High Throughput Low Delay Wireless Multicast via Multi-Channel Moving Window Codes

Fei Wu[†], Yin Sun[§], Lu Chen[†], Jackie Xu[†], Kannan Srinivasan[†], and Ness B. Shroff[†‡]
[†]Dept. of CSE, [‡]Dept. of ECE, The Ohio State University, Columbus, OH
[§]Dept. of ECE, Auburn University, Auburn, AL

Abstract—A fundamental challenge in wireless multicast has been how to simultaneously achieve high-throughput and low-delay for reliably serving a large number of users. In this paper, we show how to harness substantial throughput and delay gains by exploiting multi-channel resources. We develop a new scheme called **Multi-Channel Moving Window Codes (MC-MWC)** for multi-channel multi-session wireless multicast. The salient features of MC-MWC are three-fold. (i) High throughput: we show that MC-MWC achieves order-optimal throughput in the many-user many-channel asymptotic regime. Moreover, the number of channels required by a conventional channel-allocation based scheme is shown to be doubly-exponentially larger than that required by MC-MWC. (ii) Low delay: using large deviations theory, we show that the delay of MC-MWC decreases linearly with the number of channels, while the delay reduction of conventional schemes is no more than a finite constant. (iii) Low feedback overhead: the feedback overhead of MC-MWC is a constant that is independent of both the number of receivers in each session and the number of sessions in the network. Finally, our trace-driven simulation and numerical results validate the analytical results and show that the implementation complexity of MC-MWC is low.

I. INTRODUCTION

Mobile video is expected to contribute 75% of all the mobile traffic by 2020 [1]. Wireless multicast, by leveraging the shared nature of the wireless medium, could be an efficient way to distribute popular videos to many users. However, wireless multicast has not been widely deployed in practical systems, because it has been difficult to simultaneously achieve high throughput, low delay, and low feedback overhead for serving a large number of users.

Throughput challenge: In a wireless communication system, the channel conditions of different receivers are heterogeneous due to multipath, shadowing, mobility, etc. Since the throughput of multicast is bottlenecked by the receiver with the worst channel condition, as the number of receivers grows, the achievable throughput in multicast vanishes, offsetting the multicast gain.

Delay challenge: Existing multicast coding schemes incur large delay at the receivers. There are two categories of multicast coding schemes in the literature. The first category of schemes employ a block coding strategy, e.g., random linear network coding (RLNC) [2], LT codes [3], and Raptor codes [4], etc. However, to maintain a non-diminishing throughput,

the block size has to grow on the order[1] $\Omega(\log n)$ as the number of receivers n grows [5]. Since the decoding delay increases with the block length, such schemes are known to have a poor delay performance. The second category of schemes achieve a lower delay through an incremental network coding design, e.g., [6–10]. The data packets participate in the coding procedure progressively, and hence the receivers are able to decode packets progressively, leading to a lower decoding delay. However, when the traffic load ρ is high (close to 1), the average delay increases dramatically on the order of $\Theta\left(\frac{1}{(1-\rho)^2}\right)$ [6, 8].

Feedback challenge: To achieve reliable multicast or to improve the delay performance (e.g., [2–4, 6–10]), the transmitter needs to collect channel state information or reception status reports (such as ACKs/NAKs) from the receivers through feedback. Conventionally, the feedback overhead increases linearly with the number of receivers, e.g., [2–4, 7–10], which becomes a system bottleneck when the multicast sessions have a large number of receivers.

In modern wireless networks, multi-channel communications have become commonplace. For instance, 4G/LTE mobile wireless networks are based on OFDM and OFDMA, where a wide spectrum band is divided into many resource blocks, each with 180kHz [11]. The 802.11a standard can have 12 orthogonal channels in the 5GHz band. The availability of multiple channels provides significant flexibility in designing wireless resource allocation algorithms.

In order to address the above three fundamental challenges, we develop a multi-channel multicast code design in this paper, which can simultaneously achieve high throughput, low delay[2], and low feedback overhead. The contributions of this paper are summarized as follows.

- We propose Multi-channel Moving Window Codes (MC-MWC). The key idea behind MC-MWC is a simple Merging strategy, through which multiple multicast sessions can be jointly served by the shared multi-channel

---

[1]We use the standard order notation: for two real-valued sequences $\{x_n\}$ and $\{y_n\}$, $x_n = o(y_n)$ if $\lim_{n\to\infty} x_n/y_n = 0$; and $x_n = \omega(y_n)$ if $\lim_{n\to\infty} x_n/y_n = \infty$; and $x_n = \Omega(y_n)$ if $\lim_{n\to\infty} x_n/y_n \geq z$ for some constant $z > 0$; and $x_n = \Theta(y_n)$ if $z_1 \leq \lim_{n\to\infty} x_n/y_n \leq z_2$ for some constants $z_1 > 0$ and $z_2 > 0$.

[2]In this paper, we consider the end-to-end delay (including queueing delay and transmission delay), which is measured from the time a packet arrives at the transmitter to the time that the packet is decoded at the receiver. In comparison, the delay metric considered in e.g., [5, 6, 9, 10] only accounts for the transmission delay.

resources.

- High Throughput: We focus on a many-user many-channel asymptotic regime. First, we derive an algorithm independent lower bound on the number of channels needed for achieving any non-vanishing throughput as the number of receivers increases. Then, we show that MC-MWC achieves the lower bound in an order sense. Hence, MC-MWC achieves order-optimal throughput in the many-user many-channel asymptotic regime. Furthermore, we prove that the number of channels required by a conventional scheme based on the optimal static channel allocation and capacity-achieving codes is doubly-exponentially larger than that required by MC-MWC.
- Low Delay: Using large deviations theory, we show that the delay of MC-MWC decreases linearly as the number of channels grows, while the delay reduction of conventional channel-allocation based schemes (even incorporating with any coding schemes) is no more than a finite constant. To the best of our knowledge, MC-MWC *is the first wireless multicast scheme in which the delay decreases linearly as the number of channels grows, with no loss in the per-channel-throughput.*
- Low Feedback Overhead: By combining MC-MWC with the recently proposed anonymous feedback technique [6, 12], the total feedback overhead of MC-MWC is a constant, independent of not only the number of receivers in each session but also the number of sessions in the network.
- Trace-driven and numerical results are provided to validate the analytical results and show that 1) MC-MWC achieves significant throughput and delay improvements in the practical scenarios even when the number of users (channels) is not large, 2) the implementation complexity of MC-MWC is low in practice.

II. RELATED WORK

In wireless unicast, exploiting the multi-channel resources has been extensively studied. For example, in one line of work, e.g., [13, 14], the multi-channel resources are allocated based on the queue lengths of the users such that throughput optimality is achieved. However, throughput optimality does not necessarily imply good delay performance. Thus, another line of work, e.g., [15, 16] proposed delay-based scheduling policies, which achieve optimal throughput and provably good delay performance for multi-channel wireless unicast.

