

Performance enhancement of IP Storages over Wireless Networks

A Dissertation submitted to the
School of Computer & Systems Sciences
Jawaharlal Nehru University, New Delhi
in partial fulfilment of the requirements for the award of the degree of

Master of Technology in Computer Science and Technology

By,
Anshuman Bose

Under the supervision of
Mr. Sushil Kumar



**SCHOOL OF COMPUTER & SYSTEMS SCIENCES
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI – 110067, INDIA**



जवाहरलाल नॅहरू विश्वविद्यालय

**JAWAHARLAL NEHRU UNIVERSITY
SCHOOL OF COMPUTER & SYSTEMS SCIENCES
NEW DELHI – 110067, INDIA**

CERTIFICATE

This is to certify that the dissertation entitled “**Performance enhancement of IP Storages over wireless networks**” being submitted by **Anshuman Bose** to the School of computer & systems sciences, Jawaharlal Nehru University, New Delhi, in the partial fulfilment of the requirements for the award of the degree of **Master of Technology in Computer Science and Technology**, is a record of bonafide work carried out by him under the supervision of **Mr. Sushil Kumar** (Assistant Professor).

This work has not been submitted in part or full to any university or institution for the award of any degree or diploma.

Mr. Sushil Kumar
Assistant Professor
SC&SS, JNU

Prof. Parimala N:
Dean
School of Computer & Systems Sciences
JAWAHARLAL NEHRU UNIVERSITY
NEW DELHI-11 067
Dean
SC&SS, JNU



जवाहरलाल नॅहरू विश्वविद्यालय

**JAWAHARLAL NEHRU UNIVERSITY
SCHOOL OF COMPUTER & SYSTEMS SCIENCES
NEW DELHI – 110067, INDIA**

DECLARATION

This is to certify that the dissertation entitled “**Performance enhancement of IP Storages over wireless networks**” is being submitted to the **School of Computer & Systems Sciences, Jawaharlal Nehru University, New Delhi**, in partial fulfilment of the requirements for the award of degree of **Master of Technology in Computer Science and Technology**, is a record of bonafide work carried out by me.

The matter embodied in the dissertation has not been submitted in part or full to any university or institution for the award of any degree or diploma.

Date: 19/10/07

Place: New Delhi

Anshuman Bose

Reg. No. 05/10/MT/03

SC&SS, JNU

Acknowledgements

I would like to thank my supervisor Sushil Kumar Dohare for the encouragement, enthusiasm and direction provided to me during the entire duration of this proposed work. My parents for their dogged persistence and motivation in getting me complete my degree. I would also like to thank the entire community of SC&SS for the excellent environment they have created for research. Also, I would like to take this opportunity to thank Colin Laplace et al. for Dev-Cpp without which the simulation won't have been possible.

Anshuman Bose

Abstract

In the corporate world every decision is taken on the basis of the data available in its storage devices. The amount of data being stored is massive and the growth of such data is exponential as 'data breeds data'. The new research in the field of wireless devices has made these devices much more powerful than ever, added to this is the arrival of new high speed communication facilities like the Gigabit Ethernet etc., and the new advancement in the Ad-Hoc networks technologies. Also, the security of the data is another has become an issue of major concern in the recent past. The iSCSI protocol developed by IETF has emerged as a very good solution capable in catering the current corporate world's demands.

This work is aimed at improving the performance of iSCSI protocol over the wireless networks. We have tried to achieve this by putting in suitable changes in the architecture generally available in the corporate offices of today's world.

Contents

1. Introduction	1
Evolution of Network Storages Solutions	2
Direct Attached Storages	3
Network Attached Storages	4
Storage Area Networks	6
IP Storages	8
Emergence of Gigabit Ethernet	10
Advent of powerful wireless devices	11
Inspiration of the present work	12
The Challenges and proposed solutions	12
2. Related work	13
Fisheye State Routing	13
Secured Internet Key exchange protocol	14
Specialized crypto-processing of Data Blocks	14
Overview of the iSCSI protocol	15
3. Proposed Work	17
The Basic Proposed Model	17
Section (i)	
Hierarchical Caching of the Data Blocks	19
Application Scenario	21
Mathematical Model of the above illustration	21
Fixed Data Block size	22
Probabilistic Estimation	23
Variable Data Block size	25
Probabilistic Estimation	27
Section (ii)	
Pre-fetching of the Data Blocks	31
Application Scenario	32
Directed Links & Node generation	33
Directed Links deletion	34
Node invalidation & deletion	34
Application Scenario	35

Section (iii)		
	Cache Co-operation among Mobile Clients	37
	The Mathematical computation of κ_{MCP}	41
	Application Scenario	41
	The Mathematical Model	43
Section(iv)		
	Encryption & Decryption of Data Blocks	46
	The iSCSI Protocol (Login Authentication & Security of Data)	46
	Application Scenario	49
4.	Simulation Results	50
	Simulation Results	50
	Analysis of the result	52
5.	Conclusion & Future Work	55
	Conclusion	55
	Future Work	56
6.	Appendix	58
	Simulation Code	58
7.	References	64

List of Figures & Tables

List of Figures

1.1	The Directly Attached Storages	3
1.2	The Network Attached Storages	5
1.3	The Storage Area Networks	6
1.4	The iSCSI connections	9
3.1	Illustrative representation of the Proposed Model	17
3.2	Hierarchical Caching of Data Blocks	19
3.3	Pre-fetching of Data Blocks	31
3.4	Access history pattern of fetching of Data Blocks	32
3.5	Generation of Directed Links during short duration	33
3.6	Generation of Directed Links when the time duration is more than τ_s	33
3.7	The Ad-Hoc Network connections among MCs	37
3.8	The Network from the point of view of Mobile Client 1	42
3.9	The improvised implementation of IPSec to enhance performance	48
4.1	The Bar-Graph of the Data Block access from different location	52
4.2	The Pie-Chart of location of Data Block access when $N = 50000$	53
4.3	The Pie-Chart of location of Data Block access when $N = 125000$	54

List of Table

4.1	The location from where the Data Block was accessed	51
-----	---	----

Chapter 1

Introduction

“It took humans 300,000 years to accumulate 12 exa-bytes of data, & it would take only another two and a half years to double it.”

According to a paper of University of Berkeley.

The above quotation about the data (storage) says it all why we need new storage solutions which are more efficient in terms of

- Accessibility of Storages,
- Availability of the stored Data,
- Maintenance of the stored Data,
- Scalability of the Storage Devices,
- Security of the stored Data,
- TCO of the stored Data.

The quotation also specifies that we are living in the DATA age. In the corporate arena every decision is taken on the basis of data collected from different relevant sources [23]. The corporate executives are no longer storing data in the form of Alpha-Numeric texts only; in contrast the use of new data types such as image, audio, video, web-page etc has become a common phenomenon, all of which requires immense Data Storing capabilities. Further, it has been a trend in the corporate world to store more and more wise data as this is one of the keys to future success. The applications like Data Mining, Data Warehousing, Data Harvesting, Disaster Recovery & Management, and Data Sharing (both within the corporate network & within the trusted network of other corporate) etc are taken for granted by these organisation, though it is a tough task to perform considering the

constraints like Time, Business Deal Continuity, end user satisfaction (including customer satisfaction) and overall cater to large number of end users concurrently.

Major challenges that are being now faced are in the field of Disaster Recovery, Management and concurrent access.

The storage solution providers are excited by the drastic reduction in the cost of storages as this helped them in coping with the ever increasing size of the data at a reasonable cost [24]. But, the extremely large size of the data posed some new challenges before them and they are:-

- (i) Volume Management,
 - (ii) Accessibility.
- (i) Volume Management of the data will continue to be a challenge for the storage solution providers as the volume of data generated by the corporate is huge and also the growth is exponential in nature.
 - (ii) Accessibility of the data has become a big concern as the volume increases this task becomes even more challenging. As already mentioned the data is stored by the corporate for intelligent use in order to get benefited in the future the processing of data has to be done, and for this to happen the data should be preserved inexpensively, robustly, securely yet easily accessible to the concerned authorised users.

The security of the stored data, the disaster recovery management and the ever increasing real state price has become a major issue among the top management of any corporate over the past few years. After, the September 11 attacks on the World Trade Centre in New York the need for a remote copy of the data was further underlined.

Evolution of the Network Storage Solutions:

The storage solutions have always responded to the industry demand [17]. The evolution of the storage solutions has been in tunes to cater the demands of the industry. The years of 1990s saw a rapid growth in the hard disc technology in terms of capacity in individual PCs, which in turn lead to the growth of data from few Megabytes to tens of Gigabytes stored at different locations. We consider the following major categories of storage solutions witnessed by the industry in the recent past:-

- Direct Attached Storages (DAS),
- Network Attached Storages (NAS),

- Storage Area Networks (SAN),
- IP Storages (iSCSI, iFCP, FCIP).

Direct Attached Storages (DAS) [31]: These were the most primitive form of storage solution. These storage devices were directly attached to the servers by cables. In PC configuration these comprised of Disc attached inside the PC cabinet, while in the case of the Mainframes or large open servers the DAS are attached by cables few metres away from them. These are easy to implement and data was mainly transferred in blocks or low-level granular units using the SCSI commands.

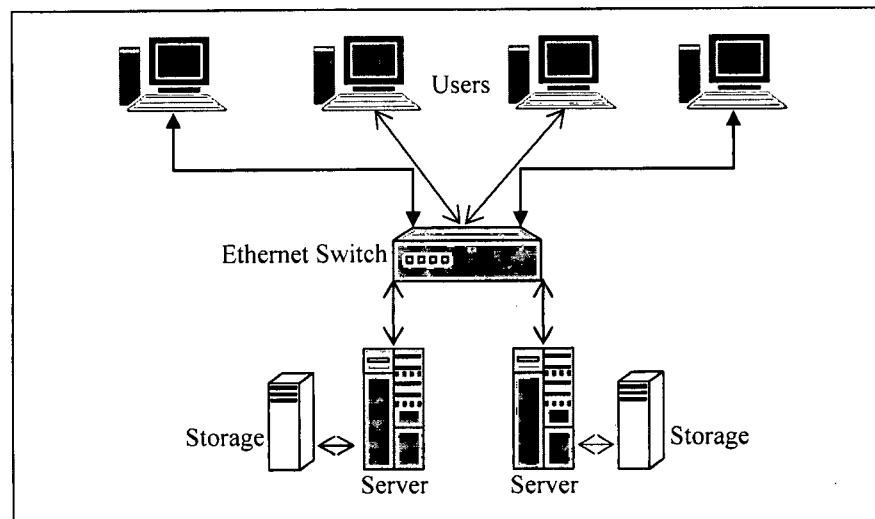


Figure No. 1.1: The Directly Attached Storages

The advantages of DAS were as follows:-

- **Low Ordering Costs**: The SCSI interconnection bus cable is cheaper and widely available commodity, this results reduction in the cost in terms of administrative overheads and other logistic cost.
- **Low Installation Cost**: As the SCSI cable are used for point-to-point connections, the skills required for the same becomes less, hence reduction in the installation cost.

- Nice Performance: As the SCSI is designed for storage purpose the software overheads are reduced and replaced by the hardware assistance, thus optimising the performance.

