Network Working Group
Internet-Draft
Intended status: Experimental
Expires: 28 November 2026

H. Song
Futurewei Technologies
T. Zhou
Huawei

27 May 2026

In-Network Aggregation Framework with Virtual Aggregation Tree and BIER
draft-song-ina-00

Abstract

AllReduce is a critical performance bottleneck for distributed deep learning and large model training in data centers for AI computing. In-Network Aggregation (INA) has been identified as an effective accelerating technique to improve its performance. The draft describes a flexible and efficient INA solution for packet routing and forwarding. The forward aggregation tree is encoded by a bitmap. The result dissemination is through BIER-based multicast which also relies on a bitmap. The two bitmaps share the same encoding scheme as specified in BIER.

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 28 November 2026.

Copyright Notice

Table of Contents

1. Introduction

Optimizing the Data Center Networks (DCN) is critical for improving the efficiency of AI computing, especially in the scenarios of parallel jobs and multiple tenants. In-Network Computing (INC), an emerging computing paradigm, aims to engage network switches to execute application functions to improve the application performance or reduce the system cost.

Among the collective communication primitives used by distributed AI computing, AllReduce has gained the most attention for in-network acceleration due to its popularity, performance impact, and suitability. "Reduce" represents the operation of sum, multiplication, max, or min on data from multiple sources. The AllReduce operation reduces a batch of arrays from the participating workers and distributes the resulting array to all the workers. Host-based AllReduce is realized by using a logical ring or tree in which the network only provides point-to-point connectivity. Specifically, the tree-based implementation involves a dedicated server, known as Parameter Server (PS), as the central point to receive data from all participating nodes, conduct data reduction, and send the result back to the nodes through unicast.

Such an implementation can be accelerated by INC through a method dubbed as In-Network Aggregation (INA). Since the network switches have memory space to buffer the arrays from the workers and have computing capability to conduct the reduction operation, the aggregation can be offloaded to the switches. The basic approach is that, for each job, an overlay aggregation tree is built on top of the DCN, in which the leaves are the workers, the root is the PS, and the internal nodes are the switches which are responsible to aggregate the arrays coming from their child nodes.

In this draft, we describe a flexible and efficient INA framework for packet forwarding. INA involves two phases: the forward aggregation phase and the backward dissemination phase. In the forward aggregation phase, we introduce Virtual Aggregation Tree (VAT) which can be mapped on a DCN topology to support INA for an AllReduce job. The bitmap mechanism is used to encode the VAT and track the aggregation status.

In the backward dissemination phase, we adopt the BIER forwarding [RFC8279] to multicast the result to the worker nodes. BIER does not require constructing a tree in advance, nor does it necessitate per-flow states in intermediate nodes. The simplicity and scalability make it ideal for aggregation result dissemination. Coincidentally, the VAT for the same AllReduce job also relies on a bitmap which has the similar encoding semantics as for multicast but is used on the opposite data moving direction. Therefore, the bitmap can be used for both aggregation and dissemination in a congruent INA solution.

2. The INA Framework

2.1. Aggregation Phase

In essence, the in-network aggregation traffic follows a tree structure. While each leaf node sends a packet towards the root, each internal tree node aggregates the packets received from its child nodes. The aggregation result at each internal node continues to be sent toward the root. The root finishes the final aggregation and multicasts the result back to all the leaves. The multicast tree does not need to overlap with the aggregation tree (except the root and leaves).

We build a VAT on top of the DCN topology. The VAT root can be a switch or a server. The VAT leaves are the server nodes. All other VAT nodes are mapped to arbitrary switches with two constraints: (1) each switch can be mapped by at most one VAT node, and (2) network connectivity exists between any two switches that are mapped to two adjacent VAT nodes.

Each server node is assigned a bit in a bitmap. For an AllReduce job, the bits for the selected workers are set to ‘1’. On a VAT, each non-leaf node is configured with a bitmap named A-BM to register the set of leaves it is responsible for aggregation. A-BM covers all the downward leaves of the node. When a worker sends a packet with an array for aggregating to the root, the packet also carries a bitmap named P-BM, in which only the bit corresponding to the worker is set to ‘1’.

When a switch mapped to a VAT node for the job receives a data packet, it performs the bit-wise AND operation on A-BM and P-BM. If it results in an all-zero bitmap, it means the packet is not supposed to be aggregated at this switch, so it continues to be forwarded towards the root. Otherwise, the packet is terminated at this switch and the array is buffered for aggregation. Once the switch collects all the arrays that need to be aggregated (i.e., the bit-wise OR of the P-BMs from the aggregated packets equals to the A-BM) and conducts the aggregation, the result packet, which carries a P-BM equal to the A-BM of the switch, is sent towards its parent VAT node. This process repeats until the root finishes the final aggregation.

Fig. 1 shows a network and a VAT constructed over it. There are 8 servers in the network. Therefore, the bitmap contains 8 bits and w_i is assigned the i-th bit in the bitmap. We assume the first 4 servers (w1 - w4) are used as workers for an AllReduce job. We decide to use s1 to aggregate the arrays form w1 and w2, use s7 to aggregate the arrays from w3 and w4, and use s6 to aggregate the arrays from s1 and s7. To achieve this, we configure the A-BMs for the job on the involved switches as shown in Fig. 1(a), which leads to the VAT as shown in Fig. 1(b).

