont mis à profit dans le développement d'un algorithme distribué qui converge vers l'équilibre du jeu.
- Nous évaluons l'efficacité de notre cadre proposé et étudions les performances et les stratégies réalisables des différentes parties prenantes du réseau via des études numériques approfondies.

2.2.3.1 Modèle de système

Nous considérons la liaison descendante d'un réseau cellulaire avec à la fois des liaisons sans fil et des communications d'accès. Nous supposons que les infrastructures et les ressources sans fil des couches de liaison et d'accès sans fil sont détenues et gérées par un ensemble $\mathcal{I} = \{1, ..., i, ..., I\}$ de service de liaison (BSP) et un ensemble $\mathcal{J} = \{1, ..., j, ..., J\}$ ASP, respectivement. Nous supposons que BSP $i \in \mathcal{K}$ possède un hub de liaison sans fil correspondant (WBH - Wireless Backhaul Hub). Chaque BSP i a une bande passante suffisamment grande et dédiée, non chevauchée avec les spectres des autres BSP. Chaque ASP j possède un BS et la bande de spectre de W_j^{A} Hz non chevauchée avec les bandes des autres ASP. Nous supposons que ces BS servent un ensemble $\mathcal{K} = \{1, ..., k, ..., K\}$ d'UE dans une zone de service particulière (c'est-à-dire une cellule) et les bandes de spectre utilisés par ces ASP ne se chevauchent pas. Par conséquent, il n'y a pas d'interférence dans le même canal entre les ASP dans cette section de réseau considérée. En pratique, les ASP peuvent réutiliser leur spectre dans d'autres zones suffisamment éloignées, introduisant ainsi certaines interférences intercellulaires. Cependant, on peut considérablement atténuer ces interférences en planifiant soigneusement les cellules. En outre, le brouillage moyen dans le même canal peut être estimé et accumulé dans la puissance de bruit de fond pour les utilisateurs de la zone de service considérée [48].

Nous supposons en outre que les ASP doivent acheter des ressources de communication de raccordement auprès des BSP pour prendre en charge les communications de bout en bout entre les UE et le réseau central (CN). Les interactions entre les différentes parties prenantes du réseau pour l'échange de ressources se produisent dans des intervalles de temps de taille fixe où le nombre d'utilisateurs actifs reste le même dans chaque intervalle de temps. Soit w_{ij} la quantité de bande passante (Hz) qu'ASP j acquiert du BSP i, et r_{ij} (bps/Hz) est l'efficacité moyenne du spectre atteinte par le correspondant lien de liaison. Nous supposons que l'UE k achète une fraction

$s_{jk} \in [0, 1]$ de la ressource spectrale de l'aSP j pour la transmission de données dans la couche d'accès; alors, l'efficacité spectrale moyenne atteinte par la liaison d'accès entre UE k et ASP j est notée comme r_{jk} (bps/Hz).

Nous supposons que chaque ASP peut transférer des données sur plusieurs liaisons de liaison via plusieurs connexions avec différents BSP simultanément. De même, chaque UE peut recevoir simultanément des données sur plusieurs liaisons d'accès associées à différents ASP. Dans la pratique, cela peut être réalisé par la technologie multi-connexions [54–57], les techniques d'agrégation de données multibandes [58, 59], les techniques de pilotage du trafic [60–64], et les protocoles de prise en charge [65, 66]. En conséquence, les ASP (UE) peuvent acheter de la bande passante de backhaul (accès) à partir de différents BSP (ASP) en même temps, permettant ainsi la connectivité à plusieurs tranches [67] pour les UE et les ASP. Les interactions entre ces parties prenantes du réseau et les utilisateurs finaux constituent un marché commercial comprenant les marchés des liaisons et des ressources d'accès.

2.2.3.2 Formulation du jeu MLMF Stackelberg

Dans cette formulation de jeu, les ASP et les UE agissent respectivement comme les leaders et les suiveurs. Les leaders jouent en premier dans la phase I en imposant des prix d'accès en tenant compte des budgets et des demandes potentiels des UE, et les ASP décident de la quantité de ressources de bande passante acquise auprès des BSP pour minimiser les coûts. Dans la deuxième étape (étape II), chaque UE achète de manière optimale les données des ASP pour maximiser son utilité compte tenu des prix imposés par les ASP. Chaque joueur maximise égoïstement sa propre fonction de gain dans ce jeu MLMF Stackelberg.

Étape I: Jeu de tarification de l'accès non coopératif entre les asps et l'acquisition de ressources de liaison Le gain de ASP j peut être défini comme le revenu généré par la fourniture de services d'accès aux UE moins les dépenses de raccordement. Plus précisément, l'aSP $j \in \mathcal{J}$ est intéressé à maximiser la fonction de gain suivante:

$$P_j(p_j, \boldsymbol{s}_j, \boldsymbol{w}_j) = \delta_j U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j) - \eta_j C_j^{\mathsf{A}}(\boldsymbol{w}_j) \tag{2.18}$$

où δ_j et η_j sont respectivement les coefficients associés à la fonction de revenu $U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j)$ et la fonction de coût $C_j^{\mathsf{A}}(\boldsymbol{w}_j)$. $\boldsymbol{w}_j = (w_{ij})_{i\in\mathcal{I}}$ désigne le vecteur de la bande passante de liaison achetée par l'aSP j auprès des BSP. $\boldsymbol{s}_j = (s_{jk})_{k\in\mathcal{K}}$ représente le vecteur des fractions de bande passante d'accès qu'ASP j alloue aux UE associés pour répondre à leurs demandes de débit $(d_{jk})_{k\in\mathcal{K}}$. La fonction de revenu $U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j)$ est définie comme

$$U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j) = \sum_{k=1}^{K} p_j d_{jk} - \theta_j \sum_{k=1}^{K} W_j^{\mathsf{A}} s_{jk}. \tag{2.19}$$

où θ_j est le coefficient associé. Parce que chaque ASP doit acheter la bande passante de liaison pour desservir ses UE, l'aSP j doit payer le coût de liaison $C_j^{\mathsf{A}}(\boldsymbol{w}_j)$, qui est défini comme suit :

$$C_j^{\mathsf{A}}(\boldsymbol{w}_j) = \sum_{i=1}^{I} q_i w_{ij} + \frac{1}{2}\sum_{i=1}^{I} w_{ij}^2 + \nu_j \sum_{i'\neq i} w_{ij} w_{i'j} \tag{2.20}$$

où q_i désigne le prix par unité de bande passante de liaison (\$/Hz) offert par le BSP i. Dans cet article, nous définissons $\nu_j \in (0, 1)$ afin que l'ASP j puisse transmettre des données dans différents spectres selon la propriété de substituabilité ci-dessus, garantissant ainsi la capacité de transmission multibande pour chaque ASP.

Pour simplifier, nous considérons l'optimisation des deux fonctions $U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j)$ et $C_j^{\mathsf{A}}(\boldsymbol{w}_j)$ indépendamment. Plus précisément, l'aSP j doit fixer le prix concurrentiel p_j pour attirer plus de demande des UE compte tenu de la ressource de spectre disponible W_j^{A} et des prix offerts par d'autres ASP. La maximisation des revenus des ASP individuels en tenant compte des prix des autres ASP constitue donc un jeu appelé *Étape-I* AG *jeu*, qui sera défini plus loin.

Le jeu Étape-I $sfAG$ *- Le jeu de tarification des ressources d'accès*

La fonction utilitaire (2.19) de l'aSP j peut être exprimée comme $U_j^{\mathsf{A}}(p_j, \boldsymbol{p}_{-j}, \boldsymbol{s}_j)$ pour décrire les interactions avec d'autres ASP. Ici, $\boldsymbol{p}_{-j} = (p_{j'})_{j'\in\mathcal{J}, j'\neq j}$ dénote les stratégies d'autres ASP sauf ASP j. Maintenant, le jeu AG peut être défini comme suit:

1. *Joueurs:* The ASPs in the set $\mathcal{J}$.

2. *Stratégie:* $p_j^{\mathsf{L}} \leq p_j \leq p_j^{\mathsf{U}},\ \forall j \in \mathcal{J}$ tel que

$$\sum_{k=1}^{K} s_{jk} \leq 1, \forall j \in \mathcal{J}. \tag{2.21}$$

3. *Fonction d'utilité:* $U_j^{\mathsf{A}}(p_j, \boldsymbol{p}_{-j}, \boldsymbol{s}_j),\ \forall j \in \mathcal{J}$.

Dans le jeu AG, p_j^{L} et p_j^{U} sont les limites inférieures et supérieures positives de le prix d'accès, imposé par la réglementation du marché pour l'aSP j. Notez que (2.21) représente la contrainte d'allocation de bande passante d'accès (fraction) pour chaque ASP j.

Étape II: Optimisation de la demande de trafic de chaque UE: La fonction de gain de chaque UE $k \in \mathcal{K}$ est définie comme suit:

$$U_k^{\mathsf{E}}(\boldsymbol{d}_k) = e_k \sum_{j=1}^{J} \log\left(1 + d_{jk}\right) \tag{2.22}$$

où $\boldsymbol{d}_k = (d_{jk})_{j \in \mathcal{J}}$ désigne le vecteur de demande de débit de l'UE k, $d_{jk} = s_{jk} W_j^{\mathsf{A}} r_{jk}$ est le débit de UE k pris en charge par ASP j, et e_k désigne le coefficient d'utilité de UE k.

Étant donné le prix imposé par les leaders (ASP) $\boldsymbol{p}^*$, UE k est intéressé à maximiser son gain en acquérant un débit approprié pour différents ASP compte tenu de son budget maximum B_k. Mathématiquement, ce problème d'optimisation de la demande de débit peut être formulé comme

$$\max_{\boldsymbol{d}_k \geq \mathbf{0}} U_k^{\mathsf{E}}(\boldsymbol{d}_k) \tag{2.23a}$$

$$\text{s.t.} \sum_{j=1}^{J} p_j d_{jk} \leq B_k \tag{2.23b}$$

où (2.23b) capture la contrainte budgétaire maximale de UE k.

2.2.3.3 Analyse de l'équilibre du jeu MLMF Stackelberg

Nous dérivons l'équilibrage de Stackelberg (SE - Stackelberg Equilibrium) du jeu de Stackelberg considéré en utilisant la méthode *induction vers l'arrière*. Plus précisément, nous dérivons d'abord la demande de débit optimal des UE dans la phase II du jeu Stackelberg. Ensuite, nous utilisons ce résultat pour dériver le NE parmi les ASP de l'étape I du jeu.

i. Demande de débit optimal des UE en phase II: Nous indiquons d'abord la demande de débit optimale des UE dans la phase II du jeu Stackelberg dans le lemme suivant.

Lemma 2.2. *Pour des prix donnés $(p_j)_{j\in\mathcal{J}}$ et budget B_k, la demande de débit optimale de UE k obtenue en résolvant le problème* (2.23) *peut être exprimée comme suit :*

$$d_{jk}^{*}=\left[\frac{B_k+\sum_{j'=1}^{J}p_{j'}}{Jp_j}-1\right]^{+}=\left[\frac{B_k+\sum_{j'\neq j}p_{j'}}{Jp_j}+\frac{1}{J}-1\right]^{+},\ \forall j\in\mathcal{K} \tag{2.24}$$

où $[x]^{+}=\max\{0,x\}$.

ii. Solution de tarification et d'allocation de ressources pour la couche d'accès dans le sous-jeu l'Étape-I AG *:* La portion de bande passante allouée par ASP j à UE k peut être exprimée comme

$$s_{jk}^{*}=\frac{B_k+\sum_{j'\neq j}p_{j'}}{JD_{jk}p_j}+\frac{1}{D_{jk}}\left(\frac{1}{J}-1\right)=\frac{\alpha_{jk}(\boldsymbol{p}_{-j})}{p_j}+\beta_{jk} \tag{2.25}$$

où $\alpha_{jk}(\boldsymbol{p}_{-j})\triangleq\frac{B_k+\sum_{j'\neq j}p_{j'}}{JD_{jk}}$ et $\beta_{jk}\triangleq\frac{1}{D_{jk}}\left(\frac{1}{J}-1\right)$. En substituant les résultats dans (2.24) et (2.25) dans (2.19), l'utilitaire atteint par ASP j devient

$$U_j^{\mathsf{A}}(p_j,\boldsymbol{s}_j,\boldsymbol{p}_{-j})=K(\frac{1}{J}-1)p_j+\sum_{k=1}^{K}\frac{B_k+\sum_{j'\neq j}p_{j'}}{J}-\theta_jW_j^{\mathsf{A}}\left(\frac{\alpha_j(\boldsymbol{p}_{-j})}{p_j}+\beta_j\right). \tag{2.26}$$

où $\alpha_j(\boldsymbol{p}_{-j})\triangleq\sum\limits_{k=1}^{K}\alpha_{jk}(\boldsymbol{p}_{-j})$ et $\beta_j\triangleq\sum\limits_{k=1}^{K}\beta_{jk}$. Nous utilisons la notation $U_j^{\mathsf{A}}(p_j,\boldsymbol{s}_j,\boldsymbol{p}_{-j})$ pour présenter l'impact des stratégies d'autres ASP $\boldsymbol{p}_{-j}$ à l'utilitaire d'ASP j. Maintenant, nous substituons le résultat de s_{jk}^{*} dans (2.25) à (2.21) et exploitons le fait que $p_j>0$, cette contrainte devient

$$\begin{aligned}&\sum_{k=1}^{K}\left(\frac{\alpha_{jk}(\boldsymbol{p}_{-j})}{p_j}+\beta_{jk}\right)\leq 1\\&\Leftrightarrow(1-\beta_j)p_j-\alpha_j(\boldsymbol{p}_{-j})\geq 0.\end{aligned} \tag{2.27}$$

Étant donné les stratégies $\boldsymbol{p}_{-j}$ des autres ASP, la stratégie optimale de ASP $j \in \mathcal{J}$ pour le sous-jeu l'Étape-I AG , i.e., sa meilleure réponse, est la solution du problème d'optimisation suivant:

$$F_j(\boldsymbol{p}_{-j}) = \underset{p_j \in \mathcal{P}_j}{\operatorname{argmax}} \; U_j^{\mathsf{A}}(p_j, \boldsymbol{p}_{-j})$$
$$\text{s.t. constraint (2.27).} \tag{2.28}$$

Nous affirmons la convexité de ce problème dans le lemme suivant.

Lemma 2.3. *Le problème* (2.28) *est convexe.*

Theorem 2.1. *Il existe un NE pour le sous-jeu l'Étape-I* AG*.*

Les conditions d'optimalité Karush–Kuhn–Tucker (KKT) du problème (2.28) sont données par

$$\frac{\partial \mathcal{L}_j^{\mathsf{A}}}{\partial p_j} = 0 \tag{2.29a}$$
$$\text{constraint (2.27)} \tag{2.29b}$$
$$\lambda_j \geq 0 \tag{2.29c}$$
$$\lambda_j \left[(1-\beta_j)p_j - \alpha_j\right] = 0. \tag{2.29d}$$

où $\mathcal{L}_j^{\mathsf{A}}(p_j, \lambda_j) = U_j^{\mathsf{A}}(p_j, \boldsymbol{p}_{-j}) + \lambda_j \left[(1-\beta_j)\, p_j - \alpha_j(\boldsymbol{p}_{-j})\right]$ et λ_j est le multiplicateur de Lagrange. Après avoir résolu (2.29), nous obtenons la meilleure fonction de réponse du prix offert aux UE d'un ASP $j \in \mathcal{J}$ comme suit:

$$p_j^{(t+1)} = \begin{cases} F_j^{\mathsf{h}}(\boldsymbol{p}_{-j}^{(t)}) = \left[\sqrt{\frac{\alpha_j(\boldsymbol{p}_{-j}^{(t)}) \sqrt{\alpha_j(\boldsymbol{p}_{-j}^{(t)})}}{(1-\beta_j)^2}} \right]_{\mathcal{P}_j} & \text{if } \theta_j < \frac{K\left(1-\frac{1}{J}\right)\sqrt{\alpha_j(\boldsymbol{p}_{-j})}}{W_j^{\mathsf{A}}(1-\beta_j)^2} \\ F_j^{\mathsf{n}}(\boldsymbol{p}_{-j}^{(t)}) = \left[\sqrt{\frac{\theta_j}{K(J-1)} \sum_{k=1}^{K} \frac{B_k + \sum_{j' \neq j} p_{j'}}{r_{jk}}} \right]_{\mathcal{P}_j} & \text{otherwise} \end{cases} \tag{2.30}$$

où t est l'indice d'itération. Sur la base des résultats dans Lemma 2.6, nous proposons un algorithme distribué, qui est résumé dans Algorithm 2.6, pour trouver le NE parmi les ASP par rapport aux prix $\boldsymbol{p}$ et $\boldsymbol{s}^*$.

Definition 2.1. *Une fonction $\boldsymbol{F}(\boldsymbol{p})$ est appelée standard et satisfait la propriété 2.s.s si les conditions suivantes sont réunies [53]:*

Algorithm 2.6. Algorithme distribué pour la tarification et l'allocation de bande passante d'accès

Initialiser $t = 0$ and ε.
Chaque ASP définit $p_j^{(t)} = p_j^{\mathsf{L}}, \forall j \in \mathcal{J}$.
Chaque ASP annonce ses prix $p_j^{(t)}$ sur le marché de l'accès.
// Ajustement de prix
while Vrai **do**
  $t = t + 1$
  for $j = 1$ à J **do**
    if $\theta_j < \frac{K\left(1-\frac{1}{J}\right)\sqrt{\alpha_j(\boldsymbol{p}_{-j})}}{W_j^{\mathsf{A}}(1-\beta_j)^2}$ **then**
      ASP j met à jour ses prix $p_j^{(t)}$ en fonction de F_j^{h} in (2.30)
    else
      ASP j met à jour ses prix $p_j^{(t)}$ en fonction de F_j^{n} dans (2.30)
    end if
  end for
  Chaque ASP annonce ses prix $p_j^{(t)}$ aux autres ASP.
  if $||\boldsymbol{p}^{(t+1)} - \boldsymbol{p}^{(t)}|| < \varepsilon$ **then**
    Quittez la boucle While
  end if
end while
Tous les ASP annoncent leurs prix aux UEs.
// Acquisition de bande passante de liaison
for Chaque ASP $j \in \mathcal{J}$ **do**
  Calculer s_{jk}, $\forall j \in \mathcal{J}, k \in \mathcal{K}$ selon (2.25).
  Calculer $\boldsymbol{w}_j = (w_{ij})_{i\in\mathcal{I}}$ en utilisant l'algorithme 2.7.
  Calculez le gain selon (2.18).
end for

- *Positivité:* $\boldsymbol{F}(\boldsymbol{p}) > \boldsymbol{0}$
- *Monotonicité: if* $\boldsymbol{p} \geq \boldsymbol{p'}$ *then* $\boldsymbol{F}(\boldsymbol{p}) \geq \boldsymbol{F}(\boldsymbol{p'})$
- *Évolutivité:* $\forall \mu > 1$, $\mu\boldsymbol{F}(\boldsymbol{p}) \geq \boldsymbol{F}(\mu\boldsymbol{p})$
- *Évolutivité bilatérale (2.s.s): Pour tout* $\mu > 1$, $\frac{1}{\mu}\boldsymbol{p} \leq \boldsymbol{p'} \leq \mu\boldsymbol{p}$ *implique* $\frac{1}{\mu}\boldsymbol{F}(\boldsymbol{p}) < \boldsymbol{F}(\boldsymbol{p'}) < \mu\boldsymbol{F}(\boldsymbol{p})$.

Lemma 2.4. $\boldsymbol{F}^{\mathsf{h}}(\boldsymbol{p}) = \left(F_j^{\mathsf{h}}(\boldsymbol{p}_{-j})\right)_{j\in\mathcal{J}}$, *où* $F_j^{\mathsf{h}}(\boldsymbol{p}_{-j})$ *est l'équation de la partie supérieure de* (2.30), *est une fonction standard avec la propriété 2.s.s dans le domaine* $\mathcal{P}$.

Lemma 2.5. $\boldsymbol{F}^{\mathsf{n}}(\boldsymbol{p}) = \left(F_j^{\mathsf{n}}(\boldsymbol{p}_{-j})\right)_{j\in\mathcal{J}}$, *où* $F_j^{\mathsf{n}}(\boldsymbol{p}_{-j})$ *est l'équation inférieure de* (2.30), *est une fonction standard avec la propriété 2.s.s dans le domaine* $\mathcal{P}$.

Lemma 2.6. *Les mises à jour itératives dans A pour* $F_j^{\mathsf{h}}(\boldsymbol{p}_{-j}^{(t)})$ *et* $F_j^{\mathsf{n}}(\boldsymbol{p}_{-j}^{(t)})$ *pour tous* $j \in \mathcal{J}$ *convergent vers le point fixe correspondant* $\boldsymbol{p}^{\mathsf{h}*}$ *et* $\boldsymbol{p}^{\mathsf{n}*}$, *respectivement.*

Algorithm 2.7. ACQUISITION DE BANDE PASSANTE DE RACCORDEMENT D'UN ASP

Initialiser γ_j^{L}, γ_j^{U}, $\forall j \in \mathcal{J}$, and ε.
while $||\boldsymbol{\gamma}^{\mathsf{U}} - \boldsymbol{\gamma}^{\mathsf{L}}|| > \varepsilon$ **do**
  Fixer $\gamma_j = (\gamma_j^{\mathsf{L}} + \gamma_j^{\mathsf{U}})/2$.
  Calculer $w_{ij} = \xi_j \left[\gamma_j \kappa_{ij} - \rho_{ij}\right]^+, \ \forall i \in \mathcal{I}, j \in \mathcal{J}$.
  Calculer $R_j = \sum_{i=1}^{I} w_{ij} r_{ij}, \ \forall j \in \mathcal{J}$.
  for $j = 1$ to J **do**
    if $R_j < R_j^{\mathsf{A}}(\boldsymbol{s}_j^*)$ **then**
      Réviser $\gamma_j^{\mathsf{L}} = \gamma_j$
    else
      Réviser $\gamma_j^{\mathsf{U}} = \gamma_j$
    end if
  end for
end while
Revenir γ_j, w_{ij}, $\forall i \in \mathcal{I}$.

2.2.3.4 Le problème de la minimisation des dépenses de transport

Au point d'équilibre du sous-jeu l'Étape-I AG, chaque ASP obtient la demande de débit totale des UE associés, qui est donnée par

$$R_j^{\mathsf{A}}(\boldsymbol{s}_j^*) \triangleq \sum_{k=1}^{K} s_{jk}^* W_j^{\mathsf{A}} r_{jk}, \ \forall j \in \mathcal{J} \tag{2.31}$$

où $(s_{jk}^*)_{j\in\mathcal{J},k\in\mathcal{K}}$ est la demande de bande passante des UE au point NE du sous-jeu l'Étape-I AG. En conséquence, l'objectif de chaque ASP est de minimiser le paiement total à tous les BSP compte tenu de leurs prix de raccordement tout en évitant la congestion du trafic à son BS. Ce problème de minimisation des coûts lié au raccordement pour chaque ASP j peut être déclaré comme suit:

$$\min_{\boldsymbol{w}_j \geq \mathbf{0}} C_j^{\mathsf{A}}(\boldsymbol{w}_j) \tag{2.32a}$$

$$\text{s.t.} \sum_{i=1}^{I} w_{ij} r_{ij} \geq R_j^{\mathsf{A}}(\boldsymbol{s}_j^*) \tag{2.32b}$$

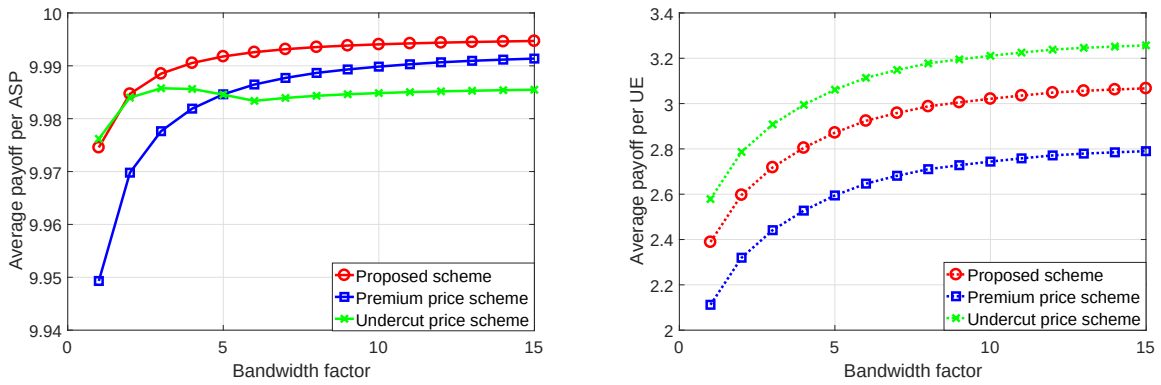
où (2.32b) est la contrainte de liaison et R_j^{A} est défini dans (2.31). This is a convex optimization problem due to the convex objective function and linear constraint (2.32b). En conséquence, nous proposons l'algorithme 2.7 pour résoudre les conditions équivalentes du problème KKT 2.32.

Lemma 2.7. *En définissant* $\nu_j \leq \min_{i\in\mathcal{I}} \left\{\frac{r_{ij}}{R_j}\right\}$, *l'algorithme 2.7 convergera.*

2.2.3.5 Résultats numériques

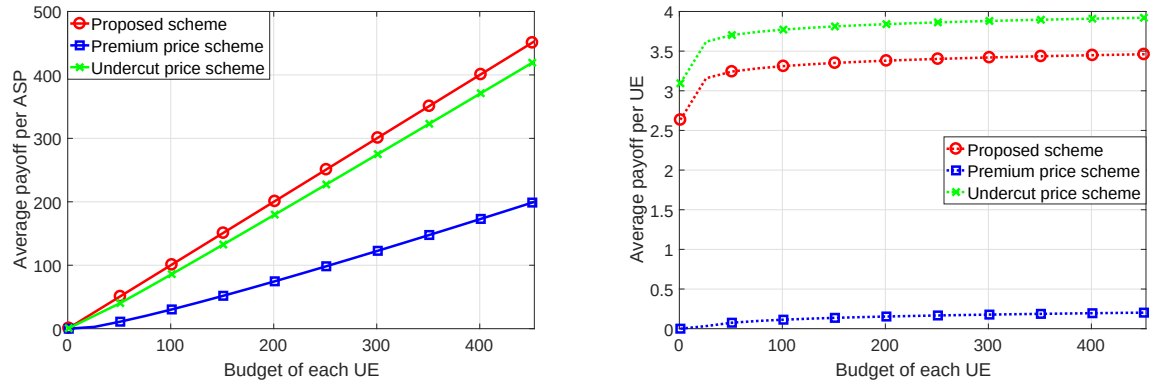
Nous évaluons les performances obtenues par le cadre théorique du jeu proposé pour un réseau sans fil avec 3 BSP ($I = 3$), 4 ASP ($J = 4$) et 20 UE ($K = 20$). Les BS des ASP et des UE sont uniformément répartis dans une zone circulaire avec un rayon de 200 mètres, tandis que les concentrateurs de liaison sans fil sont situés à des distances allant jusqu'à 400 mètres depuis l'origine. L'efficacité spectrale moyenne de chaque lien d'accès entre une BS d'un ASP et un UE (i.e., $(r_{jk})_{j\in\mathcal{J},k\in\mathcal{K}}$) de 0.35 à 12 bps/Hz, selon la distance de communication. De plus, l'efficacité spectrale moyenne de chaque liaison de liaison entre un BS d'un ASP et un WBH d'un BSP (i.e., $(r_{ij})_{i\in\mathcal{I},j\in\mathcal{J}}$) va de 1 à 5 bps/Hz.

Dans les fig. 2.3 et 2.4, nous comparons les gains moyens des ASP et des UE résultaient du cadre proposé et de deux autres schémas de référence. Dans les fig. 2.3a et 2.3b, nous montrons les gains moyens de chaque ASP et de chaque UE, respectivement en raison du cadre proposé et des schémas de base considérés car le facteur de bande passante d'accès varie. Les gains moyens de chaque APS et de chaque UE sont représentés respectivement sur les Fig. 2.4a et 2.4b pour différentes valeurs du budget de chaque UE. On peut observer que le cadre proposé permet aux ASP d'obtenir le gain moyen le plus élevé (Figs. 2.3a et 2.4a) en comparaison avec d'autres schémas de référence pour toutes les valeurs considérées du facteur de bande passante et Budget de l'UE. En raison de l'imposition de prix significativement élevés dans le cadre du système de prix premium, les ASP découragent les demandes de débit des UE (même s'ils ont de gros budgets comme le montre la figure 2.3b). Ce régime de primes se traduit donc par des gains moyens inférieurs pour les ASP et les UE par rapport à ceux dus au régime proposé, comme le montrent les Fig. 2.3 et 2.4. Dans le cadre du système de prix de sous-cotation, les UE sont en mesure de gagner un gain moyen légèrement supérieur à celui dû au cadre proposé. Le gain moyen de chaque ASP dans le cadre de ce régime ne peut cependant pas dépasser celui dû au cadre proposé. En effet, un acteur (l'aSP 4 dans ce cas) abaisse son prix en dessous du SE pour attirer plus de demandes de débit des UE tout en détériorant la force de vente d'autres ASP rivaux. L'augmentation des demandes de débit des UE gagnées par l'aSP 4, cependant, ne peut pas compenser les pertes en raison du prix peu élevé de ce lecteur. Par conséquent, les ASP, qui sont les leaders du marché, feraient mieux d'adopter le cadre proposé pour obtenir le gain moyen le plus élevé pour chacun d'eux tout en permettant un gain moyen décent pour chaque UE (le suiveur).



(a) Paiement moyen des ASPs. (b) Paiement moyen des UEs.

Figure 2.3 – Paiement moyen des ASP et des USe par rapport aux systèmes de tarification de base lorsque le facteur de bande passante d'accès varie.



(a) Paiement moyen des ASPs. (b) Paiement moyen des UEs.

Figure 2.4 – Paiement moyen des ASPs et des UEs par rapport aux systèmes de tarification de base lorsque le budget de l'UE varie.

2.3 Remarques finales

Dans cette thèse de doctorat, nous avons proposé des techniques et des algorithmes communs de gestion des ressources ainsi qu'un mécanisme d'échange de ressources basé sur la théorie des jeux pour divers environnements de réseau futurs. Plus précisément, nous avons apporté trois contributions importantes à la recherche. Premièrement, nous avons proposé une solution de gestion des ressources pour les réseaux hétérogènes à petites cellules avec allocation de bande passante et association BS compatible avec la mise en cache. Ici, nous combinons l'algorithme de recherche de ligne à faible complexité pour l'allocation de bande passante avec l'algorithme de mise en cache itérative pour le placement de contenu afin de minimiser le rapport de manque de demande de contenu des utilisateurs.

Deuxièmement, nous avons conçu une structure commune d'allocation des ressources et de mise en cache de contenu pour VWN, grâce à laquelle nous minimisons le taux de rejet de demande de contenu maximal enregistré par les utilisateurs de différents MVNO dans différentes BS dans un scénario de backhaul très encombré. Nous avons en outre proposé un algorithme basé sur la recherche de bissection efficace pour optimiser l'allocation des ressources et le placement de contenu à chaque BS, qui surpasse les autres algorithmes de référence.

Enfin, nous avons proposé un cadre basé sur le jeu MLMF Stackelberg pour la tarification et l'allocation des ressources qui capture les interactions entre les SP ainsi qu'entre les SP et les UE. Nous avons montré que le cadre basé sur le jeu proposé atteint un équilibre de jeu unique. Le, nous avons développé un algorithme distribué basé sur la mise à jour des fonctions sous-jacentes de meilleure réponse, qui s'est avéré converger vers l'équilibre du jeu. La recherche dans notre étude doctorale a donné lieu à deux publications de revues [68, 69], un article de revue technique en cours d'examen de deuxième tour [70], ainsi que six articles sur des conférences prestigieuses [42, 71–75].

Chapter 3

Introduction

The fifth-generation (5G) wireless cellular system, which has started rolling out in 2020, is expected to provide a huge network performance improvement and to support new services and applications, compared to those enabled by the current fourth-generation (4G) system [2–4]. Specifically, a 1000-fold increase of network throughput compared to that of 4G systems is the target spec promised by the 5G systems [5]. This significant network capacity increment is essential for coping with the ever-increasing mobile traffic generated from enhanced mobile broadband (eMBB) services such as mobile video streaming [6–9]. In addition to support the eMBB service type, the future 5G wireless cellular system also supports the other two key service types, namely ultra-reliable low-latency communications (uRLLC) and massive machine type communications (mMTC) for serving mission-critical applications and a massive number of simultaneous connections from wireless devices [8, 10, 11]. Fig. 3.1 illustrates key application domains factorized to the three main service types. Accordingly, a hefty burden of network traffic as well as stringent requirements are put on both the radio access network (RAN) and the backhaul network, which establish end-to-end connections between user equipments (UEs) and core network (CN) via base stations (BSs). New techniques and novel network architectures must be devised and well incorporated together to enable the 5G wireless cellular system to fulfill such stringent and versatile requirements [3, 4, 7, 8].

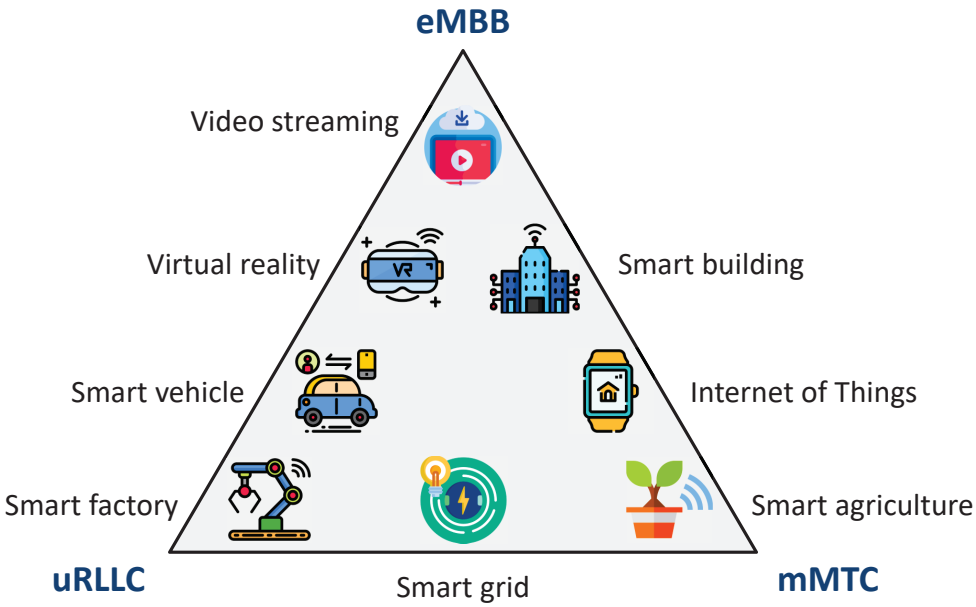


Figure 3.1 – Applications domains projected to the three main service types.

3.1 Overview on The Emerging 5G Wireless Cellular Networks

3.1.1 Advanced Resource Utilization and Management

Exploiting new radio spectrum bands and enhancing spectrum efficiency are two necessary and complementary approaches for network throughput and network quality improvement. Higher frequency bands has been recently authorized for use in 5G wireless cellular systems, thereby enabling multi-gigabit transmission through wide spectrum utilization. These spectrum bands range from sub-3 GHz (low band) [9], mid-band (i.e., 3.5 GHz) [76], to millimeter wave bands (mmWave) (i.e., 26/28 GHz and above 30 GHz) [12, 77].

Harnessing other kind of resources rather than conventional ones such as radio spectrum bandwidth and transmission power, is a promising approach to reduce communication delay and relieve traffic congestion. And storage repository is an important network resource to be leveraged in this sense [3]. In fact, by deploying storage devices at BSs in the network and pre-fetching popular content/files, which is also referred as *content caching*, [1] to these storage repositories, one can bring popular contents in closer proximity to UEs. As a result, the traffic in the backhaul links induced by accessing these contents, which are usually stored in the CN if they are not cached at the BSs,

[1] We use the term content and file interchangeably in this doctoral dissertation.

is also relieved significantly [13]. By doing so, the access delay to these contents is reduced, thus improving users' quality of service (QoS) [14–16].

Innovations in enhancing the spectrum efficiency and leveraging emerging resource dimensions, typically the content caching, are most beneficial if they are engineered jointly with other resource management frameworks [3, 15, 16]. Yet designing advanced resource management frameworks that can utilize the advantages of both the spectrum efficiency enhancement and content caching is challenging and requires much more further research.

3.1.2 Network Slicing - Wireless Network Virtualization

An illustration of the 5G wireless cellular network is given in Fig. 3.2, where the hyper-connected world with massive wireless connections requiring different QoS is envisioned. As discussed in the previous subsections, various technologies have been proposed for supporting the three main service types (eMBB, uRLLC, and mMTC) and their combinations. To help the 5G network meet stringent requirements of such diverse service types, it is crucial to design innovative network architectures that not only integrate those technologies but also make them work together in a seamless way. Ultra dense networks (UDN) [17, 78], evolved heterogeneous networks (HetNets) with dense deployment of small cells, and cloud radio access network (C-RAN) [6, 19] are the two candidates for future 5G network architecture. The HetNets can fundamentally enhance network capacity, energy efficiency, and coverage performance [17, 18], while C-RAN enables computationally communications schemes such as multi radio access technology (multi-RAT) [79, 80] and massive multi input multi output (MIMO) [81].

However, deploying and operating these novel network architectures for 5G systems as well as integrating innovative technologies into these systems require a massive overhaul in network infrastructure, both in the air interface and in the backhaul network. Such requirements can incur a surcharge of capital expenditure (CAPEX) and operation costs (OPEX), as well as slowing down the deployment time of new technologies and network services [7, 8]. Wireless network virtualization (WNV), also known as *network slicing*, has been considered as a promising networking paradigm for addressing this problem [20].

