Feedback in Coded Wireless Packet Networks

Stephan M. Günther, Maximilian Riemensberger, and Wolfgang Utschick
Associate Institute for Signal Processing, Department of Electrical Engineering
Technische Universität München
Email: {guenther,riemensberger,utschick}@tum.de

Abstract—In this paper we propose an end-to-end feedback mechanism for intra-session random linear network coding with opportunistic routing in wireless packet networks with lossy links. We focus on bidirectional network coding, i.e., forward and reverse flows between two nodes are coded together, which is key for efficient utilization of the wireless medium as it allows intermediate nodes to relay traffic in both directions with a single transmission. We analyze the performance in terms of decoding and acknowledgement times in a three-node network when nodes are fully backlogged. The results are compared to the theoretic lower bound obtained by solving the network’s flow formulation. In addition, we derive symmetric injection rates from the these results which the network should be able to sustain. The evolution of source backlogs and decoding/acknowledgement over time are simulated, demonstrating that backlogs remain bounded. The insight gained will help in developing a generalized feedback model for coded wireless mesh networks, which is to the best of our knowledge an open problem.

I. INTRODUCTION

In this paper we propose an end-to-end acknowledgement scheme for intra-session network coding that is suitable for bidirectionally coded sessions, i.e., the forward and reverse flow of the data traffic of a pair of nodes are coded together. The principle is illustrated in Figure 1: Nodes $s$ and $t$ mutually exchange data, which may be opportunistically overheard by the relay $r$. In contrast to similar intra-session coding schemes such as MORE [1], [2], the relay may code over packets belonging to the same session but different directions, i.e., a session consists of two opposite flows between the same tuple of nodes. This allows the relay to serve both traffic directions with a single broadcast. A similar scheme has been proposed for COPE in [3]. However, the paradigm followed by COPE differs as it aims at inter-session coding and avoids coding on packets destined for the same next-hop.

Feedback (acknowledgement) schemes for network coding have received little attention so far. Prototypical implementations such as MORE [2] use uncoded end-to-end feedback that is routed reliably along the shortest path from destination to source according to the ETX [4] metric. As MORE does not jointly code on different directions, this scheme may be near optimal. In [5] per-hop and end-to-end feedback are analyzed in the context of intra-session network coding. In particular it is noted that feedback does not need to include information about individual packets but about degrees of freedom, i.e., the number of linear independent packets received by the destination. COPE [3] uses per-hop acknowledgements instead of end-to-end feedback. In particular, it requires that all next-hops acknowledge successful reception of a packet. The reason is COPE’s functional principle which demands that next-hops are able to decode.

We believe that bidirectional coding with end-to-end feedback is key for efficient usage of the wireless medium when coding is restricted to packets with a common set of endpoints. However, this results in a tight coupling of both flows as endpoints must switch generations (blocks) in a synchronized manner. Therefore, a well-elaborated feedback scheme is required that is considerably more complex than just acknowledging successful decoding as done by MORE. We propose a lock-based acknowledgement scheme that allows either source of a bidirectional flow to lock the current generation. If the remote end notices the lock, it knows that no further source packets will be encoded in the current generation. Afterwards, successful decoding is indicated by matching dimensions of coding subspaces at both pairs of source and destination. Without the ability of locking, nodes cannot switch safely from one generation to the next as there is no guarantee that the remote end will not insert further packets into the current generation. Therefore, one would either have to assume instantaneous feedback or nodes would have to store an infinite number of generations in case of asymmetric traffic needs.

The remainder of this paper is organized as follows: The network model is introduced in Section II. Section III introduces bidirectional random linear coding for unicast connections. Section IV discusses a suitable feedback scheme. Details regarding channel access are given in Section V. Simulation results are given in Section VI. Finally, Section VII concludes the paper.

II. NETWORK MODEL

We consider a coded wireless packet network consisting of three nodes $N = \{s, t, r\}$ as depicted in Figure 1. The nodes $s$ and $t$ exchange data with the help of a relay node $r$. We define $S = \{s, t\}$ as the set of source/destination nodes. Time
where transmissions are broadcasts and may be overheard by any others.

