Consistent Security Protocol For Radio Detection And Ranging

Keerthy Prasannan[1], Anju Rachel Oommen[2], Smita C Thomas[3]
[1, 2, 3] Dept of Computer Science and Engineering
[1, 2, 3] Mount Zion College of Engineering, Kadammanitta, Pathanamthitta, Kerala

Abstract- *The commonly used techniques to provide fault tolerance in distributed application is snapshot. It is one of the widely used techniques for application in scientific computing database, telecommunication and critical mission applications. The aim of this snapshot security protocol is which calculates consistent global snapshot for distributed application running on radar networks inorder to add more reliability and high availability to these systems.The new designed protocol is non blocking in nature and it avoids unnecessary computation. To minimize the amount of information we use the dependability concept for minimum number of processes to take snapshot. The number and size of control messages is independent of the increase in number of process in the system.*

Keywords- snapshot, VANET, distribution application; global state, fault tolerance, global state, snapshot.

I. INTRODUCTION

In various mission critical applications, where privacy and integrity of data are very important to ensure uninterrupted operation of services,for this fault tolerant radar networks systems are widely used.One of the fault tolerant technique is snapshot that allows a system to roll back to a most recent failure-free consistent global snapshot .The main challenges in implementing snapshot protocol is maintaining low overhead otherwise, the cost will outweigh its potential benefit.Inorder to avoid this issue we propose a non-blocking coordinated snapshot protocol running on radar network for distributed applications.This protocol forces a minimum number of processes to take snapshot.An initiating radar station sends snapshot requests only to those radar stations which have communicated in the last interval of the snapshot. The faulty radar station will recover from failure because all application messages are logged at the home station of the receiver radar station.None of the radar station is affected by this subsequent recovery and fault.The dependability concept is used to minimize the amount of information for taking snapshot with minimum number of process.The number of control messages is independent of the increase in the number of processes in the radar network system.This paper explains some relavant works, proposed snapshot protocol comparative study, correctness proof and performance evaluation.

II. LITERATURE REVIEW

The protocol proposed by M.Nagpal and P.Kumar explains a coordinated blocking snapshot protocol. [1]This protocol blocks the communication while taking snapshot and broadcast request message to all processes for taking a snapshot.The main problem in this protocol is that they prevent a process from receiving application messages which makes the snapshot inconsistent.

The protocol proposed in [2] is the first non blocking snapshot coordination protocol in which markers sent snapshot request messages.The initiator in this protocol takes a snapshot and broadcast marker to all processes and on receiving the first marker it takes (each process) a snapshot request message.

The protocol proposed in [3] lies between coordinated and uncoordinated techniques for snapshot.In this process takes communication induced snapshots besides independent snapshot to reduce number of useless snapshots.

The protocol proposed in [4] is uncoordinated snapshot protocol is designed where each process takes snapshot independently without coordinating.

The proposed protocol in [5] there is a possibility of domino effect which causes loss of a large amount of useful work.
The protocol proposed in [6] uses efficient DNA operations, but it relies on static hash mapping which can be vulnerable to attacks.

III. EXISTING SYSTEM

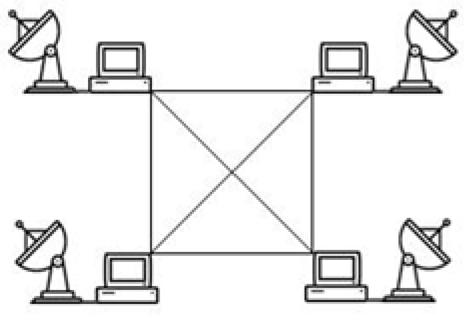
In the snapshot protocol the fundamental goal is to bring the distributed application to a consistent gobal snapshot when a failure occurs.

3.1. LIMITATIONS

- Log based snapshot protocol makes explicit use of the fact that a process execution can be modeled as a sequence of deterministic snapshot intervals.
- Process takes communication induced snapshots besides independent snapshots.
- Each process takes snapshot independently without coordinating with other processes.
- Domino effect which cause the loss of large amount of useful work
- Prevent a process from receiving application messages that could make the snapshot protocol inconsistent.

IV. PROPOSED SYSTEM (Proposed Protocol)

4.1Model of the system



The proposed non blocking protocol uses a snapshot sequence number(spki) for avoiding inconsistency.

