# 1 Economic Ecosystems in Elastic Wireless Edge Caching

Jeongho Kwak, Georgios Paschos and George Iosifidis

The delivery of content over the Internet is a multi-billion business involving different stakeholders: the content providers (CPs) which create and sell content to users; the Content Distribution Networks (CDNs) that manage large-scale content cache servers, and the Internet Service Providers (ISPs) or mobile network operators (MNOs) which are responsible for transferring the content to the end request points. In the past, the economic interactions among these entities were already complex; yet as content caches are being placed closer to end users following advances in Software-Defined Networking (SDN) and Network Function Virtualization (NFV) technologies, these interactions are becoming even more intertwined. Apart from sorting out the individual goals of each stakeholder, and finding ways to foster mutually beneficial collaboration, the economic interactions also include the interesting issue of the pricing of content caching and bandwidth leasing.

In this chapter, we study the leasing of storage resources in a wireless cloud infrastructure. The owner of storage advertises fluctuating prices, and small-sized content providers lease storage resources with the aim to improve the quality of their offered service. Not only such a scenario includes a stochastic inventory problem of meeting fluctuating video traffic demand with appropriate storage investments, but further includes complicated caching interactions due to proximity of user demand to available stored content, i.e., the closer is a user to a stored content, the higher the offered quality. We begin by motivating the economic ecosystem of elastic CDN service, including the arising business models and the elasticity of the storage resource. After we survey related works regarding economic studies of caching, we proceed by outlining the particularities of the economic ecosystem in the content caching framework, explaining in detail vertical and horizontal relations between various stakeholders in the ecosystem. In the last section of this chapter, we provide an elastic cache dimensioning, content caching and request-SBS (small BS) association problem where a CP has a limited average budget for operating costs of cache investment and a Telco CDN operator provides an elastic cache lease service. We study two possible elastic solutions for different scenarios using the Lyapunov drift-minus-benefit technique. Finally, we quantify the benefit of the elastic cache lease over the static cache lease, and provide guidelines for smart pricing.

**Table 1.1** List of abbreviations

| Abbreviation | Full name | Abbreviation | Full name |
|---|---|---|---|
| CP | Content Provider | CDN | Content Delivery Network |
| CSP | Cloud Service Provider | ISP | Internet Service Provider |
| MNO | Mobile Network Operator | Telco CDN | CDN of Telecommunication Operator |
| BS | Base Station | NFV | Network Function Virtualization |
| SDN | Software Defined Networking | SBS | Small Base Station |
| MBS | Macro Base Station | D2D | Device-to-Device |
| QoS | Quality of Service | CDNaaS | CDN as-a-Service |

## 1.1 Introduction

As demand for mobile data has been increasing exponentially in recent years [1], and Internet pipes are becoming increasingly congested by video traffic, a great deal of research has been focusing on content caching infrastructures. Since most efforts were focused on improving the efficiency of caching, the economic aspects of the operation of these large systems have been largely overlooked. Fig. 1.1 depicts just how different is the content delivery process from an economic perspective of the wireless edge caching. From a technical standpoint, the caching decisions at every edge server are tuned to maximize the caching hit ratio and/or reduce the end-to-end delay. In contrast, from an economic point of view, key players (stakeholders) such as content providers (CPs), Internet operators, and users, all have different goals, which results in delicate tradeoffs involving tuning the caches, pricing of caching service, fairness, and collaboration in traffic steering. The focus of this chapter is precisely to highlight these tradeoffs as they arise in the developing ecosystem of network virtualization, where storage will become an important commodity.

Although memory is cheap with respect to other resources [2], the total amount of installed memory in a mobile network can be significant [3]. Additionally, the edge cloud resources (e.g., such as those offered by Amazon Web Service (AWS)) are typically more expensive than those in core cloud data centers. For example, the service price of AWS CloudFront for transferring files to the edge servers is \$0.085/GB whereas that of AWS Simple Storage Service (S3) to transfer files to the cloud data center is \$0.023/GB by 10TB [4], which implies that the rental price of the edge cache storage is approximately four times more expensive than that of cloud storage. On the other hand, caching at the edge reduces backhaul utilization, known to be the performance bottleneck of dense wireless deployments. In particular, due to memory commoditization and its necessity for transferring video traffic, a market of memory is envisaged in the

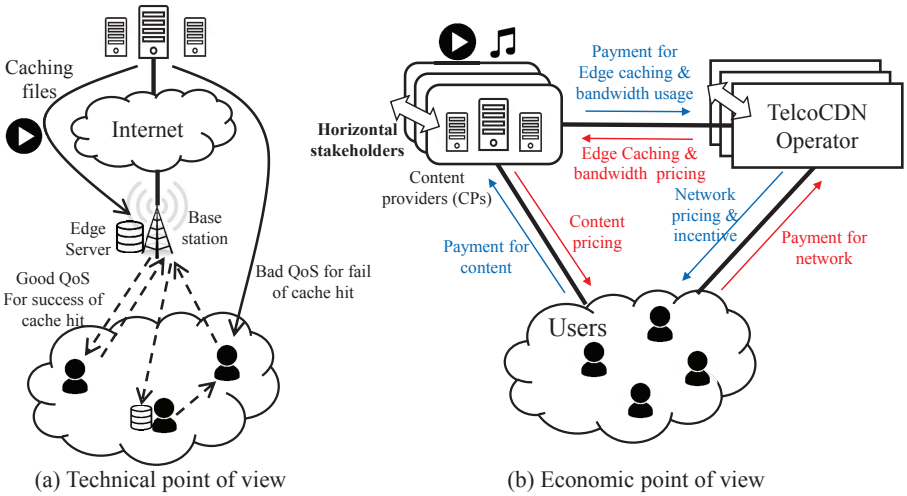



**Figure 1.1** Technical and economic perspectives of wireless edge caching: From the economic point of view, vertical stakeholders represent different types of entities such as users, CPs, ISPs, CDN providers and so on, and horizontal stakeholders represent the same type of entities, e.g., the same type of users or the same type of CPs. In general, the issues between the vertical stakeholders are pricing for the caching/networking service and collaboration between them whereas the issues between the horizontal stakeholders are making service fairness for the same service price.

near future. All the above suggest that decisions about installing/leasing wireless caches mandate a careful cost analysis [5] and respective pricing.

When it comes to moving massive amounts of video within the Internet, a number of stakeholders are involved. For example, the CPs (such as YouTube, Netflix, Facebook, and more recently Amazon) are responsible for producing content and selling content services, the telecom providers (Telcos) are responsible for delivering Internet traffic, and the users are typical consumers of the video services. However, in the evolving ecosystem of caching we encounter some more complex roles [6]. Companies such as Akamai and Limelite (called CDN providers) offer CDN as a service (CDNaaS), i.e., they rent their private networks and storages to facilitate the transfer of videos of clients. Telcos often choose to build their own CDN (known as Telco CDN[1]) in order to avoid buying CDN service. On the other hand, since CPs own the encryption rights to the video content, many times they install their own caching boxes within the Telco transport networks. Users may produce their own content (e.g. Youtube or Facebook) or choose to circulate the videos themselves (e.g. peer-to-peer networks). Caching plays a dominant role gluing together all these stakeholders, and generating multiple different business models. Perhaps the most traditional business model is the one where the Telco manages the caches and sells the CDN service

[1]Telco CDN = the CDN network of a telecommunication operator.

to the CPs. However, in this chapter, we are interested in a fresh and more interesting business model, that of *elastic CDN service*. In this business model, there are three stakeholders 1) the storage infrastructure owner (a prominent example of which is Amazon AWS ElastiCache [4]) or a Telco CDN, 2) a small-sized CP (which can not afford a private CDN), and 3) the users that generate the video demand. The CP will purchase storage on demand, and serve the requesting users. The service is called elastic because the storage can be leased in different amounts from hour to hour, allowing the CP to adjust the cache sizes according to the real-time video traffic characteristics. This provides unprecedented levels of flexibility and economization but complicates the optimization and the business interactions, which are the focal point of this chapter.

Resource elasticity and flexibility is dogmatized in modern networks. Enabled by virtualization technologies [7], cloud companies offer elastic-*anything*, i.e., AWS provides a variety of options for flexible services such as Amazon EC2 Auto Scaling, Elastic Load Balancing, Elastic File System, ElastiCache, etc [4]. Similarly, Akamai recently proposed the idea of *elastic* CDN (or cloud CDN) where storage resources are dynamically tuned to realize virtual caches [8, 9]. The arising elastic CDN service, however, not only aligns with the modernization trend, but provides two significant tangible advantages: *(i)* it allows to meet spatio-temporal demand fluctuations by installing caches where and when needed (just-in-time caching), and *(ii)* it allows small-size CPs (such as Pinterest, Snapchat, and Tumbler) to reach the market at small entry costs. Traditional CDN pricing does reflect storage usage costs or small-scale traffic fluctuations, but is rather based on a flat-rate service price across large geographical regions (e.g., continents) arranged in long-term contracts (e.g., few months to years) that reflect the traffic peaks [10]. Therefore, small-sized CPs, faced with bursty and unpredictable demand, they must either invest in building a large private CDN, or buy such flat contracts, both of which create a *barrier-to-entry* the content business. In contrast, the idea of elastic CDN service allows them to exploit opportunism and improve content delivery with a flexible pricing scheme. Indeed the first market solutions such as Akamai Aura [11] and Huawei uCDN [12] allow dynamic cache scaling and fine-grained *pay-as-you-go* service [13]. *However, due to the complicated nature of caching resource allocation and performance, pricing elastic CDN service seems to be a very complicated engineering-oriented task.* In this chapter, we propose a mathematical model towards addressing this challenge.

