Design of Efficient Robust Network Codes for Multicast Connections

Graham Booker and Alex Sprintson

Texas A&M University, College Station, Texas, USA
{gbooker,spalex}@tamu.edu

Abstract—We consider the problem of establishing reliable multicast connections across a communication network. Our goal is to provide instantaneous recovery from single edge failures. With instantaneous recovery, all destination nodes can decode the packets sent by the source node even if one of the edges in the network fails, without the need of retransmission or rerouting. We build on the novel technique of network coding that offers significant advantages over standard solutions such as disjoint path routing and diversity coding.

We begin by focusing on the case in which all network edges have equal capacity. For this case we present a network coding algorithm that constructs a robust network code over a small field. The algorithm takes advantage of special properties of the Maximum Rank Distance codes. Second, we consider a case of non-uniform edge capacities. We show that for the special case in which a small number of packets need to be transmitted from the source to destination nodes, special combinatorial properties of minimum coding networks can be exploited for constructing efficient robust network codes.

I. INTRODUCTION

In recent years, a significant effort has been devoted to improving the resilience of communication networks to failures and increasing their survivability. Edge failures are frequent in communication networks due to the inherent vulnerability of the communication infrastructure [1]. With the dramatic increase in data transmission rates, even a single failure may result in vast data losses and cause major service disruptions for many users. Accordingly, there is a significant interest in network recovery mechanisms that enable a continuous flow of data from the source to the destination with minimal data loss in the event of a failure.

In this paper, we consider the problem of establishing reliable multicast connections across a communication network with uniform and non-uniform edge capacities. Our goal is to provide instantaneous recovery from single edge failures. The instantaneous recovery mechanisms ensure continuous flow of data from the source to the destination node, with no interruption or data loss in the event of a failure. Such mechanisms eliminate the need for packet retransmissions and rerouting. Instantaneous recovery is typically achieved by sending packets over multiple paths in a way that ensures that the destination node can recover the data it needs from the received packets. The three major methods for achieving instantaneous recovery are dedicated path protection scheme [1], diversity coding [2], and network coding [3]–[7]. However, only the network coding technique can achieve the optimum in terms of the maximum number of packets that can be sent reliably from the source s to all terminals.

Consider the multicast network depicted in Figure 1 which needs to deliver two packets per communication round from the source node s to two destination nodes t1 and t2. In this network, edges (s, v1) and (s, v2) can send two packets per communication round, while all other edges can send only one packet per communication round. The figure shows an encoding scheme that delivers two packets p1 and p2 to terminals t1 and t2 over a single round such that both destination nodes can decode the packets sent by the source node in any single edge failure scenario. Note that without the encoding operation at the intermediate nodes v3, v4, and v5, it would not be possible to send two packets with instantaneous recovery.

Related work

The network coding technique has been introduced in the seminal paper of Ahlswede et al. [3]. Initial work on network coding has focused on multicast connections. It was shown in [3] that the maximum rate of a multicast network is equal to the minimum total capacity of a cut that separates the source from a terminal. This maximum rate can be achieved by using linear network codes [8]. Koetter and Médard [4] developed an algebraic framework for linear network codes.
Ho et al. [9] showed that the maximum rate can be achieved by using random linear network codes. Jaggi et al. [10] proposed a deterministic polynomial-time algorithm for finding feasible network codes in multicast networks. Network coding algorithms resilient to malicious interference have been studied in [11], [12], and [13]. Comprehensive surveys on the network coding techniques are available in the recent books [14], [15], and [16].

The idea of using network coding for instantaneous recovery from edge failures was first described by Koetter and Medard [4]. They showed that if the network has a sufficient capacity to recover from each failure scenario (e.g., by rerouting) then instantaneous recovery from each failure scenario can be achieved by employing linear network codes. Ho et al. [5] presented an information-theoretic framework for network management in the presence of edge failures. Using network coding for reliable communication was also discussed in [6] and [7]. References [17] and [18] describe practical implementations of network coding and demonstrate its benefits for improving reliability and robustness in communication networks.