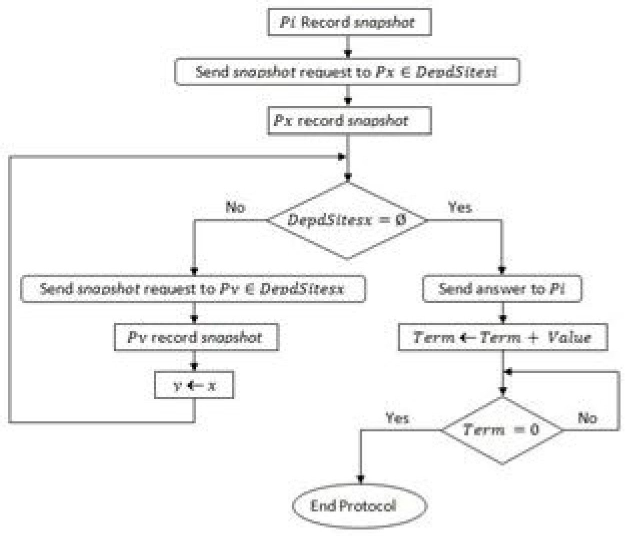
DataStructure(For every process pi)1.spki:Number of the last snapshot 2.Dpdsit:set of processes which have casual dependence relation with Pi.3.Termj:to detect protocol termination.4.Tempspki:to record last temporary.5 Permspki:to record last permanent snapshot.6.spty:1,7.Tempspkity=ϕ,8.Depdsity= ϕ

4.2Description of the protocol

It is to bring the distributed application to a consistent global snapshot after a failure is the fundamental goal of a snapshot protocol.

In the figure a radar network system which consist of a fixed number of radar stations interconnected.(wired)In this paper we use K to denote the total number of radar stations.Radar stations communication link is FIFO(First In First Out).Messages take finite amount of time during transmission in the network.Two types of messages.(a) Coordination messages.(b) Appication messages.(For coordination of the snapshot activity coordination message is used)

```
1.  Role of the initiator process Pi:
2.  Record (TempSpki);
3.  Spki := Spki + 1;
4.  Value = 1/Card(DpdSit);
5.  For All (Px ∈ DpdSit) Do
6.  Send SpkiReq (Px, Pi, Spki, Value);
7.  Endfor
8.  upon receiving answer  Px:
9.  Receive answer (Pi, Px, Value);
10. Term := Term + Value;
11. If (Term = 1) Then
12. For All (Px / x ∈ [1..n]) Do
13. Send ValidReq(Px, Pi);
14. Endfor
15. Endif
16. Role of every process Pj:
17. upon receiving snapshot request:
18. If Receive (Pj, Pi, Snpti, Value) Then
19. Snptj := Snpti;
20. Else
21. Receive SnptReq (Pj, Px, Value);
22. Endif
23. Record (TempSpkj);
24. If (DpdSit = ∅) Then
25. Send Answer (Pi, Pj, Value);
26. Else
27. Value = Value/Card(DpdSit);
28. For All (Px ∈ DpdSit) Do
29. Send SpkiReq (Px, Pj, Value);
30. Endfor
31. Endif
32. upon receiving validation request Pi:
33. Receive ValidReq (Pj, Pi);
34. If (TempSpki ≠ ∅) Then
35. PermSpki := TempSpki ;
36. Endif
37. For sending an application message to Px:
38. Send Message (Px, Pj, Spki, Msg);
39. upon receiving application message from Px:
40. Receive Message (Pj, Px, Snptx, Msg);
41. If (Spki< Sptx) Then
42. Record (TempSpki);
43. Spki := Spty;
44. Endif
45. Handle (Msg);
```



V. CONCLUSIONS

This paper, we have presented a new non-blocking snapshot protocol for distributed application running on radar network systems.One of the commonly used techniques is snapshot to provide fault-tolerance in distributed application. Therefore the application can operate even if one or more components have failed in the radar network. The important feature of the proposed snapshot protocol is; it records a very few number of snapshots with less coordinated message cost to assure a consistent global snapshot incomparison with others previous works.

VI. ACKNOWLEGEMENT

We would like to thank, Almighty God at first, without his grace and blessings this work would not have been possible. We would also like to thank the faculty members of Mount Zion College of Engineering, for their great support towards this work.

REFERENCES

[1] M. Nagpal, and P. Kumar, “Anti-Message Logging Based Checkpointing Algorithm for Mobile Distributed Systems,” *International Journal of Computer Applications*, Vol. 69, no. 14, pp. 21-27, May. 2013.

[2] S.S. Sathya, and K.S. Babu, “Survey of Fault Tolerant Techniques for Grid,” *Computer Science Review*, vol. 4, no. 2, pp. 101-120, May. 2010.

[3] J. Wu, and D. Manivannan, “A Fully Informed Model-Based Checkpointing Protocol for Preventing Useless Snapshots,” *Intl Jrnl of Parallel, Emergent and Distributed Sys*, vol. 28, no. 6, pp. 485-518, Nov. 2013

[4] M. Raynal, “Asynchronous Distributed Checkpointing,” In. *Distributed protocols for message-passing systems*, Springer, Berlin Heidelberg, pp. 189-218, 2013.

[5] M. Elnozahy, L. Alvisi, Y-M. Wang, and D.B. Johnson, “A Survey of Rollback-Recovery Protocols in Message-Passing Systems,” *ACM Computing Surveys*, vol. 34, no. 3,375-408, Sep. 2002.

[6] Y. Luo, and D. Manivannan, “HOPE: A Hybrid Optimistic Checkpointing and Selective Pessimistic Message Logging Protocol for Large Scale Distributed Systems,” *Future Generation Computer Systems*, vol. 28, no. 8, pp. 1217-1235, Oct. 2012.