Volume 5, Issue 11, November 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

FDRM: A Reliable, Fault Tolerant and Resource Allocation Algorithm in Cloud Computing

Manjit Singh* Nitesh Prabhakar, Raj Singh
School of Computer Science, Lovely Professional University
Punjab, India

Abstract: Nowadays, building secure and reliable distributed system architecture is a major challenge because of its network size in huge geographical distributed location. Security, replication of secured data; fault tolerance is the major concerns in the distributed environment. To deal with the issues, an efficient Fault Detector and Replication Manager (FDRM) algorithm is proposed. The purpose of FDRM algorithm is to increase the availability of the resources even failure occurs in network architecture.

Keywords: Distributed network, FDRM, Replication, Fault Tolerance.

I. INTRODUCTION

The rapid growth in the field of distributed networks increases the security concern. Security is one of the major and is a constant issue for distributed environment. Distributed system is a rapid emerging framework that serves for various organizations and personal applications. The rapid development of network sizes and increase in interconnectivity is the major concern regarding security in network [1]. Security is another major issue in the distributed network environment. As users depending more on the distributed system, the need of secure network become an urgent requirement. For secure network and better communication, access control and authentication are critical tools [1].

For such kind of systems, fault tolerance is important. Fault tolerance is the ability of system to work even in the presence of faults occurs in the system [4]. Various techniques have been used to build reliable and fault tolerant computer and entire network [6]. Hariri [6] discuss about fault masking, fault detection, fault containment, fault diagnosis, repair and recovery of faults.

In this paper, a new system proposed, Fault Detector and Replication Manager (FDRM) in existing network architecture to provide reliability and availability of the network resources. FDRM uses the replication, fault tolerance, failure handling system features in order to provide efficient facilities to the users. Section 2 presents the Related Work. Section 3 introduces the proposed architecture and section 4 describes FDRM and its performance analysis.

II. RELATED WORK

[1] Refers to a system called "DNCSs (Distributed Networked Control Systems), which stated as a group of systems which coordinate their actions with the help of communication networks to achieve certain goals. The introduction of this kind of networks achieves many advantages which includes low cost, reduced wire length, ease of system diagnosis and maintenance. But this also results in some others issues such as network delay, packet loss, network scheduling. [2] Propose a clustering coefficient as a local metric in order to make system fault tolerance, which is mainly implemented in wireless sensor networks. The local clustering coefficient determines the degree of the connectedness of the node's neighbor. [3] Proposed Fault indicating devices such as fault indicators have been widely used in distributed system in order to improve the reliability and to reduce the outage duration. Fault Indicators (FIs) with communication interfaces are applied into distribution automation (DA) that further reduced fault-finding time by reporting status back to control center. During the occurrences of faults, alarms and other information received from Outage Management System send to system operators. [4] Proposed a new authentication mechanism which is based is based on decentralized approach. In this the authentication key or secret key is stored on different entities rather than single one. SNAP (Secure Network Access Protocol) is used to increase the network security and availability. [5] Proposed a password authentication mechanism with smart cards. In this method user can change their passwords freely and during this the system does not required directory of password. [6] Proposed a new authentication service called Kerberos Authentication Service. It is a distributed approach that allows a client, running on behalf of a user in order to approve its identity without sending the data across the network.

III. PROPOSED ARCHITECTURE

FDRM (Fault Detector and Replication Manager) is proposed in order to provide reliability and security to the system. In this FDRM is deployed in the network, which checks for the error such as link failure and node failure. It is basically work on by sending acknowledgements to the node(s) which are participating in the network and contained secret share in itself. The nodes will also send acknowledgements in response to that. If one of the node(s) does not response to that acknowledgement then it will again send acknowledgement after certain period of time. If the node does not response

after second attempt then it will collect that secret share from that particular node(s) and save the copy of secret share and save the state of the node in its master file.

