Collective Communication Optimization(CCO): Use cases, Problem Statement, and Requirement

Personal I-Ds:
1 *https://datatracker.ietf.org/doc/draft-yao-tsvwg-cco-problem-statement-and-usecases/*
2 *https://datatracker.ietf.org/doc/draft-yao-tsvwg-cco-requirement-and-analysis/*

Kehan Yao, China Mobile

Shiping Xu, China Mobile

Yizhou Li, Huawei

Hongyi Huang, Huawei

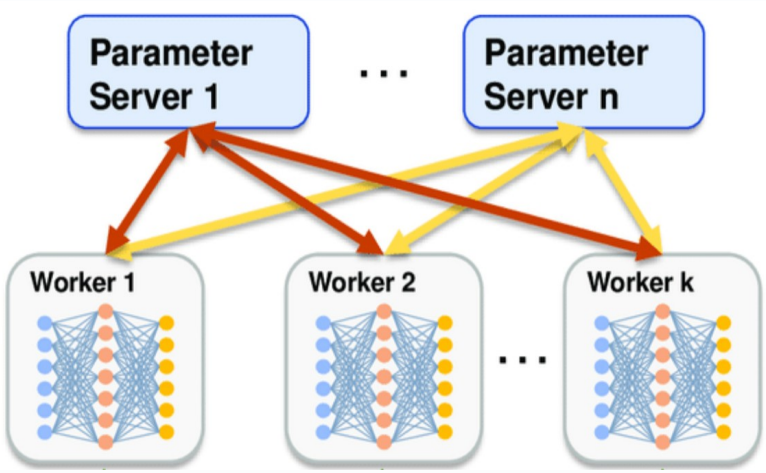
Dirk Kutscher, HKUST(GZ)

IETF 118

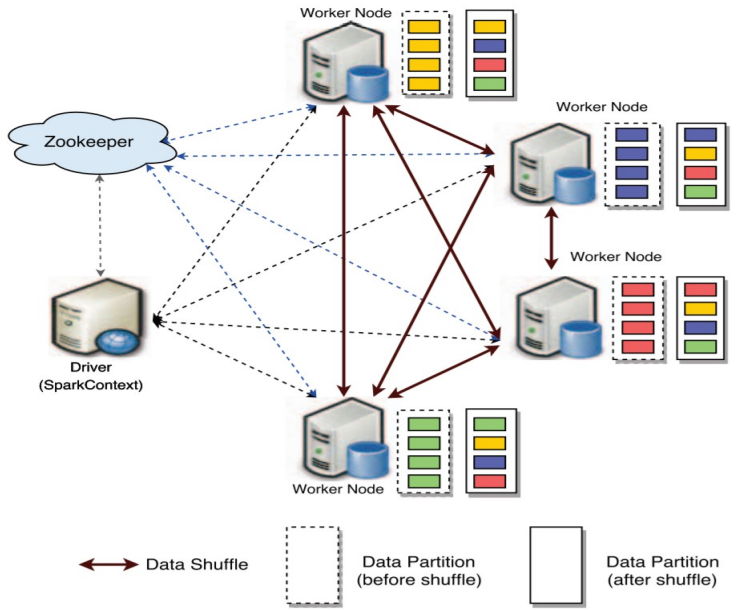
Concept:

Collective communication is an inter-process communication model which plays a key role in high performance computing and modern distributed AI model training workloads such as recommender systems and natural language processing. It involves a group or groups of processes participating in collective operations like AllReduce or AllGather. The communication model can be one-to-all, all-to-one or all-to-all and is usually realized by a sequence of unicast messages.

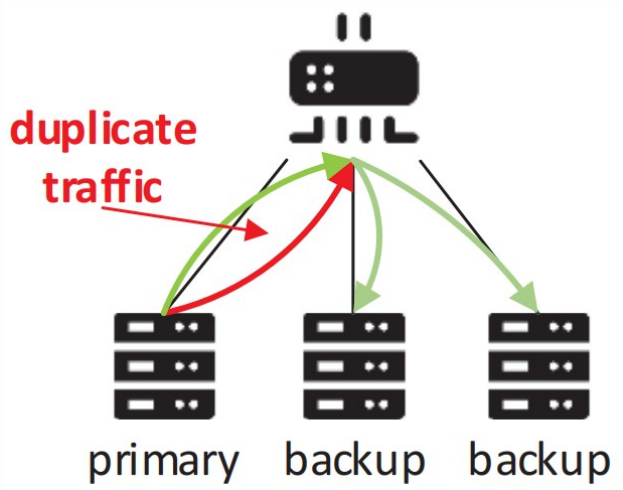
Use cases:



