Dynamic Rate Adaptation for Improved Throughput and Delay in Wireless Network Coded Broadcast

Amy Fu, Parastoo Sadeghi, *Senior Member, IEEE*, and Muriel Médard, *Fellow, IEEE*

Abstract—In this paper, we provide theoretical and simulation-based study of the delivery delay performance of a number of existing throughput-optimal coding schemes and use the results to design a new dynamic rate adaptation scheme that achieves improved overall throughput–delay performance. Under a baseline rate control scheme, the receivers' delay performance is examined. Based on their Markov states, the knowledge difference between the sender and receiver, three distinct methods for packet delivery are identified: zero state, leader state, and coefficient-based delivery. We provide analyses of each of these and show that, in many cases, zero state delivery alone presents a tractable approximation of the expected packet delivery behavior. Interestingly, while coefficient-based delivery has so far been treated as a secondary effect in the literature, we find that the choice of coefficients is extremely important in determining the delay, and a well-chosen encoding scheme can, in fact, contribute a significant improvement to the delivery delay. Based on our delivery delay model, we develop a dynamic rate adaptation scheme that uses performance prediction models to determine the sender transmission rate. Surprisingly, taking this approach leads us to the simple conclusion that the sender should regulate its addition rate based on the total number of undelivered packets stored at the receivers. We show that despite its simplicity, our proposed dynamic rate adaptation scheme results in noticeably improved throughput–delay performance over existing schemes in the literature.

Index Terms—Broadcast, delay, in-order packet delivery, network coding, rate control, throughput.

I. INTRODUCTION

IN RECENT times, there have been many advances in the capabilities of wireless communication systems [1]. A number of applications can now take advantage of these new capabilities, requiring high data rate and low delay performance. In this paper, we consider applications in which the same ordered set of packets is required by all receivers, with low delay. One application is video broadcasting where the receivers not only wish to watch live video, but also want to keep a high-quality copy for later use. This might include the broadcast of a lecture or conference recording, or perhaps simultaneous download and viewing of a purchased movie. Other potential applications include the broadcast of common information in multiplayer gaming, where users' actions must be logged in order, as well as certain scientific or mission-critical applications with low delay requirements. This differs from work such as [2]–[4] in that the applications considered in this paper do not tolerate packet losses.

In the context of these applications, there are two key measures of performance. One measure is *throughput*, defined as the average rate at which packets are delivered across receivers. This measures the efficiency with which the receivers' channel bandwidth is utilized. Since packets can only be used in order, we can only consider a packet useful once it has been *delivered*, that is if it and all preceding packets have also been correctly received. Low delay is also desirable, to avoid latency at the application. Therefore, it is equally important to minimize the *delivery delay*, the average time between when a packet is first available for transmission to the time it is delivered to the application layer.

Meeting these requirements in a wireless setting is not an easy task [5]. Receivers' independent channel conditions mean that they will experience very different erasure patterns, which in turn leads to a variety of packet demands on the sender.

A. Network Coding

Linear network coding [5]–[7] is used as an effective way to accommodate multiple receivers' packet demands while still efficiently using the transmission bandwidth. Under linear network coding, the sender divides the information into equal-sized packets and combines a number of packets into each transmission using Galois field arithmetic [7]. This combination is transmitted to the receivers along with the *coefficients* used to combine the packets. In order to recover the original packets, receivers must collect enough coded packets to decode them using Gaussian elimination [7].

Although network coding is known to enhance the throughput in many networks, the time spent waiting to receive the necessary packet combinations for decoding can result in an additional *decoding delay*. There are two problems associated with a large decoding delay. First, the decoding delay lower-bounds the achievable delivery delay since packets can only be delivered after being decoded. Second, undecoded packets can greatly increase the computational complexity of operations for both the sender and receiver. Gaussian elimination, required for receivers to decode, is known to scale as the cube of the number of packets in the set. In *full feedback* systems, the sender performs similar operations to determine

A. Fu and P. Sadeghi are with the Research School of Engineering, College of Engineering and Computer Science, The Australian National University, Canberra, 0200 ACT, Australia (e-mail: amy.fu@anu.edu.au; parastoo.sadeghi@anu.edu.au).

M. Médard is with the Department of Electrical Engineering and Computer Science, Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: medard@mit.edu).

what information is missing at the receivers. Large decoding delays mean that, on average, many undecoded packets will be stored at the receivers, resulting in more computationally expensive packet transmissions.

Network coding introduces a well-known tradeoff [8]–[12] between throughput and delivery delay. Generally, the more stringent the delay requirements, the more throughput must be sacrificed to achieve them. Many transmission schemes have been devised that aim at striking a balance between high throughput and low delay in network coded systems. We will present an overview of existing approaches in the literature and highlight the open questions that will be addressed in this paper. For brevity, we will focus on broadcast applications in wireless packet erasure channels, as they are directly related to our work.

B. Existing Methods for Delay Control

To ensure that packets can be delivered in a timely fashion, it is necessary to introduce some controlled redundancy into the sender's transmissions. This allows receivers who have experienced channel erasures to recover and deliver their missing packets. The *transmission schemes* used to achieve this can generally be divided into two components: a *rate control scheme* and a *coding scheme*. More detail will be provided in Section II. Essentially, the rate control scheme determines the *transmission rate*, the number of new packets that can be included in the sender's transmissions at each time, while the coding scheme is responsible for determining the coefficients. Each of these components can have an impact on the throughput and delay.

1) Rate Control: There are a number of ways to use rate control to reduce the delivery delay.

Under *block-based* transmission schemes, incoming packets are divided into blocks or generations [12]–[25]. The rate control scheme only allows the packets of one block to be transmitted at a time, ensuring that a block's worth of *innovative information* has been received by every receiver, before moving on to the next block. The primary advantage of this rate control scheme is that since packet delivery is done on a block-by-block basis, shorter block lengths mean smaller delivery delays. However, this comes at the cost of lower (even vanishing) throughput [14]. Another advantage of this rate control scheme is that it requires only minimal feedback from the receivers about block completion [11], [17], [26].

Other transmission schemes such as [10], [27], and [28] are non-block-based. In [10], a rate control scheme is implicitly implemented where new packets may be transmitted only if the delay performance, determined from receiver feedback, is sufficiently good. In contrast, [27] and [28] make little use of feedback in determining the transmission rate. Instead, they use a fixed transmission rate and rely on natural fluctuations in the *transmission queue* size to ensure the delivery of packets. While there has been much work studying the delay performance of block-based transmission schemes [12]–[17], so far only asymptotic limits [29] on the delay performance of [27], [28] have been found.

2) Coding: The coding scheme may be used to further improve the delay performance.

Under some coding schemes, the sender transmits network coded packets that may be noninnovative to selected receivers. A good example of this is *instantaneously decodable* network coding [20]–[24]. In this block-based transmission scheme, feedback about packets stored at the receivers is used to construct transmissions that allow immediate decoding at a subset of (or if possible all) receivers. However, instantaneous decodability comes at the cost of reduced throughput since not every receiver may receive innovative information in every transmission.

By contrast, *throughput-optimal* coding schemes do not attempt to introduce more redundancy, but instead aim to maximize the number of receivers that can obtain innovative information from each transmission. Random linear network coding (RLNC), where coefficients are chosen at random, is the most common. In [30], this was shown to achieve the capacity of a multicast network with high probability as the field size becomes large. The simplicity of implementation has led to a great deal of work including [12]–[19].

Feedback-based throughput-optimal coding schemes have also been proposed to reduce the transmission queue size [28] and minimize the delivery delay [27], however no attempt has been made to study the extent to which these schemes work.

To the best of our knowledge: 1) there has been no work on characterizing the nonasymptotic delivery behavior of the rate control scheme used in [27] and [28]; 2) the delay performance of the coding schemes presented in [27] and [28] has not yet been analyzed; and 3) there has been no systematic attempt to implement a rate control scheme that adaptively considers both the throughput and delay performance in determining the transmission rate.

C. Contributions and Distinctions With Related Work

In this paper, we take a first step in realizing a dynamic tradeoff between throughput and delivery delay in a wireless network coded broadcast system. By first understanding the mechanism by which packets are delivered in transmission schemes such as [27] and [28], we gain insight into the nature of the *throughput–delay tradeoff*, the set of throughput values and delivery delays simultaneously achievable by a system. This in itself is a difficult problem, owing to the complex interactions between the sender and receivers. To manage this, we categorize the methods of packet delivery into three categories: zero state, leader state, and coefficient-based delivery. These distinctions are made on the basis of receivers' Markov states: defined as the difference between the number of packets known by the sender and receiver at each time-step. By decoupling the contributions of each method of delivery, we can present an approximation that removes the effect of cross receiver interactions. In return for some loss of accuracy, we are able to transform a mathematically intractable problem into one that gives easily calculable results.

Based on our understanding of the mechanics of broadcast packet delivery, we propose a new transmission scheme that uses feedback information to predict the receivers' short-term
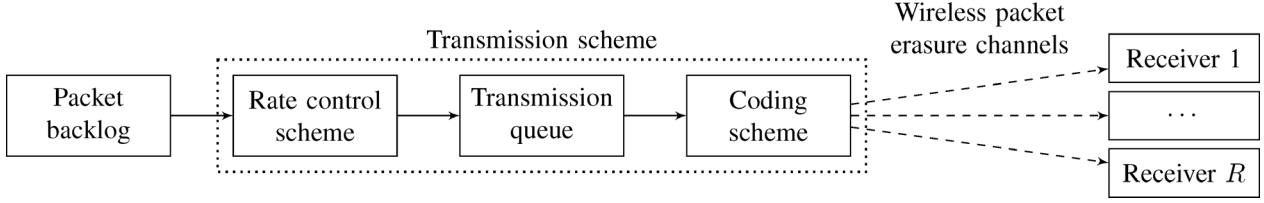
Fig. 1. Block diagram of the components of a transmission scheme.

throughput and delivery performance. This is then used to determine when to include new packets into the sender's transmissions. In effect, the sender dynamically tailors the transmission rate for noticably improved throughput–delay performance compared to [10], [27], and [28]. A related idea is considered in [25], where the block size is chosen to maximize the number of packets delivered to all receivers by a hard deadline. However, as commented in Section I-B.1 block coding is generally not conducive to good throughput.

