A Feedback-based Adaptive Broadcast Coding Scheme for Reducing In-order Delivery Delay

A Feedback-based Adaptive Broadcast Coding Scheme for Reducing In-order Delivery Delay

Jay Kumar Sundararajan
Massachusetts Institute of Technology
Cambridge, MA, USA
Email: jaykumar@mit.edu

Parastoo Sadeghi
Australian National University (ANU)
Canberra, ACT, Australia
Email: parastoo.sadeghi@anu.edu.au

Muriel Médard
Massachusetts Institute of Technology
Cambridge, MA, USA
Email: medard@mit.edu

Abstract — **We propose a new feedback-based adaptive coding scheme for a packet erasure broadcast channel. The main performance metric of interest is the delay. We consider two types of delay – decoding delay and delivery delay. Decoding delay is the time difference between the instant when the packet is decoded at an arbitrary receiver and the instant when it arrived at the sender. Delivery delay also includes the period when a decoded packet waits in a resequencing buffer at the receiver until all previous packets have also been decoded. This notion of delay is motivated by applications that accept packets only in order. Our coding scheme has the innovation guarantee property and is hence throughput optimal. It also allows efficient queue management. It uses the simple strategy of mixing only the oldest undecoded packet of each receiver, and therefore extends to any number of receivers. We conjecture that this scheme achieves the asymptotically optimal delivery (and hence decoding) delay. The asymptotic behavior is studied in the limit as the load factor of the system approaches capacity. This conjecture is verified through simulations.**

I. Introduction

In today's communication systems, the demand for supporting real-time applications is growing rapidly. In popular applications such as live video streaming and video conferencing, the user's experience is very sensitive to the per-packet delay. In pre-recorded video streaming (*i.e.*, not live), a low delay is still preferable because that would reduce the amount of buffering required for playback at the receiver.

Note that this notion of per-packet delay is very different from download delay [1]. While downloading a file, usually the main performance criterion is the time it takes to complete the download. From the system point of view, this goal essentially translates to a high throughput requirement. The implicit assumption in such a scenario is that the file is useful only as a whole.

From a throughput perspective, there are situations where coding across packets is very useful. One reason is that coding can help correct errors and erasures in the network. Another reason is, in certain network topologies such as the butterfly network from the network coding literature [2], coding is necessary to share bottleneck links across flows, in order to achieve the system capacity. Similarly, in broadcast-mode links, especially with erasures, coding across packets is critical for achieving a high throughput [3].

Now, any form of coding comes with an associated decoding delay. The receiver has to wait to collect sufficiently many coded packets before it can decode the original packets. Therefore, in delay-sensitive applications, it may be necessary to carefully design the coding scheme so that it not only satisfies the criteria needed to ensure high throughput, but also achieves a low decoding delay.

Motivated by this goal, we explore in our work, the possibility of making use of feedback in order to adapt the coding scheme in an online manner. We focus on the single hop packet erasure broadcast channel with perfect immediate feedback. We propose and study a new coding module for any number of receivers. We show that it is throughput optimal and that it allows efficient queue management. We also study two different notions of delay. The first one is the decoding delay per packet. This is simply the average over all packets of the time between arrival and decoding at an arbitrary receiver. The second notion, known as delivery delay, is a much stronger notion of delay. It assumes that packets may be delivered to the receiver's application only in the order of their arrival at the sender. These notions were also introduced in earlier work [4], [5]. We conjecture that our scheme achieves the asymptotically optimal expected decoding delay and delivery delay.

The rest of the paper is organized as follows. In Section II, we present the system model and the problem statement. Section III then motivates the problem in the context of related earlier work. Section IV presents the new generalized coding module for any number of receivers. The performance of this algorithm is described in Section V. In Section VI, we present our simulation results. Finally, the conclusions and directions for future work are presented in Section VII.

II. The system model

The system model is identical to that in [6] and [7]. It is presented here for completeness.

