WiFlix: Adaptive Video Streaming in Massive MU-MIMO Wireless Networks

D. Bethanabhotla, G. Caire and M. J. Neely

***Abstract*—We consider the problem of simultaneous on-demand streaming of stored video to multiple users in a multi-cell wireless network where multiple unicast streaming sessions are run in parallel and share the same frequency band. Each streaming session is formed by the sequential transmission of video "chunks", such that each chunk arrives into the corresponding user playback buffer within its playback deadline. We formulate the problem as a Network Utility Maximization (NUM) where the objective is to fairly maximize users' video streaming Quality of Experience (QoE) and then derive an iterative control policy using Lyapunov Optimization, which solves the NUM problem up to any level of accuracy and yields an *online protocol* with control actions at every iteration decomposing into two *layers* interconnected by the users' *request queues* : i) a video streaming adaptation layer reminiscent of Dynamic Adaptive Streaming over HTTP (DASH), implemented at each user node; ii) a transmission scheduling layer where a max-weight scheduler is implemented at each base station. The proposed chunk request scheme is a *pull* strategy where every user opportunistically requests video chunks from the neighboring base stations and dynamically adapts the quality of its requests based on the current size of the request queue. For the transmission scheduling component, we first describe the general max-weight scheduler and then particularize it to a wireless network where the base stations have Multiuser Multiple-Input Multiple-Output (MU-MIMO) beamforming capabilities. We exploit the *channel hardening* effect of large-dimensional MIMO channels (massive MIMO) and devise a low complexity user selection scheme to solve the underlying combinatorial problem of selecting user subsets for downlink beamforming, which can be easily implemented and run independently at each base station. Further, through simulations, we show that deploying MU-MIMO significantly improves video streaming performance and also that the proposed cross-layer approach is able to serve users more fairly than a baseline scheme representative of current systems running independently designed protocol layers.**

***Index Terms*—Adaptive Video Streaming, DASH, Massive MIMO, Scheduling, Network Utility Maximization, Lyapunov Optimization.**

I. Introduction

Demand for video content over wireless networks has grown dramatically in recent years and it is predicted to account for 75% of the total mobile data traffic by 2019 [1]. This is mainly due to on-demand video streaming, enabled by multimedia devices such as tablets and smartphones. In addition, recent measurement studies [2] reveal that, in 2013, around 26.9% of video streaming sessions on the Internet experienced playback interruption due to re-buffering, 43.3% were impacted by low resolution, and 4.8% failed to start altogether. At the application layer, *Dynamic Adaptive Streaming over HTTP* (DASH) [3], [4][1] has become a de-facto industry standard approach to handle video streaming over wireless networks. In DASH, each user (client) monitors the available capacity during a video streaming session and chooses adaptively and dynamically the most appropriate video quality level correspondingly. The video files are divided into "chunks", which are downloaded by sequential HTTP requests. Different quality levels can be obtained either by storing multiple versions of the same video encoded at different bit-rates, or by using scalable video coding and sending an adaptive number of refinement layers [5].

In this way, DASH attempts to maintain a reasonable quality of experience (QoE) even under changing network conditions. However, operating at the application layer only is not sufficient to achieve a fully satisfactory performance. For instance, popular video platforms such as Youtube and Netflix, which employ DASH at the application layer, have realized this fact and recently released Video Quality Reports [6], [7] where they compare and contrast different network service providers (ISP) in a given geographical area and rank/label them as either Lower Definition (LD) or Standard Definition (SD) or High Definition (HD) based on the quality of video streaming activity in their network over a certain time frame in order to inform users that the choice of ISP can affect video streaming QoE.

A. Motivation and related work

In order to cope with this problem, a *cross layer* optimization approach has been proposed in several works (e.g., see [8]–[14]). In these works, the video streaming QoE is defined in terms of performance metrics such as video quality, probability of stall events (i.e., when the playback buffer is empty and video playback stops), pre-buffering time, and re-buffering time. However, the joint optimization of these metrics by directly controlling the dynamics of the playback buffers of all the users in the network requires solving a Markov Decision Problem (MDP) which is typically quite difficult and incurs the well-known curse of dimensionality. For instance, [9] considers adaptive video transmission in a much simpler setting of a point-to-point wireless link and formulates the problem as an

All authors are with the Ming Hsieh Department of Electrical Engineering, Viterbi School of Engineering, University of Southern California, Los Angeles CA 90089. Email: bethanab, caire, mjneely@usc.edu

This work was partially supported by NSF Grant CCF-1423140 and NSF Grant CCF-0747525.

[1]This includes industry products such as Microsoft Smooth Streaming and Apple HTTP Live Streaming, which qualitatively work in the way assumed in our paper, up to minor variations which are irrelevant for the present theoretical treatment.

MDP which is then solved using the value iteration algorithm. However, even in such a simple point-to-point scenario, the value iteration policy requires extensive computation to be done offline and stored in a lookup table which is then used for the actual transmission. On the other hand, the work [13] takes a cross-layer approach and considers video delivery in the general case of a multiuser wireless network where users are served by wireless *helper nodes*.[2] In order to obtain a tractable formulation for the multiuser network, [13] adopts a "divide and conquer" approach where first the problem of maximizing a function of the time-averaged video qualities, subject to queue stability is solved, and then the delay jitter is taken care of by appropriately dimensioning the pre-buffering and re-buffering times, exploiting the fact that the playback buffer can absorb the delay fluctuations around the (bounded) mean. However, in [13] a "push" scheduling policy is considered, for which video chunks can be served out of order and may result in data loss in the presence of intermittent connectivity and/or mobility. In this paper, we fix this problem and introduce a new "pull" strategy, that is robust to fast topology variations. Our scheme allows each user to opportunistically pull data always in the correct sequential order from neighboring helper nodes. This results in smoother and more reliable performance. Another shortcoming of [13] is that it considers only helpers operating according to OFDM (Orthogonal Frequency Division Multiplexing) /TDMA (Time Division Multiple Access), i.e., serving at most one user per transmission resource (referred to as *physical layer (PHY) frame* hereafter). As a matter of fact, the current wireless technology trend is rapidly evolving towards multiuser MIMO (MU-MIMO) schemes (e.g., see [15]–[18]) where multiple users can be served on the same PHY frame by spatial multiplexing. The current work therefore allows for general wireless channel models, including MU-MIMO as a special case.

B. Contributions

Motivated by the above considerations, this paper proposes *WiFlix*, a system for efficient delivery of video content over a wireless network formed by a number of densely deployed wireless helper nodes serving multiple wireless users over a given geographic coverage area and on the same shared channel bandwidth. *WiFlix* addresses the problem of dynamic adaptive video streaming in a wireless network by *jointly optimizing* the video quality adaptation at the DASH layer (application layer) and the transmission scheduling of users at the PHY/MAC layer. This is obtained through a cross layer approach where the appropriate queue sizes maintained at the users act as a bridge between the layers. In particular, the novel contributions of this paper are as follows:

1) We introduce the notion of a *request queue*. This is a virtual queue, maintained by each user, that serves to sequentially request video chunks from helper nodes, such that the choice of the helper node and the quality at which each video chunk is requested can be adaptively adjusted. Each user, upon deciding the quality of the chunk, *requests* the bits corresponding to that chunk and places them in the request queue. Note that this does not mean the user has already downloaded the chunk, but the chunk bits are "virtually" placed in the request queue and will be taken out when the chunk is effectively delivered to the user. In this way, the user maintains in the request queue all the chunk bits that have been requested but not downloaded and adaptively adjusts the quality of future chunk requests based on its size. In addition, the user broadcasts this size to the helpers in its current vicinity and "pulls" bits from them in the right order necessary for video playback starting at the *Head Of Line* (HOL) of the request queue. Even if a mobile user gets out of range of a helper while downloading the HOL bits, it can still re-request those bits from the new helper in its current vicinity. In this way, the user always downloads chunks in the playback order and does not skip any of them. This improves significantly upon the "push" scheme proposed in [13] where the chunks could be downloaded out of order due to different transmission queue delays at different helper nodes, or skipped if a user moves out of a helper's coverage after placing a request.

2) We systematically obtain our cross-layer policy as the dynamic solution of a Network Utility Maximization (NUM) problem, where the network utility function is given in terms of the users' time-averaged video quality, and the maximization constraints are given by imposing stability of each request queue. The stability constraint implies that every requested chunk will be eventually delivered, while delivery in the right sequential order is guaranteed by the request queue mechanism described above. The proposed policy decomposes naturally into two interconnected *layers*: i) a video streaming adaptation layer reminiscent of DASH, implemented at each user node, and involving the adaptive video quality selection and placement of the video chunk requests into the request queue; ii) a transmission scheduling layer where a max-weight scheduler is implemented at the helpers. These two layers are interconnected by the users' *request queues*, which form the weights for the max-weight scheduler. Although queue stability guarantees that all requested chunks are eventually delivered, such delivery may still occur, occasionally, after the corresponding playback deadline. In this case, we are in the presence of a stall event. In order to control the stall event probability and make it sufficiently small, we follow the same divide and conquer approach of [13], and adaptively set the pre-buffering/re-buffering time by monitoring the chunk delivery delay in a sliding window. This approach has the advantage of yielding very good performances also in terms of stall event probability, while allowing for the elegant and mathematically tractable NUM framework in terms of the video quality maximization.

3) We particularize the max-weight transmission policy to a network of helpers with MU-MIMO capabilities, where the scheduling actions consist of choosing the subset of users for MU-MIMO beamforming at each helper. By exploiting the "channel hardening" effect of large-dimensional MIMO channels (massive MIMO) [19]–[21], we reduce the combi-

[2]Our treatment applies, at a very high level, to any infrastructure-based wireless network such as conventional cellular, small cells, WLAN, and heterogeneous compositions thereof (e.g., a cellular network with wifi offload). Therefore, throughout this paper, we refer to infrastructure nodes simply as "helpers".

natorial weighted sum rate maximization over the multiuser multicell network (which would involve an exponentially complex exhaustive user selection, or some polynomial complexity heuristic greedy user selection at each helper) to a simple subset selection problem which is optimally solved by a low complexity algorithm. The algorithm can be implemented independently at the MAC layer of each helper. The only information that needs to be exchanged between the layers is the length of the users' request queues, which can be easily gathered as "protocol information" via the uplink, together with the chunk requests.

4) We show through simulation in a realistic network topology and using actual encoded video data that the proposed system is very effective in improving the average video quality and reducing the percentage of time spent in buffering mode.

II. System Model

We consider a wireless network with multiple users and multiple helper stations sharing the same bandwidth. The network is defined by a bipartite graph $\mathcal{G} = (\mathcal{U}, \mathcal{H}, \mathcal{E})$, where $\mathcal{U}$ denotes the set of users, $\mathcal{H}$ denotes the set of helpers, and $\mathcal{E}$ contains edges for all pairs (h, u) such that helper h can transmit information to user u. We denote by $\mathcal{N}(u) \subseteq \mathcal{H}$ the neighborhood of user u, i.e., $\mathcal{N}(u) = \{h \in \mathcal{H} : (h, u) \in \mathcal{E}\}$. Similarly, $\mathcal{N}(h) = \{u \in \mathcal{U} : (h, u) \in \mathcal{E}\}$. Each user $u \in \mathcal{U}$ requests a video file f_u which is formed by a sequence of chunks. Each chunk corresponds to a group of pictures (GOP) that are encoded and decoded as stand-alone units [5]. Chunks have a fixed playback duration, given by $T_{\text{gop}} = (\#\text{ frames per GOP})/\eta$, where η is the frame rate, expressed in frames per second. The streaming process consists of transferring chunks from the helpers to the requesting users such that the playback buffer at each user contains the required chunks at the beginning of each chunk playback deadline. The playback starts after a short pre-buffering time, during which the playback buffer is filled by a determined amount of ordered chunks. The details related to pre-buffering and chunk playback deadlines are discussed in Section VI.

Each file f is encoded at a finite number of different quality/compression levels $m \in \{1, \ldots, N_f\}$ [4]. Due to the variable bit rate (VBR) nature of video coding [22], the quality-rate profile of a given file f may vary from chunk to chunk. In particular, we let $\tilde{D}_f(m, i)$ and $\tilde{B}_f(m, i)$ denote the video quality measure (e.g., see [23]) and the size (in number of bits) of the i-th chunk in file f at quality level m, respectively.

A. Time-scales

It is important to note that the time scale at which chunks are requested and the time scale at which PHY layer transmissions are scheduled differ by $1-3$ orders of magnitude. For instance, in current video streaming technology [3], the typical video chunk spans a duration of $0.5-2$ seconds while the duration of a *PHY frame* is of the order of milliseconds (for example, with a PHY frame duration of 10 ms (as in the LTE 4G standard [24]) and assuming $T_{\text{gop}} = 0.5$s, a video chunk spans $n = \frac{0.5}{10 \cdot 10^{-3}} = 50$ PHY frames). In the following, we consider dynamic scheduling policies that operate at the PHY frame time scale, i.e., they provide a scheduling/resource allocation decision at each PHY frame time $t \in \mathbb{Z}$. However, new chunks are requested at multiples of the chunk time, i.e., at times $t = in$ for $i \in \mathbb{Z}$ and n denoting the number of PHY frames per chunk time, assumed here to be an integer for simplicity. In the rest of the paper, we will use consistently the following notation: index t denotes the PHY frame *transmission slots*, and the index i denotes *video chunks*.

