Linear Network Coding for Sum All-Reduce over Ring Networks

Zhuoqi Tu, Yi Chen, and Shenghao Yang

***Abstract*—THIS PAPER IS ELIGIBLE FOR THE STUDENT PAPER AWARD. All-Reduce is a fundamental collective communication operation in distributed computing, where each node has a local message and all nodes require a common function of all the messages. A particularly important function for All-Reduce is the sum (or average) of all the messages, which is widely used in the training of large language models (LLMs). Existing All-Reduce algorithms for the sum function are implemented using the reduce-multicast approach, typically over a ring network topology. In this paper, we study the All-Reduce problem over ring networks using a linear network coding approach, which includes the reduce-multicast approach as a special case. We provide necessary and sufficient conditions for feasible linear network codes and apply our general results to the All-Reduce problem over a 3-node ring network to derive a tight upper bound for the achievable rate of linear network coding. We also construct feasible linear network codes to achieve the upper bound for the 3-node ring network, and find that it is necessary to use non-trivial network codes to achieve the rate upper bound for some cases. Our results lay the foundation for optimizing distributed computation in networks with repetitive topologies.**

I. Introduction

The rapid advancement of AI has driven the development of large-scale computing clusters with tens of thousands of nodes, such as the GPU clusters used for Large Language Model (LLM) training [1]. Efficient distributed computation within these clusters relies on *collective communication* [2], a paradigm comprising operations such as *Reduce-Scatter*, *All-Gather*, and *All-Reduce*. In particular, the All-Reduce for sum (or average) is fundamental to distributed LLM training, where it synchronizes local gradient vectors across the network. To accomplish this, All-Reduce algorithms—such as those implemented by NCCL [2]—aggregate gradients from all nodes (the *reduce* phase) and subsequently distribute the global sum or average back to every node (the *multicast* phase).

The scalability of AI clusters is fundamentally constrained by the high overhead of collective communication. Studies have reported that for training deep neural networks, communication time can be nearly $6\times$ greater than computation time on 1024 GPUs and $12\times$ on 2048 GPUs [3]. To mitigate this bottleneck, substantial research has focused on developing more efficient All-Reduce algorithms (see [4]–[6] and the references therein). All-Reduce is typically organized using communication topologies such as symmetric rings or trees. The optimality of ring-based All-Reduce algorithms has been discussed under the reduce-multicast framework [7].

The reduce-multicast approach underlying All-Reduce algorithms can be generalized to arbitrary network topologies using techniques from linear network coding [8], [9] and its dual framework [10], providing higher bandwidth utilization than ring or tree topologies alone. Recent work [11] has discussed how to adapt the network codes developed over finite fields to computations involving floating-point numbers, to reduce numerical errors. However, a key question remains: is the reduce-multicast paradigm optimal for All-Reduce?

The fundamental problem underlying All-Reduce has been studied in the context of network coding as an extended multicast problem [12], [13]. This problem, called *multi-sum*, is to compute the sum of multiple source messages—each generated by a source node—at all destination nodes. The aforementioned works [12], [13] analyze the computation of sums over finite fields in directed acyclic networks. An equivalence relation has been established between this problem and the multiple unicast problem [12], the latter of which is difficult to solve in general [14]. In [13], the authors investigated whether the sum can be computed in networks with a small number of sources and destinations. For the case of only two sources or two destinations, computation is possible if and only if every source-destination pair is connected by at least one path. However, for three sources and three destinations, two edge-disjoint paths per pair are required for sum computation; one path does not suffice.

In this paper, we focus on the All-Reduce problem in a ring network topology, which is commonly employed in practice. The task is to compute the sum of all message vectors held at the nodes, where message vectors take values from a general field—either finite or infinite. We assume that each communication link in the network is ideal: it can transmit one symbol per time unit without error. We study linear network coding for this setting, which includes the classic reduce-multicast method as a special case. Furthermore, since our formulation naturally extends to real-valued computations, it offers theoretical insight and practical guidelines for implementing All-Reduce with floating-point arithmetic.

Our first contribution is a necessary and sufficient condition for the feasibility of a linear network code for the All-Reduce problem over a ring network. For traditional multicast problems over networks with cycles, convolutional network codes have been studied [15]. Here we extend the idea to study All-Reduce. To facilitate the analysis, we study ordinary (non-convolutional) network codes on the directed acyclic time-expanded network induced by the ring network. The All-Reduce problem over the original ring network using

Z. Tu, Y. Chen and S. Yang are with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China.

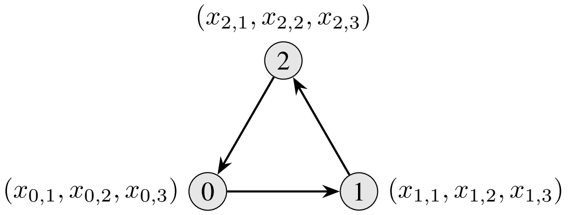

$(x_{2,1}, x_{2,2}, x_{2,3})$
2
$(x_{0,1}, x_{0,2}, x_{0,3})$
0
1
$(x_{1,1}, x_{1,2}, x_{1,3})$

Fig. 1: The 3-node ring network with nodes labeled as $0, 1, 2$. Each node holds a message vector of $K = 3$ symbols.

T time units is equivalent to the multi-sum problem over the network expanded for T time units. By benefiting from the repetitive structure of the time-expanded network, we obtain the necessary and sufficient condition for a feasible linear network code for any expansion parameter T. This condition enables us to efficiently verify the feasibility of a linear network code, and also provides insights about how to construct a feasible linear network code.

