A Fault-Tolerant Mobile Computing Model Based On Scalable Replica

Meenakshi Sati[1],Vivek Vikash[2], Vishwanath Bijalwan[3] , Pinki Kumari[4],Manish Raj[5], Meenu Balodhi[6], Priya Gairola[7], Vijay Bhaskar Semwal[8]

[1 ,2,3] *NITTR Chandigrah*
[4] *Bansatali Vidya Peeth rajstan*
[5]*IIIT Allahabad*
[6,7] *Uttarakhand Techincal Univesity*
[8] *Siemens Information System, Banglore*

Abstract **— The most frequent challenge faced by mobile user is stay connected with online data, while disconnected or poorly connected store the replica of critical data. Nomadic users require replication to store copies of critical data on their mobile machines. Existing replication services do not provide all classes of mobile users with the capabilities they require, which include: the ability for direct synchronization between any two replicas, support for large numbers of replicas, and detailed control over what files reside on their local (mobile) replica. Existing peer-to-peer solutions would enable direct communication, but suffers from dramatic scaling problems in the number of replicas, limiting the number of overall users and impacting performance. Roam is a replication system designed to satisfy the requirements of the mobile user. Roam is based on the Ward Model, replication architecture for mobile environments. Using the Ward Model and new distributed algorithms, Roam provides a scalable replication solution for the mobile user. We describe the motivation, design, and implementation of Roam and report its performance. Replication is extremely important in mobile environments because nomadic users require local copies of important data.**

Keywords **— Fault-tolerant, Mobile Computing, Roam, Data Replictaion.**

I. INTRODUCTION

REPLICATION in mobile environments requires fundamentally different solutions than those previously proposed, because nomadicity presents a fundamentally new and different computing paradigm. Before mobile computing was a feasible reality, software designers simply didn't design their systems to be mobile-enabled. Computers were largely stationary, so there was no need to consider what would happen if it were easy for people to move around geographically. Consider the case of mobile-IP. When the Internet Protocol (IP) was developed, mobility was not feasible, and therefore was not considered in the design. Now that mobility *is* a reality, many researchers are trying to fix or redesign IP to support real mobility. Like the mobile-IP case, the replication system software must also be redesigned to account for mobile computing. *Roam* is built using the Ward Model, a new replication architecture that has been designed especially with mobility in mind, though of course it applies equally well to stationary environments [1]. The Ward Model provides a new replication paradigm that is neither strictly peer nor client-server; rather, it is a hybrid model of the two that allows everyone to be effectively a peer while maintaining good scalability in the number of replicas. Additionally, *Roam* provides a number of new distributed algorithms. For instance, *Roam* contains new and improved distributed algorithms for garbage collection and for dynamic management of the version vector, the main data structure behind most optimistic replication systems.

In this paper we present an approach for data replication that considers the mobility of clients: The data sets which have to be replicated to mobile clients depend often on dynamic parameters like location of the client or time. Therefore, we introduce a generic model for specifying fragments of the global database with respect to such parameters. The increasing availability of mobile devices and wireless communication technologies opens new applications which can improve services by providing real-time access to remote information sources. Examples of such applications are not only stock tickers or messenger applications but also mobile hospital information systems and car navigation systems. However, due to the mobile and wireless nature of these devices, several limitations have to be taken into account in developing and running such mobile information systems:

(1) The structure of the network as well as the availability of mobile nodes are subject to continuous changes.

(2) The bandwidth of the communication channels is – compared to traditional wired networks – relatively low and communication is much more expensive.

(3) The resources of mobile devices, e.g. memory and computing power, are strictly limited. For a mobile information system, the first two properties prohibit data access by only remote querying.

Thus, the mobile device should keep a copy of the data that is the subject of the current operations. The third issue restricts the data managed locally on the mobile device to a subset of the overall (remote) database. This raises the need for *semantic replication*: data that is relevant in a certain

DOI: 10.9781/ijimai.2014.267

situation has to be transferred to the mobile device and in case of local updates the modifications are propagated back to the main database. A main issue in this context is the reduction of the replicated data in order to meet the requirements of minimal resource consumption (storage space, transfer time or volume). A possible application is an advanced car driver information system. In addition to street maps stored on a local disk the mobile system retrieves further information about local (i.e., relevant to the current location and time) traffic spots, accidents, detours etc. from a federated database integrating these information from various sources and displays it to the user or uses it for other operations, e.g. route planning. When this information becomes irrelevant, it can be removed from the local database. This scenario could be extended to support updates – a driver reports an accident or the system detects a traffic jam automatically – as well as other kinds of parameters, for example the kind of car (truck, automobile). In this paper we present a replication approach which is based on semantic extensions of the global view. Using this approach, location-based services and services requiring replication depending on dynamic features can be easily (on the global view level) implemented. It is embedded into the context of mobility. While looking at related work we cannot consider all requirements caused by the special characteristics of mobile computing. Therefore, we focus on work related to replication in mobile information systems.

Here we present a replication system designed to satisfy the requirements of the mobile user. Roam is based on the Ward Model, a replication architecture specifically aimed at mobile environments. Together with the Ward Model and new, distributed algorithms such as improved garbage collection techniques and version vector management, Roam provides a scalable replication solution for the mobile user. We describe not only the design and motivation of such a system, but its implementation and performance as well.

Roam addresses scalability with a three-pronged attack. The Ward Model addresses issues of replica management, consistency topologies, and update distribution. Dynamic algorithms and mechanisms handle the scalability of the versioning information, required for consistency maintenance. Finally, we address consistency itself with new algorithms and mobile-friendly semantics. In summary, Roam is a comprehensive replication system for mobile and non-mobile users alike. With it, users can truly compute while mobile, paving the way for both improved user productivity and new and unseen research along mobile computing avenues. The particular point in the solution space is one which we believe addresses a real problem and has wide applicability, not just for replicating files in a mobile context but more generally for a wide range of replication-related problems, including but not limited to military cases, software development scenarios, distributed database problems like airline reservation systems, and general-purpose distributed computing.

The rest of paper is organized as follows: section 2 reviews the related work; section 3 discuss the basic ward model; section 4 presents the advanced ward model; section 5 shows the architectural design; section 6 presents the results; and section 7 conclude the paper.