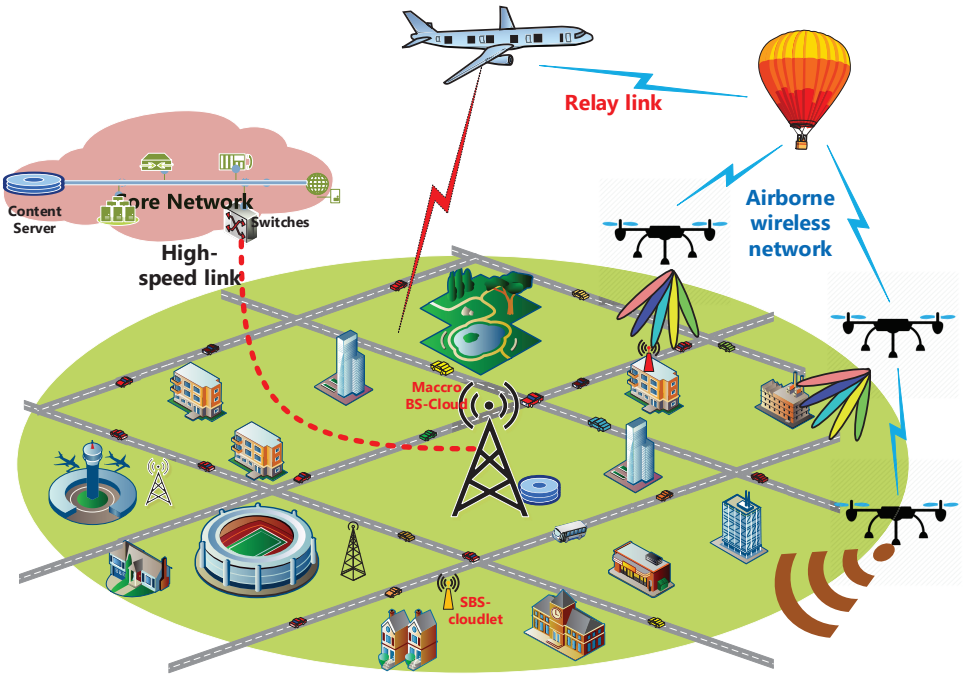


Figure 3.2 – Vision on 5G Wireless Cellular System.

In fact, wireless virtualization allows multiple mobile virtual network operators (MVNOs), also known as service providers (SPs), to share the same network infrastructure and a common resource pool owned and managed by one (or several) infrastructure provider(s) (InP). Here, the InP must be able to flexibly and efficiently allocate network resources such as transmission power and radio bandwidth to MVNOs based on their own requirements and mutual contracts so that their operations and services can be harmonized on the same infrastructure [20]. Each MVNO in turn uses the rented resources and infrastructure to provide its own services (e.g., eMBB, uRLLC and mMTC [21]) to its clients including UE or other MVNOs with committed QoS. Accordingly, network slicing helps network operators and SPs reduce CAPEX and OPEX by utilizing network resource in a flexible and efficient manner while better meeting the required QoS [20].

Thanks to scalable and flexible characteristics, network slicing also expedites technology implementation and integration into 5G wireless cellular networks [20]. For instant, content caching and mobile edge computing (MEC) can be implemented rapidly on virtualized/softwarized platforms using network slicing [67]. Fig. 3.3 illustrates a network slicing concept, where typical network slice instances, sharing a common network infrastructure and resources, are created for various application domains such as eMBB, voice, social networking, and uRLLC based applications.

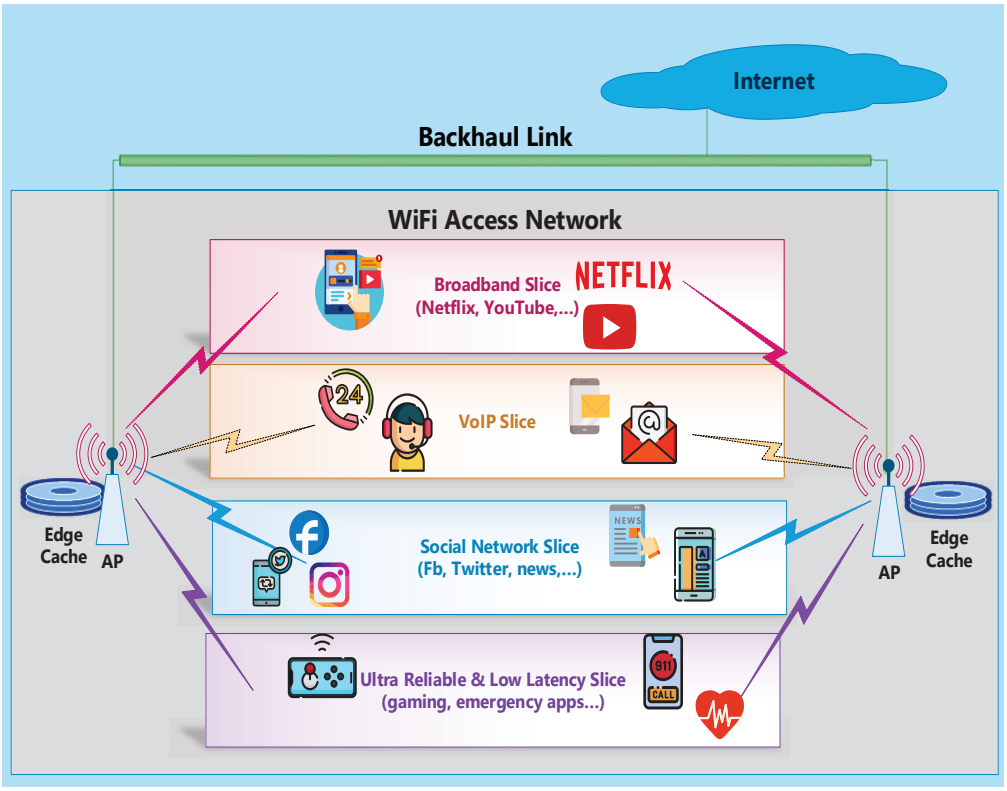


Figure 3.3 – An illustration of typical network slice for various application domains.

3.1.3 Economic Aspects of Resource Sharing Among 5G Network Tenants

Network slicing is an important technology for which the monolithic network can be virtually sliced into multiple network slices to support specialized wireless services. Appropriately designed network slices, for instance, could be designated for the high-speed streaming services such as YouTube and Netflix, or the uRLLC services for the factory control applications [22]. Network slicing also provides a paradigm shift toward multi-tenancy in the next-generation wireless network [23] where individual tenants (e.g., mobile virtual network operators (MVNOs)) own and manage corresponding network slices. By enabling service trading among tenants, this paradigm shift offers greater business opportunities and greater savings in Capital Expenditure (CAPEX) and Operational Expenditure (OPEX) [22].

Accordingly, there exists multilateral interactions between SPs regarding the economics aspect. Here, the interaction can be an economic competition between the SPs providing same service type to a market, or it can be a buy-and-sell interaction between the SPs and their customers such as UEs. These multilateral interactions among the SPs and their customers constitute to a resource trading

market. Designing an appropriate framework for operating such market is crucial for achieving efficient network serviceability and high profits. Game theory is an appropriate tool for modeling such market [24].

3.2 Research Challenges and Motivations

3.2.1 Joint Resource Allocation and Content Caching in Multi-cell HetNets

Recently, there are some existing works that study the caching problem for small-cell networks [15, 16] where they investigate the joint caching, routing, and channel assignment. However, most of these works do not consider the stochastic behavior of content request and service processes. Meanwhile, authors in [25, 26] study the joint caching and resource allocation design based on the time-varying signal-to-noise ratio (SNR). This design would require frequent cache updates, which is not cost efficient because the SNR usually varies quickly over time.

Accordingly, a general caching design for HetNets where mobile users can be associated with either a small-cell base station (SBS) or macro-cell base station (MBS) and allocated radio resource to download their desired contents should be considered. BS associations in such the heterogeneous network should take into caching decisions (i.e., being caching-aware) where users should associate with BSs which have favorable channel conditions and store their requested contents.

3.2.2 Joint Resource Allocation and Content Caching in Virtualized Wireless Networks

Different content caching frameworks have been introduced leveraging the evolution of network architecture, especially the HetNets and C-RAN based network architectures. Specifically, prefetching contents at different network levels (e.g., macrocell, smallcell, and femtocell) is a potential content caching approach that leverages the hierarchical network topology of the HetNets [29]. Another variant of such a hierarchical caching strategy is also adopted in C-RAN based network architecture, where the storage repository is not only installed at the remote radio heads (RRHs) but also in the cloud [31].

However, most of these existing works do not consider the highly congested network scenario due to the limited radio resource [32]. It is because the uRLLC and mMTC service types require a large number of reliable connections [82] while the eMBB service type consumes a great deal of network capacity. Moreover, a great deal of control signaling data, which would consume valuable radio resources in the wireless access links and result in further network congestion. The problem of lacking radio resource is more severe in wireless backhaul networks (WBN) [83–85] where wireless backhaul links are used in lieu of the traditional cable links for connection with the CN.

In the virtualized wireless environment where multiple MVNOs operate on the shared infrastructure with limited storage capacity, content caching for network performance improvement could be less significant since the InP likely partitions the available storage capacity to MVNOs. Therefore, efficient and shareable content caching among MVNOs and optimization of radio resource allocation can effectively boost the network performance [2].

3.2.3 Economic-Aware Resource Allocation in Multi-Tenant Network Slicing

As of discussion in Sec. 3.1.2, wireless virtualization is a new network paradigm where the network infrastructure and network services offered to wireless users are separated. By doing so, network slicing enables a paradigm shift toward multi-tenancy in the future wireless network [23] where individual tenants (e.g., MVNOs, and service providers (SPs)) own and manage corresponding network slices. Here, each tenant buys/sells resources and/or services from/to other tenants. For instance, an access SP (ASP) who provides radio access service to its wireless users must buy a portion of the network infrastructure (e.g, BSs) from the InP as well as the radio spectrum and backhaul links from other SPs. Furthermore, this ASP also competes against other ASPs to attract wireless users from purchasing its service. The multilateral interactions among SPs and their customers such as wireless users and other SPs, constitute a resource and service trading market. Designing an appropriate framework for operating such a market is crucial for achieving efficient network serviceability and high profits.

[2] We use the terms file and content ex changeably in this dissertation.

3.3 Literature Review

3.3.1 Joint Resource Allocation and Content Caching in HetNets and VWNs

The architecture of HetNets with dense deployment of SBSs in coexistence with the MBSs provides an important solution to better serve the ever-growing number of connected devices and rapidly increasing mobile traffic. In addition, there is increasing congestion of backhaul links, which connect the MBSs and SBSs with the CN. To mitigate this problem, the cache-enabled heterogeneous small-cell network, which caches popular contents at the network edge including MBSs and SBSs, was proposed [86]. There are some existing works that study the caching problem for small-cell networks [15, 16] where they investigated the joint caching, routing, and channel assignment. Both SBSs and MBSs in such cache-enabled systems typically cache popular contents so that requests of these highly demanded files can be served efficiently without using the backhaul links. However, all these works do not consider the stochastic behavior of content request and service processes.

Moreover, most existing works in the literature treat the WNV, content caching, resource allocation design issues separately. In particular, Poularakis et al. [13] focused on improving the caching performance, i.e., increasing the hit rate and reducing access delay, for small-cell wireless networks. The authors in [15, 16] only considered joint content caching with conventional resource allocation in wireless networks without WNV. There are a few works such as [25, 26] studying the joint caching, resource allocation, and WNV. However, adaptation of cache placement decisions based on the SNR may not be cost efficient. In fact, caching decisions at BS should be made over a long time scale while the SNR typically varies rapidly.

Different content caching frameworks have been recently introduced to leverage the evolution of network architecture. In [29] and its related work, the authors proposed to install storage repository at femtocells, which are deployed in high density and closer to UEs, to assist the macro BS through offloading content requests. Another approach called hierarchical caching is to leverage the hierarchical structure of modern network topology and coding theory for content caching as in [30]. To adapt to C-RAN based network architecture, Tang et al. in [31] proposed to install the storage repository not only at the RRHs but also in the cloud, which can be considered as another version of hierarchical caching.

Wireless networks with wireless backhaul may suffer from performance degradation if the radio resources allocated for backhaul links are not sufficient and/or heavy contents such as large video files are transferred over these backhaul links from the CN. Another innovative content caching approach which enables to significantly reduce the backhaul traffic is to leverage device caching and device-to-device (D2D) communications [87], which caches content on mobile device's storage. This approach unfortunately is hindered by the mobility nature of mobile devices and their economical selfishness in providing content caching, which could consume their limited battery and storage capacity.

3.3.2 Game Theoretic Based Resource Trading in Virtualized Wireless Networks

The Stackelberg game theory has been employed to study various resource allocation problems in different research fields [44, 45]. This game theory has recently been applied to solve different resource allocation problems in wireless communications. For instance, this game theoretic approach was leveraged to tackle spectrum sharing problems in cognitive radio networks [38, 41], resource management in full-duplex wireless networks [39], and mobile offloading market [43].

Moreover, the Stackelberg game theory has also been applied to study interactions among stakeholders of a network slicing based wireless network. Leveraging this game theory, a price-aware joint power and radio resource allocation framework was proposed for a virtualized wireless network with one InP and multiple MVNOs [42]. Meanwhile, the authors of [88] applied the Stackelberg game theory to address the joint spectrum reuse-aware resource allocation and content caching optimization problem for two different network slice instances. A tri-level Stackelberg game framework was proposed in [40] for resource trading in a virtualized wireless network. Here, each UE can be associated with only one service provider, thus resulting in multiple single-leader Stackelberg games in levels two and three of the underlying game.

A common characteristic of the above Stackelberg game based frameworks is that they assume a single operator such as the InP acts as a sole leader of the game. However, this assumption does not hold true when resource and service trading in a virtualized wireless network involves multiple SPs in the service domain. For example, various SPs such as Bell, Rogers, and Telus in Canada can provide a similar mobile service for wireless users in a certain area of the country.

In fact, SPs who control network slices providing the same service type have to compete with one another because potential customers could choose the most appropriate SPs to receive services. The interactions among these SPs, thus, form an oligopoly market in the service and resource trading. Accordingly, the multi-leader-multi-follower (MLMF) Stackelberg game theory provides a suitable tool for modeling the interactions among these stakeholders.

The MLMF Stackelberg game theory has been employed in some recent works [46–50]. Particularly, the authors of [46] and [47], which were inspired by the work [41], employed the MLMF Stackelberg game to study the spectrum sharing problem in cognitive radio networks. Several multi-leader Stackelberg game frameworks were proposed to model the traffic offloading market in LTE unlicensed bands [48, 49]. Meanwhile, the MLMF Stackelberg game [50] was applied to study the resource trading among multiples InPs, MVNOs, and their wireless users under the network slicing paradigm.

Various game theoretic approaches have been applied to tackle different resource allocation problems in the network slicing context. In particular, the authors of [34] proposed a game theoretic framework for network slicing, built upon the share-constrained proportional resource allocation mechanism, that achieves high efficiency and fairness in allocating resource to network tenants. This game framework was also applied to allocate radio remote heads (RRHs) to MVNOs in [89]. A matching game was employed to tackle the combinatorial resource trading and service selection problem for InPs, MVNOs, and UEs in [35]. Meanwhile, various bidding mechanisms [36, 90, 91] were leveraged to study the resource trading among stakeholders in network slicing. The authors of [36] and [90] applied traditional combinatorial bidding and a generalized Kelly bidding mechanisms for SPs/MVNOs to bid different network resources so as to maximize their utilities, respectively. A new bidding mechanism was recently proposed for joint computation and storage resource trading in network slicing [91]. Contract theory was also applied to model the resource trading between an MVNO and multiple InPs [37].

To the best of our knowledge, studying the multilateral interactions among peer access service providers (ASPs) that provide the same kind of service to a group of UEs in the virtualized wireless network has not been considered. Moreover, the service models in the works discussed above is limited by assuming the single-source service selection, i.e., a UE can only select one SP for purchasing service. With the multiple slice connectivity feature enabled in network slicing, however,

any wireless user and SP are able to lease services from different SPs at the same time [67], thus enabling multi-source service selection. Accordingly, it is more general (but challenging) to study the multilateral interactions among the stakeholders in such network slicing context.

3.4 Research Contributions and Organization of the Dissertation

Taking into account these issues in resource allocation and content caching design for 5G and beyond wireless networks, this Ph.D. research focuses on three main objectives. We first develop a joint radio resource allocation and content caching framework under small-cell HetNets setting, where we consider the resource allocation and content caching problem for a single network operator with its own resource pool. Second, we study the joint resource allocation and content caching in the virtualized multi-cell network environment where multiple network operators sharing a common resource pool of wireless channels and storage repositories under the coordination of a centralized controller. Third, we consider resource allocation problem concerning the multilateral interactions between SPs as well as between the SPs and their customers in a network slicing setting by using the Stackelberg game theory. All of the objectives aim to directly address important technical issues of future network scenario (HetNets) and emerging network paradigms (network slicing). The main contributions of this Ph.D. dissertation are as follows:

1. *Caching for Heterogeneous Small-Cell Networks with Bandwidth Allocation and Caching-Aware BS Association [69]:* We study the caching problem for heterogeneous small-cell networks with bandwidth allocation and caching-aware BS association. The caching control and bandwidth allocation problem aims at minimizing the request miss ratio. To solve this problem, we propose a Line-Search-based-Iterative (LSBI) algorithm which determines the solution by combining the line-search algorithm to obtain the optimal bandwidth allocation with the iterative caching algorithm to acquire a caching solution. Numerical results demonstrate that the LSBI algorithm significantly outperforms existing caching algorithms, and is on a par with the performance bound.

2. *Joint Resource Allocation and Content Caching for VWN [68]:* We study the joint resource allocation and content caching problem which aims to efficiently utilize the radio and content storage resources in the highly congested backhaul scenario of VWN. In this design, we min-

imize the maximum content request rejection rate experienced by users of different MVNOs in different cells, which results in a mixed-integer non-linear program (MINLP). We solve this difficult optimization problem by proposing a bisection-search based algorithm that iteratively optimizes the resource allocation and content caching placement. We further propose a low-complexity heuristic algorithm which achieves moderate performance loss compared to the bisection-search based algorithm. Extensive numerical results confirm the efficacy of our proposed framework which significantly reduces the maximum request outage probability compared to other benchmark algorithms.

3. *Resource Allocation for Multi-Tenant Network Slicing Using the Multi-Leader Multi-Follower Stackelberg Game Approach [70]:* We study the resource allocation and pricing problem for network slicing that captures interactions among access/backhaul service providers and their UEs by using the MLMF Stackelberg game approach. Toward this end, we show how to formulate such a Stackelberg game and prove the existence of a unique game equilibrium. Then, we develop a distributed algorithm based on updating underlying best-response functions, which is proved to converge to the game equilibrium. Numerical results are presented to provide important insights into the interactions among the involved stakeholders and demonstrate the economical efficacy of the proposed design with respect to existing benchmarks.

The remaining of this dissertation is organized as follows. Chapter 4 reviews some necessary background on wireless edge content caching techniques. The concept of wireless virtualization and the Stackelberg game theory are also briefly presented in this chapter. We discuss the caching problem for heterogeneous small-cell networks with bandwidth allocation and caching-aware BS association in Chapter 5. In Chapter 6, we present the joint resource allocation and content caching for virtualized content-centric network. In Chapter 7, we study the resource allocation and pricing problem for network slicing that captures interactions among access/backhaul service providers and their UEs by using the MLMF Stackelberg game approach. Finally, chapter 8 summarizes the main contributions of the dissertation and makes some recommendations for future research directions.

Chapter 4

Background and Fundamental

This chapter provides the necessary concepts and background on content caching and network slicing. Then, some fundamentals of Stackelberg game are briefly presented.

4.1 Wireless Edge Caching

The notion of content caching in wireless cellular networks refers to the prefetching of popular contents at storage repositories deployed at network edges such as BSs and RRHs. Content caching brings multiple advantages such as improvement of the content access latency, cache hit rate, and network congestion mitigation [3, 13–16]. First, as certain contents are placed closer to the network users, the accessing time to these contents can be significantly reduced. Second, if a popular content is cached at the network edge, the traffic incurred in the backhaul network to transmit this content from the CN to wireless users over the access network is diminished. Also, by prefetching popular contents to the storage repositories at the network edge during the off-peak hours, the network operator is able to avoid traffic overload during on-peak hours.

The popularity level of a content represents how often the underlying content is requested by UEs. Content popularity can be expressed as the number of times this content is accessed/requested by UEs during a time interval [92]. Given a list of contents, their popularity usually follows a power-law model such as Zipf distribution [93]. Here, the probability of requesting the file having rank

$f \in \{1, ..., F\}$ is given by

$$Z(f, \gamma) = \frac{f^{-\gamma}}{\sum_{n=1}^{F} n^{-\gamma}}, \tag{4.1}$$

where F is the total number of files in the list and γ is the Zipf parameter.

4.2 Wireless Virtualization

4.2.1 Basic Concepts of Wireless Virtualization

The virtualization concept, in which a virtual form of a physical entity is created through softwarization and process, was proposed since the debut of IBM CP-40 operating system (OS) in the 1960s [94]. As such, a virtual system could span across computing platforms, network resource, and storage devices [95]. Virtualization has been widely adopted for data centers, and later for networking, for securely connecting remote sites with the centralized controller through Internet in the 1980s [67].

The early form of network slicing appeared with the introduction of overlay networks, allowing a virtual network to be created over a physical network infrastructure, through nodes connected via logical links. The first-generation platforms for verifying and evaluating new network protocols were established based on overlay networks by 2000 [67]. Here, a slice was defined as a unit component with allocated resources such as computing power, storage on servers or resources existing in namespaces. The development of network virtualization was pushed forward with the introduction of software-defined networking (SDN), which enabled programmable slices via OpenFlow interface [96].

In the 5G wireless cellular system, the wireless virtualization or network slicing concept is introduced by the Next Generation Mobile Network (NGMN) [97], in which multiple logical self-contained networks can be created on top of a common physical infrastructure platform. By integrating physical and/or logical network, cloud and other kinds of resource (e.g., storage repository) into a programmable, open software-oriented multi-tenant network environment, network slicing establishes an ecosystem with different stakeholders the enables flexible and scalable deployment of novel business and technical solutions. Similar definitions of network slicing in 5G are also de-

fined in 3GPP Release 14 [98] and by International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) [99].

The network slicing process is composed of three main layers, namely the service instance layer, the network slice instance layer, and the resource layer [100]. Network slicing builds on top of the following seven main principles that shape the concept and related operations: automation, isolation, customization, elasticity, programmability, end-to-end, and hierarchical abstraction [67].

In a network slicing environment, a vertical segment or SP (e.g., application, mobile service, etc.) provides its service through one or multiple service instances, which is/are sold or allocated to clients. A set of resources is customized for each service slice, depending on the demand of that slice. This set of resources may contain none, one or several different shared or isolated sub-network instances. A sub-network instance can be a network function, a sub-set of network functions, or a set of underlying network resources. Depending on the policies and configuration arrangements, the resource allocated to each sub-network can be used in a totally isolated, disjunctive, or shared manner among different network slice instances, typically for the ones providing the same service type. Dynamic control and automation of network slice instances are realized to support dynamic service demands, thanks to open programmable interfaces and common abstractions of resources [67].

4.2.2 Enabling Technologies for Wireless Virtualization

Virtualization technologies are key enabler for the wireless virtualization. An overview on such technologies is presented in the following.

- *Hypervisor:* The hypervisor is an additional layer, placed between the physical infrastructure in the bottom and the operating layer running on the top. This middle layer is responsible for controlling and managing virtual machines, which are virtual platforms for guest OSs to execute their applications and services. Depending on how a hypervisor is implemented, it can be categorized as firmware-based, software-based, and OS level-based hypervisor. The hypervisor is also in charge of allocating resources between virtual machines and/or network slice instances [67].

- *Virtual Machine:* The creation of virtual machine (VM) provides the effect of a physical resource that runs on its own OS [67]. The actual hardware virtualization takes place on the host machine, while the guest machine is the VM. Current cloud platforms are capable of hosting multiple VMs and executing different applications in the parallel manner. Each VM shares resources such as computing, storage, memory and network, while its operation is completely isolated from that of the host and fellow guest VMs.

- *Software Defined Networking (SDN):* The idea of SDN is to decouple the control plane from the data plane (network packet forwarding process) of the traditional centralized networks and putting the network controller in a centralized intelligence. SDN is an essential network architecture approach for enabling network slicing, thanks to its key characteristics such as flexibility, service-oriented adaptation, scalability, and robustness [101]. Furthermore, SDN enables multi-tenancy through abstracting its network resource as resource groups, supporting control plane logic [23, 102]. Such resource groups, which form a functioning network slice, can be allocated to different tenants. Each tenant has capabilities to program the underlying data layer by using the data control plane function in addition with the assigned policy.

- *Network Function Virtualization (NFV):* The fundamental idea of network function virtualization (NFV) is to implement original hardware-based proprietary network functions in software, i.e., virtual network functions (VNFs). By executing VNFs in virtual environments such as cloud computing facilities, NFV enables faster time-to-market network solution, thus improving cost efficiency [103]. As VNFs are deployed on collocated and/or distributed VMs, one can flexibly chain them together, i.e., service chaining, for obtaining desired services [104]. In other words, a network slice providing a certain service can be realized through the chaining of proprietary VNFs in the virtual environment.

- *Software Defined Radio (SDR):* Relying on the notion of NFV, SDR is meant to implement components (e.g., filters, mixers, amplifiers, etc.) of a radio communication systems (or a RAN) in software installed on computing devices such as personal computer or cloud servers [105, 106]. Each piece of such software-implemented radio component can be referred as a VNF. By doing so, one can implement a new RAN by flexibly chaining necessary SDR functions running on centralized and/or distributed computing facilities together, thereby expediting the service deployment for different network operators, especially for MVNOs in the wireless virtualization environment.

4.3 Stackelberg Game Theory

To realize the network slicing, the monolithic network can be sliced into multiple network slice instances to support specialized network services, such as wireless services. Network slicing also provides the paradigm shift toward multi-tenancy in the next-generation wireless network [23] where individual tenants own and manage corresponding network slices. By enabling service trading among tenants, such as as MVNOs and/or SPs, this paradigm shift offers greater business opportunities and greater savings in CAPEX and OPEX [22]. The multilateral interactions among SPs and their customers such as UEs, constitute to a service trading market. Designing an appropriate framework for operating such a market is crucial for achieving efficient network serviceability and yielding high profits for stakeholders. Game theory [24] is a promising approach for modeling such a framework. The Stackelberg game theory is typically one of the appropriate game theories that can be applied for service and resource trading in network slicing environment, due to the existence of both hierarchical and peer interactions between SPs and their customers.

4.3.0.1 Definitions of Stackelberg Game

Stackelberg game is a special class of *non-cooperative* game theory, an important branch of game theory. Here, all the players in the game compete with each other to reach their outcome of a decision process, i.e., the interest of each individual is totally or partially in conflict with other players [24]. In a Stackelberg game, moreover, there exists a hierarchy among the players in which some of the players are in position allowing them to enforce their own strategies upon other players. The players in such a hierarchical decision-making process can be categorized as the *leaders* and *followers*. Specifically, leaders are the players in the upper-level of the hierarchy who declare decisions first and impose their strategies upon other players. Followers are the players in the lower-level of the hierarchy and react to the leaders' declared strategies. A Stackelberg game usually consists of numerous followers who interact with one or more leaders. Accordingly, a Stackelberg game is categorized as a single-leader-multi-follower (SLMF) if there is only one leader, or as a multi-leader-multi-follower (MLMF) if the number of leaders is more than one [1, 24]. Fig. 4.1 shows the structure of the SLMF and MLMF Stackelberg games.

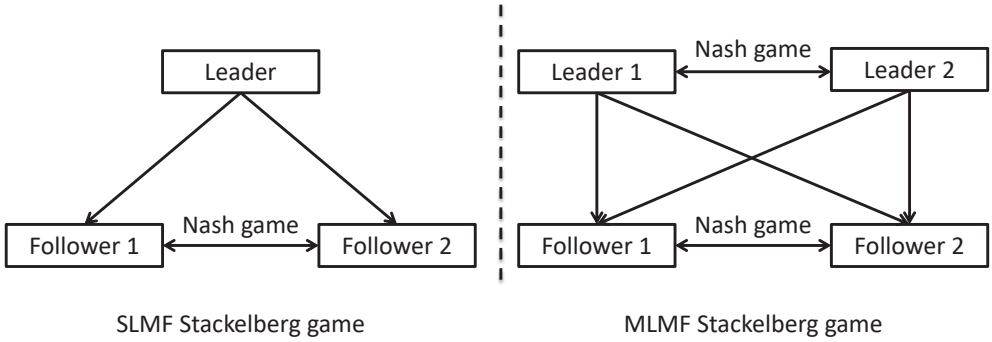


Figure 4.1 – An illustration structure of SLMF and MLMF Stackelberg games [1].

As a variant of non-cooperative game, there exists the conflict of interest or competition between the peer players in the same hierarchical level of a certain Stackelberg game, e.g., between leaders and between the followers. Finding the game equilibrium point for the same hierarchical level players, or the so-called Nash equilibrium (NE), is a crucial objective. Besides that, there is also a game equilibrium point between players in different hierarchies, i.e, between the leaders and followers, which is called Stackelberg equilibrium (SE) [24]. In what follows, we give the formal definition of NE and SE in a Stackelberg game.

4.3.1 Single Leader Multiple Follower (SLMF) Stackelberg Game

We first introduce several notations before giving the definitions of NE and SE. In a SLMF Stackelberg game, only one player plays the leader role whereas the remaining players are the followers. Denote by $\mathcal{F} = \{1, ..., f, ..., F\}$ the set of followers in this game, where f is the index of the followers. The leader is denoted by L. Moreover, $s^{\mathsf{F}}_{f,f\in\mathcal{F}}$ represents the strategy of the follower f, and $\boldsymbol{s}^{\mathsf{F}} = (s^{\mathsf{F}}_f)_{f\in\mathcal{F}}$ is the strategy vector of all the followers. A strategy of a player is the choice/action made by this player that affects the outcome of the game. Furthermore, $\boldsymbol{s}_{-f}$ is the strategy vector of all the followers except the follower f and s^{L} is the strategy of the leader.

The utility (or payoff) function of each player in the game, taking the strategy of this player's strategy and those of other players as its input, is used to evaluate the outcome of the decision making process. Here, $U^{\mathsf{L}}(s^{\mathsf{L}}, \boldsymbol{s}^{\mathsf{F}}_f)$ denotes the utility function of the leader, given the strategies $\boldsymbol{s}^{\mathsf{F}}_f$ of the followers. $U^{\mathsf{F}}_f(s^{\mathsf{L}}, s^{\mathsf{F}}_f, \boldsymbol{s}^{\mathsf{F}}_{-f})$ denotes the utility function of the follower f, given the strategies of the leader and the other followers.

Definition 4.1. *Given the leader's strategy, the NE between the followers is the point at which no follower has incentives to unilaterally change its strategy for achieving a better utility without worsening the utilities of other followers. Mathematically, given* s^{L}, $\boldsymbol{s}^{\mathsf{F}*}$ *is the NE between the followers if the following condition is satisfied:*

$$U_f^{\mathsf{F}}(s^{\mathsf{L}}, s_f^{\mathsf{F}*}, \boldsymbol{s}_{-f}^{\mathsf{F}*}) \geq U_f^{\mathsf{F}}(s^{\mathsf{L}}, s_f^{\mathsf{F}}, \boldsymbol{s}_{-f}^{\mathsf{F}*}), \forall f \in \mathcal{F}. \tag{4.2}$$

Definition 4.2. *The SE of a Stackelberg game is a NE between the leader and the followers. Specifically,* $(s^{\mathsf{L}*}, \boldsymbol{s}^{\mathsf{F}*})$ *is the SE point, if it satisfies the following conditions. For the followers,*

$$U_f^{\mathsf{F}}(s^{\mathsf{L}*}, s_f^{\mathsf{F}*}, \boldsymbol{s}_{-f}^{\mathsf{F}*}) \geq U_f^{\mathsf{F}}(s^{\mathsf{L}*}, s_f^{\mathsf{F}}, \boldsymbol{s}_{-f}^{\mathsf{F}*}), \forall f \in \mathcal{F}, \tag{4.3}$$

where $s^{\mathsf{F}*}$ *is the NE between the followers, given the leader strategy* $s^{\mathsf{L}*}$*. For the leader,*

$$U_f^{\mathsf{F}}(s^{\mathsf{L}*}, \boldsymbol{s}^{\mathsf{F}*}) \geq U_f^{\mathsf{F}}(s^{\mathsf{L}}, \boldsymbol{s}^{\mathsf{F}\dagger}), \tag{4.4}$$

where $\boldsymbol{s}^{\mathsf{F}\dagger}$ *is the NE between the followers, given the leader strategy* s^{L}*.*

4.3.2 Multiple Leader Multiple Follower (MLMF) Stackelberg Game

This is a more general form of a Stackelberg game, where multiple leaders and multiple followers play or participate in the game. The SE in this Stackelberg game consists of the NE between the followers and the NE between the leaders. Let $\mathcal{L} = \{1, ..., \ell, ..., L\}$ be the set of leaders in the game. Moreover, s_ℓ^{L} is the strategy of the leader ℓ, and $\boldsymbol{s}^{\mathsf{L}} = (s_\ell^{\mathsf{L}})_{\ell \in \mathcal{L}}$ is the strategy vector of all leaders. $\boldsymbol{s}_{-\ell}^{\mathsf{L}}$ is the strategy vector of all the leaders except the leader ℓ. $U_\ell^{\mathsf{L}}(s_\ell^{\mathsf{L}}, \boldsymbol{s}_{-\ell}^{\mathsf{L}}, \boldsymbol{s}^{\mathsf{F}})$ is the utility of the leader ℓ, given its own strategy s_ℓ^{L}, the other leaders' strategies $\boldsymbol{s}_{-\ell}^{\mathsf{L}}$, and the followers' strategies $\boldsymbol{s}^{\mathsf{F}}$. Now, the SE of the MLMF is defined as follows:

Definition 4.3. $(\boldsymbol{s}^{\mathsf{L}*}, \boldsymbol{s}^{\mathsf{F}*})$ *is the SE of the MLMF Stackelberg game, if it satisfies the following conditions. For the followers,*

$$U_f^{\mathsf{F}}(\boldsymbol{s}^{\mathsf{L}*}, s_f^{\mathsf{F}*}, \boldsymbol{s}_{-f}^{\mathsf{F}*}) \geq U_f^{\mathsf{F}}(\boldsymbol{s}^{\mathsf{L}*}, s_f^{\mathsf{F}}, \boldsymbol{s}_{-f}^{\mathsf{F}*}), \forall f \in \mathcal{F}. \tag{4.5}$$

For the leaders,

$$U_f^{\mathsf{F}}(s_\ell^{\mathsf{L}*}, \boldsymbol{s}_{-\ell}^{\mathsf{L}*}, \boldsymbol{s}^{\mathsf{F}*}) \geq U_f^{\mathsf{F}}(s^{\mathsf{L}}, \boldsymbol{s}_{-\ell}^{\mathsf{L}*}, \boldsymbol{s}^{\mathsf{F}\dagger}), \forall \ell \in \mathcal{L}, \tag{4.6}$$

where $\boldsymbol{s}^{\mathsf{F}\dagger}$ *is the NE between the followers, given the leaders' strategies* $(s^{\mathsf{L}}, \boldsymbol{s}_{-\ell}^{\mathsf{L}*})$.

4.3.3 Finding the SE of a Stackelberg Game

From the definitions of SE above, we can see that the objective of each player in a Stackelberg game is to maximize its own utility function without depreciating the other players' utilities. Specifically, given the strategy $\boldsymbol{s}^{\mathsf{L}}$ of the leaders, each follower f, for all $f \in \mathcal{F}$, needs to solve the following optimization problem:

$$\max_{s_f^{\mathsf{L}}} U_f^{\mathsf{F}}(\boldsymbol{s}^{\mathsf{L}}, s_f^{\mathsf{F}}, \boldsymbol{s}_{-f}^{\mathsf{F}*}) \tag{4.7a}$$

$$\text{subject to } g_f^{\mathsf{F}}(\boldsymbol{s}^{\mathsf{L}}, s_f^{\mathsf{F}}, \boldsymbol{s}_{-f}^{\mathsf{F}*}) \geq 0 \tag{4.7b}$$

where (4.7b) describes the constraint of the follower f, which needs not be the same for all the followers.

Meanwhile, each leader ℓ, for all $\ell \in \mathcal{L}$, maximizes its utility through the following optimization problem:

$$\max_{s_\ell^{\mathsf{F}}} U_f^{\mathsf{F}}(s^{\mathsf{L}}, \boldsymbol{s}_{-\ell}^{\mathsf{L}*}, \boldsymbol{s}^{\mathsf{F}*}) \tag{4.8a}$$

$$\text{subject to } G_\ell(s^{\mathsf{L}}, \boldsymbol{s}_{-\ell}^{\mathsf{L}*}, \boldsymbol{s}^{\mathsf{F}*}) \geq 0 \tag{4.8b}$$

$$\text{and } \boldsymbol{s}^{\mathsf{F}*} \text{ solves (4.7) for all } f \in \mathcal{F}. \tag{4.8c}$$

Here, (4.8b) represents the constraint of the leaders f, which needs not be the same for all the leaders.

A popular way to find the SE of a Stackelberg game is to use the backward induction method as follows [24]. In the SLMF Stackelberg game, given the leader's announced strategy, we first obtain the joint strategies that maximize the utilities of the followers. Such joint strategies form the optimal response (the optimal reaction set) of the followers. Then, the leader maximizes its utility through solving its optimization problem with the followers' optimal reaction set being plugged in. Similarly, we can apply the backward induction method to find the SE of the MLMF Stackelberg game, yet the treatment for each typical MLMF case is often different and complicated. Furthermore, backward induction may not be a viable method in certain Stackelberg game cases where we cannot obtained a closed-form expression of the followers' optimal response (as a function of the leaders' strategies). To this end, more complicated methods based on solving bilevel programming problems with equilibrium constraint are typically needed [1].

4.4 Concluding Remarks

In this chapter, we have discussed important technological concepts and regarding content caching at the network edge. Then, the key components for realizing the WNV have been presented. The presented concepts are helpful for the readers to better understand the content in Chapters 5 and 6. Finally, we have discussed some fundamentals of Stackelberg game theory, which is used to develop our multi-tenancy resource trading framework in the Chapter 7.

Chapter 5

Caching for Heterogeneous Small-Cell Networks with Bandwidth Allocation and Caching-Aware BS Association

The content of this chapter was published in IEEE Wireless Communications Letters in the following paper:

Thinh Duy Tran, Tuong Duc Hoang, and Long B. Le, "Caching for heterogeneous small-cell networks with bandwidth allocation and caching-aware BS association," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 49-52, Feb. 2019.