II. SYSTEM MODEL

A single sender aims to transmit a backlogged set of data packets $\mathbf{p}_1, \mathbf{p}_2, \ldots$ in the correct order to a set of R receivers. Time is slotted, denoted by $t = 1, \ldots$, and the sender can broadcast at the rate of one original or network coded packet per time-slot. The receivers are connected to the sender via independent erasure channels with *channel rate* μ, so that they successfully receive transmissions with probability μ at each time-slot.[1]

Receivers store received packets in a buffer and send an acknowledgment after each successful packet reception or a negative acknowledgment if the packet is discarded owing to an erasure, which we assume the sender detects without error.[2] The sender uses this information to record which packets receivers have stored in their buffers. Based on this information, a *transmission scheme* can be devised to determine the packet combinations the sender will transmit. The components of the transmission schemes we will study will be outlined in the remainder of this section. We now define the *delivery delay* and *throughput*, which will be used to compare the performance of the transmission schemes studied in this paper.

1) Packet Delivery: At time-slot T, a packet $\mathbf{p}_n$ is said to be *delivered* to a receiver if that receiver has already decoded all packets $\mathbf{p}_1, \ldots, \mathbf{p}_{n-1}$ and first decodes $\mathbf{p}_n$ at time $t = T$. Otherwise, $\mathbf{p}_n$ is said to be *undelivered* to that receiver.

2) Delivery Delay: The delivery delay of a transmission scheme is measured as the average number of time-slots between any packet $\mathbf{p}$ becoming available for transmission, to the time it is delivered to each of receivers.

3) Throughput: The throughput of our system is measured as the average number of packets delivered per time-slot, across receivers.

A. Transmission Scheme

Here, we outline the model for the transmission schemes we will be studying, as shown in Fig. 1. The transmission scheme employed by the sender can be divided into three components: a *rate control block*, which passes new packets into a *transmission queue*, from which a *coding block* determines $c(t)$, the network coded transmission to be sent at time t. We briefly outline the function of each block here.

1) Rate Control Block: The rate control block employs a *rate control scheme* to decide when to introduce new packets from the application into the transmission queue. In our paper, we assume the application has an infinite backlog of packets available to be transmitted by the sender. Since the sender transmits one packet per time-slot, we limit the rate control scheme to pass at most one new packet per time-slot to the transmission queue. Therefore, at each time t, the rate control block can decide whether to *add*, and place a new packet in the transmission queue, or *wait* and do nothing. If the rate control block adds, then we set the *add decision* $a(t) = 1$; if it waits, $a(t) = 0$.

2) Transmission Queue: The transmission queue stores all packets passed by the rate control scheme. Only packets in the transmission queue may be transmitted by the sender. Once all receivers have decoded a packet $\mathbf{p}$, it is removed from the transmission queue.[3] At any time t, the total number of packets that have been passed into the transmission queue is

$$A(t) = \sum_{i=1}^{t} a(i). \tag{1}$$

The remainder of this paper will focus on delivering the packets in the transmission queue to the receivers. Therefore, with a slight abuse of notation, the packets in the transmission queue will be referred to, from oldest to newest, as $\mathbf{p}_1, \ldots, \mathbf{p}_{A'(t)}$, where $A'(t)$ is the total number of packets in the transmission queue at time t.

3) Coding Block: The coding block employs a *coding scheme* to determine which of the packets in the transmission queue to code into the outgoing transmission $c(t)$ at each time-slot. Since the coding block may only choose packets from the transmission queue, transmissions are of the form

$$c(t) = \sum_{i=1}^{A'(t)} \alpha_i(t) \mathbf{p}_i \tag{2}$$

[1]In general, receivers may have different channel rates, but for clarity of explanation, we only consider the homogeneous case.

[2]Although this can be difficult to achieve in practice, it greatly simplifies analysis. We will make some comments on the effect of imperfect feedback later in this work.

[3]In [28], packets may be removed from the transmission queue before they are decoded by all receivers. However, we ignore this option, as transmission schemes other than [28] are also considered, and we do not explicitly attempt to manage the queue size.

where the coefficients $\alpha_i(t)$ are chosen at each time-slot from the field $\mathbb{F}_M$ of an appropriate size.[4] This combination is transmitted along with the corresponding *transmission vector* $v_s(t)$. If each uncoded packet $\mathbf{p}_i$ corresponds to the standard basis vector e_i whose ith entry is 1, then

$$v_s(t) = \sum_{i=1}^{A'(t)} \alpha_i(t) e_i$$
$$= [\alpha_1(t), \alpha_2(t), \ldots] \qquad (3)$$

so that the ith entry of the transmission vector $\alpha_i(t)$ corresponds to the coefficient of $\mathbf{p}_i$. Receivers use the information in the transmission vector to recover the original packets by performing Gaussian elimination on the packets in their buffers.

B. Rate Control Schemes

In this paper, we consider three rate control schemes. Two of these, the *delay threshold* and *dynamic* rate control schemes, are both *rate adaptation schemes*, which use feedback from the receivers to adjust their transmission rates. As a means of comparison, we will also study a *baseline* rate control scheme, which does not utilize feedback from the receivers to determine the transmission rate.

1) Baseline Rate Control Scheme: Under this rate control scheme, the add decision $a(t)$ is determined by a Bernoulli process with *addition rate* λ, so that the sender will add with probability $\Pr(a(t) = 1) = \lambda$ independently at each time-slot t. This is equivalent to the model used in [27] and [28].[5] By assuming the load factor $\rho = \lambda/\mu$ is appreciably less than 1, we can provide more practical nonasymptotic analysis of throughput–delay performance.

2) Delay Threshold Scheme: This delay threshold rate control scheme is taken from [10] and will be used as a comparison rate control scheme. This scheme operates under two modes, which we call *start* and *stop*. By default, the sender is set to start mode, where it adds whenever one of the receivers has decoded all packets in the transmission queue. However, if any of the packets inside the transmission queue have been present for more than some threshold T_D number of time-slots, the sender switches to stop mode. In this case, the sender waits, and the coding block transmits uncoded copies of the expired packet(s). Once all packets remaining in the transmission queue are less than T_D time-slots old, the sender reverts back to start mode.

3) Dynamic Rate Control Scheme: In this paper, we will present a rate control scheme that outperforms both the baseline and delay threshold rate control schemes. In Section VI, we shall show how add and wait decisions can be determined using a delivery model based on our transmission scheme analysis.

C. Coding Schemes

In this section, we outline the three throughput-optimal coding schemes we will study in this paper. To highlight the effects of coefficient selection on delay, we will focus on two

---

[4]The coding schemes of [27] and [28] prove that it is always possible to find an *innovative* combination for all receivers if $M \geq R$, the number of receivers.

[5]To be precise, in [27] and [28], packets are assumed to *arrive* at the application by a Bernoulli process. We have transformed this into the equivalent rate control scheme to make it comparable in terms of throughput and delay to the backlogged schemes studied in this paper.

existing schemes, the *drop-when-seen* coding scheme of [28] and [29] and the asymptotically optimal delivery scheme of [27], which we call coding schemes A and B, respectively. As a means of comparison, we also consider an RLNC scheme. Throughput-optimal coding schemes all have the *innovation guarantee property*. This means that, at each time-slot, the transmitted packet $c(t)$ will be innovative for all receivers who are still missing packets in the transmission queue. The method for selecting coefficients in each scheme is summarized below. More details can be found in [27] and [28].

1) Coding Scheme A: Coding scheme A relies on the concept of seen packets. A packet $\mathbf{p}_i$ is *seen* by a receiver if it can use the packets in its buffer to create a combination of the form $\mathbf{p}_i + f(\mathbf{p}_{>i})$, where $f(\mathbf{p}_{>i})$ is some linear combination of the packets $\mathbf{p}_{i+1}, \mathbf{p}_{i+2}, \ldots$. If this is not possible, then $\mathbf{p}_i$ is *unseen*.

Coding scheme A ensures that with each successful packet reception, a receiver sees its next unseen packet. To determine what coded packet to transmit next, the sender lists the oldest unseen packet from each receiver. Moving from oldest to newest unseen packet, it adds an appropriate multiple of each packet, so that the resulting packet is innovative to all the corresponding receivers.

2) Coding Scheme B: Under coding scheme B, the sender transmits a minimal combination based on the oldest undecoded packet of each receiver. The sender lists these oldest undecoded packets and their corresponding receivers, then beginning with the newest packet in the list, it adds in older packets only if the receiver(s) that correspond to them would not otherwise receive an innovative packet.

3) RLNC Scheme: The sender transmits a random combination of all $A'(t)$ packets in the transmission queue. Coefficients are selected randomly from $\mathbb{F}_M$. For a fair comparison, feedback is used to ensure that the final packet satisfies the innovation guarantee property. If the current set of coefficients do not have this property, new random coefficients are chosen until an appropriate combination is found.

It should be noted that, unlike the RLNC scheme, coding schemes A and B are more selective about the packets coded into each transmission. Since coding schemes A and B only code the oldest unseen or undecoded packet of each user, the sender will only code a new packet $\mathbf{p}_i$ if one of the receivers has decoded $\{\mathbf{p}_1, \ldots, \mathbf{p}_{i-1}\}$. As a result, under these coding schemes, the sender codes packets from an *effective transmission queue* that is limited to the next needed packet $\mathbf{p}_n$ of the receiver with the most delivered packets. The role of the effective transmission queue will be further discussed in Section V-A.