II. A Brief Review of Related Work

Computing in the 1970s and 1980s meant using almost completely stationary machines. Designed to be situated in one location and rarely if ever moved, these machines were heavy, awkward, and clumsy. The resulting software architectures were adequate for the time, because these machines were too clumsy to move on any regular basis. Even the emerging “portable" machines at the end of the 1980s were not really portable; they could at best be described as \luggable." However, by the 1990s the hardware industry had made rapid progress in chip and LCD technology, as well as in general miniaturization. Machines emerged that were truly portable and as powerful as their stationary cousins. Making use of this new type of machine with its attendant portability, users became increasingly mobile. In 1996, approximately one third of the computers sold were mobile-enabled: that is, portable form-factor with communications capability.

Coda [2] is an optimistically replicated file system primarily using the client-server model. Coda provides replication flexibility akin to selective replication at the clients, but not at the replicated servers. The servers run a form of peer replication. Coda clients cannot directly inter-communicate due to restrictions of the client-server model. Use of this model dramatically simplifies the consistency algorithms, at the cost of limiting the system's utility for mobility. Coda is clearly superior in the low-bandwidth scenario, having greatly optimized communications and synchronization, especially in environments with weak connectivity [3]. Some of the same ideas could be applied in Roam, though additional research would be required to incorporate them into a peer model.

The Little Work project [4] is similar to Coda, but modifies only the clients, leaving the AFS [5] servers unaltered. Congestion caused by clients' slow links is reduced in a variety of ways, including client-side modifications of AFS, Little Work's underlying RPC, and other congestion avoidance and control methods. However, clients cannot directly communicate, hindering the usability of the system in dynamic, mobile environments.

The Bayou system [6] replicates databases rather than file system objects. Like Roam, it uses the peer-to-peer model. Unlike Roam, Bayou does not attempt to provide transparent conflict detection. Applications must specify a condition that determines when a conflicting access has been made, and must specify the particular resolution process.

Ficus [7] is one of the intellectual predecessors of Roam, and Roam therefore shares many of its characteristics. Both are based on a peer model, though Roam's Ward Model scales better than the Ficus traditional peer model. Both provide selective replication control. While each maintains consistency with a periodic reconciliation process, Ficus also uses best-effort propagation at update time. Ficus is aimed at a distributed Internet environment, and works well for its target. However, it is unsuitable for mobile use, and does not scale well.

Rumor [8] is the direct predecessor of Roam; much of Roam's implementation is directly based on modified Rumor code. Rumor is in turn a descendent of Ficus, and shares many

of its characteristics and problems. It is based on the traditional peer model, and relies upon periodic reconciliation to maintain consistency. While Rumor is better suited to a mobile environment than Ficus, its scaling properties are substantially the same.

An example of remote computing is illustrated in [9] applied to the geometry over internet. Resources such as executables, languages, packages, can be used from a remote computing system. They implemented a distributed system using geometry that outsources the computing-intensive tasks to remote servers that may be located in other universities or companies, linked to grids and clusters and so on. The architecture developed stresses the interoperability of the software, and a suitable high degree of decoupling between components hosted in various locations.

The definition of a level oriented business process methodology is enhanced in [10], which encourages the adaptation of the modeling notation to the modeling and technical knowledge shown by the expert. Their approach reduces the complexity found by domain experts and enables them to model their processes completely with a level of technical detail directly proportional to their knowledge. They also generate the codes on mobile plateform.

A simple client-server system architecture and algorithms is introduced in [11] for ubiquitous live video and VOD service support. The main features of the system are: efficient usage of network resources, emphasis on user personalization, and ease of implementation. The system supports many continuous service requirements such as QoS provision, user mobility between networks and between different communication devices, and simultaneous usage of a device by a number of users.

A framework for computation offloading is proposed in [12] for mobile cloud computing. Offloading of computationally intensive application parts from the mobile platform into a remote cloud infrastructure or nearby idle computers addresses this problem. They presented Mobile Augmentation Cloud Services (MACS) middleware which enables adaptive extension of Android application execution from a mobile client into the cloud.

Clustering and regression based technique is introduced in [13] for analyzing anonymized cellular network data to identify generally important locations, and to discern semantically meaningful locations such as home and work. They test it on arbitrary cellphone users, including those with low call rates.

Traditional client-server architectures are not capable of supporting the desired infrastructure and providing the required communication capabilities among replicas. While such functionality could have potentially been added to the client-server model, we expected that modifying a fundamentally client-server design to incorporate a rich communications structure would require such basic structural changes as to imply effectively starting from scratch. The goal is therefore to design a peer-based model that scales well. Replication services based on the traditional peer model, such as Rumar [8], AFS [5], and Ficus [14], all suffer from scaling problems. In response, we have designed the Ward Model. It provides a new and different form of the peer model, one based on a hybrid between client-server and peer solutions that clusters replicas into groups without affecting the underlying any-to-any communication capability between all system participants. Ward based model can also be integrated in other domains such as in fruit diseases [15], human activity recognition [16], PCA approaches [17], 3D model [18], semantic information [19], wireless sensor network [20, 21] and learning techniques [22].

III. BASIC WARD MODEL

We describe the basic Ward Model by defining wards, and ward sets. We then explain how the model maintains consistency and how the model supports mobility.

The key idea behind wards is to group volume replicas into containers that capture the notion of common or typical communication partners. For example, given four replicas in Los Angeles and four replicas in New York, the system would perform poorly if each replica in Los Angeles typically synchronized with a replica in New York. While the topology produces correct results, each machine pays the additional cost of long- distance communication, in terms of latency, efficiency, and price per byte. Additionally, at synchronization time, there is a greater chance of a failed network connection between Los Angeles and New York than between two local partners in either location, given that generally the long distance communication depends on having the local communication operational. The obvious improved approach is to have one candidate in Los Angeles communicate with one candidate in New York; these two candidates afterwards disseminate the information among their local colleagues.

Given that synchronization should typically occur with a local partner, both for economic and efficiency reasons, we would like to group replicas together to capture the notion of synchronization locality. We therefore build wards as a collection of “nearby" volume replicas, the details of which are discussed below. The ward members are only required to be loosely connected-continual, high-quality connectivity is not necessary.

Ward set

The ward set refers to the set of replicated data stored within the ward. In the basic Ward Model, the ward set is, by definition, equivalent to the entire volume, although the definition changes in the advanced model. Like the volume itself, the ward set is dynamic in character; it changes as the volume itself changes in response to new object creations and existing object deletions. It is the ward master's responsibility to synchronize its ward set with the ward sets from other wards. In the basic Ward Model, ward-set synchronization can be accomplished simply by contacting one other ward master: since all wards store the entire volume, all ward sets are identical. In the advanced model a more complex synchronization architecture will be utilized.

