

# Mobile Cloud Computing Provides Service Availability using A Context Aware Architecture

Anusha L, Nagaraja G S



**Abstract:** Mobile systems are becoming increasingly important, and new promising paradigms such as Mobile Cloud Computing. Mobile Cloud Computing is an application that allows data to be stored and processed outside of the mobile node. There is a lot of interest in using the resources that can be accessed by transparently using distributed resource pooling offered by nearby mobile nodes. This type of device is used in emergency, education, and tourism. Systems basically use dynamic network topologies in which network partitions and disconnection occurs frequently, so the availability of the services has been compromised. In this paper proposes the context aware architecture to provide availability of the services deployed in mobile and dynamic network environments which provides better response time, the services need not be migrated at real time, so the bandwidth and energy used has been more efficient.

**Keywords:** Monitoring Services, Context Service, Replica Service.

## I. INTRODUCTION

Local Mobile Cloud is required in domains where mobile applications also need to be supported by data storage and processing services provided by the mobile platform itself in a transparent and flexible way, such as rescue teams, security forces, tourism [1]. The nodes could be turned off or disconnected (temporarily or permanently), Since these networks are generally multi-hop, this normally entails connection failures, path changes, or even network partitions, all of which could have a significant effect on the network's service availability.

The Monitor Services send information to the Context Manager Service, which is responsible for processing and storing it. The Replica Manager Service will use this information to modify the configuration of the replicas' activation/hibernation in response to the changes made in the background. A Mobile Ad-hoc Network (MANET) typically supports a local Mobile Cloud approach, which introduces new problems (e.g., battery limitations, restricted bandwidth, complex topology, transmission errors, or routing overhead,

etc.) that must be solved in order to ensure the Mobile Cloud's dependability.

The number of active replicas is directly proportional to a service's availability. An intensive replication, on the other hand, would necessitate a high level of energy consumption. As a result, an energy-conscious solution that balances system efficiency with efficient energy consumption is also needed. The rest of the paper is organized as follows: section 2 design which provides system architecture and data flow diagram, section 3 explains the methodology, section 4 provides the system testing details and finally the paper is concluded with section 5 results and analysis.

## II. DESIGN

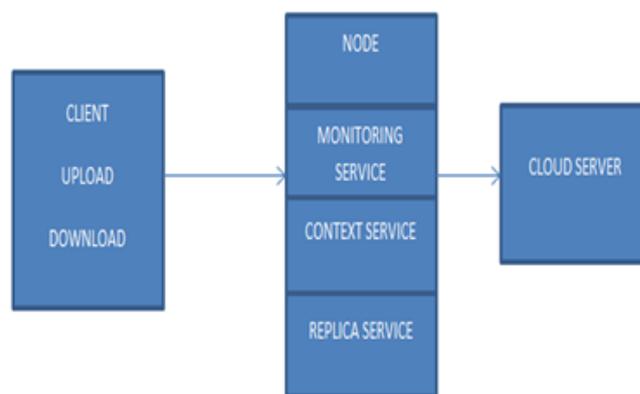


Figure 2.1: System Architecture

In figure 2.1 shows the System Architecture of the context free architecture providing service ability for the system [2].

A) Client: Client can upload and download media files to cloud using nodes which are present within its hop. Periodically client will be updated about the nodes present in its hop using UDP Multicast technique implemented in project. Randomly a Node will be selected while uploading or downloading file.

B) Node: User can create any number of nodes in different hops. Each node has context service, monitoring service and replica service implemented in it. Nodes are periodically updated with their neighbor information using UDP communication protocol.

C) Monitoring Service: This service monitors the battery level of node. If the battery is about to drain, it informs context service present in the node.

D) Context Service: Context service is used to shift the services of node to neighbor. This happens when the nodes battery is about to drain, or node changes its hop [3].

Manuscript received on May 18, 2021.

Revised Manuscript received on May 24, 2021.

Manuscript published on May 30, 2021.

\* Correspondence Author

Anusha L\*, Student, Department of Computer Science and Engineering, Rashtreeya Vidyalaya college of Engineering, Bengaluru, Karnataka, India. Email: anushal.scn19@rvce.edu.in

Dr. Nagaraja G S, Professor and Associate Dean, Department of Computer Science and Engineering, Rashtreeya Vidyalaya College of Engineering, Bengaluru, Karnataka, India. Email: nagarajags@rvce.edu.in

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

E) Replica Service: This service receives request from clients and neighboring nodes. When client sends upload or download request, it forwards it to cloud and gets the response which will be sent back to client. It also accepts service shifting request from neighboring nodes and adds the new service in current node.

