Performance Analysis of Hierarchical Fault Tolerance Protocol for Mobile Agent Systems

Heman Pathak [1] *Kumkum Garg* [2]

Abstract

A Mobile Agent (MA) is autonomous and identifiable software process that travel through a network of heterogeneous machine and act autonomously on behalf of user. Improving the survivability of MA in presence of various faults is the major issue concerns with implementation of MA.

This paper presents a brief introduction of Hierarchical Fault Tolerance Protocol (HFTP) for Mobile Agents, which can tolerate host failure, system failure as well as link failure by grouping the hosts within a network and rear guard based migration of MA in the global network.

This paper also analyzes the HFTP for its performance in presence of faults by using Colored Petri Net (CPN) based architectural model of HFTP.

I. Introduction

MA [1], [2] is an emerging technology that is becoming increasingly popular. Although potential usefulness of the MA computing paradigm has been widely accepted, MA technology has not yet found its way into today's more prominent applications. Before MA applications begin to appear on a large scale, Mobile Agent System (MAS) needs to provide infrastructure services to facilitate MA development. Among these are security, management of MA, fault tolerance, and transaction support. In this paper we are discussing the fault-tolerance issues related to MA. Faults that can occur in MA life cycle have been identified as – host failure, link failure, MAS failure, programming error or some uncaught exception.

Although several commercial and research MASs have already been developed, they either do not fully provide support for fault tolerance mechanisms [3], [4], [5], [6], [7] or provide only a partial solution to the problem. We have proposed a Hierarchical Fault Tolerance Protocol (HFTP) [8], [9], [10] for MAs and modeled it by using Colored Petri Net (CPN) [11], a powerful modeling tool for complex systems [12], [13], [14], [15].

2. Hierarchical Fault Tolerance Protocol

Based on the experienced gained from prior works, this approach has been designed to use fault masking by grouping hosts within a network at one level while fault detection and recovery by using rear guards at another.

HFTP consists of three layers. Different kinds of faults are detected and tolerated at different layers. Server at lowest layer is ***Personal Daemon Server (PDS),*** at middle layer ***Local Daemon Server (LDS)*** and at highest layer ***Global Daemon Server (GDS)***. These three layers have been implemented as proxy servers.

A. Personal Daemon Server (PDS):

It watches the MAS as well as the all MAs running on the MAS. In case MAS or its components fail, PDS is responsible to inform all other group members about the faults as well as to initiate recovery of MAS. PDS is installed on each host of the network that can host the MA.

[1] Dept. of Computer Science, Gurukul Kangri Vishwavidyalaya, Haridwar, India hemanp@rediffmail.com, nipursingh@hotmail.com

[2] Dept. of Electronics & Comp. Eng., IIT, Roorkee kgargfec@iitr.ernet.in

B. Local Daemon Server (LDS):

It is responsible to detect the host failure as well as for executing all group communication services within the group like distributing the load among the group impartially, when MA is submitted to the group as well as when a host fails. Although LDS is installed on each host, but within a group only one host is in-charge for taking decision while all other group members watch each other. LDS is installed on each host of the network as well as at the router.

C. Global Daemon Server (GDS):

It is responsible for receiving the MA from other networks and then passing them to the appropriate group of its own network. It performs all functions required for fault tolerant migration of MA in the global network of networks. In case all members of a group fail, it is responsible to recover MAs running in that group. GDS is installed on routers.

3. Parameters for Simulations

Before starting the simulation, some parameters are required to be assumed while some are generated randomly or calculated during simulation. The assignment is based on the assumption that packet transmission time is fixed and it is independent of place, time or load of network. The MA takes constant time to execute on any host. Transmission time for MA is 200 time units (TU) and for Acknowledgement is 100 TU. Logging, host assignment and recovery time is 50 TU. Execution Time for MA/host is 450 TU.

4. Overhead of using HFTP

Every fault tolerance mechanism adds some overhead to the existing systems in terms of time, space or requirement to maintain reliability. In order to observe the overhead due to HFTP, in terms of MA trip time and network overhead generated by it, we have modeled a protocol having no support for fault tolerance (without HFTP) and then compared the performance of the system using HFTP in a fault-free environment.

