ANALYSE THE PERFORMANCE OF MOBILE PEER TO PEER NETWORK USING ANT COLONY OPTIMIZATION

S.J.K. Jagadeesh Kumar[1] and R.Saraswathi[2]

[1] Professor, Dept. of Computer Science and Engineering, SKCT, Coimbatore.
[2] PG Student, Sri Krishna College of Technology, Coimbatore, Tamil Nadu, India
surswathi@gmail.com

ABSTRACT

A mobile peer-to-peer computer network is the one in which each computer in the network can act as a client or server for the other computers in the network. The communication process among the nodes in the mobile peer to peer network requires more no of messages. Due to this large number of messages passing, propose an interconnection structure called distributed Spanning Tree (DST) and it improves the efficiency of the mobile peer to peer network. The proposed method improves the data availability and consistency across the entire network and also reduces the data latency and the required number of message passes for any specific application in the network. Further to enhance the effectiveness of the proposed system, the DST network is optimized with the Ant Colony Optimization method. It gives the optimal solution of the DST method and increased availability, enhanced consistency and scalability of the network. The simulation results shows that reduces the number of message sent for any specific application and average delay and increases the packet delivery ratio in the network.

KEYWORDS

Mobile Peer to Peer Network, Distributed Spanning Tree, Global Replica Management, ACO, LRM,ORCS.

1. INTRODUCTION

1.1. Mobile Peer to Peer Network

In a mobile P2P network, the mobile nodes are connected in mesh network within their communication range. Files can be shared directly between systems on the network without the need of a central server. The communication among the mobile nodes is to be carried in multi-hop fashion due to the design considerations such as radio power limitation and channel utilization. Any communication with external networks is performed through the AP which consumes relatively more time. In a mobile P2P network, the "peers" are computer systems which are connected to each other via the Internet.

A Mobile P2P network is composed of mobile hosts that are free to move around randomly, and to organize and collaborate together to share information among themselves. Files can be shared directly between systems on the network without the need of a central server. In other words, the P2P network is called a distributed structure if the participants share a part of their own resources. These shared resources are necessary to provide the service offered by the network. The participants of such a network are both resource providers and resource consumers. The P2P network has the following characteristics:

DOI: 10.5121/ijci.2013.2302

- All nodes are both clients and servers
- Provide and consume data
- Any node can initiate a connection
- No centralized data source
- Nodes contribute content, storage, memory, CPU
- Nodes are autonomous (no administrative authority)
- Network is dynamic: nodes enter and leave the network “frequently”
- Nodes collaborate directly with each other
- Nodes have widely varying capabilities

The various benefits of P2P network has the efficient use of resources, scalability, reliability, ease of administration, Anonymity, Highly dynamic environment and Ad-hoc communication and collaboration. The various P2P applications are File sharing, Multiplayer games , Collaborative applications, Distributed computation and Ad-hoc networks. The various Challenges of the P2P network are Decentralization, Scalability, Performance, Anonymity, Fairness, Dynamism, Security, Transparency, Fault Resilience and Robustness.

1.2. Distributed Spanning Tree

The distributed spanning tree (DST) is an overlay structure designed to be scalable. It supports the growth from a small number of nodes to a large one. The DST is a tree without bottlenecks which automatically balances the load between its nodes. The DST breaks the common assumption that a tree is build of leaves and intermediate nodes. In a DST every nodes are equal. The nodes are put together into small cliques. Then, the cliques are put together into small cliques of higher level recursively. The cliques are represented in each node by a routing table. The memory space complexity of the routing tables is O(log(n)) for a n nodes DST.

The section 2 describes the related work and section 3 describes the proposed system and section 4 describes the simulation scenario and section 5 describes the conclusion of the work.

1.3. Ant Colony Optimization

In computer science and operations research, the ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. This algorithm is a member of the ant colony algorithms family, in swarm intelligence methods, and it constitutes some metaheuristic optimizations. Initially proposed by Marco Dorigo in 1992 in his PhD thesis, the first algorithm was aiming to search for an optimal path in a graph, based on the behaviour of ants seeking a path between their colony and a source of food. The original idea has since diversified to solve a wider class of numerical problems, and as a result, several problems have emerged, drawing on various aspects of the behaviour of ants.