D. Baseline and Coding Scheme B Transmission Schemes

A transmission scheme is determined by the pairing of a rate control scheme with a coding scheme. In practice, any combination is allowed. However, to simplify the presentation of this paper, two groups of transmission schemes will be studied. In Sections III–V, we analyze the *baseline transmission schemes*: transmission schemes that substitute the baseline rate control scheme into the rate control block. The baseline rate control scheme is chosen as it is the only rate control scheme for which a mathematically tractable model is possible. It is paired with throughput-optimal coding schemes A, B, and RLNC. In

Sections VI and VII, we study *coding scheme B transmission schemes*, which substitute coding scheme B into the coding block. Coding scheme B is chosen as it has the best delay performance of the three coding schemes. It is paired with each of the baseline, delay threshold, and dynamic rate control schemes.

III. MARKOV STATE

Our delivery delay analysis will be based on the receivers' *Markov states*,[6] a concept we will explain next. This allows us to categorize packet delivery methods and gives us an important tool for the estimation of the receivers' delivery delays.

A. Knowledge Spaces and the Markov State

At time t, the *transmission list* is defined as the set of standard basis vectors $V_s(t) = \{e_1, e_2, \ldots, e_{A'(t)}\}$ corresponding to the uncoded packets $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_{A'(t)}$ that are currently in the transmission queue. The sender chooses packets for transmission from the *transmission knowledge space*

$$K_s(t) = \text{span}(V_s(t)) \tag{4}$$

which is the set of all linear combinations the sender can compute using packets from the transmission queue. The size of the transmission knowledge space is given by

$$|K_s(t)| = M^{|V_s(t)|} \tag{5}$$

where M is the field size and the notation $|X|$ represents the cardinality of the set X. The *reception list* $V_r(t)$ of a receiver r is defined as the set of received transmission vectors that, after Gaussian elimination, correspond to vectors from the current transmission knowledge space $K_s(t)$. The *receiver knowledge space* is similarly defined as the set of all linear combinations that can be calculated from its reception list, $K_r(t) = \text{span}(V_r(t))$. These concepts will be used in our analysis of coefficient-based delivery in Section V.

The *Markov state* of a receiver r is defined as the difference between the size of the transmission list and reception list

$$s_r(t) = |V_s(t)| - |V_r(t)|. \tag{6}$$

It should be noted that the removal of packets from the transmission queue does not affect the Markov state since each packet removal decrements both $|V_s(t)|$ and $|V_r(t)|$.

B. Markov Chain Model

Under the baseline rate control scheme, if $\lambda < \mu$, then changes to a receiver's Markov state over time can be modeled as a traversal through a Markov chain. This is illustrated in Fig. 2, where the states $0, 1, 2, \ldots$ correspond to the values of $s_r(t)$. Whether $s_r(t)$ increases, decreases, or remains the same between time-slots depends on both the add decision $a(t)$ and the receiver's channel conditions. The allowable state transitions for states greater than zero and their probabilities
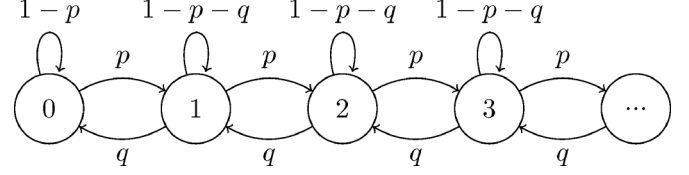
[6]The Markov state is based on the concept of *virtual queue length* in [29].

Fig. 2. Markov chain describing transitions in the Markov state of a receiver r.

TABLE I
PROBABILITY OF TRANSITIONS BETWEEN MARKOV STATES FOR $s_r(t) > 0$, WHERE THE NOTATION $\bar{x} = 1 - x$ IS USED

State transition	Probability	Shorthand notation
$s_r(t+1) = s_r(t) + 1$	$\lambda\bar{\mu}$	p
$s_r(t+1) = s_r(t) - 1$	$\bar{\lambda}\mu$	q
$s_r(t+1) = s_r(t)$	$\lambda\mu + \bar{\lambda}\bar{\mu}$	$1 - p - q$

are listed in Table I. Note that as long as $\lambda < \mu$, the Markov chain is positive recurrent. Although the Markov chain model is perfectly accurate for any receiver considered on its own, the fact that the sender is shared means that receivers' Markov states can exhibit a significant amount of correlation with one another. Nevertheless, this model still provides valuable insight into the delivery delay characteristics of the transmission schemes we will study.

Using the concept of Markov state, we can categorize the ways in which the next packet $\mathbf{p}_n$ of a receiver r can be delivered[7] to a receiver as follows.

1) *Zero state delivery*: Zero state delivery occurs when a receiver r is in the *zero state*, i.e., its Markov state $s_r(t) = 0$. At this point, the size of the reception list equals the size of the transmission list. Since coding schemes A, B, and RLNC all satisfy the innovation guarantee property, any time that $s_r(t) = 0$, all packets in the transmission queue have been delivered.

2) *Leader state delivery*: Under coding schemes A and B, a receiver r is called a *leader* if it has the minimum Markov state, i.e., $s_r(t) = \min\{s_i(t)\}$. Leader state delivery occurs when new packets are delivered by the current leader, although we require that $s_i(t) > 0$ to differentiate this from zero state delivery. In Section II-C, we noted that the effective transmission queue is limited to the receiver with most packets in their buffer. Therefore, as shown in [29], receiving a transmission while a leader results in the delivery of all packets in the effective transmission queue.

3) *Coefficient-based delivery*: Under all three coding schemes, coefficient-based delivery accounts for any packets delivered to a receiver while it is neither leading nor in the zero state. Coefficient-based delivery occurs when the inclusion of the transmission vector $v_s(t)$ into the reception list of a receiver r results in the decoding of the next needed packet $\mathbf{p}_n$. In this case, some fraction of the packets stored at the receiver are delivered.

[7]This categorization is also applicable to decoding without the in-order delivery constraint. Zero and leader state delivery both result in the decoding of all packets stored at the receiver, while *coefficient-based decoding* results in the decoding of some subset of packets stored at the receiver.

C. Distribution of Markov States

Since the Markov state will form the basis of our analysis, the first step is to find the probability $S_r(k)$ that at a randomly selected time, the receiver r is in state k. This is equivalent to finding the stationary distribution of the Markov chain corresponding to that receiver. For the Markov chain of Fig. 2, if the addition rate λ is less than the channel rate μ, a stationary distribution exists such that

$$pS_r(k) = qS_r(k+1). \tag{7}$$

Solving for $\sum_{k=0}^{\infty} S_r(k) = 1$, we obtain

$$S_r(k) = \left(1 - \frac{p}{q}\right)\left(\frac{p}{q}\right)^k. \tag{8}$$

In the following Sections IV and V, we shall analyze the effect of Markov state on the receivers' delivery delay.

IV. Zero and Leader State Delay Analysis

In this section, we study the impact of the zero and leader state delivery on the receivers' delivery delay for the baseline transmission schemes outlined in Section II-D. By using the Markov state to distinguish between different methods of packet delivery, we are able to provide insight into the delivery behavior of these throughput-optimal coding schemes that has so far been missing from the literature. Taking zero state delivery as a first approximation for our delay analysis, we use the Markov chain model of Section III-B to find the distribution of zero state delivery cycles and accurately approximate the expected zero state delivery delay. While leader state delivery has proven an intractable complication in previous analysis, we show how our model can be used to make useful observations about the impact of leader state delivery on the delivery delay.

A. Zero State Delivery

Here, we will estimate the *zero state delivery delay*, defined as the delivery delay experienced if only zero state delivery is permitted. This estimate will be used as an upper bound on the delivery delay for the baseline transmission schemes. It is important to observe that, as long as the innovation guarantee property holds, the Markov state of a receiver depends only on its channel rate μ and the addition rate λ. Therefore, zero state delivery is not affected by the coding scheme, the presence of other receivers, or even the quality of feedback. This independence makes zero state delivery analysis a valuable tool, as initial performance estimates can be made without the intractable complications that have hindered the study of network coded transmission schemes to date.

To find the zero state delivery delay, it is not sufficient to know the proportion of time a receiver spends in the zero state, calculated in (8). The zero state delivery delay depends on the distribution of times between returns to the zero state, which we call *delivery cycles*, and the distribution of transmission queue additions within each cycle. Therefore, we shall use random walk analysis to calculate the distribution of delivery cycle

lengths and, based on this work, find an accurate approximation for the zero state delivery delay of baseline transmission schemes.

1) Delivery Cycle Distributions: A receiver starting in Markov state 0 experiences a *delivery cycle* of length T if its first return to the zero state in the Markov chain occurs after exactly T time-slots. We calculate $P_{0,0}(T)$, the probability that a delivery cycle will be of length T.

We can solve this problem in two steps. First, we characterize a path through the Markov chain that consists of only *moving steps* where $s_r(t+1) = s_r(t) \pm 1$. Then we factor in the effect of *pause steps*, where $s_r(t+1) = s_r(t)$.

In the first time-step, there are two possibilities. The receiver can remain at state 0 with probability $1 - p$, which gives us $P_{0,0}(1) = 1 - p$. If it instead moves up to state 1, it must return to 0 in $T > 1$ time-steps. For a path of fixed length T to start at and return to 0, it must consist of $2k$ moving steps, k up and k down, and $T - 2k$ pause steps, where $1 \leq k \leq \lfloor T/2 \rfloor$. If no other encounters with the zero state are permitted, the first and last time-steps must be up and down steps, respectively. Therefore, the number of paths that first return to the zero state in exactly $2k$ steps without pauses is given by the $(k-1)$th Catalan number [31]

$$C_{k-1} = \frac{1}{k}\binom{2k-2}{k-1}. \tag{9}$$

Now we factor in the $T - 2k$ pauses. These pauses cannot occur in the first or last time-step, otherwise the delivery cycle length would not be T. For a given path of $2k$ moving steps, there are $\binom{T-2}{2k-2}$ choices for pause locations. Therefore, the probability of taking exactly $T > 1$ time-slots to return to the zero state is given by

$$P_{0,0}(T) = \sum_{k=1}^{\lfloor T/2 \rfloor} \frac{1}{k}\binom{2k-2}{k-1}\binom{T-2}{2k-2}p^k q^k (1-p-q)^{T-2k}. \tag{10}$$

The cumulative delivery cycle length probabilities are given for a number of values of λ and μ in Fig. 3. The greater the load factor ρ, the more slowly the probability converges to 1 and the larger the zero state delivery delay.

