Multicast Scheduling for Relay-Based Heterogeneous Networks Using Rateless Codes

Chao Chen and Seung Jun Baek, *Member, IEEE*

Abstract—We consider the multicast scheduling problem in the heterogeneous network using a half-duplex relay station (RS). Our goal is to minimize the delay of transmitting a block of packets to users over time-varying channels using rateless codes. Due to half-duplex operation, at each time slot, the RS can choose to either multicast a packet to the users, or fetch a packet from the macro base station. We formulate a fluid relaxation for the optimal decision problem, and reveal that the optimal policy has a threshold-based structure so as to exploit the opportunism of multicast channel: the RS should multicast only when the channel quality is sufficiently "high". We propose an online policy based on the relaxation which does not require the knowledge of channel distribution. When the channel distribution is symmetric across users, we provide a closed-form expression of the asymptotic performance of our policy. For two-user systems, we prove that our scheme is asymptotically optimal. When the users' channels are independent, we derive a performance bound based on water-filling rate allocation which approximates the optimal policy well. Simulation results show that our scheme performs close to theoretical bounds, under correlated as well as independent fading channels.

Index Terms—Relay networks, rateless codes, opportunistic scheduling, fluid approximation, asymptotic optimality

1 Introduction

For the purpose of throughput enhancement and coverage extension, the relay architecture is adopted as a key component in mobile communication systems, e.g., IEEE 802.16j and 3GPP LTE-Advanced [1], [2]. Both standards propose *two-hop* relay networks for the sake of simplicity and explicitness of system design [1]. Relay nodes are categorized as either half-duplex or full-duplex, depending on whether they can simultaneously transmit and receive information. While there exist many practical challenges in deploying full-duplex relays, e.g., the potential debilitating effects of self-interference, half-duplex relays have been used in most contemporary communication systems [3]. In this paper, we study multicast scheduling algorithms in the heterogeneous network using a half-duplex relay station. Recently, multicasting in cellular networks has drawn much attention, e.g., evolved multimedia broadcast/multicast services (eMBMS) [4], fueled by fast-growing demand for multimedia services for mobile subscribers. For example, the live streaming service by Twitch has attracted 100 million viewers per month, as of 2015; Twitch ranked 4th in U.S. in terms of the peak Internet traffic in 2014 [5].

As shown in Fig. 1, our model consists of a macro base station (BS), a relay station (RS) and wireless users associated with the RS. An information block consisting of K packets is to be delivered to all the users. The packets are encoded by rateless codes [6], i.e., an unlimited number of coded packets can be generated from the information block. We assume that the coded packets are infinitely backlogged at the BS. Once a user receives approximately K coded packets, the user can decode the original information block. Some well-known rateless codes are random linear network coding (RLNC) [7], LT codes [8] and Raptor codes [9].

The RS performs a multicast as follows. The RS transmits a coded packet on a single time-frequency resource in order to exploit the broadcast nature of wireless channels. Thus, every user can "see" the packet; however, the users' channels are assumed to be time-varying, e.g., some users may undergo deep fading, and their instantaneous rates seen on the transmission resource may not be high enough for the packet. Thus, only a subset of users can successfully receive the packet; we assume channel state information is available at transmitter (CSIT), hence the subset is known to the RS prior to multicasting. Due to the half-duplex operation, at each time slot, the RS must decide either to multicast a packet to the users who can successfully receive the packet, or to fetch a packet from the BS. Our goal is to make optimal scheduling decisions to minimize the average *decoding delay* defined as the number of time slots until every user receives K coded packets. Potential applications would be file distribution, software updates, news/alerts notification, and multicast video streaming similar to MPEG-DASH [10]. In such streaming methods, the video source is divided into multiple chunks called *segments* each of which consists of up to several hundred packets. A segment can be regarded as a block to be disseminated to users. Typically, the next segment can be requested from the video server only if the preceding segment is completely delivered to users. Thus, assuming each block is a file consisting of K packets over which erasure coding is applied, the reduction of segment download delay, i.e., decoding delay, will enhance the multicast throughput.

- *C. Chen is with the School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China. E-mail: eckio_491@163.com.*
- *S. Baek is with the College of Information and Communications, Korea University, Seoul 02841, Korea. E-mail: sjbaek@korea.ac.kr.*

Manuscript received 24 Aug. 2016; revised 4 Feb. 2017; accepted 22 Mar. 2017. Date of publication 27 Mar. 2017; date of current version 29 Sept. 2017. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TMC.2017.2687928

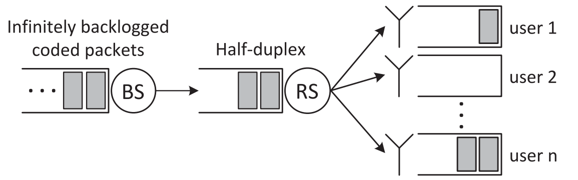

Fig. 1. System model.

Because of the time-varying channel capacity and the half-duplex operation of the RS, there exists an intrinsic tradeoff between multicasting versus fetching. Specifically, in order for the RS to multicast packets for a certain number of time slots, as many time slots must be spent in fetching new packets from the BS. Meanwhile, the number of users who can receive a packet free of error by a single multicast changes over time. Thus, it is desirable that the RS multicasts only if that number is sufficiently large so as to exploit opportunism, and fetch a new packet from the BS otherwise. By contrast if we choose to greedily multicast whenever there are packets in the RS queue, the RS queue will eventually run out of "innovative" packets with respect to those the users already have received. In such case, the RS may be forced to fetch new packets from the BS, even when the multicast channel is in "good" conditions, which results in the loss of multicast opportunities. Therefore, it is of key importance to find a good balance between multicasting and fetching so as to capture the opportunism of the wireless multicast channel. However, it proves to be difficult to find the optimal scheduling, especially when the users' channel distribution is asymmetric and/or correlated. The design of opportunistic multicast scheduler using rateless codes over a multihop network has not been well explored yet.

1.1 Related Work

There have been extensive studies on the use of rateless codes for single-hop multicast/broadcast networks. The studies [11] and [12] investigate the asymptotic throughput performance of rateless codes over single-hop broadcast erasure channel. The authors provide tight bounds on the maximum achievable throughput as a function of coding window size and the number of users. Multicast scheduling policies for single-hop cognitive radio networks are studied in [13], which incorporates two types of multiuser diversity: one among users in a multicast group, and the other among multicast groups. Note however, in[11], [12], [13], the authors mainly focus on the case where the users' channels are symmetric, while our work consider the general case of asymmetric channel distributions. The studies [14], [15], [16] examine the queueing aspects of the coded transmissions. In [14], the authors show that coding schemes can achieve a low queueing delay for lightly loaded systems, by adapting the coding window size to the number of stored packets at the source node. A queueing model associated with multiple multicast flows is considered in [15]. The capacity region of a coding scheme which operates across separate queues is characterized, which is shown to be larger than that of uncoded schemes. The work [16] introduces a virtual queue consisting of packets which are received only by partial users, and proposes scheduling policies to maximize the multicast throughput under stability constraints. Recent works [17], [18] study efficient coded multicast scheduling policies over a single-hop network. The BS selects the mutlicast rate by adjusting modulation and coding scheme (MCS), where the key tradeoff is between multiuser diversity and multicast gain. By contrast, we focus on two-hop networks with a fixed MCS at the BS; the key tradeoff is between the gains associated with multicast versus transport (fetch). Studies on multicast scheduling under deadline constraints for real-time traffic include [19], [20]. Both works propose adaptive coding schemes where [19] adopts a scheme with a varying coding window size, and the scheme in [20] adaptively chooses the encoding packets based on their priorities.

Next we point out several studies on multicasting over multihop networks. In [21], the authors show that RLNC achieves the packet-level capacity defined as the maximum reliable transmission rate in packets per unit time, for both unicast and multicast connections in lossy packet networks. However, they assume that every link has a fixed schedule for packet transmissions, which differs from our work. A recent work [22] studies optimal scheduling using half-duplex relays. However, they only focus the unicast scenario where two users communicate with each other via the relay, while our work deals with the multicast scenario. In [23], a scheduling policy for coded packet transmission in a cooperative network is proposed. However, the authors consider heuristics based on the dynamic programming which becomes computationally intractable when the number of users grows large. A recent work [24] considers multicast scheduling policies for two-hop OFDMA relay networks. Unlike our work, the authors do not consider coding, and pursue a different goal of maximizing aggregate multicast flow under proportional fairness (PF) model.

1.2 Contributions and Paper Organization

In this paper we investigate channel-aware multicast scheduling algorithms to minimize the decoding delay. Below we summarize our contributions.

(a) (Sections 2, 3, and 4) We formulate our problem as a Markov decision process (MDP) which however turns out to be intractable. Instead, we construct and investigate its fluid relaxation. By examining the fluid problem, we observe that the optimal strategy is a *threshold-based* policy such that the RS should multicast only if the revenue (defined later) exceeds a certain threshold.

(b) (Section 5) We show that even solving the fluid problem is hard because of, e.g., large problem space and requirement of full knowledge of channel distribution. We propose an online scheduling policy which periodically solves a fluid problem based on the system states and translates the solution into a feasible policy in the discrete-time domain. Our policy does not require prior knowledge of channel distribution.

(c) (Section 6) We extensively analyze the asymptotic performance of our policy. We reveal that the buffer trajectories are piecewise linear under fluid scaling. When the channel distribution is symmetric among users, we provide a closed form expression of the asymptotic delay incurred under our policy. We also prove that, our policy is asymptotically optimal for two-user systems with arbitrary channel distributions. For the important case where the users' channels are independent, we obtain a simple bound of the achievable delay under our policy. We compute the bound from a rate allocation based on water-filling. In addition, we derive a sufficient condition under which our policy is asymptotically optimal for systems of arbitrary size.

(d) (Section 7) We evaluate the performance of our policy under scenarios in which the users undergo correlated as well as independent Rayleigh fading. In all cases we demonstrate that our scheme significantly reduces the decoding delay compared to other baseline schemes such as Automatic Repeat-reQuest (ARQ), and performs close to theoretical lower bounds.

We conclude the paper in Section 8.

2 System Model

We use boldface letters to denote vectors and matrices. The ith entry of vector $\boldsymbol{x}$ is denoted by x_i. All the vectors are assumed to be column vectors, unless stated otherwise. $\boldsymbol{x}^T$ denotes the transpose of $\boldsymbol{x}$. We define $[x]^+ := \max\{x, 0\}$, and $a \wedge b := \min\{a, b\}$.

There are n users in the system. Let $I = \{1, \ldots, n\}$ denote the set of user indices. The original information block, which is to be disseminated to all the users, consists of K packets. A packet is represented as a vector of length ν over a finite coding field $\mathbb{F}_d$, and thus has the length of $\nu\lceil\log_2 d\rceil$ bits. An infinite number of coded packets are generated from those K original packets using rateless codes, and they are backlogged at the BS queue initially. The RS queue is initially empty. For simplicity, we assume that a user can recover the original information block once it receives any K coded packets, and the coding overhead is negligible [15].

We consider a time-slotted system, where the integer l denotes the time index. At each time slot, the half-duplex RS can either multicast a coded packet to users, or fetch a coded packet from the BS. The RS transmits a coded packet over a single time-frequency resource, e.g., a resource block (RB), where the packet is encoded according to a fixed modulation and coding scheme (MCS). We assume that the users' channels are time-varying as follows. The random process $C_i(l) \in \{0, 1, 2, \ldots\}$ denote the *potential* number of bits that user i can receive from the RS on the transmission resource at time slot l. Note that user i can receive a packet only if $C_i(l)$ is at least the packet size. We assume that $C_i(\cdot)$ varies independently over time slots, e.g., the users undergo fast fading. However, the joint distribution of $C_1(\cdot), \ldots, C_n(\cdot)$ at a time slot can be an arbitrary distribution function on $\mathbb{Z}_+^n$, i.e., the users' channels can be arbitrarily correlated.

