Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: 14 November 2026

Z. Wang
Q. Sun
H. Gao
BUPT

13 May 2026

Topology-Aware Construction of Collective Communication over Time-Expanded Networks

draft-wang-topology-aware-collective-communication-00

Abstract

This document describes a topology-aware method for constructing collective communication schedules in distributed systems. Instead of selecting from a small set of predefined communication algorithms, the method expands a target network topology along a time dimension, tracks per-node data state, and incrementally builds a schedule through candidate-source discovery and link-to-chunk matching. The approach is intended for heterogeneous or asymmetric topologies in which fixed communication patterns often underutilize available links or create avoidable bottlenecks. According to the source material, the resulting schedule is intended for collective communication tasks involving data distribution, aggregation, reduction, and synchronization, including all-gather, reduce-scatter, and all-reduce.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 14 November 2026.

Table of Contents

1. Introduction

Large-scale distributed training systems rely heavily on collective communication. During training, nodes repeatedly exchange model parameters, gradients, or intermediate results. As model size and cluster size increase, the communication subsystem becomes a major factor in overall job completion time.

Many existing implementations choose a communication procedure from a predefined algorithm library. This works reasonably well on regular topologies, but it becomes less effective when the underlying network contains heterogeneous links, asymmetric connectivity, or multi-level structure. In such environments, a fixed algorithm can overload some links while leaving others underused.

Other approaches attempt to generate schedules through global optimization or exhaustive search. These methods can produce good results for some inputs, but their construction cost often grows quickly with the number of nodes and links, which makes them harder to use in large systems.

This document describes a different construction framework. It uses a time-expanded network to represent both topology and time evolution, models which data chunks are currently available at which nodes, and incrementally builds communication steps until the target communication state is reached.

2. Problem Statement

The construction method described in this document is motivated by five practical issues.

1. Predefined collective algorithms do not always fit a specific topology, especially when the topology is irregular.

2. Heterogeneous and asymmetric networks make it difficult to maintain both good link utilization and low completion time with a single fixed communication pattern.

3. Many implementations do not expose a unified model for data state, temporal progression, and link occupancy during schedule construction.

4. Globally optimized schedule generation can become expensive as system size increases.

5. A practical method should support multiple collective communication modes without being tied to a single handcrafted algorithm family.

3. Design Goals

The source material indicates that the construction method is intended to meet the following goals:

* It is intended to account for topology details, including directed links, asymmetric connectivity, and bandwidth differences.

* It is intended to build a schedule for the given topology, rather than only choosing from a fixed library of existing algorithms.

* It is intended to represent communication progress as an explicit state transition process.

* It is intended to expand the schedule progressively in time instead of requiring a fully fixed time horizon in advance.

* It is intended to remain applicable to several collective communication patterns.

4. Terminology and Model

This section summarizes the key terms used by the construction method.

Node
: A compute element or processing unit participating in collective communication.

Directed Link
: A communication edge from one node to another. Link direction matters when the physical or logical topology is asymmetric.

Chunk
: The smallest schedulable unit obtained by splitting the data to be communicated.

Precondition
: The set of chunks currently available at each node at a given time layer.

Postcondition
: The target chunk distribution that defines completion of the collective task.

Unsatisfied Target
: A node-chunk pair that is required by the postcondition but has not yet been satisfied in the current state.

Candidate Source
: A node that already holds a required chunk and can potentially deliver it to a target node through an available link in the current time layer.

Time-Expanded Network
: A representation in which the original topology is unfolded across discrete time layers so that communication opportunities and state transitions can be expressed in a single structure.

5. Construction Framework

The method takes as input a topology description, link attributes, a collective communication objective, and an initial data placement. It produces a communication schedule consisting of per-time-layer send and receive actions, together with the derived transfer paths of each chunk.

5.1. Input Topology and Communication Objective

The constructor receives a set of nodes, a set of directed links, and link attributes including link bandwidth. It also receives the target collective mode. The source material explicitly mentions all-gather, reduce-scatter, and all-reduce, and more generally describes distribution, aggregation, reduction, and synchronization tasks.

The communication objective is expressed as a desired postcondition over chunks. The postcondition defines which nodes are expected to hold which chunks after the collective operation completes.

5.2. Chunk Partitioning and State Initialization

Before schedule construction begins, the payload is partitioned into chunks. Each chunk is treated as an independent schedulable item.

The constructor initializes a precondition describing which chunks are currently available at each node. It also initializes the postcondition that describes the desired final state. These two state descriptions provide the basis for deciding whether construction is complete and which transfers are still needed.

5.3. Time-Expanded Topology

The original topology is expanded into discrete time layers. Each original node is represented by a sequence of node copies, one for each layer. A directed link in the original topology becomes a temporal edge that carries a chunk from a source node in one layer to a destination node in the next layer.

This representation captures spatial connectivity and temporal progression in one model. The source material describes a layer-by-layer expansion process rather than requiring a fixed final time horizon in advance.

5.4. Candidate Source Discovery

For each unsatisfied target, the constructor searches for candidate sources that already hold the required chunk and can reach the destination through a valid temporal edge in the current layer.

According to the source material, this process starts from the target node and traces reachable links in the current time layer to identify source nodes that already hold the required chunk. The candidate set therefore reflects both current chunk availability and current temporal connectivity.

5.5. Link-to-Chunk Matching

After candidate sources have been identified, the constructor selects feasible transfers for the current time layer. A feasible transfer binds one chunk to one directed link from one source node to one destination node.

According to the source material, valid matching respects link direction and only uses links that are not already occupied in the current layer. The source material also notes that alternative embodiments may consider path length, node load, or link utilization when adjusting matching decisions.

5.6. State Update and Iterative Expansion

Once transfers have been selected for the current layer, the constructor updates node state for the next layer. Any chunk that is successfully delivered becomes part of the receiving node’s precondition in the following layer.

If unsatisfied targets remain after the update, the constructor extends the time-expanded network and repeats candidate discovery and link-to-chunk matching. This process continues until the postcondition is fully satisfied.

5.7. Resulting Schedule

The final output is a topology-aware communication schedule. According to the source material, the output includes the transfer path of each chunk, the send/receive relationships in each time layer, and the complete collective communication plan.

6. Applicability to Collective Operations

The same construction framework can be applied to several collective communication patterns by changing the initial and target state definitions.

* In all-gather, each node starts with a subset of chunks, and the postcondition requires every node to hold the union of all chunks.

* In reduce-scatter, chunks are combined according to a reduction rule and the postcondition assigns portions of the reduced result to different nodes.

* In all-reduce, the framework can be viewed as a reduce-scatter phase followed by an all-gather phase, or as another equivalent state formulation.

7. Execution Considerations

The source material explicitly lists the node set, link set, link direction, link bandwidth, and target collective mode as inputs to schedule construction. It also describes the generated schedule as an output that can later be read and executed by sending, receiving, and processing data according to the per-layer communication arrangement.

The source material further states that, after construction, the resulting communication schedule can be executed layer by layer until the target completion state is reached, and the final result can then be supplied to later training, scheduling, or control modules.

8. IANA Considerations

This document includes no request to IANA.

9. Security Considerations

This document describes a schedule construction method and does not define a new wire protocol or a new security mechanism.