International Journal of Computer Science and Security (IJCSS)

ISSN : 1985-1553

VOLUME 2, ISSUE 5

PUBLICATION FREQUENCY: 6 ISSUES PER YEAR

Editor in Chief Dr. Haralambos Mouratidis

International Journal of Computer Science and Security (IJCSS)

Book: 2008 Volume 2, Issue 5
Publishing Date: 30-10-2008
Proceedings
ISSN (Online): 1985-1553

IJCSS Journal is a part of CSC Publishers
http://www.cscjournals.org

Published in Malaysia

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers

Table of Contents

Volume 2, Issue 5, October 2008.

Pages

Application of new distance matrix to phylogenetic tree construction

P.V.Lakshmi pvl_7097@rediffmail.com
Computer Science & Engg Dept
GITAM Institute of Technology
GITAM University
Andhra Pradesh
India

Allam Appa Rao apparaoallam@gmail.com
Jawaharlal Nehru Technological University
Kakinada
Andhra Pradesh
India

Abstract

Phylogenies are the main tool for representing the relationship among biological entities. Phylogenetic reconstruction methods attempt to find the evolutionary history of given set of species. This history is usually described by an edge-weighted tree, where edges correspond to different branches of evolution, and the weight of an edge corresponds to the amount of evolutionary change on that particular branch. Phylogenetic tree is constructed based on multiple sequence alignment, but sometimes alignment fails if the data set is large and complex. In this paper a new distance matrix is proposed to reconstruct phylogenetic tree. The pair-wise scores of input sequences were transformed to distance matrix by Feng Doolittle formula before solved by neighbor-joining algorithm. Two data sets were tested with the algorithm: BChE sequences of mammals, BChE sequences of bacteria. We compared the performance and tree of our result with ClustalX and found to be similar.

Keywords: Phylogeny, Bioinformatics, Distance matrix, Phylogenetic tree, neighbor-joining algorithm, Clustal X.

1. INTRODUCTION

Phylogenetic reconstruction methods attempt to find the evolutionary history of a given set of species. Phylogenies are reconstructed using data of all kinds, from molecular data, metabolic data, morphological data to geographical and geological data [1]**.** Phylogenetic analysis elucidate functional relationship within living cells [2-4]. With more and more DNA and protein sequences have been obtained, [5-7] the problem of inferring the evolutionary history and constructing the phylogenetic tree has become one of the major problems in computational biology. There are three major methods for performing a phylogenetic analysis, distance method, maximum parsimony, and maximum likelihood methods. Distance-matrix methods such as neighbor-joining or UPGMA calculate genetic distance from multiple sequence alignments. Maximum parsimony method implies an implicit model of evolution, predicts the evolutionary tree that minimizes the

number of steps required to generate the observed variation in the sequences from common ancestral sequence. Maximum likelihood method start with simple model, in this case a model of rates of evolutionary change in nucleic acid or protein sequences and tree models that represent a pattern of evolutionary change, and then adjust the model until there is best fit of observed data. All this method requires multiple sequence alignment. Multiple sequence alignment is extension of pairwise sequence alignment. Needleman - Wunsch[8] and Smith-waterman [9] are classical dynamic programming algorithm for pair-wise sequence alignment with time and space cost of algorithm as O(mn), where m, n are lengths of two sequences to be aligned. Proposed algorithm takes O(m+n) time to generate score matrix .

2. Phylogenetic tree construction definition and previous work

A phylogenetic tree, is a model of the evolutionary history for a set of species. The neighbor-joining method by Saitou and Nei is a widely used method for constructing phylogenetic trees. It is a distance based method for constructing phylogenetic trees. It was introduced by Saitou and Nei [6], and the running time was later improved by Studier and Keppler [10].The neighbor-joining method is a greedy algorithm which attempts to minimize the sum of all branch-lengths on the constructed phylogenetic tree. It starts out with a star-formed tree where each leaf corresponds to a species, and iteratively picks two nodes adjacent to the root and joins them by inserting a new node between the root and the two selected nodes. When joining nodes, the method selects the pair of nodes i, j that minimizes the branch-length sum of the resulting new tree. Select the pair of nodes i, j that minimizes

$$Q_{ij} = (r - 2)\, d_{ij} - (R_i + R_j) \text{--- (1)}$$

where dij is the distance between nodes i and j.(assumed symmetric, i.e., $d_{ij} = d_{ji}$), R_k is the row sum over row k of the distance matrix. $R_k = \Sigma_i\, d_{ik}$ (where i ranges over all nodes adjacent to the root node), and r is the remaining number of nodes adjacent to the root. When nodes i and j are joined, they are replaced with a new node 'U' with distance to a remaining node k given by

$$d_{Uk} = (d_{ik} + d_{jk} - d_{ij})/2 \text{------- (2).}$$

Repeat this until single node. We implemented neighbor-joining method taking distance matrix obtained by multiple progressive alignment technique.

3. New distance matrix algorithm

Using dynamic programming algorithm to find the score of two sequences take O(mn) time. To reduce computational time, a new method to calculate score of two sequences was proposed. Let *X* and *Y* be two sequences with lengths of *n* and *m,* respectively. The score of their alignment, namely the length of longest common subsequences, can be calculated using dynamic programming algorithm in *O*(*mn*) time. To reduce the computation time, a score estimating algorithm[11] to approximately estimate the score of a two-sequence alignment was considered. The algorithm estimates the score of the alignment of two sequences in *O*(*m*+*n*) time. The proposed algorithm consists of 4 steps each of which scans the two sequences from a different direction. Denote *X* and *Y* as the upper and lower sequence, respectively. The four steps are denoted as Left-Upper, Right-Upper, Left-Lower and Right-Lower. The step of Left-Upper starts from the first character in *X*, say *X*[0] and searches for the first matching character in *Y* from left to right. If there is no character in *Y* matching *X*[0], it restarts the scan to search for the character matching *X*[1] in *Y* . After such character, say *Y*[*j*], being found, the algorithm searches for the first character matching with *Y*[*j*] in the rest part of *X* from left to right. If there is no character in *X* matching *Y*[*j*], it restarts the scan to search for the character matching *Y*[*j*+1] in *Y* . After such character, say *X*[*i*], being found, the algorithm searches for the first character matching with *X*[*i*] in the rest of *Y* from left to right. We alternately repeat such scans until reaching the end of the sequences. As a result, the number of the matching characters can be obtained in the scan. *count* is the number of the matching characters in *X* and *Y*.

Algorithm:
Begin

```
Lt-up(X,Y,n,m,count1);
Lt-low(X,Y,n,m,count2);
Rt-up(X,Y,n,m,count3);
Rt-low(X,Y,n,m,count4);
Return (max(count1,count2,count3,count4))
End.
```

4. Method

For given set of sequences score matrix is obtained from proposed method. Distance matrix is generated using Feng –Doolittle formula.

$$d(x^i, x^j) = -\log \frac{S(x^i, x^j) - S_{rand}}{S_{max} - S_{rand}}$$

where S_{rand} is the mean score of two random sequences. It is taken as 4 defined constant and S_{max} is the maximum attainable score for two sequences. Phylogenetic tree was constructed using neighbor-joining algorithm.

5.Results

We tested our method on two datasets of BChE sequences, mammals and bacteria. The trees are generated using the Neighbor Joining (NJ)method [6]. And all the experiments in this paper were performed on a PC with Pentium IV CPU (ZGHZ), 512KB Cache, and 256MB RAM. We chose our group of sequences from first and second data set obtained from http://www.ncbi.nlm.nih.gov/.Mammals:Human(homosapiens,NP_000046),mouse(musmusculus, NP_033868),horse(Equuscaballus,NP_00075319),Sumatranorangutan(Pongoabeli,NP_001127509), cattle (Bos taurus, NP_001070374), domestic cat(Felis catus, NP_001009364),Norway rat(Rattus norvegicus,NP_075231),chimpanzee(Pantroglodytes,XP_516857),BengalTiger (Pantheratigristigris,AAC06262),chicken(Gallusgallus,CAC37792).Bacteria:Thiocapsaroseopersicina, Chlorobium tepidum, Rhodobacter sphaeroides, Rhodobacter capsulatus, Synechocystis sp pcc6803, Rosebacter denitrificans, Heliobacillus mobilis, Bradyrhizobium japonicum, Rhodopseudomonas palustris, Rhizobium etli cFN42, Lawsonia intracellularis, Rubrivivax gelatinosus, Candidatus Kuenenia stuttgartiensis, Bradyrhizobium sp, Chloroflexus aggregans. We applied the new distance measure to the above first data sets. Fig.1 shows the tree generated by proposed method. The tree is very close and Consistent with an earlier report published in *J Biol Chem.* 1991. The data set is applied to ClustalX[12] yielded a distance matrix, which was then analyzed by the NJ program, the result is shown in Fig.2. The results presented in Fig. 1 and Fig.2 are almost the same, but there are still some differences, for example cattle grouped with mouse and rat.

6. Conclusion

In this paper, we proposed a new sequence distance measure and used it to generate distance matrix for constructing phylogenetic tree. Unlike most existing phylogeny construction methods, the proposed method does not require multiple alignments and is fully automatic. We tested our method on two datasets and applied it to analyse the evolutionary relationship among BChE sequences. It is to be noted that we use no approximations and assumptions in calculating the distances between sequences, and our distance measure does not make use of any evolutionary model. It's one of the alignment free methods for phylogenetic tree construction of given sequences and is fully automatic.

