Acknowledgement Design for Collision-Recovery-Enabled Wireless Erasure Networks

Ali ParandehGheibi, Jay Kumar Sundararajan, Muriel Médard

Department of Electrical Engineering and Computer Science

Massachusetts Institute of Technology

Email:{parandeh, jaykumar, medard}@mit.edu

Abstract—Current medium access control mechanisms are based on collision avoidance and collided packets are discarded. The recent work on ZigZag decoding departs from this approach by recovering the original packets from multiple collisions. In this paper, we view each collision as a linear combination of the original packets at the senders. The transmitted, colliding packets may themselves be a coded version of the original packets.

We design acknowledgment (ACK) mechanisms based on the idea that if a set of packets collide, the receiver can afford to ACK exactly one of them without being able to decode the packet. We characterize the conditions for an ACK mechanism under which the receiver can eventually decode all of the packets. In the context of a wireless erasure network, we show that the senders' queues behave as if the transmissions are controlled by a centralized scheduler which has access to channel state realizations at the *beginning* of each time slot. Taking advantage of this relation, we propose two ACK policies that stabilizes the system. One of these policies only requires the arrival rate information, while the other one only needs queue-length information.

We also show that our ACK policies combined with a completely decentralized transmission mechanism based on random linear network coding achieves the cut-set bound of the packet erasure network, which is strictly larger than the stability region of the centralized scheduling schemes without collision recovery.

I. INTRODUCTION

The nature of the wireless network is intrinsically different from the wired network because of the sharing of the medium among several transmitters. Such a restriction generally has been managed through forms of scheduling algorithms to coordinate access to the medium, usually in a distributed manner. The conventional approach to the Medium Access Control (MAC) problem is contention-based protocols in which multiple transmitters simultaneously attempt to access the wireless medium and operate under some rules that provide enough opportunities for the others to transmit.

Examples of such protocols in packet radio networks include ALOHA, MACAW, CSMA/CA, etc[9].

However, in many contention-based protocols, it is possible that two or more transmitters transmit their packet simultaneously, resulting in a *collision*. The collided packets are considered useless in the conventional approaches. There is a considerable literature on extracting partial information from such collisions. Gollakota and Katabi [2] showed how to recover multiple collided packets in a 802.11 system using *ZigZag decoding* when there are enough transmissions involving those packets. In fact, they suggest that each collision can be treated as a linear equation of the packets involved. In another related work, Hou *et al.* [3] demonstrated practical interference cancellation schemes for an EV-DO Rev A system with frame asynchronous users. Such interference cancellation methods require a precise estimation of channel attenuation and phase shift for each packet involved in a collision. Collision recovery approaches provide a fundamentally new approach to manage interference in a wireless setting that is essentially decentralized, and can recover losses due to collisions. In this work, we wish to understand the effects of this new approach to interference management in the high SNR regime, where interference, rather than noise, is the main limit factor for system throughput.

We provide an abstraction of a single-hop wireless network with erasures when a generalized form of ZigZag decoding is used at the receiver, and network coding is employed at the transmitters. In [17], we introduced an algebraic representation of the collisions at the receivers, and studied conditions under which a collision can be treated as a linearly independent equation (degree of freedom) of the original packets. We use this abstract model to analyze the performance of the system in various scenarios.

In [17], we analyzed a single-hop wireless erasure network with a collision recovery-enabled receiver, when each sender has one packet to deliver to the receiver. We characterized the expected time to deliver all of the packets and showed that with collision recovery we can deliver n packets in $n + O(1)$ time slots, where n is the number of the senders. This

is significantly smaller than the delivery time of centralized scheduling and contention-based mechanisms such as slotted ALOHA.

This paper is dedicated to the throughput analysis of the system with multiple senders and receivers. We study a scenario where packets arrive at each sender according to some arrival process. In this scenario, each sender *broadcasts* a random linear combination of the packets in its queue, and the receivers perform generalized form of ZigZag decoding for interference cancellation. In this work, we establish that the stability region of the collision-recovery-enabled system matches the cut-set outer bound of the erasure network, and is *strictly larger* than that of the system with a centralized scheduler based on collision avoidance.

There are various ways to achieve the stability region of the system, but we are interested in schemes that are fully decentralized and keep the queue lengths at the senders small. Towards this end, instead of waiting for all packets to get decoded, we allow the receiver to send an ACK for one of the packets involved in each collision without being able to decode the packet immediately. We characterize a set of mild conditions for an ACK mechanism under which the receiver can eventually decode all of the packets that ever arrived at the senders. We observe that under these conditions, the system imitates the behavior of a system with a genie-aided centralized scheduler with prior knowledge of channel state realizations.