5.1 Abstract

This paper studies the caching problem for heterogeneous small-cell networks with bandwidth allocation and caching-aware BS association. The caching control and bandwidth allocation problem aims at minimizing the request miss ratio. To solve this problem, we propose a Line-Search-based-Iterative (LSBI) algorithm which determines the solution by combining the line-search algorithm to obtain the optimal bandwidth allocation with the iterative caching algorithm to acquire a caching

solution. Numerical results demonstrate that the LSBI algorithm significantly outperforms existing caching algorithms, and is on a par with a performance bound.

5.2 Introduction

The heterogeneous network architecture with dense deployment of SBSs in coexistence with the MBS provides an important solution to better satisfy the ever-growing connected devices and mobile traffic. However, there is increasing congestion of backhaul links, which connect the MBSs and SBSs with the core network. To mitigate this problem, the cache-enabled heterogeneous small-cell network, which caches popular contents at the network edge including MBSs and SBSs, has been proposed [86]. In this cache-enabled system, both SBSs and MBSs typically cache popular contents so that requests of these highly demanded files can be served efficiently without using the backhaul links. Designing an efficient bandwidth allocation and content caching algorithm, which minimizes the request miss ratio, is essential to realize these benefits.

There are some existing works that study the caching problem for small-cell networks [15, 16] where they investigate the joint caching, routing, and channel assignment. However, all these works do not consider the stochastic behavior of content request and service processes. Meanwhile, authors in [25, 26] study the joint caching and resource allocation design based on the time-varying SNR. This design would require frequent cache updates, which is not cost efficient because the SNR usually varies quickly over time.

This paper tackles a general caching design where mobile users can be associated with either a SBS or MBS and allocated radio resources to download their desired contents. BS associations in such the heterogeneous network should take into caching decisions (i.e., being caching-aware) where users should associate with BSs which have favorable channel conditions and store their requested contents. To formulate the joint caching and bandwidth allocation problem considering dynamic requests and caching-aware BS association, we first derive the request miss ratio. Then, we propose a LSBI algorithm to determine the caching solution of the SBSs and MBS for a given bandwidth allocation. Moreover, the optimal bandwidth allocation solution is obtained by using a line-search procedure. We demonstrate that the proposed algorithm dramatically outperforms state-of-the-art schemes, namely the most popular caching with dynamic association (MPCWA),

most popular caching without dynamic association (MPCNA), and system most popular caching (SMPC) algorithms. Finally, we numerically show that our proposed LSBI algorithm achieves a small performance gap with the performance bound.

5.3 System Model and Problem Formulation

5.3.1 System Model

We consider a heterogeneous small-cell caching system consisting of one MBS denoted as BS 0 and S non-overlapping SBSs in the set $\mathcal{M}_s = \{1, \cdots, S\}$ deployed within the coverage area of the MBS.[1] Let $\mathcal{M} = \{0\} \cup \mathcal{M}_s$ denote the set of all BSs. We assume that the system bandwidth B is assigned orthogonally to the MBS and SBSs, and all SBSs reuse the same bandwidth. Let B_0 and B_s denote the bandwidth assigned to the MBS and all SBSs, respectively where $B_0 + B_s \leq B$ and we denote $\boldsymbol{B} = [B_0, B_s]$.

Let $w_m,\ m \in \mathcal{M}$ be the bandwidth required to serve a user in BS m [2]. Let $\boldsymbol{K} = [K_0, \cdots, K_m, \cdots, K_M]$ be the service capacity of the system, where K_m represents the maximum number of users that can be served simultaneously by BS $m \in \mathcal{M}$. We also denote $\boldsymbol{K} = [\boldsymbol{K}_0, \bar{\boldsymbol{K}}_0]$, where $\bar{\boldsymbol{K}}_0 = [K_1, \cdots, K_M]$. Then, to maintain the required users' QoS in cell m, K_m should satisfy $K_m \leq B_s/w_m\ \forall m \in \mathcal{M}_s$, $K_0 \leq B_0/w_0$, and $K_m \in Z^+$, where Z^+ denotes the set of non-negative integers.

We consider the following adaptive caching-aware BS association strategy. As user k in the coverage area of SBS m requests a file, the SBS will serve the user (i.e., user k will be associated with SBS m) if it is serving less than K_m users and the file is currently cached at the SBS. Otherwise, the request is redirected to the MBS. At the MBS, if the requested file is available in its cache and the MBS is serving less than K_0 users, the request will be served (i.e., user k will switch its association to the MBS). Otherwise, the request is missed. We now describe the file popularity, which is captured by file request probabilities. Denote $\mathcal{F} = \{f_1, \cdots, f_F\}$ as the set of F files of the same size, which can be stored in the caches of the BSs for future downloads. We assume that the

[1] The service areas of SBSs can be determined for a given BS association metric if their coverage areas are overlapping.

[2] The required bandwidth $w_m,\ m \in \mathcal{M}$ can be estimated based on the required average rate and worst-case spectral efficiency averaged over wireless fading channel considering inter-cell interference.

popularity distributions of the files in $\mathcal{F}$ depend on the service area where users in different areas can have different file preferences.

We assume that users request files in set $\mathcal{F}$. Let $\boldsymbol{p}_m = [p_{m1}, \cdots, p_{mF}]$ denote the file request probabilities of users in the coverage area of BS $m \in \mathcal{M}$ where p_{mf} denotes the probability that file f is requested by some user in the coverage area of BS m and $||\boldsymbol{p}_m||_1 = 1 \ \forall m \in \mathcal{M}$. We assume that content requests in BS $m \in \mathcal{M}$ follow the Poisson process with average rate λ_m(requests/s). We assume that $\boldsymbol{p}_m$ and λ_m are known.[3] Finally, we assume that it takes T_m seconds for BS m to serve one request (i.e., the file download time).

Let $\boldsymbol{x}_m = [x_{m1}, \cdots, x_{mF}]$ and $\boldsymbol{x} = [\boldsymbol{x}_0, \cdots, \boldsymbol{x}_S]$ represent the caching decisions of BS m and all BSs, respectively. Specifically, $x_{mf} \in \{0, 1\}$ denotes the caching status of file f at SBS m, where $x_{mf} = 1$ means that file f is cached at BS m, $x_{mf} = 0$, otherwise. We also denote the caching vector of all BSs in the system as $\boldsymbol{x} = (\boldsymbol{x}_s, \bar{\boldsymbol{x}}_s)$ where $\boldsymbol{x}_s$ and $\bar{\boldsymbol{x}}_s$ are the caching vectors of BS $s \in \mathcal{M}$ and other BSs, respectively. In the following, we formulate the caching problem which aims to minimize the request miss rate.

5.3.2 Problem Formulation

Let C_m denote the caching capacity of BS $m \in \mathcal{M}$, which is the maximum number of cached files at this BS. We first analyze the caching performance of a particular SBS, which is used in the problem formulation. Since the request rate associated with SBS $m \in \mathcal{M}_s$ is λ_m, the request rate for file f at SBS m is $\lambda_m p_{mf}$. If file f is not cached at SBS m, the request is redirected to the MBS. Denote $\lambda_{mf}^{\mathsf{redb}}(\boldsymbol{x}_m)$ as the redirected rate to the MBS from SBS m for file f due to the unavailability of file f in the cache. Then, $\lambda_{mf}^{\mathsf{redb}}(\boldsymbol{x}_m)$ can be expressed as $\lambda_{mf}^{\mathsf{redb}}(\boldsymbol{x}_m)$=$\lambda_m p_{mf}(1 - x_{mf})$. Consequently, the average request rate for all files to SBS m can be calculated as

$$\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m) = \sum_{f \in \mathcal{F}} \lambda_m p_{mf} x_{mf}. \tag{5.1}$$

Note that the aggregate request follows the Poisson process because all individual request processes are Poisson [27]. Recall that SBS m can serve at most K_m users simultaneously and it

[3]The request rate λ_m accounts for requests originated by users in the service area of BS m and requests of users handed off from neighboring BSs. Also, $\boldsymbol{p}_m$ can be estimated by using certain methods such as machine learning techniques.

takes T_m (s) to serve one request. Therefore, we can model the request/service at SBS m as an $M/D/K_m/K_m$ queue, which has Poisson arrivals, deterministic service time, K_m servers, and zero-length buffer. Hence, the probability a request being blocked [27] at SBS m can be expressed as

$$P_m^r(\boldsymbol{x}_m, K_m) = \frac{(\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)T_m)^{K_m}}{K_m!} \left(\sum_{i=0}^{K_m} \frac{(\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)T_m)^i}{i!} \right)^{-1}. \tag{5.2}$$

Note that if the request for file f is blocked by SBS m due to its limited service capability, the request is redirected to the MBS. Denote $\lambda_{mf}^{\mathsf{reda}}(\boldsymbol{x}_m, K_m)$ as the redirected request rate to the MBS from SBS m due to its limited service capability then this parameter can be calculated as $\lambda_{mf}^{\mathsf{reda}}(\boldsymbol{x}_m, K_m)$=$\lambda_{mf}^{\mathsf{req}}(\boldsymbol{x}_m)P_m^r(\boldsymbol{x}_m, K_m)$. As all requests which are rejected by SBSs due to either the limited service capability or the unavailability of requested files at the SBSs' caches are redirected to the MBS, we can calculate the total request rate of file f redirected to the MBS as $\lambda_f^{\mathsf{red}}(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{K}}_0)$= $\sum\limits_{m\in\mathcal{M}_s} \left(\lambda_{mf}^{\mathsf{redb}}(\boldsymbol{x}_m)+\lambda_{mf}^{\mathsf{reda}}(\boldsymbol{x}_m, K_m)\right)$. Therefore, the total request rate of file f to the MBS including original and redirected requests can be expressed as $\lambda_{Mf}^{\mathsf{req}}(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{K}}_0)$=$\lambda_0 p_{0f}$+$\lambda_f^{\mathsf{red}}(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{K}}_0)$.

In general, if file f is not available in the MBS's cache, it can take very long time to download the file from its content server and being transmitted through the backhaul network. This leads to very long delay degrading users' quality of experience. To investigate the efficiency of the proposed design, the event that an original or redirected request of a file which is unavailable at the MBS is considered as a request miss event.

The request miss rate associated with the MBS due to the unavailability of the files can be expressed as $\lambda_M^{\mathsf{rb}}(\boldsymbol{x}, \bar{\boldsymbol{K}}_0) = \sum_{f\in\mathcal{F}} \lambda_{Mf}^{\mathsf{req}}(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{K}}_0)(1 - x_{0f})$. Consequently, the request rate for all files at the MBS can be calculated as

$$\lambda_M^{\mathsf{reqa}}(\boldsymbol{x}, \bar{\boldsymbol{K}}_0) = \sum_{f\in\mathcal{F}} \lambda_{Mf}^{\mathsf{req}}(\bar{\boldsymbol{x}}_0, \bar{\boldsymbol{K}}_0)x_{0f}. \tag{5.3}$$

Again, the corresponding request process to the MBS for which underlying requests are served is also a Poisson process. Hence, we can model content request/service at the MBS as an $M/D/K_0/K_0$ queue. Consequently, the request miss probability due to limited serving capability of the MBS can be calculated similar to (5.2), i.e., $P_0^r(\lambda_M^{\mathsf{reqa}}((\boldsymbol{x}, \bar{\boldsymbol{K}}_0), \boldsymbol{K})) = \frac{(\lambda_M^{\mathsf{reqa}}(\boldsymbol{x}, \bar{\boldsymbol{K}}_0)T_0)^{K_0}}{K_0!} \left(\sum\limits_{i=0}^{K_0} \frac{(\lambda_M^{\mathsf{reqa}}(\boldsymbol{x}, \bar{\boldsymbol{K}}_0)T_0)^i}{i!} \right)^{-1}$. As a result, the request miss rate due to the limited serving capability of the MBS can be expressed as $\lambda_M^{\mathsf{ra}}(\boldsymbol{x}, \boldsymbol{K}) = \lambda_M^{\mathsf{reqa}}(\boldsymbol{x}, \bar{\boldsymbol{K}}_0)P_0^r(\lambda_M^{\mathsf{reqa}}((\boldsymbol{x}, \bar{\boldsymbol{K}}_0), \boldsymbol{K}))$. Finally, the total request miss rate of the system

can be calculated as $\lambda^{\mathsf{miss}}(\boldsymbol{x},\boldsymbol{K})=\lambda_M^{\mathsf{ra}}(\boldsymbol{x},\boldsymbol{K})+\lambda_M^{\mathsf{rb}}\left(\boldsymbol{x},\bar{\boldsymbol{K}}_0\right)$. Our design problem which minimizes the request miss rate can be formulated as [4]

$$\begin{aligned}\min_{\boldsymbol{x},\boldsymbol{B},\boldsymbol{K}} \quad & \lambda^{\text{miss}}(\boldsymbol{x},\boldsymbol{K}) \\ s.t. \quad & \sum_{f\in\mathcal{F}} x_{mf}\leq C_m,\ x_{mf}\in\{0,1\},\forall m\in\mathcal{M},f\in\mathcal{F} \\ & K_m\leq B_s/w_m\ \forall m\in\mathcal{M}_s, K_0\leq B_0/w_0 \\ & B_0+B_s\leq B, K_m\in Z^+\ \forall m\in\mathcal{M}.\end{aligned} \tag{5.4}$$

5.4 Algorithm Design

5.4.1 General Algorithm

Algorithm 5.1. JOINT BW ALLOCATION AND CACHING ALGORITHM (LSBI)

Initialization: $K_0^*=0$, $K_0^{\mathsf{max}}=\lfloor B/w_0\rfloor$, $\lambda_{\mathsf{opt}}^{\mathsf{miss}}=\infty$.
repeat
  $K_0^*=K_0^*+1$
  Calculate $\boldsymbol{B}^*,\boldsymbol{K}^*$ according to **(i)**, **(ii)**, and **(iii)**.
  Solve problem (5.5) by using Algorithm 5.2 to obtain $\lambda^{\mathsf{miss}}(\boldsymbol{K}^*)$ and $\boldsymbol{x}^*$.
  if $\lambda_{\mathsf{opt}}^{\mathsf{miss}}>\lambda^{\mathsf{miss}}(\boldsymbol{K})$ **then**
    Set $\lambda_{\mathsf{opt}}^{\mathsf{miss}}\leftarrow\lambda^{\mathsf{miss}}(\boldsymbol{K}^*)$.
    Set $\boldsymbol{x}_{\mathsf{opt}}\leftarrow\boldsymbol{x}^*$, and $\boldsymbol{B}_{\mathsf{opt}}\leftarrow\boldsymbol{B}^*$
  end if
until $K_0^*>K_0^{\mathsf{max}}$.
Output λ^{miss}, $\boldsymbol{B}_{\mathsf{opt}}$ and $\boldsymbol{x}_{\mathsf{opt}}$.

Note that K_0 is an integer variable, and it is limited by the system bandwidth, $K_0\leq B/w_0$. Therefore, we can perform line search for all possible solutions of K_0. For a given optimal value K_0^*, to obtain the optimal solution of problem (5.4), the optimal bandwidth allocation and service capacity of the SBS can be determined as follows: **(i)** $B_0^*=K_0^*w_0$, **(ii)** $B_s^*=B-K_0^*w_0$, and **(iii)** $K_m^*=\lfloor(B-K_0^*w_0)/w_m\rfloor\ ,\ \forall m\in\mathcal{M}_s$.

Substituting $\boldsymbol{B}^*$ and $\boldsymbol{K}^*$ to problem (5.4) yields the following caching optimization problem

$$\begin{aligned}\min_{\boldsymbol{x}} \quad & \lambda^{\text{miss}}(\boldsymbol{x},\boldsymbol{K}^*) \\ s.t. \quad & \sum_{f\in\mathcal{F}} x_{mf}\leq C_m,\ x_{mf}\in\{0,1\}.\end{aligned} \tag{5.5}$$

[4] By minimizing the request miss rate, we can indirectly reduce the backhaul traffic load and high service delay due to content download from content servers.

Based on these observations, we propose Algorithm 5.1 to solve problem (5.4). In particular, it solves problem (5.4) by line-searching over possible values of K_0. For a given value of K_0^*, Algorithm 5.1 calculates the bandwidth allocation and serving capacity vectors $\boldsymbol{B}^*$ and $\boldsymbol{K}^*$. Then, it solves the caching problem (5.5) using Algorithm 5.2 explained in the next section. The request miss rates obtained from solving problem (5.5) for different values of $\boldsymbol{K}^*$ are compared to determine the optimal solution which achieves the lowest request miss rate. In the following, we present an algorithm to solve the caching optimization problem (5.5) for a given $\boldsymbol{K}^*$.

5.4.2 Caching Algorithm

In problem (5.5), since $\boldsymbol{K}^*$ is given, we omit $\boldsymbol{K}^*$ in all related notations in the following for brevity. The objective function of (5.5) can be re-expressed as

$$\lambda^{\text{miss}}(\boldsymbol{x})=\lambda_M^{\mathsf{reqa}}(\boldsymbol{x})P_0^r(\lambda_M^{\mathsf{reqa}}(\boldsymbol{x}))-\lambda_M^{\mathsf{reqa}}(\boldsymbol{x})+\sum_{f\in\mathcal{F}}\lambda_{Mf}^{\mathsf{req}}(\boldsymbol{x}). \tag{5.6}$$

Moreover, $\sum_{f\in\mathcal{F}}\lambda_{Mf}^{\mathsf{req}}(\boldsymbol{x})$ in equation (5.6) can be expressed as

$$\sum_{f\in\mathcal{F}}\lambda_{Mf}^{\mathsf{req}}(\boldsymbol{x}) = \lambda_{\mathsf{c}}+\sum_{m\in\mathcal{M}_s}\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)P_m^r(\boldsymbol{x}_m)-\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m) \tag{5.7}$$

where $\lambda_{\mathsf{c}} = \sum_{m\in\mathcal{M}}\lambda_m$. Now, we define functions $g_0(\lambda) \triangleq \lambda P_0^r(\lambda)-\lambda$ and $g_m(\lambda) \triangleq \lambda P_m^r(\lambda)-\lambda$, which correspond to the MBS and SBSs, respectively. Then, $\lambda^{\mathsf{miss}}(\boldsymbol{x})$ can be written as

$$\lambda^{\mathsf{miss}}(\boldsymbol{x}) = \lambda_{\mathsf{c}} + g_0(\lambda_M^{\mathsf{reqa}}(\boldsymbol{x})) + \sum_{m\in\mathcal{M}_s} g_m(\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)) \tag{5.8}$$

where $\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)$ and $\lambda_M^{\mathsf{reqa}}(\boldsymbol{x})$ are given in equations (5.1) and (5.3), respectively. Next, we state the property of functions $g_m(\lambda)$ for all $m \in \mathcal{M}$ in Proposition 5.1.

Proposition 5.1. *For each $m \in \mathcal{M}$, $g_m(\lambda)$ decreases with λ.*

Proof. Let us consider the loss rate function $L(\lambda) = \lambda P(\lambda)$ where λ and $P(\lambda)$ are the arrival rate and the blocking function as in (5.2), respectively. From Section II-B in [107], $\frac{\partial L(\lambda)}{\partial \lambda} \in [0, 1]$. Therefore, $\frac{\partial g_m}{\partial \lambda} = \frac{\partial (L(\lambda)-\lambda)}{\lambda} \in [-1, 0]$, which means $g_m(\lambda)$ are decreasing functions of λ for all $m \in \mathcal{M}$. □

The decreasing property of $g_m(\lambda)$ with respect to λ is leveraged to design the caching algorithm. Specifically, $\lambda_M^{\mathsf{reqa}}(\boldsymbol{x})$ and $\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)$ are increasing functions of $\boldsymbol{x}$ and $\boldsymbol{x}_m$ for all $m \in \mathcal{M}$ in (5.8). Hence, to minimize the request miss ratio for a given $\boldsymbol{K}^*$, each BS has to cache to its full storage capacity to attain higher λ. Since solving the caching problem (5.5) optimally requires an extensive computation due to binary caching vector $\boldsymbol{x}$, we propose an iterative algorithm to solve problem (5.5) by sequentially solving the caching problem of each BS for a given caching solutions of other BSs until convergence. In what follows, we describe how to solve the caching problems of the SBSs and MBS.

5.4.2.1 Caching Decisions for the SBSs

Let $\boldsymbol{x}^t$ denote the caching solution in iteration t. Moreover, we denote $\mathcal{F}_0^t$ and $\bar{\mathcal{F}}_0^t$ as the sets of cached and un-cached files in the MBS in iteration t, respectively. Then, the caching decision sub-problem for SBS m in iteration $t+1$ can be stated as

$$\begin{aligned} &\min_{\boldsymbol{x}_m} \ \lambda^{\mathsf{miss}}(\boldsymbol{x}_m) = \lambda_{\mathsf{c}} + g_0(\lambda_M^{\mathsf{reqa}}(\boldsymbol{x}_m)) + g_m(\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)) \\ &s.t. \quad \sum_{f\in\mathcal{F}} x_{mf} \leq C_m, x_{mf} \in \{0,1\}, \ \forall f \in \mathcal{F}. \end{aligned} \tag{5.9}$$

Problem (5.9) is still a mixed integer program which is challenging to solve. Since SBS m should cache C_m files, we propose a caching scheme in which SBS m caches $C_m - \bar{C}_m$ and $\bar{C}_m$ most popular files in sets $\mathcal{F}_0^t$ and $\bar{\mathcal{F}}_0^t$, respectively. Denote $\bar{C}_m^*$ as the optimal value of $\bar{C}_m$, which can be determined by a line-search algorithm since $\bar{C}_m^* \in [0, C_m]$.

5.4.2.2 Caching Decisions for the MBS

The caching problem of the MBS can be stated as

$$\begin{aligned} &\min_{\boldsymbol{x}_0} \quad \lambda_{\mathsf{c}} + g_0(\lambda_M^{\mathsf{reqa}}(\boldsymbol{x}_0)) + \sum_{m\in\mathcal{M}_s} g_m(\lambda_m^{\mathsf{req}}(\boldsymbol{x}_m)) \\ &s.t. \quad \sum_{f\in\mathcal{F}} x_{0f} \leq C_0, x_{0f} \in \{0,1\} \ \forall f \in \mathcal{F}. \end{aligned} \tag{5.10}$$

The objective function of problem (5.10) can be written as $\lambda^{\mathsf{miss}}(\boldsymbol{x}_0)=\lambda_{\mathsf{c}}^{M}+g_0(\sum_{f\in\mathcal{F}}\lambda_{Mf}^{\mathsf{req}}(\bar{\boldsymbol{x}}_0)x_{0f})$. As $g_0(\lambda)$ is a decreasing function, the optimal solution of problem (5.10) is obtained when $\sum_{f\in\mathcal{F}}\lambda_{Mf}^{\mathsf{req}}(\bar{\boldsymbol{x}}_0)$ is maximized. Denote $\mathcal{C}_0^*$ as the set of C_0 highest values of $\lambda_{Mf}^{\mathsf{req}}(\bar{\boldsymbol{x}}_0)$. Then, to obtain the optimal solution of problem (5.10), MBS should cache all files in $\mathcal{C}_0^*$.

5.4.3 Final Caching Algorithm

Algorithm 5.2. CACHING ALGORITHM

Initialization MBS caches its C_0 most popular files and SBS m caches its C_m most popular files. Set max iteration N and tolerance ϵ.
$t = 0$
repeat
  $t = t + 1$
  $m = t$ modulo M
  if $m = 0$ **then**
    Perform caching for MBS to obtain $\boldsymbol{x}_0^{t+1^*}$
  else
    Perform caching for SBS m to obtain $\boldsymbol{x}_m^{t+1}$
  end if
  $\boldsymbol{x}_m^{t+1^*} = \text{argmin}_{\boldsymbol{x}_m^{t^*},\boldsymbol{x}_m^{t+1}}\{\lambda^{\mathsf{miss}}(\boldsymbol{x}_m^{t^*}), \lambda^{\mathsf{miss}}(\boldsymbol{x}_m^{t+1})\}$
until $|\lambda^{\mathsf{miss}}(\boldsymbol{x}_m^{t+1}) - \lambda^{\mathsf{miss}}(\boldsymbol{x}_m^{t})| < \epsilon$ or $t > N$
Output $\boldsymbol{x}^*$

From the caching design described in Algorithm 2, we can see that it creates a sequence of feasible solutions for problem (5.5) where the value of its objective function monotonically decreases over iterations. Therefore, Algorithm 2 converges to a feasible solution.

5.4.4 Complexity Analysis

We now analyze the complexity of the proposed Algorithm 5.1. The line-search procedure to find $\boldsymbol{K}^*$ requires up to K_0^{max} iterations. Algorithm 5.2 finds caching solutions for the MBS and SBSs. The caching solution for the MBS requires to calculate the request rate from its own users and SBSs' users over all the files $\boldsymbol{F}$, thus it has complexity $\mathcal{O}(MF)$. The caching solution for the SBSs is obtained by the line-search procedure over the storage capacity C_m, hence it has complexity $\mathcal{O}(\sum_{m\in\mathcal{M}} C_m) \approx \mathcal{O}(F)$ because the cache capacity at each SBS is typically much smaller than the total number of files in $\mathcal{F}$. As a result, Algorithm 5.1 has the complexity of $\mathcal{O}(K_0^{\mathsf{max}}(MF + F)) \approx \mathcal{O}(K_0^{\mathsf{max}}MF)$, which is linear with key system parameters.

5.4.5 Performance Bound

We will compare the performance of our algorithm with a performance bound which is obtained as follows. We first execute Algorithm 5.1 to obtain the bandwidth allocation solution. Then, the performance bound is obtained based on this bandwidth allocation solution and the caching solution with infinite MBS's storage capacity. In this case, the MBS should cache all the files in $\boldsymbol{F}$ and the most popularity caching strategy is the optimal caching scheme for each SBS. The resulting request miss rate is the lower bound of that under the proposed algorithm because we have indeed finite MBS's storage capacity.

5.5 Numerical Results

We consider a simulation setting with a single MBS and $|\mathcal{M}_s| = 9$ SBSs, each with coverage radius $d = 50m$, deployed within the coverage area of the MBS with coverage radius $R = 500m$ unless stated otherwise. We set $B = 20$ and $w_m = 1,\ \forall m \in \mathcal{M}$ bandwidth units. The number of files is set $F = 100$ and content requests follow the Zipf distribution with a skew parameter $\gamma = 0.8$. In addition, the storage capacities of the MBS and SBSs are assumed to be $C_0 = 20$ and $C_m = 5$, respectively. The content request processes at the MBS and SBSs are Poisson processes with the normalized rates of 10^{-5} $\mathsf{requests/s/m^2}$ and 10^{-4} $\mathsf{requests/s/m^2}$, respectively. The service times of one request for the MBS and SBS are set to $10s$ and $5s$, respectively.

We compare the performance of our proposed algorithm with three other algorithms, namely MPCWA, MPCNA, and SMPC, as introduced in Section I. All algorithms determine the bandwidth allocation solution by using the LSBI algorithm. Moreover, the MPCWA and MPCNA algorithms caches most popular files for each BS. MPCWA algorithm considers dynamic BS association while the MPCNA algorithm assumes that requests from SBSs cannot be redirected to the MBS. In the SMPC algorithm, each BS caches the most popular files where the average content popularity over cells is used. Finally, we also compare our proposed algorithm with the performance bound described in Section III-E.

Fig. 5.1 demonstrates the request miss rate versus the system bandwidth B. Next, Fig. 5.2 shows the request miss rate versus the caching capacity of each SBS C_m. In both figures, the

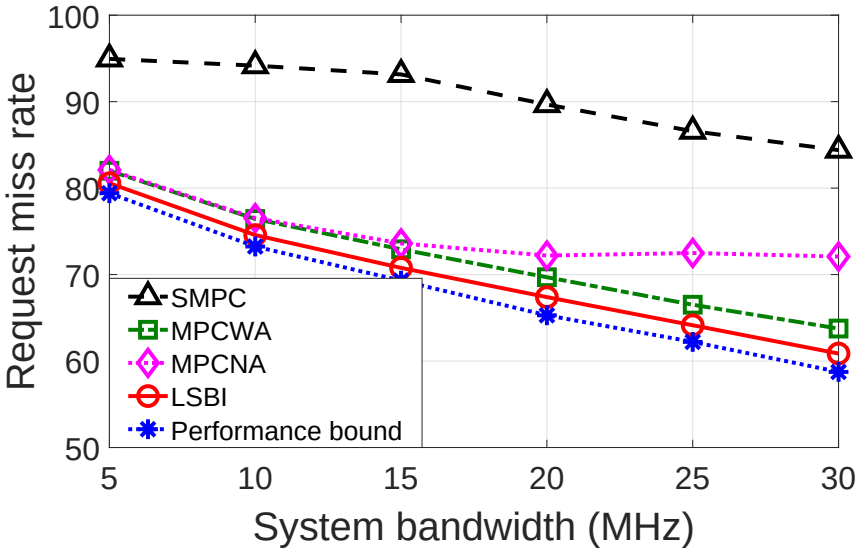


Figure 5.1 – Request miss ratio versus system bandwidth B.

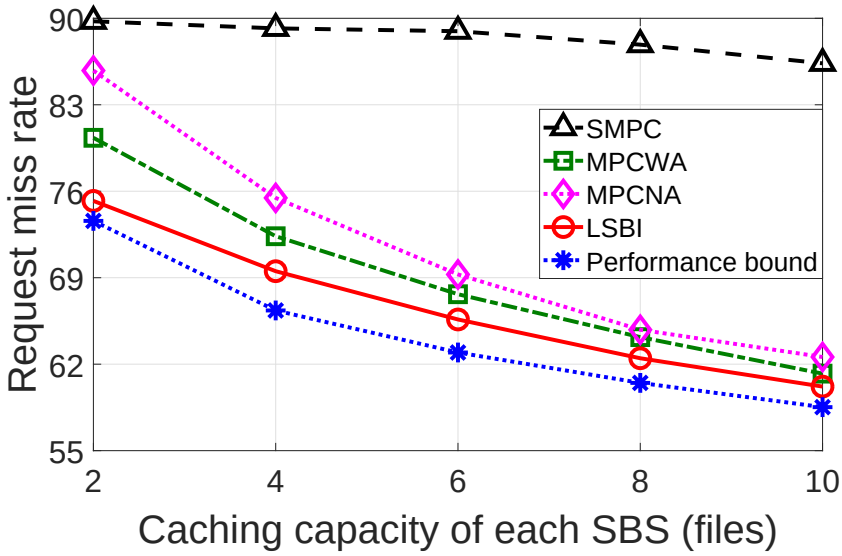


Figure 5.2 – Request miss ratio versus caching capacity of SBS C_m.

LSBI algorithm outperforms the three baseline algorithms. Figs. 5.1 and 5.2 also show the small performance gap between our proposed algorithm and the performance bound, which confirms the efficacy of our proposed framework. In fact, adding more caching capacity to the MBS only marginally decreases the request miss rate for large system bandwidth.

Fig. 5.3 illustrates the request miss rates of the LSBI algorithm versus the MBS's coverage radius for different values of γ. The request miss rate of the LSBI algorithm is smaller as γ becomes

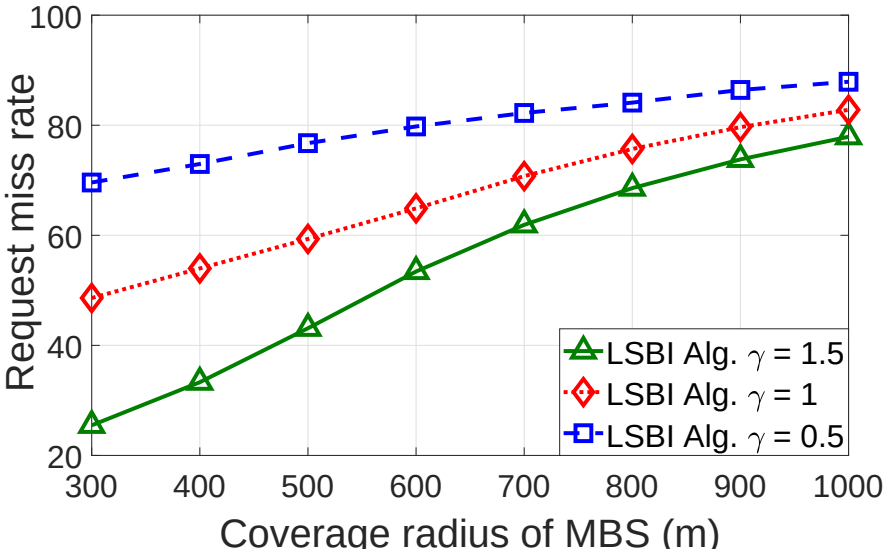


Figure 5.3 – Request miss ratio versus coverage radius of MBS R.

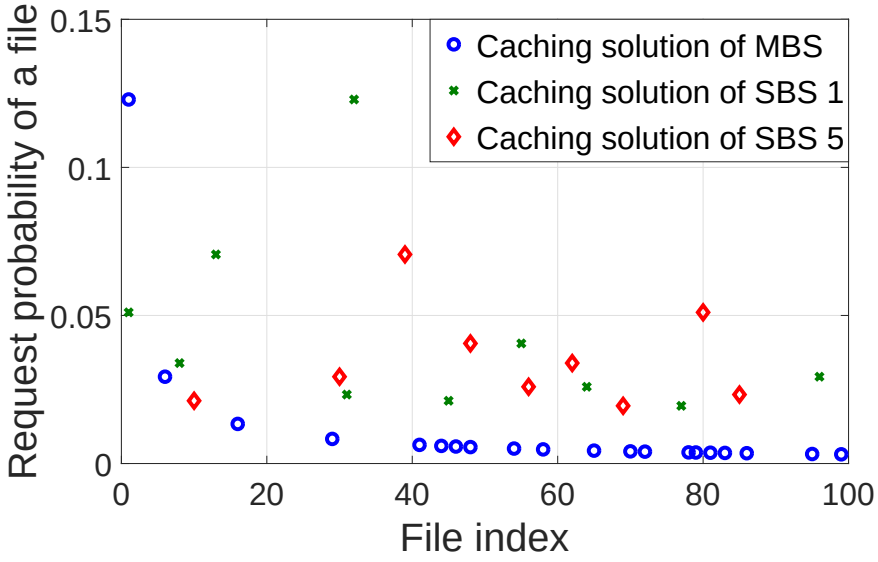


Figure 5.4 – Caching solution of MBS and SBSs.

larger. Moreover, the higher value of R results in increasing request miss rate since larger R leads to the higher request rate.

Finally, Fig. 5.4 shows the cached files at the MBS and 2 SBSs in one particular system realization where the x-axis indicates the file indices and the y-axis shows the request probabilities of different files. It can be seen that SBSs tend to cache their most popular files while the caching solution of the MBS contains files ranging from low to high request probabilities. This is because each SBS would attempt to minimize the redirected request rate to the MBS by caching its most

popular files. Moreover, the MBS accommodates redirected requests from all SBSs; therefore, its caching solution contains files spreading out from low to high preferences.

5.6 Conclusion

In this paper, we have proposed a bandwidth allocation and caching design for a heterogeneous cellular network. Numerical results have confirmed that the proposed algorithm performs significantly better than the state-of-the-art algorithms and similarly with a performance bound.

Chapter 6

Joint Resource Allocation and Content Caching in Virtualized Content-Centric Wireless Networks

The content of this chapter was published in IEEE Access in the following paper:

Thinh Duy Tran and Long Bao Le, "Joint resource allocation and content caching in virtualized content-centric wireless networks", *IEEE Access*, vol. 6, pp. 11329–11341, 2018.

6.1 Abstract

Efficient content caching plays a crucial role in QoS enhancement and congestion mitigation of the backhaul and CN for the 5G wireless cellular network, which must support a large amount of multimedia and video contents. WNV provides a novel paradigm shift in 5G system design which enables to better utilize network resources, rapid development of new services, and reduce the operation cost (OPEX). Harmonized deployment of a content caching strategy in the VWN environment, however, requires a suitable radio resource allocation framework to realize the great benefits of these technologies. In this paper, we study the joint resource allocation and content caching problem which aims at efficiently utilizing the radio and content storage resources in the

highly congested backhaul scenario. In this design, we minimize the maximum content request rejection rate experienced by users of different MVNOs in different cells, which results in a MINLP. We solve this difficult optimization problem by proposing a bisection-search based algorithm that iteratively optimizes the resource allocation and content caching placement. We further propose a low-complexity heuristic algorithm which achieves moderate performance loss compared to the bisection-search based algorithm. Extensive numerical results confirm the efficacy of our proposed framework which significantly reduces the maximum request outage probability compared to other benchmark algorithms.

6.2 Introduction

WNV has been recognized as an essential technology for the 5G wireless network [2], where WNV allows multiple MVNOs to share the same network infrastructure owned and managed by an InP. Moreover, the InP must flexibly allocate network resources such as transmission power, bandwidth, and storage to MVNOs based on their demands and mutual contracts in an efficient manner so that their operations and services can be harmonized on the same infrastructure. The MVNOs can then utilize the resources rent from the InP to provide mobile services to their UEs with committed QoS. The WNV can potentially help reduce the operation cost (OPEX) and capital expenditure cost (CAPEX) significantly while efficiently utilizing network resources, and guaranteeing the QoS. On the other hand, there has been a cloudification trend in engineering future wireless cellular systems with adaptive function splits where certain network functions and algorithms are deployed in the cloud, which is connected with BSs (also called RRHs in the literature) through the backhaul network[1] [6, 19]. When sophisticated network and communication functions such as baseband signal processing and signal detection are performed in the cloud (technically in the baseband units (BBUs)), very large backhaul bandwidth is required to meet the strict latency constraint of the I/Q data exchanges between the BSs and the cloud.

Furthermore, the future wireless network must cope with the explosion of the mobile traffic which has growth rate about 131% per year [6] where the mobile video traffic will account for about 75% of the overall mobile traffic by 2020 [5]. This huge traffic demand will put great pressure on not only the wireless access network but also the backhaul network connecting the BSs and the CN.

[1]This is called a fronthaul network some time in the literature.

Therefore, fundamental improvement of the efficiency of radio resource utilization and mitigation of backhaul congestion become very critical research issues, especially in the VWN environment. Toward this end, development of a joint efficient content caching and virtualized resource allocation framework for the 5G wireless network is required to resolve the access and backhaul network congestion while enhancing users' QoS [3, 15, 16].

Content caching at the network edge has been shown to lead to significant improvement in users' QoS [15, 16]. In particular, by deploying content storage at BSs and prefetching popular contents to these storage facilities, we can reduce access latency and mitigate traffic congestion on the backhaul links during high-traffic hours, thus improving network performance and users' QoS [14]. However, in the VWN environment where multiple MVNOs operate on the shared infrastructure with limited storage capacity, network performance improvement due to content caching could be less significant if the InP simply partitions the available storage capacity and allocates these storage partitions to MVNOs. Therefore, efficient and shareable content caching among MVNOs jointly with radio resource allocation can enable to boost the network performance.

