Network Coding Theory in Collective Communications

网络编码理论与集群通信前沿进展

Shenghao Yang / 杨升浩

School of Science and Engineering, CUHK Shenzhen / 香港中文大学（深圳）理工学院

中国电子学会信息论分会会员日报告
2026 年 8 月 21 日

Collaborators

- **Yi Chen / 陈怿**, Research Assistant Professor
  - School of Science and Engineering, CUHK Shenzhen
  - Shenzhen Future Network of Intelligence Institute (FNii)
- **Zhuoqi Tu / 屠卓奇**
  - former MPhil student at CUHK Shenzhen
  - currently a PhD student at CUHK

Outline

1. Introduction: collective communication in AI clusters
2. Part 1: Floating-point sum reduction via network coding
3. Part 2: Linear network coding for sum All-Reduce on rings
4. Takeaways and open directions

Introduction

Why collective communication?

- Large AI clusters (tens of thousands of GPUs) for LLM training
- Local computation on each node; **synchronization** via collectives
- Key operations: Broadcast, Reduce, All-Gather, **All-Reduce**, ...
- Communication can dominate runtime ($6\times$ on 1024 GPUs, $12\times$ on 2048 GPUs, vs. compute)

Networking—not FLOPs—is often the bottleneck.

All-Reduce for sum / average

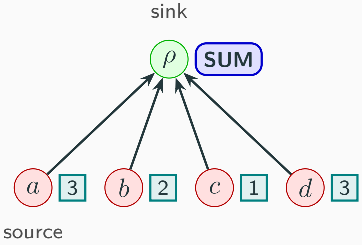
Task

Each node i holds a vector $\mathbf{x}_i$, and must obtain $\sum_i \mathbf{x}_i$ (or the average).

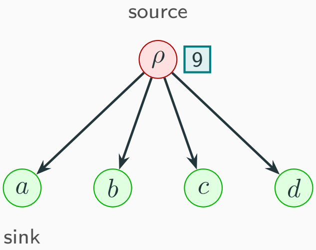
- Central in distributed SGD / data-parallel LLM training.
- Implemented in libraries such as NCCL.
- Common topologies: **rings**, trees, and richer fabrics.

Classic paradigm: the **reduce-multicast approach** (reduce, then multicast).

Example: sum reduction and multicast



(a) Sum reduction



(b) Multicast

Figure 1: All-Reduce via the reduce-multicast approach

Theoretical questions

There are several natural theoretical questions arising from All-Reduce.

- Is the reduce-multicast approach optimal for All-Reduce?
- How much can we improve the rate of All-Reduce by *network coding*?

Part 1: Floating-Point Sum Reduction

Floating-point sum reduction

Problem

In a DAG with fixed link capacities, a number of source nodes each hold real values, and one sink computes their sum using *floating-point* arithmetic.

- Goal: high throughput **and** high numerical accuracy
- Over finite fields: network coding achieves the exact sum via multicast duality
- Floating-point: roundoff introduces error; a rate-distortion tradeoff

Based on: Tu, Chen, Yang — *A Network Coding-Based Approach to Floating-Point Sum Reduction* (ISIT 2025).

Multicast $\leftrightarrow$ sum reduction duality[1]

(a) Multicast

(b) Sum reduction (dual)

- Reverse edges; transpose coding coefficients
- Multicast code $\leftrightarrow$ sum reduction code

[1] Ralf Koetter et al. **"Network codes as codes on graphs".** In: *Conference on Information Sciences and Systems (CISS)*. 2004.

Another example where simple forwarding is not enough

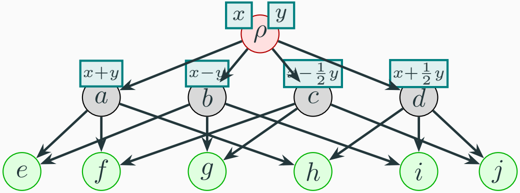


Figure 3: Multicast (linear network coding, rate 2)

- Forwarding only achieves rate 1.
- Forwarding with time-sharing achieves rate $4/3$.