2) Delay Estimate: Over a delivery cycle of length T, we estimate the number of packets added to the transmission queue and their expected delivery delay. This is combined with (10) to obtain an accurate estimate of the zero state delivery delay.

Where the delivery cycle is of length $T = 1$, the probability that one packet is added and then immediately delivered is simply $\lambda\mu$. Since the packet is immediately delivered, it incurs no delivery delay.

In Section IV-A, we established that, for all other coding cycles with length $T \geq 2$, the Markov state must increase in the first time-slot, and decrease in the last time-slot. Therefore, we must have $a(t) = 1$ in the first time-slot, and $a(t) = 0$ in the last time-slot. We now assume that in the remaining $T - 2$ time-slots, additions occur uniformly with probability λ. Then, the average number of packets delivered over a delivery cycle of length T is estimated to be
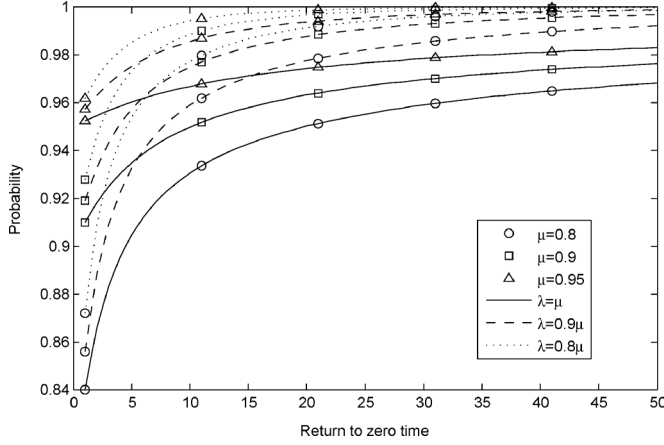
$$1 + \lambda(T - 2) \tag{11}$$

Fig. 3. Probability from (10) of a receiver taking $\leq T$ time-steps to return to the zero state under the baseline rate control scheme.
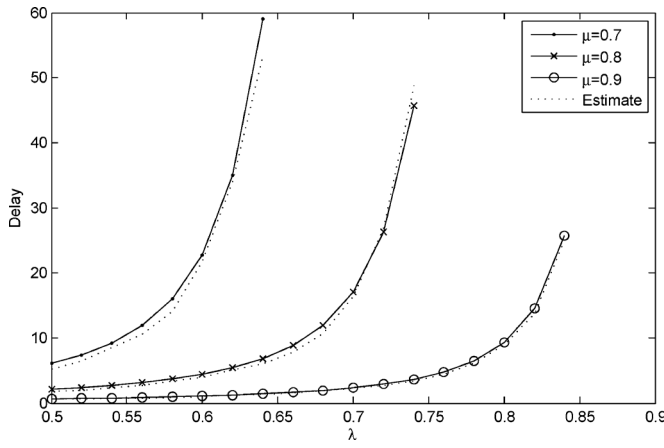

Fig. 4. Zero state delivery delay of the baseline rate control scheme, as a function of the addition rate λ. The delay estimates of (13) (dotted lines) are compared against simulation (solid lines).

and the average zero state delivery delay for each of these packets is $T/2$. The total delay incurred by these packets would then be

$$T + 0.5\lambda T(T - 2). \tag{12}$$

Therefore, the zero state delivery delay, including the $T = 1$ delivery cycle, can be estimated as

$$\frac{\sum_{T=2}^{\infty} P_{0,0}(T)(T + 0.5\lambda T(T - 2))}{\lambda\mu + 1 + \sum_{T=2}^{\infty} P_{0,0}(T)\lambda(T - 2)}. \tag{13}$$

In Fig. 4, we show how our calculated estimate, truncated at $T = 1000$, matches well with the average delivery delays obtained from simulation.

B. Leader State Delivery

In this section, we study the *leader state delivery delay*, defined as the delivery delay experienced if only zero and leader state delivery are allowed. Note that as mentioned in Section II-C, leader state delivery has an equal impact on coding schemes A and B, but does not apply to the RLNC scheme. We investigate the amount of time receivers spend leading and its impact on the delivery delay, compared to zero state delivery on its own.
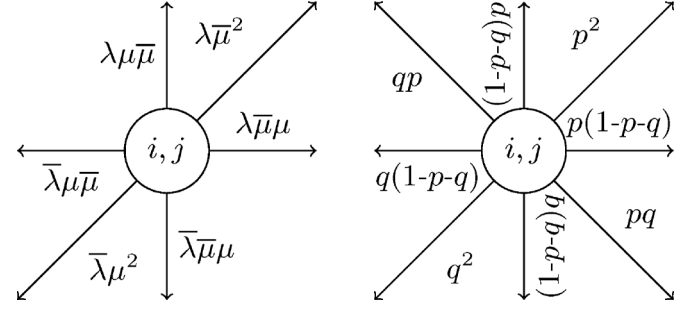

Fig. 5. Two-receiver state transition probabilities (a) in practice and (b) under the independent receiver model. The horizontal and vertical axes correspond to the Markov states of receivers 1 and 2, respectively, where $s_1(t) = i$ and $s_2(t) = j$. Since the sender is common to both receivers, it is not possible that $s_1(t + 1) = i \pm 1$ while $s_2(t + 1) = j \mp 1$.
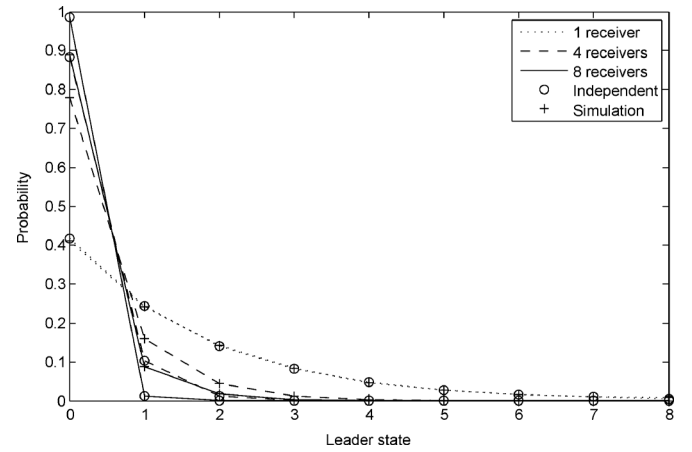

Fig. 6. Under the baseline rate control scheme with addition rate λ, the proportion of time the leader is in each state, under the independent receiver model (15) and in practice. $\lambda = 0.7, \mu = 0.8$.

1) Leader State Distribution: Based on the Markov state distribution calculated in Section III-C, we bound the average time that the leader(s) spend in each Markov state. By (8), the probability of a receiver r being in a state $\geq k$ is

$$S_r (\geq k) = \sum_{i=k}^{\infty} S_r(i) = \left(\frac{p}{q}\right)^k. \tag{14}$$

Hence, if receivers' Markov states were independent, the probability of having a leader in state k would be

$$L(k) = \left(1 - \left(\frac{p}{q}\right)^R\right)\left(\frac{p}{q}\right)^{Rk}. \tag{15}$$

However, since the sender is common to all receivers, there is a noticeable amount of correlation between receivers' Markov states. This is illustrated in Fig. 5, which compares the joint Markov state transition probabilities for two receivers under each model. In practice, the correlated transition probabilities result in the receivers being more closely grouped together than predicted by the independent receiver model. Fig. 6 shows that the probability of leading from states $k > 0$ is higher in practice than under the independent receiver model in (15).

2) Observations: The probability of the leader being in state k is bounded between the values in the single receiver case and the independent receiver model. Therefore, although
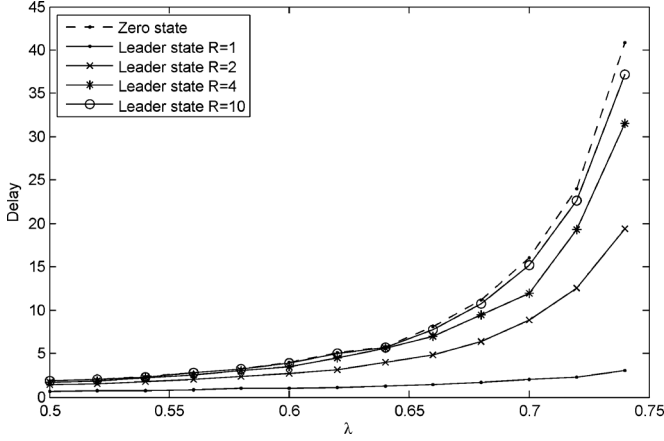
Fig. 7.   Simulated zero state and leader state delivery delays under the baseline rate control scheme with addition rate λ, for different numbers of receivers R. $\mu = 0.8$.

the independent receiver model is not entirely accurate, it can still be used to make the following observations about the leader state.

1) The probability that a receiver r is leading is $\geq 1/R$ since at least one receiver must lead at each time-slot.
2) The leader will most likely be in state $k = 0$. The larger the number of receivers R, the more likely this is the case.
3) The higher a receiver's state k, the lower its likelihood of leading.
4) By (8) and (15), as the load factor $\rho \to 1^-$, or equivalently $\lambda \to \mu^-$, the state probability distribution $S_r(\geq k)$ converges on 1 more slowly. This increases the probability that the leader will be in a state $k > 0$, and therefore the impact leader state delivery has on delay.