We then present two acknowledgement mechanisms that satisfy such conditions, called the *priority-based* and the *Longest-Connected-Queue* (LCQ) mechanisms. The former is based on fixing a priority order among the senders and acknowledging the one with highest priority. The latter is adapted from the centralized LCQ policy by Tassiulas and Ephremides [5], and is based on acknowledging the sender with the longest queue-length among those that are involved in each collision. The priority-based policy requires only the arrival rate information, while the LCQ policy requires only the queue-length information. Both of the proposed ACK mechanisms are throughput optimal, i.e., they achieve the cut-set outer bound of the erasure network.

The information theoretic capacity of wireless erasure network has been studied in the related literature. The works by Dana *et al.* [10], Lun *et al.* [13], and Smith and Hassibi [12] focus on a wireless erasure network with only broadcast constraints, while Smith and Vishwanath [11] study the capacity of an erasure network by considering only interference constraints. These works show how to achieve the cut-set bound of the multi-hop erasure network under specific constraints for a single unicast or multicast session. In contrast, our work takes into account both broadcast and interference constraints, and studies the stability *region* for multiple sessions over a single-hop wireless network. Another related literature investigates collision recovery methods such as the works by Tsatsanis *et al.* [14], and Paek and Neely [15]. In this literature, once a collision of k packets occurs, all senders
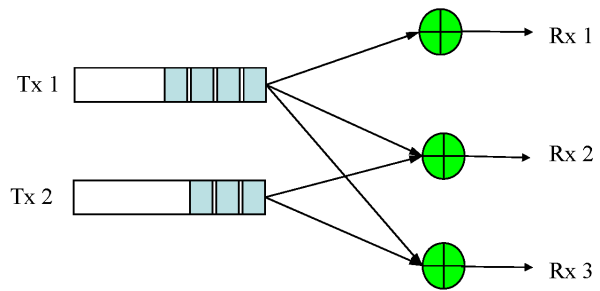

Fig. 1. Single-hop wireless network model with n senders and r receivers

remain silent until those involved in the collision retransmit another $k - 1$ times. Our proposed scheme, however, does not require such coordination among the senders.

The rest of this paper is organized as follows. In Section II, we present an abstract model of a single-hop wireless network with erasures and collision-recovery-enabled receivers. In Section III, we characterize the stability region of the single-receiver system with collision recovery. In Section IV, we generalize the results to the case of a network with multiple receivers. Finally, concluding remarks and extensions are discussed in Section V.

II. SYSTEM MODEL

The system consists of a single-hop wireless network with n senders and r receivers. We assume that a node cannot be both a sender and a receiver. The connectivity is thus specified by a bipartite graph. Fig. 1 shows an example of such a network.

We assume that time is slotted. Every sender is equipped with an infinite sized buffer. The goal of a sender is to deliver all of its packets to each of its neighbors, i.e., the set of receivers to which it is connected.

In every slot, a sender can *broadcast* a packet to its neighbors. Owing to the fading nature of the wireless channel, not all packet transmissions result in a successful reception at every neighbor. Each link between any sender i and any receiver j may experience packet erasures with probability p_{ij}. This type of erasure is to model the effect of obstacles between the senders and the receivers and channel unreliabilities. These erasures are assumed to be independent across links and over time. The independence assumption can be justified if the senders are not too close to each other and the size of each time slot is of the order of the coherence time of the channel. The channel state between i and j in time slot t is denoted by $c_{ij}(t)$.

At the end of every slot, each receiver is allowed to send an acknowledgment (ACK) to any one of the senders to which it is connected. A packet is retained in the sender's queue until it has been acknowledged by all the receivers. We ignore the overhead caused by the ACKs, and assume that the ACKs are delivered reliably without any delay.
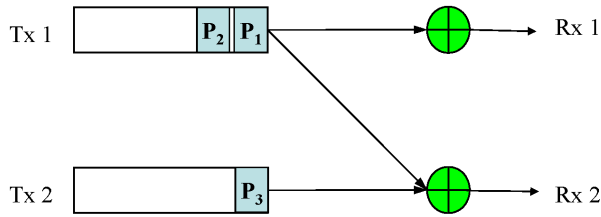
Fig. 2. Example 1: Broadcast constraint calls for network coding at the senders

Note that a collision of packets at a receiver does not immediately imply an erasure. With collision recovery schemes such as ZigZag decoding, it may be possible to extract useful information from collisions. In [17], we discussed how a collision could be thought of as a linear combination of the original packets at the sender. Next, we briefly present the main ideas from [17] for completeness.

