A Decentralized Architecture for Scientific Document Storage and Retrieval Using QUIC, Erasure Coding, and DHT-Based Overlays

Carlos Giongo[1], Adriano Fiorese[1]

[1]Departamento de Ciência da Computação (DCC)
Programa de Pós-Graduação em Computação Aplicada (PPGCAP)
Universidade Do Estado de Santa Catarina (UDESC) – Joinville, SC – Brazil

carlos.giongo@edu.udesc.br, adriano.fiorese@udesc.br

Abstract. *Decentralized architectures can reduce single points of failure in scientific document repositories. However, semantic retrieval in peer-to-peer environments introduces challenges in transport efficiency and data availability. This paper presents a Kademlia-like Distributed Hash Table (DHT) architecture with explicit separation between control and data planes. Overlay maintenance uses User Datagram Protocol (UDP), while data-intensive transfers use Quick UDP Internet Connections (QUIC). The architecture combines local Approximate Nearest Neighbor (ANN) indexing with Erasure Coding (EC) for fault-tolerant storage. Experimental results show that QUIC reduces latency and variability in publication and retrieval operations compared to Transmission Control Protocol (TCP)-based baselines.*

Resumo. *Arquiteturas descentralizadas reduzem pontos únicos de falha em repositórios científicos, mas a recuperação semântica em ambientes peer-to-peer impõe desafios de transporte e disponibilidade. Este artigo apresenta uma arquitetura baseada em Distributed Hash Table (DHT) do tipo Kademlia, com separação entre plano de controle e plano de dados. A manutenção do overlay utiliza User Datagram Protocol (UDP), enquanto transferências intensivas em dados empregam Quick UDP Internet Connections (QUIC). A arquitetura integra indexação Approximate Nearest Neighbor (ANN) local e Erasure Coding (EC) para tolerância a falhas. Resultados experimentais indicam redução de latência e variabilidade em comparação a baselines baseados em Transmission Control Protocol (TCP).*

1. Introduction

Scientific document repositories remain largely centralized, creating operational bottlenecks and single points of failure. Decentralized infrastructures offer an alternative, and Distributed Hash Tables (DHTs) are a core building block for peer-to-peer (P2P) networks because nodes can retrieve data with only partial system knowledge [Zhang et al. 2013]. DHTs such as Chord and Kademlia provide efficient exact-match lookups through consistent hashing and $O(\log N)$ routing complexity [Maymounkov and Mazières 2002]. However, semantic retrieval depends on similarity search over embeddings, which conflicts with hash-based placement because hashing removes semantic locality [Zhu and Hu 2007].

Retrieving scientific documents in decentralized environments introduces storage and transport challenges. Relevant documents may be distributed across peers without a global index, while dense vectors increase indexing and communication costs. Naive P2P search often relies on flooding [Tran and Nguyen 2008], and mapping multidimensional data into the DHT identifier space may cause index fragmentation and maintenance overhead [Zhao et al. 2015]. Data availability is also critical: replication requires $M+1$ copies to tolerate M failures [Li and Xiao 2021], where Erasure Coding (EC) is more storage efficient but adds encoding, decoding, and reconstruction traffic [Balaji et al. 2018].

This paper proposes a decentralized architecture for semantic retrieval of scientific documents. The design separates a UDP-based control plane for DHT maintenance from a QUIC-based data plane for reliable, multiplexed document transfer [Maymounkov and Mazières 2002, Ortega and Monserrat 2020]. It also integrates EC for storage resilience and local Approximate Nearest Neighbor (ANN) indexing, treating semantic search as a service layer rather than as a DHT routing primitive.

The main contributions are: (i) a hybrid transport architecture with UDP for overlay maintenance and QUIC for data transfer; (ii) an experimental comparison between TCP and QUIC for PUT and GET operations; and (iii) a service-oriented semantic retrieval model using local ANN indexing atop a resilient DHT overlay. The scope is limited to infrastructure behavior; no claim is made regarding semantic ranking metrics such as Precision@k, Recall@k, or MRR. The remainder presents background, architecture, experimental setup, results, and future work.

2. Background

DHTs provide a structured overlay that partitions data ownership among peers. Protocols such as Chord and Kademlia assign identifiers to nodes and data keys in a shared identifier space [Maymounkov and Mazières 2002]. In Kademlia, routing uses an XOR-based distance metric and reaches the responsible node in $O(\log N)$ steps without global topology knowledge [Maymounkov and Mazières 2002]. This is effective for exact-match lookup, but DHTs do not natively support full-text or semantic queries because hash-based placement disperses semantically related objects [Zhu and Hu 2007].