A user can receive a packet only if the user's instantaneous rate is high enough for the packet, as follows. Let us define binary channel state processes $X_i(l) := \mathbf{1}(C_i(l) \geq \nu\lceil\log_2 d\rceil)$ for all $i \in I$, where $\mathbf{1}(\cdot)$ denotes the indicator function. We assume for simplicity that user i can successfully receive a packet with probability 1, if $X_i(l) = 1$. Consequently, the multicast channel is naturally modeled as a collection of "ON-OFF" channels. We say that the channel state of user i is ON (resp. OFF) at time slot l, if $X_i(l) = 1$ (resp. $X_i(l) = 0$). We assume that the CSI is available at the transmitter, i.e., $X_i(\cdot), i \in I$, are known to the RS prior to the transmission at each time slot. Thus, the RS can make online scheduling decisions based on the current channel state and the cumulative number of packets received by the users which can be tracked over time due to the availability of CSI.

TABLE 1
Summary of Main Notations

K	number of packets in a block
n	number of users
I	$:= \{1, \ldots, n\}$, set of user indices
p_i	probability that user i's channel is in ON state
s	channel state
$\langle s_i \ldots s_n\rangle$	binary representation of channel state s
S	$:= \{0, \ldots, 2^n - 1\}$, set of channel states
$p(s)$	probability that the channel state is s
$\boldsymbol{w}$	$:= (w_1, \ldots, w_n)$, vector of users' weights
$\rho(\boldsymbol{w}, s)$	$:= \sum_{i\in I} s_i w_i$, revenue associated with channel state s

We refer to the channel between the BS and the RS as the wireless *backhaul*. The backhaul is assumed to be always in ON state. This assumption is reasonable, because the capacity of the backhaul is typically high, e.g., the RS can be located in the line-of-sight of the BS.

For notational convenience, we will use n-bit unsigned integers to denote the collections of channel states of n users. Suppose s is an n-bit unsigned integer. Let $\langle s_1 \ldots s_n\rangle$ denote the binary representation of s where $s_i \in \{0, 1\}$ denotes the ith bit of s. We have $s = \sum_{i\in I} s_i 2^{n-i}$, hence s_1 is the most significant bit. We will use $s = \langle s_1 \ldots s_n\rangle$ to represent a collection of channel states such that $s_i = 1$ or 0 represents the channel state of user i is ON or OFF respectively. For example, suppose $n = 3$. Then the channel state $s = \langle 110\rangle = 6$ represents that user 1 and 2's channels are in ON state. We define the set of channel states S as a collection of n-bit unsigned integers, i.e., $S = \{0, 1, \ldots, 2^n - 1\}$. For $s \in S$, we define $p(s) := \mathbb{P}(X_i(l) = s_i, i \in I)$ as the probability that the aggregate channel state is s. Let $p_i := \mathbb{P}(X_i(l) = 1)$ denote the probability that the channel state of user i is ON.

We assume that, for simplicity, once a packet is multicast, it is dropped from the RS queue.[1] The goal of the RS is to make optimal scheduling decisions, i.e., either to multicast or to fetch, at each time slot in order to minimize the decoding delay defined as the number of time slots it takes until all the users receive at least K packets. Table 1 is a summary of the main notations used in this work.

2.1 MDP Formulation

Our problem can be formulated as an MDP as follows. We define the system state as a vector $\boldsymbol{a} := (a_1, \ldots, a_n, a_R) \in \{0, 1, 2, \ldots\}^{n+1}$, where a_i is the number of packets received by user i for $i \in I$, and a_R is the number of packets backlogged at the RS. The RS can take either of the following actions: multicasting or fetching, based on channel state s and system state $\boldsymbol{a}$, which is denoted by $u(s, \boldsymbol{a})$. The value function $V(\boldsymbol{a})$ is the mean decoding delay under the optimal policy when the initial system state is $\boldsymbol{a}$. The transition probability from system state $\boldsymbol{a}$ to $\boldsymbol{a}'$ under action $u(s, \boldsymbol{a})$ is denoted by $P_{u(s,\boldsymbol{a})}(\boldsymbol{a}, \boldsymbol{a}')$. The cost function is given by $R(\boldsymbol{a}) := \mathbf{1}\ (\min_{i\in I}\{a_i\} < K)$. The Bellman equation associated with our problem is

1. The conventional way to reuse the previously multicast packets is the ARQ, such that the RS keeps multicasting every packet until all the users receive a copy. We will show that our scheme outperforms the ARQ scheme later via simulations.

$$V(\boldsymbol{a}) = R(\boldsymbol{a}) + \min_{u(s,\boldsymbol{a})} \left\{ \sum_{\boldsymbol{a}'} P_{u(s,\boldsymbol{a})}(\boldsymbol{a}, \boldsymbol{a}') V(\boldsymbol{a}') \right\}. \quad (1)$$

We can solve (1) using value iteration. However, since $\boldsymbol{a} \in \{0, 1, 2, \dots\}^{n+1}$ and $|S| = 2^n$, (1) quickly becomes computationally intractable with increasing n.

3 Fluid Approximation

In this section we construct the *fluid model* as an approximation to the MDP. The fluid model has been widely adopted as a deterministic relaxation of the original stochastic problem [25]. When the timescale of the duration of system observation is significantly larger than that of system dynamics, fluid model is a good approximation of the original MDP, i.e., the solution to fluid model well-approximates the relative value function of the associated MDP [25]. In our problem, we will assume that the block size K is sufficiently large, thus the timescale of policy execution (scheduling) is smaller than that of block dissemination. After deriving an idealized fluid control policy by solving the fluid approximation, we will refine the policy to account for the original stochastic network via a standard method outlined in [26].

We first introduce stochastic processes for our model. Let $G_s(l)$ (resp. $\Psi_s(l)$) denote the cumulative number of time slots at which the RS performed a multicast (resp. a fetching) and the channel was in state $s \in S$ up to time slot l. It is clear that $\sum_{s\in S}[G_s(l) + \Psi_s(l)] = l$. Let $B_i(l)$ be the number of packets received by user i up to time slot l. We have that

$$B_i(l) = \sum_{s\in S} G_s(l) s_i.$$

Recall that $s_i = 1$ if in channel state s user i will successfully receive. Let $B_R(l)$ be the queue length, i.e., the number of packets, at time slot l at the RS. We have that

$$B_R(l) = \sum_{s\in S} [\Psi_s(l) - G_s(l)] \geq 0.$$

Next we discuss fluid scaling of the related processes. Let $t \geq 0$ be the continuous time index. Let $\tilde{Y}(t)$ denote the continuous time process associated with $Y(\cdot)$ defined as $\tilde{Y}(t) := Y(\lfloor t \rfloor)$. Consider a sequence of systems indexed by the scaling parameter $m = 1, 2, \dots,$ as follows. In the mth system, the block size is scaled from K to mK. For a random process $\tilde{Y}(t)$, let $\tilde{Y}^{(m)}(t)$ denote the process analogous to $\tilde{Y}(t)$ in the mth system. Let us denote the following scaling of $\tilde{Y}^{(m)}(t)$ by $y^{(m)}(t)$

$$y^{(m)}(t) := \frac{1}{m} \tilde{Y}^{(m)}(mt). \quad (2)$$

For example, $\psi_s^{(m)}(t)$ is the scaling of the process $\tilde{\Psi}_s^{(m)}(t)$.

Lemma 1. *The scaled processes satisfy the following convergence uniformly over compact sets (u.o.c.) as $m \to \infty$*

$$\begin{aligned}
&g_s^{(m)}(t) \to g_s(t),\\
&\psi_s^{(m)}(t) \to \psi_s(t),\\
&b_R^{(m)}(t) \to b_R(t) = \sum_{s\in S}[\psi_s(t) - g_s(t)],\\
&b_i^{(m)}(t) \to b_i(t) = \sum_{s\in S} g_s(t) s_i, \forall i \in I,
\end{aligned}$$

where $g_s(t)$, $\psi_s(t)$, $b_R(t)$ and $b_i(t)$ are Lipschitz continuous on $[0, \infty)$.

Proof. The proof is similar to Lemma 1 in [27]. □

Hence $g_s(t)$, $\psi_s(t)$, $b_R(t)$ and $b_i(t)$ are absolutely continuous and have derivatives almost everywhere for $t \geq 0$. The points at which the derivatives exist are called *regular* points [27]. The derivative of some function $y(t)$ at regular point t is denoted by $\dot{y}(t)$.

We are interested in the asymptotic performance, thus we will properly redefine the "decoding delay". In the mth system, the users need to receive mK packets. Hence, under fluid scaling (2), we would like to minimize the time until every user receives K units of fluid. Accordingly, we will define the *asymptotic decoding delay* as follows:

$$T = \inf_t \left\{ t \in \mathbb{R}_+ \middle| \min_{i\in I}\{b_i(t)\} = K \right\}. \quad (3)$$

We formulate a deterministic *fluid problem* in order to minimize the asymptotic decoding delay as follows:

$$\begin{aligned}
\textbf{(F)} \quad & \underset{g_s(t), \psi_s(t)}{\text{minimize}} \quad T = \inf_t \left\{ t \in \mathbb{R}_+ \middle| \min_{i\in I}\{b_i(t)\} = K \right\} && (4)\\
& \text{subject to} \quad b_i(0) = b_R(0) = 0, \forall i \in I,\\
& \qquad b_i(t) = \sum_{s\in S} g_s(t) s_i, \forall i \in I, && (5)\\
& \qquad b_R(t) = \sum_{s\in S}[\psi_s(t) - g_s(t)] \geq 0, && (6)\\
& \qquad g_s(t) + \psi_s(t) = p(s)t, \forall s \in S, && (7)\\
& \qquad g_s(t) \geq 0, \psi_s(t) \geq 0, \forall s \in S. && (8)
\end{aligned}$$

$g_s(t)$ and $\psi_s(t)$ are policies associated with the fluid problem, which we call *fluid* policies. We will show that there exist optimal *stationary* fluid policies to **(F)**.

Theorem 1. *For each $s \in S$, there exists $\pi(s) \in [0,1]$ such that the following fluid policies $g_s(t)$ and $\psi_s(t)$ are optimal for* **(F)**: *for each $s \in S$*

$$g_s(t) = p(s)\pi(s)t, \psi_s(t) = p(s)[1 - \pi(s)]t. \quad (9)$$

Proof. See Appendix A. □

From the definitions of $g_s(t)$ and $\psi_s(t)$, and from (9) and (33) in Appendix A, $\pi(s)$ (resp. $1 - \pi(s)$) is interpreted as the optimal fraction of time the RS spent on multicasting (resp. fetching) conditional on that the channel state is s. Policies (9) are called stationary since $\dot{g}_s(t)$ and $\dot{\psi}_s(t)$ are constant over time. Based on Theorem 1, we will henceforth consider only stationary policies without loss of generality. Let $r_i(t) := \dot{b}_i(t)$ be the asymptotic service rate of user i's buffer, which we call the *fluid rate* achieved by user i. Under stationary policies, buffer trajectories $\{b_i(t), i \in I\}$ evolve as straight lines, thus fluid rates $\{r_i(t), i \in I\}$ are constant over time. Hence, if we drop the time index from $r_i(t)$

$$r_i = \dot{b}_i(t) = \sum_{s\in S} p(s)\pi(s)s_i, \forall i \in I.$$

Under policies (9), constraint (6) is given by

$$\sum_{s\in S} p(s)\pi(s) \leq \frac{1}{2}.$$

Since $b_i(t) = r_i t$, (3) is given by $T = K/\min_{i\in I}\{r_i\}$. Thus, our goal reduces to maximizing $\min_{i\in I}\{r_i\}$. In summary, **(F)** is equivalent to **(P)** defined as

$$\begin{aligned} (\mathrm{P}) \quad & \underset{\pi}{\text{maximize}} \quad \min_{i\in I}\{r_i\} \\ & \text{subject to} \quad \sum_{s\in S} p(s)\pi(s) \le \frac{1}{2}, \end{aligned} \tag{10}$$

$$r_i = \sum_{s\in S} p(s)\pi(s)s_i, \forall i \in I, \tag{11}$$

$$\pi(s) \in [0,1], \forall s \in S. \tag{12}$$

(P) is a linear program (LP) where the variables are $\pi(\cdot)$. Since $g_s(t)$ and $\psi_s(t)$ are determined by $\pi(s)$ for all $s \in S$, we may regard $\pi(\cdot)$ as a fluid policy. We will henceforth deal with $\pi(\cdot)$ only. From the viewpoint of the original stochastic network, one can interpret $\pi(s)$ as the *probability* of multicasting at a time slot when the channel state is s, and $1-\pi(s)$ as the probability of fetching. The constraint (10) represents flow conservation in stationarity, i.e., the outflow rate of fluid from the RS queue cannot exceed the inflow rate to the RS queue. In the following section, we will consider the optimal fluid policy for **(P)**.