A. Collisions as Degrees of Freedom

Consider a single-hop wireless erasure network with n senders and a single receiver. It is thoroughly discussed in [2] and [17] that each collision at the receiver consists of attenuated version of the signals sent from different senders. The transmitted packets are not perfectly aligned with each other. ZigZag decoding exploits these offsets by decoding the interference-free parts of the packets and successively canceling the interference caused by the decoded portions in other collisions. In [17], we demonstrated how to formally write each collision as a linear equation of the transmitted packets, and stated the decodability of the packets as invertibility of the transfer matrix of the system. The process of decoding by inverting this matrix is more general than the ZigZag decoding procedure. For example, if the offset of the two packets in two time slots are exactly the same, the ZigZag decoding process fails, while the transfer matrix may still be invertible because of the change in the channel gains over time. In what follows, we state the main assumption resulted from this abstraction.

Definition 1: A collision (reception) at a particular receiver is defined to be *useful* or a *degree of freedom* if there exists at least one packet involved in the collision that is not already decoded at that receiver.

Assumption 1: Consider a system with k packets at multiple senders, and collision-recovery-enabled receivers. Each receiver can decode all k packets if and only if it receives at least k useful collisions (degrees of freedom).

This assumption can be justified by choosing the size of each time slot of the order of the coherence time of the channel so that channel attenuation and phase randomly change over time.

In order to achieve highest possible throughput, we would like to avoid receiving useless collisions at the receiver. For a single receiver system, we can accomplish this by sending

acknowledgements and dropping packets before or once they are decoded. We shall discuss this in Section III. For the multiple receiver case, due to the broadcast constraint of the wireless medium, a sender broadcasting data to several receivers will have to code across packets over a finite field in order to avoid useless collision at all receivers and achieve the maximum possible throughput. Example 1 demonstrates the necessity for coding when broadcast constraint is present.

Example 1: Consider a single-hop wireless erasure network illustrated in Fig. 2. Receiver 1 needs packets P_1, P_2, and receiver 2 needs packets P_1, P_2, P_3. Suppose at $t = 1$, each sender transmits the head-of-line packet, and only the link between sender 1 and receiver 2 is broken, i.e., $c_{12}(1) = 0$. Hence, by the end of the first time slot, each receiver has collected one useful collision (degree of freedom). At $t = 2$, sender 1 transmits P_2 to avoid sending P_1 to receiver 1 again. Now suppose $c_{11}(2) = 0$. Therefore, by the end of the second time slot, receiver 1 has decoded P_1 and needs P_2, and receiver 2 has decoded P_2, P_3 and needs P_1. At this point, if sender 1 chooses either P_1 or P_2 to broadcast, the reception at one of the receivers is useless. However, if sender 1 transmits a linear combination of P_1 and P_2, both of the receivers can decode what they need by subtracting the packet that they have already decoded. In other words, at each receiver, the reception involves one packet that is not yet decoded and hence, is a degree of freedom.

Random linear coding is known to achieve the multicast capacity over wireless erasure networks [13]. Let us suppose that the sender codes across packets over the field $\mathbb{F}_q$ and that the coding coefficients are known at the receiver. We assume throughout this paper that the field size q is large enough that every collision counts as a degree-of-freedom if and only if it involves at least one packet that has not yet been decoded, i.e., Assumption 1 still holds. Every such collision counts as one step towards decoding the packets.

III. STABILITY REGION FOR THE SINGLE RECEIVER CASE

In this section, we study a special case where there is only one receiver in the network. For simplicity of notations, let p_i be the probability of channel erasure between sender i and the receiver. We shall show later in this paper that the results derived in this section generalize to the multiple receiver case. Consider a scenario where packets arrive at sender i according to an arrival process $A_i(t)$, where $A_i(t)$ represents the number of packets entering the ith sender's queue at slot t (cf. Fig. 3). We assume the arrival processes are *admissible* as defined in [4].

Assumption 2: The arrival processes satisfy the following conditions.

1) $\lim_{t \to \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \sum \mathbb{E}[A_i(t)] = \lambda_i$.
2) There exists a finite value A_{max} such that $\mathbb{E}[A_i^2(t)|\mathcal{H}(t)] \leq A_{max}^2$ for all i and t, where $\mathcal{H}(t)$ denotes the history up to time t.
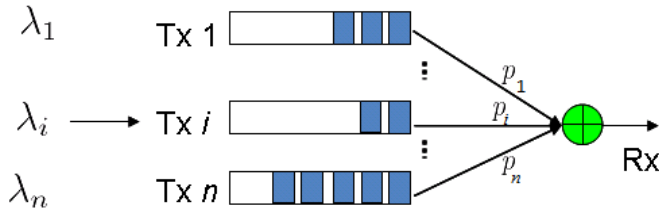3) For any $\delta > 0$, there exists an interval of size T such

Fig. 3. Single-hop network with n senders, one receiver and streaming arrivals.