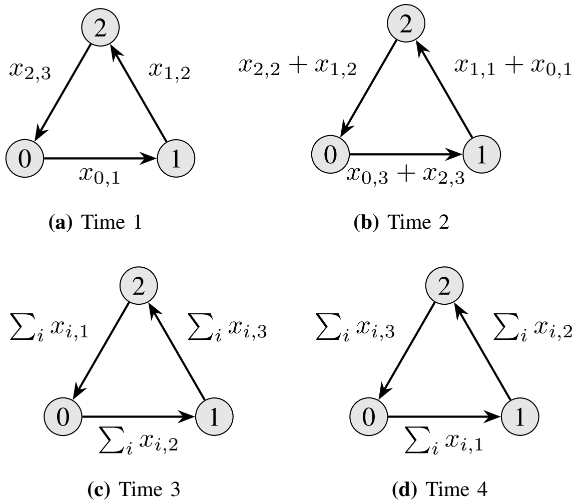
Based on the general analysis, we further study All-Reduce for the 3-node ring network. We prove that a feasible linear network code exists for the K-message multi-sum problem using T time units if and only if $T \geq \frac{4}{3}K$, which implies that the achievable rate is at most $\frac{3}{4}$. As the classic ring All-Reduce algorithm achieves the rate $\frac{3}{4}$ when $K = 3$ and $T = 4$, linear network coding beyond the reduce-multicast approach does not improve the rate. However, we find that it is necessary to use non-trivial network coding achieve the upper bound for cases with $K = 3k + 2$ and $T = 4k + 3$, where $k = 0, 1, \ldots$.

The remainder of this paper is organized as follows. In Section II, we introduce the ring network model and the All-Reduce problem. In Section III, we introduce the time-expanded network and the linear network coding framework, and prove the necessary and sufficient condition for the feasibility of a linear network code. In Section IV, we further study All-Reduce for the 3-node ring network. In Section V, we discuss the future work and the extension to other topologies.

II. All-Reduce over a Ring Network

We consider a directed N-node ring network, modeled as a directed graph $G = (\mathcal{V}, \mathcal{E})$, where the vertex set is $\mathcal{V} = \{0, 1, \ldots, N-1\}$ for some integer $N \geq 2$ and the edge set is $\mathcal{E} = \{(i, (i+1) \bmod N) : i \in \mathcal{V}\}$. Fig. 1 illustrates the 3-node ring network. Denote by $\mathbb{F}$ a field, which can be either finite or infinite. We suppose that each edge (i, j) can transmit one symbol in $\mathbb{F}$ per unit time from node i to node j without error. Let K be a positive integer. Each node $i \in \mathcal{V}$ holds a message vector $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,K}] \in \mathbb{F}^K$, a vector of K symbols over $\mathbb{F}$. The goal of the *K-message All-Reduce problem* is for every node to obtain the sum of all message vectors $\sum_{i \in \mathcal{V}} \mathbf{x}_i$.

The *classic ring All-Reduce algorithm* consists of an *reduction* phase and a *multicast* phase. Consider the All-Reduce task with $K = 1$. The reduction phase includes $N - 1$ time units: For the t-th time unit, where $1 \leq t < N$, node $t - 1$ sends the sum $\sum_{i=0}^{t-1} x_{i,1}$ to node t. By the end of time $N - 1$, node N holds the sum $\sum_{i=0}^{N-1} x_{i,1}$. In the multicast phase, node N sends the sum to node 0, which then sends it to node 1, and so on. Therefore, it takes $N - 1$ time units for the multicast phase to distribute the sum to all nodes. As one K-message sum is calculated in $2(N-1)$ time units, the rate is $\frac{1}{2(N-1)}$.



(a) Time 1 **(b)** Time 2 **(c)** Time 3 **(d)** Time 4

Fig. 2: The ring All-Reduce algorithm for a 3-node ring network with $K = 3$. Three sums are calculated using 4 time units.

To improve throughput, the ring All-Reduce algorithm adopts a time-sharing strategy for solving the All-Reduce task with $K = N$, allowing N parallel summation tasks to run concurrently, each starting at a different node. We illustrate the ring All-Reduce algorithm for a 3-node ring network in Fig. 2. These two phases are also called *reduce-scatter* and *all-gather*, respectively. Therefore, the ring All-Reduce algorithm can achieve a rate of $\frac{N}{2(N-1)}$. Notably, in the classic ring All-Reduce algorithm, additions are performed only among messages with the same index.

It is possible to extend the discussion of All-Reduce to general network topologies. The reduce-multicast approach of the classic ring All-Reduce algorithm can be extended to arbitrary network topologies using linear network coding [8], [9]. Note that multicast and sum reduce are dual problems and can use the same network coding coefficients [10]. The time-sharing strategy also remains applicable for better utilization of network bandwidth. However, the fundamental question remains: can this reduce-multicast strategy be further improved? In the remainder of this paper, we discuss a more general form of linear network coding solutions for All-Reduce over a ring network, going beyond the reduce-multicast approach.

III. Linear Network Coding for All-Reduce

In this section, we formulate a linear network coding approach for All-Reduce over the N-node ring network $G = (\mathcal{V}, \mathcal{E})$. Since G forms a cycle, one could generalize a convolutional network code as described in [15]. To facilitate the discussion of All-Reduce, we adopt an alternative approach by considering an ordinary (non-convolutional) network code applied to the time-expanded network derived from G.