2. RELATED WORK

In [6], Takahiro Hara proposed new consistency maintenance based on local conditions such as location and time need to be investigated. It attempts to classify different consistency levels according to requirements from applications and provides protocols to realize them. In [8], Ren Xun-yil et al proposed a consistency technique based on a replica clustering coefficient to classify replica nodes into multi-levels. Replica consistency has been maintained in which the updating of the data item is performed at first-level replica nodes initially and then it is propagated to the next level of nodes in sequence. Though efficiency is proved in terms of response time and the number of message passes required. In [9], Chun-Pin et al propose a Dynamic Maintenance Service to maintain the data in gird environment. The Bandwidth Hierarchy based Replication algorithm was proposed to maintain the replica dynamically in grid environment. In [9],Chao-Tung et al proposed a One-way Replica Consistency Service (ORCS) for grid environment to resolve the consistency maintenance issues and also balancing the trade off between the improving data Access performance and replica consistency.

In [11] Sang-Min Park proposed a novel dynamic replication strategy; called BHR (Bandwidth Hierarchy based Replication). It tries to maximize locality of file to reduce data access time. However, grid sites may be able to hold only small portion of overall amount of data since very large quantity of data is produced in data grid and the storage space in a site is limited. Therefore, effect from this locality is limited to a certain degree. BHR strategy takes benefit from other form of locality, called network-level locality. In [12], Haiying Shen propose an Integrated file Replication and consistency Maintenance mechanism (IRM) that integrates the two techniques in a systematic and harmonized manner. It achieves high efficiency in file replication and consistency maintenance at a significantly low cost. Instead of passively accepting replicas and updates, each node determines file replication and update polling by dynamically adapting to time-varying file query and update rates, which avoids unnecessary file replications and updates. It dramatically reduces overhead and yields significant improvements on the efficiency of both file replication and consistency maintenance approaches. In [15], Xin Sun et al proposed a bidirectional linked list based replica location service to provide a global replica view for supporting the replica management to realize a replica selection strategy and optimal replication strategy on tree-based hierarchical unstructured overlay network.

In [16], Jun Zheng et al proposed a dynamic minimum access cost based replication strategy called MAC replication strategy. It takes into account the access frequency, the status of the network connection and average response time. It calculates an appropriate site to replicate for better shortening the response time of the data source. In [17], Wanlei et al propose the Hybrid Replica Control protocol that attempts to maximize the data availability and communication overhead. In [18], Feras et al propose a Constrained Fast Spread (CFS) method to alleviate the main problems encountered in the current replication techniques and mainly concentrating on the feasibility of replicating the requested replica on each node among the network. In [19] Baskaran et al proposed a GRM in a tree structured P2P network to preserve the replica consistency through the network and reduce the traffic in the network. In [20], Sylvain Dahan et al proposed a Distributed Spanning Tree structure and it is designed to support scalable searches and traversal algorithms. The DST based searches generates less messages to send the query and avoids tree bottleneck. In [21], Sylvain Dahan et al proposed a distributed Spanning tree Structure for large scale environment. This method achieves load balancing and Fault Tolerance in the network. In [22],Xin sun et al proposed the bidirectional linked list based replica location service (BLL-RLS) on tree-based hierarchical unstructured overlay networks, including the deployment of replica location service and the design of the bidirectional linked list based replica catalog. Based on the bidirectional linked list based replica catalog, replica location and selection algorithm is also proposed.

Drawbacks:

- The Existing methods suffers from, huge number of messages sent or a higher volume of computations.
- Space complexity is very high.

- Communication overhead is high.
- When increasing the number of nodes in the network, the Consistency maintenance yields poor efficiency.

3. PROPOSED SYSTEM

In a mobile peer to peer network, each node has to exchange information and services directly with each other without any dedicated intermittent. So it develops bottlenecks in the network due to the huge volume of messages being exchanged. This could be avoided to optimize the number of messages across the network. In this paper, a distributed spanning tree approach is proposed. The proposed system consists of the following steps:

- Formation of mobile peer to peer network.
- Formation of Distributed Spanning Tree.
- Optimization of DST with Ant Colony Optimization.
- DST for Global Replica Management.