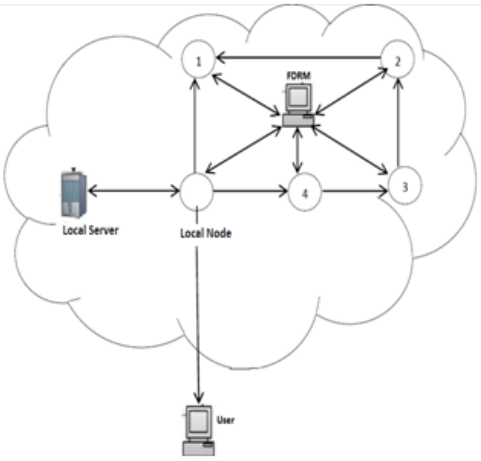


Fig 1: Proposed Architecture

3.1 Proposed Algorithm

// K = minimum number of shares that are required to access the network resource(s).
// N = total number of nodes in the network/quorum of network that stored shares of Access Code.
// AC = Access Code is the combination of all shares
// LN = Local Node gather and reconstruct the Secret shares to allow user to access the network resources.
// FDRM = Fault Detector and Replication Manager supervises the entire network and come into action during any failure(s), node(s) failure, in network.
Step1: FDRM Sends Acknowledgment to the Entire Participating Node in the cloud architecture.
Step2: if (node(s) sends messages in respond to the acknowledgment)
{

- User asks for the shares held by the participating nodes in the cloud environment.
- All shares will be collected at LN and LN will reconstruct the AC.
- LN allows user(s) to access the network resource(s).

}
Else
{ // In case of failure, the FDRM will work.

- FDRM will Sends again Acknowledgment to the Entire Participating Node(s).

If (Node(s)! = Responding)
{

- Save the state of the failed node(s) in the FDRM master file & will replicate and migrate the share(s) of failed node(s) to it and FDRM will send this Share of failed node to LN.

}
}

IV. IMPLEMENTATION

The Actual Implementation of the proposed algorithm is accomplished by using network simulator i.e. NS -2. The following diagram shows the implementation in simulation.

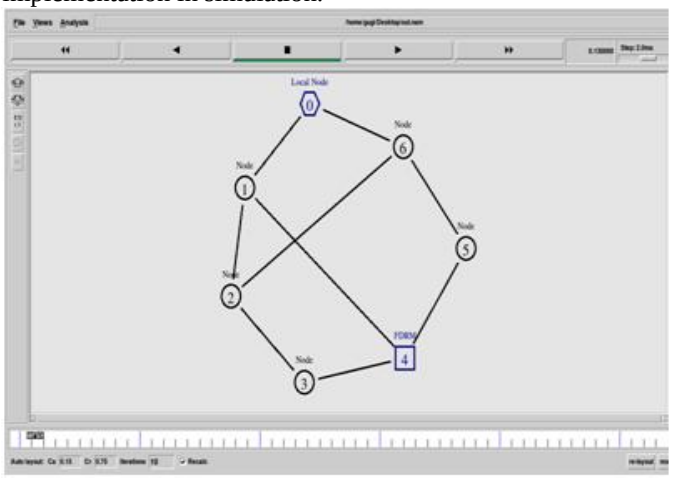


Fig 2 Connection of nodes in FDRM Network

In fig 2, the set of node are connected via communication links. In this network, node 0 is the local node which will ask for shares from nodes: node 1, node 2, node 3, node 5 and node 6. Node 4 is the FDRM node which is responsible for error checking and handling in the network architecture.

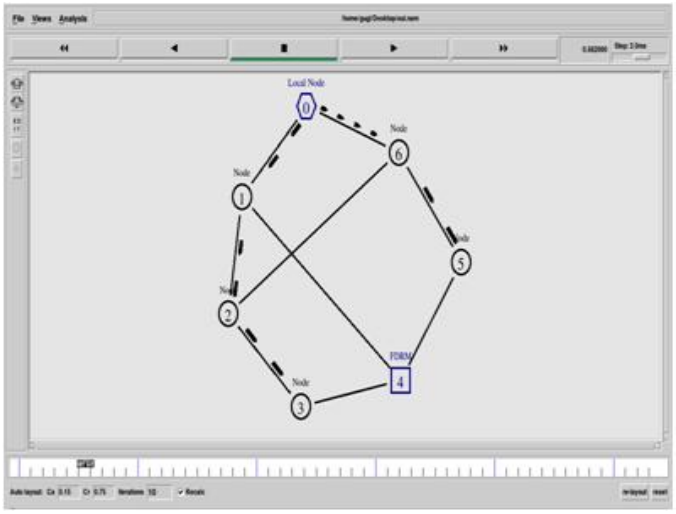


Fig 3: Sending Shares from Node 0 to all nodes

In fig 3, node 0 requesting to send shares from all the participating nodes except node 4. In response to that request all node will send secret share back to the local node.