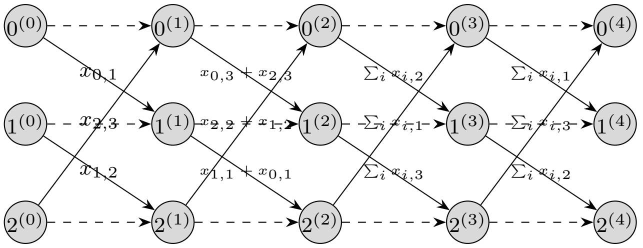


Fig. 3: Example of the time-expanded network for a 3-node ring network with $T = 4$

A. Time-Expanded Network

Definition 1 (Time-Expanded Network)**.** Given a network $G = (\mathcal{V}, \mathcal{E})$ and an integer $T \geq 1$, the time-expanded network $G_T = (\mathcal{V}_T, \mathcal{E}_T)$ is constructed as follows: $\mathcal{V}_T = \bigcup_{t=0}^{T} \mathcal{V}^{(t)}$, where for each $t \in \{0, 1, \ldots, T\}$, $\mathcal{V}^{(t)} = \{i^{(t)} : i \in \mathcal{V}\}$. The edge set $\mathcal{E}_T$ consists of two types of edges:

- **Transmission Edges:** For every edge $(i, j) \in \mathcal{E}$ and for all $t \in \{0, \ldots, T-1\}$, the edge $(i^{(t)}, j^{(t+1)})$ is included in $\mathcal{E}_T$ with the same capacity as in G.
- **Storage Edges:** For every node $i \in \mathcal{V}$ and for all $t \in \{0, \ldots, T-1\}$, the edge $(i^{(t)}, i^{(t+1)})$ is included in $\mathcal{E}_T$ with infinite capacity.

For convenience, we write $G_0 = G$.

The network G_T models the propagation of data through G over T time units, where the storage edges enable previously received symbols at a node to be carried forward to future time units. The All-Reduce problem over G thus corresponds to the following multi-sum problem over G_T:

- Nodes in $\mathcal{V}^{(0)}$ are called source nodes, each holding a message vector $\mathbf{x}_i \in \mathbb{F}^K$, and
- Nodes in $\mathcal{V}^{(T)}$ are called destination nodes, each of which must recover the sum $\sum \mathbf{x}_i$.

An All-Reduce algorithm over G using T time units can be converted to an algorithm for the corresponding multi-sum problem over G_T, and vice versa. For example, the ring All-Reduce algorithm illustrated in Fig. 2 can be represented in the time-expanded network shown in Fig. 3 as a linear network code with linear combination coefficients either 0 or 1. Henceforth, we focus on linear network coding for the multi-sum problem in G_T with more general linear combination coefficients.

B. Linear Network Coding formulation

A network code consists of the encoding of source message vectors (treated as column vectors here), the intermediate recoding of received symbols, and the final decoding at the destination nodes.

Denote by $y_{i\rightarrow}^{(t)}$ and $y_{i\leftarrow}^{(t)}$ the symbols transmitted and received by node i at time t, respectively, from the transmission edges. According to the definition of the time-expanded network, we have

$$y_{i\leftarrow}^{(t)} = y_{i'\rightarrow}^{(t-1)}, \quad i' = (i-1) \bmod N. \tag{1}$$

For $t \in \{1, 2, \ldots, T\}$, let

$$\mathbf{s}_i^{(t)} = \left[y_{i\leftarrow}^{(1)}, \ldots, y_{i\leftarrow}^{(t)} \right]^\top \in \mathbb{F}^{t \times 1} \tag{2}$$

denote the vector of symbols received by node i at time t. Let $\mathbf{s}_i^{(0)} = 0$. Using the storage edges, node $i^{(t)}$ for $t \in \{0, 1, \ldots, T-1\}$ transmits $\mathbf{x}_i$ and $\mathbf{s}_i^{(t)}$ to node $i^{(t+1)}$. At time $t \in \{1, \ldots, T\}$, node $i^{(t)}$ receives from both types of edges: $\mathbf{x}_i$, $\mathbf{s}_i^{(t-1)}$, and $y_{i\leftarrow}^{(t)}$. It then generates $\mathbf{s}_i^{(t)}$ and transmits the symbol $y_{i\rightarrow}^{(t)}$, generated as a linear combination of these received symbols:

$$y_{i\rightarrow}^{(t)} = \mathbf{m}_i^{(t)} \mathbf{x}_i + \boldsymbol{\lambda}_i^{(t)} \mathbf{s}_i^{(t)}, \tag{3}$$

where $\mathbf{m}_i^{(t)} \in \mathbb{F}^{1 \times K}$ and $\boldsymbol{\lambda}_i^{(t)} \in \mathbb{F}^{1 \times t}$ are the encoding and recoding coefficient vectors at time t, respectively. At time $t = 0$, $y_{i\rightarrow}^{(0)} = \mathbf{m}_i^{(0)} \mathbf{x}_i$.

The destination node $i^{(T)}$ receives the symbols $\mathbf{x}_i$ and $\mathbf{s}_i^{(T)}$, and then performs the decoding operation:

$$\hat{\mathbf{z}}_i = \mathbf{R}_i \mathbf{s}_i^{(T)} + \mathbf{T}_i \mathbf{x}_i, \tag{4}$$

where $\mathbf{R}_i \in \mathbb{F}^{K \times T}$ and $\mathbf{T}_i \in \mathbb{F}^{K \times K}$ are the decoding and translation matrices.