3.1. Formation of Mobile Peer to Peer Network

There are various steps for creating the P2P network. In fig 1.a shows the sample mobile peer to peer network and fig 1.b shows the simulation on mobile peer to peer network.

- Define the network options such as communication channel, propagation models, queue types and the network interface.
- Creating the instance of the simulator and set up the trace file.
- Create a topology object that keeps the movement of the mobile nodes within the topological boundary and also set the coordinate values of the boundary.
- Configure the nodes and create the number of mobile nodes in the network. Establish the communication between the nodes.
- Define the initial position of the node when it displayed in the NAM simulator.
- NS-2 uses NAM (network animator) to provide visualization. NAM also allows users to design and debug the network protocols.

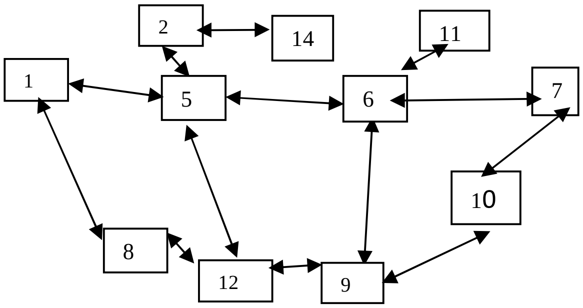

Fig :1a Structure of Mobile P2P Network

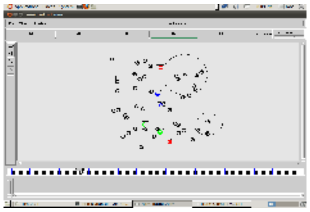

Fig :1b Simulation of Mobile P2P Network

3.2. Formation of Distributed Spanning Tree

The mobile P2P network is converted into the set of spanning trees called the Distributed Spanning Tree (DST) and the corresponding graph based algorithms are developed to optimize the number of messages across the network. The DST is an overlay structure designed to be scalable, which supports the growth of the nodes from fewer nodes to higher volumes [16,17].It allows the instantaneous creation of spanning trees rooted by any node and maintains the load balancing between the nodes [16]. This instantaneous creation of spanning trees improves the overall scalability of the intended network [18]. So, DST structures help to automatically balance and optimize the load among the nodes.

The P2P network is converted into DST and each tree should have its root node, named as the Head Node (HN) and the possible Leaf Nodes (LNs). Every HN will hold the complete details regarding its LNs and vice versa. These HNs are to be generated dynamically and should hold the replica, which is to be accessed by their corresponding LNs and indeed by other HNs also. Fig 2a shows the simulation of distributed spanning tree. The DST algorithm consists of three procedures.

1. Initialize() : This procedure create the set of Head Nodes (HNs) in a peer network based on criteria such as user approval, traffic in a particular region, etc. This procedure creates a list on each HN to hold its LNs details. This procedure assign unique id for every HN and then it calls the procedure probe ().
2. Probe() :This procedure creates probe message and flood this message to all the nodes connected to it. On receiving a probe messages, every node executes receive () procedure.
   - If the probe message is received by an HN, then it will be discarded.
   - If the message is received by the LN, which is not under any HN, then the LN stores the head variable as the HeadID. Then the procedure reply() and the forward() will be called.
   - If the reply message is received by the LN, it will be forwarded to the HN.
   - If the reply message is received by the HN, then it reads the ‘HeadId’ from rmsg. If the ‘Headid’ equals the id of the current node then it concludes that the respective head node is reached.
3. Reply() : The reply() procedure called by the corresponding LN to reply to its HN.

Definition 1: Let Ta be the graph of the peer network with HNs and LNs. Then G_a can be defined as,

$$Ga = \left\{ \begin{array}{c} DST1 = (HN1, LN11, LN12, \ldots..LN1(j1-1) \\ DST2 = (HN2, LN21, LN22, \ldots.LN2(j2-1) \\ \ldots.. \\ DSTi = (HNi, LNi1, LNi2, \ldots..LNi(ji-1) \end{array} \right\} \quad (1)$$

where,

- ‘DST_i’ is the Distributed Spanning Tree and ‘i’ is the total number of DSTs formed in the network.
- HN_i’ is the Head Node (HN) and ‘i’ is the total number of HNs in the peer network equal to the number of DSTs.
- ‘LN’ refers to the Leaf Node(s). In ‘LN_{iz} ’, refers to the corresponding HN_i and $0 < z \leq ij - 1$, where ‘$ij - 1$’ is the total number of LNs in the corresponding DST.

