ANUSANDHAN- AISECT University Journal
Vol. 06, Issue No. 13, March-2018, P-ISSN 2278-4187, E-ISSN 2457-0656

Reliable Algorithm for Enhancing QoS in Privacy Preserved Mobile Social Cloud Computing using CAN

Kiran Patidar[1]* Priti Maheshwari[2] Piyush Kuamr Shukla[3] Anand Motwani[4]

[1]RNTU, Bhopal (M.P.) India

[2]RNTU, Bhopal (M.P.) India

[3]RGPV, Bhopal (M.P.) India

[4]SISTec-R, Bhopal (M.P.) India

Abstract – The Mobile Social Cloud Computing (MSCC) paradigm is evolved keeping mobility in mind. MSCC essentially includes several user requirements including Quality of Service (QoS), which is largely affected by faults and malicious behaviour of mobile nodes. QoS is a necessary metric to evaluate the quality of MSCC. Depending on the research areas, different researchers defined QoS in different ways. Most of the schemes are proposed against random faulty clouds and works well, but these do not protect from malicious nodes present in MSCC. In MSCC, malicious users are those who take cloud services from other devices and avoid providing services to others due to one or the other reason. This paper outlines major technical challenges (problems) in MSCC and the opportunities that can be realized to overcome the challenges. The survey of related literature is also presented. Furthermore, the work proposes a Reliable Algorithm to enhance privacy and Quality of Service (QoS) in MSCC. The proposed work utilized Content Addressable Network (CAN) to give logical structure to MSCC network. The proposed algorithm is implemented in CloudSim, a renowned simulation tool. The results are evaluated on basis of QoS parameters: Execution Time and Finish Time. The experimental results are analyzed to demonstrate the efficacy of algorithm in both cases: SNS without CAN and SNS in presence of malicious environment.

Keywords: Cloud Computing, Content Addressable Network (CAN), Fault tolerance, Mobile Social Cloud Computing (MSCC), Quality of Service (QoS), Scheduling, Social Networking, Privacy, Virtualization

----------♦----------

1. INTRODUCTION

Nowadays, the computing systems are available in the form of Personal computers (PCs), Mobiles, Tablets and other handheld devices. Meanwhile, Virtualization for PCs started and Internet becomes more popular and accessible. Taking virtualization online proved a logical step in evolution of Cloud Computing (CC). This era is said to be the CC era. CC is not entirely a novel concept; it is in complicated connection with Grid Computing model, and few related technologies such as distributed, utility and cluster computing.

(a) Mobile Social Cloud Computing

The utilization of SNS is really soaring with increased use of wireless mobile devices (Sook Kyong, et.al. 2013). Integrating a mobile cloud into social networking infrastructure could open up automatic sharing and P2P multimedia access, and this will also reduce the need to back up and serve all of this data on huge servers (E.E. Marinelli, 2009).

Based on this relationship in the form of SN using SNS, users develop basic level of inherent trust for data and information sharing. Users share media and other files among each other with less or no authentication because users are eager to provide their data to other SN members even through mobile devices. The Paradigm is evolved keeping mobility in mind known as the **Mobile Social Cloud** (refer Figure 1).

Fig. 1: Mobile Social Cloud Computing

(b) Motivation

The sustained rise of CC offers the promise of quicker development and service delivery while providing cost benefits and faster replication of services. The Cloud can flawlessly deliver services to multiple devices such as smart-phones and tablets. The inherent problem referred to as Faults. One of the inherent faults is due to Malicious Behaviour. It may come under intentional faults in which, even after availing a service request, a device may not grant cloud service to other mobile devices. Therefore it is vital to deal with these faults.

In this work a Reliable Algorithm which comes under Fault Tolerant model is proposed to deal with malicious users, thus enhancing QoS and privacy in MSCC.

(c) CAN

Every cloud server has a CAN structure to manage mobile devices. Every mobile device is registered on CAN in the cloud server and is mapped on a point of CAN having a virtual logical address, namely CAN coordinates, for CAN routing. CANs are fault-tolerant, scalable and completely self-organizing peer-to-peer overlay network. As CAN is a distributed infrastructure that provides hash table-like functionality, it has been used as a base approach for large scale data management of frequently moving objects in various computing environments.

