14

X.500 Performance with Large Databases and OSInet[1]

P.V. Rajeev[2] S.V Raghavan[3]
Department of Computer Science and Engineering
Indian Institute of Technology, Madras, India
rajeev%hss@lando.hns.com
svr@iitm.ernet.in
Ashok K. Agrawala
Department of Computer Science,
University of Maryland,
College Park, MD USA.
agrawala@cs.umd.edu

Abstract

The X.500 directory provides a powerful mechanism for storing and retrieving information about objects of interest in the networking world. The rapid increase in the deployment of the directory has resulted in a very large amount of data in the global directory. Hence, the need arises for the development of a model which predicts the performance of the directory in an internet environment with large amount of data.

In this paper, we present a deterministic model for predicting the performance of X.500 directory operations. Actual measurements are carried out in a real life environment, with the directory having a large amount of data distributed across the internet. The model is validated by comparing the results of these actual measurements with the predicted values. The application of the model for the selection of DIT structure in a wide area network is explained.

Keywords:

X.500, Directory Services, Performance, Deterministic Modelling.

[1]Network Software developed at Indian Institute of Technology, Madras.
[2]Currently with Hughes Software Systems, New Delhi
[3]Further correspondence may please be addressed to Dr. S.V. Raghavan.

S. V. Raghavan et al. (eds.), *Computer Networks, Architecture and Applications*

1 INTRODUCTION

With the rapid expansion in the electronic communication, there is a growing need for an organized global information service. The X.500 directory provides such a powerful mechanism for storing and retrieving information about objects of interest in the networking world[CCITT (1988), Huitema (1989), Bumbulis *et al.*(1993)]. X.500 directory defines an architecture whereby the directory can be distributed over an arbitrary number of systems. Each system manages a portion of the global directory. Typically, in the X.500 directory, the information is distributed and managed across the network of many autonomous administrations. These systems are integrated into a single directory by a set of protocols. The directory stores information about *objects* and defines a hierarchical relationship between *objects* stored in the directory. The hierarchical structure reflects the national and organizational structure of the real world objects.

The X.500 directory was originally designed for use with X.400 message handling systems (MHS)[CCITT (1988)]. Message handling system uses directory for a variety of purposes including: retrieval of O/R address when O/R name is given, user friendly naming of users in the MHS, expansion of distribution lists, e-mail routing, etc. To provide these services to the MHS, the directory must store information about all e-mail users in the network. The number of e-mail users in the network itself will be of the order of millions. A recent survey on the usage of internet indicated a very high rate of increase in the number of users. A large amount of information has to be stored in the directory to cater to the needs of MHS.

Apart from MHS, directory is now being used for a variety of other applications. For example directory is now used to store the topology and service related information about communication networks [Mansfield *et al.*(1993), Hong *et al.*(1992a), Hong *et al.*(1992b)], to map object names to internal object identifiers in object based distributed operating systems [Rennesse *et al.*], to store bibliographic information [Barker (1992)], to store the information about ftp archives [Barker (1993)], and so on. Once all these applications start using the directory to store information, directory will be used extensively and an enormous amount of data will be stored in the directory.

The X.500 directory deployment in the internet is now progressing at a very fast pace. Presently, there are more than 450 Directory System Agents (*DSAs*) spanning 35 nations, covering a total of more than a million entries [Goodman (1992)]. The growth rate is found to be significant, with about 150 *DSAs* added every year since November 1990, with a proportional increase in the number of entries. With this rate of increase in the deployment of directory in the internet, the internet directory will soon be populated with enormous amount of information. The information about the e-mail users itself will be substantial. The internet directory is also used for storing information about various services available in the internet. Directory helps in giving an integrated view of the internet and the services available in the internet.

The concern of the designers is to estimate the critical database size and interconnection topologies among the *DSAs* and understand the implication of chaining and referral and guarantee response time whenever possible. This necessitates the study of the performance of the X.500 directory system, with special emphasis on large databases, wide area distribution and variable interconnections through different topologies.

In this paper, we present the performance measurements carried out in an internet environment with the directory distributed over two sites in the internet. A model for estimating the performance of the directory is also described.

The rest of the paper is organized as follows. Section 2 describes the X.500 and its implementation over OSInet. A deterministic model for X.500 is presented in the section 3. Section 4 validates the model with the help of actual measurements. The application of the model for the selection of DIT topology in a wide area network is explained in section 5.

2 X.500 OVER OSInet

The X.500 directory used for the performance measurements presented in this paper, is implemented [Sriram (1991), Sriram *et al.*(1994), Raghavan *et al.*(1992)] over OSInet network kernel. OSInet [Raghavan (1990), Raghavan (1993a)] is a network protocol software which provides a subset of services defined in ISO-OSI reference model. OSInet has a network kernel which provides the basic networking support and user applications that run over the kernel. The kernel provides the services required for a reliable networking environment. OSInet follows the layered architecture of the ISO-OSI reference model. The interfaces to upper and lower layers are implemented using queues and the layer specific actions are realized through finite state machines. The OSInet architecture is shown in Figure 1a.

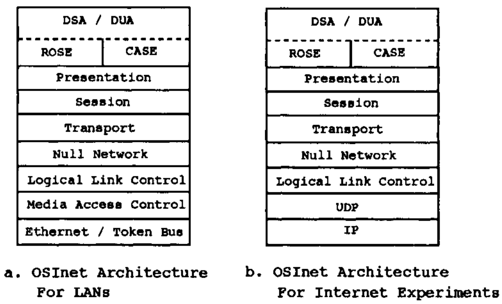


Figure 1: The OSInet Architecture.

The X.500 implementation makes use of ROSE and ACSE of OSInet. The binding and unbinding to *DSAs* are done by using the ACSE. ROSE is used to invoke the directory operations. For the internet experiments the lower layers of OSInet stack is modified to include UDP as shown in Figure 1b.

The X.500 standard [CCITT (1988)] does not specify any database mechanism for storing the directory information. It is upto the implementor to choose an appropriate storage mechanism. In the case of OSInet X.500, B-Tree++ indexing mechanism is used [Comer (1979), Raghavan *et al.*(1993b)] for the database access. The tree structure of the directory is mapped onto the underlying file systems' directory structure. The knowledge information regarding the location of various naming contexts known to *DSAs* is stored in a sub-directory called *tree directory*, which is the working directory of a *DSA*. The information about an entry is stored as pairs <attribute type, attribute value> in the directory information files. The knowledge information file is indexed using the distinguished name as the key and the directory information file is indexed using the relative distinguished name (RDN) as the key.

3 A DETERMINISTIC MODEL FOR X.500 DIRECTORY

A deterministic model for distributed directory services allows one to investigate the effect of various design choices and mode of interaction between the *DSAs* on the cost of directory operations. Very few attempts have been made so far in the literature to quantify the performance of a distributed directory[Neufeld (1992)].

The major factors that affect the performance of the directory in a large internet environment are,

- the communication delay between the Directory User Agents (*DUAs*) and *DSAs*, and between *DSAs*,
- the database access time, and
- the name resolution time.

In this section we discuss a model for the performance estimation of the directory based on these factors.

Typically, a directory service consists of a number of *DSAs* distributed throughout the internet. In our model, we assume that the *DSAs* are always available and there are no communication link failures. A *DSA* is run as a single process in the system.

The *DUAs*, or the clients of the directory services, may be a specific program, or it may be a part of another program which uses the directory for some lookup operation. For example, it may be a MHS UA which uses the directory to provide user friendly names to the MHS users. A *DUA* need only know the address of its closest *DSA*.

If a *DSA* is unable to service a request, it may chain the request to another *DSA* which may be able to answer the request. This may be repeated many times to resolve the name and to locate the requested entry. The *DSA* may also return a referral back to the requested *DUA* which in turn uses the referral to contact another *DSA* whose address is returned in the referral. This *DSA* can resolve the name if the requested object is present in its local database or can give a referral to another *DSA* which holds the entry or is more closer to the requested information. These modes of interactions are called chaining and referral respectively.

3.1 Communication

The roundtrip time between the *DUA* and the DSA_i is given by C_i. C_i strongly depends on the sites at which the *DSA* and the *DUA* are executing. This value depends on the number of gateways between the *DSA* and *DUA* and also on the speed of the intermediate transmission links. We assume that the number bytes per packet for different operations are the same. Thus the different operations have the same roundtrip time. We also ignore the variations due to the network congestion. The round trip time between a DSA_i and a DSA_j is given by C_{ij}.

3.2 Name resolution

The *DSA* resolves the name by scanning through its knowledge information base. If the requested object is located in locally held naming context, the *DSA* goes down the tree and locates the requested object. If the *DSA* finds that the requested object is not held locally, it finds out the address of the *DSA* which stores the object or a *DSA* which can give a more precise location of the *DSA* which stores the object. Then it chains to the that *DSA* or returns a referral back to the requested *DUA*. Let N_j be the average amount of time required to perform the name resolution in the DSA_i.

3.3 Database Access

The second factor that affects the performance of the directory is the database access time. As we are storing enormous amount of data in the directory, the database access time will contribute to a significant portion of the response time as seen by the client. The database access time depends on two factors: the object accessed and the type of operation requested. Let the time required to perform the database operation be D_i to perform an operation in the DSA_i. This is taken as an average amount of time taken to access different objects in the directory.

3.4 Other factors

Another factor that affect the performance of the directoty is the protocol overhead due to the lower layers. The time spent in the protocol stack, P_i, is assumed to be constant for all objects and all operations. This will be true in most of the cases because the amount of data transferred by the directory operations will be almost the same. The protocol overhead depends mainly on the size of the PDUs transferred.

Partitioning of the name space also influences the performance of the directory. If enough cross references are stored in the *DSAs*, then the *DSAs* will be able to resolve the name by contacting fewer number of *DSAs*. Also by suitably selecting the higher level *DSAs*, the performance of the system can be improved.

3.5 Scenarios

Now, we proceed to find out the response time of the directory as seen by the user of the directory. We consider the following three different cases:

i The local *DSA* itself can answer the query.

ii The local *DSA* is unable to resolve the query and uses chaining mode of interaction.

iii The local *DSA* is unable to resolve the query and uses referral.

We consider each of these three cases separately and find out the response time required in each case. In all these cases, we assume that the *DUA* is already bound to its closest *DSA* (primary *DSA*).

3.6 Case I - The Simple Case

In this case the interactions that are taking place is shown in the Figure 2. The *DUA* sends a request to the *DSA* and *DSA* reads its local database and responds to the query.

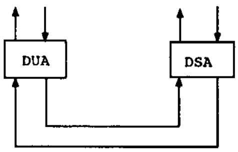


Figure 2: Single *DSA*, Single *DUA*

The response time as seen by the users of the *DUA* for performing an operation can be given by:

$$\boxed{T = P_c + C_1 + P_1 + N_1 + D_1}$$

Where,

P_c = Protocol overhead in the client (DUA).

C_1 = Round trip communication delay between the DUA to DSA_1.