The disadvantages of these directly attached storages are as follows:-

- Distance Constraints: The SCSI devices work over the parallel cables at most to a distance of few metres so the storage devices have to be placed near the servers. This increases the cost as the storage devices consume a decent amount of real state.
- Limited Scalability: As the size of data increases new disc are added to the storage locations attached to the servers. But this solution has its own limitation due to fixed size of the cabinets.
- High Maintenance Cost: As the storage devices are spread across the corporate office at different venues, the number of staff for maintenance increases and also the co-ordination between these peoples also increases the cost.
- Data Protection: Data protection becomes a challenging job in such type of storages as the data is spread across the corporate office.

Network Attached Storage (NAS) [25]: NAS is a file-based storage architecture which optimises file sharing across the network. In this architecture the recourses are directly attached to the LAN. Data is sent and received over the LAN using the TCP/IP protocol. A NAS is actually a specialised server (with its own operating system and shared storage infrastructure) which is LAN addressable.

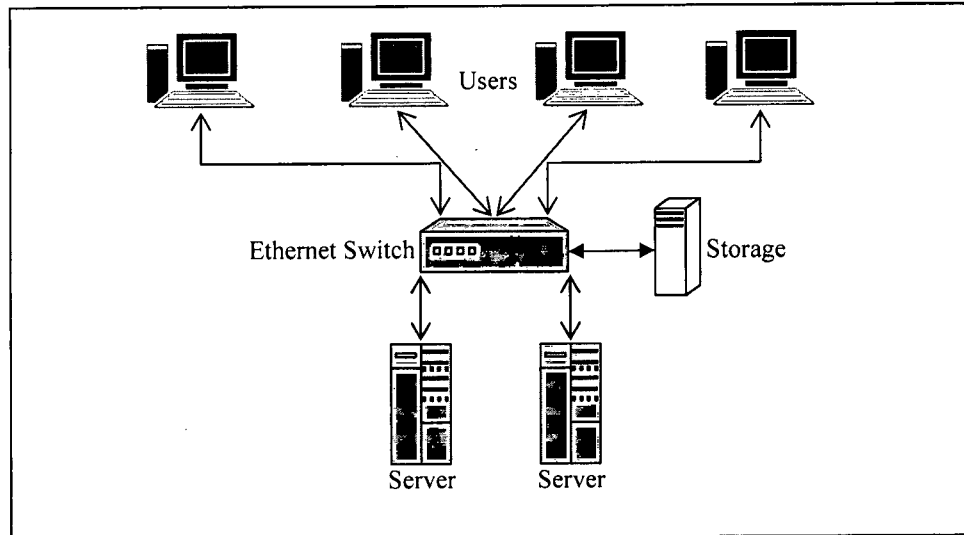


Figure No. 1.2 The Network Attached Storages

The advantages of NAS solution are as follows:-

- Simpler Implementation: As the NAS storage solutions comes in plug-and-play form and are LAN addressable hence makes them easier to install, maintain and administer.
- IP Infrastructures usage: As the NAS uses the existing IP infrastructure installed in the workspaces the cost is reduced not only in terms of infrastructure support but also in terms of the manpower required to maintain these.
- Centralised Storage and resource sharing: The centralised copy of the document removes the need for multiple copies, as in the case of DAS storage solutions, and hereby reducing the discrepancies occurring in the different copies of the same document.
- Scalability: NAS solutions are more scalable than the DAS ones as the extra storage are placed at the network attached specialised servers.
- Heterogeneous File Architecture Support: The NAS systems are designed to support heterogeneous file sharing over the network. This helps in better scalability of organisations work as the different sections within an organisation may wok on different platforms.

The disadvantages of NAS based storage solution are as follows:-

- Integrity of the Data: the integrity of data is a big concern in case of NAS. The Ethernet protocols has been mainly designed for messaging applications and not for data packet transfer, thus it ay happen in case of busy network situation a data packet being dropped without any warning.
- Database Applications Unsuitable: The NAS has been designed for file I/O transaction and thus is not suitable for the database applications which uses “raw” block I/O for high performance. For example many applications in ORACLE or DB2 exploits “raw” block I/O to improve their performance, and for this reason NAS do not scale as good as DAS, SAN or iSCSI solutions.
- Bandwidth Requirement: The NAS requires to more large blocks of data over the LAN for proper transactions, which consumes a large bandwidth. Also, the features of back-up and restore application demand a lot of bandwidth.

Storage Area Networks (SAN) [24]: Storage Area Networks are dedicated networks behind the servers that connect the servers and storage devices without burdening the enterprise LAN. The SAN are designed for improved performance, reliability, scalability and Management.

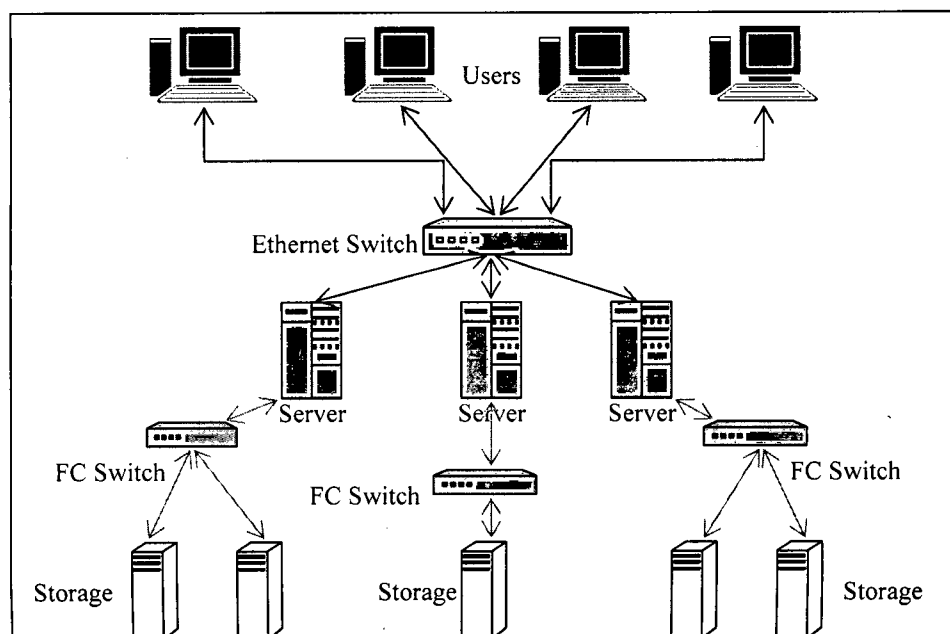


Figure No. 1.3: The Storage Area Networks

The advantages of SAN [27] are as follows:-

- Remote Storage: In SAN servers are connected to the storage devices via fibre channel; the storages now could be located at a remote (undisclosed) location which in turn results in increased flexibility and control. Also, the management cost is reduced.
- Scalability: The capacity of storage can be increased without effecting the ongoing transaction in SAN solution, this helps in business continuity.
- Increased Efficiency: As the SANs use fibre channel for connection the transfer of data is not only efficient but also very high at the same time. In fact the data transfer is faster than the gigabit Ethernet.
- Data Integrity: In SAN the use of fibre channel data integrity is handled in a better way as the sequence checking and acknowledgement of the frames is done in the Hardware which reduces the software overheads.

The disadvantages of SAN are as follows:-

- High Installation Costs: As SAN uses fibre channel for connection, new Fibre Channel Network infrastructure has to be laid for this purpose. The cost of these is high compared to their Ethernet counterparts.
- Interoperability between products from different vendors: As the technology of Fibre Channel is new, the standardisation of Fibre Channel equipments is not at the level of Ethernet equipments; hence the problem of interoperability among products from different vendor still persists.
- Skilled Personal Shortage: Unlike the Ethernet which has been for over fifteen years Fibre Channel is a new technology, thus the number of skilled personals in this field is less which increases the hiring costs.

IP Storages (iSCSI, iFCP, FCIP) [21]: IP Storages comprises of the most recent developed in the storage solution industry.

iSCSI stands for 'Internet Small Computer System Interface,'

iFCP stands for 'Internet Fibre Channel Protocol,'

FCIP stands for 'Fibre Channel over IP'.

The above protocols have been designed to cater the needs of the industry. Of the three different protocols mentioned above iFCP & FCIP are discussed briefly below:

iFCP: Internet Fibre Channel Protocol is mainly a gateway-to-gateway protocol. It helps in transmitting data to-and-from fibre channel devices via the TCP/IP connection. In this protocol the FC frames are encapsulated in the IP datagram for transportation over the TCP/IP network. The FC header is mapped to the IP header before the frame is transmitted. iFCP relies on TCP for error detection, congestion control and recovery.

FCIP: Fibre Channel over IP encapsulates the block data of the fibre channel and transports it over the TCP tunnel. The packets from the fibre channel remain unaltered and they are just encapsulated into the IP frames.

iSCSI or the Internet SCSI protocol is one of the most promising protocols in the networked storage arena. The iSCSI has been developed by IETF and already standardised. iSCSI takes advantage of the IP-based networks of the corporate organisation (some of which are very fault tolerant) to transport SCSI commands. iSCSI uses the new developments in the form of gigabit Ethernet to overcome the latency caused by putting storage and other network traffic over the same route.

iSCSI is based on the client-server architecture [1] in which the client (initiator) issues SCSI commands to request services from the server (target). Each such session, in iSCSI, between the initiator and the target is identified by its session ID which comprises of initiator ID and the target ID.

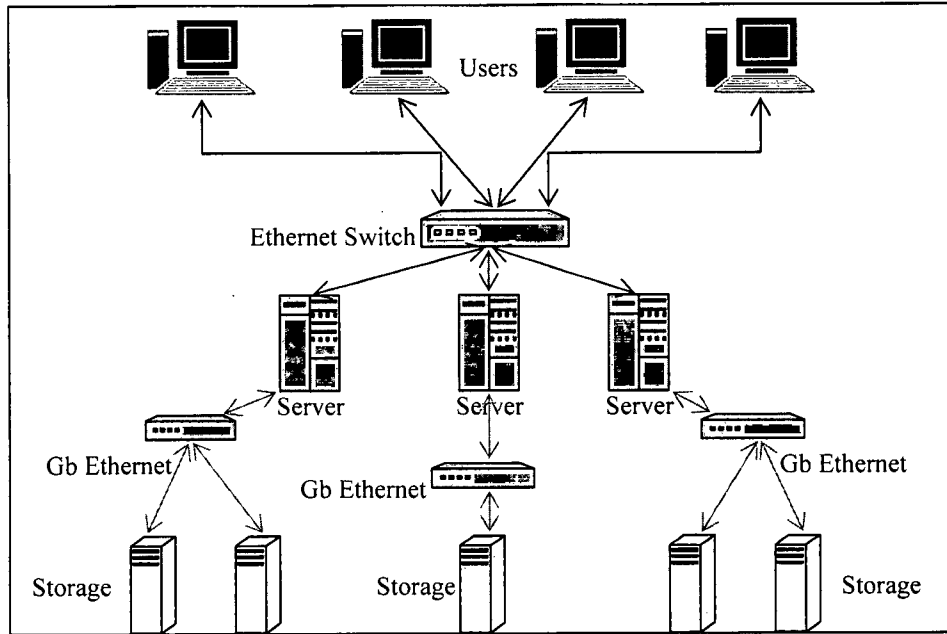


Figure No. 1.4: The iSCSI connection

The advantages of iSCSI are as follows:-

- **Existing Connectivity:** As the iSCSI uses the LAN for transportation of data the existing LAN infrastructure could be used for it. The distance is not a big concern as in the case of Fibre Channel as the data flows over the TCP/IP network.
- **Interoperability:** As iSCSI uses the TCP/IP for connection the interoperability is not at all an issue since the TCP/IP network has already been standardised many years before.
- **Management:** The management of the iSCSI devices is simple as they are managed as their SCSI counterparts. Also, the network is managed in the same way as the normal TCP/IP networks.
- **Skilled Personals:** As the TCP/IP network is time tested and the number of skilled personals in this field is far more than the number of skilled personal in the Fibre Channel the procurement of eligible candidate is not a big issue. This in turn reduces the cost.

