Cross-Layer Optimization of Fast Video Delivery in Cache- and Buffer-Enabled Relaying Networks

Lin Xiang, *Student Member, IEEE,* Derrick Wing Kwan Ng, *Member, IEEE,* Toufiqul Islam, Robert Schober, *Fellow, IEEE,* Vincent W.S. Wong, *Fellow, IEEE,* and Jiaheng Wang, *Senior Member, IEEE*

***Abstract*—In this paper, we investigate the cross-layer optimization of caching and fast video delivery for enhanced video streaming quality of experience (QoE) in two-hop relaying networks, where a base station supplies video data to multiple users with the help of relays. Different from conventional systems, each half-duplex relay node is equipped with a cache and a buffer to facilitate joint scheduling of video fetching and delivery. This introduces channel diversity gains and facilitates fast video delivery. In particular, we investigate two-stage caching and delivery control schemes for the minimization of the overall video delivery time. An offline caching and delivery optimization problem, which assumes full knowledge of user requests and channel state information (CSI), is formulated but turns out to be functional and non-convex. However, we unveil a hidden quasi-convexity and convexity in the two layers of the decomposed problem and hence solve the offline problem optimally and efficiently. Moreover, online video delivery control exploiting statistical CSI is investigated under a stochastic dynamic programming (DP) framework. To mitigate the high computational complexity of DP, we further propose a low-complexity online video delivery algorithm, which achieves close-to-optimal performance in the high buffer capacity regime. Simulation results show that our offline and online schemes can significantly reduce the overall video delivery time due to the degrees of freedom enabled by caching and buffering. Besides, an interesting trade-off between caching and buffering gains in exploiting the diversity of the wireless channel is revealed.**

I. Introduction

THE surge of video-on-demand (VoD) streaming traffic in cellular networks [2] poses two major challenges for cellular operators. On the one hand, VoD streaming imposes stringent requirements on transmission rate and latency. Supporting VoD streaming with the limited cellular frequency spectrum thus demands highly spectrally efficient transmission and resource allocation schemes in the radio access network (RAN). On the other hand, as VoD servers are usually located at the "Internet edge", a high-capacity backhaul is required to convey the aggregate VoD data from the Internet to the RAN. However, recent studies suggest that the network performance still suffers from backhaul capacity bottlenecks during VoD streaming even if state-of-the-art fourth generation (4G) Long Term Evolution (LTE) networks are deployed [3].

Instead of a traditional bottom-up upgrade from RAN to backhaul, low-complexity cost-effective solutions employing wireless caching have been proposed to address VoD streaming challenges [4]–[8]. The principle of wireless caching is similar to caching in content-centric networking (CCN) [4], [5], i.e., the most popular files are pre-stored into a cache memory deployed at the base stations (BSs) or access points (APs) of the RAN. The interest in wireless caching, however, is mainly explained by its benefits ranging from traffic offloading on the backhaul [6], through delivery capacity enhancement and delay reduction in the RAN [7], to energy savings in the entire network [8]. These benefits result from the *content reuse gains* in the video delivery phase, which are caused by the often highly correlated user preferences. Empirical measurements confirming these high correlations have been reported in [9]. Furthermore, the burden on the network introduced by caching can be kept low since the cache placement usually takes place during periods of low cellular traffic (such as early mornings).

Recently, the integration of wireless caching into the fifth generation (5G) physical layer technologies, including small cells [10], [11], cooperative multiple-input multiple-output (MIMO) [12]–[15], and cross-layer resource allocation [16], for advanced cellular video delivery has been advocated. In [10], FemtoCaching was proposed as a substitute for the backhaul in small cell networks, where the optimal file placement for the minimization of the average download delay subject to a cache capacity constraint was investigated. In [11], cooperative caching in relay nodes and user equipments was considered for the minimization of the energy consumption. Both [10] and [11] have shown that caching can effectively relieve the (wireless/wireline) backhaul capacity bottleneck in small cells by exploiting content reuse and the cellular traffic pattern in the upper layers. Moreover, the reduced cell sizes lead to additional gains in spectral efficiency in the physical layer. These gains constitute *macroscopic* caching gains in cache-enabled small cells, which are achievable without knowledge of the underlying channel characteristics and irrespective of the adopted physical layer.

Meanwhile, wireless caching has been combined with physical layer transmission techniques and scheduling protocols in [12]–[16]. Different from caching in CCNs, wireless caching has the potential to combat fading and alleviate the radio resource scarcity [12]–[16]. In [12], by caching the same data across different BSs, the authors exploited cooperative MIMO transmission for the minimization of the transmit power subject to a data rate constraint. Appealingly, caching reduces the payload sharing overhead of opportunistic cooperative MIMO trans-

 Part of this work [1] has been presented at *IEEE Global Communications Conference (Globecom)*, San Diego, CA, Dec. 2015.

L. Xiang and R. Schober are with the Institute for Digital Communications, Friedrich-Alexander University of Erlangen-Nuremberg, Germany (Email: {lin.xiang, robert.schober}@fau.de). D. W. K. Ng is with the School of Electrical Engineering and Telecommunications, University of New South Wales, Australia (Email: w.k.ng@unsw.edu.au). T. Islam is with Huawei Canada Research Center, Ottawa, Canada (Email: toufiq@ece.ubc.ca). V. W. S. Wong is with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada (Email: vincentw@ece.ubc.ca). J. Wang is with the National Mobile Communications Research Laboratory, Southeast University, Nanjing, China (Email: jhwang@seu.edu.cn).

The work of D. W. K. Ng was supported under Australian Research Councils Discovery Early Career Researcher Award funding scheme (DE170100137). The work of R. Schober was supported by the Alexander von Humboldt Professorship Program. The work of V.W.S. Wong was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). The work of J. Wang was supported by the National Natural Science Foundation of China under Grant 61571107, the Natural Science Foundation of Jiangsu Province under Grant BK20160069, and the Alexander von Humboldt Foundation.

mission and thus introduces inexpensive *spatial multiplexing gains*. Similar ideas were investigated for cooperative relaying in [13], cooperative small cell transmission in [14], and multicast beamforming in [15]. In [16], the authors proposed a channel-aware scheduling scheme for transmitting cached data in a one-hop wireless network, which enabled *multi-user diversity gains*. These cache-induced benefits in physical layer transmission and resource allocation are referred to as *microscopic* caching gains and are tightly coupled with the wireless channel characteristics.

From both the macroscopic (mostly upper-layer) and the microscopic (physical-layer) perspectives, wireless caching shows great promise for VoD streaming in future cellular networks. In fact, wireless caching has been part of a wider trend towards user-centric networking in 5G [17]. For either purpose, the same problem has gained significant importance, i.e., *how to utilize wireless caching for providing premium streaming quality of experience (QoE) such as low streaming latency and high grade of service, which are highly dependent on the delivery time*. This problem has not been well addressed in the existing works [4]–[8], [10]–[16]. In particular, orthogonal delivery of cached and uncached video data has been considered in [4]–[8], [10]–[16] and the resulting performance gains only manifest themselves in the delivery of cached data. Whenever the requested files are uncached or only partially cached, e.g. due to insufficient cache capacity or inaccurate user preference profiles[1], the delivery of the uncached video data still suffers from the capacity limitations in the backhaul/RAN. Consequently, the streaming QoEs can vary significantly across the users in the network. To address the QoE issues, joint wireless caching and buffering is proposed in this paper for enabling *fast* video delivery, which reduces the overall video delivery time with a minimum delivery rate guarantee. Our work also meets the recent paradigm shift from Real-Time Streaming Protocol (RTSP)-based [18] to Hypertext Transfer Protocol (HTTP)-based [19], [20] wireless VoD streaming in the industry. The new streaming method advocates "as-fast-as-possible" download for video data delivery and meanwhile provides deterministic quality of service (QoS) guarantees [19]. To the best of our knowledge, however, the cross-layer optimization of fast video delivery has not been systematically studied even for non-caching networks.

We consider small cell networks where half-duplex relay nodes (RNs) serve as small cell BSs, cf. Fig. 1(a). Fast delivery is achieved by exploiting the cache and the buffer equipped at each RN to overcome both the backhaul capacity bottleneck and the half-duplex relaying constraint[2]. Thereby, the cache is used as a long-term memory to store a certain amount of video data at the RNs before delivery starts. The cached video data can be instantly delivered to the requesting users in only one wireless hop, without the involvement of the backhaul. The uncached video data, however, has to be fetched and delivered in two wireless hops. Thus, the delivery of uncached video data constitutes a performance bottleneck, which is mitigated to some extent by the buffer.

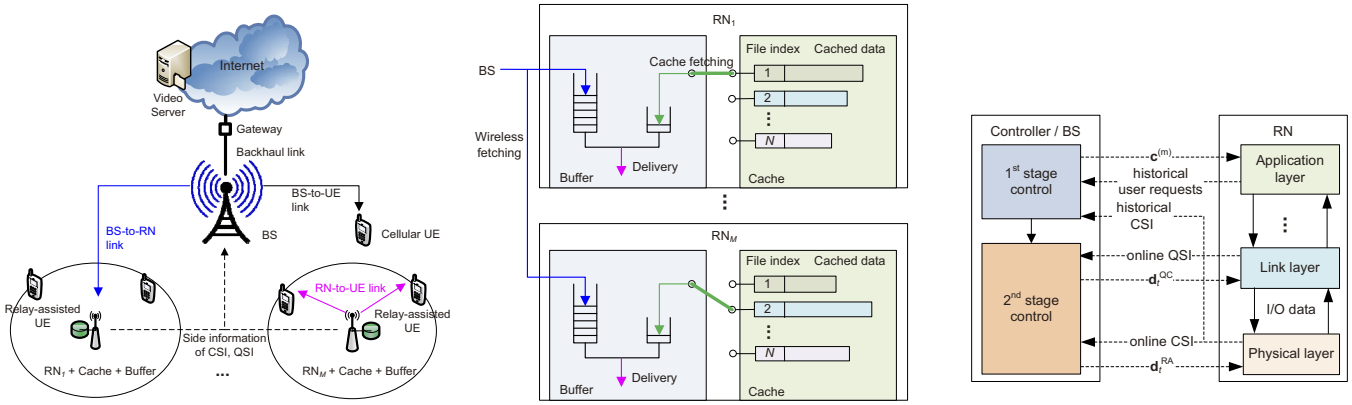
[1]Compared to CCNs, learning the users' preferences in wireless caching networks is more difficult due to the lack of traffic aggregation [5]. Yet, joint caching and buffering can reduce the performance loss caused by inaccurate knowledge of the users' preferences.

[2]For example, the RNs can be roadside units (RSUs) in wireless vehicular networks. Joint buffering and caching can be applied at these RSUs to improve the QoS of vehicular video streaming and location-based services in mobile environments [21], [22].

The buffer is activated during delivery and serves as a short-term memory to temporally store the data packets fetched from either the video server (referred to as "wireless fetching") or the cache before delivery to the requesting users. If the requested files are uncached, the buffer enables buffer-aided relaying (BaR) [23], [24], where wireless fetching and delivery links are adaptively scheduled in each time slot based on the instantaneous channel state information (CSI) and the video data buffered at the RNs. BaR leads to significant performance gains compared to conventional half-duplex relaying and avoids the self-interference typical for full-duplex relaying [24]. For the problem at hand, the channel diversity gains introduced by BaR can effectively improve the delivery of uncached video data. On the other hand, if the requested files are (partially) cached, the probability of an empty buffer for BaR is decreased as the cached video data can feed the buffer with negligible delay. Hence, the joint design of the cache and buffer operation facilitates additional diversity gains for fast delivery of the uncached video data because of the increased flexibility.

To strike a balance between system performance and hardware cost, the cache and the buffer are implemented in a two-level memory architecture. Specifically, the cache requires a larger memory capacity that matches the sizes of the requested video files, while a lower input/output (I/O) speed can be tolerated since the user preferences usually vary slowly; thus, the cache is allotted in the cheap secondary memory such as a hard disk. In contrast, the buffer requires a higher I/O speed to respond to the instantaneous variations of the wireless/backhaul channel, but a moderate memory capacity that depends on the capacity of the wireless/backhaul channel as well as its rate of variations is sufficient; therefore, the buffer is alloted in the expensive primary memory, e.g. a random access memory (RAM) or a dynamic RAM. Given the memory architecture, what portion of a file is cached and thereafter how to adjust the resource allocation have to be carefully optimized in order to ensure fast video delivery via joint caching and buffering. In the conference version of this work, we have investigated joint caching and buffering for non-HTTP VoD streaming in [1], where the optimal offline control is studied. In this paper, we extend [1] to address HTTP VoD streaming. The main contributions are summarized as follows:

- We investigate the cross-layer optimization of fast video delivery in a two-hop relay network where a buffer and a cache are utilized to minimize the overall delivery time. Different from [1], where only offline solutions were considered, we study both the optimal offline and the optimal online solutions for fast video delivery control.
- We formulate the caching and delivery control as a two-stage optimization problem. In the first stage, the cache status is optimized based on historical profiles of user requests and CSI. In the second stage, cross-layer fast delivery control aims to minimize the delivery time of the users in the network for a given cache status.
- The offline caching and delivery optimization problem for given user requests and deterministic CSI is functional (i.e., the feasible set dynamically varies with the values of the optimization variables [25]) and non-convex. However, a novel solution method exploiting the underlying quasi-convex and convex structures is proposed to solve the problem efficiently. We prove the global optimality of the obtained solution under mild conditions, where, different



(a) (b) (c)

Fig. 1. Joint caching and buffering at the relay nodes for improving the delivery of both cached and uncached video data: (a) network model; (b) schematic of cache and buffer operations at the relay nodes; (c) illustration of two-stage cross-layer control at the controller at the BS.

from [1], a self-contained proof is provided.

- The online delivery optimization for statistical CSI (i.e., the CSI distribution) knowledge is solved based on a stochastic dynamic programming (DP) framework. We also propose a low-complexity suboptimal delivery scheme, which only requires instantaneous CSI and achieves close-to-optimal performance in the high buffer capacity regime.
- Simulation results show that caching and buffering can effectively reduce the overall video delivery time by more efficiently utilizing the radio resources. The trade-off between caching and buffering gains is also investigated.

The remainder of this paper is organized as follows. We present the system model in Section II. The offline cache placement and the online cross-layer delivery optimization problems are investigated in Sections III and IV, respectively. In Section V, simulation results are provided and finally, Section VI concludes the paper.

Notation: Throughout the paper, vectors are denoted in bold lower case letters; $\mathbb{N}$ and $\mathbb{R}_+$ are the sets of natural numbers and non-negative real numbers, respectively; $\mathbf{P}$ and $\mathbf{E}$ are the probability and expectation operators, respectively; $\lceil \cdot \rceil$ denotes the ceiling function; $\mathcal{X} \times \mathcal{Y}$ denotes the Cartesian product of sets $\mathcal{X}$ and $\mathcal{Y}$; and $\mathbf{x} \succeq \mathbf{y}$ indicates that each element of $\mathbf{x} - \mathbf{y}$ is non-negative.

II. System Model

In this section, the system model of the cache- and buffer-enabled two-hop relaying network is presented.

A. Cache- and Buffer-Enabled Delivery System

Consider a time-slotted video delivery system, which consists of a BS and M overlaid half-duplex RNs (with index set $\mathcal{M} = \{1, \ldots, M\}$), cf. Fig. 1(a). The time index $t \in \mathbb{N}$ is a non-negative integer and the duration of a time slot Δ is fixed. The video server located in the Internet has N video files (with index set $\mathcal{N} = \{1, \ldots, N\}$) available for delivery. Assume that file $n \in \mathcal{N}$ consists of V_n bits, which are encoded *a priori* into segments. Each segment contains several seconds to minutes of video data and can be decoded independent of other segments.

