When consensus meets the Edge: Challenges, Algorithms, and Opportunities

Carmine Colarusso
Dept. of Engineering
University of Sannio
Benevento, Italy
ccolarusso@unisannio.it

Ida Falco
Dept. of Engineering
University of Sannio
Benevento, Italy
ifalco@unisannio.it

Eugenio Zimeo
Dept. of Engineering
University of Sannio
CINI Smart Cities Lab
Benevento, Italy
zimeo@unisannio.it

***Abstract*—Smart cities increasingly rely on distributed Edge-Cloud infrastructures to support latency-sensitive IoT services such as traffic monitoring, smart retail, and environmental monitoring. Ensuring data consistency in these environments is challenging due to network heterogeneity, intermittent connectivity, and geographically distributed nodes. Traditional consensus protocols, such as Paxos and WPaxos, provide strong consistency but introduce significant latency in Edge scenarios.**

This paper presents EdgeCloudWPaxos, a consensus protocol designed for smart-city Edge-Cloud systems. The protocol enables low-latency local operations under weak consistency while delegating global reconciliation to the Cloud. Experimental results show that EdgeCloudWPaxos improves latency and throughput under heterogeneous network conditions while preserving fault tolerance and convergence.

***Index Terms*—distributed consensus, non-linearized writes, asynchronous, reconciliation, Edge-Cloud.**

I. Introduction

Smart cities rely on large-scale distributed infrastructures composed of IoT devices, Edge nodes, and Cloud services to support applications such as intelligent transportation, environmental monitoring, and smart retail. In these scenarios, Edge computing reduces latency by processing data close to where it is generated. However, maintaining consistency across geographically distributed Edge nodes remains challenging due to variable network latency, intermittent connectivity, and concurrent accesses to replicated data. Traditional consensus protocols such as Paxos [1] and WPaxos [2] provide strong consistency guarantees through synchronous coordination and quorum-based replication. These approaches often suffer from high latency and throughput degradation in Edge-Cloud scenarios, where network delays and temporary disconnections are common. In particular, geographically distributed deployments make global linearization expensive, as each operation may require one or more Edge-Cloud round trips (Figure 1).

To address these challenges, EdgeCloudWPaxos, a Cloud-assisted consensus approach for Edge environments, is presented. The proposed design enables low-latency local operations under weak consistency while delegating global reconciliation and asynchronous strong-state convergence to the Cloud. Many smart-city workloads exhibit strong locality properties. Most operations involve data generated and consumed within the same geographical region, while only a small subset is shared across multiple Edge nodes. This observation motivates optimistic approaches that allow local operations while tolerating temporary inconsistencies and infrequent conflicts. By exploiting data locality and low conflict probability assumptions, the system reduces synchronization overhead and maintains responsiveness even under unstable networks.

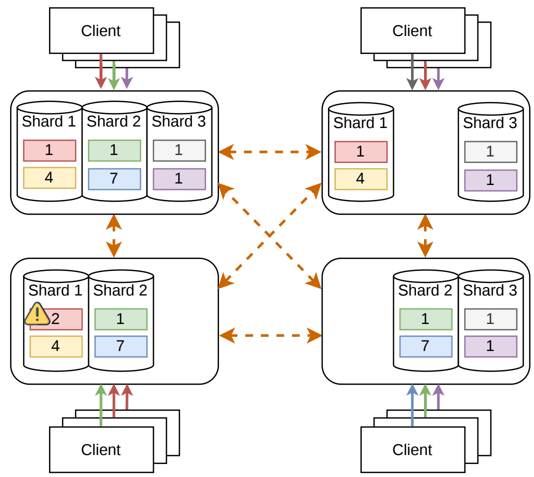


Fig. 1: Distributed synchronization.

II. EdgeCloudWPaxos

EdgeCloudWPaxos is a Cloud-assisted consensus approach designed for geographically distributed Edge environments. The protocol aims to reduce the impact of network latency and temporary disconnections by allowing Edge nodes to process operations locally while delegating global reconciliation to the Cloud [3]. The system consists of multiple Edge nodes connected to a central Cloud component. Each Edge node stores a subset of replicated data shards and serves client requests locally. Unlike traditional consensus protocols, direct Edge-to-Edge communication is not required. Instead, the Cloud acts as a coordination and reconciliation point, maintaining the mapping between shards and replicas and mediating synchronization among nodes. The protocol adopts an optimistic execution model. Operations are executed locally at the Edge under weak consistency semantics, enabling low-latency responses independently of the Edge-Cloud round-trip time. This

