Network Working Group
Internet-Draft
Intended status: Informational
Expires: 4 September 2025

D. Guzman
Technical University of Munich
D. Trossen
Huawei Technologies
J. Ott
Technical University of Munich

3 March 2025

Multicast Applicability for Distributed Consensus Systems
draft-guzman-multicast-applicability-dcs-00

Abstract

This document questions the applicability of a multicast semantic for the distributed consensus problem. For this, it details the consensus problem and the solutions taken in current systems. It outlines how point-to-multipoint communication arises as part of the consensus solution. Then, it details a peer-centric realization of a permissionless approach, identifying key issues. These issues include communication costs, performance delays, and lack of finality. Hence, it discusses a multicast strawman and its expected improvements.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://david-a-guzman.github.io/draft-guzman-multicast-applicability-dcs/draft-guzman-multicast-applicability-dcs.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-guzman-multicast-applicability-dcs/.

Source for this draft and an issue tracker can be found at https://github.com/david-a-guzman/draft-guzman-multicast-applicability-dcs.

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/.

Copyright Notice

Table of Contents

1. Introduction

Distributed applications often hold a shared state. These applications include file storage, online gaming, and financial and cryptosystems maintaining files, exchanges, and transaction updates.

Changes to a shared state need consensus (or agreement).

Agreement among the majority of participants suffices, according to [VonNewman1956][Fishburn1973].

The problem is disseminating a new (shared) state to all or (at least) the majority.

Additionally, permissionless systems remove the need for central coordination. Decentralized applications like Ethereum and Bitcoin base their consensus on these uncoordinated operations [Guzman2022].

Permissionless systems realize these uncoordinated operations through a peer-centric mechanism. A peer receives, validates, and disseminates a state to a locally maintained, constantly replenished pool of other peers until dissemination dies out.

Communication costs, low performance, and lack of finality are issues resulting from the peer-centric mechanism. Establishing and maintaining those pools (of peers) is messaging intense. The effective data transported over those pools gets duplicated. Latency for establishing and keeping pools results in consensus delay. However, even if the communication cost and latency were low, peers cannot judge the finality of state dissemination.

Multicast seems a natural approach for disseminating. But, there is no such an Internet-wide semantic [Diot2000][Farinacci2024]. Hence, we discuss an overlay strawman for a multicast-based diffusion.

This strawman alleviates the issues of the peer-centric approach. Therefore, this document describes the consensus problem in Section 2 and how current peer-centric deployments solve the state diffusion in Section 3. The last is to identify key issues of the peer-centric mechanism in Section 4, which leads to discussing, in Section 5, how a strawman for multicast improves on these issues. To conclude, Section 6 quantifies those multicast improvements.

2. The Consensus Problem

A peer in a distributed system needs to find consensus for a state change. This state change is local (at the peer). The majority of peers must agree on this (local) change.

Two state disseminations and a local update realize the consensus. A peer disseminates a new state to all or the majority. This majority or all peers locally update their (old) state. Then, all or (at least) the majority disseminate their updated state. Permissionless systems let (only) one peer, not the majority, disseminate its (updated) state.

A peer must know when the state disseminations terminate. Knowing the finality of a (new) state dissemination is key. But, terminating an update dissemination is crucial. Judging the update dissemination permits the consensus finality.

3. Peer-centric Operations

Permissionless systems deploy an uncoordinated peer-centric dissemination. Each peer realizes a pool of peers to disseminate a state [Guzman2022]. The pool of peers is a control plane, and the state is a data plane.

Inbound and outbound relations form this control plane. A peer discovers other peers by lookups and reachability tests. That peer then randomizes those discovered peers before establishing a control plane. When establishing the control plane, a peer negotiates an end-to-end TCP connection, an (overlay) security context, and capabilities. Due to the permissionless strategy, peers constantly replenish this control plane.