Each RN $m \in \mathcal{M}$ serves $K^{(m)}$ user equipments (UEs) (with index set $\mathcal{K}^{(m)} = \{1, \ldots, K^{(m)}\}$). We assume that there is no direct communication between the UEs served by the RNs and the BS. Instead, the RNs forward the user requests to the BS. In response, the BS fetches video files from the video server via a capacity-limited backhaul link and delivers them to the (relay-assisted) UEs via the RNs. The RNs perform decode-and-forward (DF) relaying and communicate with the UEs and the BS by alternating between transmission and reception. Different from the conventional systems, a cache and a buffer are deployed at each RN for improving the overall delivery performance. We generally assume finite memory capacities for both the cache and the buffer.

The system adopts an HTTP-based VoD streaming protocol with sever/BS side delivery control, as proposed in [19, Fig. 3(d) therein]. The download of each requested file consists of a series of HTTP sessions between the video server and the requesting UE. One or multiple segments are streamed in a session so that the UEs can decode and play them back without having to wait for the whole file to be download. In this paper, only the delivery time for the streaming (downloading) sessions is considered. The latency caused by user triggered interrupts during video delivery (e.g. video pause, forward, etc.) and the additional time introduced in adaptation of the transmission process are ignored.

B. Two-Stage Caching and Delivery Control

The system operation has two stages: cache placement and delivery. During placement, the portion of the video files to be proactively cached are optimized for maximizing the efficacy of the finite-capacity cache memory. Assume that each segment of file n is further encoded via rateless maximum distance separable (MDS) codes [10]. The encoded parity bits for each segment have the same size as the original segment. For each segment of file n, we can cache $c_n^{(m)} \in [0, 1]$ portion of its parity bits in RN m ahead of time. The cache allocation vector at RN m, which is normalized with respect to the file sizes, is denoted by $\mathbf{c}^{(m)} = [c_1^{(m)}, \ldots, c_N^{(m)}]$. All cache placements have to satisfy

$$\text{C1: } \sum_{n=1}^{N} c_n^{(m)} V_n \leq C_{\max}^{(m)}, \ \forall m \in \mathcal{M}, \tag{1}$$

where $C_{\max}^{(m)}$ is the cache memory capacity of RN m. We assume that the users' preferences vary slowly over time. Hence, the cache can be updated infrequently, e.g. during the off-peak network traffic periods in the early mornings.

In the delivery stage, the cached video data and the buffer are utilized to facilitate fast video delivery with QoS guarantee. We assume that there are S synchronized downloading sessions

over time, where $S \geq 1$. The session index is given by $s \in \{1, \ldots, S\}$. Let $\boldsymbol{\rho} \equiv (m, k, n, s)$ represent a request for file n by user k connected to RN m in session s and $V_{\boldsymbol{\rho}}$ be the amount of its requested video data, where $V_{\boldsymbol{\rho}} \leq V_n$. For example, if the video segments are equally divided across S sessions, we simply have $V_{\boldsymbol{\rho}} = V_n/S$.

Denote the set of user requests in RN m by $\mathcal{G}_s^{(m)} \subseteq \mathcal{K}_s^{(m)} \times \mathcal{N}$, where $\mathcal{G}_s = \bigcup_{m=1}^{M} \mathcal{G}_s^{(m)}$ is the set of user requests in the network. When request $\boldsymbol{\rho} \in \mathcal{G}_s$ is received, RN m searches the file indices in its cache. According to the cache status, $c_n^{(m)} V_{\boldsymbol{\rho}}$ parity bits of file n can be directly fetched from the cache for delivery to UE k. For correct decoding of the video segments at UE k, the remaining $(1 - c_n^{(m)})$ portion has to be further fetched from the video server [10], i.e., obtained via *wireless fetching*. With the aid of the buffer, the video data from the wireless fetching and the cache fetching as well as their delivery are jointly scheduled at RN m to enable fast delivery. To this end, efficient resource allocation and queue control schemes are required and will be presented in the remainder of this section.

C. Cross-Layer Resource Allocation for Joint File Fetching and Delivery

Consider a multi-RN multi-user delivery system with fading wireless channels and possibly time-varying backhaul capacities. The system employs orthogonal frequency division multiple access (OFDMA) at the physical layer (e.g. as in the 4G LTE standard [26]), which can effectively combat frequency-selective fading in wireless channels. We investigate cross-layer resource allocation, i.e., joint adaptive subcarrier (SC) assignment, link scheduling, and power allocation at the physical/link layer, to fully utilize the radio resources, the cache, and the buffer for fast video delivery under a minimum delivery rate guarantee. The optimal cross-layer solution provides a performance upper bound for separate optimization of each layer. We note that cross-layer control has been widely adopted for radio resource allocation [27]–[29], particularly for supporting resource-intensive wireless video transmission [30], due to its superior performance.

1) Joint SC Assignment, Link Scheduling, and Power Allocation: The RNs employ time-division duplexing and may switch between transmission and reception within each time slot. In general, we assume that the wireless BS-to-RN m, $m \in \mathcal{M}$, fetching links and the RN m-to-UE k, $k \in \mathcal{K}_n^{(m)}$, delivery links are activated for $\eta_{\mathsf{S},t}^{(m)} \in [0,1]$ and $\eta_{\mathsf{R},t}^{(m)} \in [0,1]$ fractions of time slot t, respectively, where the subscript indices S and R stand for "source" and "relay" transmissions, respectively. $\eta_{\mathsf{S},t}^{(m)}$ and $\eta_{\mathsf{R},t}^{(m)}$ satisfy $\eta_{\mathsf{S},t}^{(m)} + \eta_{\mathsf{R},t}^{(m)} = 1,\ \forall t, m$. Without loss of generality, we assume that time t is reset before each downloading session starts.

Meanwhile, the available frequency spectrum comprises F orthogonal SCs (with index set $\mathcal{F} = \{1, ..., F\}$), where each SC has a bandwidth of W Hz. We consider flat fading on each SC and assume that the duration of a time slot is less than the channel coherence time. The SCs are assigned for each BS-to-RN m-to-UE k link in each time slot. Due to time switching at time t, each SC-time slot channel, denoted by (f,t), is split into two subchannels, denoted by (f,t,i), $i \in \{\mathsf{S}, \mathsf{R}\}$, for use on the BS-to-RN and RN-to-UE links, respectively. We use $\mu_{\boldsymbol{\rho},f,t} \in \{0,1\}$ and $\mu_{\boldsymbol{\rho},f,t}^i \in [0,1]$ to indicate the assignment and the occupancy of channel (f,t) on the corresponding links for serving request $\boldsymbol{\rho} \in \mathcal{G}_s$, respectively. To avoid interference, orthogonal allocation of the SC-time resources to different links is assumed, i.e.,

$$\text{C2: } \mu_{\boldsymbol{\rho},f,t} = \mu_{\boldsymbol{\rho},f,t}^{\mathsf{S}} + \mu_{\boldsymbol{\rho},f,t}^{\mathsf{R}}, \quad \forall \boldsymbol{\rho} \in \mathcal{G}_s, f \in \mathcal{F}, \forall t,$$
$$\text{C3: } 0 \leq \mu_{\boldsymbol{\rho},f,t}^i \leq \eta_{i,t}^{(m)}, \forall \boldsymbol{\rho} \in \mathcal{G}_s, f \in \mathcal{F}, i \in \{\mathsf{S}, \mathsf{R}\}, \forall t,$$
$$\text{C4: } \mu_{\boldsymbol{\rho},f,t} \in \{0,1\}, \quad \forall \boldsymbol{\rho} \in \mathcal{G}_s, f \in \mathcal{F}, \forall t,$$
$$\text{C5: } \eta_{\mathsf{S},t}^{(m)} + \eta_{\mathsf{R},t}^{(m)} = 1, \forall m \in \mathcal{M}, \forall t,$$
$$\text{C6: } \sum\nolimits_{f \in \mathcal{F}} \mu_{\boldsymbol{\rho},f,t} \leq 1,\ \forall \boldsymbol{\rho} \in \mathcal{G}_s, \forall t,$$
$$\text{C7: } \sum\nolimits_{\boldsymbol{\rho} \in \mathcal{G}_s} \mu_{\boldsymbol{\rho},f,t} \leq 1,\ \forall f \in \mathcal{F}, \forall t,$$

where C3 specifies that the BS-to-RN m link is active if $\eta_{\mathsf{S},t}^{(m)} > 0$ and $\mu_{\boldsymbol{\rho},f,t}^{\mathsf{S}} > 0$; similarly, the RN m-to-UE k link is active if $\eta_{\mathsf{R},t}^{(m)} > 0$ and $\mu_{\boldsymbol{\rho},f,t}^{\mathsf{R}} > 0$. C6 and C7 guarantee that each SC is assigned only once in a time slot and to at most one BS-to-RN-to-UE link. Moreover, let $h_{\boldsymbol{\rho},f,t}^i$ and $p_{f,t}^i$ denote the instantaneous channel state and the transmit power allocation on subchannel $(f,t,i), i \in \{\mathsf{S}, \mathsf{R}\}$, respectively, where $p_{f,t}^i \geq 0$ is subject to optimization. Note that, $h_{\boldsymbol{\rho},f,t}^{\mathsf{S}} = h_{\boldsymbol{\rho}',f,t}^{\mathsf{S}}$ holds for all user requests $\boldsymbol{\rho}, \boldsymbol{\rho}' \in \mathcal{G}_s^{(m)}$ in each RN m. The maximum transmit power of the BS and the RNs are constrained to P_{S} and P_{R}, respectively, i.e.,

$$\text{C8: } \sum\nolimits_{f \in \mathcal{F}} \sum\nolimits_{\boldsymbol{\rho} \in \mathcal{G}_s^i} \mu_{\boldsymbol{\rho},f,t}^i p_{f,t}^i \leq P_i,\ i \in \{\mathsf{S}, \mathsf{R}\}, \forall t, \quad (2)$$

where the summation is taken over $\mathcal{G}_s^{\mathsf{S}} = \mathcal{G}_s$ and $\mathcal{G}_s^{\mathsf{R}} = \mathcal{G}_s^{(m)}$, respectively.

2) Cache/Buffer Management at the RNs: We assume that the buffer uses two first-in first-out (FIFO) queues for wireless fetching and cache fetching, respectively, cf. Fig. 1(b). The two queues are cooperatively controlled, i.e., performing joint fetching and delivery, in accordance with the proposed resource allocation policy to exploit the channel diversity. Let $b_{\boldsymbol{\rho},t}^{\mathsf{S}}$, $b_{\boldsymbol{\rho},t}^{\mathsf{C}}$, and $b_{\boldsymbol{\rho},t}^{\mathsf{R}}$ be the instantaneous rates of wireless fetching, cache fetching, and delivery in serving request $\boldsymbol{\rho} \in \mathcal{G}_s$ at time t, respectively. For given $h_{\boldsymbol{\rho},f,t}^i$ and $p_{f,t}^i$, the capacity of subchannel (f,t,i) is $\log_2(1 + p_{f,t}^i h_{\boldsymbol{\rho},f,t}^i / (N_0 W))$ bps/Hz, where N_0 is the noise power spectral density and $i \in \{\mathsf{S}, \mathsf{R}\}$. Hence, for wireless fetching and delivery, we have

$$b_{\boldsymbol{\rho},t}^i = W \sum\nolimits_{f \in \mathcal{F}} \mu_{\boldsymbol{\rho},f,t}^i \log_2 \left(1 + \frac{p_{f,t}^i h_{\boldsymbol{\rho},f,t}^i}{N_0 W}\right),\ i \in \{\mathsf{S}, \mathsf{R}\}, \quad (3)$$

and for cache fetching, we only require $b_{\boldsymbol{\rho},t}^{\mathsf{C}} \geq 0$. Moreover, let $B_{\boldsymbol{\rho},t}^{\mathsf{S}}$ and $B_{\boldsymbol{\rho},t}^{\mathsf{C}}$ be the total amount of video data fetched from the server and the cache up to time t, respectively, and $B_{\boldsymbol{\rho},t}^{\mathsf{R}}$ be the total amount of delivered video data. The trajectory of the queue evolution over time is then given by

$$B_{\boldsymbol{\rho},t}^i = B_{\boldsymbol{\rho},t-1}^i + \Delta b_{\boldsymbol{\rho},t}^i = \Delta \sum\nolimits_{\tau=1}^{t} b_{\boldsymbol{\rho},\tau}^i,\ i \in \{\mathsf{S}, \mathsf{R}, \mathsf{C}\}. \quad (4)$$

In general, the queue evolution has to satisfy the following boundary constraints,

$$\text{C9: } B_{\boldsymbol{\rho},t}^{\mathsf{R}} \leq \min\left\{B_{\boldsymbol{\rho},t}^{\mathsf{S}} + B_{\boldsymbol{\rho},t}^{\mathsf{C}},\ V_{\boldsymbol{\rho}}\right\},\ \forall \boldsymbol{\rho} \in \mathcal{G}_s, \forall t,$$
$$\text{C10: } B_{\boldsymbol{\rho},t}^{\mathsf{C}} \leq c_n^{(m)} V_{\boldsymbol{\rho}},\ \forall \boldsymbol{\rho} \in \mathcal{G}_s, \forall t,$$
$$\text{C11: } B_t^{(m)} + \sum\nolimits_{\boldsymbol{\rho} \in \mathcal{G}_s^{(m)}} b_{\boldsymbol{\rho},t}^{\mathsf{S}} \leq B_{\max}^{(m)},\ \forall m, \forall t,$$
$$\text{C12: } \sum\nolimits_{\boldsymbol{\rho} \in \mathcal{G}_s} \left(B_{\boldsymbol{\rho},t}^{\mathsf{S}} - B_{\boldsymbol{\rho},t-1}^{\mathsf{S}}\right)/\Delta \leq \gamma_t,\ \forall t,$$

where C9 and C10 guarantee data causality of the buffer/cache, i.e., the amount of delivered video data cannot exceed the

aggregate cached/buffered video data nor the video file size; C11 constrains the instantaneous queue length at the RNs to be less than the buffer capacity $B_{\max}^{(m)}$, where $B_t^{(m)}$ is the queue length of RN m at the beginning of time t satisfying $B_t^{(m)} = \sum_{\boldsymbol{\rho}\in\mathcal{G}_s^{(m)}} (B_{\boldsymbol{\rho},t-1}^{\mathsf{S}} + B_{\boldsymbol{\rho},t-1}^{\mathsf{C}} - B_{\boldsymbol{\rho},t-1}^{\mathsf{R}})$; and C12 ensures that during time slot duration Δ, the average sum rate of wireless fetching does not exceed the backhaul capacity γ_t, i.e., the backhaul is shared by all BS-to-RN links while fetching video data from the video server.

3) Video Delivery with QoS Guarantee: In the application layer, we assume that UEs have a large enough memory to store the downloaded video segments during each session. After the parity bits of a video segment are completely downloaded, the video segment is decoded and played back. Due to the advantages of MDS coding, the parity bits of each video segment can be received in arbitrary order without any impact on the decoding results. The cross-layer controller maintains a minimum delivery rate $\nu_{\min}$ at the UEs to guarantee a certain QoS [31],

$$\text{C13: } B_{\boldsymbol{\rho},t}^{\mathsf{R}} \geq \nu_{\min} \cdot \{\max\{t-\epsilon_{\boldsymbol{\rho}}, 0\}\}, \quad \forall \boldsymbol{\rho} \in \mathcal{G}_s, \forall t, \tag{5}$$

where $\epsilon_{\boldsymbol{\rho}}$ is the initial delivery delay and is known to the controller through feedback. When $t > \epsilon_{\boldsymbol{\rho}}$, C13 reduces to $B_{\boldsymbol{\rho},t}^{\mathsf{R}}/(t-\epsilon_{\boldsymbol{\rho}}) \geq \nu_{\min}$, i.e., a minimum time-averaged delivery rate is maintained.