Earlier P2P search systems used distributed inverted indexes, where peers store posting lists for subsets of terms [Li et al. 2003]. However, multi-term queries may require large posting-list transfers, and maintaining indexes under churn imposes substantial overhead [Li et al. 2003]. Modern retrieval often uses the Vector Space Model (VSM), where documents and queries are high-dimensional vectors and relevance is estimated by similarity measures such as cosine similarity [Wong et al. 1987]. ANN methods, including HNSW, support efficient top-k retrieval in centralized settings, but distributing such structures over a DHT is difficult because vector exchange and index traversal move the bottleneck from CPU to network communication.

Taking about the transport protocols, TCP provides reliability but suffers from head-of-line blocking and connection setup latency [Ortega and Monserrat 2020]. UDP avoids connection setup overhead and is suitable for Kademlia-style RPCs [Maymounkov and Mazières 2002], but it lacks reliability and congestion control for large payloads. QUIC bridges this gap by operating over UDP while providing TLS 1.3, reliable delivery, congestion control, stream multiplexing, and 0-RTT resumption

[Ortega and Monserrat 2020].

Related decentralized storage systems, including IPFS-based approaches, also explore content addressing and distributed retrieval. However, this work focuses on how control/data plane separation and QUIC-based transfers affect publication and reconstruction operations in a DHT-based scientific repository with EC and local ANN services.

3. System Architecture

The proposed architecture separates topology maintenance, data transfer, and semantic retrieval. It has four components: (i) a Kademlia-like DHT overlay; (ii) a UDP control plane for peer discovery, routing table updates, liveness detection, and keep-alive probes; (iii) a QUIC data plane for transferring embeddings, ANN results, and encoded fragments; and (iv) local indexing and storage services, including ANN search and EC-based persistence.

The control/data plane separation is deliberate. DHT maintenance requires low-latency messages with small payloads, while semantic retrieval and reconstruction require reliable transfer of vectors and file fragments. UDP is used in the control plane because Kademlia-like overlays rely on parallel asynchronous queries, where failures can be masked by contacting multiple peers concurrently [Maymounkov and Mazières 2002]. QUIC is used in the data plane because each logical operation can be mapped to an independent stream, so packet loss or congestion in one stream does not stall the others [Ortega and Monserrat 2020].

Semantic retrieval is implemented locally at each peer. Each node maintains a vector index over the embeddings associated with its stored documents or fragments. Upon receiving a natural-language query, the querying peer computes the query embedding, selects a bounded set of candidate peers through the DHT overlay, and sends the query vector through the data plane. Each candidate peer executes ANN search locally and returns compact top-k candidates, avoiding a global ANN structure in the P2P overlay.

To ensure availability under node failures while avoiding full replication, the persistence layer integrates EC. Each document is encoded into n fragments, such that any subset of k fragments is sufficient to reconstruct the original content, following the usual (n, k) erasure-coded storage model [Balaji et al. 2018]. During PUT, the publisher encodes the document, locates destination peers through the overlay, and transfers shards using QUIC streams. During GET, the querying node discovers shard holders and retrieves fragments in parallel until k fragments are obtained. EC cost is therefore reflected in encoding/decoding overhead and multi-source retrieval traffic.

4. Experimental Methodology

The experiments quantify the impact of transport-layer choices and control/data plane separation on end-to-end completion time, latency distribution, and variability during publication and retrieval.

The setup uses a fully emulated P2P network deployed with Mininet [Lantz et al. 2010]. A total of 100 logical peers were instantiated on a single physical machine with an Intel i7 13th generation CPU and $16GB$ RAM. No additional delay, loss, or bandwidth shaping was configured; thus, measured differences mainly reflect protocol behavior under shared-host contention. This controlled scenario does not model wide-area latency, heterogeneous bandwidth, NAT behavior, firewall restrictions, or distributed failures.

The corpus comprises 150 scientific documents, including master's theses and doctoral dissertations collected during the first semester of 2025 from the official repositories of the Graduate Program in Applied Computing at UDESC (Joinville) [Universidade do Estado de Santa Catarina 2026] and the Graduate Program in Computer Science at UFSC (Florianópolis) [Universidade Federal de Santa Catarina 2026], totaling approximately $1GB$ of PDF documents.