### 1.1.1 Summary of this chapter

In Section 1.2 of this chapter, we survey past works related to caching economics. Then in Section 1.3, we explain the complex interactions among various stakeholders that make up the network infrastructure for caching services and specifically categorize them into vertical and horizontal stakeholders. We will

also explain the differences between core in-network caching and wireless edge caching, as they affect the economic perspective.

In Section 1.4, we study the underlying techno-economic problem of the elastic wireless caching and provide possible solutions how to invest an available budget in cache space in order to match spatio-temporal fluctuations of content demand and storage price. Specifically, we consider joint *dynamic cache rental, content placement, and request-cache association* in a wireless scenario where the Telco CDN operator offers just-in-time CDN service to the CP. To solve this challenging control problem without knowledge of distant future content popularity and market prices, a Lyapunov drift-minus-benefit technique, which results in an instantaneous optimization problem which must be solved in a slot-by-slot fashion. We provide solutions for both non-overlapping caching coverage (simpler case) and the general overlapping case. We show that with such a weaponry of stochastic control, the system operator can exploit the flexibility of elastic CDN to improve the system benefits two times over the traditional static CDN infrastructures. The above in-depth techno-economic analysis provides a solid ground for embarking on sophisticated pricing schemes that can be mutually beneficial for all stakeholders in this ecosystem.

## 1.2 Background

Before we delve deeper into the economics of wireless edge caching, we discuss how existing literature is shaped on the topics of wireless edge caching and techno-economical caching policies, and which are the prevailing business models for content caching today.

**Wireless edge caching in heterogeneous networks.** Since the seminal work on femtocaching [14] which optimizes file caching in SBSs with capacity-limited backhaul links, there have been several wireless edge caching studies [15, 16, 17, 18]. Combes *et al.* [15] showed that device-to-device communications with a help of caching in the mobile devices make a scaling law of per-user throughput regardless of the number of users. In addition, content caching/cache dimensioning was optimized in conjunction with advanced wireless technologies such as CoMP transmission [18]. Moreover, to capture the large-scale wireless caching system, Blaszczyszyn *et al.* [19] used stochastic geometry, which modeled the location of users and base stations (BSs) with a Poisson point process. Caching in wireless edge nodes such as small BSs (SBSs) or mobile devices has different features with caching in wired core network: the file demand per edge node (i) has a smaller volume, and (ii) varies quickly due to the user mobility. Hence, it is more difficult to design optimal caching policies for such systems. The above works addressed this problem when the caches are *pre-dimensioned*, however, re-formulating this problem with the ability to re-size the caches is challenging.

**Techno-economical content caching and delivery techniques.** There are extensive studies on CDN server placement problem considering the cost of the cache memory [20, 21, 22, 23]. For example, Bektas *et al.* [20] formulated the joint problem of server deployment, file placement, and data transfer cost minimization, and solved it using Benders' decomposition, while Li *et al.* [22] used dynamic programming. Unlike server placement problems, Laoutaris *et al.* [23] formulated a storage budget allocation problem which aims to minimize the sum content retrieving costs in a hierarchical file distributed system, and proposed heuristic solutions.

Past works have proposed cooperative mechanisms between ISP, CDN operator, CP and end users as their economic relations are tightly intertwined with respect to the business models. Li *et al.* [24] studied how benefits from collaboration, e.g., joint routing and content caching are divided between entities using Nash bargaining solution. In addition, Poularakis *et al.* [25] proposed incentive-based collaboration among devices to improve caching performance via D2D (Device-to-Device) communications. More recently, various economic models of content caching framework have been proposed. For example, Krolikowski *et al.* [26] modeled an economic scenario in which Mobile Network Operator (MNO) leases its edge cache to CP to maximize cache hit ratio. Based on this models, they optimized CP's decision conditioned on the user association policy. The work in this chapter is the state-of-the-art approach to address economic aspects in the elastic wireless CDN environment.

**Business models for content caching.** *1) Akamai Intelligent Platform:* Akamai operates one of the largest CDNs offering 20% of today's Internet traffic. They have the 0.2 million caching servers distributed over the world offering 1-10 *msec* access to content around the world. In addition, Akamai and Jupiter recently proposed the idea of cloud or *elastic* CDN where storage resources are dynamically adapted to meet demand [9]. This architecture combines storage deployment with caching decisions; hence it makes imperative the efficient design of joint storage allocation and content caching policies and also gives rise to new business models for content caching. *2) Google Global Cache:* The Google Global Cache (GCC) system consists of caches installed at the ISP premises. The GCC's goal is to reduce local network bandwidth costs by providing local requests for YouTube contents [27]. The importance of GCC motivates the study of peering relations between content providers and network operators and the design of pricing models for leasing in-network caching capacity at network operators. *3) Netflix Open Connect:* The Netflix CDN is partially deployed within ISPs [28]. However, Netflix video caching faces different challenges from YouTube, mainly because its catalog is much smaller and the file popularity is more predictable. *4) Amazon AWS:* Some parts of AWS are the Amazon CloudFront, a virtual CDN that leverages the cloud storage to provide CDN services. The price to store 1TB is $20 [4] and Amazon allows one to dynamically rent caching and bandwidth resources by changing the storage capacity every one hour. *5) Cedexis and Conviva:* Until recently, major CPs contracted with a single CDN, such as Akamai,

Level 3 or Amazon CloudFront or deployed their own CDN, such as Google and Netflix. The recent rise of CDN management, namely *broker* services, such as Cedexis [29] or Conviva [30] has allowed the CP to contract with multiple CDNs for easier delivery of content [10].

## 1.3 Economic Ecosystems for Wireless Edge Caching

In this section, we study an economic ecosystem consisting of various vertical and horizontal stakeholders and features of wireless edge caching compared to in-network caching.

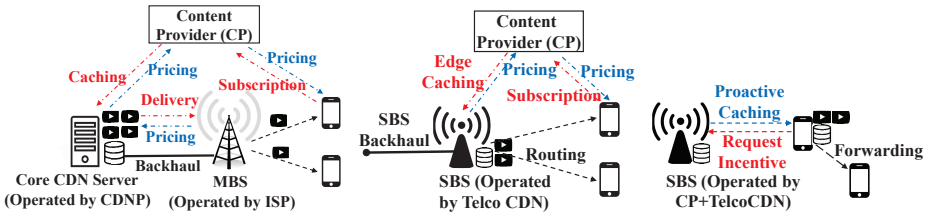
### 1.3.1 Wireless Edge Caching versus In-Network Caching

First, we study the differences between the classical in-network caching at the core versus the wireless edge caching as they affect the economic perspective. Their differences can be summarized as follows.

- The population of users reaching a cache of a BS (or edge server) is significantly smaller than that of the core network cache; correspondingly, the number of requests per unit time is also smaller, and thus timely collection of content statistics is challenging.
- Contrary to data centers, edge servers are restricted in size and have limited storage resources; hence the cost for leasing one unit of edge storage is higher.
- Wireless edge caching systems are more dynamic in nature, due to user mobility, wireless channel state variations, and multi-access coverage.

The above raise the following technical challenges particular to wireless edge caching: (a) popularity prediction is difficult, and hence caching efficiency might be compromised, especially on rapidly (second-level) fluctuating popularities, (b) the locality of storage is of high importance, and instantiating the cache at the right geographical point might allow to reach suffering users and yield a significant improvement on perceived quality, (c) storage investment can reduce backhaul link utilization (the main limitation of dense wireless architectures) but (d) the costs of leasing edge storage is very high, hence deciding an online cache sizing that addresses all these challenges is very important.

To overcome these challenges, a collaboration among stakeholders is required. Since Telcos do not have access to accurate popularity statistics, CPs must collect data at aggregate locations and offer this information to the cache management. Another proposal observes that end users often have resilient needs and can be satisfied with similar content using an appropriate incentive mechanism, in which case the system can increase the *Soft Cache hits* [31], i.e. the times user is satisfied without the delivery of the requested content. This is consistent with the recently proposed idea, which leverages recommendation systems embedded




(a) Core In-network Caching. (b) Wireless Edge Caching. (c) D2D Caching.

**Figure 1.2** Different economic scenarios for different locations of caching files. (a): File caching at the core CDN servers; (b): File caching at small cell base stations; (c): Proactive file caching at the user device and device-to-device (D2D) content delivery.

in multiple CDNs (e.g., YouTube) in order to tailor user demand towards already cached content [32]. Finally, an efficient edge cache management is possible by dynamically investing budget to the edge caches with respect to dynamic network environments including content demand, wireless channel states, and pricing for wireless bandwidth.