Fig1 Tree constructed by ClustalX tool

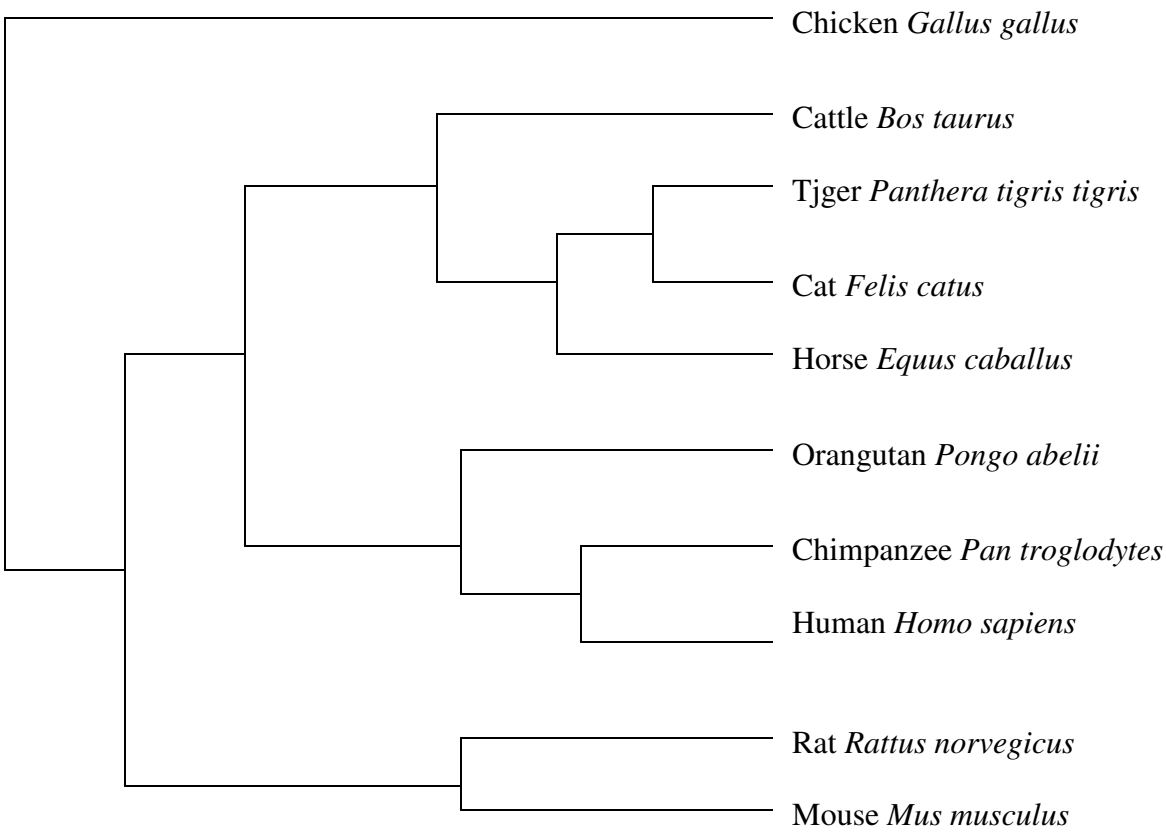
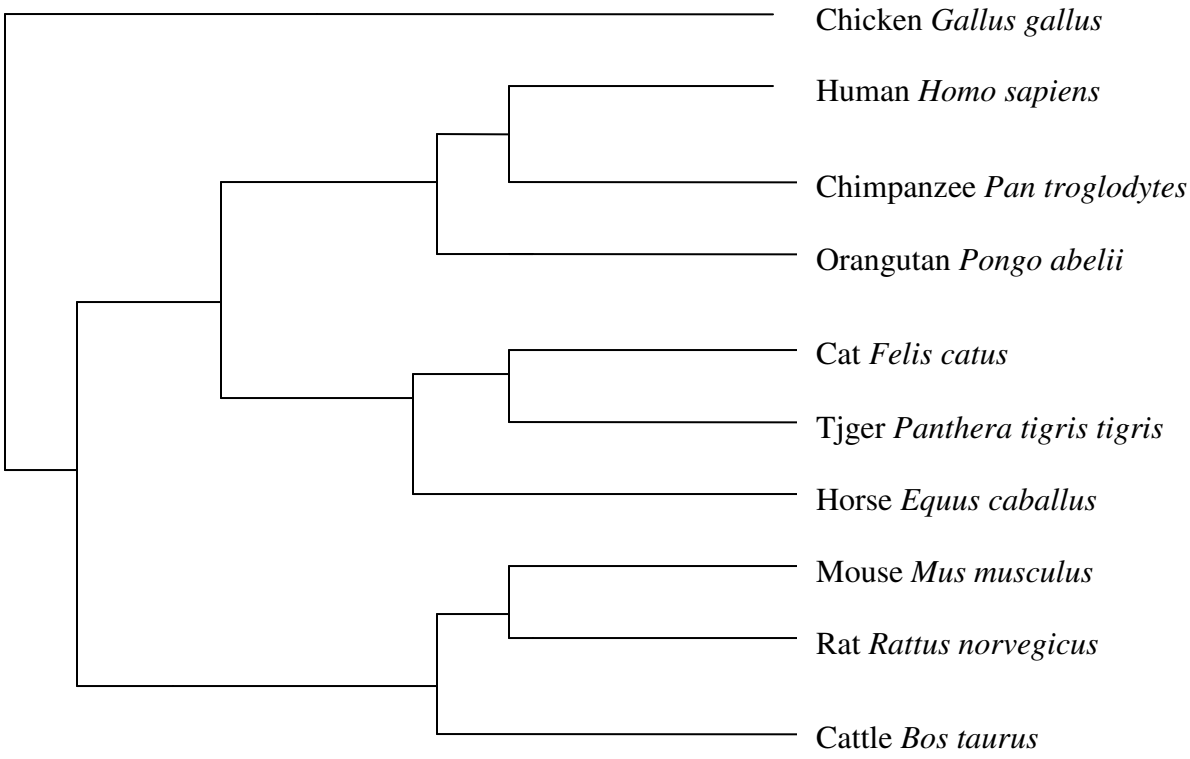


Fig2 Tree constructed by proposed method



7. References

1. Swofford, D.L., Olsen, G.J., Waddell, P.J. and Hillis, D.M. (1996).Phylogenetic inference. In *Molecular Systematics* (ed. D.M. Hillis, B.K.Mable, and C. Moritz), pp. 407.514. Sinauer Assoc.,Sunderland, MA.
2. M.Y. Galperin and E.V. Koonin. Comparative genome analysis. Methods Biochem. Anal.,43:359–392, 2001.
3 . X. Gu. Maximum-likelihood approach for gene family evolution under functional divergence.Mol. Biol. Evol., 18(4):453–464, 2001.
4. H. Zhu and J.F. Klemic et al. Analysis of yeast protein kinases using protein chips. Nature Genetics,26(3):283–289, 2000.
5. T. Hodge, M. J. T. V. Cope, "A myosin family tree,"*Journal of Cell Science*, Vol. 113, 2000, pp.3353-3354.
6. N. Saitou and M. Nei, "The neighbor-joining method: A new method for reconstructing phylogenetic trees," *Molecular Biology and Evolution*, Vol. 4, 1987, pp. 406-425.
7 T. H. Reijmers et al., "Using genetic algorithms for the construction of phylogenetic trees: application to G-protein coupled receptor sequences," *Biosystems*, Vol. 49, 1999, pp31- 43.
8. S.B.Needleman, and C.D.Wunsch, "A General method applicable to the search for similarities in amino acid sequence of two proteins", *Journal of Molecular Biology* 48pp.443-453,1970.
9 T. Smith and M. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, pp. 195–197, 1981. Mol.Biol.Evol.,4,406-425(1987) .
10 Studier JA, Keppler KJ: A Note on the Neighbor-Joining Method of Saitou and Nei. *Mol Biol Evol* 1988, 5(6):729-731.
11 Partitioned optimization algorithms for multiple sequence alignment. Yixin Chen[1] Yi Pan[2] Ling Chen[3] Juan Chen[3]
12 ClustalX program. Thompson,J.D., Gibson,T.J., Plewniak,F., Jeanmougin,F. and Higgins,D.G. (1997) The ClustalX windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools. Nucleic Acids Research, 25:4876-4882

DIRECT TRUST ESTIMATED ON DEMAND PROTOCOL FOR SECURED ROUTING IN MOBILE ADHOC NETWORKS

N.Bhalaji bhalaji.80@gmail.com
Assistant professor/department of information technology
Hindustan University
Chennai-603103, Tamilnadu, India

Druhin mukherjee, Nabamalika banerjee
School of CSE
SRM University
Chennai-603203, Tamilnadu, India

A.Shanmugam
Principal
Bannari Amman Institute of Technology,
Sathyamangalam-638401, Tamilnadu, India

Abstract

Adhoc network is a collection of wireless nodes communicating among themselves over multihop paths, without the help of any infrastructure such as base stations or access points. Although many previous techniques have been proposed for the secure routing, in this paper we propose a more reasonable and unambiguous equation for trust evaluation. Our scheme is distributed and effective without reliance on any central authority. In this paper we focus on improving the security of most commonly used Dynamic Source Routing Protocol (DSR).We improve the routing security of the existing DSR protocol by enhancing the concept of trust value, the selection of a secure route will be based on these trust values. Ns-2 simulations are performed to evaluate the impact of applying trust value based route selection to the DSR protocol.

Keywords: Adhoc, DSR, Security, Trust

1. INTRODUCTION

Adhoc networks are a collection of mobile hosts (or they can also be called as nodes), which form a temporary network. There is no fixed infrastructure in an adhoc Network and each host have a wire less interface and communicate with each other over radio or infrared. Because of node mobility the network topology changed frequently. All nodes of these wireless networks behave as routers and take part in discovery and maintenance of routes to other nodes in the network. Ad hoc networks are useful in the emergency operations and in which persons need to share information and data quickly. The security for routing protocols should be an important component in MANET. The network operations can be easily jeopardized if countermeasures are not embedded into basic routing protocol functions of MANET at the early stages of their design. Wireless mobile ad hoc network routing protocols have to be thoroughly tested and analyzed in term of their operations. Simulation experiments are the main tool for testing MANET routing protocols. Simulation experiments need to be conducted before any real implementation. The wireless and mobile nature of MANET brings new security challenges in the network design. Mobile nodes in MANET communicate with each other via open and shared broadcast wireless channels, so they are more vulnerable to security attacks. In addition, their infrastructure-less nature means that centralized security control is hard

to implement, so the network needs to rely on individual security solutions from each mobile node participating in the network. Our goal in this paper is to present the trust- based route selection to the existing implementation of the DSR routing protocol [1] of MANET to improve the security aspects of the routing protocol. We also perform detailed simulation study for the proposed secure routing protocol for MANET.
The main contributions of this paper are:

- Improving the security of the existing DSR protocol by enhancing a trust-based route [2]selection
- Comparing the implemented routing protocol with the existing DSR protocol, using simulations.

The remainder of this paper is organized as follows. Related work is discussed in Section 2, followed by a description of the proposed Trust based DSR protocol in Section 3. The simulation setup and corresponding results are outlined in section 4. Future work is outlined in Section 5 and conclusions are drawn in Section 6.

1.1. On demand routing protocol

The protocols for the adhoc network are classified based on different characteristics [2] such as

- Routing information update mechanism
- Use of temporal information for routing
- Routing topology and
- Utilization of specific resources.

The on demand routing protocol belongs to the first category of protocols which updates the information and are of reactive in nature. They obtain the necessary path when it is required, by using a connection establishment process. The most commonly used protocols under this category are Dynamic Source Routing [1] and AODV [3]. For our simulation study we consider DSR as a reference protocol.

2. RELATED WORK

2.1. DSR Protocol

Dynamic Source Routing is a protocol developed for routing in mobile ad-hoc networks and was proposed for MANET by Broch, Johnson, and Maltz [1]. In a nutshell, it works as follows: Nodes send out a ROUTE REQUEST message, all nodes that receive this message put themselves into the source route and forward it to their neighbors, unless they have received the same request before. If a receiving node is the destination, or has a route to the destination, it does not forward the request, but sends a REPLY message containing the full source route. It may send that reply along the source route in reverse order or issue a ROUTE REQUEST including the route to get back to the source, if the former is not possible due to asymmetric links. ROUTE REPLY messages can be triggered by ROUTE REQUEST messages or are gratuitous. After receiving one or several routes, the source selects the best (by default the shortest), stores it, and sends messages along that path. The better the route metrics (number of hops, delay, bandwidth, or other criteria) and the sooner the REPLY arrives at the source, the higher the preference given to the route and the longer it will stay in the cache. When a ROUTE REPLY arrives very quickly after a ROUTE REQUEST has been sent out this is an indication of a short path, since the nodes are required to wait for a time corresponding to the length of the route they can advertise, before sending it. This is done in order to avoid a storm of replies. In case of a link failure, the node that cannot forward the packet to the next node sends an error message towards the source. Routes that contain a failed link can be `salvaged' by taking an alternate partial route that does not contain the bad link.
The author in [4] developed and applied trust based routing to the DSR protocol. The idea behind this approach is to store information of the trust that one node has to the other nodes. These trust values are adjusted based on the experiences of the nodes, such as packet drops or acknowledgements receipts. The routes are evaluated based on some heuristic that uses the trust Values of the nodes as criteria. The performance study in [4] showed that this implementation had a higher throughput than standard DSR when the number of malicious nodes is slow. However it showed that DSR protocol outperformed the trust based routing in situations with a high number of malicious nodes. The malicious nodes should have very low trust values. In this paper, we improve the DSR protocol by enhancing the trust-based routing solution when exist a high number of malicious nodes in the network.
The authors in [5] evaluate the performance of some trust-based reactive routing protocols in a mobile network with varying number of malicious nodes. By doing many simulations, they demonstrate that the performance of theses protocols varies significantly even under similar attack, traffic, and mobility conditions. However, each trust-based routing protocol has its own peculiar advantage making it suitable for application in a particular extemporized environment.

Current ad hoc routing protocols are basically exposed to two different types of attack: active attacks [6] and passive attack. The active attack occurs when the malicious node bears some energy costs in order to perform the threat, whereas passive attacks are mainly due to lack of cooperation, with the purpose of saving energy selfishly. Mobile nodes that perform active Attacks with the aim of damaging other nodes by causing network outages are considered to be malicious nodes, where mobile nodes that perform passive attacks with the aim of saving battery life [7] for their own communications are considered to be selfish nodes. Malicious nodes can disrupt the functions of a routing protocol by modifying its information or by sending false routing information through the entire network. On the other hand, selfish nodes can severely degrade network performance and eventually partition the network by simply not participating to the network operation.

