Comparative Performance of Hierarchical Fault Tolerance Protocol for Mobile Agent Systems

Heman Pathak, Nipur
Dept. of Computer Science
Gurukul Kangri Vishwavidyalaya
Haridwar, India
hemanp@rediffmail.com, nipursingh@hotmail.com

Kumkum Garg
Dept. of Electronics & Comp. Eng.
IIT, Roorkee
Roorkee, India
kgargfec@iitr.ernet.in

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. A CPN model of HFTP has been used to generate simulation results for different scenarios by using various monitoring and data collection tools provided by CPN. The parameter used for analysis is trip time (time required by MA to complete its itinerary) and network overhead (number of packets transfer required over network). In order to check the efficiency of HFTP, we have modeled two other protocols Progressives Fault Tolerance Mechanism (PFTM) based of rear guards and Server Group based Agent Recovery Protocol (SG-ARP) with which our model is inspired. Performances of HFTP, PFTM and SG-ARP are compared on the basis of data obtained from simulations for different fault cases. The results of the simulation have been presented and analyzed.

Keywords-Mobile Agents, Fault Tolerance, Colored PetriNet

I. INTRODUCTION

MA [1], [2] is an emerging technology that is becoming increasingly popular. Before MA applications begin to appear on a large scale, Mobile Agent System (MAS) needs to provide infrastructure services to facilitate MA development. In this paper we are discussing the fault-tolerance issues related to MA, which is still a major obstacle that keeps the MA running only in research labs. 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].

II. HIERARCHICAL FAULT TOLERANCE PROTOCOL

HFTP consists of three 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.

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 that can host the MA 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 is also responsible to perform 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. It is installed on routers.

III. PROGRESSIVE FAULT TOLERANCE MECHANISM

In this protocol, two types of agents exist, one performing the actual computation called *Actual Agent* while the other is used to detect and recover the actual agent and is called *Witness Agent*. It uses logging and checkpoint data to perform partial or rollback recovery. To detect and recover an actual agent's failures, the witness agent monitors whether the actual agent is alive or dead. When the actual agent completes its dedicated work on a server and resumes its journey to the next server, it spawns a new witness agent at the current server. The witness agent waits for messages from the migrating agent. The agent logs an entry on arriving and before migrating from a

host. The purpose of log entries and messages is to guarantee that the actual agent has finished up to a certain point of its execution. If a server failure occurs between a log entry and its corresponding message, it can determine when and where the actual agent failed. Recovery starts after detecting the fault and by sending a probe.

IV.CPN MODEL FOR PFTM

The basic environment for executing a MA for PFTM and HFTP is the same since PFTM and HFTP both use witness agents to detect the faults. The assumptions made to simplify the model are same and are not discussed here. The global network for PFTM is also modeled in the same way as HFTP. The main page which models a host and all its components are discussed here.

A. Host Page

The main page of PFTM is the host page which models a host. Figure 6.16 shows details of the host page for PFTM.

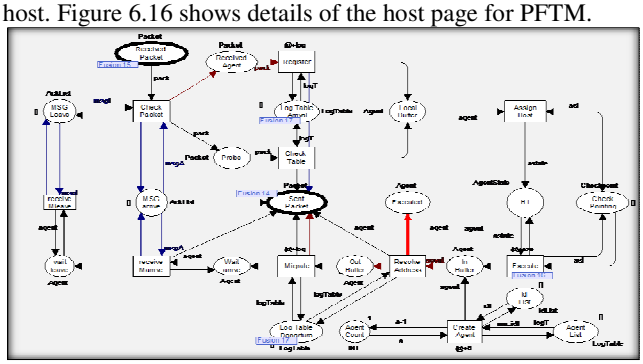

Figure A.1 – CPN Page models the Host for PFTM

V.SERVER GROUP BASED AGENT RECOVERY PROTOCOL

SG-ARP [4], [5] is inspired by the exactly once model. It provides fault-tolerance properties to MAs by replacing an agent server by a group of agent servers running on different nodes on the same local area network. The group of servers that replace an agent server is called an agent server group. In addition, SG-ARP assumes the availability of a stable storage service accessible to all group members of an agent server group. Each agent that arrives at an agent server group executes on one of the group members identified by a deterministic algorithm. An agent checkpoints its state in stable storage accessed by all members of the group any time, before migrating from the group. The members of an agent server group divide the load brought upon by the visiting agents among themselves. A group has a single logical name and appears as a single agent server to other agent servers in the distributed system.