### 1.3.2 Economic Ecosystems for Wireless Edge Caching

As the caching ecosystem becomes more complex, it is essential to adjust the interests of the key stakeholders to alleviate market inefficiencies. In this subsection, we study economic interactions among various stakeholders, classifying them into vertical and horizontal stakeholders.

**Vertical stakeholder.** These are stakeholders that perform different duties inside the system and naturally have different individual goals, e.g., users, CPs, Telcos, and CDN providers. These stakeholders need additional efforts to reach an agreement on the globally optimal caching strategy. First, we must check how the *collaboration between them* is actually mutually beneficial and then how the benefits can be distributed. An example of vertical collaboration is when the CPs offer popularity statistics data to Telcos, or when the users offer content recommendations to CDN providers.

An important tool for effective collaboration is the pricing mechanisms. Examples of existing pricing mechanism include payments from CPs to CDN providers for high quality content delivery (e.g. from Facebook to Akamai), from CDN providers to ISPs for bandwidth, from CPs to cloud infrastructure owners for in-network storage, from users to ISPs for bandwidth and in-network storage, and from users to CPs and ISPs for subscription of the services as shown in Fig. 1.1(b). On the other hand, there are cases where payments are not needed, e.g., in case of Telco CDN where ISPs and MNOs directly manage cache memory in core network or edge servers.[2]

Fig. 1.2 shows various economic scenarios for different locations of cached files.

[2]Indeed, from the perspective of a CP, traditional CDN and telco CDN are not competitive, but complementary services. A telco CDN has the advantage of proximity to end users, but

Specifically, Fig. 1.2(a) shows traditional economic relations among a CP, a CDN operator, an ISP and end users in core in-network caching. In this cache scenario, the CP makes a payment to the CDN operator to cache their files and determines prices to provide services to end users. Moreover, the ISP can be an intermediator for content delivery between CDN operator and end users. However, the collaboration between CDN and ISP, e.g., Telco CDN would be imperative in the wireless edge caching scenario since edge caching and wireless routing, e.g., user scheduling must be jointly designed [14]. The collaboration between CP and Telco CDN can be considered for D2D caching scenario since the incentives may be also offered from the federation of CP and Telco CDN to users in order to assist the network. For example, Tadrous *et al.* [33] discussed the problem of incentivizing users to exchange content by leveraging D2D communications at the off-peak time through optimized discounts. In this framework, users can submit their content requests in advance with a discounted price so as to assist the network in serving them proactively.

**Horizontal stakeholders.** Horizontal stakeholders are entities with similar goals, that compete in a common market. In this case, the notion of fairness becomes more important. When multiple CPs use the same CDN service, the CPs would expect the same caching hit performance for the same CDN service price. Since this metric is difficult to be achieved by the CDN provider, an alternative is to use an intelligent pricing methodology. For example, Gourdin *et al.* [34] modeled the economic interaction between multiple CPs and a single CDN operator with Stackelberg game, where the leader is the CDN provider and the follower is the CPs, and investigated an impact of the CDN economic behavior on the quality perceived by users and on the fairness among CPs when the CDN charges them different prices.

Prior work [35] proposed a cooperation mechanism with multiple CDN providers, based on the Nash Bargaining solution. For example, Twitter splits its content delivery across three separate CDNs. Such MultiCDN approach can be realized thanks to the virtualization and sharing of storage resources, e.g., different CDNs can jointly deploy and manage edge servers. Further, it was shown that the cooperation benefits were dispersed proportionally to the performance that each entity would have achieved under non-cooperation. Moreover, CPs are shifting delivery from a single CDN to multiple CDNs through the use of a content broker. Mukerjee *et al.* [10] addressed management issues among multiple CDN operators. The recent rise of CDN management services (*brokers*), such as Cedexis, Conviva or NicePeopleAtWork, and CDN federation techniques [36] has made it easier for CPs to enlist multiple CDNs to deliver content. They argue that CDN operator-broker collaboration is easier to achieve, as there are significantly fewer CDNs than ISPs, and business relationships are already more attuned to

typically this is limited to specific geographic locations. In contrast, traditional CDN players (e.g., Akamai) manage CDN servers globally.

collaboration (CDNs and brokers both directly optimize content delivery under contract with CPs).

In this section, we studied the classification of stakeholders and their roles in the caching ecosystem and the features of wireless edge caching. In the next section, we study a specific techno-economical problem where three vertical stakeholders, i.e., a CP, a Telco CDN operator and users coexist in the wireless edge caching scenario.

## 1.4 Elastic Wireless Cache Lease, Content Caching and Routing

With elastic CDNs, small-scale CPs without their own CDN servers can rent cache space as needed at different cloud locations from ISPs with edge servers in order to enhance their offered quality of service (QoS). This section discusses key challenges in this context, namely how to invest an available budget in cache space in order to match spatio-temporal fluctuations of file demand and storage prices. Specifically, we consider jointly *dynamic cache rental*, *file placement*, and *request-cache association* in wireless scenarios in order to provide just-in-time CDN services. The goal is to maximize the benefits of average download delay obtained by the rented caches while ensuring that the time-average rental cost is less than a fixed budget. We leverage the Lyapunov drift-minus-benefit technique to transform our infinite horizon problem into hour-by-hour subproblems which can be solved without knowledge of distant future file popularity and transmission rates. We propose efficient solutions for both non-overlapping and overlapping small cells scenarios, respectively.

### 1.4.1 Scenario

A large portion of today's Internet traffic is handled by CDNs owned by large content companies like Google and Netflix. Such deployments require a significant-often prohibitive for newcomers-investment for the cache servers and the associated control systems. For a smaller size CP such as Pinterest, Tumbler, and Snapchat, an alternative way to exploit the CDN servers is to use a cache rental service from a CDN provider such as Akamai [18] or Amazon AWS [4]. However, this can be very costly and impractical for some CPs since the leases are on the long-term basis, prices are fixed and catalog-dependent and the content placement decisions are made by the CDN provider.

As we mentioned in the introduction, disruptive solutions known as content delivery network as-a-service (CDNaaS) or *elastic CDN* (eCDN) [9] have recently emerged. Thus, the eCDNs enable a novel business model, where small CPs can dynamically rent storage and instantiate virtual CDNs to meet customer demand just-in-time and space, i.e., whenever and wherever caching is needed. Clearly, eCDNs can benefit CPs with tight monetary budgets, volatile demand and/or

seeking fine-grained caching control over their storage management and caching decisions.

At the same time, this model raises technical and economic questions. In particular, the CP must decide *(i)* how much storage to lease at each location in order to meet user demand and *(ii)* which content items (files) to cache at each of these storage reservoirs. Furthermore, these decisions need to be updated regularly, often on a per-hour basis, in order to accommodate the time-varying nature of the content demand. At each round, the CP encounters an investment dilemma: a larger cache lease will improve service quality but will also increase expenditure. With a given operational expenditure budget, the investment decisions are inherently coupled across different rounds. Overspending in one round improves the current performance but restraints subsequent decisions and limits future opportunities. The goal of a small CP is to *optimize the storage management and caching decisions that the CP needs to make when leasing storage from an eCDN system.*

We study the more challenging but increasingly relevant scenario where the eCDN is owned by a MNO and the storage resources are at small cell base stations (SBSs).[3] We consider a wireless edge caching scenario such as Fig. 1.2(b). In this scenario, the end users subscribe a content service from a CP and make a payment for the network usage to a Telco CDN operator. The Telco CDN operator provides both edge caching services and data delivery services (from core data centers to the edge servers) to the CP and mobile internet service to the end users. The end users are supposed to subscribe to the content service and use the mobile internet service with a fixed price for the constant duration, say one month. In addition, the Telco CDN operator decides the price for data delivery and edge caching service every hour based on the electricity price or network environments.

In the eCDN mobile network, a geographical area is covered by several SBSs with possibly overlapping coverage. The SBSs have caches which can be dynamically rented by the CP. When a file is served from a leased cache, there is a delay benefit as opposed to being served from a remote CP server, which is attributed to the proximity of the cache. Therefore, the CP can invest in cache space in different time slots and locations, decides which files to place in the rented space, and then enjoys a service delay benefit for the requests that were served from the caches. The business model in this section is that the CP gives an average cache rental budget to the mobile operator where all operations including cache scaling, content caching and wireless routing decisions are entrusted to the Telco CDN operator. It would be beneficial to both of the CP and the Telco CDN operator by jointly manipulating cache scaling, content caching and wireless routing since it enables to reduce backhaul congestions as well as enhances the QoS for end users. Then, the objective is to *select investment, placement, and request asso-*

[3]For example, AT&T has envisioned using their own CDN, namely *Telco CDN* which integrates content delivery with traffic engineering [37].