In wireless multicast, multiple receivers may receive the same transmitted packet, and as a result, the work-conservation principle assumed in wireless unicast is violated [17]. In addition, rateless codes or network codes, e.g., [2–4, 6–10]. are typically used to achieve good throughput performance. With coding, the receivers need additional time to decode the packets and thus the queueing delay considered in wireless unicast does not fully capture the overall delay. For the above two reasons, the methodologies developed for multi-channel wireless unicast are not applicable to the multicast scenario.

Existing studies about resource allocation in wireless multicast are either channel-statistics based or channel-state based. In the first category, e.g., [18, 19], it is assumed that the transmitter has access to the channel statistics of the receivers. To realize cooperative multicast, the authors in [18] proposed a network-coding based multicast scheme, in which the scheduling policy depends on the channel statistics of the receivers. In [19], to maximize the long-term throughput, the channels are allocated statically according to the channel statistics. The second category, e.g., [9, 17] assumes that the transmitter has perfect knowledge of the channel state information of the receivers and exploits the opportunistic gain in throughput or delay. A throughput-optimal policy was developed in [17] for wireless multicast with dynamic topology. In [9], the authors proposed an instantly decodable network coding scheme which achieves zero decoding delay at the cost of throughput loss. However, collecting channel state information from all receivers incurs a prohibitive large feedback overhead, rendering them impractical when multicast sessions have a large number of receivers. An open question is *how do we obtain a linear delay reduction in the number of channels without loss in the per-channel throughput?*

III. SYSTEM MODEL

We consider a multi-channel, multi-session wireless multicast network with m orthogonal channels and m multicast sessions, as shown in Figure 1. As in [13, 15, 16], for ease of presentation, we assume that the number of channels is equal to the number of sessions. Our throughput and delay analysis can be readily generalized to the scenario when the number of channels scales linearly with the number of sessions.

A. Multicast Sessions

For a multicast session $h \in \{1, \ldots, m\}$, the transmitter needs to send a stream of data packets to a set of receivers, denoted as $\mathcal{S}_h$. For any two different sessions h and h', their sets of receivers $\mathcal{S}_h$ and $\mathcal{S}_{h'}$ are allowed to have arbitrary intersections, i.e., a receiver may be interested in multiple sessions. Let $s_h = |\mathcal{S}_h| \geq 1$ denote the number of receivers in the session h. The number of receivers in different sessions may be different and we assume that $s_1, \ldots, s_m$ are *i.i.d.* across different sessions[3]. Let $\widehat{s}^{(n)}$ denote a random variable which has the same distribution as s_h, where n represents the expected number of receivers in one session, i.e., $n = \mathbb{E}[s_h]$.

In our model, time is slotted. The packets which need to be served in each session arrive stochastically. Let $a_h[t] \in \mathbb{N}$ denote the number of packet arrivals from session h at the beginning of time-slot t. We assume that $\{a_h[t]\}_{t \in \mathbb{N}}$ are stationary and bounded random variables, independent across time-slots, independent of the number of receivers $\{s_h\}_h$. We use $\lambda_h \triangleq \mathbb{E}[a_h[t]]$ to denote the expected packet arrival rate in session h.

---

[3]This assumption can be relaxed such that $\{s_h\}_h$ have heterogeneous distributions. The details are omitted due to space limitations.
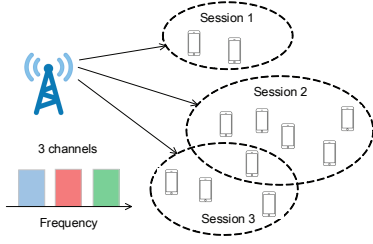
Fig. 1. Multi-channel, multi-session wireless multicast.

Fig. 2. Multi-channel Moving Window Codes.

B. Channel Model

The channel between the transmitter and the receivers is assumed to follow the standard broadcast erasure channel model. Let $c_{i,j}[t]$ denote the channel state of receiver i on channel j in time-slot t, given by

$$c_{i,j}[t] = \begin{cases} 1 & \text{if the packet sent on channel } j \text{ can be} \\ & \text{successfully received by receiver } i \\ & \text{in time-slot } t; \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

As in [2–10], it is assumed that $c_{i,j}[t]$ are *i.i.d.* across different time-slots, and let $\gamma_{i,j} \triangleq \mathbb{P}(C_{i,j}[t] = 1)$ denote the probability of successful packet reception at receiver i on channel j. We make the following assumption on $\{\gamma_{i,j}\}$.

Assumption (Heterogeneous Channel Conditions): The channel statistics $\{\gamma_{i,j}\}$ are random variables that have heterogeneous values for different receivers and different channels. Specifically, $\gamma_{i,j}$ are *i.i.d.*[4] across receivers and across channels. Let $F_\gamma(\cdot)$ and $\bar{\gamma} \triangleq \mathbb{E}[\gamma_{i,j}]$ denote the the CDF and the expectation of $\gamma_{i,j}$, respectively. Moreover, there exists $\kappa > 0$ such that $\lim_{y \to 0+} \frac{F_\gamma(y)}{y} \geq \kappa$.

The above assumptions are more general than the homogeneous network assumptions in [5, 7, 8, 10], where all $\gamma_{i,j}$ are assumed to be equal and thus fail to reflect the heterogeneous channel conditions in reality. In addition, the existence of κ is fairly weak, since it basically says that there is a nonzero probability that $\gamma_{i,j}$ is close to 0. In Section VII, we will provide simulation results that are obtained using data traces collected from a software defined radio platform. We will see that the analytical results obtained under the above assumptions are also valid in practice.

IV. MULTI-CHANNEL MOVING WINDOW CODES

In this section, we propose a Multi-Channel Moving Window Codes (MC-MWC) approach, which is built on the recent work Moving Window Codes (MWC) [6]. Compared with MWC, MC-MWC has two key novelties. First, MWC is designed for the single-channel single-session multicast, and actually we can show that a straightforward generalization of MWC to the multi-channel multi-session multicast setting will lead to poor throughput and poor delay performance. Second, owning to a new Merging strategy, MC-MWC simultaneously
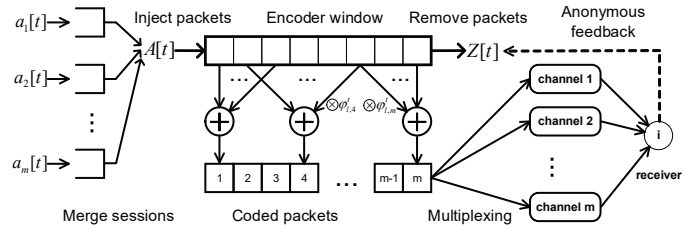
achieves good throughput and delay performance by exploiting the multi-channel resources. In the following, we first introduce a Merging strategy, and then propose MC-MWC.

A. A Simple Merging Strategy

The key idea behind MC-MWC is a simple Merging strategy.

Definition 1: (Merging) The m multicast sessions are jointly served by the multi-channel resources as follows:

Tx's side : At the transmitter, the streams of packets from all m sessions are merged to form one large multicast session. Then, the transmitter uses the m channels to send the grouped multicast session to all receivers in $\bigcup_{1 \leq h \leq m} \mathcal{S}_h$.