B. Request Queue Dynamics

At the beginning of the i-th chunk time, each user $u \in \mathcal{U}$ requests a particular quality mode for the i-th chunk of its video stream. That is, on each slot $t \in \{0, n, 2n, 3n, \ldots\}$, each user $u \in \mathcal{U}$ specifies the quality mode $m_u(t) \in \{1, 2, \ldots, N_{f_u}\}$ for its next video chunk. This decision specifies the quality $D_{f_u}(m_u(t), t)$ and the amount of bits $B_{f_u}(m_u(t), t)$ associated with the chunk requested at slot t. As these decisions are made only at times t that are multiples of n, it is convenient to define:

$$D_{f_u}(m_u(t), t) := 0 \text{ and } B_{f_u}(m_u(t), t) := 0 \quad \text{for } t \notin \{0, n, 2n, \ldots\}. \tag{1}$$

On the other hand, at time slots that are integer multiples of n, i.e., $t = in$, we have:

$$D_{f_u}(m_u(t), t) = \tilde{D}_{f_u}(m_u(in), i) \text{ and } B_{f_u}(m_u(t), t) = \tilde{B}_{f_u}(m_u(in), i). \tag{2}$$

The bits $B_{f_u}(m_u(t), t)$ are called the *requested bits* of user u at slot t, and are placed in a *request queue* $Q_u(t)$. The request queue evolves over the transmission slots $t \in \{0, 1, 2, \ldots\}$ as:

$$Q_u(t+1) = \max\{Q_u(t) - \mu_u(t) + B_{f_u}(m_u(t), t), 0\} \quad \forall\ u \in \mathcal{U}, \tag{3}$$

where $\mu_u(t)$ is the amount of bits downloaded by user u on slot t. Note that the request queue in (3) can decrease every transmission slot t as new bits are downloaded, but can only increase on slots $t = in$, i.e., on integer multiples of n. Intuitively, $Q_u(t)$ consists of bits associated with all chunks that have been requested by user u but not yet fully received.

The quantity $\mu_u(t)$ indicates the instantaneous aggregate downloading rate of user u on slot t, expressed in bits per slot. This is given by

$$\mu_u(t) = \sum_{h \in \mathcal{N}(u)} \mu_{hu}(t) 1_{hu}(t) \tag{4}$$

where $1_{hu}(t)$ is an indicator function, equal to 1 if helper h has the video file requested by user u and zero otherwise, and $\mu_{hu}(t)$ is the rate served by helper h to user u on slot t. The matrix $[\mu_{hu}(t)]$ of transmission rates is selected within a set of feasible transmission rate matrices for slot t. The set of all rate matrices supported by the network at a given slot time t is referred to as the *feasible instantaneous rate region* at time t, and depends on the network topology and channel state (e.g., on the fading channels realization). Specifically,

let $\omega(t)$ represent the topology state on slot t, being a vector of parameters that affect transmission, such as current device locations and/or channel conditions. Assume $\omega(t)$ takes values in an abstract set Ω, possibly being an infinite set. For each $\omega \in \Omega$, define $\mathcal{R}(\omega)$ to be the feasible rate region of the network for state ω. Then, the feasible instantaneous rate region is $\mathcal{R}(\omega(t))$. For example, the set $\mathcal{R}(\omega)$ may include the constraint that each user can receive a positive rate from at most one helper and/or constrain helpers to restrict transmissions to at most S users, where S denotes the maximum number of downlink data streams (spatial multiplexing gain) that the helper station can handle (see [25] for a discussion of various wireless multiple access scenarios and interference models that fit this general framework). The set $\mathcal{R}(\omega)$ can also handle models that allow simultaneous download from multiple helpers (for instance, in a cellular CDMA system with macro diversity), or information-theoretic capacity regions of various network topology models, inclusive of broadcast and interference constraints (e.g., [26]). We also mention here that this framework can also handle non-wireless scenarios. For example, it can constrain $[\mu_{hu}(t)]$ to be permutation matrices associated with packet switch constraints. However, as explained in Section I, it is desirable for current and future systems to take advantage of massive MU-MIMO capabilities at the helpers. Section V specifies $\mathcal{R}(\omega)$ for the relevant wireless scenario with helpers employing massive MU-MIMO, which is the primary focus of this paper. The simulation results in Section VII are carried out under the specific wireless model defined in Section V.

Remark 1: *Each user u maintains $Q_u(t)$ and updates it according to (3) every transmission slot t. A small amount of bookkeeping is also required by the user to associate the bits $Q_u(t)$ with their appropriate chunks. Specifically, each user maintains a list of chunks it has requested but not yet fully received, along with the quality modes it requested for each chunk. It can receive new bits on slot t only from a helper that has its requested file, and only if $Q_u(t) > 0$. In order to download these bits, the user fetches them (or "pulls" them) from the selected helper. In practice this can be implemented by an HTTP request pointing at a specific block of data corresponding to the desired quality level of the current video chunk, resulting in a DASH-like approach.*

III. Problem Formulation and Streaming Policy

When optimizing the users' video QoE we have to take into account that users compete for the same shared transmission resource (the network wireless spectrum and the helpers spatial downlink data streams) and, given the fact that the users are placed in arbitrary positions with respect to the helpers, their attainable service rates may be quite different. Hence, some fairness criterion must be enforced. In addition, we need to carefully define the notion of QoE, since the adaptive nature of the streaming process involves a possibly time-varying quality level across the streaming sessions.

As already mentioned briefly before, we remark once again that, in order to obtain a tractable formulation, we adopt the *divide and conquer approach* pioneered in [13]:

1) We first formulate the NUM problem (5), where the network utility function is a concave and component wise non-decreasing function of the time averaged users' *requested* video quality and the maximization is subject to the stability of all the request queues in the system.

2) We then solve the NUM problem using the Lyapunov Optimization framework and obtain the *drift-plus-penalty* (DPP) policy which adapts to arbitrarily changing network conditions and in fact is optimal (with respect to the NUM problem) under non-stationary and non-ergodic evolution of the underlying network state process.

3) Since all the request queues in the system are ensured to be stable, the *requested* video chunks are eventually *delivered*. However, in order to ensure that all the video chunks are delivered within their playback deadline, it suffices for every user to choose a *pre-buffering* time which exceeds the largest delay with which a chunk is delivered. In particular, when the maximum delay of each request queue in the system admits a deterministic upper bound, setting the pre-buffering time larger than such a bound makes the playback buffer under rate zero. However, for a system with arbitrary (non-stationary, non-ergodic) evolution of the underlying network state process (for e.g., arbitrary user mobility and arbitrary per-chunk fluctuations of video coding rate due to VBR coding), such deterministic upper bounds on the maximum delay may not exist or are too loose to be useful in practice. Hence, in Section VI, we propose a method to locally estimate the delays with which video chunks are delivered, such that each user can calculate its pre-buffering and re-buffering times to be larger than the locally estimated maximum delay. Through simulations in Section VII, we demonstrate the effectiveness of the combination of the DPP policy and the adaptive pre-buffering scheme.

In the rest of this section, we focus on the NUM problem formulation and its solution through the DPP approach. Throughout this work, we use the following notation for the time average quantity of interest: we let $\overline{D}_u := \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left[D_{f_u}\left(m_u(\tau), \tau\right)\right]$ denote the time average of the expected quality of user u, and $\overline{Q}_u := \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\left[Q_u\left(\tau\right)\right]$ to be the time average of the expected length of the request queue at user u, assuming that these limits exist. More in general, we use the overline notation to indicate limiting time-averages.[3] Let $\phi_u(\cdot)$ be a concave, continuous, and non-decreasing function defining network utility vs. video quality for user $u \in \mathcal{U}$. The NUM problem that we wish to solve is given by:

$$\text{maximize} \quad \sum_{u\in\mathcal{U}} \phi_u(\overline{D}_u) \tag{5a}$$

$$\text{subject to} \quad \overline{Q}_u < \infty \ \forall\ u \in \mathcal{U} \tag{5b}$$

$$[\mu_{hu}(t)] \in \mathcal{R}(\omega(t)) \ \forall\ t \tag{5c}$$

$$m_u(t) \in \{1, 2, \ldots, N_{f_u}\} \ \forall\ u \in \mathcal{U},\ \forall\ t, \tag{5d}$$

where requirement of finite $\overline{Q}_u$ corresponds to the *strong stability* condition for all the queues [27].

[3]The existence of these limits is assumed temporarily for ease of exposition of the optimization problem (5) but is not required for the derivation of the scheduling policy and for the proof of Theorem 1.

By appropriately choosing the functions $\phi_u(\cdot)$, we can impose some desired notion of fairness. For example, a general class of concave functions suitable for this purpose is given by the α-fairness network utility, defined by [28]

$$\phi_u(x) = \begin{cases} \log x & \alpha = 1 \\ \frac{x^{1-\alpha}}{1-\alpha} & \alpha > 0, \ \ \alpha \neq 1 \end{cases} \tag{6}$$

In this case, it is well-known that $\alpha = 0$ yields the maximization of the sum quality (no fairness), $\alpha \to \infty$ yields the maximization of the worst-case quality (max-min fairness) and $\alpha = 1$ yields the maximization of the geometric mean quality (proportional fairness).

In order to solve problem (5) using the stochastic optimization theory developed in [27], it is convenient to transform it into an equivalent problem that involves the maximization of a single time average. This transformation is achieved through the use of auxiliary variables $\gamma_u(t)$ and the corresponding virtual queues $\Theta_u(t)$ with buffer evolution:

$$\Theta_u(t+1) = \max\{\Theta_u(t) + \gamma_u(t) - D_{f_u}(m_u(t), t), 0\}. \tag{7}$$

Consider the transformed problem:

$$\begin{aligned} \text{maximize} \quad & \sum_{u \in \mathcal{U}} \overline{\phi_u(\gamma_u)} && \text{(8a)} \\ \text{subject to} \quad & \overline{Q}_u < \infty \ \forall \ u \in \mathcal{U} && \text{(8b)} \\ & \overline{\gamma}_u \leq \overline{D}_u \ \forall \ u \ \in \ \mathcal{U} && \text{(8c)} \\ & D_u^{\min} \leq \gamma_u(t) \leq D_u^{\max} \ \forall \ u \ \in \ \mathcal{U}, \ \forall \ t && \text{(8d)} \\ & [\mu_{hu}(t)] \in \mathcal{R}(\omega(t)) \ \forall \ t && \text{(8e)} \\ & m_u(t) \in \{1, 2, \ldots, N_{f_u}\} \ \forall \ u \in \mathcal{U}, \ \forall \ t, && \text{(8f)} \end{aligned}$$

where $D_u^{\min}$ and $D_u^{\max}$ are uniform lower and upper bounds on the quality function $D_{f_u}(\cdot, t)$. Notice that constraints (8c) correspond to stability of the virtual queues Θ_u, since $\overline{\gamma}_u$ and $\overline{D}_u$ are the time-averaged arrival rate and the time-averaged service rate for the virtual queue given in (7). We have:

Lemma 1: Problems (5) and (8) are equivalent.

Proof: The proof is well-known (see [13], [27] for instance) and is omitted due to space constraints. ∎

A. The Drift-Plus-Penalty Expression

Let $\mathbf{Q}(t)$ denote the column vector containing the backlogs of queues $Q_u(t) \ \forall \ u \in \mathcal{U}$, let $\mathbf{\Theta}(t)$ denote the column vector for the virtual queues $\Theta_u(t) \ \forall \ u \in \mathcal{U}$, $\boldsymbol{\gamma}(t)$ denote the column vector with elements $\gamma_u(t) \ \forall \ u \in \mathcal{U}$, $\mathbf{B}(t)$ denote the column vector with elements $B_{f_u}(m_u(t), t) \ \forall \ u \ \in \ \mathcal{U}$, $\mathbf{D}(t)$ denote the column vector with elements $D_{f_u}(m_u(t), t) \ \forall \ u \in \mathcal{U}$ and $\boldsymbol{\mu}(t)$ denote the column vector with elements $\mu_u(t) \ \forall \ u \in \mathcal{U}$ as defined in (4). Let $\mathbf{G}(t) = \left[\mathbf{Q}^{\mathsf{T}}(t), \mathbf{\Theta}^{\mathsf{T}}(t)\right]^{\mathsf{T}}$ be the composite vector of queue backlogs and define the quadratic Lyapunov function $L(\mathbf{G}(t)) = \frac{1}{2}\mathbf{G}^{\mathsf{T}}(t)\mathbf{G}(t)$. Intuitively, taking actions to push $L(\mathbf{G}(t))$ down tends to maintain stability of all queues. Define $\Delta(\mathbf{G}(t))$ as the one-slot drift of the Lyapunov function at slot t :

$$\Delta(\mathbf{G}(t)) \triangleq L(\mathbf{G}(t+1)) - L(\mathbf{G}(t)) \tag{9}$$

The DPP algorithm is designed to observe the queues, the current $B_{f_u}(\cdot, t)$, $D_{f_u}(\cdot, t)$ for all users u and $\omega(t)$ on each slot t, and to then choose quality mode $m_u(t)$ for all users u, matrix of transmitted bits $(\mu_{hu}(t)) \in \mathcal{R}(\omega(t))$ and $\gamma_u(t)$ subject to $D_u^{\min} \leq \gamma_u(t) \leq D_u^{\max}$ to minimize a bound on the following *drift-plus-penalty expression*:

$$\Delta(\mathbf{G}(t)) - V \sum_{u \in \mathcal{U}} \phi_u(\gamma_u(t)) \tag{10}$$

where V is a non-negative weight that affects a performance bound. Intuitively, the value of V affects the extent to which the control actions on slot t emphasize utility maximization in comparison to drift minimization.