6.2.1 Related Work

Most existing works in the literature treat the WNV, content caching, resource allocation design issues separately. In particular, Poularakis et al. [13] focus on improving the caching performance, i.e., increasing the hit rate and reducing access delay, for small-cell wireless networks. The authors in [15, 16] only consider joint content caching with conventional resource allocation in wireless networks without WNV. There are a few works such as [25, 26] studying the joint caching, resource allocation, and WNV. However, adaptation of cache placement based on the SNR may not be cost efficient, since caching decisions at BS should be made over a long time scale while the SNR typically varies rapidly.

Recently, different content caching frameworks have been introduced to leverage the evolution of network architecture. In [29] and its related work, the authors propose to install storage repository at femtocells, which are deployed in high density and closer to UEs, to assist the macro BS through offloading content requests. Another approach called hierarchical caching is to leverage the hierarchical structure of modern network topology and coding theory for content caching as in [30]. To adapt to C-RAN based network architecture, Tang et. al. in [31] propose to install the storage

repository not only at the RRH but also in the cloud, which can be considered as another version of hierarchical caching.

Most of these existing works do not consider the highly congested network scenario due to the lack of radio resource and bandwidth in the wireless access and backhaul links [32]. In fact, the uRLLC service type requires a large number of reliable connections [82]. Moreover, a great deal of control signaling data, which would consume valuable radio resources in the wireless access links and result in further network congestion.

For WBN [83–85], wireless backhaul links are used in lieu of the traditional cable links for connection with the CN. Wireless networks with wireless backhaul may suffer from performance degradation if the radio resources allocated for backhaul links are not sufficient and/or heavy contents such as large video files are transferred over these backhaul links from the CN. Another innovative content caching approach which enables to significantly reduce the backhaul traffic is to leverage device caching and D2D communication [87], which caches content on mobile device's storage. This approach unfortunately is hindered by the mobility nature of mobile devices and their economical selfishness in providing content caching, which could consume their limited battery and storage capacity.

6.2.2 Research Contributions

Motivated by the aforementioned issues, we consider the content caching design for wireless networks with highly congested backhaul links. Specifically, we study the joint radio resource allocation and content caching design for the VWN where we make the following key contributions.

- We present the problem formulation that minimizes the maximum request outage probability for all MVNOs at different BSs while avoiding content caching redundancy at the storage locations. This design captures dynamic content request arrivals/departures and possible cache misses when a requested content is not cached or radio resources are not sufficient to support the user-BS communications. Our framework considers the network scenario in which the backhaul links are highly congested and wireless access bandwidth is limited. Specifically, the design objective enables to maximize the content accesses from the BSs, which mitigates the long service latency due to content transfer over congested backhaul links. Moreover, the

similarity on the content set and content preferences among different MVNOs is exploited where different MVNOs are allowed to share the same cached contents at individual BSs.

- To solve the underlying MINLP problem, we propose a novel iterative algorithm based on the bisection-search method. In particular, the proposed algorithm exploits special properties of the Erlang-B function and the optimal channel allocation. The algorithm is proved to converge to a local optimal solution. Furthermore, we propose a caching decision rounding solution and a low-complexity algorithm with more affordable computation burden.

- Extensive numerical results are presented to demonstrate the efficacy of our proposed bisection search based algorithm, the caching decision rounding technique and the heuristic algorithm. Specifically, we study different scenarios where the content popularity patterns of different MVNOs are in same and different orders at each BS. It is confirmed through numerical studies that our proposed design performs well in both scenarios.

The rest of our paper is as follows. We describe the system model and the problem formulation in Sections 6.3 and 6.4, respectively. We then present the algorithms to solve the considered problem in Section 6.5. Section 6.6 show the numerical results and Section 6.7 concludes our paper. The summary of key notations is presented in Table 6.1.

6.3 System Model

We consider a downlink virtualized orthogonal frequency-division multiple access (OFDMA) multi-cell wireless network with caching repository deployed at each BS. The system consists of K BSs in a set denoted as $\mathcal{K} = \{1, ..., K\}$. These BSs are connected to the CN via *highly congested* backhaul links. It is assumed that the network has W^{max} wireless channels of equal bandwidth serving all the UEs associated with these BSs. To avoid severe interference in the network, we assume these channels are allocated in the orthogonal manner.[2] This network infrastructure including all BSs, the backhaul and core networks, radio and storage resources are assumed to be owned and managed by an InP. For illustration, our system model is depicted in Figure 6.1.

[2]These channels can represent frequency bands in OFDM systems or subchannels as in LTE-based systems [108].

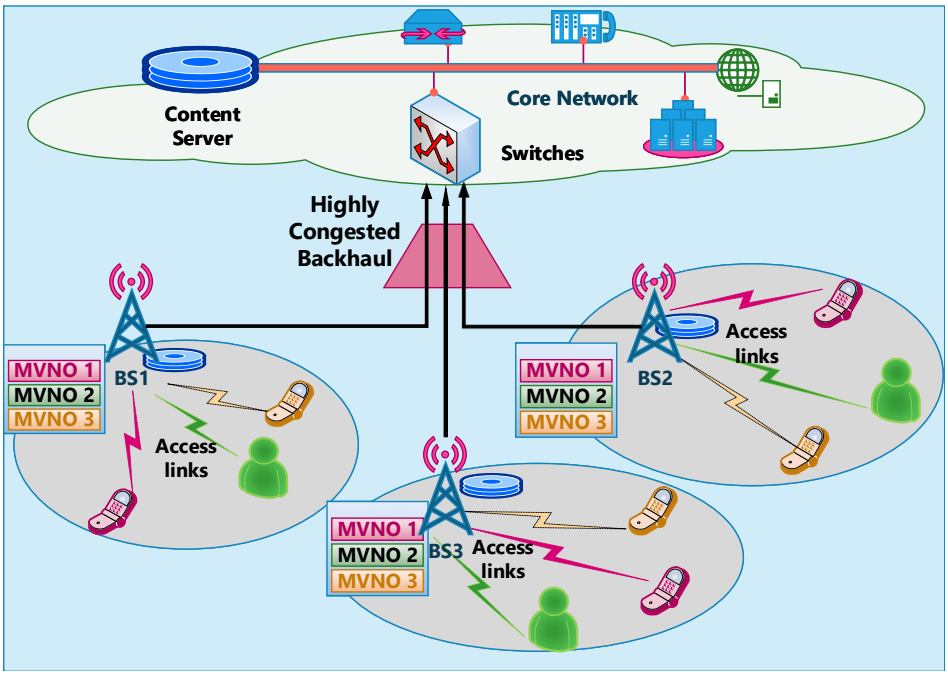


Figure 6.1 – System model

In this network, the InP serves M MVNOs in the set $\mathcal{M} = \{1, ..., M\}$, which rent resources and infrastructure to serve their UEs. For convenience, we use MVNO (m, k) to denote MVNO m associated with BS k. For the channel allocation, we denote $\boldsymbol{w} = \{w_{11}, ..., w_{km}, ..., w_{KM}\}$ as the channel allocation vector, whose elements w_{km} represent the number of wireless channels allocated to MVNO (m, k).

It is assumed that UEs of each MVNO m are interested in accessing contents in a content set $\mathcal{F} = \{f_1, ..., f_F\}$ of F files or contents. Note that it is plausible to assume a common file set F for all MVNOs as the common file set can be formed by aggregating all MVNOs' file sets. Content requests from UEs of MVNO m in the coverage of BS k are assumed to follow the Poisson process with an average rate λ_{km} (requests/s). Note that the Poisson process is the popular mathematical tool to model random arrival processes in practical telecommunication and computer systems.

Without loss of generality, we assume that each file in $\mathcal{F}$ has the normalized size of 1 unit and the BSs can cache popular contents in advance for future possible accesses [13, 25]. In practice, a large file, e.g., a movie file, can be split into equal-size chunks of data whose size can then be normalized to 1. Let C_k denote the capacity of the storage repository installed at BS k, which can cache up to C_k files where $C_k \in \mathbb{Z}_+$. Moreover, $\mathcal{Q}_{km} = \{q_{km1}, ..., q_{kmF}\}$ denotes the content request probability distribution where q_{kmf} represents the probability that UEs of MVNO (m, k) requests

Tableau 6.1 – Summary of Key Notations

Notation	Description
K	Number of base stations
M	Number of MVNOs
MVNO (m,k)	MVNO m associated with BS k
$W^{\max}$	Number of wireless channels
w_{km}	Number of channels allocated to MVNO (m,k)
$\boldsymbol{w}$	Channel allocation vector
λ_{km}	Average number of arrival request from MVNO (m,k)
C_k	Storage capacity of BS k
F	Number of contents/files
q_{kmf}	Request probability of file f by MVNO (m,k)
$\mathcal{Q}_{km}$	Request probability distribution for MVNO (m,k)
x_{kmf}	Caching decision variable for file f by MVNO (m,k)
$\boldsymbol{x}_{km}$	Caching decision vector for MVNO (m,k)
$\boldsymbol{x}$	Caching decision vector
$h_{kmf}(\boldsymbol{x})$	Cache-hit rate for file f by MVNO (m,k)
$h_{km}(\boldsymbol{x})$	Total cahe-hit rate by MVNO (m,k)
$\bar{h}_{kmf}(\boldsymbol{x})$	Cache-missed rate for file f by MVNO (m,k)
$\bar{h}_{km}(\boldsymbol{x})$	Total cahe-missed rate by MVNO (m,k)
T_{km}	Service time (s) for BS k to serve a cache-hit file request from MVNO m
$P_{km}(\boldsymbol{x},\boldsymbol{w})$	Probability that there are w_{km} ongoing cache-hit file requests from MVNO (m,k)
$\mu_{km}(\boldsymbol{x},\boldsymbol{w})$	Rejection rate for the cache-hit request from MVNO (m,k)
$\Phi_{km}(\boldsymbol{x},\boldsymbol{w})$	Total file request outage probability of MVNO (m,k)
$Z(f,\gamma)$	Zipf distribution of f-th most popular file with skewness γ

file f. Therefore, we have

$$\sum_{\mathcal{F}} q_{kmf} = 1, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}. \tag{6.1}$$

Note also that UEs must communicate with their associated BSs to download the requested file. We assume that $\mathcal{Q}_{km}$ can be different for different MVNO-BS pairs (m,k) and this allows us to capture the spatial variations of content popularity patterns.

To model the content caching placement, we introduce the caching decision vectors $\boldsymbol{x}_{km} = \{x_{km1}, ..., x_{kmF}\}$ for BS k and MVNO m and $\boldsymbol{x} = \{\boldsymbol{x}_{11}, ..., \boldsymbol{x}_{km}, ..., \boldsymbol{x}_{KM}\}$ to denote the content caching decision vector for all MVNOs (m,k), where $x_{kmf} \in \{0,1\}$ and $x_{kmf} = 1$ if file f is cached at BS k to serve requests from MVNO m, and $x_{kmf} = 0$, otherwise. Moreover, to ensure

some minimum QoS requirement, we assume that one channel (if available) must be allocated to download a requested file from the associated BS for any UE.[3]

In general, a particular UE can download its requested file from the content server (CS) in the CN and such content must be transferred over both backhaul and wireless access networks if a particular file is not cached at the UE's associated BS. With the highly congested backhaul network, the end-to-end content download time from CN can be very large, which severely affects the user's QoS. Therefore, to maintain satisfactory users' QoS, we assume that any particular content request results in a cache hit only if the requested content is cached at the UE's associated BS and there are available channels to be support the UE-BS communications.

To elaborate the content access and transmission, let us consider a particular request from MVNO m to file $f \in \mathcal{F}$ at BS k. If file f is cached at this BS (i.e., cache-hit file request) and there is an available channel in the budget of w_{km} channels, the request of file f is accepted and the file is downloaded to the requesting UE. In contrast, a content request is rejected if all w_{km} channels are allocated for serving other file requests (even if the requested content is cached at the UE's associated BS). As discussed above, due to the highly congested backhaul links between the BSs and CN, we do not account for the case where these cache-missed file requests are redirected to the CS in the CN in our design and optimization. In the next sections, we present the problem formulation and our proposed algorithm.

6.4 Problem Formulation

We now describe the joint content caching and channel allocation problem which aims to minimize the maximum file request outage probability over all MVNOs and BSs. Because different MVNOs could share the cached files at each BS, we do not cache the same file at the same BS's caching repository to serve requests from different MVNOs. Such avoidance of content caching redundancy can be mathematically expressed by the following constraints:

$$\sum_{i \in \mathcal{M}} x_{kif} \leq 1, \forall k \in \mathcal{K}, \forall f \in \mathcal{F}. \tag{6.2}$$

[3]This assumption can be relaxed where more than one channels can be required to support the content download.

Moreover, the finite storage capacity constraints at different BSs can be written as

$$\sum_{f\in\mathcal{F}}\sum_{i\in\mathcal{M}} x_{kif} \leq C_k, \forall k \in \mathcal{K}. \tag{6.3}$$

We now study the file rejections due to lack of radio resources (i.e., there is no available channel) for a given caching solution $\boldsymbol{x}$. The request rate for file f from MVNO m at BS k if file f is cached at this BS k can be calculated as

$$h_{kmf}\left(\boldsymbol{x}\right) = \lambda_{km} q_{kmf} \left(\sum_{i\in\mathcal{M}} x_{kif}\right). \tag{6.4}$$

Hence, the total request rate from MVNO m for all files in $\mathcal{F}$, if they are cached at BS k, is

$$h_{km}\left(\boldsymbol{x}\right) = \sum_{f\in\mathcal{F}} \lambda_{km} q_{kmf} \left(\sum_{i\in\mathcal{M}} x_{kif}\right). \tag{6.5}$$

Otherwise, if file f is not cached at BS k (i.e., cache-missed file), the corresponding request rate from MVNO m to this file at this BS is equal to

$$\bar{h}_{kmf}\left(\boldsymbol{x}\right) = \lambda_{km} q_{kmf} \left(1 - \sum_{i\in\mathcal{M}} x_{kif}\right), \tag{6.6}$$

and the total cache-missed file request rate from MVNO m to all files in $\mathcal{F}$ at BS k is

$$\begin{aligned}\bar{h}_{km}\left(\boldsymbol{x}\right) &= \sum_{f\in\mathcal{F}} \lambda_{km} q_{kmf} \left(1 - \sum_{i\in\mathcal{M}} x_{kif}\right) \\ &= \lambda_{km} - h_{km}\left(\boldsymbol{x}\right).\end{aligned} \tag{6.7}$$

Note that all these involved arrival processes are Poisson processes because splitting or merging Poisson processes creates Poisson processes. Assume that it takes T_{km} (s) for BS k to serve a cache-hit file request from MVNO m. T_{km} represents the download time from the content cache to the UE of MVNO (m, k). With w_{km} channels allocated by the InP to MVNO (m, k) to serve requests of UEs, at most w_{km} file requests from MVNO m can be simultaneously served by its associated BS. The file requests from MVNO m at BS k can be modeled as an $M/D/w_{km}/w_{km}$ queue with Poisson arrivals, deterministic service time, w_{km} servers, and no waiting buffer [27].

Recall that all cache-missed file requests are rejected due to high delay for downloading content from the CN. Additionally, any cache-hit file request from MVNO m at BS k is only rejected if all w_{km} channels are used to service other ongoing w_{km} requests. From [27], the probability that there are w_{km} ongoing cache-hit file requests from MVNO m being served by BS k can be calculated as

$$P_{km}(\boldsymbol{x}, \boldsymbol{w}) = \frac{(h_{km}(\boldsymbol{x})T_{km})^{w_{km}}}{w_{km}!} \left(\sum_{i=0}^{w_{km}} \frac{(h_{km}(\boldsymbol{x})T_{km})^{i}}{i!} \right)^{-1}. \tag{6.8}$$

Consequently, the rejection rate for the cache-hit request from MVNO m at BS k due to channel unavailability can be expressed as

$$\mu_{km}(\boldsymbol{x}, \boldsymbol{w}) = h_{km}(\boldsymbol{x}) P_{km}(\boldsymbol{x}, \boldsymbol{w}). \tag{6.9}$$

From (6.7) and (6.9), the total file request outage probability from MVNO m at BS k can be calculated as

$$\Phi_{km}(\boldsymbol{x}, \boldsymbol{w}) = \frac{\mu_{km}(\boldsymbol{x}, \boldsymbol{w}) + \bar{h}_{km}(\boldsymbol{x})}{\lambda_{km}}. \tag{6.10}$$

To avoid poor QoS and unfair treatment in serving file requests from different MVNOs at different BSs, we consider the joint channel allocation and content caching optimization problem which minimizes the highest outage probability among MVNOs at all BSs while accounting for the file caching redundancy avoidance and other system constraints. This problem can be formulated as follows:

$$\min_{\boldsymbol{x}, \boldsymbol{w}} \quad \max_{k \in \mathcal{K}, m \in \mathcal{M}} \Phi_{km}(\boldsymbol{x}, \boldsymbol{w}) \tag{6.11a}$$

$$\text{s.t.} \quad \sum_{m \in \mathcal{M}} x_{kmf} \leq 1, \forall k \in \mathcal{K}, \forall f \in \mathcal{F} \tag{6.11b}$$

$$\sum_{m \in \mathcal{M}} \sum_{f \in \mathcal{F}} x_{kmf} \leq C_k, \forall k \in \mathcal{K} \tag{6.11c}$$

$$w_{km} \geq W_{km}^{\mathsf{min}}, \forall k \in \mathcal{K}, \forall m \in \mathcal{M} \tag{6.11d}$$

$$\sum_{k \in \mathcal{K}} \sum_{m \in \mathcal{M}} w_{km} \leq W^{\mathsf{max}} \tag{6.11e}$$

$$x_{kmf} \in \{0, 1\} \ \forall k \in \mathcal{K}, \forall m \in \mathcal{M}, \forall f \in \mathcal{F}, \tag{6.11f}$$

where (6.11b) and (6.11c) capture the file redundancy avoidance and storage capacity constraints, respectively; (6.11d) represents the service-level-agreement (SLA) constraints for MVNO m at BS

k, which guarantees certain minimum number of allocated channels for each MVNO; (6.11e) denotes the bandwidth constraint; and (6.11f) denotes the integer caching decision variables at BSs.

6.5 Proposed Algorithms

In problem (6.11), because $\boldsymbol{x}$ and $\boldsymbol{w}$ are vectors of integer optimization variables and the elements of $\boldsymbol{w}$ are in the exponent and factorial parts of $P_{km}(\boldsymbol{x}, \boldsymbol{w})$ in (6.8), this problem is a MINLP, which is very difficult to solve optimally. Therefore, we propose a two-step iterative algorithm to tackle (6.11). In iteration i, we propose Algorithm 6.1 which is used to find the optimal channel allocation ${\boldsymbol{w}^*}_{(i)}$ based on the caching solution ${\boldsymbol{x}^*}_{(i-1)}$ obtained in the previous iteration $(i-1)$. Then, the proposed bisection-search based Algorithm 6.3 is used to determine the caching decision solution ${\boldsymbol{x}^*}_{(i)}$ based on the newly obtained value ${\boldsymbol{w}^*}_{(i)}$. With the newly obtained ${\boldsymbol{w}^*}_{(i)}$ and ${\boldsymbol{x}^*}_{(i)}$, we compute the maximum request outage probability $\varphi_{(i)}$ for iteration i. The overall procedure can be illustrated as

$$\underbrace{\boldsymbol{x}^*_{(\mathbf{0})} \rightarrow \boldsymbol{w}^*_{(\mathbf{0})}}_{\text{Initialization, } \varphi_{(0)}} \rightarrow \ldots \rightarrow \underbrace{\boldsymbol{x}^*_{(\boldsymbol{i})} \rightarrow \boldsymbol{w}^*_{(\boldsymbol{i})}}_{\text{Iteration } i, \varphi_{(0)}} \rightarrow \ldots \rightarrow \underbrace{\boldsymbol{x}^* \rightarrow \boldsymbol{w}^*}_{\text{Optimal } \varphi^*},$$

where the stopping condition is $|\varphi_{(i)} - \varphi_{(i-1)}| < \varepsilon$ with $0 < \varepsilon << 1$. Finally, we propose a heuristic fast algorithm (Section 6.5.3) for joint resource allocation and content caching based on the properties studied from the bisection search based algorithm.

6.5.1 Channel Allocation for a Given Caching Policy

In this subsection, we propose an algorithm which allocates the optimal number of channels to MVNOs at each BS to minimize the maximum request rejection rate in the network for a given caching solution (i.e., for given $\boldsymbol{x}^*$). First, we characterize the properties of $\Phi_{km}(\boldsymbol{x}^*, \boldsymbol{w})$ in the following Proposition 6.1.

Proposition 6.1. *(i) For a given $\boldsymbol{x}^*$, $P_{km}(\boldsymbol{x}^*, \boldsymbol{w})$ in (6.8) is a decreasing function of $\boldsymbol{w}$. (ii) For a given $\boldsymbol{w}^*$, $P_{km}(\boldsymbol{x}, \boldsymbol{w}^*)$ is an increasing function of $\boldsymbol{x}$.*

Algorithm 6.1. Channel Allocation for a Given Caching Solution

1: **allocate** W_{km}^{min} channels to MVNO m at BS k to satisfy (6.11d).
2: **calculate** $W^{\mathsf{free}} = W^{\mathsf{max}} - \sum_{k\in\mathcal{K}}\sum_{m\in\mathcal{M}} W_{km}^{\mathsf{min}}$
3: **while** $W^{\mathsf{free}} > 0$ **do**
4: **find** $(k^*, m^*) = \underset{k,m}{\operatorname{argmax}}\, \Phi_{km}(\boldsymbol{x}, \boldsymbol{w})$
5: $w_{k^*m^*} = w_{k^*m^*} + 1$
6: $W^{\mathsf{free}} = W^{\mathsf{free}} - 1$
7: **end while**
8: **obtain** optimal $\boldsymbol{w}^*$

Proof. Please refer to [109] for the proof of (i). Also from [109], $P_{km}(h_{km}(\boldsymbol{x})T_{km}, \boldsymbol{w})$ increases with $h_{km}(\boldsymbol{x})T_{km}$, where $T_{km} > 0, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}$. Further, $h_{km}(\boldsymbol{x})$ in (6.5) is an increasing function of $\boldsymbol{x}$. Hence, $P_{km}(\boldsymbol{x}, \boldsymbol{w})$ increases with $\boldsymbol{x}$. □

Proposition 6.1 suggests that to minimize $\Phi_{km}(\boldsymbol{x}^*, \boldsymbol{w})$, we need to make w_{km} as large as possible (i.e., allocating the largest possible number of channels). These results are leveraged to develop our channel allocation algorithm, which is described in Algorithm 6.1. Specifically, we initially attempt to satisfy all SLA bandwidth constraints by allocating W_{km}^{min} channels to MVNO m at BS k. Then, we sequentially allocate one available channel at each iteration to the MVNO m at BS k, whose $\Phi_{km}(\boldsymbol{x}, \boldsymbol{w})$ is highest at each allocation step, until all channels are used up. Lemma 6.1 stated in the following confirms the optimality of Algorithm 6.1.

Lemma 6.1. *For a given caching strategy $\boldsymbol{x}^*$, Algorithm 6.1 optimally allocates channels to individual MVNOs at all BSs to minimize the largest request outage probability in the network.*

Proof. For a given $\boldsymbol{x}^*$, denote $(k^*, m^*) = \underset{k,m}{\operatorname{argmax}}\, \Phi_{km}(\boldsymbol{x}^*, \boldsymbol{w})$ as the MVNO having current largest outage probability φ, i.e., $\varphi = \max_{k,m} \Phi_{km}(\boldsymbol{x}^*, \boldsymbol{w})$. Suppose we allocate a channel for an arbitrary MVNO (k, m) different from MVNO (k^*, m^*). As mentioned above, $\Phi_{km}(\boldsymbol{x}^*, \boldsymbol{w})$ decreases as w_{km} increases. Thus we have

$$\Phi_{km}(\boldsymbol{x}^*, w_{km} + 1) < \Phi_{km}(\boldsymbol{x}^*), w_{km}) < \varphi$$

is true for all MVNOs different from MVNO (k^*, m^*). Obviously, this strategy does not reduce φ. Therefore, allocating one available channel to the MVNO having current largest request outage probability in each step of Algorithm 6.1 is the optimal strategy.

□

6.5.2 Caching Strategy for a Given Channel Allocation Solution

We now optimize the caching decision variables $\boldsymbol{x}$ at all BSs to minimize the maximum request outage probability among MVNOs at all BSs, given the channel allocation solution $\boldsymbol{w}^*$ obtained from Algorithm 6.1. Specifically, problem (6.11) becomes

$$\min_{\boldsymbol{x}} \quad \max_{k\in\mathcal{K},m\in\mathcal{M}} \Phi_{km}\left(\boldsymbol{x},\boldsymbol{w}^*\right) \tag{6.12a}$$

$$\text{s.t.} \quad \sum_{m\in\mathcal{M}} x_{kmf} \leq 1, \forall k \in \mathcal{K}, \forall f \in \mathcal{F} \tag{6.12b}$$

$$\sum_{m\in\mathcal{M}}\sum_{f\in\mathcal{F}} x_{kmf} \leq C_k, \forall k \in \mathcal{K} \tag{6.12c}$$

$$x_{kmf} \in \{0,1\}\ \forall k \in \mathcal{K}, \forall m \in \mathcal{M}, \forall f \in \mathcal{F}. \tag{6.12d}$$

Note that problem (6.12) is still an MINLP because $\Phi_{km}\left(\boldsymbol{x},\boldsymbol{w}^*\right)$ is a non-linear function of integer variable vector $\boldsymbol{x}$. Thus, we propose to tackle this problem indirectly by exploiting the following properties of $\Phi_{km}\left(\boldsymbol{x},\boldsymbol{w}^*\right)$. First, using the results in (6.7), (6.8), and (6.9), we can rewrite (6.10) as follows:

$$\begin{aligned}\Phi_{km}\left(\boldsymbol{x},\boldsymbol{w}^*\right) &= \frac{\mu_{km}\left(\boldsymbol{x},\boldsymbol{w}^*\right)+\bar{h}_{km}\left(\boldsymbol{x}\right)}{\lambda_{km}} \\ &= \frac{h_{km}(\boldsymbol{x})P_{km}\left(h_{km}(\boldsymbol{x}),\boldsymbol{w}^*\right)+\lambda_{km}-h_{km}(\boldsymbol{x})}{\lambda_{km}} \\ &= \Phi_{km}\left(h_{km}(\boldsymbol{x}),\boldsymbol{w}^*\right).\end{aligned} \tag{6.13}$$

Hence, $\Phi_{km}\left(\boldsymbol{x},\boldsymbol{w}^*\right)$ can be considered as a function of $h_{km}(\boldsymbol{x})$ for a given $\boldsymbol{w}^*$, i.e., $\Phi_{km}\left(h_{km}(\boldsymbol{x}),\boldsymbol{w}^*\right)$. Its properties, especially its convexity, are stated in the following Proposition 6.2.

Proposition 6.2. *$\Phi_{km}\left(h_{km}(\boldsymbol{x}),\boldsymbol{w}^*\right)$ is a convex function of $h_{km}(\boldsymbol{x})$ for a given $\boldsymbol{w}^*$. Moreover, it is a decreasing function of $h_{km}(\boldsymbol{x})$.*

Proof. From [107], the loss rate $h_{km}(\boldsymbol{x})P_{km}\left(h_{km}(\boldsymbol{x}),\boldsymbol{w}^*\right)$ is a convex function of $h_{km}(\boldsymbol{x})$ for a given $\boldsymbol{w}^*$. Also from [107], $\frac{\partial[h_{km}(\boldsymbol{x})P_{km}(h_{km}(\boldsymbol{x}),\boldsymbol{w}^*)]}{\partial h_{km}(\boldsymbol{x})} \in [0,1]$. Therefore, $\frac{\partial\Phi_{km}(h_{km}(\boldsymbol{x}),\boldsymbol{w}^*)}{\partial h_{km}(\boldsymbol{x})} \leq 0$, which means $\Phi_{km}\left(h_{km}(\boldsymbol{x}),\boldsymbol{w}^*\right)$ is decreasing with $h_{km}(\boldsymbol{x})$. □

Consequently, $\max_{k\in\mathcal{K},m\in\mathcal{M}} \Phi_{km}\left(h_{km}\left(\boldsymbol{x}\right),\boldsymbol{w}\right)$ can be considered as the pointwise maximum function over $h_{km}(\boldsymbol{x})$, which is convex [33]. We now transform problem (6.12) to the following convex

optimization problem over $\boldsymbol{h}$, where $\boldsymbol{h} = \{h_{km}(\boldsymbol{x})\}, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}$.

$$\min_{\boldsymbol{h},\varphi} \quad \varphi \tag{6.14a}$$

$$\text{s.t.} \quad \Phi_{km}\left(h_{km}(\boldsymbol{x}), \boldsymbol{w}^*\right) \leq \varphi, \forall k \in \mathcal{K}, \forall m \in \mathcal{M} \tag{6.14b}$$

$$h_{km}(\boldsymbol{x}) \in \mathcal{H}, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}. \tag{6.14c}$$

In (6.14), $\mathcal{H}$ denotes the set of all feasible values of $h_{km}(\boldsymbol{x})$, which is dependent on the feasible set of $\boldsymbol{x}$ according to the constraints of problem (6.12). Particularly, $\mathcal{H}$ can be determined from the following constraints:

$$h_{km}(\boldsymbol{x}) = \sum_{f \in \mathcal{F}} \lambda_{km} q_{kmf} \left(\sum_{i \in \mathcal{M}} x_{kif} \right), \forall k \in \mathcal{K}, \forall m \in \mathcal{M} \tag{6.15a}$$

$$\sum_{m \in \mathcal{M}} x_{kmf} \leq 1, \forall k \in \mathcal{K}, \forall f \in \mathcal{F} \tag{6.15b}$$

$$\sum_{m \in \mathcal{M}} \sum_{f \in \mathcal{F}} x_{kmf} \leq C_k, \forall k \in \mathcal{K} \tag{6.15c}$$

$$x_{kmf} \in [0,1], \forall k \in \mathcal{K}, \forall m \in \mathcal{M}, \forall f \in \mathcal{F}. \tag{6.15d}$$

Constraints (6.15b) and (6.15c) are originally from (6.12b) and (6.12c), respectively. Meanwhile, (6.15d) is the relaxed version of (6.12d) to achieve the actual upper bound for all h_{km} and continuous value for $\mathcal{H}$.

Obviously Φ_{km} must be in $[0,1]$ for all k and m because it is the probability. Moreover, $\mathcal{H}$ is constrained by (6.15). Based on the results in Proposition 6.2, we can solve problem (6.14) by using the bisection search method [33] to find the optimal value φ^*, i.e., the minimum of maximal request outage probability. From φ^*, we obtain the corresponding optimal solution $h^*_{km} \in \mathcal{H}, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}$ based on (6.13). However, it is difficult to map φ^* back to h^*_{km} for all k and m due to the unknown inverse function of $\Phi_{km}\left(h_{km}(\boldsymbol{x}), \boldsymbol{w}^*\right)$, which is denoted as Φ^{-1}_{km}. Therefore, in the next subsection, we apply Newton's method [33] to find an approximate output h^*_{km} from Φ^{-1}_{km} taking the outage probability φ^* and channel allocation $\boldsymbol{w}^*$ as the inputs.

6.5.2.1 Finding h_{km} from φ

Given the request outage propability φ_{km} and channel allocation w_{km} for MVNO m at BS k, we need to find

$$h_{km} \text{ s.t. } \Phi_{km}\left(h_{km}, w_{km}\right) = \varphi_{km}. \tag{6.16}$$

Without loss of generality and for the sake of simplicity, we omit the subscripts of $\Phi_{km}\left(h_{km}, w_{km}\right)$ and the input w_{km}. Thus, (6.16) can be re-written as

$$h : \Phi\left(h\right) = \varphi. \tag{6.17}$$

According to Newton's search method, with a properly initial guess h_0, we can find a better approximation h_1 for (6.17) by

$$h_1 = h_0 - \frac{\Phi\left(h_0\right) - \varphi}{\nabla_h \Phi(h_0)}, \tag{6.18}$$

where

$$\nabla_h \Phi(h) \triangleq \frac{\partial \Phi}{\partial h} = \frac{P + \left(w - hT + hTP\right)P - 1}{\lambda}, \tag{6.19}$$

and $P \triangleq P_{km}\left(h_{km}, w\right)$, $w \triangleq w_{km}$, and $T \triangleq T_{km}$ for a particular MVNO (m, k) under consideration in (6.8). We perform the iterative update (6.18) until convergence to achieve a stable approximation of h.

Now, we present an approach to determine a feasible initial value h_0. From (6.7) and (6.10), we have

$$\varphi = \frac{hP\left(a, w\right) - h + \lambda}{\lambda} = \frac{\lambda - \frac{a(1-P(a,w))}{T}}{\lambda} = \frac{\lambda - \frac{L}{T}}{\lambda}, \tag{6.20}$$

where

$$a \triangleq hT \tag{6.21}$$

$$L \triangleq a\left(1 - P(a, w)\right). \tag{6.22}$$

Hence, we have $L = T\lambda\left(1 - \varphi\right)$. According to equation (53) in [110], we have the inequality

$$a_0 < L\left(1 + \frac{L}{w\left(w - L\right)}\right). \tag{6.23}$$

Algorithm 6.2. FINDING CACHE-HIT RATE h FROM GIVEN OUTAGE PROBABILITYφ

1: **calculate** $L = a\left(1 - P\mathsf{A}(a, w)\right)$.
2: **initialize** h_0 according to (6.24).
3: **update** h_1 according to (6.18).
4: **repeat** step (3) **until** convergence with small error ε.

Thus, we can choose the initial value h_0 as follows:

$$h_0 = \frac{a_0}{T} = \frac{L\left(1 + \frac{L}{w(w-L)}\right)}{T}. \tag{6.24}$$

The Newton's search method for calculating the hit rate h given the request outage probability φ and the parameters w, T and λ is summarized in Algorithm 6.2.

We state the convergence property and the solution uniqueness of Algorithm 6.2 in the following proposition.

Proposition 6.3. *Algorithm 6.2 converges to a unique value of h.*

Proof. Due to Proposition 6.2, Φ represents the *one-to-one mapping* function between its output φ and the input cache hit rate h. Therefore, Algorithm 6.2 returns a *unique value* of h for (6.16). □

Recall that $\Phi_{km}\left(h_{km}\left(\boldsymbol{x}\right), \boldsymbol{w}^*\right)$ is a decreasing function of $h_{km}\left(\boldsymbol{x}\right)$ for all $h_{km} \in \mathcal{H}$ according to Proposition 6.2. Therefore, given a request outage probability value φ that constraint (6.14a) is satisfied, the cache hit rate $h_{km}(\boldsymbol{x})$ must satisfy the following one-to-one relationship

$$\Phi_{km}\left(h_{km}(\boldsymbol{x}), \boldsymbol{w}^*\right) \leq \varphi \iff h_{km}(\boldsymbol{x}) \geq h, \tag{6.25}$$

where $\Phi_{km}\left(h, \boldsymbol{w}^*\right) = \varphi$. However, all $h_{km}(\boldsymbol{x})$'s are constrained by $\mathcal{H}$ as in (6.15), we need to verify the feasibility of $\boldsymbol{x}$. This motivates us to study the caching strategy in two different relevant scenarios in which the popularity orders of different files at different BS are the *same* and *different*, in the following two subsections.

6.5.2.2 Same-Order File Popularity Case

Suppose that we rank the content request probabilities for different files f of each MVNO (m, k) in the descending order of their values. In the same-order popularity case, all MVNOs at each BS

k have the same order of file indices under this ranking (i.e., the most popular file, second most popular file,... of all MVNOs at each BS k are the same). Suppose that MVNO (m^*, k^*) is the one having largest request outage probability for a given channel allocation solution $\boldsymbol{w}^*$, i.e.,

$$(k^*, m^*) = \underset{k,m}{\operatorname{argmax}} \, \Phi_{k^*m^*} \left(h_{k^*m^*} \left(\boldsymbol{x} \right), \boldsymbol{w}^* \right). \tag{6.26}$$

Given the channel allocation solution $\boldsymbol{w}^*$, $\Phi_{k^*m^*} \left(h_{k^*m^*} \left(\boldsymbol{x} \right), \boldsymbol{w}^* \right)$ decreases as $h_{k^*m^*} \left(\boldsymbol{x} \right)$ increases according to Proposition 6.2. From (6.5), $h_{k^*m^*} \left(\boldsymbol{x} \right)$ is a non-negative linear combination of caching decision vector $\boldsymbol{x}$. Consequently, increasing $h_{km} \left(\boldsymbol{x}_{km} \right)$ by caching more content preferred by MVNO (m^*, k^*) is the best strategy for reducing its request outage probability. Moreover, as file popularity ranks of different files for all $m \in \mathcal{M}$ are identical at each BS k, the *most popular caching (MPC)* is the best strategy for each BS for all MVNOs with the same-order file popularity to reduce the outage probability. In particular, this MPC strategy results in the optimal caching solution for the following problem considering the cache redundancy avoidance constraint:

$$\max_{\boldsymbol{x}_{k^*m}} \quad \sum_{m \in \mathcal{M}} h_{k^*m} \left(\boldsymbol{x}_{k^*m} \right) \tag{6.27a}$$

$$\text{s.t.} \quad \sum_{m \in \mathcal{M}} x_{k^*mf} \leq 1 \forall f \in \mathcal{F} \tag{6.27b}$$

$$\sum_{m \in \mathcal{M}} \sum_{f \in \mathcal{F}} x_{k^*mf} \leq C_{k^*} \tag{6.27c}$$

$$x_{k^*mf} \in [0, 1] \, \forall m \in \mathcal{M}, \forall f \in \mathcal{F}. \tag{6.27d}$$

Note that as each BS has its own storage repository, the caching decisions at different BSs are independent. This means that only $\boldsymbol{x}_{k^*m}$ for all $m \in \mathcal{M}$ is affected by the problem (6.27). Hence, we can apply (6.27) for all BSs having the same-order file popularity over all MVNOs. Due to Proposition 6.2 and the non-negative linear combination of $\boldsymbol{x}$ in (6.5), the solution of (6.27) will reduce the request outage probability for all MVNOs at the considered BSs.

