Mixed-Timescale Online PHY Caching for Dual-Mode MIMO Cooperative Networks

An Liu, *Senior Member, IEEE*, Vincent K. N. Lau, *Fellow, IEEE*, Wenchao Ding, and Edmund Yeh, *Senior Member, IEEE*

***Abstract*— Recently, physical layer (PHY) caching has been proposed to exploit the *dynamic side information* induced by caches at base stations (BSs) to support coordinated multi-point (CoMP) and achieve high degrees of freedom (DoF) gains. Due to the limited cache storage capacity, the performance of PHY caching depends heavily on the cache content placement algorithm. In the existing algorithms, the cache content placement is adaptive to the long-term popularity distribution in an offline manner. We propose an online PHY caching framework, which adapts the cache content placement to *microscopic spatial and temporary popularity variations* to fully exploit the benefits of PHY caching. Specifically, the joint optimization of online cache content placement and content delivery is formulated as a mixed-timescale drift minimization problem to increase the CoMP opportunity and reduce the cache content placement cost. We propose a low-complexity algorithm to obtain a throughput-optimal solution. Moreover, we provide a closed-form characterization of the maximum sum DoF in the stability region and study the impact of key system parameters on the stability region. The simulations results show that the proposed online PHY caching framework achieves large gain over existing solutions.**

***Index Terms*— Online PHY caching, CoMP, throughput-optimal resource control, stability region.**

I. Introduction

MANY recent works have shown that wireless network performance can be substantially improved by exploiting caching in content-centric wireless networks [1], [2]. The early works [3]–[5] focus on exploiting caching to reduce the backhaul loading. In [6], [7], cache-enabled opportunistic CoMP (PHY caching) is proposed to mitigate interference and improve the spectral efficiency of the physical layer (PHY) in wireless networks with limited backhaul. Beside the common benefits of caching in fixed line networks, DoF gains can be achieved by utilizing a fundamental *cache-induced PHY topology change*. Specifically, when users' requested content exists in the cache of several BSs, the BS caches induce *dynamic side information*, which can be used to cooperatively transmit the requested packets to the users, thus achieving huge DoF gains. In practice, the cache storage capacity is limited, and hence, the cache content placement algorithm plays a key role in determining the performance of PHY caching schemes. The existing algorithms can be classified into two types.

A. Offline Cache Content Placement

Offline cache content placement is adaptive to the long-term popularity distribution in an offline manner. Once the cache content placement phase has finished, the cached content cannot be changed during the content delivery phase. In [6], [7], mixed-timescale optimizations of short-term MIMO precoding and long-term cache content placement are studied to support real-time video-on-demand applications. However, these offline caching algorithms cannot capture the *microscopic spatial and temporary popularity variations*.

B. Online Cache Content Placement

Online cache content placement is dynamically adaptive to microscopic spatial and temporary popularity variations during the content delivery phase. Compared to its offline counterpart, it has more refined control over the limited cache resources and thus can potentially achieve a better performance. In [8], online cache placement and request scheduling are studied to support elastic and inelastic traffic in wireless networks. In [9], a framework for joint online forwarding and caching is proposed within the context of Name Data Networks (NDNs). The throughput-optimal solution in [9] is to cache the content with the longest request queue at each node. In other words, the caching priority is based solely on the local popularity of content. However, the solution in [9] cannot be extended easily to PHY caching with cache-induced CoMP. This is because the local popularity at different BSs may vary widely, yielding a low cooperation opportunity. As such, a reasonable online caching policy should strike a delicate balance between local popularity at individual BSs and the cooperation opportunity among cooperative BSs.

Manuscript received May 11, 2018; revised October 28, 2018 and March 1, 2019; accepted March 15, 2019. Date of publication April 2, 2019; date of current version May 8, 2019. This work was supported by the Science and Technology Program of Shenzhen, China, under Grant JCYJ20170818113908577. The work of A. Liu was supported by Global Young Experts through the China Recruitment Program. The associate editor coordinating the review of this paper and approving it for publication was M. C. Gursoy. *(Corresponding author: An Liu.)*

A. Liu is with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: anliu@zju.edu.cn).

V. K. N. Lau and W. Ding are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong (e-mail: eeknlau@ece.ust.hk; wdingae@connect.ust.hk).

E. Yeh is with the Department of Electronic and Computer Engineering, Northeastern University, Boston, MA 02115 USA (e-mail: eyeh@ece.neu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2019.2907586

In this paper, we propose an online PHY caching framework to fully exploit the benefits of cache-induced opportunistic CoMP. The main contributions are summarized as follows.

- **Online PHY caching with cache content placement cost:** In [9], [10], the cache content placement cost is ignored, and hence, the cache content placement policy depends only on the current cache state. However, for practical consideration, it is important to model the cache content placement cost. In this case, the cache content placement policy should depend on both the previous and the current cache states. As a result, both the algorithm design and throughput optimality analysis are more complicated because they involve a history-dependent policy, as explained below.
- **Mixed-timescale optimization of online PHY caching and content delivery:** We apply the Lyapunov optimization framework [11], [12] to address the joint optimization of online PHY caching and content delivery. In our design, the cache content placement is updated at a slower timescale than the other control variables to reduce the cache content placement cost. With such mixed-timescale control variables, the algorithm design is based on minimizing a T-step VIP-drift-plus-penalty function, which contains both the previous and the current cache states. As such, the conventional Lyapunov optimization algorithms based on minimizing a 1-step drift-plus-penalty function over single-timescale control variables cannot be applied. By exploiting the specific structure of the problem, we propose a low-complexity mixed-timescale optimization algorithm and establish its throughput optimality. Due to the history-dependent policy and the mixed-timescale design, the throughput optimality analysis cannot directly follow the routine of conventional Lyapunov drift plus penalty theory in [11], [12]. To over come this challenge, we first introduce the concept of conditional flow balance constraint for a given cache state, and define the network stability region under conditional flow balance. Then we establish the throughput optimality by introducing a FRAME policy as a bridge to connect the proposed solution and the optimal random policy.
- **Closed-form characterization of the stability region:** We provide a simple characterization of the stability region (in terms of DoF), incorporating the effect of cooperative caching so as to study the impact of key system parameters on the stability region.

The online PHY caching has been proposed in the conference version [13], but without the analysis of throughput optimality and stability region. The rest of the paper is organized as follows: In Section II, we introduce the system model. In Section III, we elaborate the proposed online PHY caching and content delivery schemes. In Section IV, we propose a dual-mode-VIP-based resource control framework for wireless NDNs with dual-mode PHY. In Section V, we establish the throughput optimality of the proposed resource control algorithm. In Section VI, we characterize the stability region of wireless NDNs with a dual-mode PHY. Simulations are presented in Section VII, and conclusions are given in Section VIII.

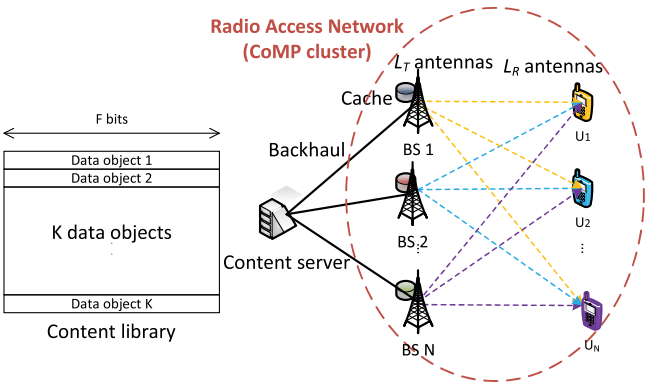


Fig. 1. Architecture of cached MIMO interference networks.

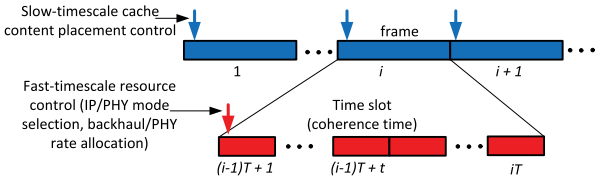


Fig. 2. Illustration of the time slot and frame.

II. System Model

A. Network Architecture

Consider a cached MIMO interference network with N BS-user pairs, as illustrated in Fig. 1. Each BS has L_T antennas and each user has L_R antennas. Each BS has transmit power P and a cache of $L_C F$ bits. There is a content server providing a content library $\mathcal{K}$ that contains K data objects, where each data object has F bits. The users request data objects from the content server via a radio access network (RAN). Each BS in the RAN is connected to the content server via a backhaul. The content server also serves as a central control node responsible for resource control of all the BSs.

For convenience, let $\mathcal{B}$ denote the set of BSs, $\mathcal{U}$ denote the set of users, $\mathcal{M} = \mathcal{B} \cup \mathcal{U}$ denote the set of all nodes, and g denote the content server. The serving BS of user $j \in \mathcal{U}$ is denoted as n_j and the associated user of BS $n \in \mathcal{B}$ is denoted as j_n. A user always sends its data object request to its serving BS. However, it may receive the requested data object from only the serving BS or from all BSs, depending on the PHY mode, as will be elaborated in the next subsection.

Time is partitioned into frames indexed by i, and each frame consists of T time slots indexed by t, as illustrated in Fig. 2. The fast-timescale resource control variables are updated at the beginning of each time slot. On the other hand, the slow-timescale resource control variables (cache content placement control) are updated at the beginning of each frame. Unless otherwise specified, t is used to index a time slot in frame i, i.e., $t \in [1 + (i-1)T, iT]$ and $i = \lceil \frac{t}{T} \rceil$.

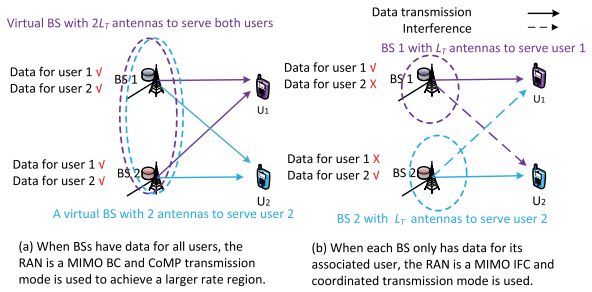


Fig. 3. Illustration of the dual-mode PHY.

B. Dual-Mode Physical Layer Model

The content server has knowledge of the global channel state information (CSI)[1] $\boldsymbol{H}(t) = \left\{\boldsymbol{H}_{jn}(t), \forall j \in \mathcal{U}, n \in \mathcal{B}\right\}$, where $\boldsymbol{H}_{jn}(t) \in \mathbb{C}^{L_R \times L_T}$ is the channel matrix between BS n and user j. $\boldsymbol{H}(t)$ is quasi-static within a time slot and i.i.d. between time slots. The time index t in $\boldsymbol{H}(t)$ will be omitted occasionally. There are two PHY modes, as elaborated below.

1) CoMP Transmission Mode (PHY Mode A): In this mode, the BSs can form a virtual transmitter and cooperatively send some data to all users. In this case, the RAN is a virtual MIMO broadcast channel (BC), as illustrated in Fig. 3-(a). Specifically, let $B_j(t)$ denote the data scheduled for delivery to user j at time slot t. In CoMP mode, $B_j(t)$ must be available at all BSs. The amount of scheduled data $\left|B_j(t)\right|$ (measured by the number of data objects) is limited by the data rate of the CoMP mode PHY, i.e., $\left|B_j(t)\right| = c_j^A(t)$, where $c_j^A(t)$ (data objects/slot) is the data rate of user j under CoMP mode with CSI $\boldsymbol{H}(t)$. To achieve the data rate $c_j^A(t)$, a coding scheme $\beta_{nj}^A: \{0,1\}^{c_j^A(t)F} \rightarrow \mathbb{C}^{L_T \times N_c}$ is applied at BS $n, \forall n \in \mathcal{B}$ for the data $B_j(t)$ requested by user j, which maps $B_j(t)$ to a codeword: $\boldsymbol{X}_{nj}^A(t) = \beta_{nj}^A\left(B_j(t)\right) \in \mathbb{C}^{L_T \times N_c}$, where N_c is the number of data symbol vectors per time slot. The signal $\boldsymbol{X}_n^A(t)$ transmitted from BS n in the t-th time slot is a superposition of the codewords for all users: $\boldsymbol{X}_n^A(t) = \sum_{j \in \mathcal{U}} \boldsymbol{X}_{nj}^A(t) \in \mathbb{C}^{L_T \times N_c}$, where each column vector in $\boldsymbol{X}_n^A(t)$ is transmitted from the L_T antennas during a symbol period in time slot t. Moreover, $\boldsymbol{X}_n^A(t)$ satisfies a power constraint $\mathrm{Tr}\left(\boldsymbol{X}_n^A(t)\left(\boldsymbol{X}_n^A(t)\right)^H\right)/N_c \leq P$. The received signal at user j in the t-th time slot is

$$\boldsymbol{Y}_j(t) = \sum_{n \in \mathcal{B}} \boldsymbol{H}_{jn}(t)\, \boldsymbol{X}_n^A(t) + \boldsymbol{Z}_j(t) \in \mathbb{C}^{L_R \times N_c}, \quad (1)$$

where $\boldsymbol{Z}_j(t)$ is the additive white Gaussian noise (AWGN). The rate $c_j^A(t)$ is achievable at the t-th time slot if user j can decode the scheduled data $B_j(t)$ with vanishing error probability as $N_c \rightarrow \infty$. The set of achievable rate vectors $\boldsymbol{c}^A(t) = \left[c_j^A(t)\right]_{j \in \mathcal{U}} \in \mathbb{R}_+^N$ forms the capacity region $C^A(\boldsymbol{H}(t)) \in \mathbb{R}_+^N$ under the CoMP mode.