Lemma 2: Under any control algorithm, the drift-plus-penalty expression satisfies:

$$\begin{aligned} &\Delta(\mathbf{G}(t)) - V \sum_{u \in \mathcal{U}} \phi_u(\gamma_u(t)) \\ &\quad \leq \mathcal{K} - V \sum_{u \in \mathcal{U}} \phi_u(\gamma_u(t)) + (\mathbf{B}(t) - \boldsymbol{\mu}(t))^{\mathsf{T}} \mathbf{Q}(t) \\ &\qquad + (\boldsymbol{\gamma}(t) - \mathbf{D}(t))^{\mathsf{T}} \mathbf{\Theta}(t). \end{aligned} \tag{11}$$

where $\mathcal{K}$ is a uniform upper bound on the term

$$\begin{aligned} \frac{1}{2} \Big[&(\mathbf{B}(t) - \boldsymbol{\mu}(t))^{\mathsf{T}} (\mathbf{B}(t) - \boldsymbol{\mu}(t)) \\ &+ (\boldsymbol{\gamma}(t) - \mathbf{D}(t))^{\mathsf{T}} (\boldsymbol{\gamma}(t) - \mathbf{D}(t)) \Big] \end{aligned} \tag{12}$$

under the realistic assumption that the chunk sizes, the transmission rates and the video quality are bounded.

Proof: Expanding the quadratic Lyapunov function, we have

$$\begin{aligned} &L(\mathbf{G}(t+1)) - L(\mathbf{G}(t)) \\ &= \frac{1}{2} \left(\mathbf{Q}^{\mathsf{T}}(t+1)\mathbf{Q}(t+1) - \mathbf{Q}^{\mathsf{T}}(t)\mathbf{Q}(t) \right) \\ &\quad + \frac{1}{2} \left(\mathbf{\Theta}^{\mathsf{T}}(t+1)\mathbf{\Theta}(t+1) - \mathbf{\Theta}^{\mathsf{T}}(t)\mathbf{\Theta}(t) \right) \\ &= \frac{1}{2} \Big[(\max\{\mathbf{Q}(t) - \boldsymbol{\mu}(t) + \mathbf{B}(t), \mathbf{0}\})^{\mathsf{T}} (\max\{\mathbf{Q}(t) - \boldsymbol{\mu}(t) \\ &\qquad + \mathbf{B}(t), \mathbf{0}\}) - \mathbf{Q}^{\mathsf{T}}(t)\mathbf{Q}(t) \Big] \\ &\quad + \frac{1}{2} \Big[(\max\{\mathbf{\Theta}(t) + \boldsymbol{\gamma}(t) - \mathbf{D}(t), \mathbf{0}\})^{\mathsf{T}} (\max\{\mathbf{\Theta}(t) \\ &\qquad + \boldsymbol{\gamma}(t) - \mathbf{D}(t), \mathbf{0}\}) - \mathbf{\Theta}^{\mathsf{T}}(t)\mathbf{\Theta}(t) \Big], \end{aligned} \tag{13}$$

where we have used the queue evolution equations (3) and (7) and "max" is applied componentwise.

Using the fact that for any non-negative scalar quantities Θ, γ and D we have the inequalities

$$\begin{aligned} (\max\{\Theta + \gamma - D, 0\})^2 &\leq (\Theta + \gamma - D)^2 \\ &= \Theta^2 + (\gamma - D)^2 + 2\Theta(\gamma - D), \end{aligned} \tag{14}$$

we have

$$\begin{aligned} &L(\mathbf{G}(t+1)) - L(\mathbf{G}(t)) \\ &\leq \frac{1}{2} (\mathbf{B}(t) - \boldsymbol{\mu}(t))^{\mathsf{T}} (\mathbf{B}(t) - \boldsymbol{\mu}(t)) + (\mathbf{B}(t) - \boldsymbol{\mu}(t))^{\mathsf{T}} \mathbf{Q}(t) \\ &\quad + \frac{1}{2} (\boldsymbol{\gamma}(t) - \mathbf{D}(t))^{\mathsf{T}} (\boldsymbol{\gamma}(t) - \mathbf{D}(t)) + (\boldsymbol{\gamma}(t) - \mathbf{D}(t))^{\mathsf{T}} \mathbf{\Theta}(t) \end{aligned} \tag{15}$$

Under the realistic assumption that the chunk sizes, the transmission rates and the video quality measures are bounded above by some constants, independent of t, the term

$$\frac{1}{2}\Big[(\mathbf{B}(t)-\boldsymbol{\mu}(t))^{\mathsf{T}}(\mathbf{B}(t)-\boldsymbol{\mu}(t)) + (\boldsymbol{\gamma}(t)-\mathbf{D}(t))^{\mathsf{T}}(\boldsymbol{\gamma}(t)-\mathbf{D}(t))\Big] \quad (16)$$

is bounded above by a constant $\mathcal{K}$. Using this fact and adding the penalty term $-V\sum_{u\in\mathcal{U}}\phi_u(\gamma_u(t))$ on both sides of the inequality (15) yields the result. ∎

The DPP policy described below acquires information about the queue states $\mathbf{G}(t)$, the rate-quality profile $(B_{f_u}(\cdot,t), D_{f_u}(\cdot,t))$ for all users u and the channel state $\omega(t)$ at every slot t, and chooses control actions $m_u(t)$, $[\mu_{hu}(t)] \in \mathcal{R}(\omega(t))$ and $\gamma_u(t)$, subject to $D_u^{\min} \le \gamma_u(t) \le D_u^{\max}$, in order to minimize the last three terms on the right hand side of the inequality (11).

The non-constant part in the right hand side of (11) can be re-written as:

$$\left[\mathbf{B}^{\mathsf{T}}(t)\mathbf{Q}(t)-\mathbf{D}^{\mathsf{T}}(t)\boldsymbol{\Theta}(t)\right] - \left[V\sum_{u\in\mathcal{U}}\phi_u(\gamma_u(t)) - \boldsymbol{\gamma}^{\mathsf{T}}(t)\boldsymbol{\Theta}(t)\right] - \boldsymbol{\mu}^{\mathsf{T}}(t)\mathbf{Q}(t). \quad (17)$$

The resulting control actions are given by the minimization, at transmission slot t, of the expression in (17). Notice that the first term of (17) depends only on $m_u(t)\ \forall\ u\in\mathcal{U}$, the second term of (17) depends only on $\boldsymbol{\gamma}(t)$ and the third term of (17) depends only on $\boldsymbol{\mu}(t)$. Thus, the overall minimization decomposes into three separate sub-problems, yielding the layered scheme given below.

B. The Drift-Plus-Penalty Policy

We address the minimization of (17) focusing separately on its (separable) components.

1) Control actions at the user nodes (pull congestion control): The first term in (17) is given by

$$\sum_{u\in\mathcal{U}}\left\{Q_u(t)B_{f_u}(m_u(t),t)-\Theta_u(t)D_{f_u}(m_u(t),t)\right\}. \quad (18)$$

The minimization variables $m_u(t)$ appear in separate terms of the sum and hence can be optimized separately over each user $u\in\mathcal{U}$. Thus, each user observes the queues $Q_u(t), \Theta_u(t)$ and is aware of the the rate-quality profile $(B_{f_u}(\cdot,t), D_{f_u}(\cdot,t))$ on slot t (video metadata), so that it can choose the quality level of the requested chunk at every video chunk slot i, i.e., at transmission slots $t\in\{in : i\in\mathbb{Z}\}$ as:

$$m_u(t) = \text{argmin}\left\{Q_u(t)B_{f_u}(m,t)-\Theta_u(t)D_{f_u}(m,t) : m\in\{1,\dots,N_{f_u}\}\right\}. \quad (19)$$

As defined in (1), for all transmission slots t which are not integer multiples of n, there is no chunk requested and therefore $B_{f_u}(m_u(t),t)$ and $D_{f_u}(m_u(t),t)$ are equal to 0. The second term in (17), after a change of sign, is given by

$$\sum_{u\in\mathcal{U}}\left\{V\phi_u(\gamma_u(t))-\gamma_u(t)\Theta_u(t)\right\}. \quad (20)$$

Again, this is maximized by maximizing separately each term, yielding the simple one-dimensional maximization (e.g., solvable by line-search):

$$\gamma_u(t) = \text{argmax}\left\{V\phi_u(\gamma)-\Theta_u(t)\gamma\ :\ \gamma\in[D_u^{\min}, D_u^{\max}]\right\}, \quad (21)$$

We refer to the policy (19) and (21) as *pull congestion control* since each user u selects the quality level at which this chunk is requested by taking into account the state of its request queue Q_u. It chooses an appropriate video quality level that balances the desire for high quality (reflected by the term $-\Theta_u(t)D_{f_u}(m,t)$ in (19)) and the desire for low request queue lengths (reflected by the term $Q_u(t)B_{f_u}(m,t)$ in (19)) and then opportunistically pulls the chunk at that video quality level from the helpers in its current vicinity. This policy is reminiscent of the current DASH technology [5], where the client (user) progressively fetches a video file by downloading successive chunks, and makes adaptive decisions on the source encoding quality based on its current knowledge of the congestion of the underlying server-client connection. Notice also that, in order to compute (19) and (21), each user needs to know only *local information* formed by the locally maintained request queue backlog $Q_u(t)$ and by the locally computed virtual queue backlog $\Theta_u(t)$.

2) Control actions at the helper nodes (transmission scheduling): At transmission slot t, the network controller observes the queues $Q_u(t)$ of all users u and the topology state $\omega(t)$, and chooses the feasible instantaneous rate matrix $[\mu_{hu}(t)]\in\mathcal{R}(\omega(t))$ to maximize the weighted sum rate of the transmission rates achievable in transmission slot t. Namely, the network of helpers must solve the Max-Weighted Sum Rate (MWSR) problem:

$$\text{maximize}\ \sum_{h\in\mathcal{H}}\sum_{u\in\mathcal{N}(h)}Q_u(t)\mu_{hu}(t) \quad \text{subject to}\ [\mu_{hu}(t)]\in\mathcal{R}(\omega(t)) \quad (22)$$

where $\mathcal{R}(\omega(t))$ is the feasible instantaneous rate region of the network at slot t. It is immediate to see that, after a change of sign, the maximization of the third term in (17) yields the problem (22).

Remark 2: *In summary, the implementation of the DPP policy decomposes into the* pull congestion control *decisions at the* users *and the transmission scheduling decisions at the* helpers*. The* pull congestion control *decisions (19) and (21) are implemented independently at each user u using only the local knowledge of the queue lengths $Q_u(t)$ and $\Theta_u(t)$. Thus, each user u can implement (19) and (21) without having to learn such information over-the-air from other nodes in the network. On the other hand, in order to implement the* general MWSR transmission scheduling *decision (22), the network of helpers need to know the request queues $Q_u(t)$ of the users and the feasible instantaneous rate region of the network $\mathcal{R}(\omega(t))$. Under certain system assumptions, the solution to the general MWSR problem lends itself to a simple distributed implementation where each helper h makes its own scheduling decisions using the knowledge of the $Q_u(t)$'s for the subset of neighboring users $u\in\mathcal{N}(h)$. This information can be learnt over-the-air at the cost of a small protocol overhead.*

Notice that such overhead is nothing more than a rate priority request in the form of a recursively computed rate weight, as currently implemented in Proportional Fairness scheduling [29]. Therefore, it is expected that the implementation of the proposed DPP is not significantly more complicated than current DASH on top of standard PHY resource allocation schemes. When particularizing the general DPP policy to the massive MIMO multicell network of Section V, we will see that the proposed Wiflix falls in this fortunate class of systems (see also Remark 4).