Figure 1 shows that trip time increases linearly for both HFTP and without HFTP. Since simulation has been performed in an ideal fault-free environment, here all the steps including execution and migration of MAs takes constant time, trip time increases linearly as the number of servers in its itinerary increases. Trip time is higher for HFTP because it requires logging the arrival and departure at the router, also the in-charge has to execute a deterministic algorithm to assign a host to the arrived MA in the group. Check-pointing is also required, even if no faults occur during MA execution.

Figure 2 shows that, although network overhead increases linearly for both cases, HFTP generates more overhead as the number of servers increase. This is because HFTP requires sending an acknowledgement for every migrating MA to detect link failure. Number of acknowledgements increases with the number of servers in the MA itinerary. Further implementation of group communication services generates network overhead in Local Area Networks.

5. Fault cases and tolerance through HFTP

In order to observe the performance of HFTP in the presence of faults, we have generated various faults in the CPN model of HFTP by changing the failure probability rate and then measured its performance in terms of trip time and network overhead.

For each case, a MA with ten servers in its itinerary is launched. Simulation has been repeated hundred times and its average value has been used to predict the performance pattern.

A. Case 1: Mobile Agent System Failure

The MAS fails during execution of a MA according to its failure probability rate. Here it is assumed that at least one active host within each group to share the load of the failed host and MA does not get blocked.

Figure-3 shows that system failure rate is tolerated by HFTP. The global network overhead remains constant, while the local network overhead increases exponentially with failure rate, because a recovered agent may fail again and again. Every time a failed agent recovers, it adds some network overhead locally as it is required to transfer the recovered agent to its new host.

Figure- 4 show that trip time increases exponentially as failure rate increases, because every time the system fails, more time and extra execution steps required for detecting the fault, recovering the agent and resuming its execution on the new host. For small failure rates, the performance does not degrade too much.

B. Case 2: Host Failure

During the execution of MA, the host machine may go down and all MAs hosted by it are lost. HFTP tolerates host failure provided there is at least one active host per group to avoid blocking. Host failure is tolerated in the same way as system failure so the performance is expected to be similar as in case of system failure. Unlike system failure, where a fault is detected by a thread, host failure is detected by other members of the group. The fault detection mechanism does not increase any load. Again, in case of host failure, only local network overhead increases, global package transfer remains constant.

Figure-5 shows that local network overhead increases almost exponentially with host failure rate. Result is same as in case of system failure, but slightly better as recovery of a failed host is the responsibility of network manager and has not been considered while system recovery is initiated by PDS.

Figures- 6 verify our claim that host failure is tolerated by HFTP and gives a similar performance as in case of system failure provided blocking does not occur.

C. Case 3: Link Failure

A link may fail during the migration of a MA from one host to another within a Local Area Network or between networks. Due to link failure, a MA may get lost on its way. HFTP tolerates link failure unless it leads to network partitioning. Link failure during migration within a network is tolerated by using TCP and has not been used for performance analysis. Due to link failure, a MA or acknowledgement may get lost in a global network and require retransmission of acknowledgements or probes, which not only increase network overhead and execution steps but also trip time. Since failure is detected only after waiting time is over, so delay increases more as compared to network overhead and number of execution steps.

Figure-7 shows the pattern of network growth overhead as link failure rate increases. It also proves our claim that HFTP is able to tolerate link failure. However when failure rate is more than 25%, overhead increase significantly.

Figures-8 shows that if failure rate is more than 25%, performance of the system degraded significantly and more time is required while for low failure rate performance is comparable.

Since MA failure detection and recovery takes place only at the host, it does not increase network overhead and not been observed.

D. Case 4: Agent Failure

Since MA failure detection and recovery takes place only at the host, it does not increase network overhead and not been observed.

Figure- 9 shows that trip increases almost linearly and performance is not degraded much until failure rate goes beyond 30%.