Rx's side : In each time-slot, a receiver $i \in \bigcup_{1 \leq h \leq m} \mathcal{S}_h$ listens on all the m channels to receive as many packets as it can. Then, the received packets are used to decode the merged multicast session, including the session(s) receiver i is interested in.

Notice that Merging only describes some high-level multicast transmission rules. The coding, transmission, and feedback procedures of a specific example of Merging will be explained next in detail.

In practice, the sessions can be divided into groups and Merging is applied to the sessions within each group. The choice of group size with Merging depends on the performance requirements on throughput, delay, and implementation complexity, as will be systematically analyzed and shown in Sections V, VI, and VII.

B. Multi-Channel Moving Window Codes

As shown in Figure 2, MC-MWC is comprised of three modules, i.e., the encoder at the transmitter side, the decoder at the receiver side, and the feedback modules.

1) Encoder: At the transmitter, all the packets from all sessions $\{1, \ldots, m\}$ are grouped together to form one large multicast session. The packets in the merged multicast session are indexed as $p_1, p_2, \ldots$, where the packets are indexed according to the order of their arrivals and the tie breaking rule for the packets arriving at the same time-slot is arbitrary. All the newly-arrived packets at the beginning of time-slot t are instantly injected[5] to an encoder window. At time-slot t, the total number of packet arrivals from all the m sessions is denoted as $a[t] \triangleq \sum_{h=1}^{m} a_h[t]$. Then, the number of packets

---

[4]Note that while $\{\gamma_{i,j}\}_{i,j}$ are *i.i.d.*, they correspond to the probability of successful packet reception of receiver i on channel j. So, these probabilities can be different for different realizations of $\gamma_{i,j}$, hence the channel conditions are heterogeneous.

[5]The injection rate has to be within the capacity region of MC-MWC, which will be discussed and analyzed in Section V.

that the encoder has received up to the beginning of time-slot t is $A[t]$, i.e.,

$$A[t] = \sum_{\tau=1}^{t} a[\tau] = \sum_{\tau=1}^{t} \sum_{h=1}^{m} a_h[t]. \qquad (2)$$

To prevent the encoder window from growing indefinitely over time, at the beginning of time-slot t, $Z[t]$ packets which are "determinable"[6] at all receivers $\bigcup_{1 \le h \le m} \mathcal{S}_h$ ($Z[t]$ can be determined via a low overhead feedback mechanism described later) are removed from the encoder window.

At time-slot t, the encoder generates m coded packets through linear combinations of the data packets in the encoder window, which are transmitted over the m channels to all receivers $\bigcup_{1 \le h \le m} \mathcal{S}_h$. Specifically, the coded packet $x_j[t]$ which is transmitted on the channel $j \in \{1, \dots, m\}$ in time-slot t is generated by

$$x_j[t] = \sum_{l=Z[t]+1}^{A[t]} \varphi_{l,j}^t \times p_l, \qquad (3)$$

where p_l denotes the lth packet of the merged session, "$\times$" is the product operator on a Galois field $GF(2^q)$, and $\left\{ \varphi_{l,j}^t \right\}_{\forall l,j,t}$ are independently drawn according to a uniform distribution on $\{GF(2^q)\}$. The values of $A[t]$ and $Z[t]$ are embedded in the packet header of $x_j[t]$. Moreover, $\left\{ \varphi_{l,j}^t \right\}_{\forall l,j,t}$ can be known at each receiver by feeding the same seed to the random number generators of the transmitter and all the receivers.

2) Decoder: With Merging, a receiver $i \in \bigcup_{1 \le h \le m} \mathcal{S}_h$ receives all information sent on the m channels and decode all the m sessions including the session(s) it is interested in, i.e., $\{h : i \in \mathcal{S}_h\}$.

To facilitate a clear understanding of the decoding procedure, we restate the definition of *"determinable" packet* that was originally defined in [7].

Definition 2: ("Determinable" packet) A packet p_l is said to be "determinable" at a receiver if the receiver has enough information to express p_l as a linear combination of some packets $p_{l+1}, p_{l+2}, \dots$ with greater indices, and "indeterminable" otherwise.

Let $S_i[t] \in \mathbb{N}$ be the number of "determinable" packets at receiver i by the end of time-slot t. Define a *virtual decoder queue*

$$Q_i[t] = A[t] - S_i[t] \qquad (4)$$

for each receiver $i \in \bigcup_{1 \le h \le m} \mathcal{S}_h$. Then, $Q_i[t]$ is the number of packets which have arrived from one of the m sessions but are "indeterminable" at receiver i at the end of time-slot t.

Suppose at the beginning of time-slot t, the packets $p_1, \dots, p_{l_1}$ are "determinable" at receiver i and the total number of packets that the encoder received is $A[t] = l_2$. Similar to [6, 7], when the field size 2^q is sufficiently large, for every successfully received packet over the m channels at time-slot

---

[6]"Determinable" will be formally defined later in the decoder module.

---

t, with a high probability, the next "indeterminable" packet becomes "determinable" until all the packets $p_1, \dots, p_{l_2}$ are "determinable". To better see this, consider the following simple example. The receiver i receives 3 coded packets from channels $2, 3, 5$ in time-slot 1, i.e., $x_2[1] = p_1 + p_2 + 2p_3$, $x_3[1] = p_1 + 2p_2 + p_3$ and $x_5[1] = 2p_1 + p_2 + 2p_3$. From the first packet, $p_1 = x_2[1] - p_2 - 2p_3$, thus, by definition, p_1 becomes "determinable". Similarly, from the second and third packets, we have $p_2 = x_3[1] - x_2[1] + p_3$ and $4p_3 = 3x_2[1] - x_3[1] - x_5[1]$, by which p_2 and p_3 are "determinable" in turn. Therefore, the evolution of $S_i[t]$ is given by

$$S_i[t] = S_i[t-1] + \min \left\{ \sum_{j=1}^{m} c_{i,j}[t], A[t] - S_i[t-1] \right\}. \qquad (5)$$

Combining Equations (4), (5) and letting $c_i[t] \triangleq \sum_{j=1}^{m} c_{i,j}[t]$, the evolution of $Q_i[t]$ can be expressed as:

$$Q_i[t] = \left\{ Q_i[t-1] + \sum_{h=1}^{m} a_h[t] - \sum_{j=1}^{m} c_{i,j}[t] \right\}^{+}$$
$$= \{ Q_i[t-1] + a[t] - c_i[t] \}^{+}. \qquad (6)$$

Similar to [6, 7], decoding occurs at receiver i in time-slot t if and only if all the packets, which have arrived from one of the m sessions are "determinable" at receiver i, i.e.,

$$Q_i[t] = 0, \qquad (7)$$

when all packets $p_1, \dots, p_{A[t]}$ could be decoded.