6.5.2.3 Different-Order File Popularity Case

In this case, the popularity ranks of different files for different MVNOs at each BS k can be different (e.g., the most popular file for MVNO 1 can be different from the most popular file for MVNO 2). Hence, maximizing $h_{k^*m^*} \left(\boldsymbol{x} \right)$ may cause the decrement of $h_{k^*m} \left(\boldsymbol{x} \right)$ for some $m \in \mathcal{M} / \{m^*\}$. In fact,

caching more content preferred by MVNO (m^*, k^*) may evict other MVNOs' most favorite content due to the storage capacity limitation. This results in the increment of $\Phi_{k^*m}\left(h_{k^*m}\left(\boldsymbol{x}\right), \boldsymbol{w}^*\right)$ for some $m \in \mathcal{M}/\{m^*\}$ according to Proposition 6.2. Denote the pre-caching-decision largest request outage probability by φ. IF the post-caching-decision request outage probability Φ_{k^*m} for some $m \neq m^*$ exceeds φ, then the caching decision problem by (6.27) fails to obtain better outage probability, i.e., $\varphi^{\mathsf{new}} < \varphi$. Therefore, the MPC strategy in (6.27) may not be the best strategy in the case of different-order file popularity. To this end, we need to guarantee that optimizing $\boldsymbol{x}_{k^*m^*}$ for MVNO (k^*, m^*) does not make post-caching-decision Φ_{k^*m} exceed φ for all $m \in \mathcal{M}/\{m^*\}$, which is mathematically imposed by

$$\Phi_{km}\left(h_{km}\left(\boldsymbol{x}\right), \boldsymbol{w}^*\right) \leq \varphi, \forall m \in \mathcal{M}, \forall k \in \mathcal{K}. \tag{6.28}$$

Due to Proposition 6.2, constraint (6.28) is equivalent to

$$h_{km}\left(\boldsymbol{x}\right) \geq h_{km}^{\mathsf{low}}, \forall m \in \mathcal{M}, \forall k \in \mathcal{K}, \tag{6.29}$$

where

$$\Phi_{km}\left(h_{km}^{\mathsf{low}}, \boldsymbol{w}^*\right) = \varphi, \forall m \in \mathcal{M}, \forall k \in \mathcal{K}. \tag{6.30}$$

In other words, h_{km}^{low} is the output of the inverse function Φ_{km}^{-1} taking φ as the input. We will use Algorithm 6.2 to find h_{km}^{low}. With constraint (6.29), the caching decision problem (6.27) becomes (6.31) and we state its properties in Proposition 6.4.

$$\max_{\boldsymbol{x}_{km}} \quad \sum_{m \in \mathcal{M}} h_{km}\left(\boldsymbol{x}_{km}\right) \tag{6.31a}$$

$$\text{s.t.} \quad h_{km}\left(\boldsymbol{x}\right) \geq h_{km}^{\mathsf{low}}, \forall m \in \mathcal{M} \tag{6.31b}$$

$$\sum_{m \in \mathcal{M}} x_{k^*mf} \leq 1 \forall f \in \mathcal{F} \tag{6.31c}$$

$$\sum_{m \in \mathcal{M}} \sum_{f \in \mathcal{F}} x_{kmf} \leq C_k \tag{6.31d}$$

$$x_{kmf} \in [0, 1] \, \forall m \in \mathcal{M}, \forall f \in \mathcal{F}. \tag{6.31e}$$

Proposition 6.4. *Problem* (6.32) *covers problem* (6.27) *in both cases of same-order and different-order file popularity.*

Proof. First, if we remove constraint (6.31b) from (6.31), then it is equivalent to (6.27). Hence, the feasible solution set of (6.31) is a subset of the feasible solution set of (6.27). Second, in the case of same-order file popularity at BS k, the constraint (6.31b) is always satisfied. In fact, all the cache hit rate $h_{km}(\boldsymbol{x})$ for all m will increase by caching any file due to the identical set of file popularity $\mathcal{Q}_{km}$ for all m, i.e., $h_{km}(\boldsymbol{x}^*) \geq h_{km}^{\mathsf{low}}, \forall m \in \mathcal{M}$. □

With Proposition 6.4, solving (6.31) is sufficient for obtaining caching decisions for the both cases of file popularity at each BS. Recall that each BS is equipped with an independent storage repository. Therefore, by taking summation over $\mathcal{K}$ (all BSs) in the objective function of (6.31), we obtain the caching decision optimization problem for all BSs. Mathematically, this caching problem can be stated as

$$\max_{\boldsymbol{x}} \quad \sum_{k\in\mathcal{K}} \sum_{m\in\mathcal{M}} h_{km}(\boldsymbol{x}) \tag{6.32a}$$

$$\text{s.t.} \quad h_{km}(\boldsymbol{x}) \geq h_{km}^{\mathsf{low}}, \forall m \in \mathcal{M}, \forall k \in \mathcal{K} \tag{6.32b}$$

$$\sum_{m\in\mathcal{M}} x_{kmf} \leq 1, \forall k \in \mathcal{K}, \forall f \in \mathcal{F} \tag{6.32c}$$

$$\sum_{m\in\mathcal{M}} \sum_{f\in\mathcal{F}} x_{kmf} \leq C_k, \forall k \in \mathcal{K} \tag{6.32d}$$

$$x_{kmf} \in [0,1] \, \forall m \in \mathcal{M}, \forall f \in \mathcal{F}. \tag{6.32e}$$

We can solve this linear programming problem by using any available solver such as CVX [111]. Finally, Algorithm 6.3 summarizes our iterative joint channel allocation (Algorithm 6.1) and content caching placement strategy in (6.32).

6.5.2.4 Rounding Caching Decision Variables

After obtaining the result from Algorithm 6.3, we need to round the caching decision variables due to the underlying constraint relaxation. For the same-order file popularity case, the optimal solution $\boldsymbol{x}^*$ of (6.32) is actually an integral vector as this is the result of the most popular caching policy. For different-order file popularity case, $\boldsymbol{x}^*$ is a real-value vector in $[0,1]^{K\times M\times F}$ due to the relaxed constraint (6.32e). To efficiently round $\boldsymbol{x}^*$ to a binary vector, we propose an algorithm which is based on the modified version of problem (6.32). Specifically, we first loosen the constraint

Algorithm 6.3. Iterative Channel Allocation and Content Caching Placement

set $i = 1$ and tolerance $\varepsilon > 0$.
initialize $\boldsymbol{x}^*_{(i)}$ according to most popular caching strategy with equal storage partition.
initialize channel allocation $\mathbf{w}_{(i)}$ using Algorithm 6.1 given $\boldsymbol{x}^*_{(i)}$.
calculate $\Phi_{km}, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}$.
find largest outage probability $\varphi_{(i)} = \max_{k,m} \Phi_{km}$.
set $\Delta_\varphi = 1$
while $\Delta_\varphi > \varepsilon$ **do**
  $i = i + 1$
  set $\phi^{\mathsf{up}} = 1$
  set $\phi^{\mathsf{low}} = 0$
  while $\phi^{\mathsf{up}} - \phi^{\mathsf{low}} > \varepsilon$ **do**
    $\phi_{(i)} = \left(\phi^{\mathsf{up}} + \phi^{\mathsf{low}}\right)/2$
    find h^{low}_{km} from $\phi_{(i)}$ by using Algorithm 6.2.
    solve problem (6.32) to find $\boldsymbol{x}^*$.
    if $\boldsymbol{x}^*$ is feasible **then**
      $\phi^{\mathsf{up}} = \phi_{(i)}$
      $\boldsymbol{x}^*_{(i)} = \boldsymbol{x}^*$
    else
      $\phi^{\mathsf{low}} = \phi_{(i)}$
    end if
  end while
  find optimal $\boldsymbol{w}^*_{(i)}$ by using Algorithm 6.1.
  calculate $\Phi^{(i)}_{km}, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}$.
  find $\varphi_{(i)} = \max_{k,m} \Phi^{(i)}_{km}$
  calculate $\Delta_\varphi = |\varphi^*_{(i)} - \varphi^*_{(i-1)}|$
end while
obtain final $\boldsymbol{w}^*$ and $\boldsymbol{x}^*$ from Algorithm 6.1 given $\boldsymbol{x}^*_{(i)}$.

Algorithm 6.4. Rounding Caching Decision Variables

initialize small $\varepsilon > 0$
obtain the optimal request outage probability value φ from Algorithm 6.3.
repeat
  obtain $h^{\mathsf{low}}_{km}, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}$ with Algorithm 6.2.
  solve problem (6.32) with integral constraint.
  if integral solution $\boldsymbol{x}^*_{\mathsf{INT}}$ is not found **then**
    $\varphi = \varphi + \varepsilon$
  end if
until integral solution $\boldsymbol{x}^*_{\mathsf{INT}}$ is found.

(6.32b) by adding a positively small ε to the optimal φ obtained from the original problem (6.32), and thus obtaining the corresponding h^{low}_{km} for all MVNO (m, k). Next, we transform (6.32e) to the corresponding integral constraint. Finally, we solve the modified version of (6.32) using a solver supporting integer linear programming (ILP) such as Gurobi [112]. We continue loosening the constraint (6.32b) by adding ε to current φ, then solve the modified (6.32) iteratively until obtaining a feasible integral solution $\boldsymbol{x}^*_{\mathsf{INT}}$. This procedure is summarized in the Algorithm 6.4.

6.5.2.5 Convergence and Complexity Analysis

First, the convergence of Algorithm 6.3 is stated in Lemma 6.2

Lemma 6.2. *Algorithm 6.3 converges to a local optimum point of channel allocation and caching decision.*

Proof. The bisection search based Algorithm 6.3 consists of two main steps: channel allocation (Algorithm 6.1) and relaxed caching decision (problem (6.32)) through the Newton's search method (Algorithm 6.2). In each step, a single type of variables is optimized while the remaining variables remain the same as in the previous iteration. Algorithm 6.1 achieves the optimal channel allocation solution $\boldsymbol{w}^*$ in each iteration according to Lemma 6.1. Meanwhile, Algorithm 6.2 returns a unique h_{km} given the outage probability φ due to the one-to-one mapping property induced by Proposition 6.2. This results in a unique caching decision for problem (6.32). Consequently, Algorithm 6.3 implements the block coordinate descent (BCD) search, whose convergence is guaranteed [113]. However, we can only guarantee that Algorithm 6.3 converges to a local optimal. This is because the caching decision problem (6.32) is solved through approximation of h_{km} for all k and m.

□

Regarding the complexity, the inner while-loop of Algorithm 6.3 is upper-bounded by $\lceil \log_2 \left(\frac{\Phi^{\mathsf{up}} - \Phi^{\mathsf{low}}}{\varepsilon} \right) \rceil$ search iterations. Algorithm 6.1 has $\mathcal{O}(KMW)$ complexity, where K, M and W are the total numbers of BSs, MVNOs and wireless channels, respectively. The Newton's search method in Algorithm 6.2 converges within tens of iterations, where each iteration has complexity of $\mathcal{O}(KM)$. Solving linear programming problem (6.32) involves polynomial time complexity. Algorithm 6.1 thus has polynomial time complexity. Its running time is affordable for the underlying resource provisioning optimization as it is only repeated once over a long time period, e.g., hours or days.

6.5.3 Proposed Heuristic Algorithm

For performance evaluation, we now present another heuristic algorithm. In this fast algorithm, we first equally split the storage repository into M partitions at each BS, each partition is then assigned to one MVNO for fairness. However, we allow the contents cached on these partitions to be shared

among all MVNOs co-located at each BS. Moreover, the contents are cached in an iterative manner as described in the following. In each iteration and considered storage partition corresponding to a particular MVNO, we cache the MVNO's most favorite content which still not exists in any storage partitions. The procedure is repeated until all the partition capacity is fully cached. Feeding the newly derived caching solution to Algorithm 6.1, we finally obtain the channel allocation solution.

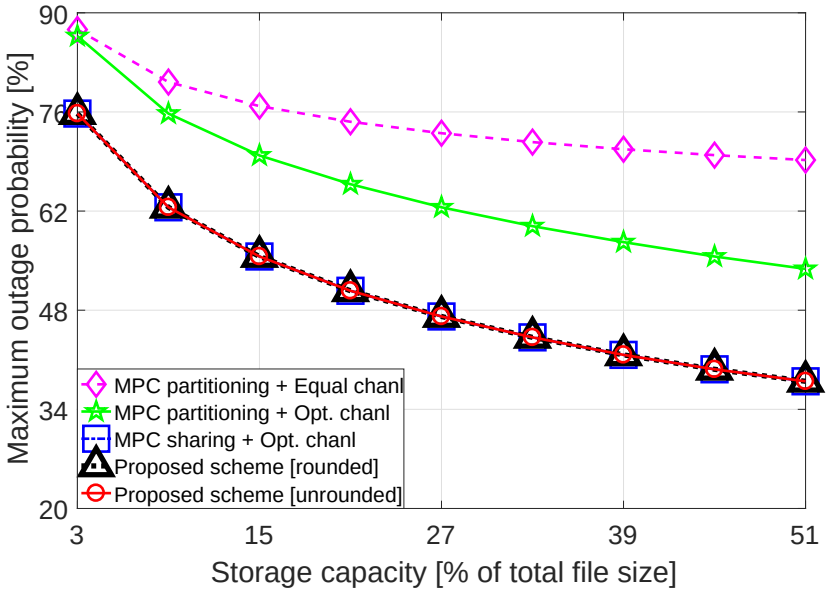
It can be verified that this algorithm requires about $K \times M \times C_k$ steps for caching files in each BS, thus resulting in the total complexity of $\mathcal{O}(KMC_k)$. Note that this is much faster than the proposed bisection-search algorithm for caching decision. Moreover, Algorithm 6.1 has $\mathcal{O}(KMW)$ complexity. Hence, the proposed heuristic algorithm has overall complexity of $\mathcal{O}(KM(W + C_k))$.

6.6 Numerical Results

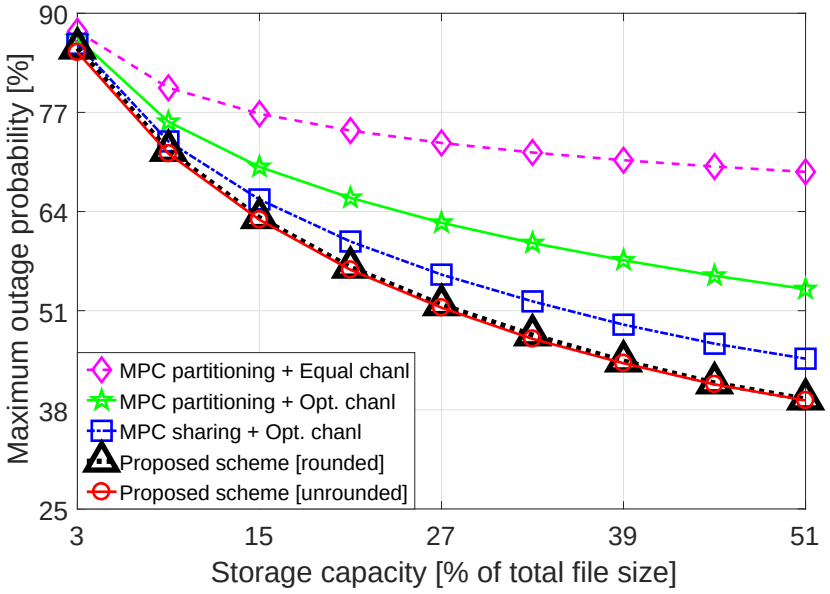
In this section, we evaluate the performance of our proposed algorithms through computer simulation under the following setting. We consider the network with 5 BSs serving 3 MVNOs, which access a list of 100 files, i.e., $K = 5, M = 3$ and $F = 100$. The average request rates for each MVNO are randomly chosen in the range of $[1, 15]$, which results in the total of request rates from tens to hundreds requests arriving to the considered network in one second. We assume that wireless channels are allocated and accessed in the orthogonal manner to avoid strong interference. File requests (i.e., content popularity) are assumed to follow the Zipf distribution where the probability of requesting the file having rank $f \in \{1, ..., F\}$ is given by

$$Z(f, \gamma) = \frac{f^{-\gamma}}{\sum_{n=1}^{F} n^{-\gamma}}, \tag{6.33}$$

where F is the total number of files in the list and γ is the Zipf parameter. This parameter is chosen as $\gamma \in [0.3, 1.2]$ in the simulation setting, and MVNOs co-located at the same BS are assumed to have the same γ value. However, the rank of each file in the list is set randomly with respect to each MVNO and BS. The obtained numerical results are averaged over 100 realizations of the file ranking in the list. The obtained results, therefore, represent the average performance of the proposed algorithms over different realizations of file popularity including same-order and different-order cases. We assume the same service time with $T_{km} = 1$ for all MVNO (m, k).



(a) Same-order file popularity



(b) Different-order file popularity

Figure 6.2 – Maximum outage probability vs storage capacity

We assume that all BSs share $W^{\max} = 90$ wireless channels in the orthogonal manner to serve file requests from MVNOs. Each SLA requirement is set with $W_{km}^{\min} = 2, \forall k \in \mathcal{K}, \forall m \in \mathcal{M}$. All the BSs have equal storage capacity which varies from 3% to 51% of the total file size. Note that the storage capacity can be much smaller than the total size of all files in the list in practice [14]. Finally, we set $\varepsilon = 10^{-4}$ in the stopping condition for all experiments.

For performance evaluation, we compare our proposed algorithms with baseline algorithms based on the most popular caching (MPC) strategy and different channel allocation strategies. The the following baseline algorithms and the proposed heuristic algorithm are considered in the performance evaluation.

- **MPC partitioning + Equal Chanl:** The storage repository and the channel budget are equally divided to all MVNOs at each BS. The MPC strategy is independently applied for each MVNO. The file sharing among MVNOs at the same BS is disabled.
- **MPC partitioning + Opt Chanl:** This setting is similar to the one above for storage repository partitioning. However, the channels are allocated to MVNOs following the proposed channel allocation in Algorithm 6.1.
- **MPC sharing + Opt Chanl:** This is our proposed fast heuristic algorithm in Section 6.5.3.

Storage resource strongly impacts the caching performance and thus our proposed algorithms. Figures 6.2a and 6.2b show that the proposed bisection-search based algorithm (Algorithm 6.3) with cache sharing consistently achieves the smallest maximum request outage probability in the cases with same-order and different-order file popularity, respectively. Moreover, the proposed rounding operation for caching decision variables result in negligible performance loss compared to the achieved performance before rounding, which confirms the efficacy of our design (the request outage probability obtained under relaxation from Algorithm 3 is the lower bound of the optimum value). Moreover, the proposed heuristic algorithm achieves performance very close to the proposed bisection-search based algorithm in the different-order popularity case, and both algorithms result in the same solution in the same-order file popularity case.

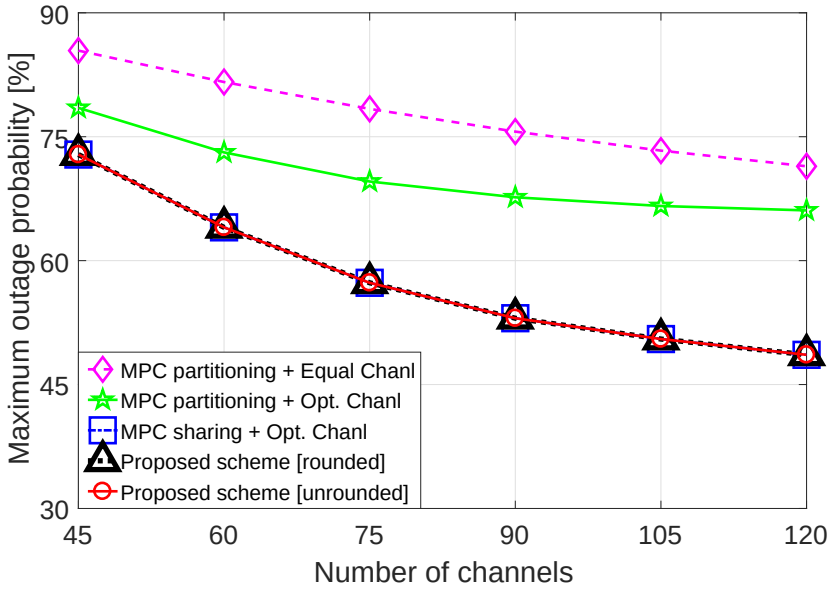
Moreover, these figures also confirm that cache sharing results in significant performance enhancement in the virtualized wireless network serving multiple MVNOs. In fact, sharing the cache allows efficient coordination among different MVNOs to avoid file caching redundancy, thus leaving more storage space to store more files. This in turns leads to reduction of the request outage probability. In contrast, the baselines schemes perform worse than our proposed algorithms, especially at the high storage capacity regime. This is because the larger number of files cached at BS is, the more flexibility is available for file sharing, especially in the case of different-order file popularity. Further, channel allocation with knowledge about the caching solution also helps to improve per-

formance. This is confirmed by the fact that the baseline schemes with optimal channel allocation result in lower maximum request outage probability in comparison with those employing the equal channel allocation.

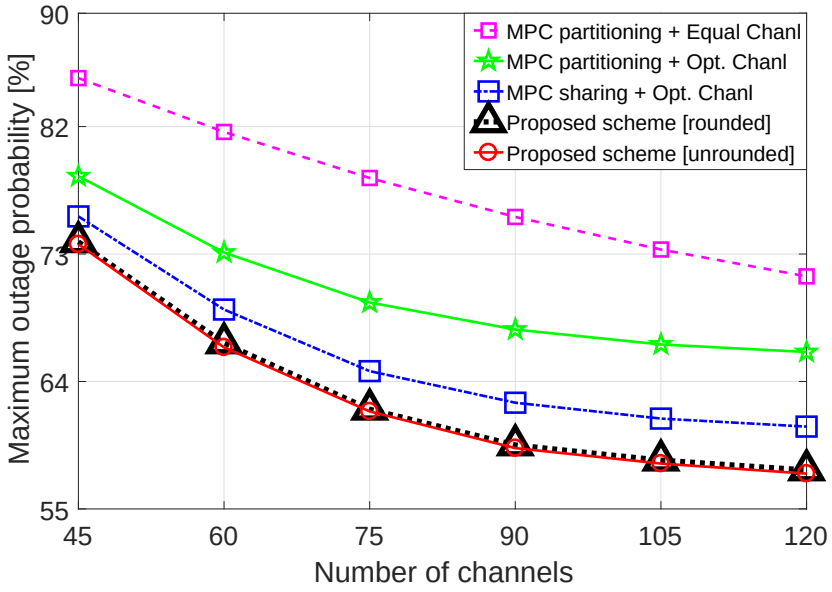
We present the maximum request outage probability among MVNOs at all BSs versus the total number of channels in Figures 6.3a and 6.3b for the cases of same-order and different-order file popularity, respectively. The results are obtained with the storage capacity equal to 15% of the total size of all files. Similar to Figure 6.2, Figure 6.3 confirms the greatest performance of our proposed bisection-search based algorithm as it achieves the lowest request outage probability compared with the remaining baselines. Figures 6.2 and 6.3 imply that instead of partitioning the available storage space to individual MVNOs, it is better to share it among MVNOs co-located at the same BS.

Figure 6.4 shows the maximum request outage probability as a function of the Zipf parameter γ, which is set equal for all MVNOs and BSs. In this experiment, the storage capacity at each BS is set equal to 18% of the total size of all files in the file list. Meanwhile, the numerical results are averaged over four settings with different number of wireless channels which are equal to 60, 90, 120, and 150. The content request rate is generated randomly in the range of $[1, 15]$. Similar to the results in previous figures, our proposed algorithm achieves the best performance in terms of maximum request outage probability. Moreover, the proposed bisection-search based algorithm and heuristic algorithms result in significant reduction of the request outage probability when γ increases in the range $\gamma \in [0.3, 1.2]$, whereas the request outage probability decreases more slowly with γ when $\gamma > 1.2$. This is due to the property of Zipf distribution, in which its cumulative distribution function (CDF) increases faster with the increment of γ, given the same number of files.

Further, the probability mass function (PMF) of the Zipf distribution becomes long-tailed for large values of γ. Hence, caching the same number of files with larger Zipf parameter γ results in the greater hit rate $h_{km}(\boldsymbol{x})$, which in turn greatly decreases the rejection rate $\mu_{km}(\boldsymbol{x}, \boldsymbol{w})$ according to Proposition 6.2. However, when γ enters the large-value regime, the additional contributions to the hit rate by the low-rank files are negligible. Note that MVNOs co-located at the same BS would have very similar file ranking in practice (due to the their similar UEs' file preferences). This explains why our proposed heuristic algorithm, i.e., most popular caching strategy with our



(a) Same-order file popularity

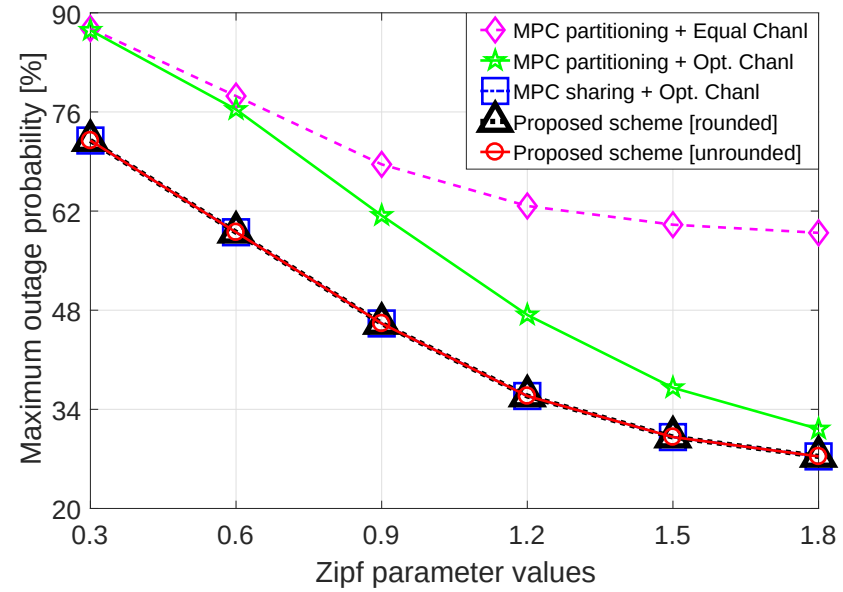


(b) Different-order file popularity

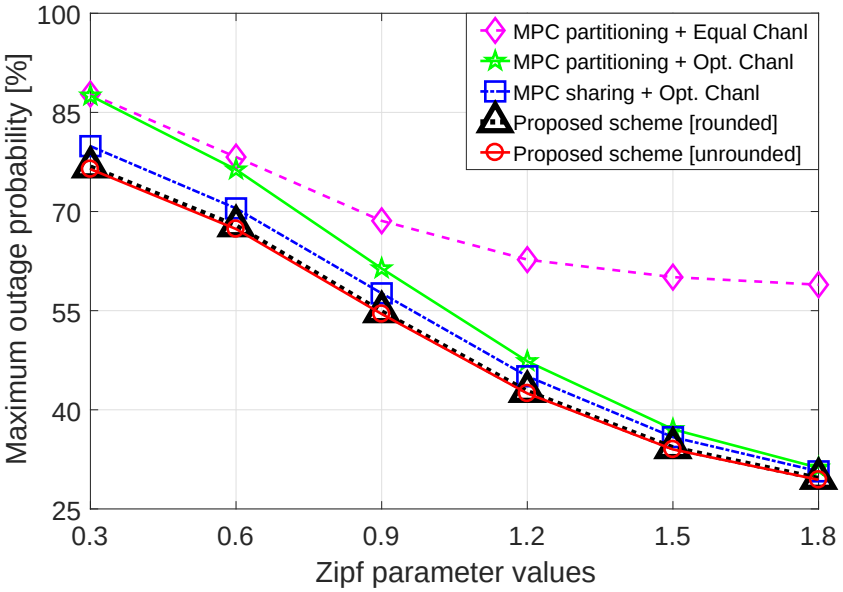
Figure 6.3 – Maximum outage probability vs number of channels

proposed channel allocation algorithm, can achieve very similar performance with our proposed bisection-search based algorithm.

Figure 6.5 shows the maximum request outage probability among MVNOs and BSs as we vary the maximum request rate. Specifically, in this experiment, the average request rate is set in the range $[1, 15 + \Delta]$, where $\Delta \in \{0, 5, 10, 15, 20, 25\}$. The numerical results are averaged over four



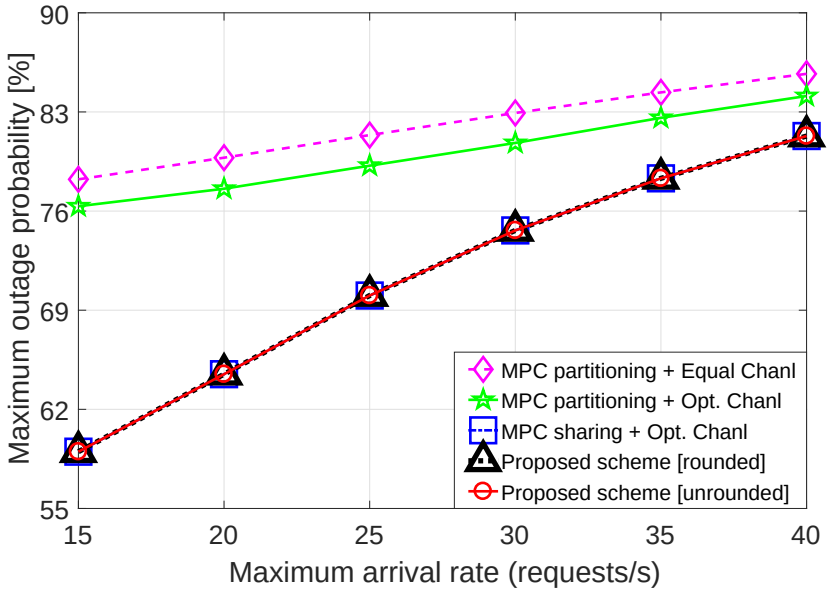
(a) Same-order file popularity



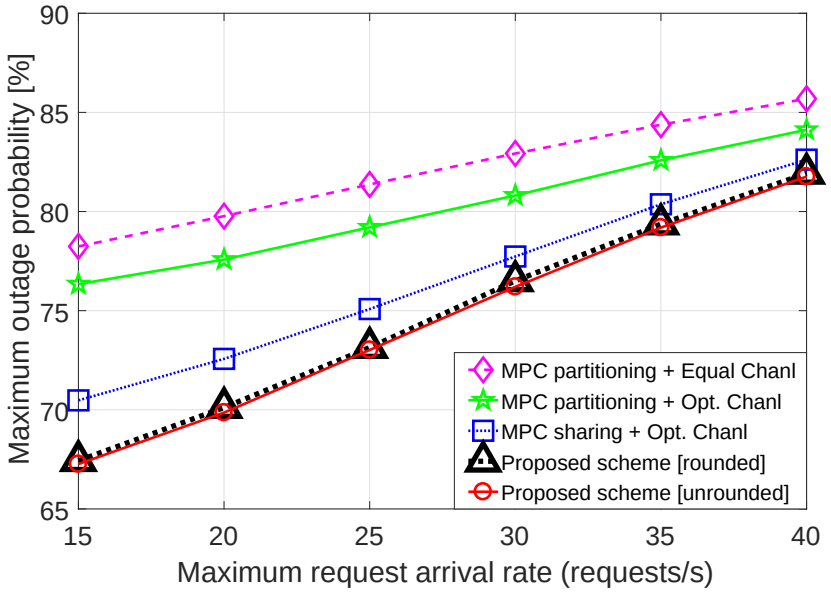
(b) Different-order file popularity

Figure 6.4 – Maximum outage probability vs Zipf parameter

different settings of channel budget with 60, 90, 120 and 150 wireless channels. Meanwhile, we fix the storage capacity at 18% of the total size of all files and the Zipf parameter at $\gamma = 0.6$. As Δ increases, i.e., the maximum file request rate increases, the maximum request outage probabilities of all schemes increase as well. Again, our proposed algorithm significantly outperforms other baselines over all values of Δ.



(a) Same-order file popularity



(b) Different-order file popularity

Figure 6.5 – Maximum outage probability vs maximum request arrival rate

All the presented figures confirm the great performance of our proposed bisection-search based algorithm and the proposed heuristic algorithm (i.e, joint MPC sharing strategy with optimal channel allocation in Algorithm 6.1). In the case of same-order file popularity, Figures 6.2a, 6.3a, 6.4a, and 6.5a show that the proposed bisection-based algorithm and the heuristic algorithm achieve the same solution.

In the case of different-order file popularity, the performance of Algorithm 6.3 with caching-variable rounding is worse than that of the standalone Algorithm 6.3 (without rounding of caching decision variables), yet the performance loss due to cache decision variable rounding is negligible, as shown in Figures 6.2b, 6.3b, 6.4b, and 6.5b. Meanwhile, the proposed heuristic algorithm suffers from less than 5% performance loss in comparison with the standalone Algorithm 6.3. The performance gap between Algorithm 6.3 and the proposed heuristic algorithm is only visible in the regimes of large storage capacity, large channel budget, small Zipf parameter, and small maximum request arrival rate.

6.7 Conclusion

We have studied the joint resource allocation and content caching problem in wireless networks with congested backhaul considering the dynamics of arrival/departure requests and the cache redundancy avoidance. We have proposed a bisection-search based algorithm to tackle the underlying design problem and we have proved that it converges to a local optimal solution in polynomial time. We further proposed a fast heuristic algorithm which attains moderate performance loss in comparison with the bisection-search based algorithm. Numerical results confirms our dominant performance in comparison with baseline schemes in different relevant network settings. Specifically, the results have confirmed that sharing the storage space among different MVNOs can enable to improve the caching performance significantly. Also, joint optimization of content caching and resource allocation is important to achieve the best performance, especially if the content popularity patterns of individual MVNOs at each BS are different.

Chapter 7

Resource Allocation for Multi-Tenant Network Slicing: A Multi-Leader Multi-Follower Stackelberg Game Approach

The content of this chapter was presented in the following paper:

Thinh Duy Tran and Long Bao Le, "Resource allocation for multi-tenant network slicing: A multi-leader multi-follower Stackelberg game approach," submitted to *IEEE Transactions on Vehicular Technology*.

7.1 Abstract

Network slicing provides great opportunities for network SPs to scale up and achieve higher revenue by enabling fast and efficient creation of new services and customizable networks for enterprises and UEs. This can only be achieved at the expense of more complicated radio resource management and service/trading interactions among different involved stakeholders. In this paper, we study the resource allocation and pricing problem for network slicing that captures interactions among

access/backhaul service providers and their UEs by using the MLMF Stackelberg game approach. Toward this end, we show how to formulate such a Stackelberg game and prove the existence of a unique game equilibrium. Then, we develop a distributed algorithm based on updating underlying best-response functions, which is proved to converge to the game equilibrium. Numerical results are presented to provide important insights into the interactions among the involved stakeholders and demonstrate the economical efficacy of the proposed design with respect to existing benchmarks.

7.2 Introduction

The next-generation wireless network must rely on effective resource allocation and network orchestration to cope with the exponential growth of mobile traffic and support emerging applications [7]. Network slicing has been recognized as one of the most important technologies where the monolithic network can be sliced into multiple network slices to support specialized wireless services. Appropriately designed network slices, for instance, could be designated for the high-speed streaming services such as YouTube and Netflix, or the uRLLC services for the factory control applications [22]. Network slicing also provides the paradigm shift toward multi-tenancy in the next-generation wireless network [23] where individual tenants (e.g., MVNOs) own and manage corresponding network slices. By enabling service trading among tenants, also known as SPs, this paradigm shift offers greater business opportunities and greater savings in Capital Expenditure (CAPEX) and Operational Expenditure (OPEX) [22]. The multilateral interactions among SPs and their customers such as UEs, which constitute to a service trading market, can be modeled by using the Stackelberg game theory. Designing an appropriate framework for operating such market is crucial for achieving efficient network serviceability and high profits.

7.2.1 Related Works

The Stackelberg game theory has been employed to study various resource allocation problems in different research fields [44, 45]. This game theory has recently been applied to solve different resource allocation problems in wireless communications. In particular, the authors of [38, 41] leveraged this game theoretic approach to tackle spectrum sharing problems in cognitive radio networks. A tri-level Stackelberg game framework was proposed for the full-duplex wireless network

in [39] where the spectrum provider acts as a single leader in the game. Another Stackelberg game framework was proposed to investigate the mobile offloading market in [43].

Moreover, the Stackelberg game theory has also been applied to study interactions among stakeholders of a network slicing based wireless network. Leveraging this game theory, a price-aware joint power and radio resource allocation framework was proposed for a virtualized wireless network with one InP and multiple MVNOs [42]. Meanwhile, the authors of [88] applied the Stackelberg game theory to address the joint spectrum reuse-aware resource allocation and content caching optimization problem for two different network slice instances. A tri-level Stackelberg game framework was proposed in [40] for resource trading in a virtualized wireless network. Here, each UE can be associated with only one service provider, thus resulting in multiple single-leader Stackelberg games in levels two and three of the underlying game.

A common characteristic of the above Stackelberg game based frameworks is that they assume a single operator such as the InP acts as a sole leader of the game, i.e., single-leader-multiple-follower (SLMF) Stackelberg game theory. However, this assumption does not hold true when resource and service trading in a virtualized wireless network involves multiple SPs in the service domain. For example, various SPs such as Bell, Rogers, and Telus in Canada can provide a similar mobile service for wireless users in a certain area of the country. In fact, SPs who control network slices providing the same service type have to compete with one another because potential customers could choose the most appropriate SPs to receive services. The interactions among these SPs, thus, form an oligopoly market in the service and resource trading. Accordingly, the multi-leader-multi-follower (MLMF) Stackelberg game theory provides a suitable tool for modeling the interactions among these stakeholders.

The MLMF Stackelberg game theory has been employed in some recent works [46–50]. Particularly, the authors of [46] and [47], which were inspired by the work [41], employed the MLMF Stackelberg game to study the spectrum sharing problem in cognitive radio networks. Several multi-leader Stackelberg game frameworks were proposed to model the traffic offloading market in LTE unlicensed bands [48, 49]. Meanwhile, the MLMF Stackelberg game [50] was applied to study the resource trading among multiples InPs, MVNOs, and their wireless users under the network slicing paradigm.

Various game theoretic approaches have been applied to tackle different resource allocation problems in the network slicing context. In particular, the authors of [34] proposed a game theoretic framework for network slicing, built upon the share-constrained proportional resource allocation mechanism, that achieves high efficiency and fairness in allocating resource to network tenants. This game framework was also applied to allocate radio remote heads (RRHs) to MVNOs in [89]. A matching game was employed to tackle the combinatorial resource trading and service selection problem for InPs, MVNOs, and UEs in [35]. Meanwhile, various bidding mechanisms [36, 90, 91] were leveraged to study the resource trading among stakeholders in network slicing. The authors of [36] and [90] applied traditional combinatorial bidding and a generalized Kelly bidding mechanisms for SPs/MVNOs to bid different network resources so as to maximize their utilities, respectively. A new bidding mechanism was recently proposed for joint computation and storage resource trading in network slicing [91]. Contract theory was also applied to model the resource trading between an MVNO and multiple InPs [37].