2. LITERATURE REVIEW

Sook Kyong Choi et al. (2013) proposed fault tolerance and QoS (Quality of Services) scheduling using CAN (Content Addressable Network) in Mobile Social Cloud Computing (MSCC). Fault tolerance and QoS scheduling consists of four sub-scheduling algorithms: malicious-user filtering, cloud service delivery, QoS provisioning, and replication and load-balancing. Under the SNS, a mobile device is used can also be used as a resource for providing cloud services. They simulated scheduling both with and without CAN and shows improvement in cloud service execution time, finish time, reliability and reduces the cloud service error rate.

Authors Qian, (2010) defined basic and extended QoS for evaluating scheduling algorithms. Time and Cost are considered in basic QoS while reliability, availability, security, and reputation covered in extended QoS.

The work M. Reza Rahimi (2013) discusses state of the art in the MSCC. Authors M. Reza Rahimi (2013) illustrated the applicability of MCC in various domains including Social Networking, learning, health/wellness and commerce. The research gaps are identified covering critical aspects of realization and effective utilization. The authors suggests that improved resource allocation can be achieved through efficient task distribution.

The paper Hoang T. et.al. (2011) presents a survey of MSCC that includes overview of the definition of MCC, architecture, and applications. MSCC issues, existing solutions, and approaches are presented. The main issues discussed includes performance (e.g., low bandwidth, storage and battery life), environment (e.g., availability, scalability and heterogeneity), and security (e.g., issues related to privacy and reliability).

Elio Goettelmann et al. (2013) proposed an approach for deploying business processes on the cloud supporting security constraints; thereby ensuring sensitive data exchange. They consider additional requirements related to data-dependencies and Quality of Service (QoS) disparities to optimize the execution of the outsourced process.

3. PROPOSED WORK

(a) MSCC Environment for Proposed Work

Figure 2 shows the MSCC computing environment. MSCC is using CAN which is type of P2P network. As an instance let us assume that User-1 and User-2 are on same social network. When User-1 requests cloud service from server, the server returns device information of User-2, finally both user connect and share the resources and / or services, without much authentication. Also a mobile device can be a member of any or every social network.

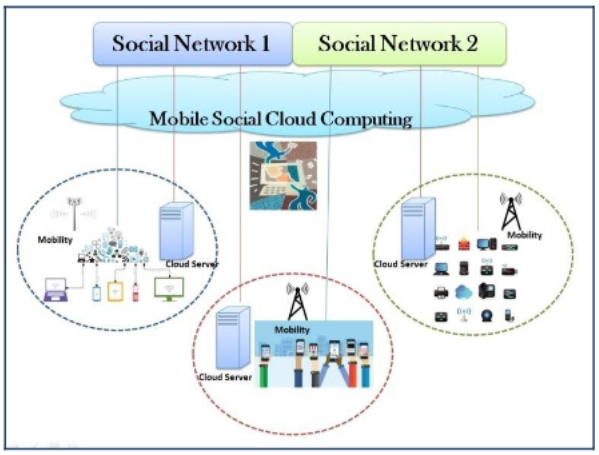

Fig. 2: Global View of MSCC Environment

(b) Proposed Reliable Algorithm for Enhanced QoS using CAN

The nodes who intend to join the network will get a unique identifier (resource_ID) and a launch message from cloud server. The server sets the primary reputation of nodes to a minimum value. The nodes that are malicious or not intend to coordinate in the network will send the negative reply. The server collects the reply and lists the status of nodes in two categories i.e. "True" and "Initial Malicious nodes" Also, the server sets the reputation of True nodes and initially detected malicious nodes to a minimum value. This is required for further authentication and filtering the suspicious True nodes. In the next step, the server sends one small test file, for storing, to all nodes. That who responds positively in the context i.e. if correct acknowledgement code is received then the reputation of such nodes is increased using formula mentioned in equation 1.

Equation 1.

$$reputation_{i-1} = reputation_i \quad (\text{if } i=0)$$

$$reputation_{i-1} = reputation_i + reputation_i * \log_{10} e^{(reputation_i)} \quad (\text{if } i>=1)$$

If mobile node is true, then its reputation value is high, if reputation value is low then higher the probability of malicious. When the reputation value is below certain threshold (minimum) value, the node would not be allowed to connect to MSCC.

4. EXPERIMENTAL SETUP AND RESULT ANALYSIS

(a) Simulation Scenario

There are various scenarios configured for experiment purpose, considering various existing and proposed work. Here, we are only showing a single case in 10 scenarios. Table 1 show for simulation scenario according to Presence of Malicious nodes, Reliable and QoS Aware Algorithm. Here the experiment presented is simulated 10 times. For each of the experiment the average of following 02 parameters: Finish time and Execution time is taken.

