CAFE: Causal Algorithms for Fault-tolerant Environments

Submitted by: Sinchan Sengupta, Nantes Université
Supervisors: Matthieu Perrin (MCF) and Pascal Molli (Professeur),
Gestion de données distribuées (GDD), LS2N, Nantes Université.

1 Motivations

Our research aims to develop Byzantine-tolerant causal broadcast mechanisms with stronger guarantees against adversarial entities. This work aligns with the objectives of the PEPR **eNSEMBLE** program, particularly the PILOT PC, which focuses on fostering secure, interoperable, and sovereign collaborative environments that enhance trust and digital well-being.

Collaborative editing use case Google Docs popularized real-time collaborative editing, however, its centralized architecture raises privacy concerns. Various methods have been proposed to decentralize collaborative editors [8, 14], yet decentralization introduces new security challenges. For instance, consider a scenario in which a group of users collaborates to rank a list of proposals. If the causal delivery of operations is rigged, the convergence property of decentralized collaborative editors may be compromised, resulting in inconsistent rankings among participants. Consequently, the final published ranking might not reflect a true consensus. A decentralized collaborative editor built on top of a Byzantine-tolerant causal broadcast mechanism can prevent such a catastrophic outcome.

If we consider the enforcement of access control policies in a decentralized collaborative system [10], a causality attack might allow a malicious participant to retaliate even after losing permissions. By reordering specific writes to make them appear as though they occurred before the revocation, the attacker can compromise the integrity of the shared data and undermine the enforcement of these policies.

Insider trading manipulations use case Beyond collaborative editing, Byzantine attacks on causality can severely disrupt various forms of collaboration. In financial systems, such attacks can critically undermine fairness, as seen in insider trading manipulations [11, 1], where the absence of causal enforcement allows adversarial actors to reorder transactions to their advantage. Enforcing causal broadcast ensures that transactions are processed in a fair sequence, preventing manipulations such as artificial price inflation.

2 Research Objectives and Methodology

Objectives System designers have long recognized that capturing causality is essential for achieving consistency in large-scale, fault-tolerant, and highly available systems [6, 13]. In fault-prone environments, communication mechanisms are often designed to enforce modularity by abstracting low-level message exchanges into higher-level constraints. *Causal Broadcast* is one such abstraction, ensuring that a message is only delivered once all causally preceding messages have been received.

Several definitions of Byzantine-tolerant causal broadcast have been proposed in the literature [1, 4, 7], emphasizing the necessity of a FIFO property that restricts the order in which correct processes deliver messages from Byzantine processes. However, these approaches primarily focus on ensuring that correct processes propagate causal dependencies accurately, without preventing Byzantine processes from falsifying or omitting causal relationships. This limitation renders existing solutions inadequate for addressing the use cases presented in the previous section, where adversarial behavior exploits gaps in causality enforcement.

The CAFE project aims to specify, implement, and evaluate a new Byzantine-tolerant causal broadcast abstraction tailored to these challenges. Our objectives are threefold:

1. **Specification:** We will define a new Byzantine Causal Broadcast specification that prevents Byzantine processes from concealing causal dependencies from correct processes.

2. **Algorithm Design and Proofs:** We will develop a new algorithm that adheres to this specification and formally prove its correctness.

3. **Implementation and Evaluation:** We will integrate our new primitive into the collaborative editor MUTE [8] and analyze how it affects the performance of collaborative editors.

Approach Specifying the expected behavior of such a broadcast abstraction is challenging, as correctness cannot be defined solely in terms of execution traces. In the use cases discussed above, the Byzantine behaviors we aim to prevent do not only arise from the execution of the broadcast itself but also depend on broader system-level expectations and intended usage. To address this, we aim to explore how Knowledge Theory [2] can provide a framework for specifying the semantic causal relationships that Byzantine processes must respect, ensuring that their actions align with the expected information flow of the system.

On the algorithmic side, we aim to build on our previous work on privacy-preserving atomic registers [5], to explore the use of secret sharing schemes [12] to regulate how and when Byzantine processes gain access to the content of broadcasted messages. Our approach introduces a phased timeline to control the flow of information, ensuring that correct processes learn about the existence of a message m before Byzantine processes gain access to its content. During this intermediate phase, correct processes have the flexibility to decide whether or not to enforce a causal link between m and messages broadcast by Byzantine processes. Crucially, once Byzantine processes learn the content of m, correct processes will always be able to infer the correct causality relationship. This progressive disclosure mechanism prevents adversarial nodes from arbitrarily manipulating causality while preserving the integrity of the broadcast. Our work will investigate how this phased approach can strengthen Byzantine-tolerant causal broadcast and enhance reliable communication in collaborative environments.

Experimental Study and Evaluation Strategy The experimental aims to answer two important questions:

Q1: Are users able to detect convergence failure due the presence of Byzantine participants manipulating causality in a collaborative editing?

Q2: How the usage of Byzantine-tolerant causal broadcast mechanism affect the delay in real-time collaborative editing.

To address Q1, we plan to modify the real-time collaborative editor MUTE [8] by introducing Byzantine participants. Selected users will perform a sorting task to establish a ranking, following the methodology described in [3]. The Byzantine participants will deliberately attack the collaborative editing session, potentially disrupting convergence. The objective of the experiment is to determine whether participants can detect the presence of Byzantine behavior before the task concludes. If users cannot detect byzantine attack, then the underlying system has to be trusted.