A. Our contribution

In this paper we propose efficient algorithms for construction of robust network codes over small finite fields. We consider two major cases. In the first case, we assume that all edges of the network have uniform capacity, while in the second case the capacity of network edges can vary. For the first case we present an efficient network coding algorithm that identifies a robust network code over a small field. The algorithm takes advantage of special properties of Maximum Rank Distance (MRD) codes [19]. For the second case, we focus on settings in which the source node needs to deliver two packets per time unit to all terminals. We show that in this case, a special topological properties of robust coding networks can be exploited for constructing a network code over a small finite field.

A robust network code for multicast networks can be established through the standard network coding algorithm presented in [10]. However, this algorithm is designed to handle arbitrary failure patterns, and, as a result, requires a field size of $O(|E|)$ in the case of single edge failures, where $E$ is the set of network edges. In contrast, our scheme requires a small field size ($O(k)$, where $k$ is the number of terminals), which does not depend on the size of the underlying communication network. The size of the finite field is a very important factor in practical implementation schemes [17] as it determines the amount of communication and computational overhead. In addition, the computational complexity of our algorithm is smaller than that of the existing solutions.

II. Model

A. Multicast Network

We consider a multicast network $N$ that uses a directed acyclic graph $G(V, E)$ to send data from source $s$ to a set $T$ of $k$ destination nodes $\{t_1, \ldots, t_k\} \subset V$. The data is delivered in packets. We assume that each packet is an element of a finite field $F_q = GF(q)$. We also assume that the data exchange is performed in rounds, such that each edge $e \in E$ can transmit $c(e)$ packets per communication round. We assume that $c(e)$ is an integer number and refer to it as the capacity of edge $e$. At each communication round, the source node needs to transmit $h$ packets $R = (p_1, p_2, \ldots, p_h)^T$ from the source node $s \in V$ to each destination node $t \in T$. We refer to $h$ as the rate of the multicast connection. It was shown in [3] and [8] that the maximum rate of the network, i.e., the maximum number of packets that can be sent from the source $s$ to a set $T$ of terminals per time unit, is equal to the minimum capacity of a cut that separates the source $s$ from a terminal $t \in T$. Accordingly, we say that a multicast network $N$ is feasible if any cut that separates $s$ and a terminal $t \in T$ has at least $h$ edges. We say that a coding network $N$ is minimal if any network formed from $N$ by removing an edge or decreasing the capacity of an edge is no longer feasible. It is easy to verify that the capacity of each edge in a minimal network is bounded by $h$.

B. Coding Networks

For clarity of presentation, we define an auxiliary graph $\tilde{G}(V, A)$ formed by the network graph $G(V, E)$ by substituting each edge $e \in E$ by $c(e)$ parallel arcs that have the same tail and head nodes as $e$; each arc can transmit one packet per communication round. We denote by $A(e) \subseteq A$ the set of arcs that correspond to edge $e$. In what follows we only refer to packets sent at the current communication round. The packets sent in the subsequent rounds are handled in a similar manner.

A network code is defined by associating each arc $a(v, u) \in A$ in the network with a local encoding function $f_a$. The local encoding function specifies the packet transmitted by arc $a$ as a function of the packets available at or received by the tail node of $a$ in the current communication round. More specifically, for each outgoing arc $a(s, u) \in A$ of the source node $s$, $f_a$ is a function of the original $h$ packets $P_i$, i.e., $f_a : P^h \rightarrow F$. For any other arc $a(v, u) \in A$, $v \neq s$, $f_a$ is a function of the packets received by node $v$ at the current round, i.e., $f_a : P^l \rightarrow F$, where $l$ is the number of incoming arcs of $v$ in $G$. A network code $C$ is a set of encoding functions associated with the arcs in $A$, i.e., $C = \{f_a \mid a \in A\}$. In a linear network code all packets are elements of a finite field and all local encoding functions are linear functions over that field.

C. Robust Coding Networks

As mentioned in the introduction, we assume that only one of the edges in the network can fail at any time. Since a failed edge $e$ cannot transmit packets, we assume that the encoding function $f_a$ of each arc $a \in A(e)$ is identically equal to zero, i.e., $f_a = \mathbf{0}$. To guarantee instantaneous recovery, it is sufficient to ensure that for each edge failure there exists a set of $h$ linearly independent packets received by $t$.