Time is slotted. Packets of a fixed size arrive into an infinite capacity buffer at a sender node according to a Bernoulli process of rate λ. There are n receivers. This stream belongs to a broadcast session that needs to be delivered to all the receivers. The sender is connected to the receivers by a packet erasure broadcast channel. This channel accepts one packet as input in every slot and

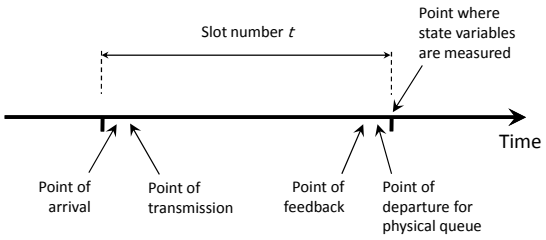

Figure 1: Relative timing of arrival, service and departure points within a slot

either reproduces this packet without errors (with probability μ), or outputs an erasure symbol (with probability $1-\mu$), independently at each receiver. The erasures are independent across receivers and across time-slots, and can be detected by the receivers. The presence of a perfect delay-free feedback link enables the sender to know the exact state of knowledge of the receivers in every slot, *before* deciding the transmission of that slot. The exact order in which the various events occur in our system is shown in Figure 1.

The load factor ρ is defined to be $\rho := \lambda/\mu$. In this paper, we assume that the sender can only use linear codes. In other words, every transmission is a linear combination of the current contents of the buffer. The coefficient vector corresponding to a linear combination is conveyed to the receiver through the packet header.

We will use two notions of delay. These notions were also studied in [5].

Definition 1 (Decoding Delay)**.** *The* decoding delay *of a packet with respect to a receiver is the time between the arrival of the packet at the sender and the decoding of the packet by the receiver under consideration.*

Definition 2 (Delivery Delay)**.** *The* delivery delay *of a packet with respect to a receiver is the time between the arrival of the packet at the sender and the delivery of the packet by the receiver to the application, with the constraint that packets may be delivered only in order.*

It is easily seen from these definitions that *the delivery delay is in general, longer than the decoding delay.* It is well known that in this model, the queue can be stabilized as long as $\rho < 1$, by using linear network coding [3]. In this work, we are interested in the rate of growth of the decoding and delivery delay, in the asymptotic regime of ρ approaching 1.

The problem we study in this paper is the following: Is there an adaptive coding scheme that is throughput optimal and at the same time achieves the best possible rate of growth of the decoding and delivery delay, as a function of $1/(1-\rho)$?

III. Motivation and related earlier work

Coding for per-packet delay has been studied in earlier work by Martinian *et al.* [8]. However, that work considered a point-to-point setting unlike our broadcast scenario. The problem of the delay for recovering packets from a file has been studied in the rateless code framework with or without feedback, by [9] and [10]. Reference [11] also considered the problem of coding for delay using feedback. The setting there is in terms of a fixed delay model for point-to-point communication, where each packet has a deadline by which it needs to be delivered. A packet which does not meet its deadline is considered to be in error, and the corresponding error exponents are characterized.

In contrast, we consider the expected per-packet delay in a queuing theoretic framework, with no strict deadlines. Besides, our setting is a point-to-multipoint (broadcast) packet erasure channel.

For the case of packet erasure broadcast channel with two receivers, Durvy *et al.* [12] have proposed a feedback-based throughput optimal coding scheme that ensures that every successful innovative reception at any receiver will cause it to decode an original packet. This property is called *instantaneous decodability.* However, the authors provided an example to show that for the three receiver case, instantaneous decodability cannot be achieved without losing throughput.

Keller *et al.* [13] also studied this problem and proposed and compared several algorithms to reduce the decoding delay. This work did not consider the in-order delivery problem. Both [12] and [13] consider the transmission of a given finite set of packets. In contrast, [14] assumes that packets arrive at the source according to a stochastic process in a streaming manner and proposes a coding scheme for two receivers. The focus however, is to ensure stable throughput and not low delay. In [15], the authors propose a greedy coding scheme for the case of more than 2 receivers, which aims to maximize the number of receivers that can decode a packet instantaneously, at the expense of losing throughput.