Fig. 4. Stability region of a two-sender case with (a) centralized scheduling (b) collision recovery.

that for any initial slot t_0

$$\mathbb{E}\left[\frac{1}{T}\sum_{\tau=0}^{T-1}A_i(t_0+\tau)|\mathcal{H}(t_0)\right] \leq \lambda_i + \delta, \quad \text{for all } i.$$

The above conditions are easily satisfied if the arrival processes are Bernoulli processes with mean λ_i. According to the communication protocol described in Section II, a packet is dropped from a sender's queue if and only if it is *acknowledged* by the receiver. Let $\mu_i(t)$ be the number of packets dropped from the queue of the ith sender during time slot t. We assume that the $A_i(t)$ arrivals occur at the end of slot t. Thus, the evolution of $Q_i(t)$, the queue-length at sender i at time t, is given by

$$Q_i(t+1) = \max\{Q_i(t) - \mu_i(t), 0\} + A_i(t). \quad (1)$$

The goal is to characterize the stability region, which is defined as the closure of the set of arrival rates for which there exist a service policy such that the each queue in the system is stable in the following sense:

$$\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}[Q_i(\tau)] < \infty, \quad \text{for all } i. \quad (2)$$

A *centralized scheduling policy* involves choosing at most one of the senders for transmission (service) so that any collision is avoided. If the packet is delivered successfully at the receiver, an acknowledgment is fed back to the sender and that packet is dropped from the sender's queue. The centralized scheduler requires coordination among the senders as well as information about the queue-length or the arrival rates. However, it does not have access to channel state before it is realized. Since the erasure events are independent across time and different senders, no matter what scheduling policy is used, each scheduled sender may suffer from an erasure with probability p_i and hence, the assigned time slot remains empty. Thus, we have the following *necessary* conditions for the stability region:

$$\sum_{i=1}^{n}\frac{\lambda_i}{1-p_i} < 1,$$
$$\lambda_i \geq 0, \quad i=1,\ldots,n. \quad (3)$$

In fact, it can be shown that the above conditions are also sufficient. The queues can be stabilized by a centralized scheduling policy that selects the sender with the longest
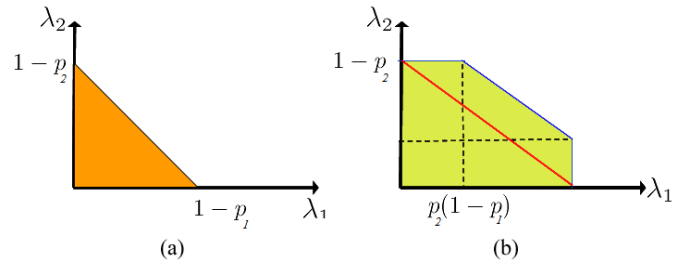
queue for transmission [4]. In summary the stability region for centralized scheduling policies is an n-dimensional simplex given by (3). An example of such region for a two-sender system is illustrated in Fig. 4(a).

Under the centralized scheduling policy the assigned sender may experience an erasure, and hence, waste time slots even when there are other senders that would not have suffered an erasure. However, if the realization of the channel state in the next time slot is known, such wastes can be avoided by choosing the transmitter from those that are connected to the receiver. Tassiulas and Ephremides [5] show that if information about channel state realization is available a priori, the following set of arrival rates can be stabilized:

$$\sum_{i\in S}\lambda_i < 1 - \prod_{i\in S}p_i, \quad \text{for all } S\subseteq\{1,\ldots,n\},$$
$$\lambda_i \geq 0, \quad i=1,\ldots,n, \quad (4)$$

The region described in (4) can be achieved by serving the sender with longest queue-length among those that are connected to the receiver. Moreover, Tassiulas and Ephremides [5] show that it is not possible to stabilize the queues for any point outside the region described in (4). This can also be seen as a consequence of *Cut-Set bound* (cf. [6]) applied to this setup. The stability region for a two-sender system is illustrated in Fig. 4(b).

In the following, we first show how to use a collision-recovery-enabled receiver to achieve the stability region given in (4) without prior knowledge about channel state realizations. We first show that under a proper acknowledgement mechanism, as long as the sender side queues are stable, the receiver will eventually decode every packet that arrives at any sender. This allows us to design acknowledgement policies based on the centralized scheduling algorithms for the case where channel state realizations are known a priori. Let us first formally define an acknowledgement policy.

Definition 2: [Acknowledgement policy] Consider a single-hop wireless network of a single receiver and n senders. Let $\boldsymbol{c}(t) \in \{0,1\}^n, \boldsymbol{Q}(t) \in \mathbb{Z}^n$ denote the channel state and queue-length vectors, respectively. Define an *ACK policy* as a mapping

$$\phi : \{0,1\}^n \times \mathbb{Z}^n \to \{0,1,\ldots,n\},$$

where $\phi(\boldsymbol{c}(t),\boldsymbol{Q}(t))$ gives the index of *at most* one sender

to be acknowledged at time t. If no sender is acknowledged, ϕ returns 0.

