AVOIDING DATA CORRUPTION IN DROP COMPUTING MOBILE NETWORKS

[1] KUMPATLA KRISHNA KISHORE,

, [2] Mr. Naga Srinivasa Rao

[1]PG STUDENT ,DEPT OF MCA

[2] **Asst. Prof**, Dept of MCA

SREE KONASEEMA BHANOJI RAMARS P.G. COLLEGE (AMALAPURAM)

ABSTRACT

Drop computing is a network paradigm that aims to address the issues of the mobile cloud computing model, which has started to show limitations especially since the advent of the Internet of Things and the increase in the number of connected devices. In drop computing, nodes are able to offload data and computations to the cloud, to edge devices, or to the social-based opportunistic network composed of other nodes located nearby. In this paper, we focus on the lowest layer of drop computing, where mobile nodes offload tasks and data to and from each other through close-range protocols, based on their social connections. In such a scenario, where the data can circulate in the mobile network on multiple paths (and through multiple other devices), consistency issues may appear due to data corruption or malicious intent. Since there is no central entity that can control the way information is spread and its correctness, alternative methods need to be employed. In this paper, we propose several mechanisms for ensuring data consistency in drop computing, ranging from a rating system to careful analysis of the data received. Through thorough experimentation, we show that our proposed solution is able to maximize the amount of correct (i.e., uncorrupted) data exchanged in the network, with percentages as high as 100%.

Keywords: malicious intent, data corruption, drop computing

1 INTRODUCTION:

In drop computing mobile networks, where devices frequently connect and disconnect from the network due to factors such as mobility, intermittent connectivity, and limited bandwidth, ensuring data integrity becomes a significant challenge. Data corruption can occur during transmission, storage, or processing, leading to loss of critical information, compromised network performance, and degraded user experience. Therefore, the primary objective of this project is to develop robust strategies and mechanisms to mitigate the risk of data corruption in drop computing mobile networks.

Key Challenges:

Intermittent Connectivity: Mobile devices in drop computing environments experience intermittent connectivity as they move between network zones or encounter network disruptions. This intermittent connectivity increases the likelihood of data corruption during transmission.

Limited Bandwidth: Drop computing mobile networks often operate with limited bandwidth, leading to packet loss, congestion, and retransmission delays. These factors contribute to the vulnerability of data to corruption during transmission.

Dynamic Network Topology: The dynamic nature of drop computing networks, characterized by frequent device mobility and network reconfiguration, poses challenges for maintaining consistent data paths and ensuring reliable communication.

Resource Constraints: Mobile devices and network nodes in drop computing environments have limited computational resources and energy reserves, limiting the complexity and overhead of data integrity mechanisms.

Project Objectives:

Develop Reliable Data Transmission Protocols: Design and implement communication protocols that prioritize reliability and data integrity, incorporating error detection, correction, and recovery mechanisms tailored to drop computing mobile networks.

Implement Robust Error Detection Mechanisms: Develop algorithms and techniques for detecting data corruption during transmission, storage, and processing, leveraging checksums, cyclic redundancy checks (CRC), and forward error correction (FEC) codes.

Ensure Seamless Data Recovery: Design fault-tolerant mechanisms for recovering from data corruption events, including packet retransmission, redundant data storage, and dynamic rerouting strategies to minimize data loss and disruption.

Optimize Resource Utilization: Develop resource-efficient data integrity solutions that minimize computational overhead, energy consumption, and bandwidth utilization, ensuring compatibility with the constraints of drop computing mobile networks.

Validate Performance and Scalability: Evaluate the effectiveness and scalability of the proposed solutions through simulations, experiments, and real-world deployments, assessing factors such as data throughput, latency, reliability, and network overhead.

Expected Outcomes:

A comprehensive understanding of the challenges and requirements for avoiding data corruption in drop computing mobile networks.

Novel strategies and mechanisms for ensuring data integrity during transmission, storage, and processing in dynamic network environments.

Validation of proposed solutions through experimentation and performance evaluation, demonstrating their effectiveness and scalability.

Guidelines and best practices for implementing data integrity mechanisms in drop computing mobile networks, benefiting network operators, device manufacturers, and end-users.

2 LITERATURE SURVEY:

Mobile cloud computing for computation offloading: Issues and challenges

Despite the evolution and enhancements that mobile devices have experienced, they are still considered as limited computing devices. Today, users become more demanding and expect to execute computational intensive applications on their smartphone devices. Therefore, Mobile Cloud Computing (MCC) integrates mobile computing and Cloud Computing (CC) in order to extend capabilities of mobile devices using offloading techniques. Computation offloading tackles limitations of Smart Mobile Devices (SMDs) such as limited battery lifetime, limited processing capabilities, and limited storage capacity by offloading the execution and workload to other rich systems with better performance and resources. This paper presents the current offloading frameworks, computation offloading techniques, and analyzes them along with their main critical issues. In addition, it explores different important parameters based on which the frameworks are implemented such as offloading method and level of partitioning. Finally, it summarizes the issues in offloading frameworks in the MCC domain that requires further research.