During the contention phase nodes draw independently a random, non-negative delay to wait before starting to transmit. The node with the unique shortest delay wins the contention phase and broadcasts a single frame. All other nodes remain silent. This mechanism essentially resembles a time-continuous variant of CSMA/CA with ideal sensing, no interference and no back-off. Channel access of node \( i \in \mathcal{N} \) is modeled by means of a binary random variable \( Z_i(\tau) \), which is one if \( i \) transmits during time slot \( \tau \) and zero otherwise. Ideal sensing implies that exactly one node is transmitting in any time slot \( \tau \), i.e., \( \sum_{i \in \mathcal{N}} Z_i(\tau) = 1 \). The delay of each node \( i \) is distributed according to an exponential distribution mapped continuously and monotonically onto the contention phase. The rate of the exponential distribution is denoted by \( w_i(\tau) \geq 0 \), referred to as node weight, and is adapted over time to prioritize nodes which have more packets to send than others.

After the contention phase, the node which has won the medium access transmits a fixed length data packet. Packet transmissions are broadcasts and may be overheard by any node within range. The binary random variables \( E_{ij}(\tau) \) for all \( i, j \in \mathcal{N} \) represent packet losses from \( i \) to \( j \) in time slot \( \tau \). Packet losses are assumed to be i.i.d. and time-invariant with average \( \mathbb{E}[E_{ij}] = \epsilon_{ij} \). A packet is successfully transmitted from \( i \) to \( j \) at time \( \tau \) if \( Z_i(\tau)(1-E_{ij}(\tau)) = 1 \), i.e., \( i \) has won the contention phase and the packet has not been lost at node \( j \) in the transmission phase.

Packets are generated by a source process at all nodes in \( \mathcal{S} \). The random variable \( X_i(\tau) \) denotes the number of packets generated by \( i \) at time \( \tau \). We assume that the source processes are i.i.d. over time and independent across all nodes. The average rate at which packets are generated at \( i \) is \( \mathbb{E}[X_i(\tau)] = \lambda_i \). Newly generated packets are stored in a queue for later transmission. The length of this queue, namely the backlog \( U_i(\tau) \), evolves as

\[
U_i(\tau + 1) = U_i(\tau) - Y_i(\tau) + X_i(\tau),
\]

where \( Y_i(\tau) \) denotes the number of packets that node \( i \) adds to the current generation (block) of packets. In each time slot \( \tau \), \( Y_i(\tau) \) is chosen as the minimum of the current backlog \( U_i(\tau) \) and the remaining space in the current generation. If a new generation is started at time \( \tau \), as many backlogged packets as possible are added immediately.

### III. Bidirectional Random Linear Network Coding

Data packets from all nodes are coded using a random linear block code on a finite field \( \mathbb{F}_q \) of order \( q \). In each code generation \( N \) packets are encoded. The code can be represented by an \( N \)-dimensional vector space \( V \) over the field \( \mathbb{F}_q \), without loss of generality we may choose \( V = \mathbb{F}_q^N \). Each coded packet that is transmitted gets its \( N \) coding coefficients prepended to communicate the random code to the decoder at the intended destination. We assume that the generations are transmitted subsequently, i.e., only one generation is active in the network at all times. Therefore, we omit an explicit generation index. Furthermore, we omit the time index \( \tau \) as well if there is no ambiguity regarding the time slot.

A bidirectional coding session consists of two flows: one from \( s \) to \( t \) and the other from \( t \) to \( s \). The flows are assigned to coding subspaces \( V^s \) and \( V^t \), respectively, separating the coding space into two subspaces such that \( V = V^s \oplus V^t \). The total dimension of the coding space is thus partitioned as \( \dim V = \dim V^s + \dim V^t \). The available packets at time \( \tau \) at source node \( s(\tau) \) assigned to the current generation are represented by the subspace \( V^s_{\tau} = V^s \cap \{ V^t_{\tau} \} \). If the source process has generated at least \( \dim V^s + \dim V^t \) packets at \( s(\tau) \) since the previous generation was completed, then \( V^s_{\tau} = V^s \cap \{ V^t_{\tau} \} \) holds. Each coded packet that is transmitted in the network at time \( \tau \) is element of the current coding subspace \( V^s \oplus V^t \).