IV. Policy Performance

As outlined in Section II, VBR video yields time-varying quality and rate functions $D_f(m,t)$ and $B_f(m,t)$, which depend on the individual video file. Furthermore, arbitrary user motion yields slower time variations of the pathloss coefficients at the same time-scale of the video streaming session. As a result, any stationarity or ergodicity assumption about the topology state $\omega(t)$, the rate function $B_f(m,t)$ and quality function $D_f(m,t)$ is unlikely to hold in most practically relevant settings. Therefore, we consider the optimality of the DPP policy for an *arbitrary sample path* of the topology state $\omega(t)$, the quality function $D_f(m,t)$ and the rate function $B_f(m,t)$. Following in the footsteps of [27], [30], we compare the network utility achieved by our DPP policy with that achieved by an optimal oracle policy with T-slot lookahead, i.e., knowledge of the future sample path over an interval of length T slots. Time is split into frames of duration T slots and we consider F such frames. For an arbitrary sample path of $\omega(t)$, $D_f(m,t)$ and $B_f(m,t)$, we consider the static optimization problem over the j-th frame

$$\text{maximize } \sum_{u\in\mathcal{U}} \phi_u \left(\frac{1}{T} \sum_{\tau=jT}^{(j+1)T-1} D_{f_u}(m_u(\tau),\tau) \right) \quad (23)$$

$$\text{subject to } \frac{1}{T} \sum_{\tau=jT}^{(j+1)T-1} \left[B_{f_u}(m_u(\tau),\tau) - \mu_u(\tau) \right] \le 0 \quad \forall\, u \in \mathcal{U}, \quad (24)$$

$$[\mu_{hu}(\tau)] \in \mathcal{R}(\omega(\tau)) \quad \forall\, \tau \in \{jT,\ldots,(j+1)T-1\}, \quad (25)$$

$$m_u(\tau) \in \{1,2,\ldots,N_{f_u}\}\ \forall\, u \in \mathcal{U}, \quad \forall\, \tau \in \{jT,\ldots,(j+1)T-1\}, \quad (26)$$

and denote by ϕ_j^{opt} the maximum of the network utility function for frame j, achieved over all policies which have future knowledge of the sample path over the j-th frame subject to the constraints (24)-(26). We have the following result:

Theorem 1: The DPP scheduling policy achieves per-sample path network utility

$$\sum_{u\in\mathcal{U}} \phi_u\left(\overline{D}_u\right) \ge \lim_{F\to\infty} \frac{1}{F} \sum_{j=0}^{F-1} \phi_j^{\text{opt}} - O\left(\frac{1}{V}\right) \quad (27)$$

with bounded queue backlogs satisfying

$$\lim_{F\to\infty} \frac{1}{FT} \sum_{\tau=0}^{FT-1} \left(\sum_{u\in\mathcal{U}} Q_u(\tau) + \sum_{u\in\mathcal{U}} \Theta_u(\tau) \right) \le O(V) \quad (28)$$

where $O(1/V)$ indicates a term that vanishes as $1/V$ and $O(V)$ indicates a term that grows linearly with V, as the policy control parameter V grows large.

Proof: See Appendix A. ∎

An immediate corollary of Theorem 1 is:

Corollary 1: For the system defined in Section II, when the evolution of the topology state $\omega(t)$, the rate function $B_f(m,t)$ and the quality function $D_f(m,t)$ is stationary and ergodic, then

$$\sum_{u\in\mathcal{U}} \phi_u(\overline{D}_u) \ge \phi^{\text{opt}} - O\left(\frac{1}{V}\right), \quad (29)$$

where ϕ^{opt} is the optimal value of the NUM problem (5) in the stationary ergodic case, and

$$\sum_{u\in\mathcal{U}} \overline{Q}_u + \sum_{u\in\mathcal{U}} \overline{\Theta}_u \le O(V). \quad (30)$$

In particular, if the network state is i.i.d., the bounding term in (29) is explicitly given by $O(1/V) = \frac{\mathcal{K}}{V}$, and the bounding term in (30) is explicitly given by $\frac{\mathcal{K}+V(\phi_{\max}-\phi_{\min})}{\epsilon}$, where $\phi_{\min} = \sum_{u\in\mathcal{U}} \phi_u(D_u^{\min})$, $\phi_{\max} = \sum_{u\in\mathcal{U}} \phi_u(D_u^{\max})$, $\epsilon > 0$ is the slack variable corresponding to the constraint (24), and the constant $\mathcal{K}$ is defined in (11).

Proof: See Appendix A. ∎

V. Wireless System Model with Massive MU-MIMO Helpers

In this section, we first specify the region of instantaneous service rates $\mathcal{R}(\omega(t))$ for the specific PHY layer model comprising of massive MU-MIMO at each helper. We then specialize the weighted sum-rate maximization problem (22) to this system. By exploiting the channel-hardening effect of high dimensional MIMO channels, we observe that the MWSR problem is optimally solved by a low complexity greedy algorithm which can be implemented in a distributed manner with each helper independently choosing user subsets for MU-MIMO beamforming.

A. Helpers with Massive MU-MIMO

For simplicity of exposition, we present the system under the assumptions that the user fading channel vectors are formed by i.i.d. (across time-frequency slots) zero-mean unit-variance complex circularly symmetric coefficients, which are perfectly known at the helper transmitters. In the spirit of massive MIMO [19], we consider per-base station MU-MIMO precoding, i.e., we do not allow base station cooperation. Therefore, the inter-cell interference at each user receiver is treated as additive noise. Extensions and alternative (more realistic) PHY models are briefly discussed in Remark 3 at the end of this section. Each helper h, equipped with a large number of antennas M, implements MU-MIMO to serve the users $\mathcal{N}(h)$ in its vicinity. As a result, helper h can serve

simultaneously, in the spatial domain, any subset of size not larger than $\min\{M, |\mathcal{N}(h)|\}$ of the users in $\mathcal{N}(h)$. We further assume that each helper performs Linear Zero-Forcing Beamforming (LZFBF) to the set of selected users (referred to in the following as "active users").

The wireless channel is modeled by the well-known and widely accepted OFDM block-fading model, where at each transmission slot t, the channel corresponding to the helper-user link (h,u) in $\mathcal{E}$, on the OFDM subcarrier $\nu = 1,\ldots,\mathcal{V}$ is given by

$$\begin{aligned} y_u(t;\nu) = &\sqrt{g_{hu}(t)}\boldsymbol{\xi}_{hu}^{\mathsf{H}}(t;\nu)\mathbf{V}_h(t;\nu)\mathbf{x}_h(t;\nu) \\ &+ \sum_{h'\neq h}\sqrt{g_{h'u}(t)}\boldsymbol{\xi}_{h'u}^{\mathsf{H}}(t;\nu)\mathbf{V}_{h'}(t;\nu)\mathbf{x}_{h'}(t;\nu) \\ &+ z_u(t;\nu), \end{aligned} \tag{31}$$

where $\boldsymbol{\xi}_{hu}(t;\nu)$ is the $M\times 1$ column vector of channel coefficients from the antenna array of helper h to the receiving antenna of user u, $g_{hu}(t)$ is the large-scale distance dependent pathloss from helper h to user u (independent of the subcarrier index ν since the pahtloss is frequency-flat), $\mathbf{V}_h(t;\nu)$ is the downlink precoding matrix of helper h, and $\mathbf{x}_h(t;\nu)$ is the corresponding vector of transmitted complex symbols. Also, $z_u(t;\nu)$ denotes the additive Gaussian noise at the u-th user receiver. Notice that (31) takes fully into account the inter-cell interference of the signals sent by other helpers $h'\neq h$, on the link from helper h to user u.

We use $\mathcal{S}_h(t)$ to denote the subset of users scheduled for transmission by helper h in slot t. Collecting the channel vectors $\boldsymbol{\xi}_{u,h}(t;\nu)$ for $u\in\mathcal{S}_h(t)$ as the columns of a $M\times|\mathcal{S}_h(t)|$ channel matrix $\boldsymbol{\Xi}_h(t;\nu)$, the LZFBF precoding matrix of dimension $M\times|\mathcal{S}_h(t)|$ is given by the column-normalized pseudo-inverse

$$\mathbf{V}(t;\nu) = \boldsymbol{\Xi}_h(t;\nu)(\boldsymbol{\Xi}_h^{\mathsf{H}}(t;\nu)\boldsymbol{\Xi}_h(t;\nu))^{-1}\Lambda^{1/2}(t;\nu) \tag{32}$$

where $\Lambda(t;\nu)$ is a diagonal matrix with u-th diagonal element given by

$$\Lambda_u(t;\nu) = \frac{1}{\left[\left(\boldsymbol{\Xi}_h^{\mathsf{H}}(t;\nu)\boldsymbol{\Xi}_h(t;\nu)\right)^{-1}\right]_{uu}} \tag{33}$$

($[\cdot]_{uu}$ denotes the u-th diagonal element of the matrix argument). Using the fact that $\boldsymbol{\Xi}_h^{\mathsf{H}}(t;\nu)\mathbf{V}_h(t;\nu) = \Lambda^{1/2}(t;\nu)$, the resulting downlink channel to user $u\in\mathcal{S}_h(t)$ becomes

$$\begin{aligned} y_u(t;\nu) = &\sqrt{g_{hu}(t)\Lambda_u(t;\nu)}x_{hu}(t;\nu) \\ &+ \sum_{h'\neq h}\sqrt{g_{h'u}(t)}\boldsymbol{\xi}_{h'u}^{\mathsf{H}}(t;\nu)\mathbf{V}_{h'}(t;\nu)\mathbf{x}_{h'}(t;\nu) \\ &+ z_u(t;\nu). \end{aligned} \tag{34}$$

Our goal here is to obtain an accurate yet simple characterization of the feasible instantaneous rate region $\mathcal{R}(\omega(t))$ for the channel model (34), where the network state is defined by the pathloss coefficients (reflecting the network topology), i.e., $\omega(t) = [g_{hu}(t)]$. To this purpose, we shall exploit some standard results in large random matrix theory (see [17], [31]–[34]), consider achievable rates under Gaussian random coding and worst-case uncorrelated additive noise plus interference [35], and a simple application of Jensen's inequality. By dividing the channel coefficient by $\sqrt{M}$ and scaling up the helpers' transmit power by M we obtain an equivalent channel model, for which the following deterministic approximation holds: as M and $|\mathcal{S}_h(t)|$ become large with fixed ratio $\frac{|\mathcal{S}_h(t)|}{M}\leq 1$, it is well-known that

$$\Lambda_u(t;\nu) \approx \left(1 - \frac{|\mathcal{S}_h(t)|-1}{M}\right) \tag{35}$$

where the approximation error $\Lambda_u(t;\nu) - \left(1-\frac{|\mathcal{S}_h(t)|-1}{M}\right)$ is asymptotically Gaussian with a variance that vanishes as $O(1/M^2)$ [31], [32]. By treating the inter-cell interference as (uncorrelated) additive noise, and assuming a very large number of time-frequency symbols per slot, it is immediate to show that the following rate is achievable:

$$R_{hu}(t) = \frac{1}{\mathcal{V}}\sum_{\nu=1}^{\mathcal{V}}\log\left(1+\frac{\Lambda_u(t;\nu)g_{hu}(t)\frac{MP_h}{|\mathcal{S}_h(t)|}}{1+\sum\limits_{h'\neq h}g_{h'u}(t)\|\boldsymbol{\xi}_{h'u}^{\mathsf{H}}(t;\nu)\mathbf{V}_{h'}(t;\nu)\|^2\frac{MP_{h'}}{|\mathcal{S}_{h'}(t)|}}\right). \tag{36}$$

Now, using the deterministic approximation (35), replacing the arithmetic mean over the subcarrier index with an ensemble average over the fading statistics, and using Jensen's inequality, we find the desired approximate achievable rate expression as

$$\begin{aligned} R_{hu}(t) \approx &\\ &\mathbb{E}\left[\log\left(1+\frac{M-|\mathcal{S}_h(t)|+1}{|\mathcal{S}_h(t)|}\times\right.\right. \\ &\left.\left.\frac{g_{hu}(t)P_h}{1+\sum\limits_{h'\neq h}g_{h'u}(t)\|\boldsymbol{\xi}_{h'u}^{\mathsf{H}}(t;\nu)\mathbf{V}_{h'}(t;\nu)\|^2\frac{MP_{h'}}{|\mathcal{S}_{h'}(t)|}}\right)\right] &(37)\\ \geq &\log\left(1+\frac{M-|\mathcal{S}_h(t)|+1}{|\mathcal{S}_h(t)|}\times\right. \\ &\left.\frac{g_{hu}(t)P_h}{1+\sum\limits_{h'\neq h}g_{h'u}(t)\mathbb{E}[\|\boldsymbol{\xi}_{h'u}^{\mathsf{H}}(t;\nu)\mathbf{V}_{h'}(t;\nu)\|^2]\frac{MP_{h'}}{|\mathcal{S}_{h'}(t)|}}\right) &(38)\\ = &\log\left(1+\frac{M-|\mathcal{S}_h(t)|+1}{|\mathcal{S}_h(t)|}\frac{g_{hu}(t)P_h}{1+\sum\limits_{h'\neq h}g_{h'u}(t)P_{h'}}\right), &(39) \end{aligned}$$

(see also [34] for a similar argument). In (38) we used the fact that, by construction of the normalization of the LZFBF precoder in (32), we have $\mathbb{E}[\|\boldsymbol{\xi}_{h'u}^{\mathsf{H}}(t;\nu)\mathbf{V}_{h'}(t;\nu)\|^2] = \frac{1}{M}\mathrm{tr}(\mathbb{E}[\mathbf{V}_{h'}^{\mathsf{H}}(t;\nu)\mathbf{V}_{h'}(t;\nu)]) = |\mathcal{S}_{h'}(t)|/M$.