Emergence of Gigabit Ethernet:

The increased usage of applications like Data Warehousing, Data Mining, Data Harvesting, CAD, 3D modelling, medical imaging, Streaming Videos etc to-and-from the server has become a common phenomenon. The enhanced server processing capacities are not the only requirement for such purposes; in fact they are of not much use unless supported by high speed network connections. The need for such high speed network connections resulted into the form of Gigabit Ethernet [22].

The development of Gigabit Ethernet has resulted in enhancing the productivity as it is helpful in many ways like:-

- **Collaborative Work Environment:** The physical location of the executives is no longer any constraint as they could interact with their counterparts via high quality video conferencing while continuing their own work (say on CAD) at the same time.
- **Sharing of Large Files:** The Data files used by the corporate are no longer only of alpha-numeric in nature, but today these may contain spreadsheets, promotion videos, design images, animations etc all of them form very large sized files. The advent of Gigabit Ethernet has made it possible to share such kind of file within a reasonable amount of time.
- **Multi-Tasking:** Today a executive can have couple of spreadsheets opened in his/her PC while detaching a large file from the e-mail attachment, and simultaneously downloading a video sent to him by one of his/her customers. Such a scenario is possible today largely due to increased power of the PCs and the Gigabit Ethernet.

Thus, the ever increasing demand of bandwidth due to such applications, increased processing powers of the processors and powerful operating systems etc all contributed in bringing in the revolution in high capacity networks.

Advent of powerful wireless Devices:

The wireless devices available today are more powerful than their counterparts in the recent past. This sector has seen a massive technological growth in the recent years. The processing power of the mobile device processor has increased many times. The storage capacity of these devices has also increased substantially. Also the development in the technologies related to powerful batteries which are source of power to such devices has been decent. All the above has lead to more powerful wireless devices than ever before.

The Laptops have become almost equivalent to the PCs in almost every aspect, while the PDAs now have additional features like word processing, spread sheets, wireless LAN connectivity, and Ad-Hoc network ready. The multimedia rich applications can be run easily on such devices these days (something which was not even thought of couple of years back).

The wireless LAN architectures have also been made better.

- IEEE 802.11b offers a data rate of 2 Mb/s to 11 Mb/s at 2.4 GHz,
- IEEE 802.11a offers a data rate of 6 Mb/s to 54 Mb/s at 5 GHz,
- IEEE 802.11g offers a data rate over 20 Mb/s and upto 54 Mb/s at 2.4 GHz,
- IEEE 802.11n is expected to give a data rate of at least 100 Mb/s which would be 25 times faster than IEEE 802.11b and 5 times faster than IEEE 802.11a & IEEE 802.11g.
- HIPERLAN/2 gives a data rate up to 54 Mb/s

The technology for the wireless LAN is growing rapidly and the slow data rate is becoming a thing of past.

The popularity of such devices among the executive is natural. These devices help in their day to day work and increase their efficiency manifold. The ever increasing usage of such devices has forced the storage industry to work out solutions which can cater to such demands.

Inspiration of the present work:

Our work has been inspired by the above existing scenario. In our work we have focussed on iSCSI protocol only as this is likely to become more ubiquitous in the years to come. The wireless network has been chosen because of the enhanced capabilities of these devices.

The proposed work has been categorised into four sections each of which helps in improving the performance of iSCSI over the wireless networks.

The Challenges and proposed solutions:

One of the first and foremost challenges we encountered during our study of “improving the performance iSCSI over the wireless networks” was whether each Data Block asked by the user should be made available only from the Remote Storage Server? This was an important issue as much of the performance depends on the rate of retrieval of the query submitted by a user. We propose the ‘Hierarchical Caching of the Data Blocks’, in *section (i)*, as a solution to this problem.

The next aspect we thought of was the issue of whether an intelligent guess of the Data Block needed by a user can be done? We have tried to address this issue in our *section (ii)* of our proposed work in the form of ‘Pre-fetching of the Data Blocks’.

Another aspect was the imbedded in the nature of wireless devices. These devices are prone to frequent disconnection from their Access Point. The issue that arose due to this was what about the user’s request during this period of disconnection? We put forward a solution to this aspect in the form of ‘Cache Co-operation among Mobile Clients’ in our *section (iii)* of our proposed work.

Related Work

The iSCSI protocol being a relatively new and extensively versatile protocol for the solution to the storage needs of today's corporate world, the development of different aspects of it in the last few years has been rapid. Also the Ad-Hoc networks have been a hot topic of research interest which has resulted into development of good number of protocols in the recent past. We discuss few of the important research papers which have been a motivating force behind our proposed work.

Fisheye State Routing:

FSR is a hierarchical routing protocol [11]. At each node of this routing the following information is preserved:-

- A list of the neighbouring nodes,
- A table for the topology of the network,
- A table for the next hop,
- A table for the distance of the route to the destination.

In the FSR accurate information concerning the distance and the path's quality about the immediate neighbourhood of the node is maintained. Such information about the far nodes is maintained in lesser updated form. This is achieved by periodic exchange of node information. While this periodic update in the case of neighbouring nodes (which is known as scope of the node) is high, for those nodes which lie beyond this scope the periodic updating is less. In this way the overhead of the routing table has been reduced, the accuracy of a packet being forwarded to a destination increases with the movement of the packet.

This type of routing helps in keeping the overheads associated in the routing table with increase in the size of network low; hence we have chosen this protocol for the Ad-Hoc routing of the packets in our proposed work.

Secured Internet Key exchange protocol:

The information on the internet is carried using the IP, which is a connectionless protocol and does not provide any privacy or security. In order to integrate security on per packet basis IETF developed the IPsec protocol. IPsec provides connectionless data integrity, authentication, data confidentiality, anti-replay protection and data origin authentication. It also provides traffic flow confidentiality to some extent. The IPsec is actually a suite of protocols which includes Authentication Header (AH), Encapsulation Security Payload (ESP), Internet Key Exchange (IKE), etc. IKE is used by two security gateways to exchange the key materials used by the IPsec functions.

There are many variants of the IKE. The original IKE based on the DH key agreement protocol is vulnerable to attack from the middle man. In 2004 the algorithm proposed by Haddad was made the standard for IKE. We proposed to use the faster method as given by *Chang et al.*[12].

The choice of this IKE algorithm is due to the fact that in our proposed work the data of the corporate (which can include mission critical data) transmission could be carried over the public networks also.

Specialized crypto-processing of Data Blocks:

In their work Design, Implementation and evaluation of Security in iSCSI-based Network Storage Systems, *Chaitanya et al.* [14] a unique procedure of encryption and decryption has been implemented which they term as '*Lazy Decryption*' and '*Lazy Authentication*'. The remote storage in case of iSCSI-based storage solutions doesn't use the data; the data is only used by the end user. Hence, for the data is stored at the remote storage location there is no need for decryption, (which has reached the remote storage location in encrypted form via the network).

Lazy Decryption: The iSCSI defines its packet which consists of a header and possible data is known as iSCSI PDU or iSCSI protocol data unit. The basic header is of 48 bytes followed by an extended additional header segment. The length of this additional header is given at the header at TotalAHSLength field. During read command from an initiator the symmetric key from the IPsec is used to decrypt the complete header including the additional segment which is considered to be a single block. During the write command from an initiator the Block Encryption Key (BEK) is generated for every block that is to be written to the server. This BEK are stored sequentially in a new Additional Header Segment

appended at the end of the iSCSI PDU header. The confidentiality of this is protected by the IPsec encryption key.

Overview of the iSCSI Protocol:

The iSCSI protocol specifies how to access SCSI storage devices over the TCP/IP networks. The interactions between the initiator and the target are based on sessions. The following is the description of iSCSI protocol features as given by J. Satran et al in [1] and J. Satran et al in [36].

- Sessions: An iSCSI session is a collection of TCP connections between the initiator and the target to transfer the SCSI commands and data between them. There can be multiple connections between an initiator and a target for better performance.
- iSCSI PDU: The iSCSI defines its own data units known as the iSCSI PDU. It consists of a Header and followed by data segment. These iSCSI PDUs are sent as contents of one or more TCP packets: Generally, the iSCSI PDU are of four types:
 - SCSI Command/ Response,
 - Data In/Out,
 - Ready to Transfer (R2T),
 - Login Request/ Response.

The SCSI command PDU is used to transfer a SCSI command from the initiator to the target. In the case of read data request, the target simply sends the data to the initiator while in case of write data request the target sends the complete data or partial data as requested by R2T command for updating it.

- Login: This is one of the most important aspects of the iSCSI protocol. Immediately after the establishment of TCP connections between an initiator and a target the login phase starts. In the login phase at first the initiator and the target authenticate each other and some other operational parameters are also exchanged.
- Naming: An iSCSI entity is identified by its name and not by its location. iSCSI uses URL like naming procedure which are unique in nature worldwide. This helps in proper routing of the packets to the intended destinations.

- Data Integrity: In addition to the checksum provided by TCP, iSCSI PDUs contains its own CRC as it is not desirable to have any error in the storage scenario. Initiator and targets may negotiate with each other whether to use the CRC or not.
- Security: If the organisation is using a physically secured separate network for such type of data movement then security is not a big issue, but in the case when transactions are taken place on general data network security becomes a must and challenging task. The IPSec protocol is used to provide security to the data of iSCSI on per packet basis.
- Direct Data Placement: usually in the case of TCP, the data is copied at several places like the buffer etc for proper functioning of the entire transfer. The iSCSI PDU headers contain sufficient information for proper functioning of the iSCSI HBA. Thus, the data can be directly placed hence better performance is resulted.

Proposed Work

The Basic Proposed Model:

As we have already mentioned, this proposed work has been inspired by the challenges that are being faced by the experts of the industry in providing their clients with much more data and information processing capabilities over the wireless networks in a cost effective manner.