2) Coordinated Transmission Mode (PHY Mode B): In this case, user j can only be served by the serving BS n_j and the RAN is a MIMO interference channel (IFC), as illustrated in Fig. 3-(b). Similarly, we have $\left|B_j(t)\right| = c_j^B(t)$, where $c_j^B(t)$ (data objects/slot) is the data rate of user j under coordinated mode with CSI $\boldsymbol{H}(t)$. To achieve the data rate $c_j^B(t)$, a coding scheme $\beta_j^B: \{0,1\}^{c_j^B(t)F} \rightarrow \mathbb{C}^{L_T \times N_c}$ is applied at BS j_n to map $B_j(t)$ to a codeword: $\boldsymbol{X}_j^B(t) = \beta_j^B\left(B_j(t)\right) \in \mathbb{C}^{L_T \times N_c}$, where $\boldsymbol{X}_j^B(t)$ satisfies a power constraint $\mathrm{Tr}\left(\boldsymbol{X}_j^B(t)\left(\boldsymbol{X}_j^B(t)\right)^H\right)/N_c \leq P$. The received signal at user j in the t-th time slot is

$$\boldsymbol{Y}_j(t) = \sum_{j' \in \mathcal{K}} \boldsymbol{H}_{jn_{j'}}(t)\, \boldsymbol{X}_{j'}^B(t) + \boldsymbol{Z}_j(t) \in \mathbb{C}^{L_R \times N_c}. \quad (2)$$

The rate $c_j^B(t)$ at the t-th time slot is achievable if the user j can decode the scheduled data $B_j(t)$ with vanishing error probability as $N_c \rightarrow \infty$. The set of achievable rate vectors $\boldsymbol{c}^B(t) = \left[c_j^B(t)\right]_{j \in \mathcal{U}} \in \mathbb{R}_+^N$ forms the capacity region $C^B(\boldsymbol{H}(t)) \in \mathbb{R}_+^N$ under PHY mode B.

We do not restrict the PHY to be any specific CoMP/coordinated transmission scheme (coding schemes $\left\{\beta_{nj}^A\right\}/\left\{\beta_j^B\right\}$) but consider an abstract PHY model represented by the capacity regions $C^A(\boldsymbol{H})$ and $C^B(\boldsymbol{H})$. As such, the proposed online PHY caching framework with the above abstract PHY model has the flexibility to incorporate various PHY technologies. Note that $C^A(\boldsymbol{H})$ and $C^B(\boldsymbol{H})$ depend on the transmit power P at each BS. Moreover, we have $C^B(\boldsymbol{H}) \subseteq C^A(\boldsymbol{H})$ according to the definitions of $C^A(\boldsymbol{H})$ and $C^B(\boldsymbol{H})$. In the following, we use linear precoding to illustrate the dual mode physical layer.

CoMP Mode Under Linear Precoding: In this mode, the N users are served using CoMP linear precoding between the BSs. The received signal for user j can be expressed as:

$$\boldsymbol{y}_j = \widetilde{\boldsymbol{H}}_j \boldsymbol{V}_j^A \boldsymbol{x}_j^A + \sum_{j' \neq j} \widetilde{\boldsymbol{H}}_j \boldsymbol{V}_{j'}^A \boldsymbol{x}_{j'}^A + \boldsymbol{z}_j, \quad (3)$$

where $\widetilde{\boldsymbol{H}}_j = \left[\boldsymbol{H}_{j1}, \ldots, \boldsymbol{H}_{jN}\right] \in \mathbb{C}^{L_R \times NL_T}$ is the composite channel matrix between all the BSs and user j; $\boldsymbol{x}_j^A \in \mathbb{C}^{d_j^A} \sim \mathcal{CN}(0, \boldsymbol{I})$ and d_j^A are respectively the data vector and the number of data streams for user j; $\boldsymbol{V}_j^A \in \mathbb{C}^{NL_T \times d_j^A}$ is the composite precoding matrix for user j, and $\boldsymbol{z}_j$ is the AWGN. For given CSI $\boldsymbol{H}$ and precoding matrices $\boldsymbol{V}^A = \left\{\boldsymbol{V}_j^A : \forall j\right\}$, the data rate (bps) of user j under the CoMP mode is [6]

$$c_j^A\left(\boldsymbol{H}, \boldsymbol{V}^A\right) = B_W \log_2 \left|\boldsymbol{I} + \widetilde{\boldsymbol{H}}_j \boldsymbol{V}_j^A \boldsymbol{V}_j^{AH} \widetilde{\boldsymbol{H}}_j^H \widetilde{\Omega}_j^{-1}\right|, \quad (4)$$

where B_W is the channel bandwidth, and $\widetilde{\Omega}_j = \boldsymbol{I} + \sum_{j' \neq j} \widetilde{\boldsymbol{H}}_j \boldsymbol{V}_{j'}^A \boldsymbol{V}_{j'}^H \widetilde{\boldsymbol{H}}_j^H$ is the interference-plus-noise covariance matrix of user j.

[1]The global CSI assumption is a common requirement for a lot of interference mitigation schemes, such as coordinated beamforming or cooperative MIMO [14]. The CSI signaling is not the bottleneck of backhaul loading compared with the payload symbols, because the former is done on the timescale of a coherence time (e.g. 20 ms for pedestrian users), while the latter has to be done on the timescale of payload symbols (e.g. 10 us in LTE). Within the LTE-A, there are already signaling mechanism for the eNB to acquire CSI from cooperating base stations (via CSI processes) [15]. There are also commercial deployment of LTE-A systems which demonstrates that global CSIT for CoMP can be achieved in practical systems.

Coordinated Mode Under Linear Precoding: In this mode, the user j can only be served by BS j using coordinated linear precoding. The received signal for user j can be expressed as:

$$\boldsymbol{y}_j = \boldsymbol{H}_{jn_j}\boldsymbol{V}_j^B\boldsymbol{x}_j^B + \sum_{j'\neq j}\boldsymbol{H}_{jn_{j'}}\boldsymbol{V}_{j'}^B\boldsymbol{x}_{j'}^B + \boldsymbol{z}_j, \tag{5}$$

where $\boldsymbol{x}_j^B \in \mathbb{C}^{d_j^B} \sim \mathcal{CN}(0,\boldsymbol{I})$ and d_j^B are respectively the data vector and the number of data streams for user j; and $\boldsymbol{V}_j^B \in \mathbb{C}^{L_T\times d_j^B}$ is the precoding matrix for user j. For given CSI $\boldsymbol{H}$ and precoding matrices $\boldsymbol{V}^B = \left\{\boldsymbol{V}_j^B{:}\forall j\right\}$, the data rate of user j under coordinated mode is [6]

$$c_j^B\left(\boldsymbol{H},\boldsymbol{V}^B\right) = B_W\log_2\left|\boldsymbol{I} + \boldsymbol{H}_{jn_j}\boldsymbol{V}_j^B\boldsymbol{V}_j^{BH}\boldsymbol{H}_{jn_j}^H\Omega_j^{-1}\right|, \tag{6}$$

where $\boldsymbol{\Omega}_j = \boldsymbol{I} + \sum_{j'\neq j}\boldsymbol{H}_{jn_{j'}}\boldsymbol{V}_{j'}^B\boldsymbol{V}_{j'}^{BH}\boldsymbol{H}_{jn_{j'}}^H$ is the interference-plus-noise covariance matrix.

III. Mixed-Timescale Online PHY Caching and Content Delivery Scheme

A. Slow-Timescale Online PHY Caching Scheme

In the proposed online PHY caching scheme, the cached data objects at each BS are updated once every frame (T time slots), as illustrated in Fig. 2. Since the local popularity variations at each BS usually change at a timescale much slower than the instantaneous CSI (slot interval), in practice, we may choose $T \gg 1$ to reduce the cache content placement cost without losing the ability to track microscopic spatial and temporary popularity variations.

Let $s_n^k(i) \in \{0,1\}$ denote the *cache state* of data object k at BS n, where $s_n^k(i) = 1$ means that data object k is in the cache of BS n at frame i and $s_n^k(i) = 0$ means the opposite. The *cache placement control action* at the beginning of the i-th frame is denoted by $\left\{p_n^k(i) \in \{-1,0,1\}, \forall n,k\right\}$, where $p_n^k(i) = -1$ and $p_n^k(i) = 1$ mean that the data object k is removed from and added to the cache of BS n at the beginning of the i-th frame respectively, and $p_n^k(i) = 0$ means that the cache state is unchanged. Note that there is no need to add an existing data object to the cache or remove a non-existing data object, i.e.,

$$\begin{aligned} &p_n^k(i) \neq 1, \text{when } s_n^k(i-1) = 1.\\ &p_n^k(i) \neq -1, \text{when } s_n^k(i-1) = 0. \end{aligned} \tag{7}$$

As such, the cache state dynamics is

$$s_n^k(i) = s_n^k(i-1) + p_n^k(i), \forall n \in \mathcal{B}, k \in \mathcal{K}. \tag{8}$$

The cache placement control action $\left\{p_n^k(i)\right\}$ must satisfy the following cache size constraint:

$$\sum_{k\in\mathcal{K}} s_n^k(i) \leq L_C, \forall i, \forall n \in \mathcal{B}. \tag{9}$$

Let $\boldsymbol{s}(i) = \left\{s_n^k(i), \forall n \in \mathcal{B}, k \in \mathcal{K}\right\}$ denote the *aggregate cache state*. When $p_n^k(i) = 1$, BS n needs to obtain data object k from the backhaul, which induces some *cache content placement cost*. To accommodate the traffic caused by cache content placement, the available backhaul capacity R (data objects/slot) at each BS is divided into a *data sub-channel* and a *control sub-channel* as $R = R_c + R_d$, where the data sub-channel with rate R_d is used for transmitting the data objects requested by users, and the control sub-channel is used for transmitting the data objects induced by the cache placement control and other control signaling. The cache content placement cost for BS n to cache data object k at frame i is $\Gamma_n^k(i) = \gamma\,\mathbf{1}_{\{p_n^k(i)=1\}}$, where $\mathbf{1}$ denotes the indication function, and γ is the price of fetching one data object using the control sub-channel. The total cost function at frame i is

$$\Gamma(i) = \sum_{n\in\mathcal{B},k\in\mathcal{K}}\Gamma_n^k(i) = \sum_{n\in\mathcal{B},k\in\mathcal{K}}\gamma\,\mathbf{1}_{\{p_n^k(i)=1\}}, \tag{10}$$

and the average cache content placement cost is

$$\overline{\Gamma} = \limsup_{J\to\infty}\frac{1}{J}\sum_{i=1}^{J}\mathbb{E}\left[\Gamma(i)\right]. \tag{11}$$

Remark 1: In this paper, we consider a two-hop wireless NDN where each BS is directly connected to the content server via a backhaul. This allows us to focus on the edge caching (PHY caching) design at the BSs. In a more complicated network, the BSs may connect to the content server via a gateway and core network. In this case, we can also employ centralized caching at the gateway. The centralized caching and PHY caching have different roles and they are complementary to each other. The role of centralized caching is to reduce the number of hops from the content server to the consumer as well as reduce the backhaul consumption between the gateway and core network, while the role of PHY caching is to strike a balance between inducing MIMO cooperation gain and reducing the BS backhaul consumption.

B. Fast-Timescale Dual-Mode Content Delivery Scheme

We consider a *dual-mode content delivery scheme*. Specifically, each data object is divided into D data chunks, and each data chunk is allocated a unique ID. The content delivery operates at the level of data chunks using two types of packets: *Interest Packets* (IPs) and *Data Packet*s (DPs). To request a data chunk, a user sends out an IP, which carries the ID of the data chunk, to the content server via its serving BS. For each IP from user j, the content server determines its mode according to an *IP mode selection* policy that will be elaborated in Section IV-C. If it is marked as a *CoMP Mode IP*, the corresponding DP will be delivered to all BSs and stored in the *CoMP mode data buffer* at each BS. If it is marked as a *coordinated mode IP*, the corresponding DP will be delivered to the serving BS n_j only, and stored in a *coordinated mode data buffer* at BS n_j. At each time slot, the content server also needs to determine the PHY mode $M_a(t) \in \{0,1\}$ according to the *PHY mode selection* policy elaborated in Section IV-C. If $M_a(t) = 0$ ($M_a(t) = 1$), the BSs will employ the coordinated (CoMP) mode to transmit some data from the coordinated (CoMP) mode data buffers to the users. The dual-mode content delivery has four components.

1) Component 1 (IP Mode Selection at Content Server): Let $Da_j^k(t)$ denote the number of IPs of data object k received by the content server from user j at time slot t, where $a_j^k(t)$ can be interpreted as the instantaneous arrival rate of IPs in the unit of data object/slot since each data object corresponds to D IPs. The content server will mark all these $Da_j^k(t)$ IPs using the same *IP mode*, denoted by $m_j^k(t) \in \{0, 1\}$.

2) Component 2 (Coordinated Mode DPs' Delivery to Each BS): If $m_j^k(t) = 0$, the corresponding $Da_j^k(t)$ DPs are called *coordinated mode DPs*, which will be delivered to the serving BS n_j only. Specifically, if BS n_j has data object k in the local cache ($s_{n_j}^k(i) = 1$), it creates $Da_j^k(t)$ DPs containing the requested data chunks indicated by the $Da_j^k(t)$ IPs. Otherwise, the content server will create the requested $Da_j^k(t)$ DPs and store them in the n_j-th data buffer with queue length Q_{gn_j} at the content server. These will be sent to BS n_j via backhaul when they become the head-of-the-queue DPs. In both cases, after obtaining the $Da_j^k(t)$ DPs, BS n_j will store them in the coordinated mode data buffer $Q_{n_j j}^B$.

3) Component 3 (CoMP Mode DPs' Delivery to Each BS): If $m_j^k(t) = 1$, the corresponding $Da_j^k(t)$ DPs are called *CoMP mode DPs*, which will be delivered to all BSs. For any $n \in \mathcal{B}$, if BS n has data object k in the local cache ($s_n^k(i) = 1$), it creates the requested $Da_j^k(t)$ DPs. Otherwise, the content server will create the requested $Da_j^k(t)$ DPs and store them in the n-th data buffer Q_{gn} at the content server. These will be sent to BS n via backhaul when they become the head-of-the-queue DPs. In both cases, after obtaining the $Da_j^k(t)$ DPs, BS n will store the $Da_j^k(t)$ DPs in the j-th CoMP mode data buffer Q_{nj}^A.