F) Cloud Server: It is used as storage for media files. All the files uploaded by clients will be stored in cloud server.

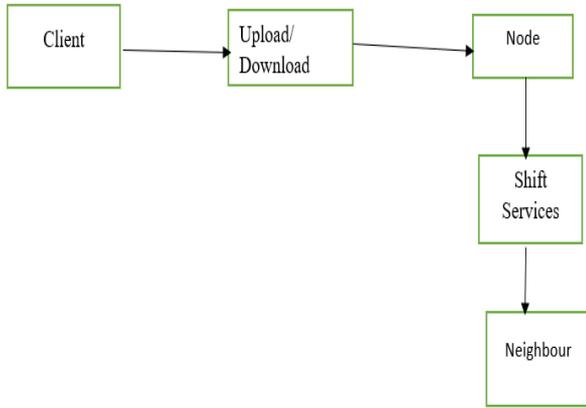


Figure 2.2: Data Flow Diagram Level 0

In Data flow diagram level 0 shows that in Mobile Cloud Computing system the client can upload or download the audio or video files to the cloud server with respect to the nodes present in the particular hops in the networks. The nodes it will shift its services available in particular node to its neighbor providing context awareness to the system.

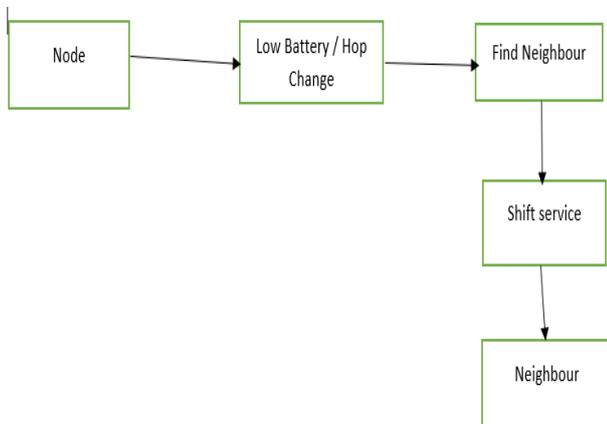


Figure 2.3: Data Flow Diagram Level 1

In Fig 2.3 shows Data Flow Diagram Level 1 in Mobile Cloud Computing the node with low battery or hop change with find the corresponding Neighbor with highest battery monitoring device and shift the services like audio or video to the neighbor so there is no compromising in the services in the systems.

III. METHODOLOGY

A) Java Swing

Java Swing is the Graphical User Interface used to design the front end of the application. It was earlier called Abstract Window Toolkit, serving as lightweight application which is

more customizable and provides “pluggable” look and feel for the application.

B) TCP/IP Client Sockets

TCP/IP sockets are used to create stable, bidirectional, permanent, point-to-point, stream-based connections between Internet hosts. A socket connects Java’s I/O device to other programmers running on the local computer or on some other machine connected to the Internet. When you create a Socket object, you’re automatically establishing a link between the client and the server. There are no methods or constructors that reveal the specifics of making the relation directly [4].

C) TCP/IP Server Sockets

For server applications, Java has a separate socket class that must be used. The Server Socket class is used to build servers that listen on published ports for connections from local or remote client programmers. Server Sockets are not the same as regular Sockets.

D) Java and the Net

TCP/IP is supported by Java by expanding the existing stream I/O interface. The TCP and UDP protocol families are also supported by Java. TCP is a network protocol that allows for stable stream-based I/O. UDP supports a point-to-point datagram-oriented model, which is simpler and therefore faster.

E) InetAddress

Both the numerical IP address and the domain name for that address are encapsulated in the InetAddress class. We communicate with this class using an IP host’s name, which is more convenient and understandable than its IP address. The number is hidden within the InetAddress class. InetAddress can handle both IPv4 and IPv6 addresses as of Java 2 version 1.4[5].