Distributed AI Model Training

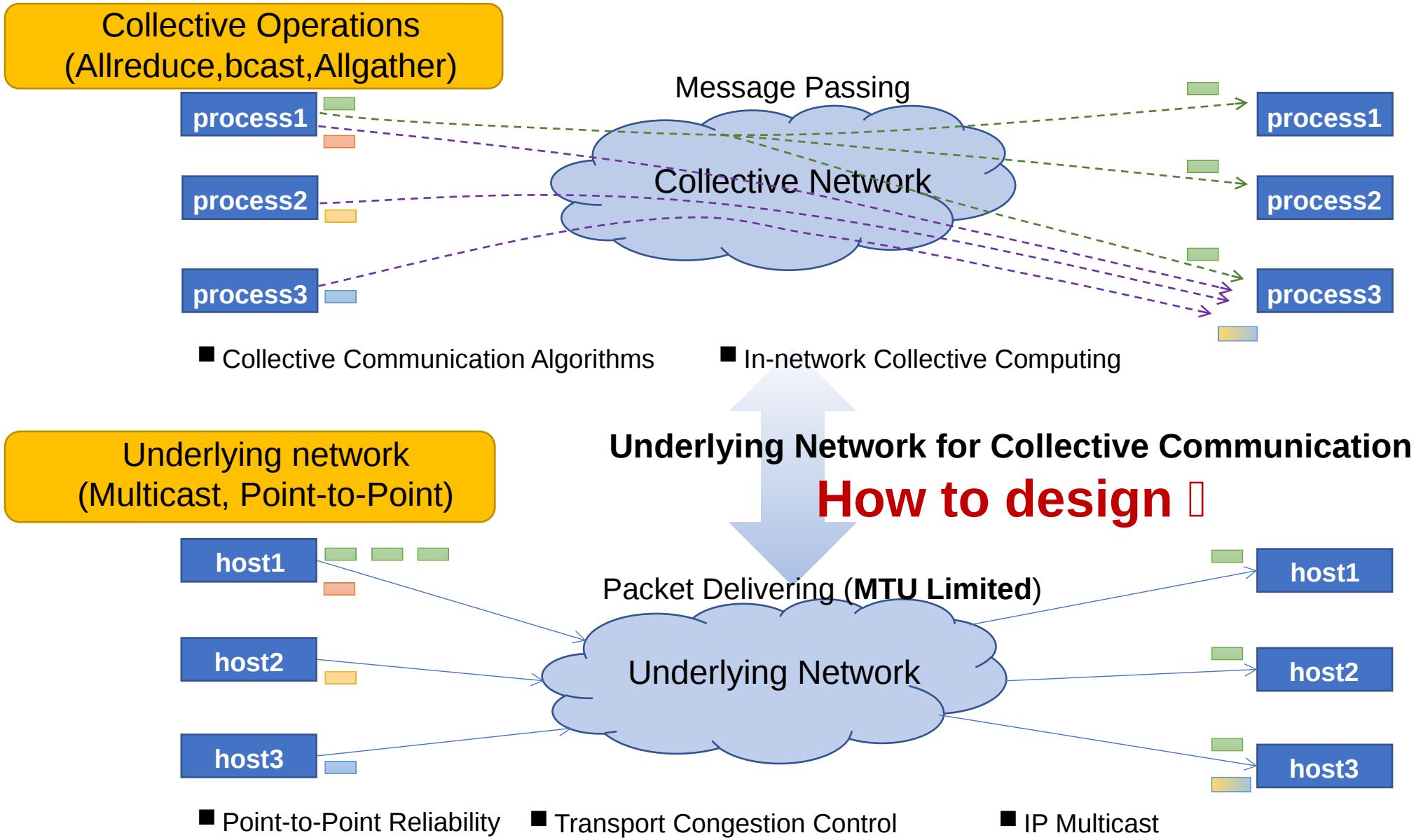


Spark Shuffle
in Big Data Anlaysis



Distributed Storage

Communication Pattern:



Major Problems & Observation:

- P2P implementation of Collective Communication incurs much overhead, reflected in:
  - **large bandwidth occupancy(duplications & redundancy)**
  - **much data movement(end-to-end transmission)**
  - **large number of data copies at endpoints(sending one pkt needs to copy at least one time).**

Communication bottleneck & performance degradation

- It should
  - save bandwidth(This is extremely important for BW-sensitive Apps like distributed AI model training workloads, **since BW is the new oil**).