3) Feedback: Recall that to prevent the encoder window from growing indefinitely over time, $Z[t]$ packets are removed from the encoder window at the beginning of time-slot t. To achieve reliable multicast, these $Z[t]$ packets must be "determinable" at all receivers $\bigcup_{1 \le h \le m} \mathcal{S}_h$ at the beginning of time-slot t, and hence $Z[t]$ must be controlled based on the status feedback from all the receivers. To avoid the prohibitively large overhead caused by traditional per-receiver feedback mechanisms, we modify the anonymous feedback mechanism proposed in the recent work [6, 12] to ensure reliable multicast reception at all receivers $i \in \bigcup_{1 \le h \le m} \mathcal{S}_h$ with a negligible feedback overhead.

Anonymous feedback was devised for a single multicast session [6, 12], which guarantees reliable multicast with constant feedback overhead, regardless of the number of receivers. However, without Merging, applying anonymous feedback to the multi-session multicast would incur an overhead which increases with the number of sessions.

In MC-MWC, anonymous feedback can be applied to the merged session. The key idea is to let the receiver(s), for which the oldest m packets in the encoder are not all "determinable", send a NAK in a shared feedback channel at the end of each time slot. This feedback is "anonymous" in the sense that the transmitter does not differentiate the actual ID(s) of the receiver(s) sending NAK. As long as the transmitter can detect the existence of NAK signal from feedback, it will keep the

oldest m packet in the encoder buffer; otherwise, the oldest m packets are removed from the buffer. This mechanism ensures that

$$Z[t] \leq \min_{i \in \bigcup_{1 \leq h \leq m} \mathcal{S}_h} S_i[t-1], \qquad (8)$$

which guarantees reliable multicast. *The anonymous feedback in MC-MWC only requires a short sub-slot to detect the existence of NAK in the shared feedback channel. Thus, the total feedback overhead of MC-MWC is* **a constant**, *not only independent of the number of receivers in each session, but also independent of the number of sessions in the network.* Compared with conventional schemes, e.g., [2–4, 7–10], the feedback overhead of MC-MWC is $\Theta(mn)$ times smaller. Compared with MWC [6], the feedback overhead of MC-MWC is m times smaller. Therefore, MC-MWC effectively reduces the feedback overhead when there are a large number of sessions/receivers. In [12], it is shown that anonymous feedback can be easily implemented and incurs a low overhead in practice.

V. HIGH THROUGHPUT OF MC-MWC

In this section, we analyze the throughput performance of MC-MWC. Noting that multicast is most beneficial when there are a large number of receivers, we focus on a scenario when both the number of receivers and the number of channels scale to infinity. We refer to this setting as the *many-user many-channel asymptotic regime*. We emphasize that although our analysis is in the asymptotic regime, the results provide important insights on the practical scenario when the number of users (channels) is not large (e.g., only 10-20 users per session), as illustrated in Section VII. To facilitate the analysis in this regime, we make the following assumption on the distribution of the number of receivers, i.e., $\{\widehat{s}^{(n)}\}_{n \in \mathcal{N}}$. There is a function $z_s(\cdot)$ such that for any $\alpha \geq 0$ and for any $n \in \mathbb{N}$,

$$\mathbb{P}\left(\frac{\widehat{s}^{(n)}}{\mathbb{E}[\widehat{s}^{(n)}]} \leq \alpha \right) \leq z_s(\alpha), \qquad (9)$$

and

$$\lim_{\alpha \to 0} z_s(\alpha) = 0. \qquad (10)$$

Note that this is a mild assumption. For instance, the assumption is satisfied when $\widehat{s}^{(n)}$ follows common discrete distributions such as geometric distribution, discrete uniform distribution, Poisson distribution, degenerate distribution, etc.

A. Algorithm Independent Lower Bound

In the single-channel single-session multicast case, the achievable throughput is bottlenecked by the receiver with the worst channel condition. As a result, with the number of receivers increasing, it is more and more likely that there exists one receiver whose channel condition happens to be poor, leading to a vanishing multicast throughput. We are interested in understanding the following fundamental question *in a multi-channel, multi-session multicast: how many multi-channel resources are required to achieve a non-vanishing per-*

session throughput? We answer the question by deriving an algorithm independent scaling law on the number of channels required to achieve a non-vanishing throughput as the number of receivers increases.

First, given m, n and the channel statistics $\Gamma = \{\gamma_{i,j}\}_{i,j}$, we define the full capacity region as the set of arrival rate vectors that can be stabilized by some multicast scheme that has the instantaneous channel state information (CSI) of all receivers at the transmitter.

$$\Lambda_{\text{Full}}^{(m,n,\Gamma)} = \left\{ \vec{\lambda}^{(m,n,\Gamma)} = \left(\lambda_1^{(m,n,\Gamma)}, \ldots, \lambda_m^{(m,n,\Gamma)} \right) \, \Big| \right.$$
$$\vec{\lambda}^{(m,n,\Gamma)} \text{ can be stabilized by some}$$
$$\left. \text{multicast scheme with CSI at } \mathsf{Tx}\text{'s side.} \right\}. \quad (11)$$

Note that it is well known that without coding, $\Lambda_{\text{Full}}^{(m,n,\Gamma)}$ can be expressed as the convex hull of all feasible schedules. Nevertheless, with the possibility of inter-session coding, it is generally very difficult to explicitly characterize $\Lambda_{\text{Full}}^{(m,n,\Gamma)}$ even when the channel statistics $\Gamma = \{\gamma_{i,j}\}_{i,j}$ is given. Instead, we derive a fundamental property of $\Lambda_{\text{Full}}^{(m,n,\Gamma)}$ in the many-user many-channel asymptotic regime.

For any session h, given the channel statistics $\Gamma = \{\gamma_{i,j}\}_{i,j}$, define the maximum achievable throughput of session h in the full capacity region $\Lambda_{\text{Full}}^{(m,n,\Gamma)}$ as

$$\overline{\lambda}_h^{(m,n,\Gamma)} \triangleq \max_{\vec{\lambda}^{(m,n,\Gamma)} \in \Lambda_{\text{Full}}^{(m,n,\Gamma)}} \lambda_h^{(m,n,\Gamma)} \qquad (12)$$

Then we have the following theorem regarding $\overline{\lambda}_h^{(m,n,\Gamma)}$.

Theorem 1: If the number of channels scales slower than logarithmically[7] with the expected number of receivers, i.e., $m = O\left((\log n)^\delta\right)$ for some $\delta \in (0,1)$, then the achievable throughput of any multicast scheme vanishes as $n \to \infty$:

$$\overline{\lambda}_h^{(m,n,\Gamma)} \xrightarrow{P} 0, \quad \forall h \in \{1, \ldots, m\}, \qquad (13)$$

in which $\xrightarrow{P}$ denotes "convergence in probability".

Due to space limitations, the proof of Theorem 1 is relegated in our online technical report (please see [20]).

Remark 1.1: Theorem 1 suggests that, to achieve any non-vanishing throughput, the number of channels must scale at least logarithmically with the expected number of receivers, i.e., $m = \Omega(\log n)$.