Maintenance of consistency

Consistency is maintained simultaneously both within each ward and among wards. In both intra- and inter-ward scenarios, the consistency topology refers to the communication pattern used between replicas. All of our consistency algorithms are topology-independent. For example, a quadratic message complexity results from an all-pairs reconciliation topology, but a ring (using a gossip-based transfer of information) [20-25] reduces the message complexity to a linear cost. A superior messaging plan avoids the quadratic cost when inter-site communication is available, but gracefully handles degraded communication. Using two separate topologies within the ward and between wards reduces the generality of the model and increases the requirement for special-case code. Although one could potentially identify hypothetical reasons why two topologies would be required, a topology that applies equally well to both scenarios is clearly a better choice for generality and simplicity. The adaptive ring allows rarely or never-communicating sites to share data by relying on third-party replicas to gossip on their behalf. For these reasons, it is an attractive topology to consider, and one that applies equally well both within and between wards.[25-30]

For instance, assume the ward contains replicas 2, 4, and 6. If replica 5 joins but only replica 4 is knowledgeable of 5's existence, then replica 6 will synchronize with replica 4, which 6 believes to be the next replica in the ring. However, as part of 6's synchronization it learns of 5's existence, so the ring automatically “heals" itself as the new information propagates. Since the underlying algorithms are topology independent, correctness is not affected by temporary spokes on the adaptive ring. The example is illustrated graphically in Figure 1. (Ward is used generally here, and also refers to the meta-ward that contains all ward masters.). Ward masters are illustrated as solid black replicas; wards are double circles surrounding the replicas. Arrows indicate synchronization paths, although the synchronization topology adapts itself to network topology.

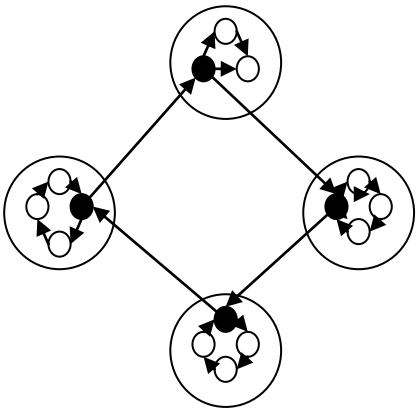

Fig. 1. The basic adaptive ring topology, both within and between wards.

Support for mobility

The model provides support for intra-ward mobility essentially for free, since everyone in the ward is a peer in the traditional sense. Intra-ward mobility occurs anytime the user is mobile within a restricted geographic area and is therefore only likely to encounter other machines from the same ward. For instance, moving within one's once from the desk to the couch demonstrates an intra-ward mobile action, as does moving around the once or within town to the local coffee shop. In all examples, the degree of motion is large enough to potentially change the set of “best" or most efficient communication partners, but not large enough to bring the user outside of his or her ward. For example, two colleagues in the once may never directly synchronize, typically relying on a set of third-party replicas to relay updates. However, if they meet at someone's home to discuss plans for the following day, they will usually want their machines to directly communicate.

IV. Advanced Ward Model

In the advanced Ward Model we introduce selective replication, the ability for a ward member to physically store only select portions of the complete volume. The details concerning how the reconciliation algorithms and controls handle selective replication are discussed in the next section. Here we describe the changes selective replication makes to the ward definitions, controls, and constructs.

Wards

The basic definition of a ward remains unchanged. The ward is still a collection of “nearby" volume replicas. However, all replicas need not store the same portions of the volume, which impacts both the ward set and synchronization topologies, as described in the following sections. The characteristics and abilities of ward members similarly remain unchanged. Specifically, any-to-any communication is still enabled between any two ward members. However, due to selective replication, any-to-any communication between all replicas may not always make physical sense for any particular layout of the file objects onto the set of volume replicas.

Ward set

The ward set is defined to be set of replicated data stored within the ward. In the basic Ward Model the ward set is equivalent to the volume. However, selective replication allows each ward member to store select portions of the volume, meaning that the set of data stored at each replica may be smaller than the entire volume. The ward set itself, therefore, may be smaller than volume, though of course it can never be larger. It is equivalent to the union of all replica sets for all replicas in the ward. The ward set changes dynamically as the set of data stored within the ward changes. For instance, if a given ward member uses the selective replication controls to locally add a new file system object, the ward set expands to include this new object. The ward set similarly decreases in size when replicas locally drop file replicas. Additionally, ward motion can change the ward set, as replicas move into and out of a given ward.

Ward masters

The ward master is responsible for the inter-ward synchronization of the entire ward set, and therefore must be able to identify the complete ward set. Use of both selective replication and ward changing, however, can change the ward set. Since any ward member can dynamically and optimistically change its replica set using the selective replication controls, and because new machines can at any time move into the ward (carrying with them their accompanying replica set), the ward set changes dynamically, lazily, and without global coordination. Without selective replication the ward set could be identified simply by naming the volume. With selective replication, however, the best mechanism for the ward master to identify its complete ward set is to individually name all objects in it.

Finally, we would like selective replication to help alleviate disk storage at the ward master. A ward member that wants to store a 100MB file should not necessarily force the ward master to also store that 100MB file solely for inter-ward synchronization purposes. We allow the ward master to store only a virtual replica-the equivalent to naming the object without storing _le data. When data is required for synchronization purposes, the ward master relays it via the physical data site.

Maintenance of consistency

Since each replica in the ward has a potentially different replica set, it follows that each ward master has a potentially different ward set. Therefore, in both intra- and inter-ward synchronization, we must use a more robust topology than a simple adaptive ring. We augment the basic adaptive ring to account for the differences between replica sets and ward sets. When the ward master is viewed as a “super-replica," the ward set appears exactly the same as a replica set for a super-replica, meaning that one approach can again be used in both intra- and inter-ward synchronization. The solution uses an adaptive ring for each file object, rather than one for the whole volume, and then coalesces multiple per-file rings into a single ring based on the intersection between replica.

Support for mobility

Selective replication introduces new difficulties for the ward motion algorithms. In the basic Ward Model, a machine moving into a new ward is guaranteed to have the same replica set as the new ward's ward set, because all participants store full volumes. Since the ward set is equivalent to the moving replica's replica set, it is straightforward for the new machine to integrate with the new ward. With selective replication, the mobile machine's replica set may differ from the new ward's existing ward set. As a result, the advanced Ward Model requires more rich and robust ward motion algorithms.