IV. SYSTEM TESTING

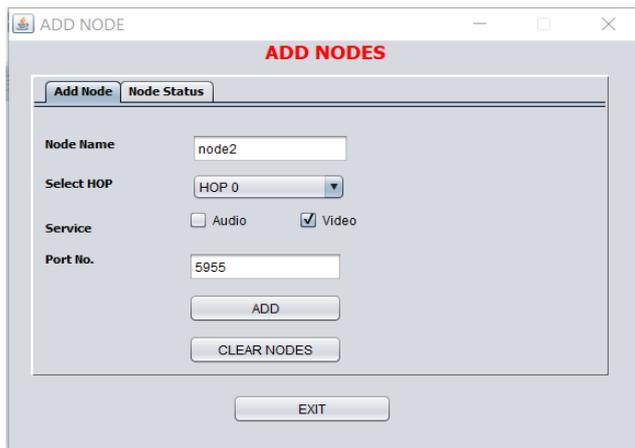
The aim of testing is to know the errors. Testing is the method of attempting to find any possible flaws or weaknesses in a work product. It allows you to test the functionality of individual parts, subassemblies, assemblies, and/or a finished product. It is the process of testing software to ensure that it meets its specifications and meets user expectations, and that it does not fail in an inappropriate way. Testing results are mentioned in the table below for the module Client, Node and Add node modules.

| Module | Given input   | Actual output                                | Result |
|--------|---------------|--|--------|
| Client | File Upload   | Upload request should be sent to node        | OK     |
| Client | File Download | File Download request should be sent to node | OK     |

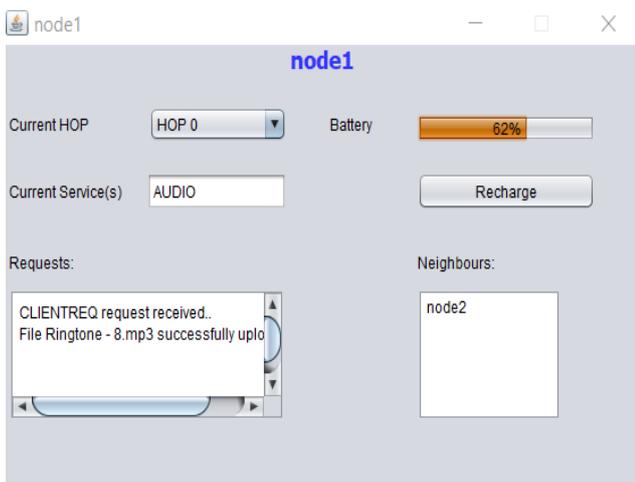


|           |                                     |   |    |
|-----------|-------------------------------------|---|----|
| Node      | Update Service                      | Service successfully updated                                | OK |
| Node      | Update Service                      | Service Updated   | OK |
| Node      | Change HOP                          | Request successfully sent to context service                | OK |
| Node      | Battery Status                      | Context Service successfully notified on low battery status | OK |
| Add nodes | Node name, hop, service and port no | Node successfully added in DB                               | OK |
| Add nodes | Clear DB                            | Node successfully removed from DB                           | OK |