B. Order Throughput Optimality of MC-MWC

Now we analyze the achievable throughput of MC-MWC. Given the number of receivers in each session $\{s_h\}_{1 \leq h \leq m}$ as well as the channel statistics $\{\gamma_{i,j}\}_{i,j}$, the maximum achievable throughput of MC-MWC can be easily derived as follows. For a receiver i, it receives information on all the m channels, the sum capacity of which is $\sum_{j=1}^m \gamma_{i,j}$. Recall the capacity of multicast is limited by the worst receiver, the

---

[7]In this paper, $\log(\cdot)$ denotes the natural logarithm.

maximum sum throughput of all sessions is then given by

$$\sum_{h=1}^{m} \lambda_h \leq \min_{i \in \bigcup_{h=1}^{m} \mathcal{S}_h} \sum_{j=1}^{m} \gamma_{i,j}. \quad (14)$$

On the other hand, according to Equation (6), $\{Q_i[t]\}_t$ is a random walk on $[0, \infty)$, and has a steady state distribution if Equation (14) holds. Hence, Equation (14) is achievable by MC-MWC.

Similar to the definition of $\Lambda_{\text{Full}}^{(m,n,\Gamma)}$, we define the capacity region of MC-MWC.

$$\Lambda_{\text{MC-MWC}}^{(m,n,\Gamma)} = \left\{ \vec{\lambda}^{(m,n,\Gamma)} \middle| \vec{\lambda}^{(m,n,\Gamma)} \text{ stabilizable by MC-MWC} \right\}.$$

To characterize $\Lambda_{\text{MC-MWC}}^{(m,n,\Gamma)}$, let us define a convex polytope $\mathcal{P}_\lambda^{(m)} \in \mathcal{R}^m$ for a given m and any given $\lambda < \bar{\gamma}$ as

$$\mathcal{P}_\lambda^{(m)} = \left\{ \vec{\lambda}^{(m)} = \left(\lambda_1^{(m)}, \ldots, \lambda_m^{(m)} \right) \middle| \right.$$
$$\left. \frac{1}{m} \sum_{h=1}^{m} \lambda_h^{(m)} \leq \lambda, \lambda_h^{(m)} \geq 0, \forall h \right\}. \quad (15)$$

The polytope $\mathcal{P}_\lambda^{(m)} \in \mathcal{R}^m$ contains all rate vectors such that the average rate is no greater than λ.

Then we derive the following theorem regarding $\Lambda_{\text{MC-MWC}}^{(m,n,\Gamma)}$.

Theorem 2: If the number of channels scale logarithmically with the expected number of receivers, i.e., $m = \left\lceil \frac{1}{(\bar{\gamma}-\lambda)^2} \log n \right\rceil$ for any $\lambda < \bar{\gamma}$, then the achievable per-session throughput of MC-MWC is lower bounded by λ in the following sense:

$$\lim_{n \to \infty} \mathbb{P}\left(\mathcal{P}_\lambda^{(m)} \subseteq \Lambda_{\text{MC-MWC}}^{(m,n,\Gamma)} \right) = 1. \quad (16)$$

Proof sketch of Theorem 2: Let us prove that, with MC-MWC, for any $p \in (0,1)$, if $m = \left\lceil \frac{1}{(\bar{\gamma}-\lambda)^2} \log n \right\rceil$, then $\mathcal{P}_\lambda^{(m)} \subseteq \Lambda_{\text{MC-MWC}}^{(m,n,\Gamma)}$ with probability no less than p, as $n \to \infty$.

First, based on Equations (6) and (14), we show that,

$$\mathbb{P}\left(\mathcal{P}_\lambda^{(m)} \subseteq \Lambda_{\text{MC-MWC}}^{(m,n,\Gamma)} \right) = \mathbb{P}\left(\frac{1}{m} \min_{i \in \bigcup_{h=1}^{m} \mathcal{S}_h} \sum_{j=1}^{m} \gamma_{i,j} \geq \lambda \right). \quad (17)$$

Note that for any $\tau \geq 1$, it can be shown that

$$\mathbb{P}\left(\left| \bigcup_{h=1}^{m} \mathcal{S}_h \right| \geq \tau m n \right) \leq \frac{1}{\tau}. \quad (18)$$

Then, using Hoeffding's inequality, we derive a lower bound of Equation (17).

$$\mathbb{P}\left(\frac{1}{m} \min_{i \in \bigcup_{h=1}^{m} \mathcal{S}_h} \sum_{j=1}^{m} \gamma_{i,j} \geq \lambda \right)$$
$$\geq \left(1 - \frac{1}{\tau} \right) \left(1 - \lceil \tau m n \rceil e^{-2(\bar{\gamma}-\lambda)^2 m} \right). \quad (19)$$

For any $p \in [0,1)$, let $\tau = \frac{2}{1-p}$ and $m = \left\lceil \frac{1}{(\bar{\gamma}-\lambda)^2} \log n \right\rceil$.

Combining with Equations (17) and (19), we have

$$\mathbb{P}\left(\mathcal{P}_\lambda^{(m)} \subseteq \Lambda_{\text{MC-MWC}}^{(m,n,\Gamma)} \right) \geq \frac{p+1}{2} (1 - o(1)), \quad (20)$$

where $o(1)$ converges to 0 as $n \to \infty$.

Noting that $\frac{p+1}{2} \geq p$ and p can be arbitrarily close to 1, the proof is complete.

Due to space limitations, we provide the proof details in our online technical report (please see [20]). ∎

Remark 2.1: Equation (16) suggests that if the number of channels scales logarithmically with the number of receivers, with high probability MC-MWC can stabilize all the rate vectors such that the average rate is less than $\bar{\gamma}$.

Remark 2.2: Comparing Theorem 2 with Theorem 1, we can see that to achieve a non-vanishing throughput, the number of channels required by MC-MWC achieves the algorithm independent lower bound in an order sense. Hence, MC-MWC achieves order-optimal throughput in the many-user many-channel asymptotic regime.

C. Throughput Gain over a Conventional Scheme

Finally, we compare MC-MWC with a conventional scheme, which can be considered as a straightforward extension of [18, 19] to the multi-channel, multi-session multicast setting. First, let us define capacity-achieving code.

Definition 3: (Capacity-achieving code) A coding scheme is said capacity-achieving if it can achieve the capacity of the broadcast erasure channel.

The conventional scheme is based on the following static channel allocation.