3. NEW TRUST BASED ROUTING SCHEME

This section presents the improvement of the trust based Route selection to be applied to the DSR protocol in order to enhance the security of the routing protocol. The purpose of applying the Trust based route selection to the DSR protocol is to fortify the existing implementation by selecting the best and securest route in the network. In difference to the process of route selection in the DSR protocol which involves the selecting of the shortest route to the destination node, in our proposed Protocol we choose the most reliable and secure route to the destination based on the trust values of all nodes that found in the administrator of the trust unit. A separate acknowledgement module is there to monitor the received acknowledgments and adjust the trust values for the nodes on the route. For each node in the network, a trust value will be stored that represent the value of the trustiness to each of its neighbor nodes. This trust value will be adjusted based on the experiences that the node has with its neighbor nodes. When a node receives data packets or acknowledgements from its neighbor node, the trust value for this neighbor node will be upgraded. Neighbor node that is encountered for the first time will have an initial trust value assigned based on trust formation strategy. If a route contains known nodes, the trust values of these neighbor nodes are used to assign the initial trust value. If a requested acknowledgement was not received, the trust value for this neighbor node should be decreased.

3.1. Components of the proposed protocol

The proposed protocol consists of the following components.

1. Trust Unit
1.1. Initialiser
1.2. Upgrader
1.3. Administrator
2. Monitor
3. Router

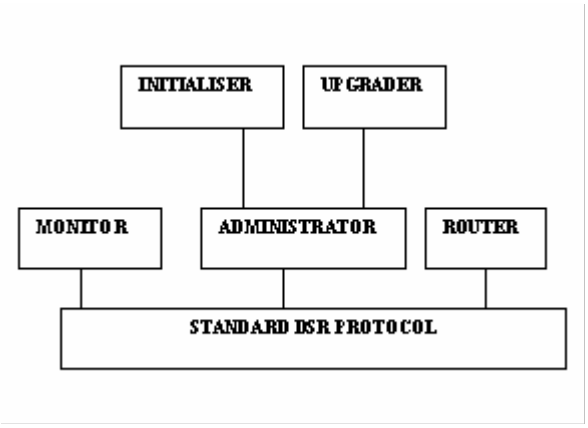


FIGURE 1: Components of Relationship Based DSR

Trust Unit

***Initialiser Module*:** This module is used to assign a trust value for unknown new mobile nodes in the network. It would be best to assign a low trust value in an environment with many malicious nodes. If a route contains known nodes, the trust value of these nodes is used to base the assignment of the initial trust value for the new node.

Upgrader Module: The upgrader module of trust unit is used to implement the Functions for upgrading trust. The updating of the trust values will depend on a given node experience in a given situation. We use the

following equation to upgrade the trust value for each node encountered in the route the function for upgrading trust depends on two parameters, previous trust values and the nature of Experience. It is calculated as below

T = tanh [(Δ+W)*Te]

Where
T: The upgraded trust value
Te: The existing trust value
W: The weight of the experience. For acknowledgement related operations it is assumed to be 1 and for data forward and receiving it takes 0.5.
Δ: assumes +1 for positive experience and 0 for negative experience.
The positive and negative experience is calculated based on the acknowledgment. If the acknowledgement is received within the time frame then it's counted as positive experience else if its not received with in the stipulated time it is counted as a negative experience.
Administrator: The Administrator module of the trust unit stores trust information about all known nodes during run time, and it offers methods to query for information about stored trust values. So it is used as the interface between the existing DSR protocol on one hand and the Initialiser and Upgrader modules on the other hand.
Monitor: The purpose of the monitor module is to adjust the trust values from the received acknowledgements. Since the trust values are used on routing selecting decisions, it is important that a missing acknowledgement is detected fast. When an acknowledgement is received, the trust upgrader module upgrades the trust values for nodes on the stored route. If a requested acknowledgement is not received, the packet is considered dropped, so the trust values should be adjusted in a negative way.
Router: The router module is responsible to evaluate routes based on trust values of nodes. In this paper we are going to discuss about different routing strategies which we are going to apply over the proposed protocol and test its performance in presence of malicious nodes.
Route selection strategy 1:
The first route selection strategy will return the average trust value of all nodes on the route. Based on this value the route is rated and the route with highest rating is preferred.
Route selection strategy 2:
The second one is extension to the first. In order to favor shorter routes average of the trust values is divided by the number of nodes. Thus the route with high value are given high ratings and subsequently selected for the routing.

4. SIMULATIONS AND RESULTS

In our simulations we use performance metric to compare the trust based DSR protocol fortified with the above route selection strategies under the presence of the malicious nodes and the standard DSR. The throughput is considered for our experiment which is defined as a important metric for the determination of the routing protocol performance [8].
Throughput: This gives the fraction of the channel capacity used for data transmission
Throughput = Total amount of Data received correctly / Total time
For the performance analysis of the protocol extensions, a regular well-behaved DSR network is used as a reference. We then introduce compromised stranger nodes into the network which doesn't forward the packets. The simulation is being implemented In Network Simulator 2 [9], a simulator for mobile adhoc networks.

PARAMETER	VALUE
Application traffic	CBR
Radio range	250 m
Packet size	512 bytes
Transmission rate	4 packets/s
Pause time for nodes	60 s
Maximum speed	1 m/s
Simulation time	600 s
Number of nodes	25
Area	1000 m *

	1000 m
Available bandwidth	1 Mb/s

The speed of 1 m/s corresponds to slow moving. For a simulation that last 600 seconds, approximately 30000 CBR packets are sent. This number is considered high enough to eliminate any deviations influence on the results. With 1 Mb/s bandwidth, a packet size of 512 bytes and a transmission rate of 4 packets/s, congestion of the network is not likely to occur.

The following graphs illustrate the performance of the different routing strategies and their throughput values. The route selection performance of standard DSR protocol is shown below and it assumes all the nodes are functioning proper and does not bother about the malicious behaviour of the nodes. No trust enhancements are used in the route selection of the DSR protocol.

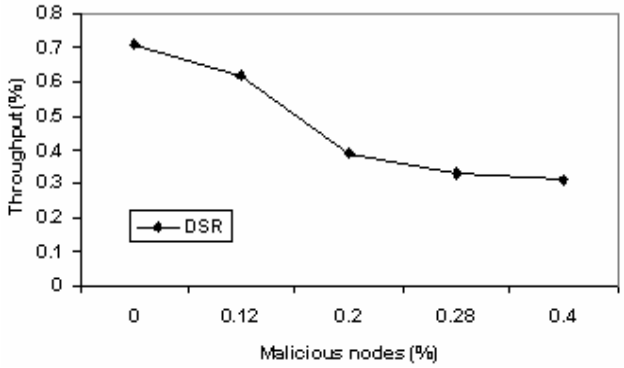


FIGURE 2: Throughput of Standard DSR

The following graph is the result of employing the route selection strategy 1 which is based on the average trust value of nodes. The highly rated routing path is selected and thus the following graph shows some improvement over the standard DSR.

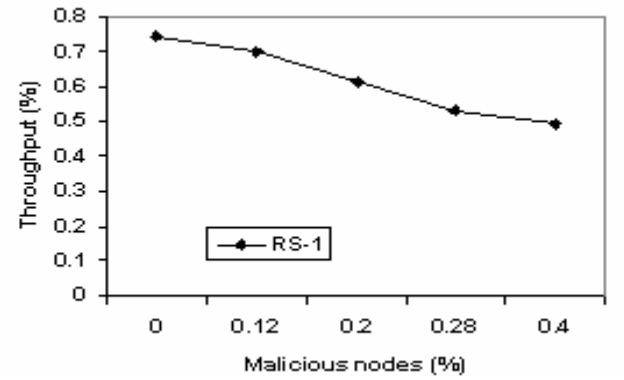


FIGURE 3: Throughput of Route Selection strategy-1

Then we conduct another simulation to analyse the performance of the routing protocol when provided with the scheme of Routing strategy -2 which we have discussed in the previous section. This method determines the rating of the route based on the values obtained from dividing the average of the trust values by the number of nodes. This figure also gives better performance than the standard DSR.

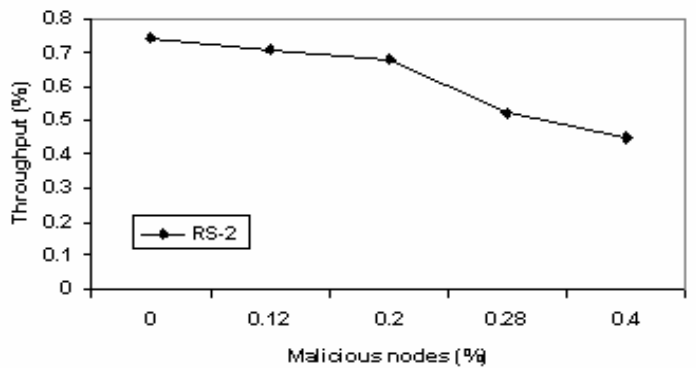


FIGURE 4: Throughput of Route selection strategy-2

The throughput obtained by using all above routing techniques are compared and the results reveal that the routing performed with fortified trust values yield better results than the standard DSR routing.

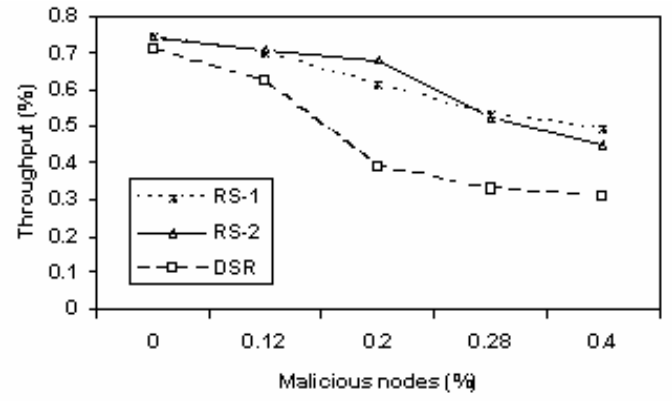


FIGURE 5: Comparison of throughput

5. FUTURE WORK

In this paper we proposed a two routing schemes based on the trust values of the nodes. In future instead of using only the trust values, the experience factor may be considered for calculating the rating of the route. the proposed scheme may also be tested under different attack scenarios.

6. CONCLUSION

We presented improvement in mobile adhoc routing over on demand type of protocol namely Dynamic Source Routing protocol and analysed the performance of the proposed scheme and compared it with the existing DSR protocol. Ns-2 simulator [9] was used for the analysis of the performance. The results show that the proposed trust based routing performs better than the existing standard DSR.

7. REFERENCES

1 D. Johnson, D. Maltz, Y. Hu, and J. Jetcheva. "*The dynamic source routing protocol for mobile ad hoc networks. Internet Draft, Internet Engineering Task Force*" Mar. 2001.http://www.ietf.org/internetdrafts/ draft-ietf.

2 C. Siva Ram Murthy and B. S. Manoj, "*Ad Hoc Wireless Networks: Architectures and Protocols*" Prentice Hall, 2004.

3 C.E.Perkins and E.M.Royer, "*Adhoc On-Demand Distance Vector Routing*" proceedings of IEEE workshop on mobile computing systems and Applications 1999, pp. 90-100, February 1999.

4 John Keane, "*Trust-based Dynamic Source Routing in Mobile Ad Hoc Networks*", MS thesis, Department of Computer Science, Trinity College Dublin, September 2002.

5 A. Pirzada, C. McDonald and A. Datta,"*Performance Comparison of Trust-based Reactive Routing Protocols*", IEEE Transactions on Mobile Computing, Vol 5(6), pages 695-710, 2006.

6S. Murphy, "*Routing Protocol Threat Analysis," Internet Draft, draft-murphy-threat-00.txt*, October 2002. https://forum.eviloctal.com/redirect.php?tid=1992&goto=lastpost

7 Djamel Djenouri, Nadjib Badache "*New power-aware routing protocol for mobile Adhoc network*" International journal Adhoc and ubiquitous computing, volume 1.No.3, 2006. pp 126-136. DOI: 10.1504/IJAHUC.2006.009882.

8 J. Broch, D. Johnson, D. Maltz, Y. Hu, J.Jetcheva, "*A Performance Comparison of Multihop Wireless Ad Hoc Networking Protocols*", Proceedings of 4th ACM/IEEE International Conference on Mobile Computing and Networking, 1998.