Definition 2 (Feasibility)**.** For the All-Reduce problem over the ring network $G = (\mathcal{V}, \mathcal{E})$, a linear network code formulated in (3) and (4) with respect to the time-expanded network G_T with coefficient matrices $\mathbf{m}_i^{(t)}, \boldsymbol{\lambda}_i^{(t)}, \mathbf{R}_i, \mathbf{T}_i$ for $i \in \mathcal{V}$ is *feasible* if for any $i \in \mathcal{V}$ and $\mathbf{x}_i \in \mathbb{F}^K$, $\hat{\mathbf{z}}_i = \sum_{j \in \mathcal{V}} \mathbf{x}_j$.

The rate of a linear network code described above is K/T. A rate R is said to be *achievable* by linear network coding for All-Reduce in a ring network if for any $\epsilon > 0$, there exists a feasible linear network code with rate at least $R - \epsilon$.

The reduce-multicast All-Reduce algorithms yield linear network coding solutions with the following characteristics:

- The entries of all coefficient matrices are binary.
- Each vector $\mathbf{m}_i^{(t)}$ contains at most one entry equal to 1, i.e., at most one message symbol is selected for encoding in (3). Similarly, $\boldsymbol{\lambda}_i^{(t)}$ contains at most one 1. If both vectors select a symbol, the received symbol selected by $\boldsymbol{\lambda}_i^{(t)}$ must correspond to the same message index as the symbol selected by $\mathbf{m}_i^{(t)}$.
- The matrix $\mathbf{T}_i$ contains at most one entry equal to 1 per row. Likewise, $\mathbf{R}_i$ contains at most one 1 per row. For any given row, if $\mathbf{T}_i$ selects a message symbol, the corresponding row in $\mathbf{R}_i$ must select a received symbol with the same message index.

C. Necessary and Sufficient Conditions for Feasibility

In general, if a linear network code can solve the multi-sum problem in G_T, it can solve the multicast problem from any source node to all the destination nodes. Thus, there must be at least K edge-disjoint paths from any source node to any destination node. By counting the number of edge-disjoint paths in G_T, we obtain the following necessary condition for the existence of a feasible linear network code.

Proposition 1. *For the time-expanded graph G_T of the N-node ring network G, a feasible linear network code to the K-message multi-sum problem exists only if $T \geq K + N - 2$.*

The proposition implies that the rate of All-Reduce satisfies $\frac{K}{T} \leq \frac{K}{K+N-2} < 1$. When $K = N$, the rate is upper-bounded by $\frac{N}{2(N-1)}$, and this bound is achieved by the ring All-Reduce algorithm.

Define the encoding and recoding matrices for node $i \in \mathcal{V}$ across time as follows

$$\mathbf{M}_i = \begin{bmatrix} \mathbf{m}_i^{(0)} \\ \mathbf{m}_i^{(1)} \\ \vdots \\ \mathbf{m}_i^{(T-1)} \end{bmatrix} \in \mathbb{F}^{T \times K}, \ \mathbf{\Lambda}_i = \begin{bmatrix} 0 & \mathbf{0} \\ \boldsymbol{\lambda}_i^{(1)} & \mathbf{0} \\ \vdots & \vdots \\ \boldsymbol{\lambda}_i^{(T-1)} & 0 \end{bmatrix} \in \mathbb{F}^{T \times T}.$$

We note that $\mathbf{\Lambda}_i$ is a strictly lower triangular matrix. Given $i \in \mathcal{V}$, define the backward index shift along the ring as

$$i_{\leftarrow j} = (i - j) \bmod N, \quad j = 1, \dots, N,$$

with $i_{\leftarrow N} = i$. Let $i' = i_{\leftarrow 1}$ denote the predecessor of i. Let $\mathbf{\Lambda}_{i',0} = \mathbf{I}_t$ (the $t \times t$ identity). For $j > 0$, define the recursion

$$\mathbf{\Lambda}_{i',j} = \mathbf{\Lambda}_{i',j-1} \mathbf{\Lambda}_{i_{\leftarrow j}},$$

which represents the chained product of recoding matrices from node i' through node $i_{\leftarrow j}$.

Theorem 2. *For the time-expanded graph G_T of the N-node ring network $G = (\mathcal{V}, \mathcal{E})$, a linear network code for the K-message multi-sum problem is feasible if and only if the following conditions hold:*

$$\begin{aligned} \mathbf{R}_i \left(\mathbf{I}_T - \mathbf{\Lambda}_{i',N}\right)^{-1} \mathbf{\Lambda}_{i',j-1} \mathbf{M}_{i_{\leftarrow j}} = \mathbf{I}_K, \\ \forall i \in \mathcal{V}, j \in \{1, \dots, N-1\}, \qquad (5) \\ \mathbf{R}_i \left(\mathbf{I}_T - \mathbf{\Lambda}_{i',N}\right)^{-1} \mathbf{\Lambda}_{i',N-1} \mathbf{M}_i + \mathbf{T}_i = \mathbf{I}_K, \forall i \in \mathcal{V}. \quad (6) \end{aligned}$$

Remark 1. The essential condition is to verify (5), which involves only $\mathbf{R}_i$, $\mathbf{\Lambda}_i$, and $\mathbf{M}_i$. Once feasible matrices $\mathbf{R}_i$, $\mathbf{\Lambda}_i$, and $\mathbf{M}_i$ are found, the translation matrices $\mathbf{T}_i$ can be determined by solving (6).