To address question Q2, we will adopt the experimental protocol proposed in [3]. As a first step, we will integrate Byzantine Causal Broadcast into MUTE, which may introduce additional operational overhead affecting real-time collaboration due to extra verification and message ordering steps. We will measure key performance metrics such as document convergence time, message overhead (latency and bandwidth impact), and state synchronization accuracy. We will then conduct user studies to observe how participants adapt their collaboration strategies in response to delays, assessing their sensitivity to increased latency and its effect on coordination and task performance. This will provide insights into the trade-offs between Byzantine resilience and usability in collaborative environments.

3 Project Consortium and Duration

CAFE aims to contribute to both the theoretical Distributed Computing and CSCW communities. To achieve this, we leverage the expertise of Matthieu Perrin, who will oversee the development of the causal broadcast algorithm, and Pascal Molli, who has extensive experience in real-time collaborative editors and has published in the CSCW scientific community [9].

We plan to collaborate with the COAST team in Nancy for the experimental studies, to leverage their expertise in MUTE and has previously conducted similar user studies.

CAFE is proposed to run from May 2025 to April 2027, following this timeline: i) Specification Development – Defining the Causal Broadcast abstraction consistent with Knowledge of Preconditions (May 2025 – October 2025). ii) Algorithm Design and Correctness – Designing the algorithm and proving its correctness (November 2025 – May 2026). iii) Experimental Validation – Implementing and testing the approach in MUTE (June 2026 – April 2027).

We plan to publish scientific results in conferences and journals of the distributed computing (PODC), and CSCW communities. We will take part into the animations of the PEPR **eNSEMBLE** program.

4 References

[1] A. Auvolat, D. Frey, M. Raynal, and F. Taïani. Byzantine-tolerant causal broadcast. *Theoretical Computer Science*, 885:55–68, 2021.

[2] C. Dwork and Y. Moses. Knowledge and common knowledge in a byzantine environment: crash failures. *Information and Computation*, 88(2):156–186, 1990.

[3] C.-L. Ignat, G. Oster, O. Fox, V. L. Shalin, and F. Charoy. How do user groups cope with delay in real-time collaborative note taking. In *ECSCW 2015: Proceedings of the 14th European Conference on Computer Supported Cooperative Work, 19-23 September 2015, Oslo, Norway*, pages 223–242. Springer, 2015.

[4] V. Kowalski, A. Mostefaoui, and M. Perrin. Causal mutual byzantine broadcast. In *Proceedings of the 2024 Workshop on Advanced Tools, Programming Languages, and PLatforms for Implementing and Evaluating algorithms for Distributed systems*, pages 1–8, 2024.

[5] V. Kowalski, A. Mostéfaoui, M. Perrin, and S. Sengupta. Byzantine-tolerant privacy-preserving atomic register. In *Proceedings of the 26th International Conference on Distributed Computing and Networking*, ICDCN '25, page 201–210, New York, NY, USA, 2025. Association for Computing Machinery.

[6] W. Lloyd, M. J. Freedman, M. Kaminsky, and D. G. Andersen. Don't settle for eventual: scalable causal consistency for wide-area storage with cops. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, pages 401–416, 2011.

[7] A. Misra and A. D. Kshemkalyani. Byzantine fault-tolerant causal ordering. In *24th International Conference on Distributed Computing and Networking, ICDCN 2023, Kharagpur, India, January 4-7, 2023*, pages 100–109. ACM, 2023.

[8] M. Nicolas, V. Elvinger, G. Oster, C.-L. Ignat, and F. Charoy. Mute: A peer-to-peer web-based real-time collaborative editor. In *ECSCW 2017-15th European Conference on Computer-Supported Cooperative Work*, volume 1, pages 1–4. EUSSET, 2017.

[9] G. Oster, P. Urso, P. Molli, and A. Imine. Data consistency for P2P collaborative editing. In P. J. Hinds and D. Martin, editors, *Proceedings of the 2006 ACM Conference on Computer Supported Cooperative Work, CSCW 2006, Banff, Alberta, Canada, November 4-8, 2006*, pages 259–268. ACM, 2006.

[10] P.-A. Rault, C.-L. Ignat, and O. Perrin. Distributed access control for collaborative applications using crdts. In *Proceedings of the 9th Workshop on Principles and Practice of Consistency for Distributed Data*, PaPoC '22, page 33–38, New York, NY, USA, 2022. Association for Computing Machinery.

[11] M. K. Reiter and K. P. Birman. How to securely replicate services. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 16(3):986–1009, 1994.

[12] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[13] M. Shapiro, N. Preguiça, C. Baquero, and M. Zawirski. Conflict-free replicated data types. In *Stabilization, Safety, and Security of Distributed Systems: 13th International Symposium, SSS 2011, Grenoble, France, October 10-12, 2011. Proceedings 13*, pages 386–400. Springer, 2011.

[14] S. Weiss, P. Urso, and P. Molli. Logoot: A scalable optimistic replication algorithm for collaborative editing on p2p networks. In *2009 29th IEEE International Conference on Distributed Computing Systems*, pages 404–412. IEEE, 2009.