In this proposed work, we propose the following architecture to achieve the perspective goal, the diagrammatic illustration of which is given below:--

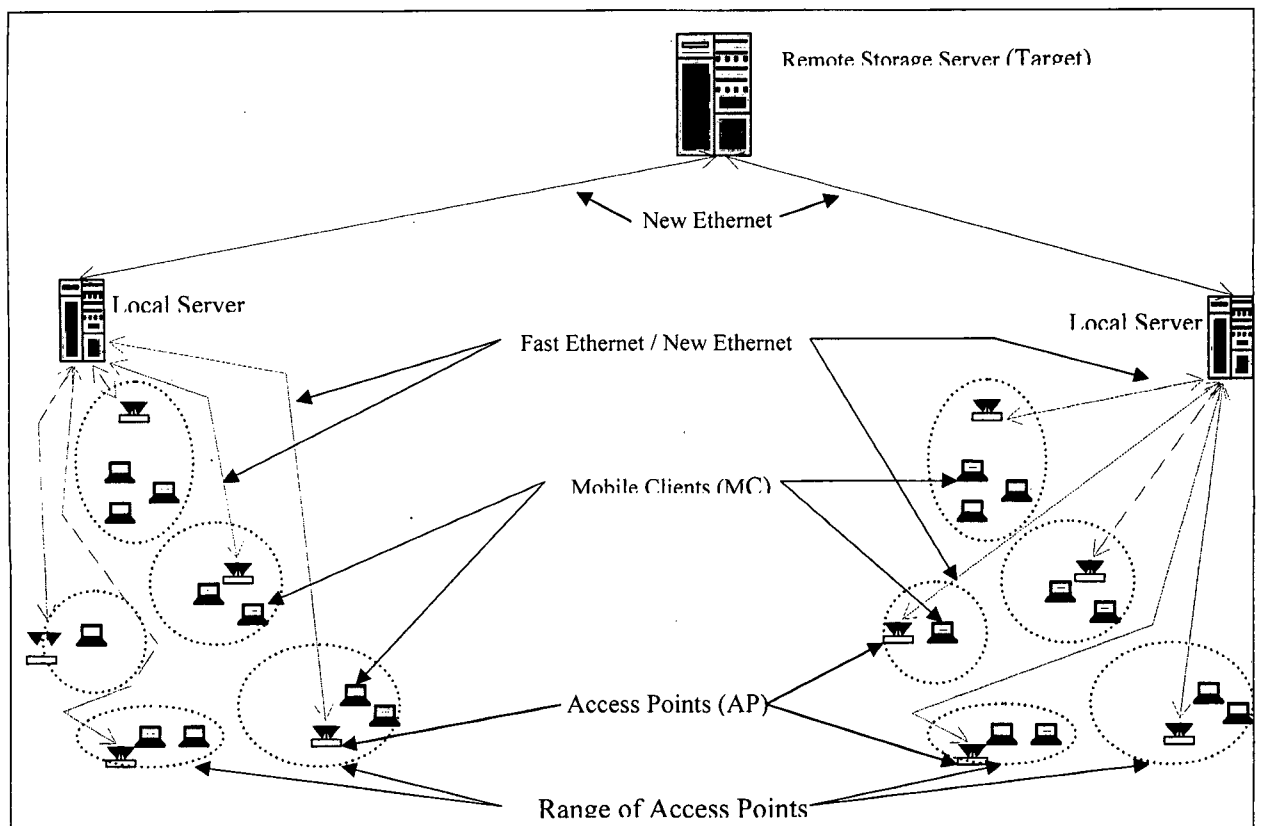



Figure No. 3.1: Illustrative representation of the proposed model

As already mentioned, the corporate organisations stores all its data including the sensitive data items at a Remote Storage Devices also known as Target. This Remote Storage Device is in turn connected to the Local Servers at the various offices of the organisation via New Ethernet also known as Gigabit LAN.

Each of the Local Server is equipped with low cost cache memory such as cache discs or cache disks etc. These Local servers cater to both wired connections and wireless connections. Since, this proposed work is confined to improving the performance over the wireless networks we do not consider the wired connections' performance enhancement aspects, though this proposed work will also improve the wired connections' performance.

The Local servers are connected to the various Access Points (AP, shown in the figure as ) either by Fast Ethernet or by New Ethernet. Each of the Access Points is equipped with cache memory in the form of NVRAMs.

These Access Points caters to various Mobile Clients (MCs, shown in the figure above by Blue coloured Laptops) in its proximity. The MCs also have an arrangement of Ad-Hoc Network among themselves. In this proposed work we have assumed that the MCs use Fisheye State Routing (FSR) for Ad-Hoc routing of packets among themselves.

This work is divided into four sections:--

- (i) Hierarchical Caching of the Data Blocks.
- (ii) Pre-fetching of the Data Blocks.
- (iii) Cache cooperation among mobile nodes by using Ad-Hoc networks.
- (iv) Encryption and Decryption of the Data Blocks.

Hierarchical Caching of the Data Blocks

As per the iSCSI protocol the data is stored in specialised storage devices at a remote location which may be hundreds of miles away from the user of the data. Thus, the data is stored at the Remote Storage Server called 'Target' and connected to the Local Server via the Gigabit Ethernet. Since the size of the data is huge, the transmission of the same from the target to the Mobile Client or user (called 'initiator' in iSCSI protocol) consumes considerable amount of time. Further, the link may not be always able to cater to a particular Mobile Client every time as it has to cater to the whole corporate organisation. In view of such constraint we propose the concept of Hierarchical Caching of the Data Blocks.

In Hierarchical Caching as already mentioned we use cache memory at the Local Server and Access

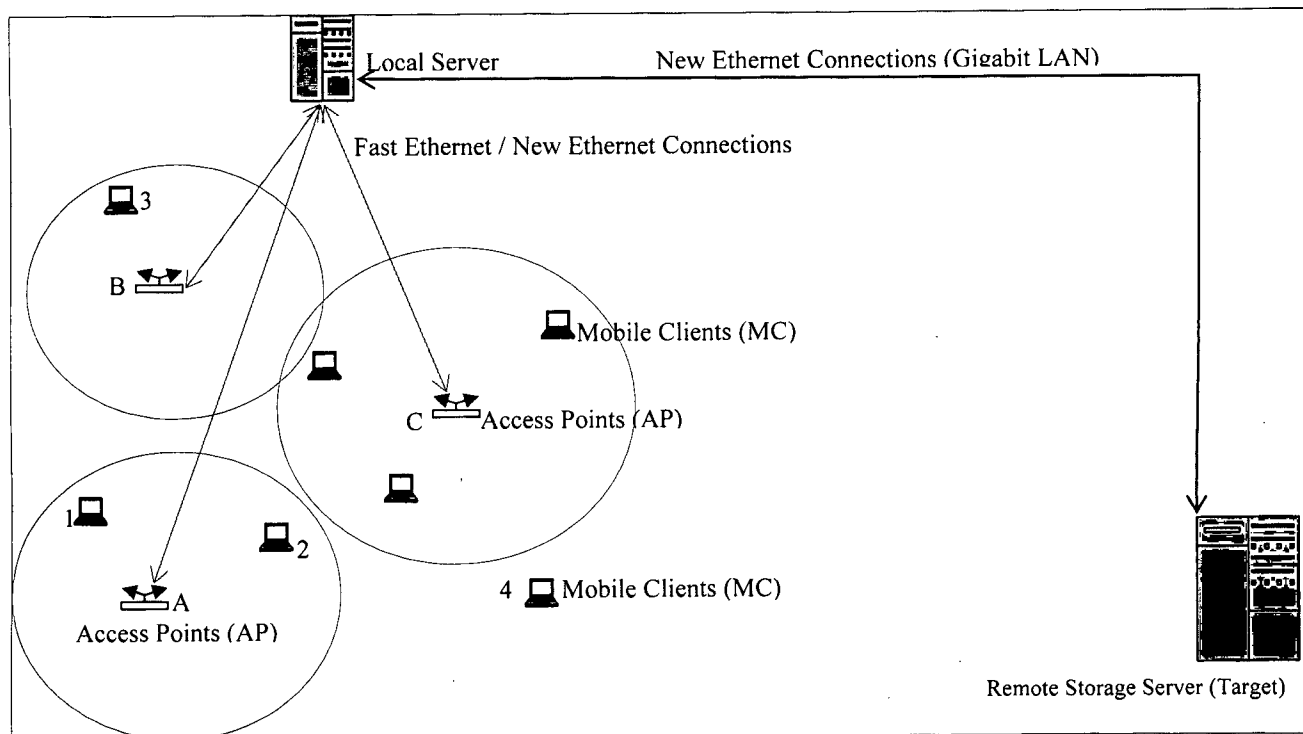


Figure No. 3.2: Hierarchical caching of data blocks

point level. This architecture helps in improving the performance of the iSCSI over the Wireless networks. When a Data Block is asked by a user the following events follows:-

- (i) The Mobile Client looks for the Data Block in its own cache, if it is not available in its cache it will send a request for the same to the Remote Storage Server via the Access Point it is connected.
- (ii) On receiving such a request from any Mobile Client, the Access Point at first looks for the same in its cache (NVRAMs), if the concerned Data Block is found in its cache the Mobile Client will be immediately served with the request. In case the requested Data Block is not available with the cache at this Access Point the request will be forwarded to the Local Server.
- (iii) As soon as such a request reaches the Local Server, an algorithm (which has been discussed in the section (ii) of this proposed work) starts working. Here also if the Data Block is available at the cache memory (Cache Discs or similar Memory Modules), the Mobile Client will get the service. In case the asked Data Block being not present in the cache at the Local Server the request will be forwarded to the Remote Storage Server or Target.

The Remote Storage Server caters to the request of the user if the Data Block asked by the user is a valid one or it is not being updated by any other user (who has sufficient rights to do so). On its way back of the requested Data Block the following activities will be performed:-

- (i) As soon as the Data Block (in encrypted form) arrives at the Local Server, the Local Server will put a copy of the Data Block (in encrypted form only) at its cache.
- (ii) Again as the Data Block arrives at the concerned Access Point, a copy of the Data Block will be stored at the cache memory of the Access Point in encrypted form.

In this way the hierarchical caching of the Data Blocks will be done. This hierarchical caching will help in reducing the latency and the time to retrieve asked Data Block. We now present an application scenario in which we put forward how the reduction in fetch time is achieved by using such technique.

Application Scenario:

A Mobile Client (MC), also known as 'initiator', say '1', asks for a data block 'X' for the first time (which is at present available only in encrypted format at the Remote Storage Device, also known as 'Target'), the request of this MC-1 is first processed by the Access Point (AP), say AP-A. Now since the AP-A doesn't possess this data block X at its cache so the request is forwarded to the local server, again as the cache of the local server also doesn't possess this data block hence the request is forwarded to the remote storage device. The corresponding data block is then retrieved from its storage location and sent back. On its way back the data block (in encrypted form) is stored first at the cache of the Local server and then at the cache of the corresponding access point i.e. AP-A.

Now consider a situation when MC-2 wants to access the same data block X, in such a situation as soon as the request for this data block X is received at the access point AP-A, this data block X is immediately sent to MC-2 as it was already available at its cache.

Now consider another situation when MC-3 (which is connected via another access point AP-B) asks for the same data block X. As AP-B doesn't possess any copy of this data block at its cache it forwards the request to the Local server. The Local server (as it possesses a copy of the required data block) sends it to the access point AP-B (which stores a copy of it at its cache).

In this way the hierarchical caching of the Data Block helps in improving the performance. We now present the mathematical model of the above proposed architecture.

Mathematical Model of the above illustration:

In this mathematical model we consider two aspects: —

- (i) When the data block being accessed is of fixed size.
- (ii) When the data block is of variable size.