4) Component 4 (PHY Mode Determination at Content Server): The content server determines the *PHY mode* $M_a(t)$. If $M_a(t) = 0$, coordinated transmission mode will be used to send the data in $Q_{n j_n}^B$ to user j_n, $\forall j_n \in \mathcal{U}$, at rate $c_{j_n}^B(t)$ (data objects/slot). If $M_a(t) = 1$, CoMP transmission mode will be used to send the data in Q_{nj}^A to user j, $\forall j \in \mathcal{U}$, at rate $c_j^A(t)$. Let $\boldsymbol{c}^B(t) = \left[c_j^B(t)\right]_{j\in\mathcal{U}} \in \mathbb{R}_+^N$ and $\boldsymbol{c}^A(t) = \left[c_j^A(t)\right]_{j\in\mathcal{U}} \in \mathbb{R}_+^N$ denote the *PHY rate allocation* for the coordinated and CoMP modes respectively.

C. Data Packet Queue Dynamics

The dynamics of the queue Q_{gn} at the content server is

$$Q_{gn}(t+1) = \left(Q_{gn}(t) - c_{ng}(t)\right)^+ + b_n(t), \tag{12}$$

where $b_n(t) = \sum_{k\in\mathcal{K}} \overline{s}_n^k(i) \left(\sum_{j\in\mathcal{U}} m_j^k(t)\, a_j^k(t) + \overline{m}_{j_n}^k(t)\, a_{j_n}^k(t)\right)$ is the arrival rate of $Q_{gn}(t)$, $\overline{s}_n^k(i) = \mathbf{1}_{s_n^k(i)=0}$, $\overline{m}_j^k(t) = \mathbf{1}_{m_j^k(t)=0}$, and $c_{ng}(t)$ is the allocated transmission rate of DPs from the content server to BS n during time slot t. Let $Db_{n j_n}^B(t)$ ($Db_{nj}^A(t)$) denote the number of coordinated mode DPs of user j_n (CoMP mode IPs of user j) obtained at BS n from either the backhaul or the local cache. The dynamics of the queues at BS n are

$$\begin{aligned} Q_{n j_n}^B(t+1) &= \left(Q_{n j_n}^B(t) - \overline{M}_a(t)\, c_{j_n}^B(t)\right)^+ + b_{n j_n}^B(t), \\ Q_{nj}^A(t+1) &= Q_{nj}^A(t) - \min\left(M_a(t)\, c_j^A(t),\, \check{Q}_j^A(t)\right) \\ &\quad + b_{nj}^A(t), \end{aligned} \tag{13}$$

where $\overline{M}_a(t) = \mathbf{1}_{M_a(t)=0}$ and $\check{Q}_j^A(t) = \min_n Q_{nj}^A(t)$. Note that the queue length is measured using the number of data objects and the arrival rate is measured in data objects/slot because the proposed resource control framework operates at the data object level.

IV. Dual-Mode-VIP-Based Resource Control

In the proposed scheme, the slow-timescale cache content placement policy $\left\{p_n^k(i)\right\}$ is adaptive to the local popularity information at each node. The fast-timescale controls include the IP mode selection $\left\{m_j^k(t)\right\}$, backhaul rate allocation $\left\{c_{ng}(t)\right\}$, PHY mode selection $M_a(t)$, and PHY rate allocation $\left\{\boldsymbol{c}^A(t), \boldsymbol{c}^B(t)\right\}$, which are adaptive to the cache state $\left\{s_n^k(i)\right\}$ and global CSI $\boldsymbol{H}(t)$. However, the local popularity information is unavailable in the actual plane (network) due to interest collapsing and suppression, which refers to the operation that multiple unsatisfied requests (IPs) for the same DP at a node will be aggregated into a single IP [9]. This is good for efficiency, but also bad since we lose track of the actual expressed demand once the suppression happens. To overcome this challenge, we consider a *dual-mode VIP framework*, which creates a virtual plane (network) in which multiple interests are not suppressed via the introduction of Virtual Interest Packets (VIPs). As such, resource control algorithms operating in the virtual plane can take advantage of local popularity information (as represented by the VIP counts). Moreover, this dual-mode-VIP-based approach also reduces the algorithm complexity considerably (as compared with operating on DPs/IPs in the actual plane).

A. Transformation to a Virtual Network

The dual-mode VIP framework relies on the concept of VIPs flowing over a *virtual network*, as illustrated in Fig. 4. The virtual network is simulated at the content server and it has exactly the same topology, cache state $\left\{s_n^k(i)\right\}$ and global CSI $\boldsymbol{H}(t)$ as the actual network. Each virtual node $m \in \mathcal{N}$ maintains a VIP queue $V_m^k(t)$ for each data object k, which is implemented as a counter in the content server. The VIP queue $V_m^k(t)$ captures the local popularity at each (virtual) node, and the set of all VIP queues $\mathbf{V}(t) = \left\{V_m^k(t), \forall m \in \mathcal{M}, k \in \mathcal{K}\right\}$ captures microscopic popularity variations. Initially, all VIP queues are set to 0, i.e., $V_m^k(1) = 0, \forall m \in \mathcal{M}, k \in \mathcal{K}$. As the content server receives data object requests (IPs requesting the starting chunk of data objects) from users, the corresponding VIP queues $V_j^k(t)$, $j \in \mathcal{U}$ are incremented accordingly. After some number of VIPs in $V_j^k(t)$, $j \in \mathcal{U}$ have been "forwarded" to the virtual BSs (in the virtual network), the VIP queues $V_j^k(t)$, $j \in \mathcal{U}$ are decreased and the VIP queues $V_n^k(t)$, $n \in \mathcal{B}$ are increased by the same number accordingly. Similarly, after some number of VIPs in $V_n^k(t)$, $n \in \mathcal{B}$ have been "forwarded"

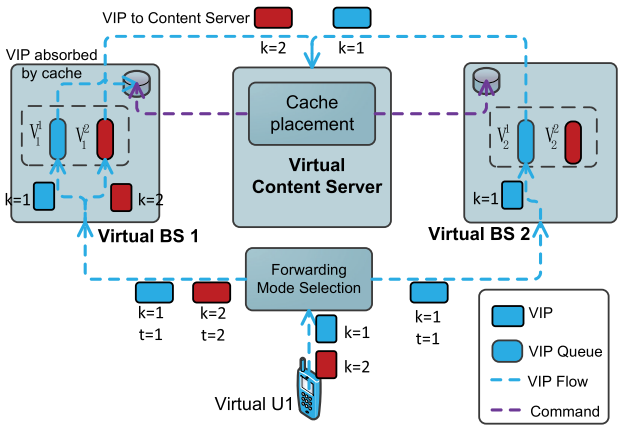


Fig. 4. Illustration of the VIP flow in the virtual network. In this example, user 1 requests data object 1 at time slot 1 and data object 2 at time slot 2. BS 1's cache contains data object 1 and BS 2's cache contains data object 2. The forwarding mode is CoMP mode at time slot 1, and coordinated mode at time slot 2. Therefore, the blue VIP with ID $k = 1$ is forwarded to both BSs, and the red VIP with ID $k = 2$ is forwarded to BS 1 only. Moreover, BS 1 (2) forwards the red (blue) VIP associated with data object 2 (1) to content server since data object 2 (1) is not stored at BS1 (2).

to the virtual content server (content source) and local cache, the VIP queues $V_n^k(t), n \in \mathcal{B}$ are decreased by the same number accordingly. Specifically, in the virtual network, there are two modes for "forwarding" the VIPs from the virtual users to virtual BSs, corresponding to the two PHY modes in the actual plane.

In the *CoMP forwarding mode*, VIPs in $V_j^k(t)$ are forwarded to all virtual BSs, and thus at time $t+1$, the VIP queues become

$$\begin{aligned} V_j^k(t+1) &= \left(V_j^k(t) - \mu_j^{Ak}(t)\right)^+ + A_j^k(t), \\ V_n^k(t+1) &= \Bigg(\left(V_n^k(t) - \mu_{ng}^k(t)\right)^+ \\ &\quad + \sum_{j\in\mathcal{U}} \mu_j^{Ak}(t) - r_n s_n^k(i)\Bigg)^+, \forall j \in \mathcal{U}, n \in \mathcal{B} \end{aligned} \tag{14}$$

where $A_j^k(t)$ is the number of exogenous data object request arrivals at the VIP queue $V_j^k(t)$ during slot t, $\mu_j^{Ak}(t)$ is the allocated transmission rate of VIPs for data object k from virtual user j to all virtual BSs during time slot t with CoMP forwarding mode, $\mu_{ng}^k(t)$ is the allocated transmission rate of VIPs for data object k from virtual BS n to the virtual content server during time slot t, and r_n is the maximum rate (in data objects/slot) at which BS n can produce copies of cached object k (e.g., the maximum rate r_n may reflect the I/O rate of the storage disk). On the other hand, in the coordinated forwarding mode, VIPs in $V_j^k(t)$ are "forwarded" to the serving virtual BS n_j only, and thus at time $t+1$, the VIP queues become

$$\begin{aligned} V_j^k(t+1) &= \left(V_j^k(t) - \mu_j^{Bk}(t)\right)^+ + A_j^k(t), \\ V_n^k(t+1) &= \left(\left(V_n^k(t) - \mu_{ng}^k(t)\right)^+ + \mu_{j_n}^{Bk}(t) - r_n s_n^k(i)\right)^+, \end{aligned} \tag{15}$$

TABLE I
KEY NOTATIONS IN THE ACTUAL AND VIRTUAL NETWORKS

Notations in the actual and virtual networks
PHY mode M_a and IP modes $\{m_j^k\}$
PHY rates $\{\boldsymbol{c}^A, \boldsymbol{c}^B\}$
Backhaul rates $\{c_{ng}\}$
DP queues $\{Q_{nj}^A, Q_{njn}^B, Q_{gn}\}$
IPs arrival rates $\{a_j^k\}$
Stability region Λ_c
Forwarding mode M
Forwarding rates between BSs and users $\{\boldsymbol{\mu}^A, \boldsymbol{\mu}^B\}$
Forwarding rates between BSs and server $\{\mu_{ng}^k\}$
VIP queues $\{V_n^k, V_j^k\}$
VIPs arrival rates $\{A_j^k\}$
VIP stability region Λ_v

$\forall j \in \mathcal{U}, n \in \mathcal{B}$, where $\mu_j^{Bk}(t)$ is the allocated transmission rate of VIPs for data object k from virtual user j to virtual BS n_j during time slot t with coordinated forwarding mode.

Combining the above two cases, the VIP queue dynamics can be expressed in a compact form:

$$\begin{aligned} V_j^k(t+1) &= \left(V_j^k(t) - \mu_j^{Ak}(t)M(t) - \mu_j^{Bk}(t)\overline{M}(t)\right)^+ \\ &\quad + A_j^k(t) \\ V_n^k(t+1) &= \Bigg(\left(V_n^k(t) - \mu_{ng}^k(t)\right)^+ + \sum_{j\in\mathcal{U}} \mu_j^{Ak}(t)M(t) \\ &\quad + \mu_{j_n}^{Bk}(t)\overline{M}(t) - r_n s_n^k(i)\Bigg)^+, \end{aligned} \tag{16}$$

$\forall j \in \mathcal{U}, n \in \mathcal{B}, k \in \mathcal{K}$, where $M(t) \in \{0, 1\}$ is the forwarding mode at time slot t in the virtual plane ($M(t) = 0$ stands for the coordinated forwarding mode and $M(t) = 1$ stands for the CoMP forwarding mode), and $\overline{M}(t) = \mathbf{1}_{M(t)=0}$.

In Table I, we list the key notations in the actual network and the corresponding notations in the virtual network for easy reference.

B. Mixed-Timescale Resource Control in Virtual Network

A mixed-timescale resource control algorithm determines the slow-timescale cache content placement policy $\{p_n^k(i)\}$ and the fast-timescale policies in the virtual network, aiming at solving the following problem:

$$\begin{aligned} &\min \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=1}^{t} \sum_{m\in\mathcal{M},k\in\mathcal{K}} \mathbb{E}\left[V_m^k(\tau)\right] \\ &\quad + W \limsup_{J\to\infty} \frac{1}{J}\sum_{i=1}^{J} \mathbb{E}\left[\Gamma(i)\right], \\ &\text{s.t.} \sum_{k\in\mathcal{K}} s_n^k(i) \le B_C, \forall n \in \mathcal{B}, \forall i; \sum_{k\in\mathcal{K}} \mu_{ng}^k(t) \le R_d, \forall n, \\ &\quad \boldsymbol{\mu}^A(t) \in C^A(\boldsymbol{H}); \boldsymbol{\mu}^B(t) \in C^B(\boldsymbol{H}), \forall t, \end{aligned} \tag{17}$$

where the objective function is a weighted sum of the total average VIP queue length and average cache placement cost, W is a price factor which can be used to control the tradeoff between the stability (measured by the total average VIP queue

length) and average cache placement cost, the optimization variables are $\left\{p_n^k(i), M(t), \mu_j^{Ak}(t), \mu_j^{Bk}(t), \mu_{ng}^k(t)\right\}$. However, it is difficult to directly solve this problem. Therefore, we resort to the Lyapunov optimization framework [11], [12] and approximately solve this problem by minimizing a mixed-timescale drift-plus-penalty as follows.

1) Slow-Timescale Cache Content Placement Solution: Define $\mathcal{L}(\mathbf{V}(t)) \triangleq \sum_{n\in\mathcal{N},k\in\mathcal{K}} \left(V_n^k(t)\right)^2$ as the Lyapunov function, which is a measure of unsatisfied requests in the network. The cache content placement $\left\{p_n^k(i)\right\}$ is designed to minimize the *T-step VIP-drift-plus-penalty* defined as

$$\Delta_T(i) \triangleq \mathbb{E}\left[\mathcal{L}(\mathbf{V}(t_0^i+T)) - \mathcal{L}(\mathbf{V}(t_0^i))|\boldsymbol{X}(t_0^i)\right] + W\mathbb{E}\left[\sum_{n\in\mathcal{B},k\in\mathcal{K}} \gamma\, 1_{\{p_n^k(i)=1\}}|\boldsymbol{X}(t_0^i)\right], \quad (18)$$

where t_0^i is the starting time slot of the i-th frame, and $\boldsymbol{X}(t_0^i) = \left[\mathbf{V}(t_0^i)\, \boldsymbol{s}(i-1)\right]$ is the observed system state at the beginning of the frame. Intuitively, if the first term in $\Delta_T(i)$ is negative, the VIP lengths tend to decrease. On the other hand, the second term in $\Delta_T(i)$ is the weighted cache content placement cost, and W is a price factor. Therefore, minimizing $\Delta_T(i)$ helps to strike a balance between stability and cost reduction. Following a similar analysis to that in the proof of Lemma 3 in [16], we obtain an upper bound of $\Delta_T(i)$.