The system uses EC with parameters $(n = 14, k = 10)$, where each document is split into 10 data fragments and 4 parity fragments. A global redundancy constant $K = 8$ was also applied, resulting in an effective storage footprint of approximately:

$$1\,\text{GB} \times 1.4 \times 8 = 11.2\,\text{GB} \tag{1}$$

Three configurations were evaluated: I) Control TCP + Data TCP, used as baseline; II) Control UDP + Data TCP; and III) Control UDP + Data QUIC. Each configuration executed full PUT and GET cycles over the corpus. The choice of 150 documents was motivated by resource contention, since all peers ran on the same physical processor.

5. Results and Analysis

Figure 1 presents the distribution of per-file publication times. The baseline configuration (Control TCP + Data TCP) exhibits the highest median latency and the widest spread, with outliers exceeding 100 seconds per file. Replacing TCP with UDP in the control plane reduces both median publication time and variability, since peer discovery and overlay maintenance avoid connection establishment overhead.

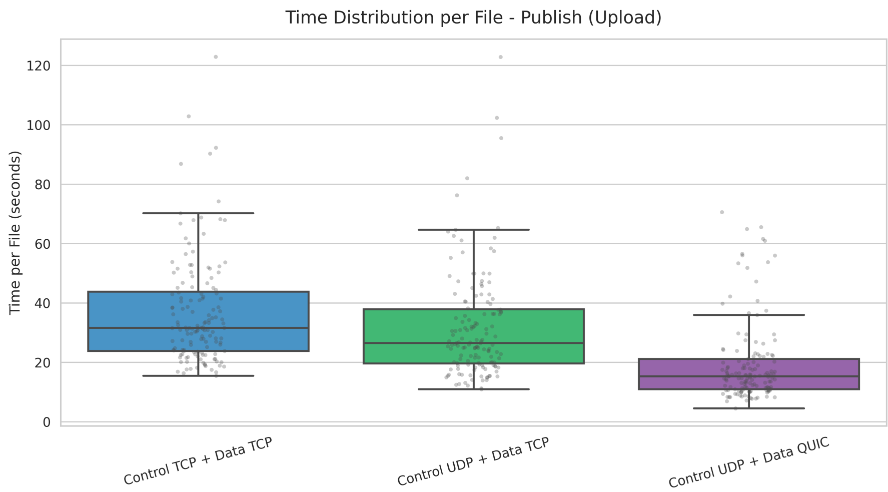

Figure 1. Per-file publication time distribution by transport configuration

The largest improvement occurs in the Control UDP + Data QUIC configuration. QUIC enables parallel shard uploads through independent streams, preventing slow or congested transfers from blocking others. In accumulated publication time for all 150 documents, Control UDP + Data QUIC reduces total publication time by approximately 38.9% compared with Control UDP + Data TCP, and by more than 46% compared with the TCP-only baseline.

Similar trends appear during retrieval. Figure 2 shows that the TCP-only configuration presents higher variability and a heavier tail, reflecting sequential transfer constraints during shard reconstruction. UDP improves shard holder discovery, while QUIC provides the best overall performance by allowing concurrent shard downloads until the required k fragments are obtained.

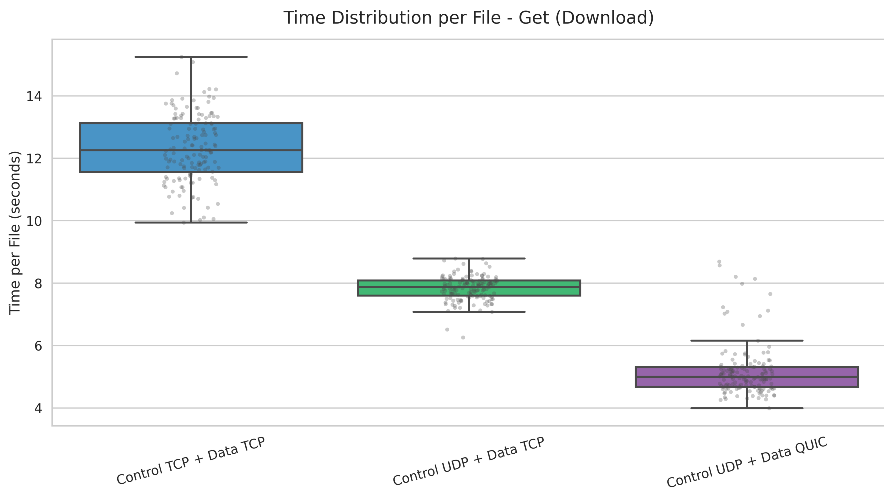

Figure 2. Per-file retrieval time distribution during GET operations

For accumulated retrieval time, the QUIC-based data plane reduces total retrieval time by approximately 26.3% compared with Control UDP + Data TCP. This confirms that transport-layer multiplexing is effective in multi-source reconstruction scenarios induced by EC. The results support the intuition that UDP is appropriate for lightweight overlay maintenance, while QUIC is better suited to concurrent data-intensive transfers.