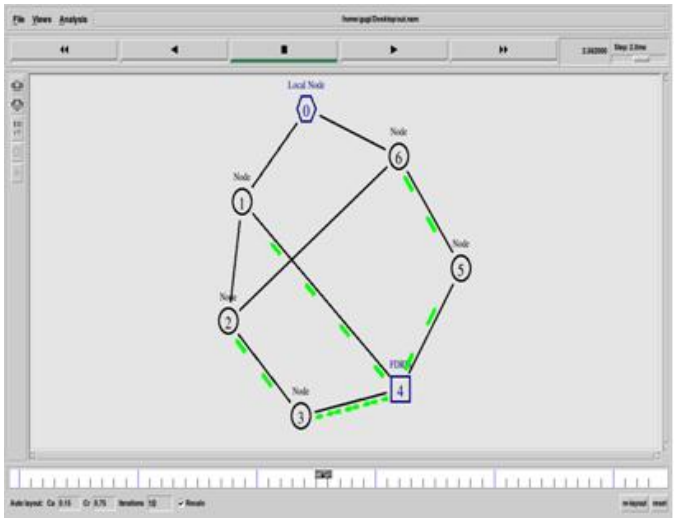


Fig 4: Sending Acknowledgement by FDRM to all nodes expect node

In fig 4, to check the aliveness of the nodes in the network, FDRM sending the acknowledgment messages to all the nodes. It also ensures that all the nodes working properly.

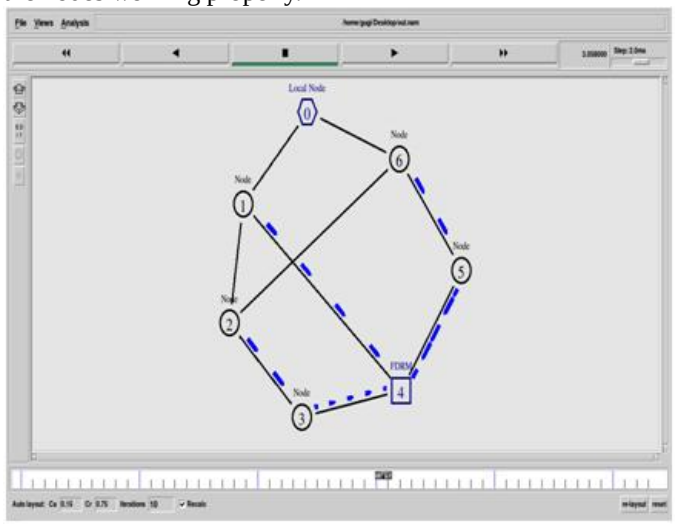


Fig 5: All Nodes sending messages to FDRM

In fig 5, all the nodes respond back to the FDRM. The response message contains the information about the status (working or failed) of each node.

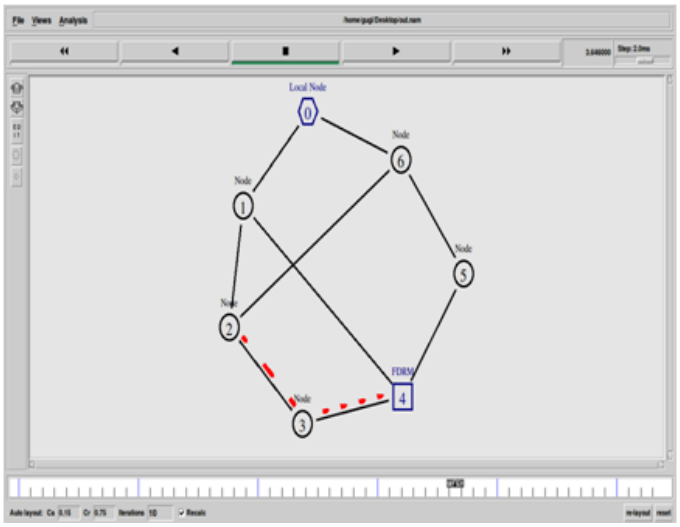


Fig 6: Failure of node 2

In fig 6, node 2 sends the failure messages to FDRM. FDRM will save the state of the node 2.