    --- "***The metaphor is not from me, but I think it is quite impressive.*** "
  - reduce data movement.
  - decrease data copies.

- **Offloading collective operations to the network** is important for achieving benefits above and very necessary, especially for these performance-driven Apps.

Design Issues:

- **Transport Issues:**
  - Reliability

  underlying network lacks collective communication reliability
  - Semantic Gap

  message passing vs packet delivering
  - Blocking & Non-blocking

  different optimizations for different communication modes

- **One-to-Group Transmission:**
  - IP Multicast for Message Bcast/AlltoAll/...

  IP multicast is the most direct way, perhaps there is a better way

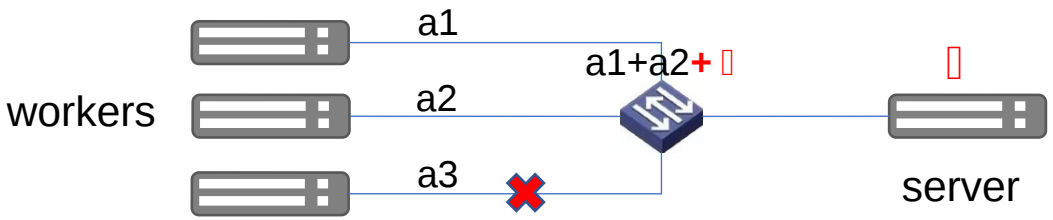
- **Data & Control & Management:**
  - In-network Primitives

  collective operations based on unified In-network primitives
  - Topology Awareness

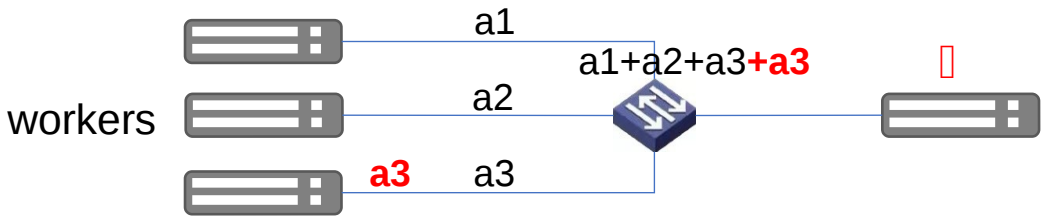
  to improve existing topology aware algorithms to support in-network computing

Transport Issues:

- **Reliability**
  - Point-to-Point reliability doesn't work well when collective operations are offloaded to intermediate nodes(switch and/or NIC).
  - Two potential ways to guarantee reliability in Group-to-one mode:
    - The intermediate node(switch) acts as an endpoint, two different session will be established between worker-to-switch, and switch-to-server.
      - **(Switch needs to maintain full transport function and states, encryption will make it worse.)**
    - The intermediate node(switch) doesn't act as an endpoint. End-to-End principle is maintained.
      - **(Switch doesn't need to maintain full transport function and states, but it needs to be aware of the operations it should perform on the pkts and needs some loss recovery and correctness guarantee mechanisms.)**



Sender side pkt loss



Duplicate pkts for aggregation

Transport Issues:

- **Semantic Gap**
  - Collective Communication is inter-process mode, it carries **messages.**
    - **Messages** have no limitation for size**.**
  - Underlying network delivers **packets.**
    - **Packets** have upper limitation**, MTU.**
  - There needs a **mapping function or mechanism** for intermediate node to **recognize, combinate and compute** on the incoming packets, to form messages.
  - Some affecting issues:
    - Message and packet sending rate.
    - Switch buffer management.
    - In-order or out-of-order delivery.

Will impact transport performance as well as correctness.

- **Blocking and Non-blocking transmission**
  - Collective communication supports two types of communication methods simultaneously
  - Collective operations offloading should adjust to different modes.