Theorem 1: (T-step Drift-Plus-Penalty Upper Bound). An upper bound of $\Delta_T(i)$ *is* $\widetilde{\Delta}_T(i) \triangleq \mathbb{E}\left[\Delta_T^U(i)\,|\boldsymbol{X}(t_0^i)\right]$, *where*

$$\Delta_T^U(i) = W\sum_{n\in\mathcal{B},k\in\mathcal{K}} \gamma\, 1_{\{p_n^k(i)=1\}} + \overline{\Delta} - 2T\sum_{n\in\mathcal{B},k\in\mathcal{K}} V_n^k(t_0^i) r_n \left[s_n^k(i-1) + p_n^k(i)\right], \quad (19)$$

and $\overline{\Delta}$ *is a term independent of* $\left\{p_n^k(i)\right\}$.

Please refer to [17] for the detailed proof. The slow-timescale drift minimization problem is

$$\min_{\{p_n^k(i)\}} \Delta_T^U(i) \text{ :s.t. } (7-9) \text{ are satisfied.} \quad (20)$$

The detailed steps to find the optimal solution of (20) are summarized in Algorithm 1, which only has linear complexity w.r.t. the number of data objects K. In Algorithm 1, $V_n^k = V_n^k\left(t_0^i\right)$, $\mathcal{C}$ is the set of currently cached data objects at BS n, $\mathcal{C}'$ is a set of B_C data objects with the highest VIP counts (popularity), $\mathcal{O}' = \mathcal{C}'/\mathcal{C}$ is the set of the most popular data objects which have not been cached, and $\mathcal{O}$ is the set of currently cached data objects which are not in $\mathcal{C}'$. Each data object k_i' in $\mathcal{O}'$ will be added to the cache (i.e., $p_n^{k_i'}(i) = 1$) if the benefit of caching it, as indicated by the backlog difference $\left(V_n^{k_i'} - V_n^{k_i}\right) r_n$, exceeds the cache content placement cost threshold $\frac{W}{2T}\gamma$. If data object k_i' is added to the cache and $i > \left(\left|\mathcal{O}'\right| - |\mathcal{O}|\right)^+$, data object k_i in $\mathcal{O}$ will be removed (i.e., $p_n^{k_i}(i) = -1$) to save space for caching data object k_i'. It is not hard to prove that Algorithm 1 finds the optimal solution of (20). The detailed proof is given in [17].

Algorithm 1 Slow-Timescale Cache Content Placement Solution at Frame i

For each base station n,

- Let $\mathcal{C} = \left\{k|s_n^k(i-1) = 1, k\in\mathcal{K}\right\}$.
- Let $\mathcal{C}' = \left\{k|s_n^{k\star}(i) = 1\right\}$, where $\left\{s_n^{k\star}(i)\right\}$ is the optimal solution of $\max_{\{s_n^k\}} \sum_{k\in\mathcal{K}} V_n^k s_n^k$, $\sum_{k\in\mathcal{K}} s_n^k \le B_C$.
- Let $\mathcal{O}' = \mathcal{C}'/\mathcal{C}$, $\mathcal{O} = \mathcal{C}/\mathcal{C}'$.

  Sort the queue backlogs as, $V_n^{k_1'} \ge \cdots \ge V_n^{k_{|\mathcal{O}'|}'}$, $0 \le 0 \le \cdots \le V_n^{k_{|\mathcal{O}'|-|\mathcal{O}|+1}} \le \cdots \le V_n^{k_{|\mathcal{O}'|}}$.
- For $i = 1$: $\left(\left|\mathcal{O}'\right| - |\mathcal{O}|\right)^+$, if $V_n^{k_i'} r_n \ge \frac{W}{2T}\gamma$, then let $p_n^{k_i'}(i) = 1$.
- For $i = \left(\left|\mathcal{O}'\right| - |\mathcal{O}|\right)^+ + 1$: $\left|\mathcal{O}'\right|$, if $\left(V_n^{k_i'} - V_n^{k_i}\right) r_n \ge \frac{W}{2T}\gamma$, then let $p_n^{k_i'}(i) = 1$ and $p_n^{k_i}(i) = -1$.

2) Fast-Timescale Control Solution: The fast-timescale control solution is obtained by solving the following 1-*step VIP-drift minimization problem:*

$$\begin{aligned}\min\ & M(t)\sum_{j\in\mathcal{U},k\in\mathcal{K}} \mu_j^{Ak}(t)\left(\sum_{n\in\mathcal{B}} V_n^k(t) - V_j^k(t)\right) \\ & + \overline{M}(t)\sum_{j\in\mathcal{U},k\in\mathcal{K}} \mu_j^{Bk}(t)\left(V_{n_j}^k(t) - V_j^k(t)\right) \\ & - \sum_{n\in\mathcal{B},k\in\mathcal{K}} V_n^k(t)\mu_{ng}^k(t) \quad (21)\\ \text{s.t.}\ & \sum_{k\in\mathcal{K}} \mu_{ng}^k(t) \le R_d, \forall n, \\ & \boldsymbol{\mu}^A(t) \in C^A(\boldsymbol{H}); \boldsymbol{\mu}^B(t) \in C^B(\boldsymbol{H}), \quad (22)\end{aligned}$$

where $\boldsymbol{\mu}^A(t) = \left[\sum_{k\in\mathcal{K}} \mu_j^{Ak}(t)\right]_{j\in\mathcal{U}} \in \mathbb{R}_+^N$, $\boldsymbol{\mu}^B(t) = \left[\sum_{k\in\mathcal{K}} \mu_j^{Bk}(t)\right]_{j\in\mathcal{U}} \in \mathbb{R}_+^N$, (22) is the link capacity constraint, and the optimization variables are $\left\{M(t), \mu_{ng}^k(t), \mu_j^{Ak}(t), \mu_j^{Bk}(t)\right\}$. The detailed steps to solve (21) are summarized in Algorithm 2. Note that the weighted sum-rate maximization problems in (23) and (24) can be solved by existing algorithms for different CoMP/coordinated transmission schemes [18].

C. Virtual-to-Actual Control Policy Mapping

In the following, we propose a *virtual-to-actual control policy mapping* which can generate a resource control policy for the actual network from that in the virtual network.

1) Mapping for Cache Placement Control Policy $\left\{p_n^k(i)\right\}$*:* The cache placement control action in the actual network is the same as that in the virtual network.

2) Mapping for IP Mode Selection Policy $\left\{m_j^k(t)\right\}$*:* For a given forwarding mode selection and rate allocation policy $\left\{M(t), \mu_j^{Ak}(t)\right\}$ in the virtual network that achieves an average transmission rate of VIPs $\overline{\mu}_j^{Ak} = \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=1}^t M(\tau)\,\mu_j^{Ak}(\tau)$ for data object k from virtual user j to all virtual BSs, we need to construct an IP mode selection policy $\left\{m_j^k(t)\right\}$ in the actual network such that the

Algorithm 2 Fast-Timescale Control Solution at Slot t

1. Backhaul rate allocation

Let $\mu_{ng}^{k}(t) = \begin{cases} R_d & k = k_n^*(t) \\ 0 & otherwise \end{cases}, \forall n \in \mathcal{B}$, where $k_n^*(t) \triangleq \arg\max_k V_n^k(t)$.

2. Forwarding mode selection and rate allocation

Let

$$\left\{\mu_j^{Ak*}(t)\right\} = \underset{\left\{\mu_j^{Ak}(t)\right\}}{\text{argmax}} \sum_{j\in\mathcal{U},k\in\mathcal{K}} \mu_j^{Ak}(t)\left(V_j^k(t) - \sum_{n\in\mathcal{B}} V_n^k(t)\right) \quad \text{s.t. } \boldsymbol{\mu}^A(t) \in C^A(\boldsymbol{H}) \tag{23}$$

$$\left\{\mu_j^{Bk*}(t)\right\} = \underset{\left\{\mu_j^{Bk}(t)\right\}}{\text{argmax}} \sum_{j\in\mathcal{U},k\in\mathcal{K}} \mu_j^{Bk}(t)\left(V_j^k(t) - V_{n_j}^k(t)\right) \quad \text{s.t. } \boldsymbol{\mu}^B(t) \in C^B(\boldsymbol{H}). \tag{24}$$

Let $\Delta_1^A = \sum_{j\in\mathcal{U},k\in\mathcal{K}} \mu_j^{Ak*}(t)\left(V_j^k(t) - \sum_{n\in\mathcal{B}} V_n^k(t)\right)$ and $\Delta_1^B = \sum_{j\in\mathcal{U},k\in\mathcal{K}} \mu_j^{Bk*}(t)\left(V_j^k(t) - V_{n_j}^k(t)\right)$.

If $\Delta_1^A \geq \Delta_1^B$, **let** $M(t) = 1$, $\left\{\mu_j^{Ak}(t)\right\} = \left\{\mu_j^{Ak*}(t)\right\}$, $\mu_j^{Bk}(t) = 0, \forall j, k$;

Else, **let** $M(t) = 0$, $\mu_j^{Ak}(t) = 0, \forall j, k$, $\left\{\mu_j^{Bk}(t)\right\} = \left\{\mu_j^{Bk*}(t)\right\}$.

same average transmission rate of IPs for data object k from user j to all BSs can be achieved, i.e.,

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=1}^{t} m_j^k(\tau)\, a_j^k(\tau) = \overline{\mu}_j^{Ak}. \tag{25}$$

To achieve this, the content server maintains a set of *virtual CoMP queues* as

$$U_j^{Ak}(t+1) = U_j^{Ak}(t) - m_j^k(t)\, a_j^k(t) + M(t)\, \mu_j^{Ak}(t), \forall j, k, \tag{26}$$

where $U_j^{Ak}(1) = 0, \forall j, k$. Clearly, by setting the forwarding mode in the actual plane as $m_j^k(t) = \mathbf{1}_{U_j^{Ak}(t+1)>0}, \forall t$, we can achieve a bounded $\left|U_j^{Ak}(t+1)\right|, \forall t$ (since $\mu_j^{Ak}(t)$ is bounded), which implies that (25) can also be satisfied.

3) Mapping for PHY Mode Selection and Rate Allocation Policy: For the PHY mode selection and rate allocation policy in the actual plane, we let $c_{ng}(t) = \sum_{k\in\mathcal{K}} \mu_{ng}^k(t)$, $M_a(t) = M(t)$, $\boldsymbol{c}^A(t) = \boldsymbol{\mu}^A(t)$ and $\boldsymbol{c}^B(t) = \boldsymbol{\mu}^B(t)$.

V. Throughput Optimality Analysis

In this section, we first introduce the concept of the *network stability region under conditional flow balance*. Then we establish the equivalence between the virtual and actual networks, and the throughput optimality of the proposed VIP-based resource control algorithm. For all the theoretical analysis in this and the next section, we make the following standard assumptions on the arrival processes $A_j^k(t)$'s: (i) The arrival processes $\left\{A_j^k(t); t = 1, 2, \ldots\right\}$ are mutually independent with respect to j and k; and (ii) for all $j \in \mathcal{U}, k \in \mathcal{K}$, $\left\{A_j^k(t); t = 1, 2, \ldots\right\}$ are i.i.d. with respect to t and $A_j^k(t) \leq A_{max}^k$ for all t.

A. Motivation of Conditional Flow Balance

In practice, a basic QoS requirement is to maintain the queue stability for the data flow of each user. Specifically, a queue $Q(t)$ is stable if

$$\overline{Q} \triangleq \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=1}^{t} \mathbb{E}\left[Q(\tau)\right] < \infty, \tag{27}$$

where $\overline{Q}$ is called the *limiting average queue length* of $Q(t)$. A necessary condition for a $Q(t)$ with dynamics $Q(t+1) = (Q(t) - b(t))^+ + a(t)$ to be stable is that the following *flow balance constraint* is satisfied:

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=1}^{t} \mathbb{E}\left[a(\tau) - b(\tau)\right] \leq 0. \tag{28}$$

The conventional *network stability region* Λ is defined as the closure of the set of all arrival rate tuples $\boldsymbol{\lambda} = \left(\lambda_j^k\right)_{j\in\mathcal{U},k\in\mathcal{K}}$ for which there exists some resource control policy which can guarantee that all data queues are stable, where $\lambda_j^k = \lim_{t\to\infty} \frac{1}{t}\sum_{\tau=1}^{t} A_j^k(\tau)$ is the long-term exogenous VIP arrival rate at the VIP queue $V_j^k(t)$. To guarantee the basic QoS requirements of all users, the system should not operate at a point outside the stability region.

However, the flow balance or queue stability constraint is not sufficient to guarantee a good performance for practical cached interference networks, as explained below. In cached networks, the cache state $\boldsymbol{s}(i)$ has a huge impact on the arrival rate of $Q_{gn}(t)$. Specifically, let $\mathcal{K}_n$ denote the subset of data objects that need to be delivered to BS n. When $s_n^k(i) = 1, \forall k \in \mathcal{K}_n$, all the requests about data objects in $\mathcal{K}_n$ will be served by the local cache at BS n, and thus $b_n(t) = 0$. When $s_n^k(i) = 0, \forall k \notin \mathcal{K}_n$, all the requests about data objects in $\mathcal{K}_n$ will be forwarded to the content server, and thus $b_n(t)$ is large. Recall that in practice, $\boldsymbol{s}(i)$ is a *slow-timescale process* which can only change at the timescale of frames ($T \gg 1$ time slots) to avoid frequent cache content placement. If only the flow balance or queue stability constraint is considered, $s_n^k(i), \forall k \in \mathcal{K}_n$ may remain 0 for several frames, during which $Q_{gn}(t)$ may keep growing. As a result, the average delay of DPs would be in the order of several frames, which is unacceptable in practice. To address this issue, we introduce the concept of the *network stability region under conditional flow balance* $\Lambda_c \subset \Lambda$, as will be formally defined in the next subsection. When the arrival rates $\left(\lambda_j^k\right) \in \Lambda_c$, there exists some resource control policy which can guarantee that the flow balance is satisfied conditioned on any *cache state* $\boldsymbol{s}$ of non-zero probability. This stronger notion of stability ensures that the system will not operate at a point with excessively large delay. Besides the above practical consideration, imposing the conditional flow balance constraint also makes it more tractable to establish the throughput optimality of the proposed algorithm.