The ward master is responsible for the inter-ward synchronization of all intra-ward data. When a new machine enters the ward and brings with it data files not in the current ward set, there are two options: either the ward set expands to incorporate the new data objects, or it doesn't. In the former case the ward set changes, possibly causing changes at other ward masters since they must keep track of what is stored at the various ward masters to properly form their adaptive per-file rings. In the latter case the ward set remains unchanged, and there are no ripple effects affecting the other ward masters, but the mobile machine cannot synchronize all of its data completely within the new ward. Some of the data must be synchronized with another ward, most likely the original ward that the mobile machine came from. To properly decide which option is best, we must look at how physical motion actually occurs. Real mobility seems to occur in one of two modes:

visit: a temporary trip to a remote location, measured in hours or a small number of days

long stay: a longer stay at a remote location, perhaps for \a while," perhaps more permanently.

Each mode has accompanying expectations of cost and performance. Users expect a temporary mobile move to be lightweight and inexpensive: since they're not planning on staying very long, they don't want to pay a large up-front cost. Additionally, users will generally accept sub-optimal performance, given that they know the motion is temporary and the up-front cost is minimal. On the other hand, users moving for longer periods of time are generally willing to pay a more expensive up-front cost to gain better performance. Since they know they will remain at the remote location for a long time, they want good performance while there. The up-front cost is amortized over the length of the stay; users staying a short time do not gain significant benefit from the large cost, and are therefore generally unwilling to pay it. Real motion, of course, occurs over a continuous time spectrum, and doesn't always fall exactly into one of the two classifications. However, as a general paradigm the classifications seem to work fairly well, especially since positions in the middle of the spectrum can essentially be placed in either category.

V. System Architecture

As we discussed earlier, we are having several models to achieve services of Replication, but any of them is not able to fulfill our requirements. So we are using a new model, called Ward Model. Ward model is a hybrid model which combines the features of peer-to-peer and client server model as shown in figure 4. This model provides all the four basics qualities of replication system for mobility. In Ward Model, several new terms are used. These terms are Wards, Ward Member (MC), Ward Master (WM) and Ward set which we discussed already in previous sections. Figure 2 describes the ward model and Figure 3 shows the graphical views of ward models.

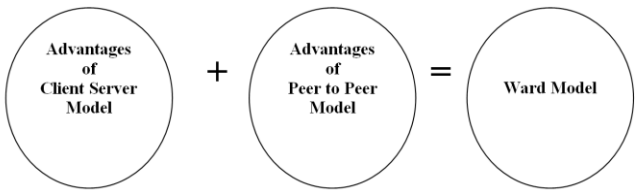

Fig. 2: Ward Model

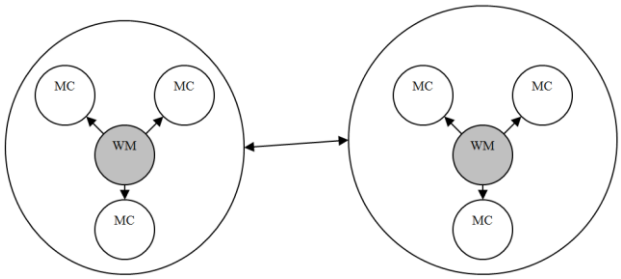

Fig. 3. Graphical view of Ward Model

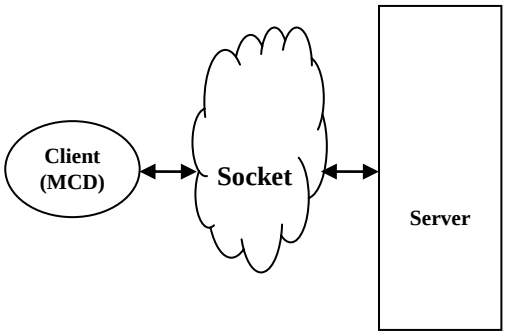

Fig. 4. Client server using socket.

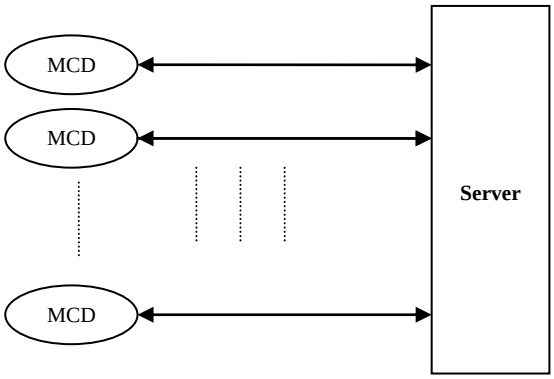

Fig. 5. Multiple Clients connected to Server.

Figure 4-9 shows the block architectures of the various phases of the approach such as, working of client server using socket, how multiple clients are connected to server, the process of client authentication, pinging of clients at server using hash table, receiving file from server and sending file to server.

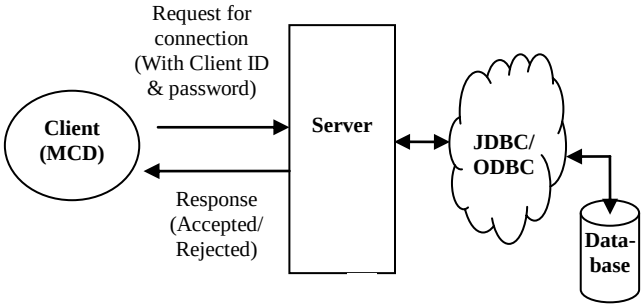

Fig. 6. Client Authentication

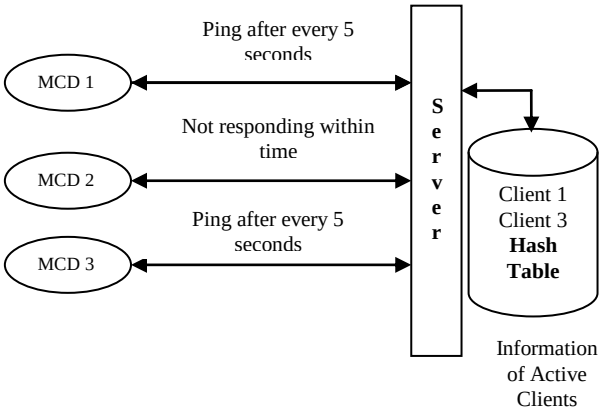

Fig. 7. Pinging of Clients at Server using hash table.

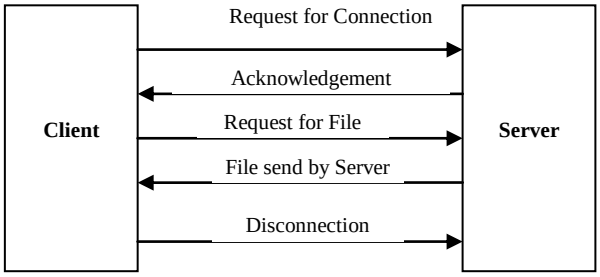

Fig. 8: Receiving file from server.