When a node in an agent server group fails, the group membership protocol notifies all the remaining group members about this failure and creates a new group. All the agents that were running on the failed member are distributed to the remaining group members. The remaining group members then restart those agents from their last checkpoint state. A group is implemented via a group communication service that provides support for maintaining synchronization and state consistency between different group members.

The protocol uses TCP for agent transfer and can recover from the loss of an agent during migration due to communication failure. But in order to tolerate network partitioning, it allows the user to specify a list of alternate agent server group. If migration is still unsuccessful, it throws an exception. Although this protocol claims to achieve exactly once as well as network partitioning, the use of TCP, replication and check-pointing makes it slow and costly.

VI.CPN MODEL FOR SG-ARP

SG-ARP has been modeled using the hierarchical CPN, where different pages model different components of the protocol. Assumptions for the model are again the same as in case of HFTP. The SG-ARP model uses multiple pages.

A. GlobalNetwork

The page at the highest level of hierarchy models the global network. This page has been modeled in the same way as the *GlobalNetwork* page in HFTP. The only difference is that this page uses TCP communication, so the delay and overhead through this network is different.

B. LocalAreaNetwork

The page at the second level of hierarchy models the Local Area Network. Figure 6.17 gives the description of *LocalAreaNetwork* page.

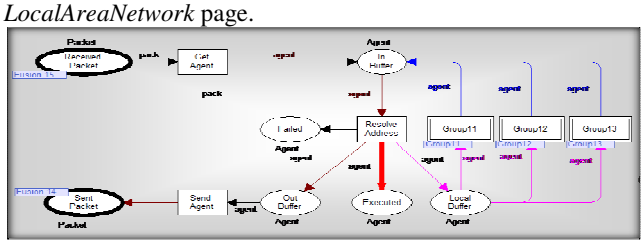

Figure B.1 – CPN Page models the Local Area Network for SG-ARP

C. ServerGroup

The Page at the next level is *ServerGroup*, which models the Agent Server group. Figure 6.18 shows the description of this page.

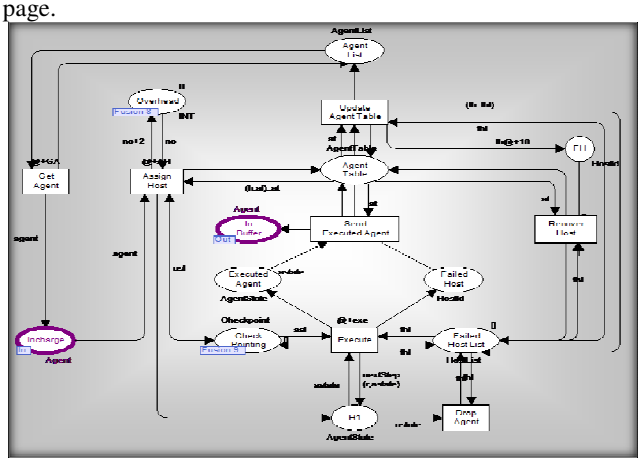

Figure C.1 – CPN Page models the Server Group for SG-ARP

VII. Performance Analysis

Performance analysis of fault tolerance approaches is very difficult due to the variety of approaches and large number of system parameters. Currently, there are three methods for evaluation of performance, namely; analytical, real-system measurement and simulation methods. Simulation is a flexible, reproducible and inexpensive method for performance analysis, especially for the MA paradigm. In this paper, comparative performance analysis of HFTP has been done on the basis of simulation results obtained from CPN model of HFTP.

VIII. 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	= 200 time units
Transmission time for Acknowledgement	= 100 time units
Logging (Arrival/Departure) time	= 50 time units
Host assignment for In-charge	= 50 time units
Execution Time for MA/host	= 450 time units
Recovery time for Mobile Agent	= 50 time units
Time to Checkpoint data and state	= 100 time units

IX. Comparison for different fault cases

In this section, we compare the performance of HFTP with PFTM and SG-ARP for various cases.

