Data corruption avoidance in cloud computing

Prof. Nagesh.M [1, a)] Vardhan reddy Rangineni [2, b)],
Nithin reddy Gantla[3, c)],Sri teja reddy Kotha [4, d)], Shashi Kumar. B

[1] *Professor, CSE, Sreyas Institute of Engineering & Technology, Hyderabad-500068, India*
[2,3,4&5]*Students,4th CSE, Sreyas Institute of Engineering &Technology, Hyderabad-500068, India*
Author Emails nageshmathew@gmail.com vardhanbannu11@gmail.com
nithingnr14@gmail.com
tejareddy2468@gmail.com shashikumarb000@gmail.com

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 after the increase in usage of the Internet of Things and the rise in the number of connected devices causing load on the cloud. In drop computing, nodes can 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 project, we focus on the lowest layer of drop computing, if a mobile node needs data or has to compute a task and does not have the necessary resources, it can try to get the results from the ad-hoc cloud of mobile devices existing around it. Using these close-range nodes is cheaper even than contacting the edge, because short-distance protocols such as Bluetooth and Wi-Fi Direct tend to consume less power than mobile broadband protocols, while at the same time having lower latencies due to the short distances. Inconsistencies formed due to malicious intent (nodes intentionally tamper with the data to spread false information in the network) or hardware failures or signal loss or network congestion. This may lead to data corruption. We propose several mechanisms for ensuring data consistency in drop computing, ranging from storing the path of a task and a rating system to careful analysis of the data received. Through thorough experimentation, we show that our proposed solution can maximize the amount of correct (i.e., uncorrupted by using Hamming codes) data exchanged in the network, with percentages as high as 100%.

INTRODUCTION

Mobile applications nowadays offer a large number of innovative features for end-users. However, although smartphones generally have high computing power and plenty of resources, these applications generally rely on cloud support to offer the best interaction for the user, through mobile cloud computing. Because of the numerous cloud interactions, there

are certain limitations and challenges regarding the network load, since even two devices located close to one another need to pass through the cloud to interact.

Communicating with a cloud system is necessary for devices with limited resources, but costly in terms of infrastructure and delay. To reduce these interactions, the amount of data sent to cloud platforms needs to be reduced, by moving some processing at the edge of the network, closer to where data is generated. In a network composed of mobile devices (smartphones, sensors, things, etc.), to have access to data or computing power, a node needs to request the cloud. To avoid high latencies in the cloud, as well as the cost of virtual resources, edge computing is employed. This paradigm refers to the existence of routers, switches, or set-top- cloud or even help process some tasks. Thus, when a node requests data or compute-intensive tasks to be solved, it first contacts the edge devices, which can offer the reply without needing to contact the cloud. However, since edge and fog computing has already begun to show some limitations, new concepts have become necessary.

By further extending the edge computing model to reduce latency and costs, even more, we developed the Drop Computing paradigm, which adds an extra layer between the mobile devices and the edge nodes. Direct tends to consume less power than mobile broadband protocols, while at the same time having lower latencies due to the short distances. To optimize the process of selecting the suitable mobile node that can help with a request (for data or computations), social and historic metadata about the nodes making up the mobile layer are employed.

Thus, only nodes that are considered familiar and trustworthy are selected for serving the requests of a mobile device. Drop Computing implies that data or tasks are spread into the network composed of mobile devices, for quicker access and lower consumption of resources.

However, in such situations where data belonging to a node passes through other peers, extra care should be taken to ensure data consistency. Since at the lowest layer of Drop Computing we are dealing with a decentralized network (composed of mobile devices that only have information about and from the nodes they encounter), the classical data consistency methods cannot be employed.

There is no central entity for ensuring consistency, so nodes need to govern themselves and decide together which data are correct. Some nodes might have hardware failures or might be malicious, so they should be avoided. Therefore, in this paper, we propose several solutions for ensuring data consistency for task computation in Drop Computing, while striving to have a low effect on the overall processing latency and the number of tasks computed. Through thorough experimental simulations, we show that our solution is even able to achieve 100% correctness in certain situations, while keeping the effect on latency down.