Denote the CSI on the wireless/backhaul channels and the queue status information (QSI) by $\mathbf{h}_t \triangleq [h_{\boldsymbol{\rho},f,t}^{\mathsf{S}}, h_{\boldsymbol{\rho},f,t}^{\mathsf{R}}, \gamma_t]$ and $\mathbf{q}_t \triangleq [B_{\boldsymbol{\rho},t}^{\mathsf{S}}, B_{\boldsymbol{\rho},t}^{\mathsf{R}}, B_{\boldsymbol{\rho},t}^{\mathsf{C}}]$, respectively. We assume that a central controller located at the BS is responsible for collecting the CSI and QSI from the RNs and UEs. Moreover, the controller optimizes over the resource allocation vector, denoted by $\mathbf{d}_t^{\mathsf{RA}} \triangleq [\mu_{\boldsymbol{\rho},f,t}, \mu_{\boldsymbol{\rho},f,t}^i, \eta_{i,t}^{(m)}, p_{f,t}^i]_{i\in\{\mathsf{S},\mathsf{R}\}}$, for joint SC assignment, link scheduling, and power allocation and the queue control vector, denoted by $\mathbf{d}_t^{\mathsf{QC}} \triangleq [b_{\boldsymbol{\rho},t}^{\mathsf{S}}, b_{\boldsymbol{\rho},t}^{\mathsf{C}}, b_{\boldsymbol{\rho},t}^{\mathsf{R}}]$, for cache/buffer management. The optimal decisions are then broadcast to the RNs[3]. The proposed two-stage cross-layer control is illustrated in Fig. 1(c). Note that vectors $\mathbf{d}_t^{\mathsf{RA}}$ and $\mathbf{d}_t^{\mathsf{QC}}$ are related by (3) and (4), or $\mathbf{d}_t^{\mathsf{QC}} = \mathbf{g}\left(\mathbf{d}_t^{\mathsf{RA}}, \mathbf{h}_t\right)$ for short. If treating $b_{\boldsymbol{\rho},t}^{\mathsf{C}}$ as an auxiliary variable, $\mathbf{g}$ is an (affine) surjective function [33]. Thus, $\mathbf{d}_t^{\mathsf{QC}}$ is known once $\mathbf{d}_t^{\mathsf{RA}}$ is determined. This can be exploited for system design in two manners: (i) by eliminating $\mathbf{d}_t^{\mathsf{QC}}$ according to $\mathbf{g}$, the offline optimization can be formulated as a single resource allocation problem (with the aid of caching/buffering), cf. Section III; (ii) by decomposing the optimization space of $\mathbf{d}_t^{\mathsf{QC}}$ and $\mathbf{d}_t^{\mathsf{RA}}$, the complexity of the online optimization can be reduced, cf. Section IV.

Remark 1. Although the above model focuses on relay-assisted UEs, it is general enough to also include UEs communicating with the BS directly, cf. Fig. 1(a). Consider that user k' is served directly by the BS for delivering file n'. This is equivalent to defining the request $\boldsymbol{\rho}' = (m', k', n', s)$ to be served by a "virtual" RN m' if and only if the CSI and the QSI satisfy $h_{\boldsymbol{\rho}',t}^{\mathsf{S}} = h_{\boldsymbol{\rho}',t}^{\mathsf{R}}$ and $B_{\boldsymbol{\rho}',t}^{\mathsf{S}} = B_{\boldsymbol{\rho}',t}^{\mathsf{R}}$, respectively. Hence, the one-hop communication between UEs and the BS is included as a special case of the considered two-hop communication model.

Remark 2. The considered system design improves performance at the expense of an increased hardware cost and computational complexity. In particular, the performance benefits enabled by joint buffering and caching at the RNs introduce additional hardware costs. Besides, compared to either separate optimization conducted in each layer or decentralized optimization, the proposed centralized cross-layer optimization provides a better performance but entails a higher computational complexity as well as a higher feedback overhead. However, as will be shown below, caching and buffering can effectively mitigate the radio resource scarcity in wireless cellular networks by utilizing a new type of resource, namely, storage memory. Moreover, these benefits can be obtained via polynomial-time optimization algorithms, which can be effectively implemented in practical systems. Since the radio spectrum is limited, investing in memory and computing resources may be an appealing option to enhance the QoE of video streaming.

[3]The CSI and QSI feedback and the decision broadcasting may incur a significant signaling overhead for highly dense RN deployments. In this case, quantization methods can be applied to reduce the amount of information exchange needed [32].

III. Offline Cross-Layer Caching and Delivery Control

In this section, the cross-layer caching and delivery control is formulated as a two-stage optimization problem. In the first stage, the cache controller optimizes the cache status based on historical profiles of user requests and CSI measurements. For given user requests and cache status, cross-layer delivery control is performed in the second stage, which aligns the resource allocation and the queue management decisions for minimizing the overall delivery time. Assuming full knowledge of the CSI of all links over a sufficiently long time period, an offline algorithm is proposed to solve the two-stage optimization problem. The offline caching algorithm is well suited for a historical data driven caching control [34]. The offline delivery control provides a performance upper bound for the minimum video delivery time, based on which online delivery schemes requiring only causal CSI knowledge are further studied in Section IV. In the following, we first study the simpler delivery control problem of the second stage before tackling the caching problem of the first stage.

A. Second Stage Fast Delivery Control

For a given cache status, the *delivery time* is defined as the number of time slots needed to complete the file delivery for all users in each session. Denote the delivery time for session s by $T_{\mathrm{a},s}$, where $T_{\mathrm{a},s} \in \mathbb{N}$ and $s \in \{1,\ldots,S\}$. Let $\mathbf{d}(T_{\mathrm{a},s}) = [\mathbf{d}_t^{\mathsf{RA}}, b_{\boldsymbol{\rho},t}^{\mathsf{C}}]_{t=0}^{T_{\mathrm{a},s}}$ be the delivery control vector (where $b_{\boldsymbol{\rho},t}^{\mathsf{S}}$ and $b_{\boldsymbol{\rho},t}^{\mathsf{R}}$ are eliminated according to $\mathbf{g}$) belonging to the feasible set $\mathcal{D}(T_{\mathrm{a},s}) \triangleq \{\mathbf{d} \mid \mathbf{d} \succeq \mathbf{0},$ C2–C13$\}$. Both $\mathbf{d}$ and $\mathcal{D}$ depend on the delivery time $T_{\mathrm{a},s}$.

Given the delivery file sizes $V_{\boldsymbol{\rho}}$ in session s, the cross-layer scheduler computes the optimal delivery vector $\mathbf{d}$ that minimizes $T_{\mathrm{a},s}$ subject to the constraints at the physical, data link, and application layers. Then, for each session s, the following offline delivery time minimization problem is considered,

$$\begin{aligned} \text{(P1)} \quad & \underset{T_{\mathrm{a},s}\in\mathbb{N},\, \mathbf{d}\in\mathcal{D}(T_{\mathrm{a},s})}{\text{minimize}} \quad T_{\mathrm{a},s} \\ & \text{subject to} \quad \text{C14: } B_{\boldsymbol{\rho},T_{\mathrm{a},s}}^{\mathsf{R}} = V_{\boldsymbol{\rho}},\ \forall \boldsymbol{\rho} \in \mathcal{G}_s, \end{aligned} \tag{6}$$

where constraint C14 indicates delivery completion at $T_{\mathrm{a},s}$.

Problem (P1) is a mixed-combinatorial and non-convex optimization problem. It involves the integer optimization variables $\mu_{\boldsymbol{\rho},f,t}$ and $T_{\mathrm{a},s}$ and non-convex constraints including C8, which contains bilinear terms, and C9, C10, C11, C12, which

involve differences of convex functions. This type of problem is generally intractable [35]. Moreover, different from typical resource allocation problems defined over a fixed time period, (P1) admits a *free* delivery time $T_{\mathrm{a},s}$ [25, Section 3.4.3], which is to be determined via optimization. This endows (P1) with the *functional* attribute since the optimal delivery time $T^*_{\mathrm{a},s}$ hinges on the profile of the CSI and the delivery control over the period $[0,\, T^*_{\mathrm{a},s}]$, which in turn depends on $T^*_{\mathrm{a},s}$ itself. As a result, in searching for the optimal value $T^*_{\mathrm{a},s}$, the feasible set $\mathcal{D}(T_{\mathrm{a},s})$ and the set of relevant CSI vary dynamically with the values of $T_{\mathrm{a},s}$, which further complicates the solution of (P1).

To overcome the above difficulties, a novel solution method is proposed based on suitable decomposition and transformation techniques. Rather than solving the functional problem (P1) directly, we first decompose it into two layers of subproblems. Thereby, the outer-layer subproblem is concerned with the delivery time minimization while the inner-layer subproblem is reduced to a typical resource allocation optimization for a given delivery time. We find that the structure of the two-layer decomposition is not unique by which several instances of decomposition are constructed in Theorem 1; see (D1a) and (D1b) below.

For notational convenience, let $\mathbb{I}_{\mathcal{A}}(x)$ be the indicator function defined over set $\mathcal{A}$,

$$\mathbb{I}_{\mathcal{A}}(x) \triangleq \begin{cases} 0, & \text{if } x \in \mathcal{A}, \\ \infty, & \text{otherwise}, \end{cases}$$

which indicates the membership of x in $\mathcal{A}$. Theorem 1 is then stated as follows.

Theorem 1. (Two-layer decomposition of (P1)) Let $T^*_{\mathrm{a},s}$ be the optimal delivery time of (P1). Define

$$\text{(D1)} \quad T^*_{i,s} \triangleq \min_{T_{i,s}\in\mathbb{N}\cup\{\infty\}} \; T_{i,s} + \mathbb{I}_{\mathcal{A}_{i,s}}(T_{i,s}). \tag{7}$$

Then, $T^*_{\mathrm{a},s} = T^*_{i,s}$ holds for the $\mathcal{A}_{i,s}$ given in the following: i) $\mathcal{A}_{1,s} \triangleq \left\{T_{1,s} \mid \beta_{\mathrm{a}}(T_{1,s}) = \sum_{\boldsymbol{\rho}\in\mathcal{G}_s} V_{\boldsymbol{\rho}}\right\}$, where $\beta_{\mathrm{a}}(T_{1,s})$ is the maximum amount of data delivered within $[0,\, T_{1,s}]$,

$$\text{(D1a)} \quad \beta_a(T_{1,s}) \triangleq \max_{\mathbf{d}\in\mathcal{D}(T_{1,s})} \quad \sum\nolimits_{\boldsymbol{\rho}\in\mathcal{G}_s} B^{\mathsf{R}}_{\boldsymbol{\rho},T_{1,s}}. \tag{8}$$

ii) $\mathcal{A}_{2,s} \triangleq \left\{T_{2,s} \mid \sum_{\boldsymbol{\rho}\in\mathcal{G}_s} B^{\mathsf{R}}_{\boldsymbol{\rho},T_{2,s}} = \sum_{\boldsymbol{\rho}\in\mathcal{G}_s} V_{\boldsymbol{\rho}}\right\}$, that is, the $B^{\mathsf{R}}_{\boldsymbol{\rho},T_{2,s}}$'s are feasible for the following problem

$$\begin{aligned} \text{(D1b)} \quad & \underset{\mathbf{d}\in\mathcal{D}(T_{2,s})}{\text{maximize}} \quad 1 \\ & \text{subject to} \quad \sum\nolimits_{\boldsymbol{\rho}\in\mathcal{G}_s} B^{\mathsf{R}}_{\boldsymbol{\rho},T_{2,s}} = \sum\nolimits_{\boldsymbol{\rho}\in\mathcal{G}_s} V_{\boldsymbol{\rho}}. \end{aligned} \tag{9}$$

Several remarks are in order. First, applying the indicator function $\mathbb{I}_{\mathcal{A}_{i,s}}(T_{i,s})$ in Theorem 1, the feasible set (as well as the range of the objective function) of problem (D1) is extended onto $\mathbb{N}\cup\{\infty\}$ while the set of delivery times (i.e., the set of $T_{i,s}$ satisfying $\mathbb{I}_{\mathcal{A}_{i,s}}(T_{i,s}) = 0$) is constrained to $\mathcal{A}_{i,s}\cap\mathbb{N}$, $i = 1, 2$. As a result, the feasibility of $T_{i,s}$ in problems (D1a) and (D1b) is indicated by the value of $\mathbb{I}_{\mathcal{A}_i}(T_{i,s})$ directly, where $\mathcal{A}_{i,s}$ depends on the resource allocation policies defined in problems (D1a) and (D1b), respectively.

Second, due to Theorem 1 and the fact that $\min_{x,y} f(x,y) = \min_x [\min_y f(x,y)]$ [36, Section 4.1.3], (P1) can be decomposed into two layers of subproblems: the inner-layer problems (D1a) and (D1b), which are typical resource allocation optimization or feasibility problems for a given delivery time, and the outer-layer problem (D1), which seeks the optimal delivery time. This result can be understood intuitively: (D1a) and (D1b) construct two candidate sets of delivery trajectories, i.e., the profiles of $\phi(T_{1,s})$ or $\sum_{\boldsymbol{\rho}\in\mathcal{G}_s} B^{\mathsf{R}}_{\boldsymbol{\rho},T_{2,s}}$ over time period $[0, T_{i,s}]$ satisfying the termination condition C14. Due to the $(T^*_{i,s}-1)$ degrees of freedom[4] overall in $\mathcal{D}(T^*_{i,s})$, $i = 1, 2$, however, the decomposition is non-unique. Nevertheless, the optimal delivery trajectory, which achieves the delivery time $T^*_{i,s}$, is subject to the outer-layer optimization in (D1).

Finally, problem (D1a) also maximizes the effective throughput in a given delivery time period $[0, T_{1,s}]$. Problems (P1) and (D1a) are dual in the sense that $T^*_{\mathrm{a},s} = T^*_{1,s}$, i.e., the control policy that maximizes the amount of delivered video data within a given time period also minimizes the time needed for delivering the same amount of video data. A similar duality between delivery time minimization and throughput maximization was also established in the energy harvesting literature [37]. However, we show here that such a duality holds even for a general problem formulation with complicated boundary constraints.

We now provide a sketch of the proof below.

Proof: We first prove ii). It is easy to verify that the delivery completion condition C14 is equivalent to $\mathbb{I}_{\mathcal{A}_{2,s}}(T_{2,s}) = 0$ or $\mathcal{A}_{2,s} \neq \emptyset$, since $\mathbb{I}_{\mathcal{A}_{2,s}}(T_{2,s}) = \sum_{\boldsymbol{\rho}} \mathbb{I}_{\mathcal{A}_{2,s,\boldsymbol{\rho}}}(T_2)$, where $\mathcal{A}_{2,s,\boldsymbol{\rho}}(T) \triangleq \{T \mid T \text{ satisfying C14 for } \boldsymbol{\rho}\}$. When the completion condition is satisfied, the objective functions and the feasible sets of (P1) and (D1) become the same for $i = 2$. Thus, $T^*_{\mathrm{a},s} = T^*_{2,s}$ holds.

Now, we can prove i) based on ii). From problem (D1a), $\sum_{\boldsymbol{\rho}} B^{\mathsf{R}}_{\boldsymbol{\rho},T} \leq \beta_{\mathrm{a}}(T)$ holds for any $T \in \mathbb{N}$. If $\mathbb{I}_{\mathcal{A}_{2,s}}(T) = 0$, then $\mathbb{I}_{\mathcal{A}_{1,s}}(T) = 0$ also holds, i.e., $\mathcal{A}_{1,s} \subseteq \mathcal{A}_{2,s}$. Thus, $T^*_{1,s} \geq T^*_{2,s}$. However, if $T^*_{2,s} \in \mathcal{A}_{2,s} \neq \emptyset$ and $\mathbf{d}^*$ solves problem (D1b), then $\mathbf{d}^*$ is also feasible for problem (D1a) with $T_{1,s} = T^*_{2,s} \in \mathcal{A}_{2,s}$, because $\mathcal{A}_{1,s}$, by its definition, contains all delivery control vectors that can lead to delivery completion. Thus, $T^*_{2,s} \in \mathcal{A}_{1,s}$. We then have $T^*_{1,s} \leq T_{1,s} = T^*_{2,s}$ since $T^*_{1,s}$ is the optimal solution to (D1) for $i = 1$. Therefore, $T^*_{1,s} = T^*_{2,s}$ holds, which completes the proof. ∎

Based on the above two-layer decomposition, we show in Section III-C that the functional difficulty can be divided and then conquered by exploiting the underlying convex and quasi-convex structures. The inner-layer subproblems (D1a) and (D1b), which, similar to problem (P1), are mixed-combinatorial and non-convex optimization problems, can be transformed into equivalent convex problems and then be solved efficiently. Herein, either subproblem (D1a) or (D1b) can be solved in the inner layer with comparable computational complexities. The outer-layer subproblem can be solved via a simple one-dimensional search. Particularly, an efficient bisection method is applicable due to the underlying quasi-convexity of the outer-layer subproblem [36]. The overall solution is shown to be globally optimal for (P1) under mild conditions.