P_1 = Protocol overhead in the DSA_1 (including queuing delay).

N_1 = Name resolution time in DSA_1.

D_1 = Database access time in DSA_1.

3.7 Case II - Chaining

In this case, the primary *DSA*, DSA_1, decides to chain the request to a remote *DSA* which can answer the query. The interactions that are taking place are shown in the Figure 3. The primary *DSA* has to bind to the remote *DSA* before invoking the remote operation on the second *DSA*. Thus the total time required to perform the operation includes the time required to bind to the remote *DSA* + the time required to chain the request to the remote *DSA* + the time required to merge the results obtained from the remote *DSA* and send the result back to the *DUA*.

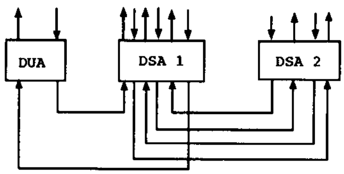


Figure 3: Multiple *DSAs*, Chaining mode of interaction

Thus the response time as seen by the user of the system is

T = time taken by the *DUA* to send the request + time taken by the *DSA* to perform the complete operation (include the time to bind to the second *DSA* and to invoke remote operation on the remote *DSA*) + time required to merge the results and return the result to the *DUA*

$$T = P_c + C_1 + T_{o1} + rm_1 + P_1$$

P_c = Total protocol overhead in the client (DUA).

C_1 = Round trip communication delay between DUA and DSA_1.

T_{o1} = Total time required by the DSA_1 to complete the operation including chaining.

rm_1 = Time required to merge the results.

P_1 = Total protocol overhead in the DSA_1 (including the queuing delay).

T_{o1} consists of three parts. Name resolution time in the first *DSA* + time required to bind to remote *DSA* + time required to perform the operation on the remote *DSA*.

$$T_{o1} = N_1 + T_b + T_o$$
$$T_b = bf_1 + C_{12} + P_2 + b_2 + P_1$$
$$T_o = qf_1 + C_{12} + P_2 + N_2 + D_2 + P_1$$

$N_1 =$ time required for name resolution in DSA_1.

$bf_1 =$ Time taken to form the bind request.

$C_{12} =$ round trip communication delay between DSA_1 to DSA_2.

$P_2 =$ protocol over head in DSA_2 (including the queuing delay).

$b_2 =$ time required to perform bind operation.

$qf_1 =$ time required to form the query.

$N_2 =$ time required for name resolution in DSA_2.

$D_2 =$ Time required to perform database operation o in DSA_2.

Thus the total time is given by:

$$T = P_c + C_1 + 3P_1 + N_1 + bf_1 + 2(C_{12} + P_2) + b_2 + qf_1 + N_2 + D_2 + rm_1$$

Now consider a general case with the chaining goes to an arbitrary depth, say n *DSAs*. All but the last *DSA* act as relays to pass the query to the next *DSA* and all but the first *DSA* requires binding. It should be noted that the unbind time will not affect the total response time as unbinding is done after sending the result to the *DUA* or to the previous *DSA* in the chain. Thus the following actions are repeated in each *DSA* except the last one:

- resolve the name,
- bind to next *DSA*,
- chain the operation on the next *DSA* in the chain, and
- merge the result obtained and pass it back to the previous *DSA*.

Thus the total time taken is given by:

$$T = P_c + C_1 + 3P_1 + N_1 + bf_1 + 2C_{12} + qf_1 + \sum_{k=2}^{n}(P_k + b_k) +$$
$$\sum_{k=2}^{n-1}[3P_k + N_k + bf_k + 2C_{k(k+1)} + rm_k] + P_n + N_n + D_n$$

3.8 Case III - Referral

In this case the *DSA* returns a referral to the *DUA*, which in turn uses this referral to contact another *DSA* which can answer the query. The interactions that are taking place are given in Figure 4.

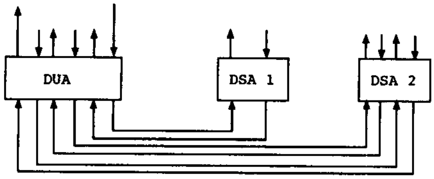


Figure 4: Multiple *DSAs*, Referral mode of interaction

The operation can be split into the following sub-operations:

- The *DUA* sends the request to DSA_1 which returns a referral.
- *DUA* uses these referral and binds to a *DSA* whose address is given in the referral.
- *DUA* sends the operation request to the second *DSA* which services it and returns the result.

Timing for each of these operations are given below.

$T_1 = P_c + C_1 + P_1 + N_1$
$T_{2b} = bf_c + P_c + C_2 + P_2 + b_2$
$T_{2o} = qf_c + P_c + C_2 + P_2 + N_2 + D_2$
Thus the total time T is given by:

$T = 3P_c + C_1 + P_1 + N_1 + qf_c + bf_c + 2(C_2 + P_2) + b_2 + N_2 + D_2$
If the referral extends to *n DSAs*, the total time required is given by:

$$T = (2n-1)P_c + (n-1)bf_c + n * qf_c + \sum_{k=2}^{n}(C_k + P_k + b_k) + \sum_{k=1}^{n}(C_k + P_k + N_k) + D_n$$

Hitherto, we developed a deterministic model to predict T, the time required to perform the directory operations, in terms of the parameters P_c, P_1, P_2, N_1, N_2, C_1, C_2, C_{12}, D_1, D_2, b_1, bf_c, bf_1, qf_c, qf_1 and rm_1. The completion of the model requires estimation of these parameters.

4 PARAMETER ESTIMATION OF THE MODEL

A series of experiments were designed to measure the parameters identified in the section 3. Routines for logging the time are included at various points in both *DUA* and *DSA*. A set of ten experiments are conducted and the average values are taken. The parameters required by the model is estimated from these measured values. Figure 5 shows the architecture of *DSA* and *DUA* in the experimental setup. The shaded area indicates the routines added to measure the timings.

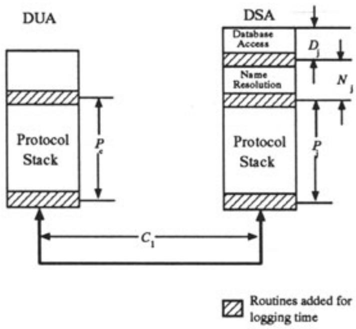


Figure 5: Experimental Setup

In the *DUA*, the time is logged in just before the query is passed onto the protocol stack and once again before the packet is passed to the network. Similarly, in the *DSA*, time is logged when a packet is received from the network, before and after the name resolution and during the database access. That way, database access time and the name resolution time can be calculated. As the clocks in the two systems are not synchronized with each other, it is not possible to measure the oneway communication delay. But by measuring the packet transmission time and packet reception time at both ends, it is possible to measure the round trip communication time.

Experiment Number	DSA_1	DSA_2	DUA	Where reported	Mode
1	IITM	N/A	IITM	Table 2	N/A
2	UMD	N/A	IITM	Table 3	N/A
3	IITM	UMD	IITM	Table 4	Chaining
4	IITM	UMD	IITM	Table 5	Referral

Table 1: Experiments Conducted.

The DIT structure used for the measurements is shown in the Figure 6. Experiments were conducted with different network topologies and different modes of interactions. The experiments conducted are listed in the Table 1.

The database used for the experiments is generated from a subscriber database used by telecommunication department. The entries in the database are duplicated to create bigger databases. An unique identifier is appended so as to make the name of every subscriber unique. This name is then used as the RDN of the entry. Subordinates of a given entry is stored in a separate file and this file is indexed using the B-Tree++ algorithm, with RDN as the key.

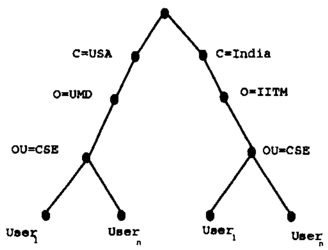


Figure 6: DIT structure

Tables 2 and 3 present the measurements that are taken between three different machines in the internet. Table 2 gives the measurements with *DSA* running in the Sparc machine in IIT, Madras, India and *DUA* running in the same LAN in a Tektronix machine in IIT, Madras, India.

Table 3 shows the measurements with the *DSA* running in the Ultrix machine in University of Maryland, USA, and the *DUA* running in the Sparc machine in IIT, Madras, India. All values given in the tables are the average over a set of ten measurements.

Table 4 shows the measurements when the directory is distributed among two systems in the internet, using *chaining mode* of interaction between the *DSAs*. The *DSAs* were running in the Ultrix machine in University of Maryland, USA and in the Sparc machine in IIT, Madras and *DUA* in the tektronix machine in IIT, Madras. Table 5 shows the measurements done with the same configuration using the *referral mode* of interaction.

From the tables, the following observations can be made:

- The time required to perform the database operation depends on the type of operation. From the tables, it can be seen that the most time consuming directory operation is *modifyrdn* operation. This is because the *modifyrdn* operation requires searching the database twice to perform the operation - once to check the existance of the entry whose modification is requested and then to confirm that no other entry with the same name exists.
- The database operation time and the name resolution time increases with the increase in the number of entries in the directory database. This is because of the increase in the depth of the B-Tree.
- The time required to perform the *list* operation does not depends on the database size. As the number of subordinates of the base object is always larger than the sizelimit parameter, the *list* operation always returns the first few entries and returns the *sizelimit exceeded* error. This is a limitation of the 1988 standard of X.500[CCITT (1988)], which has been overcome in 1993 version[CCITT (1993)].

The estimated values of model parameters are used in predicting the directory operation time T. In all the experiments, the directory operation time T was also measured. Table 6 gives the parameters related to directory operations. The values D_1 is the database operation time and N_1 the name resolution time in the DSA where the entry is located. Table 7 gives the $DSA - DUA$ interaction parameters. The values presented include the bind query formation time, the bind operation time, the query formation time and the result merging time. Table 8 gives the parameters for communication delay and protocol overhead. In the table, C_l is the communication delay in a LAN environment and C_i is the communication delay in an internet environment.

The variations in the predicted value and the actually measured value are computed. The error rate is calculated as *(predicted value - measured value) / predicted value * 100*. The error rate is plotted against database size for all directory operation for all the four experiments in the Figure 8, 9, 10, and 11. The graph given in the Figure 9 shows the variation in the percentage error rate with the number of entries for experiment 2, with no referral or chaining in an internet environment. From this graph, it is evident that the percentage error rate decreases as the number of entries increase. The estimated database access time is more accurate than the round trip communication time. The round trip communication time is considerably smaller when compared to the database access time, when the database size reaches 50,000. Thus the percentage variation in the predicted response time will become smaller with the increase in the number of entries in the directory. This can be predicted accurately. But the round trip communication time across the internet cannot be predicted accurately [Sanghi *et al.*(1993)]. When the database size is small the round trip communication time will be small and thus the percentage variation in the predicted value from the measured value will be high.