Define the following properties for an ACK policy ϕ:

(i)   [Feasibility] $\phi(\boldsymbol{c}(t), \boldsymbol{Q}(t)) \neq i$ when $c_i(t)Q_i(t) = 0$.

(ii)   [Efficiency] All the queues in the system are stable for any arrival rate in the stability region of the system, where the service process of queue i is given by

$$\mu_i(t) = 1, \quad \text{if and only if} \quad \phi(\boldsymbol{c}(t), \boldsymbol{Q}(t)) = i.$$

Theorem 1: Consider a single-hop wireless erasure network with n senders and one receiver which is capable of collision recovery. Each sender transmits the head-of-line packet of its queue at every time slot until acknowledged by the receiver. If the ACK policy at the receiver satisfies conditions (i) and (ii) of Definition 2, then for any arrival rate in the stability region of the system, every packet that arrives at any sender will eventually get decoded by the receiver.

Proof: First, recall that by Definition 1, every reception (potentially collision) at the receiver can be thought of as linearly independent (innovative) combination of the packets, unless all of the packets involved in the reception are already decoded. Hence, feasibility of the ACK policy implies that whenever a packet is acknowledged the receiver must have received at least one innovative linear equation involving that packet. On the other hand, if a packet is acknowledged, it will be dropped from the sender's queue. This means that the total number of degrees of freedom (linearly independent equations) at the receiver is not less than the total number of packets that have been dropped from any sender's queue.

By efficiency of the acknowledgement policy, the queue at each sender is stable. Hence, all the queues will eventually become simultaneously empty. If all the queues are empty at the same time, this means the receiver has received as many degrees of freedom (linearly independent equations) as the number of packets that ever arrived at the senders. Therefore, by Assumption 1 it can decode them all. ∎

Next, we present two acknowledgement policies and show that they satisfy both feasibility and efficiency conditions of Definition 2.

Definition 3: [Priority-based policy] Consider a permutation π of the senders with π_1 being the sender with highest priority. The *priority-based* acknowledgement policy, denoted by ϕ^π, is given by

$$\phi^\pi(\boldsymbol{c}(t), \boldsymbol{Q}(t)) = \begin{cases} 0, & \text{if } c_i(t)Q_i(t) = 0, \text{ for all } i, \\ \\ \pi_j, & \text{if } c_{\pi_j}(t)Q_{\pi_j}(t) > 0, \\ & \text{and } c_{\pi_i}(t)Q_{\pi_i}(t) = 0, \text{ for all } i < j. \end{cases}$$

In words, upon every reception the receiver acknowledges the packet from the sender with highest priority among those packets that are involved in the collision.

Note that the priority-based policy does not require the queue-length information. The acknowledgement decision merely depends on whether or not a packet is transmitted without being erased. We may easily verify that the priority-based policy is feasible. In the following, we prove efficiency of this ACK policy by showing that it achieves the vertices of the stability region given by (4). First, we provide a simple characterization of the vertices of the dominant face of the region.

Lemma 1: There exists a one-to-one correspondence between permutations of $\{1, \ldots, n\}$ and vertices of the dominant face of the region described in (4). In particular, for any permutation π, the corresponding vertex is given by

$$\lambda_{\pi_i} = (1 - p_{\pi_i}) \prod_{j < i} p_{\pi_j}, \quad i = 1, \ldots, n.$$

Proof: See [8]. ∎

Theorem 2: Consider a single-hop wireless erasure network with one receiver and n senders, where the arrival process $A_i(t)$ satisfies Assumption 2. Any vertex on the dominant face of the region given by (4) can be achieved without prior knowledge about channel state realization by employing the priority-based acknowledgement policy defined in Definition 3.

Proof: Fix a vertex, V, on the dominant face of the stability region. By Lemma 1, it corresponds to a permutation π of the senders. Without loss of generality, assume $\pi = (1, 2, \ldots, n)$. By Lemma 1, the rate-tuple corresponding to V is given by

$$\bar{\lambda}_i = (1 - p_i) \prod_{j < i} p_j, \quad i = 1, \ldots, n. \tag{5}$$