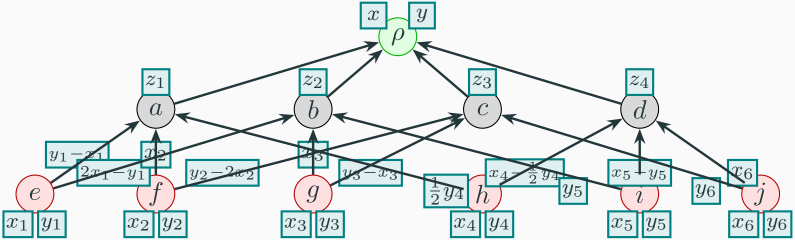


Figure 4: Sum reduction (dual)

Linear network coding over the real numbers

Network coding formulation

- Source message $\mathbf{x} \in \mathbb{R}^{1\times r}$.
- On edge e: $x_e = \mathbf{x}\,\mathbf{c}_e$ with recoding

$$\mathbf{c}_e = \sum_{e' \in \mathrm{In}(v)} \beta_{e',e}\,\mathbf{c}_{e'}$$

- Transfer matrix $\mathbf{F}_t$ at sink t: decodable iff $\mathrm{rank}(\mathbf{F}_t) = r$

- The capacity of the network is infinite if each link can transmit one real number per time.
- $\min_t \mathrm{MaxFlow}$ is NOT an upper bound on the achievable rate.
- In practice, we implement real-valued network coding using floating-point arithmetic.

Existence

For any $r \leq \min_t \mathrm{MaxFlow}$, there exist real coefficients making each $\overline{\mathbf{F}}_t$ full rank (density of $\mathbb{R}$), where $\overline{\mathbf{F}}_t$ is an $r \times r$ submatrix of the transfer matrix at sink t.

Floating-point: where error enters

Encoding at source t uses $\overline{\mathbf{F}}_t^{-1}$ in floating-point:

- Representation error: $\mathrm{float}(\overline{\mathbf{F}}_t^{-1}) - \overline{\mathbf{F}}_t^{-1}$
- Multiplication roundoff in $\mathbf{x}_t(\mathrm{float}(\overline{\mathbf{F}}_t^{-1}))^\top$

Decoding and recoding introduce further perturbation $\mathbf{E}_t^{(2)}$.

Distortion bound (sketch)

$$\frac{\|\mathbf{x}_t^{(1)} - \mathbf{x}_t\|}{\|\mathbf{x}_t\|} \lesssim \kappa^3(\overline{\mathbf{F}}_t) \cdot (\cdots) + \kappa^2(\overline{\mathbf{F}}_t) \cdot (\cdots) + \kappa(\overline{\mathbf{F}}_t) \cdot (\cdots)$$

Minimize condition numbers $\kappa(\overline{\mathbf{F}}_t)$.

Optimizing the network code

- Construct code along a sequence of growing subgraphs (Jaggi-style)
- When adding edge e, choose $\mathbf{c}_e \in V$ to keep full rank **and** reduce $\prod_{t \in \mathcal{T}_e} \kappa(\mathbf{F}_t)$
- Frobenius / determinant bound: $\kappa(\mathbf{F}_t) \leq \frac{2}{|\det(\mathbf{F_t})|} \left(\frac{\|\mathbf{F}_t\|_F}{\sqrt{r}} \right)^r$.
  - $\det(\mathbf{F}_t)$ is a linear function of $\mathbf{c}_e$
  - $(\|\mathbf{F}_t\|_F)^2$ is a quadratic function of $\mathbf{c}_e$
- Proxy for reducing $\prod_t \kappa(\mathbf{F}_t)$:

$$\min_{\boldsymbol{\alpha}} \prod_t \frac{(\boldsymbol{\alpha}^\top \boldsymbol{\alpha} + \gamma_t^2)^{r/2}}{|\mathbf{b}_t^\top \boldsymbol{\alpha}|},$$ where $\mathbf{b}_t$ and γ_t are constants.