Static Channel Allocation: In the static channel allocation, the transmitter allocates the channels according to the channel statistics $\Gamma = \{\gamma_{i,j}\}_{i,j}$. Let $\vec{g} = (g_1, \ldots, g_m)$: $\{1, 2, \ldots, m\} \mapsto \{1, 2, \ldots, m\}$ represent a one-to-one mapping from the sessions $\{h\}_{1 \leq h \leq m}$ to the channels $\{j\}_{1 \leq j \leq m}$. The static channel allocation is to find a $\vec{g} = (g_1, \ldots, g_m)$ such that the long-term sum throughput of all sessions is maximized. Note that given a static channel allocation $\vec{g}$, the maximum achievable throughput of any session h is bottlenecked by the receiver with the worst channel condition on the channel g_h, i.e., $\min_{i \in \mathcal{S}_h} \gamma_{i,g_h}$. Then, the optimal static channel allocation is formulated by

$$\max_{\vec{g}=(g_1,\ldots,g_m)} \left(\sum_{h=1}^{m} \min_{i \in \mathcal{S}_h} \gamma_{i,g_h} \right). \quad (21)$$

Given the optimal channel allocation $\vec{g}^*$ to Equation (21), the transmitter encodes the packets from session h using a capacity-achieving coding scheme, such as rateless codes (e.g., [3, 4]) and some network coding schemes (e.g., [2, 7]), and sends the coded packets over the channel $\vec{g}_h^*$.

With the conventional scheme, given m, n and the channel statistics $\Gamma = \{\gamma_{i,j}\}_{i,j}$, there is a maximum achievable throughput for each session h, denoted as $\bar{\lambda}_h^{(m,n,\Gamma,\text{static})}$. We derive the following theorem regarding $\bar{\lambda}_h^{(m,n,\Gamma,\text{static})}$.

Theorem 3: If the number of channels scales slower than exponentially with the expected number of receivers, i.e., $m =$

$o(v^n)$ for all $v > 1$, then the achievable throughput of the optimal static channel allocation with capacity-achieving codes vanishes as $n \to \infty$, that is,

$$\overline{\lambda}_h^{(m,n,\Gamma,\text{static})} \xrightarrow{P} 0, \quad \forall h \in \{1,\ldots,m\}. \tag{22}$$

Please see [20] for proof details.

Remark 3.1: Theorem 3 suggests that, to achieve a non-vanishing throughput, the number of channels must scale at least exponentially with the expected number of receivers, i.e., there exists some $v > 1$ such that $m = \Omega(v^n)$.

Remark 3.2: Comparing Theorem 3 with Theorem 2, it can be observed that to achieve a non-vanishing throughput, the multi-channel resource required by the optimal static channel allocation with capacity-achieving codes is doubly-exponentially larger than that required by MC-MWC.

VI. LOW DELAY OF MC-MWC

In this section, we show that MC-MWC achieves significant delay reduction by exploiting the multi-channel resources. We focus on the uniform traffic scenario when the arrival rates for all sessions are equal, i.e., $\lambda_h = \lambda, \forall h$.

Without loss of generality, we focus on analyzing the delay performance of an arbitrary session h. Let p_l^h be the lth packet in session h. The delay of the packet p_l^h with respect to a receiver $i \in \mathcal{S}_h$ is defined as the time between the arrival of the packet at the transmitter to the decoding of p_l^h and all the packets in session h with smaller indices, denoted as $D_{i,l}$.

Then, assuming the system is stationary and ergodic, for receiver $i \in \mathcal{S}_h$, the delay violation probability that the delay of a packet exceeds a threshold k is given by

$$\mathbb{P}(D_i > k) = \lim_{L \to \infty} \frac{1}{L} \sum_{l=1}^{L} 1_{\{D_{i,l} > k\}}. \tag{23}$$

We first analyze the delay performance of MC-MWC.

Let the time-slots t_i^d $(d = 1, 2, \ldots)$ be the decoding moments of receiver i satisfying (7). Suppose that packet p_l^h arrives at time-slot t, which is between two successive decoding moments $t_i^d < t \leq t_i^{d+1}$. Then, packet p_l^h and all packets with smaller indices in session h will be decoded in time-slot t_i^{d+1}. The delay of packet p_l^h at the receiver i is

$$D_{i,l} = t_i^{d+1} - t. \tag{24}$$

The following theorem characterizes the delay experienced at a receiver i, which is shown to be independent of the channel statistics of the other receivers and n.

Theorem 4: For a receiver $i \in \mathcal{S}_h$ such that $\sum_{j=1}^{m} \gamma_{i,j} > m\lambda$, the asymptotic decay rate of the delay violation probability of MC-MWC is

$$-\lim_{k \to \infty} \frac{1}{k} \log \mathbb{P}(D_i > k) = \Phi_i^m,$$

where

$$\Phi_i^m = \sup_{\theta \in \mathbb{R}} \left\{ -m \log \mathbb{E}\left(e^{-\theta a_h[t]}\right) - \sum_{j=1}^{m} \log\left(\gamma_{i,j}e^\theta + 1 - \gamma_{i,j}\right) \right\}. \tag{25}$$

In addition, for any $\lambda < \bar{\gamma}$, with probability 1 we have

$$\lim_{m \to \infty} \frac{\Phi_i^m}{m} = \sup_{\theta \in \mathbb{R}} \left\{ -\log \mathbb{E}\left(e^{-\theta a_h[t]}\right) - \mathbb{E}\left[\log\left(\gamma_{i,j}e^\theta + 1 - \gamma_{i,j}\right)\right] \right\}. \tag{26}$$

Proof sketch of Theorem 4: We focus on receiver 1. Define $I_d \triangleq t_1^{d+1} - t_1^d$ as the interval between the dth decoding moment and the $(d+1)^{\text{th}}$ decoding moment, which can be expressed as

$$I_d = \min\left\{ t \geq 1 : \sum_{\tau = t_1^d + 1}^{t_1^d + t} \left(\sum_{h=1}^{m} a_h[\tau] - \sum_{j=1}^{m} c_{1,j}[\tau] \right) \leq 0 \right\},$$

indicating that the decoding process is a renewal process. Noting that analyzing D_1 directly is very difficult, in the first step, we connect $\mathbb{P}(D_1 > k)$ with $\mathbb{P}(I_d > b)$.

Lemma 1: $\mathbb{P}(D_1 > k)$ is upper and lower bounded by

$$\frac{\mathbb{P}(I_d > k)}{m\lambda \mathbb{E}[I_d]} \leq \mathbb{P}(D_1 > k) \leq \frac{k\mathbb{P}(I_d > k) + \sum_{b=k}^{\infty} \mathbb{P}(I_d > b)}{\lambda \mathbb{E}[I_d]}.$$

Then, using large deviations theory, we derive the decay rate for I_d.

Lemma 2: The decay rate of the decoding interval in the steady state is given by

$$-\lim_{b \to \infty} \frac{1}{b} \log \mathbb{P}(I_d > b) = \Phi_1^m, \tag{27}$$

where Φ_1^m is the rate function defined in Equation (25).

By combining Lemma 1 and Lemma 2, the decay rate of D_1 is derived.

Due to space limitations, please see our online technical report [20] for proof details. ∎

Remark 4.1: Theorem 4 suggests that with MC-MWC, $\mathbb{P}(D_i > k) \approx e^{-\Phi_i^m k}$. From Equation (26), the delay improvement with MC-MWC is asymptotically linear in the number of channels m. Hence, MC-MWC could dramatically reduce delay by leveraging the multi-channel resources.