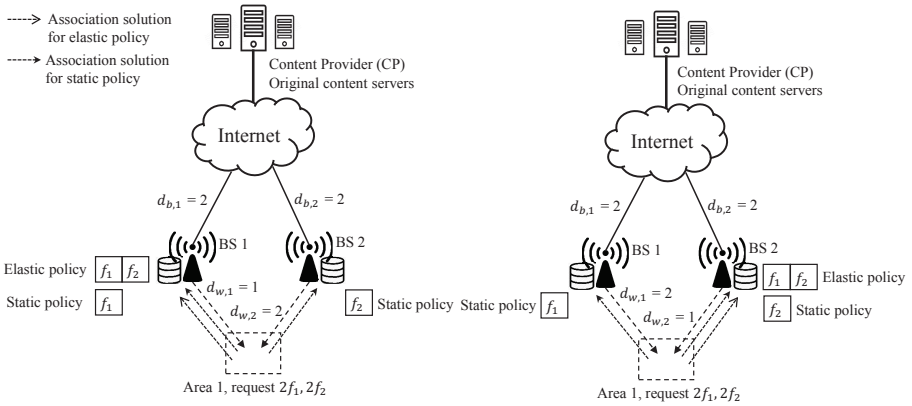



**Figure 1.3** An example of the elastic policy and the static policy in a cache rental system. For the simplicity of the model, the only experienced wireless delay is a time-varying parameter in this model. Note that the association solution for static policy is fair time-sharing between BS 1 and BS 2 for both time slots since time-average delays for both BSs are the same.

*ciation in order to minimize the average service delay for a given time-average budget.*

Apart from this model, we can also consider different types of business models. For example, the CP invests its budget to lease the edge cache capacity and makes a decision of file caching itself while file routing to different BSs can be controlled by the Telco CDN operator [26]. This model also makes sense in a case that the information of video files must be kept secure from the Telco CDN operator, yet the CP cannot benefit from the joint optimization of file routing and file caching.

### 1.4.2 Motivating Example of Elastic Cache Lease

Fig. 1.3 shows an example of the elastic cache lease, file caching and area-BS association policy and the static policy of them in a cache rental system.[4] In this example, we assume a scenario that a CP provides content service for users in area 1 and a Telco CDN operator has two edge servers attached to each BS to provide cache memory space lease service to the CP. Moreover, there are backhaul delay $d_{b,n}$ which denotes the experienced delay to transfer/receive files between the CP and each BS and wireless delay $d_{w,n}$ which denotes the experienced delay to transmit files from each BS to the end-users in area 1, respectively where $n \in 1, 2$ denotes the BS index. In this example, we consider following dynamic states, i.e., states change over time slots and static states, i.e., states do not change over time slots: (i) *dynamic states:* $d_{w,n}$ for all BSs over time slots, (ii) *static states:* $d_{b,n}$ for all BSs and the number of file requests for all files $f_1, f_2$. We assume that

[4]To simplify, we do not consider the units of all parameters.

$d_{b,n}$ for both BSs are 2 and file request arrivals for both files are 2 every time slot. At time slot 1, the CP which adopts the elastic policy is likely to lease the memory space at BS1 for all files and all requests would be associated with BS1 since the wireless delay from BS1 to area 1 is smaller than that from BS2 to area 1. At time slot 2, the CP which adopts the elastic policy is likely to lease the memory space at BS 2 to cache all files and all requests would be associated with BS2 since the wireless delay from BS2 to area 1 is smaller than that from BS1 to area 1. However, the CP which adopts the static policy is likely to lease both memory spaces at BS1 and BS2 to cache $f_1$ at BS 1 and cache $f_2$ at BS 2, respectively and use fair time-sharing association between two BSs since the time-average wireless delays are the same from both BSs to area 1.

Then, we evaluate the elastic policy and the static policy using three metrics as follows: (i) total cache lease costs, (ii) total backhaul bandwidth usage costs, and (iii) total end-to-end delays (from the closest location which caches the corresponding file to area 1). First, the total cache lease costs of both policies are the same since the memory space to cache two files every time slot is leased by the Telco CDN operator every time slot. Second, the total backhaul bandwidth usage cost for the elastic policy is 0 and for the static policy is $4R$ where $R$ denotes the bandwidth usage price to transmit a file over a backhaul link. The cost for the static policy comes from the fact that one request of each file must be transmitted from the original content servers via BS1 or BS 2 in both time slots. Finally, the end-to-end delay of the elastic policy is 2 (2 file requests for file $f_1$) +2 (2 file requests for file $f_2$) = 4 at time slot 1 and 2+2 = 4 at time slot 2 and that of the static policy is 1 ($f_1$ from BS1 caching) + 3 ($f_2$ from the CP to area 1 via BS1)+ 2 ($f_2$ from BS2 caching) + 4 ($f_1$ from the CP to area 1 via BS2) = 10 at time slot 1 and 1 ($f_2$ from BS2 caching) + 3 ($f_1$ from the CP to area 1 via BS2) + 2 ($f_1$ from BS1 caching) + 4 ($f_2$ from BS2 caching) = 10 at time slot 2, respectively, i.e., the total end-to-end delay of the elastic policy is 8 and that of the static policy is 20, respectively. In summary, the elastic policy is able to attain the better QoS of end users with the same edge memory lease cost and less backhaul usage cost than the static policy.[5]

### 1.4.3 System Model

The cache rental system consists of a macro BS (MBS) (we denote it using letter $s$) and several SBSs collected in set $\mathcal{J}$. All stations together $\mathcal{J} \cup \{s\}$ provide coverage to a given geographical area, cf. Fig. 1.4. Specifically, we partition the geographical area into $\mathcal{I}$ non-overlapping subareas and use $\mathcal{J}_i \subseteq \mathcal{J}$ to denote the subset of SBSs that are *reachable* by subarea $i \in \mathcal{I}$. The MBS is always reachable. Each SBS offers *storage for lease*, which can be used to cache files and facilitate their delivery.

[5] In this example, we assume that file popularity can be exactly predicted. There exist many studies which address the prediction of the file popularity using machine learning techniques such as the one in [38] and references therein.

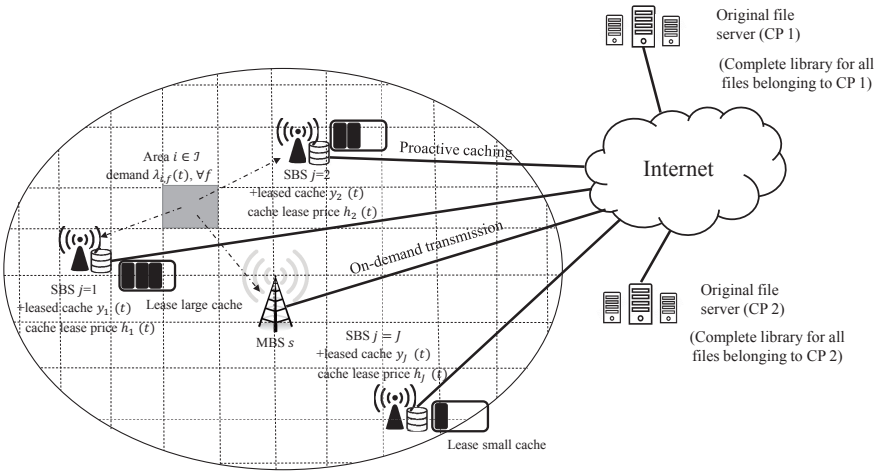



**Figure 1.4** Overview of cache rental, file caching and association in wireless elastic CDNs.

**Table 1.2** Summary of the notations

| Notation | Definition | Notation | Definition |
|---|---|---|---|
| $i \in \mathcal{I}$ | area index | $d_{is}(t)$ | average delay for serving area $i$ by remote servers |
| $j \in \mathcal{J}$ | SBS index | $d_{ij}(t)$ | average delay for serving area $i$ by SBS $j$ |
| $s$ | MBS index | $h_j(t)$ | price to lease cache storage per unit bit |
| $f \in \mathcal{F}$ | file index | $\lambda_{i,f}(t)$ | demand profile |
| $B_{avg}$ | average budget constraint | $y_j(t)$ | leased cache space at SBS $j$ |
| $t$ | hour index (time slot) | $z_{j,f}(t)$ | file caching indicator |
| $x_{ij,f}(t)$ | association probability | | |

Time is slotted in hours $t = 0, 1, \ldots$. For each file $f$ in a catalog $\mathcal{F}$, we denote with $\lambda_{i,f}(t)$ the traffic demand emanating from subarea $i$ in slot $t$. We assume that $\lambda_{i,f}(t)$ is an i.i.d. process with finite mean and variance.[6] The traffic demand reflects the file popularity in time and space, and therefore it will be crucial to adjust caching decisions over time.

When a user requests a file, there is an associated download delay $d_{ij}(t), j \in \mathcal{J}_i \cup \{s\}$, which depends on *(i)* the subarea $i$, where the user is located, and *(ii)* the station $j \in \mathcal{J} \cup \{s\}$ from which the file is retrieved, which together determine the communication path used; delay is associated to a path for various reasons such as wireless interference, congestion, propagation time, etc., which are all path-specific. When the file is retrieved from the MBS, a remote server is contacted

[6]It is possible to extend the model to Markovian arrivals using the framework of [39].

to obtain the file (Fig. 1.4), and although this ensures delivery of every file–and hence feasibility of our mathematical problem–the corresponding download delay $d_{is}(t)$ is generally large. To improve QoS, the file can be retrieved from a nearby SBS cache, instead of the MBS.