B. Stability Region under Conditional Flow Balance

The *limiting probability* that a cache state $\boldsymbol{s}$ occurs is

$$\pi_{\boldsymbol{s}} = \limsup_{i\to\infty} \mathbb{E}\left[\frac{|\mathcal{T}(\boldsymbol{s}, i)|}{i}\right], \tag{29}$$

where $\mathcal{T}(\boldsymbol{s},i) = \{\tau \le i : \boldsymbol{s}(\tau) = \boldsymbol{s}\}$. Let $\mathcal{S} = \{\boldsymbol{s} : \pi_{\boldsymbol{s}} > 0\}$ denote the set of all cache states with non-zero limiting probability. Then the *conditional flow balance constraint* is

$$\overline{b}_{n|\boldsymbol{s}} \le \overline{c}_{ng|\boldsymbol{s}}, \overline{b}^B_{nj_n|\boldsymbol{s}} \le \overline{c}^B_{j_n|\boldsymbol{s}}, \overline{b}^A_{nj|\boldsymbol{s}} \le \overline{c}^A_{j|\boldsymbol{s}} \tag{30}$$

$\forall j \in \mathcal{U}, n \in \mathcal{B}$ and $\forall \boldsymbol{s} \in \mathcal{S}$, where $\left(\overline{b}_{n|\boldsymbol{s}}, \overline{c}_{ng|\boldsymbol{s}}\right)$, $\left(\overline{b}^B_{nj_n|\boldsymbol{s}}, \overline{c}^B_{j_n|\boldsymbol{s}}\right)$ and $\left(\overline{b}^A_{nj|\boldsymbol{s}} \overline{c}^A_{j|\boldsymbol{s}}\right)$ are *$\boldsymbol{s}$-conditional average rates* of $\left(b_n(t), c_{ng}(t)\right)$, $\left(b^B_{nj_n}(t), \overline{M}_a(t) c^B_{j_n}(t)\right)$ and $\left(b^A_{nj}(t), M_a(t) c^A_j(t)\right)$ (arrival/departure rates of $Q_{gn}(t)$, $Q^B_{nj_n}(t)$ and $Q^A_{nj}(t)$, respectively), i.e., average rates over all frames with cache state $\boldsymbol{s}$. For example, the *$\boldsymbol{s}$-conditional average departure rate* of $Q^B_{nj_n}(t)$ is defined as

$$\overline{c}^B_{j_n|\boldsymbol{s}} = \limsup_{i\to\infty} \mathbb{E}\left[\frac{\sum_{\tau\in\mathcal{T}(\boldsymbol{s},i)}}{T\left|\mathcal{T}(\boldsymbol{s},i)\right|} \sum_{t=1+(\tau-1)T}^{\tau T} \overline{M}_a(t) c^B_{j_n}(t)\right]. \tag{31}$$

The other *$\boldsymbol{s}$-conditional average rates* are defined similarly.

Definition 1: The *network stability region under conditional flow balance* Λ_c is the closure of the set of all arrival rate tuples $\boldsymbol{\lambda}$ for which there exists some resource control policy which can guarantee that all data queues are stable and also satisfies the cache size constraint (9), conditional flow balance constraint (30), and the following link capacity constraint:

$$c_{ng}(t) \le R_d, \forall n; \boldsymbol{c}^A(t) \in C^A(\boldsymbol{H}); \boldsymbol{c}^B(t) \in C^B(\boldsymbol{H}); \forall t. \tag{32}$$

Similarly, we can define the *VIP stability region under conditional flow balance*. In the virtual plane, the conditional flow balance constraint is

$$\lambda^k_j \le \overline{\mu}^{Ak}_{j|\boldsymbol{s}} + \overline{\mu}^{Bk}_{j|\boldsymbol{s}}; \sum_{j\in\mathcal{U}} \overline{\mu}^{Ak}_{j|\boldsymbol{s}} + \overline{\mu}^{Bk}_{j_n|\boldsymbol{s}} \le \overline{\mu}^k_{ng|\boldsymbol{s}} + r_n s^k_n, \tag{33}$$

$\forall j \in \mathcal{U}, k \in \mathcal{K}, n \in \mathcal{B}$ and $\forall \boldsymbol{s} \in \mathcal{S}$, where $\overline{\mu}^{Ak}_{j|\boldsymbol{s}}$, $\overline{\mu}^{Bk}_{j|\boldsymbol{s}}$ and $\overline{\mu}^k_{ng|\boldsymbol{s}}$ are *$\boldsymbol{s}$-conditional average rates* of $\mu^{Ak}_j(t)M(t)$, $\mu^{Bk}_j(t)\overline{M}(t)$ and $\mu^k_{ng}(t)$, whose definitions are similar to (31).

Definition 2: The *VIP stability region under conditional flow balance* Λ_v is the closure of the set of all arrival rate tuples $\boldsymbol{\lambda}$ for which there exists some virtual resource control policy which makes all VIP queues stable and satisfies the cache size constraint (9), conditional flow balance constraint (33), and link capacity constraint (22) in the virtual network.

In the rest of the paper, "the stability region" always refers to the stability region under conditional flow balance.

C. Equivalence Between the Virtual and Actual Networks

Unlike the virtual network, where each data object k corresponds to an individual VIP queue, each DP queue in the actual network contains DPs of all data objects. Hence, the virtual network is not exactly a "time reversal mirror" (TRM) of the actual network, and it is non-trivial to establish the equivalence between them. This challenge is addressed in the following theorem, which is proved in Appendix A.

Theorem 2: (Equivalence between virtual and actual networks). $\Lambda_c = \Lambda_v$, and for any arrival rate tuple $\boldsymbol{\lambda} \in int\Lambda_v$, we have $\overline{\Gamma}^*_c(\boldsymbol{\lambda}) = \overline{\Gamma}^*_v(\boldsymbol{\lambda})$, where $\overline{\Gamma}^*_c(\boldsymbol{\lambda})$ and $\overline{\Gamma}^*_v(\boldsymbol{\lambda})$ are the minimum cache content placement costs required for stability in the actual and virtual networks respectively, under the arrival rate tuple $\boldsymbol{\lambda}$.

D. Optimality of the Dual-Mode-VIP-Based Resource Control

The throughput optimality of the proposed resource control is summarized below.

Theorem 3: (Throughput optimality for the virtual network). If there exists $\boldsymbol{\epsilon} = \left(\epsilon^k_n = \epsilon\right)_{n\in\mathcal{N},k\in\mathcal{K}} \succ \mathbf{0}$ such that $\boldsymbol{\lambda} + \boldsymbol{\epsilon} \in \Lambda_v$, then the VIP queues and cache content placement cost under the mixed-timescale resource control algorithm in Section IV-B (Algorithms 1 and 2) satisfies

$$\limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=1}^{t} \sum_{m\in\mathcal{M},k\in\mathcal{K}} \mathbb{E}\left[V^k_m(\tau)\right] \le \frac{TB + W\overline{\Gamma}^*_v(\boldsymbol{\lambda})}{\epsilon},$$

$$\limsup_{J\to\infty} \frac{1}{J} \sum_{i=1}^{J} \mathbb{E}\left[\Gamma(i)\right] \le \frac{TB}{W} + \overline{\Gamma}^*_v(\boldsymbol{\lambda}), \tag{34}$$

where B is a constant depending on the maximum endogenous rates $\mu^{out}_{j,max} \triangleq \max_{\boldsymbol{H},\boldsymbol{\mu}^A} \mu^A_j$, s.t. $\boldsymbol{\mu}^A \in C^A(\boldsymbol{H}), \forall j \in \mathcal{U}$, $\mu^{out}_{n,max} \triangleq R_d, \forall n \in \mathcal{B}$, maximum exogenous rates $\mu^{in}_{n,max} \triangleq \max_{\boldsymbol{H},\boldsymbol{\mu}^A} \sum_{j\in\mathcal{U}} \mu^A_j$, s.t. $\boldsymbol{\mu}^A \in C^A(\boldsymbol{H})$ and the maximum arrival rate A_{max} at each user.

Please refer to Appendix B for the proof. Theorem 3 states that the proposed solution is throughput-optimal for the virtual network since for any $\boldsymbol{\lambda} \in \text{int}\Lambda_v$, the average cost can be made arbitrarily close to the minimum cost $\overline{\Gamma}^*_v(\boldsymbol{\lambda})$ with bounded VIP queue lengths by choosing a sufficiently large W. Note that the virtual-to-actual control policy mapping in Section IV-C is designed to satisfy the following property: if the virtual resource control policy satisfies the conditional flow balance constraint (33), the resulting resource control policy also satisfies the conditional flow balance constraint (30) in the actual network. Therefore, Theorem 3 implies that the proposed solution is also throughput-optimal for the actual network.

VI. Characterization of the Stability Region

Since the network stability region is equal to the VIP stability region, we shall focus on the characterization of the VIP stability region, which is easier.

A. VIP DoF Stability Region under Identical User Preference

The VIP stability region for the general case is given in Lemma 3 in Appendix C. To obtain insight for practical design, we focus on studying the *VIP DoF stability region under identical user preference* defined as

$$\mathcal{D}_{ve} \triangleq \lim_{P\to\infty} \left(\Lambda_v \cap \Lambda_e\right)/P, \tag{35}$$

where $\Lambda_e = \left\{\boldsymbol{\lambda} : \lambda^1_j \ge \lambda^2_j \ge \ldots \ge \lambda^K_j, \forall j\right\}$. $\mathcal{D}_{ve}$ captures the VIP stability region when the SNR is high and the popularity orders of the K data objects at all users are identical.

Theorem 4: (VIP DoF stability region). For any arrival rate tuple $\boldsymbol{\lambda} \in int\left(\Lambda_v \cap \Lambda_e\right)$*, the minimum cache content placement cost required for stability is given by* $\overline{\Gamma}_v^*\left(\boldsymbol{\lambda}\right) = 0$*, which is achieved by a fixed cache placement* $s_n^k = 1, \forall k \in \mathcal{K}_p, n$*, and* $s_n^k = 0, \forall k \in \overline{\mathcal{K}}_p, n$*, where* $\mathcal{K}_p = \{1, \ldots, L_C\}$ *and* $\overline{\mathcal{K}}_p = \mathcal{K} \backslash \mathcal{K}_p$*. Moreover, if*

$$\lim_{P\to\infty} C^A\left(\boldsymbol{H}\right)/P = \mathcal{D}^A, \lim_{P\to\infty} C^B\left(\boldsymbol{H}\right)/P = \mathcal{D}^B, \forall \boldsymbol{H}, \tag{36}$$

where $\mathcal{D}^A$ *and* $\mathcal{D}^B$ *are DoF regions under the CoMP and coordinated modes respectively, then* $\mathcal{D}_{ve}$ *consists of all DoF tuples* $\boldsymbol{d} = \left(d_j^k\right)_{k\in\mathcal{K}, j\in\mathcal{U}}$ *such that there exists a set of variables* $\left\{\alpha \in [0,1], \mu_j^{Ak} \geq 0, \mu_j^{Bk} \geq 0, \mu_{ng}^k \geq 0\right\}$ *satisfying*

$$\begin{aligned} d_j^k \leq\ & (1-\alpha)\,\mu_j^{Ak} + \alpha\mu_j^{Bk}, \forall j,k \\ & \times \left(\sum_{k\in\mathcal{K}_p}\left[(1-\alpha)\sum_{j\in\mathcal{U}}\mu_j^{Ak} + \alpha\mu_{j_n}^{Bk}\right] - r_n\right)^+ \\ & + \sum_{k\in\overline{\mathcal{K}}_p}\left[(1-\alpha)\sum_{j\in\mathcal{U}}\mu_j^{Ak} + \alpha\mu_{j_n}^{Bk}\right] \leq R_d, \forall n \\ & \times \left[\sum_{k\in\mathcal{K}}\mu_j^{Ak}\right]_{j\in\mathcal{U}} \in \mathcal{D}^A, \left[\sum_{k\in\mathcal{K}}\mu_j^{Bk}\right]_{j\in\mathcal{U}} \in \mathcal{D}^B. \end{aligned} \tag{37}$$

Please refer to Appendix C for the proof. Note that (36) is a mild conditon because it holds for many channel distributions (such as Rayleigh or Rice fading channels). Moreover, the DoF region is determined by the distribution of $\boldsymbol{H}$ instead of the realization of $\boldsymbol{H}$ [19]. Therefore, $\mathcal{D}^A$ and $\mathcal{D}^B$ are not expressed as a function of $\boldsymbol{H}$.

Remark 2: For the special case in Theorem 4 with stationary popularity and identical user preference, the fixed offline cache placement (caching the most popular L_C data objects) is sufficient to achieve the minimum cache content placement cost $\overline{\Gamma}_v^*\left(\boldsymbol{\lambda}\right) = 0$. This is because, in this special case, each user at each frame sees the same stationary arrival rate process $\left\{A_j^k(t)\right\}$. In order to satisfy the conditional flow balance constraint within each frame, we should cache the most popular L_C data objects to induce CoMP to handle the large arrival rates caused by the more frequent requests of popular data objects, since this will save more backhaul resources to serve the requests of the other data objects. As can be seen from (37), when the cache size L_C is larger, the CoMP probability $1-\alpha$ is larger and more data object requests can be handled by the cache-induced CoMP. Therefore, the DoF stability region increases with the cache size L_C. Note that even for the special case in Theorem 4, the proposed online cache placement still has an advantage in terms of the delay performance. In practice, the popularity varies over time and different users have different preferences. Moreover, the random user requests and wireless fading will also cause microscopic spatial and temporary popularity variations. In this case, the online cache placement can achieve much better delay performance (with the same backhaul capacity R), as will be shown in simulations.