TH-14680

(i). Fixed Size Data Block:

Let us consider the following assumptions:

the mean block size = S_{mean}

the time taken for a block of size S_{mean} to arrive from the target to the local server = $T_{\text{target access}}$

the time taken for a block of size S_{mean} to arrive from the local server to the access point = $T_{\text{AP access}}$

the time taken for a block of size S_{mean} to arrive from the AP to initiator (MC) = T_{wireless}

So, in the case when a MC asks for a document for the first time total time taken would = $T_{\text{total first}}$

$$\text{Where } T_{\text{total first}} = 2 * (T_{\text{target access}} + T_{\text{AP access}} + T_{\text{wireless}})$$

Similarly, in the case of the document being asked by an initiator (MC) being available with the access point AP cache = $T_{\text{Av. AP}}$

$$\text{Where } T_{\text{Av. AP}} = 2 * T_{\text{wireless}}$$

And, when the document being available with the local server cache = $T_{\text{local cache}}$

$$\text{Where } T_{\text{local cache}} = 2 * (T_{\text{AP access}} + T_{\text{wireless}})$$

We now consider a situation for a time interval τ , during this interval let there be N number of data blocks being accessed by any one MC, for simplicity, we assume that all the data block that are accessed are of almost same size (i.e. S_{mean}).

Suppose, out of these N data blocks

N_s be the number of blocks accessed from the target (remote storage device).

N_o be the number of blocks accessed from the Local server.

N_c be the number of blocks accessed from the Access Points (AP).

Therefore, we have

$$N = N_s + N_o + N_c$$

So, the total mean time $T_{\text{Total mean}}$ could be computed as:

$$T_{\text{Total mean}} = 1/N (\sum_{j=1}^{N_s} T_{\text{total first}} + \sum_{k=1}^{N_o} T_{\text{local cache}} + \sum_{l=1}^{N_c} T_{\text{Av. AP}})$$

Thus, the standard deviation σ from this mean time for a data block is given by

$$\sigma = \sqrt{ [1/N \{ \sum_{j=1}^{N_s} (T_{\text{total first}} - T_{\text{Total mean}}) + \sum_{k=1}^{N_o} (T_{\text{local cache}} - T_{\text{Total mean}}) + \sum_{l=1}^{N_c} (T_{\text{Av. AP}} - T_{\text{Total mean}}) \}^2]}$$

Probabilistic Estimation:

We now present the probabilistic estimation of the usage pattern by users of the proposed architecture. We assume that the arrival of request of a Data Block from a user follows the Poisson Distribution.

Let us assume the following:-

The arrival rate of the access request of the data block = λ_w

The rate of the data block being available from the cache at the MC = λ_K

The rate of the data block being available from the cache at the AP = λ_C

The rate of the data block being available from the cache at the local server = λ_o

The rate of the data block being available from the cache at the remote storage (target) = λ_S

Further, we consider a situation in an infinitesimal time duration t and $t+\Delta t$, which may be also thought as the time interval between the request for n^{th} data block and the $(n+1)^{\text{th}}$ data block.

Now, the following events are considered:

E_K = Event of the data block is being accessed from the cache at the MC.

E_C = Event of the data block is being accessed from the cache at the AP.

E_O = Event of the data block is being accessed from the cache at the Local server.

E_S = Event of the data block is being accessed from the cache at the remote storage (target).

Then the corresponding probability would be given by (where P_i is the probability associated with the event E_i):

Since, the inter-arrival density for the data block request = $\lambda_w e^{-\lambda_w t}$

And, the Probability of occurrence of data block being accessed from the cache at the MC by time t is given by $e^{-\lambda_k \cdot t}$

$$\text{So, } P_K = \int_0^{\infty} e^{-\lambda_k \cdot t} \lambda_w e^{-\lambda_w \cdot t} dt$$

And, the Probability of occurrence of data block being accessed from the cache at the AP by time t is given by $e^{-\lambda_c \cdot t} (1 - e^{-\lambda_k \cdot t})$

$$\text{So, } P_C = \int_0^{\infty} e^{-\lambda_c \cdot t} (1 - e^{-\lambda_k \cdot t}) \lambda_w e^{-\lambda_w \cdot t} dt$$

And, the Probability of occurrence of data block being accessed from the cache at the Local server by time t is given by $(1 - e^{-\lambda_c \cdot t})(1 - e^{-\lambda_k \cdot t}) e^{-\lambda_o \cdot t}$

$$\text{So, } P_o = \int_0^{\infty} e^{-\lambda_o \cdot t} (1 - e^{-\lambda_c \cdot t}) (1 - e^{-\lambda_k \cdot t}) \lambda_w e^{-\lambda_w \cdot t} dt$$

And, the Probability of occurrence of data block being accessed from the cache at the Local server by time t is given by $(1 - e^{-\lambda_c \cdot t})(1 - e^{-\lambda_k \cdot t})(1 - e^{-\lambda_o \cdot t}) e^{-\lambda_s \cdot t}$

$$\text{So, } P_s = \int_0^{\infty} e^{-\lambda_s \cdot t} (1 - e^{-\lambda_c \cdot t})(1 - e^{-\lambda_k \cdot t})(1 - e^{-\lambda_o \cdot t}) \lambda_w e^{-\lambda_w \cdot t} dt$$

Thus, the probability P_{success} of successful accessing of a data block is given by:

$$P_{\text{success}} = 1/N (P_s \cdot N_s + P_o \cdot N_o + P_c \cdot N_c)$$

(ii). Variable Data Block Size

We assume the following:

Time taken for the p^{th} block of size S_p to arrive from the target to the local server = $T_{\text{target access, p}}$

Time taken for the p^{th} block of size S_p to arrive from the local server to the AP = $T_{\text{AP access, p}}$

Time taken for the p^{th} block of size S_p to arrive from the AP to initiator (MC) = $T_{\text{wireless, p}}$

So, in the case when a MC asks for a document for the first time total time taken would = $T_{\text{total first, p}}$

$$\text{Where } T_{\text{total first, p}} = 2 * (T_{\text{target access, p}} + T_{\text{AP access, p}} + T_{\text{wireless, p}})$$

Similarly, in the case of the document being asked by an initiator (MC) being available with the access point AP cache = $T_{Av. AP, p}$

$$\text{Where } T_{Av. AP, p} = 2 * T_{wireless, p}$$

And, when the document being available with the local server cache = $T_{local cache, p}$

$$\text{Where } T_{local cache, p} = 2*(T_{AP access, p} + T_{wireless, p})$$

Again we consider the total number of blocks accessed in some time interval τ be N such that:

N_s be the number of blocks accessed from the target (remote storage device).

N_o be the number of blocks accessed from the Local server.

N_c be the number of blocks accessed from the Access Points (AP).

$$\text{Thus, } N = N_s + N_o + N_c$$

So, the total mean time $T_{Total mean}$ could be computed as:

$$T_{Total mean} = 1/N (\sum_{j=1}^{N_s} T_{total first, j} + \sum_{k=1}^{N_o} T_{local cache, k} + \sum_{l=1}^{N_c} T_{Av. AP, l})$$

In addition, the standard deviation σ from this mean time for a data block is given by

$$\sigma = \sqrt{ [1/N \{ \sum_{j=1}^{N_s} (T_{total first, j} - T_{Total mean}) + \sum_{k=1}^{N_o} (T_{local cache, k} - T_{Total mean}) + \sum_{l=1}^{N_c} (T_{Av. AP, l} - T_{Total mean}) \}^2] }$$

Probabilistic Estimation:

We now assume that the pattern of accessing the data blocks by an MC follows *Poisson Process*.

The arrival rate of the access request of the p^{th} data block = $\lambda_{w,p}$

The rate of the p^{th} data block being available from the cache at the MC = $\lambda_{K,p}$

The rate of the p^{th} data block being available from the cache at the AP = $\lambda_{C,p}$

The rate of the p^{th} data block being available from the cache at the local server = $\lambda_{O,p}$

The rate of the p^{th} data block being available from the cache at the remote storage (target) = $\lambda_{S,p}$

Further, we consider a situation in an infinitesimal time duration t and $t+\Delta t$, which may be also thought as the time interval between the request for n^{th} data block and the $(n+1)^{\text{th}}$ data block.

Now, the following events are considered:

E_K = Event of the p^{th} data block is being accessed from the cache at the MC.

E_C = Event of the p^{th} data block is being accessed from the cache at the AP.

E_O = Event of the p^{th} data block is being accessed from the cache at the Local server.

E_S = Event of the p^{th} data block is being accessed from the cache at the remote storage (target).

Then the corresponding probability would be given by (where P_i is the probability associated with the event E_i):

Since, the inter-arrival density for the p^{th} data block request

$$= \lambda_{w,p} e^{-\lambda_{w,p} t}$$

And, the Probability of occurrence of p^{th} data block being accessed from the cache at the MC by time t is given by

$$e^{-\lambda_{k,p} \cdot t}$$

$$\begin{aligned} \text{So, } P_K &= \int_0^{\infty} e^{-\lambda_{k,p} \cdot t} \lambda_{W,P} e^{-\lambda_{w,p} \cdot t} dt \\ &= \lambda_{W,P} \int_0^{\infty} e^{-\lambda_{k,p} \cdot t} e^{-\lambda_{w,p} \cdot t} dt \\ &= \lambda_{W,P} \int_0^{\infty} e^{-(\lambda_{w,p} + \lambda_{k,p})t} dt \\ &= \lambda_{W,P} / (\lambda_{W,P} + \lambda_{K,P}) \end{aligned}$$

The Probability of occurrence of p^{th} data block being accessed from the cache at the AP by time t is given by

$$e^{-\lambda_{c,p} \cdot t} (1 - e^{-\lambda_{k,p} \cdot t})$$

$$\begin{aligned} \text{So, } P_C &= \int_0^{\infty} e^{-\lambda_{c,p} \cdot t} (1 - e^{-\lambda_{k,p} \cdot t}) \lambda_{W,P} e^{-\lambda_{w,p} \cdot t} dt \\ &= \lambda_{W,P} \int_0^{\infty} \{ e^{-(\lambda_{c,p} + \lambda_{w,p})t} - e^{-(\lambda_{c,p} + \lambda_{w,p} + \lambda_{k,p})t} \} dt \\ &= \lambda_{W,P} \{ 1 / (\lambda_{C,P} + \lambda_{W,P}) - 1 / (\lambda_{C,P} + \lambda_{W,P} + \lambda_{K,P}) \} \\ &= \lambda_{W,P} / (\lambda_{C,P} + \lambda_{W,P}) - \lambda_{W,P} / (\lambda_{C,P} + \lambda_{W,P} + \lambda_{K,P}) \\ &= \lambda_{W,P} \cdot \lambda_{K,P} / \{ (\lambda_{C,P} + \lambda_{W,P}) \cdot (\lambda_{C,P} + \lambda_{W,P} + \lambda_{K,P}) \} \end{aligned}$$