Figure 10 shows the percentage variation of error with increase in database size for chaining mode of interaction. The variation in this case is also small for large database size. In our experiments, the *DUA* and one *DSA* was in the same LAN. Thus the communication delay between the *DUA* and the first *DSA* will be very less. Figure 11 in the appendix gives the percentage variations of the error with increase in database size for referral mode of interaction. Again, the variation in the predicted value from the actual measurements decreases with increase in the database size.

5 APPLICATION OF THE MODEL FOR ESTIMATING THE NAME LOOKUP TIME

We now use the deterministic model to predict the expected response time for read operation in a national network. Two different DIT structures are considered and the response time are computed for each structure with the help of the model.

We consider the *ERNET*[4] topology, and estimate the response time using the analytical model discussed in the section 3.

The network topology of *ERNET* is shown in the Figure 7a. *ERNET* consists of 8 different sites, connected by point-to-point links of capacity 9.6 kbps. We estimate the average response time for a query given to any one of the site in the *ERNET* to access the data in any other site. The label on the arcs connecting the sites are the round trip time between the nodes. The actual communication delay between the sites is measured using the `ping` command.

Figure 7b shows the logical tree structure used to estimate the response time for a directory operation. The average response time estimated using the analytical model for the *ERNET* directory for the DIT structure shown in the Figure 7b is shown in the Table 9. Figure 7c shows another logical directory tree structure with the first level *DSA* located at a different site. The response time with this DIT configuration is given in the Table 10. The values given are for a single lookup (read) operation with 500 entries in each *DSA*. The value in column i and row j indicates the time required to access an object in site i by a *DUA* in site j. We assume that *DSAs* stores no cross references to any other *DSAs*, thus the name resolution is done by strict tree traversal.

The Table 9 and 10 show the response time for directory operations with different DIT structures. The readings given are the estimated time for directory read operations. The difference in the readings clearly show the advantage of selecting a proper DIT structure

[4]Education and Research NETwork, setup by Department of Electronics and UNDP.

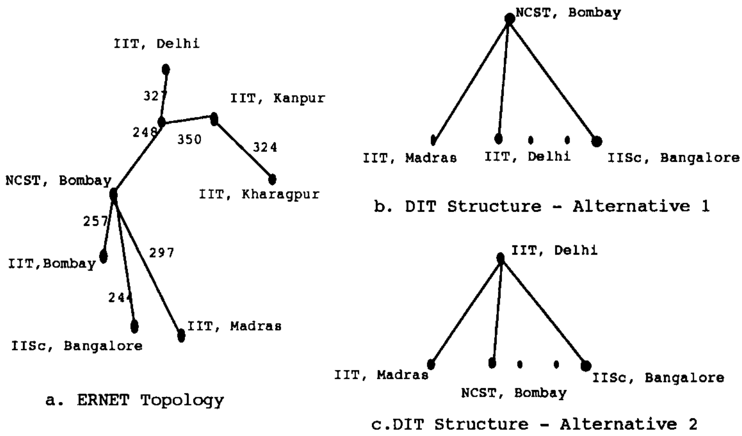


Figure 7: ERNET directory

for a given topology. The directory structure given in Figure 7b is best among the two topologies as it gives the minimum response time.

6 CONCLUSION

In this paper, we presented a simple deterministic model for predicting the performance of the X.500 directory. This model will be helpful for the X.500 directory managers to tune the directory to get a better performance. The model assists the directory manager in estimating the critical database size and the optimal topology to be used. The model is validated by comparing the results predicted using the model and the actual measurements carried out in the internet with a large database. The percentage deviation in the predicted results from the measured results was found to be very less thereby confirming the usefulness of the model. Also, the deviation of the predicted results from the measured value decreases with increase in the database size because the variations in the communication delay, which is an unpredictable factor, will be a small percentage of the total time required to perform the directory operation. The model is used to select the optimal topology for DIT, when the physical interconnection topology of a large national network is given.

References

[CCITT (1988)] CCITT. *Data Communication Networks, Message Handling Systems, Recommendations X.400 - X.420.* 1988.

[CCITT (1988)] CCITT. *The Directory - Overview of Concepts, Models and Services, Recommendations X.500 - X.521.* 1988.

[CCITT (1993)] CCITT. *The Directory - Overview of Concepts, Models and Services, Draft Recommendations X.500 - X.521.* 1993.

[Goodman (1992)] David Goodman. *PARADISE International Report 4.* PARADISE Pilot Group, University College London, November 1992.

[Sanghi *et al.*(1993)] Dheraj Sanghi, Olafur Gudmundsson, Ashok Agarwala. Study of network dynamics. *Computer Networks and ISDN Systems*, 26:371–378, 1993.

[Comer (1979)] Douglas Comer. The Ubiquitous B-trees. *ACM Computing Surveys*, 11(2):121–137, June 1979.

[Huitema (1989)] G. Huitema. The X.500 Directory Services. *Computer Networks and ISDN Systems*, 16:161–166, September 1989.

[Neufeld (1992)] Gerald Neufeld, Barry Brachman, Murray Goldberg, Duncan Stickings. The EAN X.500 Directory Services. *Internetworking : Research and experience*, 3:55–81, 92.

[Mansfield *et al.*(1993)] Glenn Mansfield, K Jayanthi, K Higuchi, Thomas Johannsen, Y Nemoto. Mapping communication networks in the directory. *Computer networks and ISDN Systems*, 26:327–335, 1993.

[Hong *et al.*(1992a)] James W Hong, Micheal A Bauer, J Michael Bennet. Integration of Directory Services in the network management framework. In *Third International Symposium on Integrated Network Management*, April 1992.

[Hong *et al.*(1992b)] James W Hong, Micheal A Bauer, J Micheal Bennet. Integragion of the Directory Services in the Distributed systems management. In *International Conference on Parallel and Distributed Systems*, 1992.

[Barker (1992)] Paul Barker. DIT Structures to support searching for Bibilographic information and information resources. Technical report, University College London, December 1992.

[Barker (1993)] Paul Barker. Implementing FTP archive searching using X.500. Internet Draft : OSI-DS 40, March 1993.

[Bumbulis *et al.*(1993)] P.J. Bumbulis, D.D. Cowan, C.M. Durance, T.M. Stepien. An Introduction to the OSI Directory Services. *Computer Networks and ISDN Systems*, 26:239–249, 1993.

[Sriram (1991)] R Sriram. Studies in OSInet Directory Services. Master's thesis, Indian Institute of Technology Madras, January 1991.

[Sriram *et al.*(1994)] R Sriram, S V Raghavan. Example implementation of Name Services using Directory Services. To appear in Computer Networks and ISDN Systems, 1994.

[Raghavan (1990)] S.V. Raghavan. *Solutions to Local Area Networks, The Indian Context.* Tata McGraw Hill Publishing Company Limited, 1990.

[Rennesse *et al.*] Robbert Van Rennesse, Andrew S Tanenbaum. A Directory service supporting avaliability and consistency. Vrije Universiteit, Amsterdam.

[Raghavan *et al.*(1992)] S. V. Raghavan, R. Sriram. Implementation of X.500 Directory Services in OSInet. *IETE Technical Review*, 9(3), May - June 1992.

[Raghavan (1993a)] S.V. Raghavan. OSInet: Open Systems Interconnection network Software:: Description of its Protocol Architecture, Design, Implementation, Application and Performance. Technical Report ERNET/06/93/IITM, CSE/NSL/08/93/IITM, Indian Institute of Technology, Madras, April 1993.

[Raghavan *et al.*(1993b)] S.V. Raghavan, B. Prabhakaran, K. Shivkumar. An enhanced B tree design for databases. Technical Report ERNET/10/93/IITM CSE/NSL/12/93/IITM, Indian Institute of Technology, Madras, May 1993.

7 AUTHOR INFORMATION

Rajeev P.V. obtained his Bachelor's degree in 1991 from the University of Calicut, India in the field of Computer Science and Engineering. He completed his Master's in Computer Science and Engineering from Indian Institute of Technology, Madras, India. He was working as a research fellow in Networks Systems Laboratory, Department of Computer Science, Indian Institute of Technology, Madras, India. At present, he is with Hughes Software Systems, Delhi. His research interests are networks and multimedia systems.

S.V. Raghavan is on the faculty of the Department of Computer Science and Engineering, Indian Institute of Technology, Madras. He is also the Chief Investigator of the project on *Education and Research in Computer Networking* jointly sponsored by the Department of Electronics, Government of India and the United Nations Development Programme. He is a life member of the Computer Society of India, a member of the Institution of Engineers and a fellow of institute of Electronics and Telecommunication Engineers. He is presently serving on the Board of Editors of the journal of IETE for computers and control. He is also a member of the Editorial Advisory Board for Computer Communications, Butterwoth-Heinemann Ltd. His research interests are networks, protocols, multimedia systems and performance.

Ashok K. Agarwala is a Professor of Computer Science at the University of Maryland, College Park. For the past twenty years he has been actively involved in the research in various aspects of computer system design, implementation and performance. He received his the B.Sc. degree from Agra University in 1960, the B.E degree in Electrical Technology, and the M.E. degree in Applied Electronics and Servomechanisms from Indian Institute of Science in 1963 and 1965, respectively. He earned the A.M. and Ph.D. degrees in Applied Mathematics from Harward University, Cambridge, MA in 1970. Dr. Agrawala is fellow of the IEEE, and a member of the Association for Computing Machinery, American Association for the Advancement of Science Xi.

DSA running in Sparc, IIT Madras and DUA in Tektronix, IIT Madras							
Number of Entries = 5000							
Time	Read	Compare	List	Add	Modify	Remove	ModifyRDN
D_1	531	337	301	1427	1011	210	2203
N_1	1096	1118	781	1219	1137	1060	1256
P_1	33	31	20	31	19	29	26
C_1	38	29	27	38	25	31	38
P_c	18	19	17	29	17	22	17
T	1710	1534	1146	2744	2209	1352	3540
Number of Entries = 10000							
D_1	593	392	1610	1213	232	1932	1496
N_1	1461	1460	798	1432	1472	1503	1496
P_1	34	25	25	22	30	25	27
C_1	38	38	37	32	26	33	29
P_c	25	25	18	18	21	17	27
T	2151	1940	1250	3114	2762	1810	3511
Number of Entries = 25000							
D_1	581	410	363	3793	2478	241	3975
N_1	2262	2145	728	2453	2320	2413	2417
P_1	23	32	32	34	34	31	33
C_1	26	35	37	38	31	36	29
P_c	19	30	20	20	30	18	25
T	2911	2652	1201	6338	4893	2739	6479
Number of Entries = 50000							
D_1	512	409	410	6781	4923	303	7013
N_1	3813	3623	813	3713	3691	3686	3594
P_1	29	25	32	19	34	34	24
C_1	26	30	35	27	33	34	26
P_c	22	18	30	18	25	26	22
T	4402	4105	1320	10558	8706	4083	10679

All values in milliseconds

Table 2: DSA and DUA on LAN.