We can observe some of these effects in Fig. 7. $R = 1$ represents the extreme case where there is only one receiver who is always leading, and so results in extremely low delivery delays. As R increases, however, the leader state delivery delay quickly converges toward the zero state delivery delay. Even at moderate values of R, for example $R = 10$, the difference between the zero and leader state delivery delay is negligibly small. This behavior can be attributed to observations 1 and 2, made above. By contrast, as the load factor ρ increases, so does the impact of leader state delivery, consistent with observation 4.

Under imperfect feedback conditions, the contribution of leader state delivery would be further diminished since the sender would not always know which packets the leader has received. In order to maintain the innovation guarantee property, the sender would need to account for the possibility the leader has received all packets for which the outcome has not yet been determined. This overestimation of the leader's channel rate would result in an effective transmission queue closer in size to that of the actual transmission queue.

V. Coefficient-Based Delivery

Coefficient-based delivery accounts for any remaining packets delivered while a receiver is neither leading nor in the zero state. The impact of coefficient-based delivery is not well understood because of the difficulty of analyzing its effects. In the literature, it is generally speculated to contribute a small, if not negligible, improvement on the delivery delay. However,

through simulation, we demonstrate two important principles for improving the likelihood of coefficient-based delivery: minimizing the coding field size M, and maintaining sparse codes [i.e., minimizing the number of nonzero coefficients $\alpha_i(t)$ in (2)]. When these conditions are met, coefficient-based delivery can reduce the delivery delay significantly.

Say that at time t the sender transmits a packet $c(t)$ with transmission vector $v_s(t)$. Then, using the concepts from Section III-A, the next needed packet will be delivered if and only if the following condition holds.

Lemma 1: At time t, a receiver can deliver their next needed packet $\mathbf{p}_n$ iff they receive a packet with transmission vector $v_s(t) \in \mathrm{span}(K_r(t-1) \cup e_n) \setminus K_r(t-1)$.

Proof: A packet $\mathbf{p}_k$ is decoded iff $e_k \in K_r(t)$. Say that $e_n \notin K_r(t-1)$. Then, for $\mathbf{p}_n$ to be delivered at time t, $e_n \in \mathrm{span}(K_r(t-1) \cup v_s(t))$. To satisfy the innovation guarantee property, $v_s(t) \notin K_r(t-1)$. Therefore, to deliver packet $\mathbf{p}_n$ at time t, $v_s(t) \in \mathrm{span}(K_r(t-1) \cup e_n) \setminus K_r(t-1)$. ∎

As we shall show, the probability of coefficient-based delivery depends on both the coding scheme used and the *effective Markov state*, which we now define.

A. Effective Markov State

The *effective transmission list* $V_s^*(t)$ is defined as the set of basis vectors corresponding to packets in the effective transmission queue. In the RLNC scheme, typically $V_s^*(t) = V_s(t)$, unless all coefficients selected for the newest packet happen to be 0. In contrast, under coding schemes A and B, the effective transmission queue is limited by the number of packets known by the leading receiver(s), so that

$$|V_s^*(t)| = \min\left(\max_{r \in 1,\ldots,R}(|V_r(t-1)| + 1), |V_s(t)|\right) \quad (16)$$

and $V_s^*(t) = \{e_1, e_2 \ldots, e_{|V_s^*(t)|}\}$. Similarly to (4), the *effective transmission knowledge space* is $K_s^*(t) = \mathrm{span}(V_s^*(t))$. The *effective Markov state* of a receiver r can then be defined as

$$s_r^*(t) = |V_s^*(t)| - |V_r(t-1)|. \quad (17)$$

This differs from (6) in that, in order to calculate the probability that the *current* transmission $c(t)$ will deliver $\mathbf{p}_n$, it compares the effective transmission list to the reception list *prior* to packet receptions in the current time-slot.

Lemma 2: A receiver r can only coefficient-based deliver its next needed packet $\mathbf{p}_n$ when its effective Markov state decreases, i.e., $s_r^*(t) = s_r^*(t-1) - 1$.

Proof: It is always true that $K_r(t-1) \subset K_s^*(t-1)$. If the receiver is not a leader, then they have not decoded all packets in $K_s^*(t-1)$ and $e_n \in K_s^*(t-1)$. Therefore, by Lemma 1, coefficient-based delivery can only occur if $v_s(t) \in K_s^*(t-1)$, so that $V_s^*(t) = V_s^*(t-1)$. Receiving an innovative packet means that $|V_r(t)| = |V_r(t-1)| + 1$, so by (17), $s_r^*(t) = s_r^*(t-1) - 1$. ∎

Hence, in order for a coefficient-based delivery opportunity to arise, three conditions must first be satisfied.

- The receiver is neither a leader nor in the zero state.
- No new packets are encoded by the sender.
- The receiver successfully receives the transmitted packet.

Therefore, of the time-slots a receiver is neither in the zero state or a leader, approximately $\bar{\lambda}\mu$ of these provide an opportunity for coefficient-based delivery to occur. We now investigate

the effectiveness of coding schemes A and B and RLNC in utilizing this fraction of *coefficient-based deliverable* time-slots to minimize delay.

B. RLNC Scheme

To gain some insight into the probability of coefficient-based delivery, we first study the RLNC scheme. Here, we will demonstrate how the effective Markov state affects the probability of coefficient-based delivery.

We first calculate for a single receiver the probability that with receiver knowledge space $K_r(t)$, the next needed packet $\mathbf{p}_n$ will be delivered. The total number of possible transmissions is given by the size of the transmission knowledge space $K_s(t)$ minus the receiver knowledge space. Therefore, by Lemma 1, the probability of selecting a packet under the RLNC scheme that allows $\mathbf{p}_n$ to be delivered is

$$\frac{|\text{span}(K_r(t-1) \cup e_n) \setminus K_r(t-1)|}{|K_s^*(t) \setminus K_r(t-1)|} = \frac{M-1}{M^{s_r^*(t)}-1}. \quad (18)$$

Therefore, the probability of coefficient-based delivery depends only on the effective Markov state of the receiver and the field size M. The exponential dependence on both of these factors means that the coefficient-based delivery probability will be very small for high effective Markov states and large field sizes.

For the multiple-receiver case, simulations show that there is a fairly negligible difference between the RLNC coefficient-based delivery probabilities for the single- and multiple-receiver cases, provided they are coded using the same field size M. Some of these probabilities, normalized over coefficient-based deliverable time-slots, are shown in Fig. 8, and the resulting delay performance for $M = 4$ is given in Fig. 9. As expected, the small coefficient-based delivery probability results in only a slight improvement over zero state delivery.

C. Coding Scheme A

Under coding scheme A, the sender codes only the first unseen packet of each receiver. Furthermore, the coefficients chosen are the smallest that will satisfy the innovation guarantee property.[8] Although a field size $M \geq R$ is necessary to guarantee innovation, the majority of the time coefficients from a much smaller field size $\mathbb{F}_2$ are sufficient.

In Fig. 8, the coefficient-based delivery probability of coding scheme A is compared against the RLNC scheme. Under coding scheme A with four receivers and $\mathbb{F}_4$, the probability of coefficient-based delivery in fact lies between the probabilities for the $\mathbb{F}_2$ and $\mathbb{F}_4$ RLNC scheme. This occurs since in practice the sender usually selects binary field coefficients, effectively coding from the field $\mathbb{F}_2$. With eight receivers and a field $\mathbb{F}_8$, packet coefficients are nearly always selected from the field $\mathbb{F}_4$. This results in coefficient-based delivery probabilities close to the RLNC $\mathbb{F}_4$ case. In Fig. 9, we can observe that coding scheme A has significantly better delay performance compared to the RLNC scheme. This is primarily due to the role of leader state delivery, with a slight contribution from coefficient-based delivery.

[8]In [28], it is suggested that any coefficient satisfying the innovation guarantee is suitable, but in our implementation the smallest allowable coefficient is chosen.
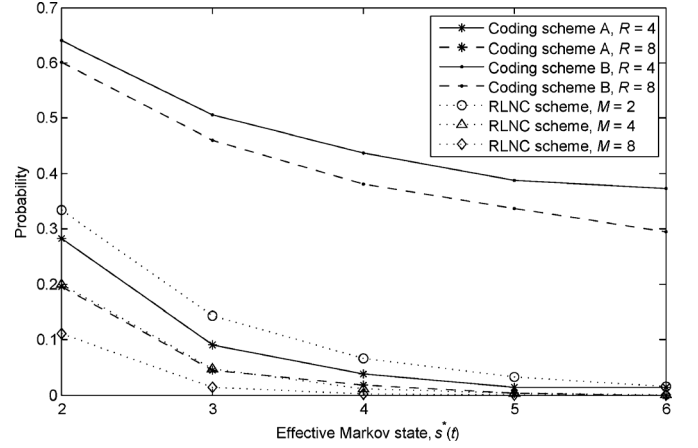

Fig. 8. Probability of coefficient-based delivery as a function of Markov state, for baseline transmission schemes with addition rate $\lambda = 0.7$. Probabilities are normalized over coefficient-based deliverable time-slots. The field size $M = R$ the number of receivers, and $\mu = 0.8$.
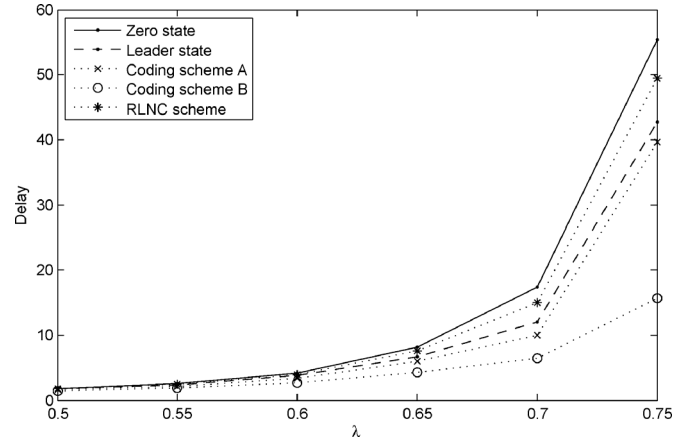

Fig. 9. Average delivery delay of coding schemes A, B, and RLNC under the baseline rate control scheme with addition rate λ. Zero and leader state delivery delays are included for comparison. $R = 4$, $\mu = 0.8$.