Definition 2: The number of messages required to pass the nodes in the MP2P network with the DST structure can be evaluated as the following equation,

$$n(msgpass)=((P/LN) \times P \times R) + ((P/LN) \times T) \quad (2)$$

where,

- ‘T’ refers to the number of message pass between one HN and another HN.
- ‘R’ refers to the number of message pass between HN and LN.
- ‘P’ refers to the number of peers in the network.
- ‘LN’ refers to the number of LN under HN (assume equal number of LN for all HN).
- ‘(P/LN)’ refers to the number of DSTs in the MP2P network (or equals n(HN)).
- It can be interpreted that $1 \leq M \leq N \leq L$ and $1 \leq P \leq L$.

In Eq. (2), ‘P/LN’ gives the number of DSTs formed in the network which is also equal to n(HN). So the Eq. (2) can be rewritten as,

$$n(msgpass) = (L \times R) + (n(HN) \times T) \quad (3)$$

In other words, the total number of message passes required to form a DST in the MP2P network n(msgpass) is equal to the cumulative sum of the distance of each LN from its HN in terms of edges between them.

ALGORITHM FOR THE FORMATION OF DST:

Initialize()

1. Test the node v .If the test condition is satisfied then the node v is Head node.
2. Create an array to store all the LNs and call the function probe().
3. Create a variable Head as the LN.

Receive()

1. Check whether the message is the probe message and the node is HN means, discard the message
2. If the Head variable of the LN is also null means, delete the message.
3. If the Head variable of the LN equals the probe message of the HeadID means, call the reply() and forward() function.
4. If the node is HN then reply message of the HeadID equals the HeadID means, add the LN to the array else route the message to all the nodes connected to it.

Probe()

Send the probe message to all the LNs connected to the particular HN.

Forward()

Send the forward message to all nodes connected to it.

Reply()

1. Create a reply message to all the nodes.
2. Set the LeafID and HeadID field of the reply message as v.ID and v.Head.
3. Send the reply message to the Head node HN.

Fig 2.Simulation of Distributed Spanning Tree

3.3. An Ant Colony Optimization of DST

The Ant Colony Optimization technique is a probabilistic optimization technique which could find the optimal path in a graph, which is based on behaviours of ant seeking a path between their colonies and source of food. By applying the ACO over the formulated DST, we can obtain the optimal path in terms of reduced number of message passes among the nodes in the network. The Ant colony optimization Algorithm for DST optimization is presented in the fig 3.This algorithm consists of four procedure; optimization (G),propagate (),construction() and daemon action().In Fig 3 shows the simulation of ant colony optimization of DST. The Procedure for optimization (G) consists of two operations:

1. Finding the optimal path between every HNs. Let HN_i is a HN among { HN_1,HN_2,.....HN_n}, where n is the number of HNs in the P2P network. HN_i use probe message ‘p’ to find the optimal path between HN_i and other HNs.
2. The propagate (G,x,p) ,which propagates messages through different paths is called which takes graph G, HN_i as ‘x’ and probe message ‘p’ as parameters. Probe message ‘p’ is flooded through the possible path which increase the number of feasible path discovered between the HN_i and other HN.
3. The construction (G,Ʈ,x,z) which calculates the edge value through the destination HN, is called by HN_i which takes the graph G, start HN ‘x’ start HN ‘x’ specific end HN ‘Z’ and ‘Ʈ’ as the parameter. The ‘Ʈ’ is the measure of cumulative edge value between ‘x’ and ‘z’.’ $Ʈ_i$’ value is used to decide the optimal path between the nodes. The value of ‘Val’ can be given as

   $$\text{Val}= \sum_{i=0}^{p}(Ʈi) \quad (4)$$

   Where

   - ‘val’ is a variable to count the value of Ʈ on each edge from ‘x’ to ‘z’.
   - ‘p’ is the number of edges between HNs ‘x’ and ‘z’ in the MP2P network.
   - ‘Ʈ’ is the cumulative edge value between the node ‘x’ and ‘z’.