Probability of occurrence of p^{th} data block being accessed from the cache at the Local server by time t is given by

$$(1 - e^{-\lambda_{c,p} \cdot t})(1 - e^{-\lambda_{k,p} \cdot t}) e^{-\lambda_{o,p} \cdot t}$$

$$\begin{aligned} \text{So, } P_O &= \int_0^{\infty} e^{-\lambda_{o,p} \cdot t} (1 - e^{-\lambda_{c,p} \cdot t})(1 - e^{-\lambda_{k,p} \cdot t}) \lambda_{w,p} e^{-\lambda_{w,p} \cdot t} dt \\ &= \lambda_{w,p} \int_0^{\infty} \left\{ e^{-(\lambda_{o,p} + \lambda_{w,p})t} - e^{-(\lambda_{o,p} + \lambda_{w,p} + \lambda_{k,p})t} - e^{-(\lambda_{o,p} + \lambda_{w,p} + \lambda_{c,p})t} + e^{-(\lambda_{o,p} + \lambda_{w,p} + \lambda_{c,p} + \lambda_{k,p})t} \right\} dt \\ &= \lambda_{w,p} \left\{ 1 / (\lambda_{o,p} + \lambda_{w,p}) - 1 / (\lambda_{o,p} + \lambda_{w,p} + \lambda_{k,p}) - 1 / (\lambda_{o,p} + \lambda_{w,p} + \lambda_{c,p}) + \right. \\ &\quad \left. 1 / (\lambda_{o,p} + \lambda_{w,p} + \lambda_{c,p} + \lambda_{k,p}) \right\} \end{aligned}$$

Probability of occurrence of p^{th} data block being accessed from the cache at the Local server by time t is given by

$$(1 - e^{-\lambda_{c,p} \cdot t})(1 - e^{-\lambda_{k,p} \cdot t})(1 - e^{-\lambda_{o,p} \cdot t}) e^{-\lambda_{s,p} \cdot t}$$

$$\begin{aligned} P_S &= \int_0^{\infty} e^{-\lambda_{s,p} \cdot t} (1 - e^{-\lambda_{c,p} \cdot t})(1 - e^{-\lambda_{k,p} \cdot t})(1 - e^{-\lambda_{o,p} \cdot t}) \lambda_{w,p} e^{-\lambda_{w,p} \cdot t} dt \\ &= \lambda_{w,p} \int_0^{\infty} \left\{ e^{-(\lambda_{s,p} + \lambda_{w,p})t} - e^{-(\lambda_{s,p} + \lambda_{w,p} + \lambda_{c,p})t} - e^{-(\lambda_{s,p} + \lambda_{w,p} + \lambda_{k,p})t} - \right. \\ &\quad e^{-(\lambda_{s,p} + \lambda_{w,p} + \lambda_{o,p})t} + e^{-(\lambda_{s,p} + \lambda_{w,p} + \lambda_{c,p} + \lambda_{k,p})t} + e^{-(\lambda_{s,p} + \lambda_{w,p} + \lambda_{c,p} + \lambda_{o,p})t} + \\ &\quad \left. e^{-(\lambda_{s,p} + \lambda_{w,p} + \lambda_{k,p} + \lambda_{o,p})t} - e^{-(\lambda_{s,p} + \lambda_{w,p} + \lambda_{c,p} + \lambda_{k,p} + \lambda_{o,p})t} \right\} dt \end{aligned}$$

$$\begin{aligned}
&= \lambda_{W,P} \{ 1 / (\lambda_{S,P} + \lambda_{W,P}) - 1 / (\lambda_{S,P} + \lambda_{W,P} + \lambda_{C,P}) - 1 / (\lambda_{S,P} + \lambda_{W,P} + \lambda_{K,P}) - \\
&\quad 1 / (\lambda_{S,P} + \lambda_{W,P} + \lambda_{O,P}) + 1 / (\lambda_{S,P} + \lambda_{W,P} + \lambda_{C,P} + \lambda_{K,P}) + 1 / (\lambda_{S,P} + \lambda_{W,P} + \\
&\quad \lambda_{C,P} + \lambda_{O,P}) + 1 / (\lambda_{S,P} + \lambda_{W,P} + \lambda_{K,P} + \lambda_{O,P}) - 1 / (\lambda_{S,P} + \lambda_{W,P} + \lambda_{C,P} + \lambda_{K,P} + \lambda_{O,P}) \}
\end{aligned}$$

This is how the Hierarchical caching of the Data Blocks will improve the performance of the IP Storages over the wireless networks.

Pre-fetching of the Data Blocks

We now put forward the concept of 'Pre-fetching of the Data Blocks'. As the data is no longer stored permanently at the Local Server, we propose to use these powerful computers for enhancing the performance of IP Storages. The Local Storages will run an algorithm which keeps track of the performance of IP Storages. The Local Storages will run an algorithm which keeps track of the access pattern of the Data Blocks fetched by both the wired terminals and the wireless ones. We propose that this data be kept at the Local Server in a directed graph format. The data at each of the node of such a graph will contain the name of the accessed resource (e.g. file name) and the location of the Data Block at the Remote Storage Device; it won't contain the Data Block itself.

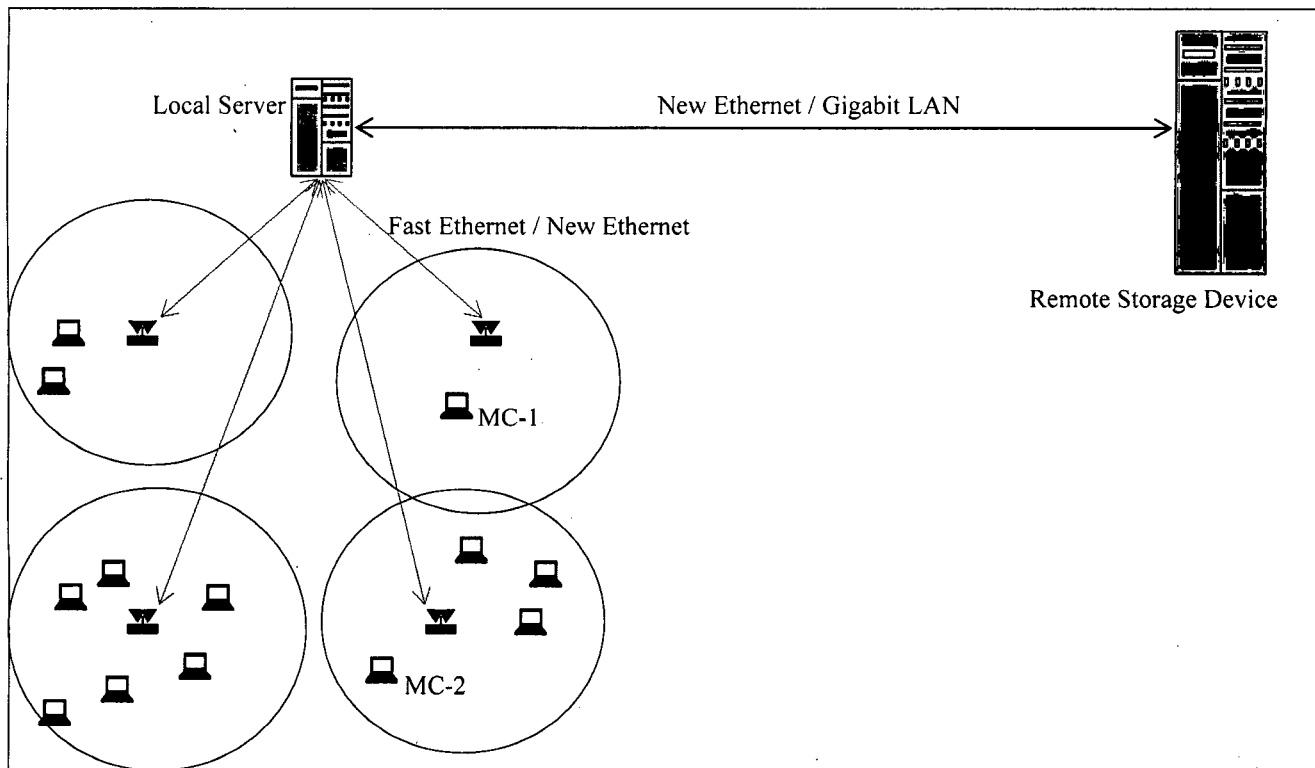


Figure No. 3.3: Pre-fetching of the Data Blocks

Application Scenario:

Assume that the following directed graph (Fig: 3.4) is created at the Local Server. This graph has been created by keeping a trail over the request submitted by a particular client. For example:- MC-1 as shown in the fig: 3.3 asks for Data Block 4(say), this information is stored in the local server by creating a node, then again it is recorded that MC-1 asks for Data Block 3 and Data Block 8 (within a very short duration of time), at this moment a link from the node 4 will be created that connects node 3 and a similar link from node 4 to node 8 will be also created. After sometime if the same MC-1 asks for Data Block 9, and then links from both nodes 3 and 8 to node 9 will be created.

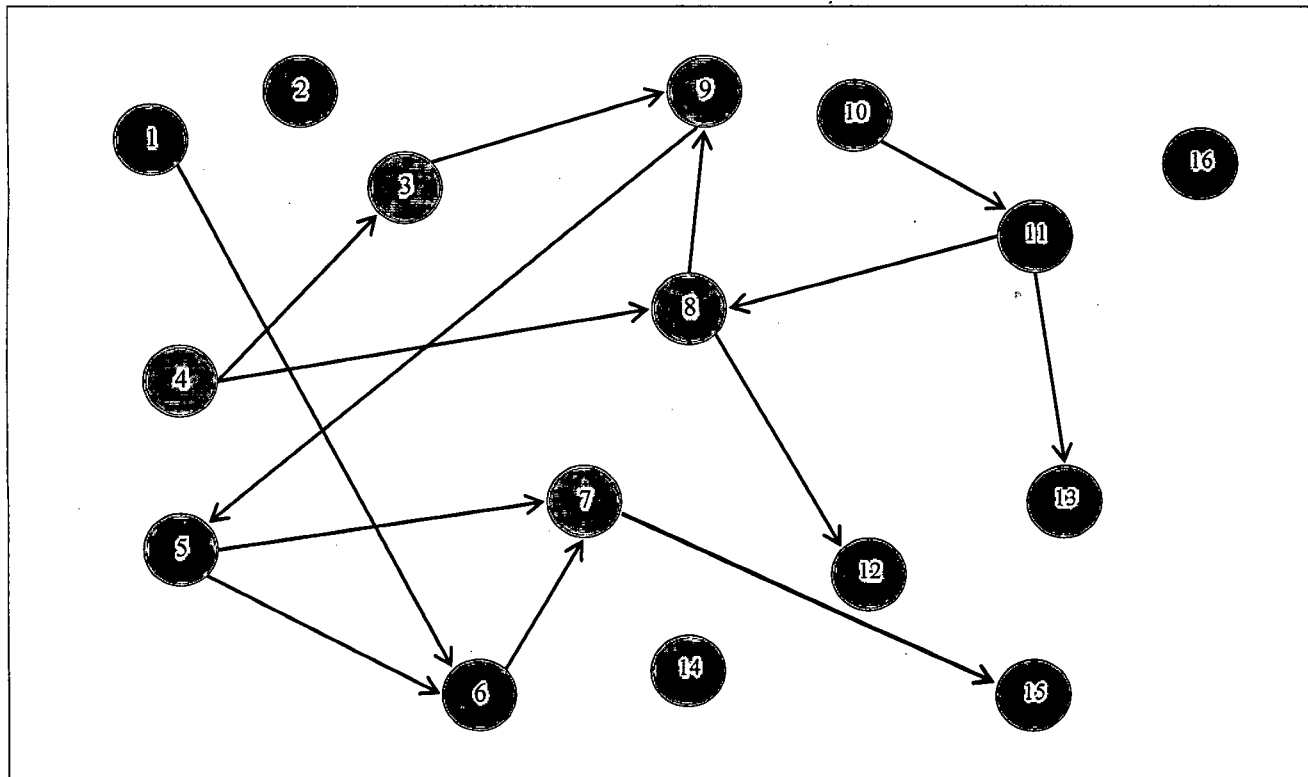


Figure No. 3.4: Access history of pattern of fetching of Data Blocks

In this proposed work we have assumed that all the Mobile clients have to get themselves authenticated themselves when they log in for the first time or if the period of disconnection is too large, also each of the wired clients have to get themselves

authenticated on first log-in. Each of such session will be treated as separate sessions by this graph generating algorithm. In addition to this the algorithm also provides time slots (which will be separate for each client logged in during that time) for the purpose of maintaining the graph.