For the sake of notational simplicity, we define the rate (raw) vectors $\mathbf{c}_h(\mathcal{S}_h(t),t)\in\mathbb{R}_+^{|\mathcal{U}|}$ with components

$$c_{hu}(\mathcal{S}_h(t),t) = \begin{cases} 0 & \text{if } u\notin\mathcal{S}_h(t) \\ sR_{hu}(t) & \text{if } u\in\mathcal{S}_h(t) \end{cases} \tag{40}$$

where $R_{hu}(t)$ is given by (38) and s denotes the number of time-frequency symbols per slot. Notice that for each helper h and each set of scheduled users $\mathcal{S}_h(t)\subseteq\mathcal{N}(h)$ the vector

$\mathbf{c}_h(\mathcal{S}_h(t),t)$ yields the number of information bits that the users $u \in \mathcal{S}_h(t)$ can successfully decode from helper h during slot t. Hence, the desired concise expression of $\mathcal{R}(\omega(t))$ is given as follows:

Proposition 1: **Feasible instantaneous rate region:** For every t, the region of instantaneous feasible rates $\mathcal{R}(\omega(t))$ is formed by all $|\mathcal{H}| \times |\mathcal{U}|$ rate matrices $[\mu_{hu}(t)]$ whose h-th row is $\mathbf{c}_h(\mathcal{S}_h(t),t)$, for some $\mathcal{S}_h(t) \subseteq \mathcal{N}(h)$, for all $h \in \mathcal{H}$. □

We assume that the receiver at every user is "smart" in the sense that it can decode multiple streams in the same transmission slot, i.e., user u, in transmission slot t, can receive $\mu_u(t) = \sum_{h\in\mathcal{N}(u)} \mu_{hu}(t)$ information bits by simultaneously downloading $\mu_{hu}(t)$ bits from helpers $h \in \mathcal{N}(u)$. Notice that each stream is achievable (in an information theoretic sense), by treating the other streams as Gaussian noise, i.e., we do not make use of multiuser detection schemes (e.g., based on successive interference cancellation) at the user receivers. Therefore, our rate expressions are representative of what can be achieved with today's user device technology.

For the sake of comparison, in the simulation results of Section VII we also consider a *dumb receiver heuristic* where each user u decodes only the strongest data stream and therefore downloads only $\max_{h\in\mathcal{N}(u)} \mu_{hu}(t)$ information bits. While the dumb receiver heuristic is a degradation of the optimal solution involving advanced receivers, the simulation results in Section VII show that this degradation is almost negligible. This also implicitly indicates that, in most relevant practical topologies and pathloss scenarios, it is unlikely that the same user is scheduled by more than one helper in the same transmission slot, i.e., $\mathcal{S}_h(t) \cap \mathcal{S}_{h'}(t)$ is empty with high probability for $h \neq h'$.

Remark 3: *Although here we have chosen, for the sake of clarity, to consider the somehow simplistic case of i.i.d. zero-mean channel vectors and perfect channel state information, we hasten to say that introducing deterministic approximations of the user instantaneous rates in more involved and realistic cases including base-station antenna correlation [36]–[40], and/or pilot-based channel state information, typically obtained through uplink pilots and TDD reciprocity [19]–[21], [41] and including the effect of pilot contamination, is just a matter of a simple exercise. However, since the focus of this paper is the cross-layer optimization of video-streaming over a multicell multiuser wireless network employing (massive) MU-MIMO, but not to give a tutorial presentation of well-known rate deterministic approximations, which are already provided elsewhere, we have chosen to stay with the simple model defined above. In fact, while the feasible instantaneous rate region takes on the same form given in Proposition 1, the expression of the instantaneous feasible rates may be significantly more complicated. As a simple example, note that if one wishes to take into account the uplink pilot overhead, the number of symbols per slot s introduced before should be changed into $s - |\mathcal{S}_h(t)|$, since on each slot the users $u \in \mathcal{S}_h(t)$ must transmit mutually orthogonal uplink pilots, requiring $|\mathcal{S}_h(t)|$ symbols per slot [19].*

B. Transmission Scheduling with Massive MU-MIMO Helpers

We now particularize the problem (22) to the specific wireless system with massive MU-MIMO helpers. Since the rate vectors at each helper (i.e., the rows $\mathbf{c}_h(\mathcal{S}_h(t),t)$) can be chosen independently at each helper, (22) decouples into separate maximizations for each helper h given by the following discrete optimization problem:

$$\begin{aligned} &\text{maximize} \sum_{u\in\mathcal{N}(h)} Q_u(t)\mu_{hu}(t) \\ &\text{subject to } \{\mu_{hu}(t)\}_{u\in\mathcal{N}(h)} \in \{\mathbf{c}_h(\mathcal{S}_h,t) : \mathcal{S}_h \subseteq \mathcal{N}(h)\}. \end{aligned} \tag{41}$$

This corresponds to maximizing, at each helper h, the weighted sum rate over the discrete set of vectors $\{\mathbf{c}_h(\mathcal{S}_h,t) : \mathcal{S}_h \subseteq \mathcal{N}(h)\}$ with an exponential number $2^{|\mathcal{N}(h)|} - 1$ of choices for the active user subset. The key observation that allows to eliminate this exponential complexity is that when helper h schedules the subset $\mathcal{S}_h$ of users for MU-MIMO beamforming, the rate of each user $u \in \mathcal{S}_h$ depends only on the cardinality $|\mathcal{S}_h|$ but not on the identity of the members of the subset $\mathcal{S}_h$. This is an important consequence of the massive MIMO "deterministic" rate behavior, due to the channel hardening in the presence of a large number of antennas. As a consequence, for a fixed subset size S, the subset $\mathcal{U}^*(S,t)$ of users maximizing the weighted sum rate can be obtained by sorting the users in $\mathcal{N}(h)$ according to the weighted rate $Q_u(t)\log\left(1+\frac{M-S+1}{S}\frac{P_h g_{hu}(t)}{\left(1+\sum_{h'\neq h} P_{h'u} g_{h'u}(t)\right)}\right)$ and choosing greedily the best S users. Thus, we have

$$\begin{aligned} &\mathcal{U}_h^*(S,t) = \\ &\text{argmax-}S \left\{ Q_u(t)\log\left(1+\frac{M-S+1}{S}\times \right.\right. \\ &\qquad \left.\left. \frac{P_h g_{hu}(t)}{1+\sum\limits_{h'\neq h} P_{h'u} g_{h'u}(t)}\right) \;:\; u\in\mathcal{N}(h) \right\}, \end{aligned} \tag{42}$$

where argmax-S denotes the operation of choosing the first S elements of a set of real numbers sorted in decreasing order.

This *sort & greedy selection* procedure is repeated for every subset size yielding all the subsets $\{\mathcal{U}^*(S,t)\}_{S=1}^{|\mathcal{N}(h)|}$. Then, from these subsets, the subset $\mathcal{U}^*(t)$ which has the maximum weighted sum rate is picked as

$$\begin{aligned} &\mathcal{U}_h^*(t) = \\ &\text{argmax} \left\{ \sum_{u\in\mathcal{U}_h^*(S,t)} Q_u(t)\log\left(1+\frac{M-S+1}{S}\times \right.\right. \\ &\qquad \left.\left. \frac{P_h g_{hu}(t)}{1+\sum\limits_{h'\neq h} P_{h'u} g_{h'u}(t)}\right) \;:\; \mathcal{U}_h^*(S,t)\ \forall\ S \right\} \end{aligned} \tag{43}$$

yielding the optimal solution to (41).

A typical sorting algorithm has complexity $O\left(|\mathcal{N}(h)|\log(|\mathcal{N}(h)|)\right)$ and since the sorting procedure is repeated for every subset size, the algorithm has complexity $O\left(|\mathcal{N}(h)|^2\log(|\mathcal{N}(h)|\right)$ which improves upon existing user scheduling algorithms [42] for the MIMO broadcast channel.

Remark 4: *Notice that for a practical implementation of the transmission scheduling algorithm, each helper h first needs to learn locally the request queue lengths Q_u of the users in its neighborhood $\mathcal{N}(h)$. Then, it has to greedily pick the user subset $\mathcal{U}_h^*(t)$ according to (43) and again learn the small scale fading channel vectors $\boldsymbol{\xi}_{hu}$ of the users in $\mathcal{U}_h^*(t)$ for LZFBF pre-coding through some form of channel state feedback from the users. Furthermore, the $\mu_{hu}(t)$ video-encoded bits transmitted by helper h to user u should correspond to the chunks at the head of line of the request queue Q_u, encoded at the quality level chosen by user u in a previous video chunk slot based on the pull scheme (19). Thus, each user u must also broadcast the metadata (chunk number and quality level) of the chunks at the head of line along with Q_u to the helpers in $\mathcal{N}(u)$.*

VI. Pre-buffering and re-buffering chunks

As described in Section II, the playback process consumes chunks at a fixed playback rate $1/T_{\rm gop}$ (one chunk per video chunk slot i), while the number of chunks received per video chunk slot is a random variable due to the fact that $\omega(t)$ is a random process and the transmission resources are dynamically allocated by the DPP scheduling policy. In order to prevent stall events, each user u should choose its pre-buffering time T_u to be larger than the maximum delay with which a chunk is delivered to it. However, such maximum delay is neither deterministic nor known a priori. Moreover, even in special cases where the maximum delay of each request queue in the system admits a deterministic bound (e.g., see [25]), such a bound may be loose and setting the pre-buffering time to be larger than that bound might be simply unacceptable in a practical system implementation. Therefore, in this section, following our previous work in [13], we propose a simple method where each user u adaptively determines the pre-buffering time T_u on the basis of estimates of local delays obtained by monitoring the delivery times in a sliding window spanning a fixed number of video chunk slots. The key difference with respect to [13] is that the present scheme is much simpler, since the proposed pull congestion control ensures that chunks are received in the right playback order.

An example of the playback buffer dynamics is illustrated in Table I and Fig. 1. The table indicates the chunk numbers and their respective arrival times. The blue curve in Fig. 1 shows the time evolution of the number of chunks downloaded in the playback buffer. The green curve indicates the evolution with time of the number of chunks consumed by playback. The playback consumption starts after an initial pre-buffering delay $T_u = d$, as indicated in the figure. At any video chunk slot i, the chunk requested at $i-d$ is expected to be available in the playback buffer. However, if the chunk is delivered with a delay greater than d, the two curves meet and a buffer underrun event occurs. In order to prevent these events, each user u should choose its pre-buffering time T_u to be larger than the maximum delay.

More formally, the goal here is to determine the delay T_u after which user u should start playback, with respect to the time at which the first chunk is requested (beginning of the streaming session). We define the size of the playback buffer $\Psi_u(i)$ as the number of chunks available in the buffer at video chunk slot i but not yet played out. Without loss of generality, assume that the streaming session starts at $i = 1$. Then, $\Psi_u(i)$ evolves at the video timescale over video chunk slots $i \in \{1, 2, 3, \ldots\}$ as:

$$\Psi_u(i) = \max\{\Psi_u(i-1) - 1\{i > T_u\}, 0\} + a_i. \qquad (44)$$

where $1\{\mathcal{K}\}$ denotes the indicator function of a condition or event $\mathcal{K}$ and a_i is the number of chunks which are completely downloaded in the transmission slots between $t = (i-1)n$ and $t = in$. Note that the playback buffer is updated every video chunk slot i, i.e., at the time scale of seconds. Thus, if the download of a chunk is completed between $t = (i-1)n$ and $t = in$, from the playback buffer's perspective, the chunk is considered to have arrived at the end of the i-th video chunk slot, i.e., at $t = in$. Let A_k denote the video chunk slot in which chunk k arrives at the user and let W_k denote the delay (measured in video chunk slots) with which chunk k is delivered. Note that the longest period during which $\Psi_u(i)$ is not incremented is given by the maximum delay to deliver chunks. Thus, each user u needs to adaptively estimate W_k in order to choose T_u. In the proposed method, at each video chunk slot $i = 1, 2, \ldots$, user u calculates the maximum observed delay E_i in a sliding window of size Δ, by letting:

$$E_i = \max\{W_k \ : \ i - \Delta + 1 \leq A_k \leq i\}. \qquad (45)$$

Finally, user u starts its playback when Ψ_i crosses the level ρE_i, i.e., $T_u = \min\{i \ : \ \Psi_u(i) \geq \rho E_i\}$ where ρ is an algorithm control parameter. If a stall event occurs at video chunk slot T, i.e., $\Psi_i = 0$ for $i > T$, the algorithm enters a re-buffering phase in which the same algorithm presented above is employed again to determine the new instant $T + T_u + 1$ at which playback is restarted. With slight abuse of notation, we re-use T_u to denote the re-buffering delay although it is re-estimated using the sliding window method at each new stall event.

VII. Numerical Experiment

Our simulations are based on a network topology formed by a 80m×80m region with 5 helpers (indicated by ∘'s) as shown in Fig. 2. The users (indicated by ∗'s) are generated according to a non-homogeneous Poisson point process with higher density in a central region of size $\frac{80}{3}$m×$\frac{80}{3}$m, as shown in Fig. 2.

Each helper has M antennas and serves user sets of size upto S, with transmission power of 35dBm. The pathloss from a helper to a user is given by $\frac{1}{1+(\frac{d}{40})^{3.5}}$, with d representing the helper-user distance (assuming a torus wrap-around model to avoid boundary effects). We assume a PHY fame duration of 10 ms and a total system bandwidth of 18 Mhz as specified in the LTE 4G standard. With one OFDM resource block (7×12 channel symbols) spanning 0.5 ms in time and 180 Khz in bandwidth (corresponding to 12 adjacent subcarriers each with 15 KHz bandwidth), each transmission slot spans $s = 84 \times 100 \times 20$ channel symbols.