To this end, SBS $j \in \mathcal{J}$ leases storage for purposes of caching. The storage is leased at a fluctuating price $h_j(t)$ per unit, which is extrinsic to our system. The price changes over time following electricity price fluctuations [40] and a spot market of storage, where storage owners sell their left-overs, and hence the price is affected by temporal ebbs and flows of traffic and storage demand.

We introduce the investment variables $y_j(t)$ to denote the amount of SBS storage that is leased for caching operations in slot $t$. The investment decisions are subject to an economic constraint. Specifically, we have in mind an average budget $B_{avg}$, which must be satisfied over a long horizon. On one hand, cloud service providers, like Amazon AWS, provide storage lease (called S3) and back-haul transmission (called CloudFront) at prices that are adapted every hour, which motivates investment decisions on hourly-basis. On the other hand, CPs must meet operational expenditure (OpEx) billing targets only at a much larger time scale, e.g., over a month. With these particularities in mind, we introduce our time-average budget constraint, expressed as follows:

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{j\in\mathcal{J}} y_j(t) h_j(t) \leq B_{avg}, \quad \text{(billing constraint)} \tag{1.1}$$

where the term $\sum_{j\in\mathcal{J}} y_j(t)h_j(t)$ represents the total investment in slot $t$, the LHS is the time average investment, and $B_{avg}$ is the available *average budget* in dollars per hour to be spent on storage leasing.

To determine the average delay experienced within an hour, we must describe carefully how each file is served. For this, we introduce two more sets of decision variables: (i) file placement variable $z_{j,f}(t) \in \{0,1\}$ takes value 1 iff file $f$ is cached at SBS $j$ in slot $t$, and (ii) demand association variable $x_{ij,f}(t) \in [0,1]$ denotes the fraction of location $i$ traffic demand for file $f$ that is served by SBS $j$, again in slot $t$. We can now express the hourly end-to-end delay benefit from file caching for the subarea $i$ and SBS $j$ as:

$$D_{ij}(\boldsymbol{x}(t), \boldsymbol{z}(t); \boldsymbol{\lambda}(t), \boldsymbol{d}(t)) = (d_{is}(t) - d_{ij}(t)) \sum_{f\in\mathcal{F}} x_{ij,f}(t) z_{j,f}(t) \lambda_{i,f}(t). \tag{1.2}$$

Observe that the delay depends on the fraction of traffic served at this location $x_{ij,f}(t)$, on whether the file is actually cached here $z_{j,f}(t)$, and finally on the volume of demand $\lambda_{i,f}(t)$. The total delay benefit in slot $t$ is:

$$g_t(\boldsymbol{x}(t), \boldsymbol{z}(t); \boldsymbol{\lambda}(t), \boldsymbol{d}(t)) = \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}_i} D_{ij}(\boldsymbol{x}(t), \boldsymbol{z}(t); \boldsymbol{\lambda}(t), \boldsymbol{d}(t)). \tag{1.3}$$

Below we will drop $(\boldsymbol{\lambda}(t), \boldsymbol{d}(t))$ from the argument of $g$, though the dependence on these parameters remains implied. We note that the total delay and the total delay benefit add up to a constant term (equal to the total delay without caching)

and hence minimizing total delay is equivalent to maximizing total delay benefit. Below, we focus on the latter.

A number of constraints must be satisfied at each time slot. Specifically, the entire demand emanating from each subarea must be served (by SBSs or MBS)

$$\sum_{j \in \mathcal{J}_i \cup \{s\}} x_{ij,f}(t) = 1, \quad \forall i, f, t, \quad \text{(service constraint)} \tag{1.4}$$

and the file placement is limited by the available (leased) storage:

$$\sum_{f \in \mathcal{F}} z_{j,f}(t) \leq y_j(t)/b, \quad \forall\, j, t, \quad \text{(storage space constraint)} \tag{1.5}$$

where $b$ is the size of each file; we assume that $b$ is the same for all files for simplicity, but we can model a heterogeneous file size scenario by dividing the different size of files into same size chunks. To facilitate reading, we summarize the notations in Table 1.2.

### 1.4.4 Problem Formulation

The system is operated with an *elastic CDN strategy*, which at slot $t$ maps the current state of the system to a decision tuple $(x_{ij,f}(t), y_j(t), z_{j,f}(t))$. An elastic CDN strategy is called *feasible* if it satisfies the billing constraint (1.1) and the instantaneous constraints of service (1.4) and caching space (1.5) explained above. We would like to address the mobile operator question *"what is the feasible elastic CDN strategy that maximizes average delay benefit?"* This question can be addressed by the following control problem:

$$\begin{aligned} \mathbf{(P)}: \max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} g_t(\boldsymbol{x}(t), \boldsymbol{z}(t)), \\ s.t. \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{j \in \mathcal{J}} y_j(t) h_j(t) \leq B_{avg}, \\ \sum_{j \in \mathcal{J}_i} x_{ij,f}(t) = 1, \forall i, f, t, \quad \sum_{f \in \mathcal{F}} z_{j,f}(t) \leq y_j(t)/b, \forall j, t. \end{aligned} \tag{1.6}$$

Note that this control problem is challenging for the following reasons:

- Crucial factors for the objective such as future traffic demand $\lambda_{i,f}(t)$ and future delay gains $d_{is}(t) - d_{ij}(t)$ are unknown at the time the investment decisions $y_j(\tau)$ are taken ($\tau < t$),
- Due to the time average billing constraint, a large investment $y_j(\tau)$ reduces the available budget in future slots $t > \tau$, which can be problematic in combination with the unknown future costs $h_j(t)$, delays $d_{ij}(t), d_{is}(t)$ and traffic demand $\lambda_{i,f}(t)$.

### 1.4.5 Lyapunov-based Elastic CDN Strategy

Since problem **(P)** involves the challenging time-average constraint (1.1), a promising approach is to couple the fate of this constraint with an evolving controllable counter. To this end, we introduce a virtual queue (or counter) whose backlog is updated by

$$Q_B(t+1) = \Big[Q_B(t) + \sum_{j\in\mathcal{J}} y_j(t)h_j(t) - B_{avg}\Big]^+. \tag{1.7}$$

Prior work [41] shows that if weak stability conditions hold for the virtual queue, i.e.,

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} Q_B(t) < \infty, \tag{1.8}$$

then constraint (1.1) is asymptotically satisfied, in the sense that its residual tends to zero as $T \to \infty$. Intuitively, the backlog $Q_B(t)$ estimates the total excess budget spent in the previous time slots (instantaneous residual), and therefore $Q_B(t)$ is valuable information for deciding how to invest at slot $t$. Then, let us focus on slot $t$. The decision maker is aware of 1) the mean traffic demand profile for the next hour $[\lambda_{i,f}(t)]_{i,f}$, which in practice is achieved by measurements and use of machine learning methods, cf. [38], 2) the delay profile realizations $[d_{ij}(t)]_{i,j}$ available by measurements, and the readily available 3) prices $[h_j(t)]_j$ and 4) virtual queue length $Q_B(t)$, while file size $b$ is assumed known. Therefore, the elastic CDN strategy is applied on the state $([\lambda_{i,f}(t)]_{i,f}, [d_{ij}(t)]_{i,j}, [h_j(t)]_j, Q_B(t))$. To design a strategy that solves **(P)** we employ a *Lyapunov drift-minus-benefit* framework as follows.

We first define the quadratic Lyapunov function and arising drift as follows:

$$L(t) \triangleq \frac{1}{2}Q_B(t)^2, \tag{1.9}$$

$$\Delta(t) \triangleq \mathbb{E}\{L(t+1) - L(t)|Q_B(t)\}. \tag{1.10}$$

Note that the Lyapunov drift $\Delta(t)$ depends on slot $t$ decision $(\boldsymbol{y}(t), \boldsymbol{x}(t), \boldsymbol{z}(t))$ implicitly via the update of $Q_B(t+1)$, and provides information about the expected improvement of constraint satisfaction when taking a specific decision; a decision with small cost $\sum_{j\in\mathcal{J}} y_j(t)h_j(t)$ will tend to reduce the length $Q_B(t+1)$, and therefore make the drift negative. Since we are also interested in maximizing the time average of $g_t$ using instantaneously feasible decisions (satisfying demand and storage space constraints), we next introduce the Lyapunov drift-minus-benefit function ($DMB$) which balances the drift and the instantaneously obtained delay benefit:

$$DMB(\boldsymbol{x}(t), \boldsymbol{z}(t)) = \Delta(t) - V\mathbb{E}\{g_t(\boldsymbol{x}(t), \boldsymbol{z}(t))|\boldsymbol{Q}(t)\}, \tag{1.11}$$

where $V$ is a constant parameter to balance the tradeoff between two conflicting objectives, (i) reducing the billing constraint residual and (ii) increasing the delay benefit.