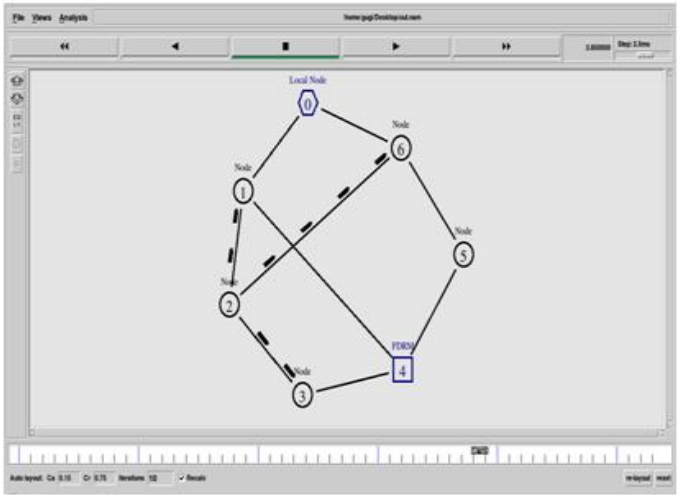


Fig 7: Node 2 sending Data to adjacent nodes (1.3 and 6)

In Fig 7 Before failure node 2 sending the secret share to its adjacent or nearby nodes , i.e. node 1, node 3, node 6 respectively.

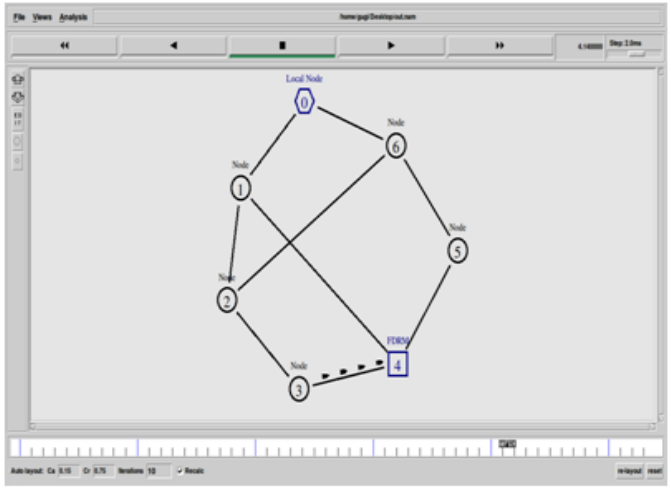


Fig 8: Node 3 sending secret share of node 2 to FDRM

Fig 8, after collecting the copy of secret share node 3 sends it to the FDRM, because it chooses the nearest path to send data to FDRM node.

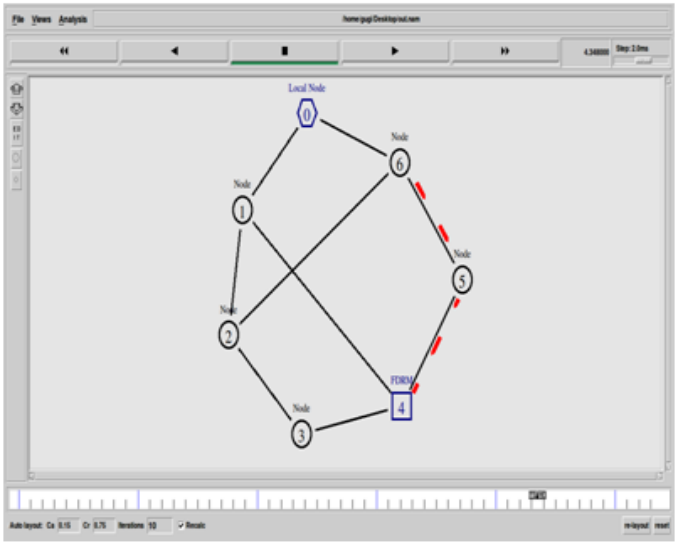


Fig 9: Failure of Node 6

Fig. 9 depicts the failure state of node 6. Node 6 holds the secret share and that secret share are migrated to the FDRM. So before occurrence of failure, such as crash failure, node 6 sends its secret shares to the FDRM. Failure aliveness message is sending by the node 6 to FDRM.