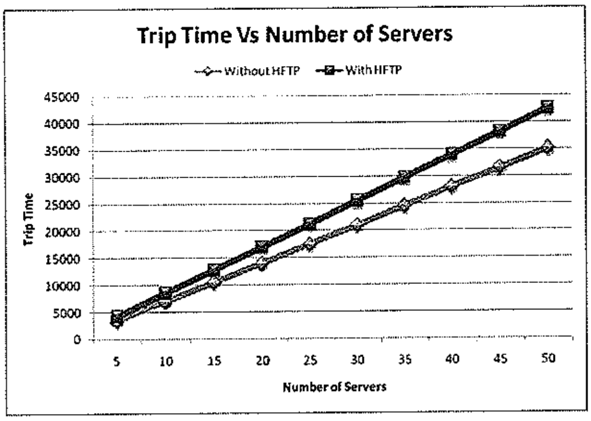

Figure 1: Overhead of the HFTP in terms of trip time in fault free environment

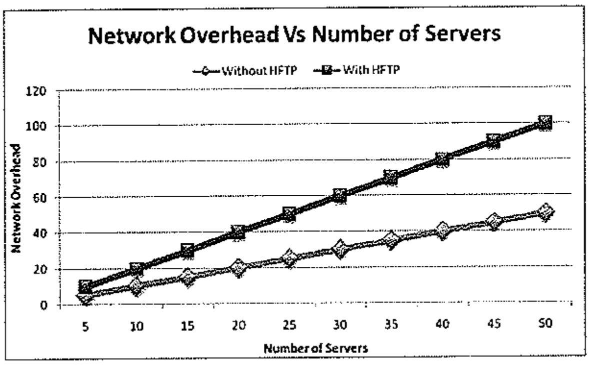

Figure 2: Network Overhead of the HFTP in fault free environment

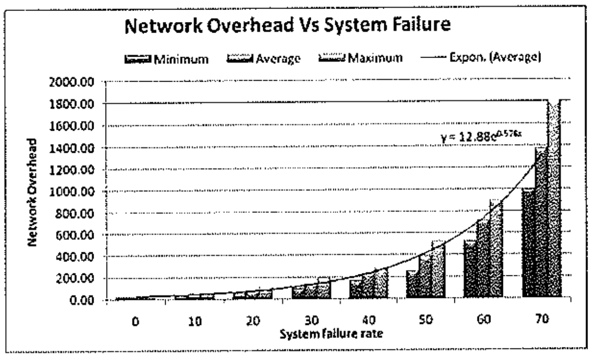

Figure 3 : Network Overhead by HFTP in the presence of System Failure

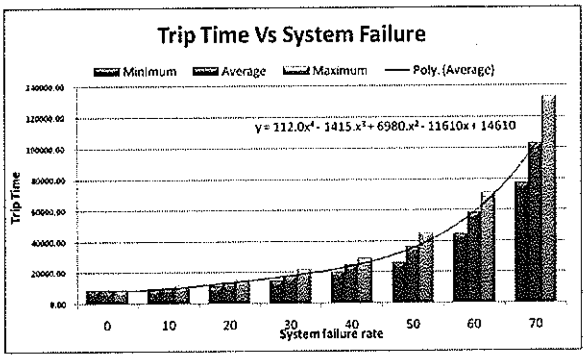

Figure 4 : Performance in terms of trip time in the presence of System Failure

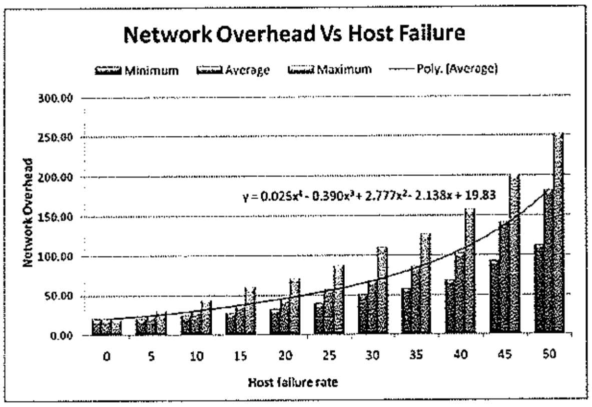

Figure 5: Network Overhead generated by HFTP in the presence of Host Failure

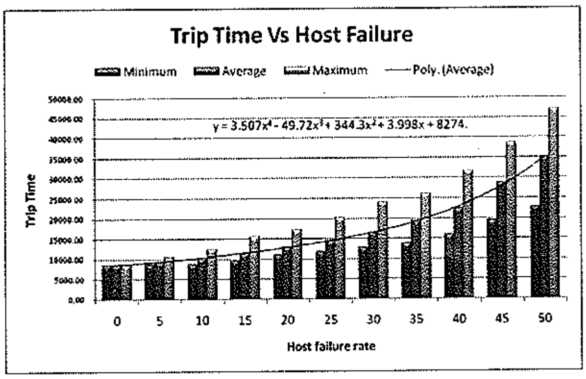

Figure 6 : Performance in terms of trip time in the presence of Host Failure

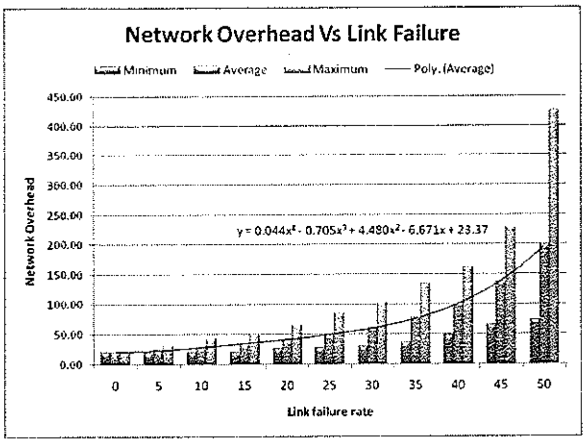

Figure 7: Network Overhead generated in the presence of Link Failure

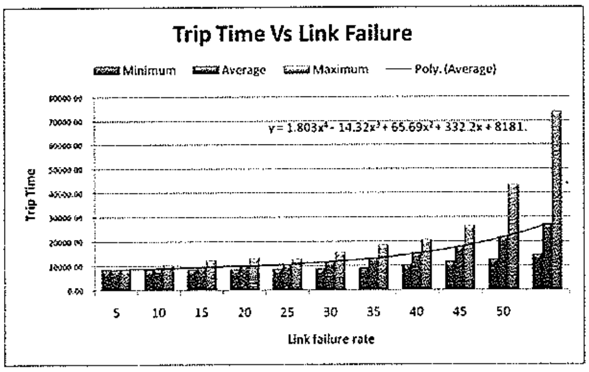

Figure 8 : Performance in terms of trip time in the presence of Link Failure

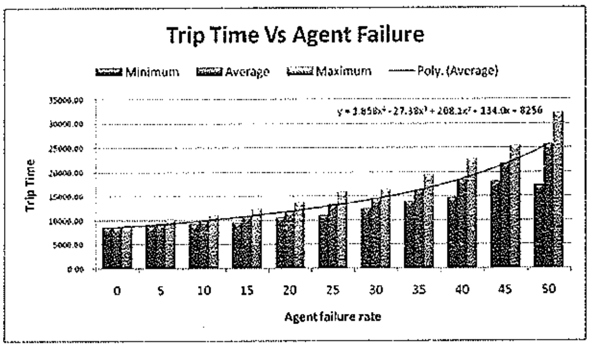

Figure 9 : Performance in terms of trip time in the presence of Agent Failure

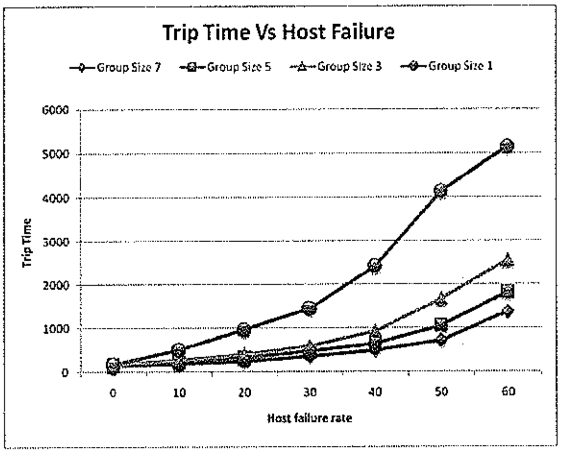

Figure 10 : Trip time Vs Host Failure rate for different Group Sizes

6. Conclusion

The results show that HFTP is able to tolerate all kinds of faults without degrading the performance significantly. For low failure rate, the survivability of MA in HFTP is ensured and it is able to achieve tolerance without increasing network overhead or time delay substantially. If host/ system failure rate increases, then the MA may be blocked within a group. This blocking may be avoided by properly selecting the group size. But these failures are not frequent so the results are acceptable. Link failures in the global network may lead to network partitioning. This extreme case of link failure is tolerated by HFTP, if an alternative list of hosts is defined in its itinerary. Also, if the order of the itinerary is not fixed, the MA can visit some other host in its itinerary and may try to visit the disconnected host latter when at least one of the links resumes. In the worst case when all the target hosts are disconnected with current network, MA will be blocked within the network.