Experiment: combination network $(10, 4)$

- Root $+$ 10 intermediate nodes $+$ $\binom{10}{4}$ sinks; max-flow $= 4$
- float32 sum reduction, $N = 1024$ trials

	Optimized	Random
$\sum_t \log \kappa(\overline{\mathbf{F}}_t)$	5.7	487
distortion rate	3.9×10^{-7}	1.1×10^{-6}

Optimization lowers κ and the measured relative error.

Part 1 Summary

1. When the rate equals max-flow, we show how to reduce floating-point error by optimizing the network code.
2. The general rate-distortion tradeoff remains open.
3. A natural question is *whether the reduce-multicast approach is optimal for All-Reduce*.

Part 2: Linear Network Coding for Sum All-Reduce on Rings

A general All-Reduce problem

Multi-sum

Each source generates a message; **every** destination wants the sum of all source messages.

- A network-coding view of All-Reduce (finite fields, DAGs) [2, 3]
- Equivalent to multiple unicast $\Rightarrow$ hard in general, and linear coding is not sufficient [2, 4]
- 2 sources or 2 destinations: feasible iff every pair has a path
- 3 sources and 3 destinations: **two** edge-disjoint paths per pair are necessary; one path is not sufficient

Linear network coding for sum All-Reduce on rings

(N, K) Ring All-Reduce

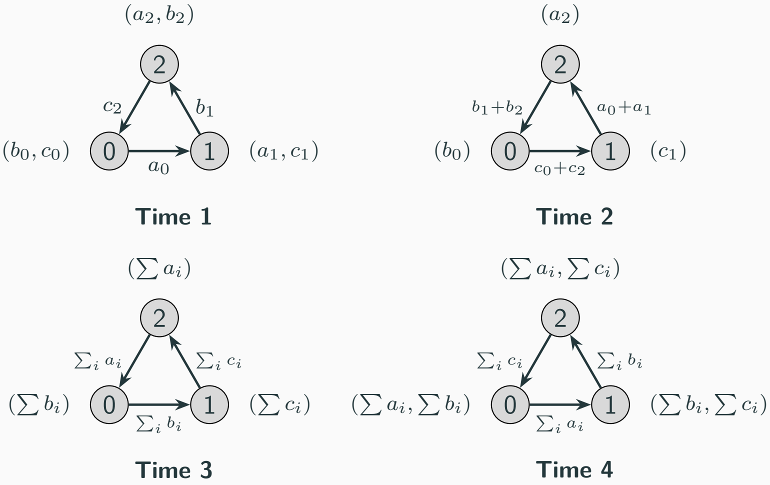
N-node directed ring; each link carries one symbol from field $\mathbb{F}$ per time.

Node i holds $\mathbf{x}_i \in \mathbb{F}^K$. Goal: every node recovers $\sum_i \mathbf{x}_i$.

- Classic ring All-Reduce: $K = N$
  - Packing N instances of the reduce-multicast approach.
  - Rate $\frac{N}{2(N-1)}$.
- Question: can network coding do better?

Based on: Tu, Chen, Yang — *Linear Network Coding for Sum All-Reduce over Ring Networks* (ISIT 2026).

Classic ring All-Reduce ($N = 3$, $K = 3$)

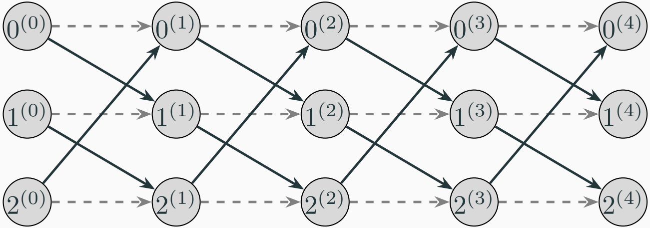


Three sums in $T = 4$ steps $\Rightarrow$ rate $K/T = 3/4$.