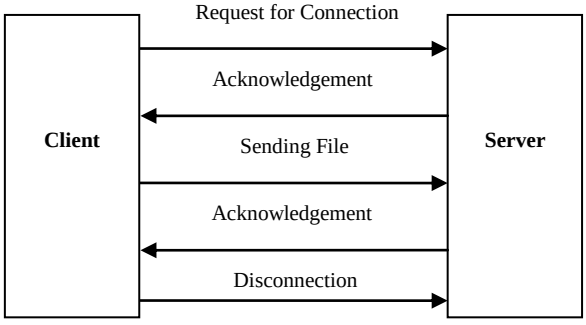

Fig. 9: Sending file to server.

Implementation Model: Implementation model is described with the help of following flow charts

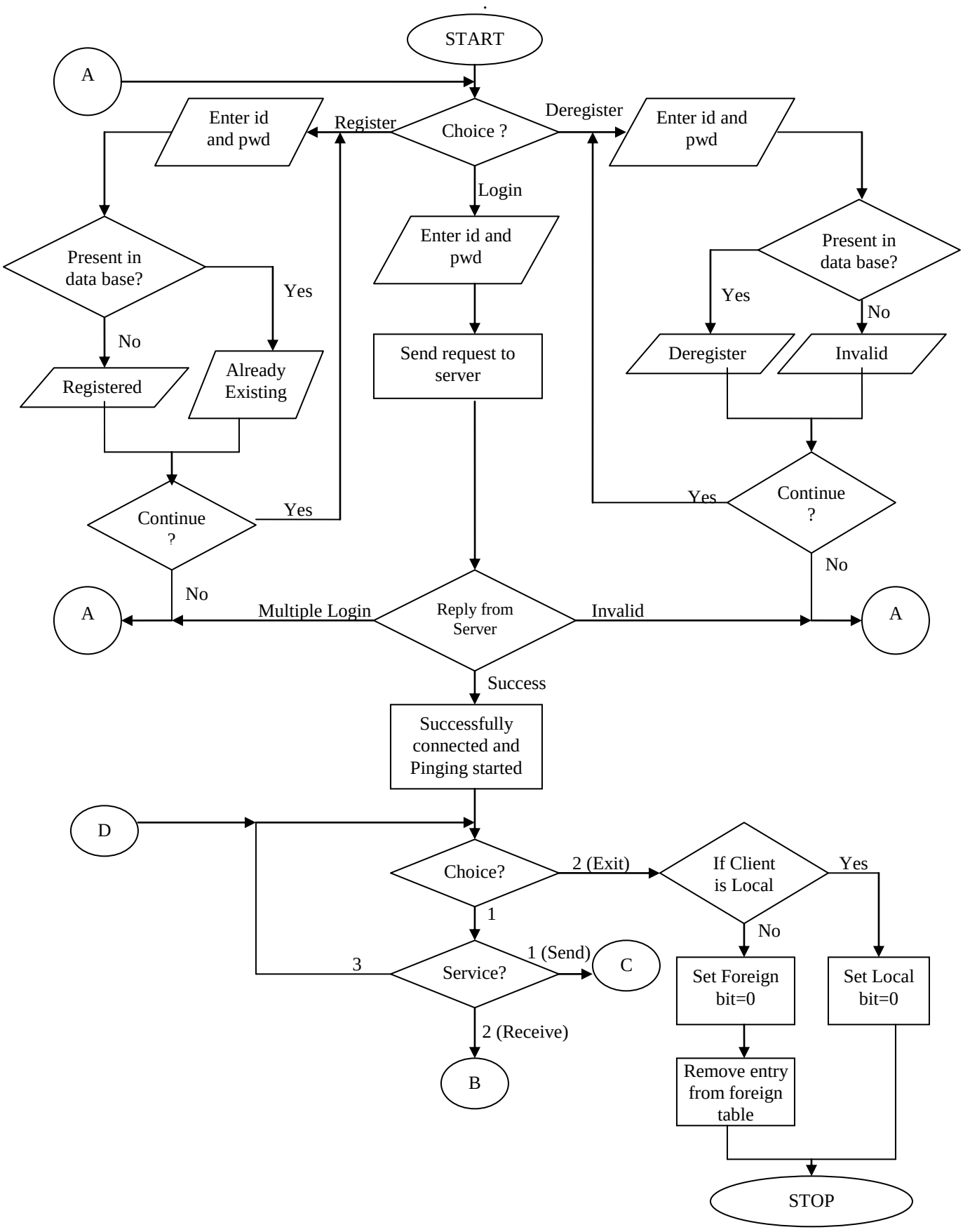

Fig. 10: Client Side Flow Chart

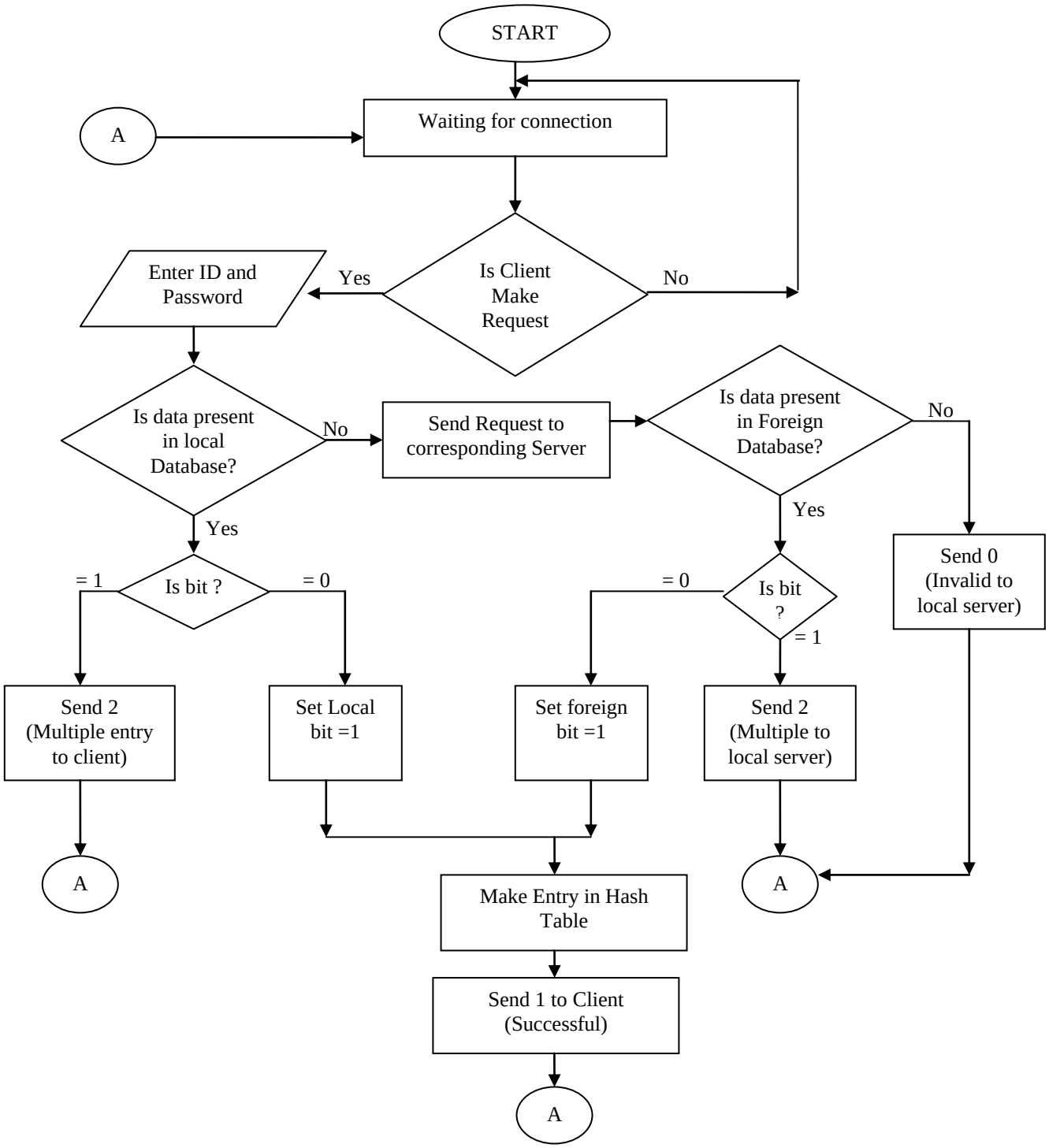

Fig. 11: Server Side Flow Chart

Figure 10 and Figure 11 describe the steps involved in the implementation of client and server side processes respectively with the help of scalable replica concepts while client made a request to the server.

VI. RESULTS

Database

We have used Microsoft- Access as Database. In this we have used following tables.

TABLE I
SERVER

Server ID	Server Name	Server Password
S1	Ser1	Pwd1
S2	Ser2	Pwd2

TABLE 2
SERVER1

Server ID	Server Name
M11	Pwd11
M12	Pwd12

TABLE 3
SERVER2

MCD_ID	MCD_PWD
M21	Pwd21
M22	Pwd22

Experimental Results

Figure 12 to 21 shows the output over command prompt while testing our proposed approach for different scenario. Figure 12 shows the registration of a new client. Client will enter the desired ID and password; if ID is available it will be assigned otherwise prompt for different ID. Figure 13 depicts the scenario of deletion of a registered ID. To delete the ID, password will be required.