DUA running in Sparc, IIT, Madras and DSA in Univ. of Maryland							
Number of Entries = 5000							
Time	Read	Compare	List	Add	Modify	Remove	ModifyRDN
D_1	531	320	343	1461	1075	180	2304
N_1	1220	1180	712	1207	1230	1141	1162
P_1	26	29	27	31	27	26	29
C_1	1290	1214	1195	1175	1108	1178	1187
P_c	16	21	18	29	22	23	23
T	3126	2764	2293	3903	3262	2558	4695
Number of Entries = 10000							
D_1	532	393	327	1772	1246	210	2802
N_1	1400	1364	719	1767	1879	1364	1782
P_1	24	27	23	31	29	20	32
C_1	1239	1206	1195	1223	1156	1259	1150
P_c	23	20	27	22	18	23	21
T	3318	3010	2270	5015	4528	2886	5067
Number of Entries = 25000							
D_1	546	403	308	3632	2590	230	3815
N_1	2561	2232	738	2762	2260	2123	2974
P_1	27	23	29	33	29	23	29
C_1	1319	1294	1299	1226	1213	1302	1264
P_c	21	18	21	29	27	19	23
T	4464	3952	2395	7682	6122	3897	7805
Number of Entries = 50000							
D_1	573	420	380	6585	4833	287	6952
N_1	3905	3685	790	3917	3729	3519	3923
P_1	29	23	31	28	26	29	25
C_1	1278	1123	1145	1292	1240	1153	1231
P_c	21	18	27	21	23	22	19
T	5808	5226	2363	11841	9851	5010	12150

All values in milliseconds

Table 3: DSA and DUA Across Internet.

DSAs in Maryland and Sparc, IIT, Madras and DUA in Tektronix Chaining mode of interaction Number of Entries = 5000							
Time	Read	Compare	List	Add	Modify	Remove	ModifyRDN
D_2	618	526	428	1198	1063	230	1839
N_2	1003	1029	721	1102	1121	1017	1482
N_1	307	314	418	472	381	238	325
P_1	26	22	29	30	22	28	24
P_2	20	21	28	21	30	20	24
P_c	19	23	19	20	23	22	20
C_1	34	34	36	26	29	28	31
C_{12}	1162	1227	1241	1181	1286	1280	1143
bf_1	38	32	39	30	31	34	35
qf_1	45	38	35	42	39	45	48
rm_1	50	36	49	46	38	39	30
b_2	122	110	112	142	124	106	146
T	4678	4704	4482	5572	5547	4443	6362
Number of Entries = 10000							
D_2	597	387	441	1780	1312	348	1899
N_2	1492	1480	892	1471	1398	1501	1496
N_1	318	329	393	403	307	321	481
P_1	28	29	30	20	33	20	34
P_2	30	33	26	27	21	33	22
P_c	22	18	24	23	20	18	18
C_1	32	35	35	25	32	31	30
C_{12}	1286	1156	1108	1259	1257	1295	1291
bf_1	33	39	33	33	35	30	35
qf_1	45	49	35	48	49	44	42
rm_1	48	38	54	49	35	43	36
b_2	117	107	143	100	120	114	125
T	5420	4947	4408	6564	5963	5166	6890
Number of Entries = 25000							
D_2	588	476	480	3578	2489	480	3857
N_2	2378	2247	760	2782	2490	2389	2415
N_1	321	370	280	420	338	264	407
P_1	33	32	31	32	22	31	33
P_2	23	32	27	23	28	27	27
P_c	18	19	24	20	19	24	18
C_1	25	38	37	34	36	37	30
C_{12}	1304	1212	1162	1257	1222	1162	1264
bf_1	38	30	37	39	34	37	33
qf_1	37	39	47	36	43	47	38
rm_1	53	52	51	32	36	51	33
b_2	115	122	106	107	114	106	133
T	6326	5977	4293	9704	8165	5906	9645
Number of Entries = 50000							
D_2	621	508	405	6817	4293	421	6983
N_2	3873	3692	821	3985	3961	3796	3975
N_1	298	264	269	347	469	256	442
P_1	20	28	31	32	22	34	22
P_2	26	33	20	31	33	24	26
P_c	21	22	24	23	21	19	21
C_1	30	36	27	26	38	27	26
C_{12}	1142	1198	1313	1235	1323	1243	1248
bf_1	37	30	31	30	37	30	31
qf_1	38	35	46	41	35	48	36
rm_1	50	40	44	40	48	35	52
b_2	138	135	147	124	117	108	128
T	7502	7308	4573	14061	11797	7376	14308

All values in milliseconds

Table 4: DSAs Across Internet with Chaining + DUA.

DSAs in Maryland and Sparc, IIT, Madras and DUA in Tektronix Referral mode of interaction Number of Entries = 5000							
Time	Read	Compare	List	Add	Modify	Remove	ModifyRDN
D_2	639	570	400	1247	1036	242	1793
N_2	1027	1001	658	1051	1150	1073	1407
N_1	329	360	394	463	414	217	334
P_1	25	34	32	34	26	23	23
P_2	34	27	33	33	23	21	23
P_c	20	22	18	22	24	23	22
C_1	38	33	25	37	30	33	30
C_2	1228	1222	1233	1154	1136	1164	1194
bf_c	32	38	34	32	39	39	38
qf_c	43	36	41	48	46	40	38
b_2	115	117	138	116	138	138	112
T	4832	4753	4308	5468	5269	4244	6275
Number of Entries = 10000							
D_2	607	384	448	1785	1296	299	1850
N_2	1430	1512	918	1445	1391	1483	1510
N_1	284	369	396	432	349	349	496
P_1	25	32	31	22	33	21	24
P_2	32	33	29	29	24	29	30
P_c	22	23	21	23	24	23	18
C_1	39	37	26	25	38	28	31
C_2	1223	1281	1154	1149	1167	1120	1130
bf_c	33	32	38	36	36	32	32
qf_c	38	41	49	47	38	35	44
b_2	100	133	135	141	134	139	104
T	5132	5237	4470	6358	5769	4753	6465
Number of Entries = 25000							
D_2	591	449	444	3533	2486	494	3836
N_2	2336	2211	726	2711	2465	2348	2417
N_1	303	407	302	383	332	222	372
P_1	25	34	21	25	23	21	32
P_2	23	23	26	24	26	34	26
P_c	20	19	19	24	20	23	22
C_1	25	26	33	25	35	25	37
C_2	1114	1343	1100	1165	1157	1211	1134
bf_c	35	36	34	37	33	36	37
qf_c	48	38	40	44	42	39	35
b_2	128	114	145	113	111	117	120
T	5825	6104	4054	9321	7953	5861	9272
Number of Entries = 50000							
D_2	670	499	386	6816	4265	432	6960
N_2	3918	3624	760	4025	3920	3737	3998
N_1	264	302	244	361	483	271	461
P_1	21	23	30	32	30	27	25
P_2	23	30	32	25	29	33	30
P_c	22	21	23	24	18	19	24
C_1	26	31	29	39	31	30	27
C_2	1285	1114	1282	1240	1289	1127	1243
bf_c	31	30	34	35	38	38	39
qf_c	37	46	47	39	42	35	45
b_2	104	116	146	145	104	116	116
T	7753	7022	4373	14094	11603	7063	14289

All values in milliseconds

Table 5: DSAs Across Internet with Referral + DUA.

Variable Estimated	Read	Compare	Add	Modify	Remove	ModifyRDN
Number of entries = 5000						
D_1	531	320	1500	1075	200	2300
N_1	1220	1180	1200	1230	1140	1160
Number of entries = 10,000						
D_1	532	393	1772	1246	210	2802
N_1	1400	1364	1767	1879	1364	1782
Number of entries = 25,000						
D_1	546	403	3632	2590	230	3815
N_1	2561	2232	2762	2260	2123	2874
Number of entries = 50,000						
D_1	573	420	6585	4833	287	6952
N_1	3905	3685	3917	3729	3519	3923

Table 6: Estimated Values of Model Parameters Related to Directory Operations.

Estimated Variable	Value
Time to search knowledge information (N_1)	550
Bind request formation time (bf_1)	40
Bind time (b_i)	130
Query Formation time (qf_1)	45
Result merging time (rm_1)	45

Table 7: Estimated Values of *DSA* - *DUA* interaction parameters.

Estimated Variable	Minimum Value	Maximum Value	Mean	Standard Deviation
P_1	19	34	27	0.427
P_2	20	34	26	0.588
P_c	16	30	21	0.380
C_l	25	39	30	0.860
C_i	1100	1343	1198	7.020

Table 8: Estimated Values of Protocol Overhead and Communication Delay Parameters.

	ncst	iitm	iitb	iisc	doe	iitkgp	iitk	iitd
ncst	1667	3282	3090	3086	3109	4759	3880	3850
iitm	3230	1607	4244	4280	4311	5511	5204	5154
iitb	3174	4336	1549	4152	4199	5685	5146	5054
iisc	3084	4304	4098	1458	4205	5657	5198	5144
doe	3159	4330	4088	4180	1499	4665	4062	3958
iitkgp	4723	6928	7048	7022	6109	1582	5496	6774
iitk	3908	5964	5850	5986	4887	4861	1498	5574
iitd	3763	5792	5666	5810	4641	6005	5446	1542

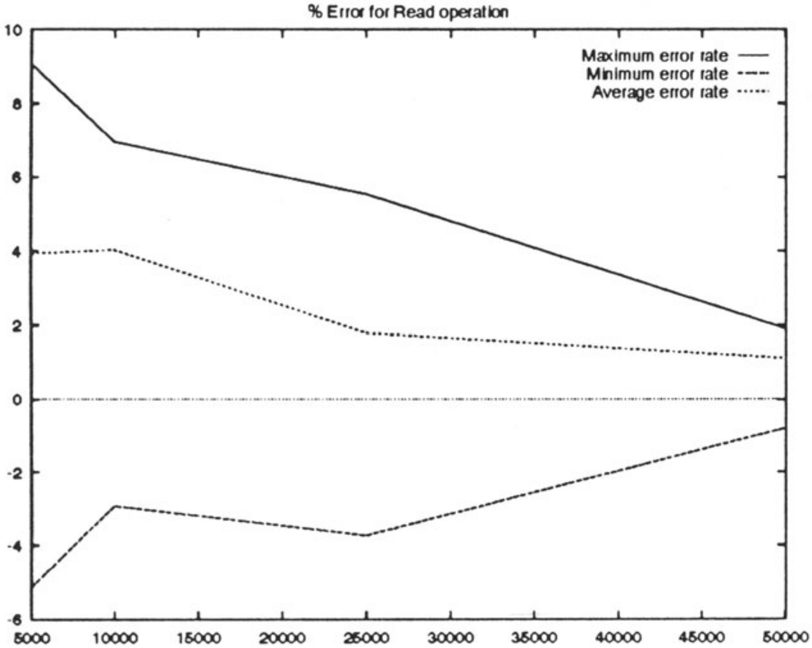
All values in milliseconds.