Next, we show a lower bound on the delay performance of a general class of conventional schemes defined as follows.

Definition 4: (Channel-allocation based schemes) A multicast scheme is said to be channel-allocation based, if different sessions are allocated to and served by different channels. The channel allocation can be either static over time (as formulated by Equation (21)), or dynamically adapted across time-slots based on the instantaneous channel state information of all the receivers and all time-slots. To reduce the technical difficulties, it is assumed that one session can be allocated with at most one channel in each time-slot, which makes sense considering that there are m channels and m sessions with $\lambda_h = \lambda, \forall h$. Given the channel allocation, the transmitter may serve the multicast sessions by employing any possible coding scheme, including but not limited to rateless codes and network codes, e.g., [2–4, 6–10].

Theorem 5: When $\mathbb{P}(a_h[t] > 1) > 0$, for any receiver $i \in \mathcal{S}_h$, the decay rate of delay in any channel-allocation based

scheme is upper bounded by

$$- \lim_{k \to \infty} \frac{1}{k} \log \mathbb{P}(D_i > k) \leq \Phi^{\text{const}}, \qquad (28)$$

where

$$\Phi^{\text{const}} = \sup \left\{ \theta > 0 : \log \mathbb{E} \left(e^{\theta a_h[t]} \right) < \theta \right\}$$

is a constant independent of the number of channels m.

Please see [20] for proof details.

Remark 5.1: Theorem 5 suggests that the conventional channel-allocation based schemes (even incorporating any coding schemes), can achieve at most a constant delay improvement (not a function of the number of channels) by exploring the multi-channel resources.

Remark 5.2: The intuition behind Theorem 5 is as follows. For a queueing system, the delay of packets typically originates from two sources: 1) the stochastic and bursty arrival process; 2) the stochastic service process. Note that with a channel-allocation based scheme, a receiver can receive at most one packet from any given session in one time-slot. Even when the receiver could always receive one packet in each time-slot, there is a lower bound of the delay violation probability which originates from the stochastic and bursty arrival process. The lower bound is independent of the number of channels. However, owing to Merging, MC-MWC is not limited to the assumption that a receiver can receive at most one packet from any given session in one time-slot.

VII. SIMULATIONS

In this section, we provide trace-driven and numerical results to investigate the throughput, delay and implementation complexity performance of MC-MWC.

A. Throughput Performance

To evaluate the throughput performance of MC-MWC in practice, we collect traces from experiments on a software defined radio platform. We use NI PXIe-1082 platform with NI-5791 RF front end for the experiments. The carrier frequency is set to be 2.5 GHz and the bandwidth used is 40 MHz. Experiments are performed in an indoor lab environment to get the CSI measurements. We note that the effects of multipath is obvious: The SNR difference across this 40 MHz bandwidth can be more than 20 dB. Hence, we collect the CSI traces at 100 different client locations and use these traces to emulate multicast receivers. Groups are formed randomly among these receivers. Packet reception probability is calculated from the subband CSI traces using the method in [21].

Figure 3 shows the CDF of throughput under different schemes, where 5 groups are formed with on average 20 receivers in each group and 1000 random group formations are performed to get the figure. Here we consider the optimal static channel allocation, as well as a random channel allocation strategy, which assigns channels to sessions randomly. Both the optimal static channel allocation and random channel allocation schemes incorporate random linear network codes (RLNC) [2], which is one capacity-achieving code. We can
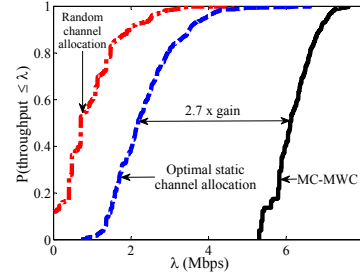

Fig. 3. Multicast throughput in trace-driven simulations with 5 sessions, in which each session has on average 20 receivers.
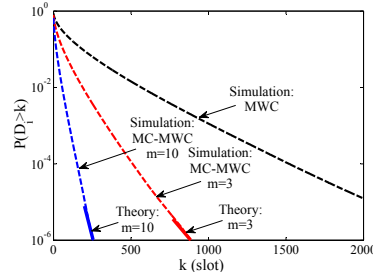

Fig. 4. Delay violation probability.

see that MC-MWC shows a 2.7x gain over optimal static channel allocation scheme, which addresses the concern that: *in real world, even when the channel conditions are correlated across receivers and channels, and when the number of receivers/channels is not large, the proposed scheme could still have a significant throughput gain.*

B. Delay Performance

Recall that in the heterogeneous network, MC-MWC achieves a larger throughput than the optimal static channel allocation. To make the delay comparison fair, we consider a homogeneous network scenario, i.e., $\gamma_{i,j} = 0.6$ for all receivers and for all channels, in which the maximum achievable throughput of MC-MWC and the optimal static channel allocation are the same. There are 100 receivers in each session. In each session $h \in \{1, \ldots, m\}$, the packets arrive according to a Bernoulli process with rate $\lambda = 0.54$.

Figure 4 plots the delay violation probability of one receiver $i \in \mathcal{S}_h$ with different schemes, i.e., MC-MWC and moving window codes (MWC) [6]. It can be observed that $\mathbb{P}(D_i > k)$ of MC-MWC decays exponentially with k and matches the predicted asymptotic decay rate in Equation (25). Furthermore, the decay rate is linear in the number of channels m, which shows that MC-MWC achieves a significant delay gain over MWC [6] by exploring the multi-channel resources.

Then, we consider the average delay performance of the same network scenario under different arrival rate λ. Figure 5 shows the average delay of different schemes with respect to the traffic load $\rho \triangleq \frac{\lambda}{\gamma_{i,j}}$. From Figure 5, we have the following observations. First, the average delay of MC-MWC is much lower than that of RLNC [2], which is one of the rateless codes. It is worthy to note that the average delay of other rateless codes, such as LT codes [3] and Raptor codes [4], is close to that of RLNC. Second, compared with MWC [6], MC-MWC achieves a delay reduction which is roughly linear
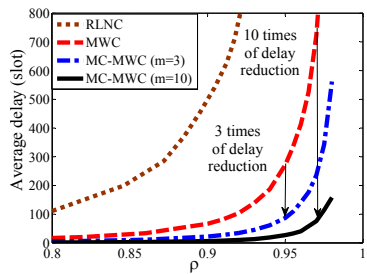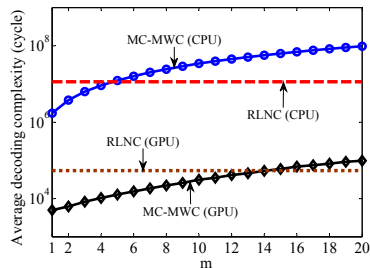
Fig. 5. Average delay vs. the traffic load ρ.

Fig. 6. Average decoding complexity vs. m.

in the number of channels m. Third, the delay gain holds for any load $\rho \in (0, 1)$.