D. Coding Scheme B

Coding scheme B, by contrast, attempts to closely mimic a systematic, uncoded scheme by coding additional packets into each transmission only if it is necessary to maintain the innovation guarantee property. Each of these extra packets has the additional property that, if received, the coded transmission will allow the corresponding receiver to deliver their next needed packet.

We can expect that at least λ of the sender's transmissions, corresponding to the first transmission of each new packet, will be uncoded. Fig. 10 agrees with this prediction, and we can observe that the four-receiver case has a slightly higher proportion of uncoded packets compared to the eight-receiver case. This can be attributed to the fact that the smaller the number of receivers, the lower the probability that additional packets need to be included in each transmission.

Coding scheme B is shown in Fig. 8 to have a coefficient-based delivery probability that is significantly higher than coding scheme A and decays more slowly as a function of the effective Markov state. The coefficient-based delivery probability for the eight-receiver case, which has a smaller
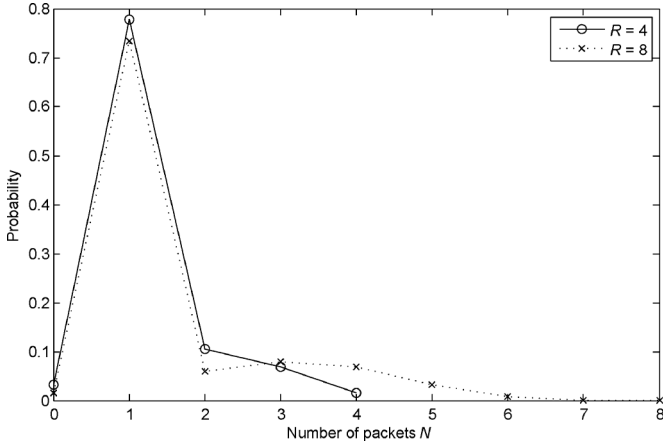
Fig. 10.   For the baseline rate control scheme with coding scheme B, the probability that N packets are coded into a sender transmission. $R = 4, 8, \lambda = 0.7, \mu = 0.8$. Note that if the transmission queue is empty, no packets will be coded.

fraction of uncoded packets, is somewhat less than its four-receiver counterpart. From Fig. 9, we can observe that the higher coefficient-based delivery probabilities of coding scheme B result in significantly better delivery delay compared to both coding scheme A and the RLNC scheme. The improvements are especially notable at high addition rates, with an almost threefold improvement in the delivery delay compared to the leader state delivery delay.

We can give an intuitive explanation for the link between coding sparsity and a higher probability of coefficient-based delivery. If a large fraction of undelivered packets is already decoded, this effectively reduces the size of the system of equations corresponding to unknowns in the receiver's buffer. If the transmitted combination is itself sparse, then there is a good probability that its few nonzero elements are those previously decoded by the receiver. Where the elements corresponding to other receivers are already known, the sender will ensure that the transmitted combination allows the delivery of the receiver's next needed packet.

It should be noted that under coding schemes A and B, infrequent feedback could potentially degrade the delivery delay performance. If the sender were to make decisions based on incomplete information about the contents of receivers' buffers, throughput optimality would only be achievable at the cost of larger field sizes and less sparse coding, both of which would have a detrimental impact on the delivery delay.

VI. DYNAMIC RATE CONTROL SCHEME

In this section, we outline the decision metric used in the *dynamic rate control scheme*. This rate control scheme, which builds upon the Markov state model analyzed in Sections II–V, determines whether the sender adds new packets to the transmission queue or waits, based on the receivers' predicted throughput and delay performance. Using this decision metric, we will demonstrate in Section VII that improved throughput–delay performance can be achieved, compared to both the baseline and delay threshold rate control schemes.

A. Dynamic Delivery Model

The first step is to outline the *dynamic delivery model* used by the dynamic rate control scheme to predict the receivers' throughput and delivery performance. Under the dynamic delivery model, we continue to model sender additions to the transmission queue by a Bernoulli process, as was done for the baseline rate control scheme. Although the dynamic rate control scheme clearly differs from the baseline rate control scheme, the dynamic nature of the rate control scheme means that there is no straightforward way to predict when the sender will choose to add. Therefore, the zero state delivery model continues to model sender additions with a Bernoulli process, replacing the previously known addition rate λ with an *addition rate estimate*, λ_{est}. Using (1), λ_{est} is given by the average observed addition rate, upper-bounded by μ, the maximum possible delivery rate

$$\lambda_{\text{est}} = \min(A(t)/t, \mu - \epsilon) \tag{19}$$

where ϵ is a small, positive value, taken to be $\epsilon = 0.0001$ in our simulations.[9] This is a practical way to choose λ for our dynamic delivery model because it allows the sender to dynamically adjust its transmission rate to accurately reflect its throughput and delay performance priorities. The stability of this system under inaccurate values of λ_{est} is discussed in Section VI-C.2.

The dynamic delivery model also assumes that zero state delivery is the only method by which packets can be delivered to the receivers. It should be noted that this is a somewhat pessimistic performance estimate, as it does not take into account the effects of leader state and coefficient-based delivery studied in Sections IV and V. However, since the effects of leader state and coefficient-based delivery were found to be difficult to predict, the zero state delivery delay provides a tractable upper bound on the expected delivery delay.

In many situations, zero state delivery provides a reasonable estimate of the delivery delay. For moderate to large values of R, leader state delivery has a relatively minor impact, and under coding schemes A and RLNC, the effects of coefficient-based delivery on the delay performance were relatively insignificant. Under imperfect feedback conditions, the contributions from leader and coefficient-based delivery are further diminished. However, the zero state delivery model is not always accurate. If coding scheme B is used, the addition rate $\lambda \to \mu$, or R the number of receivers is small, then leader state and coefficient-based delivery can still have a significant impact on the actual delivery performance.[10]

B. Decision Metric

Now that the dynamic delivery model has been established, we can outline how add and wait decisions are determined in the dynamic rate control scheme. Previously, we studied the baseline transmission schemes, where packets were added to the transmission queue with probability $\lambda < \mu$. By contrast, in the

---

[9]It is of course reasonable to assume that a receiver's delivery rate λ will be less than the channel capacity μ, but the reason for this explicit upper bound will become apparent in (33).

[10]Although in this section we shall use the zero state delivery model to make predictions about the receivers' throughput and delay performance, the performance measurements of Section VII are obtained from the combination of all three delivery methods.

dynamic rate control scheme we are about to introduce, a *decision metric* M is used to determine whether to add or wait. This allows the sender to dynamically adjust its addition rate based on the delivery performance of the receivers, resulting in better throughput–delay performance. The sender calculates M by weighing throughput performance measures $P_T^A(r)$, $P_T^W(r)$ as well as delivery performance measures $P_D^A(r)$ and $P_D^W(r)$ for each receiver r, under the add and wait decisions, respectively.

1) Throughput Performance: Our first task is to measure the throughput performance of a receiver r under the add and wait decisions. Every time the sender waits, there is no increase in the total throughput. Therefore

$$P_T^W(r) = 0. \tag{20}$$

On the other hand, adding increases the total throughput by one packet, giving

$$P_T^A(r) = 1. \tag{21}$$

2) Expected Time to Zero State Delivery: To measure the delivery delay performance, we compare the expected delivery delay for a receiver in Markov state k, under add and wait decisions. Our first task is to calculate the average time it takes for a receiver in Markov state k to zero state deliver. This is equivalent to finding the average time to move from state k to 0 for the first time, under the Markov chain in Fig. 2.

Let E_k be the expected *time to zero state*, i.e., expected number of time-steps it takes for a receiver starting at state k to reach 0 for the first time. Since our Markov chain is positive recurrent, we know that E_k exists.

The receiver's journey from state k to 0 can be considered as a series of traversals through the Markov chain from k to $k - 1$ for the first time, $k - 1$ to $k - 2$ for the first time, and so on. Because the transition probabilities between adjacent states in the Markov chain are the same for all $k > 0$, the average time required for each traversal is the same. Therefore, the expected time to zero state starting from k can be expressed as

$$E_k = kE_1. \tag{22}$$

Studying the Markov chain of Fig. 2, there are three possible transitions at each time-step. Starting at state k, the receiver's Markov state may increment, decrement, or remain the same, with the probabilities listed in Table I adapted to λ_{est}, as calculated in (19). Each state transition corresponds to a one-unit increase in delay. From this information, we can establish the relationship between the time to the zero state for different values of k

$$E_k = 1 + qE_{k-1} + pE_{k+1} + (1 - p - q)E_k. \tag{23}$$

Substituting $k = 1$, we obtain

$$E_1 = 1 + qE_0 + pE_2 + (1 - p - q)E_1. \tag{24}$$

As a starting condition, $E_0 = 0$, since in this case the receiver is already at $k = 0$. Additionally from (22), $E_2 = 2E_1$. Substituting in these values, we obtain

$$E_1 = \frac{1}{q - p}. \tag{25}$$

Therefore, using (22) and the values in Table I, we obtain the result

$$E_k = \frac{k}{\mu - \lambda}. \tag{26}$$

3) Delivery Performance: We are now in a position to determine $P_D^W(r)$ and $P_D^A(r)$, the expected time to zero state under add and wait decisions. For a receiver r starting in state k_r, adding will result in one of two possible Markov states, $k_r + 1$ and k_r, depending on whether the receiver experiences an erasure or not. Therefore, the expected time to zero state under adding is given by

$$P_D^A(r) = \bar{\mu}E_{k_r+1} + \mu E_{k_r}$$
$$= \frac{k_r + \bar{\mu}}{\mu - \lambda}. \tag{27}$$

Similarly, waiting will result in receiver r moving to state k_r or $k_r - 1$. Therefore, the expected time to zero state under waiting is given by

$$P_D^W(r) = \bar{\mu}E_{k_r} + \mu E_{k_r-1}$$
$$= \frac{k_r - \mu}{\mu - \lambda}. \tag{28}$$

4) Benefits of Adding and Waiting: For a given receiver r, we now calculate the benefit $B_W(r)$ and $B_A(r)$ of waiting and adding, respectively.