Our current work considers stochastic packet arrivals. Whereas the earlier works did not consider the in-order delivery constraint, we study the delivery delay as well. We focus on throughput optimal schemes. Since instantaneously decodability cannot be guaranteed for more than 2 receivers, we consider the relaxed requirement of asymptotically optimal decoding and delivery delay, where the asymptotics are in the limit of the load factor $\rho \to 1$.

Reference [7] presented a lower bound on the asymptotic growth of the expected decoding delay of $O\left(\frac{1}{1-\rho}\right)$ by arguing that even the single receiver case has this linear rate of growth in terms of $\frac{1}{1-\rho}$. For the two receiver case, it can be proved that the algorithm of [12] indeed achieves this lower bound for decoding delay, and seems to achieve it for delivery delay as well, based on simulations.

In other work, [6] (also in [5]) proposed a feedback-based coding scheme for any number of receivers. The main focus there, however, was to ensure efficient queue management. The queue size growth was shown to be $O\left(\frac{1}{1-\rho}\right)$. However, the decoding delay of the schemes

proposed there, are seen to have a *quadratic* growth in $1/(1-\rho)$ based on simulations.

Reference [7] proposed a coding scheme for the case of three receivers that was conjectured to achieve the asymptotic lower bound. However, it was not generalizable to multiple receivers. Reference [16] considers the case of heterogeneous channels to the receivers, and proposes a systematic online encoding scheme that sends uncoded packets to enable frequent decoding at the receivers. However, no characterization of the asymptotic behavior of the decoding or delivery delay is provided. For more related work, the reader is also referred to [16] and [5].

The contribution of our current paper is to provide a new coding module for any number of receivers, that is at the same time, throughput optimal, allows asymptotically optimal queue sizes and is conjectured to achieve an asymptotically optimal $O\left(\frac{1}{1-\rho}\right)$ growth for both decoding and delivery delay. It can be shown that the two-receiver algorithm of [12] is a special case of our algorithm. The delay performance conjecture is verified through simulations.

IV. The coding algorithm

We now present the new coding module for the general case of any number of receivers. First, we describe the main ideas behind the algorithm. Then, we present the detailed specification.

A. Intuitive description

The intuition behind the algorithm is to first identify for each receiver, the oldest packet that it has not yet decoded, which we will call the *request* of that receiver. The algorithm then transmits a linear combination that involves packets from only within this set.

The linear combination is constructed incrementally. The receivers are grouped according to their request, and the groups are processed in descending order of their requested packet's index. First, the newest request (*i.e.*, the one with the largest index) is included in the linear combination, as otherwise, the corresponding receivers, having decoded everything older, will find the transmission non-innovative[1]. Then, the algorithm checks whether the linear combination formed thus far is innovative to every receiver in the next group. If it is not innovative, then the coefficient of the next group's request is adjusted till it is simultaneously innovative to the whole group. The key idea is that, since the groups are processed in descending order of their requests, the choices made for the coefficient of subsequent groups' requests will not affect the innovation of earlier groups. This is because, the earlier groups have already decoded the subsequent groups' requests.

After processing all the groups in this order, the transmitted linear combination is thus chosen so that it satisfies the innovation guarantee property[2].

B. Representing knowledge

Before specifying the algorithm, we first propose a way to systematically represent the state of knowledge of the receivers. This is based on the representation used in [6], with a key difference described below.

The k^{th} packet to have arrived at the sender is said to have an *index* k and is denoted by $\mathbf{p_k}$. Suppose the total number of packets that have arrived at any time t is denoted by $A(t)$. Since we have a restriction that the coding must be linear, we can represent the state of knowledge of a node by a vector space consisting of all the linear combinations that a node can compute using what it has received thus far. We represent the state of knowledge using a basis of this vector space. The basis is represented as the rows of a matrix which is in the row-reduced echelon form (RREF). The matrix has $A(t)$ columns, one for each packet that has arrived thus far. While all this is identical to the representation in [6], the main difference is in the ordering of the columns of the basis matrix. We use the same framework, except that in our current work, the columns are ordered so that packet $\mathbf{p_k}$ maps to column $A(t)-k$. In other words, the columns are arranged in reverse order with respect to the order of arrival at the sender.