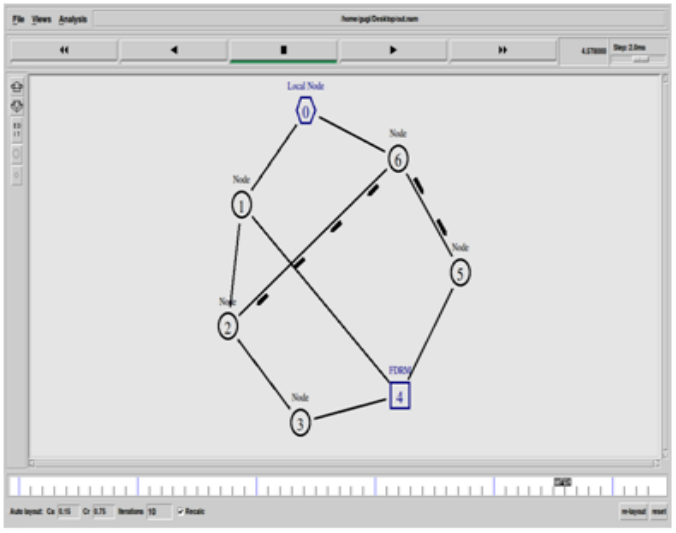


Fig 10: Node 6 sending Data to adjacent nodes (2 and 5)

In Fig 10, node 6 sending the copy of its secret shares to its adjacent nodes i.e. node 2 and node 5 respectively after the aliveness message sent to the FDRM.

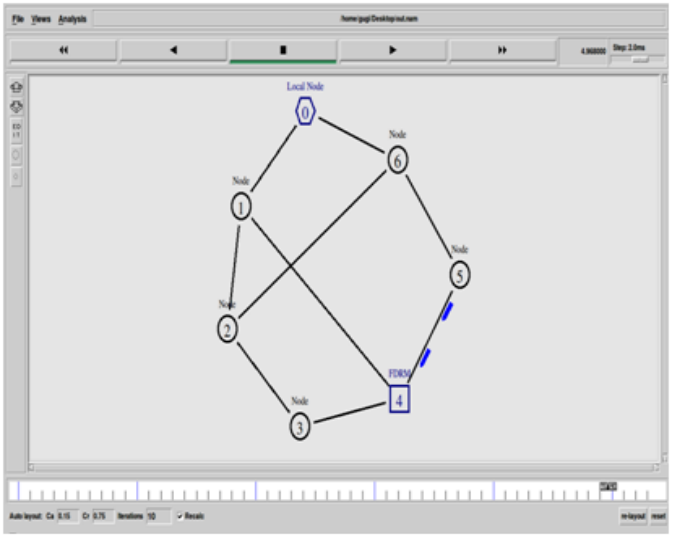


Fig 11: Node 5 sending Secret share to node 4 i.e. FDRM

In fig 11, the node 5 sending the secret shares which held by the node 6. Because as shown in figure, the node 6 send the secret shares to node 2 and node 5 nearest node 5 then send the secret data of node 6 to the FDRM to complete the authentication procedure.

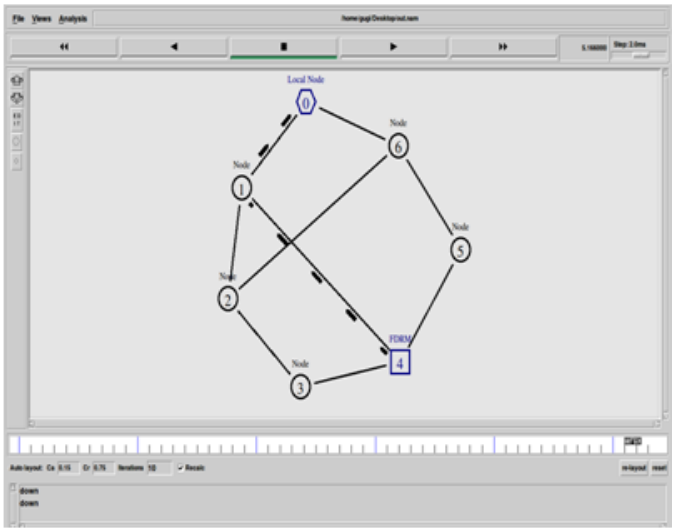


Fig 12: FDRM sending shares to Local Node 0

In Fig 12, FDRM will send the data collected from faulty node(s) to Local node 0, so that Local Node will reconstruct the secret share and create a key, which will allow the user to access the network resources.