4. DaemonAction(Val) is called by the end HN ‘z’ which takes the ‘Val’ as parameter and decides the optimal path between the HNs ‘x’ and ‘z’ based on the value of ‘t’ along the path of each probe ‘p’. Every probe reaches ‘z’ with its ‘val’ then ‘z’ decides the optimal path based on the ‘val’ and the component type of ‘t’.

ALGORITHM FOR ANT COLONY OPTIMIZATION

Optimize()

1. Consider the graph that consists of vertices v and edges e such that G=(v,e)
2. Consider the HN x and z such that create a probe message p on x .
3. Call the function propagate(),construction() and daemonaction() to all HNs in the DST.
4. Consider the LN y for the particular HN x ,create a probe message p on y.
5. Call the function propagate(),construction() and daemonaction() to all the LNs.

Probe()

Forward the probe message p to all nodes.

1. For each non visited vertices in G, count the value of edge from node x to z and also calculate the cumulative edge value between the node x and z.

Construction()

1. Initialize val(z) and trial(z).
2. For each non visited vertices in G, count the value of edge from node x to z and also calculate the cumulative edge value between the node x and z.

DaemonAction()

1. Compare the cumulative edge value between the node x and z.
2. Find the optimal path between the node x and z based on the cumulative edge value ‘Ʈ’.
3. If the value ‘Ʈ ’ can be compared with the positive Qos attribute like bandwidth and transmission speed of the node. If the node has the highest ‘Ʈ ’ value, then it chooses the optimal path between the nodes.

Let ‘τ ’ be a positive QoS attribute like bandwidth, transmission speed, etc., then the path with highest value of ‘val’ is chosen to be optimal. On the other hand, if ‘τ ’ is one of the negative QoS attributes like Hop count, congestion delay, propagation delay, etc., then the path with lowest value of ‘val’ is chosen to be optimal.

Definition 3: Let n(ACOmsgpass) be the number of message passes required for applying ACO in DST MP2P network and it can be estimated as,

$$n(ACOmsgpass) = ((L/P) \times N) + ((P \times M) \times (L/P)) \qquad (5)$$

where,

- ‘N’ refers to the number of message pass between one HN and another HN.
- ‘M’ refers to the number of message pass between HN and LN.
- ‘L’ refers to the number of peers in the network.
- ‘P’ refers to the number of LN under HN (assume equal number of LN for all HN).
- ‘(L/P)’ refers to the number of DSTs in the MP2P network (or equals n(HN)).
- It can be interpreted that $1 \le M \le N \le L$ and $1 \le P \le L$.

Since ‘L/P‘ gives the number of DSTs formed in the network which is also equal to n(HN). So the Eq. (5) can be rewritten as,

$$n(ACOmsgpass) = (n(HN) \times N) + (M \times L). \quad (6)$$

Thus the total number of message passes required to perform ACO technique in DST P2P network is directly proportional to total number of peers and HNs in the network.

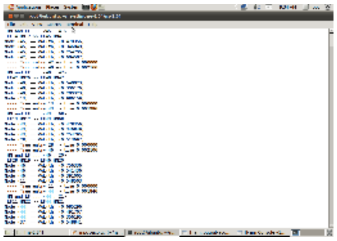

Fig : 3 Simulation of Ant Colony Optimization Method

4. SIMULATION SCENARIO

This section describes the experimental set-up developed for investigating the proposed method with different performance criteria. The NS2 simulator is used for comparing the performance of mobile peer to peer network. A Mobile Peer-to-Peer network is simulated with 50 mobile nodes (N1, N2,.. .N50) moving at constant speed within a bounded region. Unit propagation delay of the wireless medium has been assumed as 10 ms. The experimentation and analysis have been carried out under two different scenarios: MP2P with DST and MP2P with ACO optimized DST networks. The Table 1 specifies the number of parameters used for the simulation of mobile peer to peer network.