4 Optimal Fluid Policy

4.1 Formulation as a Max-Weight Problem

We will investigate the properties of the solution to **(P)**. It is difficult to directly solve **(P)**, since the number of variables in $\pi(\cdot)$ is 2^n. We first show that, **(P)** can be formulated as a max-weight problem. Define $\boldsymbol{w} := (w_1, \ldots, w_n)$ as the weight vector for the users. Consider the following LP:

$$\begin{aligned} (\mathrm{MW}) \quad & \underset{\pi}{\text{maximize}} \sum_{i\in I} w_i r_i \\ & \text{subject to} (10) - (12). \end{aligned}$$

Theorem 2. *There exist nonnegative weights w_i, $i \in I$, such that **(P)** is equivalent to **(MW)**.*

Proof. See Appendix B. □

We will interpret w_i as the *revenue* earned by user i from each reception of a packet. Let us define $\rho(\boldsymbol{w}, s) := \sum_{i\in I} w_i s_i$ as the total revenue earned by the system when the RS multicasts under channel state s. If we express the objective of **(MW)** in terms of $\pi(\cdot)$, we have that

$$\sum_{s\in S} p(s)\pi(s)\rho(\boldsymbol{w}, s). \tag{13}$$

Note that (13) represents the expected revenue under the scheduling policy $\pi(\cdot)$. Thus, we can regard **(MW)** as a problem of revenue maximization.

We will show that, the optimal policy is a *threshold-based* policy based on the revenue earned by the system. Let us introduce the following notations which will be useful for further discussion. For given $\boldsymbol{w}$, let $\eta(k) \in S$ denote the channel state such that, the revenue associated with $\eta(k)$ is kth largest (ties are arbitrarily broken) among all $\{\rho(\boldsymbol{w}, s), s \in S\}$. Also define k^* as

$$k^* := \min\left\{k \in \{1, \ldots, |S|\} \,\middle|\, \sum_{j=1}^{k} p(\eta(j)) > \frac{1}{2}\right\}. \tag{14}$$

Theorem 3. *The policy $\pi^*(\cdot)$ is optimal for **(MW)***

$$\pi^*(s) = \begin{cases} 1, & \rho(\boldsymbol{w}, s) > \xi, \\ \beta, & \rho(\boldsymbol{w}, s) = \xi, \\ 0, & \text{otherwise}, \end{cases} \tag{15}$$

where

$$\xi := \rho(\boldsymbol{w}, \eta(k^*)), \beta := \frac{\frac{1}{2} - \sum_{s:\rho(\boldsymbol{w},s)>\xi} p(s)}{\sum_{s:\rho(\boldsymbol{w},s)=\xi} p(s)}. \tag{16}$$

Proof. See Appendix C. □

We provide some intuitions behind the optimal policy. Let us interpret the optimal fluid policy in the original stochastic network. Suppose the channel state is $s \in S$ at a time slot. The optimal policy for the RS is to multicast with probability (w.p.) 1, or $\pi^*(s) = 1$, if the revenue associated with s or $\rho(\boldsymbol{w}, s)$ exceeds the threshold ξ; if $\rho(\boldsymbol{w}, s)$ is exactly ξ, the RS performs a multicast w.p. β and a fetching w.p. $1-\beta$; if $\rho(\boldsymbol{w}, s)$ is less than ξ, the RS performs a fetching. The policy shows the opportunistic nature of the scheduling as follows. As mentioned earlier, there exists a tradeoff between fetching a new packet over the backhaul in order to replenish the RS queue, versus exploiting the opportunism of the time-varying multicast channel. According to the policy, the RS should multicast only when the channel condition is sufficiently favorable, i.e., when the system earns a high revenue. In other words, one can regard the revenue as a measure of channel quality. Our result explicitly provides the optimal point of the tradeoff between fetching versus multicasting.

We can solve **(P)**, if we can find the optimal weights and solve the associated **(MW)** instead. We define a *symmetric-user system* as a system in which users' channels are symmetric, i.e., the distribution is invariant to permutation of the users, which implies that $p(s) = p(\sigma)$ for all channel states $s, \sigma \in S$ such that $\|s\| = \|\sigma\|$, where $\|s\| := \sum_{i=1}^{n} s_i$ denotes the number of ones in the binary representation of s. By symmetry, it is easily seen that the optimal weights for Theorem 2 are identical, i.e., we may assume $\boldsymbol{w} = 1_n$, where 1_n represents the n-dimensional vector of ones. For such $\boldsymbol{w}$ the solution to **(MW)** is identical to **(P)** as follows.

Corollary 1. *For symmetric-user systems, the following policy is optimal for **(P)***

$$\pi^*(s) = \begin{cases} 1, & \|s\| > \xi, \\ \beta, & \|s\| = \xi, \\ 0, & \text{otherwise}, \end{cases}$$

where ξ and β are as defined in Theorem 3.

In symmetric-user systems, interpretation of the revenue is simple: it is proportional to the number of users' channels which are in ON state, i.e., $\|s\|$. Hence the RS should multicast only if the number of ON channels exceeds certain threshold.

4.2 Hardness

In general, it is hard to solve **(MW)** due to the following reasons: first, it requires the full knowledge of channel distribution, i.e., $p(s), s \in S$, which may be difficult in practice; second, it seems to be hard to find optimal $\boldsymbol{w}$ in general for given $p(s), s \in S$; third, it turns out that **(MW)** is hard to solve even if the optimal $\boldsymbol{w}$ is known. Specifically, evaluating ξ turns out to be hard, which makes **(MW)** NP-hard as follows.

Theorem 4. *(MW) is NP-hard.*

Proof. We first introduce an NP-hard problem called the kTH LARGEST SUBSET [28]. For finite multiset A, define $\|A\| := \sum_{a \in A} a$ as the sum-element of A.

PROBLEM: kTH LARGEST SUBSET.

INSTANCE: Finite multiset A, positive integers k and c.

QUESTION: Are there k or more distinct subsets $B \subseteq A$ such that $\|B\| \leq c$?

We will refer to the sum-element of a subset as a *subset-sum*. **(MW)** is related to kTH LARGEST SUBSET as follows. Let $A = \{w_1, \ldots, w_n\}$. For channel state $s = \langle s_1 \ldots s_n \rangle$, define the associated multiset $B_s := \{w_i | s_i = 1, i \in I\}$. Clearly, B_s is a subset of A, and we have $\|B_s\| = \sum_{i \in I} w_i s_i = \rho(\boldsymbol{w}, s)$. Thus, the set of all possible revenues $\{\rho(\boldsymbol{w}, s), s \in S\}$ is equivalent to the set of all the subset-sums of A. Consequently, finding the kth largest subset-sum of A is equivalent to finding the kth largest revenue in **(MW)**. Note that, in order to compute $\pi^*(\cdot)$, one needs to evaluate ξ which is the k^*th largest revenue; see (16). Accordingly, one can show that there exists a polynomial-time reduction from kTH LARGEST SUBSET to **(MW)**. Due to space constraints, the detailed proof is relegated to [29]. □

5 PROPOSED POLICY

Our goal is to propose an online scheduling policy that achieves low decoding delay. We will (i) propose a method to solve **(MW)** approximately; (ii) propose an online scheduling policy based on the approximation, which does not require the knowledge of channel distributions; (iii) prove our policy is optimal in certain asymptotic sense.

5.1 Weight-Based Priority (WBP) Rule

Let us first examine the optimal policy to **(MW)**. Channel states $s \in S$ are ranked in the order of revenues generated by the states, or $\rho(\boldsymbol{w}, s)$. Specifically, given two channel states $s, \sigma \in S$, we say that *s precedes σ* if $\rho(\boldsymbol{w}, s) > \rho(\boldsymbol{w}, \sigma)$ holds. That is, if s generates more revenue than σ, we treat s as the "better" channel state than σ. Under the optimal policy, if channel state s precedes certain threshold, the channel is regarded to be in "good" condition, in which case the RS performs a multicast so as to exploit opportunism.

Next we introduce the WBP rule which is a simplified precedence rule for channel states. Without loss of generality, assume $w_1 \geq \cdots \geq w_n$ in this section. Consider two channel states $s = \langle s_1 s_2 \ldots s_n \rangle \in S$ and $\sigma = \langle \sigma_1 \sigma_2 \ldots \sigma_n \rangle \in S$ such that $s \neq \sigma$. Let $l^* := \min\{l : s_l \neq \sigma_l\}$ be the first bit position in which s differs from σ. In WBP, we say s precedes σ if $s_{l^*} > \sigma_{l^*}$, i.e., $s_{l^*} = 1$ and $\sigma_{l^*} = 0$. Equivalently, WBP is a precedence rule such that

$$\boxed{\begin{array}{l} s \text{ precedes } \sigma \text{ if } s > \sigma, \text{where we compare } s \\ \text{and } \sigma \text{ in terms of the sizes of unsigned} \\ \text{binary integers they represent.} \end{array}} \tag{17}$$

For example, suppose there are three users. Consider channel states $s = \langle 100 \rangle = 4$ and $\sigma = \langle 011 \rangle = 3$. Since as integer values $4 > 3$, we have $s > \sigma$. Under the WBP rule, s is the "better" channel state than σ. Hence, when we assess the quality of channel states under the WBP rule, *users associated with larger weights are given higher priority*. Note that the WBP rule may not be the optimal channel assessment, in the above example if $w_1 = 3$ and $w_2 = w_3 = 2$, the revenue (sum-weight) associated with s and σ are given by 3 and 4, in which case σ is the better channel.

We will approximately solve **(MW)** using the WBP rule as follows. Consider the following problem:

$$\begin{aligned} (\text{P-WBP}) \quad & \underset{\pi}{\text{maximize}} \quad \sum_{i \in I} w_i r_i \\ & \text{subject to} \quad (10) - (11), \\ & \pi(s) = \begin{cases} 1, & s > \xi, \\ \beta, & s = \xi, \\ 0, & s < \xi, \end{cases} \\ & \beta \in [0, 1), \xi \in S. \end{aligned} \tag{18}$$

Due to (18), **(P-WBP)** is identical to **(MW)** except that $\pi(\cdot)$ is constrained to be a threshold-based policy under the WBP rule, i.e., the RS should multicast w.p. 1 only if the current channel state, as an unsigned binary integer, exceeds ξ. We only need to determine threshold parameters ξ and β. The objective of **(P-WBP)** is monotonically decreasing (resp. increasing) in ξ (resp. β), hence we can simply minimize ξ (resp. maximize β) under the flow constraints (10) and (11). Hence, unlike the revenue-based precedence rule optimal for **(MW)**, the solution for **(P-WBP)** does not depend on the absolute values of $w_1, \ldots, w_n$, but only on their relative order. Later, we will discuss how **(P-WBP)** can be solved efficiently and adaptively without the knowledge of $p(s), s \in S$.

5.2 Modified Fluid Problem

Here we propose a deterministic, continuous-time fluid problem called **(F-WBP)**, which is modified from **(F)**. The following is an outline of **(F-WBP)**.

(a) The system is reviewed at the beginning of every constant interval which is called a *review period*;

(b) At each review period we maximize the expected revenue earned by the system under the WBP rule;

(c) The weight of each user is set to the remaining amount of fluid to be received.