Next, we show the priority-based policy defined in Definition 3 can achieve the vertex V, i.e., for any $\epsilon > 0$, the priority-based policy stabilizes the queues with arrival rates

$$\lambda_i = (1 - p_i) \prod_{j < i} p_j - \epsilon, \quad i = 1, \ldots, n.$$

As described in the definition of the priority-based acknowledgement policy, a sufficient condition for acknowledging sender i is to have the link of sender i not erased and the links of all other senders with higher priorities erased. Note that an acknowledgment to sender i is equivalent to serving the queue at sender i by one packet. By independence of the erasures across links we obtain the following expected service rate for each sender i

$$\mathbb{E}[\mu_i(t)] \geq \bar{\mu}_i = \prod_{j < i} p_j (1 - p_i), \quad i = 1, \ldots, n.$$

Hence, by Definition 3.5 of [4], the server process $\mu_i(t)$ is admissible with rate $\bar{\mu}_i$. Moreover, the arrival process $A_i(t)$ is also admissible with rate λ_i by Assumption 2. Since $\bar{\mu}_i = \bar{\lambda}_i > \lambda_i$ for any $\epsilon > 0$, by Lemma 3.6 of [4] the sender side queues are stable. In other words, arrival rates arbitrarily close to that of vertex V can be achieved. ∎

Corollary 1: Any point in the stability region described in (4) is achievable in a distributed manner without prior

knowledge about channel state realizations by employing the acknowledgement mechanisms of the form of the priority-based ACK policy given by Definition 3.

Proof: It is sufficient to show the achievability for the rates on the dominant face of the stability region (4). Every point on the dominant face of the stability region can be written as a convex combination of the extreme points of the dominant face. Moreover, each extreme point can be achieved by a priority-based policy given in Definition 3, corresponding to that vertex. Therefore, every point on the dominant face can be achieved by time-sharing or randomization between such ACK policies.

Note that the difference between the policies achieving different vertices is in the priorities assigned to different senders. Since all of these decisions about acknowledgements are taken by the receiver, no coordination among the transmitters is necessary. ∎

The priority-based ACK policy requires knowledge of the arrival rates at the receiver to tune the acknowledgement mechanism. However, if the senders' queue-length information is available at the receiver, we can mimic the centralized scheduling algorithm by Tassiulas and Ephremides [5] that uses both queue-length information and channel state realizations. Then, we shall not need to know the arrival rates. Achievability of the stability region in (4) is then a direct consequence of the results in [5].

Definition 4: [LCQ policy] The *Longest-Connected-Queue* acknowledgement policy is defined as

$$\phi^{LCQ}(\boldsymbol{c}(t), \boldsymbol{Q}(t)) = \arg \max_i c_i(t) Q_i(t),$$

i.e., among those senders that are not suffering from an erasure, acknowledge the one with the longest queue length.

Theorem 3: The LCQ acknowledgement policy defined in Definition 4 satisfies both feasibility and efficiency conditions of Definition 2.

Proof: The feasibility of the LCQ policy easily follows form Definition 4. The proof of efficiency of the LCQ policy is the same as Lemma 2 of [5]. ∎

IV. MULTIPLE RECEIVER CASE

In this section, we generalize the results of the preceding parts to the case of a single-hop wireless erasure channel with multiple senders and receivers. Denote by $\Gamma_O(i)$ the set of receivers that can potentially receive a packet from sender i, and write $\Gamma_I(j)$ for the set of senders that can reach receiver j. Recall that the senders are constrained to *broadcast* the packets on all outgoing links. The goal of each sender is to deliver all the packets in its queue to each of its neighbors.

In the following we characterize the stability region of the network for the collision recovery approach, where packets arrive at sender i according to the arrival process $A_i(t)$. We assume the arrival processes satisfy Assumption 2 for some rate λ_i. Note that in this scenario, both broadcast and interference constraints are present, and there are multiple broadcast sessions. We show that the cut-set outer bound is achievable by combining network coding at the senders and collision recovery at the receivers.

First, let us state the outer bound given by the cut-set bound. This region is the intersection of the stability regions given by (4) for individual receivers.

Theorem 4: [Outer bound] Consider a single-hop wireless erasure network modeled as a bipartite graph, where the erasure probability of link between sender i and receiver j is denoted by p_{ij}. Assume that packets arrive at sender i with rate λ_i. For every receiver j, it is necessary for stability of the system to have

$$
\begin{aligned}
\sum_{i \in S} \lambda_i &\leq 1 - \prod_{i \in S} p_{ij}, \quad \text{for all } S \subseteq \Gamma_I(j), \\
\lambda_i &\geq 0, \quad \text{for all } i,
\end{aligned}
\tag{6}
$$

where $\Gamma_I(j)$ is the set of senders in the neighborhood of receiver j.