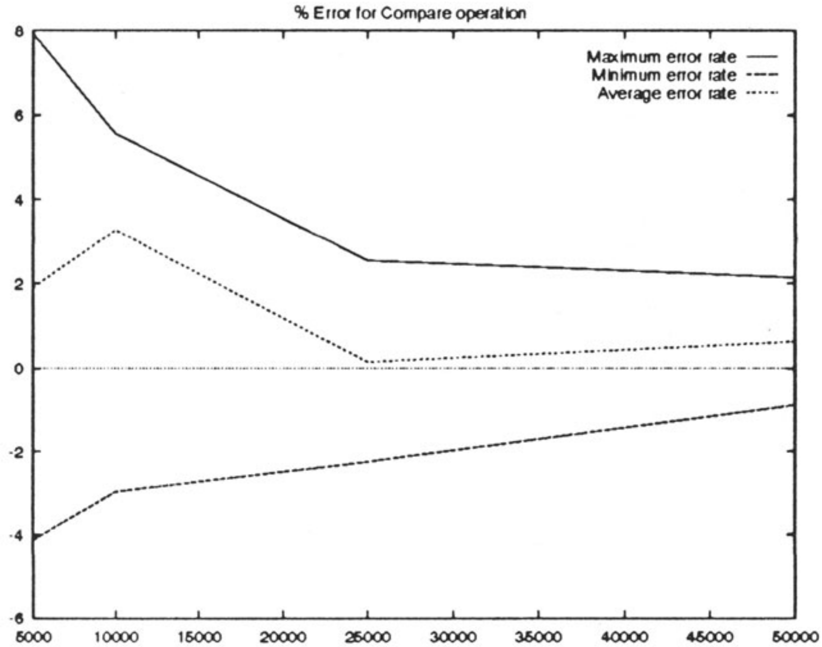
Table 9: Predicted Read Operation Timing in DIT Alternative 1.

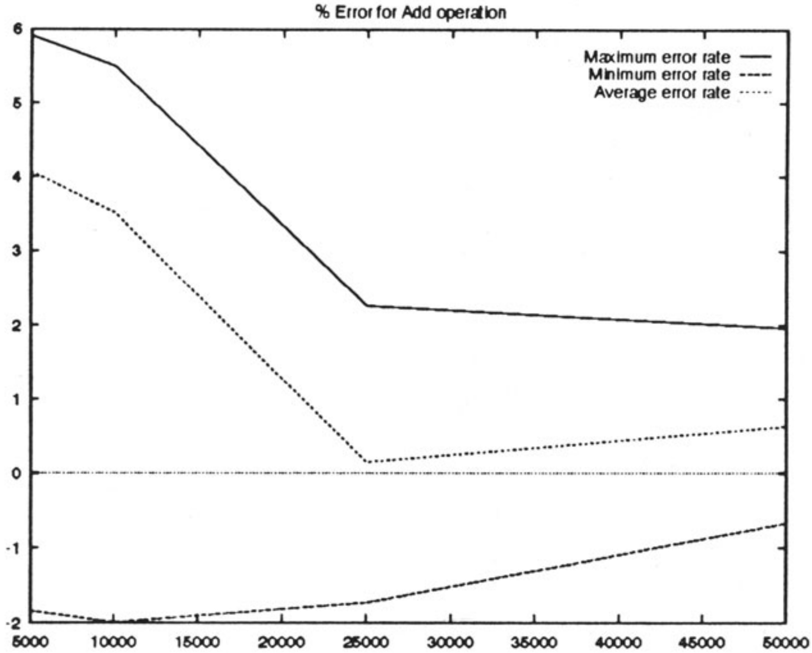
	ncst	iitm	iitb	iisc	doe	iitkgp	iitk	iitd
ncst	1667	4534	4342	4338	4361	6011	5132	3850
iitm	5089	1607	5180	5216	5247	6447	6140	4231
iitb	4989	5300	1549	5116	5163	6649	6110	4203
iisc	5079	5434	5228	1458	5335	6787	6328	4279
doe	3893	4258	4016	4108	1499	4593	3990	3152
iitkgp	6709	6620	6740	6714	5801	1582	5188	4480
iitk	5509	5948	5834	5970	4871	4845	1498	3957
iitd	3763	4182	4056	4200	3031	4395	3836	1542

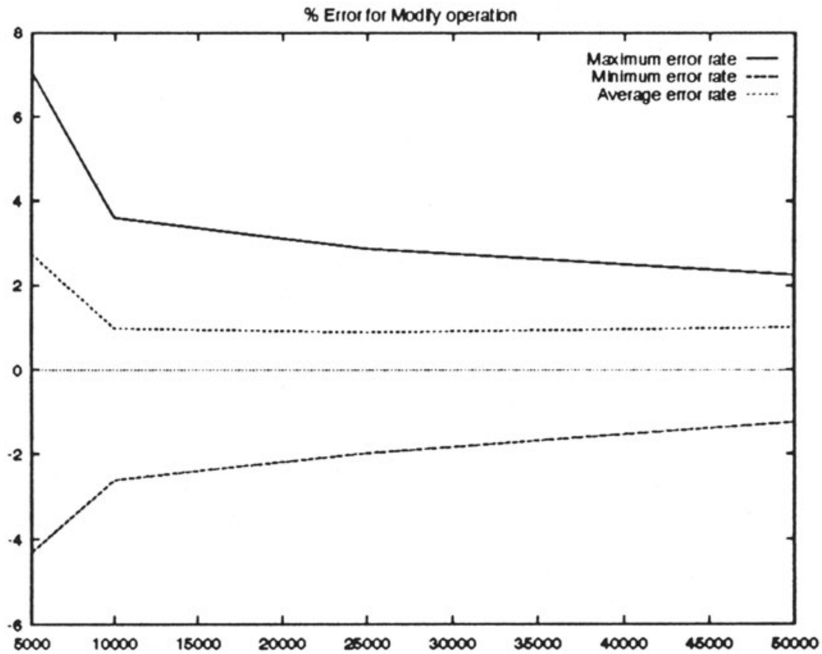
All values in milliseconds.

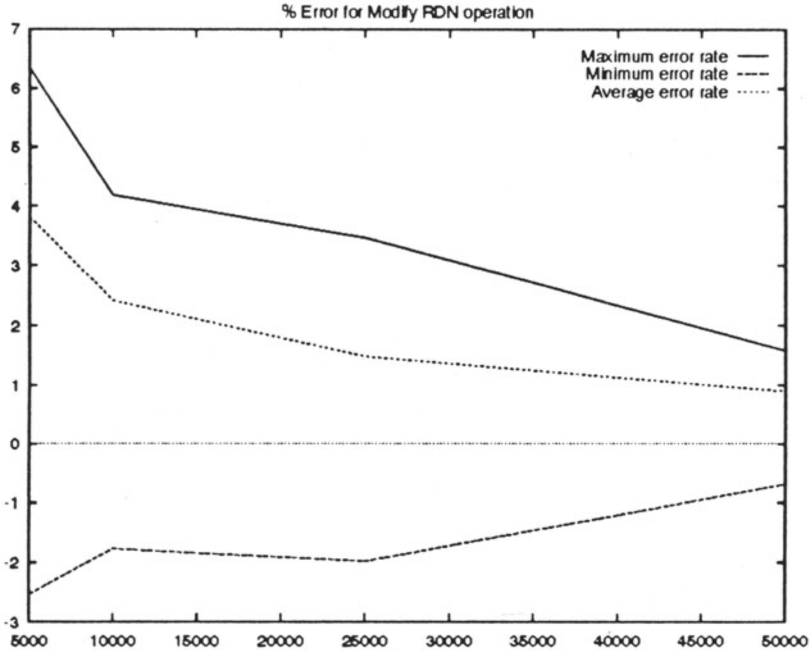
Table 10: Predicted Read Operation Timings in DIT Alternative 2.


% Error for Read operation
Maximum error rate
Minimum error rate
Average error rate
10
8
6
4
2
0
-2
-4
-6
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000


% Error for Compare operation
Maximum error rate
Minimum error rate
Average error rate
8
6
4
2
0
-2
-4
-6
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000


% Error for Add operation
Maximum error rate
Minimum error rate
Average error rate
6
5
4
3
2
1
0
-1
-2
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000


% Error for Modify operation
Maximum error rate
Minimum error rate
Average error rate
8
6
4
2
0
-2
-4
-6
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000