B. Maximum Sum DoF Under Identical User Popularity

We shall derive a closed-form expression of the sum DoF under *identical user popularity*, which is a special case of identical user preference when the arrival rate of any data object is the same for all users. Specifically, the average arrival rates (in terms of DoF) have the form $d_j^k = d\rho_k, \forall j,k$, where ρ_k can be interpreted as the probability of requesting data object k, and $d = \sum_{k\in\mathcal{K}} d_j^k$ is the total average arrival rate of user j (in terms of DoF). In this case, the maximum sum DoF that can be achieved under the stability constraint is

$$D^* = \max_d Kd, \text{ s.t. } \left(d_j^k = d\rho_k\right)_{j\in\mathcal{U}, k\in\mathcal{K}} \in \mathcal{D}_{ve}. \tag{38}$$

Theorem 5: (Maximum sum DoF under Zipf popularity). Consider identical user popularity and suppose (36) is satisfied. When $r_n \geq ND^A$*, the maximum sum DoF* D^* *in (38) is*

$$D^* = \begin{cases} D_A, & R_d \geq R_A^* \\ \left(1-\alpha^*\right)D_A + \alpha^* D_B, & R_d \in \left(R_B^*, R_A^*\right) \\ R_d \Big/ \sum_{k=L_c+1}^{K}\rho_k, & R_d \leq R_B^* \end{cases}, \tag{39}$$

where $D_A = \max_d d,\ s.t.\ , \left(d_j = d\right)_{j\in\mathcal{U}} \in \mathcal{D}^A$, $D_B = \max_d d,\ s.t.\ , \left(d_j = d\right)_{j\in\mathcal{U}} \in \mathcal{D}^B$, $R_A^* = ND_A\sum_{k=L_c+1}^{K}\rho_k$, $R_B^* = D_B\sum_{k=L_c+1}^{K}\rho_k$ *and*

$$\alpha^* = \begin{cases} \hat{\alpha} \triangleq \dfrac{R_A^* - NR_d}{R_A^* - NR_B^*} & \text{if } \dfrac{(1-\hat{\alpha})R_A^*}{N} + \hat{\alpha}R_B^* \leq \hat{\alpha}D_B \\ \dfrac{R_A^* - R_d}{R_A^* - NR_B^* + (N-1)D_B} & \text{otherwise.} \end{cases} \tag{40}$$

Please refer to Appendix D for the proof. The assumption $r_n \geq ND_A$ helps to simplify the expression of the maximum sum DoF. This assumption is usually satisfied in practice since the I/O speed of the storage device is typically much larger than the wireless transmission rate. From Theorem 5, we have the following observations.

1) Impact of Backhaul Capacity: When $R_d \geq R_A^*$, there is enough backhaul capacity to support CoMP transmission with probability 1. In this case, the sum DoF is D_A, which is completely limited by the RAN. When $R_d \in \left(R_B^*, R_A^*\right)$, the backhaul capacity can only support CoMP transmission mode with a non-zero probability less than 1. In this case, the sum DoF is between D_B and D_A, which is limited by both the RAN and backhaul. Moreover, as R_d increases from R_B^* to R_A^*, α^* decreases from 1 to 0, and D^* increases from D_B to D_A. When $R_d \leq R_B^*$, the sum DoF is less than D_B, which is completely limited by the backhaul.

2) Impact of Cache Size L_C*:* For a larger cache size L_C, both R_A^* and R_B^* become smaller, i.e., a smaller backhaul capacity is required to support full CoMP transmission. Moreover, both the CoMP transmission probability α^* and the sum DoF increase with L_C. On the other hand, when L_C is very

small, the backhaul capacity has to be larger than $R_A^* \approx ND_A$ in order to support full CoMP transmission.

3) Impact of Popularity Distribution: As an example, consider the *Zipf popularity distribution* [20], where

$$\rho_k = \frac{k^{-\varsigma}}{\sum_{k=1}^{K} k^{-\varsigma}}, k = 1, \ldots, K, \tag{41}$$

and $\varsigma \geq 0$ is the *popularity skewness parameter*. A larger popularity skewness ς means that the user requests concentrate more on a few popular files. As a result, for a larger ς, both backhaul thresholds R_A^* and R_B^* become smaller. Moreover, both the CoMP transmission probability α^* and the sum DoF increase with ς.

VII. Simulation Results

Consider a cached MIMO interference network with seven BS-user pairs placed in seven wrapped-around hexagonal cells. Each BS is equipped with two antennas, and each user is equipped with one antenna. The backhaul capacity per BS is 30 Mbps except for Fig. 10. The channel bandwidth is 10 MHz, the slot size is 2 ms and the frame size is 0.5 s. The pathloss model between BS n and user j is $PL_{j,n} = 140.7 + 36.7 \log 10\left(d_{j,n}\right)$ [21], where $d_{j,n}$ is the distance between BS n and user j. The channel between BS n and user j is modeled as $\boldsymbol{H}_{jn} = PL_{j,n}\overline{\boldsymbol{H}}_{jn}$, where $\overline{\boldsymbol{H}}_{jn}$ has i.i.d. Gaussian entries of zero mean and unit variance.

Zero-forcing beamforming (ZFBF), which is a special case of linear precoding, is used at the PHY for both CoMP and coordinated transmission modes. In the CoMP mode, all users can be simultaneously served by the BSs, and the corresponding ZFBF precoder is given by $\boldsymbol{V}_j^A = \xi \widetilde{\boldsymbol{H}}^H \left(\widetilde{\boldsymbol{H}}\widetilde{\boldsymbol{H}}^H\right)^{-1}$, where $\widetilde{\boldsymbol{H}} = \left[\widetilde{\boldsymbol{H}}_j\right]_{j=1,\ldots,N}^H \in \mathbb{C}^{N\times 2N}$ is the composite channel matrix between all BSs and all users; and ξ is chosen to satisfy the power constraint. In the coordinated mode, we randomly select a subset of two users $\mathcal{U}^B$ for transmission at each time slot. For given user selection $\mathcal{U}^B$, the corresponding ZFBF precoder is given by $\boldsymbol{V}_j^B = \sqrt{P}\overline{\boldsymbol{V}}_j^B$, where $\overline{\boldsymbol{V}}_j^B \in \mathbb{C}^2$ with $\left\|\overline{\boldsymbol{V}}_j^B\right\| = 1$ is obtained by the projection of $\boldsymbol{H}_{jn_j}$ on the orthogonal complement of the subspace spanned by $\left[\boldsymbol{H}_{j'n_j}\right]_{j'\in\mathcal{U}^B\setminus\{j\}}$.

There are $K = 1000$ data objects in the content server. The data chunk size is 50 KB and the data object size is 1 MB. At each user, object requests arrive according to a Poisson process with a total average arrival rate of λ Mbps. To verify the performance under both spatial and temporary popularity variations, we assume user j only requests a subset $\mathcal{F}_j$ of 100 data objects whose indices are randomly generated. The average arrival rate of data object $\mathcal{F}_j(k) \in \mathcal{F}_j$ at user j is $\lambda_j^{\mathcal{F}_j(k)} = \lambda\rho_k$, where $\mathcal{F}_j(k)$ is the k-th data object in $\mathcal{F}_j$ and ρ_k's follow the Zipf distribution in (41). The following baselines are considered.

Baseline 1 (Offline caching with dual-mode PHY [6], [7]**):** Each BS caches the most popular L_C data objects in an offline manner. Dual-mode PHY is employed at the RAN.

Baseline 2 (LFU with dual-mode PHY): In Least Frequently Used (LFU) caching, the nodes record how often each data object has been requested and choose to cache the new data object if it is more frequently requested than the least frequently requested cached data object (which is replaced). Dual-mode PHY is employed at the RAN.

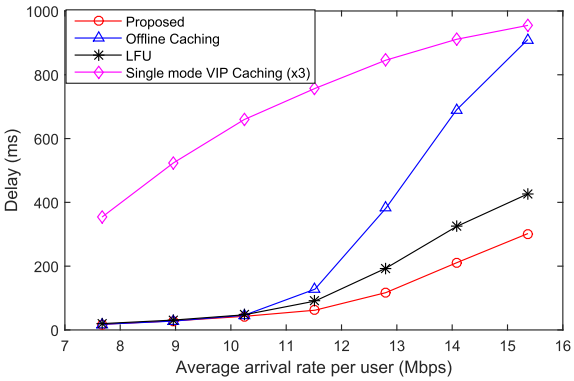


Fig. 5. Delay versus per user average arrival rate with cache size $L_C = 80$ data objects and skewness $\varsigma = 0.5$.

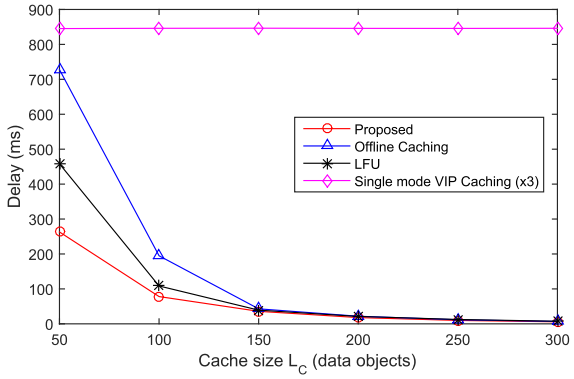


Fig. 6. Delay versus cache size L_C with per user average arrival rate $\lambda = 13.25$ Mbps and skewness $\varsigma = 0.5$.

Baseline 3 (VIP caching with single-mode PHY [9]**):** The cache placement is determined by the VIP framework in [9] and only coordinated mode is considered at the PHY.

For fair comparison, the data sub-channel R_d and control sub-channel R_c are assumed to share the same R Mbps backhaul capacity for all schemes. In Fig. 5 – 7, we plot the delay performance of the schemes versus the average arrival rate of each user λ, the cache size L_C at each BS and the skewness parameter ς respectively. For Baseline 3, the delay shown in the figure is the actual delay divided by 3. The delay for an IP request is the difference between the fulfillment time (i.e., time of arrival of the requested DP) and the creation time of the IP request. It can be seen that the delay of all schemes increases with the average arrival rate λ, and decreases with the cache size L_C and skewness parameter ς. Moreover, the proposed scheme achieves better performance than all baseline schemes.

In Fig. 8 – 9, we plot the cache hit rate versus the cache size L_C and the skewness parameter ς. The cache hit rate of all schemes increases with L_C and ς. In Fig. 10, we plot the delay performance versus the backhaul capacity. It can be seen that the delay of all schemes decreases with the backhaul capacity. Again, the proposed scheme achieves the best cache hit rate

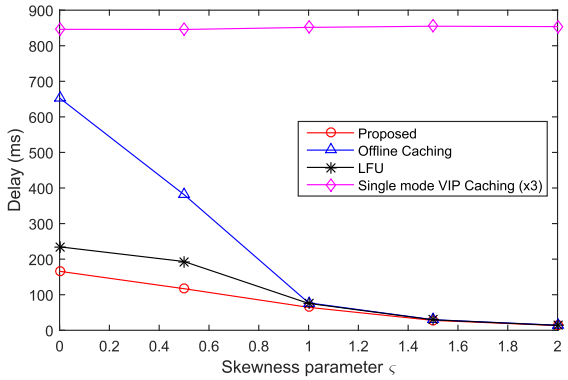


Fig. 7. Delay versus skewness ς with per user average arrival rate $\lambda = 13.25$ Mbps and cache size $L_C = 80$ data objects.

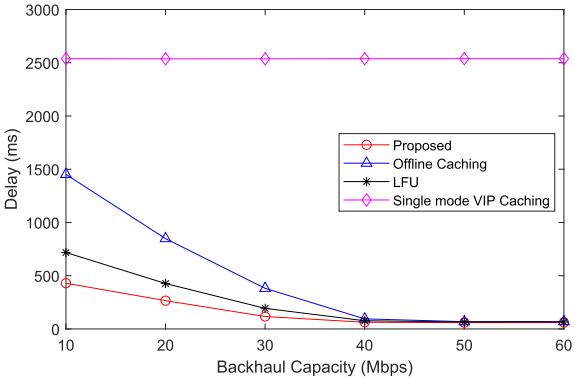


Fig. 10. Delay versus backhaul capacity with per user average arrival rate $\lambda = 13.25$ Mbps, cache size $L_C = 80$ data objects and skewness $\varsigma = 0.5$.

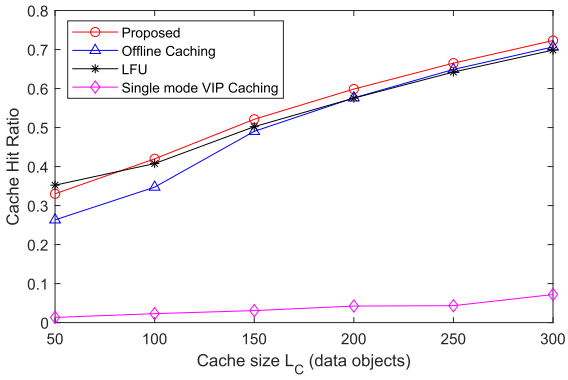


Fig. 8. Cache hit rate versus cache size L_C with per user average arrival rate $\lambda = 13.25$ Mbps and skewness $\varsigma = 0.5$.

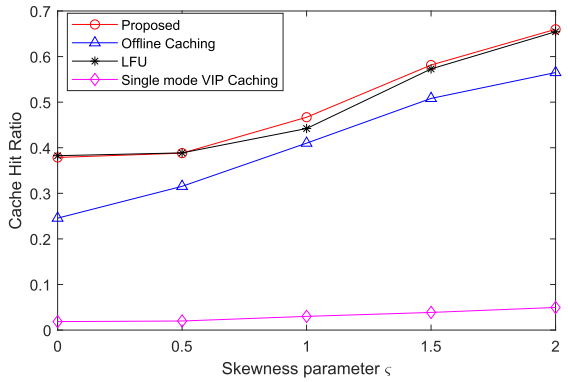


Fig. 9. Cache hit rate versus skewness ς with per user average arrival rate $\lambda = 13.25$ Mbps and cache size $L_C = 80$ data objects.

and delay-backhaul tradeoff performance. Note that the CoMP probability is not necessarily proportional to the average cache hit rate because the CoMP mode transmission requires simultaneous cache hit at all BSs. As a result, the cache hit rate cannot completely reflect the delay performance. Even when the cache hit rate of LFU is not low relative to the proposed scheme, the delay performance of LFU can be much worse than the proposed scheme in some cases.