TABLE I: Arrival times of chunks

Chunk number	1	2	3	4	5	6	7	8	9	10	11	12	13
Arrival time	3	4	6	9	10	11	12	13	16	17	19	20	21

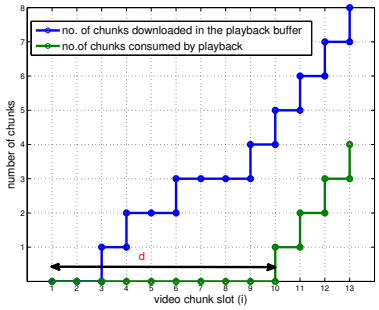


Fig. 1: Evolution of number of ordered and consumed chunks

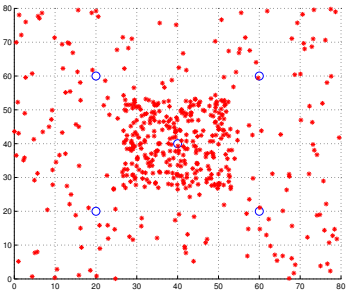


Fig. 2: Simulation setup

We assume that all the users request chunks successively from VBR-encoded video sequences. Each video file is a long sequence of chunks, each of duration 0.5 seconds and with a frame rate 30 frames per second. We consider a specific video sequence formed by 800 chunks, constructed using several standard video clips from the database in [43]. The chunks are encoded into different quality modes with the quality index measured using the *Structural SIMilarity* (SSIM) index defined in [44]. The chunks from 1 to 200 are encoded into 8 quality modes with an average bitrate of 631 kbps. Chunks 201 to 400 are encoded in 4 quality modes at an average bitrate of 3908 kbps. Similarly, chunks $401-600$ and $601-800$ are encoded into 4 and 8 quality modes with average bitrates of 6679 kbps and 556 kbps respectively. In the simulation, each user starts its streaming session of 1000 chunks from some arbitrary position in this reference video sequence and successively requests 1000 chunks by cycling through the sequence.

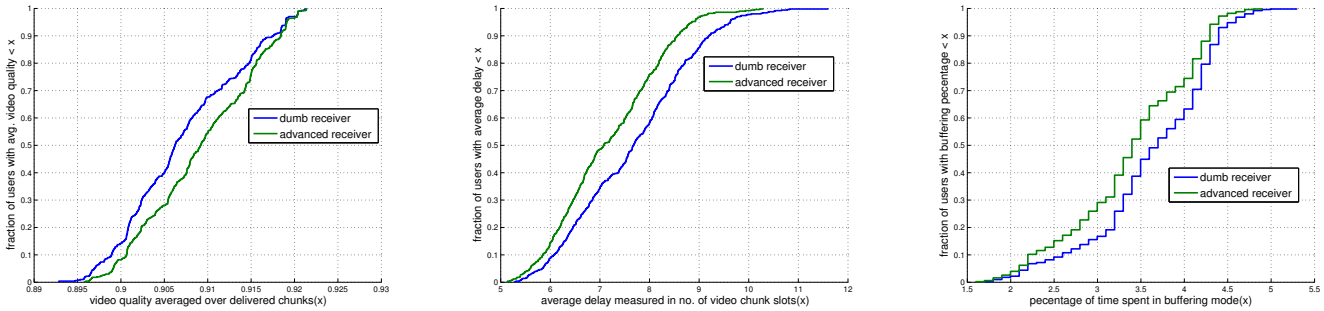
We choose the utility function $\Phi_u(\cdot) = \log(\cdot)\ \forall\ u \in \mathcal{U}$ to impose proportional fairness. We set the pre-buffering algorithm control parameter (described in Section VI) $\rho = 3$. We simulate our algorithm for the layout shown in Fig. 2 (with around 500 users generated according to a non-homogenous Poisson point process as explained above). At $t = 1$, all the users simultaneously start streaming 1000 chunks.

We studied the performance of our algorithm with $M = 40$ antennas and maximum active user subset size $S = 10$ for different values of the policy control parameter V and observed that both the QoE metrics average video quality and the $\%$ of time spent in buffering mode are satisfactory for the choice of $V = 2 * 10^{14}$. We use that value for the rest of the simulations in this section.

We now study the performance loss experienced under the *dumb receiver heuristic* where the receiver at every user u decodes only the strongest signal and downloads only $\max_{h\in\mathcal{N}(u)} \mu_{hu}(t)$ in contrast to the macro diversity advanced receiver which can decode multiple signals simultaneously and download all the $\sum_{h\in\mathcal{N}(u)} \mu_{hu}(t)$ bits. Using $M = 40$ and $S = 10$, we simulate our algorithm and plot the CDF's over the user population of a) the average video quality b) the average delay in the reception of video chunks measured in video chunk slots and c) the $\%$ of playback time spent in buffering mode in Figs. 3a, 3b and 3c respectively. We observe that the performance loss in using a dumb receiver is fairly negligible and therefore use a dumb receiver for the rest of the simulations in this section.

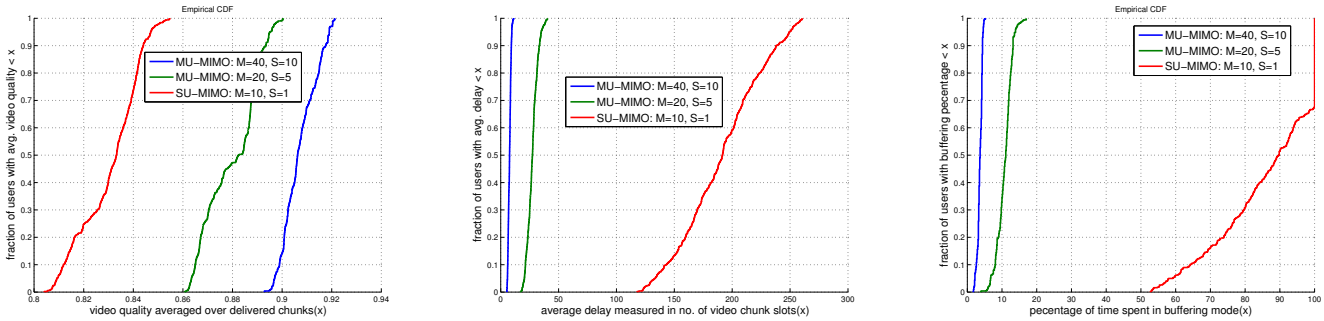
We next study the QoE improvement when MU-MIMO is deployed at the helpers in place of legacy single-user MIMO (SU-MIMO) systems. We plot the CDF over the user population of the same video streaming QoE metrics as in the previous figures for three different cases 1) MU-MIMO with $M = 40$ antennas and maximum active user subset size $S = 10$; 2) MU-MIMO with $M = 20$ antennas and maximum active user subset size $S = 5$; 3) SU-MIMO with $M = 10$ antennas. From Figs. 4a, 4c and 4b, we can observe that there is significant improvement of video streaming performance in terms of the average video quality, the average delay (or alternately the percentage of time spent in buffering mode) when MU-MIMO is employed at the PHY layer in comparison to SU-MIMO. This clearly indicates that upgrading current SU-MIMO systems to massive MU-MIMO is a promising approach to meet the increasing demands for HD video streaming.

Finally, we study the benefits of using a cross layer approach in comparison to a baseline scheme representative of legacy wireless systems. We perform this comparison for the case where every helper employs SU-MIMO with $M = 10$ antennas. For the baseline scheme, every user first fixes its



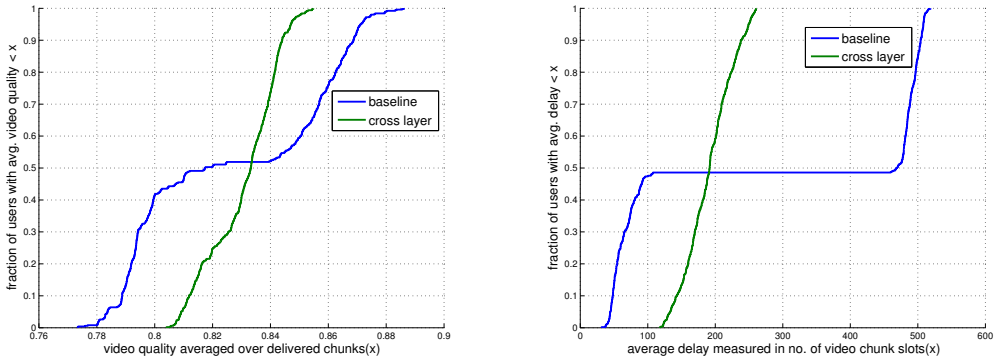
(a) (b) (c)

Fig. 3: Performance comparison of advanced and dumb receivers.



(a) (b) (c)

Fig. 4: Video streaming QoE improvement with MU-MIMO over SU-MIMO



(a) (b)

Fig. 5: Performance comparison of a cross-layer approach with a baseline scheme.

association with the unique helper that provides the maximum received signal strength (RSSI) $P_h g_{hu}$ and then uses the same control decision (19) to choose the quality levels for the chunks that arrive into the request queue every video chunk slot. Furthermore, we assume that the helpers *locally* employ Proportional Fairness scheduling [29], which under the massive MIMO deterministic rate approximation reduces to *equal air-time* scheduling. In brief, each helper h schedules the users associated with it through the max-RSSI scheme in a round-robin fashion across the transmission slots *independent of the request queue lengths at the users*. This baseline scheme is representative of current practical systems where the decisions across different layers are independent and there is no interaction between the upper and lower layers. We plot the CDFs over the user population of the average video quality and the average delay in the reception of chunks in Figs. 5a and 5b respectively. We can observe that the cross layer scheme treats the users in a fair manner while the baseline scheme favors some users at the expense of other users in the system. Since the topology in Fig. 2 has a central region with higher user density (hotspot) and the user-helper association is max-RSSI based in the baseline scheme, most of the users

in the hotspot get associated to the helper in the center of the topology. As a consequence, this helper get *overloaded* with many users associated to it. Moreover, since the helper employs the round-robin scheduling policy in the baseline scheme, each user in the hotspot gets scheduled only on a small fraction of transmission slots resulting in poor video quality and delay performance. However, the users which are outside the hotspot get associated to the lightly loaded helpers and therefore experience better video streaming QoE. This explains the skewed nature of the distributions in Figs. 5a and 5b corresponding to the baseline scheme.

On the other hand, with the cross-layer scheme, the user-helper association is dynamic and it changes during the course of control policy depending on the transmission scheduling decisions which in turn depend on the achievable rates of the users and their dynamically changing request queue lengths. Such dynamic user-helper association implicitly balances the load across all the helpers in the system leading to fairness in QoE performance across the user population.

VIII. Conclusions

In this work we propose WiFlix, a system for efficient streaming of video content in a network of helpers capable of implementing the advanced physical layer technique massive MU-MIMO. We formulate a NUM problem to maximize the video streaming QoE of the users in the network and solve it using the Lyapunov Optimization technique to derive a control policy which decomposes into congestion control decisions at the users and transmission scheduling decisions at the helpers. We devise a low complexity greedy user selection scheme to solve optimally the combinatorial problem of scheduling users for multiuser beamforming. The transmission scheduling decisions consist of each base station choosing the subset of users for MU-MIMO beamforming. By exploiting the channel hardening effect of high dimensional MIMO channels, we reduce the combinatorial weighted sum rate maximization over the multiuser multicell network (which would involve an exponentially complex exhaustive user selection, or some polynomial complexity heuristic greedy user selection at each base station) to a simple subset selection problem which is optimally solved by a low complexity algorithm. The algorithm is amenable to easy implementation with local, independent user scheduling decisions at the helpers. Similarly, the congestion control decisions can be implemented independently at the users with each user opportunistically pulling chunks from its neighboring helpers and adapting the quality of the chunk requests in response to changing network conditions reflected by changing request queue sizes. Possible future considerations include implementing and testing WiFlix in practice.