(i) **Directed Links & Node generation:**

If a Client have accessed a Data Block, say X, and then after sometime if the same Client sends request to access two Data Blocks Y and Z within a very short duration (say τ_s) then directed links from the node (corresponding to X) to

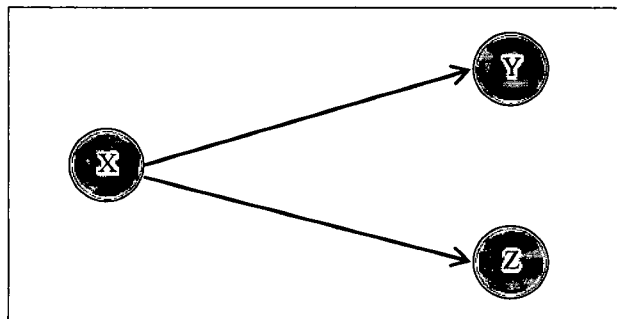


Figure No. 3.5: Generation of Directed Links during short duration

the nodes (corresponding to Y and Z) are created. Thus, two links are formed from the node corresponding to X.

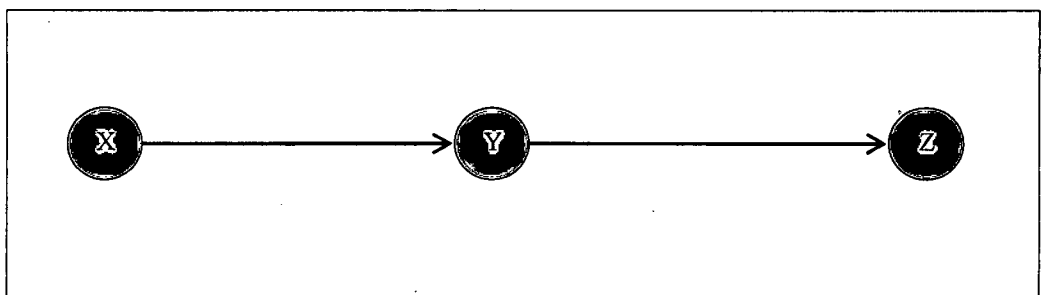


Figure No. 3.6:-Generation of Directed Links when time duration is more than τ_s

But, if the patterns of accessing the Data Blocks have greater time duration than this duration τ_s then the creation of links in the graph is somewhat different. Say, if Y is accessed and then after time duration greater than τ_s Z is accessed, in this case first a directed link from node (corresponding to) X to the node (corresponding to) Y is created and then from this node a directed link to the node (corresponding to) Z is created. One exception to this circumstance would be when the client asks for a Data Block for the first time.

Now when some other client access the Data Block Y new node is not created but the existing node is used. If this Client accesses some other Data Block, say W, then at first the algorithm will look for a node (corresponding to) W, if such a node exist then a directed Link from node (corresponding to) Y to this node is created, otherwise, a node (for W) is first created and then the directed link is created.

(ii) **Directed Links deletion:**

The number of hits for each directed link irrespective of the client will be maintained for a specified duration of time, say one week or may be a fortnight. The algorithm will also have a threshold value, set by some administrator. This threshold value will be used by the algorithm to determine whether a directed link should be continued to be stored in the appropriate data structure or should be deleted.

For example:- Consider in the fig: 3.4 if the number of hits to the directed link of node 11 to node 8 is below the threshold (set by the administrator) during the requisite time duration (a week or a fortnight as set by the administrator) then this directed link will be deleted, while the nodes will remain.

(iii) **Node invalidation & deletion:**

A node will be invalidated or deleted from this graph if one the following events occur.

- (a) Suppose a Data Block is updated/augmented by any one of the clients, this client will then sends a small packet called 'write-info' to the Remote Storage Device. On receiving such a packet the Remote Storage Device will broadcast an updating message to all the Local

Servers about the concerned Data Block. On reception of any such message from the Remote Storage Device the following events will be scheduled at the Local Server level.

(i) The node corresponding to this Data Block, if present, will have its entire links, both incoming and outgoing, deleted and then the node itself will be deleted. We propose to delete this node because when the client updates/ augments the Data Block it may change its size drastically, which in turn will force the disk space management software at the Remote Storage Device to relocate this Data Block.

(ii) When such a broadcast is received from the Remote Storage Device all the copies of the concerning Data Block from the different caches are immediately removed.

(b) Suppose a Data Block is deleted from the Remote Storage Devices, (this could be done only by clients who enjoy sufficient rights for such type of activities). In this case also the corresponding node along with its entire incoming and outgoing links will be deleted.

Finally, the Local Server enacts on the application related to the graph only either when it sees that the incoming request from a MC is not available in its cache or when it receives a request related to 'write-info' from the Remote Storage Device.

The Pre-fetching algorithm will be using the above generated graph to pre-fetch the appropriate Data Blocks and storing a copy of this pre-fetched Data Block in the large inexpensive cache of the Local server. Also only one Data Block after a request from a client will be fetched.

Application Scenario:

Consider a situation when this system related to the graph has become stable i.e. the number of nodes in the graph has reached a suitable level for the proper working of our pre-fetching algorithm. Suppose the MC-1 asks for the Data Block 5 (assuming that Data Block 5 is at present not available in the cache at the AP), so the request reaches at the Local Server level.

At the Local Server level if the Data Block 5 is available then it is given to the AP for further transmission, and if it is not available then its request is forwarded to the Remote Storage Device, but in any case our pre-fetching algorithm will start working.

This algorithm learns, from the already generated graph of access pattern of the Data Blocks, that usually when a client asks for the Data Block 5 it will also ask for the Data Blocks 6 and 7. So, the algorithm looks for the availability of these Data Blocks in the cache of its Local Server, if any of them is not present in the cache then it immediately sends a request of the corresponding Data Block to the Remote Storage Device, and after receiving the Data Block it stores the copy in the cache. At this moment of time the pre-fetching algorithm will not do any such work related to Data Block 15 even though it is connected to Data Block 7, this is due to the fact the algorithm works only for the first directed link from the concerned node.

Now, suppose that any Mobile Client accesses the Data Block 6 (assuming that it is not available at the concerned AP). This Mobile Client will get the service immediately as the Data Block is already present with the cache at Local server. Now, suppose that MC-2 asks for the Data Block 7 since at Local Server level it is available so the request will be immediately served, also as the next directed link from the node corresponding to Data Block 7 is that of Data Block 15, so the pre-fetching algorithm will start working as mentioned above.

Now, if a MC accesses the Data block 15, and then within the same session it sends a request for another Data Block (assuming the requested Data Block doesn't exist at the concerned AP) graph algorithm will start working, as soon as the request reaches the Local Server (irrespective of the fact whether the requested Data Block is present in its cache or not), to update the graph and the directed links.

This is how the pre-fetching algorithm will work to improve the performance of the IP Storages over the wireless networks.

Cache Co-operation among Mobile Clients

The inspiration of this section of our proposed work lies in the nature of wireless communications. The Mobile clients working in any wireless environment are prone to frequent disconnection, non accessibility of wireless channel, hard and soft hand-offs etc. Each of the following puts immense road block in the performance of the IP Storages. In this section we mainly consider the scenarios pertaining to disconnection of MCs from their nearest APs and non accessibility of the wireless channel. We assume that the MCs form an Ad-Hoc network to co-operate among themselves. It is also assumed that the MCs use *Fisheye state routing* for this Ad-Hoc network.

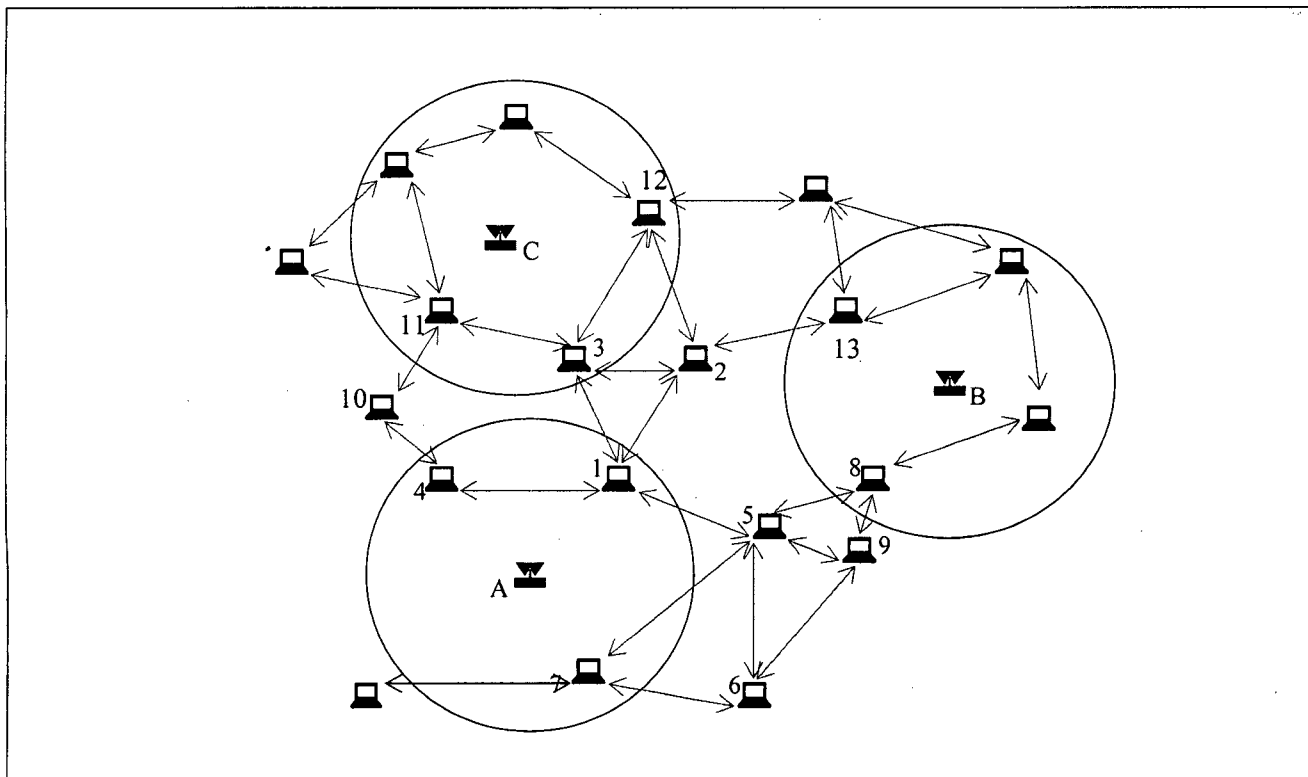


Figure No. 3.7: The Ad-Hoc Network connections among the MCs

In the above figure the Ad-Hoc connections among the MCs is shown by Red arrows. It further assumed that the scope of each of the MCs is one hop.