Proof: Assume that the system is operating under some policy $\mathcal{P}$ and is stable. Hence, the Markov chain corresponding to the queue lengths at the senders is ergodic and has a stationary distribution. Therefore, the departure rate μ_i of the queue at sender i is equal to its arrival rate λ_i. On the other hand, by independence of the information at different senders, the departure (transmission) rates should satisfy the following conditions given by the cuts between each receiver j and the senders over a bipartite graph:

$$\sum_{i \in S} \mu_i \leq 1 - \prod_{i \in S} p_{ij}, \quad \text{for all } S \subseteq \Gamma_I(j),$$

which implies the desired result. ∎

Next, we present transmission and acknowledgement policies that achieve the outer bound given by Theorem 4. The transmission policy is based on random linear network coding, and the acknowledgement policy is based on the notion of "seen" packets as defined in [16], and is built upon a single-receiver acknowledgement policy.

Definition 5: [Code-ACK policy] Consider a single-hop wireless erasure network. The Code-ACK policy is as follows:

- **Transmission mechanism**: Each sender transmits a random linear combination of the packets in its queue at every time slot.
- **Acknowledgement mechanism**: Each receiver j acknowledges the last *seen* packet of the sender given by $\phi_j\big(\boldsymbol{c}^{(j)}(t), \boldsymbol{Q}^{(j)}(t)\big)$, where ϕ_j is a single-receiver ACK policy (cf. Definition 2) of receiver j when other receivers are not present, and

$$\boldsymbol{c}^{(j)}(t) = \{c_{ij}(t) : i \in \Gamma_I(j)\},$$

$$\boldsymbol{Q}^{(j)}(t) = \{Q_{ij}(t) : i \in \Gamma_I(j)\},$$

where $Q_{ij}(t)$ the backlog of the packets at sender i not yet seen by receiver j.

Theorem 5: Consider a single-hop wireless erasure network with multiple receivers all capable of collision recovery.
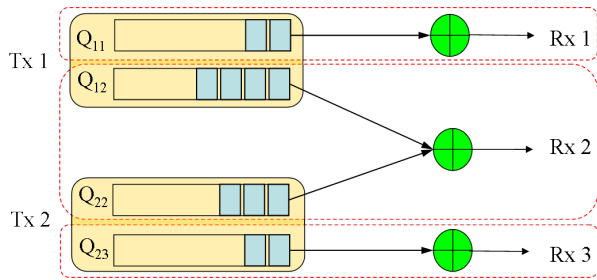
Fig. 5. Each sender-side queue viewed as two virtual queues containing the packets not seen by the corresponding receiver.

Assume the arrival processes at the senders satisfy Assumption 2. The Code-ACK policy given in Definition 5 achieves any point in the interior of the region given by (6), if the single-server ACK policies ϕ_j used in Code-ACK policy are both feasible and efficient. Moreover, every packet that arrives at a sender will eventually get decoded by all of the its neighbor receivers.

Proof: Since each sender needs to deliver all of its packets to all of its neighbors, we can think of a senders's queue as multiple virtual queues targeted for each that sender's neighbors. Each of these virtual queues contain the packets still needed by the corresponding receiver. An arrival at the sender corresponds to an arrival to each of its virtual queues, and an ACK from a receiver results in dropping a packet from the virtual queue of that receiver. A packet is dropped from a sender's original queue, if it is ACKed by all of its neighbors, in other words, if it is dropped from all its virtual queues (See Fig. 5). Therefore, we can relate the queue-length at sender i to those of the virtual queues as follows:

$$Q_i(t) \leq \sum_{j \in \Gamma_O(i)} Q_{ij}(t). \tag{7}$$

In the Code-ACK policy, receivers acknowledge a seen packet from a sender. Thus, the virtual queues at sender i corresponding to receiver j coincides with the packets at sender i not yet seen by receiver j. Moreover, upon every reception at receiver j, the corresponding virtual queue of sender $\phi_j\big(\boldsymbol{c}^{(j)}(t), \boldsymbol{Q}^{(j)}(t)\big)$ is served. Therefore, we can isolate each receiver j and its corresponding virtual queues from the rest of the network, and treat the isolated part as single-receiver erasure network.

By comparing the regions described in (6) and (4), we observe that the region for the multiple-receiver case is a subset of the one for the single-receiver case. Since ϕ_j is an efficient single-receiver ACK policy for every receiver j, all of the virtual queues are stable by Definition 2. Therefore, by (7) all of the sender-side queues are stable.

It remains to show that that all of the packets arriving at a sender are eventually decodable at its neighbor receivers. Similarly to the proof of Theorem 1, it is sufficient to show that for every ACK sent by receiver j, a degree of freedom (innovative packet) is received at receiver j. If this is the case,

by stability of the virtual queues corresponding to receiver j, they all eventually become empty and there are as many degrees of freedom at the receiver as there are unknowns. Hence, every packet arrived at the senders in $\Gamma_I(j)$ are decodable.