Appendix A
Proof of Theorem 1 and of Corollary 1

As in Section III, we consider the following problem, equivalent to (23) – (26), which involves a sum of time-averages instead of functions of time averages and introduces the auxiliary variables $\gamma_u(t)$:

$$\text{maximize } \frac{1}{T}\sum_{\tau=jT}^{(j+1)T-1}\sum_{u\in\mathcal{U}}\phi_u\left(\gamma_u(\tau)\right) \tag{46a}$$

$$\text{subject to } \frac{1}{T}\sum_{\tau=jT}^{(j+1)T-1}\left[B_{f_u}\left(m_u(\tau),\tau\right)-\mu_u\left(\tau\right)\right]\leq 0\ \forall\ u\in\mathcal{U} \tag{46b}$$

$$\frac{1}{T}\sum_{\tau=jT}^{(j+1)T-1}\left[\gamma_u\left(\tau\right)-D_{f_u}\left(m_u(\tau),\tau\right)\right]\leq 0\ \forall\ u\in\mathcal{U} \tag{46c}$$

$$D_u^{\min}\leq\gamma_u(\tau)\leq D_u^{\max}\ \forall\ u\in\mathcal{U},\ \forall\ \tau\in\{jT,\ldots,(j+1)T-1\} \tag{46d}$$

$$[\mu_{hu}(\tau)]\in\mathcal{R}(\omega(\tau))\ \forall\ \tau\in\{jT,\ldots,(j+1)T-1\} \tag{46e}$$

$$m_u(\tau)\in\{1,2,\ldots,N_{f_u}\}\ \forall\ u\in\mathcal{U},\ \forall\ \tau\in\{jT,\ldots,(j+1)T-1\} \tag{46f}$$

The update equations for the request queues $Q_u\ \forall\ u\in\mathcal{U}$ and the virtual queues $\Theta_u\ \forall\ u\in\mathcal{U}$ are given in (3) and in (7), respectively. Let $\mathbf{G}(\tau)=\left[\mathbf{Q}^{\mathsf{T}}(\tau),\boldsymbol{\Theta}^{\mathsf{T}}(\tau)\right]^{\mathsf{T}}$ be the combined queue backlogs column vector, and define the quadratic Lyapunov function $L(\mathbf{G}(\tau)=\frac{1}{2}\mathbf{G}^{\mathsf{T}}(\tau)\mathbf{G}(\tau)$. Fix a particular slot τ in the j-th frame. We first consider the one-slot drift of $L(\mathbf{G}(\tau))$. From (15), we know that

$$L(\mathbf{G}(\tau+1))-L(\mathbf{G}(\tau))\leq\mathcal{K}+\left(\mathbf{B}(\tau)-\boldsymbol{\mu}(\tau)\right)^{\mathsf{T}}\mathbf{Q}(\tau)+\left(\boldsymbol{\gamma}(\tau)-\mathbf{D}(\tau)\right)^{\mathsf{T}}\boldsymbol{\Theta}(\tau) \tag{47}$$

where $\mathcal{K}$ is a uniform bound on the term

$$\frac{1}{2}\left[\left(\mathbf{B}(\tau)-\boldsymbol{\mu}(\tau)\right)^{\mathsf{T}}\left(\mathbf{B}(\tau)-\boldsymbol{\mu}(\tau)\right)+\left(\boldsymbol{\gamma}(\tau)-\mathbf{D}(\tau)\right)^{\mathsf{T}}\left(\boldsymbol{\gamma}(\tau)-\mathbf{D}(\tau)\right)\right] \tag{48}$$

, which exists under the realistic assumption that the chunk sizes, the transmission rates and the video quality measures are upper bounded by some constants, independent of τ. We choose $\mathcal{K}$ such that

$$\mathcal{K}>2\boldsymbol{\kappa}^{\mathsf{T}}\boldsymbol{\kappa} \tag{49}$$

where $\boldsymbol{\kappa}$ is a vector whose components are all equal to the same number κ and this number is a uniform upper bound on the maximum possible magnitude of drift in any of the queues (both the request queues Q_u and the virtual queues Θ_u) in one slot. With the additional penalty term $-V\sum_{u\in\mathcal{U}}\phi_u(\gamma_u(\tau))$ added on both sides of (47), we have the following DPP inequality:

$$L(\mathbf{G}(\tau+1)) - L(\mathbf{G}(\tau)) - V\sum_{u\in\mathcal{U}}\phi_u(\gamma_u(\tau))$$
$$\leq \mathcal{K} + (\mathbf{B}(\tau)-\boldsymbol{\mu}(\tau))^{\mathsf{T}}\mathbf{Q}(\tau) + (\boldsymbol{\gamma}(\tau)-\mathbf{D}(\tau))^{\mathsf{T}}\boldsymbol{\Theta}(\tau)$$
$$- V\sum_{u\in\mathcal{U}}\phi_u(\gamma_u(\tau)) \quad (50)$$

Let the DPP policy which minimizes the right hand side of the *drift-plus-penalty* inequality (50) comprise of the control actions $\{m_u(\tau)\}_{\tau=jT}^{(j+1)T-1}\ \forall\ u\in\mathcal{U}$, $\{\boldsymbol{\gamma}(\tau)\}_{\tau=jT}^{(j+1)T-1}$ and $\{(\mu_{hu}(\tau))\}_{\tau=jT}^{(j+1)T-1}$. Since the DPP policy minimizes the expression on the RHS of (50), any other policy comprising of the control actions $\{m_u^*(\tau)\}_{\tau=jT}^{(j+1)T-1}\ \forall\ u\in\mathcal{U}$, $\{\boldsymbol{\gamma}^*(\tau)\}_{\tau=jT}^{(j+1)T-1}$ and $\{(\mu_{hu}^*(\tau))\}_{\tau=jT}^{(j+1)T-1}$ would give a larger value of the expression. We therefore have

$$L(\mathbf{G}(\tau+1)) - L(\mathbf{G}(\tau)) - V\sum_{u\in\mathcal{U}}\phi_u(\gamma_u(\tau))$$
$$\leq \mathcal{K} + (\mathbf{B}^*(\tau)-\boldsymbol{\mu}^*(\tau))^{\mathsf{T}}\mathbf{Q}(\tau) + (\boldsymbol{\gamma}^*(\tau)-\mathbf{D}^*(\tau))^{\mathsf{T}}\boldsymbol{\Theta}(\tau)$$
$$- V\sum_{u\in\mathcal{U}}\phi_u(\gamma_u^*(\tau)). \quad (51)$$

Further, we note that the maximum change in the queue length vectors $Q_u(\tau)$ and $\Theta_u(\tau)$ from one slot to the next is bounded by κ. Thus, we have for all $\tau\in\{jT,\ldots,(j+1)T-1\}$

$$|Q_u(\tau)-Q_u(jT)| \leq (\tau-jT)\kappa\ \ \forall\ u\in\mathcal{U} \quad (52)$$
$$|\Theta_u(\tau)-\Theta_u(jT)| \leq (\tau-jT)\kappa\ \ \forall\ u\in\mathcal{U} \quad (53)$$

Substituting the above inequalities in (51), we have

$$L(\mathbf{G}(\tau+1)) - L(\mathbf{G}(\tau)) - V\sum_{u\in\mathcal{U}}\phi_u(\gamma_u(\tau))$$
$$\leq \mathcal{K} + (\mathbf{B}^*(\tau)-\boldsymbol{\mu}^*(\tau))^{\mathsf{T}}(\mathbf{Q}(jT)+(\tau-jT)\boldsymbol{\kappa})$$
$$+ (\boldsymbol{\gamma}^*(\tau)-\mathbf{D}^*(\tau))^{\mathsf{T}}(\boldsymbol{\Theta}(jT)+(\tau-jT)\boldsymbol{\kappa})$$
$$- V\sum_{u\in\mathcal{U}}\phi_u(\gamma_u^*(\tau)). \quad (54)$$

Then, summing (54) over $\tau\in\{jT,\ldots,(j+1)T-1\}$, we obtain the T-slot Lyapunov drift over the j-th frame:

$$L(\mathbf{G}((j+1)T)) - L(\mathbf{G}(jT)) - V\sum_{\tau=jT}^{jT+T-1}\sum_{u\in\mathcal{U}}\phi_u(\gamma_u(\tau))$$
$$\leq \mathcal{K}T + \left(\sum\nolimits_{\tau=jT}^{jT+T-1}(\mathbf{B}^*(\tau)-\boldsymbol{\mu}^*(\tau))\right)^{\mathsf{T}}\mathbf{Q}(jT)$$
$$+ \left(\sum\nolimits_{\tau=jT}^{jT+T-1}(\mathbf{B}^*(\tau)-\boldsymbol{\mu}^*(\tau))(\tau-jT)\right)^{\mathsf{T}}\boldsymbol{\kappa}$$
$$+ \left(\sum\nolimits_{\tau=jT}^{jT+T-1}(\boldsymbol{\gamma}^*(\tau)-\mathbf{D}^*(\tau))\right)^{\mathsf{T}}\boldsymbol{\Theta}(jT)$$
$$+ \left(\sum\nolimits_{\tau=jT}^{jT+T-1}(\boldsymbol{\gamma}^*(\tau)-\mathbf{D}^*(\tau))(\tau-jT)\right)^{\mathsf{T}}\boldsymbol{\kappa}$$
$$- V\sum\nolimits_{\tau=jT}^{jT+T-1}\sum_{u\in\mathcal{U}}\phi_u(\gamma_u^*(\tau)) \quad (55)$$

Using the inequalities $\mathbf{B}^*(\tau)-\boldsymbol{\mu}^*(\tau)\leq 2\boldsymbol{\kappa}$, $\boldsymbol{\gamma}^*(\tau)-\mathbf{D}^*(\tau)\leq 2\boldsymbol{\kappa}$ in (55), we have

$$L(\mathbf{G}((j+1)T)) - L(\mathbf{G}(jT)) - V\sum_{\tau=jT}^{jT+T-1}\sum_{u\in\mathcal{U}}\phi_u(\gamma_u(\tau))$$
$$\leq \mathcal{K}T + \left(\sum\nolimits_{\tau=jT}^{jT+T-1}(\mathbf{B}^*(\tau)-\boldsymbol{\mu}^*(\tau))\right)^{\mathsf{T}}\mathbf{Q}(jT)$$
$$+ 2\left(\sum\nolimits_{\tau=jT}^{jT+T-1}(\tau-jT)\right)\boldsymbol{\kappa}^{\mathsf{T}}\boldsymbol{\kappa}$$
$$+ \left(\sum\nolimits_{\tau=jT}^{jT+T-1}(\boldsymbol{\gamma}^*(\tau)-\mathbf{D}^*(\tau))\right)^{\mathsf{T}}\boldsymbol{\Theta}(jT)$$
$$+ 2\left(\sum\nolimits_{\tau=jT}^{jT+T-1}(\tau-jT)\right)\boldsymbol{\kappa}^{\mathsf{T}}\boldsymbol{\kappa}$$
$$- V\sum\nolimits_{\tau=jT}^{jT+T-1}\sum_{u\in\mathcal{U}}\phi_u(\gamma_u^*(\tau)) \quad (56)$$

Using $\boldsymbol{\kappa}^{\mathsf{T}}\boldsymbol{\kappa}\leq\frac{\mathcal{K}}{2}$, $\sum_{\tau=jT}^{jT+T-1}(\tau-jT)=\frac{T(T-1)}{2}$, we get

$$L(\mathbf{G}((j+1)T)) - L(\mathbf{G}(jT)) - V\sum_{\tau=jT}^{jT+T-1}\sum_{u\in\mathcal{U}}\phi_u(\gamma_u(\tau))$$
$$\leq \mathcal{K}T^2 + \left(\sum\nolimits_{\tau=jT}^{jT+T-1}(\mathbf{B}^*(\tau)-\boldsymbol{\mu}^*(\tau))\right)^{\mathsf{T}}\mathbf{Q}(jT)$$
$$+ \left(\sum\nolimits_{\tau=jT}^{jT+T-1}(\boldsymbol{\gamma}^*(\tau)-\mathbf{D}^*(\tau))\right)^{\mathsf{T}}\boldsymbol{\Theta}(jT)$$
$$- V\sum\nolimits_{\tau=jT}^{jT+T-1}\sum_{u\in\mathcal{U}}\phi_u(\gamma_u^*(\tau)) \quad (57)$$

We now consider the policy comprising of the control actions $\{m_u^*(\tau)\}_{\tau=jT}^{(j+1)T-1}\ \forall\ u\in\mathcal{U}$, $\{\boldsymbol{\gamma}^*(\tau)\}_{\tau=jT}^{(j+1)T-1}$ and $\{(\mu_{hu}^*(\tau))\}_{\tau=jT}^{(j+1)T-1}$, and satisfying the following constraints:[4]

$$\frac{1}{T}\sum_{\tau=jT}^{(j+1)T-1}\left[B_{f_u}^*(m_u(\tau),\tau)-\mu_u^*(\tau)\right] < -\epsilon\ \forall\ u\in\mathcal{U} \quad (58)$$
$$\frac{1}{T}\sum_{\tau=jT}^{(j+1)T-1}\left[\gamma_u^*(\tau)-D_{f_u}^*(m_u(\tau),\tau)\right] < -\epsilon\ \forall\ u\in\mathcal{U} \quad (59)$$

where $\epsilon>0$ is arbitrary. We plug in the inequalities (58), (59) in (57) and obtain

[4]It is easy to see that such policy is guaranteed to exist provided that we allow, without loss of generality, for a virtual video layer of zero quality and zero rate, and in the assumption that, at any slot t, each user u has at least one link $(h,u)\in\mathcal{E}$ with $h\in\mathcal{N}(u)\cap\mathcal{H}(f_u)$ with peak rate lower bounded by some strictly positive number $C_{\min}$. This prevents the case where a user gets zero rate for a whole frame of length T. This assumption is not restrictive in practice since a user experiencing unacceptably poor link quality to all the helpers for a long time interval would be disconnected from the network and its streaming session halted.

$$L(\mathbf{G}((j+1)T)) - L(\mathbf{G}(jT)) - V\sum_{\tau=jT}^{jT+T-1}\sum_{u\in\mathcal{U}}\phi_u(\gamma_u(\tau)) < \mathcal{K}T^2 - \epsilon T\sum_{u\in\mathcal{U}}Q_u(jT) - \epsilon T\sum_{u\in\mathcal{U}}\Theta_u(jT) - V\sum\nolimits_{\tau=jT}^{jT+T-1}\sum_{u\in\mathcal{U}}\phi_u(\gamma_u^*(\tau)) \quad (60)$$