Peers disseminate states through these control planes. Each peer sends a state to its control plane (pool) and the peers in this control plane (pool) iteratively to other peers. Peers iteratively diffuse a control plane, such as transactions and blocks, in blockchain networks [Nakamoto2008][Guzman2023]. This peer-centric process is a (randomized) iterative diffusion.

4. Peer-centric Issues

The control plane establishment and maintenance are end-to-end and comprise at least three handshakes. This control plane is a probabilistic sequence (Table 1 in [Guzman2024c]) worsened by unreachability [Guzman2024a], churn, and costly constant replenishment. The number of bytes required to establish and maintain communication relations defines this control plane overhead.

In iterative diffusion, an incoming state keeps arriving even when a peer has already received that state [Guzman2024b]. The number of bytes wasted while disseminating a state determines this data plane overhead.

Disseminating a state results in many stages. Pool establishment, replenishment, and data duplication worsen the delay of these stages. The time required to agree on a single state determines this resulting consensus latency.

In iterative diffusion, those (many) disseminating stages are unknown. Therefore, peers cannot assess when dissemination terminates. As a result, peers cannot judge the finality of consensus.

5. How may Multicast improve on the Peer-centric Issues?

Control Plane Overhead: Dedicated State Replication Points (SRPs) reduce the number of bytes establishing and maintaining end-to-end relations. SRPs placed in clusters of peers establish and maintain peer-to-SRP and SRP-to-SRP relations. SRP-to-SRP is an inter-domain (multipoint) unicast relation, and peer-to-SRP is an intra-domain multicast [Deering1990]. Hence, a peer sets and keeps a single relation instead of a pool of peers.

Peer announcements build peer-to-SRP, and timing these announcements is a heart-beat mechanism (Further details in [Guzman2024c]). SRP-to-SRP builds a (full or partial) mesh and a spanning tree between SRPs. For this, the strawman proposes shared trees to build Source-Specific Multicast (SSM) [rfc3569] or (unicast) tunneling [Bartczak2012].

Data Plane Overhead: An SRP-to-peer relation does not waste bytes while replicating a state. Instead, the multicast relations coordinate a surplus to ensure the majority rule.

SRPs aggregate peer announcements, which results in peer counters. These counters extend a Next Hop Information Base (NHIB) with the number of peers (downstream) an interface. These counters approximate the majority rule. This number is key since, unlike IP multicast [rfc4601], packets are not replicated to all peers but to a dynamic number of peers. The dynamic number is the majority of announced peers, plus some surplus to account for possible packet losses and churn.

Then, a peer queries the majority to the closer SRP to send a state. This peer sends a state to the (nearer) SRP. The SRP forwards toward its interfaces based on the number of peers and the majority rule specified by the sending peer.

Consensus Latency: Replicating state results in three (known) short-timed stages (peer-to-SRP, SRP-to-SRP, and SRP-to-peer).

Consensus Finality: The first replication stage realizes the finality by letting the (closer) SRP guarantee the specified majority rule to the sending peer.

6. Expected Improvements

Empirical data conceive and parametrize models for peer-centric and system-centric views to compare iterative and multicast diffusion. Passive crawls captured this data from approximately 72k Ethereum peers between 2022 and 2023.

Control Plane Overhead: Peers establish a relation with a minimum of 2712 and 127 bytes in iterative and multicast diffusion, respectively [Guzman2024c]. From a system-centric view, each peer establishes a pool of a minimum of fifty other peers in contrast to a single relation (peer-to-SRP) in multicast. Hence, iterative diffusion may need at least 9 GBs and multicast diffusion 8.7 MBs. However, the establishment and maintenance of relations are probabilistic processes. The models in [Guzman2024c](Figures 5 and 7) capture and parameterize this behavior to conclude that multicast needs 790x and 30x fewer bytes during relation establishment and maintenance, respectively.