Seamless application execution in mobile cloud computing: Motivation, taxonomy, and open challenges

Seamless application execution is vital for the usability of various delay-sensitive mobile cloud applications. However, the resource-intensive migration process and intrinsic limitations of the wireless medium impede the realization of seamless execution in mobile cloud computing (MCC) environment. This work is the first comprehensive survey that studies the state-of-the-art cloud-based mobile application execution frameworks (CMAEFs) in perspective of seamless application execution in MCC and investigates the frameworks suitability for the seamless execution. The seamless execution enabling approaches for the CMAEFs are identified and classified based on the implementation locations. We also investigate the seamless application execution enabling approaches to identify advantages and disadvantages of employing such approaches for attaining the seamless application execution in MCC. The existing frameworks are compared based on the significant parameters derived from the taxonomy of the seamless application execution enabling approaches. The principles for enabling the seamless application execution within the MCC are also highlighted. Finally, open research challenges in

realizing the seamless application execution are discussed.

Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges

For various reasons, the cloud computing paradigm is unable to meet certain requirements (e.g. low latency and jitter, context awareness, mobility support) that are crucial for several applications (e.g. vehicular networks, augmented reality). To fulfill these requirements, various paradigms, such as fog computing, mobile edge computing, and mobile cloud computing, have emerged in recent years. While these edge paradigms share several features, most of the existing research is compartmentalized; no synergies have been explored. This is especially true in the field of security, where most analyses focus only on one edge paradigm, while ignoring the others. The main goal of this study is to holistically analyze the security threats, challenges, and mechanisms inherent in all edge paradigms, while highlighting potential synergies and venues of collaboration. In our results, we will show that all edge paradigms should consider the advances in other paradigms.

Drop computing: Ad-hoc dynamic collaborative computing

Mobile applications nowadays generally consist of a frontend component running on the device, and a backend component running on the cloud that performs the larger computations. However, this usage model leads to high costs for developing the application (since a cloud infrastructure that should scale to the number of users must be maintained), and to a potentially bad user experience (if the latency is high or the users employ mobile broadband they pay for). Thus, we introduce the Drop Computing paradigm, which proposes the concept of decentralized computing over multilayered networks, combining cloud and wireless technologies over a social crowd formed between mobile and edge devices. Mobile devices and people interconnect to form ad-hoc dynamic collaborations to support the equivalent of a crowd- based edge multilayered cloud of clouds, where the capabilities of any mobile device are extended beyond the local technology barriers, to accommodate external resources available in the crowd of other devices. Thus, instead of every data or computation request going directly to the cloud, Drop Computing employs the mobile crowd formed of devices in close proximity for quicker and more efficient access. Devices in the mobile crowd are leveraged for requesting already downloaded data or performing computations, and the cloud acts as the second (or even third)

3 PROBLEM STATEMENT:

In drop computing mobile networks, where devices frequently connect and disconnect from the network due to factors such as mobility, intermittent connectivity, and limited bandwidth, ensuring data integrity becomes a significant challenge. Data corruption can occur during transmission, storage, or processing, leading to loss of critical information, compromised network performance, and degraded user experience. Therefore, the primary objective of this project is to develop robust strategies and mechanisms to mitigate the risk of data corruption in drop computing mobile networks

4 PROPOSED METHOD:

- We propose upgrading the expected versions mechanism by specifying that a given percentage of executors of a task need to be different. Furthermore, we add the restriction that a minimum number of final relay nodes per task should be expected. These restrictions can help increase consistency because of the following:

- By accepting task versions executed by different nodes, the chances that the information is not corrupt increase, regardless of the way data are corrupted _ by accepting task versions from different nodes, we allow the task to have a more diverse path from executor to owner, which is useful in the scenario where tasks are corrupted after they are computed..

1. Data Redundancy Mechanism:

Objective: Ensure data availability and integrity despite network dynamics. Specification:
Define redundancy policies specifying the number of replicas and their distribution across network nodes.
Implement a consensus algorithm (e.g., Paxos, Raft) to ensure consistency among replicas.
Enable automatic replication and synchronization mechanisms to maintain redundancy.

2. *Data Consistency Management:*

Objective: Maintain data consistency across distributed nodes to prevent corruption and conflicts.
Specification:
Implement distributed consensus protocols to agree on the order of data updates.
Employ techniques like vector clocks or Lamport timestamps to track causality and resolve conflicts.
Enforce strict consistency models (e.g., linearizability) or eventual consistency based on application requirements.

3. *Error Detection and Correction:*

Objective: Detect and correct errors to prevent data corruption during transmission.
Specification:
Utilize error-detection codes (e.g., CRC, checksums) to verify data integrity. Implement forward error correction (FEC) techniques to recover lost or corrupted data. Integrate retransmission mechanisms to request missing or corrupted data packets.