Table 1
Simulation Scenarios

	SNS without CAN	SNS using CAN with Malicious Nodes	SNS using CAN with Reliable (QoS) Algorithm
Case	Yes	Yes	Yes

The scenario is configured in famous cloud simulation tool: CloudSim [8], using the entities as shown in Table 2. For simulation purpose SNS is configured with 3 Data centres, 4 access points, 30 virtual machines. The users with 100 mobile devices are considered which asks for or executes 50 services.

Table 2
Simulation Configuration

Entities		Nos.
Social Network Services (SNS)	Data Centre	**03**
Access Points (AP)s	Brokers	**04**
Virtual Machines (VMs)		**30**
Mobile Devices	Hosts	**100**
Cloud services (Tasks)	Cloudlets	**50**

(b) Performance Parameters

(i) Service or Task Execution Time: It is referred to as time taken to execute the service that is requested by mobile device.

(ii) Cloud Service Finish Time: Finish Time represents the end of all tasks running at DC. It is also represented as Maximum Turnaround time taken by a process.

(a) Result Analysis

The performance comparison of Execution Time and Finish Time is shown in Table 3 and 4 respectively. The graphical comparison of Execution Time and Finish Time is shown in Figure 3 and 4 respectively. Figure 5 and Figure 6 are respectively showing the Average Execution and Finish Time of scenario as presented in Table 1.

	SNS without CAN	SNS using CAN with Malicious	Proposed Reliable (QoS) Algorithm
Scenario 1	301.75	317.96	310.6
2	301.75	321.6	306.9
3	301.75	318.25	307.3
4	301.75	315.24	310.7
5	301.75	316.23	314.2
6	301.75	318.3	315
7	301.75	319.26	309.1
8	301.75	316.28	300.77
9	301.75	315.81	309.74
10	301.75	317	312.12

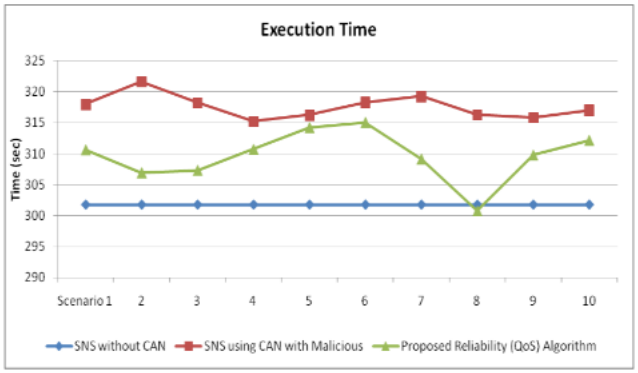

Fig. 3: Comparison of Execution Time (sec)

	SNS without CAN	SNS using CAN with Malicious	Proposed Reliable (QoS) Algorithm
Scenario 1	339.8	359.4	356.6
2	339.8	363.6	349.6
3	339.8	363.6	352.4
4	339.8	356.6	355.2
5	339.8	355.2	369.2
6	339.8	359.4	359.4
7	339.8	366.4	355.2
8	339.8	355.2	352.4
9	339.8	352.4	356.6
10	339.8	356.6	359.4

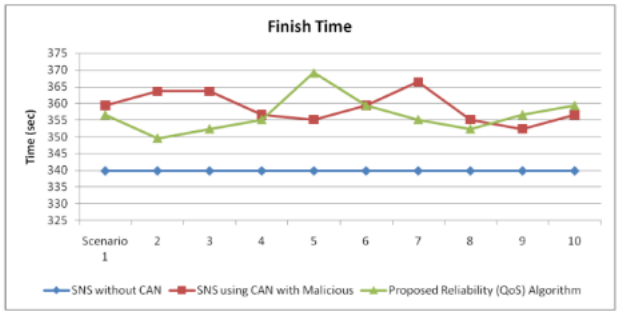

Fig. 4: Comparison of Finish Time (sec)