6. Conclusion and Future Work

This paper presented a decentralized architecture for scientific document storage and semantic retrieval, combining a UDP-based Kademlia-like control plane, a QUIC-based data plane, local ANN indexing, and EC for fault-tolerant storage.

Results show that separating control and data planes improves publication and retrieval performance. QUIC reduces latency and variability under concurrent shard transfers, while UDP improves overlay responsiveness. These gains are transport-driven and do not require changes to DHT routing logic or semantic indexing algorithms.

The evaluation was conducted in a controlled Mininet environment, which improves reproducibility but limits external validity because all peers ran on a single host and no delay, loss, bandwidth shaping, NAT, or firewall constraints were modeled. Future studies should evaluate real or hybrid deployments, compare the proposal with state-of-the-art decentralized storage systems such as IPFS-based solutions, and analyze EC costs in greater depth. Future work also includes adaptive EC strategies, availability-aware peer selection, and semantic-aware routing mechanisms.

References

Balaji, S. B., Krishnan, M. N., Vajha, M., Ramkumar, V., Sasidharan, B., and Kumar, P. V. (2018). Erasure coding for distributed storage: an overview. *Science China Information Sciences*, 61(10):100301.

Lantz, B., Heller, B., and McKeown, N. (2010). A network in a laptop: rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Hotnets-IX, New York, NY, USA. Association for Computing Machinery.

Li, J., Loo, B. T., Hellerstein, J. M., Kaashoek, M. F., Karger, D. R., and Morris, R. (2003). On the feasibility of peer-to-peer web indexing and search. In Kaashoek, M. F. and Stoica, I., editors, *Peer-to-Peer Systems II*, pages 207–215, Berlin, Heidelberg. Springer Berlin Heidelberg.

Li, Z. and Xiao, C. (2021). Er-store: A hybrid storage mechanism with erasure coding and replication in distributed database systems. *Scientific Programming*, 2021(1):9910942.

Maymounkov, P. and Mazières, D. (2002). Kademlia: A peer-to-peer information system based on the XOR metric. In Druschel, P., Kaashoek, F., and Rowstron, A., editors, *Peer-to-Peer Systems*, volume 2429 of *Lecture Notes in Computer Science*, pages 53–65, Berlin, Heidelberg. Springer.

Ortega, V. and Monserrat, J. F. (2020). Semantic distributed data for vehicular networks using the Inter-Planetary File System. *Sensors*, 20(22):6404.

Tran, D. A. and Nguyen, K. (2008). Multidimensional information retrieval in peer-to-peer networks. In *2008 IEEE International Symposium on Parallel and Distributed Processing*, pages 1–5.

Universidade do Estado de Santa Catarina (2026). Dissertações – programa de pós-graduação em computação aplicada (PPGCA). Acessado em: 07 jan. 2026.

Universidade Federal de Santa Catarina (2026). Programa de pós-graduação em ciência da computação (PPGCC). Acessado em: 07 jan. 2026.

Wong, S. K., Ziarko, W., Raghavan, V. V., and Wong, P. C. (1987). On modeling of information retrieval concepts in vector spaces. *ACM Trans. Database Syst.*, 12(2):299–321.

Zhang, H., Wen, Y., Xie, H., and Yu, N. (2013). *Distributed Hash Table: Theory, Platforms and Applications*. SpringerBriefs in Computer Science. Springer New York, New York, NY.

Zhao, F., Liu, J., Zhou, J., Jin, H., and Yang, L. T. (2015). Ls-ams: An adaptive indexing structure for realtime search on microblogs. *IEEE Transactions on Big Data*, 1(4):125–137.

Zhu, Y. and Hu, Y. (2007). Efficient semantic search on dht overlays. *Journal of Parallel and Distributed Computing*, 67(5):604–616.