Data Plane Overhead: An extended exponential (growth) model captures the data duplication of iterative diffusion. The findings for the randomized selection of peers in [Guzman2024b] (Figure 4) derive the likelihood for state duplication. Parametrizing this model results in the observation that multicast-based replication requires 22.44 MB less traffic per state dissemination of 256 bytes (e.g., a transaction).

Consensus Latency: The exponential and the randomized peer selection model (as before) characterized the iterative diffusion, and the three-stage latency model captured the multicast diffusion. Measurements and the key insight of peer concentration in ten well-known infrastructures [Guzman2024c] parametrize both models. The last (detailed in [Guzman2024b]) is to conclude that multicast-based consensus is at least 4x faster than iterative (diffusion) consensus.

Consensus Finality: In multicast-based replication, the commitment to a shared state is part of the control plane.

7. IANA Considerations

This document has no IANA actions.

8. References

8.1. Normative References

[rfc4601] Fenner, Handley, Kouvelas, and Holbrook, "{Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)}", August 2006, <https://www.rfc-editor.org/info/rfc4601>.

8.2. Informative References

[Bartczak2012]
Bartczak and Zwierzykowski, "Performance evaluation of Source Specific Multicast routing protocols for IP networks", 2012, <https://doi.org/10.1109/CSNDSP.2012.6292693>.

[Deering1990]
Deering and Cheriton, "Multicast Routing in Datagram Internetworks and Extended LANs", May 1990.

[Diot2000] Diot, Levine, Lyles, Kassem, and Balensiefen, "Deployment issues for the IP multicast service and architecture", 2000, <https://doi.org/10.1109/65.819174>.

[Farinacci2024]
Farinacci, D., Giuliano, L., McBride, M., and N. Warnke, "Multicast Lessons Learned from Decades of Deployment Experience", Work in Progress, Internet-Draft, draft-ietf-pim-multicast-lessons-learned-04, July 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-pim-multicast-lessons-learned-04>.

[Fishburn1973]
Fishburn, "The Theory of Social Choice", 1973.

[Guzman2022]
Guzman, Trossen, McBride, and Fan, "Insights on Impact of Distributed Ledgers on Provider Networks", 2022, <https://doi.org/10.1007/978-3-031-23495-8_1>.

[Guzman2023]
Guzman, Trossen, and Ott, "If Iterative Diffusion Is The Answer, What Was The Question?", 2023, <https://doi.org/10.1145/3607504.3609288>.

[Guzman2024a]
Guzman, Trossen, Doan, and Ott, "Proliferation of the Service-centric Distributed Consensus Model and its Impact on Ethereum", 2024, <https://doi.org/10.1109/ICBC59979.2024.10634450>.

[Guzman2024b]
Guzman, Trossen, and Ott, "Distributed Consensus Through Network Support", 2024, <https://doi.org/10.23919/IFIPNetworking62109.2024.10619905>.

[Guzman2024c]
Guzman, Trossen, and Ott, "Communication Cost for Permissionless Distributed Consensus at Internet Scale", 2024, <https://doi.org/10.1145/3694809.3700743>.

[Nakamoto2008]
Nakamoto, "{Bitcoin: A Peer-to-Peer Electronic Cash System}", 2008, <https://doi.org/10.1007/s10838-008-9062-0>.

[rfc3569] Bhattacharya, "{An Overview of Source-Specific Multicast (SSM)}", July 2003, <https://www.rfc-editor.org/info/rfc3569>.

[VonNewman1956]
VonNewman, "Probabilistic Logics and the Synthesis of Reliable Organism from Unreliable Components", 1956.

Acknowledgments

We thank the co-authors of the research papers supporting the multicast-based ideas.

Authors’ Addresses

David Guzman
Technical University of Munich
Email: david.guzman@tum.de

Dirk Trossen
Huawei Technologies
Email: dirk.trossen@huawei.com

Joerg Ott
Technical University of Munich
Email: ott@in.tum.de