9 Kevin Fall, Kannan Varadhan: *The ns manual*, http://www.isi.edu/nsnam/ns/ doc/index.html

Dynamic Load Balancing Architecture for Distributed VoD using Agent Technology

H S Guruprasad hs_gurup@yahoo.com
Research Scholar, Dr MGR University
Asst Prof& HOD / Dept of ISE
BMSCE, Bangalore, India

Dr. H D Maheshappa hdmappa@gmail.com
Prof & HOD / Dept of E & C
RevaITM
Bangalore, India

Abstract

This paper proposes a load balancing algorithm for a distributed VoD architecture using agents. A mobile agent is used to frequently update the popularity of the videos based on which channel allocation is done effectively. The proposed approach groups a set of local proxy servers into a Local Proxy Server Group [LPG] for load balancing among the proxy servers, to reduce the load on the central multimedia server, to reduce storage redundancy among the proxy servers and to maximize the channel utilization. The simulation results prove the load balancing among the local proxy servers, reduction of load on central multimedia server, maximum channel utilization and more channel allocation for popular videos.

Keywords: Load Balancing, Mobile Agent, Channel Allocation, Waiting Time, Popular Videos, Proxy Server

1. INTRODUCTION

Agents are autonomous programs which can understand an environment, take actions depending upon the current status of the environment using its knowledge base and also learn so as to act in the future. Autonomy, reactive, proactive and temporally continuous are mandatory properties of an agent. The other important properties are commutative, mobile, learning and dependable. These properties make an agent different from other programs. The agents can move around in a heterogeneous network to accomplish their assigned tasks. The mobile code should be independent of the platform so that it can execute at any remote host in a heterogeneous network [1, 3, 7].

A video-on-demand system can be designed using any of the 3 major network configurations – centralized, networked and distributed. In a centralized system configuration, all the clients are connected to one central server which stores all the videos. All the client requests are satisfied by this central server. In a network system configuration, many video servers exist within the

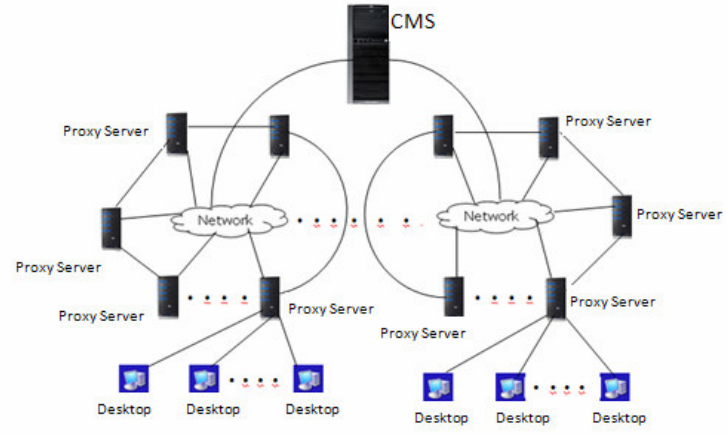
network. Each video server is connected to a small set of clients and this video server manages a subset of the videos. In a distributed system configuration, there is a central server which stores all the videos and smaller servers are located near the network edges. When a client requests a particular video, the video server responsible for the requests ensures continuous playback for the video [2].

Proxy servers are widely used in multimedia networks to reduce the load on the central server and to serve the client requests faster. In [4], Tay and Pang have proposed an algorithm called GWQ [Global Waiting Queue] which shares the load in a distributed VoD system and hence reduces the waiting time for the client requests. This load sharing algorithm balances the load between heavily loaded proxy servers and lightly loaded proxy servers in a distributed VoD. They assumed that videos are replicated in all the servers and videos are evenly required, which requires very large storage capacity in the individual servers. In [6], Sonia Gonzalez, Navarro, Zapata proposed a more realistic algorithm for load sharing in a distributed VoD system. Their algorithm maintains small waiting times using less storage capacity servers by allowing partial replication of videos. The percentage of replication is determined by the popularity of the videos.

In this paper, we propose a new load balancing algorithm and VoD architecture for distributed VoD system. This architecture consists of a Central Multimedia Server [CMS]. A set of local Proxy servers are connected together in the form of a ring to form a Local Proxy Server Group [LPG]. All the LPG's are connected to the CMS. All connections are made through fiber optic cables. The rest of the paper is organized as follows: Section 2 presents the proposed architecture, section 3 presents the proposed algorithm, Section 4 presents the simulation model, Section 5 presents the simulation results and discussion, Section 6 finally concludes the paper and further work.

2. PROPOSED ARCHITECTURE

The proposed VoD architecture is as shown below:



A group of Proxy Servers are connected together in the form of a ring called Local Proxy Server Group [LPG]. A set of clients (users) are connected to each Proxy Server. This group of local Proxy Servers is connected to the Central Multimedia Server [CMS] through fiber optic cables. One of the Proxy Servers in the LPG acts as a coordinator and maintains a database which contains the information of the videos present in each Proxy Server in that LPG and also the popularity of the videos in that LPG. A mobile agent tours through the Proxy Servers of a LPG periodically to find and update the set of videos present in each Proxy Server of the LPG and the popularity of the videos in the LPG. This information is shared among all the Proxy Servers in the LPG.

Initially, all the N number of videos are stored in the CMS. The distribution of the videos is done as follows:

First, all the N videos are arranged based on their popularity. The popularity of a video is directly proportional to the number of hits for the video. The number of requests to a video follows Zipf law of distribution. We select the first m videos from the popularity based sorted list and stored in each proxy server. The remaining videos are stored depending on the local popularity in the proxy servers.

When a request for a video arrives at the PS, the following cases happen:

- The requested video is present at the PS[Proxy Server]
- The requested video is not present at the PS, but is present either in LPS[Left neighboring Proxy Server] or RPS[Right neighboring Proxy Server]
- The requested video is present in both LPS and RPS
- The requested video is not present in both LPS and RPS, but is present in one of the Proxy Servers in that LPG.
- The requested video is not present in any of the Proxy Servers in the LPG

If the requested video is present in the PS, then the real time transmission of the video starts immediately with the video content being streamed to the client from the PS. If the requested video is not present in the PS, then we check whether it is present in the LPS or in RPS.

If the requested video is present only in LPS, then we check the number of channels allocated for popular videos b/w LPS & PS and CMS & PS. If more numbers of channels are allocated for popular videos b/w LPS & PS, then path LPS-PS is selected, otherwise the path CMS-PS is selected. If the requested video is the first request, then all the channels are allocated for this video. Otherwise, depending on the popularity of the requested video, the channels are allocated as follows: If the requested video is more popular than the videos being streamed in the channels, then more number of channels are allocated for the requested video by deallocating channels from the lesser popular videos being streamed in the channels. Otherwise, appropriate numbers of channels are allocated depending on its popularity and the popularity of the videos streamed. Then the channel allocation of the other videos is dynamically adjusted, if needed.

If the requested video is present only in RPS, then we check the number of channels allocated for popular videos b/w RPS & PS and CMS & PS. If more numbers of channels are allocated for popular videos b/w RPS & PS, then path RPS-PS is selected, otherwise the path CMS-PS is selected. If the requested video is the first request, then all the channels are allocated for this video. Otherwise, the channel allocation is done in the same way as given above.

If the requested video is present in both RPS and LPS, then we check the number of channels allocated for popular videos b/w RPS & PS, LPS & PS and CMS & PS. Among these 3 paths, we select the path in which more number of channels are allocated for most popular videos. If the requested video is the first request, then all the channels are allocated for this video. Otherwise, the channel allocation is done in the same way as given above.

If the requested video is not present in both LPS and RPS, but is present in one of the Proxy Servers in that LPG, then appropriate number of channels are allocated within the LPG by finding the optimal path from the PS and the server in LPG having the requested video depending on the popularity of the requested video.

If the requested video is not present in any of the Proxy Servers in the LPG, Then the path PS-CMS is selected.

If the requested video is the first request, then all the channels are allocated for this video. Otherwise, the channel allocation is done in the same way as given above.

3. PROPOSED ALGORITHM

Nomenclature:
[PS: Proxy Server
LPS: Left neighboring Proxy Server
RPS: Right neighboring Proxy Server
NPS: Neighboring Proxy Servers
LPG: Local Proxy Server Group
CMS: Central Multimedia Server
NCAPV(x): Number of channels allocated for popular videos between PS and x]

```
Channel_allocation(y)
 {
   If (requested video is the first request)
         All the channels between PS and y are allocated to this video
   else
      {
       - Number of channels between PS and y are allocated depending on the available number
         of channels and proportional to the popularity of the requested video
       - Dynamically adjust the channel allocation for the other videos(if required)
      }
  }
```

When a request for a video m arrives at a particular time t, do the following:

```
If (requested video is present in PS)
        Start streaming from PS
else
 {
        If (requested video is present in only LPS)
          {
           If (NCAPV (LPS) >= NCAPV (CMS))
                   Channel_allocation (LPS)
            else
                   Channel_allocation (CMS)
          }
          else if (requested video is present in only RPS)
                  {
                      If (NCAPV (RPS) > =NCAPV (CMS))
                         Channel_allocation (RPS)
                       else
                          Channel_allocation (CMS)
                  }
          else If (requested video is present in LPS & RPS only)
                  {
                    If (NCAPV (RPS) >= NCAPV (CMS) and (NCAPV (RPS) >=NCAPV (LPS))
                            Channel_allocation (RPS)
                    If (NCAPV (LPS) >= NCAPV (CMS) and (NCAPV (LPS) >=NCAPV (RPS))
                            Channel_allocation (RPS)
                       else
                            Channel_allocation (CMS)
                  }
                 else if (requested video is present in LPG – NPS's)
                         {
```

```
                    If ((NCAPV (LPG-NPS's)>= NCAPV (CMS))
                        Channel allocation is done within the LPG by finding the optimal path
                        from the PS and the server in LPG having the requested video
                    else
                        Channel_allocation (CMS)
                    }
                else
                        Channel_allocation (CMS)
}
```

4. SIMULATION MODEL

The simulation model consists of a single Central Multimedia Server [CMS], and a few proxy servers in one local proxy server Group [LPG]. The following are the assumptions made in the model:

The user requests for the video follows Zipf law of distribution. The sizes of the videos are uniformly distributed over a range. The number of channels between PS & LPS, between PS & RPS, between the Proxy Servers in a LPG and between PS & CMS are assumed to be same.

The performance parameters are load sharing among the proxy servers, reduction of load on the CMS and channel utilization between PS & LNPS, PS & RNPS and PS & CMS and between the Proxy Servers of a LPG.