Define a sequence of time intervals of length ϵ in $\mathbb{R}$

$$\mathcal{J}_k := [k\epsilon, (k+1)\epsilon), \quad k = 0, 1, 2, \ldots. \tag{19}$$

$\mathcal{J}_k$ is called kth review period in fluid regime. Then **(F-WBP)** for $t \in \mathcal{J}_k$, $k = 0, 1, 2, \ldots$, can be defined as follows:

$$\begin{aligned} (\text{F-WBP}) \quad & \underset{\pi}{\text{maximize}} \quad \sum_{i \in I} w_i(k\epsilon) r_i(t) \\ & \text{subject to} \quad (5) - (8), \\ & w_i(k\epsilon) = [K - b_i(k\epsilon)]^+, i \in I, \end{aligned} \tag{20}$$

$$r_i(t) = \dot{b}_i(t), i \in I, \tag{21}$$

$$b_i(k\epsilon) = \begin{cases} 0, & k = 0, \\ \lim_{\tau \uparrow k\epsilon} b_i^*(\tau), & k = 1, 2, \dots, \end{cases} \tag{22}$$

Policy is threshold–based under the WBP rule.

In **(F-WBP)** we use the same notation for fluid trajectories as **(F)**, e.g., $b_i(t)$. In (22), $b_i^*(t)$ denotes the optimal trajectory to **(F-WBP)** for user i's buffer. Hence the initial state for $b_i(t)$, $t \in \mathcal{J}_k$, is given by the terminal state of $b_i^*(t)$ for $t \in \mathcal{J}_{k-1}$, which is the solution from the previous review period. Hence **(F-WBP)** can be solved sequentially for $\mathcal{J}_0$, $\mathcal{J}_1$,.... It is easy to show that the optimal policy to **(F-WBP)** is *piecewise constant*, i.e., the policy is determined at every ϵ, and remains constant during each review period. By construction, a feasible policy to **(F-WBP)** has a threshold-based structure under the WBP rule. Therefore, the optimal policy during $\mathcal{J}_k$ is equal to the solution to **(P-WBP)** with input weights $w_i(k\epsilon)$, $i \in I$.

The main idea behind **(F-WBP)** is as follows. Our goal is to minimize the decoding delay. Thus, we prioritize users who have relatively large amounts of unfinished work by assigning them with large weights, and by maximizing the expected revenue earned by the system under the WBP rule. The policy reviews the system regularly at the beginning of each review period, so as to update weights according to the time-varying buffer trajectories. The priority of service is assigned in the order of work to be done, hence the WBP rule captures the *overall* high revenue by treating channel states as binary numbers, and assigning most significant bits to high workload (revenue) users. Thus, our policy realizes a trade-off between the ease of policy computation versus the achievable revenue from multicast channel.

5.3 WBP-TRACK Policy

We will propose an online policy called WBP-TRACK for the discrete-time stochastic network based on the optimal fluid policy to **(F-WBP)**. Control policies defined in fluid regime should be appropriately *translated* to online policies for stochastic networks. In such policy translation, we would like the online policy to be *asymptotically optimal* [30] in the following sense:

Definition 1 ([30]). *Consider a policy* Π *for the original stochastic network. Also consider a deterministic fluid problem* $(\mathcal{F})$. *Suppose we take the fluid limit of the network controlled under policy* Π. *If the trajectories obtained from such fluid scaling achieve the optimal cost for* $(\mathcal{F})$, *we say* Π *is fluid-scale asymptotically optimal (FSAO) for* $(\mathcal{F})$.

WBP-TRACK is designed to be FSAO for **(F-WBP)**. Similar to **(F-WBP)**, time is broken into review periods with constant length L_r. At the beginning of each review period, we set weight w_i as the remaining number of packets[2] to be received by user i for all $i \in I$. The idea behind WBP-TRACK is to solve **(P-WBP)** based on the weights in order to obtain a WBP rule-based fluid policy, and translate the fluid policy into an implementable discrete-time policy. We further divide each review period into a training period and an executing period. The purpose of the training period is to solve **(P-WBP)** adaptively without the knowledge of channel distribution. The executing period is the policy translation step according to the method of constructing tracking policies [26]. Algorithm 1 describes WBP-TRACK.

2. In fact, the weight can be set to any strictly increasing function of the remaining number of packets, because only the relative order of weight size matters under the WBP rule. Refer to Lemma 2 for details.

Algorithm 1. WBP-TRACK Policy

for the beginning of each review period $l = kL_r, k = 0, 1, 2, \dots$ **do**
  $w_i \leftarrow [K - B_i(l)]^+, \forall i \in I.$
  Sort $w_1, \dots, w_n$ in the decreasing order where ties are broken randomly. Assign the users with the indices in that order.
  Solve **(P-WBP)** using Algorithm 2 during the training period. Denote the obtained policy by $\hat{\pi}(\cdot)$.
  for each time slot τ in the remaining of the review period **do**
    Observe the channel state s.
    RS performs a multicast w.p. $\hat{\pi}(s)\mathbf{1}(B_R(\tau) > 0)$, and performs a fetching w.p. $1 - \hat{\pi}(s)\mathbf{1}(B_R(\tau) > 0)$.
  end for
end for

Algorithm 2. Adaptively Finding ξ for kth Review Period

$\xi \leftarrow 2^{n-1}$.
for $z = 1$ to n **do**
  $T_{\text{in}} \leftarrow 0, T_{\text{out}} \leftarrow 0.$
  for time slots $l = kL_r + (z-1)L_s, \dots, kL_r + zL_s - 1$ **do**
    Observe the channel state s.
    RS performs a multicast if $s \geq \xi$ and $B_R(l) > 0$; RS performs a fetching, otherwise.
    $T_{\text{out}} \leftarrow T_{\text{out}} + \mathbf{1}(s \geq \xi)$, $T_{\text{in}} \leftarrow T_{\text{in}} + \mathbf{1}(s < \xi)$.
  end for
  if $z < n$ **then**
    $\xi \leftarrow \xi + 2^{n-1-z}$ if $T_{\text{out}} > T_{\text{in}}$; $\xi \leftarrow \xi - 2^{n-1-z}$ if $T_{\text{out}} < T_{\text{in}}$; break if $T_{\text{out}} = T_{\text{in}}$.
  else
    $\xi \leftarrow \xi - 1$ if $T_{\text{out}} < T_{\text{in}}$.
  end if
end for

Below we outline how **(P-WBP)** can be solved adaptively during the training period. Our goal is to find the threshold parameters ξ and β of **(P-WBP)** without knowing $p(s), s \in S$. Due to flow conservation, the time fraction of multicasting at the RS cannot exceed $\frac{1}{2}$ in the long run, i.e.,

$$\mathbb{P}(\langle X_1 \dots X_n \rangle > \xi) = \mathbb{P}\left(\sum_{i \in I} 2^{n-i} X_i > \xi\right) \leq \frac{1}{2}, \tag{23}$$

where we omitted the time index from $X_i(l)$ for notational simplicity. We would like to find the minimum ξ which satisfies (23). Define a binary random variable $Z_\xi := \mathbf{1}(\sum_{i \in I} 2^{n-i} X_i(l) > \xi)$. We would like to search for ξ such that $\mathbb{E}[Z_\xi] = \mathbb{P}(Z_\xi = 1)$ is closest to $\frac{1}{2}$ as in (23). For fixed ξ, we need to estimate $\mathbb{E}[Z_\xi]$ without knowing the channel distribution. This is a simple estimation problem for Bernoulli random variables, since we can obtain an independent sample of Z_ξ at every time slot. Hence we can iterate the following in order to find the optimal ξ: (i) take an empirical average of L_s samples of Z_ξ for some large enough constant L_s, (ii) compare the average with $\frac{1}{2}$, and (iii) adjust ξ

accordingly so as to make (23) as tight as possible. The search space for ξ is S of which the size grows exponentially in n. However, since $\mathbb{E}[Z_\xi]$ is monotonic in ξ, we can use *bisection* method to adjust ξ, which guarantees to find the desired ξ by iterating the above steps (i)-(iii) for at most n times. This is outlined in Algorithm 2. After finding the optimal ξ, one can use a similar bisection method to estimate the optimal β. We will apply μ iterations of bisection in order to estimate β for some positive integer μ. Since bisection is used to estimate $\beta \in [0,1]$, the error is bounded by $2^{-\mu}$, i.e., we assume μ-bit precision in estimating β. Now let us denote the longest possible training period by L_t. The above construction shows that $L_t = (n+\mu)L_s$ holds.

5.4 Asymptotic Optimality of WBP-TRACK

We will show that WBP-TRACK is FSAO for **(F-WBP)** by appropriately scaling parameters L_s, L_r and L_t. Suppose a sequence of systems for the fluid scaling parameter $m = 1, 2, \ldots$. Let $b_i^{(m)}(t)$ denote the scaled buffer trajectory of user i in the mth system under WBP-TRACK. Recall that the block size is scaled from K to mK in the mth system. Let $L_s^{(m)} := f_s(mK)$, $L_t^{(m)} := f_t(mK)$ and $L_r^{(m)} := f_r(mK)$ be the scaled versions of parameters L_s, L_t and L_r for the mth system, respectively, for some functions $f_s(\cdot)$, $f_t(\cdot)$ and $f_r(\cdot)$.

Theorem 5. *Suppose $f_s(\cdot)$, $f_t(\cdot)$ and $f_r(\cdot)$ satisfy*

$$f_s(x) \to \infty, \text{as } x \to \infty, \tag{24}$$

$$\frac{f_t(x)}{x} = \frac{f_s(x)(n+\mu)}{x} \to 0, \text{as } x \to \infty, \tag{25}$$

$$\frac{f_r(mK)}{m} \to \epsilon, \text{as } m \to \infty, \tag{26}$$

then WBP-TRACK is FSAO for **(F-WBP)**, *i.e., for all $t \geq 0$*

$$(b_1^{(m)}(t), \ldots, b_n^{(m)}(t)) \to (b_1^*(t), \ldots, b_n^*(t)), \text{as} m \to \infty.$$

Proof. The scaling of L_s, L_r and L_t is based on the construction of tracking policies [26]. Thus, the trajectories of our policy converge to the optimal ones due to the functional strong law of large numbers. It remains to show that the length of training period is long enough so that **(P-WBP)** is exactly solved during the training period, but vanishes under the fluid scaling so that the training period has a negligible effect on fluid trajectories. This requires $L_t^{(m)} \to \infty$ and $\frac{L_t^{(m)}}{m} \to 0$ as $m \to \infty$, which holds from the definition of $f_t(\cdot)$.[3] □

In the rest of the paper, we will investigate the asymptotic performance of WBP-TRACK. To that end, we will examine the optimal trajectories to **(F-WBP)**.

3. To prove Theorem 5, we need the random processes $C_i(t)$'s satisfy the strong law of large numbers (SLLN), i.e., with probability 1, $\lim_{l\to\infty} \sum_{\tau=1}^{l} C_i(\tau)/l = \lambda_i$ for some constant λ_i for all $i \in I$. Hence the time-independent assumption of users' channels can be relaxed, e.g., one can assume that $C_i(t)$'s are stationary and ergodic discrete-time Markov chains (DTMC). Under such assumptions, Theorem 5, as well as the analysis on the asymptotic performance of WBP-TRACK provided in the following two sections, will still be valid.

6 Performance Analysis

6.1 Properties of the Trajectories

We evaluate the asymptotic performance of WBP-TRACK for systems with general channel distribution. We focus on analyzing the buffer trajectories under fluid scaling.

Lemma 2. *The fluid rates $\{r_i(t), i \in I\}$ achieved under WBP-TRACK for $t \in \mathcal{J}_k$ depend only on the relative order, not the absolute sizes, of trajectories $\{b_i(k\epsilon), i \in I\}$.*

Proof. By Theorem 5, the rates $\{r_i(t), i \in I\}$ achieved under WBP-TRACK are the optimal rates to **(F-WBP)**, i.e., the optimal rates to **(P-WBP)** with input weights $\{w_i(k\epsilon), i \in I\}$. Recall that, in Section 5.1, we have argued that the solution to **(P-WBP)** depends only on the relative order of users' weights. Since the relative order of the weights are determined by that of trajectories $\{b_i(k\epsilon), i \in I\}$, the result follows. □

Such invariance of rates to absolute buffer sizes is useful for analyzing the buffer trajectories; especially if the order of users' trajectories is fixed over time, the rates of trajectories will remain fixed as well. Next we examine the properties of the fluid trajectories under WBP-TRACK as ϵ vanishes.[4]

Theorem 6. *Any buffer trajectory under WBP-TRACK is* linear *under fluid scaling as $\epsilon \to 0$.*

Proof. See Appendix D. □

The linearity of buffer trajectories is crucial in proving key properties of WBP-TRACK which we introduce in the sequel.