Applying the queue update equation (1.7) and lemma 4.3 from [42], we obtain under any possible decision $([y_j(t)]_j, [x_{ij,f}(t)]_{ijf}, [z_{j,f}(t)]_{jf})$:

$$\begin{aligned} DMB(\boldsymbol{x}(t), \boldsymbol{z}(t)) \leq P - V\mathbb{E}\{g_t(\boldsymbol{x}(t), \boldsymbol{z}(t))|Q_B(t)\} \\ -\mathbb{E}\Big\{\Big(B_{avg} - \sum_{j\in\mathcal{J}} y_j(t)h_j(t)\Big)Q_B(t)|Q_B(t)\Big\}, \end{aligned} \tag{1.12}$$

where $P = (B_{avg}^2 + |\mathcal{J}|y_{max}^2 h_{max}^2)/2$ is a positive constant, and $y_{max}$ and $h_{max}$ denote the maximum storage that can be leased at any SBS during an hour, and the maximum price respectively. Neely [39] showed that we can uncover optimal decisions by minimizing the RHS of (1.12).

We propose the elastic CDN strategy (*SBSD*) which at slot $t$ takes actions $(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{z}(t)) = (\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{z}^*)$, where

$$(\boldsymbol{x}^*, \boldsymbol{y}^* \boldsymbol{z}^*) \in \arg\max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} V g_t(\boldsymbol{x}, \boldsymbol{z}) - \sum_{j\in\mathcal{J}} Q_B(t) y_j h_j(t), \tag{1.13}$$

$$s.t. \qquad \sum_{j\in\mathcal{J}_i} x_{ij,f} = 1, \forall i, f, t, \quad \sum_{f\in\mathcal{F}} z_{j,f} \leq y_j/b, \forall j, t.$$

The first straightforward result is that *SBSD* is a feasible elastic CDN policy. First, the instantaneous constraints of service (1.4) and storage space (1.5) are automatically satisfied at each slot by the design of the policy. Then, we may observe that *SBSD* minimizes the RHS of (1.12), therefore using lemma 4.6 in Neely's book [39], we can show that *SBSD* also stabilizes $Q_B(t)$, and hence the billing constraint (1.1) is asymptotically satisfied.

Some further remarks are in order:

- As long as the technical requirement "$\boldsymbol{\lambda}(t)$ and $\boldsymbol{d}(t)$ have finite second moments" is satisfied (used in the proof of asymptotic feasibility), *SBSD* will satisfy the budget constraint (even if for example their statistics are not Markovian, or their predictions are provided in a delayed or erroneous fashion).
- It achieves a near-optimal average delay benefit without *a priori* knowledge of popularity or delay statistics, but rather by looking at the hourly learned popularity and delay profile and keeping a budget counter.
- [43] has shown that $Q_B(t)$ plays the role of a stochastic Lagrangian multiplier. Therefore, when $\boldsymbol{\lambda}(t)$ and $\boldsymbol{d}(t)$ are stationary, we also anticipate that $Q_B(t)$ will vary around a value related to $V$, hence we can pick a large $V$. However, with non-stationary demand and delay profiles, we expect $Q_B(t)$ to vary following non-stationary trends, and therefore it becomes essential to pick a small $V$ in order to make the algorithm robust.

It remains to show that *SBSD* in fact solves **(P)**, by achieving a near optimal time average delay benefit. We show this next for two different cases, *(i)* non-overlapping, and *(ii)* overlapping SBS coverages.

**Non-overlapping SBS coverage.** When SBS coverage is non-overlapping,

each subarea can reach a single SBS cache, which immediately simplifies routing splits $x_{ij,f}(t)$, such that $x_{ij,f}(t) = 1,\ \forall t$ if subarea $i$ can reach SBS $j$ and 0 otherwise, for all $i, j, f$. In essence, each request can be served only by the reachable cache (or the MBS when the file is not cached there). We will see that this makes our problem relatively easy to solve.

First, we note that caching file $f$ at SBS $j$ in slot $t$ brings the following delay benefit:

$$K_{j,f}(t) \triangleq \sum_i (d_{is}(t) - d_{ij}(t)) x_{ij,f}(t) \lambda_{i,f}(t),$$

which is computable using known parameters $\boldsymbol{d}, \boldsymbol{x}, \boldsymbol{\lambda}$ ($\boldsymbol{x}$ is a parameter here because it is fully determined by the reachability of the cache) and independent of the decisions $\boldsymbol{y}(t), \boldsymbol{z}(t)$. Consequently, the *SBSD* optimization becomes:

$$\begin{aligned} \max_{\substack{y_j \geq 0 \\ z_{j,f} \in \{0,1\}}} \quad & V \sum_{j,f} K_{j,f}(t) z_{j,f} - Q_B(t) \sum_{j \in \mathcal{J}} y_j h_j(t), \\ s.t. \quad & \sum_{f \in \mathcal{F}} z_{j,f} \leq y_j / b, \quad \forall j, f. \end{aligned} \tag{1.14}$$

Due to its simple form, (1.14) can be solved by inspection. At each pair SBS-slot $(j, t)$, we order files in decreasing values of $K_{j,f}(t)$. For an investment $y_j(t)$, the highest delay benefit is collected by caching the $y_j(t)/b$ files that rank higher in this list. This provides directly the solutions $\boldsymbol{z}$ as a function of $\boldsymbol{y}$, it remains now to determine the latter. With a slight abuse of notation, let us call $\sigma$ the permutation of file indices that implies $K_{j,\sigma(1)}(t) \geq \cdots \geq K_{j,\sigma(|\mathcal{F}|)}(t)$ (the abuse is because we do not explicitly denote the dependence of $\sigma$ on $j, t$ to reduce clutter), then we can decompose the investment decisions per SBS, and find $y_j^*(t)$ by maximizing:

$$y_j^*(t) \in \arg\max_{y_j \geq 0} \sum_{f=1}^{\lfloor y_j / b \rfloor} K_{j,\sigma(f)}(t) - \frac{Q_B(t)}{V} h_j(t) y_j.$$

Above, $y_j^*(t)$ can be efficiently computed by listing partial sums $\sum_{f=1}^{\lfloor y_j/b \rfloor} K_{j,\sigma(f)}(t)$ for $y_j/b = 1, 2, \ldots$ until the difference of one partial sum from the previous becomes smaller than $\frac{Q_B(t)}{V} h_j(t)$.

Mathematically speaking, the above might include cases where the solution is to avoid investment altogether ($\sum_j y_j^*(t) = 0$), or buy storage for all files ($y_j^*(t) = |\mathcal{F}|$), however in practice these cases are extremely rare, because of the skewness of popularity: we will always benefit from storing popular files and we will seldom benefit from storing unpopular ones. Moreover, since we decouple the problem with the subarea-SBS association, the problem can be decomposed into each SBS's problem. Below we give the algorithmic steps to find $\boldsymbol{y}_j$ and $\boldsymbol{z}_j$ for all SBS $j \in \mathcal{J}$ optimally in detail:

1. Calculate $K_{j,f}(t) = \sum_i (d_{is}(t) - d_{ij}(t)) x_{ij,f}(t) \lambda_{i,f}(t)$ for all files.

2. Sort $K_{j,f}(t)$ with permutation $\sigma$, such that $K_{j,\sigma(1)}(t) \geq \cdots \geq K_{j,\sigma(|\mathcal{F}|)}(t)$.
3. Set partial sums $S(e) = \sum_{f=1}^{e} K_{j,\sigma(f)}(t)$, for $e = 1, 2, \ldots$.
4. Find $e^*$ which is the smallest $e$ that ensures $S(e) - S(e-1) < \frac{Q_B(t)}{V} h_j(t)$.
5. Choose cache lease: $y_j^*(t) = e^* b$.
6. Choose file placement:

$$z_{j,\sigma(f)}^*(t) = \begin{cases} 1 & \text{if } f \leq \lfloor y_j^*(t)/b \rfloor, \\ 0 & \text{otherwise.} \end{cases}$$

Then, the algorithm, namely *Optimal* algorithm in the non-overlapping SBS case has the following features:

- Given virtual queue length, storage price, and parameter $V$, the algorithm finds the amount of storage that if leased it optimizes a weighted sum of delay benefits and budget penalties.
- For the found storage amount that is leased, files are cached at each SBS according to which yields the highest delay benefit, until the available leased storage is completely filled up.