A. Case 1: No Failure

The performance of all the three protocols has been compared in a fault-free environment to check the overhead caused by them and shown in Fig. 1&2.

It is clear from Figure-1 that there is not much difference between the performances of the protocols in terms of trip time. HFTP is slightly better than others.

Figure-2 shows that HFTP is the most effective in terms of bandwidth consumption among these three protocols. The network overhead is maximized for SG-ARP due to the use of TCP. PFTM requires more message transfer over the network than HFTP for fault detection as well as for recovery.

B. Case 2: Link Failure

The performance of all three protocols has also been observed in the presence of link failure. Link failure has been generated by changing the failure probability rate.

Figure-3 shows that the network overhead increases almost exponentially as network failure rate increases. For network failure rate less than 25%, PFTM is better than SG-ARP, but for higher failure rate, SG-ARP gives better results. Since PFTM watches a MA and its host from another host and communicates through message passing, so as link failure rate increases, message loss also increases and more messages are required to perform recovery, which increases network overhead exponentially.

Figure-4 shows the comparative performances of all three protocols in terms of trip-time in the presence of link failures. PFTM is slowest among the three, as it detects faults only after waiting time is over. Also for recovery it uses unreliable links. Trip time increases exponentially for PFTM with failure rate. For failure rates less than 25%, HFTP is the best. But, as failure rate increases UDP proves to be slower than TCP [14] and hence TCP based SG-ARP gives better results. Since in real applications, link failure rate is not high, hence HFTP is the protocol of our choice.

C. Case 3: Host Failure

To model host failure during MA execution, the host failure rate is set by the user. Accordingly, the machine may fail during execution. Host/server failure is tolerated by SG-ARP and HFTP by detecting the fault and then by distributing the load of the failed host among the active members of the group. PFTM can detect host failure, if it does not receive messages from the target host. But it cannot continue execution of MA until the host recovers. Although the MA does not get lost, it will be blocked until the host recovers. So its performance has not been compared with HFTP and SG-ARP.

Figure-5 & 6 show the comparative performance of HFTP and SG-ARP in terms of network overhead and trip time verses host failure rate respectively. It can be seen from these figures that both protocols masks host failure by using the concept of grouping but HFTP gives improved results due to better check-pointing and communication.

It has been assumed here that there are enough hosts in each group and there is always at least one active host per group to share the load of the failed hosts. Failed hosts are recovered by using networks own fault tolerance mechanism. This assumption has been made to compare the performance of both protocols in terms of execution time and network overhead, in the presence of host failure.

D. Case 4: Agent Failure

PFTM and HFTP are able to detect and tolerate agent failure and restart MA execution from the last check-pointed state. SG-ARP is not able to detect or tolerate such faults because it has no such provision.

For detecting or recovering agent faults, HFTP does not require any message passing. Because the PDS installed at the same host detects agent failure and recovery is performed from the last check-pointed state. So there is no network overhead due to agent failure for HFTP. However, PFTM detects faults only if it fails to receive message^{leave} and sends the probe to recover the failed agent. Hence, there is increase in the network overhead. Figure-7 shows that Network Overhead is constant for HFTP but increases linearly for PFTM.

Figure-8 shows that trip time increases linearly for HFTP but rate of increase is high for PFTM.