Remark 3. In (P1), we typically set

$$V_{\boldsymbol{\rho}} = V_n/S, \; \boldsymbol{\rho} \in \mathcal{G}_s, \tag{10}$$

for simplicity [19]. (P1) is then solved independently for each session $s \in \{1, \ldots, S\}$ to deliver partial files of size V_n/S. We note that the overall delivery time increases with S as

[4]In the special case of $T^*_{i,s} = 1$, the two candidate sets of delivery trajectories are identical.

the dependencies from one session to the next are ignored, i.e., single-session streaming with $S = 1$ is optimal. This is confirmed by the simulation results in Section V. On the other hand, although (P1) is tractable for arbitrary $S \geq 1$, the computational complexity decreases with S. The interesting trade-off between performance and computational complexity by leveraging S is further characterized and shown in Section V.

B. First Stage Cache Control

We advocate a historical data based cache control to guarantee the streaming QoE for different user requests [34]. The cache status is determined based on historical profiles of user requests and CSI, which are referred to as "scenario" data. We assume that each user requests only one file in a set of scenario data and Ω sets of scenario data can be obtained from the system records. The delivery decision for session $s \in \{1, \ldots, S\}$ of scenario $\omega \in \{1, \ldots, \Omega\}$ is denoted by $\mathbf{d}_{s,\omega}$ with the corresponding feasible set $\mathcal{D}_{s,\omega}(T_{s,\omega})$. However, the caching decisions $\mathbf{c}^{(m)}$ are scenario independent. To obtain a tractable problem formulation, we assume that the caching decisions are optimized for each session individually. We define $\mathbf{c}^{(m)} \triangleq [\mathbf{c}_1^{(m)}, \ldots, \mathbf{c}_S^{(m)}]$, where the feasible set of $\mathbf{c}_s^{(m)} \triangleq [\mathbf{c}_{1,s}^{(m)}, \ldots, \mathbf{c}_{N,s}^{(m)}]$ is given by $\mathcal{C}_s^{(m)} \triangleq \{\mathbf{c}_s^{(m)} \mid \sum_{n=1}^{N} c_{n,s}^{(m)} V_{\boldsymbol{\rho}} \leq C_{\max}^{(m)}/S, \forall m \in \mathcal{M}, \forall \boldsymbol{\rho} \in \mathcal{G}_s\}$.

For providing ubiquitous QoE, the caching control aims to minimize the worst-case delivery time over the Ω pre-selected scenarios for each session $s \in \{1, \ldots, S\}$,

$$\text{(P2)} \quad \underset{\mathbf{c}_s^{(m)} \in \mathcal{C}_s^{(m)}}{\text{minimize}} \quad \max_{\{T_{s,\omega}\} \in \mathbb{N}^{\Omega}} \quad T_{s,\omega} \tag{11}$$
$$\text{subject to} \quad \text{C15: } B^{\mathsf{R}}_{\boldsymbol{\rho},T_{s,\omega}} = V_{\boldsymbol{\rho}}, \ \forall \boldsymbol{\rho} \in \mathcal{G}_s, \forall \omega,$$
$$\mathbf{d}_{s,\omega} \in \mathcal{D}_{s,\omega}(T_{s,\omega}), \ \forall s, \forall \omega,$$

where C15 guarantees delivery completion for each scenario.

Similar to (P1), we perform two-layer decomposition of (P2), cf. Theorem 2.

Theorem 2. (Two-layer decomposition of (P2)) Let $T_{\mathrm{b},s}^*$ be the optimal delivery time of (P2). Define

$$\text{(D2)} \quad T_{i,s}^* \triangleq \min_{T_{i,s} \in \mathbb{N} \cup \{\infty\}} \ T_{i,s} + \mathbb{I}_{\mathcal{B}_{i,s}}(T_{i,s}). \tag{12}$$

Then, $T_{\mathrm{b},s}^* = T_{i,s}^*$ holds for the $\mathcal{B}_{i,s}$ given in the following:
i) $\mathcal{B}_{1,s} \triangleq \left\{T_{1,s} \mid \beta_{\mathrm{b}}(T_{1,s}) = \sum_{\omega} \sum_{\boldsymbol{\rho} \in \mathcal{G}_s} V_{\boldsymbol{\rho}}\right\}$, where $\beta_{\mathrm{b}}(T_{1,s})$ is the maximal amount of data delivered within $[0, T_{1,s}]$,

$$\text{(D2a)} \ \beta_{\mathrm{b}}(T_{1,s}) \triangleq \underset{\mathbf{c}^{(m)} \in \mathcal{C}^{(m)}}{\text{maximize}} \quad \sum_{\omega} \sum_{\boldsymbol{\rho} \in \mathcal{G}_s} B^{\mathsf{R}}_{\boldsymbol{\rho},T_{1,s},\omega} \tag{13}$$
$$\text{subject to} \quad \mathbf{d}_{s,\omega} \in \mathcal{D}_{s,\omega}(T_{1,s}), \ \forall \omega;$$

ii) $\mathcal{B}_{2,s} \triangleq \left\{T_{2,s} \mid \sum_{\omega} \sum_{\boldsymbol{\rho} \in \mathcal{G}_s} B^{\mathsf{R}}_{\boldsymbol{\rho},T_{2,s},\omega} = \sum_{\omega} \sum_{\boldsymbol{\rho} \in \mathcal{G}_s} V_{\boldsymbol{\rho}}\right\}$. In other words, the $B^{\mathsf{R}}_{\boldsymbol{\rho},T_{2,s},\omega}$'s are feasible for problem

$$\text{(D2b)} \quad \underset{\mathbf{c}^{(m)} \in \mathcal{C}^{(m)}}{\text{maximize}} \quad 1 \tag{14}$$
$$\text{subject to} \quad \sum_{\omega} \sum_{\boldsymbol{\rho} \in \mathcal{G}_s} B^{\mathsf{R}}_{\boldsymbol{\rho},T_{2,s},\omega} = \sum_{\omega} \sum_{\boldsymbol{\rho} \in \mathcal{G}_s} V_{\boldsymbol{\rho}}$$
$$\mathbf{d}_{s,\omega} \in \mathcal{D}_{s,\omega}(T_{2,s}), \ \forall \omega.$$

Proof: Let $\max_{\omega} T_{s,\omega} \leq T_{1,s}$, $\forall \omega \in \Omega$, $\forall s \in \{1, \ldots, S\}$. By applying the epigraph transformation to (P2), we obtain (D2a). Note that in (D2a), $\mathcal{D}_{s,\omega}$ becomes dependent on $T_{1,s}$ instead of $T_{s,\omega}$. Then, (D2) can be proved in a similar manner as Theorem 1. The details are omitted here. ∎

Different from Theorem 1, the inner-layer subproblems in Theorem 2 are joint optimization or feasibility problems for cache control and resource allocation. However, the caching and delivery decisions are only coupled via constraints C9, C10, and C11 in (D2a). We reveal in Section III-C that, the same solution techniques are applicable for both (D1a) and (D2a). Thus, problem (P2) can be efficiently solved.

C. Solutions of Problems (P1) and (P2)

1) Solving Non-Convex Inner-Layer Subproblems: For the inner-layer subproblems, taking (D1a) and (D2a) as examples, we apply constraint relaxation to deal with the combinatorial variables $\mu_{\boldsymbol{\rho},f,t}$ and variable transformations to address the non-convex constraints C8, C9, C10, C11, and C12. The solution is summarized in two steps, which is also directly applicable for solving (D1b) and (D2b).

Step 1: Binary Relaxation for the SC Assignment Variables: We relax constraint C4 by extending the binary SC assignment variables to $\mu_{\boldsymbol{\rho},f,t} \in [0, 1]$. The resulting relaxed problem is solved below without adopting complex combinatorial solution methods. We will show in Theorem 3 that the solution of the relaxed problem becomes optimal for (D1a) and (D2a) when the number of SCs is large, i.e., $F \to \infty$. Moreover, as shown in [38]–[40] for the same relaxation, in practical OFDMA systems such as 4G LTE, the number of SCs is sufficiently large such that the optimum solution is closely approached.

Step 2: Equivalent Convex Problem: We introduce two new variables $b^{\mathsf{S}}_{\boldsymbol{\rho},f,t}$ and $b^{\mathsf{R}}_{\boldsymbol{\rho},f,t}$ which denote the effective fetching and delivery rates on subchannel (f, t, i), respectively, i.e.,

$$b^i_{\boldsymbol{\rho},f,t} = W \mu^i_{\boldsymbol{\rho},f,t} \log_2 \left(1 + \frac{p^i_{f,t} h^i_{\boldsymbol{\rho},f,t}}{N_0 W}\right) \geq 0, \ i \in \{\mathsf{S}, \mathsf{R}\}. \tag{15}$$

We have $b^i_{\boldsymbol{\rho},f,t} \to 0$ when $\mu^i_{\boldsymbol{\rho},f,t} \to 0$. Eliminating $p^{\mathsf{S}}_{f,t}$ and $p^{\mathsf{R}}_{f,t}$ in C8 based on (15), we have

$$\text{C8} \iff \sum_{\boldsymbol{\rho},f} \frac{\mu^i_{\boldsymbol{\rho},f,t}}{h^i_{\boldsymbol{\rho},f,t}} \left[\exp\left(\frac{b^i_{\boldsymbol{\rho},f,t} \ln 2}{\mu^i_{\boldsymbol{\rho},f,t} W}\right) - 1\right] \leq \frac{P_i}{W N_0},$$
$$i \in \{\mathsf{S}, \mathsf{R}\}, \forall t,$$

which is a convex set, cf. *Lemma 1.*

Lemma 1. The function $\theta(ax + b, y) = (ax + b) \exp\left(\frac{y}{ax+b}\right)$ is jointly convex in $(x, y) \in \{x \mid ax + b \geq 0\} \times \mathbb{R}_+$.

Proof: When $a = 1$, $b = 0$, $\theta(x, y)$ is a perspective function and thus jointly convex in $(x, y) \in \mathbb{R}_+^2$. Then, $\theta(ax + b, y)$ is the composition of $\theta(x, y)$ with an affine function, which is also jointly convex [36]. ∎

Moreover, based on (15), we can transform $B^{\mathsf{S}}_{\boldsymbol{\rho},t}$ and $B^{\mathsf{R}}_{\boldsymbol{\rho},t}$ into affine functions of $b^{\mathsf{S}}_{\boldsymbol{\rho},f,t}$ and $b^{\mathsf{R}}_{\boldsymbol{\rho},f,t}$, respectively,

$$B^i_{\boldsymbol{\rho},t} = \Delta \sum\nolimits_{f \in \mathcal{F}} \sum\nolimits_{\tau=1}^{t} b^i_{\boldsymbol{\rho},f,\tau}, \ i \in \{\mathsf{S}, \mathsf{R}\}, \forall t. \tag{16}$$

Accordingly, the cache/buffer management constraints C9, C10, C11, and C12 are convex. Based on these relaxation and transformation steps, (D1a) and (D2a) can be reformulated as equivalent convex problems for which strong duality holds. Thus, efficient polynomial-time algorithms, e.g. interior point methods [36], can be employed to obtain the optimal solution.

2) Solving Quasi-Convex Outer-Layer Subproblems: For the outer-layer subproblems (D1) and (D2), we reveal their quasi-convex structures in Proposition 1 and then adopt the bisection method to solve them [36]. Without loss of generality, let T be the delivery time, which corresponds to $T_{\mathrm{a},s}$ for (D1a) and $T_{\mathrm{b},s}$ for (D2a). For mathematical rigor, the domain of T is extended onto $\mathbb{R}_+ \cup \{\infty\}$ by defining $T = \lceil t_c \rceil$, where $t_c \in \mathbb{R}_+$.

Proposition 1. (Quasi-convexity of outer-layer subproblems) Problems (D1) and (D2) are quasi-convex programs in t_c.

Proof: It is easy to verify that $\beta_{\mathrm{a}}(\cdot)$ and $\beta_{\mathrm{b}}(\cdot)$ defined in (D1a) and (D2a) are non-decreasing and quasi-linear in t_c (and also in T). Then, $\mathcal{A}_{1,s}$ and $\mathcal{B}_{1,s}$ are convex sets (specifically rays) in t_c. Moreover, the objective functions of (D1) and (D2) are quasi-convex in t_c as they are non-increasing on $\mathbb{N}\backslash\mathcal{A}_{1,s}$ and $\mathbb{N}\backslash\mathcal{B}_{1,s}$ but non-decreasing on $\mathcal{A}_{1,s}\cap\mathbb{N}$ and $\mathcal{B}_{1,s}\cap\mathbb{N}$, respectively [36, Section 3.4.2]. Therefore, the outer-layer subproblems are quasi-convex. ■

Based on the above, both subproblems are solvable now. The overall solution procedure for (P1) and (P2) is summarized in Algorithm 1, which performs a doubling search from line 2 to line 7 to determine an upper bound on the delivery time (i.e., an initial search range $[l_0, u_0]$), and a bisection search from line 8 to line 17 to optimize the delivery time. During each iteration of the search, an inner problem (e.g. (D1a) and (D2a)) needs to be solved [36], [41]. For an initial step size $T_{\text{step}} = 1$ and delivery time bound $[l_0, u_0]$, both the doubling search and the bisection search terminate after $\ell \triangleq \lceil \log_2(u_0 - l_0) \rceil$ iterations. Appealingly, under mild conditions, Algorithm 1 converges to the global optimum, as stated in Theorem 3.

Theorem 3. (Global optimality condition) Algorithm 1 converges to the unique global optimal delivery time in a finite number of iterations, if $F \to \infty$ and $\mathbb{I}_{\mathcal{A}}(l_0) = \mathbb{I}_{\mathcal{B}}(l_0) = \infty$ (i.e., l_0 is small enough such that video delivery is not completed at $T = l_0$).

Proof: Please refer to the Appendix. ■

Upon obtaining the optimal delivery time, the corresponding caching or delivery decisions are also available. Thus, (P1) and (P2) are solved. The hidden convexity of the inner-layer subproblem guarantees that the obtained caching and delivery decisions are optimal for a given delivery time during each iteration of the bisection search and that, together with the quasi-convexity of the outer-layer subproblem, the overall solution is globally optimal upon reaching convergence.