Throughout this paper, we will use the RREF representation of the basis matrix, with this reverse ordering of the packets. We also make use of the notion of seen packets that was introduced in [6]. Note however that the definition becomes quite different from that in the previous work, if we use the reverse ordering on the packets.

Definition 3 (Seeing a packet)**.** *A node is said to have* seen *a packet with index k if and only if the k^{th} column from the right, of the RREF basis B of the knowledge space V of the node, is a pivot column. Alternatively, a node has seen a packet $\mathbf{p_k}$ if it has received enough information to compute a linear combination of the form $(\mathbf{p_k}+\mathbf{q})$, where $\mathbf{q}$ is itself a linear combination involving only packets with an index **less** than that of $\mathbf{p}$. (Decoding implies seeing, as we can pick $\mathbf{q}=\mathbf{0}$.)*

In contrast, the definition used in [6] had replaced the word "less" with the word "greater" in the above statement. We believe the reverse ordering is better suited to analyzing the delivery delay. We now make some observations about the new definition.

Observation 1: As with the forward ordering, the notion of seen with the reverse ordering also has connections to the dimension of the knowledge space. In particular, we can show that every innovative reception causes a new

[1] A linear combination is said to be *innovative* if its coefficient vector is linearly independent of previously received linear combinations.

[2] *Innovation guarantee* means the transmission is innovative to every receiver except when the receiver already knows everything that the sender knows.

packet to be seen. In other words, the number of seen packets is equal to the dimension of the knowledge space.

Observation 2: Due to the reverse ordering of the packets, we have an interesting property. For any $k > 0$, if all packets $\mathbf{p_1}$ to $\mathbf{p_k}$ have been seen, then they have also been decoded, and hence can be delivered to the application.

C. Algorithm specification

Now, we present the formal coding algorithm. Let us first define $\{u_1, u_2, \ldots, u_m\}$ to be the set of indices of the oldest undecoded packets of the n receivers, sorted in descending order ($m \leq n$, since the oldest undecoded packet may be the same for some receivers). Exclude receivers whose oldest undecoded packet has not yet arrived at the sender. We call this resulting set of packets the *transmit set*, since the coding module will use only these packets in computing the linear combination to be transmitted.

Let $R(u_i)$ be the group of receivers whose request is $\mathbf{p_{u_i}}$. We now present the coding module to select the linear combination for transmission.

Initialize the transmit coefficient vector $\mathbf{a}$ to an all zero vector of length Q, the current sender queue size.
for $j = 1$ to m **do** (*Loop over the transmit set*)
  Initialize the veto list[3] to the empty set.
  for all $r \in R(u_j)$ **do**
    Zero out the coefficient of all packets seen by receiver r from the current transmission vector $\mathbf{a}$ by subtracting from $\mathbf{a}$, suitably scaled versions of the rows of the current RREF basis matrix, to get the vector $\mathbf{a'}$. (This is essentially the first step of Gaussian elimination.) Hence, find out which packet will be newly seen if the linear combination corresponding to $\mathbf{a}$ is transmitted. This is simply the index of the packet corresponding to the first non-zero entry in $\mathbf{a'}$.
    if no packet is newly seen **then**
      Append 0 to the veto list
    else if the newly seen packet's index is u_j **then**
      Append the additive inverse of the leading non-zero entry of $\mathbf{a'}$ to the veto list
    else if the newly seen packet is anything else **then**
      Do not add anything to the veto list
    end if
  end for
  Arrange the elements of the finite field in any order, starting with 0. Choose a_{u_j} to be the first element in this order that is not in the veto list.
end for
Compute the transmit packet: $\mathbf{g} := \sum_{k=1}^{Q} a_k \mathbf{p_k}$

V. Properties of the algorithm

A. Throughput

To ensure correctness, the algorithm uses a finite field of size at least as large as the number of receivers. Theorem 1 shows that this is a sufficient condition to guarantee innovation.