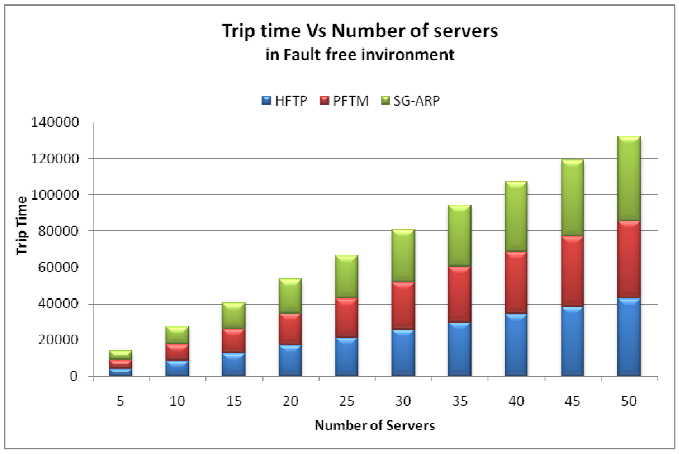

Fig. 1: Comparison based on trip time in fault free environment

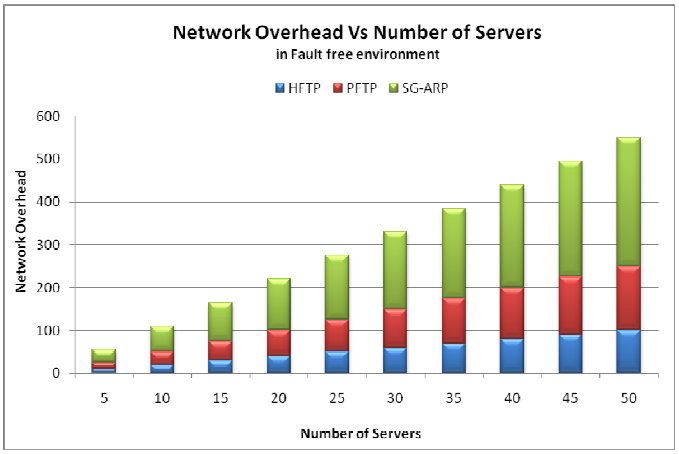

Fig. 2: Comparison based on network overhead in fault free environment

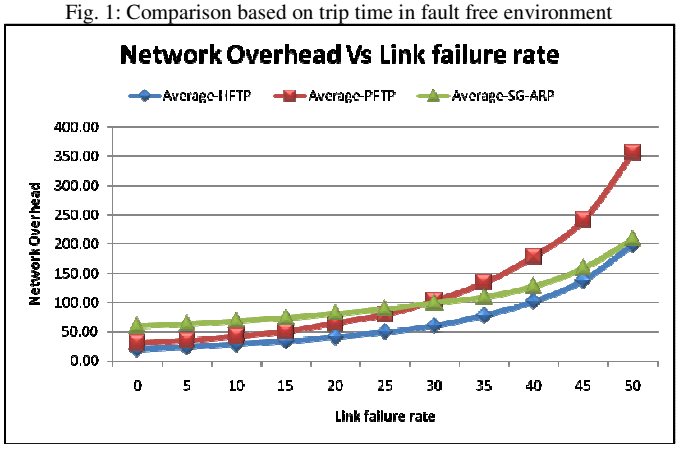

Fig. 3: Comparison based on network overhead in presence of Link failure

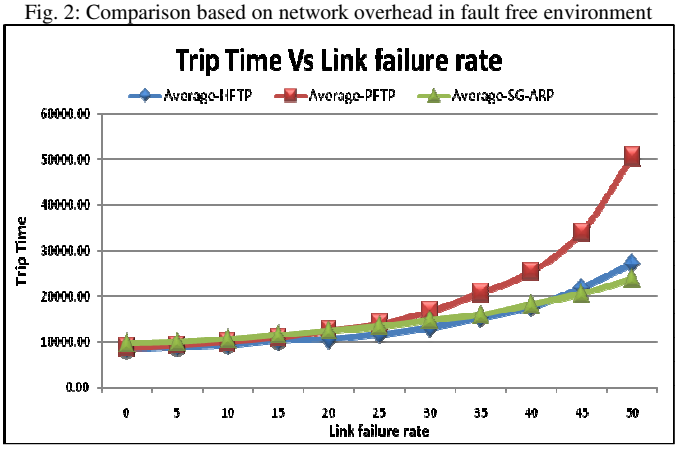

Fig. 4: Comparison based on trip time in presence of Link failure

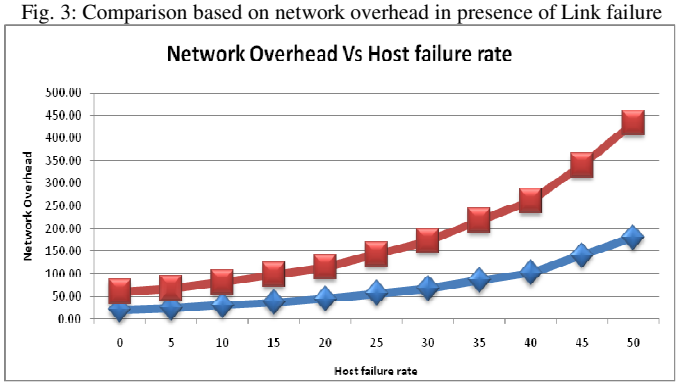

Fig. 5: Comparison of HFTP & SG-ARP (Net. Overhead Vs Host failure)