Definition 2. *We say trajectories $b_i(t)$ and $b_j(t)$* coalesce *if*

$$\max_{t\geq 0} |b_i(t) - b_j(t)| \to 0, \text{as} \epsilon \to 0. \tag{27}$$

The trajectories under WBP-TRACK are piecewise linear if $\epsilon > 0$. However as $\epsilon \to 0$, a set of trajectories may "glue" together, and become a single, merged linear trajectory, which we call *trajectory coalescing*. The coalescing occurs because WBP-TRACK prioritizes smaller trajectories to which higher rates are allocated at the next review period; these trajectories may eventually surpass others. If this process is repeated across users over subsequent review periods, the trajectories may eventually merge together as ϵ vanishes. Fig. 2 shows an example of trajectory coalescing.

6.2 Symmetric-User Systems

We consider symmetric-user systems defined in Section 4, i.e., $p(s) = p(\sigma)$ holds for all $s, \sigma \in S$ such that $\|s\| = \|\sigma\|$. We will show that all the buffer trajectories coalesce. The coalesced buffer trajectory is a straight line over time, thus the associated buffers are filled at an identical and constant rate. Denote the marginal probability p_i by p for all $i \in I$.

Theorem 7. *For symmetric-user systems, let $\tilde{\pi}(\cdot)$ denote the optimal solution to* **(P-WBP)** *obtained by assuming*

4. Recall $L_r^{(m)} := f_r(mK)$ denotes the length of review period for the mth system. By choosing any $f_r(\cdot)$ such that $f_r(x)/\log(x) \to \infty$ and $f_r(x)/x \to 0$ as $x \to \infty$, we have $\epsilon = \lim_{m\to\infty} L_r^{(m)}/m = 0$. Under such scaling of review periods, one can show (see Remark 3 of [26]) that the policy translation via tracking policies is valid.

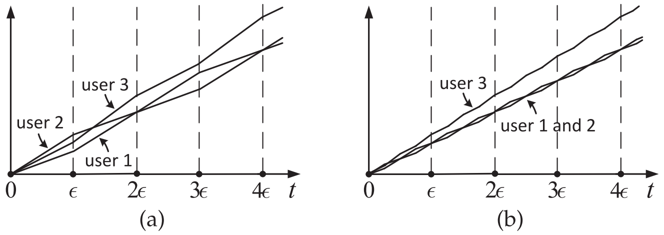

Fig. 2. Example of trajectory coalescing with independent users: $n=3$, $p_1=0.5$, $p_2=0.6$, and $p_3=0.8$. The length of review period is ϵ and $\frac{\epsilon}{4}$ for (a) and (b), respectively. We observe that all trajectories converge to straight lines, and trajectories of user 1 and 2 coalesce as $\epsilon \to 0$.

$w_1 \geq \cdots \geq w_n$. The fluid trajectories of the users' buffers coalesce, and the coalesced buffer trajectory is of the form ct, $\forall t \geq 0$, where the rate c is given by

$$c = \frac{1}{n}\sum_{s\in S} p(s)\tilde{\pi}(s)\|s\|. \tag{28}$$

Thus, WBP-TRACK achieves the delay K/c asymptotically.

Proof. See Appendix E. □

Theorem 7 enables us to explicitly compute the asymptotic decoding delay achieved under WBP-TRACK. Note that it is easy to numerically find $\tilde{\pi}$ using the bisection method, which follows a similar idea as Algorithm 2. Moreover, due to symmetry, i.e., $p(s)$ depends only on $\|s\|$, the summation (28) can be computed in $O(n)$ time. Fig. 3 is a comparison of the asymptotic decoding delay K/c and the optimal decoding delay for **(F)**, for which we assumed that the users' channels are i.i.d. Under varying system sizes and the values of p, we observe that the delay incurred by WBP-TRACK is at most 17 percent higher than the optimal one.

6.3 Two-User Systems

We show that WBP-TRACK is FSAO for original fluid problem **(F)** when $n=2$. Hence WBP-TRACK asymptotically minimizes the decoding delay.

Theorem 8. *When $n=2$, WBP-TRACK achieves the optimal decoding delay for* **(F)** *under fluid scaling as $\epsilon \to 0$.*

Proof: See Appendix F. □

Since WBP-TRACK is constructed from the modified fluid problem **(F-WBP)** instead of the original fluid problem **(F)**, WBP-TRACK may not be FSAO in general for the original problem **(P)**. However, in the proof of Theorem 8, we showed that the minimum fluid rate of users' buffers achieved under WBP-TRACK is equal to the optimal rate. This is a surprising result that it is optimal to schedule users in a prioritized way over the multicast channel as in WBP-TRACK for two-user systems.

6.4 Systems with Independent Users' Channels

The analysis so far considered systems with arbitrarily correlated channels. In this section, we consider the important special case where the users' channels are independent, but are not necessarily identically distributed. Unfortunately, it is still hard to analytically derive the asymptotic delay under our policy, because of complex system behaviours such as trajectory coalescing. Instead, we aim to provide a performance bound which can be easily computed. To that end, we introduce a lemma on the fluid rates of the users.

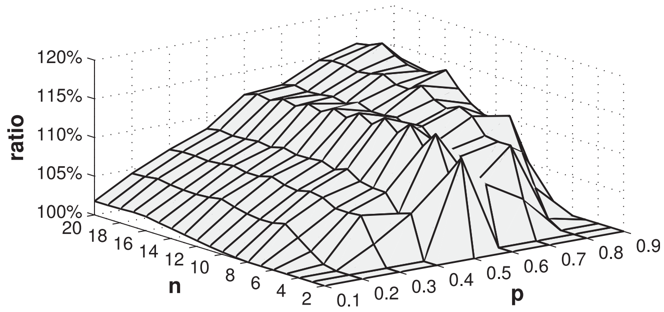

Fig. 3. The ratio (in percentage) between asymptotic decoding delay achieved under WBP-TRACK and the optimal decoding delay for **(F)**, for independent and symmetric users' channels with varying n and p.

Lemma 3. *For systems with independent channel distributions, the fluid rate of user i is at least $p_i/2$ under WBP-TRACK for all $i \in I$.*

Proof. The proof is relegated to [29]. □

Since the fluid rate of user i cannot exceed p_i under any policy, Lemma 3 immediately implies that the asymptotic delay achieved under WBP-TRACK is at most twice the optimal one. However, it is our goal to derive a tighter bound in the rest of this section.

Our approach is summarized as follows. We first introduce a reference system denoted by $\mathcal{S}$. It is easy to compute the users' service rates in the reference system. We then compare the achievable fluid rates in the original system and those in $\mathcal{S}$, and show that the minimum fluid rate among the users under WBP-TRACK is at least that in $\mathcal{S}$.

Throughout this section, we will assume $p_1 \leq p_2 \leq \cdots \leq p_n$ for notational simplicity, i.e., user 1 has the worst channel condition. $\mathcal{S}$ is a n-user system; user i is allocated with the rate given by $\frac{q_i}{2}t_i + \frac{p_i}{2}$ where $q_i := p_i \wedge (1-p_i)$ for t_i fraction of time. In $\mathcal{S}$, the rates are allocated such that that the minimum service rate among the users is maximized. Thus, the time fractions $t_1, \ldots, t_n$ solve the following problem:

$$\begin{aligned} (P-\boldsymbol{\mathcal{S}}) \quad & \underset{t_1,\ldots,t_n}{\text{maximize}} \quad \min_{i\in I}\{r_i\} \\ & \text{subject to} \quad r_i = \frac{q_i}{2}t_i + \frac{p_i}{2}, \forall i \in I, \\ & \sum_{i=1}^{k} t_i = 1, \quad t_i \geq 0, \forall i \in I. \end{aligned}$$

Let r_i^* and t_i^* denote the optimal r_i and t_i for **(P-$\mathcal{S}$)**, for all $i \in I$. Let $z^* = \min_{i\in I}\{r_i^*\}$ be the optimal value of **(P-$\mathcal{S}$)**. It is easy to show that the solution to **(P-$\mathcal{S}$)** has the "water-filling" structure such that

$$t_i^* = \left[\frac{2z^* - p_i}{q_i}\right]^+, r_i^* = \max\left\{z^*, \frac{p_i}{2}\right\},$$

where z^* satisfies $\sum_{i=1}^n t_i^* = 1$. Let k^* be the number of users such that $t_i^* > 0$. Due to the water-filling property, the following inequality holds in $\mathcal{S}$

$$z^* = r_1^* = \cdots = r_{k^*}^* \leq r_{k^*+1}^* \leq \cdots \leq r_n^*. \tag{29}$$

An example of the solution structure is shown in Fig. 4a.

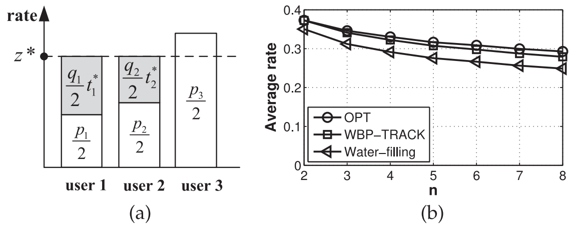

Fig. 4. (a) Water-filling solution structure: $n = 3$ and $k^* = 2$. (b) Comparison of asymptotic rates: for each n, p_i $(i \in I)$ are uniformly and independently generated from $[0.3, 0.7]$ for 3,000 times. "OPT" represents the optimal rate to **(P)**; "Water-filling" represents z^*.

Next, we will show that WBP-TRACK in the original system achieves the minimum fluid rate of at least z^*.

Theorem 9. *Suppose that the users' channels are independently distributed. The minimum fluid rate of the users under WBP-TRACK is at least z^* as $\epsilon \to 0$. Thus, the asymptotic decoding delay is at most K/z^*.*

Proof. Let $\hat{k}$ be the number of trajectories which coalesce *and* achieve the minimum rate under WBP-TRACK. For example, $\hat{k} = 1$ implies that the trajectory which achieved the minimum rate did not coalesce with the other trajectories. Let $\hat{r}$ denote the minimum rate. Let $\hat{I} := \{f(1), \ldots, f(\hat{k})\} \subseteq I$ denote the set of indices of the coalesced trajectories. Without loss of generality, assume $f(i) < f(j)$ for all $1 \le i < j \le \hat{k}$. Let $\hat{t}_i$ denote the fraction of time during which user i is served with the highest priority under WBP-TRACK. During $\hat{t}_i$ fraction of time, user i is served at rate $p_i \wedge \frac{1}{2}$, because user i will be served whenever its channel is in ON state under WBP-TRACK; hence the rate is p_i, but cannot exceed $\frac{1}{2}$ due to the half-duplex operation. During the remaining $1 - \hat{t}_i$ fraction of time, user i is still served at a nonzero rate, due to multicasting by the RS. This rate is at least $\frac{p_i}{2}$ due to Lemma 3. Thus, the average service rate of user i is at least

$$\left(p_i \wedge \frac{1}{2}\right)\hat{t}_i + \frac{p_i}{2}(1 - \hat{t}_i) = \frac{q_i}{2}\hat{t}_i + \frac{p_i}{2}. \tag{30}$$

We consider two cases:

Case 1. $\hat{k} \le k^*$: Since every trajectory is linear under WBP-TRACK due to Theorem 6, the trajectories in $\hat{I}$ coalesce into $\hat{r}t$. Hence the coalesced trajectory associated with $\hat{I}$ is always below the other trajectories, i.e., users *not* in $\hat{I}$ cannot be served at the highest priority under WBP-TRACK at any instant. Thus, $\sum_{i=1}^{\hat{k}} \hat{t}_{f(i)} = 1$ holds, and we have that for any $1 \le i \le \hat{k}$

$$z^* = \frac{p_i}{2} + \frac{q_i}{2}t_i^*, \tag{31}$$

$$\hat{r} \ge \frac{p_{f(i)}}{2} + \frac{q_{f(i)}}{2}\hat{t}_{f(i)} \ge \frac{p_i}{2} + \frac{q_i}{2}\hat{t}_{f(i)}. \tag{32}$$

(31) is due to (29). The first inequality of (32) holds due to (30); the second inequality is due to assumptions $p_1 \le\le p_n$ and $f(i) \ge i$. Now suppose $z^* > \hat{r}$. We must have $t_i^* > \hat{t}_{f(i)}$, $\forall i = 1, \ldots, \hat{k}$, due to (31) and (32). However, since $\hat{k} \le k^*$, we have that

$$\sum_{i=1}^{\hat{k}} t_i^* \le \sum_{i=1}^{k^*} t_i^* = 1 = \sum_{i=1}^{\hat{k}} \hat{t}_{f(i)},$$

which contradicts $t_i^* > \hat{t}_{f(i)}$, $\forall i = 1, \ldots, \hat{k}$. Hence $z^* \le \hat{r}$ holds.