V. RESULT AND PERFORMANCE ANALYSIS

To perform the real time implementation of the FDRM algorithm. A performance comparison is made with the existing SNAP protocol. NS2 network simulation is used to implement algorithm used behind FDRM approach. Three types of packets are used:

Table 1: Variable Size

Variable	**Size (bytes)**
Acknowledgement packets	16
Feedback Messages	16
Failure Acknowledgement packets	4

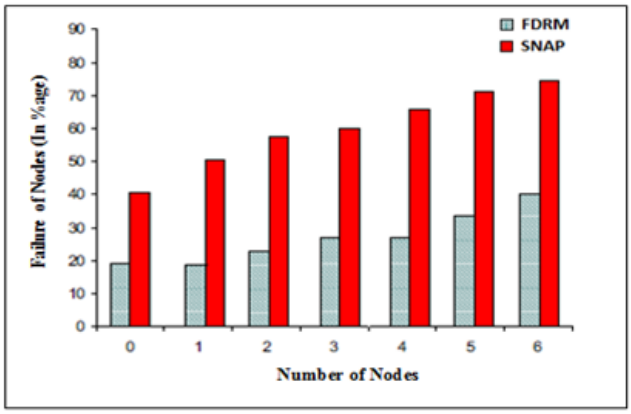
- Acknowledgment Packets: These Packets generated by the FDRM to all participating nodes except Local node (i.e. Node 0) to check the aliveness of the each node. Each packet size is of 16 bytes in size.
- Feedback Messages: These messages are generated in response to the acknowledgment send by the FDRM. The message contains the information about the working state of all the nodes in the network. All these messages collected by FDRM.
- Failure Acknowledgment packets: These messages used to notify the failure state of the nodes and they respond it to the FDRM to collect the share of that node.

Table 2: Analysis of SNAP and FDRM

Parameters	**SNAP**	**FDRM**
Failure Handler	No	Yes
Replication	No	Yes
Handling Link Failure	No	Yes
Handling Node Failure	No	Yes

5.1 Analysis of SNAP and FDRM

The comparison is made between the FDRM network architecture with the existing SNAP protocol. FDRM uses the replication technique to provide the data availability to the distributed clients. Failure Handler mechanism is embossed in FDRM network architecture to deal with failure such as node failure, communication failure. Secret data replication well performed by the FDRM to make the architecture morel reliable and even work in the presence of failures occur in the network architecture. Following are the parameter which is considered for the comparative relation:



Impact on network size using FDRM

In order to study the impact on network size using the FDRM, these proposed techniques simulated using network simulator i.e. NS -2. It is observed that by using FDRM in the network architecture, the size of the network also increased. FDRM is added to the network to check the failure in the network such as node failure, replication of data, communication link failure. As the network size is increased due to FDRM the chances of delay of the messages also increased. The figure (….), shows the impact on network using FDRM and SNAP as well is:

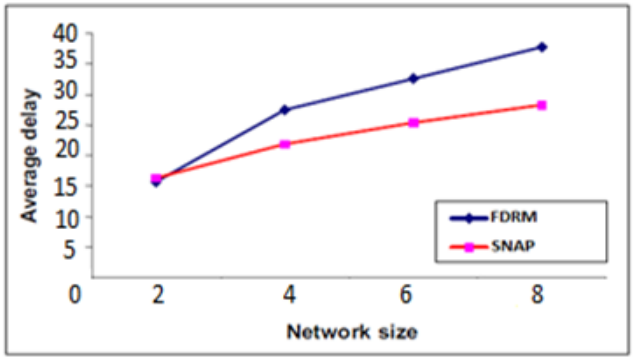


Fig 14 Fault Tolerance

Fault Tolerance and Failure Handling: Fig 14. indicates the effect of FDRM in SNAP Network. It depicts that when we implement FDRM in SNAP Network then the failure rate varies gradually as the number of nodes increases. But in case of SNAP Network, failure of nodes, without FDRM, increases. When the size of the network increases, the FDRM tolerates the failure occurred in the network system efficiently, but SNAP tolerate less faults as compared to the FDRM.