Remark 4. Assume that the interior-point method, which has been implemented in various numerical solvers such as CVX [42], is applied for solving inner-layer problems (D1a) and (D1b) in line 4 and line 11 of Algorithm 1 [41]. Let T_S be the total delivery time for solving problem (P1) with S-session streaming, where $T_{S_1} \leq T_{S_2}$ generally holds for $S_1 \leq S_2$. To estimate the computational complexity of Algorithm 1, we assume that the average delivery time per session, $\frac{T_S}{S}$, satisfies $\frac{T_S}{S} \in [l_0, u_0]$ and $\frac{c_S T_S}{S} = u_0$, i.e., the initial search bounds for session $s \in \{1, \ldots, S\}$ are given by $[l_0, \frac{c_S T_S}{S}]$, where $c_S \geq 1$ is a parameter dependent on the value of S. In each step of the bisection or doubling search with $T \in [l_0, u_0]$, the computational complexity of solving problem (D1a) with the interior-point method is approximated in the big-O notation as $\mathcal{O}\left((KFT)^{3.5}\right)$ [41], since the size of the optimization problem scales with the number of UEs K, the number of SCs F, and the delivery time T. Moreover, the series of delivery times generated by the bisection (doubling) search iterations in the worst case is approximated by $\{T\} = \{u_0, u_0 - \frac{u_0 - l_0}{2^{\ell}}, u_0 - \frac{u_0 - l_0}{2^{\ell-1}}, \ldots, u_0 - \frac{u_0 - l_0}{2^0} = l_0\}$, where $\ell = \lceil \log_2(u_0 - l_0) \rceil$. Since at most $(\ell + 2)$ problems of sizes $KFT \leq KFu_0 = KFc_S T_S / S$ have to be solved during the bisection and doubling searches, respectively, the overall computational complexity of Algorithm 1 for solving problem (P1) can be approximated as

$$\begin{aligned} &\mathcal{O}\left(2S(KFu_0)^{3.5}(\ell+2)\right) \\ =&\mathcal{O}\left(2S^{-2.5}\left(c_S KFT_S\right)^{3.5}\log_2\left(\tfrac{c_S T_S}{S}\right)\right), \end{aligned} \tag{17}$$

where the factor "2" accounts for both the bisection and the doubling searches. We observe from (17) that multi-session streaming ($S > 1$) has a much lower computational complexity than single-session streaming ($S = 1$) as long as the difference in the respective values of $c_S T_S$ is small. Similarly, the computational complexity of Algorithm 1 for solving (P2) can be approximated as $\mathcal{O}\left(2S^{-3.5}\left(c_S KFT_S\Omega\right)^{3.5}\log_2\left(\tfrac{c_S T_S}{S}\right)\right)$ since the problem size of (P2) is Ω times that of (P1) and the S subproblems of (D2a) and (D2b) can be solved in parallel.

Algorithm 1 Search for the Optimal Offline Delivery Time

1: **initialization**: Given l; $T_{\text{step}} \leftarrow 1$, $T \leftarrow l$, tolerance $\epsilon \leftarrow 1$;
2: *%Phase 1: Doubling search for delivery time bound;*
3: **repeat**
4: Solve inner-layer problems (D1a) (or (D1b)) and (D2a) (or (D2b)) in $[0, T]$;
5: $l \leftarrow T$, $T \leftarrow T + T_{\text{step}}$, $T_{\text{step}} \leftarrow 2 * T_{\text{step}}$;
6: **until** $\mathbb{I}_{\mathcal{A}}(T) = 0$ or $\mathbb{I}_{\mathcal{B}}(T) = 0$.
7: $u \leftarrow T$;
8: *%Phase 2: Bisection search for optimal delivery time;*
9: **repeat**
10: $T \leftarrow \lceil (l + u)/2 \rceil$;
11: Solve inner-layer problems (D1a) (or (D1b)) and (D2a) (or (D2b)) in $[0, T]$;
12: **if** $\mathbb{I}_{\mathcal{A}}(T) = 0$ or $\mathbb{I}_{\mathcal{B}}(T) = 0$ **then**
13: $u \leftarrow T$;
14: **else**
15: $l \leftarrow T$;
16: **end if**
17: **until** $u - l < \epsilon$.

IV. Online Cross-Layer Delivery Control

In this section, we consider *online* fast delivery control schemes which require causal CSI only. The optimal online delivery scheme is first studied in Section IV-A for providing a performance benchmark. Due to the causality requirement, the offline optimization technique is not fully applicable. Instead, by decomposition and discretization techniques, we could cast the optimal online delivery control as a stochastic shortest path (SSP) problem and thereby solve it by a stochastic DP algorithm. However, due to its exponential computational complexity, it is difficult to apply the proposed DP algorithm in a large system. To resolve this problem, a suboptimal online delivery scheme having polynomial-time computational complexity is further proposed in Section IV-B. Throughout this section, the cache status is assumed to be pre-determined by the historical cache control in Section III-B, cf. (P2) and Algorithm 1. For a succinct presentation, single-session streaming is assumed below. The session index s is hence omitted. The delivery control problem associated with multi-session streaming can be addressed by just applying the developed model and algorithm to each session individually.

A. Optimal Online Fast Delivery

Assume a given user request scenario and statistical CSI. Before delivery starts, the optimal delivery policy is computed as a function of user requests and CSI values. During online VoD streaming, specific delivery control decisions are instantaneously obtained by evaluating the delivery policy according to the

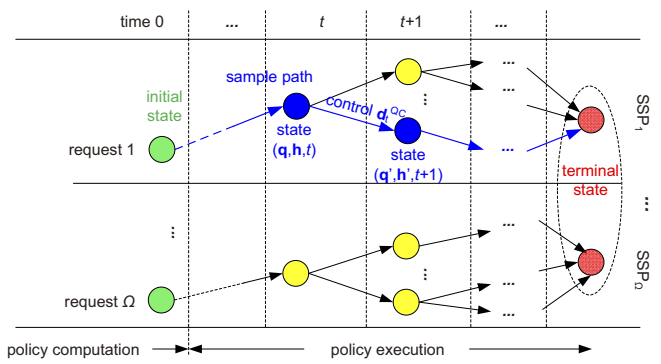


Fig. 2. State transition diagram for online delivery control.

current user requests and the instantaneous CSI realizations, cf. Fig. 2. However, finding the optimal online delivery policy involves a nontrivial optimization over a space of functions rather than that of variables. This new challenge is due to the CSI causality requirement and will be tackled below.

1) Problem Decomposition: The delivery process for serving user request $\boldsymbol{\rho}$ involves two dynamical FIFO queues to coordinate wireless fetching, cache fetching, and next-hop delivery (cf. Fig. 1(b)) subject to constraints C2–C14. We note that the queue management control, whose decisions are coupled over time due to constraints C9, C10, C11, C13, and C14, constitutes the major difficulty for online delivery optimization. Nevertheless, the underlying resource allocation decisions are easily solvable based on the techniques in Section III. Therefore, a decomposition as in Theorem 1 is worthwhile to separate the resource allocation subproblem from the online delivery problem.

Specifically, for a given queue control vector $\mathbf{d}_t^{\mathsf{QC}}$, there exists a feasible resource allocation (fulfilling constraints C2–C8 and C12) if and only if

$$\mathbf{d}_t^{\mathsf{QC}} \in \mathcal{D}^{\mathsf{QC}}(\mathbf{h}_t) \triangleq \{\mathbf{d}_t^{\mathsf{QC}} = \mathbf{g}\left(\mathbf{d}_t^{\mathsf{RA}}, \mathbf{h}_t\right) \mid \mathbf{d}_t^{\mathsf{RA}} \in \mathcal{D}^{\mathsf{RA}}(\mathbf{h}_t)\}, \quad (18)$$

or equivalently, the following resource allocation subproblem (RASP) is feasible,

$$\text{(P3)} \quad \underset{\mathbf{d}_t^{\mathsf{RA}} \in \mathcal{D}^{\mathsf{RA}}(\mathbf{h}_t)}{\text{maximize}} \quad 1 \quad (19)$$
$$\text{subject to} \quad \mathbf{g}\left(\mathbf{d}_t^{\mathsf{RA}}, \mathbf{h}_t\right) = \mathbf{d}_t^{\mathsf{QC}},$$

where we define $\mathcal{D}^{\mathsf{RA}}(\mathbf{h}_t) \triangleq \{\mathbf{d}_t^{\mathsf{RA}} \succeq \mathbf{0} \mid \text{C2–C8}, \text{C12}\}$ (C12 is rewritten as $\sum_{\boldsymbol{\rho}\in\mathcal{G}} \sum_{f\in\mathcal{F}} b^{\mathsf{S}}_{\boldsymbol{\rho},f,t} \le \gamma_t \Delta$). The online control problem is then decomposed into the RASP (19) in the inner layer, and a queue control subproblem (QCSP) in the outer layer; the latter determines the optimal queue control policy that minimizes the expected delivery time subject to (18) and the remaining (queuing) constraints C9, C10, C11, C13, and C14. The RASP given in problem (P3) is non-convex but can be efficiently solved by an equivalent convex problem as in Section III. The formulation and solution of the QCSP is addressed next.

2) Stochastic Shortest Path (SSP) Formulation of QCSP: We consider discrete-valued CSI $\mathbf{h}_t \in \hat{\mathcal{H}}$ and QSI $B^i_{\boldsymbol{\rho},t} \in \hat{\mathcal{V}}_i \triangleq \{v_0 = 0, v_1, v_2, \ldots, v_{L_i - 1} = \alpha_i V_{\boldsymbol{\rho}}\}$ ($i \in \{\mathsf{S}, \mathsf{R}, \mathsf{C}\}$) for ease of derivation, where $v_0 < v_1 < \ldots < v_{L_i - 1}$, $\alpha_{\mathsf{S}} = 1 - c_n^{(m)}$, $\alpha_{\mathsf{R}} = 1$, $\alpha_{\mathsf{C}} = c_n^{(m)}$, and L_i is the discretization level. The queue control vectors are also discretized based on the discrete-valued QSI and we have $\mathbf{d}_t^{\mathsf{QC}} \in \hat{\mathcal{D}}^{\mathsf{QC}}(\mathbf{h}_t)$, where

$$\hat{\mathcal{D}}^{\mathsf{QC}}(\mathbf{h}_t) \triangleq \Big\{\mathbf{d}_t^{\mathsf{QC}} \in \mathcal{D}^{\mathsf{QC}}(\mathbf{h}_t) \mid b^i_{\boldsymbol{\rho},t} = v_{j+k} - v_j, \; j,k \in \mathbb{N},\; j+k \le L_i,\; i \in \{\mathsf{S},\mathsf{R},\mathsf{C}\}\Big\}. \quad (20)$$

A sufficient condition is provided in the sequel to guarantee that the online problem remains feasible after discretization, i.e., the discretization in (20) is *proper* [25, Chapter 6].

As a starting point, the evolution of the instantaneous QSI $\mathbf{q}_t$ along an arbitrary sample path, i.e., a time series of the CSI values and the queue control vectors as shown in Fig. 2, is considered. Given the instantaneous CSI, $\mathbf{h}_t$, and queue control vector, $\mathbf{d}_t^{\mathsf{QC}} \in \hat{\mathcal{D}}^{\mathsf{QC}}(\mathbf{h}_t)$, $\mathbf{q}_t$ evolves according to C9 and C10, i.e.,

$$B^i_{\boldsymbol{\rho},t+1} = \min\left\{B^i_{\boldsymbol{\rho},t} + b^i_{\boldsymbol{\rho},t}, \alpha_i V_{\boldsymbol{\rho}}\right\},\; \boldsymbol{\rho} \in \mathcal{G}, i \in \{\mathsf{S},\mathsf{C}\}, \quad (21)$$
$$B^{\mathsf{R}}_{\boldsymbol{\rho},t+1} = \min\left\{B^{\mathsf{R}}_{\boldsymbol{\rho},t} + b^{\mathsf{R}}_{\boldsymbol{\rho},t}, B^{\mathsf{S}}_{\boldsymbol{\rho},t} + B^{\mathsf{C}}_{\boldsymbol{\rho},t}, \alpha_{\mathsf{R}} V_{\boldsymbol{\rho}}\right\},\; \boldsymbol{\rho} \in \mathcal{G},$$

where $\mathbf{q}_t$ has to further fulfill the buffer capacity constraint C11 and the QoS constraint C13. The queues defined in (21) are *non-decreasing*, i.e., $\mathbf{q}_{t+1} - \mathbf{q}_t \succeq \mathbf{0}$, and *non-stationary*, e.g., the sets of feasible queue states and queue control vectors are time-varying due to C13. As a result, the time evolution accompanying the queue evolution should be considered explicitly. For this purpose, we define the system time index (STI) $t \in \mathcal{T}$ as the number of time slots elapsed since delivery started. The STI increases by 1, i.e., from t to $t+1$, for each transition of the QSI or the CSI. We have $\mathcal{T} \subset \mathbb{N}$ and $|\mathcal{T}| < +\infty$ because the file sizes are finite.

To capture both the queue and the time evolution, the system state is defined as $\boldsymbol{\phi}_t \triangleq [\mathbf{q}, \mathbf{h}, t]$ (or alternatively $\boldsymbol{\phi}_t \triangleq [\mathbf{q}_t, \mathbf{h}_t]$). Using the notation of $\boldsymbol{\phi}_t$, we denote (21) as $\boldsymbol{\phi}_{t+1} = \mathbf{f}(\boldsymbol{\phi}_t, \mathbf{d}_t^{\mathsf{QC}})$. The state space, i.e., the set of feasible system states, is given by $\mathcal{S} \triangleq \{\boldsymbol{\phi}_t \succeq \mathbf{0} \mid \text{C11}, \text{C13}\}$. The values of $\boldsymbol{\phi}_t$ are observed and provided as side information for online delivery control. The control space, i.e., the set of $\mathbf{d}_t^{\mathsf{QC}}$ satisfying constraints C2–C13, is defined for each state $\boldsymbol{\phi}_t$,

$$\mathcal{Q}(\boldsymbol{\phi}_t) = \left\{\mathbf{d}_t^{\mathsf{QC}} \in \hat{\mathcal{D}}^{\mathsf{QC}}(\mathbf{h}_t) \mid \mathbf{f}(\boldsymbol{\phi}_t, \mathbf{d}_t^{\mathsf{QC}}) \in \mathcal{S}\right\},\; \boldsymbol{\phi}_t \in \mathcal{S}. \quad (22)$$

Note that the null action $\mathbf{0} \in \mathcal{Q}(\boldsymbol{\phi}_t)$ holds trivially for any $\boldsymbol{\phi}_t$. The state transition terminates, when the delivery is completed, upon reaching the "terminal state" defined as $\boldsymbol{\varphi} \triangleq \{\boldsymbol{\phi}_t \in \mathcal{S} \mid \min_{\boldsymbol{\rho}} B^{\mathsf{R}}_{\boldsymbol{\rho}}/V_n = 1\}$. Here, $\boldsymbol{\varphi}$ is an aggregation of the states satisfying C14. Thus, the state and control spaces are defined with all relevant constraints taken into account. Note that we have assumed fixed CSI in the derivations so far.

Next, the statistical transitions of system states under online delivery control and statistical CSI are studied. Assume that the channel process $\{\mathbf{h}_t\}$ is independent over time and that the probability distribution at each time is given. Then, for given $\mathbf{d}_t^{\mathsf{QC}} \in \mathcal{Q}(\boldsymbol{\phi}_t)$, the state transitions are *Markovian*, i.e., the transition is independent of all the previous states except for the current state $\boldsymbol{\phi}_t$. Consequently, the probability of transition from $\boldsymbol{\phi}_t$ to its next state $\boldsymbol{\phi}_{t+1}$ is given by

$$\mathbf{P}\left\{\boldsymbol{\phi}_{t+1} \mid \boldsymbol{\phi}_t, \mathbf{d}_t^{\mathsf{QC}}\right\} = \mathbf{P}\left\{\mathbf{h}_{t+1}\right\} \cdot \mathbf{1}\left\{\boldsymbol{\phi}_{t+1} = \mathbf{f}(\boldsymbol{\phi}_t, \mathbf{d}_t^{\mathsf{QC}})\right\}, \quad (23)$$

where $\mathbf{d}_t^{\mathsf{QC}} \in \hat{\mathcal{D}}^{\mathsf{QC}}(\boldsymbol{\phi}_t)$ and we have $\mathbf{1}\{X\} = 1$ if condition X holds true and $\mathbf{1}\{X\} = 0$ otherwise.