**General case with overlapping SBS coverage.** Next, we consider the general case, where the areas that SBS cover may overlap. In this case, area-SBS association variables $x_{ij,f}(t)$ must be jointly decided with cache rental and file placement, and we may no longer use the trick with $K_{j,f}(t)$, since the user can be served from possibly multiple SBSs and the actual collected delay benefit depends on which SBS is selected. We remind the reader that the *SBSD* strategy determines the decisions solving:

$$\begin{aligned} \max_{\substack{y_j \geq 0 \\ x_{ij,f} \in [0,1] \\ z_{j,f} \in \{0,1\}}} \quad & V g_t(\boldsymbol{x}, \boldsymbol{z}) - Q_B(t) \sum_{j \in \mathcal{J}} y_j h_j(t), \\ s.t. \quad & \sum_{f \in \mathcal{F}} z_{j,f} \leq \frac{y_j}{b}, \ \forall j. \\ & \sum_{j \in \mathcal{J}_i} x_{ij,f} = 1, \forall i, f. \end{aligned} \tag{1.15}$$

We note that (1.15) as formulated is a mixed integer *non-linear* program (MINLP) due to the product of variables $x_{ij,f}$ and $z_{j,f}$ that appears inside $g_t$ in the objective. To proceed, we linearize the objective by replacing products $x_{ij,f} z_{j,f}$ with simply $x_{ij,f}$, and adding an extra constraint $x_{ij,f}(t) \leq z_{j,f}(t)$. Note that if $z_{j,f}(t) = 0$ then the constraint implies that $x_{ij,f}(t) = 0$ as well, eliminating any delay benefit at the objective. Also if $z_{j,f}(t) = 1$, variable $x_{ij,f}(t)$ is not affected by the new constraint, and works as before. To deal with the case where a file is not found in any cache (which by our new construction would imply $\sum x_{ij,f} = 0$ and would violate the last constraint), we must also add a *dummy* allocation variable $x_{is,f}$ to ensure that the constraint (1.4) can be always

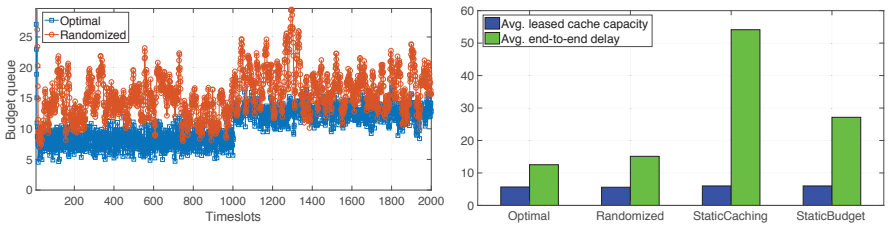
satisfied. We remark that due to the maximization, these dummy variables are always forced to take zero value if a positive delay benefit can be obtained.

Now, the problem becomes a Mixed Integer Linear Program (MILP). To solve this problem, we can consider three approaches as follows:

- We may solve the linear relaxation of (1.15), and then use a rounding technique to obtain an approximation guarantee, e.g., a possibility is to combine the relaxation with randomized rounding [44]. By §4.7 of [45], our approximate solution of (1.15) will provide an elastic CDN strategy with approximate feasibility and average delay benefit. In turn, the approximate feasibility can lead to a feasible strategy with some extra losses.
- As explained in [14], it is possible to use MDS codes to achieve an effective "fractional file placement". In essence, each cache stores a number of linear combinations of file chunks which correspond to fractions of a file, and then each user can combine different such coded chunks to produce the original file. In this context, the *SBSD* becomes a Linear Program and can be solved quickly to optimality.
- A third approach is to obtain an efficient approximate solution is to apply the idea of "Low complexity scheduling" from [46]. This method assigns to the leased cache capacity by smoothly increasing it or decreasing it with a small step size. The sign of the change is randomly chosen. Then it resolves our *SBSD* optimization to get a new average delay benefit, and if these new values outperform previous delay benefits, the random solution is applied.

In this chapter, we take the third method as an example to derive an algorithm in the general case. In this context, we provide a stability guarantee for the budget queue length $Q_B(t)$, which implies that the produced strategy is asymptotically feasible. The strategy, namely *Randomized* algorithm is described in the following steps:

1. For the first time slot, leased cache capacity $y_j^*(1)$ is chosen as $B_{avg}/(|\mathcal{J}|h_{avg})$ for all SBSs.
2. Based on the decided leased cache capacity for each SBS, file caching and user association solutions $(\boldsymbol{x}^*(t), \boldsymbol{z}^*(t))$ are obtained using a greedy file caching (GFC) policy and an optimal user association (OUA) policy for a given file caching solution which are described in the following.
3. For time slots $t > 1$, leased cache capacity $y_j'(t)$ is chosen as $y_j^*(t-1) + \delta \cdot U_j(t-1)$ where $\delta$ denotes small step size and $U_j(t-1)$ is uniformly chosen in $\{-1, 1\}$ for all SBSs.
4. Based on the decided leased cache capacity for each SBS, file caching and user association solutions $(\boldsymbol{x}'(t), \boldsymbol{z}'(t))$ are obtained using a GFC policy with an OUA policy for a given file caching solution.
5. Compare $Vg_t(\boldsymbol{x}'(t), \boldsymbol{z}'(t)) - Q_B(t)\sum_{j\in\mathcal{J}} y_j'(t)h_j(t)$ and $Vg_t(\boldsymbol{x}^*(t-1), \boldsymbol{z}^*(t-1)) - Q_B(t)\sum_{j\in\mathcal{J}} y_j^*(t-1)h_j(t)$ and choose a set of solutions whose objective value is greater as an optimal set of solutions, i.e., $(\boldsymbol{x}^*(t), \boldsymbol{z}^*(t))$.




(a) Budget queue statistics. (b) Avg. performance metrics.

**Figure 1.5** Improvement in performance due to the elastic cache lease and file caching and the elastic behavior in a non-overlapping SBS scenario: 2 SBSs, 2 areas, 10 files in each SBS, average cache lease budget is the same with caching 6 files in each time slot.

The greedy file caching (GFC) policy begins with an empty cache set in all SBSs. Then, this policy iteratively caches files one-by-one in all SBSs where an added file in each step is selected so as to maximize the differential objective value in (1.15). For a given set of cached file, the OUA policy is to choose association variables by

$$x^*_{ij,f}(t) = \underset{x_{ij,f}(t)}{\operatorname{argmax}} \, (d_{is}(t) - d_{ij}(t)) z^*_{j,f}(t) \lambda_{i,f}(1), \quad \forall i, f.$$

This solution is robust due to the comparison mechanism between the solution of the current time slot and that of the previous time slot. Namely, if the budget queue increases due to the excessive investment for cache lease, it reduces the objective value, hence it forces the decision maker to choose the solution of the previous time slot. On the other hand, if the budget queue decreases due to the less investment for cache lease, it increases the objective value, hence it forces the decision maker to choose the solution of the current time slot. This mechanism stabilizes the budget queue.

Moreover, for a given leased cache capacity, a joint file caching and user association problem is shown to be a monotone submodular problem with matroid constraints in respect to cached files in SBSs according to the recent literature, e.g., [47]. This implies a greedy-fashioned file caching algorithm in conjunction with the optimal user association (OUA) policy (for a given file caching solution) probably achieves a constant factor approximation $(1 - 1/e)$ to the optimal performance.

To quantify the performance improvement of the elastic cache lease and file caching over the static policies, we run simulations under a simple non-overlapping SBS scenario (2 SBSs, 2 areas, 10 files in each SBS). We assume that each area is associated with the nearest SBS. In this scenario, delay for the serving area by SBS and delay for the serving area by remote servers in each time slot are drawn from the Gaussian distribution with various parameters and taken only positive values. To capture the spatio-temporal diversity of file popularity, the

arrival rate of each file is drawn from the Zipf distribution [38] and different Zipf parameters are used for each area and each period of time slots.[7]

We compare the proposed algorithms, i.e., *Optimal* and *Randomized* with *StaticCaching* and *StaticBudget* policies. The StaticCaching policy caches the files based on the general content popularity with the static cache investment, i.e., caching the same number of files at all SBSs whereas the StaticBudget policy uses the static cache lease for all time slots but file caching is chosen so as to maximize our objective function in SBSD, i.e., this policy adopts an adaptive file caching for a given cache capacity.

Fig. 1.5 shows the budget queue statistics of the elastic algorithms, total end-to-end statistics and leased cache capacity for all algorithms. The elastic algorithms (i.e., Optimal and Randomized) opportunistically exploit the dynamics of network delays and file request arrivals with keeping average leased cache capacity whereas the static algorithms (i.e., StaticCaching and StaticBudget) leases a fixed amount of budget every time slot. For example, when the traffic demand is higher and Zipf parameter is low, then the higher cache capacity is leased, and vice versa. As a result, we could find interesting observations from this simulations as follow: (i) the elastic algorithms perform better than the static algorithms in terms of total end-to-end delay (at least 53% reduction) in even less leased cache capacity. (ii) Randomized policy shows the close to the optimal performance (83% in terms of average end-to-end delay). These results can be found in real spatio-temporal traffic and content popularity variation scenarios such that the traffic arrival is high and Zipf parameter is small during the day and the traffic arrival is low and Zipf parameter is high during the night (in temporal diversity), and such daily traffic and content popularity distribution can be changed when the CDN simultaneously serves Europe and Russia with 4 hours time shift (in spatial diversity). In this experiment, the idea is to save money by not using memory at low traffic. However this will not bring significant benefit in practice because the CDN operator, e.g., AWS will be unhappy with this and raise the price of peak hours to mitigate the low utilization of the cloud system. Hence, the real economical benefits will appear if the filing of one CP can be covered by other CPs.

## 1.5 Conclusion

The economics of caching is one of the less explored research areas than the technical content caching studies even though it is quickly gaining momentum thanks to the advances in network virtualization, which enables elastic control of storage resources. These flexible technologies in conjunction with dynamic wireless network environments introduce new business models in caching services. To reflect

[7]The sum traffics for all files at each time slot and each area are drawn from the Gaussian distribution with various parameters and taken only positive values.

these new business models, it is required to redesign techno-economic mechanisms and policies and specifically to devise novel cache investment schemes that are suitable for these multilateral and almost real-time environmental variations in the presence of limited information about future content demand and wireless channel states.