In this phase, the performance of the Mobile P2P network is evaluated using the Ant colony optimization method. In fig 4.1a specifies the Number of nodes Vs number of packet request for that node. It shows when the number of nodes increases, the message sent for the node was also decreases by using DST method. But in optimized DST (ACO) method, the number of requested packets from the network was also decreased. In the second graph 4.1b shows the number of node Vs packet delivery ratio. By using ACO method, the packet delivery ratio for the node was increased compared to that of using DST method. In the third graph 4.1c shows number of nodes Vs average delay. By using optimized DST method, the average delay for the node was decreased when the number of node was increased.

Table 1 : Simulation Parameters

Parameters (unit)	**Value(default)**
Number of Mobile nodes	50
Radius of communication range	1~19
Size of the Network	100X100
Propagation delay	12 ms
Average moving speed	0 ~ 25m/s
Maximum transmission range	250 m
Simulation time	900 sec
Nodes Mobility	1,5,10,15,20 m/s
Maximum velocity of the node	1

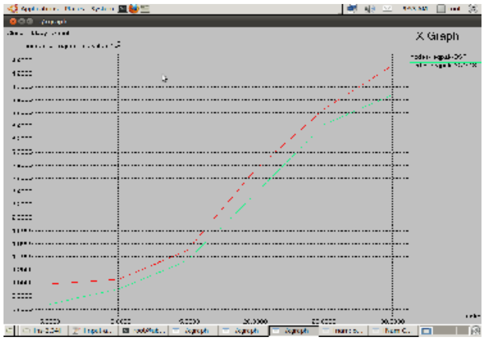

Fig:4.1a No. of Node Vs No.of Request Packets

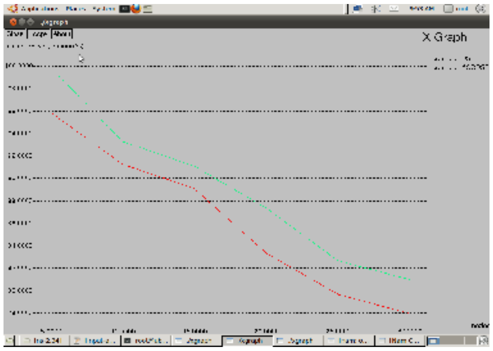

Fig 4.1b : Number of Node Vs Packet Delivery ratio

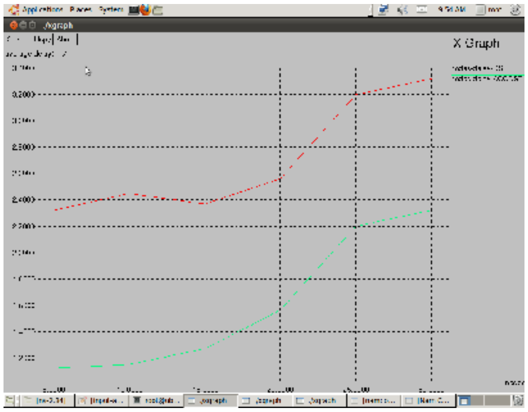

Fig : 4.1c Number of Node Vs Average Delay

5. CONCLUSION

By employing the DST structures in the P2P network, the consistency and replication efficiency can be achieved with the few messages compared to the traditional method. The scalability of the P2P network can be improved with the application of DST structures. The proposed model increases the data availability, reduces the bandwidth conception and number of messages in the network and also improves the fault tolerant capacity of the overall system. Further to enhance the effectiveness of the proposed system, the DST network is optimized with the Ant Colony Optimization method. It gives the optimal solution of the DST method and reduces the message sent and average delay and increases the packet delivery ratio in the network. We have to plan to achieve the cluster based replica allocation for mobile peer to peer networks and also achieve the effective service cache management in the network.

REFERENCES

[1] Feng Jie, Lisong Xu, Byrav Ramamurthy, “Overlay construction in mobile peer-to-peer networks”, in: Mobile Peer-to-Peer Computing for Next Generation Distributed Environments: Advancing Conceptual and Algorithmic Applications. IGI Global, 2009. Web. pp. 51–67.