Proof. By (1), (2) and (3), we have

$$\mathbf{s}_i^{(T)} = \mathbf{M}_{i'} \mathbf{x}_{i'} + \mathbf{\Lambda}_{i'} \mathbf{s}_{i'}^{(T)}. \qquad (7)$$

By expanding (7) iteratively for $i = 0$, we have

$$\begin{aligned} \mathbf{s}_0^{(T)} &= \mathbf{M}_{N-1}\mathbf{x}_{N-1} + \mathbf{\Lambda}_{N-1}\left(\mathbf{M}_{N-2}\mathbf{x}_{N-2} + \mathbf{\Lambda}_{N-2}\mathbf{s}_{N-2}^{(T)}\right) \\ &= \mathbf{M}_{N-1}\mathbf{x}_{N-1} + \mathbf{\Lambda}_{N-1}\mathbf{M}_{N-2}\mathbf{x}_{N-2} + \\ &\quad \mathbf{\Lambda}_{N-1}\mathbf{\Lambda}_{N-2}\mathbf{M}_{N-3}\mathbf{x}_{N-3} + \cdots + \\ &\quad \mathbf{\Lambda}_{N-1}\mathbf{\Lambda}_{N-2}\cdots\mathbf{\Lambda}_1\mathbf{M}_0\mathbf{x}_0 + \\ &\quad \mathbf{\Lambda}_{N-1}\mathbf{\Lambda}_{N-2}\cdots\mathbf{\Lambda}_1\mathbf{\Lambda}_0\mathbf{s}_0^{(T)}. \end{aligned}$$

A similar formula can be derived for any $i \in \mathcal{V}$. For every $i \in \mathcal{V}$, we have the following general formula:

$$\mathbf{s}_i^{(T)} = \left(\mathbf{I}_T - \mathbf{\Lambda}_{i',N}\right)^{-1} \sum_{j=1}^{N} \mathbf{\Lambda}_{i',j-1} \mathbf{M}_{i_{\leftarrow j}} \mathbf{x}_{i_{\leftarrow j}},$$

where the inverse $\left(\mathbf{I}_T - \mathbf{\Lambda}_{i',N}\right)^{-1}$ exists because $\mathbf{\Lambda}_{i',N}$ is a product of strictly lower triangular matrices and is therefore itself strictly lower triangular. Consequently, $\mathbf{I}_T - \mathbf{\Lambda}_{i',N}$ is a lower triangular matrix with 1's on the diagonal, which is guaranteed to be invertible.

For node $i^{(T)}$ to successfully recover the global sum, the decoded output must satisfy

$$\begin{aligned} \hat{\mathbf{z}}_i =& \mathbf{R}_i \mathbf{s}_i^{(T)} + \mathbf{T}_i \mathbf{x}_i \\ =& \mathbf{R}_i \left(\mathbf{I}_T - \mathbf{\Lambda}_{i',N}\right)^{-1} \sum_{j=1}^{N} \mathbf{\Lambda}_{i',j-1} \mathbf{M}_{i_{\leftarrow j}} \mathbf{x}_{i_{\leftarrow j}} + \mathbf{T}_i \mathbf{x}_i \\ =& \sum_{j=0}^{N-1} \mathbf{x}_j. \end{aligned}$$

To ensure this equality holds universally for arbitrary input vectors $\mathbf{x}_j$, the effective coefficient matrix for each distinct source vector must be the identity matrix $\mathbf{I}_K$. The proof is completed by noting that $\mathbf{x}_{i_{\leftarrow N}} = \mathbf{x}_i$. □

IV. All-Reduce over a Three-Node Ring Network

In this section, we apply Theorem 2 to the 3-node ring network to study the achievable rates of the All-Reduce problem by linear network coding.

When $N = 3$, from Theorem 2, we can find a feasible linear network code for the K-message multi-sum problem if and only if there exist matrices $\mathbf{R}_i$, $\mathbf{\Lambda}_i$, and $\mathbf{M}_i$ for $i = 0, 1, 2$ satisfying the following equations:

$$\mathbf{R}_0 \left(\mathbf{I}_T - \mathbf{\Lambda}_2\mathbf{\Lambda}_1\mathbf{\Lambda}_0\right)^{-1} \quad \mathbf{M}_2 = \mathbf{I}_K \qquad (8)$$

$$\mathbf{R}_0 \left(\mathbf{I}_T - \mathbf{\Lambda}_2\mathbf{\Lambda}_1\mathbf{\Lambda}_0\right)^{-1} \mathbf{\Lambda}_2 \mathbf{M}_1 = \mathbf{I}_K \qquad (9)$$

$$\mathbf{R}_1 \left(\mathbf{I}_T - \mathbf{\Lambda}_0\mathbf{\Lambda}_2\mathbf{\Lambda}_1\right)^{-1} \quad \mathbf{M}_0 = \mathbf{I}_K \qquad (10)$$

$$\mathbf{R}_1 \left(\mathbf{I}_T - \mathbf{\Lambda}_0\mathbf{\Lambda}_2\mathbf{\Lambda}_1\right)^{-1} \mathbf{\Lambda}_0 \mathbf{M}_2 = \mathbf{I}_K \qquad (11)$$

$$\mathbf{R}_2 \left(\mathbf{I}_T - \mathbf{\Lambda}_1\mathbf{\Lambda}_0\mathbf{\Lambda}_2\right)^{-1} \quad \mathbf{M}_1 = \mathbf{I}_K \qquad (12)$$

$$\mathbf{R}_2 \left(\mathbf{I}_T - \mathbf{\Lambda}_1\mathbf{\Lambda}_0\mathbf{\Lambda}_2\right)^{-1} \mathbf{\Lambda}_1 \mathbf{M}_0 = \mathbf{I}_K. \qquad (13)$$

Based on these equations, we can derive the following theorem for the existence of a feasible linear network code.