Lower times to zero state are desirable, therefore we multiply the delivery performance of each receiver by a factor of -1. We also observe that when a receiver moves to the zero state, they will deliver *all* the packets in their buffer. Therefore, we also scale the receiver's delivery performance by the number of undelivered packets $u(r)$ currently stored in their buffer. The information provided by $u(r)$ is particularly important because it tells us how many packets' delivery delays will be affected, and therefore how great an impact the sender's decision will have on the delay performance.

On the other hand, throughput and delay may not be of equal importance. Therefore, we scale the throughput performance by a weighting factor f, which determines the relative importance of one unit of throughput, compared to one unit of delay. This single free parameter f is important because it allows us to study the throughput delay sensitivity of the system. This idea is used in other work such as [11], where the parameter p effectively dictates the balance between the throughput and delivery delay.

Combining our results, we obtain the benefits of adding and waiting

$$B_A(r) = fP_T^A(r) - u(r)P_D^A(r)$$
$$= f - u(r)\frac{k_r + \bar{\mu}}{\mu - \lambda} \tag{29}$$
$$B_W(r) = fP_T^W(r) - u(r)P_D^W(r)$$
$$= -u(r)\frac{k_r - \mu}{\mu - \lambda}. \tag{30}$$

5) Decision Metric: We now combine these results to determine whether the sender should add or wait. Of course, in practice we are not able to determine the addition rate λ in advance, so we substitute λ with the estimate λ_{est} from (19). For a

single receiver r, the difference in performance between adding and waiting is given by

$$M(r) = B_A - B_W \tag{31}$$

$$= f - \frac{u(r)}{\mu - \lambda_{\text{est}}}. \tag{32}$$

If $M(r) > 0$, it indicates that adding is more beneficial to receiver r's performance than waiting. It is interesting to note that $M(r)$ does not depend on the actual Markov state k_r of the receiver. This comes about because in (22), the expected time to zero E_k is linearly dependent on k. Adding a new packet merely increments the Markov state by 1, compared to waiting. Therefore, the difference in performance between adding and waiting does not depend on the receiver's Markov state k.

Our decision metric is therefore given by the sum of the receivers' performance differences

$$M = \sum_{r=1}^{R} M(r)$$

$$= Rf - \frac{\sum_{r=1}^{R} u(r)}{\mu - \lambda_{\text{est}}}. \tag{33}$$

If $M > 0$, then the sender decides to add a new packet to the transmission queue. Otherwise, it waits. It is important to note that the $\lambda < \mu$ constraint set by (19) ensures that the weighting of $u(r)$ will always be negative.

C. Feedback Variables

There are two feedback variables that affect the decision metric, M: the total number of undelivered packets stored at the receivers, $\sum_{r=1}^{R} u(r)$, and the delivery rate estimate λ_{est}. We briefly discuss the impact of each variable on the sender's decision.

1) Number of Undelivered Packets $\sum_{r=1}^{R} u(r)$: Let us assume for now that λ_{est} is constant. When this is the case, $u(r)$ becomes the only variable required to determine M. From (33), we can observe that there is a threshold T_U number of undelivered packets stored among the receivers above which, $M < 0$ and the sender will wait, but below which the sender will add. The threshold value can be found by solving the equation

$$Rf - \frac{T_U}{\mu - \lambda_{\text{est}}} = 0 \tag{34}$$

yielding the solution

$$T_U = Rf(\mu - \lambda_{\text{est}}). \tag{35}$$

Therefore, the sender's decision strategy can be equivalently phrased as follows.

If the total number of undelivered packets $\sum_{r=1}^{R} u(r)$ stored at the receivers is greater than or equal to T_U, then wait. Otherwise, add a new packet to the transmission queue.

The greater the value of T_U, the longer the sender will spend adding new packets to the transmission queue, before waiting to reduce the number of undelivered packets at the receivers. This
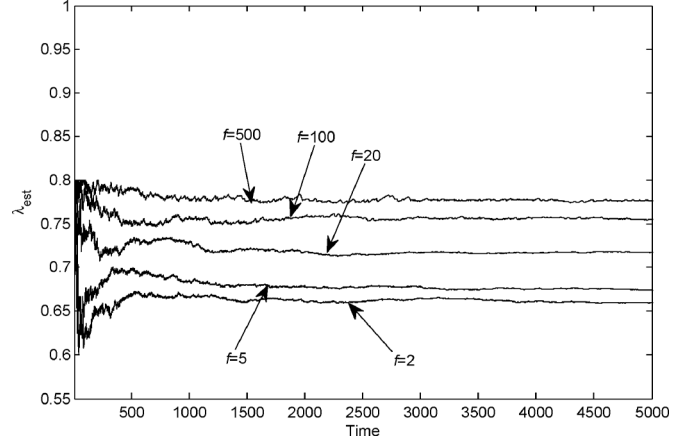

Fig. 11. Sender's estimated addition rate λ_{est} as a function of time, for the weighting factors f listed in the figure. The simulation was implemented with $R = 4$ and $\mu = 0.8$.

means that greater values of T_U will generally result in higher throughput, but also higher delivery delays.

2) Addition Rate Estimate, λ_{est}: Here, we investigate the consequences of an inaccurate addition rate estimate, λ_{est}. Let us assume that there is some value λ, which is the correct addition rate for the system, and let $\lambda_{\text{est}} = \lambda + \Delta$ be the (possibly inaccurate) estimate based on the observed addition rate so far. The effect of the discrepancy Δ on the undelivered packet threshold of (35) is

$$T'_U = Rf(\mu - \lambda - \Delta).$$

Notice that this only differs from (35) by a Δ term. The result is that the greater λ_{est} is compared to λ, the smaller T_U will be, and vice versa.

Interestingly, this results in a feedback loop where if λ_{est} is too high, the addition rate will drop below λ, in turn reducing the measurement λ_{est}. On the other hand, a low value of λ_{est} will have the opposite effect, raising the addition rate, and therefore the estimate λ_{est}. The end result is a stable addition rate, where fluctuations will to some degree be corrected by the system. It can be observed in Fig. 11 that, in line with our analysis, λ_{est} quickly converges upon a stable value.

VII. PERFORMANCE COMPARISON

Here, we compare the performance of the three coding scheme B transmission schemes. Coding scheme B was chosen since it has the best delay performance of the three coding schemes studied in Section V. In Fig. 12, we compare the throughput–delay performance of the coding scheme B transmission schemes for four and eight receivers. The results are discussed here.

A. Baseline Rate Control Scheme

As expected, the baseline rate control scheme exhibits the worst throughput–delay performance of the three rate control schemes. As the throughput increases, so does the average delivery delay, with the four-receiver case performing marginally better than the eight-receiver case.
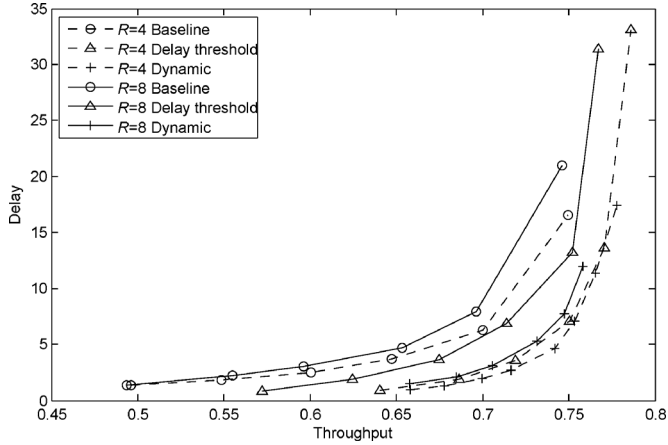
Fig. 12. Throughput–delay performance for the coding scheme B transmission schemes. From left to right, the parameters for each transmission scheme are as follows: Baseline: $\lambda = 0.5, \dots, 0.75$. Delay threshold: $T_D = 2, \dots, 100$. Dynamic: $f = 2, \dots, 500$. For all transmission schemes $R = 4$ and $\mu = 0.8$.

B. Delay Threshold Rate Control Scheme

The delay threshold rate control scheme performs significantly better than the baseline rate control scheme. The strategy of reducing the rate when one or more receivers is experiencing significant delays greatly improves the delivery delay. However, the delay performance for the eight-receiver case is significantly worse than the four-receiver case. This is most likely caused by the transmission inefficiency of the stop mode.

In stop mode, the sender transmits an uncoded packet to allow the worst performing receiver to deliver its next needed packet(s), thus improving its delivery delay. However, in doing so, every other receiver will incur a one-unit throughput penalty since the uncoded transmission will not provide them with any innovative information. With larger numbers of receivers, this penalty can become quite significant.

C. Dynamic Rate Control Scheme

The dynamic rate control scheme further improves upon the delay threshold scheme. We can observe in Fig. 12 that, as intended, increasing the weighting factor f results in higher throughput. The real improvement over the delay threshold rate control scheme can be seen in the eight-receiver case, where for the same throughput, the dynamic rate control scheme experiences approximately half the delivery delay of the delay threshold scheme. This can be attributed to the fact it is a fairer, more well informed rate control scheme.

Unlike the delay threshold scheme, the dynamic rate control scheme does not disproportionately weight the needs of the worst performing receivers. When determining whether to add or wait, the sender weighs the requirements of *all* receivers, instead of only the receiver(s) with the worst delay performance.