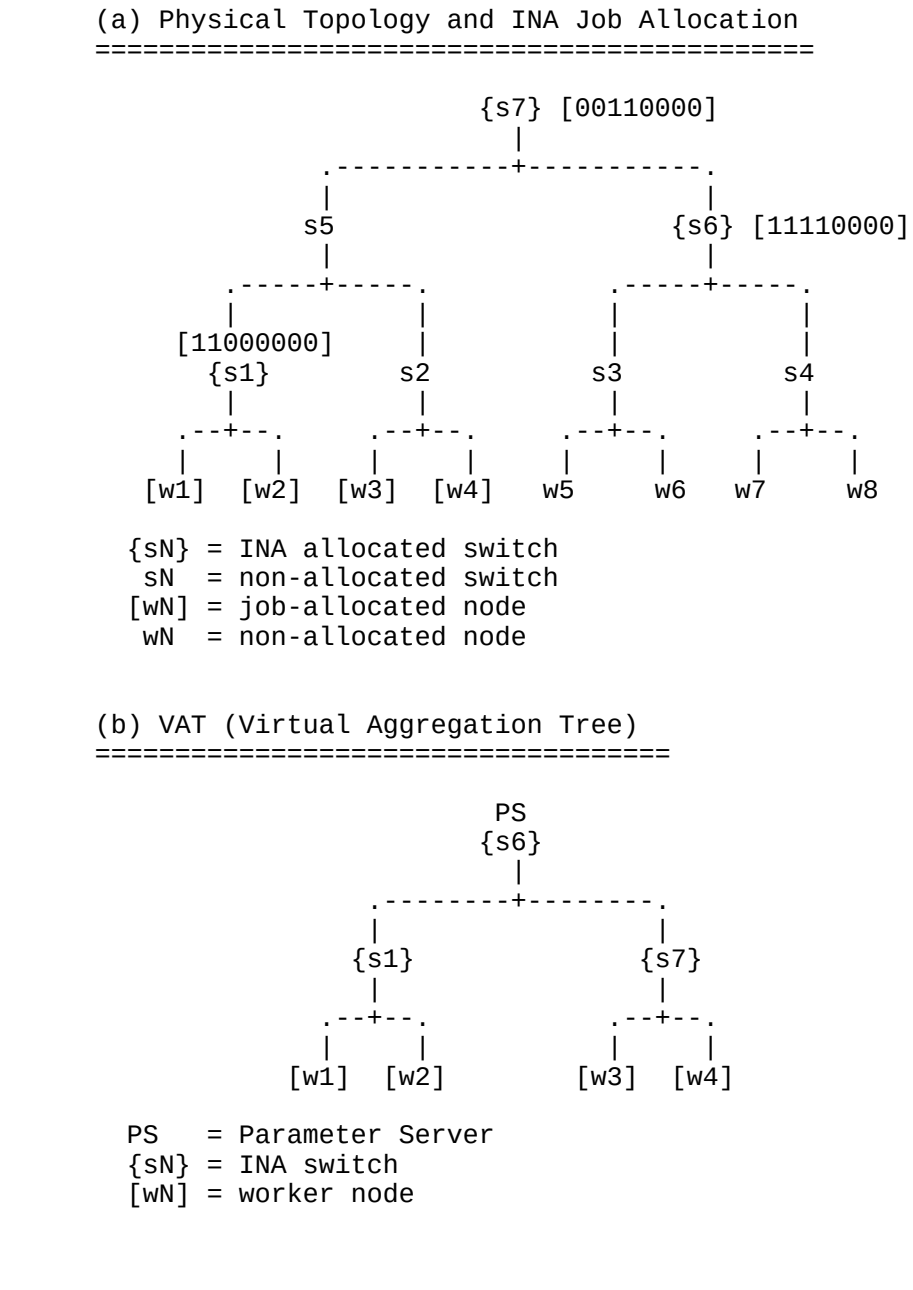

Figure 1: Physical Topology and VAT

The algorithm to construct VATs and the protocol for packet routing and forwarding between VAT nodes are out of the scope of this document.

2.2. Dissemination Phase

The most efficient way for result dissemination is through a multicast tree. The multicast tree shares the root and the leaves with the corresponding VAT, but may have different shape. Most existing multicast protocols require building explicit multicast trees and maintaining per-flow state at intermediate nodes. Instead, the BIER forwarding architecture allows each multicast packet to carry a succinct bitmap in a BIER header to identify the targets. Therefore, BIER is used for the result dissemination. In this context, the root is the BFIR, the leave nodes are BFERs.

2.3. VAT Bitmap Encoding

While the dissemination phase can use the BIER multicast directly, the header format for the aggregation phase needs to be defined. Due to the semantic similarity, the VAT bitmap adopts the same specification as BIER, i.e., the method to encode BFR IDs. Consequently, the A-BM configured at the VAT root node can be directly used as the BIER bitmap for multicast.

3. Security Considerations

TBD.

4. IANA Considerations

TBD.

5. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <https://www.rfc-editor.org/info/rfc8279>.

6. Informative References

Authors' Addresses

Haoyu Song
Futurewei Technologies
United States of America
Email: haoyu.song@futurewei.com

Tianran Zhou
Huawei
China
Email: zhoutianran@huawei.com