Theorem 3. *For the time-expanded graph G_T of the 3-node ring network G, a feasible linear network code for the K-message multi-sum problem exists if and only if $T \geq \frac{4K}{3}$.*

The theorem implies that for the All-Reduce problem over a 3-node ring network, using linear network coding can achieve the rate at most $\frac{3}{4}$. As the classic ring All-Reduce algorithm can achieve the rate $\frac{3}{4}$ when $K = 3$ and $T = 4$, linear network coding does not improve the rate. However, it provides more flexibility to construct a feasible solution for more general values of K and T. As we illustrated in the proof, it may be necessary to use non-trivial network coding for some cases.

A. Proof of Necessity

The equations (8) and (9) for node 0 imply that

$$\mathbf{R}_0 \left(\mathbf{I}_T - \mathbf{\Lambda}_2\mathbf{\Lambda}_1\mathbf{\Lambda}_0\right)^{-1} \left(\mathbf{M}_2 - \mathbf{\Lambda}_2\mathbf{M}_1\right) = \mathbf{0}.$$

Let $\mathbf{\Phi}_0 = \mathbf{M}_2 - \mathbf{\Lambda}_2\mathbf{M}_1$. Since $\mathbf{R}_0\left(\mathbf{I}_T - \mathbf{\Lambda}_2\mathbf{\Lambda}_1\mathbf{\Lambda}_0\right)^{-1}$ is an $K \times T$ matrix with rank K (necessary to satisfy (8)), its null space has dimension $T - K$. As the columns of $\mathbf{\Phi}_0$ must lie in this null space, we have $\operatorname{rank}(\mathbf{\Phi}_0) \leq T - K$.

Apply the same argument to node 1 and 2. Let $\mathbf{\Phi}_1 = \mathbf{M}_0 - \mathbf{\Lambda}_0\mathbf{M}_2$ and $\mathbf{\Phi}_2 = \mathbf{M}_1 - \mathbf{\Lambda}_1\mathbf{M}_0$. We have $\operatorname{rank}(\mathbf{\Phi}_1) \leq T - K$ and $\operatorname{rank}(\mathbf{\Phi}_2) \leq T - K$.

We can express $\mathbf{M}_i$ as follows: $\mathbf{M}_0 = \mathbf{\Lambda}_0\mathbf{M}_2 + \mathbf{\Phi}_1$, $\mathbf{M}_1 = \mathbf{\Lambda}_1\mathbf{M}_0 + \mathbf{\Phi}_2$ and $\mathbf{M}_2 = \mathbf{\Lambda}_2\mathbf{M}_1 + \mathbf{\Phi}_0$. Substituting the expressions for $\mathbf{M}_1$ and $\mathbf{M}_2$ into the equation for $\mathbf{M}_0$ yields

$$\begin{aligned}\mathbf{M}_0 &= \mathbf{\Lambda}_0\mathbf{M}_2 + \mathbf{\Phi}_1\\ &= \mathbf{\Lambda}_0(\mathbf{\Lambda}_2\mathbf{M}_1 + \mathbf{\Phi}_0) + \mathbf{\Phi}_1\\ &= \mathbf{\Lambda}_0(\mathbf{\Lambda}_2(\mathbf{\Lambda}_1\mathbf{M}_0 + \mathbf{\Phi}_2) + \mathbf{\Phi}_0) + \mathbf{\Phi}_1\\ &= \mathbf{\Lambda}_0\mathbf{\Lambda}_2\mathbf{\Lambda}_1\mathbf{M}_0 + \mathbf{\Lambda}_0\mathbf{\Lambda}_2\mathbf{\Phi}_2 + \mathbf{\Lambda}_0\mathbf{\Phi}_0 + \mathbf{\Phi}_1.\end{aligned}$$

Since each $\mathbf{\Lambda}_i$ is strictly lower triangular, their product is also strictly lower triangular. Therefore, $\mathbf{I}_T - \mathbf{\Lambda}_0\mathbf{\Lambda}_2\mathbf{\Lambda}_1$ is a lower triangular matrix with 1s on the diagonal, making it invertible. Rearranging gives

$$\mathbf{M}_0 = (\mathbf{I}_T - \mathbf{\Lambda}_0\mathbf{\Lambda}_2\mathbf{\Lambda}_1)^{-1}(\mathbf{\Lambda}_0\mathbf{\Lambda}_2\mathbf{\Phi}_2 + \mathbf{\Lambda}_0\mathbf{\Phi}_0 + \mathbf{\Phi}_1).$$

For a feasible solution, $\mathbf{M}_0$ must have full rank K. The rank of the right-hand side is at most the sum of the ranks of its terms (noting that multiplication by full-rank or triangular matrices does not increase rank):

$$K = \operatorname{rank}(\mathbf{M}_0) \leq \sum_{i=0}^{2} \operatorname{rank}(\mathbf{\Phi}_i) \leq 3(T - K),$$

which implies $T \geq \frac{4}{3}K$.

B. Proof of Sufficiency

We establish sufficiency through an explicit construction. For any K and T satisfying $T \geq \frac{4}{3}K$, i.e., $T \geq \lceil \frac{4}{3}K \rceil$, we can construct matrices $\mathbf{R}_i$, $\mathbf{M}_i$ and $\mathbf{\Lambda}_i$ to satisfy the six equations (8) to (13).