5. RESULTS & DISCUSSION

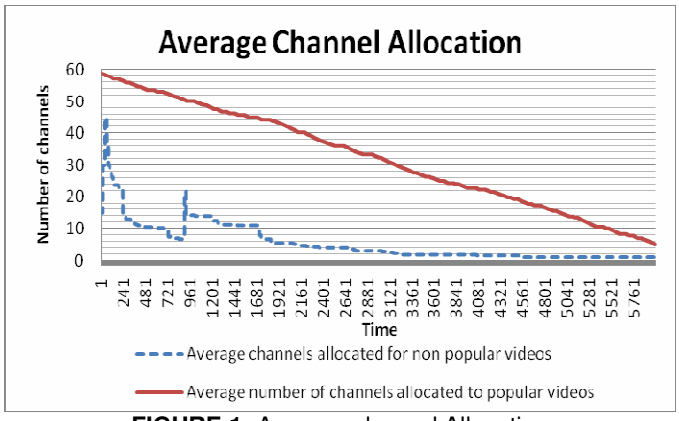


FIGURE 1: Average channel Allocation

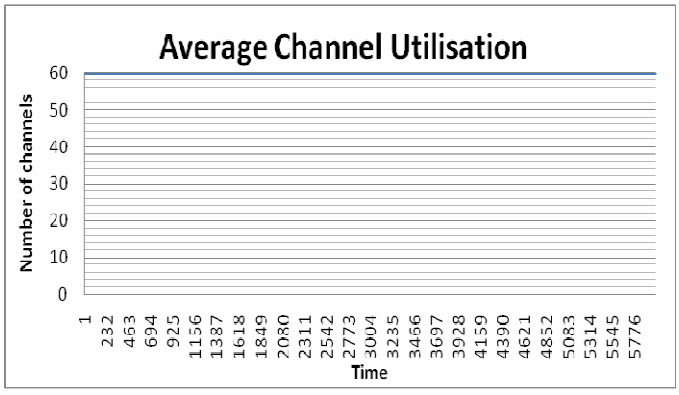


FIGURE 2: Average Channel Utilisation

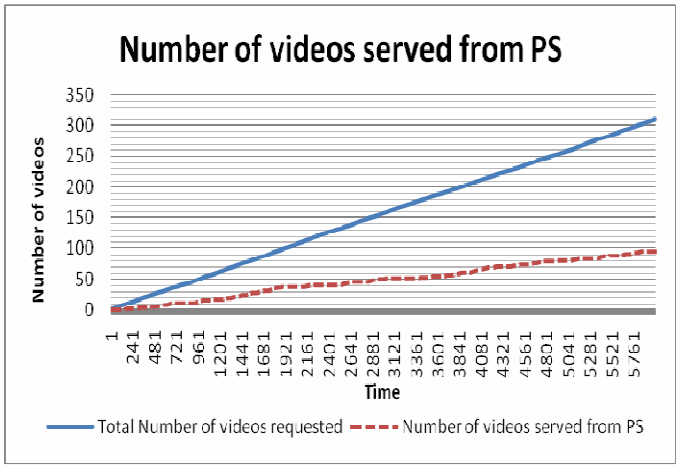


FIGURE 3: No. of videos served from PS

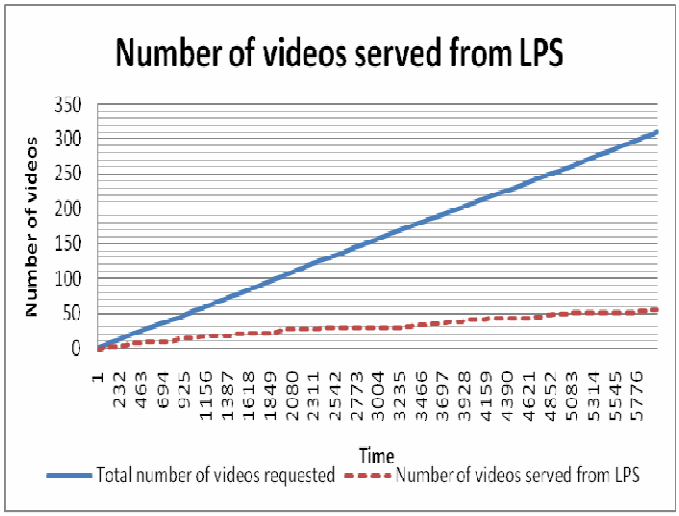


FIGURE 4: No. of videos served from LPS

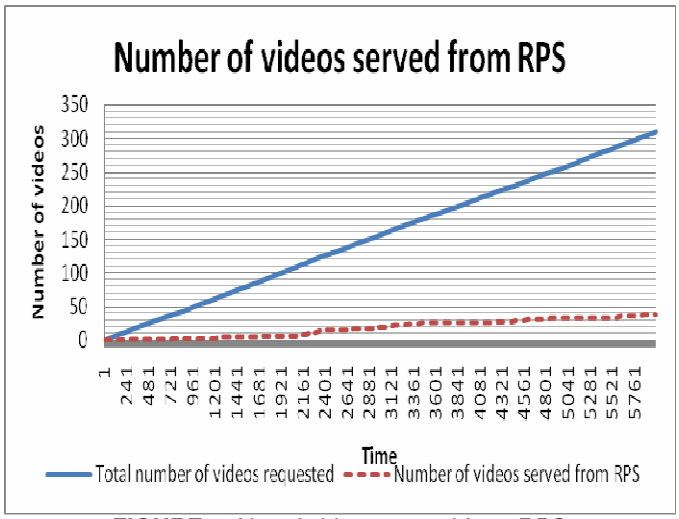


FIGURE 5: No. of videos served from RPS

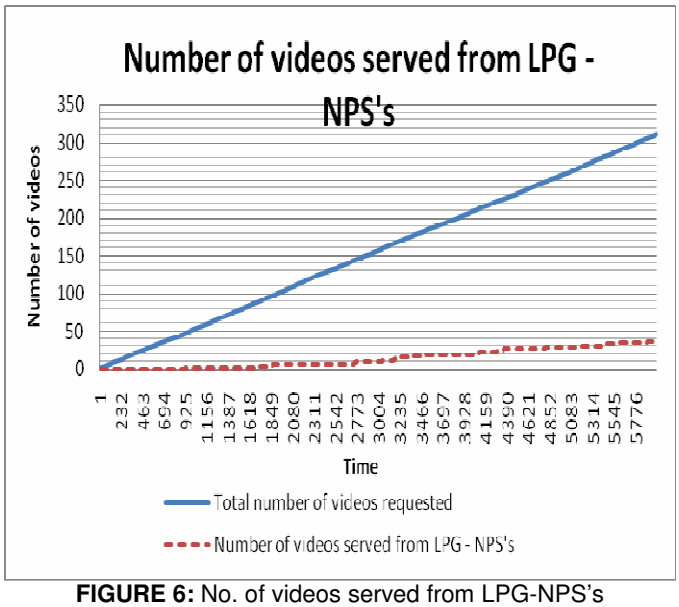


FIGURE 6: No. of videos served from LPG-NPS's

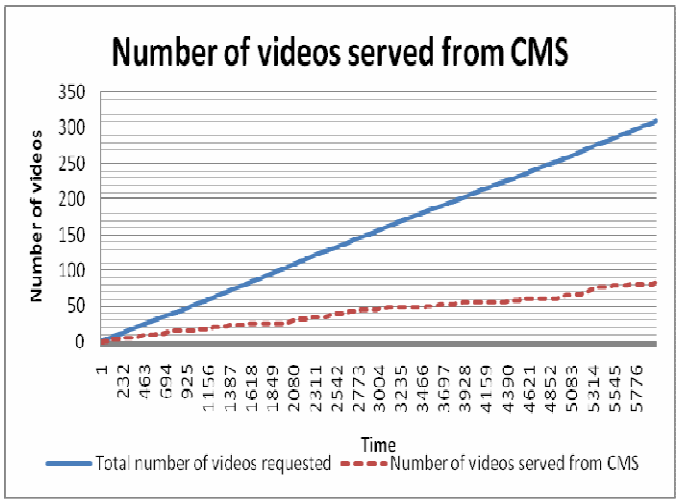


FIGURE 7: No. of videos served from CMS

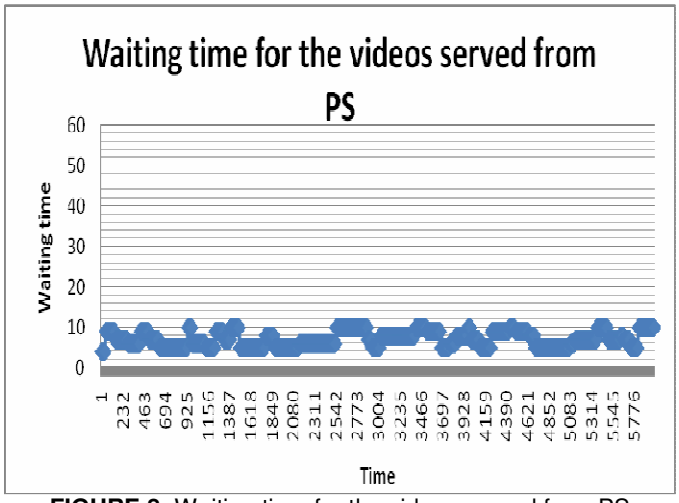


FIGURE 8: Waiting time for the videos served from PS

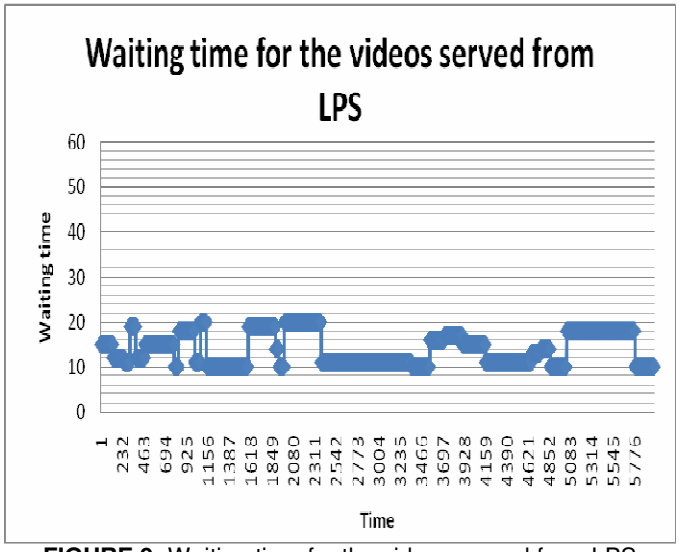


FIGURE 9: Waiting time for the videos served from LPS

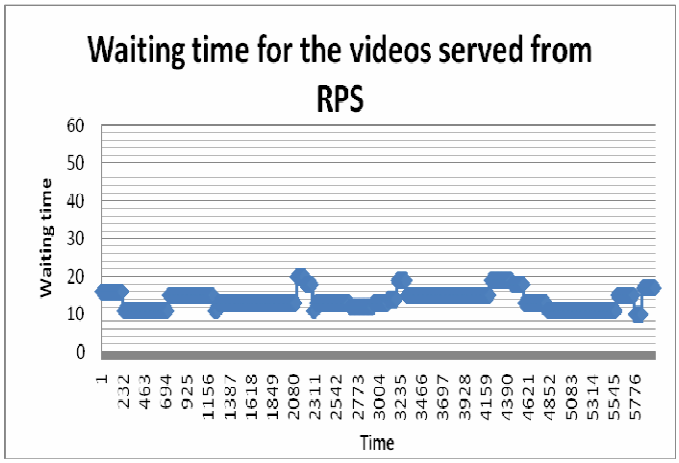


FIGURE 10: Waiting time for the videos served from RPS

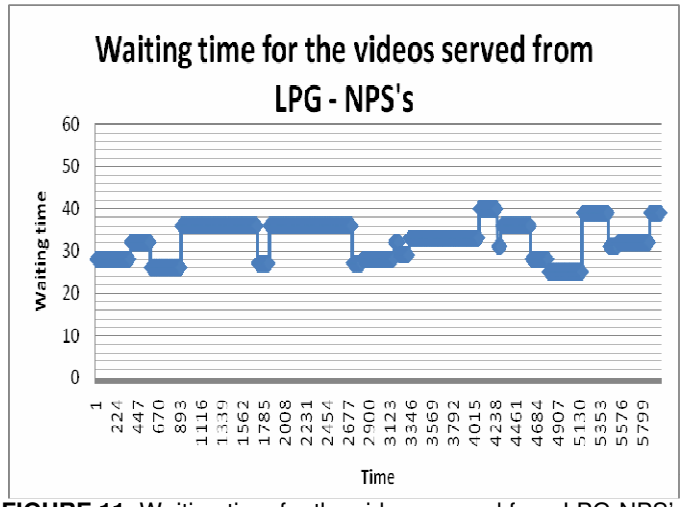


FIGURE 11: Waiting time for the videos served from LPG-NPS's

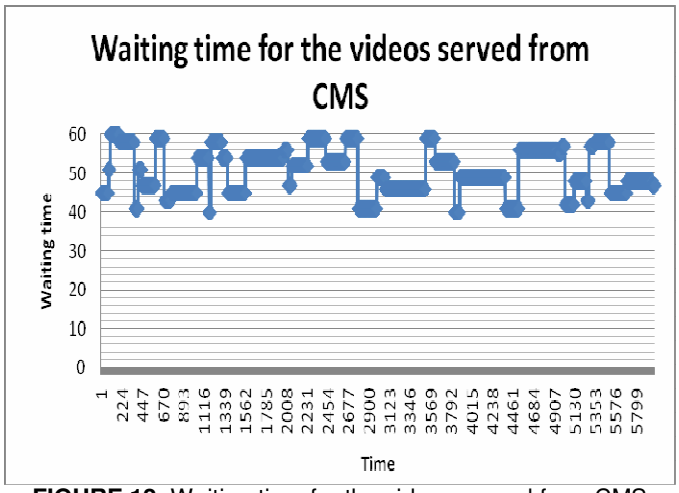


FIGURE 12: Waiting time for the videos served from CMS

The results presented are an average of several simulations conducted on the model. The sizes of the videos are taken in the range 300MB to 800MB. The number of proxy servers are considered in LPG is 6 and each simulation is carried out for 6000 seconds.