We distinguish between two types of robust networks codes. In strongly robust network codes the local encoding coefficients of all arcs in $A$ remain the same, except for the arcs $A(e)$ that correspond to the failed edge $e$ which are assigned zero encoding coefficients. In weakly robust network codes, the arcs that are located downstream of the failed edge $e$ are allowed to change their encoding coefficients, while all the encoding coefficients that correspond to other edges must remain the same.
Definition 1 (Strongly Robust Network Code): A network code $\mathcal{C}$ is said to be strongly robust if for each $e \in E$ it holds that the network code $\mathcal{C}'$ formed from $\mathcal{C}$ by assigning zero encoding coefficients to arcs in $A(e)$ is feasible.

We proceed to discuss weakly robust network codes. When an edge $e(v, u) \in E$ fails, the set of nodes of the network can be divided into two subsets $V'_e$ and $V''_e$. The set $V'_e$ includes all descendants the head node $u$ of $e$, while the set $V''_e = V \setminus V'_e$ includes all ancestors of the tail node $v$ of $e$, as well as all other nodes not included in $V''_e$. Figure 2 shows an example of a cut $(V'_e, V''_e)$ in a coding network. We assume that node $u$ can detect the failure of edge $e$ and notify its immediate descendants, which, in turn, can change their encoding coefficients so that each affected terminal will be able to decode the original packets. Indeed, changing encoding coefficients for the nodes in $V''_e$ can be done with minimum penalty because the information about the failure is attached to the packets to carry the information. In contrast, in order to modify the network code for arcs that originate from nodes in $V''_e$ we need to send special control messages, which will incur additional delay.

Definition 2 (Weakly Robust Network Code): A network code $\mathcal{C}$ is weakly robust to single edge failures if each edge $e \in E$, there exists a feasible code $\mathcal{C}'$ that satisfies the following two conditions:

1) The encoding coefficients of all arcs that originate from nodes in $V''_e$ have the the same encoding coefficients in $\mathcal{C}'$ as in $\mathcal{C}$ (except arcs in $A(E)$)
2) The global encoding coefficients for all arcs in $A(e)$ have zero encoding coefficients
3) Each terminal $t \in T$ can decode the original packets.

D. Necessary Condition

A necessary condition for existence of robust network codes (both weakly and strongly robust) is that for each $e \in E$ a network $G''$ formed from $G$ by removing $e$ must admit an $(s, t)$-flow of value $h$. This condition is equivalent to

$$\min_{\mathcal{C}} \left[ \sum_{e \in E(C)} c(e) - \max_{e \in E(C)} c(e) \right] \geq h,$$

where the minimum is taken over all $(s, t)$-cuts $C(V_1, V_2)$ that separate $s$ and $t$ in $G$, and $E(C)$ is the set of edges that belong to $C$, i.e., the set of edges that connect a node in $V_1$ to a node in $V_2$. In [4] it was shown that this condition is also sufficient for providing instantaneous recovery from edge failures. Moreover, it was shown that the instantaneous recovery can be achieved by using linear network codes. Therefore, we refer to a graph $G(V, E)$ that satisfies this condition as a feasible graph or network.

III. STRONGLY ROBUST CODES FOR NETWORKS WITH UNIFORM CAPACITIES

In this section we assume that all edges of the network have uniform capacity $c$, i.e., each edge can send exactly $c$ packets per time unit. We present an efficient algorithm that can construct a robust network code over a finite field of size $O(k)$. We observe that without loss of generality, we can assume that the capacity of each edge is one unit. Indeed, a feasible network code for unit capacity edges can be extended into the case in which the capacity of each edges is equal to $c$ by combining $c$ communication rounds into a single round. Accordingly, for the rest of this section, we assume that all edges have unit capacity.

In [10] it was shown that communication at rate $h$ with instantaneous recovery from single edge failures is possible if and only if for each edge $e \in E$, it holds that the network $G'(V', E')$ formed from $G(V, E)$ by removing $e$, contains at least $h$ edge-disjoint paths from source $s$ to each terminal node $t \in T$. This implies that a necessary and sufficient condition for the feasibility of network $N$ is the existence of $h+1$ edge-disjoint paths between $s$ and each $t \in T$.

Our approach can be summarized as follows. First, we generate a special parity check packet, referred to as $p_{h+1}$. This packet is a linear combination of the original packets and is constructed as described in Section III-A. Then, we use a standard network coding algorithm due to Jaggi et al. [10] for sending $\mathbb{R} = \{p_1, p_2, \ldots, p_h, p_{h+1}\}$ packets from $s$ to $T$.

