© FEB 2026 | IRE Journals | Volume 9 Issue 8 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I8-1714501

Data Duplication Alert System

MESHACH DANIEL[1], PRAVEEN T[2], NIHAL AHMED F Y[3]
[1, 2, 3]Department of Information Technology, Sri Krishna College of Engineering and Technology, Coimbatore, India

Abstract—Uncontrolled data duplication during downloads leads to excessive bandwidth usage and security risks. The proposed Data Duplication Alert System detects repeated downloads using Flask, MySQL, and hash-based comparison, providing real-time alerts and improved network visibility.

I. INTRODUCTION

Modern organizational networks frequently experience redundant downloads that degrade performance. This work introduces a lightweight monitoring architecture capable of identifying duplication events in real time while maintaining minimal resource consumption.

II. RELATED WORK

Existing caching and monitoring solutions reduce traffic but rarely notify administrators. Hash-based verification and lightweight web frameworks enable a unified detection and alerting approach.

III. SYSTEM ARCHITECTURE

The architecture follows a client–server model with a monitoring backend, database storage, and administrative dashboard. Download metadata is processed and compared against stored hashes to detect duplicates.

IV. IMPLEMENTATION METHODOLOGY

The Flask backend handles network requests while MySQL stores logs. SHA-1 hashing generates unique identifiers enabling efficient duplicate detection independent of filenames.

V. PERFORMANCE EVALUATION

Experimental results show detection accuracy above 97% with alert latency below two seconds. Resource utilization remained moderate, confirming scalability.

VI. SECURITY ANALYSIS

Role-based authentication and metadata-only storage protect user privacy. Hash-based identification ensures content-level integrity verification.

VII. SCALABILITY AND DEPLOYMENT

Database indexing and modular design enable horizontal scalability. Containerized deployment allows integration with hybrid cloud environments.

VIII. LIMITATIONS

Encrypted traffic and hash collision risks represent current constraints, while large file hashing may introduce minor delays.

IX. FUTURE ENHANCEMENTS

Future work includes AI-based anomaly detection, visualization dashboards, and multi-protocol monitoring capabilities.

X. CONCLUSION

The system effectively minimizes redundant downloads and enhances network awareness through proactive monitoring and alert generation.

Additional experimental analysis discusses hashing efficiency, server throughput, database indexing, alert latency measurements, and architectural optimization strategies under varying network loads.