We impose additional structure on the coded transmissions to simplify the following analysis on linear dependencies, yet without loss of generality. In particular, we suppose that the source nodes transmit only coded packets from \( V^s \) and \( V^t \), respectively, but not from any subspace of received coding vectors. Therefore, all received packets at the relay \( r \) are either from \( V^s \) or \( V^t \). The received packets are represented by two subspaces \( V^r_{\tau} \) and \( V^t_{\tau} \) for flow \( V^s \) and \( V^t \), respectively. For example if \( r \) receives a coded packet from \( s \), this packet is from the linear hull \( V^s_{\tau} + V^t_{\tau} \) of \( V^s_{\tau} \) and \( V^t_{\tau} \), and \( t \) can remove the contribution of \( V^t_{\tau} \) from that packet by linear elimination since \( V^t_{\tau} \subset V^s_{\tau} \). The remaining packet after this elimination is from \( V^s_{\tau} \subset V^t_{\tau} \). This structure shows that we can treat the coding and decoding of both flows separately within each generation. For the remainder of this section we discuss only the flow \( s \) with source \( s \) and destination \( t \). The results apply analogously to the opposite flow \( t \).

#### A. Innovative packets

In this section we consider a packet transmission at time \( \tau \) for flow \( s \) and thus drop both flow and time indices. We assume that nodes choose coding vectors independently and uniformly distributed from their coding subspace. We are interested in the probability that a broadcast by the source \( s \) or the relay \( r \) is innovative with respect to the previously received packets at the destination, namely the current coding subspace \( V_i \) of the destination. To this end let \( v \) denote a random coded packet drawn from the coding subspace of the transmitting node. That is, the packet \( v \) is uniformly drawn from the vector space \( V_i \) when \( i \in \mathcal{N} \) transmits. Broadcasting the packet \( v \) changes the dimensions of the coding subspaces at the receiving nodes. In order to characterize innovative packets, i.e., packets that are not in the coding subspace of the destination \( V_i \), we use the generalized quotient spaces \( V_i/V_i \) at the source and \( V_i/V_i \) at the relay.
Definition 1. Let $V, W$ be subspaces of a vector space $U$. We define the equivalence class $[v] = \{v + w : w \in V \cap W\}$ for all $v \in V$ and the generalized quotient space denoted by $V/W = \{[v] : v \in V\}$.\footnote{If $W \subset V$, the generalized quotient space coincides with the quotient space since $V/W = V/(W \cap V)$ holds.}

The quotient space $V/W$ can be interpreted as the vector space $V$ where vectors that differ only by elements in $W$ are declared equivalent, and hence elements in $W$ are declared irrelevant. In the context of network coding, we are interested particularly in the quotient spaces $V_s/V_t$ and $V_r/V_t$ as packets that have already reached the destination are irrelevant (not innovative) to all nodes in the network. Since we consider uniformly distributed random packets, it is sufficient to track the dimensions of the relevant coding subspaces and quotient spaces. To this end we define the dimensional shift $\Delta V_i$ and $\Delta V_i/V_j$, respectively, for coding subspaces and quotient spaces.

First, we consider the evolution of the coding space $V_t$ at the terminal with respect to the random packet $v$ that is transmitted. Packet $v$ is innovative if it is not element of $V_t$. In this case $\dim V_t$ increases by one, namely $\Delta V_t = 1$. The probability that $t$ receives an innovative packet is given by

$$
\Pr[\Delta V_t = 1] = \Pr[Z_s = 1](1 - \epsilon_{st})(1 - q^{-\dim V_s/V_t}) + \Pr[Z_r = 1](1 - \epsilon_{rt})(1 - q^{-\dim V_r/V_t}).
$$

The last factors of both terms are the probabilities that packets generated at the source and at the relay, respectively, are innovative to $t$. These probabilities depend only on the dimensions of the quotient spaces $V_s/V_t$ and $V_r/V_t$. Therefore, we study their dimensional shift next.