The standard algorithm will treat the packets in $\mathbb{R}$ as generated by independent random processes. The algorithm ensures that in the normal network conditions, each destination node receives $h+1$ independent linear combination of the packets in $\mathbb{R}$. The following lemma shows that after a single edge failure, each destination node receives at least $h$ linearly independent combinations of packets in $\mathbb{R}$.

Lemma 3: Upon an edge failure, each terminal $t \in T$ receives at least $h$ linear combinations of packets in $\mathbb{R}$.

Proof: Since we assume that all edges are of unit capacity, each edge in the network can be represented by a single arc. For each arc $a \in A$ we define the global encoding vector

$$\Gamma_e = [\gamma_1^e \ldots \gamma_{h+1}^e]^T \in \mathbb{F}_q^{h+1},$$

that captures the relation between the packet $p_a$ transmitted on arc $a$ and the original packets in $\mathbb{R}$:

$$p_e = \sum_{i=1}^{h+1} p_i \cdot \gamma_i^e.$$
Let \( t \) be a terminal in \( T \). We define the transfer matrix \( M_t \) that captures the relation between the original packets \( R \) and the packets received by the terminal node \( t \in T \) over its incoming edges. The matrix \( M_t \) is defined as follows:
\[
M_t = \left[ \begin{array}{ccc}
\Gamma_{a_1} & \Gamma_{a_2} & \ldots & \Gamma_{a_{h+1}} \\
\end{array} \right],
\]
where \( E_t = \{a_1, \ldots, a_{h+1}\} \) is the set of incoming arcs of \( t \).

Let \( a' \) be a failed arc. A failure of \( a' \) might result in a change of the transfer matrix \( M_t \). We note that the new network matrix \( M_t \) can be written as:
\[
M_t = M_t - \Gamma_{a'} T_{a'},
\]
where \( T_{a'} \) is an \( 1 \times h \) matrix that depends on the location of arc \( a' \) in the network. Note that \( \Gamma_{a'} T_{a'} \) is an \((h+1) \times (h+1)\) matrix of rank no more than one. The subadditivity property \(^1\) of rank implies that the rank of \( M_t \) is at least \( h \).

A. Creating parity check packet

Lemma 3 implies that in the event of any single edge failure, each terminal node receives at least \( h \) independent linear combinations of the packets in \( \{p_1, p_2, \ldots, p_h, p_{h+1}\} \). Since packet \( p_{h+1} \) is a linear combination of \( h \) original packets \( R = \{p_1, p_2, \ldots, p_h\} \), each destination node receives, in fact, \( h \) linear combinations of \( R \). Accordingly, our goal is to construct packet \( p_{h+1} \) in such a way that each destination node receives \( h \) independent linear combinations of \( R \). This will allow each destination node to decode the original packets.

For clarity of presentation we first focus on the case of \( h = 2 \). In this case we have two original packets, \( p_1 \) and \( p_2 \), and one parity check packet \( p_3 = \gamma_1 p_1 + \gamma_2 p_2 \). Suppose that a terminal \( t \in T \) receives two linearly independent combinations of \( p_1 \), \( p_2 \), and \( p_3 \).

There are three different forms of packets a sink may receive, differing by their coefficients for packet \( p_3 \). In the first case, a sink receives two packets which both have zero coefficients for \( p_3 \). In this case, decoding of \( p_1 \) and \( p_2 \) is trivial. If one packet has a non-zero coefficient for \( p_3 \), and the second has a zero coefficient for \( p_3 \), we can express the packets as in Equation 4 below by dividing the first packet by its coefficient for \( p_3 \):
\[
\begin{pmatrix}
\beta_1 & \beta_2 & 1 \\
\beta_3 & \beta_4 & 0
\end{pmatrix}
\begin{pmatrix}
p_1 \\
p_2 \\
p_3
\end{pmatrix},
\]
where \( \beta_1, \ldots, \beta_4 \) are coefficients that belong to \( F_q \). By substituting \( p_3 = \gamma_1 p_1 + \gamma_2 p_2 \) we get:
\[
\begin{pmatrix}
(\beta_1 + \gamma_1) & (\beta_2 + \gamma_2) \\
\beta_3 & \beta_4
\end{pmatrix}
\begin{pmatrix}
p_1 \\
p_2
\end{pmatrix},
\]
\[
(\beta_1 + \gamma_1)(\beta_2 + \gamma_2) = \beta_3 \beta_4.
\]