Fig 1 shows the average channel allocation for popular videos and non popular videos. Our channel allocation algorithm allocates more channels for popular videos than the non popular videos. The average channel utilization is always maximum because all the channels are allocated among the videos being streamed as shown in Fig 2.

The videos that are requested frequently are stored in the PS. When there is a request for these videos, streaming starts immediately and hence the waiting time for these videos is very less as shown in Fig 3 and Fig 8.

When the requested video is found in LPS or RPS, the streaming is initiated to the requested proxy server from the LPS or RPS and the waiting time is very small as shown in Fig 4, Fig 5, Fig 9 and Fig10.

When the requested video is found in some proxy servers other than LPS or RPS, the streaming is initiated to the requested proxy server and the waiting time is considerable as shown in Fig 6 and Fig 11.

When the requested video is not found in any of the proxy servers in the LPG, the video has to be streamed from the CMS and the waiting time is more as shown in Fig 7 and Fig12.

6. CONCLUSION

In this paper, we have concentrated on the load balancing among the proxy servers and central multimedia server using agents. The simulation shows promising results. The algorithm always uses maximum number of channels between the proxy servers in a LPG and also between the CMS and the proxy servers of a LPG by allocating more channels to the more popular videos. Further work is being carried out to investigate load balancing by optimally balancing the channels and the buffer.

REFERENCES

1. M Dakshayini, H S Guruprasad, H D Maheshappa, A S Manjunath, “Load Balancing in Distributed VoD using Local Proxy Server Group [LPSG]”, International Conference on

Computational Intelligence And Multimedia Applications 2007 (ICCIMA'07), Dec 13th -15th 2007, India.

2. Santosh Kulkarni "Bandwidth Efficient Video on Demand Algorithm (BEVA) " *10th International conference on Telecommunications*, Vol 2 pp 1335-1342, 2003

3. S S Manvi and P Venkataram, "Mobile Agent based online Bandwidth allocation Scheme in Multimedia Communications", *IEEE GLOBECOM 2001 Conference* USA

4. Y.C Tay and HweeHwa Pang, "Load Sharing in Distributed Multimedia-On-Demand Systems", *IEEE Transactions on Knowledge and data Engineering*, Vol.12, No.3, May/June 2000.

5. Meng Guo and Mostafa H. Ammar and Ellen W. Zegura, "Selecting among Replicated Batching Video-on-Demand Servers", *Proceedings of the 12th International Workshop on Network and Operating System Support for Digital Audio and Video*, pp 155—163, 2002

6. S. Gonzalez, A. Navarro, J. Lopez and E.L. Zapata, "Load Sharing in Distributed VoD (Video on Demand) Systems*". Int'l Conf. on Advances in Infrastructure for e-Business, e-Education, e-Science, and e-Medicine on the Internet (SSGRR 2002w*), L'Aquila, Italy, January 21-27, 2002.

7. Mohammed A. M. Ibrahim, "Distributed Network Management with Secured Mobile Agent Support", *International Conference on Hybrid Information Technology (ICHIT'06)*, 2006

8. P R Rao, Prasanna H Bammigatti," Delegation in Role Based Access Control Model for Workflow Systems", IJCSS: International Journal of Computer Science and Security, Volume 2, Issue 2, 2008.

9. Frederic Thouin, Mark Coates, Dominic Goodwill, "Video-on-Demand Equipment Allocation," *Fifth IEEE International Symposium on Network Computing and Applications (NCA'06*), pp 103-110, 2006

10. Hongliang Yu, Dongdong Zheng, Ben Y. Zhao, Weimin Zheng, "Understanding User Behavior in Large-Scale Video-on-Demand Systems", *Proceedings of the 2006 EuroSys conference*, Volume 40 , Issue 4 (October 2006), PP: 333 - 344

11. A.M.Natarajan, C.Venkatesh, R.Asokan, "Ant Based Dynamic Source Routing Protocol to Support Multiple Quality of Service (QoS) Metrics in Mobile Ad Hoc Networks", IJCSS: International Journal of Computer Science and Security, Volume 2, Issue 3, 2008.

12. A Dan, D Sitaram and P Shabuddin, "Dynamic batching policies for an on demand video server ", *Multimedia Systems,* pp 51-58, 1996

13. Gonzalez, A. Navarro, J. Lopez and E.L. Zapata, "A case study of Load sharing based on popularity in Distributed VoD systems", IEEE Transactions on Multimedia, Vol 8, No. 6, December 2006

14. Jiang Yu, Chun Tung Chou, Xu Du, Tai Wang, "Internal popularity of streaming video and its implication on caching",20th International Conference on Advanced Information Networking and Applications(AINA '06), 2006

15. Kamaksi Prasad V2, N H Ayachit1, Santosh L Deshpande1, "The time efficient security for broadcast Networks", IJCSS: International Journal of Computer Science and Security, Volume 2, Issue 2, 2008.

IMPLEMENTATION OF ARTIFICIAL NEURAL NETWORK IN CONCURRENCY CONTROL OF COMPUTER INTEGRATED MANUFACTURING(CIM) DATABASE

P. Raviram ravirampedu@gmail.com
Research Scholar/Department of Computer Science and Engineering
Vinayaka Mission's University
Salem, 636 308, INDIA

R.S.D. Wahidabanu rsdwb@yahoo.com
Professor & Head
Department of Electronics and Communication Engineering
Government College of Engineering
Salem, 636 011, INDIA

ABSTRACT

Manufacturing database store large amount of interrelated data. The designers access specific information or group of information in the data. Each designer accessing an entity tries to modify the design parameters meeting the requirements of different customers. Sister concerns of the same group of company will be modifying the data as per design requirements. When information is updated with new modification by different group of designers, what is the order in which modification of the data has to be allowed. If simultaneous access of the information is done, how to maintain the consistency of the data. and a designer voluntarily corrupts the data, how to make sure the designer is responsible for the corruption of data. In any case if the transaction process corrupts the data, how to maintain the consistency of the data. Deleting the information wantedly can be identified with extra security for the data. However, when transaction protocol is not implemented properly, then corruption of data in the form of misleading information that showing less numerical value than what it has to be or showing more numerical than before updation. In this research work, we have proposed a neural network method for the managing the locks assigned to objects and the corresponding transactions are stored in a data structure. The main purpose of using the ANN is that it will require less memory in storing the lock information assigned to objects. We have attempted to use backpropagation algorithm for storing lock information when multi users are working on computer integrated manufacturing (CIM) database.

Keywords: Concurrency control, locks, backpropagation algorithm, neural network, CIM database, Knowledge management.

1. INTRODUCTION

Knowledge management in advanced database have been considered as an interesting research area in the recent past. Real-Time database systems (RTDB) and Active database systems have been discussed and implemented respectively to support non-traditional applications. Few researches concentrate on the integration of active and real-time database systems and is very much used in computer integrated manufacturing (CIM). New problems are evolved in concurrency control (CC)[4,13] of real-time database systems. Conventional CC protocols are more concerned about the serializability but real-time database systems also focus on transaction deadlines. The situation is more complicated when real-time databases integrated with active characteristic. RTDBs must not only respond to the external transactions but also the internal triggered events. Due to the triggering structure in RTDBs, a dynamic CC algorithm is needed. If we simply apply existing conventional database or real time database CC protocols, a lot of CPU processing time will be wasted and transactions may not be able to complete before their deadlines.

2. TRANSACTION PROCESS

Transaction is series of actions, carried out by user or application, which accesses or changes contents of database. It is a logical unit of work on the database. It transforms database from one consistent state to another, although consistency may be violated during transaction[2,3]. Concurrency is the process of managing simultaneous operations on the database without having them interfere with one another. It prevents interference when two or more users are accessing database simultaneously and atleast one is updating data. Although two transactions may be correct in themselves, interleaving of operations may produce an incorrect result. Three potential problems caused by concurrency are lost update, uncommitted dependency and inconsistent analysis. Executions of transactions guaranteed to ensure consistency is identified by the concept of serializability with those schedule of reads / write. Serial schedule is where operations of each transaction are executed consecutively without any interleaved operations from other transactions. Nonserial Schedule: Schedule where operations from set of concurrent transactions are interleaved techniques used for concurrency Control are Locking and Timestamping. Both are conservative approaches when delay transactions in case they conflict with other transactions. Optimistic methods assume conflict is rare and only check for conflicts at commit.

Transaction uses locks to deny access to other transactions and so prevent incorrect updates. A transaction must claim a shared (read) or exclusive (write) lock on a data item before read or write. Lock prevents another transaction from modifying item or even reading it, in the case of a write lock. Rules of locking are, if transaction has shared lock on item, can read but not update item, and if transaction has exclusive lock on item, can both read and update item, Reads cannot conflict, so more than one transaction can hold shared locks simultaneously on same item, Exclusive lock gives transaction exclusive access to that item.

3. TRANSACTIONS REQUIREMENTS IN CAD DATABASE

Design and development of a product (shown in Figure 1) is the first and foremost step in a manufacturing industry. This process is recurrent and repetitive until it reaches a final approved design and development stage. Design and development activity involves defining and describing the product, drawing the product in the computer using computer aided drafting(CAD) software making modifications in the drawing, proving suitable material combinations for the product, defining various sizes for the product, providing safety factor provision based on the end application, satisfying customer requirements. The entire process will be generally interactive between a designer and the customer with one to one direct contact, or interactive discussions between designers at various locations, or independent design decisions by various designers who are located at different places and are accessing the same database which is centralized and sometimes distributed. When many designers are involved in designing an object in the database a major problem of concurrency as well as version of the product developed occurs.

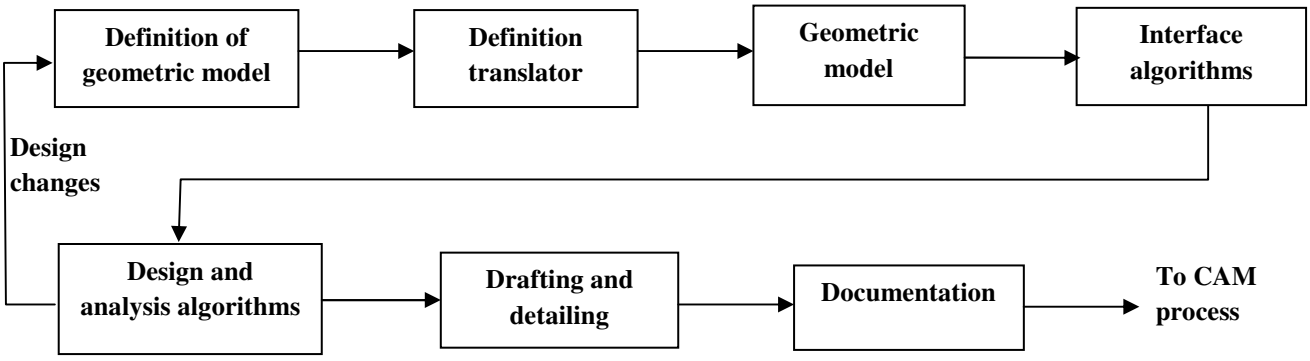


FIGURE 1: Schematic diagram of Design and development.

Majority of transaction will be done with long time gap. In the existing commercial database, all the equations and procedures are already coded with all the rules required which may change time to time.[12] In such case, whenever a user is accessing the data in the form of read / write the transaction quickly. He may not choose the choice of interest charged for the cash he swiped. using his credit card. This indicates transaction with minimum amount of time.

Transaction with longer time period : In case of computer design process even though design formulae have been encoded at the time of software development, the users will have their choice of choosing their design requirements[1,2]. They will do design based on their previous experiences and on the requirements of customers, based on the availability of machine capabilities in the workshop. The activity of deciding the optimum design will take long time for completing the transaction. Suppose, a similar design is done by another user very quickly, which may be based on some criteria, the question is whether the second person can be permitted to update the database or should he have to wait for the first person to the design process. What type of transaction concept that has to be adopted is based on many criteria which may not be readily fixed.

Controlling transaction with users choice: Most of the CAD transactions are based on interaction among many users. Atleast two users would transact the knowledge, discuss and come to a conclusion whether such design can be finalized. In such case, the final commit in the database should be possible. This should not become impossible because of basic transaction rules.