Time-expanded network G_T

- Vertices $i^{(t)}$ for $t = 0, \ldots, T$
- **Transmission** edges: $(i^{(t)}, (i{+}1)^{(t+1)})$
- **Storage** edges: $(i^{(t)}, i^{(t+1)})$
- All-Reduce in T steps $\Leftrightarrow$ multi-sum on G_T:
  sources at $t = 0$, destinations at $t = T$ recover $\sum \mathbf{x}_i$

We apply ordinary (time-varying) linear network coding on the DAG G_T.



(K,T) linear network code

At time t, node i sends

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

and at time T decodes

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

Definition (Feasibility)

$\hat{\mathbf{z}}_i = \sum_j \mathbf{x}_j$ for all i and all messages.

- Rate of a feasible code: K/T
- Reduce-multicast approach: sparse $0/1$ coefficients, same-index additions only

Necessary condition (any N)

Proposition

A feasible (K, T) *code requires*

$$T \geq K + N - 2.$$

- From edge-disjoint path counts in G_T
- When $K = N$:
  - $T \geq 2(N-1) \Rightarrow$ rate $\leq \frac{N}{2(N-1)}$
  - Classic ring All-Reduce **meets** this bound
- In general, $K/T \leq 1 - \frac{N-2}{T} \rightarrow 1$

3-node ring: tight characterization

Theorem

A (K, T) linear network code for All-Reduce on the 3-node ring is feasible
iff

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

Corollary

- *Achievable rate $\leq 3/4$.*
- *Classic ring All-Reduce achieves $3/4$ at $(K, T) = (3, 4)$.*

General linear network coding does not improve the asymptotic rate on the 3-node ring.

Feasibility equations for $N = 3$

From the general characterization, a (K, T) linear network code is feasible **iff** there exist matrices $\mathbf{M}_i$, $\mathbf{\Lambda}_i$, $\mathbf{R}_i$ $(i = 0, 1, 2)$ such that, for every i (indices mod 3),

$$\mathbf{R}_i' \mathbf{M}_{i-1} = \mathbf{I}_K \quad \text{and} \quad \mathbf{R}_i' \mathbf{\Lambda}_{i-1} \mathbf{M}_{i-2} = \mathbf{I}_K,$$

where

$$\mathbf{R}_i' = \mathbf{R}_i \big(\mathbf{I}_T - \mathbf{\Lambda}_{i-1} \mathbf{\Lambda}_{i-2} \mathbf{\Lambda}_i\big)^{-1}.$$

- Six equations in total (two per node i).
- Node i extracts each other source's message so it can form the sum.
- $\mathbf{\Lambda}_i$ about storage only is strictly lower triangular.

Proof idea (necessity)

From the feasibility equations (previous slide), define

$$\mathbf{\Phi}_0 = \mathbf{M}_2 - \mathbf{\Lambda}_2\mathbf{M}_1, \quad \mathbf{\Phi}_1 = \mathbf{M}_0 - \mathbf{\Lambda}_0\mathbf{M}_2, \quad \mathbf{\Phi}_2 = \mathbf{M}_1 - \mathbf{\Lambda}_1\mathbf{M}_0.$$

- Each $\mathrm{rank}(\mathbf{\Phi}_i) \leq T - K$
- Express $\mathbf{M}_0$ via $(\mathbf{I} - \mathbf{\Lambda}_0\mathbf{\Lambda}_2\mathbf{\Lambda}_1)^{-1}$ and the $\mathbf{\Phi}$'s
- Full-rank $\mathbf{M}_0$ forces

$$K \leq \sum_i \mathrm{rank}(\mathbf{\Phi}_i) \leq 3(T - K) \;\Rightarrow\; T \geq \tfrac{4}{3}K.$$

Sufficiency by construction

It is enough to realize base cases and tile:

(K, T)	Rate	Type
$(1, 2)$	$1/2$	reduce-multicast
$(2, 3)$	$2/3$	non-trivial network coding
$(3, 4)$	$3/4$	classic ring All-Reduce

For general K: pack $\lfloor K/3 \rfloor$ blocks of $(3, 4)$ plus a residual $(1, 2)$ or $(2, 3)$.