Finally, let us assign an immediate cost of $c(\boldsymbol{\phi}_t) = 1$ to state $\boldsymbol{\phi}_t$ if $\boldsymbol{\phi}_t$ is non-terminating, and $c(\boldsymbol{\phi}_t) = 0$ otherwise, i.e., $c(\boldsymbol{\phi}_t) \triangleq \mathbf{1}\{\boldsymbol{\phi}_t \in \mathcal{S} \backslash \boldsymbol{\varphi}\}$. Denote the initial system state by $\boldsymbol{\phi}_0 \triangleq [\mathbf{0}, \cdot, 0]$. The QCSP can then be formulated as seeking the

optimal control policy $\boldsymbol{\pi}^*$ that achieves the minimal expected cost, or equivalently the minimal expected delivery time T^*, starting from $\boldsymbol{\phi}_0$. We have

$$\text{(P4)} \quad T^*(\boldsymbol{\phi}_0) = \min_{\boldsymbol{\pi}} \mathbf{E}\left[\sum\nolimits_{t=1}^{\infty} c(\boldsymbol{\phi}_t)\right], \tag{24}$$

and $\boldsymbol{\pi}^* = \arg\min_{\boldsymbol{\pi}} \mathbf{E}\left[\sum_{t=1}^{\infty} c(\boldsymbol{\phi}_t)\right]$, where the expectation is taken with respect to the probability distribution of the state transitions. Due to the Markovian property, delivery policies of the form $\boldsymbol{\pi} \triangleq \left[\mathbf{d}_t^{\mathsf{QC}}(\boldsymbol{\phi}_t)\right]_{\boldsymbol{\phi}_t \in \mathcal{S}}$, where $\mathbf{d}_t^{\mathsf{QC}}(\cdot) : \mathcal{S} \to \mathcal{Q}$ are deterministic functions mapping $\boldsymbol{\phi}_t \in \mathcal{S}$ to $\mathbf{d}_t \in \mathcal{Q}$, can be considered without loss of optimality [25].

However, if $\hat{\mathcal{D}}^{\mathsf{QC}}$ and $\hat{\mathcal{V}}_i$ are not properly discretized, a control policy that leads to delivery completion in a finite delivery time irrespective of the channel process may not exist. That is, (P4) becomes infeasible. To resolve this problem, a sufficient condition is stated in Proposition 2 to guarantee proper discretization.

Proposition 2. (Sufficient condition for proper discretization) The discretization is proper if, for each QSI $\mathbf{q}_t$, there exists some CSI $\mathbf{h}_t^0 \in \hat{\mathcal{H}}$, such that $\mathcal{Q}\left([\mathbf{q}, \mathbf{h}^0, t]\right) \backslash \{\mathbf{0}\}$ is non-empty, i.e., $\bigcup_{\mathbf{h}} \left(\mathcal{Q}([\mathbf{q}, \mathbf{h}, t]) \backslash \{\mathbf{0}\}\right) \neq \emptyset$ holds for all $\boldsymbol{\phi}_t = [\mathbf{q}, \mathbf{h}, t]$.

The properness is obvious since, under the condition of Proposition 2, we can always traverse from QSI $\mathbf{q}$ to some QSI $\mathbf{q}'$, where $\mathbf{q}' \succeq \mathbf{q}$ and $\mathbf{q}' \neq \mathbf{q}$, in a finite number of state transitions using a non-zero control policy. On the contrary, if there exist some QSI $\mathbf{q}_t$ such that $\mathcal{Q}\left([\mathbf{q}, \mathbf{h}, t]\right) \backslash \{\mathbf{0}\} = \emptyset$ holds for all $\mathbf{h}_t \in \hat{\mathcal{H}}$, the state transition may possibly end in an infinite-loop within the subset of states $\{[\mathbf{q}, \mathbf{h}, t)] \mid \mathbf{h}_t \in \hat{\mathcal{H}}\}$. Based on these discussions, by starting from any non-terminating state $\boldsymbol{\phi} \in \mathcal{S} \backslash \boldsymbol{\varphi}$, the terminal state $\boldsymbol{\varphi}$ can be *achieved* after some finite number of transitions; moreover, it remains in the terminal state thereafter with probability 1, i.e., $\mathbf{P}\left\{\boldsymbol{\varphi} \mid \boldsymbol{\varphi}, \mathbf{d}_t^{\mathsf{QC}}\right\} \equiv 1$. This ensures that (P4) is a well-defined SSP problem if properly discretized [25, Chapter 7], [43, Chapter 2].

3) Bellman Optimality Equation and Solution: The SSP formulation in (P4) allows us to compute the optimal online delivery policy via DP algorithms. Let $T^*(\boldsymbol{\phi}_t)$ be the optimal cost-to-go functional at state $\boldsymbol{\phi}_t \in \mathcal{S}$, which determines the minimal expected delivery time or the minimum expected number of steps for transition from state $\boldsymbol{\phi}_t$ to terminal state $\boldsymbol{\varphi}$. The DP equation is given as

$$T^*(\boldsymbol{\phi}_t) = \min_{\mathbf{d}_t^{\mathsf{QC}} \in \mathcal{Q}(\boldsymbol{\phi}_t)} c_t(\boldsymbol{\phi}_t) + \sum_{\boldsymbol{\phi}_{t+1} \in \mathcal{S}} \mathbf{P}\left\{\boldsymbol{\phi}_{t+1} \mid \boldsymbol{\phi}_t, \mathbf{d}_t^{\mathsf{QC}}\right\} T^*(\boldsymbol{\phi}_{t+1}), \quad \boldsymbol{\phi}_t \in \mathcal{S}. \tag{25}$$

If $\boldsymbol{\phi}_t = \boldsymbol{\varphi}$, we have $T^*(\boldsymbol{\varphi}) = \mathbf{P}\left\{\boldsymbol{\varphi} \mid \boldsymbol{\varphi}, \mathbf{d}_t^{\mathsf{QC}}\right\} T^*(\boldsymbol{\varphi}) = 0$. Otherwise, vector $[T^*(\boldsymbol{\phi}_t)]_{\boldsymbol{\phi}_t \in \mathcal{S} \backslash \boldsymbol{\varphi}}$, appearing on both sides of (25), defines the fixed point of (25), which is unique and can be obtained by the successive approximation (value iteration) method [43, Chapter 2]. In particular, by solving the problem in (26) at iteration step k,

$$T_{k+1}(\boldsymbol{\phi}_t) = \min_{\mathbf{d}_t^{\mathsf{QC}} \in \mathcal{Q}(\boldsymbol{\phi}_t)} c_t(\boldsymbol{\phi}_t) + \sum_{\boldsymbol{\phi}_{t+1} \in \mathcal{S}} \mathbf{P}\left\{\boldsymbol{\phi}_{t+1} \mid \boldsymbol{\phi}_t, \mathbf{d}_t^{\mathsf{QC}}\right\} T_k(\boldsymbol{\phi}_{t+1}), \quad \boldsymbol{\phi}_t \in \mathcal{S} \backslash \boldsymbol{\varphi}, \tag{26}$$

the successive approximation can approach the (unique) fixed point of the DP equation (and thus the online optimum) in the limit as $k \to \infty$, i.e., $T^*(\boldsymbol{\phi}_t) = \lim_{k \to \infty} T_k(\boldsymbol{\phi}_t)$. Upon obtaining $T^*(\boldsymbol{\phi}_t)$, the optimal online delivery policy is obtained by the deterministic policy $\boldsymbol{\pi}^* = [\mathbf{d}_t^{\mathsf{QC}*}(\boldsymbol{\phi}_t)]_{\boldsymbol{\phi}_t \in \mathcal{S} \backslash \boldsymbol{\varphi}}$ [43, Chapter 2], where

$$\mathbf{d}_t^{\mathsf{QC}*}(\boldsymbol{\phi}_t) = \arg\min_{\mathbf{d}_t^{\mathsf{QC}} \in \mathcal{Q}(\boldsymbol{\phi}_t)} \left[\sum_{\boldsymbol{\phi}_{t+1} \in \mathcal{S}} \mathbf{P}\left\{\boldsymbol{\phi}_{t+1} \mid \boldsymbol{\phi}_t, \mathbf{d}_t^{\mathsf{QC}}\right\} T^*(\boldsymbol{\phi}_{t+1})\right], \quad \boldsymbol{\phi}_t \in \mathcal{S} \backslash \boldsymbol{\varphi}. \tag{27}$$

Algorithm 2 Computation of the Optimal Online Delivery Policy

1: *%Phase 1: Initialization of control space, state space, and state transition tables*;
2: **given** Discretization level L_i, cache status $\mathbf{c}^{(m)}$, request scenario $\boldsymbol{\omega} \in \Omega$;
3: **repeat**
4: Discretize state space $\mathcal{S}$ and control set $\hat{\mathcal{D}}^{\mathsf{QC}}$ according to (20);
5: Check feasibility of each control vector $\mathbf{d}_t^{\mathsf{QC}} \in \hat{\mathcal{D}}_1$ and determine control space $\mathcal{Q}(\boldsymbol{\phi}_t)$ (cf. (22));
6: Increase discretization level $L_i, i \in \{\mathsf{S}, \mathsf{R}, \mathsf{C}\}$;
7: **until** $\bigcup_{\mathbf{h}} \left(\mathcal{Q}(\mathbf{q}, \mathbf{h}, t) \backslash \{\mathbf{0}\}\right) \neq \emptyset$ for all $\boldsymbol{\phi}_t = (\mathbf{q}, \mathbf{h}, t)$ (cf. Proposition 2);
8: Build state transition tables for all $\mathbf{d}_t^{\mathsf{QC}}$ based on (23);
9: *%Phase 2: Successive approximation for obtaining optimal online policy;*
10: **given** tolerance $\epsilon > 0$ and initial value functionals $T_0(\boldsymbol{\phi}_t)$, $k \leftarrow 0$;
11: **repeat**
12: Update $T_{k+1}(\boldsymbol{\phi}_t)$ based on (26);
13: Update iteration index $k \leftarrow k + 1$;
14: **until** $\max_{\boldsymbol{\phi}_t} |T_{k+1}(\boldsymbol{\phi}_t) - T_k(\boldsymbol{\phi}_t)| < \epsilon$;
15: Obtain delivery policy based on (27).

The computation steps for obtaining the optimal online delivery policy are summarized in Algorithm 2. From line 1 to line 7, discretization of the state and the control spaces is repeatedly performed using an increasingly finer granularity until the properness condition in Proposition 2 is satisfied. Then, a state transition table associated with the discrete state space is built for each discrete delivery control in line 8. Applying the obtained transition tables and the discrete control space in the successive approximation procedure between line 9 and line 15, the optimal policy is finally obtained. Algorithm 2 can run offline and bears a computational complexity of $\mathcal{O}\left((L_{\mathsf{S}} L_{\mathsf{R}} L_{\mathsf{C}})^K\right)$, which depends on the number of discretization levels L_i, $i \in \{\mathsf{S}, \mathsf{R}, \mathsf{C}\}$, and the number of users K. The obtained policies are stored in the controller as a look-up table, from which online delivery decisions for different system states can be retrieved instantaneously.

The problem formulations (P3), (P4) and solution techniques in Sections IV-A are valid for arbitrary user request scenarios (as well as for all downloading sessions if multi-session streaming is adopted). The delivery control policies for all request scenarios can be obtained independent of each other in parallel, as illustrated in Fig. 2. Note that the optimal online optimization for S-session streaming control incurs an overall computational complexity of $\mathcal{O}\left(S(L_{\mathsf{S}} L_{\mathsf{R}} L_{\mathsf{C}})^K\right)$, which scales linearly with S. Due to the parallel implementation, however, the overall computational time does not increase with Ω.

B. Suboptimal Scheme

The non-polynomial time computational complexity of the optimal DP approach prohibits its application in large systems with tens or hundreds of VoD streaming users. Effective suboptimal schemes are needed instead in these cases to leverage a better trade-off between performance and computational complexity. For example, at the cost of performance losses, the SSP optimization problem can be alternatively solved by approximate DP

algorithms with low complexity [43, Section 2], [44, Sections 4 & 6].

Alternatively, exploiting the structure of the offline optimization scheme, we propose here a greedy suboptimal online scheme for the considered joint caching and buffering system. Our approach is inspired by [23], where similar greedy heuristics have been shown to be optimal for simple BaR systems with sufficiently large buffer capacity. This optimality can be explained by the fact that for large buffer capacities, the time-coupled queuing constraints become inactive most of the time (i.e., only active in a vanishing fraction of the delivery time) and thus have a limited effect on the system performance.

In particular, rather than solving the non-causal inner-layer problem given in Theorem 1, we propose to optimize the effective instantaneous throughput based on the instantaneous CSI only. To this end, the following problem is defined at each time instance,

$$\text{(P5)} \quad \underset{\mathbf{d}_t \in \mathcal{D}}{\text{maximize}} \quad \sum\nolimits_{\boldsymbol{\rho} \in \mathcal{G}} \left(B^{\mathsf{R}}_{\boldsymbol{\rho},t} - B^{\mathsf{R}}_{\boldsymbol{\rho},t-1} \right) \qquad (28)$$
$$\text{subject to} \quad \overline{\text{C14}}\text{: } B^{\mathsf{R}}_{\boldsymbol{\rho},t} \leq V_{\boldsymbol{\rho}}, \ \forall \boldsymbol{\rho} \in \mathcal{G},$$

which is equivalent to maximizing the instantaneous total queue length $\sum_{\boldsymbol{\rho}} B^{\mathsf{R}}_{\boldsymbol{\rho},t}$ over the instantaneous delivery control vector $\mathbf{d}_t$. Problem (P5) is non-convex but can be transformed into an equivalent convex problem based on (15) in Section III-C. At time t, the suboptimal algorithm solves problem (P5) and then updates the queue status using the obtained optimal solutions. This process continues as time increases until constraint $\overline{\text{C14}}$ becomes active for all requests, i.e., the delivery is completed. As a result, the instantaneous total queue length is maximized in a greedy manner.

For given delivery time T, the proposed suboptimal algorithm has a polynomial time computational complexity of $\Theta^{\text{subopt}} = \mathcal{O}\left(T(KF)^{3.5}\right)$, where the optimization problem solved per iteration has a size that scales with K and F. In addition, the proposed suboptimal algorithm requires only instantaneous CSI (and QSI) and is thus appealing for real-time implementation. Note that Θ^{subopt} scales with T linearly. This implies that the proposed suboptimal scheme has even a much lower complexity than the polynomial-time optimal offline scheme, cf. (17).

The suboptimal algorithm causes a performance degradation compared to optimal DP as the time coupling in the queue evolution is ignored. However, we find in Section V that the proposed greedy suboptimal scheme is close-to-optimal in the high buffer capacity regime.

V. Performance Evaluation

In this section, the performance of the proposed schemes is evaluated for residential small cells where most of the VoD streaming traffic is expected to occur [2]. Consider $M = 3$ RNs equally distributed in a cell of radius 750 m. Each RN is located at a distance of 500 m from the BS and provides coverage in a radius of 250 m, e.g. to accommodate the VoD traffic for a few households. We assume there are $N = 5$ video files, each of size 500 MB (Bytes), that have to be delivered to $K = 3$ users. The UEs are uniformly and randomly distributed in the cell while the minimum distance between UE and BS/RN is 50 m. Each user requests one file independently. Let θ_n be the probability of file $n \in \mathcal{N}$ being requested and $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_N]$ be the probability distribution of the requests for the different files. We set $\boldsymbol{\theta}$ = [0.57, 0.20, 0.11, 0.07, 0.05], which follows the Zipf distribution [9]. Moreover, the 3GPP path loss model ("Macro + Outdoor Relay, NLOS scenario") in [45] is adopted. The small-scale fading coefficients are independent and identically distributed (i.i.d.) Rayleigh random variables. The other relevant system parameters are given in Table I. Before video delivery starts, Ω = 50 scenarios are randomly generated based on the user preference distribution and the channel model to optimize the initial cache status, cf. (P2).