Cooperating the views of the designers in synchronization Many users working on shared objects cannot be serialized. The shared objects shall pass among them when modifying two parts of the same object parallelly to create new version of the object. In this case complete serializability is not possible. New concept is being developed to meet maximum seriazability in such type of shared interactive design environment.

4. EARLIER APPROACHES TO GROUP TRANSACTIONS

Developers of large project often work in small teams. A policy is to define the kinds of interactions allowed among members of the same team as opposed to interactions between teams. A group paradigm to deal with consistency of replicated data in an unreliable distributed system[3]. We can hierarchically divide the problem of achieving serializability into two simpler ones: (1) a local policy that ensures a total ordering of all transactions within a group; and (2) a

global policy that ensures correct serialization of all groups. Groups, like nested transactions, is an aggregation of a set of transactions[5]. There are differences between groups and nested transactions. A nested transaction is designed *a priori* in a structured manner as a single entity that may invoke subtransactions, which may themselves invoke other subtransactions. Groups do not have any *a priori* assigned structure and no predetermined precedence ordering imposed on the execution of transactions within a group. Another difference is that the same concurrency control policy is used to ensure synchronization among nested transactions at the root level and within each nested transaction.. The group paradigm was introduced to model inter-site consistency in a distributed database system. It can be used, to model teams of developers, where each team is modeled as a group with a local concurrency control policy that supports synergistic cooperation.

CAD Transactions in Groups : The group-oriented model does not use long-lived locks on objects in the public database. The conversational transactions model sets long-lived locks on objects that checked out from the public database until they are checked back into the public database. The group-oriented model categorizes transactions into *group transactions* (GT) and *user transactions* (UT). Any UT is a subtransaction of a GT. The model also provides primitives to define groups of users with the intention of assigning each GT a user group. Each user group develops a part of the project in a *group database*. A GT reserves objects from the public database into the group database of the user group it was assigned. Within a group database, individual designers create their own user database, and they invoke UTs to reserve objects from the group database to their user database.

In the group-oriented model, user groups are isolated from each other. One user group cannot see the work of another user group until the work is deposited in the public database. Group transactions are thus serializable. Within a group transaction, several user transactions can run concurrently. These transactions are serializable unless users intervene to make them cooperate in a non-serializable schedule. The basic mechanism provided for relaxing serializability is a version concept that allows parallel development (branching) and notification. Versions are derived, deleted, and modified explicitly by a designer only after being locked in any one of a range of lock modes.

The model supports five lock modes on a version of an object: (1) read-only, which makes a version available only for reading; (2) read/derive, which allows multiple users to either read the same version or derive a new version from it; (3) shared derivation, which allows the owner to both read the version and derive a new version from it, while allowing parallel reads of the same version and derivation of different new versions by other users; (4) exclusive derivation, which allows the owner of the lock to read a version of an object and derive a new version, and allows only parallel reads of the original version; and (5) exclusive lock, which allows the owner to read, modify and derive a version, and allows no parallel operations on the locked version. Using these lock modes, several designers can cooperate on developing the same design object. The exclusive lock modes allow for isolation of development efforts (as in traditional transactions), if that is what is needed. To guarantee consistency of the database, designers are only allowed to access objects as part of a transaction. Each transaction in the group-oriented model is two-phase, consisting of an acquire phase and a release phase. Locks can only be strengthened (i.e., converted into a more exclusive mode) in the acquire phase, and weakened (converted into a more flexible lock) in the release phase. If a user requests a lock on a particular object and the object is already locked with an incompatible lock, the request is rejected and the initiator of the requesting transaction is informed of the rejection. This avoids the problem of deadlock, which is caused by blocking transactions that request unavailable resources. The initiator of the transaction is notified later when the object he requested becomes available for locking. In addition to this flexible locking mechanism, the model provides a read operation that breaks any lock by allowing a user to read any version, knowing that it might be about to be changed. This operation provides the designer (more often a manager of a design effort) the ability to observe

the progress of development of a design object, without affecting the designers doing the development.

5. BACKPROPAGATION ALGORITHM (BPA)

An artificial neural network(ANN)[7] is an abstract simulation of a real nervous system that contains a collection of neuron units, communicating with each other via axon connections. Such a model bears a strong resemblance to axons and dendrites in a nervous system. Due to this self-organizing and adaptive nature, the model offers potentially a new parallel processing paradigm. This model could be more robust and user-friendly than the traditional approaches. ANN can be viewed as computing elements, simulating the structure and function of the biological neural network. These networks are expected to solve the problems, in a manner which is different from conventional mapping. Neural networks are used to mimic the operational details of the human brain in a computer. Neural networks are made of artificial 'neurons', which are actually simplified versions of the natural neurons that occur in the human brain. It is hoped, that it would be possible to replicate some of the desirable features of the human brain by constructing networks that consist of a large number of neurons. A neural architecture (shown in Figure 2) comprises massively parallel adaptive elements with interconnection networks, which are structured hierarchically.

The BPA[6] uses the steepest-descent method to reach a global minimum. The number of layers and number of nodes in the hidden layers are decided. The connections between nodes are initialized with random weights. As shown in Table 1 a pattern from the training set is presented in the input layer of the network and the error at the output layer is calculated. The error is propagated backwards towards the input layer and the weights are updated. This procedure is repeated for all the training patterns. At the end of each iteration, test patterns are presented to ANN, and the classification performance of ANN is evaluated. Further training of ANN is continued till the desired classification performance is reached.

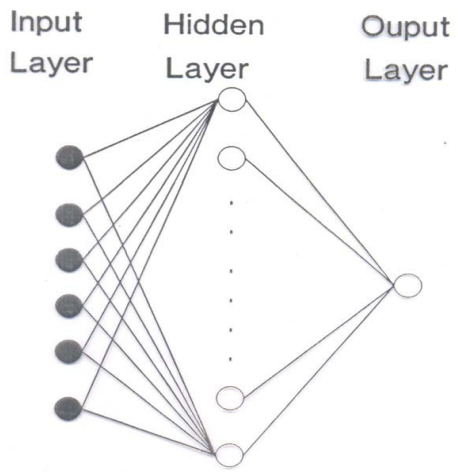


FIGURE 2: Multilayer perception.

6. INTELLIGENT LOCKING STRATEGY PROPOSED LOCKING IMPLEMENTATION

Inbuilt library functions for the bolt are available in standard CAD[8, 9,10,11, 12] softwares. In Figure 3, the bolt and its entities are shown schematically. The bolt drawing (Figure 3) is used to manufacture one variety of bolt that has to be used to clamp two plates. This drawing file will be accessed by many designers who will choose their choice of designs and the designs are stored

in the same location of the server. Each designer can choose their option of changing the shaped , dimension of different components of the subassembly of the bolt. When any modification is done for one subassembly, due to associative dimensioning concept, the dimensions of the entire bolt shape and dimensions can change.

Problems faced in storing the modified drawing. If it is associative dimensioning, then all the changes in shape and size of the bolt system have to be updated for a small change in the dimension of the subassembly. At the same time, another designer would want to retain earlier version. of the drawing.

When more than one user adopts similar changes in the file, and when they commit at the same time, how to maintain the consistency of the bolt file.

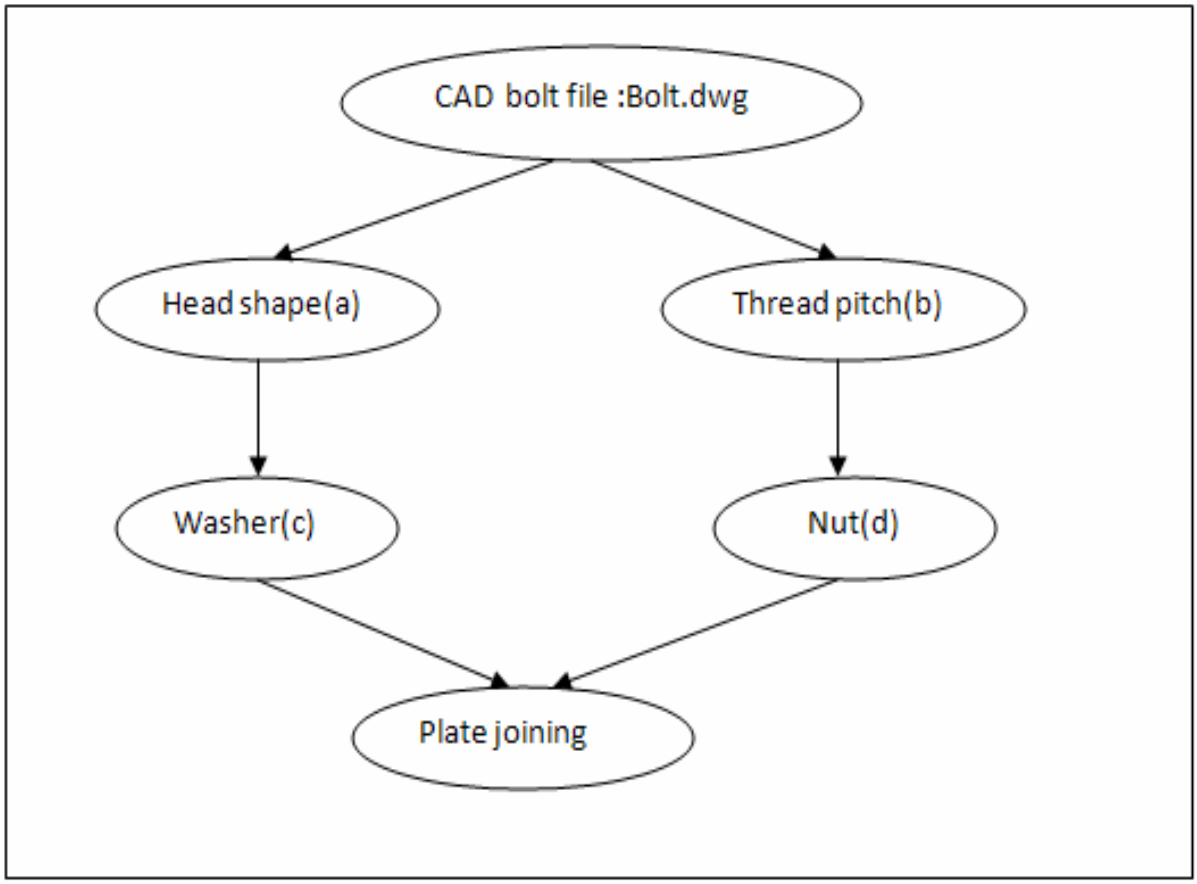


FIGURE 3: A bolt file contains entities.

STEPS INVOLVED.

FORWARD PROPAGATION

The weights and thresholds of the network are initialized.

The inputs and outputs of a pattern are presented to the network.

The output of each node in the successive layers is calculated.

o(output of a node) = $1/(1+\exp(\sum w_{ij} x_i + \Theta))$

The error of a pattern is calculated

$E(p) = (1/2) \sum(d(p) - o(p))^2$

REVERSE PROPAGATION

The error for the nodes in the output layer is calculated

δ(output layer) = o(1-o)(d-o)

The weights between output layer and hidden layer are updated

$W(n+1) = W(n) + \eta\delta$(output layer) o(hidden layer)

The error for the nodes in the hidden layer is calculated

δ(hidden layer) = o(1-o) $\sum\delta$(output layer) W(updated weights between hidden and output layer)

The weights between hidden and input layer are updated.

$W(n+1) = W(n) + \eta\delta$(hidden layer) o(input layer)

The above steps complete one weight updation

Second pattern is presented and the above steps are followed for the second weight updation.

When all the training patterns are presented, a cycle of iteration or epoch is completed.

The errors of all the training patterns are calculated and displayed on the monitor as the mean squared error(MSE).

$E(MSE) = \sum E(p)$

TABLE 1: Back-propagation algorithm.

In Table 2 defines the variables used for training the ANN about locks assigned to different objects.