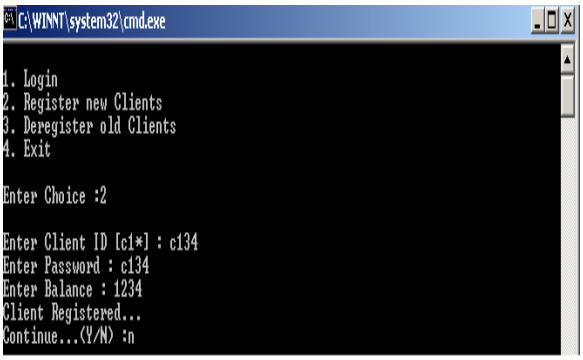

Fig. 12: Register new Clients

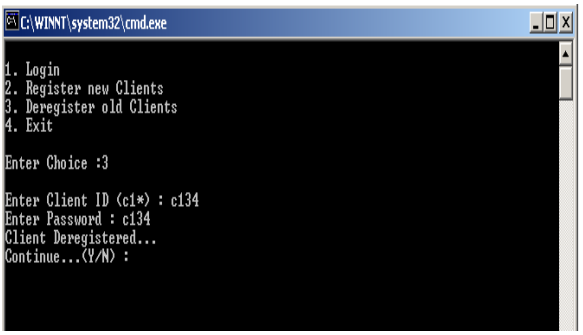

Fig. 13: De-registering old clients

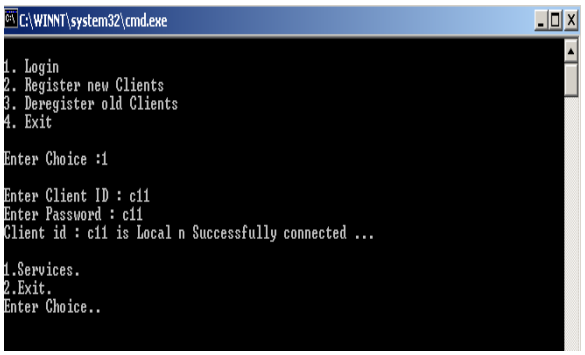

Fig. 14: Login

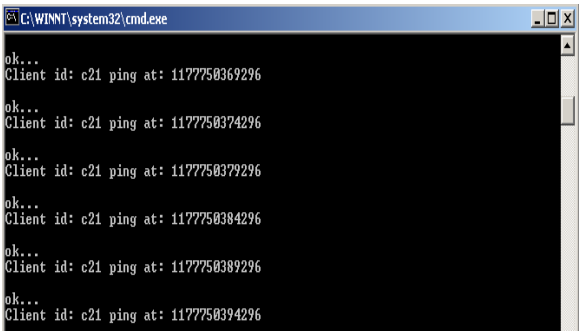

Fig. 15: Pinging

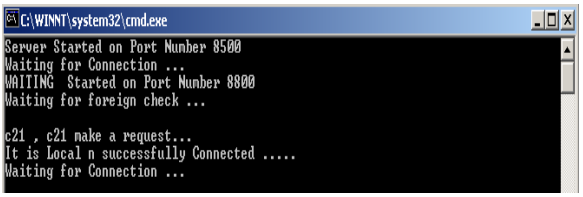

Fig. 16: Server Window

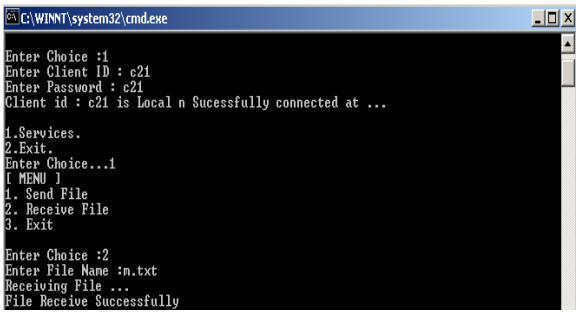

Fig. 17: Services used by clients

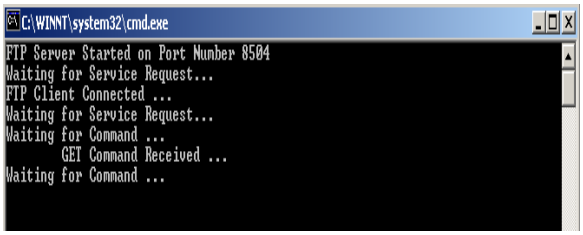

Fig. 18: FTP Server Window

Fig. 19: Connecting Foreign Client in VLR

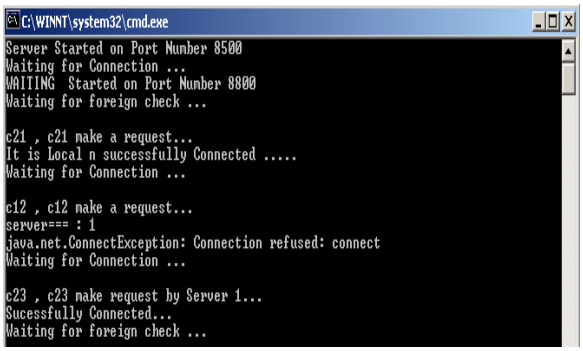

Fig. 20: Foreign Server Window

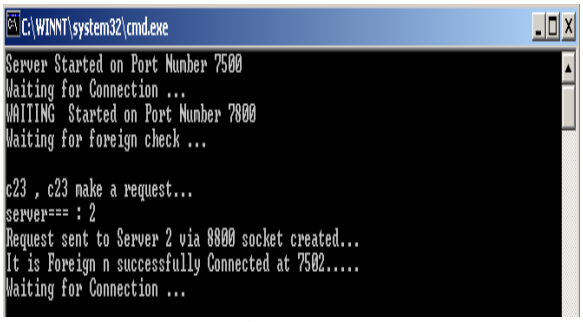

Fig. 21: Local Server Window (after connecting Foreign Client)

Figure 14 shows the steps of login; ID and password is required to login and after successful login some choices will be generated. Figure 15 is the case to test that any client is active or not. If it is pinging means it is active. Figure 16 shows the actions at the server side. At port number 8500, server is waiting for connection and if any request is made then it verifies its credential and after successful verification, client will be connected. Figure 17 shows the commands available for performing services to a client such as sending or receiving a file. Figure 18 shows the activities involved at the server side for file transfer to the clients. The example of connecting a foreign client is depicted in the Figure 19. Figure 20 shows the connection with a foreign server from different clients. Figure 21 shows the status of a local server window after connecting a foreign client. Foreign clients are connected at the port 7502 of local server.

From the experiments, it is deduced that scalable replica facilitates efficient client and server operations in the mobile computing which is purely based on the client and server architecture.

VII. Conclusion

In this paper we proposed a number of significant problems in large scale replication, and produced a scalable replication system for mobile environments. With the arrival of machines capable of supporting truly mobile computing came users wanting to access and update their data while mobile. If the user has more than one machine, for instance a laptop and a desktop, or if the data must be shared between multiple users, then the data must be replicated. Unfortunately, the existing replication systems are not mobile-compliant. Designed for stationary environments, they do not provide users with the abilities they require when mobile. Furthermore, we build into Roam's algorithms the ability to automatically handle and update outdated