Why $(K, T) = (2, 3)$ requires network coding

- Reduce-multicast approach (one '1' per row of $\mathbf{M}_i, \mathbf{R}_j$) is **impossible** for $(2, 3)$
- Explicit feasible code uses denser rows, e.g.

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

- Same phenomenon for all $K = 3k + 2$, $T = 4k + 3$

Message

Rate ceiling matches classic ring All-Reduce,
but **non-trivial network coding is necessary** for some (K, T).

Beyond $N = 3$ (glimpse)

Recall the general necessary condition (any N): $T \geq K + N - 2$.

N	General bound	Improved bound (tight)	Rate cap K/T	Classic ring ($\frac{N}{2(N-1)}$)
4	$T \geq K + 2$	$T \geq \max\{K + 2, \lceil \frac{3K}{2} \rceil\}$	$\leq \frac{2}{3}$	$\frac{2}{3}$
5	$T \geq K + 3$	$T \geq \max\{K + 3, \lceil \frac{8K}{5} \rceil\}$	$\leq \frac{5}{8}$	$\frac{5}{8}$

- The improved bound dominates the general bound for large K
- When $N = 4$, the improved bound is tight for all (K, T) satisfying the necessary condition.
  - Linear network coding does not beat the reduce-multicast approach asymptotically,
  - but it may be necessary for certain values of K and T.

Part 2 Summary

1. The classic reduce-multicast approach is rate-optimal when $K = N$.
2. Network coding is necessary to achieve the optimal rate in some cases with $K < N$.
3. Can non-linear network coding do better?

Closing

Unified picture

1. **Real-valued network coding solution**
   Find the largest rate achievable by a linear network code over $\mathbb{R}$ for sum All-Reduce.
2. **Floating-point network coding with lower error**
   Design a floating-point network code that achieves that rate with low error.

Takeaways

- AI does not yet handle network codes well, but it can help generate Lean code from our proofs, enabling formal machine verification of our results.
- Open problems:
  - Generalize the improved bound to arbitrary N.
  - Extend the results non-ring topologies.

Selected references

[1] Ralf Koetter et al. **"Network codes as codes on graphs".** In: *Conference on Information Sciences and Systems (CISS)*. 2004.

[2] Brijesh Kumar Rai and Bikash Kumar Dey. **"On Network Coding for Sum-Networks".** In: *IEEE Transactions on Information Theory* 58.1 (2012), pp. 50–63.

[3] Aditya Ramamoorthy and Michael Langberg. **"Communicating the Sum of Sources Over a Network".** In: *IEEE Journal on Selected Areas in Communications* 31.4 (2013), pp. 655–665.

[4] Sudeep Kamath et al. **"The two-unicast problem".** In: *IEEE Transactions on Information Theory* 64.5 (2016), pp. 3865–3882.

[5] Zhuoqi Tu, Yi Chen, and Shenghao Yang. **"A network coding-based approach to floating-point sum reduction".** In: *IEEE International Symposium on Information Theory (ISIT)*. 2025.

[6] Zhuoqi Tu, Yi Chen, and Shenghao Yang. **"Linear network coding for sum**

Thank you! / 谢谢！

Questions? / 提问欢迎

Shenghao Yang / 杨升浩 shyang@cuhk.edu.cn

The Chinese University of Hong Kong, Shenzhen

香港中文大学（深圳）