1.1 Problem Statement

Communicating with a cloud system is necessary for devices with limited resources, but costly in terms of infrastructure and delay. To reduce these interactions, the amount of data sent to cloud platforms need to be reduced, by moving some processing at the edge of the network, closer to where data is generated. There is a possibility of data corruption while using Drop Computing is to be avoided.

1.2 Project Scope

This project explores the use of a drop computing approach implemented in the mobile nodes that want to solve some computation tasks and thus load them to devices in proximity in an opportunistic fashion. This project aims to solve the data corruption caused in the process. It helps in solving the inconsistencies formed due to malicious intent (nodes intentionally tamper with the data to spread false information in the network) or network congestion. So this can be used in networks where data is to be shared in smaller regions like within a stadium or information sharing among IoT devices.

1.3 Objectives

The main focus of this model is to propose a solution to the data corruption that occurred whileusing Drop Computing to share information in mobile nodes.

Existing System

- There exists a solution where devices with common goals work together to solve tasks. Each node can compute parts of a task, and then all these partial results are merged into the final solution, which is shared by all the contributing nodes.
- Another solution was proposed where a similar mobile cloud framework was used, but this solution has the drawback of only allowing single-hop device-to-device communication using Bluetooth. Furthermore, users of this framework are incentivized through monetary transactions, which might prove difficult to implement in real-life, especially since no data corruption mechanisms are proposed and communication is performed in a decentralized fashion between mobile devices.

Proposed System

- We propose upgrading the expected versions mechanism by specifying that a given percentage of executors of a task needs to be different. Furthermore, we add the restriction that a minimum number of _nal relay nodes per task should be expected.

- 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 the executor to the owner, which is useful in the scenario where tasks are corrupted after they
are computed.