meta-data as part of the normal synchronization of user data. Roam makes extensive use of the optimistic approach in its management of the system. Roam is a successful piece of research, based not just on the performance analysis but also on real-world experience and use. It is our hope that Roam paves the way for real mobile use and future mobile computing research; simultaneously, we would like the underlying ideas and concepts to bear fruit and diseases and become used in other areas of computer science.

Refrences

[1] D. Ratner, P. Reiher, and G. J. Popek, “Roam: a scalable replication system for mobile computing,” in: Proceedings of the 10th International Workshop on Database and Expert Systems, 1999, pp. 96 – 104.

[2] M. Satyanarayanan, J. J. Kistler, P. Kumar, M. E. Okasaki, E. H. Siegel, and D. C. Steere, “Coda: A highly available file system for a distributed work station environment,” IEEE Transactions on Computers, vol. 39, no. 4, pp. 447-459, 1990.

[3] J. J. Kistler and M. Satyanarayanan, “Disconnected operation in the Coda file system,” ACM Transactions on Computer Systems, vol. 10, no. 1, pp. 3-25, 1992.

[4] P. Honeyman, L. Huston, J. Rees, and D. Bachmann, “The Little Work project,” in: Proceedings of the 3rd Workshop on Workstation Operating Systems, 1992, pp. 11-14.

[5] J. Howard, M. Kazar, S. Menees, D. Nichols, M. Satyanarayanan, R. Sidebotham, and M. West, “Scale and performance in a distributed file system,” ACM Transactions on Computer Systems, vol. 6, no. 1, pp. 51- 81, 1988.

[6] D. B. Terry, M. M. Theimer, K. Petersen, A. J. Demers, M. J. Spreitzer, and C. H. Hauser, “Managing update conicts in Bayou, a weakly connected replicated storage system,” in: Proceedings of the 15th Symposium on Operating Systems Principles, 1995, pp. 172-183.

[7] Jatav, Vinod Kumar, et al. "Wireless Sensor Networks: Attack Models and Detection." 2012 IACSIT Hong Kong Conferences, IPCSIT vol. 30 (2012)©(2012) IACSIT Press, Singapore. 2012.

[8] P. Reiher, J. Popek, M. Gunter, J. Salomone, and D. Ratner, "Peer-to-peer reconciliation based replication for mobile computers," in: Proceedings of the ECOOP Workshop on Mobility and Replication, 1996.

[9] Semwal, Vijay Bhaskar, K. Susheel Kumar, Vinay S. Bhaskar, and Meenakshi Sati. "Accurate location estimation of moving object with energy constraint & adaptive update algorithms to save data." arXiv preprint arXiv:1108.1321 (2011).

[10] M. Ferreira and M. Casquilho, "An Example in Remote Computing Over the Internet applied to Geometry," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 2, no. 3, pp. 38-43, 2013.