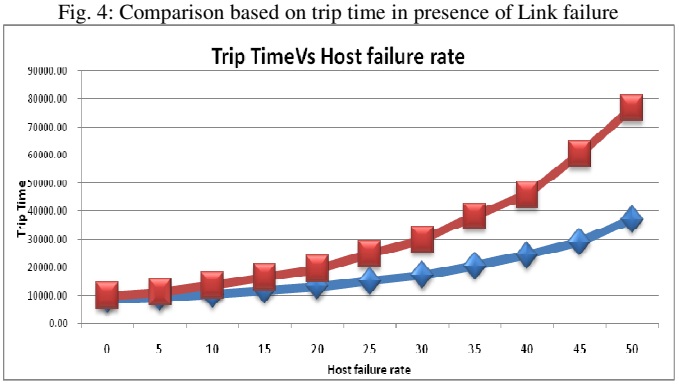

Fig. 6: Comparison of HFTP & SG-ARP (Trip Time Vs Host Failure)

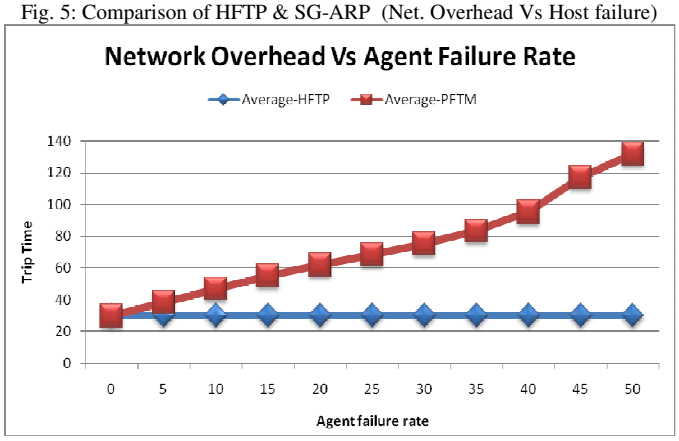

Fig. 7: Comparison of HFTP & PFTM (Network overhead Vs Agent failure)

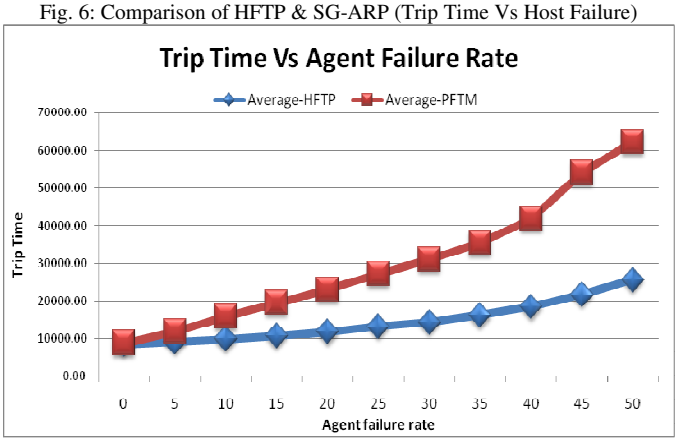

Fig. 8: Comparison in terms of trip time in presence of Agent failure

X. Conclusion

The results of comparison show that HFTP is able to tolerate all kinds of faults and has proved to be more efficient than SG-ARP and PFTM. 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

[2]. 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.

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

[4]. 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

[5]. 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.

[6]. 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.

[7]. 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.

[8]. 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

[9]. 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.

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

[11]. 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.

[12]. 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.

[13]. 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

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

[15]. George Xylomenos and George C. Polyzos,"*TCP and UDP Performance over a Wireless LAN",* in proceedings of the IEEE infocom 1999, pp. 439–446.