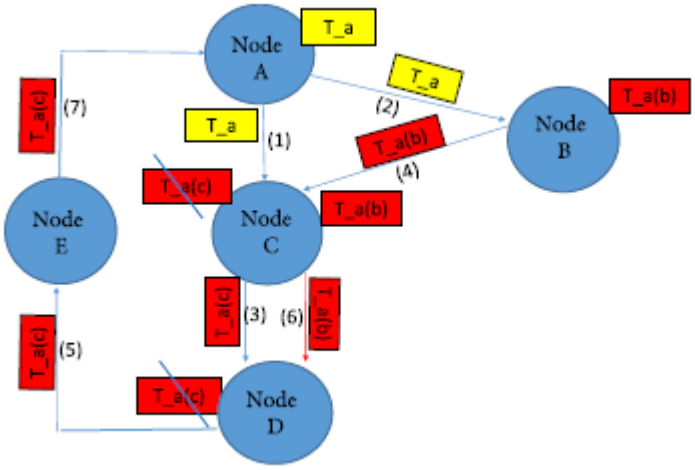


Figure 1: Architecture diagram

LITERATURE SURVEY

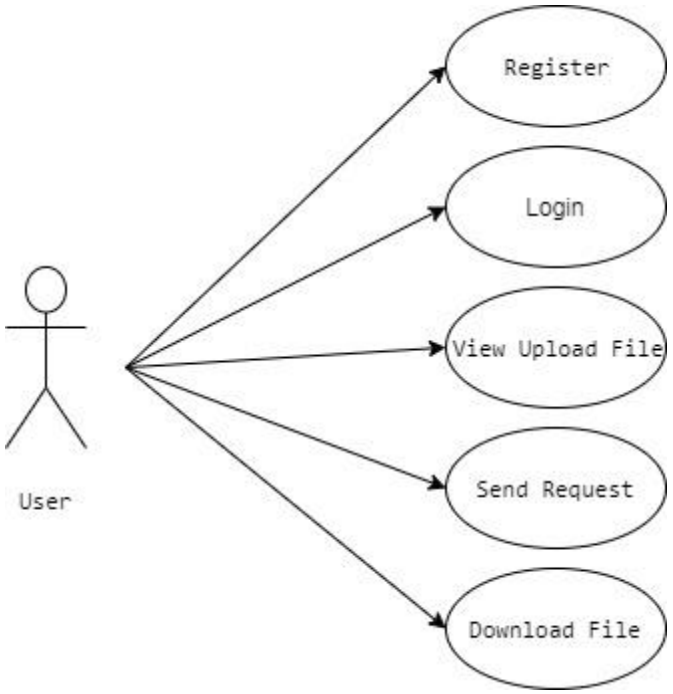
Despite the evolution and enhancements that mobile devices have experienced, they are still considered as limited computing devices. Today, users become more demanding and expect to execute computational intensive applications on their smartphone devices. Therefore, Mobile Cloud Computing (MCC) integrates 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.

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.

In the past years, the use of smartphones and mobile devices on the widest scale has led to an evolution of existing mobile networks. This has generated different routing and dissemination algorithms compared to the existing ones for classic wired networks. Therefore, we introduced 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 which will receive every data or computation request. The problem addressed in this paper is related to the consistency of data received from other peers, as there may be situations where it has been intentionally corrupted or has suffered inevitable changes on its path. Our main contributions include a rating mechanism of the nodes and a waiting mechanism for more versions of the same information in order to establish the correctness of the received data. Using simulations, we show that the proposed solutions increase the consistency level up to 100% in some situations. The simulation of the network interactions is based both on a synthetic mobility model, as well as on a real-life mobility trace.