User	Object	mode

TABLE 2: Variables considered for training and testing of ANN for lock management.

where
User represents the client
Object represents the entire Manufacturing related file or an entity in the file
Mode represents type of lock assigned to an object.

exclusive *(X) mode*. Data item can be both read as well as written.
shared *(S) mode*. Data item can only be read..
intention-shared (IS): indicates explicit locking at a lower level of the tree but only with shared locks.
intention-exclusive (IX): indicates explicit locking at a lower level with exclusive or shared locks
shared and intention-exclusive (SIX): the subtree rooted by that node is locked explicitly in shared mode and explicit locking is being done at a lower level with exclusive-mode locks.
A intention locks allow a higher level node to be locked in S or X mode without having to check all descendent nodes.

In Table 3 , column 1 represents the lock type. Column 2 represents the value to be used in the input layer of the ANN in module 1 and module 3. Column 3 gives binary representation of Lock type to be used in the output layer of module 1 and module 3. The values are used as target outputs in the module 1 and module 3 during lock release on a data item.

Table 4 shows two transactions T1 and T2 in the first column. Each transaction requests object a or b with a lock mode S or X. The fourth column indicates if any one of the lock is assigned for the object and otherwise '0' if no lock is assigned to the object.

Lock type	(Input layer representation numerical value).	Binary representation in target layer of the ANN
S	1	001
X	2	010
IS	3	011
IX	4	100
Object Not locked	0	000

TABLE 3: Binary representation of lock type.

User / Intermediate transaction	Object (a)	Object (b)	Mode S,X ,IS,IX	Lock Enabled – (1) Otherwise (0)
T1	a	-	S	1

T2	a	-	S	1
T1	a	-	X	1
T1	-	b	S	1
T2	a	-	X	1
T1	-	b	X	1

TABLE 4: Sample sequence of object access by two users.

This work uses for modules of algorithm which work using BPA given in Table 1. The modules given in Table 5 gives their usage for learning and finding the lock states. OML(Object, Mode, Lock) and OL (Object Mode)

Module	Name	Training / Testing	ANN Topology
1	UML	Training (Figure 4)	2{user number and mode} x {no. of nodes in hidden layer} x 3{Lock value}
2	UML	Testing (Figure 5)	2{user number and mode} x (no. of nodes in hidden layer) x 3(Lock value)
3	XL	Training (Figure 6)	1{user} x 2 {no. of nodes in hidden layer} x 3{lock value}
4	XL	Testing (Figure 7)	1{user} x 2 {no. of nodes in hidden layer} x 3{lock value}
In the fourth column of this table, 3 values are given in the order, no. of nodes in the input layer, no. of nodes in the hidden layer which can be anything and no. of nodes in the output layer which is 3(fixed)			

TABLE 5: Modules used for learning the lock status of an object.

OML training

IL **HL** **OL**

(Object name, mode **(Lock value is trained)**

are presented)

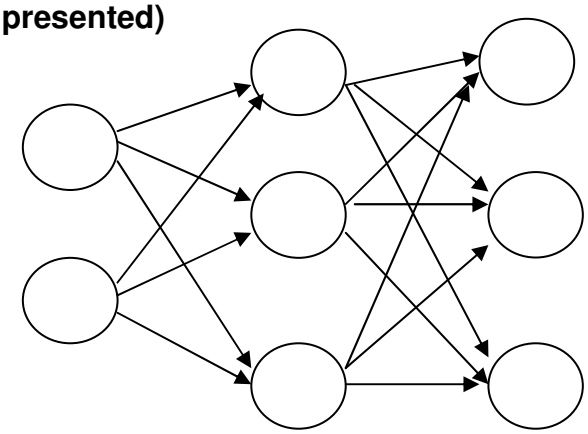


FIGURE 7: OML trining

OML testing

IL **HL** **OL**

(Object name, mode **(Lock value is obtained)**

are presented)

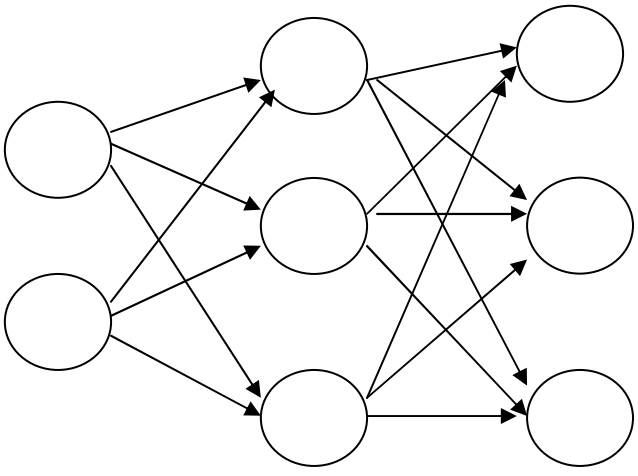


FIGURE 5: OML testing

OL training

IL **HL** **OL**

(Object name **(Lock value is trained)**

is presented)

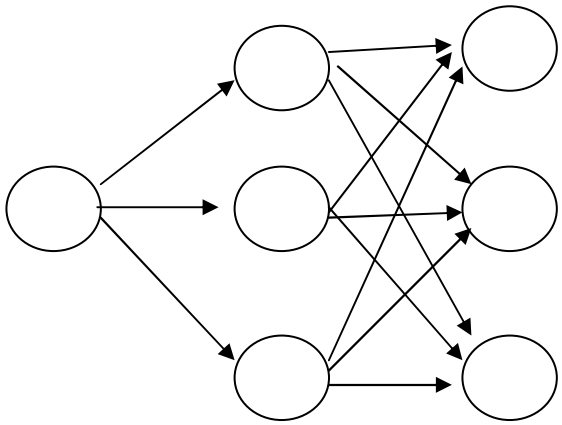


FIGURE 6: OL trianing

OL testing

IL **HL** **OL**

(Object name **(Lock value is obtained)**

is presented)

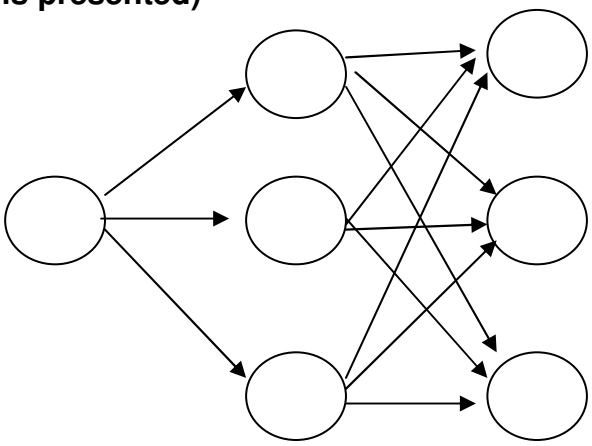


FIGURE 7: OL testing

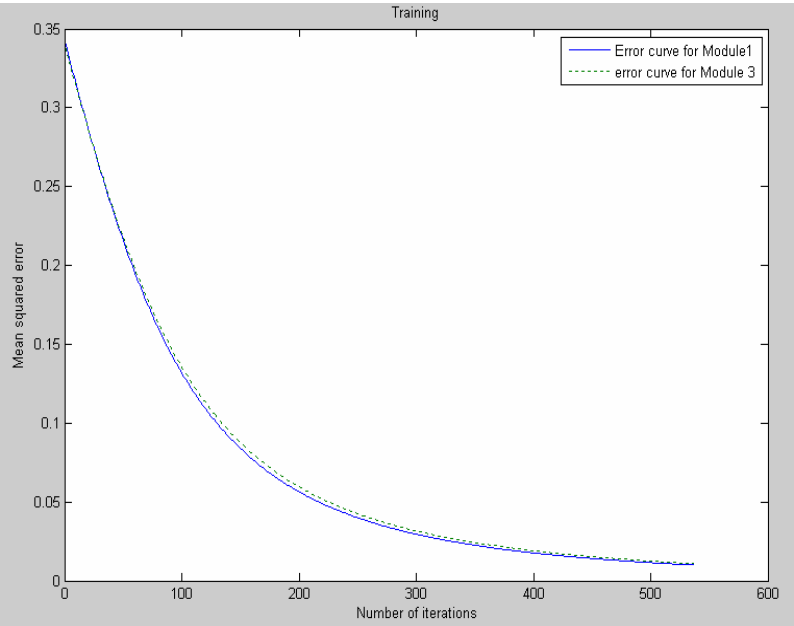


FIGURE 8: Mean squared error convergence.

Sequence of modules executed when a transaction requests lock or releases lock

1. Initialize randomly the weights of module 1 and module 3
2. A transaction T_i requests lock on an object (a ,b,)
3. Module 4 is tested with object (a,b, …) requested in step 2 to obtain binary value. If '000' is output in the output layer of module 4, then the object is free to be accessed. If (001, 010, 011, 100 is output then the object is under use. If the output value is 001, then the transaction in step is given access to the requested object
4. In any case , if T_i is given transaction to requested object, then module 1 and module 3 are weight updated using the backpropagation algorithm (forward and backward steps)
5. In any case , if the object is under any lock mode other than shared or no lock, then the transactions is kept under queue.

7. RESULTS AND DISCUSSIONS

The proposed ANN for lock state learning and lock state finding have been implemented using Matlab 7. Module 1 and Module 3 are trained until a Mean Square Error value of 0.01 is reached. The time for convergence to reach 0.01 is at an average of 3sec. The Figure 7, shows number of iterations versus mean squared error for Module 1 and Module 3.

8. CONCLUSION

An approach has been attempted to implement ANN in concurrency control. The approach has to be verified with different types of files operated by many users in a distributed environment. Different types of ANN algorithms can be attempted to achieve concurrency control in CAD database application.

9. REFERENCES

1. F. Bancilhon, W. Kim and H. Korth. “A Model of CAD Transactions”. In Proceedings of the 11th International Conference on Very Large Data Bases, Morgan Kaufmann, August, 1985, pp. 25-33

2. K. Salem, H. Garcia-Molina and R. Alonso. “Altruistic Locking: A Strategy for Coping with Long Lived Transactions”. In Proceedings of the 2nd International Workshop on High Performance Transaction Systems , September, 1987, pp. 19.1 - 19.24

3. P. Klahold, G. Schlageter, R. Unland and W. Wilkes, “A Transaction Model Supporting Complex Applications in Integrated Information Systems”. In Proceedings of the ACM SIGMOD International Conference on the Management of Data, ACM Press, May, 1985, pp. 388-401

4. A. H. Skarra, and S. B. Zdonik. “Concurrency Control and Object- Oriented Databases”. In Kim, W., and Lochovsky, F. H., Ed., *Object-Oriented Concepts, Databases, and Applications*, ACM Press, New York, NY, 1989, pp. 395-421

5. M. F. Fernandez and S. B. Zdonik. “Transaction Groups: A Model for Controlling Cooperative Work”. In Proceedings of the 3rd International Workshop on Persistent Object Systems, January, 1989

6. S. Purushothaman, Y. G. Srinivasa. “A back Propagation Algorithm applied to Tool wear Monitoring”. International Journal of Tools Manufacture ,1994, Vol 34, No 5, pp 625-631

7. S.Purushothaman, Y.G Srinivasa. “A procedure for training an artificial neural network with application to tool wear monitoring”. INT J.PROD RES., 1998, Vol., 36, No.3, 635-651

8. M. L. Brodie, B. Blanstein, U. Dayal, F. Manola, A. Rosenthal. “CAD/CAM Database Management”. IEEE Database Engineering, vo1.7, N0.2, pp. 12-20, June 1984

9. M. P. Groover and E. W. Zim mers. “CAD/CAM: Computer-Aided Design and Manufacturing”. Prentice-Hall, New York, *NY,* 1984

10. M. A. Ketabchi, V. Berzins. “Modeling and Managing CAD Databases”. IEEE Computer, February 1987, pp. 93-102

11. S. G. Landis. “Design Evolution and History in an Object-Oriented CAD/CAM Database”. IEEE, Proc.

12. Alexandtos Biliris, Huibin Zhao. “Design Versions in a Distributed CAD Environment”. 1989 IEEE, PP 354-359

13. A. A. Akintola, G. A. Aderounmu and A. U. Osakwe and M.O. Adigun. “Performance Modeling of an Enhanced Optimistic Locking Architecture for Concurrency Control in a Distributed Database System”. Journal of Research and Practice in Information Technology, Vol. 37, No. 4, November 2005

COMPUTER SCIENCE JOURNALS SDN BHD
M-3-19, PLAZA DAMAS
SRI HARTAMAS
50480, KUALA LUMPUR
MALAYSIA