When the cache size L_C or skewness ς is large, the CoMP probability is high for any caching scheme, and thus the performance gap between different schemes will vanish, except for the single mode VIP caching scheme, which still has a large performance gap w.r.t. the proposed scheme because it cannot enjoy the cache-enabled CoMP gain. However, the proposed scheme has significant gain over all the baseline schemes for practical scenarios when the cache size is limited compared to the total content size and the popularity does not concentrate on a few data objects. The fact that the proposed scheme achieves a better performance than the LFU demonstrates the effectiveness of the proposed dual-mode-VIP-based mixed-timescale resource control algorithm. The LFU can achieve a better performance than the offline caching scheme because it is an online caching scheme which can better adapts the cached content according to the microscopic spatial and temporary popularity variations. Finally, the single mode VIP caching scheme is worse than the other schemes because it cannot exploit the cache-enabled opportunistic CoMP to enhance the capacity of RAN.

VIII. Conclusion

We propose a mixed-timescale online PHY caching and content delivery scheme for wireless NDNs with a dual-mode PHY. The cache content placement is performed once per frame (T time slots) to avoid excessive cache content placement cost. For a given cache state at each frame, the PHY mode selection and rate allocation is performed once per time slot to fully exploit the cached content at the BS. To facilitate efficient resource control design, we introduce a *dual-mode VIP framework*, which transforms the original network into a virtual network, and formulate the resource control design in the virtual network. We establish the throughput optimality of the proposed solution. Moreover, we obtain a closed-form expression for the maximum sum DoF in the stability region under the Zipf popularity distribution. Simulations show that the proposed solution outperforms the existing solutions.

Appendix

A. Proof of Theorem 2

For a given $\boldsymbol{\lambda} \in \text{int}\Lambda_v$, the minimum cost $\overline{\Gamma}_v^*(\boldsymbol{\lambda})$ for VIP stability is given by the solution of the problem in Theorem 3, minimized over the class of all stationary randomized policies defined in Section C. If $\boldsymbol{\lambda} \in \text{int}\Lambda_v$, there exists

a positive value ϵ such that $\boldsymbol{\lambda}+\boldsymbol{\epsilon} \in \text{int}\Lambda_v$. It follows that there exists a stationary random resource control policy $\left\{p_n^k(i), M(t), \mu_j^{Ak}(t), \mu_j^{Bk}(t), \mu_{ng}^k(t)\right\}$ in the virtual network such that the corresponding conditional flow balance constraint is satisfied:

$$\lambda_j^k+\epsilon \le \overline{\mu}_{j|\boldsymbol{s}}^{Ak}+\overline{\mu}_{j|\boldsymbol{s}}^{Bk}, \quad \sum_{j\in\mathcal{U}} \overline{\mu}_{j|\boldsymbol{s}}^{Ak}+\overline{\mu}_{j_n|\boldsymbol{s}}^{Bk} \le \overline{\mu}_{ng|\boldsymbol{s}}^k + r_n s_n^k. \tag{42}$$

By modifying the rate allocation policy to $\mu_j^{Ak}(t)=\mu_j^{Ak}(t)-\frac{0.5\epsilon}{N+1}$ and $\mu_j^{Bk}(t)=\mu_j^{Bk}(t)-\frac{0.5\epsilon}{N+1}$, the resulting control policy satisfies the following conditional flow balance constraint:

$$\lambda_j^k+\frac{N\epsilon}{N+1} \le \overline{\mu}_{j|\boldsymbol{s}}^{Ak}+\overline{\mu}_{j|\boldsymbol{s}}^{Bk}, \quad \sum_{j\in\mathcal{U}} \overline{\mu}_{j|\boldsymbol{s}}^{Ak}+\overline{\mu}_{j_n|\boldsymbol{s}}^{Bk}+\frac{\epsilon}{2} \le \overline{\mu}_{ng|\boldsymbol{s}}^k + r_n s_n^k, \tag{43}$$

and we define $\overline{\Gamma}^*(\epsilon)$ as the minimum average cost consumed by any such stationary policy.

In the following, we construct a random control policy $\left\{p_n^k(i), m_j^k(t), c_{ng}(t), M_a(t), \boldsymbol{c}^A(t), \boldsymbol{c}^B(t)\right\}$ in the actual network such that the following conditional flow balance constraint is satisfied when the arrival rate tuple is $\boldsymbol{\lambda}$:

$$\overline{b}_{n|\boldsymbol{s}}+\frac{K\epsilon}{2} \le \overline{c}_{ng|\boldsymbol{s}}, \overline{b}_{nj_n|\boldsymbol{s}}^B+\frac{KN\epsilon}{2(N+1)} \le \overline{c}_{j_n|\boldsymbol{s}}^B, \quad \overline{b}_{nj|\boldsymbol{s}}^A+\frac{KN\epsilon}{2(N+1)} \le \overline{c}_{j|\boldsymbol{s}}^A. \tag{44}$$

Specifically, the cache placement control policy $\left\{p_n^k(i)\right\}$ in the actual network is the same as that in the virtual network. For a given cache state $\boldsymbol{s}$, the forwarding mode at time slot t in the actual plane is randomly chosen with $\Pr\left[m_j^k(t)=1\right]=\min\left(\left(\overline{\mu}_{j|\boldsymbol{s}}^{Ak}-\frac{N\epsilon}{2(N+1)}\right)/\lambda_j^k, 1\right)$. For the other control actions in the actual plane, we let $c_{ng}(t)=\sum_{k\in\mathcal{K}}\mu_{ng}^k(t)$, $M_a(t)=M(t)$, $\boldsymbol{c}^A(t)=\boldsymbol{\mu}^A(t)$ and $\boldsymbol{c}^B(t)=\boldsymbol{\mu}^B(t)$. Then, it can be verified that (44) is satisfied and the above control policy achieves the same average cost $\overline{\Gamma}^*(\epsilon)$ as that in the virtual network. (44) implies that the average departure rate of each DP queue is strictly larger than the average arrival rate, and hence the network is stable [11]. Therefore, we have proved that $\Lambda_v \subseteq \Lambda_c$. Moreover, following a similar argument to Footnote 3 in [11], it can be shown that $\overline{\Gamma}^*(\epsilon) \to \overline{\Gamma}_v^*(\boldsymbol{\lambda})$ as $\epsilon \to 0$, so that stability in the actual network can be attained with an average cost that is arbitrarily close to $\overline{\Gamma}_v^*(\boldsymbol{\lambda})$.

Similarly, it can be shown that for any $\boldsymbol{\lambda} \in \text{int}\Lambda_c$, the VIP stability in the virtual network can be attained with average cost that is arbitrarily close to $\overline{\Gamma}_c^*(\boldsymbol{\lambda})$. This implies that $\Lambda_c \subseteq \Lambda_v$. Therefore, we have $\Lambda_c=\Lambda_v$ and $\overline{\Gamma}_c^*(\boldsymbol{\lambda})=\overline{\Gamma}_v^*(\boldsymbol{\lambda})$ for any arrival rate tuple $\boldsymbol{\lambda} \in \text{int}\Lambda_v$.

B. Proof of Theorem 3

Let Ω_R^* denote the optimal random policy (the optimal solution of (50)) in Lemma 3. Let $\Delta_T^{\text{R}}(i)$ denote the T-step drift-plus-penalty under Ω_R^*.We first obtain an upper bound of $\Delta_T^{\text{R}}(i)$.

Lemma 1: $\Delta_T^R(i) \le \widetilde{\Delta}_T^R(i) = T^2B_0 - T\epsilon\sum_{j\in N,k\in\mathcal{K}} V_j^k(t_0) + TW\overline{\Gamma}_v^*(\boldsymbol{\lambda})$, *where* B_0 *is a constant depending on* $\left\{\mu_{j,max}^{out}, \mu_{n,max}^{out}, \mu_{n,max}^{in}\right\}$ *and* A_{max}.

Proof: Following a similar analysis to the proof of Theorem 3 in [16], it can be shown that

$$\begin{aligned}
\Delta_T^{\text{R}}(i) &\le \widetilde{\Delta}_T^{\text{R}}(i)=T^2B_0+\Delta_{T1}^{\text{R}}(i)+\Delta_{T2}^{\text{R}}(i)+TW\overline{\Gamma}_v^*(\boldsymbol{\lambda}),\\
\widetilde{\Delta}_{T1}^{\text{R}}(i) &= 2T\sum_{j\in\mathcal{U},k\in\mathcal{K}} V_j^k(t_0^i)\mathbb{E}\left[\lambda_n^k-\frac{1}{T}\sum_{\tau=t_0}^{t_0+T-1}\left(M^*(\tau)\sum_{j\in\mathcal{U}}\mu_j^{Ak*}(\tau)+\overline{M}^*(\tau)\sum_{j\in\mathcal{U}}\mu_j^{Bk*}(\tau)\right)|\boldsymbol{s}(i-1)\right],\\
\widetilde{\Delta}_{T2}^{\text{R}}(i) &= -2T\sum_{j\in\mathcal{B},k\in\mathcal{K}} V_j^k(t_0)\\
&\quad\times\mathbb{E}\left[\frac{1}{T}\sum_{\tau=t_0}^{t_0+T-1}\mu_{jg}^k(\tau)^*+r_n\left[s_n^k(t_0^i-1)+p_n^k(t_0)^*\right]\right.\\
&\quad-\frac{1}{T}\sum_{\tau=t_0}^{t_0+T-1}M^*(\tau)\sum_{j\in\mathcal{U}}\mu_j^{Ak*}(\tau)\\
&\quad\left.-\frac{1}{T}\sum_{\tau=t_0}^{t_0+T-1}\overline{M}^*(\tau)\sum_{j\in\mathcal{U}}\mu_j^{Bk*}(\tau)\,|\boldsymbol{s}(i-1)\right],
\end{aligned}$$

where the control actions with superscript * are given by the optimal random policy Ω_R^*. Since for any given $\boldsymbol{s}(i)$, Ω_R^* satisfies the conditional flow balance (33), we have $\widetilde{\Delta}_{T1}^{\text{R}}(i) \le -2T\sum_{j\in\mathcal{U},k\in\mathcal{K}} V_j^k(t_0^i)\epsilon$ and $\widetilde{\Delta}_{T2}^{\text{R}}(i) \le -2T\sum_{j\in\mathcal{B},k\in\mathcal{K}} V_j^k(t_0^i)\epsilon$, from which Lemma 1 follows. ■

To obtain an upper bound of the drift for the proposed solution, we need to consider a FRAME policy which serves as a bridge to connect the proposed solution and the optimal random policy Ω_R^*. In the FRAME policy, the cache content placement is the same as the proposed solution, while the mode selection and rate allocation is the optimal solution of a modified version of the drift minimization problem in (21), with the current VIP length $\left\{V_j^k(t), V_n^k(t)\right\}$ replaced by the outdated VIP length $\left\{V_j^k(t_0^i), V_n^k(t_0^i)\right\}$. Let $\widetilde{\Delta}_T^{\text{P}}(i)$ and $\widetilde{\Delta}_T^{\text{F}}(i)$ denote the upper bound of the T-step drift-plus-penalty defined in Theorem 1 under the proposed solution and the FRAME policy respectively. The following lemma states the relationship between the different policies.

Lemma 2: $\widetilde{\Delta}_T^P(i)-\widetilde{\Delta}_T^F(i) \le T^2B_1$, *where* B_1 *is a constant depending on* $\mu_{n,max}^{out}$, $\mu_{n,max}^{in}$ *and* A_{max}. *Moreover,* $\widetilde{\Delta}_T^F(i) \le \widetilde{\Delta}_T^R(i)$.

Proof: Similar to the proof of Lemma 5 in [16], the result $\widetilde{\Delta}_T^{\text{P}}(i)-\widetilde{\Delta}_T^{\text{F}}(i) \le T^2B_1$ follows from the fact that the expected magnitude of change in a single VIP queue is at most $(\mu_{j,\max}^{out}+A_{\max})$, $\forall j\in\mathcal{U}$ and $(\mu_{n,\max}^{out}+\mu_{n,\max}^{in}+r_n)$, $\forall n\in\mathcal{B}$, and the caching-related term in $\widetilde{\Delta}_T^{\text{P}}(i)$ and $\widetilde{\Delta}_T^{\text{F}}(i)$ is identical. The detailed proof is omitted for conciseness. On the other hand, the result $\widetilde{\Delta}_T^{\text{F}}(i) \le \widetilde{\Delta}_T^{\text{R}}(i)$ follows from the fact that the FRAME policy minimizes $\Delta_T(i)$. ■

From Lemma 1 and 2, we conclude that

$$\begin{aligned}\Delta_T^{\mathrm{P}}(i) &\le \widetilde{\Delta}_T^{\mathrm{P}}(i) \le \widetilde{\Delta}_T^{\mathrm{F}}(i) + T^2 B_1 \\ &\le T^2 B - T\epsilon \sum_{j\in N, k\in\mathcal{K}} V_j^k(t_0) + TW\overline{\Gamma}_v^*(\boldsymbol{\lambda}), \quad (45)\end{aligned}$$

where $B \triangleq B_0 + B_1$. Finally, Theorem 3 can be proved from (45) by applying Theorem 4.2 of [12] and the same technique as in Theorem 3 of [16].