UML DIAGRAMS



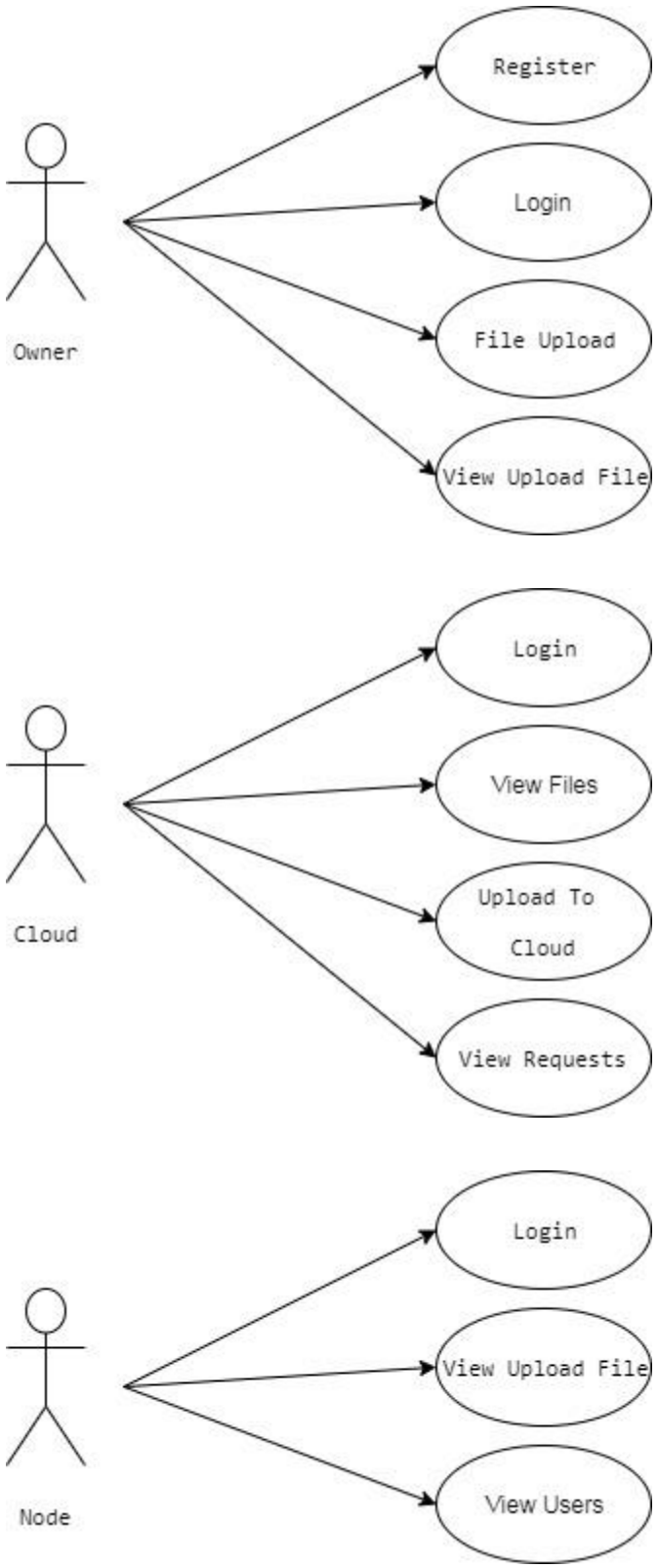


Figure 2 Use Case Diagram

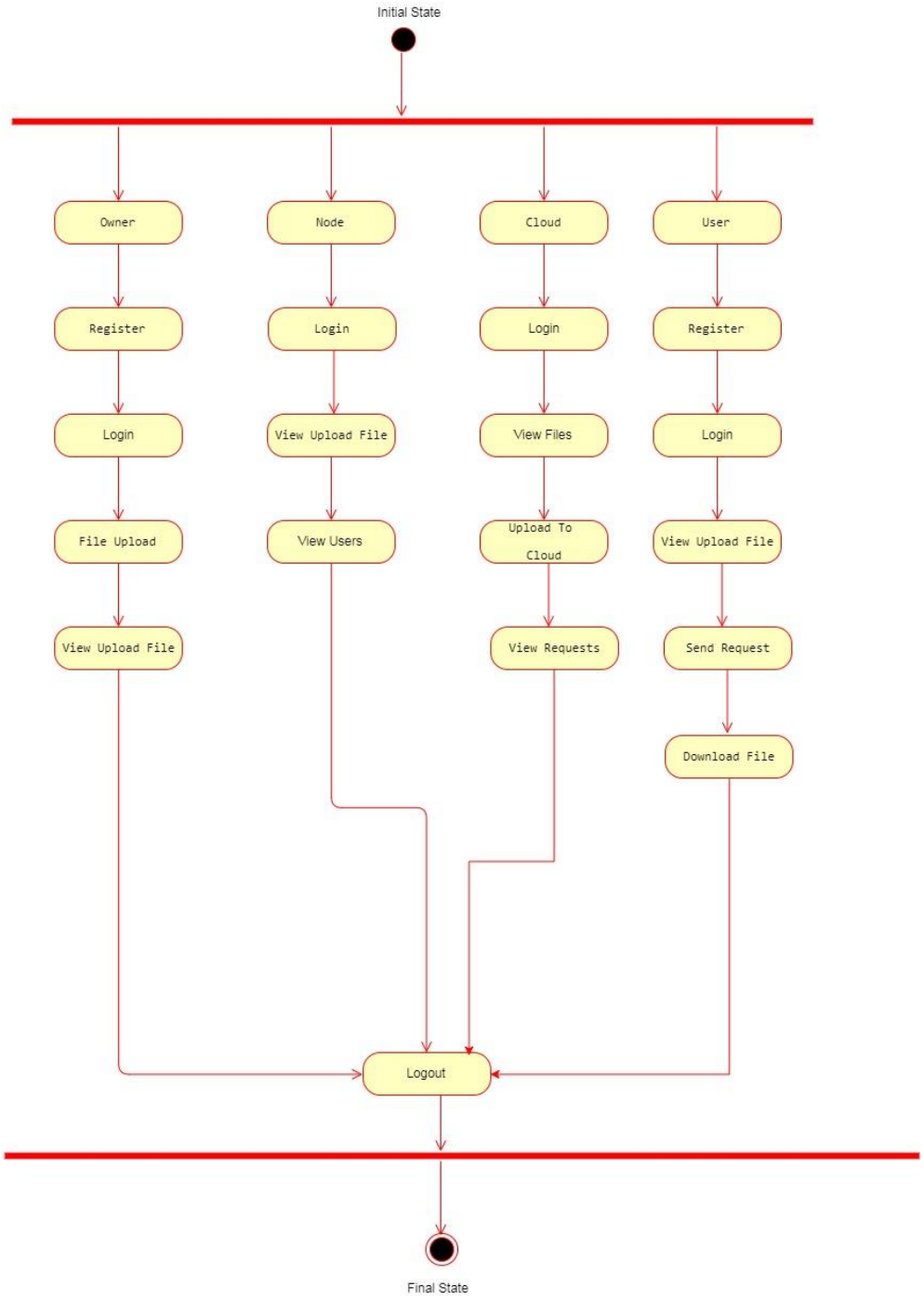


Figure 3: Activity Diagram

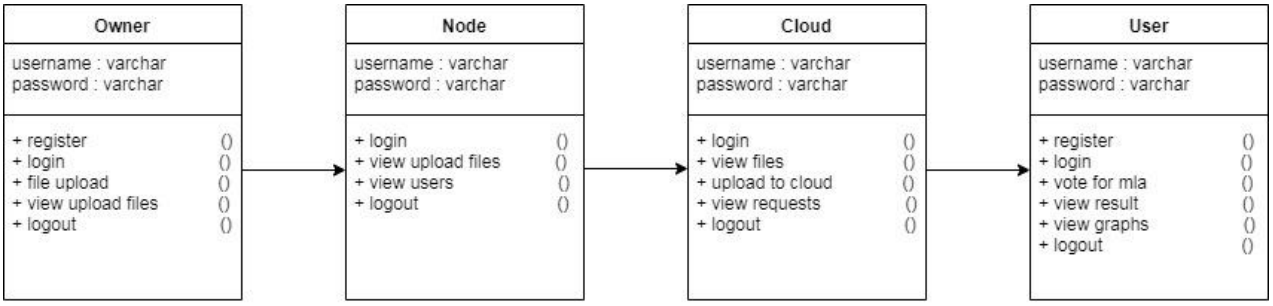


Figure .4: Class Diagram

TEST CASES

Table 1: **Test case for user registration**

Test Case	Input	Test case Description	Expected Output	Actual Output	Status
1	Invalid user id and word	User registration	User selects already existing user name	Displays message to choose different username	pass
	l id and word	User registration	User Enters valid details	user registered successfully	pass

Table 1: **Test case for user registration**

Table 1 shows that, user has to register in order to login to the account if the details entered already exists, then it displays message to choose different username. If the user enters the valid details then registration will be successful.

- Login Test case

Test Case	Input	Test case Description	Expected Output	Actual Output	Status
2	Invalid user id and word	Account login	User enters wrong user id and password	Displays the message user id or password is incorrect	pass
	l User id and password	Account login	User enters correct user id and password	User logs in successfully	pass

Table 2: **Test case for user login**

9. 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.

Future scope:

For future work, our aim is to improve the rating mechanism in order to obtain higher hit rates, as well as lower latencies and overhead. Moreover, we wish to come up with methods to increase the altruism of Drop Computing nodes, incentivizing them to participate in the collaborative network. This would be done by integrating reward mechanisms for nodes that execute and disseminate computing tasks. Finally, we also wish to implement a Reed-Solomon code [31] as an improvement over the Hamming mechanism, since it would be able to detect corrupted bits based on the number of parity bits added in the payload.

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.