It suffices to demonstrate existence for the cases $(K, T) \in \{(1,2), (2,3), (3,4)\}$. For any $K \geq 4$, let $T = \lceil \frac{4K}{3} \rceil$. We partition the K symbols into $\lfloor \frac{K}{3} \rfloor$ groups of 3 symbols, each consuming 4 time units. The remaining $K \bmod 3$ symbols form a residual group allocated to the remaining $\lceil \frac{4K}{3} \rceil - 4\lfloor \frac{K}{3} \rfloor$ time units. Since the residual parameters correspond to $(1, 2)$ or $(2, 3)$, the construction is complete. For $T > \lceil \frac{4K}{3} \rceil$, the result holds by leaving the excess time units unused.

1) $K = 3$ and $T = 4$: A feasible network code is given as follows:

$$\mathbf{M}_0 = \begin{bmatrix} 1 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 0\\ 0 & 0 & 0 \end{bmatrix}, \mathbf{M}_1 = \begin{bmatrix} 0 & 1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0 \end{bmatrix}, \mathbf{M}_2 = \begin{bmatrix} 0 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{\Lambda_0} = \mathbf{\Lambda_1} = \mathbf{\Lambda_2} = \begin{bmatrix} 0 & 0 & 0 & 0\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{R}_0 = \begin{bmatrix} 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{R}_1 = \begin{bmatrix} 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{R}_2 = \begin{bmatrix} 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

This corresponds to the classic ring All-Reduce algorithm.

2) $K = 2$ and $T = 3$: A feasible network code is given as follows:

$$\mathbf{M}_0 = \begin{bmatrix} 0 & 1\\ 1 & 0\\ 1 & 1 \end{bmatrix}, \quad \mathbf{M}_1 = \begin{bmatrix} 1 & 1\\ 0 & 1\\ 1 & 0 \end{bmatrix}, \quad \mathbf{M}_2 = \begin{bmatrix} 1 & 0\\ 1 & 1\\ 0 & 1 \end{bmatrix},$$

$$\mathbf{\Lambda_0} = \mathbf{\Lambda_1} = \mathbf{\Lambda_2} = \begin{bmatrix} 0 & 0 & 0\\ 1 & 0 & 0\\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{R}_0 = \begin{bmatrix} 0 & 1 & -1\\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{R}_1 = \begin{bmatrix} 0 & 1 & 0\\ 0 & -1 & 1 \end{bmatrix}, \quad \mathbf{R}_2 = \begin{bmatrix} 0 & 0 & 1\\ 0 & 1 & 0 \end{bmatrix}.$$

3) $K = 1$ and $T = 2$: A feasible network code is given as follows:

$$\mathbf{M}_0 = \mathbf{M}_1 = \mathbf{M}_2 = \begin{bmatrix} 1\\ 1 \end{bmatrix}, \quad \mathbf{\Lambda_0} = \mathbf{\Lambda_1} = \mathbf{\Lambda_2} = \begin{bmatrix} 0 & 0\\ 1 & 0 \end{bmatrix},$$

$$\mathbf{R}_0 = \mathbf{R}_1 = \mathbf{R}_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

The proof is completed. However, let us take a further look at the three constructions. The first and third constructions correspond to reduce-multicast ring All-Reduce algorithm. In the second construction, the matrices $\mathbf{M}_i$ and $\mathbf{R}_j$ have two "1"s in some rows. This is a characteristic of non-trivial network coding. When $K = 2$ and $T = 3$, it is actually not possible to construct a feasible network code where the matrices $\mathbf{M}_i$ and $\mathbf{R}_j$ have only one "1" in each row (see Appendix).

V. Concluding Remarks

The necessary and sufficient condition derived in this paper enables us to efficiently verify the feasibility of a linear network code for the All-Reduce problem, and it also provides insights into code construction. In the context of multicast problems, analogous conditions have lead to polynomial-time algorithms for constructing feasible linear network codes [16]. The development of constructive algorithms for the All-Reduce problem remains a subject for future research.

Although this study focuses on ring networks, our analysis is extensible to other topologies with cycles. For general networks, the time-expanded network remains a directed acyclic graph with a similar repetitive structure, though nodes may have multiple incoming and outgoing transmission edges. To accommodate general network topologies, the scalar notation $y_{i\rightarrow}^{(t)}$ and $y_{i\leftarrow}^{(t)}$ must be generalized to vectors $\mathbf{y}_{i\rightarrow}^{(t)}$ and $\mathbf{y}_{i\leftarrow}^{(t)}$, respectively. It is of interest to study whether network coding can bring advantages over the reduce-multicast approach in general network topologies.

References

[1] Z. Jiang, H. Lin, Y. Zhong, Q. Huang, Y. Chen, Z. Zhang, Y. Peng, X. Li, C. Xie, S. Nong *et al.*, "MegaScale: Scaling large language model training to more than 10,000 GPUs," in *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2024, pp. 745–760.

[2] Nvidia, "Nvidia collective communication library (NCCL) documentation," 2020. [Online]. Available: https://docs.nvidia.com/deeplearning/nccl/user-guide/docs/index.html

[3] N. Dryden, N. Maruyama, T. Moon, T. Benson, A. Yoo, M. Snir, and B. Van Essen, "Aluminum: An asynchronous, GPU-aware communication library optimized for large-scale training of deep neural networks on HPC systems," in *2018 IEEE/ACM Machine Learning in HPC Environments (MLHPC)*, 2018, pp. 1–13.