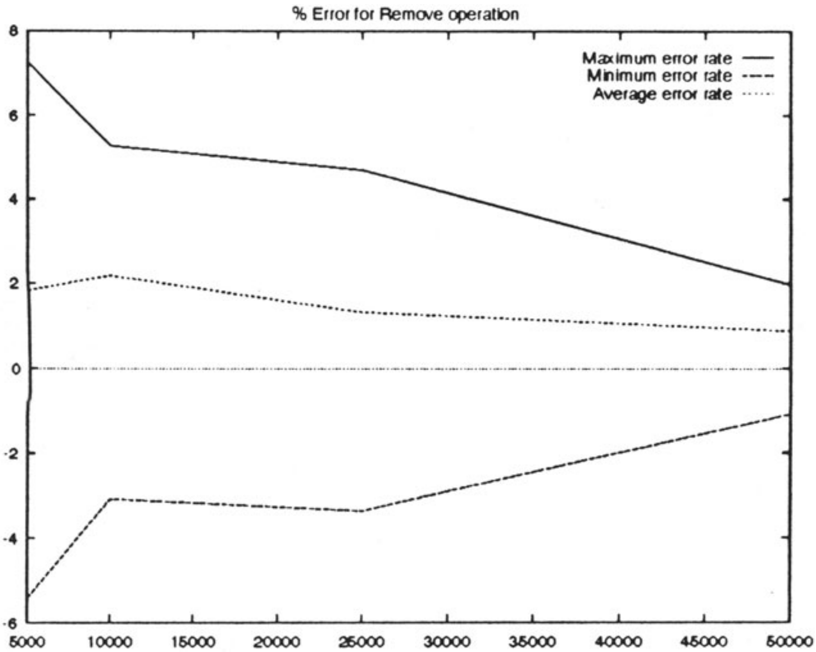
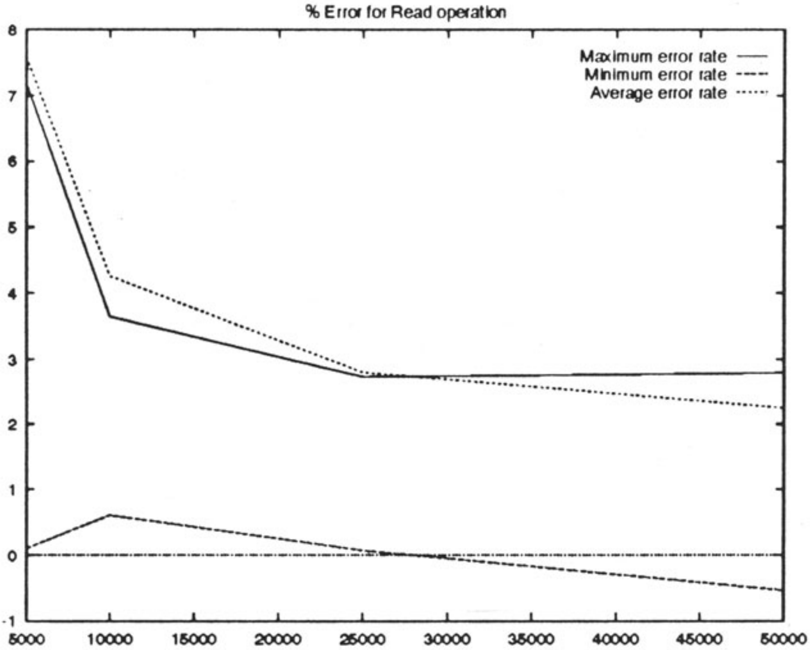
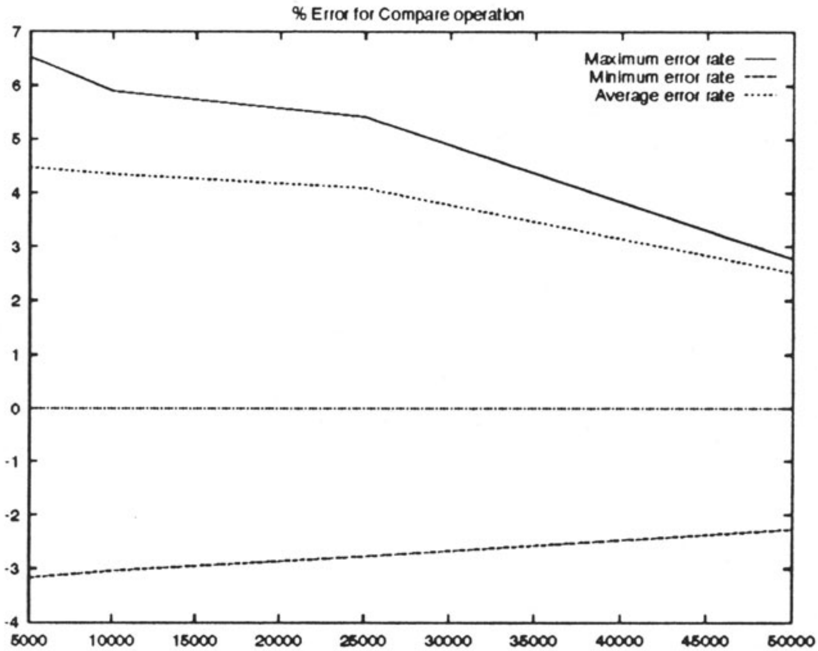
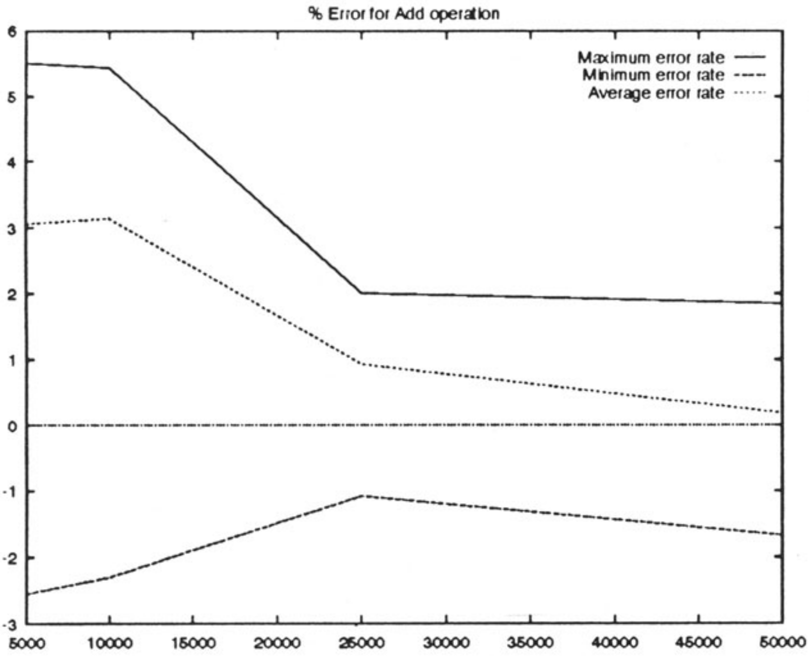
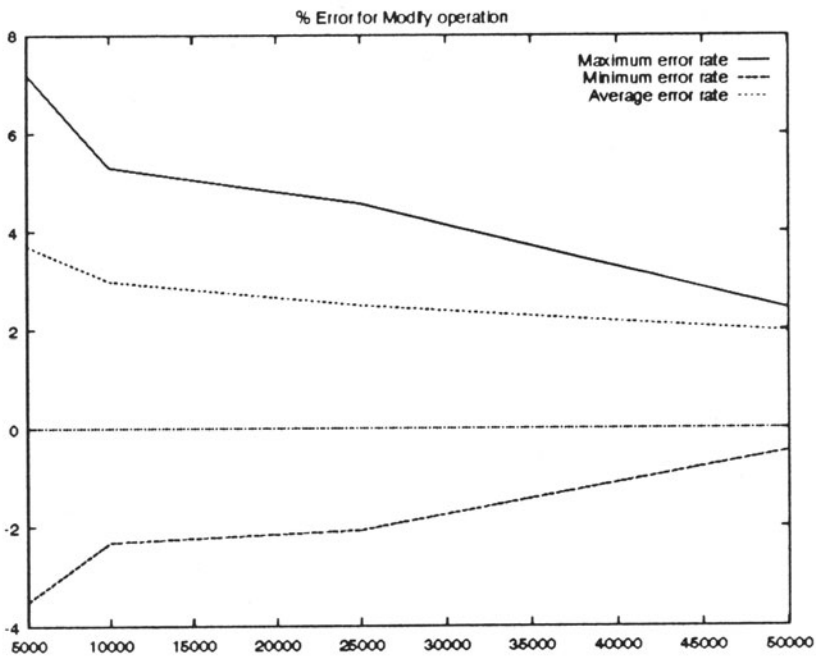


Figure 8: Percentage Error rate Vs Database size, Experiment 1.


% Error for Read operation
Maximum error rate
Minimum error rate
Average error rate
8
7
6
5
4
3
2
1
0
-1
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000


% Error for Compare operation
Maximum error rate
Minimum error rate
Average error rate
7
6
5
4
3
2
1
0
-1
-2
-3
-4
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000


% Error for Add operation
Maximum error rate
Minimum error rate
Average error rate
6
5
4
3
2
1
0
-1
-2
-3
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000


% Error for Modify operation
Maximum error rate
Minimum error rate
Average error rate
8
6
4
2
0
-2
-4
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000