Case 2. $\hat{k} > k^*$: By definition, Trajectory $f(\hat{k})$ is one of the coalesced trajectories that achieve rate $\hat{r}$. Hence due to Lemma 3, $\hat{r} \ge \frac{p_{f(\hat{k})}}{2}$ holds. In system $\mathcal{S}$, we have that $z^* \le r^*_{k^*+1} = \frac{p_{k^*+1}}{2}$. By assumption, we have that $f(\hat{k}) \ge \hat{k}$ and $\hat{k} \ge k^* + 1$. Hence we have that

$$z^* \le \frac{p_{k^*+1}}{2} \le \frac{p_{\hat{k}}}{2} \le \frac{p_{f(\hat{k})}}{2} \le \hat{r}.$$

Consequently, z^* is a lower bound on the minimum service rate of users' buffers under WBP-TRACK. □ □

Theorem 9 enables us to estimate the performance of WBP-TRACK for a large system using reference system $\mathcal{S}$; one can compute z^* for a performance bound. It is easy to verify that $z^* \ge \frac{p_1}{2}$ always holds, i.e., Theorem 9 provides a better bound than that implied by Lemma 3. Moreover, we can derive a sufficient condition under which WBP-TRACK is asymptotically optimal, as follows.

Theorem 10. *Suppose $p_1 \le p_2/2$ and the users' channels are independent. WBP-TRACK achieves the optimal decoding delay for* **(F)** *as $\epsilon \to 0$ irrespective of n.*

Proof. One can show that, if $p_1 \le \frac{p_2}{2}$, $z^* = \min\{p_1, \frac{1}{2}\} = p_1$ holds by solving **(P-$\mathcal{S}$)**. However, the fluid rate of user 1 cannot exceed p_1 under any policy for **(F)**. Thus, the result follows from Theorem 9. □

Fig. 4b shows a numerical evaluation of asymptotic rate achieved under WBP-TRACK. We observe that WBP-TRACK is optimal when $n = 2$ as predicted by Theorem 8. Overall, the fluid rate achieved by WBP-TRACK is very close to the optimal rate. Moreover, we find that the proposed bound, z^*, approximates WBP-TRACK well.

7 Simulation Results

The analysis so far focused on the asymptotic performance of WBP-TRACK. In this section, we evaluate the performance in discrete-time stochastic networks by simulations.

We perform simulations using MATLAB under its GUI environment. We choose the spatially correlated Rayleigh-fading channel [31] as our channel model as follows:

$$y_i[l] = h_i[l]x_i[l] + z_i[l], \forall i \in I, \forall l = 0, 1, 2, \ldots,$$

where $x_i[l]$ (resp. $y_i[l]$) is transmitted (resp. received) signal to user i at time slot l; $h_i[l] \sim \mathcal{CN}(0, \sigma_i^2)$ is the fading parameter for user i at time slot l, where $\mathcal{CN}$ denotes the complex Gaussian distribution; and $z_i[l] \sim \mathcal{CN}(0, N_0)$ denotes the i.i.d. additive noise to user i at time slot l. Define vector $\boldsymbol{v}[l] := (v_1[l], \ldots, v_n[l])$, where $v_i[l] := |h_i[l]|$ for all $i \in I$. Hence, $v_i[l]$ is Rayleigh distributed for all $i \in I$ and l. We generate $\boldsymbol{v}[l]$ independently over time slots, while $v_1[l], \ldots, v_n[l]$ are correlated through covariance matrix $\boldsymbol{\Sigma} := \mathbb{E}[(\boldsymbol{v}[l] - \mathbb{E}[\boldsymbol{v}[l]])(\boldsymbol{v}[l] - \mathbb{E}[\boldsymbol{v}[l]])^T]$. Let P denote the transmit power of the RS. The number of bits which can be transmitted to user i at time slot l, i.e., $C_i(l)$, is given by $\delta B\log(1 + |h_i[l]|^2 \frac{P}{N_0})$ bits, where δ is the time slot length and B is the bandwidth.

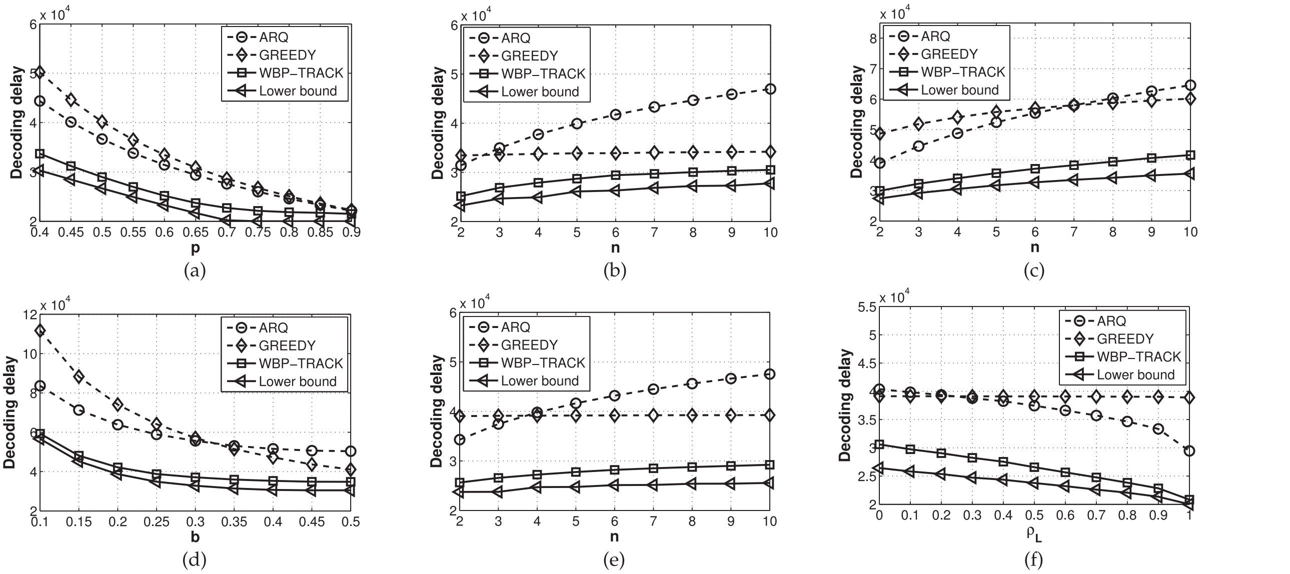

Fig. 5. Comparison of decoding delays: (a) Independent and symmetric channels with $n = 2$ and varying p. (b) Independent and symmetric channels with $p = 0.6$ and varying n. (c) Independent channels with p_i's are uniformly generated from $[0.3, 0.7]$ and varying n. (d) Independent channels with $n = 6$, p_i's are uniformly generated from $[b, 1 - b]$ and varying b. (e) Correlated channels with $\mathbf{\Sigma}_{i,i} = 1$ for all $i \in I$, $\mathbf{\Sigma}_{i,j} = \frac{1}{2}$ for $i \neq j$, $i, j \in I$, and varying n. (f) Correlated channels with $n = 3$, $\mathbf{\Sigma}_{i,i} = 1$ for all $i \in I$, $\mathbf{\Sigma}_{i,j} = \rho_L$ for all $i \neq j$, $i, j \in I$, and varying ρ_L.

The following are baseline schemes in our simulations:

ARQ: This is the traditional ARQ scheme. The RS first fetches a packet from the BS, then the RS keeps multicasting the packet until every user receives a copy. This procedure is repeated for K times.

GREEDY: In this scheme, the RS greedily performs a multicast whenever possible, i.e., the RS performs a fetching only when the RS queue is empty or when all the users' channels are in OFF state.

In case of finite-time horizon problems, the optimal cost of an MDP is bounded below by the solution to the associated fluid problem [25]. In our simulations, we will compare our scheme with a lower bound of the decoding delay, which is obtained by numerically solving **(P)**. We let $K = 10^4$. For the WBP-TRACK policy, we set $L_s = 20$, $\mu = 3$ and $L_r = 200n$.

Fig. 5a shows the average decoding delay when $n = 2$. The users' channels are independent and symmetric where $p_1 = p_2 = p$, and we vary p in the plot. WBP-TRACK performs the best, which is only 7-11 percent higher than the lower bound. WBP-TRACK reduces the decoding delay up to 25 and 33 percent compared with ARQ and GREEDY, respectively. Interestingly, GREEDY is shown to perform worse than ARQ. This implies that, if a scheduling policy adopting coding is oblivious of channel conditions, and thus does not exploit opportunism, the policy may incur higher delay than ARQ.

Next we evaluate the performance under independent and symmetric channels with $p = 0.6$ by varying the number of users in Fig. 5b. WBP-TRACK performs the best, and incurs 8-15 percent higher delay than the lower bound. The delay incurred by GREEDY is slightly affected by n, since all the users are symmetric and they finish receiving K coded packets "simultaneously". By contrast, the delay achieved by ARQ increases rapidly with n, which is up to 48 percent higher than WBP-TRACK.

The delay performance for independent and asymmetric channels with varying n is illustrated in Fig. 5c. For each n, we generated p_i's uniformly from $[0.3, 0.7]$. WBP-TRACK significantly reduces the decoding delay compared to other schemes; the gains of reduction are 24-31 and 32-39 percent relative to ARQ and GREEDY, respectively. Fig. 5d shows the performance when n is fixed to 6, where p_i's are uniformly and independently generated from $[b, 1 - b]$ and b varies from 0.1 to 0.5. Hence the smaller b, there is more variability in the channel parameters. The relative reduction in delay by WBP-TRACK compared to other schemes also becomes greater for smaller b. WBP-TRACK reduces decoding delay up to 29 and 47 percent compared with ARQ and GREEDY, respectively. Interestingly, WBP-TRACK performs even better as b decreases, i.e., it performs closer to the lower bound. We conclude that WBP-TRACK achieves near-optimal delay even under a wide range of variability in the users' channel quality.

Next we compare the performance under correlated Rayleigh fading channels. We set $\frac{P}{N_0} = 20$ dB, $\delta = 1$ ms, $B = 1$ MHz, $\nu = 10^3$ and $d = 32$.

In Fig. 5e, we set covariance matrix $\mathbf{\Sigma}$ such that $\mathbf{\Sigma}_{i,i} = 1$ for all $i \in I$, and $\mathbf{\Sigma}_{i,j} = \frac{1}{2}$ for $i \neq j$, $i, j \in I$. We vary n in the simulation. WBP-TRACK performs well, and is close to the lower bound. GREEDY incurs 35-53 percent higher decoding delay than WBP-TRACK, and ARQ achieves 51-63 percent higher delay than WBP-TRACK. Fig. 5f shows the performance where $n = 3$, $\mathbf{\Sigma}_{i,i} = 1$ for all $i \in I$, and $\mathbf{\Sigma}_{i,j} = \rho_L$ for all $i \neq j$, $i, j \in I$. We vary ρ_L from 0 to 1 with interval 0.1 in the simulation. We observe that, as the correlation among channels increases, the decoding delay decreases. Hence the channel correlation tends to help reducing decoding delay. This can be perhaps deduced from the extreme case such that $\rho_L = 1$, i.e., the channels are identical. In this case, every user receives the packet simultaneously at every multicast. Thus, the variance of the time instants at which the users complete receiving the whole block is 0. Since the decoding delay is determined by the user with the worst completion time, a lower variance will result in a lower delay, if the mean parameter of the channel is fixed. When $\rho_L = 1$, WBP-TRACK is only

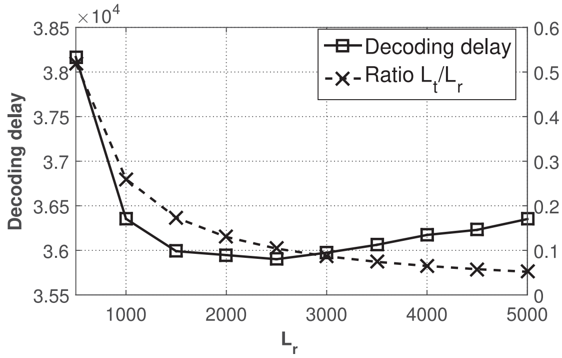

Fig. 6. Decoding delay performance of WBP-TRACK with varying length of review periods. We assume that the users' channels are independent, and that $n = 10$, $p_1 = p_2 = 0.4$, $p_3 = p_4 = 0.5$, $p_5 = p_6 = 0.6$, $p_7 = p_8 = 0.7, p_9 = p_{10} = 0.8, L_s = 20$, and $\mu = 3$.