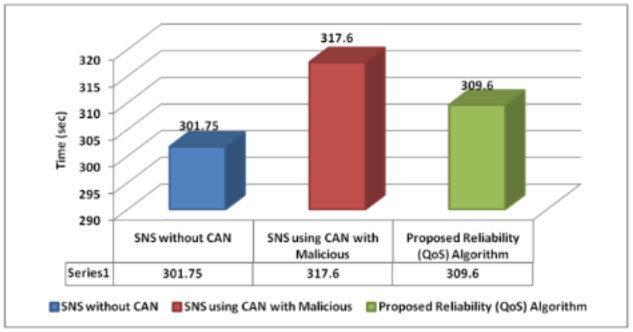

Fig. 5: Average Execution Time (sec)

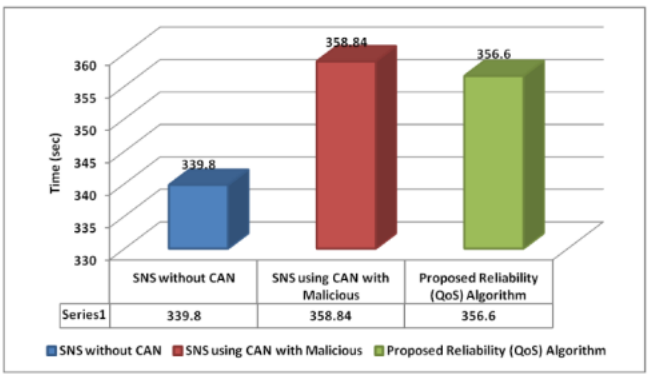

Fig. 6: Average Finish Time (sec)

5. CONCLUSION AND FUTURE DIRECTIONS

The work presents opportunities that can be realized to overcome the major technical challenges (problems) in MSCC along with the survey of related literature. The effect of malicious nodes on SNS is also shown on two QoS parameters. Furthermore, a Reliable algorithm to deal with malicious users, who intend to join the SNS, is proposed to enhance QoS and privacy in MSCC. Depending on the research areas, different researchers defined QoS in different ways. In future the work will be evaluated on the basis of various extended QoS metrics. The complete frameworks are still in need for almost all works proposed in the field. So, the framework for the algorithm can also be proposed in future. In dynamic MSCC the major challenge is to determine the effectiveness of the QoS algorithms. Almost all works proposed in the field have not evaluated their techniques against reliability. The methods to evaluate the effectiveness of the works' proposed in literature are still in need. So, in future we propose a method to evaluate the accuracy of such techniques and frameworks.

REFERENCES

E.E. Marinelli, Hyrax (2009): cloud computing on mobile devices using MapReduce, Masters Thesis, Carnegie Mellon University.

Elio Goettelmann, Walid Fdhila and Claude Godart, (2013). "Partitioning and Cloud Deployment of Composite Web Services under Security Constraints", IEEE International Conference on Cloud Engineering, pp. 193-200.

Hoang T. Dinh, Chonho Lee, Dusit Niyato* and Ping Wang, (2011). "A survey of mobile cloud computing: architecture, applications, and approaches", John Wiley & Sons, Ltd. WIRELESS COMMUNICATIONS AND MOBILE COMPUTING.

Ian Foster, Yong Zhao, Ioan Raicu, Shiyong Lu (2011). "Cloud Computing and Grid

Computing 360-Degree Compared", Available at: https://arxiv.org/ftp/arxiv/papers/0901/0901.0131.pdf

M. Reza Rahimi · Jian Ren · Chi Harold Liu · Athanasios V. Vasilakos · Nalini Venkatasubramanian, (2013). "Mobile Cloud Computing: A Survey, State of Art and Future Directions", Springer Science + Business Media New York.

Qian, T., Huiyou, C., Yang, Y., Chunqin, G. (2010): A trustworthy management approach for cloud services QoS data. In: ICMLC, pp.1626–1631

Rakshit Gupta, Piyush Kuamr Shukla, Rajeev Pandey (2011). "Survey on Mobile Social Cloud Computing (MSCC), " International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169, Volume: 5 Issue: 6, pp. 1332 – 1340.

Rodrigo, N.C., Rajiv, R., Anton, B., De Rose, C.A.F., Buyya, R. (2011). Cloud Sim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. SPE J. 41(1), pp. 23–50. ISSN:0038-0644

Sook Kyong Choi, Kwang Sik Chung and Heonchang Yu (2013). "Fault Tolerance and QoS Scheduling using CAN in Mobile Social Cloud Computing", Springer Cluster Computing, DOI 10.1007/s10586-013-0286-3.

Corresponding Author

Kiran Patidar*

RNTU, Bhopal (M.P.) India

E-Mail – kiranpatidar21@gmail.com