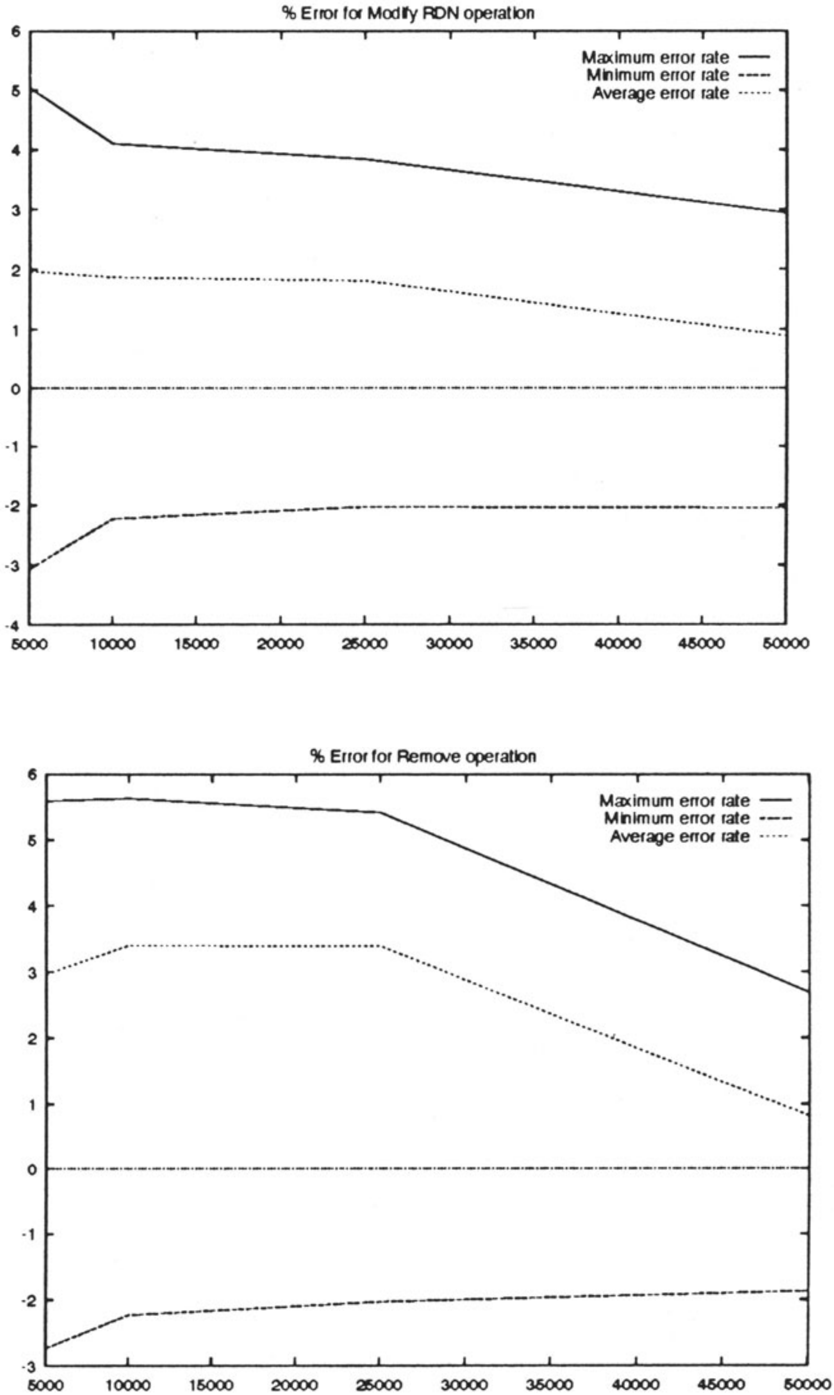
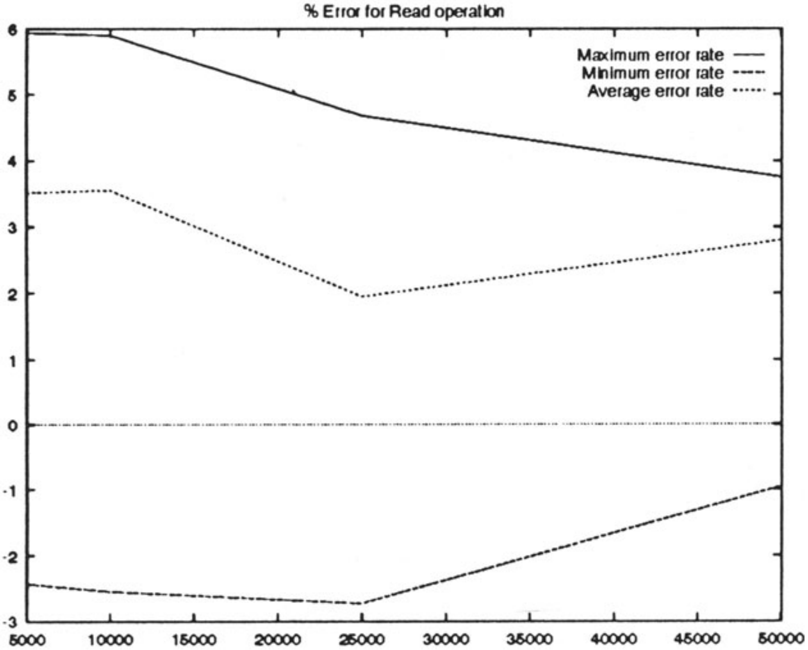
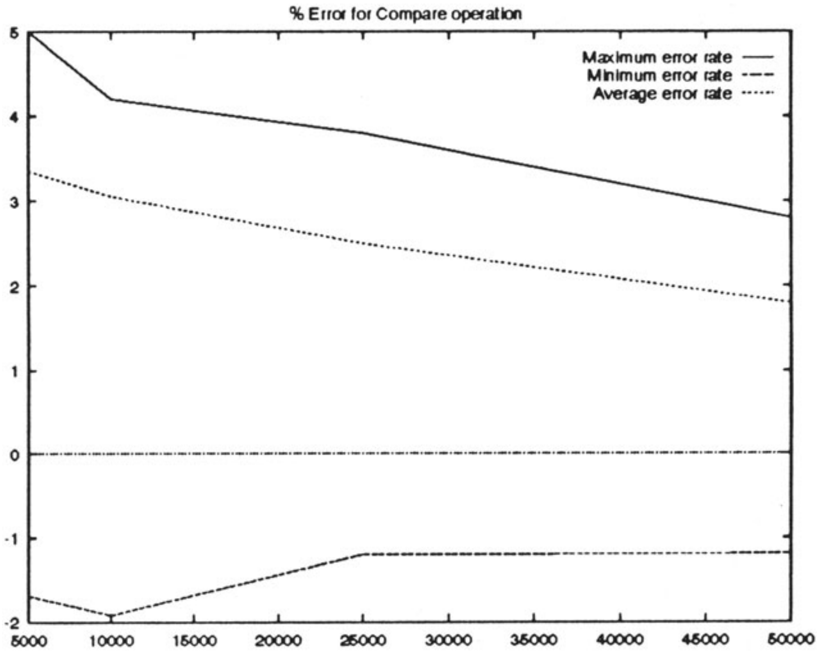
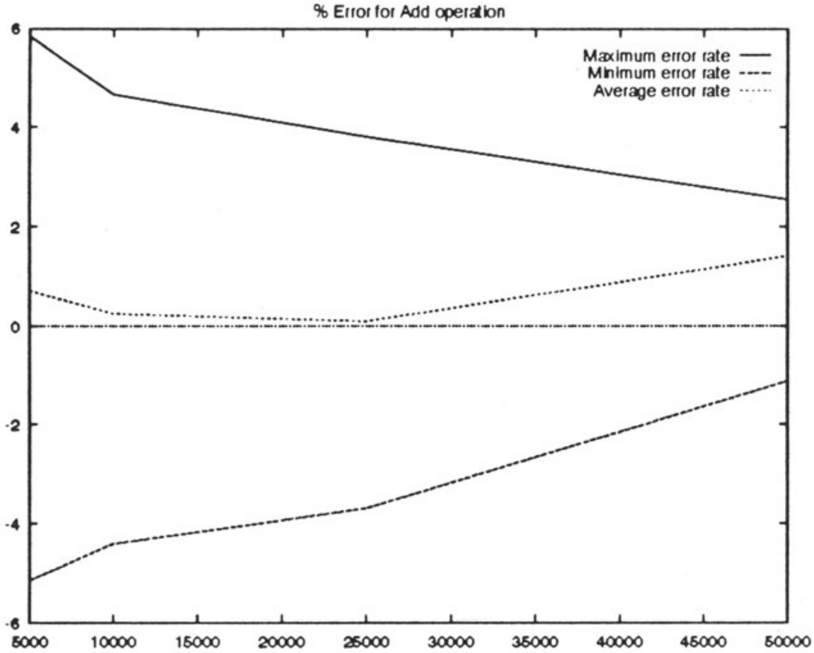
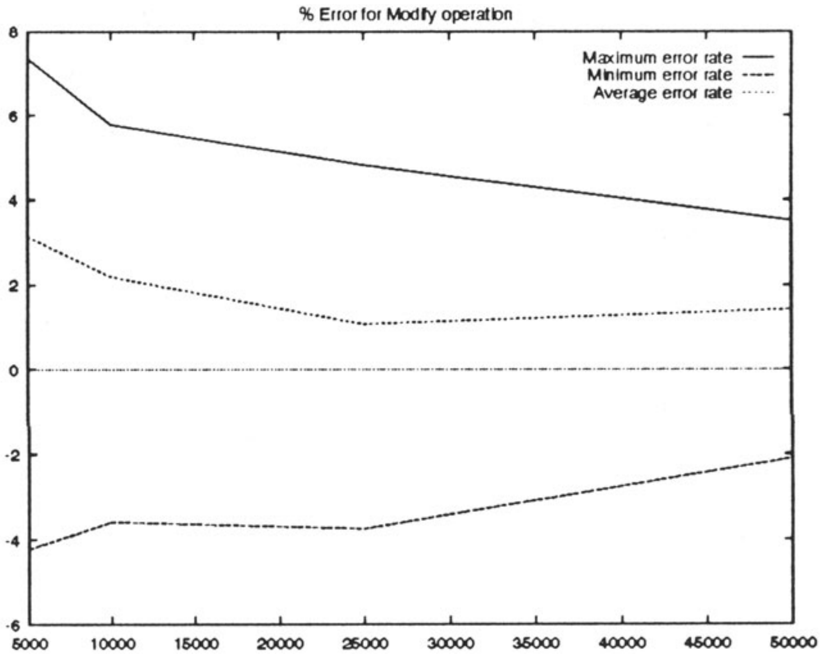


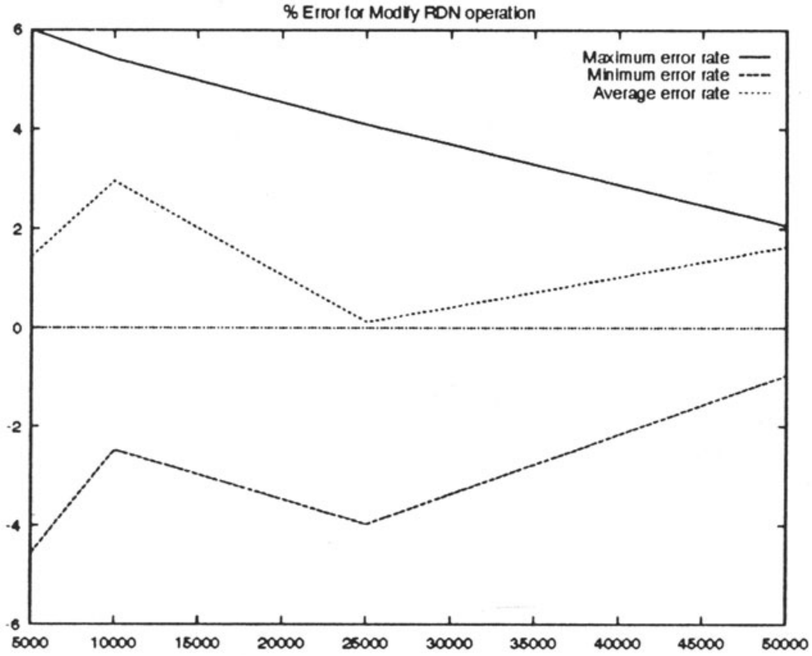
Figure 9: Percentage Error rate Vs Database size, Experiment 2.


% Error for Read operation
Maximum error rate
Minimum error rate
Average error rate
6
5
4
3
2
1
0
-1
-2
-3
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000


% Error for Compare operation
Maximum error rate
Minimum error rate
Average error rate
5
4
3
2
1
0
-1
-2
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000


% Error for Add operation
Maximum error rate
Minimum error rate
Average error rate
6
4
2
0
-2
-4
-6
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000


% Error for Modify operation
Maximum error rate
Minimum error rate
Average error rate
8
6
4
2
0
-2
-4
-6
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000