We wish to find values for \( \gamma_1 \) and \( \gamma_2 \) such that matrix
\[
R = \begin{pmatrix}
(\beta_1 + \gamma_1) & (\beta_2 + \gamma_2) \\
\beta_3 & \beta_4
\end{pmatrix}
\]
is of full rank. In particular, our goal is to select the values of \( \gamma_1 \) and \( \gamma_2 \) in such a way that the determinant
\[
\det(R) = \beta_1 \beta_4 - \beta_2 \beta_3 + \gamma_1 \beta_4 - \gamma_2 \beta_3
\]
of \( R \) is not equal to zero, for any possible choice of \( \beta_1, \ldots, \beta_4 \). It is easy to verify that this cannot be done if \( \gamma_1 \) and \( \gamma_2 \) belong to the same field, \( F_q \), as \( \beta_1, \ldots, \beta_4 \).

\(^1\)The subadditivity property of rank implies that \( \text{Rank}(X + Y) \leq \text{Rank}(X) + \text{Rank}(Y) \).

Accordingly, in our approach we select \( \gamma_1 \) and \( \gamma_2 \) in an extension field of \( F_q \) and construct the parity check packet \( p_3 \) using an MRD code [19].

In an MRD code, one uses a vector of elements in \( GF(q) \) to create an element in an extension field. A vector of \( N \) elements from \( GF(q) \) can be treated as an element in \( GF(q^N) \) where \( GF(q^N) \) is an extension field of \( GF(q) \). A \((n, m)\) MRD code over \( GF(q^N) \) takes \( m \) information symbols and generates \( n \) encoded symbols. It is capable of correcting \( n - m \) rank erasures, or otherwise stated, can recover \( m \) information packets from any \( m \) linearly independent combinations of the transmitted packets [19]. Since we are interested in recovery of the packets of \( p_1 \) and \( p_2 \) from any two linearly independent packets originating from \( p_1 \), \( p_2 \), and \( p_3 \), a \((3, 2)\) MRD code is sufficient. Such a code can be constructed using the following parity check matrix [20]:
\[
H = \begin{pmatrix}
\alpha & \alpha^2 & 1
\end{pmatrix},
\]
where \( \alpha \) is the primitive element of \( GF(q^3) \). Using \( H \), we can set \( p_3 = -\alpha p_1 - \alpha^2 p_2 \). Note that this way we are forced to work in the extension field \( GF(q^3) \) of \( F_q = GF(q) \). This implies that the original packets \( p_1 \), \( p_2 \), \( p_3 \) must belong to \( GF(q^3) \) as well. This, however, can be achieved by combining three communication rounds into a single round and treating a vector of size three in \( GF(q) \) as a single element of \( GF(q^3) \).

If we re-examine \( R \), we see that elements from \( GF(q) \) are mixed with elements from \( GF(q^3) \). Since \( GF(q^3) \) is constructed as an extension field of \( GF(q) \), then \( \beta_1, \ldots, \beta_4 \) can be treated as belonging to \( GF(q^3) \) which behave as elements from \( GF(q) \). This implies that \( \beta_1 \beta_4 - \beta_2 \beta_3 \) belong to \( GF(q) \), and \( \alpha \beta_4 - \alpha^2 \beta_3 \) might also be in \( GF(q) \) if \( \alpha - \alpha^2 \in GF(q) \). However, this is not the case since \( \begin{pmatrix} 1 & \alpha & \alpha^2 \end{pmatrix} \) is a valid MRD code [19]. Therefore,
\[
\det(R) = \beta_1 \beta_4 - \beta_2 \beta_3 + \alpha \beta_4 - \alpha^2 \beta_3
\]
is equal to zero only if \( \beta_3 = \beta_4 = 0 \), which contradicts our assumption that the destination node receives two linearly independent vectors.