The quotient space of innovative packets at the source $V_s/V_t$ can increase only when new packets are added to the current coding generation by the source process. Packets belonging to the $st$ flow which are transmitted by nodes other than $s$ are known to $s$ already and, therefore, cannot increase the dimension of $V_s/V_t$. However, $\dim V_s/V_t$ can be decreased by one if no new packets are added and the destination receives an innovative packet, which increases $\dim V_t$ by one. Therefore, we can conclude that

$$
\Pr[\Delta V_s/V_t - Y_s = -1] = \Pr[\Delta V_t = 1]
$$

where $Y_s$ denotes the number of packets that the source process adds to the coding space at $s$.

Finally, we need to characterize the change of $\dim V_r/V_t$, which influences both $\Delta V_t$ and $\Delta V_s/V_t$. Suppose the relay transmits a packet. Then $\dim V_r/V_t$ decreases by one if the packet is innovative and received at $t$. Furthermore, $\dim V_r/V_t$ decreases also if $s$ transmits an innovative packet that would be not innovative if the relay space were known at the terminal. That is, the packet $v$ is an element from the linear hull $V_t + V_r$ of $V_t$ and $V_r$. The dimension of this space is given by

$$
\dim(V_t + V_r) = \dim V_t + \dim V_r - \dim(V_t \cap V_r) = \dim V_t + \dim V_r/V_t.
$$

The probability that the vector $v$ drawn uniformly from $V_s$ is in $V_t + V_r$ and innovative to $t$ is given by

$$
\Pr[v \in V_s \cap V_t + V_r] = \Pr[v \in V_s \cap V_t + V_r | v \in V_s] \Pr[v \in V_s] = (1 - q^{-\dim(V_t + V_r) + \dim(V_s + V_t)}) q^{-\dim V_t + \dim(V_t + V_r)} = (1 - q^{-\dim V_r/V_t}) q^{-\dim V_r/V_t + \dim V_t/V_s + \dim V_t/V_r} = (q^{\dim V_t/V_s - 1}) q^{-\dim V_r/V_t}.
$$

We remark that $v \in V_r + V_t$ implies $v \in V_s$ since $V_r$ and $V_s$ are subspaces of $V_t$. Therefore, the probability that $\dim V_r/V_t$ decreases by one due to a transmission by the relay or the source is given by

$$
\Pr[\Delta V_r/V_t = -1] = \Pr[Z_r = 1](1 - \epsilon_{rt})(1 - q^{-\dim V_r/V_t}).
$$

On the other hand, the dimension of the relay quotient space can also increase when $s$ transmits a packet $v$. This happens when the packet $s$ is not in $V_r + V_t$ and is only received by the relay but not the destination. The probability that $\dim V_t/V_r$ increases by one is thus given by

$$
\Pr[\Delta V_t/V_r = 1] = \Pr[Z_s = 1](1 - \epsilon_{st})(1 - q^{-\dim V_s/V_t} + \dim V_t/V_r).
$$

We remark that $\dim V_s = \dim V_t + \dim V_s/V_t$ is implied by $V_t \subset V_s$. Therefore, $\dim V_s$ is redundant given the other two dimensions. Additionally, since the aforementioned probabilities (2), (3), (6) and (7) do not depend explicitly on the dimensions of $V_r$, $V_s \cap V_t$, and $V_r + V_t$, these are also redundant for the description of the behavior of a generation over time. Therefore, we can conclude that the coding space at the terminal $V_t$ and the quotient spaces $V_s/V_t$ and $V_r/V_t$ yield a complete description of the network coding state.

IV. FEEDBACK

Feedback and synchronous generation switching is an issue in bidirectional network coding, which is explained by the following example: Suppose $s$ has generated $\dim V_s^{st}(\tau) = \dim V_s^{st}$ packets destined for $t$ at time $\tau$, whereas $t$ has generated $\dim V_t^{ts}(\tau) < \dim V_t^{ts}$ packets destined for $s$. At this point both session endpoints should agree to switch to the next generation as soon as possible. Otherwise, the flow $st$ is blocked indefinitely while waiting for the reverse flow, or $s$ must make a one-sided decision to switch to the next generation. The former case is obviously not desirable, for the latter case both endpoints would have to allocate an arbitrary amount of generations in case of asymmetric traffic needs.
Therefore, $t$ needs to signal to $s$ that it does not intend to add any further packets to this generation. Finally, after $t$ has decoded all packets from $s$ and vice versa they need to acknowledge successful decoding to each other.