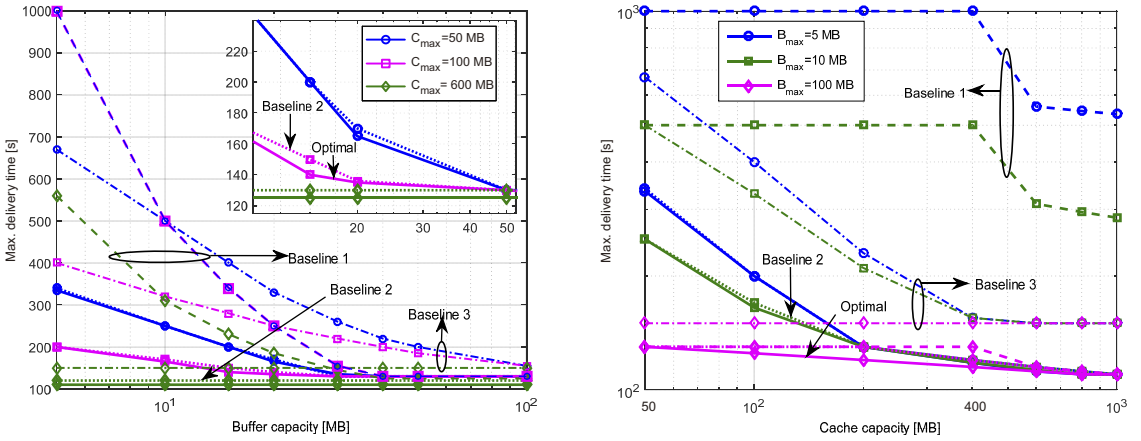
TABLE I
Simulation parameters.

Parameters	Settings
System bandwidth	20 MHz
Subcarriers	F = 64
Bandwidth of a SC	W = 313 kHz
Duration of time slot	Δ = 20 ms
Max. transmit power	$P_{\mathsf{S}} = 46$ dBm, $P_{\mathsf{R}} = 40$ dBm
Noise power density	$N_0 = -172.6$ dBm/Hz
Backhaul capacity	γ_t = 1 Gbps, $\forall t$
UE rate requirement	$\nu_{\min}$ = 1 kbps
Initial delay	$\epsilon_{\boldsymbol{\rho}} = 0$

First, the offline scheme (cf. Algorithm 1) is considered assuming perfect knowledge of the user requests and the CSI. Unless specified otherwise, we consider single-session streaming, i.e., $S = 1$. For comparison, we consider two heuristic caching policies and one suboptimal delivery scheme as baselines:

- *Baseline 1 (Preference-based Caching):* In this case, the most popular files are cached. Assuming $\boldsymbol{\theta}$ is known, the cache control decision is made based on
$$\max_{\mathbf{c}^{(m)} \in \mathcal{C}^{(m)}} \sum\nolimits_{m,n} \theta_n c_n^{(m)} V_n. \qquad (29)$$
- *Baseline 2 (Uniform Caching):* In this case, the same amount of data is cached for each file, i.e., $c_n^{(m)} V_n = \frac{1}{N} \times \min\{C_{\max}^{(m)}, \sum_{n=1}^{N} V_n\}, \forall m, n$, and the user's preference is not taken into account. For both Baselines 1 and 2, the optimal delivery scheme in (P1) is adopted.
- *Baseline 3 (Joint SC Assignment and Power Allocation with Fixed Link Schedule)*: This scheme is basically the same as the one obtained from (P1) except that a fixed link schedule as in the conventional half-duplex relaying protocol [24] is assumed, i.e., $\mu^{\mathsf{S}}_{\boldsymbol{\rho},f,2t} = \mu^{\mathsf{R}}_{\boldsymbol{\rho},f,2t-1} = 0, \forall t$, holds. Hence, the benefits of BaR cannot be exploited, but joint SC assignment and power allocation is performed for minimizing the delivery time. For Baseline 3, the same initial cache status as for the optimal delivery scheme is adopted. We note that Baseline 3 focuses on the delivery design in the physical layer only and does not require the exchange of QSI. The computational complexity of Baseline 3 is $\mathcal{O}\left(T_S(KF)^{3.5}\right)$ for S-session streaming.

In Figs. 3(a) and 3(b), the maximum delivery time of all considered offline schemes is evaluated for different values of the buffer and cache capacities, respectively. For a small cache capacity, we observe from Fig. 3(a) that the performance of the optimal scheme can be significantly improved by increasing the buffer capacity, which increases the joint scheduling opportunities of (wireless) fetching and delivery for uncached data in the two-hop relaying system. The buffering gains saturate at large buffer capacities when the maximal benefits are achieved. As the cache capacity increases, smaller buffer capacities are



(a) (b)

Fig. 3. Maximum delivery time versus (a) buffer capacity and (b) cache capacity for the proposed scheme (solid line), Baseline 1 (dashed line), Baseline 2 (dotted line), and Baseline 3 (dash-dotted line).

sufficient to achieve the maximal buffering gains. This is because the amount of uncached data decreases, which reduces the joint control opportunities for wireless fetching and delivery of uncached data.

The caching gains are further investigated in Fig. 3(b). For a small buffer capacity, the performance of the optimal scheme improves significantly by increasing the cache capacity. This is expected since, on the one hand, the cache facilitates the macroscopic gains of content reuse and reduced delivery distance for the delivery of cached data. Unlike the buffering gains, the macroscopic caching gains do not diminish for large buffer capacities, i.e., they cannot be compensated by buffering gains. On the other hand, similar to the buffer, the cache improves the microscopic diversity gains for uncached data because of joint (wireless and cache) fetching and delivery control. Therefore, some trade-off between the buffering and the (microscopic) caching gains can be observed in Figs. 3(a) and 3(b).

From Figs. 3(a) and 3(b) we observe that the optimal scheme achieves a significant performance gain in terms of the maximum delivery time compared to Baseline 3, which underlines the benefits of cross-layer delivery optimization compared to optimizing each layer separately. Indeed, the optimal scheme has a $\left(\frac{T_S}{S}\right)^{2.5}\log_2\left(\frac{T_S}{S}\right)$ times higher computational complexity than Baseline 3, cf. (17), and requires the exchange of QSI; however, as will be shown in Fig. 6(b), by increasing the number of sessions S, the computational cost of the optimal scheme can be significantly reduced without degrading the performance noticeably. We note that the results in Figs. 3(a) and 3(b) seem to contradict the common belief that BaR achieves a throughput gain at the expense of an increased transmission delay [24]. However, considering the dual relation between throughput and delivery time, cf. Theorem 1, it is not surprising that an improved throughput also benefits the delivery time.

Comparing the optimal scheme with Baselines 1 and 2, our results suggest that preference-based caching is least efficient in utilizing the cache capacity for overall delivery enhancement, particularly when the cache and buffer capacities are small. The reason is that the uncached files constitute the performance bottleneck. For example, considering (29), the less popular files would not be cached unless $C_{\max} > V_1$, where V_1 is the size of the most popular file, i.e., file 1 for the given probability

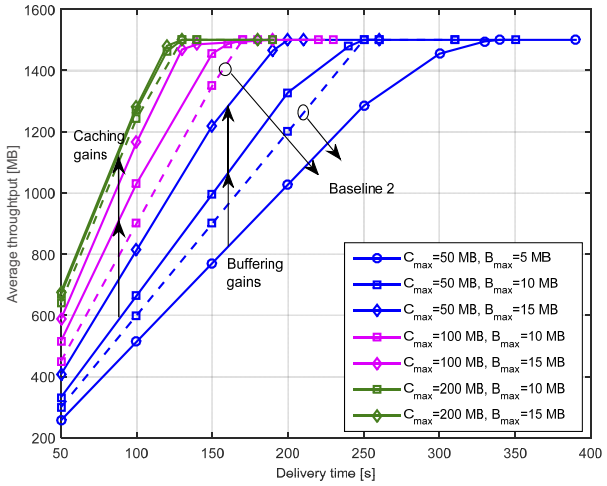


Fig. 4. Average amount of delivered video data versus delivery time for the proposed scheme (solid line) and Baseline 2 (dashed line). Results for Baseline 2 are only shown for $B_{\max}$ = 10 MB.

distribution $\boldsymbol{\theta}$. This explains why the performance of Baseline 1 does not improve when $C_{\max}$ increases from 50 MB to 400 MB in Fig. 3(b). On the other hand, uniform caching combined with the optimal delivery scheme in (P1), i.e., Baseline 2, performs very close to the optimal scheme. For a detailed analysis of this phenomenon, the average amount of delivered video data with respect to the delivery time is illustrated in Fig. 4. We observe that the delivered data of Baseline 2 increases steadily with the delivery time because the cache status is independent of the user requests and the delivery process. With both the users' requests and the delivery process considered in the caching decisions, the optimal scheme shows a seemingly faster delivery progress than Baseline 2. However, in an i.i.d. channel fading environment, the optimal scheme only improves the delivery completion time slightly compared to Baseline 2.

Next, the optimal (cf. Algorithm 2) and suboptimal (cf. (P5)) online delivery schemes are evaluated assuming availability of causal CSI only. Due to the limited scalability of DP, the simulation is performed in a single-RN single-UE subsystem of the relaying network, i.e., $M = 1$ and $K = 1$. The channel on each SC follows an independent two-point distribution with $\mathbf{P}\{h = 0.6\} = 0.4$ and $\mathbf{P}\{h = 1.0\} = 0.6$. We

discretize the queue states and the delivery control vectors in (20) uniformly using an initial step size $\delta = v_{k+1} - v_k = 0.1 \times \min\{B_{\max}^{(m)}, C_{\max}^{(m)}\}, \forall k \in \mathbb{N}$.

Figs. 5(a) and 5(b) show the maximal delivery time of the considered online delivery schemes for different values of the buffer and cache capacities, respectively. Figs. 5(a) and 5(b) show that the optimal online and offline schemes can effectively exploit the cache and the buffer capacity for reducing the maximal delivery time of the users in the network. This behavior is similar to the effects in Figs. 3(a) and 3(b). However, compared with the optimal offline scheme, the optimal online scheme suffers from a performance loss due to the lack of non-causal CSI. It is interesting to note that the performance gap between the optimal online and the offline schemes remains roughly constant as the cache capacity increases, cf. Fig. 5(b). In contrast, the gap quickly diminishes as the buffer capacity increases and becomes negligible in the high buffer capacity regime, where the effect of the queuing constraints on the system performance becomes negligible, cf. Fig. 5(a). This result facilitates the design of efficient online delivery schemes with low complexities and even with no need for statistical CSI knowledge, as shown below for the suboptimal online scheme. Recall that the suboptimal online scheme has a much lower computational complexity compared to the optimal online scheme. However, because of the ignorance of the time coupling in the queue evolution, the suboptimal online scheme suffers from a performance degradation as shown in Fig. 5(b). The performance loss decreases slowly as the cache capacity increases. Thus, compared to the optimal online scheme, a larger cache capacity is needed for the suboptimal online scheme to achieve a certain performance. Nevertheless, the performance loss of the proposed suboptimal online scheme becomes negligible in the high buffer capacity regime, cf. Fig. 5(a).

Finally, we extend our consideration to larger systems and discuss some alternatives for achieving scalable delivery control. In Fig. 6(a), the proposed delivery schemes are evaluated for different numbers of users in the system with $M = 3$ RNs. The optimal online scheme is not considered due to its high computational complexity. From Fig. 6(a), we observe that the performance gap between the suboptimal online and the optimal offline schemes increases with the number of users. However, for the considered number of users, the performance gap remains small when the buffer capacity per user is large. This result suggests that scaling the buffer capacity at the RNs with the number of users is necessary for the suboptimal online scheme to achieve a close-to-optimal performance.

In Fig. 6(b), multi-session streaming ($S > 1$) is compared with single-session streaming ($S = 1$), where the requested file size for each session is chosen according to (10). The suboptimal online scheme is not evaluated herein since both its performance and computational complexity are independent of the value of S. As can be observed from Fig. 6(b), for $S \leq 200$ and large cache or buffer capacities, the performance loss incurred by multi-session streaming is negligible. On the other hand, (17) reveals that the computational complexity can be significantly reduced by increasing S. For example, when S is increased from 1 to 200, the average computation time per simulation scenario is dramatically reduced from 20.5 minutes to 0.3 seconds, if CVX is run on a desktop computer with an Intel(R) Core(TM) Quad 3.40-GHz CPU and 16 GB of memory.

VI. Conclusions

In this paper, cross-layer caching and delivery control was investigated for minimizing the overall video delivery time in a downlink network, where a BS sends video data to multiple users via buffer- and cache-enabled relay nodes. A two-stage offline optimization problem was formulated for given user requests and full CSI knowledge, and turned out to be functional and non-convex. Based on the proposed decomposition and transformation techniques, a novel efficient offline algorithm was developed to solve the problem. Moreover, online delivery optimization under statistical CSI knowledge was investigated based on the DP framework and optimal and suboptimal online schemes were proposed. Simulation results revealed that joint caching and buffering can effectively reduce the overall delivery time by exploiting the channel diversity of the fetching and delivery links. Besides, our results unveiled a trade-off between the caching gain and the buffering gain in both the online and the offline settings and suggested the existence of low-complexity close-to-optimum online delivery schemes in the high buffer capacity regime.

Appendix
Proof of Theorem 3

The proof involves two steps. In the first step, we show that in the limiting case of $F \to \infty$, the binary relaxation in the inner-layer subproblems becomes tight. Note that the binary constraint C4: $\mu_{\boldsymbol{\rho},f,t} \in \{0,1\}$ is equivalent to $\mu_{\boldsymbol{\rho},f,t} \in [0,1]$ and $\xi(\mathbf{x}) \geq 0$ with $\xi(\mathbf{x}) \triangleq \sum_{\boldsymbol{\rho},f,t}(\mu_{\boldsymbol{\rho},f,t}^2 - \mu_{\boldsymbol{\rho},f,t})$. Without loss of generality, the inner-layer subproblems (D1a), (D1b), (D2a), and (D2b) can be written in general form as,

$$\beta^* \triangleq \underset{\mathbf{x}\in\mathcal{X}}{\text{minimize}} \quad \beta(\mathbf{x}) \qquad \text{subject to} \quad \xi(\mathbf{x}) \geq 0, \tag{30}$$

where $\mathbf{x}$, $\beta(\mathbf{x})$, and $\mathcal{X}$ denote the optimization variable, the objective function, and the remaining constraints of (D1a), (D1b), (D2a), and (D2b). Note that $\mathcal{X}$ is a convex set but (30) is non-convex due to constraint $\xi(\mathbf{x}) \geq 0$. Assume that (30) is feasible and its primal optimal value is β^*.

The Lagrangian of (30) is given by $\mathcal{L}(\mathbf{x},\lambda) = \beta(\mathbf{x}) - \lambda\xi(\mathbf{x})$, where $\lambda \geq 0$ is the Lagrangian multiplier for $\xi(\mathbf{x})$. The dual problem of (30) is

$$\beta^+ \triangleq \max_{\lambda\geq 0} \min_{\mathbf{x}\in\mathcal{X}} \mathcal{L}(\mathbf{x},\lambda), \tag{31}$$

where β^+ is the dual optimal value. In general, $\beta^* \geq \beta^+$ holds for the non-convex problem (30) due to weak duality and $\beta^* - \beta^+$ is the duality gap of (30).