[4] A. Shah, V. Chidambaram, M. Cowan, S. Maleki, M. Musuvathi, T. Mytkowicz, J. Nelson, O. Saarikivi, and R. Singh, "TACCL: Guiding collective algorithm synthesis using communication sketches," in *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2023, pp. 593–612.

[5] D. De Sensi, T. Bonato, D. Saam, and T. Hoefler, "Swing: shortcutting rings for higher bandwidth allreduce," in *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2024, pp. 1445–1462.

[6] L. Zhao, S. Pal, T. Chugh, W. Wang, J. Fantl, P. Basu, J. Khoury, and A. Krishnamurthy, "Efficient direct-connect topologies for collective communications," in *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2025, pp. 705–737.

[7] P. Patarasuk and X. Yuan, "Bandwidth optimal all-reduce algorithms for clusters of workstations," *Journal of Parallel and Distributed Computing*, vol. 69, no. 2, pp. 117–124, 2009.

[8] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, 2003.

[9] S.-Y. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[10] R. Koetter, M. Effros, T. Ho, and M. Médard, "Network codes as codes on graphs," in *Conference on Information Sciences and Systems (CISS)*, 2004.

[11] Z. Tu, Y. Chen, and S. Yang, "A network coding-based approach to floating-point sum reduction," in *IEEE International Symposium on Information Theory (ISIT)*, 2025.

[12] B. K. Rai and B. K. Dey, "On network coding for sum-networks," *IEEE Transactions on Information Theory*, vol. 58, no. 1, pp. 50–63, 2012.

[13] A. Ramamoorthy and M. Langberg, "Communicating the sum of sources over a network," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 4, pp. 655–665, 2013.

[14] S. Kamath, V. Anantharam, D. Tse, and C.-C. Wang, "The two-unicast problem," *IEEE Transactions on Information Theory*, vol. 64, no. 5, pp. 3865–3882, 2016.

[15] R. W. Yeung, *Information Theory and Network Coding*. Springer New York, NY, 2010.

[16] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1973–1982, 2005.

Appendix
Proof of Impossibility of Reduce-Multicast Approach for $K = 2$ and $T = 3$

In this appendix, we prove that when $K = 2$ and $T = 3$, it is impossible to construct a feasible reduce-multicast network code for the All-Reduce problem over a 3-node ring network.

Recall that a feasible network code must satisfy the six equations in (8) to (13) for the 3-node ring network. When $K = 2$ and $T = 3$, we further have that the multiplications of $\mathbf{\Lambda}_0$, $\mathbf{\Lambda}_1$, and $\mathbf{\Lambda}_2$ in any order must be the all-zero matrices. Therefore, we can simplify the six equations to the following ones:

$$\begin{aligned}
\mathbf{R}_0\mathbf{M}_2 &= \mathbf{I}_2, && (14)\\
\mathbf{R}_0\mathbf{\Lambda}_2\mathbf{M}_1 &= \mathbf{I}_2, && (15)\\
\mathbf{R}_1\mathbf{M}_0 &= \mathbf{I}_2, && (16)\\
\mathbf{R}_1\mathbf{\Lambda}_0\mathbf{M}_2 &= \mathbf{I}_2, && (17)\\
\mathbf{R}_2\mathbf{M}_1 &= \mathbf{I}_2, && (18)\\
\mathbf{R}_2\mathbf{\Lambda}_1\mathbf{M}_0 &= \mathbf{I}_2, && (19)
\end{aligned}$$

where $\mathbf{M}_i \in \mathbb{F}^{3\times 2}$ and $\mathbf{R}_j \in \mathbb{F}^{2\times 3}$.

For the reduce-multicast approach, the matrices $\mathbf{M}_i$ and $\mathbf{R}_j$ have the following constraints:

- Each row of $\mathbf{M}_i \in \mathbb{F}^{T\times K}$ has at most one 1 and zeros elsewhere.
- Each row of $\mathbf{R}_j \in \mathbb{F}^{K\times T}$ has at most one 1 and zeros elsewhere.

As $\mathbf{\Lambda}_i$ is strictly lower triangular, the 1's in $\mathbf{R}_j$ should not appear in the first column. Therefore, $\mathbf{R}_j$ must be of the form:

$$\mathbf{R}_j = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ or } \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

From (14), we have that $\mathbf{M}_2$ must have the form:

$$\mathbf{M}_2 = \begin{bmatrix} x & y \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ or } \begin{bmatrix} x & y \\ 0 & 1 \\ 1 & 0 \end{bmatrix},$$

where $x, y \in \{0, 1\}$. Further by (17), we have that $\mathbf{M}_2$ must have the form:

$$\mathbf{M}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \text{ or } \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Now we consider two cases. Case 1: $\mathbf{M}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$. In this case, to satisfy (14) and (17), we must have that

$$\mathbf{R}_0 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } \mathbf{R}_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Then by (16) and (19), we must have that

$$\mathbf{M}_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \text{ and } \mathbf{R}_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Finally, substituting the above form of $\mathbf{R}_0$ and $\mathbf{R}_2$ into (15) and (18), we find there is no solution for $\mathbf{M}_1$.

Case 2: $\mathbf{M}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$. We can similarly show that there is no solution to satisfy all the simplified equations. Therefore, there is no feasible network code for the All-Reduce problem over a 3-node ring network when $K = 2$ and $T = 3$.