We introduce lock and decoding state estimators to handle generation locking and acknowledgements, which comprise the knowledge of each node about the state of the source and the decoder at time $\tau$. We denote by $a^st_s$ ($a^ts_s$) the estimated $\dim V^st_s$ ($\dim V^ts_s$) at node $i$ and by $l^st_s$ ($l^ts_s$) the estimated binary lock state at node $i$. A lock state $l^st_s(\tau) = 1$ means that $s$ does no longer insert packets into the current generation, and therefore, $\dim V^st_s$ will stay constant until the current generation is finished. The respective lock states at the other nodes $r$ and $t$ signal that they are aware of the lock due to the source. When a node broadcasts a packet in time slot $\tau$, it includes these estimators in the packet header. Additionally, the actual number of source packets for each flow $\dim V^st_s$ and $\dim V^ts_s$ are included in the header after the generation is locked for the respective flow. This ensures that the destinations know how many packets they need to expect for decoding.

The evolution of the decoding estimators over time can be described by simple maximum operations. For flow $st$ the estimators at time $\tau + 1$ at nodes $j \in \mathcal{N}\setminus \{t\}$ are given by

$$a^st_s(\tau + 1) = \max \{a^st_s(\tau), Z_s(\tau)(1 - E_{ij}(\tau))a^st_s(\tau)\}, \quad (8)$$

i.e., the maximum of the estimator at node $i \in \mathcal{N}$ and the estimator included in the received packet. The destination node $t$ of flow $st$ sets

$$a^st_s(\tau) = \dim V^st_s(\tau) \quad (9)$$

for all $\tau$. The analogous equations hold for the flow $ts$.

The lock states couple both flows to allow for a synchronous switching from one generation to the next. The lock state $l^st_s$ means that the source node of flow $st$ has stopped inserting packets into the current generation and has communicated the actual generation size together with the lock signal to node $i$. The locks of flow $st$ are updated as follows: The source node $s$ sets its lock state $l^st_s(\tau + 1)$ of flow $st$ to one if either $\dim V^st_s(\tau + 1) = \dim V^st_s$, or if it receives a packet from node $i$ in time slot $\tau$ where the included lock state $l^st_s(\tau)$ of the reverse flow $ts$ is equal to one. The destination and relay nodes set their lock state $l^ts_s(\tau + 1)$ of flow $st$ to one if they receive a packet from a neighboring node $j$ in time slot $\tau$ where the included lock state $l^ts_s(\tau)$ of the same flow is equal to one. The locks of flow $ts$ are updated analogously with the roles of source and destination reversed.

Based on the lock states and acknowledgements we can define when a generation is complete. First, a generation is decoded if $\dim V^st_s(\tau) = \dim V^st_s(\tau), \dim V^ts_s(\tau) = \dim V^ts_s(\tau)$, and $l^st_s(\tau) = l^ts_s(\tau) = 1$. That is, both flows are locked at their respective source and all packets that have been injected into this generation are decoded at their respective destinations. If in addition $l^st_s(\tau) = l^ts_s(\tau) = 1$ holds, then both destinations are aware of having received and decoded all packets of the respective flow in the current generation.

Finally, if in addition to those conditions $a^st_s = \dim V^st_s$ and $a^ts_s = \dim V^ts_s$, then the generation is acknowledged, i.e., both sources know that their respective destinations have successfully decoded all packet that they have injected into this generation. Therefore, they can safely switch to the next generation and start adding packets from the backlog to the next generation. At this point all coding, acknowledgement, and lock states are reset.

V. NODE WEIGHT AND CHANNEL ACCESS

To achieve good performance of the bidirectional coding scheme, decoding and acknowledgement times need to be kept as low as possible. Therefore, the medium access needs to take into account which nodes have packets to transmit. In our considered random access scheme, this can be achieved by designing the weights $w_i(\tau)$ appropriately. For this we adapt a packet transmission credit scheme, which has been proposed for the unidirectional network coding approach MORE [2].