VI. CONCLUSION

In this paper, a secure architecture is proposed in the existing SNAP protocol. This architecture is based on fault detection and replication. FDRM ensures that the entire network architecture resources remain available for users even in the presence of failure. The proposed architecture enhance availability, fault tolerance, distributed trust. SNAP does not employ the replication or third party database in case of failure. FDRM results in additional Messages overhead, network size and delay issues. Component ranking which lead to the failure occur in the network architecture will considered and implement for further work to make the distributed environment more reliable and efficient. Self- stabilization will also implemented which results in less communication overhead and message overhead and delay issues that hamper the performance of the distributed environment.

REFERENCES

[1] Al Shahri, A. F., D. G. Smith, and J. M. Irvine. "A Secure network access protocol (SNAP)." *Computers and Communication, 2003.(ISCC 2003). Proceedings. Eighth IEEE International Symposium on*. IEEE, 2003.

[2] S. H. Hosseini, J. G. Kuhl, and S. M. Reddy, "A diagnosis algorithm for distributed computing systems with dynamic failure and repair," *Computers, IEEE Transactions on*, vol. 100, no. 3, pp. 223–233, 1984.

[3] L. B. Bhajantri and N. Nalini, "A fault tolerance approach to topology control in Distributed Sensor Networks," in *Advanced Communication Control and Computing Technologies (ICACCCT), 2012 IEEE International Conference on*, 2012, pp. 208–212.

[4] J. Al-Jaroodi, N. Mohamed, and K. A. Nuaimi, "An efficient fault-tolerant algorithm for distributed cloud services," in *Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on*, 2012, pp. 1–8.

[5] S. Chandrasekar and P. K. Srimani, "A new fault tolerant distributed algorithm for longest paths in a DAG," in *Software Reliability Engineering, 1993. Proceedings., Fourth International Symposium on*, 1993, pp. 202–206.

[6] S. Hariri, A. Choudhary, and B. Sarikaya, "Architectural support for designing fault-tolerant open distributed systems," *Computer*, vol. 25, no. 6, pp. 50–62, 1992.

[7] Z. Feng and G. Hu, "Distributed fault identification and fault-tolerant control for multi-agent systems," in *Control Conference (CCC), 2014 33rd Chinese*, 2014, pp. 1476–1481.

[8] V. Reppa, M. Polycarpou, and C. Panayiotou, "Distributed Sensor Fault Diagnosis for a Network of Interconnected Cyber-Physical Systems," 2014.

[9] M. Stumm and S. Zhou, "Fault tolerant distributed shared memory algorithms," in *Parallel and Distributed Processing, 1990. Proceedings of the Second IEEE Symposium on*, 1990, pp. 719–724.

[10] M. Kutlu, G. Agrawal, and O. Kurt, "Fault tolerant parallel data-intensive algorithms," in *High Performance Computing (HiPC), 2012 19th International Conference on*, 2012, pp. 1–10.

[11] G. Dini, "Increasing security and availability of an Internet voting system," in *Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on*, 2002, pp. 347–354.

[12] U. Malladi, "Notice of Violation of IEEE Publication Principles Design, analysis and performance evaluation of a new algorithm for developing a fault tolerant distributed system," in *Parallel and Distributed Systems, 2006. ICPADS 2006. 12th International Conference on*, 2006, vol. 1, p. 10–pp.

[13] P. Broadfoot and G. Lowe, "On distributed security transaction that use secure transport protocol," in *Computer Security Foundations Workshop, 2003. Proceedings. 16th IEEE*, 2003, pp. 141–151.

[14] P. Thamidurai, Y.-K. Park, and K. S. Trivedi, "On reliability modelling of fault-tolerant distributed systems," in *Distributed Computing Systems, 1989., 9th International Conference on*, 1989, pp. 136–142.

[15] Z. Ding and M. Jiang, "Reliability Computing for Service Composition," in *Secure Software Integration and Reliability Improvement, 2009. SSIRI 2009. Third IEEE International Conference on*, 2009, pp. 449–450.