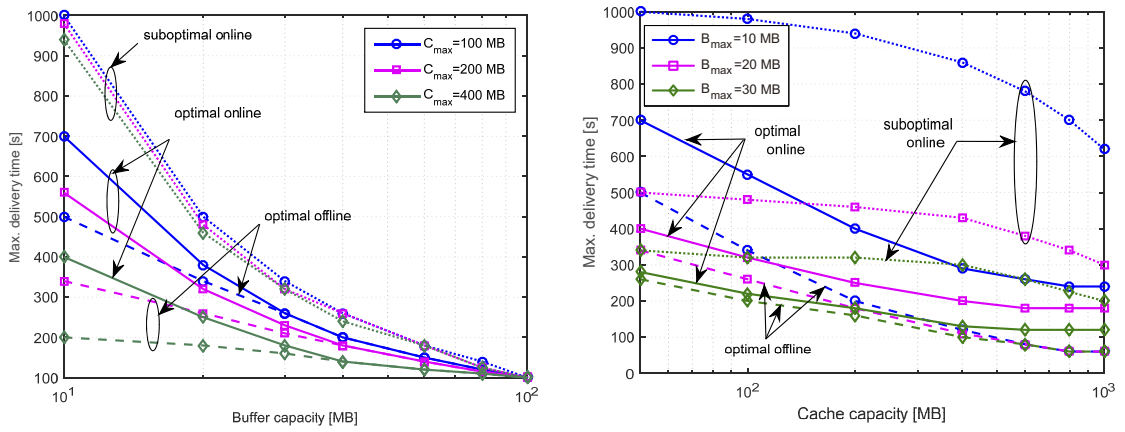
Define the perturbation function of (30) as [46, Chapter 6.2],

$$v(y) \triangleq \underset{\mathbf{x}\in\mathcal{X}}{\text{minimize}} \quad \beta(\mathbf{x}) \qquad \text{subject to} \quad \xi(\mathbf{x}) \geq y. \tag{32}$$

Note that $v(y)$ is a non-decreasing function of y. Moreover, we show below that $v(y)$ is a convex function as $F \to \infty$, i.e., for any $\varrho \in [0,1]$, we have

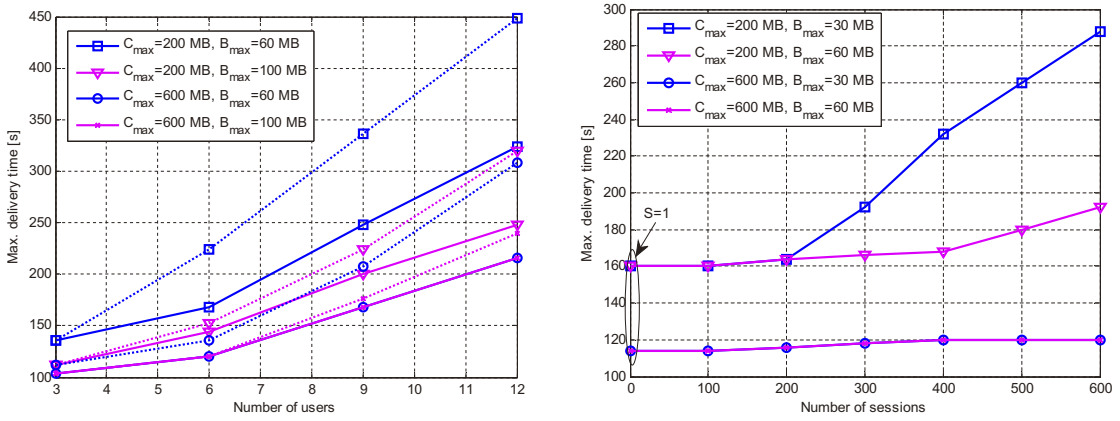
$$v(\varrho y_1 + (1-\varrho)y_2) \leq \varrho v(y_1) + (1-\varrho)v(y_2),\ \forall y_1, \forall y_2. \tag{33}$$

Specifically, let $\mathbf{x}_1^*$ and $\mathbf{x}_2^*$ be the optimal control policy of (32) for arbitrary perturbation variables y_1 and y_2, respectively. We assume that both y_1 and y_2 are feasible for (32), since,



(a) (b)

Fig. 5. Maximum delivery time versus (a) buffer capacity and (b) cache capacity for the optimal online scheme (solid line), optimal offline scheme (dashed line), and suboptimal online scheme (dotted line).



(a) (b)

Fig. 6. (a) Maximum delivery time versus number of users K for the optimal offline scheme (solid line) and the suboptimal online scheme (dotted line) and (b) maximum delivery time versus number of streaming sessions S for the optimal offline scheme and $K = 6$.

otherwise, $v(y) = +\infty$ and (33) trivially holds. For $F \to \infty$, the total bandwidth is divided into a set of infinitesimally narrow SCs, where several adjacent SCs within the coherence bandwidth have approximately the same channel gains. Now, we construct a "frequency-sharing" policy by dividing the coherence bandwidth into two portions, i.e., ϱ and $(1-\varrho)$, and apply the control policies $\mathbf{x}_1^*$ and $\mathbf{x}_2^*$ in each portion, respectively. It is easy to verify that the constructed policy, denoted by $\mathbf{x}_\varrho$, is feasible for $v(\varrho y_1 + (1-\varrho)y_2)$ since $\mathcal{X}$ is convex and $\xi(\mathbf{x}_\varrho) = \varrho\xi(\mathbf{x}_1^*) + (1-\varrho)\xi(\mathbf{x}_2^*) \geq \varrho y_1 + (1-\varrho)y_2$. Meanwhile, $\mathbf{x}_\varrho$ achieves an objective value of $\varrho v(y_1) + (1-\varrho)v(y_2)$. Due to the minimization operation in (32), thus (33) is true.

Because of the monotonicity and convexity of $v(y)$, there exist $\overline{\lambda} \geq 0$ satisfying $v(y) \geq v(0) + \overline{\lambda}(y)$, $\forall y$, where $\overline{\lambda}$ is the subgradient of $v(\cdot)$. Let $\overline{\mathbf{x}} \in \mathcal{X}$ be the optimal solution of (30). Then, according to [46, Theorem 6.2.7], $(\overline{\mathbf{x}}, \overline{\lambda})$ is a saddle point of $\mathcal{L}(\mathbf{x}, \lambda)$, i.e., $\mathcal{L}(\overline{\mathbf{x}}, \lambda) \leq \mathcal{L}(\overline{\mathbf{x}}, \overline{\lambda}) \leq \mathcal{L}(\mathbf{x}, \overline{\lambda})$, $\forall \mathbf{x} \in \mathcal{X}$, $\forall \lambda \geq 0$. Moreover, we have $\beta^* = \beta^+$, i.e., the duality gap vanishes as $F \to \infty$, due to [46, Theorem 6.2.5].

On the other hand, the binary relaxation problem of (30) is given by $\beta^- \triangleq \min_{\mathbf{x}\in\mathcal{X}} \beta(\mathbf{x})$, which is a convex problem and strong duality holds. Moreover, based on Lagrangian duality theory [36, Chapter 5], the dual problem of the relaxed problem is identical to (31), i.e., $\beta^- = \beta^+$. Consequently, $\beta^- = \beta^*$ as $F \to \infty$ and the tightness of the binary relaxation is proved.

In the second step, we prove the global optimality of Algorithm 1 for the outer-layer subproblems based on Proposition 1. Let l_k and u_k be the lower and upper bounds on the delivery time in iteration step $k \in \mathbb{N}$ during the bisection search, respectively. It is easy to verify that sequence $\{u_k\}$ is monotonically non-increasing while $\{l_k\}$ is monotonically non-decreasing. As a result, the delivery time bounds are improved in each iteration of the algorithm. Moreover, the search region in the kth iteration of the bisection search is given by $\mathcal{I}_k \triangleq \cap_{j=0}^{k}[l_j, u_j]$. Based on Proposition 1, $\mathcal{I}_k$ is a convex set. Specifically, we have $\mathcal{I}_k = [l_k, u_k]$. Thus, the search region shrinks in each iteration of the algorithm, i.e., $\mathcal{I}_{k+1} \subseteq \mathcal{I}_k$. If l_0 satisfies $\mathbb{I}_{\mathcal{A}}(l_0) = \mathbb{I}_{\mathcal{B}}(l_0) = \infty$, then the optimal delivery time T^* is contained in $\mathcal{I}_0 = [l_0, u_0]$ as the doubling search procedure terminates.

On the other hand, from the discussions above, the global optimum of the inner-layer problems can be obtained in each iteration of the bisection search (cf. line 11 of Algorithm 1) by solving their equivalent convex problems. As a result, T^* is always contained in $\cap_{j=0}^{k}[l_j, u_j]$ in the kth iteration of the bisection search. Since $u_0 < \infty$, the termination condition $u_k < l_k + 1$ will be satisfied in a finite number of iterations; moreover, for sufficiently large $J < \infty$, the solution obtained $\cap_{j=0}^{J}[l_j, u_j]$ is globally optimal, where $\{T^*\} = \cap_{j=0}^{J}[l_j, u_j] \subseteq [l_0, u_0]$. The

uniqueness of T^* is due to the monotonicity of the delivery time.

REFERENCES

[1] L. Xiang, D. W. K. Ng, T. Islam, R. Schober, and V. W. S. Wong, "Cross-layer optimization of fast video delivery in cache-enabled relaying networks," in *Proc. IEEE Global Comm. Conf. (GLOBECOM)*, San Diego, CA, Dec. 2015.

[2] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update 2015-2020," Feb. 2016.

[3] R. Knutson, "Video boom forces Verizon to upgrade network," *The Wall Street Journal*, Dec. 2013.

[4] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131–139, Feb. 2014.

[5] G. Paschos, E. Baştuğ, I. Land, G. Caire, and M. Debbah, "Wireless caching: Technical misconceptions and business barriers," *arXiv preprint arXiv:1602.00173*, Jun. 2016.

[6] F. Pantisano, M. Bennis, W. Saad, and M. Debbah, "Match to cache: Joint user association and backhaul allocation in cache-aware small cell networks," in *Proc. IEEE Int'l Conf. Comm. (ICC)*, London, UK, Jun. 2015.

[7] H. AhleHagh and S. Dey, "Video caching in Radio Access Network: Impact on delay and capacity," in *Proc. IEEE Wireless Comm. and Netw. Conf. (WCNC)*, Paris, France, Apr. 2012.

[8] D. Liu and C. Yang, "Energy efficiency of downlink networks with caching at base stations," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 907–922, Apr. 2016.

[9] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, New York, NY, Mar. 1999.

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

[11] C. Yang, Y. Yao, Z. Chen, and B. Xia, "Analysis on cache-enabled wireless heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 131–145, Jan. 2016.

[12] A. Liu and V. Lau, "Mixed-timescale precoding and cache control in cached MIMO interference network," *IEEE Trans. Signal Process.*, vol. 61, no. 24, pp. 6320–6332, Dec. 2013.

[13] ——, "Cache-enabled opportunistic cooperative MIMO for video streaming in wireless systems," *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 390–402, Jan. 2014.

[14] Z. Chen, J. Lee, T. Q. Quek, and M. Kountouris, "Cooperative caching and transmission design in cluster-centric small cell networks," *arXiv preprint arXiv:1601.00321*, Jan. 2016.

[15] M. Tao, E. Chen, H. Zhou, and W. Yu, "Content-centric sparse multicast beamforming for cache-enabled cloud RAN," *IEEE Trans. Wireless Commun.*, vol. 62, no. 24, pp. 6320–6332, Sep. 2016.

[16] S. Shariatpanahi, H. Shah-Mansouri, and B. Khalaj, "Caching gain in wireless networks with fading: A multi-user diversity perspective," in *Proc. IEEE Wireless Comm. and Netw. Conf. (WCNC)*, Istanbul, Turkey, Apr. 2014.

[17] M. Chiang and T. Zhang, "Fog networking and IoT: An overview on research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[18] I. Elsen, F. Hartung, U. Horn, M. Kampmann, and L. Peters, "Streaming technology in 3G mobile communication systems," *IEEE Computer*, no. 9, pp. 46–52, Sep. 2001.

[19] K. J. Ma, R. Bartos, S. Bhatia, and R. Nair, "Mobile video delivery with HTTP," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 166–175, Apr. 2011.

[20] A. Begen, T. Akgul, and M. Baugher, "Watching video over the web: Part 1: Streaming protocols," *IEEE Internet Comput.*, vol. 15, no. 2, pp. 54–63, Mar. 2011.

[21] B. Liu, W. Zhou, T. Zhu, L. Gao, T. Luan, and H. Zhou, "Silence is golden: Enhancing privacy of location-based services by content broadcasting and active caching in wireless vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9942–9953, Dec. 2016.

[22] L. Idir, S. Paris, and F. Naït-Abdesselam, "Optimal caching of encoded data for content distribution in vehicular networks," in *Proc. IEEE Int'l Conf. Comm. (ICC) - Workshop*, London, UK, Jun. 2015.

[23] N. Zlatanov, R. Schober, and P. Popovski, "Buffer-aided relaying with adaptive link selection," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 8, pp. 1530–1542, Aug. 2013.

[24] N. Zlatanov, A. Ikhlef, T. Islam, and R. Schober, "Buffer-aided cooperative communications: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 52, no. 4, pp. 146–153, Apr. 2014.

[25] D. P. Bertsekas, *Dynamic Programming and Optimal Control, vol. 1*, 3rd ed. Athena Scientific, 2005.

[26] E. Dahlman, S. Parkvall, and J. Sköld, *4G: LTE/LTE-Advanced for Mobile Broadband*. Academic Press, 2013.

[27] R. A. Berry and E. M. Yeh, "Cross-layer wireless resource allocation," *IEEE Signal Process. Mag.*, vol. 21, no. 5, pp. 59–68, Sep. 2004.

[28] G. Song and Y. Li, "Cross-layer optimization for OFDM wireless networks-part I: Theoretical framework," *IEEE Trans. Wireless Commun.*, vol. 4, no. 2, pp. 614–624, Mar. 2005.

[29] J. Tang and X. Zhang, "Cross-layer resource allocation over wireless relay networks for quality of service provisioning," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 4, pp. 645–656, May 2007.

[30] M. van Der Schaar and S. S. N, "Cross-layer wireless multimedia transmission: Challenges, principles, and new paradigms," *IEEE Wireless Commun. Mag.*, vol. 12, no. 4, pp. 50–58, Aug. 2005.

[31] D. Wu and R. Negi, "Effective capacity: A wireless link model for support of quality of service," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 630–643, Jul. 2003.

[32] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud RAN for mobile networks: A technology overview," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 405–426, 2015.

[33] G. B. Folland, *Real Analysis: Modern Techniques and Their Applications*. John Wiley & Sons, 2013.

[34] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*. Springer Science & Business Media, 2011.

[35] R. Horst, *Introduction to Global Optimization*. Springer Science & Business Media, 2000.

[36] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[37] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus, and A. Yener, "Transmission with energy harvesting nodes in fading wireless channels: Optimal policies," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 8, pp. 1732–1743, Aug. 2011.

[38] W. Yu and R. Lui, "Dual methods for nonconvex spectrum optimization of multicarrier systems," *IEEE Trans. Commun.*, vol. 54, no. 7, pp. 1310–1322, Jul. 2006.

[39] K. Seong, M. Mohseni, and J. Cioffi, "Optimal resource allocation for OFDMA downlink systems," in *Proc. Int'l. Sym. Inf. Theory (ISIT)*, Seattle, WA, Jul. 2006.

[40] D. W. K. Ng, E. S. Lo, and R. Schober, "Energy-efficient resource allocation in multi-cell OFDMA systems with limited backhaul capacity," *IEEE Trans. Wireless Commun.*, vol. 11, no. 10, pp. 3618–3631, Oct. 2012.

[41] Y. Ye, *Interior Point Algorithms: Theory and Analysis*. John Wiley & Sons, 1997.

[42] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," [Online] Available: http://cvxr.com/cvx, Mar. 2014.

[43] D. P. Bertsekas, *Dynamic Programming and Optimal Control, vol. 2*, 4th ed. Athena Scientific, 2012.

[44] A. Kolobov, "Planning with Markov decision processes: An AI perspective," *Synth. Lect. Artificial Intell. & Mach. Learn.*, vol. 6, no. 1, pp. 1–210, Jun. 2012.

[45] 3GPP TR 36.814, "Further advancements for E-UTRA physical layer aspects (Release 9)," Mar. 2010.

[46] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, 3rd ed. John Wiley & Sons, 2013.