Lastly, we consider the case where both received vectors contain non-zero coefficients of \( p_3 \), as in Equation 6. Through Gaussian elimination, two vectors can be subtracted to construct vectors as in Equation 4, hence by using the argument described above we can show that packets \( p_1 \) and \( p_2 \) can be recovered as well.
\[
\begin{pmatrix}
\beta_1 & \beta_2 & 1 \\
\beta_3 & \beta_4 & 1
\end{pmatrix}
\begin{pmatrix}
p_1 \\
p_2 \\
p_3
\end{pmatrix},
\]

B. General case

We turn to consider a more general case of \( h > 2 \). Since we need to only recover from a single failure, we need to find a \((h+1, h)\) MRD code. For this case, we can use the parity check matrix [19]:
\[
H = \begin{pmatrix}
\alpha & \alpha^2 & \ldots & \alpha^h & 1
\end{pmatrix}
\]
over the field \( GF(q^{h+1}) \), with
\[
p_{h+1} = - \sum_{i=1}^{h} \alpha^i p_i.
\]
Theorem 4: The proposed scheme achieves an instantaneous recovery from any single edge failure.

Proof: Follows directly from the properties of MRD codes [19].

IV. NETWORK WITH NON-UNIFORM CAPACITIES

In this section, we assume that different network edges have different capacities. We focus on a special case in which only two packets need to be delivered from the source to all terminals at each communication round. The design of robust network codes for $h = 2$ in the context of unicast connections has been studied in [21]. In this work, we build on the results of [21] for constructing a robust network code for multicast connections.

Let $G(V, E)$ be a minimum robust coding network, i.e., a feasible robust network such that the removal of an edge or a reduction in the capacity of an edge results in a violation of its feasibility. Note that the capacity of any edge $e \in E$ is at most two. For each terminal $t \in T$ let $G_t(V_t, E_t)$ to be a subgraph of $G(V, E)$ that contains a minimum coding network with respect to terminal $t$. That is, $G_t(V_t, E_t)$ only contains edges of $G(V, E)$ that are necessary to guarantee the conditions defined by Equation 1 for terminal $t$. Furthermore, any reduction of the capacity of edges in $G_t(V_t, E_t)$ will result in a violation of this condition for at least one of the $(s, t)$ cuts.

For each $t \in T$ we denote by $A_t$ the set of arcs that correspond to edges in $E_t$. In [21] it was shown that it is possible to construct a robust unicast network code over $GF(2)$. In this code, each arc is transmitted either one of the original packets $p_1$ and $p_2$ or their sum ($p_1 + p_2$). For each $t \in T$ we denote such network code by $\mathcal{C}_t$. We use $\mathcal{C}_t$ to divide $A_t$ into three disjoint subsets $A_{t1}$, $A_{t2}$, and $A_{t3}$, where all the arcs in $A_{t1}$ carry the same packet.

To extend this to multicast, we construct a set of $f = 3^k$ linearly independent vectors $\Phi = \{\Phi_1, \Phi_2, \ldots, \Phi_f\}$ of the original two packets over $GF(q)$.

We divide the arcs in $A_t$ into $f$ different subsets $A_{t1}, \ldots, A_{tf}$ such that for each subset $A_t$ it holds that all arcs in $A_t$ belong to the same subsets $A_{t1}$ for all $t$ for which the arc belongs to $A_t$. That is, for any two arcs $a_1$ and $a_2$ in $A_t$ and for each terminal $t$ such that $a_1, a_2 \in A_t$ it holds that both $a_1$ and $a_2$ belong to the same subset, say $A_{t1}$, of $A_t$. Then each subset $A_{tf}$ is associated with a linearly independent vector $\Phi_f$ from $GF(q)$. We construct a code for which it holds that the global encoding coefficient of each arc in $A_t$ is equal to $\Phi_t$. It is easy to verify that such a network code is feasible, i.e., it is possible to select the set of local encoding coefficients that satisfy this property. The proof is based on the fact that for $h = 2$ any two linearly independent packets are sufficient for constructing any linear combination in $\Phi$. Also, it is easy to verify that the code has a weakly robust property. Due to space limitations, the proof is omitted.

V. CONCLUSION

In this paper, we considered the problem of establishing reliable multicast connections across a communication network. For the case of uniform edge capacities we presented an efficient network coding algorithm based on the MRD codes that requires a small finite field $O(k)$. For the case of non-uniform capacities, we focused on a special case of $h = 2$ and showed that it is also possible to construct a robust code over a small field. Future research includes the construction of network codes with non-uniform capacities and transmission of more than two packets per communication round.

REFERENCES