[2] Frank H.P. Fitzek, Hassan Charaf, “Mobile peer-to-peer networks: An Introduction to the Tutorial Guide”, John Wiley & Sons, Ltd., ISBN: 978-0-470- 69992-8, 2009, pp. 4–18.

[3] Mobile peer-to-peer systems: overview, issues and potential usages. Webpage: http://medialab-prado.es/mmedia/2328

[4] A.A. Helal, A.A. Heddaya, B.B. Bhargava, Replication Techniques in Distributed Systems, Kluwer Academic, ISBN: 0-7923-9800-9, 1996.

[5] B. Ciciani, D.M. Dias, P.S. Yu, Analysis of replication in distributed database systems, IEEE Trans. Knowl Data Eng. 2 (2) (1990) 247–261.

[6] Takahiro Hara, Sanjay Kumar Madria, Consistency management strategies for data replication in mobile ad hoc networks, IEEE Trans. Mob. Comput. 8 (7) (2009) 950–967.

[7] S.Y. Hwang, K.K.S. Lee, Y.H. Chin, Data replication in a distributed system: a performance study, in: Proc. Conf. Database and Expert Systems Applications, 1996, pp. 708–717.

[8] Ren Xun-yi, Wang Ru-chuan, Kong Qiang, Efficient model for replica consistency maintenance in data grids, in: International Symposium on Computer Science and its Applications, IEEE, 2008.

[9] Chao-Tung Yang, Chun-Pin Fu, Ching-Hsien Hsu, File replication, maintenance, and consistency management services in data grids, J. Supercomput. 53 (3)(2009) 411–439.

[10] R.M. Rahman, K. Barker, R. Alhajj, Replica placement design with static optimality and dynamic maintainability, in: Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid, CCGRID’06, pp. 434–437.

[11] S.M. Park, J.H. Kim, Y.B. Ko, S. YoonW-, Dynamic data grid replication strategy based on internet hierarchy, in: The IInd Intl Workshop on Grid and Cooperative Computing, GCC2003, pp. 838–846.

[12] Haiying Shen “IRM: Integrated File Replication and Consistency Maintenance in P2P Systems” IEEE transactions on Parallel and Distributed systems,Vol.21 pp.100-113. Top of FoBottom of Form

[13] J. Luo, J.P. Hubaux, P. Eugster, PAN: providing reliable storage in mobile ad hoc networks with probabilistic quorum systems, in: Proc. ACM MobiHoc, 2003, pp. 1–12.

[14] A. Datta, M. Hauswirth, K. Aberer, Updates in highly unreliable, replicated peer-to-peer systems, in:Proceedings OF IEEE ICDCS’03, Providence, RI, USA, May 2003.

[15] Xin Sun, Kan Li,Yushu Liu,” An Efficient Replica Location Method in Hierarchical P2P Networks”,IEEE ICIS Intl Conf. Pp.769-774.

[16] Xin Sun, Jun Zheng, Qiongxin Liu, Yushu Liu, “Dynamic data replication based on access cost in distributed systems”, in: 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology, IEEE 2009.

[17] Wanlei Zhou,Holmes,” The design and simulation of a hybrid replication control protocol” proc in.IEEE Intl sym. on PAAN, pp.210-215.1999.

[18] Feras Hanandeh and Mutaz,” CFS: A New Dynamic Replication strategy for Data grids”,Intl,Arab JIT Vol.9 2012.

[19] Baskaran .R and Paul.P.V,”Algorithm and direction for analysis of global Replica Management in P2P network”,IEEE Intl.Conf. in ICRTIT 2012.

[20] Sylvain Dahan, Laurent Philippe, Jean-Marc Nicod, “The distributed spanning tree structure”, IEEE Trans. Parallel Distrib. Syst. (2009) 1738–1751.

[21] Sylvain Dahan, Distributed spanning tree algorithms for large scale traversals,in: 11th International Conference on Parallel and Distributed Systems,ICPADS’05, IEEE, 2005.

[22] Xin Sun and Kan Li “ An Efficient Replica Location Method in Hierarchical P2P Networks” IEEE /ACIS Intl Conf.pp.769-774.2009.