C. Low Implementation Complexity

With MC-MWC, a receiver needs to decode the sessions it may not be interested in. Hence, it is important to understand the decoding complexity of MC-MWC. To this end, we emulate the decoding procedure of MC-MWC on the commercial CPU and GPU.

In the emulation, the finite field size in MC-MWC is set to be 256, and we use a simple table loop-up approach [22] to do the operations on the finite field. Each packet has 1024 bytes. We emulate the decoding process at a receiver with $\gamma_{i,j} = 0.6$ for all channels, where packets arrive according to a Bernoulli process with rate $\lambda = 0.48$ such that $\rho = 0.8$. We emulate two versions of the decoding algorithm, the serial version on a single core CPU (Intel Core i7-2600 CPU clocked at 3.4 GHz), and the parallel version on a GPU (NVIDIA GeForce GTX 860M). Note that the GPU provides 640 shader units clocked at 1029 MHz, and we use 16 blocks and 64 threads per blocks in CUDA. For comparison, we also emulate RLNC [2] with a block length of 50 packets.

In Figure 6, we plot the average decoding complexity (in terms of cycles) of different schemes to decode one packet in a session. From Figure 6, we have the following observations. First, as m grows, the decoding complexity of MC-MWC increases on both CPU and GPU, which is consistent with our expectation that the decoding complexity with MC-MWC increases with m. Second, for both MC-MWC and RLNC, GPU leverages its ability to decode packets in parallel, and thus dramatically reduce the decoding complexity. Third, the decoding complexity of MC-MWC is smaller than that of RLNC when m is small. This is because the moving window coding strategy results in a sparser decoding matrix compared with the dense decoding matrix in RLNC, as shown in [6, 23]. Finally, MC-MWC can support 10 sessions, the throughput of each is more than $100Mbps$ on our GPU. Therefore,

the decoding complexity of MC-MWC is low in practice for moderate m, in which case MC-MWC already exhibits significant throughput and delay gains as shown previously.

VIII. Conclusion

In this paper, we develop a Multi-Channel Moving Window Codes (MC-MWC) and prove that it achieves high throughput, low delay, and requires very limited feedback. We verify our theoretical results using trace-driven simulations, and show that the complexity of implementing MC-MWC is low in practice. This new approach, which exploits multi-channel capability, moving window coding, and anonymous feedback, has the potential to finally realize in practice the significant promise of wireless multicast.

References

[1] "Cisco visual networking index: Forecast and methodology, 2015-2020." http://www.cisco.com.

[2] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, pp. 4413–4430, Oct. 2006.

[3] M. Luby, "LT codes," in *IEEE FOCS 2002*, pp. 271–280, 2002.

[4] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, pp. 2551–2567, Jun. 2006.

[5] Y. Yang and N. Shroff, "Throughput of rateless codes over broadcast erasure channels," *IEEE/ACM Trans. Netw.*, vol. 23, no. 1, pp. 126–137, 2015.

[6] F. Wu, Y. Sun, Y. Yang, K. Srinivasan, and N. Shroff, "Constant-delay and constant-feedback moving window network coding for wireless multicast: Design and asymptotic analysis," *IEEE J. Select. Areas Commun.*, vol. 33, pp. 127–140, Feb 2015.

[7] J. K. Sundararajan, D. Shah, and M. Médard, "ARQ for network coding," in *IEEE ISIT 2008*, pp. 1651–1655, Jul. 2008.

[8] J. K. Sundararajan, D. Shah, and M. Médard, "Feedback-based online network coding," *CoRR*, vol. abs/0904.1730, 2009.

[9] S. Parastoo, S. Ramtin, and T. Danail, "An optimal adaptive network coding scheme for minimizing decoding delay in broadcast erasure channels," *EURASIP J. Wirel. Commun. Netw.*, vol. 2010, Jan. 2010.

[10] A. Fu, P. Sadeghi, and M. Medard, "Dynamic rate adaptation for improved throughput and delay in wireless network coded broadcast," *IEEE/ACM Trans. Netw.*, in press.

[11] H. Holma and A. Toskala, *LTE for UMTS-OFDMA and SC-FDMA based radio access*. John Wiley & Sons, 2009.

[12] F. Wu, Y. Yang, O. Zhang, K. Srinivasan, and N. B. Shroff, "Anonymous-query based rate control for wireless multicast: Approaching optimality with constant feedback," ACM MobiHoc, pp. 191–200, 2016.

[13] S. Bodas, S. Shakkottai, L. Ying, and R. Srikant, "Low-complexity scheduling algorithms for multichannel downlink wireless networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 5, pp. 1608–1621, 2012.

[14] M. Ouyang and L. Ying, "Approaching throughput optimality with limited feedback in multichannel wireless downlink networks," *IEEE/ACM Trans. Netw.*, vol. 21, pp. 1827–1838, Dec. 2013.

[15] B. Ji, G. R. Gupta, X. Lin, and N. B. Shroff, "Low-complexity scheduling policies for achieving throughput and asymptotic delay optimality in multichannel wireless networks," *IEEE/ACM Trans. Netw.*, vol. 22, no. 6, pp. 1911–1924, 2014.

[16] B. Ji, G. R. Gupta, M. Sharma, X. Lin, and N. B. Shroff, "Achieving optimal throughput and near-optimal asymptotic delay performance in multichannel wireless networks with low complexity: a practical greedy scheduling policy," *IEEE/ACM Trans. Netw.*, vol. 23, no. 3, pp. 880–893, 2015.

[17] A. Sinha, L. Tassiulas, and E. Modiano, "Throughput-optimal broadcast in wireless networks with dynamic topology," ACM MobiHoc '16, pp. 21–30, 2016.

[18] D. Koutsonikolas, Y. C. Hu, and C.-C. Wang, "Pacifier: High-throughput, reliable multicast without "crying babies" in wireless mesh networks," *IEEE/ACM Trans. Netw.*, vol. 20, pp. 1375–1388, Oct. 2012.

[19] K. C.-J. Lin and D.-N. Yang, "Multicast with intraflow network coding in multirate multichannel wireless mesh networks," *IEEE Trans.on Veh. Technol.*, vol. 62, no. 8, pp. 3913–3927, 2013.

[20] F. Wu, Y. Sun, L. Chen, J. Xu, K. Srinivasan, and N. B. Shroff, "High throughput low delay wireless multicast via multi-channel moving window codes." https://arxiv.org/abs/1801.01613, Jan. 2018.

[21] W. Zhou, T. Das, L. Chen, K. Srinivasan, and P. Sinha, "Basic: backbone-assisted successive interference cancellation," ACM MobiCom, pp. 149–161, 2016.

[22] M. A. Hasan, "Look-up table-based large finite field multiplication in memory constrained cryptosystems," *IEEE Trans. on Comput.*, vol. 49, pp. 749–758, Jul 2000.

[23] A. Tassi, I. Chatzigeorgiou, and D. E. Lucani, "Analysis and optimization of sparse random linear network coding for reliable multicast services," *IEEE Trans. on Commun.*, vol. 64, no. 1, pp. 285–299, 2016.