To the best of our knowledge, studying the multilateral interactions among peer access service providers (ASPs) that provide the same kind of service to a group of UEs in the virtualized wireless network has not been considered. This work aims to fill this gap in the current research literature. Different from the works discussed above, where a UE can only select one SP for purchasing service (i.e., single-source service selection), we relax this assumption to have a more general service model. In our work, we allow any UE and ASP to be able to lease services from different ASPs and backhaul service providers (BSPs) at the same time, respectively, thereby enabling the multiple slice connectivity [67][1]. Accordingly, the multilateral interactions among the stakeholders considered our network slicing model become more general but challenging to study.

7.2.2 Research Contributions

In this paper, we consider the interactions among the SPs and their UEs in the network slicing-based wireless network with wireless backhauling. In fact, wireless backhauling enables rapid and flexible network network deployment to support emerging 5G use cases such as fixed wireless access (FWA) [114, 115]. We consider a network infrastructure with wireless access and backhaul networks which are owned and managed by wireless ASPs and backhaul service providers (BSPs), respectively where

[1]More detailed descriptions of the multiple slice connectivity are given in Section 7.3.

the BSPs sell backhaul services to the ASPs while the ASPs compete with each other for selling access services to UEs. The interactions among different stakeholders (UEs, ASPs, and BSPs) are studied by using the Stackelberg game theory which specifically models the competitions among the ASPs (i.e., peer SPs) and the buy-and-sell interactions between the ASPs and their UEs and between ASPs and BSPs. Specifically, this paper makes the following contributions.

- We formulate the interactions among UEs, ASPs and BSPs as a multi-leader multi-follower (MLMF) Stackelberg game [51]. Specifically, the ASPs and UEs act as the leaders and followers in this game, respectively where each ASP offers its access price for selling service to the UEs based on their anticipated budgets and the pricing policies of other ASPs. Moreover, each ASP also optimizes the backhaul capacity purchased from the BSPs to serve the UEs' demands aggregated at their access gateways while the UEs optimally purchase access bandwidth from the ASPs considering the offered prices.
- Assuming that players in the underlying Stackelberg game act rationally to maximize their own payoffs, the interactions of the players potentially result in a game equilibrium. By applying backward induction method, we derive the price best-response functions for the ASPs and the throughput best-response function for the UEs in the access layer. We prove the existence of a unique Stackelberg game equilibrium.
- We further prove that these best-response functions belong to the class of standard functions [52] and they satisfy the so-called two-sided scalability (2.s.s) property [53]. These results are leveraged in developing a distributed algorithm that converges to the game equilibrium. In this algorithm, the iterative updates using these best-response functions are adopted and an iterative strategy for cost-minimum backhaul resource acquisition for each ASP is proposed to meet the aggregated throughput demand in the wireless access layer.
- We evaluate the efficacy of our proposed framework and investigate the achievable performance and strategies of different network stakeholders via extensive numerical studies. Moreover, the payoffs achieved by the ASPs and UEs under the proposed Stackelberg game framework are compared with those achieved under three different baselines.

The remainder of this paper is organized as follows. We describe the system model and the game formulation in Sections 7.3 and 7.4, respectively. The detailed game analysis and proposed algorithm

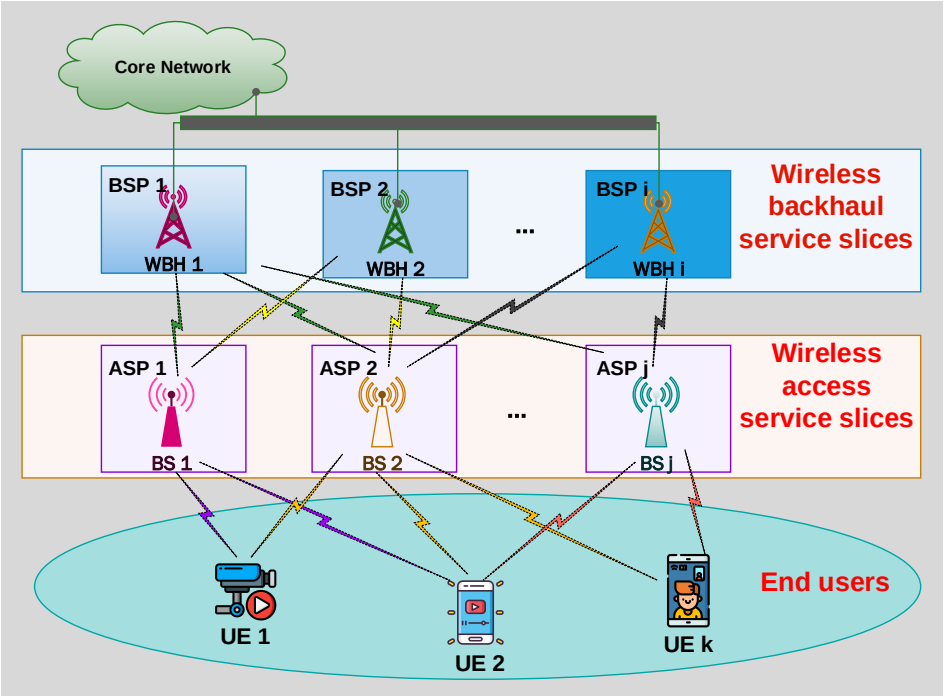


Figure 7.1 – The infrastructure-based network slicing framework. The access and backhaul network slices are respectively managed by multiple ASPs and BSPs. The ASPs (UEs) can simultaneously purchase services from different BSPs (ASPs).

to achieve the game equilibrium are described in Section 7.5. Section 7.6 presents numerical results and Section 7.7 concludes our paper. Key notations used in this paper are summarized in Table 7.1.

7.3 System Model

We consider the downlink of a cellular network with both wireless backhaul and access communications. We assume that the infrastructures and wireless resources of the wireless backhaul and access layers are owned and manged by I backhaul service providers (BSPs) and J access service providers (ASPs), respectively. These BSPs are collected in a set $\mathcal{I} = \{1, ..., i, ..., I\}$ where i is the index of the i-th BSP. For simplicity, we assume that BSP $i \in \mathcal{K}$ owns one corresponding wireless backhaul hub (WBH) and each BSP i has a sufficiently large and dedicated bandwidth that is non-overlapped with the spectra of other BSPs. Meanwhile, the ASPs are collected in a set $\mathcal{J} = \{1, ..., j, ..., J\}$ where each ASP j possesses one base station (BS) and spectrum band of W_j^{A} Hz. We assume that these BSs serve a set $\mathcal{K} = \{1, ..., k, ..., K\}$ of UEs in a particular service area (i.e., a cell) and the

Tableau 7.1 – Summary of Key Notations

Notation	Description
I	Number of BSPs
J	Number of ASPs
K	Number of UEs
W_j^{A}	Spectrum bandwidth (Hz) of ASP j
s_{jk}	Fraction of bandwidth allocated by ASP j to UE k
w_{ij}	Amount of bandwidth (Hz) that ASP j purchases from BSP i
r_{ij}	Average spectrum efficiency (bps/Hz) of the backhaul link between ASP j and BSP i
r_{jk}	Average spectrum efficiency (bps/Hz) of the access link between UE k and ASP j
d_{jk}	Throughput demand (bps) of UE k served by ASP j
$\boldsymbol{d}_k$	Throughput demand vector of UE k
$\boldsymbol{d}_{-k}$	Throughput demand vector of all UEs except UE k
p_j	Price per throughput demand unit (\$/bps) imposed by ASP j
$\boldsymbol{p}_{-j}$	Price vector imposed by all ASPs except ASP j
q_i	Price per backhaul bandwidth unit (\$/Hz) offered by BSP i
$P_j(p_j, \boldsymbol{s}_j, \boldsymbol{w}_j)$	Payoff function of ASP j
$U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j)$	Revenue function of ASP j
δ_j	Coefficient associated with $U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j)$
θ_j	Resource allocation cost coefficient of ASP j
$C_j^{\mathsf{A}}(\boldsymbol{w}_j)$	Backhaul cost function of ASP j
η_j	Coefficient associated with $C_j^{\mathsf{A}}(\boldsymbol{w}_j)$
$U_k^{\mathsf{E}}(\boldsymbol{d}_k)$	Payoff function of UE k
e_k	Utility coefficient of UE k
B_k	Budget of UE k (\$)

spectrum bands used by these ASPs are non-overlapped. In practice, the ASPs can reuse their spectrum in other areas which are sufficiently far away, thus introducing certain inter-cell interference. However, one can greatly mitigate such interference with careful cell planning. Furthermore, the average co-channel interference can be estimated and accumulated in the background noise power for users in the considered service area [48].

We further assume that the ASPs must purchase backhaul communication resources from BSPs to support end-to-end communications between UEs and the core network (CN). Interactions among different network stakeholders for resource trading occur in fixed-size time intervals where the number of active users remains the same in each time interval. Let w_{ij} denote the amount of bandwidth (Hz) that ASP j acquires from the BSP i, and r_{ij} (bps/Hz) represent the average spectrum efficiency achieved by the corresponding backhaul link. We assume that UE k purchases

a fraction $s_{jk} \in [0, 1]$ of the ASP j's spectrum resource for data transmission in the access layer. The average spectrum efficiency achieved by the access link between UE k and ASP j is denoted as r_{jk} (bps/Hz).[2]

We assume that each ASP can transfer data over multiple backhaul links through multiple connections with different BSPs simultaneously. Similarly, each UE can receive data over multiple access links associated with different ASPs simultaneously. In practice, this can be achieved by the multi-connection technology [54–57] and multi-band data aggregation techniques [58, 59]. Further, several traffic steering techniques [60–64] together with supporting protocols such as Multipath TCP [65] and IETF 0-RTT-Convert [66] could be exploited for splitting/merging data streams to/from multiple link interfaces. As a result, ASPs (UEs) are able to purchase backhaul (access) bandwidth from different BSPs (ASPs) at the same time, thereby enabling multiple slice connectivity [67] for both the UEs and ASPs. The multilateral interactions among these network stakeholders and end users constitute a trading market including the backhaul and access resource markets. Fig. 7.1 depicts our system model.

7.4 MLMF Stackelberg Game Formulation

In this section, we describe how to formulate the interactions among network stakeholders as a two-stage MLMF Stackelberg game. Specifically, the ASPs and UEs act as the leaders and the followers of the game, respectively. The leaders play first in Stage I by imposing access prices taking into account the potential budgets and demands of the UEs, and the ASPs decide the amount of bandwidth resource acquired from the BSPs for cost minimization. In the second stage (Stage II), each UE optimally purchases data from the ASPs to maximize its utility given the prices imposed by the ASPs. We assume that each player selfishly maximizes its own payoff function in this MLMF Stackelberg game. In the following, we present the utility functions and the decision making operations of different network stakeholders.

[2]Note that r_{ij} and r_{jk} are obtained by averaging the spectrum efficiency over wireless fading channel in the underlying time slot considering potential interference among co-channel network sections.

7.4.1 Stage I: Noncooperative Access Pricing Game among ASPs and Backhaul Resource Acquisition

7.4.1.1 Payoff Function of Each ASP

By anticipating throughput demand from associated UEs, each ASP needs to buy backhaul bandwidth from BSPs to guarantee the congestion-free end-to-end connection between the UEs and the CN. Specifically, each ASP is a middleman who plays as a seller and a buyer in the access and backhaul resource markets, respectively. Consequently, the payoff of ASP j can be defined as the revenue earned from providing access services to UEs minus the backhaul expenditure. Specifically, the ASP $j \in \mathcal{J}$ is interested in maximizing the following payoff function:

$$P_j(p_j, \boldsymbol{s}_j, \boldsymbol{w}_j) = \delta_j U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j) - \eta_j C_j^{\mathsf{A}}(\boldsymbol{w}_j) \tag{7.1}$$

where δ_j and η_j are respectively the coefficients associated with the revenue function $U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j)$ and cost function $C_j^{\mathsf{A}}(\boldsymbol{w}_j)$, and δ_j and η_j are typically set to ensure that $P_j(p_j, \boldsymbol{s}_j, \boldsymbol{w}_j) \geq 0$. In (7.1), $\boldsymbol{w}_j = (w_{ij})_{i\in\mathcal{I}}$ denotes the vector of backhaul bandwidth purchased by the ASP j from the BSPs and $\boldsymbol{s}_j = (s_{jk})_{k\in\mathcal{K}}$ represents the vector of access bandwidth fractions that ASP j allocates to the associated UEs to meet their throughput demands $(d_{jk})_{k\in\mathcal{K}}$. Moreover, the revenue function $U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j)$ is defined as

$$U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j) = \sum_{k=1}^{K} p_j d_{jk} - \theta_j \sum_{k=1}^{K} W_j^{\mathsf{A}} s_{jk} \tag{7.2}$$

which is the revenue earned from selling the access service to UEs with a price p_j, subtracted by the resource allocation cost with a coefficient θ_j.

Because each ASP must purchase the backhaul bandwidth to serve its UEs, the ASP j has to pay the backhaul cost $C_j^{\mathsf{A}}(\boldsymbol{w}_j)$, which is defined as follows:

$$C_j^{\mathsf{A}}(\boldsymbol{w}_j) = \sum_{i=1}^{I} q_i w_{ij} + \frac{1}{2}\sum_{i=1}^{I} w_{ij}^2 + \nu_j \sum_{i'\neq i} w_{ij} w_{i'j} \tag{7.3}$$

where q_i denotes the price per backhaul bandwidth unit (\$/Hz) offered by the BSP i. Here, we adopt a quadratic backhaul cost function as differentiating this function resulting in a linear backhaul throughput demand structure for the ASPs, thus making the analysis tractable. This quadratic cost

function also helps the ASPs avoid the over purchasing of backhaul resource due to its concavity. Moreover, this cost function incorporates the substitutability property with the parameter $\nu_j \in (0, 1)$ [116]. In this paper, we adopt this property to allow the ASP j to transmit data over different backhaul links purchased from different BSPs, thus ensuring the multi-band transmission ability for each ASP. [3]

For simplicity, we consider the optimization of the two functions $U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j)$ and $C_j^{\mathsf{A}}(\boldsymbol{w}_j)$ independently. Specifically, the ASP j has to set the competitive price p_j to attract more demand from UEs considering the available spectrum resource W_j^{A} and the prices offered by other ASPs. The revenue maximization of individual ASPs taking into account the prices of other ASPs thus constitute to a game called *Stage-I* AG *game*, which is defined later. Meanwhile, the ASP j has to compete with other ASPs in buying bargain backhaul resources from the BSPs to reduce the backhaul expenditure $C_j^{\mathsf{A}}(\boldsymbol{w}_j)$. Finally, the following constraints must be imposed for each ASP/BS j to meet the traffic demand of its associated UEs:

$$\sum_{i=1}^{I} w_{ij} r_{ij} \geq \sum_{k=1}^{K} d_{jk}. \tag{7.4}$$

We will elaborate this cost minimization problem later in Section 7.5.3.

7.4.1.2 The Stage-I AG Game - The Access Resource Pricing Game

The utility function (7.2) of the ASP j can be expressed as $U_j^{\mathsf{A}}(p_j, \boldsymbol{p}_{-j}, \boldsymbol{s}_j)$ to describe the interactions with other ASPs. Here, $\boldsymbol{p}_{-j} = (p_{j'})_{j' \in \mathcal{J}, j' \neq j}$ denote the strategies of other ASPs except ASP j. Now, the AG game can be defined as follows:

1. *Players:* The ASPs in the set $\mathcal{J}$.
2. *Strategy:* $p_j^{\mathsf{L}} \leq p_j \leq p_j^{\mathsf{U}},\ \forall j \in \mathcal{J}$ such that

$$\sum_{k=1}^{K} s_{jk} \leq 1, \forall j \in \mathcal{J}. \tag{7.5}$$

[3] Quadratic cost function was established in economics when considering the price and quantity competition in a duopoly market [116]. Due to its concavity, it is able to represent the saturation of customer satisfaction when its demand increases. Moreover, this cost function incorporates the spectrum substitutability with the parameter ν_j. Here, goods are substitutable when $\nu_j > 0$, meaning that the goods produced by different firms/manufacturers in a certain market have a similar property/functionality. Therefore, a customer can replace goods produced by a certain firm by those from another firms for its need. Quadratic cost function is also adopted in other works [39, 41, 117].

3. *Utility function:* $U_j^{\mathsf{A}}(p_j, \boldsymbol{p}_{-j}, \boldsymbol{s}_j),\ \forall j \in \mathcal{J}$.

In the AG game, p_j^{L} and p_j^{U} are the positive lower and upper bounds on the access price, which are imposed by the market regulations for the ASP j. Note that (7.5) represents the access bandwidth (fraction) allocation constraint for each ASP j.

7.4.2 Stage II: Traffic Demand Optimization of Each UE

7.4.2.1 Payoff Function of Each UE

Inspired by [48] and [118], the payoff function of each UE $k \in \mathcal{K}$ is defined as follows:

$$U_k^{\mathsf{E}}(\boldsymbol{d}_k) = e_k \sum_{j=1}^{J} \log\left(1 + d_{jk}\right) \tag{7.6}$$

where $\boldsymbol{d}_k = (d_{jk})_{j\in\mathcal{J}}$ denotes the throughput demand vector of UE k, $d_{jk} = s_{jk} W_j^{\mathsf{A}} r_{jk}$ is the throughput of UE k supported by ASP j, and e_k denotes the utility coefficient of UE k. The logarithm-based payoff function is adopted since it is a strictly increasing and concave function of the throughput. Moreover, this payoff function returns diminishing utility for UE k with respect to the throughput d_{jk}. This captures the practical situation where the increase in service satisfaction of end users would diminish with the increase in the achievable throughput.

7.4.2.2 UE's Throughput Demand Optimization Problem in Stage II

Given the price imposed by the leaders (ASPs) $\boldsymbol{p}^*$, UE k is interested in maximizing its payoff by acquiring appropriate throughput for different ASPs considering its maximum budget B_k. Mathematically, this throughput demand optimization problem can be formulated as

$$\max_{\boldsymbol{d}_k \geq \boldsymbol{0}} \ U_k^{\mathsf{E}}(\boldsymbol{d}_k) \tag{7.7a}$$

$$\text{s.t.} \ \sum_{j=1}^{J} p_j d_{jk} \leq B_k \tag{7.7b}$$

where (7.7b) captures the maximum budget constraint of UE k.

7.5 Analysis of the MLMF Stackelberg Game Equilibrium

We first define the Stackelberg equilibrium (SE) and Nash equilibrium (NE) in Definition 7.1 in the following.

Definition 7.1. *The SE of a Stackelberg game is the NE between the leaders and followers. The NE of a game is a point at which no player has incentives to unilaterally change its strategy for achieving a better utility without deteriorating the utilities of other players. In this paper, a point* $(\boldsymbol{p}^*, \boldsymbol{d}^*)$*, where* $p_j^* \in \mathcal{P}_j$*, is the SE of the Stackelberg game if it satisfies the following conditions. For each ASP* j *in Stage I,*

$$U_j^{\mathsf{A}}(p_j^*, \boldsymbol{p}_{-j}^*, \boldsymbol{d}^*) \geq U_j^{\mathsf{A}}(p_j, \boldsymbol{p}_{-j}^*, \boldsymbol{d}^*),\ \forall p_j \in \mathcal{P}_j,\ j \in \mathcal{J}. \tag{7.8}$$

For each UE k *in Stage II,*

$$U_k^{\mathsf{E}}(\boldsymbol{p}^*, \boldsymbol{d}_k^*, \boldsymbol{d}_{-k}^*) \geq U_k^{\mathsf{E}}(\boldsymbol{p}^*, \boldsymbol{d}_k, \boldsymbol{d}_{-k}^*), \forall k \in \mathcal{K}. \tag{7.9}$$

Here, (7.8) *is the conditions for achieving the NE between the leaders (ASPs), and* (7.9) *is the condition for achieving the NE between the followers (UEs).*

In the following, we derive the SE of the considered Stackelberg game by using the *backward induction* method. Specifically, we first derive the optimal throughput demand of UEs in Stage II of the Stackelberg game. Then, we use this result to derive the NE among the ASPs in Stage I of the game.

7.5.1 Optimal Throughput Demand of UEs in Stage II

We first state the optimal throughput demand of UEs in Stage II of the Stackelberg game in the following lemma.

Lemma 7.1. *For given prices $(p_j)_{j\in\mathcal{J}}$ and budget B_k, the optimal throughput demand of UE k obtained by solving problem (7.7) can be expressed as follows:*

$$\begin{aligned} d_{jk}^* &= \left[\frac{B_k + \sum_{j'=1}^{J} p_{j'}}{Jp_j} - 1\right]^+ \\ &= \left[\frac{B_k + \sum_{j'\neq j} p_{j'}}{Jp_j} + \frac{1}{J} - 1\right]^+, \ \forall j \in \mathcal{J} \end{aligned} \tag{7.10}$$

where $[x]^+ = \max\{0, x\}$.

Proof. The proof is given in Appendix A. □

From (7.10), d_{jk}^*, which is the optimal throughput of UE k given by ASP j, is a function of the access prices $(p_j, \boldsymbol{p}_{-j})$ of the ASPs. Three characteristics of $d_{jk}^*(p_j, \boldsymbol{p}_{-j})$ can be drawn from (7.10). First, the throughput demand of each UE is not interfered by other UEs' throughput demand. The throughput demand solution in (7.10) is thus the NE in Stage II. Second, if ASP j offers higher price p_j then UE k will acquire smaller throughput from this provider, which reflects an intuitive buy-and-sell interaction between UEs and ASPs. Third, given a fixed price p_j of ASP j, UE k will demand higher throughput from this ASP if other ASP $j' \neq j$ increases its price. This characteristic reflects a market-competition interactions among the peer ASPs. The throughput demand solution in (7.10) also represents the criteria for which each UE k purchases access services from different ASPs. Particularly, if $\frac{B_k+\sum_{j=1}^{J} p_j}{Jp_j} \leq 1$, then UE k is better off purchasing the ASP j's service.

7.5.2 Pricing and Resource Allocation Solution for the Access Layer in Stage-I AG Subgame

To accommodate the throughput d_{jk} demanded by UE k, ASP j must allocate a portion s_{jk} of its bandwidth, which is calculated by

$$s_{jk} = \frac{d_{jk}}{W_j^{\mathsf{A}} r_{jk}} = \frac{d_{jk}}{D_{jk}} \tag{7.11}$$

where $D_{jk} \triangleq W_j^{\mathsf{A}} r_{jk}$. We now analyze the NE of the Stage-I AG subgame with respect to the pricing and resource allocation for the access layer based on the UEs' throughput demand in Stage-II of the Stackelberg game. Suppose that UE k demands access service of ASP j, we have $d_{jk}^* \geq 0$,

thus we can remove the $[\cdot]^+$ operator in (7.10). By substituting the result in (7.10) into (7.11), the bandwidth portion allocated by ASP j to UE k can be expressed as

$$s_{jk}^* = \frac{B_k + \sum_{j' \neq j} p_{j'}}{JD_{jk}p_j} + \frac{1}{D_{jk}}\left(\frac{1}{J} - 1\right) = \frac{\alpha_{jk}(\boldsymbol{p}_{-j})}{p_j} + \beta_{jk} \tag{7.12}$$

where

$$\alpha_{jk}(\boldsymbol{p}_{-j}) \triangleq \frac{B_k + \sum_{j' \neq j} p_{j'}}{JD_{jk}} \tag{7.13}$$

$$\beta_{jk} \triangleq \frac{1}{D_{jk}}\left(\frac{1}{J} - 1\right). \tag{7.14}$$

By substituting the results in (7.10) and (7.12) into (7.2), the utility achieved by ASP j becomes

$$U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j, \boldsymbol{p}_{-j}) = K(\frac{1}{J} - 1)p_j + \sum_{k=1}^{K} \frac{B_k + \sum\limits_{j' \neq j} p_{j'}}{J} - \theta_j W_j^{\mathsf{A}} \sum_{k=1}^{K} \left(\frac{\alpha_{jk}(\boldsymbol{p}_{-j})}{p_j} + \beta_{jk}\right) \tag{7.15}$$

where we use the notation $U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j, \boldsymbol{p}_{-j})$ to present the impact of $\boldsymbol{p}_{-j}$ to the utility of ASP j, i.e., the impacts of other ASPs' strategies to the ASP j's strategy. For convenience, we define

$$\alpha_j(\boldsymbol{p}_{-j}) \triangleq \sum_{k=1}^{K} \alpha_{jk}(\boldsymbol{p}_{-j}) \tag{7.16}$$

$$\beta_j \triangleq \sum_{k=1}^{K} \beta_{jk} \tag{7.17}$$

then, (7.15) becomes

$$U_j^{\mathsf{A}}(p_j, \boldsymbol{s}_j, \boldsymbol{p}_{-j}) = K(\frac{1}{J} - 1)p_j + \sum_{k=1}^{K} \frac{B_k + \sum_{j' \neq j} p_{j'}}{J} - \theta_j W_j^{\mathsf{A}} \left(\frac{\alpha_j(\boldsymbol{p}_{-j})}{p_j} + \beta_j\right). \tag{7.18}$$

Now, we substitute the result for s_{jk}^* in (7.12) to (7.5) and exploit the fact that $p_j > 0$, this constraint becomes

$$\begin{aligned} &\sum_{k=1}^{K} \left(\frac{\alpha_{jk}(\boldsymbol{p}_{-j})}{p_j} + \beta_{jk}\right) \leq 1 \\ &\Leftrightarrow (1 - \beta_j)p_j - \alpha_j(\boldsymbol{p}_{-j}) \geq 0. \end{aligned} \tag{7.19}$$

Given the strategies $\boldsymbol{p}_{-j}$ of other ASPs, the optimal strategy of ASP $j \in \mathcal{J}$ for the Stage-I AG subgame, i.e., its best response, is the solution of the following optimization problem:

$$F_j(\boldsymbol{p}_{-j}) = \underset{p_j \in \mathcal{P}_j}{\operatorname{argmax}} \; U_j^{\mathsf{A}}(p_j, \boldsymbol{p}_{-j})$$
$$\text{s.t. constraint (7.19).} \tag{7.20}$$

We state the convexity of this problem in the following lemma.

Lemma 7.2. *Problem* (7.20) *is convex.*

Proof. We first prove that the utility function $U_j^{\mathsf{A}}(p_j, \boldsymbol{p}_{-j})$ is a concave function of $p_j \in \mathcal{P}_j$. The first-order derivative of $U_j^{\mathsf{A}}(p_j, \boldsymbol{p}_{-j})$ with respect to p_j is given by

$$\frac{\partial U_j^{\mathsf{A}}}{\partial p_j} = K(\frac{1}{J} - 1) + \frac{\theta_j W_j^{\mathsf{A}} \alpha_j(\boldsymbol{p}_{-j})}{p_j^2}. \tag{7.21}$$

Moreover, the second-order derivative of $U_j^{\mathsf{A}}(p_j, \boldsymbol{p}_{-j})$ with respect to p_j is given by

$$\frac{\partial^2 U_j^{\mathsf{A}}}{\partial^2 p_j} = -\frac{2\theta_j W_j^{\mathsf{A}} \alpha_j(\boldsymbol{p}_{-j})}{p_j^3}. \tag{7.22}$$

With $\alpha_{jk}(\boldsymbol{p}_{-j}) > 0$, we have $\frac{\partial^2 U_j^{\mathsf{A}}}{\partial^2 p_j} < 0, \; \forall p_j > 0$. Hence, $U_j^{\mathsf{A}}(p_j, \boldsymbol{p}_{-j})$ is a concave function of $p_j \in \mathcal{P}_j$ where $\mathcal{P}_j$ denotes the positive segment of p_j [33]. Furthermore, (7.19) is a linear constraint of $p_j \in \mathcal{P}_j$. Thus, problem (7.20) is a convex problem of p_j. □

We state the existence of a NE for the Stage-I AG subgame in Theorem 7.1.

Theorem 7.1. *There exists a NE for the Stage-I* AG *subgame.*

Proof. $\mathcal{P}$ is a compact convex subset on $\mathbb{R}^J$. In problem (7.20), its objective function is concave according to Lemma 7.2. Constraint (7.19) is convex on $\mathcal{P}_j$ for all $j \in \mathcal{J}$. Then, according to Theorem 1, Remarks 1 and 2 in [119], the Stage-I AG subgame possesses a NE. □

With the existence of the NE for the Stage-I AG subgame stated in Theorem 7.1, the solution of (7.20) forms the NE. Now, we proceed to derive the solution of (7.20). The Lagrangian of problem

(7.20) with the equivalent constraint (7.19) is given by

$$\mathcal{L}_j^{\mathsf{A}}(p_j, \lambda_j) = U_j^{\mathsf{A}}(p_j, \boldsymbol{p}_{-j}) + \lambda_j \left[(1-\beta_j)\, p_j - \alpha_j(\boldsymbol{p}_{-j})\right] \tag{7.23}$$

where λ_j is the Lagrange multiplier. The KKT optimality conditions of problem (7.20) are given by

$$\frac{\partial \mathcal{L}_j^{\mathsf{A}}}{\partial p_j} = 0 \tag{7.24a}$$

$$\text{constraint (7.19)} \tag{7.24b}$$

$$\lambda_j \geq 0 \tag{7.24c}$$

$$\lambda_j \left[(1-\beta_j)p_j - \alpha_j\right] = 0. \tag{7.24d}$$

Here, (7.24b) and (7.24c) are the primal and dual feasibility conditions, respectively. Moreover, (7.24d) is the complementary slackness condition. Because problem (7.20) is convex, its optimal solution can be obtained from solving the above KKT optimality conditions. By solving (7.24a) with $p_j > 0$, we obtain

$$p_j^* = \sqrt{\frac{\theta_j W_j^{\mathsf{A}} \alpha_j(\boldsymbol{p}_{-j})}{K\left(1-\frac{1}{J}\right) - \lambda_j(1-\beta_j)}}. \tag{7.25}$$

The nominator of the fractional term inside the square root of (7.25) is positive for all positive $\boldsymbol{p}_{-j}$. Therefore, to guarantee a positive denominator for this fractional term of (7.25) thus making p_j^* feasible, we need the following condition on λ_j:

$$\lambda_j < \frac{K(1-\frac{1}{J})}{1-\beta_j}. \tag{7.26}$$

Note that $(1-\beta_j)$ is always positive, thus the condition (7.26) does not conflict with $\lambda_j \geq 0$. Moreover, according to the complementary slackness condition (7.24d), there are two cases: the equality condition of (7.19) holds and does not hold, i.e., the ASP j allocates all of its access bandwidth to UEs or not, respectively. We study these two cases in the following.

7.5.2.1 Case 1 - The equality condition of constraint (7.19) holds

In this case, each ASP j allocates all of its access bandwidth to UEs, then we have $\lambda_j > 0$ and

$$(1-\beta_j)p_j - \alpha_j(\boldsymbol{p}_{-j}) = 0. \tag{7.27}$$

By solving (7.27), we obtain

$$\lambda_j = \frac{K\left(1-\frac{1}{J}\right)}{1-\beta_j} - \frac{\theta_j W_j^{\mathsf{A}}(1-\beta_j)}{\sqrt{\alpha_j(\boldsymbol{p}_{-j})}}. \tag{7.28}$$

Due to the positive subtrahend term in (7.28), λ_j in (7.28) satisfies condition (7.26). Furthermore, because of the condition (7.24c), we have $\lambda_j > 0$ or equivalently

$$\theta_j < \frac{K\left(1-\frac{1}{J}\right)\sqrt{\alpha_j(\boldsymbol{p}_{-j})}}{W_j^{\mathsf{A}}(1-\beta_j)^2}. \tag{7.29}$$

Because the right hand side of (7.29) depends on $\alpha_j(\boldsymbol{p}_{-j})$, we need to impose the tighter constraint for the right hand side of (7.29), i.e.,

$$\begin{aligned} \theta_j &< \min_{\boldsymbol{p}_{-j}} \left[\frac{K\left(1-\frac{1}{J}\right)\sqrt{\alpha_j(\boldsymbol{p}_{-j})}}{W_j^{\mathsf{A}}(1-\beta_j)^2}\right] \\ &\Leftrightarrow \theta_j < \frac{K\left(1-\frac{1}{J}\right)\sqrt{\alpha_j(\boldsymbol{p}^{\mathsf{L}}_{-j})}}{W_j^{\mathsf{A}}(1-\beta_j)^2} \end{aligned} \tag{7.30}$$

where $\boldsymbol{p}^{\mathsf{L}}_{-j} = (p^{\mathsf{L}}_{j'})_{j'\neq j}$. With the condition on θ_j given in (7.30), we can make sure that λ_j in (7.28) is always positive, which guarantees the feasibility of Case 1. Also from (7.27), we have

$$K\left(1-\frac{1}{J}\right) - \lambda_j(1-\beta_j) = \frac{\theta_j W_j^{\mathsf{A}}(1-\beta_j)^2}{\sqrt{\alpha_j(\boldsymbol{p}_{-j})}}. \tag{7.31}$$

We can see that the denominator of (7.25) is equal to the right hand side of (7.31), thus by substituting the right hand side of (7.31) into (7.25), we have

$$F_j^{\mathsf{h}}(\boldsymbol{p}_{-j}) = p_j^* = \sqrt{\frac{\alpha_j(\boldsymbol{p}_{-j})\sqrt{\alpha_j(\boldsymbol{p}_{-j})}}{(1-\beta_j)^2}}. \tag{7.32}$$

$\boldsymbol{F}^{\mathsf{h}}(\boldsymbol{p}) = (F_j^{\mathsf{h}}(\boldsymbol{p}_{-j}))_{j\in\mathcal{J}}$ is the best response function for the price offered to the UEs in Case 2.

To find the price for ASP j, we propose an iterative mechanism to update p_j^* according to (7.32) given the current prices of other ASPs $\boldsymbol{p}_{-j}$ as follows:

$$p_j^{(t+1)} = F_j^{\mathsf{h}}(\boldsymbol{p}_{-j}^{(t)}) = \left[\sqrt{\frac{\alpha_j(\boldsymbol{p}_{-j}^{(t)})\sqrt{\alpha_j(\boldsymbol{p}_{-j}^{(t)})}}{(1-\beta_j)^2}}\right]_{\mathcal{P}_j} \quad \forall j \in \mathcal{J} \tag{7.33}$$

where t is the iteration index and $[\cdot]_{\mathcal{P}_j}$ is the projection onto $\mathcal{P}_j$.

We can prove that the proposed iterative update in (7.33) indeed converges to a unique fixed point. Toward this end, we will show that $\boldsymbol{F}(\boldsymbol{p})$ is a *standard function* [52] with *two-sided scalability* (2.s.s) property [53] on $\mathcal{P} = \bigcap(\mathcal{P}_j)_{j\in\mathcal{J}}$. According to [53], the iterative update based on this function converges to a unique fixed point. Specifically, a function $\boldsymbol{F}(\boldsymbol{p})$ is called standard and satisfies the 2.s.s property if the following conditions hold:

- Positivity: $\boldsymbol{F}(\boldsymbol{p}) > \boldsymbol{0}$
- Monotonicity: if $\boldsymbol{p} \geq \boldsymbol{p'}$ then $\boldsymbol{F}(\boldsymbol{p}) \geq \boldsymbol{F}(\boldsymbol{p'})$
- Scalability: $\forall \mu > 1$, $\mu\boldsymbol{F}(\boldsymbol{p}) \geq \boldsymbol{F}(\mu\boldsymbol{p})$
- Two-sided scalability (2.s.s): For all $\mu > 1$, $\frac{1}{\mu}\boldsymbol{p} \leq \boldsymbol{p'} \leq \mu\boldsymbol{p}$ implies $\frac{1}{\mu}\boldsymbol{F}(\boldsymbol{p}) < \boldsymbol{F}(\boldsymbol{p'}) < \mu\boldsymbol{F}(\boldsymbol{p})$.

Lemma 7.3. *$\boldsymbol{F}^{\mathsf{h}}(\boldsymbol{p}) = \left(F_j^{\mathsf{h}}(\boldsymbol{p}_{-j})\right)_{j\in\mathcal{J}}$, where $F_j^{\mathsf{h}}(\boldsymbol{p}_{-j})$ is given in (7.32), is a standard function with 2.s.s property in domain $\mathcal{P}$.*

Proof. First, $F_j^{\mathsf{h}}(\boldsymbol{p}_{-j}) > 0$ in $\mathcal{P}_{-j}$ for all $j \in \mathcal{J}$, thus $\boldsymbol{F}(\boldsymbol{p})$ is positive in $\mathcal{P}$. Second, given $\boldsymbol{p} \geq \boldsymbol{p'} > 0$, for all $j \in \mathcal{J}$ we have

$$\begin{aligned} &\alpha_j(\boldsymbol{p}_{-j}) \geq \alpha_j(\boldsymbol{p'}_{-j}) \\ &\Rightarrow \boldsymbol{F}_j^{\mathsf{h}}(\boldsymbol{p}_{-j}) \geq \boldsymbol{F}_j^{\mathsf{h}}(\boldsymbol{p'}_{-j}) \\ &\Rightarrow \boldsymbol{F}^{\mathsf{h}}(\boldsymbol{p}) \geq \boldsymbol{F}^{\mathsf{h}}(\boldsymbol{p'}) \end{aligned} \tag{7.34}$$

which confirms the monotonicity of $\boldsymbol{F}^{\mathsf{h}}(\boldsymbol{p})$ in $\mathcal{P}$. Third, for all $\mu > 1$, for all $j \in \mathcal{J}$, we have

$$\begin{aligned}
&\mu\alpha_j(\boldsymbol{p}_{-j}) = \sum_{k=1}^{K} \frac{\mu B_k + \sum\limits_{j' \neq j} \mu p_{j'}}{JD_{jk}} \geq \alpha_j(\mu \boldsymbol{p}_{-j}) \\
&\Rightarrow \mu F_j^{\mathsf{h}}(\boldsymbol{p}_{-j}) = \sqrt{\sqrt{\mu}\frac{(\mu\alpha_j(\boldsymbol{p}_{-j}))\sqrt{\mu\alpha_j(\boldsymbol{p}_{-j})}}{(1-\beta_j)^2}} \\
&\geq \sqrt{\sqrt{\mu}\frac{\alpha_j(\mu\boldsymbol{p}_{-j})\sqrt{\alpha_j(\mu\boldsymbol{p}_{-j})}}{(1-\beta_j)^2}} \\
&\geq F_j^{\mathsf{h}}(\mu\boldsymbol{p}_{-j}).
\end{aligned} \tag{7.35}$$

Thus, $\boldsymbol{F}(\boldsymbol{p})$ satisfies the scalability property. As a result, $\boldsymbol{F}(\boldsymbol{p})$ is a standard function in $\mathcal{P}$.