Furthermore, the dynamic rate control scheme considers both throughput and delay performance when determining whether to add or wait. This is in stark contrast to the baseline scheme, which only attempts to control the throughput, and the delay threshold scheme, for which the addition rate is controlled purely on the basis of the delay performance of the worst receiver. By recognizing that the add/wait decision is a tradeoff

between throughput and delay performance, our dynamic rate control scheme is able to determine at each time-slot which performance measure can most be improved upon under the current circumstances.

VIII. CONCLUSION

We have demonstrated that the transmission rate of a broadcast transmission scheme can be dynamically adapted to improve both throughput and delivery delay performance.

Analyzing the baseline transmission schemes, we used receivers' Markov states to distinguish among three methods for packet delivery: zero state, leader state, and coefficient-based delivery. We were able to accurately model the zero state delivery delay and found that, in many cases, zero state delivery alone provided a reasonable approximation for the expected delivery delay. Where there were more than a few receivers, leader state delivery was observed to have a negligible impact on the delivery delay. Although the RLNC scheme and coding scheme A had only a small impact on the delivery delay, coding scheme B resulted in significant improvements over zero state delivery alone, by capitalizing on more coefficient-based delivery opportunities.

Based on these observations, we developed a dynamic rate adaptation scheme that determined whether the sender should add or wait by comparing the benefit of each decision to the throughput and delay performance. We found that this decision-making process was equivalent to regulating the sender's addition rate based on the total number of undelivered packets stored at the receivers. The dynamic rate adaptation scheme allowed noticeably better throughput–delay tradeoffs to be achieved, compared to existing approaches in the literature.

So far, our work has only been in the context of receivers with homogeneous channel rates. While our analysis is equally applicable to heterogeneous networks, a number of other issues including resource allocation and fairness must also be considered.

REFERENCES

[1] C. Fragouli, J. Widmer, and J. Le Boudec, "Efficient broadcasting using network coding," *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 450–463, Apr. 2008.
[2] P. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 390–404, Apr. 2006.
[3] H. Seferoglu and A. Markopoulou, "Video-aware opportunistic network coding over wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 5, pp. 713–728, Jun. 2009.
[4] J. Chakareski and P. Frossard, "Rate-distortion optimized distributed packet scheduling of multiple video streams over shared communication resources," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 207–218, Apr. 2006.
[5] C. Fragouli, D. Katabi, A. Markopoulou, M. Médard, and H. Rahul, "Wireless network coding: Opportunities & challenges," in *Proc. IEEE Military Commun. Conf.*, Oct. 2007, pp. 1–8.
[6] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2002.

[7] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[8] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless broadcast using network coding," *IEEE Trans. Veh. Technol.*, vol. 58, no. 2, pp. 914–925, Feb. 2009.

[9] X. Li, C.-C. Wang, and X. Lin, "Throughput and delay analysis on uncoded and coded wireless broadcast with hard deadline constraints," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–5.

[10] J. Barros, R. Costa, D. Munaretto, and J. Widmer, "Effective delay control in online network coding," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 208–216.

[11] W. Zeng, C. Ng, and M. Médard, "Joint coding and scheduling optimization in wireless systems with varying delay sensitivities," in *Proc. IEEE SECON*, 2012, pp. 416–424.

[12] A. Eryilmaz, A. Ozdaglar, M. Médard, and E. Ahmed, "On the delay and throughput gains of coding in unreliable networks," *IEEE Trans. Inf. Theory*, vol. 54, no. 12, pp. 5511–5524, Dec. 2008.

[13] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. Annu. Allerton Conf. Commun., Control, Comput.*, Oct. 2003.

[14] B. Swapna, A. Eryilmaz, and N. Shroff, "Throughput-delay analysis of random linear network coding for wireless broadcasting," in *Proc. NETCOD*, June 2010, pp. 1–6.

[15] M. Vehkapera and M. Médard, "A throughput-delay trade-off in packetized systems with erasures," in *Proc. IEEE ISIT*, 2005, pp. 1858–1862.

[16] J. Park, M. Gerla, D. Lun, Y. Yi, and M. Médard, "Codecast: A network-coding-based ad hoc multicast protocol," *IEEE Wireless Commun. Mag.*, vol. 13, no. 5, pp. 76–81, Oct. 2006.

[17] D. Lucani, M. Stojanovic, and M. Médard, "Random linear network coding for time division duplexing: When to stop talking and start listening," in *Proc. IEEE INFOCOM*, 2009, vol. 9, pp. 1800–1808.

[18] D. Lun, M. Médard, and R. Koetter, "Network coding for efficient wireless unicast," in *Proc. IZS*, 2006, pp. 74–77.

[19] C. Gkantsidis and P. Rodriguez, "Network coding for large scale content distribution," in *Proc. IEEE INFOCOM*, 2005, vol. 4, pp. 2235–2245.

[20] P. Sadeghi, R. Shams, and D. Traskov, "An optimal adaptive network coding scheme for minimizing decoding delay in broadcast erasure channels," *EURASIP J. Wireless Commun. Netw.*, pp. 1–14, Apr. 2010.

[21] S. Sorour and S. Valaee, "On minimizing broadcast completion delay for instantly decodable network coding," in *Proc. IEEE ICC*, May 2010, pp. 1–5.

[22] P. Sadeghi, D. Traskov, and R. Koetter, "Adaptive network coding for broadcast channels," in *Proc. NETCOD*, Jun. 2009, pp. 80–85.

[23] X. Li, C.-C. Wang, and X. Lin, "On the capacity of immediately-decodable coding schemes for wireless stored-video broadcast with hard deadline constraints," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 5, pp. 1094–1105, May 2011.

[24] S. Sorour and S. Valaee, "Minimum broadcast decoding delay for generalized instantly decodable network coding," in *Proc. IEEE GLOBECOM*, 2010, pp. 1–5.

[25] L. Yang, Y. Sagduyu, J. Li, and J. Zhang, "Adaptive network coding for scheduling real-time traffic with hard deadlines," 2012 [Online]. Available: http://arxiv.org/abs/1203.4008

[26] C. Fragouli, D. Lun, M. Médard, and P. Pakzad, "On feedback for network coding," in *Proc. CISS*, 2007, pp. 248–252.

[27] J. Sundararajan, P. Sadeghi, and M. Médard, "A feedback-based adaptive broadcast coding scheme for reducing in-order delivery delay," in *Proc. NETCOD*, 2009, pp. 1–6.

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

[29] J. Sundararajan, D. Shah, and M. Médard, "Feedback-based online network coding," 2009 [Online]. Available: http://arxiv.org/abs/0904.1730

[30] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[31] R. Brualdi and K. Bogart, *Introductory Combinatorics*. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.

Amy Fu received the Bachelor of Engineering degree in electrical engineering from the University of Sydney, Sydney, Australia, in 2009, and is currently pursuing the Ph.D. degree in engineering at the Australian National University, Canberra, Australia.

Her research interests are in the area of network coding for wireless broadcast networks.

Miss Fu was awarded First Class Honours and the University Medal from the University of Sydney in 2009.

Parastoo Sadeghi (S'02–M'06–SM'07) received the B.E. and M.E. degrees in electrical engineering from Sharif University of Technology, Tehran, Iran, in 1995 and 1997, respectively, and the Ph.D. degree in electrical engineering from The University of New South Wales, Sydney, Australia, in 2006.

From 1997 to 2002, she worked as a Research Engineer and then as a Senior Research Engineer with Iran Communication Industries (ICI), Tehran, Iran, and with Deqx (formerly known as Clarity Eq), Sydney, Australia. She is currently a Fellow with the Research School of Engineering, The Australian National University, Canberra, Australia. She has visited various research institutes, including the Institute for Communications Engineering, Technical University of Munich, Munich, Germany, in 2008 and the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2009 and 2013. She has coauthored more than 100 refereed journal or conference papers and the book *Hilbert Space Methods in Signal Processing* (Cambridge Univ. Press, 2013). She is a Chief Investigator in a number of Australian Research Council Discovery and Linkage Projects. Her research interests are mainly in the areas of network coding, wireless communications systems, and signal processing.

Dr. Sadeghi received two IEEE Region 10 Student Paper awards for her research in the information theory of time-varying fading channels in 2003 and 2005.

Muriel Médard (S'90–M'95–SM'02–F'08) received the B.S. degrees in electrical engineering and computer science (EECS) and mathematics, the B.S. degree in humanities, and the M.S. and Sc.D. degrees in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 1989, 1990, 1991, and 1995, respectively.

She is a Professor with the Department of EECS, MIT. She was previously an Assistant Professor with the Electrical and Computer Engineering Department, University of Illinois at Urbana–Champaign, Urbana, IL, USA, and a Staff Member with the MIT Lincoln Laboratory, Lexington, MA, USA. Her research interests are in the areas of network coding and reliable communications, particularly for optical and wireless networks.

Prof. Médard has served as an Associate Editor for the Optical Communications and Networking Series of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, the IEEE/OSA JOURNAL OF LIGHTWAVE TECHNOLOGY, the IEEE TRANSACTIONS ON INFORMATION THEORY, and the *OSA Journal of Optical Networking*. She has served as a Guest Editor for the IEEE/OSA JOURNAL OF LIGHTWAVE TECHNOLOGY, IEEE Transactions on Information Theory (twice), the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (twice), and IEEE IFS. She is a member of the Board of Governors of the IEEE Information Theory Society and serves as the Junior Past President. She has served as TPC Co-Chair of ISIT, WiOpt, and CONEXT. She was awarded the 2009 IEEE Communication Society and Information Theory Society Joint Paper Award, the 2009 IEEE William R. Bennett Prize in the Field of Communications, and the 2002 IEEE Leon K. Kirchmayer Prize Paper Award. She was co-winner of the 2004 MIT Harold E. Edgerton Faculty Achievement Award. In 2007, she was named a Gilbreth Lecturer by the National Academy of Engineering.