C. Proof of Theorem 4

Consider a *stationary random policy* Ω_R as follows. At the beginning of the i-th frame, $\left\{p_n^k(i)\right\}$ is randomly chosen from the set of all feasible cache placement control actions $\mathcal{A}_p^{\boldsymbol{s}'}$ (i.e., satisfying the cache size constraint (9)) with probability $\boldsymbol{\alpha}_p^{\boldsymbol{s}'} = \left[\alpha_{p1}^{\boldsymbol{s}'}, \ldots, \alpha_{p\left|\mathcal{A}_p^{\boldsymbol{s}'}\right|}^{\boldsymbol{s}'}\right]$, where $\boldsymbol{s}' = \boldsymbol{s}(i-1)$. Here, we use the superscript $\boldsymbol{s}'$ to indicate that $\mathcal{A}_p^{\boldsymbol{s}'}$ and $\boldsymbol{\alpha}_p^{\boldsymbol{s}'}$ depend on $\boldsymbol{s}'$. At each time slot t, $M(t)$ is randomly chosen with $\Pr[M(t)=0] = \alpha_M^{(\boldsymbol{s},\boldsymbol{H})}$ and $\Pr[M(t)=1] = 1-\alpha_M^{(\boldsymbol{s},\boldsymbol{H})}$, where $\boldsymbol{s} = \boldsymbol{s}(i)$ and $\boldsymbol{H} = \boldsymbol{H}(t)$; $\mu_j^{Ak}(t)$ is randomly chosen from $N+1$ rate tuples $\left\{\mu_{ji}^{Ak(\boldsymbol{s},\boldsymbol{H})}, i = 1, \ldots, N+1\right\}$ satisfying

$$\left[\sum_{k\in\mathcal{K}} \mu_{ji}^{Ak(\boldsymbol{s},\boldsymbol{H})}\right]_{j\in\mathcal{U}} \in C^A(\boldsymbol{H}), i = 1, \ldots, N+1, \quad (46)$$

with probability $\boldsymbol{\alpha}_A^{(\boldsymbol{s},\boldsymbol{H})} = \left[\alpha_{A1}^{(\boldsymbol{s},\boldsymbol{H})}, \ldots, \alpha_{A(N+1)}^{(\boldsymbol{s},\boldsymbol{H})}\right]$; $\mu_j^{Bk}(t)$ is randomly chosen from $N+1$ rate tuples $\left\{\mu_{ji}^{Bk(\boldsymbol{s},\boldsymbol{H})}, i = 1, \ldots, N+1\right\}$ satisfying

$$\left[\sum_{k\in\mathcal{K}} \mu_{ji}^{Bk(\boldsymbol{s},\boldsymbol{H})}\right]_{j\in\mathcal{U}} \in C^B(\boldsymbol{H}), i = 1, \ldots, N+1, \quad (47)$$

with probability $\boldsymbol{\alpha}_B^{(\boldsymbol{s},\boldsymbol{H})} = \left[\alpha_{B1}^{(\boldsymbol{s},\boldsymbol{H})}, \ldots, \alpha_{B(N+1)}^{(\boldsymbol{s},\boldsymbol{H})}\right]$; and $\mu_{ng}^k(t) = \mu_{ng}^{k(\boldsymbol{s},\boldsymbol{H})}$ satisfying

$$\sum_{k\in\mathcal{K}} \mu_{ng}^{k(\boldsymbol{s},\boldsymbol{H})} \le R_d. \quad (48)$$

Moreover, the above random policy $\Omega_R = \left\{\mathcal{A}_p^{\boldsymbol{s}'}, \boldsymbol{\alpha}_p^{\boldsymbol{s}'}, \alpha_M^{(\boldsymbol{s},\boldsymbol{H})}, \mu_{ji}^{Ak(\boldsymbol{s},\boldsymbol{H})}, \boldsymbol{\alpha}_A^{(\boldsymbol{s},\boldsymbol{H})}, \mu_{ji}^{Bk(\boldsymbol{s},\boldsymbol{H})}, \boldsymbol{\alpha}_B^{(\boldsymbol{s},\boldsymbol{H})}, \mu_{ng}^{k(\boldsymbol{s},\boldsymbol{H})}\right\}$ is called a stationary random policy if the resulting Markov cache state process $\boldsymbol{s}(i)$ is stationary. Let $P_{\boldsymbol{s}',\boldsymbol{s}}^{\Omega_R}$ denote the transition probability of the controlled Markov Process $\boldsymbol{s}(i)$ from state $\boldsymbol{s}'$ to state $\boldsymbol{s}$, and $\pi_{\boldsymbol{s}}^{\Omega_R}$ denote the steady state probability of $\boldsymbol{s}(i) = \boldsymbol{s}$, under the stationary random policy Ω_R. Then we have the following lemma.

Lemma 3: (VIP stability region). The VIP stability region Λ_v consists of all arrival rate tuples $\boldsymbol{\lambda}$ such that there exists a stationary random policy Ω_R satisfying (46), (47), (48) and

$$\begin{aligned}&\lambda_j^k \le \mathbb{E}_{\boldsymbol{H}}\Bigg[\left(1-\alpha_M^{(\boldsymbol{s},\boldsymbol{H})}\right)\sum_{i=1}^{N+1}\alpha_{Ai}^{(\boldsymbol{s},\boldsymbol{H})}\mu_{ji}^{Ak(\boldsymbol{s},\boldsymbol{H})} \\ &\qquad +\alpha_M^{(\boldsymbol{s},\boldsymbol{H})}\sum_{i=1}^{N+1}\alpha_{Bi}^{(\boldsymbol{s},\boldsymbol{H})}\mu_{ji}^{Bk(\boldsymbol{s},\boldsymbol{H})}\Bigg], \\ &\mathbb{E}_{\boldsymbol{H}}\Bigg[\left(1-\alpha_M^{(\boldsymbol{s},\boldsymbol{H})}\right)\sum_{j\in\mathcal{U}}\sum_{i=1}^{N+1}\alpha_{Ai}^{(\boldsymbol{s},\boldsymbol{H})}\mu_{ji}^{Ak(\boldsymbol{s},\boldsymbol{H})} \\ &\qquad +\alpha_M^{(\boldsymbol{s},\boldsymbol{H})}\sum_{i=1}^{N+1}\alpha_{Bi}^{(\boldsymbol{s},\boldsymbol{H})}\mu_{j_n i}^{Bk(\boldsymbol{s},\boldsymbol{H})}\Bigg] \le \mathbb{E}_{\boldsymbol{H}}\left[\mu_{ng}^{k(\boldsymbol{s},\boldsymbol{H})}\right] + r_n s_n^k, \quad (49)\end{aligned}$$

$\forall j \in \mathcal{U}, k \in \mathcal{K}, n \in \mathcal{B}$ and $\forall \boldsymbol{s} \in \mathcal{S}$. Moreover, for any $\boldsymbol{\lambda} \in int\Lambda_v$, the minimum cache content placement cost required for stability $\overline{\Gamma}_v^*(\boldsymbol{\lambda})$ is given by

$$\begin{aligned}\overline{\Gamma}_v^*(\boldsymbol{\lambda}) = \min_{\Omega_R} &\sum_{\left(\boldsymbol{s}',\boldsymbol{s}\right)\in\mathcal{S}_T} \pi_{\boldsymbol{s}'}^{\Omega_R} P_{\boldsymbol{s}',\boldsymbol{s}}^{\Omega_R} \sum_{n\in\mathcal{B},k\in\mathcal{K}} \mathbf{1}_{\left\{s_n^k - s_n'^k = 1\right\}} \quad (50) \\ &\text{s.t. (46), (47), (48) and (49) are satisfied} \\ &\forall j \in \mathcal{U}, k \in \mathcal{K}, n \in \mathcal{B}, \boldsymbol{s} \in \mathcal{S}.\end{aligned}$$

The proof follows from arguments similar to those in [11], [12] and please refer to [17] for the detailed proof. Finally, Theorem 4 follows from Lemma 3 and the definition of $\mathcal{D}_{ve}$.

D. Proof of Theorem 5

It follows from the symmetry property of the problem that $\mu_j^{Ak} = \mu^{Ak}, \forall j$ and $\mu_j^{Bk} = \mu^{Bk}, \forall j$ at the optimal solution of (38). As a result, when $r_n \ge ND^A$, Problem (38) is equivalent to the following problem:

$$\begin{aligned}&\max_{d,\alpha,\mu^{Ak},\mu^{Bk}} Kd, \\ &\text{s.t. } d\rho_k = (1-\alpha)\mu^{Ak} + \alpha\mu^{Bk}, \forall k \\ &\qquad N(1-\alpha)\sum_{k=L_c+1}^{K}\mu^{Ak} + \alpha\sum_{k=L_c+1}^{K}\mu^{Bk} \le R_d, \\ &\qquad \sum_{k\in\mathcal{K}}\mu^{Ak} \le D_A, \sum_{k\in\mathcal{K}}\mu^{Bk} \le D_B.\end{aligned}$$

By finding the optimal solution of the above problem, we can obtain the maximum sum DoF as in Theorem 5.

REFERENCES

[1] J. Li, W. Chen, M. Xiao, F. Shu, and X. Liu, "Efficient video pricing and caching in heterogeneous networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 8744–8751, Oct. 2016.

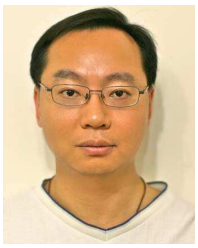
[2] J. Li, J. Sun, Y. Qian, F. Shu, M. Xiao, and W. Xiang, "A commercial video-caching system for small-cell cellular networks using game theory," *IEEE Access*, vol. 4, pp. 7519–7531, 2016.

[3] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless video content delivery through distributed caching helpers," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 1107–1115.

[4] J. Dai, F. Liu, B. Li, B. Li, and J. Liu, "Collaborative caching in wireless video streaming through resource auctions," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 2, pp. 458–466, Feb. 2012.

[5] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of distributed caching in D2D wireless networks," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Sep. 2013, pp. 1–5.
[6] A. Liu and V. K. 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.
[7] A. Liu and V. K. N. Lau "Cache-enabled opportunistic cooperative MIMO for video streaming in wireless systems," *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 390–402, Jan. 2014.
[8] N. Abedini and S. Shakkottai, "Content caching and scheduling in wireless networks with elastic and inelastic traffic," *IEEE/ACM Trans. Netw.*, vol. 22, no. 3, pp. 864–874, Jun. 2014.
[9] E. Yeh, T. Ho, Y. Cui, M. Burd, R. Liu, and D. Leong, "VIP: A framework for joint dynamic forwarding and caching in named data networks," in *Proc. 1st ACM Conf. Inf.-Centric Netw.*, Sep. 2014, pp. 117–126.
[10] M. M. Amble, P. Parag, S. Shakkottai, and L. Ying, "Content-aware caching and traffic management in content distribution networks," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 2858–2866.
[11] M. J. Neely, "Energy optimal control for time-varying wireless networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, Jul. 2006.
[12] M. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lect. Commun. Netw.*, vol. 3, no. 1, pp. 1–211, Sep. 2010.
[13] A. Liu, V. Lau, W. Ding, and E. Yeh, "Mixed timescale online PHY caching and content delivery for content-centric wireless networks," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.
[14] D. Gesbert, S. Hanly, H. Huang, S. S. Shitz, O. Simeone, and W. Yu, "Multi-cell MIMO cooperative networks: A new look at interference," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 9, pp. 1380–1408, Dec. 2010.
[15] C. Cox, *An Introduction to LTE: LTE, LTE-Advanced, SAE and 4G Mobile Communications*. Hoboken, NJ, USA: Wiley, 2012.
[16] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time-varying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.
[17] A. Liu, V. Lau, W. Ding, and E. Yeh. (2018). "Mixed-timescale online PHY caching for dual-mode MIMO cooperative networks." [Online]. Available: https://arxiv.org/abs/1810.07503
[18] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4331–4340, Sep. 2011.
[19] R. Tandon, S. Mohajer, H. V. Poor, and S. Shamai (Shitz), "Degrees of freedom region of the MIMO interference channel with output feedback and delayed CSIT," *IEEE Trans. Inf. Theory*, vol. 59, no. 3, pp. 1444–1457, Mar. 2013.
[20] T. Yamakami, "A zipf-like distribution of popularity and hits in the mobile web pages with short life time," in *Proc. 7th Int. Conf. Parallel Distrib. Comput. Appl. Technol. (PDCAT'06)* Taipei, Taiwan, Dec. 2006, pp. 240–243.
[21] *Tech. Specification Group Radio Access Network; Further Advancements for E-UTRA Physical Layer Aspects*, document 3GPP TR 36.814. [Online]. Available: http://www.3gpp.org

An Liu (S'07–M'09–SM'17) received the Ph.D. and B.S. degrees in electrical engineering from Peking University, China, in 2011 and 2004, respectively. From 2008 to 2010, he was a Visiting Scholar with the Department of Electrical Computer and Energy Engineering, University of Colorado at Boulder. He was a Post-Doctoral Research Fellow and a Research Assistant Professor with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, from 2011 to 2013 and from 2015 to 2017, respectively, where he has been a Visiting Assistant Professor since 2014. He is currently a Distinguished Research Fellow with the College of Information Science and Electronic Engineering, Zhejiang University. His research interests include wireless communications, stochastic optimization, and compressive sensing.



Vincent K. N. Lau (SM'04–F'12) received the B.Eng. (Hons.) from The University of Hong Kong in 1992 and the Ph.D. degree from Cambridge University in 1997. He was with Bell Labs., from 1997 to 2004. He joined the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology (HKUST), in 2004, where he is currently a Chair Professor and the Founding Director of Huawei, HKUST Joint Innovation Lab. His current research interests include robust and delay-optimal cross layer optimization for MIMO/OFDM wireless systems, interference mitigation techniques for wireless networks, massive MIMO, and M2M and network control systems.

Wenchao Ding received the B.S. degree in electronic and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2015. He is currently pursuing the Ph.D. degree with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong. His current research interests include wireless communication, robotics, and autonomous navigation.

Edmund Yeh received the B.S. degree (Hons.) in electrical engineering from Stanford University in 1994, the M.Phil. degree in engineering from Cambridge University in 1995, and the Ph.D. degree in electrical engineering and computer science from MIT, under the supervision of Prof. R. Gallager in 2001. He was an Assistant and Associate Professor of electrical engineering, computer science, and statistics with Yale University. He is currently a Professor of electrical and computer engineering with Northeastern University, Boston, MA, USA. He was a recipient of the Winston Churchill Scholarship, the Alexander von Humboldt Research Fellowship, the Army Research Office Young Investigator Award, and the Best Paper Award from the 2015 IEEE International Conference on Communications Communication Theory Symposium and the 2017 ACM Conference on Information-Centric Networking.