4 percent higher than the lower bound; by contrast, ARQ and GREEDY incur decoding delays 41 and 87 percent higher than WBP-TRACK, respectively. The results show that WBP-TRACK performs well across various degrees of correlation among users' channels.

Next we study the effect of the length of review period on the performance of WBP-TRACK. Fig. 6 shows 1) decoding delay; 2) the ratio between training period and review period, against the length of review period. We fixed L_t to the value used in the previous simulations. Intuitively, on the one hand, longer L_r is desirable because relatively less fraction of time is wasted in learning threshold parameters in each review period; on the other hand, shorter L_r enables a faster update of weights, i.e., we can promptly track changes in trajectories. The figure shows that there exists a best tradeoff point, in which case the training period occupies about 10 percent of the review period.

8 Conclusions

We studied the problem of minimizing the decoding delay for coded multicasting in two-hop relay networks. We proved that the optimal scheduling under fluid scaling has a threshold-based structure. We proposed an adaptive online policy which prioritizes users based on the remaining workload, and provided asymptotic bounds on performance guarantees. Simulation results show that, our policy in fact performs substantially close to the achievable bound.

Appendix A
Proof of Theorem 1

Denote the optimal policies to **(F)** by $g_s^*(t)$ and $\psi_s^*(t)$. Let T^* denote the optimal decoding delay. Denote the optimal buffer trajectory achieved by user i by $b_i^*(t)$. Let

$$\pi(s) := \frac{g_s^*(T^*)}{p(s)T^*}, \forall s \in S. \tag{33}$$

It is easy to show that policies (9) with $\pi(\cdot)$ defined in (33) are feasible for **(F)**, i.e., the policies satisfy (6), (7), and (8). Next we show that they achieve the optimal delay T^*. We have that

$$\begin{aligned} b_i(t) &= \sum_{s\in S} g_s(t)s_i = t\sum_{s\in S} p(s)\pi(s)s_i \\ &= t\sum_{s\in S}\frac{g_s^*(T^*)}{T^*}s_i = \frac{t}{T^*}b_i^*(T^*). \end{aligned}$$

Thus, we have that $b_i(T^*) = b_i^*(T^*)$, i.e., by time T^*, user i receives the same amount of fluid as that of the optimal policy. We conclude that policies (9) are optimal for **(F)**.

Appendix B
Proof of Theorem 2

Let $\boldsymbol{\pi} \in \{0,1\}^{|S|}$ denote the $|S|$-dimensional vector of $\pi(s)$, $s \in S$. Let $\boldsymbol{r} = (r_1, \dots, r_n)$. From (11), $\boldsymbol{r}$ is a linear transformation of $\boldsymbol{\pi}$. Thus, we can transform **(P)** into an equivalent LP with respect to $\boldsymbol{r}$ instead of $\boldsymbol{\pi}$. From (10), (11), and (12), one can show that the feasible region of $\boldsymbol{r}$ is a convex polyhedron. Hence, **(P)** is equivalent to the following problem for some matrix $\boldsymbol{A}$ and vector $\boldsymbol{b}$:

$$\underset{\boldsymbol{r}}{\text{maximize}} \min_{i\in I}\{r_i\} \quad \text{subject to } \boldsymbol{A}\boldsymbol{r} \le \boldsymbol{b}. \tag{34}$$

Thus, the dual of **(P)** is equivalent to

$$\begin{aligned} &\underset{\boldsymbol{\lambda},\boldsymbol{\nu}}{\text{minimize}}\ \boldsymbol{b}^T\boldsymbol{\nu} \\ &\text{subject to } \mathbf{1}^T\boldsymbol{\lambda} = 1, \boldsymbol{A}^T\boldsymbol{\nu} = \boldsymbol{\lambda}, \\ &\qquad \boldsymbol{\lambda} \ge \mathbf{0}, \boldsymbol{\nu} \ge \mathbf{0}. \end{aligned} \tag{35}$$

The dual of **(MW)** is

$$\begin{aligned} &\underset{\boldsymbol{\nu}}{\text{minimize}}\ \boldsymbol{b}^T\boldsymbol{\nu} \\ &\text{subject to } \boldsymbol{A}^T\boldsymbol{\nu} = \boldsymbol{w}, \boldsymbol{\nu} \ge \mathbf{0}. \end{aligned} \tag{36}$$

Let $\boldsymbol{\lambda}^*$ be the dual optimal $\boldsymbol{\lambda}$ for (35). If we let $\boldsymbol{w} = \boldsymbol{\lambda}^*$, we see that (36) is equivalent to (35). Consequently, **(P)** and **(MW)** are equivalent.

Appendix C
Proof of Theorem 3

Let $x(s) := p(s)\pi(s), \forall s \in S$. We can rewrite **(MW)** as

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & \sum_{s\in S} x(s)\rho(\boldsymbol{w}, s) \\ \text{subject to} \quad & \sum_{s\in S} x(s) \le \frac{1}{2}, \\ & 0 \le x(s) \le p(s), \forall s \in S. \end{aligned} \tag{37}$$

The above problem can be interpreted as a constrained portfolio optimization as follows. $x(s)$ represents the allocation of budget over asset s, and the unit return is given by $\rho(\boldsymbol{w}, s)$. Thus, one can maximize the total return by spending the maximum possible budget on the asset with the highest return, then on that with the next highest return, etc. This yields the threshold-based solution in the theorem.

Appendix D
Proof of Theorem 6

We will compare the trajectories under WBP-TRACK for different lengths of review period. Let $b_i^{(1)}(t)$ (resp. $b_i^{(2)}(t)$) denote the buffer trajectory of user $i \in I$ when the length of review period is given by ϵ (resp. $\frac{\epsilon}{2}$). Let $r_i^{(j)}(t) := \dot{b}_i^{(j)}(t)$ denote the associated rate of user i, for $j = 1, 2$.

We will use induction to show $b_i^{(1)}(k\epsilon) = 2b_i^{(2)}(\frac{k\epsilon}{2}), \forall i \in I$, for $k = 0, 1, 2, \dots$. For $k = 0$, we have that $b_i^{(1)}(0) =$

$b_i^{(2)}(0)=0$, thus the hypothesis holds. Suppose the hypothesis holds for some $k>0$. Since $b_i^{(2)}(t)$ is piecewise linear, for $t\in\left[\frac{k\epsilon}{2},\frac{(k+1)\epsilon}{2}\right)$, we have that

$$b_i^{(2)}(t)=b_i^{(2)}\left(\frac{k\epsilon}{2}\right)+r_i^{(2)}\left(\frac{k\epsilon}{2}\right)\left(t-\frac{k\epsilon}{2}\right).$$

Also for $t\in[k\epsilon,(k+1)\epsilon)$, we have that

$$\begin{aligned}b_i^{(1)}(t)&=b_i^{(1)}(k\epsilon)+r_i^{(1)}(k\epsilon)(t-k\epsilon)\\&\overset{(a)}{=}2b_i^{(2)}\left(\frac{k\epsilon}{2}\right)+r_i^{(2)}\left(\frac{k\epsilon}{2}\right)(t-k\epsilon)=2b_i^{(2)}\left(\frac{t}{2}\right).\end{aligned}$$

The equality (a) holds because; due to induction hypothesis $b_i^{(1)}(k\epsilon)=2b_i^{(2)}(\frac{k\epsilon}{2})$, the relative order of $b_i^{(1)}(k\epsilon)$ among users is equal to that of $b_i^{(2)}(\frac{k\epsilon}{2})$; this implies that, due to Lemma 2, $r_i^{(1)}(k\epsilon)=r_i^{(2)}(\frac{k\epsilon}{2})$ for all $i\in I$. Due to continuity of trajectories, the above equation further implies that $b_i^{(1)}((k+1)\epsilon)=2b_i^{(2)}(\frac{(k+1)\epsilon}{2})$ holds for all $i\in I$, which proves the induction step. Thus

$$b_i^{(1)}(t)=2b_i^{(2)}\left(\frac{t}{2}\right),\quad \forall t\geq 0. \tag{38}$$

As $\epsilon\to 0$, $|b_i^{(1)}(t)-b_i^{(2)}(t)|\to 0$ uniformly over $t\geq 0$. Hence from (38). we have $|b_i^{(1)}(t)-\frac{1}{2}b_i^{(1)}(2t)|\to 0$ as $\epsilon\to 0$, $\forall t\geq 0$, Thus, $b_i^{(1)}(t)$ converges to a linear function of t.

Appendix E
Proof of Theorem 7

We first show that all the trajectories coalesce as $\epsilon\to 0$. We specifically show that the following holds for $k=0,1,2,\ldots$

$$|b_i(t)-b_j(t)|\leq p\epsilon, \forall i,j\in I, t\in\mathcal{J}_k, \tag{39}$$

by induction on k. First (39) holds for $k=0$; since $b_i(0)=0$, $\forall i\in I$, and the rate of any trajectory is at most p. Now suppose (39) holds for $k=z-1$ for some $z\in\mathbb{Z}_+$, i.e., during $\mathcal{J}_{z-1}=[(z-1)\epsilon,z\epsilon)$. Due to continuity of trajectories, this also assumes that $|b_i(z\epsilon)-b_j(z\epsilon)|\leq p\epsilon$. Without loss of generality, assume $b_i(z\epsilon)<b_j(z\epsilon)$. Since $w_i(z\epsilon)>w_j(z\epsilon)$, user i gets the higher priority and achieves a higher or equal rate under the WBP rule at time $t=z\epsilon$. Hence $r_i(z\epsilon)\geq r_j(z\epsilon)$ holds and since $r_i(t)$ and $r_j(t)$ are constant during $\mathcal{J}_z$, we have that

$$\begin{aligned}&|b_i(t)-b_j(t)|=|b_i(z\epsilon)-b_j(z\epsilon)+[r_i(z\epsilon)-r_j(z\epsilon)](t-z\epsilon)|\\&\leq\max\{b_j(z\epsilon)-b_i(z\epsilon),[r_i(z\epsilon)-r_j(z\epsilon)](t-z\epsilon)\}\\&\leq p\epsilon,\quad\forall t\in\mathcal{J}_z.\end{aligned}$$

Hence (39) holds for $k=z$. Consequently, (39) holds by induction, which implies that $b_1(t),\ldots,b_n(t)$ coalesce.

Next, we analyze rate c of the coalesced trajectories. For $t\in\mathcal{J}_k$ and $i\in I$, we have that

$$b_i(t)=\sum_{z=0}^{k-1}r_i(z\epsilon)\epsilon+r_i(k\epsilon)(t-k\epsilon). \tag{40}$$

Meanwhile from (39), for sufficiently small ϵ and for $t\in\mathcal{J}_k$

$$b_i(t)\approx\frac{1}{n}\sum_{j\in I}b_j(t) \tag{41}$$

$$=\frac{1}{n}\left\{\sum_{z=0}^{k-1}\left[\sum_{j\in I}r_j(z\epsilon)\right]\epsilon+\left[\sum_{j\in I}r_j(k\epsilon)\right](t-k\epsilon)\right\} \tag{42}$$

$$=\frac{t}{n}\sum_{j\in I}\sum_{s\in S}p(s)\tilde{\pi}(s)s_i=\frac{t}{n}\sum_{s\in S}p(s)\tilde{\pi}(s)\|s\|=ct. \tag{43}$$

where (43) is derived from (42) as follows. $\sum_{j\in I}r_j(z\epsilon)$ is the sum-rate achieved by the system, and is determined by the solution to **(P-WBP)** with weights $w_1(z\epsilon),\ldots,w_n(z\epsilon)$. However, due to symmetry in the channel, the sum-rate is constant irrespective of the relative order of the weights. Hence the sum-rate is given by nc where c is given by (28). The approximation in (41) becomes the uniform convergence as $\epsilon\to 0$, which proves the theorem.