Theorem 1. *If the field is at least as large as the number of receivers, then the above algorithm will always find values for the a_k's such that the resulting transmission satisfies the innovation guarantee property.*

Proof. We first show that the choices made by the algorithm guarantee innovation. For any $j > 0$, consider the j^{th} request group. Let $\mathbf{a}(j-1)$ be the value of the coefficient vector just before processing group j (Note, $\mathbf{a}(\mathbf{0}) = \mathbf{0}$.).

Any receiver in group j has not decoded $\mathbf{p_{u_j}}$ yet. Hence, it cannot know a linear combination of the form $\mathbf{a}(j-1) + \beta \mathbf{e_{u_j}}$ for more than one value of β, where $\mathbf{e_{u_j}}$ is the unit vector with a 1 in the u_j^{th} coordinate and 0 elsewhere. (If it knew two such combinations, it could subtract one from the other to find $\mathbf{p_{u_j}}$, a contradiction.)

Suppose the receiver knows exactly one such linear combination. Then, after the row reduction step, the vector $\mathbf{a}(j-1)$ will get transformed into $\mathbf{a'} = -\beta \mathbf{e_{u_j}}$. Hence, the leading non-zero coefficient of $\mathbf{a'}$ is $-\beta$, and its additive inverse gives β. (Note: the resulting value of β could be 0. This corresponds to the non-innovative case.) If the receiver does not know any linear combination of this form, then packet u_j is not seen, and nothing is added to the veto list.

In short, the values that are vetoed are those values of β for which some receiver knows a linear combination of the form $\mathbf{a}(j-1) + \beta \mathbf{e_{u_j}}$. Hence, by picking a value of a_{u_j} from outside this list, we ensure innovation. Thus, the algorithm essentially checks for innovation by considering different coefficients β for including $\mathbf{p_{u_j}}$ into the transmission and eliminating the ones that do not work. Finally, processing subsequent groups will not affect the innovation of the previous groups because the subsequent groups will only change the coefficient of their requests, which have already been decoded by the previous groups.

We now show that the algorithm always has enough choices to pick such an a_{u_j} even after excluding the veto list. As argued above, at any point in the algorithm, each receiver adds at most one field element to the veto list. Hence, the veto list can never be longer than the number of receivers in the corresponding request group. Now, we consider two cases.

Case 1: If the group requesting the highest request u_1 does not include all the receivers, then none of the groups contain n receivers. Hence, the veto list for any group will always be strictly shorter than n, and hence if the field size is at least n, there is always a choice left for a_{u_j}.

Case 2: If all n receivers request the highest packet u_1, then it has to be the case that they have all decoded every packet before u_1. Hence, the only coefficient that any receiver would veto for $\mathbf{p_{u_1}}$ is 0, thus leaving other choices for a_{u_1}.

This completes the proof. □

[3]This will hold the list of unacceptable coefficients of $\mathbf{p_{u_j}}$.

B. Decoding and delivery delay

We conjecture that the coding module described above has good delay performance.

Conjecture 1. *For the coding module in Section IV C, the expected decoding delay per packet, as well as the expected delivery delay per packet with respect to a particular receiver, grow as* $O\left(\frac{1}{1-\rho}\right)$ *as* $\rho \to 1$*, which is asymptotically optimal.*

The exact analysis of the delay and the proof of this conjecture are open problems. We believe that the notion of seen packets will be useful in this analysis. In particular, to analyze the delivery delay, we can make use of Observation 2 from Section IV B. A packet is delivered if and only if this packet and all packets with a lower index have been seen. This condition is the same as what arises in problems involving a resequencing buffer. Thus, we can formulate our delivery delay problem in terms of traditional queuing problems.

In our formulation, we break down the delivery delay of a packet for a particular receiver into two parts, as though the packet has to traverse two queues in tandem. The first part is simply the time till the packet is seen. Once it is seen, the packet moves into a second queue which is essentially a resequencing buffer. The second part is the time spent in this buffer waiting for all older packets to be seen.