Now, we prove the above claim. Let receiver j send and ACK to sender i at the end of slot t. First, by feasibility of the ACK policy ϕ_j we observe that the link between i and j should be connected during slot t, and $Q_{ij}(t) > 0$. Sender i broadcasts a random linear combination of the packets in its queue which include the packets in the virtual queue Q_{ij}. If the field size is large enough, we can assume that the coefficients corresponding to at least one of the packets in virtual queue Q_{ij} is nonzero. Hence, the reception at receiver j at time slot t should have involved a packet from sender i that was not seen by receiver j. Since all decoded packets are seen [16], the collision at receiver j at time t involves a packet that is not yet decoded, and hence, it is a new degree of freedom. ∎

V. Conclusions

In this paper, we have studied the throughput performance of collision recovery methods, e.g. ZigZag decoding [2]. An algebraic representation of the collisions allowed us to view receptions at a receiver as linear combinations of the packets at the senders. The algebraic framework provides alternative collision recovery methods and generalizations for the case when the transmitted packets are themselves coded versions of the original packets.

We focused on the streaming arrivals scenario of a single-hop wireless erasure networks where both broadcast and interference constraints are present. We characterized the stability region of the collision-recovery-enabled system, showing it is strictly larger than a system with a centralized scheduler based on collision avoidance. Moreover, we demonstrated how to achieve the stability region of the system in a decentralized manner by designing proper acknowledgement mechanisms. Further, we showed that under properly designed ACK policies, the queue-lengths at the senders behave as if they are serviced by a centralized scheduler with prior knowledge of channel state realizations. Our conclusion is that collision recovery approach allows significant improvements upon conventional contention resolution approaches while not requiring any coordinations among the senders.

In this work, we assumed that each node in the network is fixed to be a sender or receiver. However, in general nodes can either be a sender or a receiver. An interesting extension of this work would be designing scheduling algorithms that properly partition the nodes into senders and receivers in each time slot.

References

[1] R. Koetter and M. Médard, *An Algebraic Approach to Network Coding*, IEEE/ACM Trans. on Networking, vol.11, no.5, pp. 782-795, Oct. 2003.
[2] S. Gollakota and D. Katabi, *ZigZag Decoding: Combating Hidden Terminals in Wireless Networks*, ACM SIGCOMM, 2008.

[3] J. Hou, J.E. Smee, H.D. Pfister and S. Tomasin, *Implementing interference cancellation to increase the EV-DO Rev A reverse link*, IEEE Communications Magazine, 2006.

[4] L. Georgiadis, M. J. Neely and L. Tassiulas, *Resource Allocation and Cross-Layer Control in Wireless Networks*, Foundations and Trends in Networking, Vol. 1, no. 1, pp. 1-144, 2006.

[5] L. Tassiulas and A. Ephremides, *Dynamic server allocation to parallel queues with randomly varying connectivity*, IEEE Trans. on Information Theory, Vol. 39, No. 2, pp. 466-478, 1993

[6] T. Cover and J. Thomas, Elements of Information Theory, Wiley & Sons, New York, 1991. Second edition, 2006.

[7] R. Motwani and P. Raghavan, Randomized Algorithms, Cambridge University Press, Ninth edition, 2007.

[8] R. E. Bixby, W. H. Cunningham, and D. M. Topkis: Partial order of a polymatroid extreme point, Math. Oper. Res., 10 (1985), 367-378.

[9] D. Bertsekas and R. Gallager, *Data networks (2nd ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1992.

[10] A. F. Dana, R. Gowaikar, R. Palanki, B. Hassibi and M. Effros, *Capacity of wireless erasure networks*, IEEE/ACM Trans. on Networking, vol.52, no.3, pp. 789-804, Mar. 2006.

[11] B. Smith and S. Vishwanath, *Unicast Transmission Over Multiple Access Erasure Networks: Capacity and Duality*, IEEE Information Theory Workshop, 2007.

[12] B. Smith and B. Hassibi, *Wireless Erasure Networks With Feedback*, Proc. of ISIT, 2008.

[13] 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, Mar. 2008.

[14] D. S. Lun, M. Médard, R. Koetter, and M. Effros, *On coding for reliable communication over packet networks* , IEEE Trans. on Signal Processing , vol.48, no.3, Mar. 2000.

[15] J. Paek and M. J. Neely, *Mathematical Analysis of Throughput Bounds in Random Access with ZigZag Decoding*, Proc. of WiOpt, 2009.

[16] J. K. Sundararajan, D. Shah, and M. Médard, *ARQ for network coding*, Proc. of ISIT, 2008.

[17] A. ParandehGheibi, J. K. Sundararajan, and M. Médard, *Collision Helps - Algebraic Collision Recovery for Wireless Erasure Networks*, Proc. of IEEE Wireless Network Coding Workshop, 2010.