First, we focused on the recent economic ecosystems of caching services. Collaboration and benefit distribution between vertical stakeholders and competition between the horizontal stakeholders were investigated. We then discussed differences of the core in-network caching and wireless edge caching in a perspective of economics. Second, we focused on a specific elastic cache lease, content caching, and request-SBS association problem where a CP has a limited average budget for operating costs of cache and bandwidth investment. Accordingly, we provided possible cache lease, content caching, and request-SBS association solutions by exploiting the Lyapunov optimization and randomized scheduling techniques. Finally, we quantified the benefits of the elastic cache lease over the static cache lease, and provided discussions for smart pricing.

# References


[1] Cisco. San Jose, CA, "Cisco visual networking index: Global mobile data traffic forecast update, 2015-2020." [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.pdf

[2] S.-E. Elayoubi and J. Roberts, "Performance and cost effectiveness of caching in mobile access networks," in *Proc. of ACM ICN*, 2015, pp. 79–88.

[3] M. Leconte, G. Paschos, L. Gkatzikis, M. Draief, S. Vassilaras, and S. Chouvardas, "Placing dynamic content in caches with small population," in *Proc. of IEEE INFOCOM*, Apr. 2016, pp. 1–9.

[4] "Amazon web service." [Online]. Available: https://aws.amazon.com

[5] S. Elayoubi and J. Roberts, "Performance and cost effectiveness of caching in mobile access networks," in *Proc. of ACM ICN*, 2015, pp. 79–88.

[6] G. Paschos, G. Iosifidis, M. Tao, D. Towsley, and G. Caire, "The role of caching in future communication systems and networks," *to appear, IEEE JSAC*, 2018.

[7] S. Vassilaras, L. Gkatzikis, N. Liakopoulos, I. N. Stiakogiannakis, M. Qi, L. Shi, L. Liu, M. Debbah, and G. S. Paschos, "The algorithmic aspects of network slicing," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 112–119, 2017.

[8] Akamai White Paper, "The case for a virtualized CDN(vCDN) for delivering operator OTT video."

[9] "The elastic CDN solution (akamai-juniper)." [Online]. Available: https://www.juniper.net/assets/kr/kr/local/pdf/solutionbriefs/3510532-en.pdf

[10] M. K. Mukerjee, I. N. Bozkurt, D. Ray, B. M. Maggs, S. Seshan, and H. Zhang, "Redesigning cdn-broker interactions for improved content delivery," in *Proc. of ACM CoNEXT*, 2017, pp. 68–80.

[11] "Akamai collaborates with orange on NFV initiative to dynamically scale CDN capacity for large events." [Online]. Available: https://www.akamai.com/us/en/about/news/press/2016-press/akamai-collaborates-with-orange-on-nfv-initiative.jsp

[12] "Huawei uCDN solution." [Online]. Available: http://carrier.huawei.com/en/solutions/cloud-powered-digital-services/ucdn

[13] "Amazon ElastiCache." [Online]. Available: https://aws.amazon.com/elasticache/

[14] K. Shanmugam, N. Golrezaei, A. Dimakis, A. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. on Inform. Theory*, vol. 59, no. 12, pp. 8402–8413, Sep. 2013.

[15] R. Combes, A. Ghorbel, M. Kobayashi, and S. Yang, "Opportunistic content delivery in fading broadcast channels," *to appear, IEEE JSAC*, 2018.

[16] A. Zewail and A. Yener, "Combination networks with or without secrecy constraints: The impact of caching relays," *to appear, IEEE JSAC*, 2018.

[17] N. Economides and B. E. Hermalin, "The strategic use of download limits by a monopoly platform," *Journal of Economics*, vol. 46, no. 2, pp. 297–327, 2015.

[18] Akamai White Paper, "The case for a virtualized CDN (vCDN) for delivering operator OTT video."

[19] B. Blaszczyszyn and A. Giovanidis, "Optimal geographic caching in cellular networks," in *Proc. of IEEE ICC*, 2015, pp. 3358–3363.

[20] T. Bektas, O. Oguz, and I. Ouveysi, "Designing cost-effective content distribution networks," *Computers & Operations Research*, vol. 34, no. 8, pp. 2436–2449, 2007.

[21] K. Ho, S. Georgoulas, M. Amin, and G. Pavlou, "Managing traffic demand uncertainty in replica server placement with robust optimization," *Proc. of NETWORKING*, pp. 727–739, 2006.

[22] W. Li, E. Chan, Y. Wang, D. Chen, and S. Lu, "Cache placement optimization in hierarchical networks: Analysis and performance evaluation," *Proc. of NETWORKING*, pp. 385–396, 2008.

[23] N. Laoutaris, V. Zissimopoulos, and I. Stavrakakis, "On the optimization of storage capacity allocation for content distribution," *Computer Networks*, vol. 47, no. 3, pp. 409–428, 2005.

[24] B. Li, M. J. Golin, G. F. Italiano, X. Deng, and K. Sohraby, "On the optimal placement of web proxies in the internet," in *Proc. of IEEE INFOCOM*, 1999, pp. 1282–1290.

[25] K. Poularakis, G. Iosifidis, I. Pefkianakis, L. Tassiulas, and M. May, "Mobile data offloading through caching in residential 802.11 wireless networks," *IEEE Transactions on Network and Service Management*, vol. 13, no. 1, pp. 71–84, 2016.

[26] J. Krolikowski, A. Giovanidis, and M. Renzo, "A decomposition framework for optimal edge-cache leasing," *to appear, IEEE JSAC*, 2018.

[27] "Google Global Cache (GCC)." [Online]. Available: https://peering.google.com/#/infrastructure

[28] A. Berglund, "How Netflix works with ISPs around the globe to deliver a great viewing experience," *Netflix Blog*, 2016.

[29] "Cedexis," 2017. [Online]. Available: https://www.cedexis.com/

[30] "Conviva," 2015. [Online]. Available: http://www.conviva.com/

[31] P. Sermpezis, T. Giannakas, T. Spyropoulos, and L. Vigneri, "Soft cache hits: Improving performance through recommendation and delivery of related content," *to appear, IEEE JSAC*, 2018.

[32] L. E. Chatzieleftheriou, M. Karaliopoulos, and I. Koutsopoulos, "Caching-aware recommendations: Nudging user preferences towards better caching performance," in *Proc. of IEEE INFOCOM*, Apr. 2017, pp. 1–9.

[33] J. Tadrous, A. Eryilmaz, and H. El Gamal, "Joint smart pricing and proactive content caching for mobile services," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2357–2371, 2016.

[34] E. Gourdin, P. Maillé, G. Simon, and B. Tuffin, "The economics of cdns and their impact on service fairness," *IEEE Transactions on Network and Service Management*, vol. 14, no. 1, pp. 22–33, 2017.

[35] L. Wang, G. Tyson, J. Kangasharju, J. Crowcroft, L. Wang, G. Tyson, J. Kangasharju, and J. Crowcroft, "Milking the cache cow with fairness in mind," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2686–2700, 2017.

[36] "Cisco Report: The CDN federation - Solution for SPs and content providers to scale a great customer experience." 2012. [Online]. Available: http://www.cisco.com/

[37] "AT&T business, content delivery network." [Online]. Available: https://www.business.att.com/solutions/Family/cloud/content-delivery-network/

[38] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, Feb. 2014.

[39] M. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, pp. 1–211, 2010.

[40] "CAISO: California independent system operator." [Online]. Available: http://www.caiso.com/

[41] M. Neely, "Energy optimal control for time varying wireless networks," *IEEE Trans. on Inform. Theory*, vol. 52, no. 7, pp. 2915–2934, Jul. 2006.

[42] L. Georgiadis, M. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundation and Trends in Networking*, vol. 1, no. 1, pp. 1–149, 2006.

[43] L. Huang, S. Moeller, M. J. Neely, and B. Krishnamachari, "Lifo-backpressure achieves near-optimal utility-delay tradeoff," *IEEE/ACM Transactions on Networking*, vol. 21, no. 3, pp. 831–844, Jun. 2013.

[44] P. Raghavan and C. D. Tompson, "Randomized rounding: a technique for provably good algorithms and algorithmic proofs," *Combinatorica*, vol. 7, no. 4, pp. 365–374, 1987.

[45] L. Georgiadis, M. J. Neely, L. Tassiulas *et al.*, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends® in Networking, Now Publishers, Inc.*, vol. 1, no. 1, pp. 1–144, 2006.

[46] L. Tassiulas, "Linear complexity algorithms for maximum throughput in radio networks and input queued switches," in *Proc. of IEEE INFOCOM*, Apr. 1998, pp. 533–539.

[47] M. Dehghan, A. Seetharam, B. Jiang, T. He, T. Salonidis, J. Kurose, D. Towsley, and R. Sitaraman, "On the complexity of optimal routing and content caching in heterogeneous networks," in *Proc. of IEEE INFOCOM*, Apr. 2015, pp. 936–944.