References

[1]. J. Chen, "*A Hierarchical Fault-Tolerance Framework for Mobile Intelligent Agent Systems*," Master of Science thesis, The University of British Columbia, Faculty Of Graduate Studies, Department of Computer Science, April 2002.

[2]. D. Kotz, R. Gary, "*Agent Tcl: Targeting the Need of Mobile Computer*", IEEE Internet Computing, pp. 58-67 July/August 1997.

[3]. R. Michael, T. Y. Wong, "*A Progressive Fault Tolerant Mechanism in Mobile Agent Systems*," in Proc. of the 7th world Multi-conference on Systematics, Cybernetics and Informatics, Vol. IX, Orlando, Florida, July 2003, P.P. 299-306

[4]. S. Mishra, Y. Huang, "*Fault Tolerance in Agent-Based Computing Systems*," Proceedings of the 13th ISCA International Conference on Parallel & Distributed Computing, Las Vegas, N V. August 2000.

[5]. S. Mishra, "*Agent Fault Tolerance Using Group Communication*," Proceedings of the 2001 International Conference on Parallel & Distributed processing Techniques and Application (PDPTA-2001), Las Vegas, N V. June 2001.

[6]. H. Pals, S. Petri, and C. Grewe, "*FANTOMAS : Fault Tolerance for Mobile Agents in Clusters*," Proceedings International Parallel and Distributed Processing Symposium (IPDPS), 2000, Worksoft J.D.P. Rollim (ed.) pp. 1236-1247, 2002.

[7]. R. B. Patel, K. Garg, "*Fault-Tolerant Mobile Agents Computing On Open Networks*", www.caip.rutgers.edu/~parashar/AAW-HiPC2003/patel-aaw-hipc-03.pdf

[8]. H. Pathak, K. Garg, Nipur, "*Fault Tolerance Approaches for Mobile Agent Systems:A Parameter Based Comparative Study*", in proceedings of National Conference on Trends of Computational Techniques in Engineering (TCTE '2004), SLIET Punjab (India), pp 119-123, October 2004.

[9]. H. Pathak, Nipur, "*Hierarchical Fault Tolerance Model for Mobile Agent Systems*", National Conference on Statistics, Computer & Applications, November 2005, Amarawati, M.H. India.

[10]. H. Pathak, K. Garg, Nipur, "*Fault Tolerance Problem & Challenges for Mobile Agent Systems and Proposed Solution*", in proceedings of National Conference on Communication & Computational Techniques: Current & Future Trends (NCCT – 06), DIT Dehradun, India, pp 381-386, February 2006.

[11]. H. Pathak, K. Garg, Nipur, "*CPN model for Hierarchical Fault Tolerance Protocol for Mobile Agent Systems*", in proceedings 2008 International Conference of Networks (ICON 2008), New Delhi, India, December 2008.

[12]. K. Jensen, "*Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use*," Volume 1,2 and 3, Monographs in Theoretical Computer Science, Springer-Verlag. ISBN: 3-540-60943-1, 3-540-58276-2, 3-540-62867-3

[13]. CPN Tool website: www.daimi.au.dk/CPNtools

[14]. Ouyang, C. and Billington, J.,"*On verifying the Internet Open Trading Protocol*," In:Proceedings of EC-Web 2003, Lecture Notes in Computer Science 2738, Springer-Verlag, 2003, 292-302

[15]. Ouyang, C. and Billington, J., *"An improved formal specification of the Internet Open Trading Protocol,"* In: Proceedings of the 2004 ACM Symposium on Applied Computing, Nicosia, Cyprus, 2004, 779-783.

Author's Biography

Dr.Heman Pathak is currently working as Assistant professor at Gurukul kangri Vishwavidyalaya, Haridwar. She has published nine research papers in international and national level conferences and journals. Her research areas are parallel and distributed computing and mobile agent technology.

Dr. Kumkum Garg is currently working as Professor of Computing in Department of Electronics & Computer Engg.,IIT Roorkee. She has published eighty nine research papers in international and national level conferences and journals and organizes five conference and seminars. She has published three books and received various honor form national and international agencies. Her areas of interests are Computer Networks, Mobile Computing, Artificial Intelligence.

Dr.Nipur is currently working as Associate professor at Gurukul kangri Vishwavidyalaya, Haridwar. She has published twenty two research papers in international and national level conferences and journals. Her research areas are Interconnection networks, image processing, mobile adhoc network and mobile agent technology.