The expectation of the first part is easy to calculate, since every innovative reception causes a new packet to be seen. By Little's theorem, the delay is directly proportional to the size of the queue of unseen packets. This queue's behavior was studied in [6]. Although that work used the older notion of seeing a packet, it can be shown that the analysis still holds even if we use the new notion of seen packets based on reverse ordering. Hence, we get a $O\left(\frac{1}{1-\rho}\right)$ bound on the first part of the delay. The analysis of the second part of the delay however, seems more complicated.

C. Queue management

The coding module described above makes use of only the oldest undecoded packet of each receiver in any given time-slot. Since our definition of seen packets uses reverse ordering of the packets (see Section IV B), the oldest undecoded packet is always an unseen packet. In other words, the algorithm never uses packets that have been seen by all the receivers. This implies that the algorithm is compatible with the drop-when-seen queuing algorithm that was proposed and analyzed in [6], provided we use the new definition of seen. As pointed out in Observation 1 in Section IV B, the new definition of seeing a packet has the same relation to the dimension of the knowledge space as the old definition of [6]. Thus, we can obtain all the queue size guarantees that were obtained in the earlier work. In other words, we can get a provable $O\left(\frac{1}{1-\rho}\right)$ growth of the expected queue size at the sender, in addition to the provable innovation guarantee property and the conjectured delay guarantees.

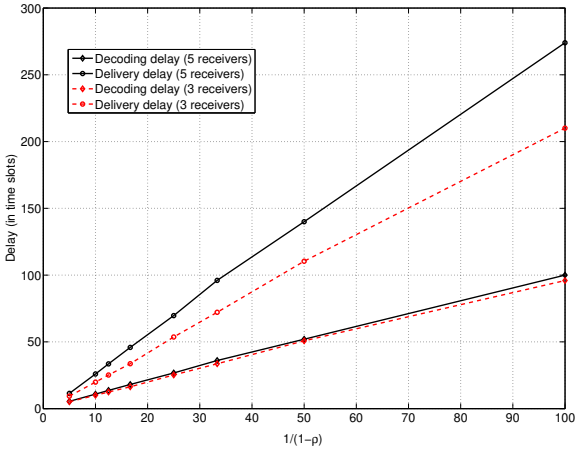

Figure 2: Linear plot of the decoding and delivery delay

VI. Simulation results

We now evaluate the performance of the newly proposed coding module through simulations. In particular, we study the behavior of the decoding delay and the delivery delay as a function of the load factor ρ, in the limit as ρ approaches 1, *i.e.*, as the loading on the system approaches capacity.

The probability of reception in any slot is $\mu = 0.5$. The packets arrive according to a Bernoulli process, whose arrival rate is calculated according to the load factor ρ. The load factor is varied through the following values: 0.8, 0.9, 0.92, 0.94, 0.96, 0.97, 0.98 and 0.99. The decoding delay and delivery delay are averaged across the packets over a large number of slots. The number of slots is set to 10^6 for the first four data points, 2×10^6 for the next two points, and at 5×10^6 for the last two points.

We consider two different cases. In the first case, there are three receivers. The entire operation is therefore performed over a $GF(3)$ (*i.e.*, integer operations modulo 3). In the second case, we consider the situation where there are five receivers. In this case, the operations are performed over a field of size 5.

Figure 2 shows the plot of the decoding and delivery delay as a function of $\frac{1}{1-\rho}$ for both the three and the five receiver cases. Figure 3 shows the same plot in a logarithmic scale. From both these figures, it is clearly seen that the algorithm achieves a linear growth of the delay in terms of $\frac{1}{1-\rho}$. We have thus verified Conjecture 1 for the case of 3 and 5 receivers, using simulations.