Each node $i \in \mathcal{N}$ is assigned a nonnegative credit $\alpha^st_s$ ($\alpha^ts_s$) that quantifies the fraction of packets it should broadcast to serve flow $st$ ($ts$) for each received (from other nodes) or injected (from the source process) packet. Additionally, at the destination it quantifies the fraction of acknowledgments that should be injected towards the source of the reverse flow $ts$ ($st$). The theoretically optimal credits for unidirectional coding have been computed in [1]. We use the simpler credit approximation by MORE [2] separately for each flow $st$ and $ts$ with the fixed order source, relay, destination instead of an ETX metric based order. Then we implement a bidirectional credit based access by combining the weight functions for both flows.

We define weight functions for flow $st$ at nodes $s$ and $r$ by

$$w^st_s(\tau + 1) = \max \{1, w^st_s(\tau) - Z_s(\tau) + \alpha^st_s Z_s(\tau)\}, \quad (10)$$

$$w^st_r(\tau + 1) = \max \{1, w^st_r(\tau) - Z_r(\tau) + \alpha^st_s Z_s(\tau)(1 - E_{sr}(\tau))\}, \quad (11)$$

respectively, where $Z_s(\tau)(1 - E_{sr}(\tau))$ indicates that $r$ received a packet from $s$. At the destination $t$ for flow $st$ the weight function is fixed at $w^st_t(\tau) = 1$ as long as the generation is not locked and decoded at $t$. After the generation has been decoded at $t$ and needs to be acknowledged, the weight function is fixed at $w^st_t(\tau) = \alpha^ts_s$, i.e., the credit for one packet that needs to be delivered from $t$ to $s$. The weight functions for the reverse flow $ts$ are defined analogously with the roles of source and destination exchanged. The channel access weight function for all nodes $i \in \mathcal{N}$ is then defined as

$$w_i(\tau) = \max \{w^st_i(\tau), w^ts_i(\tau)\}. \quad (12)$$

VI. SIMULATION RESULTS

We simulate bidirectional random linear network coding with our feedback scheme and determine the times $\tau_{dec}$ and $\tau_{ack}$ between initializing a new generation and successful decoding and acknowledgement, respectively. To derive the maximum injection rate from these time values, we assume a
network in fully backlogged state, i.e., both source nodes always have a complete generation of source packets to be sent. The results are compared to a lower bound $\tau_{dec}$ for decoding, which is obtained by solving the network flow problem [6]. We assume orthogonal scheduling for the lower bound and neglect random linear dependencies between packets. As a result $\tau_{dec}$ is invariant to the generation size $\dim V = N$ and the coding field size. For better comparability of results we normalize time values with respect to the number of packets transmitted per generation, i.e., $\tau$ is divided by $N$. Results are shown in Figure 2a. Considering $\epsilon_r = 0.2$ (solid lines) we see a clear advantage for larger generations. The reason is that locking and acknowledging has to be done at the end of each generation and thus larger generations exhibit a lower overhead. A similar trend applies to higher error rates as indicated by dashed lines.

The maximum symmetric injection rate that can be sustained by the simulated network is $\lambda = N / (2st)$. For $\epsilon = 0.8$ we derive $\lambda \approx 0.278$ from Figure 2a. The development of backlogs over time and absolute values for $\tau_{dec}$ and $\tau_{ack}$ are shown in Figure 2b. The mean values for decoding and acknowledgement are $\tau_{dec}^* \approx 221$ and $\tau_{ack}^* \approx 227$. The long term averages, when divided by $N$, converge to the results shown in Figure 2a.

VII. CONCLUSION

We presented a bidirectional network coding and feedback scheme, which codes jointly over two flows with opposite directions. The scheme is able to keep the flows in both directions synchronized with respect to the network coding generation, which is necessary to prevent the flows from stalling each other. It achieves good performance with respect to throughput and decoding and acknowledgement times over a wide range of parameters in a three-node relay network with lossy packet broadcasts. Further research should be directed towards such schemes in larger mesh networks with general topology and nonorthogonal medium access.

Locking and acknowledging generations induces delays in the order of the round trip time (end-to-end) delay between source and destination. To mitigate the negative effect of these stall times, a sliding window approach may be applied, i.e., the source may start the next generation before the previous one has been acknowledged as long as minimum and maximum generation numbers do not exceed a certain threshold. A similar approach is employed by TCP [7] for congestion avoidance, and, in the context of network coding, has been proposed for in [8] to increase throughput.

REFERENCES