We now prove the 2.s.s. property of $\boldsymbol{F}(\boldsymbol{p})$. From the monotonicity property, we have $\boldsymbol{F}(\mu\boldsymbol{p}) \geq \boldsymbol{F}(\boldsymbol{p'})$ given $\mu\boldsymbol{p} \geq \boldsymbol{p'}$. By scalability, we have $\mu\boldsymbol{F}(\boldsymbol{p}) > \boldsymbol{F}(\mu\boldsymbol{p})$ which implies $\boldsymbol{F}(\boldsymbol{p'}) < \mu\boldsymbol{F}(\boldsymbol{p})$. For all $\mu > 1$, we can see that $\frac{1}{\mu}\alpha_j(\boldsymbol{p}_{-j}) < \alpha_j(\frac{1}{\mu}\boldsymbol{p}_{-j})$ for all $j \in \mathcal{J}$. Thus, we have $\frac{1}{\mu}\boldsymbol{F}^{\mathsf{h}}(\boldsymbol{p}_{-j}) < \boldsymbol{F}^{\mathsf{h}}(\frac{1}{\mu}\boldsymbol{p}_{-j})$. By monotonicity, we further have $\boldsymbol{F}^{\mathsf{h}}(\frac{1}{\mu}\boldsymbol{p}) \leq \boldsymbol{F}^{\mathsf{h}}(\boldsymbol{p'})$ for $\frac{1}{\mu}\boldsymbol{p} \leq \boldsymbol{p'}$. Consequently, $\frac{1}{\mu}\boldsymbol{F}^{\mathsf{h}}(\boldsymbol{p}) < \boldsymbol{F}^{\mathsf{h}}(\boldsymbol{p'})$. As a result, we have $\frac{1}{\mu}\boldsymbol{F}^{\mathsf{h}}(\boldsymbol{p}) < \boldsymbol{F}^{\mathsf{h}}(\boldsymbol{p'}) < \mu\boldsymbol{F}^{\mathsf{h}}(\boldsymbol{p})$ for $\frac{1}{\mu}\boldsymbol{p} \leq \boldsymbol{p'} < \mu\boldsymbol{p}$, which proves the 2.s.s property of $\boldsymbol{F}^{\mathsf{h}}(\boldsymbol{p})$. □

7.5.2.2 Case 2 - The equality condition of constraint (7.19) does not hold

In this case, each ASP j does not allocate all of its access bandwidth to UEs. we have $\lambda_j = 0$, and p_j^* in (7.25) becomes

$$p_j^* = F_j^{\mathsf{n}}(\boldsymbol{p}_{-j}) = \sqrt{\frac{\theta_j W_j^{\mathsf{A}} \alpha_j(\boldsymbol{p}_{-j})}{K\left(1-\frac{1}{J}\right)}} = \sqrt{\frac{\theta_j}{K(J-1)} \sum_{k=1}^{K} \frac{B_k + \sum\limits_{j' \neq j} p_{j'}}{r_{jk}}}. \tag{7.36}$$

This is the best response function of ASP j with respect to the prices of other ASPs. The best-response price of ASP j in this case only depends on the prices of other ASPs, but not the access bandwidth W_j^{A} for all $j \in \mathcal{J}$. Similar to Case 1, we propose an iterative routine to find p_j^* in (7.36) as follows:

$$p_j^{(t+1)} = F_j^{\mathsf{n}}(\boldsymbol{p}_{-j}^{(t)}) = \left[\sqrt{\frac{\theta_j}{K(J-1)} \sum_{k=1}^{K} \frac{B_k + \sum\limits_{j' \neq j} p_{j'}}{r_{jk}}}\right]_{\mathcal{P}_j} \tag{7.37}$$

where t is the iteration index.

Lemma 7.4. $\boldsymbol{F}^{\mathsf{n}}(\boldsymbol{p}) = \left(F_j^{\mathsf{n}}(\boldsymbol{p}_{-j})\right)_{j\in\mathcal{J}}$, *where* $F_j^{\mathsf{n}}(\boldsymbol{p}_{-j})$ *is given in* (7.36)*, is a standard function with 2.s.s property in domain* $\mathcal{P}$.

Proof. Let us define $f_j(\boldsymbol{p}_{-j}) \triangleq \sum_{k=1}^{K} \frac{B_k + \sum_{j'\neq j} p_{j'}}{r_{jk}}, \forall j \in \mathcal{J}$. For all $\boldsymbol{p} \geq \boldsymbol{0}$, $\boldsymbol{F}^{\mathsf{n}}(\boldsymbol{p})$ is positive. Moreover, for given $\boldsymbol{p} \geq \boldsymbol{p'}$, we have

$$f_j(\boldsymbol{p}_{-j}) \geq f_j(\boldsymbol{p'}_{-j}) \tag{7.38}$$

$$\Rightarrow F_j^{\mathsf{n}}(\boldsymbol{p}_{-j}) \geq F_j^{\mathsf{n}}(\boldsymbol{p'}_{-j}) \tag{7.39}$$

$$\Rightarrow \boldsymbol{F}^{\mathsf{n}}(\boldsymbol{p}) \geq \boldsymbol{F}^{\mathsf{n}}(\boldsymbol{p'}). \tag{7.40}$$

Thus, $\boldsymbol{F}^{\mathsf{n}}(\boldsymbol{p})$ is monotonic in $\boldsymbol{p} \in \mathcal{P}$. For all $\mu > 1$, we have

$$\mu F_j^{\mathsf{n}}(\boldsymbol{p}_{-j}) \geq \sqrt{\frac{\mu\theta_j}{K(J-1)} f_j(\boldsymbol{p}_{-j})} \geq \underbrace{\sqrt{\frac{\theta_j}{K(J-1)} f_j(\mu\boldsymbol{p}_{-j})}}_{F_j^{\mathsf{n}}(\mu\boldsymbol{p}_{-j})} \tag{7.41}$$

for all $j \in \mathcal{J}$, which confirms the scalability property of $\boldsymbol{F}^{\mathsf{n}}(\boldsymbol{p})$. Now, by monotonicity, we have $\boldsymbol{F}^{\mathsf{n}}(\boldsymbol{p'}) \leq \boldsymbol{F}^{\mathsf{n}}(\mu\boldsymbol{p})$ for $\boldsymbol{p'} \leq \mu\boldsymbol{p}$. Further, by scalability, $\boldsymbol{F}^{\mathsf{n}}(\mu\boldsymbol{p}) < \mu\boldsymbol{F}^{\mathsf{n}}(\boldsymbol{p})$. Thus, we have $\boldsymbol{F}^{\mathsf{n}}(\boldsymbol{p'}) < \mu\boldsymbol{F}^{\mathsf{n}}(\boldsymbol{p})$. Next, by monotonicity, $\boldsymbol{F}^{\mathsf{n}}(\boldsymbol{p'}) \geq \boldsymbol{F}^{\mathsf{n}}(\frac{1}{\mu}\boldsymbol{p})$ for $\boldsymbol{p'} \geq \frac{1}{\mu}\boldsymbol{p}$ with $\mu > 1$. Further, by scalability, $\boldsymbol{F}^{\mathsf{n}}(\frac{1}{\mu}\boldsymbol{p}) > \frac{1}{\mu}\boldsymbol{F}^{\mathsf{n}}(\boldsymbol{p})$. Hence, $\boldsymbol{F}^{\mathsf{n}}(\boldsymbol{p'}) > \frac{1}{\mu}\boldsymbol{F}^{\mathsf{n}}(\boldsymbol{p})$ for $\boldsymbol{p'} \geq \frac{1}{\mu}\boldsymbol{p}$. Thus, $\boldsymbol{F}^{\mathsf{n}}(\boldsymbol{p})$ satisfies the 2.s.s property. □

Lemma 7.5. *The iterative updates in* (7.33) *and* (7.37) *for Cases 1 and 2, respectively, converge to the corresponding fixed point* $\boldsymbol{p}^{\mathsf{h}*}$ *and* $\boldsymbol{p}^{\mathsf{n}*}$*, respectively.*

Proof. According to Lemmas 7.3 and 7.4, $\boldsymbol{F}^{\mathsf{h}}(\boldsymbol{p})$ and $\boldsymbol{F}^{\mathsf{h}}(\boldsymbol{p})$ are standard functions with 2.s.s property. Therefore, with $\boldsymbol{p}$ is bounded by $\mathcal{P}$ and the iterative updates (7.33) and (7.37), each will converge to the corresponding fixed points according to [53]. □

Based on the results in Lemma 7.5, we propose a distributed algorithm, which is summarized in Algorithm 7.1, to find the NE among the ASPs with respect to the prices $\boldsymbol{p}$ and $\boldsymbol{s}^*$. In this algorithm, each ASP initially announces its initial price to the market (line 3). After observing the prices from other ASPs, each ASP recalculates its price according to its best response function. This

routine, from lines 4 to 17, is repeated until the market reaches the NE. Then, the ASPs announce their best-response prices to the UEs (line 18). From lines 19 to 23, each ASP calculates the total demand of backhaul bandwidth, then executes Algorithm 7.2 to optimize the amount of backhaul bandwidths purchased from the BSPs. Note that Algorithm 7.2 is described in the next subsection.

7.5.2.3 Message Exchanges among Game Stakeholders

Here, we discuss how Algorithms 7.1 and 7.2 can be executed in a distributed manner through exchanging information among ASPs, UEs, and BSPs. In network slicing, there exists a slice orchestrator/service broker which is in charge of collecting network information, handling service operation, and providing resource abstraction for network slices [67]. Accordingly, the information required for resource trading between the ASPs, UEs, and BSPs, typically the prices offered by these SPs and the budgets of UEs, can be exchanged via the slice orchestrator. The UEs are also able to update both the physical and abstracted resource information, as well as to provide their own information to the network, by communicating with the slice orchestrator via the physical downlink and uplink control channels of the 5G wireless cellular system. As a result, it is feasible for involved network entities to exchange the required information, thereby enabling the successful execution of Algorithms 7.1 and 7.2.

7.5.3 The Backhaul Expenditure Minimization Problem

At the equilibrium point of the Stage-I AG subgame, each ASP obtains the total throughput demand from associated UEs, which is given by

$$R_j^{\mathsf{A}}(\boldsymbol{s}_j^*) \triangleq \sum_{k=1}^{K} s_{jk}^* W_j^{\mathsf{A}} r_{jk},\ \forall j \in \mathcal{J} \tag{7.42}$$

where $(s_{jk}^*)_{j\in\mathcal{J},k\in\mathcal{K}}$ is the bandwidth demand of UEs at the Stage-I AG subgame's NE point. Accordingly, the objective of each ASP is to minimize the total payment to all the BSPs given their backhaul prices while avoiding the traffic congestion at its BS. This backhaul constrained cost

Algorithm 7.1. Distributed Algorithm for Pricing and Access Bandwidth Allocation

Initialize $t = 0$ and ε.
Each ASP set $p_j^{(t)} = p_j^{\mathsf{L}}, \forall j \in \mathcal{J}$.
Each ASP announces its prices $p_j^{(t)}$ to the access market.
// Price adjustment
while True **do**
  $t = t + 1$
  for $j = 1$ to J **do**
    if Condition (7.29) is satisfied **then**
      ASP j updates its prices $p_j^{(t)}$ according to (7.33)
    else
      ASP j updates its prices $p_j^{(t)}$ according to (7.37)
    end if
  end for
  Each ASP announces its prices $p_j^{(t)}$ to other ASPs.
  if $||\boldsymbol{p}^{(t+1)} - \boldsymbol{p}^{(t)}|| < \varepsilon$ **then**
    Exit the While loop
  end if
end while
All the ASPs announce their prices to the UEs.
// Backhaul bandwidth acquisition
for Each ASP $j \in \mathcal{J}$ **do**
  Calculate $s_{jk},\ \forall j \in \mathcal{J}, k \in \mathcal{K}$ according to (7.12).
  Calculate $\boldsymbol{w}_j = (w_{ij})_{i \in \mathcal{I}}$ using Algorithm 7.2.
  Calculate the payoff according to (7.1).
end for

minimization problem for each ASP j can be stated as follows:

$$\min_{\boldsymbol{w}_j \geq \boldsymbol{0}} C_j^{\mathsf{A}}(\boldsymbol{w}_j) \tag{7.43a}$$

$$\text{s.t.} \sum_{i=1}^{I} w_{ij} r_{ij} \geq R_j^{\mathsf{A}}(\boldsymbol{s}_j^*) \tag{7.43b}$$

where (7.43b) is the backhaul constraint and R_j^{A} is defined in (7.42). This is a convex optimization problem due to the convex objective function and linear constraint (7.43b).

Lemma 7.6. *Given $\nu_j \in [0, 1)$, each ASP j only acquires an amount of backhaul bandwidth such that the aggregated backhaul throughput and the access throughput at each BS are equal. Furthermore, the optimal solution $\boldsymbol{w}_j^*$ of* (7.43) *for each ASP j can be obtained by solving the following system*

of equations:

$$w_{ij} = \xi_j \left[\gamma_j \kappa_{ij} - \rho_{ij}\right]^+ \tag{7.44a}$$

$$\sum_{i=1}^{I} w_{ij} r_{ij} = R_j^{\mathsf{A}}(\boldsymbol{s}_j^*) \tag{7.44b}$$

$$\gamma_j > 0 \tag{7.44c}$$

where

$$\xi_j \triangleq \frac{1}{(1-\nu_j)[1+\nu_j(I-1)]} \tag{7.45a}$$

$$\kappa_{ij} \triangleq r_{ij}[1+\nu_j(I-1)] - \nu_j \sum_{i=1}^{I} r_{ij} \tag{7.45b}$$

$$\rho_{ij} \triangleq q_i[1+\nu_j(I-1)] - \nu_j \sum_{i=1}^{I} q_i \tag{7.45c}$$

and γ_j represent the Lagrange multiplier associated with constraint (7.43b)*.*

Proof. The proof is given in Appendix B. □

In (7.44a), w_{ij} for all $i \in \mathcal{I}$ linearly depends on the value of γ_j, which can be considered as the water level at ASP j. Accordingly, we develop a water-filling based algorithm, which is summarized in Algorithm 7.2, to find an optimal water level γ_j, from which we can determine the acquired backhaul bandwidth $w_{ij},\ \forall i \in \mathcal{I}$. Since the total backhaul throughput at each ASP j must equal to $R_j^{\mathsf{A}}(\boldsymbol{s}_j^*)$, we can adopt a bisection search method to determine the backhaul bandwidth w_{ij} in Algorithm 7.2. The convergence of this algorithm is stated in Lemma 7.7 in the following.

Lemma 7.7. *By setting $\nu_j \leq \min\limits_{i\in\mathcal{I}} \left\{\frac{r_{ij}}{R_j}\right\}$, Algorithm 7.2 will converge.*

Proof. In (7.44a), w_{ij} are increasing functions of γ_j for all $i \in \mathcal{I}$ when $\kappa_{ij} > 0\ \forall i \in \mathcal{I}$, i.e., $r_{ij}[1+\nu_j(I-1)] - \nu_j \sum\limits_{i=1}^{I} r_{ij} > 0, \forall i \in \mathcal{I}$. We can see that

$$r_{ij}[1+\nu_j(I-1)] - \nu_j \sum_{i=1}^{I} r_{ij} \geq r_{ij} - \nu_j \sum_{i=1}^{I} r_{ij}, \forall i \in \mathcal{I} \tag{7.46}$$

Algorithm 7.2. Backhaul Bandwidth Acquisition of An ASP

Init γ_j^{L}, γ_j^{U}, $\forall j \in \mathcal{J}$, and ε.
while $||\boldsymbol{\gamma}^{\mathsf{U}} - \boldsymbol{\gamma}^{\mathsf{L}}|| > \varepsilon$ **do**
  Set $\gamma_j = (\gamma_j^{\mathsf{L}} + \gamma_j^{\mathsf{U}})/2$.
  Calculate w_{ij}, $\forall i \in \mathcal{I}, j \in \mathcal{J}$ according to (7.44a).
  Calculate $R_j = \sum_{i=1}^{I} w_{ij} r_{ij}$, $\forall j \in \mathcal{J}$.
  for $j = 1$ to J **do**
    if $R_j < R_j^{\mathsf{A}}(\boldsymbol{s}_j^*)$ **then**
      Update $\gamma_j^{\mathsf{L}} = \gamma_j$
    else
      Update $\gamma_j^{\mathsf{U}} = \gamma_j$
    end if
  end for
end while
Return γ_j, w_{ij}, $\forall i \in \mathcal{I}$.

due to $1 + \nu_j(I-1) > 1$ for all $\nu_j \geq 0$. Furthermore,

$$r_{ij} - \nu_j \sum_{i=1}^{I} r_{ij} > 0 \Leftrightarrow \nu_j < \frac{r_{ij}}{\sum_{i=1}^{I} r_{ij}}, \forall i \in \mathcal{I}. \tag{7.47}$$

Thus, by setting $\nu_j \leq \min_{i \in \mathcal{I}} \left\{ \frac{r_{ij}}{\sum_{i=1}^{I} r_{ij}} \right\}$, we can guarantee the monotonically increasing property of w_{ij} for all $i \in \mathcal{I}$. As a result, when we increase the water level γ_j, the weighted sum in (7.44b) will be met, i.e., the convergence of Algorithm 7.2 holds. □

Regarding the complexity, Algorithm 7.2 is based on the bisection search method, which requires at most $\mathcal{O}(\log \frac{1}{\varepsilon})$ comparisons with the accuracy tolerance ε (i.e. the worst-case logarithmic complexity). Meanwhile, Algorithm 7.1 consists of two main tasks, the best-response price calculation and the backhaul bandwidth acquisition, executed by each ASP. The first task only requires each ASP to update the two 2.s.s functions (7.33) or (7.37) in each iteration. Furthermore, the iterative update based on the 2.s.s function has a fast convergence speed [120, 121]. The second task requires each ASP to execute Algorithm 7.2, having the worst-case logarithmic running time. Therefore, the distributed Algorithm 7.1 has a low computational complexity.

7.6 Numerical Results

We evaluate the performance achieved by the proposed game theoretic framework for a wireless network with 3 BSPs ($I = 3$), 4 ASPs ($J = 4$) and 20 UEs ($K = 20$). The BSs of ASPs and UEs are uniformly distributed in a circular area with the radius of 200 meters, while the wireless backhaul hubs are located with distances up to 400 meters from the origin. The average spectrum efficiency of each access link between a BS of an ASP and a UE (i.e., $(r_{jk})_{j\in\mathcal{J},k\in\mathcal{K}}$) ranges from 0.35 to 12 bps/Hz, depending on the communication distance. Also, the average spectrum efficiency of each backhaul link between a BS of an ASP and a WBH of a BSP (i.e., $(r_{ij})_{i\in\mathcal{I},j\in\mathcal{J}}$) ranges from 1 to 5 bps/Hz. ASPs 1, 2, 3, and 4 have the access bandwidth of 50, 150, 250 and 350 kHz respectively unless stated otherwise. Each UE k has a budget of $B_k = 10$ and the utility coefficient is set as $e_k = 0.1$ for all $k \in \mathcal{K}$. We set $\delta_j = 1$ and $\eta_j = 10^{-11}$ for all $j \in \mathcal{K}$, $p_j^{\mathsf{L}} = 10^{-10}$ and $p_j^{\mathsf{U}} = 10^3$ for all $j \in \mathcal{J}$, $\theta_j = 2.5 \times 10^{-8}$ and $\nu_j = 0.1$ for all $j \in \mathcal{J}$. Finally, BSPs 1, 2, and 3 offer the prices of 10^{-8}, 2×10^{-8} and 3×10^{-8} (\$/Hz) for their backhaul bandwidth, respectively.

Figs. 7.2a and 7.2b demonstrate the convergence of the Algorithms 7.1 and 7.2, respectively. Here, we set $\varepsilon = 10^{-6}$ as the accuracy achieved by both algorithms at convergence. As shown in Fig. 7.2a, Algorithm 7.1 converges very quickly after 3 iterations for different values of θ_j. For the higher value of θ_j, Algorithm 7.1 converges to the higher-price equilibrium point. Algorithm 7.2 also converges quite quickly, as shown in Fig. 7.2b.

Fig. 7.3 shows the prices of ASPs as functions of the amount of access bandwidth for different values of θ_j. Here, we set the baseline bandwidths for ASPs 1, 2, 3 and 4 equal to 50, 150, 250 and 350 KHz, respectively, i.e., $\boldsymbol{W}_{\mathsf{base}}^{\mathsf{A}} = (50, 150, 250, 350)$ KHz. The horizontal axis represents the bandwidth factor/multiplier where the access bandwidth is equal to the baseline bandwidth $\boldsymbol{W}_{\mathsf{base}}^{\mathsf{A}}$ multiplied by this bandwidth factor. We can see that the prices imposed by the ASPs increase with the cost coefficient θ_j. When $\theta_j = 10^{-5}$ for all $j \in \mathcal{J}$, the conditions (7.29) does not hold and in this case the ASPs choose $\boldsymbol{F}^{\mathsf{n}}$ in (7.37) as their best response functions, which are independent of the value of access bandwidth. In contrast, when $\theta_j = 2.5 \times 10^{-8}$, the ASPs opt for $\boldsymbol{F}^{\mathsf{h}}$ in (7.33) as their best response functions because the condition (7.29) holds. In this case, the ASPs impose lower prices when more resource becomes available to maintain their competitive advantage in the access service market. In fact, when ASP j possesses more bandwidth resource, this provider is able to sell its access service at a lower price, thus attracting more UEs' demands. The UEs thus tend to

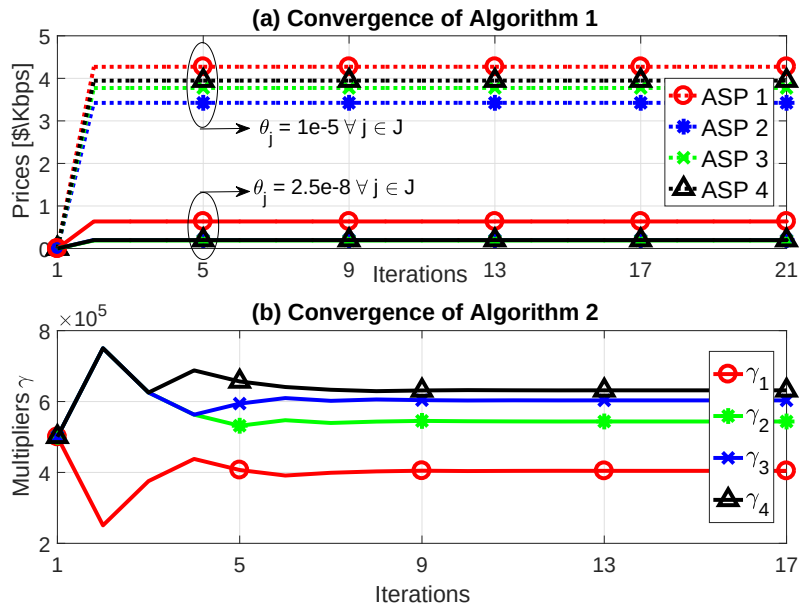


Figure 7.2 – Convergence of Algorithm 7.1 and Algorithm 7.2.

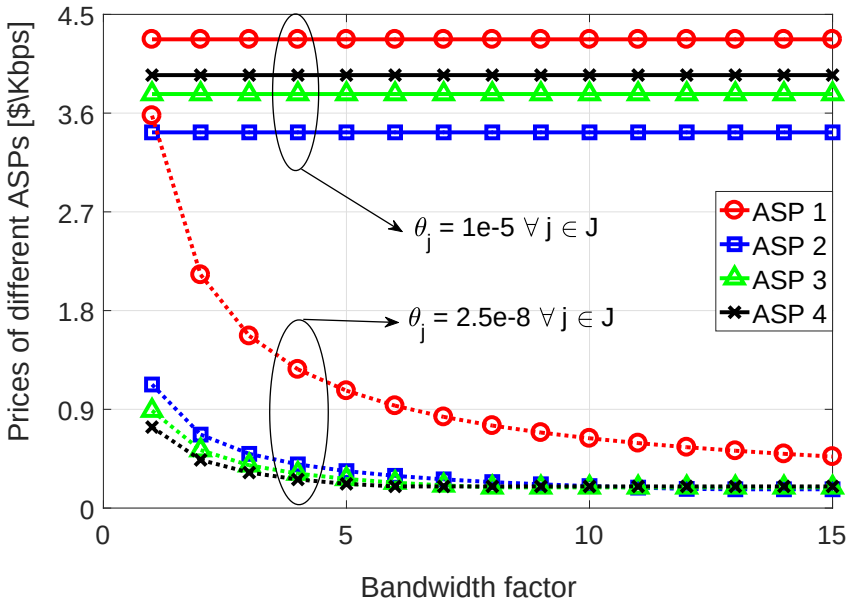


Figure 7.3 – Prices of ASPs vs. access bandwidth factor.

buy less throughput from other ASPs, as can be verified in the throughput demand given in (7.10). Moreover, other ASPs have to lower their prices for counter-reacting the market deviation of ASP j. Furthermore, Fig. 7.3 also shows that the smaller the access bandwidth possessed by an ASP, the higher the price offered by this ASP. Since ASP 1 has the smallest value of bandwidth resource in our setting, this provider always imposes a higher price than those of other ASPs.

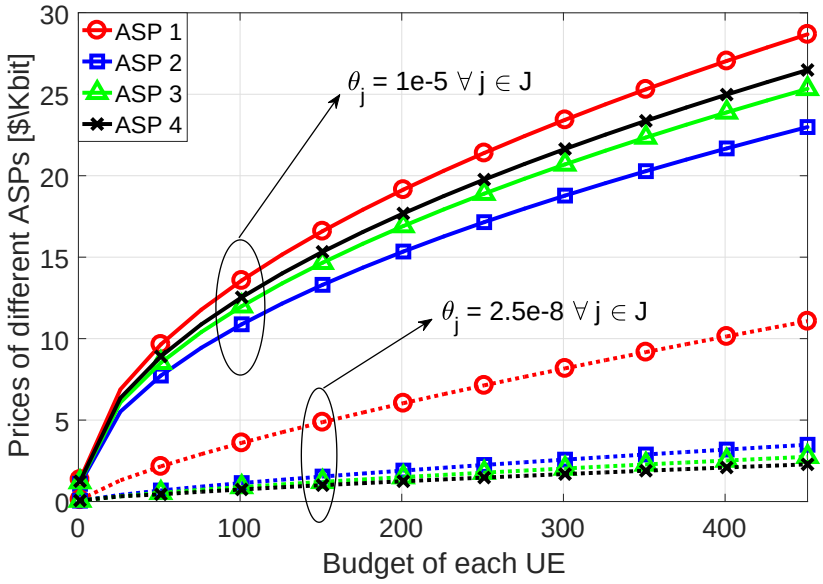


Figure 7.4 – Prices of ASPs vs. Budget of each UE.

Fig. 7.4 shows the prices of ASPs as functions of the UE' budget for different values of θ_j. It can be seen that the ASPs offer higher prices when each UE has a larger budget. The ASPs also impose higher prices when they have to pay more, i.e., a higher value of θ_j. Furthermore, the smaller the bandwidth resource possessed by an ASP, the higher the price is imposed by this ASP. This indeed confirms the relations between the price p_j and the UE's budget B_k given in (7.32) and (7.36).

Fig. 7.5 shows the average prices of ASPs versus the number of UEs in the network. This figure reveals that the sellers (ASPs) impose higher prices with higher demand from the larger number of buyers (UEs). When the number of UEs exceeds 80, their very large total rate demand forces the ASPs to exhaust their resources, resulting in a price surge. This phenomenon is amplified with the larger UEs' budget amount, which reflects the higher rate demand from the UEs.

Figs. 7.6 and 7.7 illustrate the payoffs achieved by the ASPs and the UEs, respectively for different values of the access bandwidth factor. These figures show that the payoffs increase when there is more access bandwidth resources. By setting lower prices (as shown in Fig. 7.3) with more bandwidth resources, the ASPs can attract larger UEs' demand, thus enabling to increase the ASPs' payoffs. As the UEs can purchase more throughput at lower price, their payoff becomes higher as well. However, the payoff of the ASPs becomes lower when θ_j increases, which implies a higher

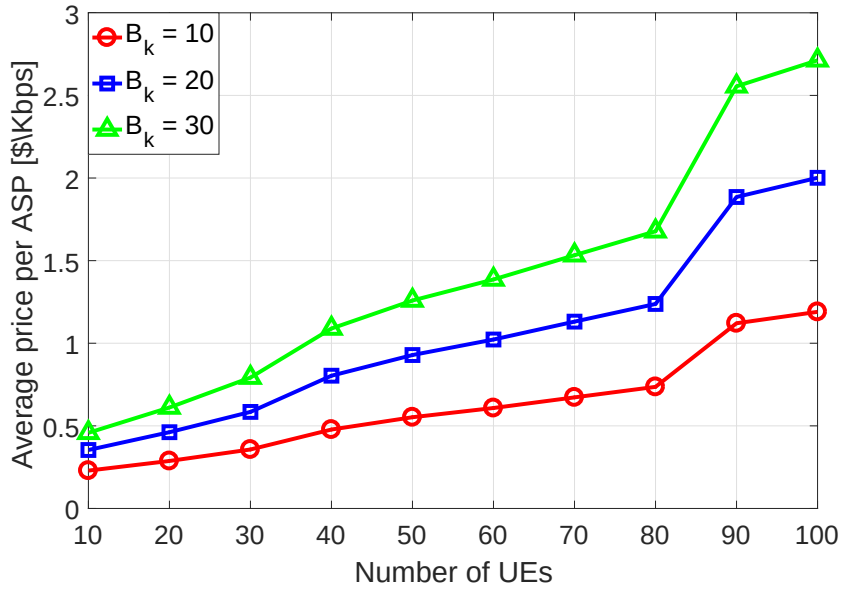


Figure 7.5 – Average price of ASPs vs. number of UEs.

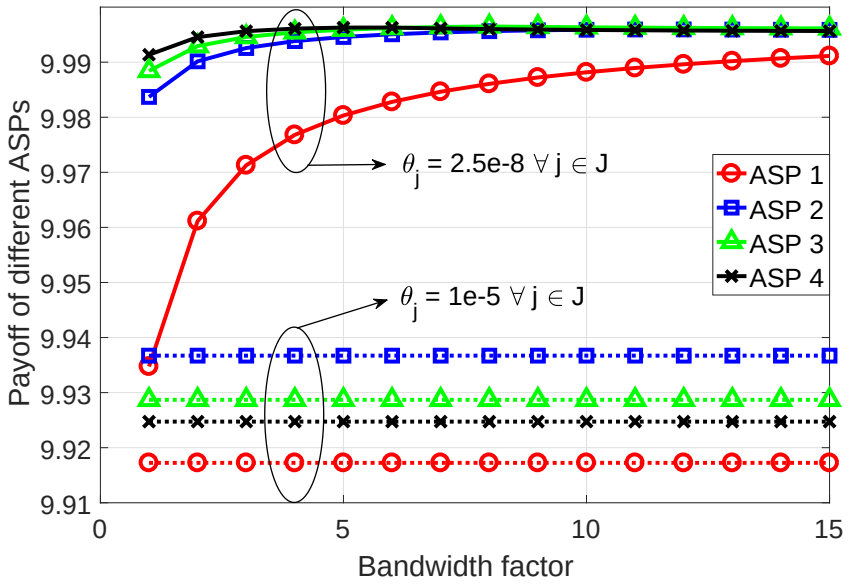


Figure 7.6 – Payoff of the ASPs vs. access bandwidth factor.

cost. Moreover, when the UEs have larger budget, their payoff increase as these UEs are able to purchase more throughput from the ASPs.

In Figs. 7.8 and 7.9, we compare the average payoffs of the ASPs and UEs resulted from the proposed framework and two other baselines schemes. The first baseline scheme is the premium price scheme [122] where the ASPs impose higher prices. This scheme is often used by sellers in

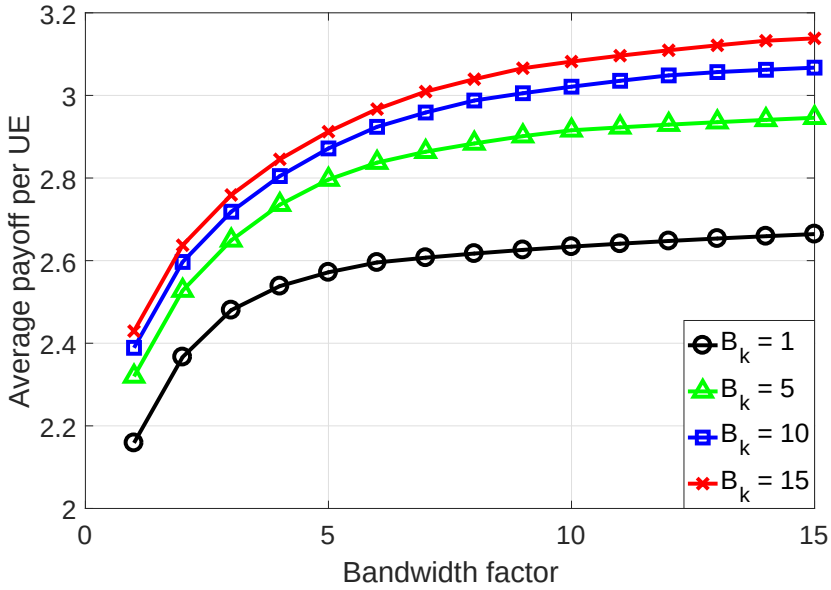
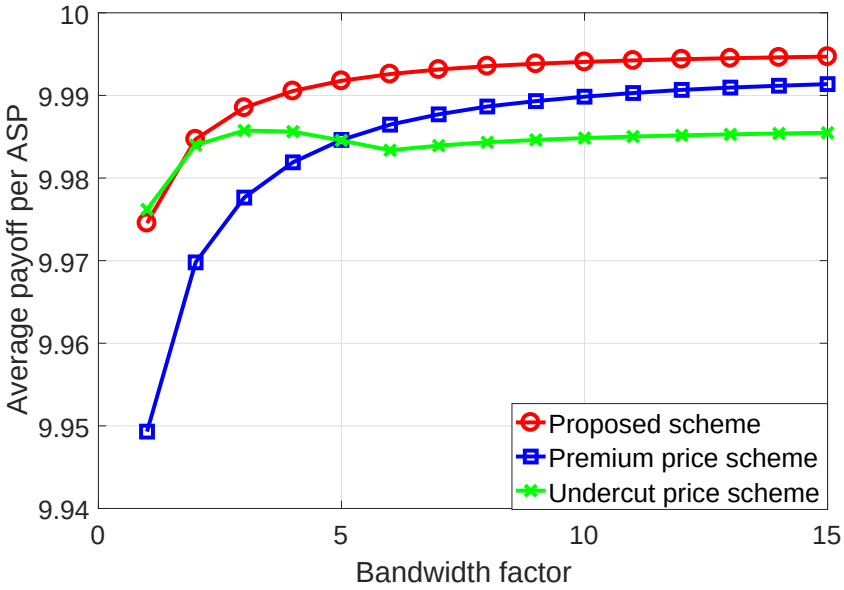


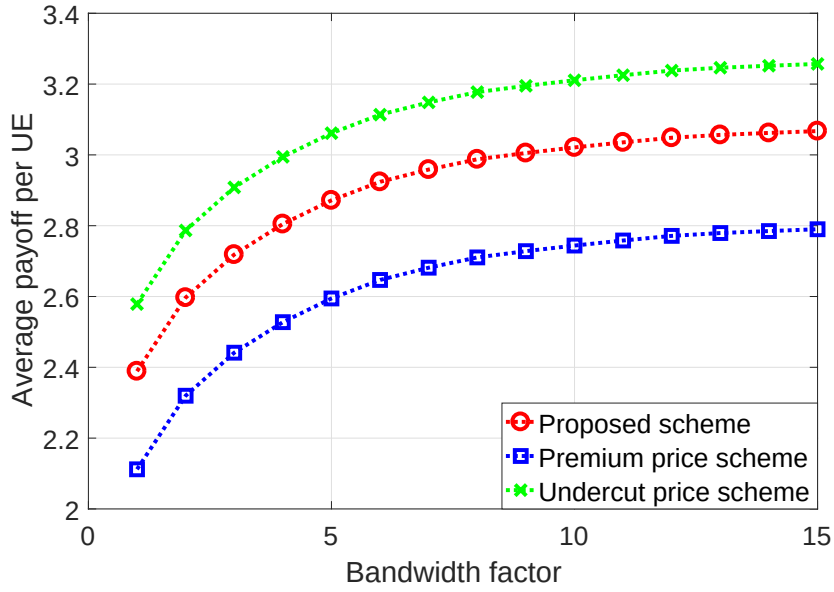
Figure 7.7 – Average payoff of the UEs vs. access bandwidth factor for different UE's budget.

luxury markets, or by sellers in oligopoly markets. Here, we set the ASPs' prices twice the prices obtained from the NE point of our proposed framework. The second baseline scheme is the undercut price scheme where one ASP deviates from the NE point by imposing a very low price for driving other ASPs out of the market [123, 124]. Here, we assume that ASP 4 undercuts the market by setting its price by 15% of the price corresponding to the NE point obtained from our proposed framework.