[11] J. Solis-Martinez, N. Garcia-Menendez, C. Pelayo G-Bustelo, and J. M. Cueva-Lovelle, "BPLOM: BPM Level-Oriented Methodology for Incremental Business Process Modeling and Code Generation on Mobile Platforms," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 2, no. 2, pp. 13-27, 2013.

[12] Neha Gandotra, Vishwanath Bijalwan "coexistence model of zigbee & ieee 802.11b (wlan) in ubiquitousnetwork environment" IJARCET, volume 1, Issue 4,June 2012.

[13] Vishwanath Bijalwan, Dr. Sanjay Singh "ANALYSIS &DESIGN OF JOINT PHY-MAC MODEL OF IEEE 802.15.4" IJSETR, volume 2, Issue 9, September 2013..

[14] S. Isaacman, R. Becker, R. Caceres, S. Kobourov,M. Martonosi, J. Rowland, and A. Varshavsky. Identifying important places in people's lives from cellular network data. In Proc. Int. Conf. on Pervasive Computing, San Francisco, Jun. 2011.

[15] R. G. Guy, "Ficus: a very large scale reliable distributed file system," Ph.D. dissertation, University of California, Los Angeles, June 1991.

[16] J. P. Gupta, N. Singh, P. Dixit, V. B. Semwal, and S. R. Dubey, "Human Activity Recognition using Gait Pattern," International Journal of Computer Vision and Image Processing, vol. 3, no. 3, pp. 31 – 53, 2013.

[17] K. K. Susheel, V. B. Semwal and R. C. Tripathi, "Real time face recognition using adaboost improved fast PCA algorithm," arXiv preprint arXiv:1108.1353 (2011).

[18] K. S. Kumar, V. B. Semwal, S. Prasad and R. C. Tripathi, "Generating 3D Model Using 2D Images of an Object," International Journal of Engineering Science, 2011.

[19] N. Singh, S. R. Dubey, P. Dixit and J. P. Gupta, "Semantic Image Retrieval by Combining Color, Texture and Shape Features," In the Proceedings of the International Conference on Computing Sciences, pp. 116-120, 2012.

[20] V. B. Semwal, V. B. Semwal, M. Sati and S. Verma, "Accurate location estimation of moving object in Wireless Sensor network," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 1, no. 4 , pp. 71-75, 2011.

[21] E. Felembane, C. Lee and E. Ekici, "MMSPEED: Multipath Multi-SPEED Protocol for QoS Guarantee of Reliability and Timeliness in Wireless Sensor Networks," IEEE Transactions on Mobile Computing, vol. 5, no. 6, pp. 738-754, 2006.

[22] S. R. Dubey and A. S. Jalal, "Adapted Approach for Fruit Disease Identification using Images," International Journal of Computer Vision and Image Processing, vol. 2, no. 3, pp. 44-58, 2012.

[23] Sanjay Singh, Vishwanath Bijalwan, "Design Of Wireless Sensor Network Node On Zigbee For Water Level Detection" Volume 3,Issue8, August 2013.

[24] Anchit Bijalwan, Vishwanath Bijalwan, "Examining the Crimninology using Network Forensic" 8th National Conference USCSTC ,Dec 2013.

[25] Vishwanath Bijalwan, Vinay Kumar, Pinki Kumari,Jordan Pascual," KNN based Machine Learning Approach for Text and Document Mining" International Journal of Database Theory and Application Vol.7, No.1 (2014), pp.61-70.

[26] Pinki Kumari and Vikas Pareek, RAKSHITA- A Novel web based Approach for Protecting Digital Copyrights Using Public Key Digital Watermarking and Human Fingerprints" International conference on methods and models in computer science (ICM2CS-2010).

[27] Pinki Kumari and Abhishek Vaish, "Instant Face detection and attributes recognition" in International Journal of Advanced Computer Science and Applications (IJACSA - ISSN 2156 5570):2011

[28] Pinki Kumari and Abhishek Vaish,"A Comparative study of Machine Learning algorithms for Emotion State Recognition through Physiological signal", Advances in Intelligent Systems and Computing, Vol.236-Springer; ISBN 978-81-322-1601-8

[29] Pinki Kumari and Abhishek Vaish "Brainwave's Energy feature Extraction using wavelet Transform" proceeding of IEEE SCEECS, 2014, MANIT, Bhopal ISBN: 978-1-4799-2526-1.

[30] Semwal, Vijay Bhaskar, K. Susheel Kumar, Vinay S. Bhaskar, and Meenakshi Sati. "Accurate location estimation of moving object with energy constraint & adaptive update algorithms to save data." arXiv preprint arXiv:1108.1321 (2011).

Meenakshi Sati presently working as Assistant Professor, India. She is B.Tech from Graphic Era University Dehradun and pursing M.Tech from NITTR Chandigrah. Her major research work Interest in Image Processing, Sensor Network, Design and Analysis of Algorithm, Data Structure.

Vivek Vikash is presently working as Software Engineer, Honeybell, India. He completed his M.Tech form Indian Institute of Information Technology, Allahabad, his major research work Interest in Image Processing, computer sensor network & Pattern Recognition.

Vishwanath Bijalwan is working as assistance professor at Uttarakhand Technical University Dehradun. He is person with lot of potential. His research interest is wireless sensor network, WiMax, Wi-Fi, machine learning, Informtion Retrieval.So far he has published 4 high quality researches paper and work on many MHRD funded project.

Pinki Kumari received her M.Tech in Computer Science and Engineering from Banasthali University, Rajasthan, and her Ph.D in Information Security. Her research interests include Pattern recognition, Cloud Computing, Big Data and Distributed Database.

Manish Raj is currently pursuing his Ph.D fm IIITA.. He received his M.Tech from IIITA India. He has more than 5 years of research and teaching experience. He supervised 5 PG and 21 UG students. He has 2 research papers in reputed international journal and conference.

Meenu Balodhi is currently pursuing her fm Uttaranchal technical university. His research area biometric identification and wireless sensor network

Priya Gairola is currently pursuing her fm Uttaranchal technical university. His research area biometric identification and wireless sensor network, Wi-Max

Vijay Bhaskar Semwal has been severed as a PCC member in several International conferences and Journals. He was also head for 5th IEEE conference on big data at COER Roorkee. He is carrying more than 5 year of academic and industry experience. He served for various top organizations like Siemens, Newgen etc. Earlier he received his M.tech in wireless sensor domain and was awarded by gold medal. His major research work interest in Wireless Sensor Network, Artificial Intelligence, Image Processing, Computer Network &Security, and Design & Analysis of Algorithm, Machine Learning, and Information Retrieval Soft Computing.