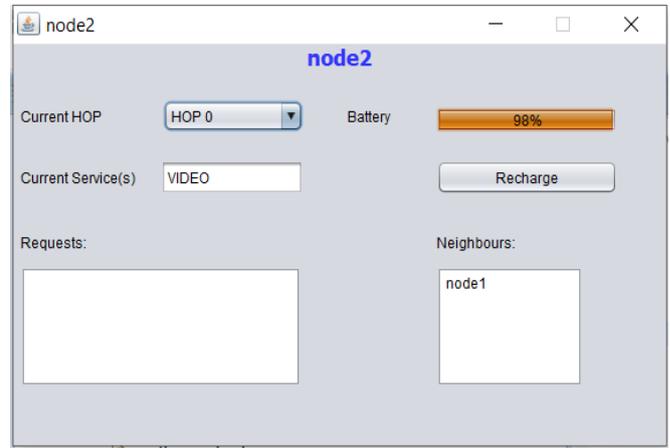
**V. RESULTS AND ANALYSIS**



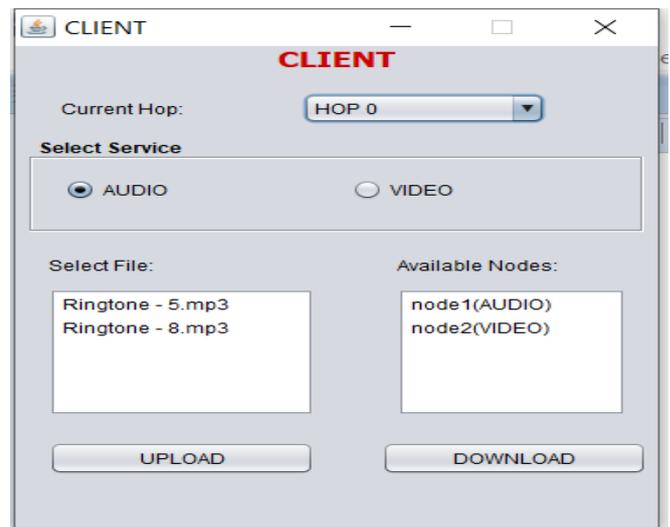
**FIG 5.1: Add Nodes**



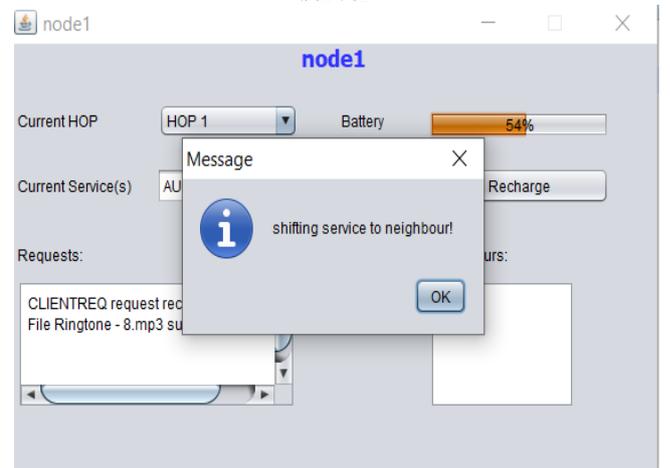
**FIG 5.2: Service provided by node 1.**



**FIG 5.3: Service provided by node 2.**



**FIG 5.4: Client will upload or download service to cloud server**



**FIG 5.5: Shifting service to neighbor.**

In Fig 5.1 shows Add Nodes field for context awareness architecture for sifting services in Mobile Cloud Computing which contains fields like Node name where any number of nodes can be added in the system, Select hop field which contains 3 hops in the network, services in this project provides audio and video services, port no has been randomly generated from the system.



In Fig 5.2 and Fig 5.3 shows the description of nodes in the project which contains current hop field, the monitoring services i.e. battery for the system, Current services it may be audio or video services, requests provide communication between client and the node using TCP/IP socket programming. Neighbors field provides the neighbor node in the hops.

In Fig 5.4 where client will upload or download the audio and video services in the cloud server, In Fig 5.5 shows the shifting service in the neighbor, if node1 want to change the next hop then it will shift its present service to its neighbors so there is no need to compromise in the services, the monitoring services in which the battery drain it will shift its services to its neighbor.

### VI. CONCLUSION

The paper provides implementation details about context aware architecture provides services like audio, video, monitoring in mobile cloud computing. So there is no compromise of services in mobile cloud computing which provides be better utilization of energy and bandwidth in the network.

### REFERENCES

1. G. Guerrero-Contreras, J. L. Garrido, S. Balderas-Díaz and C. Rodríguez-Domínguez, "A Context-Aware Architecture Supporting Service Availability in Mobile Cloud Computing," in IEEE Transactions on Services Computing, vol. 10, no. 6, pp. 956-968, 1 Nov.-Dec. 2017, doi: 10.1109/TSC.2016.2540629.
2. J. Zhou, J. Chen, L. Li and Z. Zhang, "The Context Awareness Architecture in Mobile Cloud Computing," 2012 Fifth International Symposium on Computational Intelligence and Design, 2012, pp. 302-305, doi: 10.1109/ISCID.2012.83.
3. N. Gupta and A. Agarwal, "Context aware Mobile Cloud Computing: Review," 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), 2015, pp. 1061-1065.
4. L. Guan, X. Ke, M. Song and J. Song, "A Survey of Research on Mobile Cloud Computing," 2011 10th IEEE/ACIS International Conference on Computer and Information Science, 2011, pp. 387-392, doi: 10.1109/ICIS.2011.67.
5. M. Zhao, H. Xiang and J. Chang, "A context-aware architecture for intelligent home environment," 2011 6th International Conference on Pervasive Computing and Applications, 2011, pp. 208-211, doi: 10.1109/ICPCA.2011.6106506

### AUTHORS PROFILE



**Anusha L** is a M.Tech student at Department of Computer Science and Engineering, Rashtriya Vidyalaya College of Engineering, Bengaluru, Karnataka. [anushal.scn19@rvce.edu.in](mailto:anushal.scn19@rvce.edu.in)



**Dr. Nagaraja G S** is working as professor and Associate Dean at Department of Computer Science and Engineering, Rashtriya Vidyalaya College of Engineering, Bengaluru, Karnataka, India. [nagarajags@rvce.edu.in](mailto:nagarajags@rvce.edu.in)