In Figs. 7.8a and 7.8b, we show the average payoffs of each APS and each UE, respectively due to the proposed framework and the considered baseline schemes as the access bandwidth factor varies. The average payoffs of each ASP and each UE are respectively shown in Figs. 7.9a and 7.9b for different values of each UE's budget. It can be observed that the proposed framework enables the ASPs to achieve highest average payoff (Figs. 7.8a and 7.9a) in comparison with other baseline schemes for all considered values of the bandwidth factor and UE's budget. Due to imposing significantly high prices under the premium price scheme, the ASPs discourage throughput demands from the UEs (even though when they have large budgets as shown in Fig. 7.8b). This premium scheme thus results in inferior average payoffs for both the ASPs and UEs compared to those due to the proposed scheme, as shown in Figs. 7.8 and 7.9. Under the undercut price scheme, the UEs are able to earn a slightly higher average payoff than that due to the proposed framework. The average payoff of each ASP under this scheme, however, cannot surpass the one due to the proposed



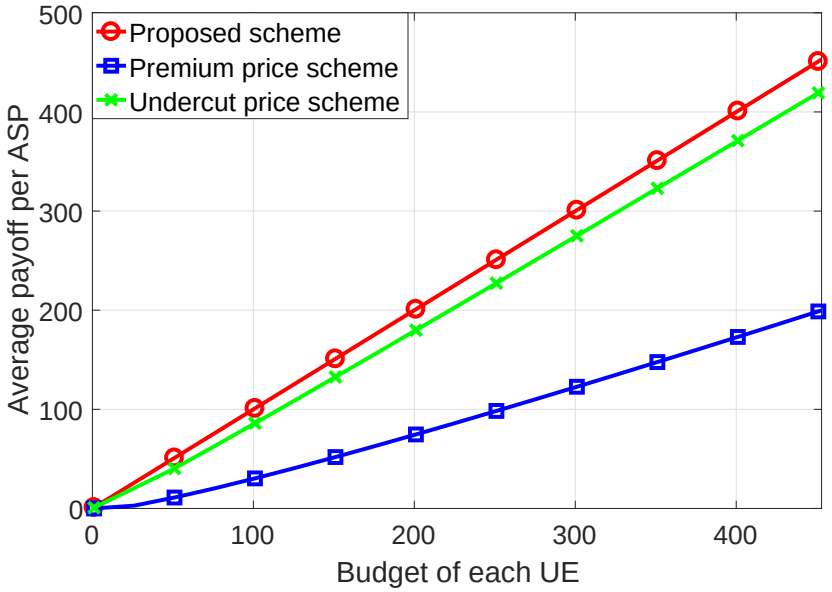
(a) Average payoff of ASPs.



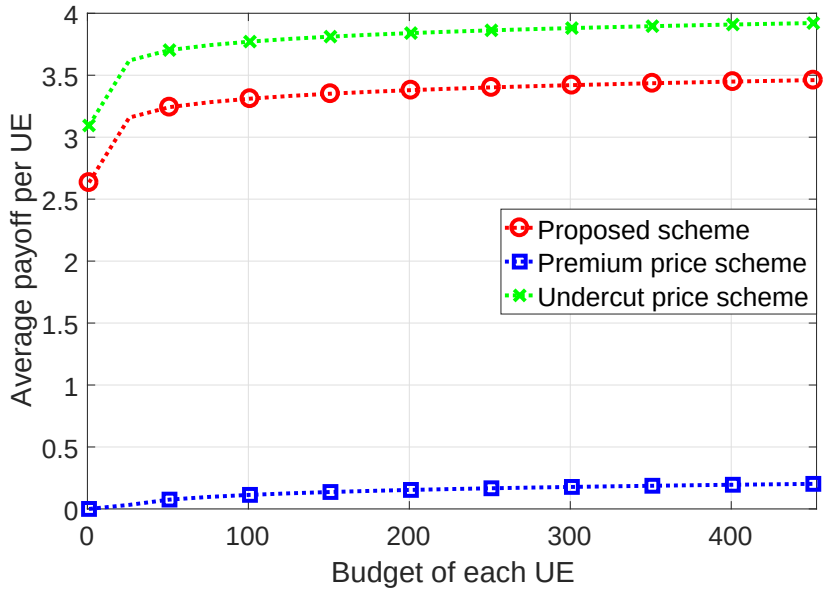
(b) Average payoff of UEs

Figure 7.8 – Average payoff of APSs and UEs in comparison with baseline pricing schemes when the access bandwidth factor varies.

framework. This is because a player (the ASP 4 in this circumstance) lowers its price below the SE to attract more throughput demands from the UEs while deteriorating the sale force of other rival ASPs. The increment in throughput demands from the UEs gained by the ASP 4, however, cannot compensate the lost due to the low-imposing price of this player. Therefore, the ASPs, who are the



(a) Average payoff of ASPs.



(b) Average payoff of UEs

Figure 7.9 – Average payoff of APSs and UEs in comparison with baseline pricing schemes when the UE's budget varies.

market leaders, are better off adopting the proposed framework to achieve highest average payoff for each of them while allowing a decent average payoff for each UE (the follower).

7.7 Conclusion

We have studied the interactions among the peer ASPs and between these ASPs and their UEs by employing the MLMF Stackelberg game model. The game equilibrium has been analyzed and we have shown how to reach the game equilibrium via a distributed algorithm, which has been built upon the derived price and best-response functions of the throughput demand for the ASPs and UEs, respectively. Numerical results have demonstrated fast convergence of the proposed algorithms and the economical efficacy of our proposed framework. Specifically, the ASPs tend to decrease and increase their prices with larger access bandwidth and smaller UE's budget, respectively. Moreover, both ASPs and UEs enjoy higher payoffs when the access bandwidth become higher. Finally, the proposed framework leads to higher average payoff for the ASPs and decently high payoff for the UEs compared to three considered baseline schemes.

Appendix A: Proof of Lemma 7.1

It can be verified that the optimization problem (7.7) is convex, thus its optimal solution $\boldsymbol{d}^* = (d_{jk}^*)_{j\in\mathcal{J}}$ can be obtained from solving its Karush-Kuhn-Tucker (KKT) optimality conditions [33], which are given as follows:

$$\frac{\partial \mathcal{L}_k^{\mathsf{E}}(\boldsymbol{d},\lambda_k)}{\partial d_{jk}} = \frac{e_k}{1+d_{jk}^*} - \lambda_k p_j = 0,\ \forall j \in \mathcal{J} \tag{7.48}$$

$$\text{constraint (7.7b)} \tag{7.49}$$

$$\lambda_k \geq 0 \tag{7.50}$$

$$\lambda_k \left(B_k - \sum_{j=1}^{J} p_j d_{jk}^* \right) = 0 \tag{7.51}$$

where $\mathcal{L}_k^{\mathsf{E}}(\boldsymbol{d},\lambda_k) \triangleq U_k^{\mathsf{E}}(\boldsymbol{d}) + \lambda_k(B_k - \sum_{j=1}^{J} p_j d_{jk})$ is the Lagrangian of problem (7.7) with the Lagrange multiplier λ_k. Constraints (7.7b) and (7.50) are the primal and dual feasible conditions, respectively. Moreover, (7.51) represents the complementary slackness condition.

Now, we proceed to find d_{jk}^* using these KKT conditions. Specifically, by solving (7.48) we have

$$d_{jk}^* = \left[\frac{e_k}{\lambda_k p_j} - 1 \right]^+ . \tag{7.52}$$

Because $d_{jk}^* \geq 0$, we must have $\lambda_k > 0$. Consequently, from (7.51) and the complementary slackness condition, we have $B_k - \sum_{j=1}^{J} p_j d_{jk}^* = 0$. By substituting (7.52) into this equation, we obtain

$$\lambda_k = \frac{Je_k}{B_k + \sum_{j=1}^{J} p_j}. \tag{7.53}$$

By substituting (7.53) into (7.52), we finally obtain (7.10).

Appendix B: Proof of Lemma 7.6

The Lagrangian of the optimization problem (7.43) is given by

$$\mathcal{L}_j^{\mathsf{C}} = C_j^{\mathsf{A}}(\boldsymbol{w}_j) + \gamma_j \left(R^{\mathsf{A}} - \sum_{i=1}^{I} w_{ij} r_{ij} \right) \tag{7.54}$$

where γ_j is the Lagrange multiplier. As problem (7.43) is convex, its solution can be obtained from solving its KKT conditions given as follows:

$$\frac{\partial \mathcal{L}_j^{\mathsf{C}}}{\partial w_{ij}} = q_i + w_{ij} + \nu_j \sum_{i' \neq i} w_{i'j} - \gamma_j r_{ij} = 0, \ \forall i \in \mathcal{I} \tag{7.55}$$

$$w_{ij} \geq 0, \ \forall i \in \mathcal{I} \tag{7.56}$$

$$\text{constraint (7.43b)} \tag{7.57}$$

$$\gamma_j \geq 0 \tag{7.58}$$

$$\gamma_j \left(R^{\mathsf{A}} - \sum_{i=1}^{I} w_{ij} r_{ij} \right) = 0 \tag{7.59}$$

where (7.56) and (7.43b) represent the primal feasibility conditions, while (7.58) captures the dual feasibility condition. Further, (7.59) is the complementary slackness condition.

From (7.55), we have

$$w_{ij} = \gamma_j r_{ij} - q_i - \nu_j \sum_{i' \neq i} w_{i'j}, \ \forall i \in \mathcal{I}. \tag{7.60}$$

We can see that if $\gamma_j = 0$, then $w_{ij} < 0$ for all $i \in \mathcal{I}$, which is not plausible. Therefore, we must have $\gamma_j > 0$, and by the complementary slackness condition (7.59), constraint (7.43b) must be active. This confirms the first statement of Lemma 7.6. Next, by solving the set of equations in (7.55) and by (7.56), we obtain w_{ij} as shown in (7.44a). This completes the proof of Lemma 7.6.

Chapter 8

Conclusions and Future Works

8.1 Major Research Contributions

In the first contribution, we have studied the joint resource allocation and content caching problem which aims to efficiently utilize the radio and content storage resources in the VWN with highly congested backhaul links. This problem aims to minimize the maximum content request rejection rate experienced by users of different MVNOs in different cells, which results in a MINLP. We have solved this difficult optimization problem by proposing a bisection-search based algorithm that iteratively optimizes the resource allocation and content caching placement. We have further proposed a low-complexity heuristic algorithm which achieves moderate performance loss compared to the bisection-search based algorithm. Extensive numerical results have confirmed the efficacy of our proposed framework which significantly reduces the maximum request outage probability compared to other benchmark algorithms.

In the second contribution, we have proposed the caching problem for heterogeneous small-cell networks with bandwidth allocation and caching-aware BS association. The caching control and bandwidth allocation problem aims at minimizing the request miss ratio. To solve this problem, we have proposed a LSBI algorithm which determines the solution by combining the line-search algorithm to obtain the optimal bandwidth allocation with the iterative caching algorithm to acquire a caching solution. Numerical results have demonstrated that the LSBI algorithm significantly outperforms existing caching algorithms, and is on a par with a performance bound.

In the third contribution, we have studied the resource allocation and pricing problem for network slicing that captures interactions among access/backhaul service providers and their UEs by using the MLMF Stackelberg game approach. Toward this end, we have shown how to formulate such a Stackelberg game and prove the existence of a unique game equilibrium. Then, we have developed a distributed algorithm based on updating underlying best-response functions, which is proved to converge to the game equilibrium. Numerical results have been presented to provide important insights into the interactions among the involved stakeholders and demonstrate the economical efficacy of the proposed design with respect to existing benchmarks.

8.2 Future Research Directions

Our research work in this dissertation focuses on the joint resource allocation and resource trading/pricing for future wireless cellular networks employing the wireless virtualization technology. The following research directions are of importance and deserve further investigation.

8.2.1 Content Popularity Prediction

Content caching at the network edge is typically designed by leveraging content popularity information. However, the content popularity is a sporadic and time-varying statistical information. The accuracy of content popularity estimation has a great impact on the efficacy of the content caching system in terms of network traffic reduction and QoS improvement. As a result, improving content popularity prediction accuracy for mobile edge caching is an important research direction deserving further study.

8.2.2 Machine Learning-based End-to-End Slice Orchestration and Management

Recently, various frameworks have been proposed for enabling software-based network slice orchestration and management. These frameworks, however, solely focus on the flexibility and scalability of slice orchestration. More advanced frameworks need to be devised so as to improving the resource utilization, i.e., avoiding over- and under-provisioning issues, while guaranteeing the QoS of

each slice. To this end, sophisticated machine learning techniques such as reinforcement learning and deep learning are promising solutions for realizing self-optimizing network orchestration and management.

8.2.3 Mobility Management in Network Slicing

With the rapidly increasing number of smart devices and the urgent need for the 5G wireless system to support different vertical industries, mobility management has become a critical issue for 5G network slicing. Specifically, 5G network slices must be created and managed in a way to efficiently meet different characteristics and requirements with regards to mobility and latency. For example, it is necessary for network-slicing-based 5G networks to support seamless handover where highly mobile user can move from different SDN controllers and/or resource substances.

8.2.4 Security and Privacy Challenges in Network Slicing

A security concern is raised for network slicing as network slice instances share a common resource pool. Different slices serving different vertical domains my also have different security levels and privacy policy requirements. Accordingly, in allocating resource to a particular slice in virtualized 5G networks, advanced security and privacy protocols and constraints need to be taken into account.

8.3 List of Publications

8.3.1 Journals

[J1] *Thinh Duy Tran* and Long Bao Le, "Resource allocation for multi-tenant network slicing: A multi-leader multi-follower Stackelberg game approach," submitted to *IEEE Transactions on Vehicular Technology* (under second-round review).

[J2] *Thinh Duy Tran*, Tuong Duc Hoang, and Long B. Le, "Caching for heterogeneous small-cell networks with bandwidth allocation and caching-aware BS association," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 49-52, Feb. 2019.

[J3] *Thinh Duy Tran* and Long Bao Le, "Joint resource allocation and content caching in virtualized content-centric wireless networks", *IEEE Access*, vol. 6, pp. 11329–11341, 2018.

8.3.2 Conferences

[C1] *Thinh Duy Tran*, Long Bao Le, Thanh Tung Vu, and Duy Trong Ngo, "Stackelberg game-based network slicing for joint wireless access and backhaul resource allocation," In proceeding of *IEEE International Conference on Communications (IEEE ICC)*, Shanghai, China, 2019, pp. 1-7.

[C2] *Thinh Duy Tran* and Long Bao Le, "Hybrid backscatter and underlay transmissions in rf-powered cognitive radio networks," In proceeding of *26th International Conference on Telecommunications (ICT)*, Hanoi, Vietnam, 2019, pp. 11-15.

[C3] *Thinh Duy Tran* and Long Bao Le, "Joint wireless access-backhaul network slicing and content caching optimization," In proceeding of *IEEE ICC Workshop - Information-Centric Edge Computing and Caching for Future Networks (ICECC)*, Kansas city, USA, May 2018.

[C4] *Thinh Duy Tran* and Long Bao Le,"Joint resource allocation and content caching in virtualized multi-cell wireless networks," In proceeding of *IEEE Global Communications Conference (IEEE GLOBECOM)*, Singapore, December 2017.

[C5] *Thinh Duy Tran* and Long Bao Le, "Stackelberg game approach for wireless virtualization design in wireless networks," In proceeding of *IEEE International Conference on Communications (IEEE ICC)*, Paris, France, May 2017.

[C6] *Thinh Duy Tran* and Long Bao Le, "Resource allocation for efficient bandwidth provisioning in virtualized wireless networks," In proceeding of *IEEE Wireless Communications and Networking Conference (IEEE WCNC)*, San Francisco, USA, March 2017.

References

[1] S. Leyffer and T. Munson, "Solving multi-leader-common-follower games," *Optimisation Methods & Software*, vol. 25, no. 4, pp. 601–623, 2010.

[2] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5G be?," *IEEE J. Select. Areas in Commun.*, vol. 32, no. 6, pp. 1065–1082, 2014.

[3] H. Liu, Z. Chen, and L. Qian, "The three primary colors of mobile systems," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 15–21, 2016.

[4] Cisco, "Fog computing and the internet of things: Extend the cloud to where the things are." Available: `https://goo.gl/Y62JcZ`., Nov. 2015. Accessed: 2020-01-06.

[5] Qualcomm, "The 1000x mobile data challenge." Available: `https://www.qualcomm.com/invention/1000x`, 2013. Accessed: 2020-01-06.

[6] C. M. R. Institute, "C-RAN white paper: The road towards green RAN." Available: `http://labs.chinamobile.com/cran/`, 2011. Accessed: 2016-06-20.

[7] Cisco, "Cisco visual networking index: global mobile data traffic forecast update, 2016-2021." Available: `https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html`, 2017. Accessed: 2020-01-06.

[8] Cisco, "Cisco visual networking index: Forecast and trends, 2017-2022 white paper." Available: `https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.html`, 2018. Accessed: 2020-01-06.

[9] Qualcomm and NOKIA, "Making 5G a reality: Addressing the strong mobile broadband demand in 2019 and beyond." Available: `https://goo.gl/HqfG1w`, Sep. 2017. Accessed: 2020-01-06.

[10] Qualcomm, "Making 5G NR a reality." Available: `https://goo.gl/dmLvJQ`, Dec. 2016. Accessed: 2020-01-06.

[11] 3GPP, "Study on physical layer enhancements for NR ultra-reliable and low latency case (URLLC)." Available: `https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3498`, Mar. 2019. Accessed: 2020-01-06.

[12] Y. Niu, Y. Li, D. Jin, L. Su, and A. V. Vasilakos, "A survey of millimeter wave communications (mmWave) for 5G: Opportunities and challenges," *Wireless Networks*, vol. 21, pp. 2657–2676, Nov. 2015.

[13] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, "Exploiting caching and multicast for 5G wireless networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2995–3007, 2016.

[14] G. Paschos, E. Bastug, I. Land, G. Caire, and M. Debbah, "Wireless caching: Technical misconceptions and business barriers," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 16–22, 2016.

[15] A. Khreishah, J. Chakareski, and A. Gharaibeh, "Joint caching, routing, and channel assignment for collaborative small-cell cellular networks," *IEEE J. Select. Areas in Commun.*, vol. 34, no. 8, pp. 2275–2284, 2016.

[16] Z. Chen, J. Lee, T. Q. S. Quek, and M. Kountouris, "Cooperative caching and transmission design in cluster-centric small cell networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 3401–3415, 2017.

[17] S. Samarakoon, M. Bennis, W. Saad, M. Debbah, and M. Latva-aho, "Ultra dense small cell networks: Turning density into energy efficiency," *IEEE J. Select. Areas in Commun.*, vol. 34, no. 5, pp. 1267–1280, 2016.

[18] J. Zheng, Y. Wu, N. Zhang, H. Zhou, Y. Cai, and X. Shen, "Optimal power control in ultra-dense small cell networks: A game-theoretic approach," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4139–4150, 2017.

[19] Ericsson, "Cloud RAN the benefits of virtualization, centralization and coordination." Available: `https://www.ericsson.com/assets/local/publications/white-papers/wp-cloud-ran.pdf`. Accessed: 2020-01-06.

[20] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega, D. Aziz, and H. Bakker, "Network slicing to enable scalability and flexibility in 5G mobile networks," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 72–79, 2017.

[21] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5G wireless network slicing for embb, urllc, and mmtc: A communication-theoretic view," *IEEE Access*, vol. 6, pp. 55765–55779, 2018.

[22] Ericsson, "Scalable network opportunities: An economic study of 5G network slicing for iot service deployment." Available: `https://www.ericsson.com/en/digital-services/trending/economic-study-5g-network-slicing`, 2017. Accessed: 2020-01-06.

[23] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, "From network sharing to multi-tenancy: The 5G network slice broker," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 32–39, 2016.

[24] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game theory in wireless and communication networks: theory, models, and applications*. Cambridge university press, 2012.

[25] C. Liang, F. R. Yu, H. Yao, and Z. Han, "Virtual resource allocation in information-centric wireless networks with virtualization," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9902–9914, 2016.

[26] Q. Chen, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "Joint resource allocation for software defined networking, caching and computing," in *Proc. IEEE Global Commun. Conf.*, pp. 1–6, 2016.

[27] R. B. Cooper, *Introduction to Queuing Theory*. Elsevier, North Holland, 1981.

[28] A. Kumar, D. Manjunath, and J. Kuri, *Communication networking: an analytical approach*. Elsevier, 2004.

[29] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *Proc. IEEE INFOCOM*, pp. 1107–1115, March 2012.

[30] N. Karamchandani, U. Niesen, M. A. Maddah-Ali, and S. N. Diggavi, "Hierarchical coded caching," *IEEE Trans. Inform. Theory*, vol. 62, no. 6, pp. 3212–3229, 2016.

[31] J. Tang and T. Q. S. Quek, "The role of cloud computing in content-centric mobile networking," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 52–59, 2016.

[32] M. Peng, C. Wang, V. Lau, and H. V. Poor, "Fronthaul-constrained cloud radio access networks: Insights and challenges," *IEEE Wireless Commun.*, vol. 22, no. 2, pp. 152–160, 2015.

[33] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[34] P. Caballero, A. Banchs, G. de Veciana, and X. Costa-Pérez, "Network slicing games: Enabling customization in multi-tenant networks," in *Proc. IEEE INFOCOM*, pp. 1–9, 2017.

[35] S. M. A. Kazmi, N. H. Tran, T. M. Ho, and C. S. Hong, "Hierarchical matching game for service selection and resource purchasing in wireless network virtualization," *IEEE Commun. Letters*, vol. 22, no. 1, pp. 121–124, 2018.

[36] K. Zhu and E. Hossain, "Virtualization of 5G cellular networks as a hierarchical combinatorial auction," *IEEE Trans. Mobile Comput.*, vol. 15, no. 10, pp. 2640–2654, 2016.

[37] D. Zhang, Z. Chang, T. Hämäläinen, and W. Gao, "A contract-based resource allocation mechanism in wireless virtualized network," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 474–479, 2018.

[38] X. Kang, R. Zhang, and M. Motani, "Price-based resource allocation for spectrum-sharing femtocell networks: A Stackelberg game approach," *IEEE J. Select. Areas in Commun.*, vol. 30, no. 3, pp. 538–549, 2012.

[39] G. Liu, F. R. Yu, H. Ji, and V. C. M. Leung, "Virtual resource management in green cellular networks with shared full-duplex relaying and wireless virtualization: A game-based approach," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7529–7542, 2016.

[40] J. Hu, Z. Zheng, B. Di, and L. Song, "Tri-level Stackelberg game for resource allocation in radio access network slicing," in *Proc. IEEE Global Commun. Conf.*, pp. 1–6, Dec 2018.

[41] D. Niyato and E. Hossain, "Competitive pricing for spectrum sharing in cognitive radio networks: Dynamic game, inefficiency of nash equilibrium, and collusion," *IEEE J. Select. Areas in Commun.*, vol. 26, no. 1, pp. 192–202, 2008.

[42] T. D. Tran and L. B. Le, "Stackelberg game approach for wireless virtualization design in wireless networks," in *Proc. IEEE Int. Conf. Commun.*, pp. 1–6, May 2017.

[43] T. M. Ho, N. H. Tran, L. B. Le, W. Saad, S. M. A. Kazmi, and C. S. Hong, "Coordinated resource partitioning and data offloading in wireless heterogeneous networks," *IEEE Commun. Letters*, vol. 20, no. 5, pp. 974–977, 2016.

[44] D. Yue and F. You, "Game-theoretic modeling and optimization of multi-echelon supply chain design and operation under stackelberg game and market equilibrium," *Computers and Chemical Engineering*, vol. 71, pp. 347–361, 2014.

[45] S. Alaei, R. Alaei, and M. Behravesh, "A theoritical game approach for two echelon stochastic inventory system," *Acta Polytechnica Hungarica*, vol. 12, no. 4, 2015.

[46] S. Kim, "Multi-leader multi-follower stackelberg model for cognitive radio spectrum sharing scheme," *Comput. Netw.*, vol. 56, pp. 3682–3692, Nov. 2012.

[47] Y. Mao, T. Cheng, H. Zhao, and N. Shen, "Multiple-seller and multiple-buyer spectrum sharing model in cognitive radio-based wireless sensor network," *Int J Distrib Sens Netw*, vol. 13, no. 11, p. 1550147717742888, 2017.

[48] K. Wang, F. C. M. Lau, L. Chen, and R. Schober, "Pricing mobile data offloading: A distributed market framework," *IEEE Trans. Wireless Commun.*, vol. 15, no. 2, pp. 913–927, 2016.

[49] H. Zhang, Y. Xiao, L. X. Cai, D. Niyato, L. Song, and Z. Han, "A multi-leader multi-follower Stackelberg game for resource management in LTE unlicensed," *IEEE Trans. Wireless Commun.*, vol. 16, no. 1, pp. 348–361, 2017.

[50] D. B. Rawat, A. Alshaikhi, A. Alshammari, C. Bajracharya, and M. Song, "Payoff optimization through wireless network virtualization for IoT applications: A three layer game approach," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2797–2805, 2019.

[51] D. Fudenberg and J. Tirole, *Game Theory.* Cambridge, MA: MIT Press, 1991.

[52] R. D. Yates, "A framework for uplink power control in cellular radio systems," *IEEE J. Select. Areas in Commun.*, vol. 13, no. 7, pp. 1341–1347, 1995.

[53] Chi Wan Sung and Kin-Kwong Leung, "A generalized framework for distributed power control in wireless networks," *IEEE Trans. Inform. Theory*, vol. 51, no. 7, pp. 2625–2635, 2005.

[54] 3GPP, "Dual connectivity (release 16)." Available: `http://www.3gpp.org/DynaReport/FeatureOrStudyItemFile-800058.htm`, 2019. Accessed: 2019-10-15.

[55] J. Lianghai, A. Weinand, B. Han, and H. D. Schotten, "Multi-RATs support to improve V2X communication," in *Proc. IEEE Wireless Commun. and Networking. Conf.*, pp. 1–6, April 2018.

[56] V. Petrov, D. Solomitckii, A. Samuylov, M. A. Lema, M. Gapeyenko, D. Moltchanov, S. Andreev, V. Naumov, K. Samouylov, M. Dohler, and Y. Koucheryavy, "Dynamic multi-connectivity performance in ultra-dense urban mmWave deployments," *IEEE J. Select. Areas in Commun.*, vol. 35, no. 9, pp. 2038–2055, 2017.

[57] X. Ba, Y. Wang, H. Hai, Y. Chen, and Z. Liu, "Performance comparison of multi-connectivity with comp in 5G ultra-dense network," in *Proc. IEEE Veh. Technol. Conf.*, (Porto, Portugal), pp. 1–5, 2018.

[58] D. López-Pérez, A. García-Rodríguez, L. G. Giordano, M. Kasslin, and K. Doppler, "IEEE 802.11be - extremely high throughput: The next generation of wi-fi technology beyond 802.11ax," *CoRR*, vol. abs/1902.04320, 2019.

[59] L. Cariou, "802.11 EHT proposed PAR," report, IEEE 802 Study Group, 2019. Accessed: 2019-10-15.

[60] 3GPP, "Study on access traffic steering, switch and splitting support in the 5G System (5GS) architecture Version 16.0.0." Available: `https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3254`, 2018. Accessed: 2019-10-15.

[61] S. Zhang, N. Zhang, S. Zhou, J. Gong, Z. Niu, and X. Shen, "Energy-sustainable traffic steering for 5G mobile networks," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 54–60, 2017.

[62] A. Prasad, F. S. Moya, M. Ericson, R. Fantini, and O. Bulakci, "Enabling RAN moderation and dynamic traffic steering in 5G," in *Proc. IEEE Veh. Technol. Conf.*, pp. 1–6, Sep. 2016.

[63] N. Zhang, S. Zhang, S. Wu, J. Ren, J. W. Mark, and X. Shen, "Beyond coexistence: Traffic steering in LTE networks with unlicensed bands," *IEEE Wireless Commun.*, vol. 23, no. 6, pp. 40–46, 2016.

[64] N. Akhtar, I. Matta, A. Raza, L. Goratti, T. Braun, and F. Esposito, "Virtual function placement and traffic steering over 5G multi-technology networks," in *4th IEEE Conference on Network Softwarization and Workshops, NetSoft 2018, Montreal, QC, Canada, June 25-29, 2018*, pp. 114–122, 2018.

[65] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP extensions for multipath operation with multiple addresses." Available: `https://www.rfc-editor.org/info/rfc6824`, Jan. 2013. Accessed: 2019-10-15.

[66] O. Bonaventure, M. Boucadair, S. Gundavelli, S. Seo, and B. Hesmans, "0-RTT TCP convert protocol." Available: `https://datatracker.ietf.org/doc/html/draft-ietf-tcpm-converters-10`, 2019. Accessed: 2019-10-15.

[67] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, pp. 2429–2453, Third quarter 2018.

[68] T. D. Tran and L. B. Le, "Joint resource allocation and content caching in virtualized content-centric wireless networks," *IEEE Access*, vol. 6, pp. 11329–11341, 2018.

[69] T. D. Tran, T. D. Hoang, and L. B. Le, "Caching for heterogeneous small-cell networks with bandwidth allocation and caching-aware bs association," *IEEE Wireless Commun. Letters*, vol. 8, no. 1, pp. 49–52, 2019.

[70] T. D. Tran and L. B. Le, "Resource allocation for multi-tenant network slicing: A multi-leader multi-follower Stackelberg game approach." submitted to IEEE Trans. Veh. Technol.

[71] T. D. Tran and L. B. Le, "Resource allocation for efficient bandwidth provisioning in virtualized wireless networks," in *Proc. IEEE Wireless Commun. and Networking. Conf.*, pp. 1–6, March 2017.

[72] T. D. Tran and L. B. Le, "Joint resource allocation and content caching in virtualized multi-cell wireless networks," in *Proc. IEEE Global Commun. Conf.*, pp. 1–6, Dec 2017.

[73] T. D. Tran and L. B. Le, "Joint wireless access-backhaul network slicing and content caching optimization," in *Proc. IEEE Int. Conf. Commun.*, pp. 1–6, May 2018.

[74] T. D. Tran, L. B. Le, T. T. Vu, and D. T. Ngo, "Stackelberg game-based network slicing for joint wireless access and backhaul resource allocation," in *Proc. IEEE Int. Conf. Commun.*, pp. 1–7, May 2019.

[75] T. D. Tran and L. B. Le, "Hybrid backscatter and underlay transmissions in RF-powered cognitive radio networks," in *2019 26th International Conference on Telecommunications (ICT)*, pp. 11–15, Apr. 2019.

[76] GSMA, "5G spectrum - GSMA public policy position." Available: `https://www.gsma.com/spectrum/wp-content/uploads/2019/09/5G-Spectrum-Positions.pdf`, July 2019. Accessed: 2020-01-06.

[77] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, "Millimeter wave mobile communications for 5G cellular: It will work!," *IEEE Access*, vol. 1, pp. 335–349, 2013.

[78] M. Kamel, W. Hamouda, and A. Youssef, "Ultra-dense networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, pp. 2522–2545, Fourth quarter 2016.

[79] R. Wang, H. Hu, and X. Yang, "Potentials and challenges of C-RAN supporting multi-RATs toward 5G mobile networks," *IEEE Access*, vol. 2, pp. 1187–1195, 2014.

[80] L. Diez, A. Garcia-Saavedra, V. Valls, X. Li, X. Costa-Perez, and R. Agüero, "LaSR: A supple multi-connectivity scheduler for multi-RAT OFDMA systems," *IEEE Trans. Mobile Comput.*, vol. 19, no. 3, pp. 624–639, 2020.

[81] C. Pan, H. Zhu, N. J. Gomes, and J. Wang, "Joint precoding and RRH selection for user-centric green MIMO C-RAN," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 2891–2906, 2017.

[82] A. Maeder, A. Ali, A. Bedekar, A. F. Cattoni, D. Chandramouli, S. Chandrashekar, L. Du, M. Hesse, C. Sartori, and S. Turtinen, "A scalable and flexible radio access network architecture for fifth generation mobile networks," *IEEE Commun. Mag.*, vol. 54, no. 11, pp. 16–23, 2016.

[83] U. Siddique, H. Tabassum, E. Hossain, and D. I. Kim, "Wireless backhauling of 5g small cells: challenges and solution approaches," *IEEE Wireless Commun.*, vol. 22, no. 5, pp. 22–31, 2015.

[84] M. Gregori, J. Gómez-Vilardebó, J. Matamoros, and D. Gündüz, "Wireless content caching for small cell and D2D networks," *IEEE J. Select. Areas in Commun.*, vol. 34, no. 5, pp. 1222–1234, 2016.

[85] J. Zhao, T. Q. S. Quek, and Z. Lei, "Heterogeneous cellular networks using wireless backhaul: Fast admission control and large system analysis," *IEEE J. Select. Areas in Commun.*, vol. 33, no. 10, pp. 2128–2143, 2015.

[86] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inform. Theory*, vol. 59, no. 12, pp. 8402–8413, 2013.

[87] M. Ji, G. Caire, and A. F. Molisch, "Wireless device-to-device caching networks: Basic principles and system performance," *IEEE J. Select. Areas in Commun.*, vol. 34, no. 1, pp. 176–189, 2016.

[88] Y. Sun, M. Peng, S. Mao, and S. Yan, "Hierarchical radio resource allocation for network slicing in fog radio access networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3866–3881, 2019.

[89] O. Narmanlioglu and E. Zeydan, "Learning in SDN-based multi-tenant cellular networks: A game-theoretic perspective," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 929–934, 2017.

[90] Y. K. Tun, N. H. Tran, D. T. Ngo, S. R. Pandey, Z. Han, and C. S. Hong, "Wireless network slicing: Generalized kelly mechanism-based resource allocation," *IEEE J. Select. Areas in Commun.*, vol. 37, no. 8, pp. 1794–1807, 2019.

[91] M. Jiang, M. Condoluci, and T. Mahmoodi, "Network slicing in 5G: An auction-based model," in *Proc. IEEE Int. Conf. Commun.*, pp. 1–6, 2017.

[92] H. Pinto, J. M. Almeida, and M. A. Gonçalves, "Using early view patterns to predict the popularity of Youtube videos," in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, (New York, NY, USA), p. 365–374, Association for Computing Machinery, 2013.

[93] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon, "Analyzing the video popularity characteristics of large-scale user generated content systems," *IEEE/ACM Trans. Netw.*, vol. 17, no. 5, pp. 1357–1370, 2009.

[94] S. Meier, B. Virun, J. Blumert, and M. T. Jones, "IBM systems virtualization: Servers, storage, and software," *IBM Redbook*, 2008.

[95] R. P. Goldberg, "Survey of virtual machine research," *Computer*, vol. 7, no. 6, pp. 34–45, 1974.

[96] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[97] N. Alliance, "MGMN 5G white paper." Available: `https://www.ngmn.org/wp-content/uploads/NGMN_5G_White_Paper_V1_0.pdf`, 2015. Accessed: 2020-01-06.

[98] 3GPP, "Study on architecture for next generation system, release 14." Available: `https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3008`, Dec. 2016. Accessed: 2020-01-06.

[99] 3GPP, "Framework of network virtualization for future networks, next generation network — future networks." Available: `https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-Y.3011-201201-I!!PDF-E&type=items`, Jan. 2012. Accessed: 2020-02-13.

[100] P. Hedman, "Decription of network slicing concept (NGMN 5G p1)," *NGMN (Next Generation Mobile Networks) Alliance*, 2016.

[101] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 17, pp. 27–51, First quarter 2015.

[102] ONF, "SDN architecture overview." Available: `https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN-ARCH-Overview-1.1-11112014.02.pdf`, Nov. 2014. Accessed: 2020-02-13.

[103] ETSI, "Network functions virtualisation: An introduction, benefits, enablers, challenges & call for action. issue 1." Available: `https://portal.etsi.org/NFV/NFV_White_Paper.pdf`, Oct. 2012. Accessed: 2020-01-06.

[104] J. d. J. Gil Herrera and J. F. Botero Vega, "Network functions virtualization: A survey," *IEEE Latin America Transactions*, vol. 14, no. 2, pp. 983–997, 2016.

[105] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "SoftRAN: Software defined radio access network," in *Proc. ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, p. 25–30, 2013.

[106] F. K. Jondral, "Software-defined radio—basics and evolution to cognitive radio," *EURASIP J Wirel Commun Netw*, vol. 2005, no. 3, 2005.

[107] K. R. Krishnan, "The convexity of loss rate in an Erlang loss system and sojourn in an erlang delay system with respect to arrival and service rates," *IEEE Trans. Commun.*, vol. 38, no. 9, pp. 1314–1316, 1990.

[108] E. Hossain, M. Rasti, and L. B. Le, *Radio Resource Management in Wireless Networks: An Engineering Approach*. Cambridge University Press, 2017.

[109] G. Zeng, "Two common properties of the Erlang-B function, Erlang-C function, and Engset blocking function," *Mathematical and Computer Modelling*, vol. 37, no. 12, pp. 1287–1296, 2003.

[110] D. L. Jagerman, "Methods in traffic calculations," *AT&T Bell Laboratories Technical Journal*, vol. 63, no. 7, pp. 1283–1310, 1984.

[111] CVX, "CVX: Matlab software for disciplined convex programming." Availabe: `http://cvxr.com/cvx/`.

[112] I. Gurobi Optimization, "Gurobi optimizer reference manual." Availabe: `http://www.gurobi.com`, 2016.

[113] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM J Optim*, vol. 23, no. 2, pp. 1126–1153, 2013.

[114] Ericsson, "Ericsson mobility report." Available: `https://www.ericsson.com/en/digital-services/trending/economic-study-5g-network-slicing`, June 2019. Accessed: 2019-10-15.

[115] Ericsson, "Fixed wireless access handbook." Available: `https://www.ericsson.com/assets/local/networks/documents/fwa-handbook-june-2019.pdf?_ga=2.35318625.1745410969.1569518187-1074726230.1566927018`, June 2019. Accessed: 2019-10-15.

[116] N. Singh and X. Vives, "Price and quantity competition in a differentiated duopoly," *The RAND Journal of Economics*, vol. 15, no. 4, pp. 546–554, 1984.

[117] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, "A double-auction mechanism for mobile data-offloading markets," *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1634–1647, 2015.

[118] T. M. Ho, N. H. Tran, L. Le, Z. Han, S. M. A. Kazmi, and C. S. Hong, "Network virtualization with energy efficiency optimization for wireless heterogeneous networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 10, pp. 2386–2400, 2018.

[119] G. Arslan, M. F. Demirkol, and S. Yüksel, "On games with coupled constraints," *IEEE Trans. Autom. Control*, vol. 60, no. 2, pp. 358–372, 2015.

[120] C.-Y. Huang and R. D. Yates, "Rate of convergence for minimum power assignment algorithms in cellular radio systems," *Wireless Networks*, vol. 4, no. 3, pp. 223–231, 1998.

[121] H. R. Feyzmahdavian, M. Johansson, and T. Charalambous, "Contractive interference functions and rates of convergence of distributed power control laws," *IEEE Trans. Wireless Commun.*, vol. 11, no. 12, pp. 4494–4502, 2012.

[122] C. E. Gittings and S. Brierley, *The Advertising Handbook*. USA: Routledge, 2002.

[123] J. S. McGee, "Predatory price cutting: The standard oil (n. j.) case," *The Journal of Law and Economics*, vol. 1, pp. 137–169, 1958.

[124] P. Bolton, J. F. Brodley, and M. H. Riordan, "Predatory pricing: Strategic theory and legal policy," *Georgetown Law Review*, vol. 88, p. 2239, 1999.