4. *Network Partition Handling:*

Objective: Mitigate the impact of network partitions on data integrity.

Specification:
Implement partition-tolerant algorithms (e.g., CRDTs) for data structures to reconcile conflicting updates.

Define strategies for graceful degradation, such as prioritizing consistency over availability during partitions.

Monitor network connectivity and trigger recovery mechanisms upon partition resolution.

5. Secure Communication Protocols:

Objective: Ensure secure data transmission to prevent unauthorized access or tampering.

Specification:

Utilize secure communication protocols (e.g., TLS/SSL) to encrypt data in transit.

Implement mutual authentication mechanisms between communicating nodes to prevent spoofing or man-in-the-middle attacks.

Manage cryptographic keys securely and regularly update them to maintain confidentiality and integrity.

6. Continuous Monitoring and Diagnostics:

Objective: Monitor system health and detect anomalies to prevent potential data corruption.

Specification:

Implement monitoring agents to collect and analyze network performance metrics. Utilize anomaly detection algorithms to identify deviations from normal behavior.

Integrate logging and auditing mechanisms to trace data transactions and diagnose potential corruption incidents.

7. Offline Data Handling and Synchronization:

Objective: Enable seamless data synchronization and integrity maintenance in disconnected or intermittent connectivity scenarios.

Specification:

Implement caching mechanisms to store data locally on mobile devices during network disconnections. Define synchronization protocols to reconcile local data changes with the central repository upon reconnection.

Ensure conflict resolution strategies are in place to handle divergent updates during synchronization.

8. *Disaster Recovery and Backup:*

Objective: Ensure data availability and integrity through robust disaster recovery and backup strategies.

Specification:

Establish backup procedures to regularly replicate data to offsite or cloud-based storage.

Implement data snapshotting to capture the state of distributed systems at specific points in time.

Define disaster recovery plans outlining procedures for data restoration and system recovery in case of catastrophic failures.

7 CONCLUSION

In this project, we have presented the Drop Computing paradigm, which combines edge and fog computing with mobile network and social information to decrease latency and power consumption. We presented several use cases for Drop Computing, including an AAL scenario where we show that it can be employed in an elderly care facility to reduce costs. Then, we proposed data consistency mechanisms for Drop Computing, assuming a scenario where mobile nodes want to solve some computation tasks and thus offload them to devices in proximity in an opportunistic fashion. The thorough experimental testing showed that, through setting appropriate restrictions, our consistency solution can satisfy the requirements of a network with regard to a desired trust level, since the proposed rating mechanism can lead to a task correctness as high as 100%. Furthermore, this happens without the latency and the number of tasks executed in the network being affected too much.

REFERENCES:

[1] K. Akherfi, M. Gerndt, and H. Harroud, ‘‘Mobile cloud computing for computation offloading: Issues and challenges,’’ Appl. Comput. Informat., vol. 14, no. 1, pp. 1–16, 2018.

[2] E. Ahmed, A. Gani, M. K. Khan, R. Buyya, and S. U. Khan, ‘‘Seamless application execution in mobile cloud computing: Motivation, taxonomy, and open challenges,’’ J. Netw. Comput. Appl., vol. 52, pp. 154–172, Jun. 2015.

[3] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, ‘‘Mobile edge computing—A key technology towards 5G,’’ ETSI White Paper, vol. 11, no. 11, pp. 1–16, 2015.

[4] R. Roman, J. Lopez, and M. Mambo, ‘‘Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges,’’ Future Gener. Comput. Syst., vol. 78, pp. 680–698, Jan. 2018.

[5] R.-I. Ciobanu, C. Negru, F. Pop, C. Dobre, C. X. Mavromoustakis, and G. Mastorakis, ‘‘Drop computing: Ad-hoc dynamic collaborative computing,’’ Future Gener. Comput. Syst., vol. 92, pp. 889–899, Mar. 2017.

[6] V.-C. Tabusca, R.-I. Ciobanu, and C. Dobre, ‘‘Data consistency in mobile collaborative networks based on the drop computing paradigm,’’ in Proc. IEEE Int. Conf. Comput. Sci. Eng. (CSE), Oct. 2018, pp. 29–35.

[7] G. Huerta-Canepa and D. Lee, ‘‘A virtual cloud computing provider for mobile devices,’’ in Proc. 1st ACM Workshop Mobile Cloud Comput. Services Social Netw. Beyond (MCS). New York, NY, USA: ACM, 2010, pp. 6:1–6:5. doi: 10.1145/1810931.1810937.

[8] N. Fernando, S. W. Loke, and W. Rahayu, ‘‘Dynamic mobile cloud computing: Ad hoc and opportunistic job sharing,’’ in Proc. 4th IEEE Int. Conf. Utility Cloud Comput. (UCC), Washington, DC, USA: IEEE Comput. Soc., Dec. 2011, pp. 281–286. doi: 10.1109/UCC.2011.45.