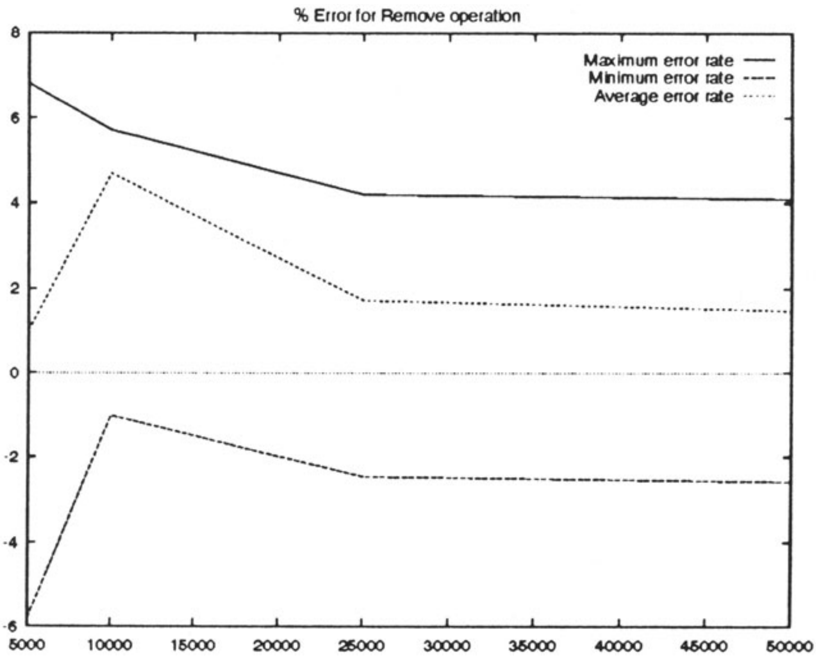
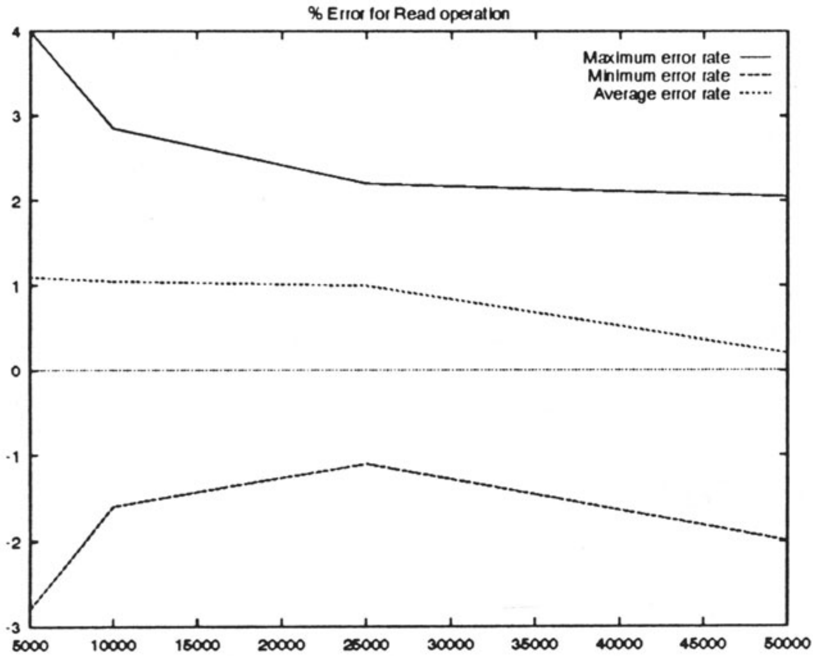
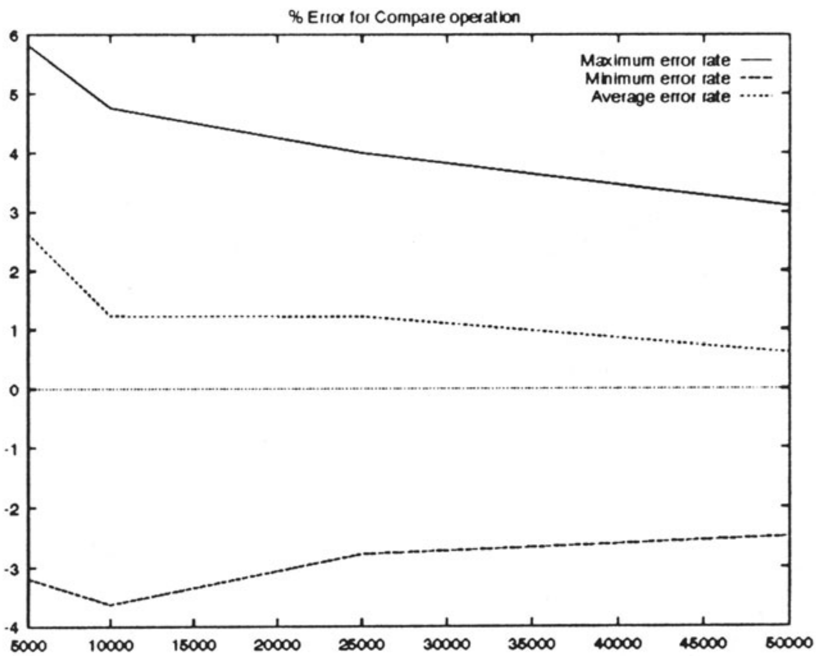
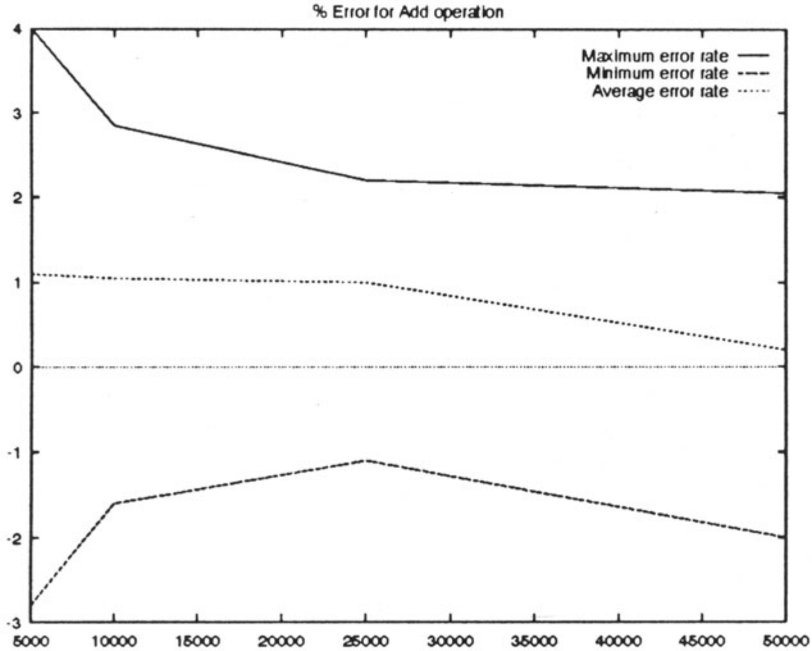
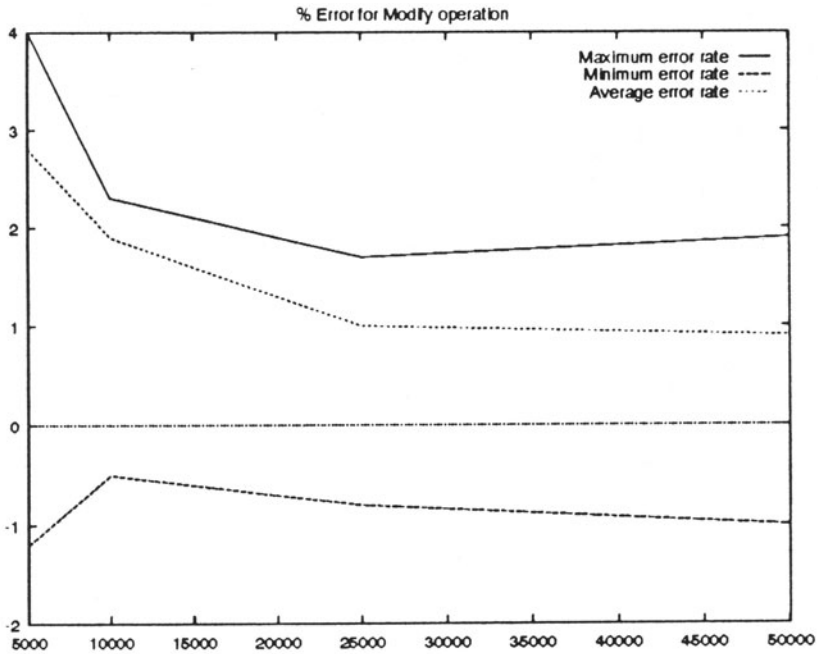


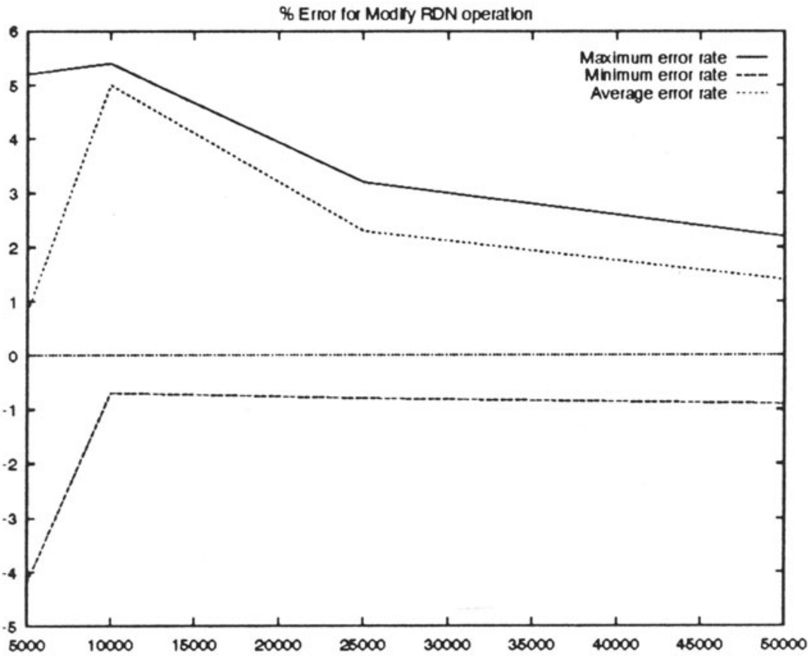
Figure 10: Percentage Error rate Vs Database size, Experiment 3.


% Error for Read operation
Maximum error rate
Minimum error rate
Average error rate
4
3
2
1
0
-1
-2
-3
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000


% Error for Compare operation
Maximum error rate
Minimum error rate
Average error rate
6
5
4
3
2
1
0
-1
-2
-3
-4
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000


% Error for Add operation
Maximum error rate
Minimum error rate
Average error rate
4
3
2
1
0
-1
-2
-3
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000


% Error for Modify operation
Maximum error rate
Minimum error rate
Average error rate
4
3
2
1
0
-1
-2
5000
10000
15000
20000
25000
30000
35000
40000
45000
50000



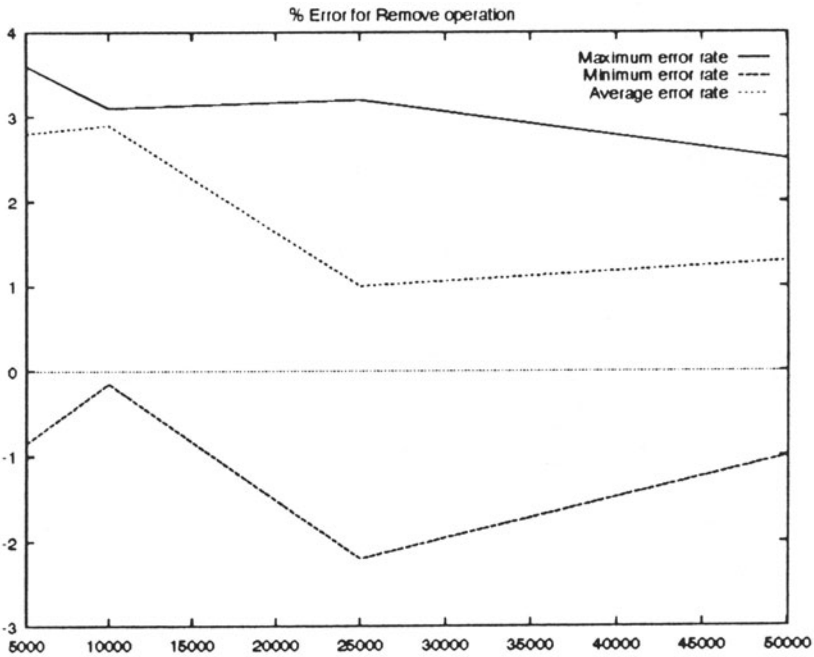


Figure 11: Percentage Error rate Vs Database size, Experiment 4.