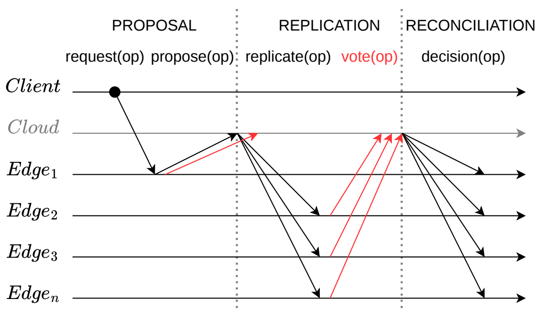
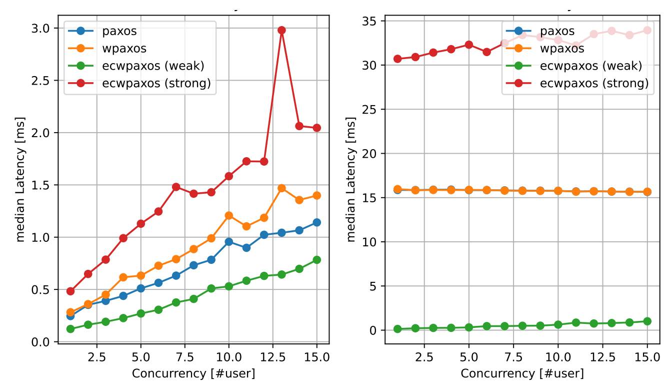


Fig. 2: Phases of the algorithm.



(a) 0.5 [s] RTT network.

(b) 15 [s] RTT Network.

Fig. 4: Median latency varying cuncurrency.

design is motivated by the observation that many smart-city workloads exhibit strong locality, with most accesses involving data that are rarely shared across geographically distant nodes. As a consequence, conflicts are expected to occur infrequently. When strong consistency is required, operations are synchronized through the Cloud, which acts as the global acceptor and linearization point. Edge replicas execute the operation locally and return a vote to the Cloud. Once all required votes are collected, the Cloud decides whether to commit or abort the operation and propagates the final decision back to the replicas (Fig. 2). In case of temporary disconnections, Edge nodes continue operating locally in weak-consistency mode and reconcile their state once connectivity is restored.

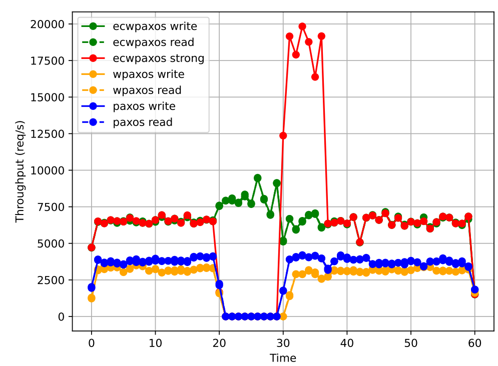


Fig. 5: Network interruption.

III. Experimental Evaluation

The experimental evaluation was conducted on a distributed setup composed of three Edge nodes and one Cloud node deployed on virtual machines. Different network conditions were emulated by introducing controlled latencies between the Edge and the Cloud in order to reproduce realistic smart-city wide-area environments.

Results show that EdgeCloudWPaxos significantly improves both throughput and latency compared to traditional consensus protocols such as Paxos and WPaxos. In particular, the protocol maintains stable throughput even under high network latency, while Paxos and WPaxos rapidly saturate due to the synchronous coordination required for each operation (Fig. 3). Latency analysis further highlights the benefits of local execution. Weak-consistency operations in EdgeCloudWPaxos exhibit very low latency and remain largely unaffected by Edge-Cloud round-trip delays. Conversely, Paxos and WPaxos show latency growth proportional to the network RTT. Strong-consistency operations in EdgeCloudWPaxos incur an additional synchronization cost (Fig. 4). The evaluation also confirms the resilience of the proposed approach under heterogeneous network conditions and temporary disconnections. During network disruptions, Edge nodes continue serving local operations in weak-consistency mode and reconcile their state once connectivity is restored (Fig. 5). These results demonstrate that EdgeCloudWPaxos effectively decouples local responsiveness from global synchronization costs, making it particularly suitable for latency-sensitive smart-city applications deployed over geographically distributed Edge infrastructures.

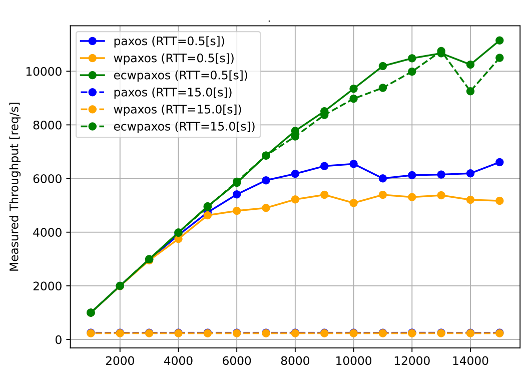


Fig. 3: Throughput saturation.

References

[1] L. Lamport, "Paxos made simple," *ACM SIGACT News (Distributed Computing Column) 32, 4 (Whole Number 121, December 2001)*, pp. 51–58, 2001.

[2] A. Ailijiang, A. Charapko, M. Demirbas, and T. Kosar, "Wpaxos: Ruling the archipelago with fast consensus," *arXiv preprint arXiv:1703.08905*, 2017.

[3] C. Colarusso, I. Falco, and E. Zimeo, "Business continuity of cloud-based iot applications through a seamless continuum," *Internet of Things*, p. 101723, 2025.