VII. Conclusion

In this work, we have thus proposed a new coding module which not only achieves optimal throughput, but is conjectured to achieve asymptotically optimal decoding

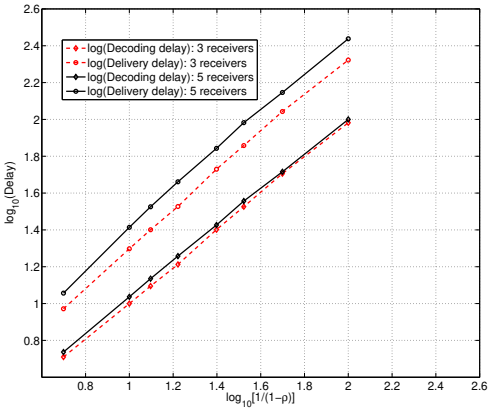

Figure 3: Log plot of the decoding and delivery delay

and in-order delivery delay as well. In addition, it also allows efficient queue management, leading to asymptotically optimal expected queue size. The algorithm applies to the case of any number of receivers. The conjecture on low delay is verified through simulations.

Our work introduces a new way of adapting the encoding process based on feedback so as to ensure low delay in combination with high throughput. In the future, several extensions are possible. Of particular interest is the study of the effect of delayed or imperfect feedback on the code design. We believe that the main ideas of the coding algorithm will extend to the case where we have imperfections in the feedback link.

Also of interest for the future is the proof of Conjecture 1. The delivery delay is closely related to the problems concerning resequencing buffers, which have been studied in a different context in the literature. The techniques used in those works might be useful in studying this problem as well.

Acknowledgments

The authors would like to thank Prof. Devavrat Shah for several useful discussions. This research was supported by the following grants: subcontract # S0176938 issued by UC Santa Cruz, supported by the United States Army under Award No. W911NF-05-1-0246, and the DARPA under Grant No. HR0011-08-1-0008. The work of P. Sadeghi was supported under ARC Discovery Projects funding scheme (project no. DP0984950).

References

[1] A. Eryilmaz, A. Ozdaglar, and M. Médard, "On delay performance gains from network coding," in *40th Annual Conference on Information Sciences and Systems*, 2006.

[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. on Information Theory*, vol. 46, pp. 1204–1216, 2000.

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

[4] C. T. K. Ng, M. Médard, and A. Ozdaglar, "Cross-layer optimization in wireless networks under different packet delay metrics," in *Proceedings of IEEE INFOCOM*, 2009.

[5] J. K. Sundararajan, D. Shah, and M. Médard, "Feedback-based online network coding," *Submitted to IEEE Trans. on Information Theory*, 2009. [Online]. Available: http://arxiv.org/abs/0904.1730

[6] ——, "ARQ for network coding," in *Proceedings of IEEE ISIT*, July 2008.

[7] ——, "Online network coding for optimal throughput and delay - the three-receiver case," in *International Symposium on Information Theory and its Applications*, December 2008.

[8] E. Martinian, "Dynamic information and constraints in source and channel coding," PhD Thesis, Massachusetts Institute of Technology, Dept. of EECS, 2004.

[9] S. Sanghavi, "Intermediate performance of rateless codes," in *Proceedings of IEEE Information Theory Workshop (ITW)*, September 2007.

[10] A. Beimel, S. Dolev, and N. Singer, "RT oblivious erasure correcting," in *Proceedings of IEEE Information Theory Workshop (ITW)*, October 2004.

[11] A. Sahai, "Why delay and block length are not the same thing for channel coding with feedback," in *Proc. of UCSD Workshop on Information Theory and its Applications. Invited Paper*, Feb. 2006.

[12] M. Durvy, C. Fragouli, and P. Thiran, "Towards reliable broadcasting using ACKs," in *Proceedings of IEEE ISIT*, 2007.

[13] L. Keller, E. Drinea, and C. Fragouli, "Online broadcasting with network coding," in *NetCod*, 2008.

[14] Y. E. Sagduyu and A. Ephremides, "On broadcast stability region in random access through network coding," in *44th Allerton Annual Conference on Communication, Control and Computing*, 2006.

[15] ——, "On network coding for stable multicast communication," in *IEEE Military Communications Conference (MILCOM)*, October 2007.

[16] J. Barros, R. A. Costa, D. Munaretto, and J. Widmer, "Effective delay control in online network coding," in *Proceedings of IEEE INFOCOM*, 2009.