Also, considering the fact that for any vector $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_{|\mathcal{U}|})$ we have

$$\sum_{u\in\mathcal{U}}\phi_u(D_u^{\min}) = \phi_{\min} \leq \sum_{u\in\mathcal{U}}\phi_u(\gamma_u) \leq \phi_{\max} = \sum_{u\in\mathcal{U}}\phi_u(D_u^{\max}), \quad (61)$$

we can write:

$$L(\mathbf{G}((j+1)T)) - L(\mathbf{G}(jT)) < \mathcal{K}T^2 + VT(\phi_{\max} - \phi_{\min}) - \epsilon T\sum_{u\in\mathcal{U}}Q_u(jT) - \epsilon T\sum_{u\in\mathcal{U}}\Theta_u(jT) \quad (62)$$

Once again using (52), (53), we have:

$$L(\mathbf{G}((j+1)T)) - L(\mathbf{G}(jT)) < \mathcal{K}T^2 + VT(\phi_{\max} - \phi_{\min}) - \epsilon\sum_{\tau=jT}^{jT+T-1}\sum_{u\in\mathcal{U}}Q_u(\tau) - \epsilon\sum_{\tau=jT}^{jT+T-1}\sum_{u\in\mathcal{U}}\Theta_u(\tau) + \epsilon\kappa|\mathcal{U}|T(T-1) \quad (63)$$

Summing the above over the frames $j \in \{0, \ldots, F-1\}$ yields

$$L(\mathbf{G}((FT)) - L(\mathbf{G}(0)) < \mathcal{K}T^2F + VFT(\phi_{\max} - \phi_{\min}) - \epsilon\sum_{\tau=0}^{FT-1}\sum_{u\in\mathcal{U}}Q_u(\tau) - \epsilon\sum_{\tau=0}^{FT-1}\sum_{u\in\mathcal{U}}\Theta_u(\tau) + \epsilon\kappa|\mathcal{U}|FT(T-1) \quad (64)$$

Rearranging and neglecting appropriate terms, we get

$$\frac{1}{FT}\sum_{\tau=0}^{FT-1}\sum_{u\in\mathcal{U}}Q_u(\tau) + \frac{1}{FT}\sum_{\tau=0}^{FT-1}\sum_{u\in\mathcal{U}}\Theta_u(\tau) < \frac{\mathcal{K}T}{\epsilon} + \frac{V(\phi_{\max} - \phi_{\min})}{\epsilon} + \frac{L(\mathbf{G}(0))}{\epsilon FT} + \kappa|\mathcal{U}|(T-1) \quad (65)$$

Taking limits as $F \to \infty$

$$\boxed{\lim_{F\to\infty}\frac{1}{FT}\sum_{\tau=0}^{FT-1}\left(\sum_{u\in\mathcal{U}}Q_u(\tau) + \sum_{u\in\mathcal{U}}\Theta_u(\tau)\right) < \frac{\mathcal{K}T}{\epsilon} + \frac{V(\phi_{\max} - \phi_{\min})}{\epsilon} + \kappa|\mathcal{U}|(T-1)} \quad (66)$$

such that (28) is proved.

We now consider the policy comprising of the decisions which achieves the optimal solution ϕ_j^{opt} to the problem (46a) – (46e). Using (46b) and (46c) in (57), we have

$$L(\mathbf{G}((j+1)T)) - L(\mathbf{G}(jT)) - V\sum_{\tau=jT}^{jT+T-1}\sum_{u\in\mathcal{U}}\phi_u(\gamma_u(\tau)) \leq \mathcal{K}T + \mathcal{K}T(T-1) - VT\phi_j^{\text{opt}} \quad (67)$$

Summing this over $j \in \{0, \ldots, F-1\}$, yields

$$L(\mathbf{G}((FT)) - L(\mathbf{G}(0)) - V\sum_{\tau=0}^{FT-1}\sum_{u\in\mathcal{U}}\phi_u(\gamma_u(\tau)) \leq \mathcal{K}T^2F - VT\sum_{j=0}^{F-1}\phi_j^{\text{opt}}. \quad (68)$$

Dividing both sides by VFT and using the fact that $L(\mathbf{G}((FT)) > 0$, we get

$$\frac{1}{FT}\sum_{\tau=0}^{FT-1}\sum_{u\in\mathcal{U}}\phi_u(\gamma_u(\tau)) \geq \frac{1}{F}\sum_{j=0}^{F-1}\phi_j^{\text{opt}} - \frac{\mathcal{K}T}{V} - \frac{L(\mathbf{G}(0))}{VTF}. \quad (69)$$

At this point, using Jensen's inequality, the fact that $\phi_u(\cdot)$ is continuous and non-decreasing for all $u \in \mathcal{U}$, and the fact that the strong stability of the queues (66) implies that $\lim_{F\to\infty}\frac{1}{FT}\sum_{\tau=0}^{FT-1}\Theta_u(\tau) < \infty \ \forall\ u \in \mathcal{U}$, which in turns implies that $\overline{\gamma}_u \leq \overline{D}_u \ \forall\ u \in \mathcal{U}$, we arrive at

$$\boxed{\sum_{u\in\mathcal{U}}\phi_u\left(\overline{D}_u\right) \geq \lim_{F\to\infty}\frac{1}{F}\sum_{j=0}^{F-1}\phi_j^{\text{opt}} - \frac{\mathcal{K}T}{V}.} \quad (70)$$

such that (27) is proved.

Thus, the utility under the DPP policy is within $O(1/V)$ of the time average of the ϕ_j^{opt} utility values that can be achieved only if knowledge of the future states up to a look-ahead of blocks of T slots. If T is increased, then the value of ϕ_j^{opt} for every frame j improves since we allow a larger look-ahead. However, from (70), we can see that if T is increased, then V can also be increased in order to maintain the same distance from optimality. This yields a corresponding $O(V)$ increase in the queues backlog.

For the case where the rate function $B_f(m,t)$, the quality function $D_f(m,t)$ and the topology state $\omega(t)$is stationary and ergodic, the time average in the left hand side of (66) and in the right hand side of (70) become ensemble averages because of ergodicity. Thus, we obtain (29) and (30). Furthermore, if the network state is i.i.d., we can take $T = 1$ in the above derivation, obtaining the bounds given in Corollary 1.

REFERENCES

[1] "Cisco visual networking index: Global mobile data traffic forecast update, 2014-2019." [Online]. Available: http://goo.gl/1XYhqY

[2] "Conviva viewer experience report, 2014." [Online]. Available: http://www.conviva.com/vxr/

[3] "MPEG DASH standard." [Online]. Available: http://dashif.org/mpeg-dash

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

[5] Y. Sánchez, T. Schierl, C. Hellge, T. Wiegand, D. Hong, D. De Vleeschauwer, W. Van Leekwijck, and Y. Lelouedec, "iDASH: improved dynamic adaptive streaming over http using scalable video coding," in *ACM Multimedia Systems Conference (MMSys)*, 2011, pp. 23–25.

[6] "Google video quality report." [Online]. Available: https://www.google.com/get/videoqualityreport/

[7] "Netflix ISP speed index." [Online]. Available: http://ispspeedindex.netflix.com/

[8] V. Joseph and G. de Veciana, "Nova: Qoe-driven optimization of dash-based video delivery in networks," *arXiv preprint arXiv:1307.7210*, 2013.

[9] C. Gong and X. Wang, "Adaptive transmission for delay-constrained wireless video," *Wireless Communications, IEEE Transactions on*, vol. 13, no. 1, pp. 49–61, January 2014.

[10] A. A. Khalek, C. Caramanis, and R. W. Heath Jr, "Loss visibility optimized real-time video transmission over mimo systems," *arXiv preprint arXiv:1301.3174*, 2013.

[11] M. Zhao, X. Gong, J. Liang, W. Wang, X. Que, and S. Cheng, "Scheduling and resource allocation for wireless dynamic adaptive streaming of scalable videos over http," in *Communications (ICC), 2014 IEEE International Conference on*, June 2014, pp. 1681–1686.

[12] C. Chen, R. Heath, A. Bovik, and G. de Veciana, "Adaptive policies for real-time video transmission: A markov decision process framework," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 2249–2252.

[13] D. Bethanabhotla, G. Caire, and M. J. Neely, "Adaptive video streaming for wireless networks with multiple users and helpers," *Communications, IEEE Transactions on*, vol. 63, no. 1, pp. 268–285, Jan 2015.

[14] K. Miller, D. Bethanabhotla, G. Caire, and A. Wolisz, "A control-theoretic approach to adaptive video streaming in dense wireless networks," *Multimedia, IEEE Transactions on*, vol. 17, no. 8, pp. 1309–1322, 2015.

[15] "IEEE 802.11ac - Wikipedia," https://en.wikipedia.org/wiki/IEEE_802.11ac, Last Updated: January 14, 2014.

[16] J. Andrews, S. Buzzi, W. Choi, S. Hanly, A. Lozano, A. Soong, and J. Zhang, "What will 5G be?" *Selected Areas in Communications, IEEE Journal on*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[17] D. Bethanabhotla, O. Y. Bursalioglu, H. C. Papadopoulos, and G. Caire, "Optimal user-cell association for massive MIMO wireless networks," *Wireless Communications, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[18] "pCell white paper." [Online]. Available: http://goo.gl/lDrofd

[19] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. on Wireless Communications*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.

[20] H. Huh, G. Caire, H. Papadopoulos, and S. Ramprashad, "Achieving massive MIMO spectral efficiency with a not-so-large number of antennas," *IEEE Trans. on Wireless Communications*, vol. 11, no. 9, pp. 3226–3239, 2012.

[21] J. Hoydis, S. Ten Brink, and M. Debbah, "Massive MIMO in the UL/DL of cellular networks: How many antennas do we need?" *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 2, pp. 160–171, 2013.

[22] A. Ortega, "Variable bit-rate video coding," *Compressed Video over Networks*, pp. 343–382, 2000.

[23] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.

[24] A. F. Molisch, *Wireless communications*. Wiley, 2010, vol. 15.

[25] M. J. Neely, "Wireless peer-to-peer scheduling in mobile networks," in *2012 46th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2012, pp. 1–6.

[26] A. El Gamal and Y. Kim, "Lecture notes on network information theory," *Arxiv preprint arxiv:1001.3404*, vol. 3, 2010.

[27] M. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[28] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Transactions on Networking (ToN)*, vol. 8, no. 5, pp. 556–567, 2000.

[29] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3. IEEE, 2005, pp. 1794–1803.

[30] M. Neely, "Universal scheduling for networks with arbitrary traffic, channels, and mobility," in *Decision and Control (CDC), 2010 49th IEEE Conference on*. IEEE, 2010, pp. 1822–1829.

[31] A. Tulino and S. Verdú, *Random matrix theory and wireless communications*. Now Publishers Inc, 2004, vol. 1.

[32] R. Couillet and M. Debbah, *Random matrix methods for wireless communications*. Cambridge University Press, 2011.

[33] H. Huh, A. M. Tulino, and G. Caire, "Network MIMO with linear zero-forcing beamforming: Large system analysis, impact of channel estimation, and reduced-complexity scheduling," *Information Theory, IEEE Transactions on*, vol. 58, no. 5, pp. 2911–2934, 2012.

[34] Y. Lim, C. Chae, and G. Caire, "Performance analysis of massive MIMO for cell-boundary users," *Wireless Communications, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[35] B. Hassibi and B. M. Hochwald, "How much training is needed in multiple-antenna wireless links?" *Information Theory, IEEE Transactions on*, vol. 49, no. 4, pp. 951–963, 2003.

[36] A. Adhikary, J. Nam, J.-Y. Ahn, and G. Caire, "Joint spatial division and multiplexingthe large-scale array regime," *IEEE Trans. on Inform. Theory*, vol. 59, no. 10, pp. 6441–6463, 2013.

[37] J. Nam, A. Adhikary, J.-Y. Ahn, and G. Caire, "Joint spatial division and multiplexing: Opportunistic beamforming, user grouping and simplified downlink scheduling," *IEEE J. of Sel. Topics in Sig. Proc. (JSTSP)*, vol. 8, no. 5, pp. 876–890, 2014.

[38] A. Adhikary, E. Al Safadi, M. K. Samimi, R. Wang, G. Caire, T. S. Rappaport, and A. F. Molisch, "Joint spatial division and multiplexing for mm-wave channels," *IEEE J. on Sel. Areas on Commun. (JSAC)*, vol. 32, no. 6, pp. 1239–1255, 2014.

[39] H. Yin, D. Gesbert, M. Filippou, and Y. Liu, "A coordinated approach to channel estimation in large-scale multiple-antenna systems," *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 2, pp. 264–273, 2013.

[40] S. Wagner, R. Couillet, M. Debbah, and D. Slock, "Large system analysis of linear precoding in correlated miso broadcast channels under limited feedback," *Information Theory, IEEE Transactions on*, vol. 58, no. 7, pp. 4509–4537, 2012.

[41] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser mimo systems," *Communications, IEEE Transactions on*, vol. 61, no. 4, pp. 1436–1449, 2013.

[42] T. Yoo and A. Goldsmith, "On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 3, pp. 528–541, 2006.

[43] http://media.xiph.org/video/derf/.

[44] "The SSIM Index for Image Quality Assessment." [Online]. Available: http://goo.gl/ngR0UL