Appendix F
Proof of Theorem 8

Since WBP-TRACK is FSAO for **(F-WBP)**, we first analyze the optimal trajectories to **(F-WBP)**. Assume $p_1\leq p_2$, without loss of generality. By Lemma 2, rates $\{r_i(t),i\in I\}$ for $t\in\mathcal{J}_k$ depend only on the relative order of trajectories $\{b_i(k\epsilon),i\in I\}$. Accordingly let $r_i^{(j)}$ denote the fluid rate achieved by user i when user j receives the higher priority, for all $i,j\in\{1,2\}$. We consider the following cases:

Case 1: $p(\langle 11\rangle)\geq\frac{1}{2}$. Irrespective of users' buffer sizes, the optimal policy is given by $\pi(s)=\frac{1}{2p(\langle 11\rangle)}$ if $s=\langle 11\rangle$, and $\pi(s)=0$ otherwise. Hence, one can find that $r_i^{(j)}=\frac{1}{2}$ for all $i,j\in\{1,2\}$. This means that both users achieve the same rate of $\frac{1}{2}$ during the whole transmission.

Case 2: $p(\langle 11\rangle)<\frac{1}{2}$ and $p(\langle 11\rangle)\geq 2p_1-\frac{1}{2}$. We have that

$$r_1^{(1)}=p_1\leq\left(p(\langle 11\rangle)+\frac{1}{2}-p_1\right)\wedge p_2=r_2^{(1)}.$$

Also we can show $r_1^{(2)}<r_2^{(2)}$. Hence, user 1 always achieves a smaller rate than user 2 irrespective of the priority. As a result, user 1 will receive the higher priority during the transmission and achieve a rate of p_1.

Case 3: $p(\langle 11\rangle)<\frac{1}{2}$ and $p(\langle 11\rangle)<2p_1-\frac{1}{2}$. We will first find $r_i^{(j)}$ for all $i,j\in\{1,2\}$. We consider three subcases. If $p_1\leq p_2\leq\frac{1}{2}$, we have that

$$r_1^{(1)}=p_1, r_2^{(1)}=p(\langle 11\rangle)+\frac{1}{2}-p_1, \tag{44}$$

$$r_1^{(2)}=p(\langle 11\rangle)+\frac{1}{2}-p_2, r_2^{(2)}=p_2. \tag{45}$$

If $p_1\leq\frac{1}{2}\leq p_2$, we have that

$$r_1^{(1)}=p_1,\quad r_2^{(1)}=p(\langle 11\rangle)+\frac{1}{2}-p_1, \tag{46}$$

$$r_1^{(2)}=p(\langle 11\rangle),\quad r_2^{(2)}=\frac{1}{2}. \tag{47}$$

If $\frac{1}{2}\leq p_1\leq p_2$, we have that

$$r_1^{(1)}=r_2^{(2)}=\frac{1}{2},\quad r_2^{(1)}=r_1^{(2)}=p(\langle 11\rangle). \tag{48}$$

In all of the above subcases, we have that

$$r_1^{(1)}>r_2^{(1)},\quad r_1^{(2)}<r_2^{(2)},$$

i.e., the user who receives the higher priority will achieve the higher rate. Hence the order of the priority will alternate between the users over review periods. As $\epsilon\to 0$, two trajectories will eventually coalesce, which can be proved similar to Theorem 7. Thus, $b_1(t)$ and $b_2(t)$ will coalesce into a line, say ct, where we compute c as follows.

Let t_1 (resp. t_2) denote the fraction of time user 1 (resp. user 2) receives the higher priority as $\epsilon \to 0$. It is clear that $t_1 + t_2 = 1$. Due to trajectory coalescing, we have that

$$t_1 r_1^{(1)} + t_2 r_1^{(2)} = t_1 r_2^{(1)} + t_2 r_2^{(2)} = c.$$

Hence c is given by

$$c = \frac{\Delta_2}{\Delta_1 + \Delta_2} r_1^{(1)} + \frac{\Delta_1}{\Delta_1 + \Delta_2} r_1^{(2)},$$

where $\Delta_1 := r_1^{(1)} - r_2^{(1)}$ and $\Delta_2 := r_2^{(2)} - r_1^{(2)}$. One can show that c is equal to $\frac{1}{4} + \frac{p(\langle 11 \rangle)}{2}$ in all cases of $r_1^{(j)}$ and $r_2^{(j)}$, $j = 1, 2$, given by (44), (45), (46), (47), and (48).

Next, one can find the optimal fluid rates by directly solving **(P)**. It turns out that, the solution to **(P)** is identical to the minimum rate among users under WBP-TRACK derived in all the above cases. We omit the details here to save space.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government (NRF-2016R1A2B1014934) and ZheJiang Provincial Natural Science Foundation of China under Grant LQ17F010001.

REFERENCES

[1] K. Loa, et al., "IMT-advanced relay standards [WiMAX/LTE update]," *IEEE Commun. Mag.*, vol. 48, no. 8, pp. 40–48, Aug. 2010.

[2] A. Damnjanovic, et al., "A survey on 3GPP heterogeneous networks," *IEEE Wireless Commun.*, vol. 18, no. 3, pp. 10–21, Jun. 2011.

[3] A. Sabharwal, P. Schniter, D. Guo, D. W. Bliss, S. Rangarajan, and R. Wichman, "In-band full-duplex wireless: Challenges and opportunities," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 9, pp. 1637–1652, Sep. 2014.

[4] D. Lecompte and F. Gabin, "Evolved multimedia broadcast multicast service (eMBMS) in LTE-advanced: Overview and Rel-11 enhancements," *IEEE Commun. Mag.*, vol. 50, no. 11, pp. 68–74, Nov. 2012.

[5] (2014). [Online]. Available: http://blog.twitch.tv/2014/02/twitch-community-4th-in-peak-us-internet-traffic

[6] D. J. MacKay, "Fountain codes," *IEE Proc.-Commun.*, vol. 152, no. 6, pp. 1062–1068, 2005.

[7] T. Ho, et al., "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[8] M. Luby, "LT codes," in *Proc. IEEE 43th Annu. Symp. Found. Comput. Sci.*, 2002, pp. 271–280.

[9] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.

[10] T. Stockhammer, "Dynamic adaptive streaming over HTTP–: Standards and design principles," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst.*, 2011, pp. 133–144.

[11] Y. Yang and N. Shroff, "Throughput of rateless codes over broadcast erasure channels," in *Proc. 13th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2012, pp. 125–134.

[12] B. Swapna, A. Eryilmaz, and N. Shroff, "Throughput-delay analysis of random linear network coding for wireless broadcasting," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6328–6341, Oct. 2013.

[13] P. Polacek, C. W. Huang, and J. W. Chiang, "Unified opportunistic scheduling for layered multicast over cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 10, pp. 2627–2639, Oct. 2016.

[14] B. Shrader and A. Ephremides, "Queueing delay analysis for multicast with random linear coding," *IEEE Trans. Inf. Theory*, vol. 58, no. 1, pp. 421–429, Jan. 2012.

[15] R. Cogill, B. Shrader, and A. Ephremides, "Stable throughput for multicast with random linear coding," *IEEE Trans. Inf. Theory*, vol. 57, no. 1, pp. 267–281, Jan. 2011.

[16] N. Moghadam and H. Li, "Queue stability analysis in network coded wireless multicast network," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 950–953, May 2016.

[17] G. H. Sim, J. Widmer, and B. Rengarajan, "Opportunistic finite horizon multicasting of erasure-coded data," *IEEE Trans. Mobile Comput.*, vol. 15, no. 3, pp. 705–718, Mar. 2016.

[18] T.-P. Low, M.-O. Pun, Y.-W. P. Hong, and C.-C. J. Kuo, "Optimized opportunistic multicast scheduling (OMS) over wireless cellular networks," *IEEE Trans. Wireless Commun.*, vol. 9, no. 2, pp. 791–801, Feb. 2010.

[19] L. Yang, Y. E. Sagduyu, J. Zhang, and J. H. Li, "Deadline-aware scheduling with adaptive network coding for real-time traffic," *IEEE/ACM Trans. Netw.*, vol. 23, no. 5, pp. 1430–1443, Oct. 2015.

[20] H. Li, B. Li, T. T. Tran, and D. C. Sicker, "Transmission schemes for multicasting hard deadline constrained prioritized data in wireless multimedia streaming," *IEEE Trans. Wireless Commun.*, vol. 15, no. 3, pp. 1631–1641, Mar. 2016.

[21] D. Lun, M. Médard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," *Phys. Commun.*, vol. 1, no. 1, pp. 3–20, 2008.

[22] M. Shaqfeh, A. Zafar, H. Alnuweiri, and M. S. Alouini, "Joint opportunistic scheduling and network coding for bidirectional relay channel," in *Proc. IEEE Int. Symp. Inf. Theory Proc.*, 2013, pp. 1327–1331.

[23] H. Khamfroush, D. E. Lucani, and J. Barros, "Network coding for wireless cooperative networks: Simple rules, near-optimal delay," in *Proc. IEEE Int. Conf. Commun. Workshops*, 2014, pp. 255–260.

[24] K. Sundaresan and S. Rangarajan, "Cooperation versus multiplexing: Multicast scheduling algorithms for OFDMA relay networks," *IEEE/ACM Trans. Netw.*, vol. 22, no. 3, pp. 756–769, Jun. 2014.

[25] S. Meyn, *Control Techniques for Complex Networks*. Cambridge, U.K. : Cambridge Univ. Press, 2008.

[26] N. Bäuerle, "Asymptotic optimality of tracking policies in stochastic networks," *Ann. Appl. Probability*, vol. 10, pp. 1065–1083, 2000.

[27] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting, "Scheduling in a queuing system with asynchronously varying service rates," *Probability Eng. Inf. Sci.*, vol. 18, no. 02, pp. 191–217, 2004.

[28] M. R. Garey and D. S. Johnson, *Computers and Intractability*, vol. 29. San Francisco, CA, USA: Freeman, 2002.

[29] C. Chen and S. Baek, "Channel-aware multicast scheduling algorithms for relay networks using rateless codes," (2015). [Online]. Available: http://sing.korea.ac.kr/papers/coded_relay_techreport_2.pdf

[30] C. Maglaras, "Discrete-review policies for scheduling stochastic networks: Trajectory tracking and fluid-scale asymptotic optimality," *Ann. Appl. Probability*, vol. 10, no. 3, pp. 897–929, 2000.

[31] B. Makki and T. Eriksson, "Multiuser diversity in correlated rayleigh-fading channels," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, no. 1, pp. 1–9, 2012.

Chao Chen received the BS degree in electrical engineering from the Harbin Institute of Technology, in 2009 and the PhD degree in electrical engineering from Korea University, in 2015. Since December 2015, he is in the School of Information and Electronic Engineering, Zhejiang Gongshang University, China, where he currently is a lecturer. His research interests include network coding and optimization problems in communication networks.

Seung Jun Baek (S'04–M'09) received the BS degree from Seoul National University, in 1998, and the MS and PhD degrees from the University of Texas at Austin, in 2002 and 2007, respectively, in electrical and computer engineering. From 2007 to 2009, he was a member of technical staff with DSP Systems R&D Center, Texas Instruments. In 2009, he joined the College of Information and Communications, Korea University, Korea, where he is currently an associate professor. His research interests include modeling and analysis of communication networks, cellular networks, wireless ad-hoc and sensor networks, game theory, and compressive sensing. He is a member of the IEEE.