One-to-all transmission:

- What we need is **a low latency multi-destination delivery mechanism** (potentially with reliability or at least failure detection).
- IP multicast is more a potential solution (maybe not a good one), but still worth trying.
- IP multicast has been designed to support broadcast related applications like live streaming and video conferencing. However, collective communication based distributed AI workloads are not co-designed with IP multicast right now.
- Collective operations like Bcast and AlltoAll can be augmented by extending IP multicast protocols, as well as other composite operations like reduce-scatter and all-gather.
- Some standardized IP multicast protocols, like PIM and BIER may be able to be extended to support.

Data, Control & Management:

- **In-network primitives**
  - Collective operations: Allreduce, Bcast, AlltoAll…
    - can be called as functions to implement collective communication.
  - In-network primitives:
    - Switch/NIC implementations of collective operations.

- Current in-network primitives is designed for different applications case-by-case, in “chimney-mode”.
- Should have a standard definition on these in-network primitives, to avoid waste in network configurations:
  - Data structure
  - Date type…

- **Topology awareness based on collective operations offloading.**
  - Existing topology awareness algorithms are not co-designed with In-network computing capabilities.
  - Should adjust the algorithms to be suitable for collective operations offloading
    - For example, in clos-based topology, find the most suitable in-network tree node for offloading
    - Switch memory, distance, availability…

Requirements:

- R1. Transport layer **MUST** support RMA function.
- R2. Memory sharing is **RECOMMENDED** to support collective operations.
- R3. The implementation of blocking or non-blocking communication of applications **MUST** be adjusted according to different communication modes.
- R4. The transport layer **MUST** provide appropriate reliability mechanisms to adapt to different communication modes.
- R5. The transport layer **MUST** carry messages that network devices can recognize to complete offloaded collective operations processing.
- R6. The transport layer **MUST** support fallback mechanism, in case network devices are not sufficient for collective operations offloading.
- R7. **MUST** support unified definition and management of data types, data structures supported by network devices considering collectives offloading, and the unified management of resources such as memory of network devices.
- R8. It is **RECOMMENDED** to achieve topology awareness, task scheduling, and allocation in collaboration between the end and network.
- R9. IP multicast protocols **SHOULD** be extended to support collective communication. However, whether to design new multicast algorithms and protocols that are dedicated for collective communication is out of the scope.
- R10. The mechanism of choosing alternative node for implementing collective operations **MUST** be designed, to ensure system robustness and reliability.

Analysis:

- **Other on-going work which may be of interest to the topic:**

- **COINRG(Computing in the Network Research Group):**
  - COIN investigate how network data plane programmability can improve Internet architecture.
  - Broad applications(network functions offloading, machine learning acceleration, in-network caching and in-network control, etc.)
  - Collective communication optimization not necessarily designed with network programmability.

- **SHARP(Scalable Hierarchical Aggregation and Reduction Protocol):**
  - Collective operations offloading based on Infiniband network architecture.
  - Currently, not interoperable with the Internet architecture.

- **UCX(Unified Communication Framework):**
  - Design a framework for collective communication implementation and address the cross-platform functionality and performance portability challenges.

Security and Operational Considerations:

- Collective communication optimization may introduce some security and privacy concerns:

- On one hand, it may impact data confidentiality, integrity, and authentication.
  - Both security-enabled and security-less deployments should be considered.
  - It's suggested to deploy in **limited domains**[RFC8799] at first, since it does not have to pay the penalty of expensive crypto or authority operations. Applications can choose to trust the network within limited domains or they can trust mutually if both of them belong to the same administrator.
  - Extending the technology to the Internet should be designed together with some intrinsic protective actions.

- On the other hand, decrypting and encrypting data on network devices is not only inefficient, but also involves issues such as key management and authorization.

Related Side Meeting in IETF118:

- *https://wiki.ietf.org/meeting/118/sidemeetings*

Title：Collective Communication Optimization(CCO),

Time Schedule: 9th, Nov, Thursday, 14:30 -- 16:00, Palmovka ½

Agenda: https://github.com/CCO-IETF/ietf118-side-meeting

Online Meeting tool (WebEX) :https://ietf.webex.com/meet/ietfsidemeeting2

Welcome for more discussions and contributions.