When a user of a disconnected (from its access point) MC asks for a particular Data Block which is not present in the MC's cache, the MC will activate its Ad-Hoc channel to get the Data Block. The MC will at first broadcast a request packet in its first scope (which may be one at Hop or two Hops or more as according to the implementation of *Fisheye state Routing*). In addition to other fields this packet will have two fields for the following: –

- (i) One which contains the number of hops travelled by the packet,
- (ii) Another will contain the Time-To-Live for the packet.

The Time-To-Live field helps in restricting the response time before the parent MC sends another such request packet for the same Data Block if in the case it does not receive any Return-Reply packet from any other MC.

Now, the MCs which receives such packets at first looks in their respective cache for the Data Block, if the Data Block is present in the Local cache a Return-Reply packet is sent to the MC which had made the request via the same path. But in the situation it is not available the MC will react in the following fashion.

- (i) At first, it will look whether it is connected to any Access Point or not and whether the link to this Access Point is free or not (it may be possible that the link to this Access Point is being used by itself). If it is connected and the link is free this MC will send a request for this Data Block to the Access Point, and also send a Return-Reply packet to the MC which had made the request for the Data Block.
- (ii) If it is not connected to an Access Point then it broadcast the request packet further in its scope provided the Client-Power (discussed later) is greater than the threshold value (pre-determined by the concerning Mobile Client).

- (iii) And, in case it is connected to the Access Point but the link is not free it will wait for the link to be free or wait for a time period equal to the Time-To-Live of the packet, whichever is lesser.

The Client-Power (κ_{MCP}) will be determined by the following:

- (a) Type of Mobile Device (Laptops, Notepads, Personal Digital Assistants etc).
- (b) The amount of Battery Power left.
- (c) The No. of Hops the initial requester (of the particular Data Block) is away.
- (d) The Processing capabilities of the MC.

A Mobile Client with lesser κ_{MCP} will not broadcast the request packet further in its scope.

The Client-Power (κ_{MCP}) will be computed in the following manner:-

- (a) The field corresponding to the 'type of the Mobile Client' will be assigned a number between 0 to 1. The assignment scheme will follow the following pattern
 - (i) A static device, like a desktop, will be assigned the highest value 1,
 - (ii) A PDA or a device with similar capabilities will be assigned a smaller number (say between 0.1 ~ 0.35) depending on the capabilities of the device,
 - (iii) A Laptop will be assigned a moderate number (say between 0.5 ~ 0.8) depending upon the device capabilities.
- (b) The amount of battery power left is computed directly from the system resources.
- (c) The number of Hops the Request packet has travelled is determined from the respective field in the packet itself.
- (d) The processing power of the Mobile Client is determined by the hardware features of the Mobile Client.

The Return-Reply packet, in addition to other standard fields, will contain the following fields to help the initial requester of the Data Block to choose the path of data transfer.

- (i) Type of the Mobile Client in the path: – in this field the minimum of all the value of ‘Type of Mobile Client’ in that particular path will be stored. This will be achieved in the following manner: when a Return-Reply packet starts to travel its way to the MC, which had put the request for the concerned Data Block, each of the MC in its way will compare the value of this field with it’s own value, if it’s own value is lesser than the existing value then it replaces the value with it’s own value, otherwise it does not makes any change. In this way when the packet reaches its destination MC, the destination MC will be capable to read the minimum of the ‘Type of Mobile Client’ in that particular path. Hence the destination MC will try to avoid a path which contains a PDA or device with similar capabilities.
- (ii) In a similar manner another field will contain the minimum of all the Client-Power (κ_{MCP}) lying in the path. This field is very useful in determining the optimum path, in case there is a tie between two or more paths, by the destination MC.
- (iii) Another important field of this packet will be containing the complete path information that the Return-Reply packet will traverse.

The MC (say parent MC) which had put the request packet of the Data Block may receive more than one Return-Reply packet from different MCs or from same MC with different paths. Now, this parent MC will have to choose an optimum path for the data transfer. The choice of the optimum path is depends upon the two fields of Return-Reply packets discussed above.

The following steps are followed by the parent MC to choose the optimum path:

- (i) The parent MC identifies the shortest paths (least number of hops) among all the paths available,

- (ii) From these paths, the paths in which one or more Mobile Client is a PDA or a similar mobile device with limited capabilities will be ruled out.
- (iii) Out of the remaining paths the path with highest κ_{MCP} is selected.

If none of the path qualifies for data transfer, the parent MC will look for the optimum path among the next longer (more number of hops) paths.

Also, in case of link breakage or non availability of the path the parent MC moves to next available path.

The mathematical computation of κ_{MCP} :

Let the 'Type of device' be = T_{Device} ,

Let the amount of battery power of the Mobile Client be = B_{Power} ,

Let the Number of Hops travelled by the packet be = N_{Packet} ,

Let the Processing power of the Mobile Client be = $P_{ProPower}$.

$$\text{Thus, } \kappa_{MCP} = \{T_{Device} \times \lambda \cdot B_{Power} \times \mu \cdot P_{ProPower}\} / N_{Packet}$$

where λ and μ are constants for making the quantities B_{Power} and $P_{ProPower}$ into numeric values compatible with other Mobile Clients.

Application Scenario:

Consider a situation when the Mobile client 1 is not connected to its nearest Access Point A (this may be due to the fact that the Access Point is catering to a large data transfer to some other Mobile Client or may be down due to some technical reasons), then in such a situation this MC 1 will immediately look for the user's request (say Data Block X) among the neighbouring Mobile Clients.

The Mobile Client 1 will broadcast the required Data Block X request among the Mobile Clients lying in its scope (which is assumed to be first hop neighbours). Thus, such request will be received by MCs 2 to 5. If any of these MCs have this Data Block

X then a unicast Return-Reply packet is sent back to the MC 1. If the Mobile Client doesn't have the required Data Block X (and is also disconnected) then further broadcast of the request will depend upon the Client-Power (κ_{MCP}) of the Mobile Client.

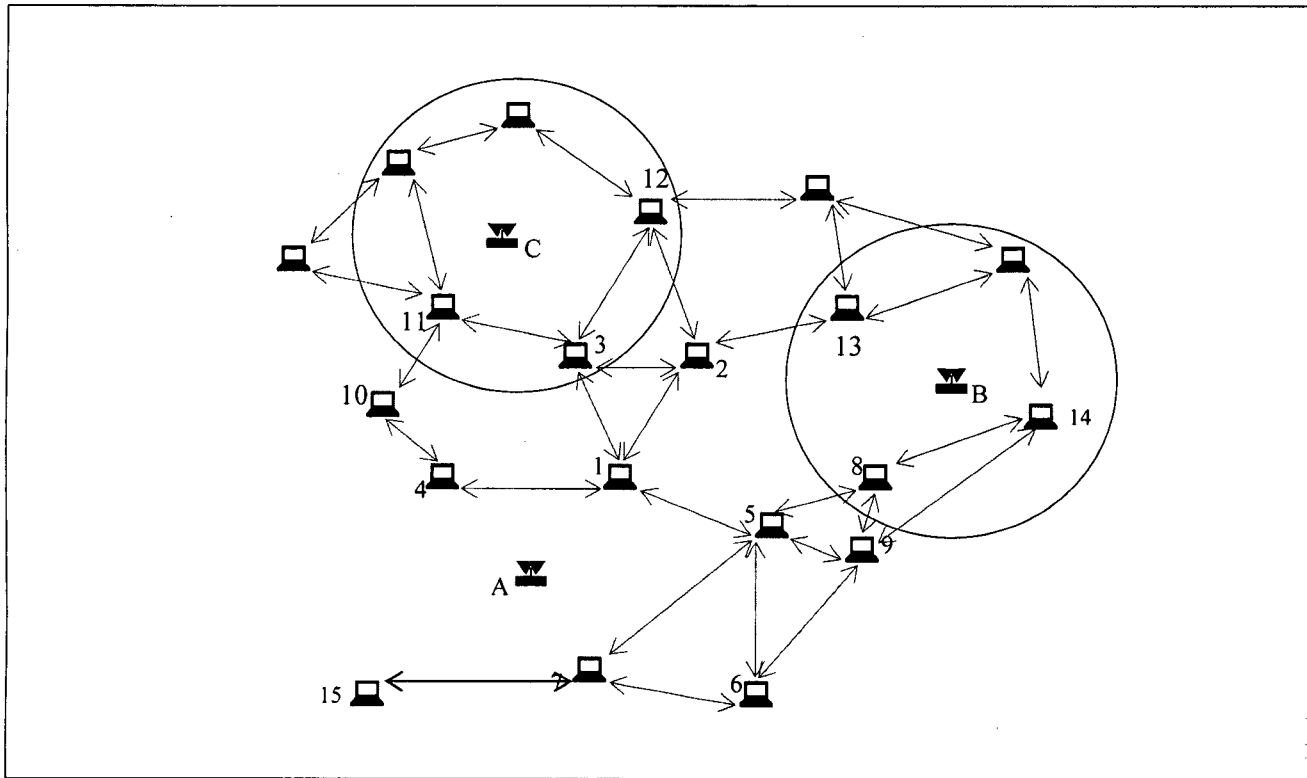


Figure No. 3.8: The Network from the point of view of Mobile Client 1

Now the MC-2 to MC-5 all receives the request for the Data Block from the MC-1. Suppose the Mobile Clients 2, 4 and 5 all doesn't have the requested Data Block X in their respective cache, so they forward the request to the next hop in their respective neighbourhood. The Mobile client 3 will act in a different way as it is connected to an Access Point C, if this MC 3 is not using its link for some kind of data transfer it will send back a Return-Reply packet to the MC 1 along the same path. But, in the case this MC 3 is using the particular link for its own data transfer, it will store the request packet in its buffer until the Time-To-Live duration of the packet is not over, if the link is freed before the expiry of the Time-To-Live of the request packet the Return-Reply packet is sent.

Again, let's assume that

- (i) the MC 3 is not in a position to provide the service,
- (ii) the required Data Block X is available with the MC 15,
- (iii) the MC 13 has the requested Data Block X and is provide the service,
- (iv) the MC 8 is a PDA,
- (v) the MC 14 is not using its link and may retrieve the Data Block X from its Access Point B.

Thus, the following Return-Reply packet will arrive at the MC 1

- (i) from MC 15 with route $15 - 7 - 5 - 1$,
- (ii) from MC 14 with route $14 - 8 - 5 - 1$,
- (iii) from MC 14 with route $14 - 9 - 5 - 1$,
- (iv) from MC 13 with route $13 - 2 - 1$.

On reception of the Return-Reply packets the MC 1 will take the decision of the best route keeping in view the number of hops, K_{MCP} and the type of the Mobile Clients in the path.

In the above scenario the first choice will be the option (iv) as it has the minimum number of hops. Now, assume that this path $13 - 2 - 1$ breaks up to some reason (say the Mobile Client 2 has moved away or has gone down), in such a scenario the MC 1 will look for the next better path which is path no. (i) that is $15 - 7 - 5 - 1$. Further, if this link also goes down, the MC 1 looks for the next better path from the available two options (ii) and (iii). Out here option number (ii) will be ruled out in favour of option number (iii), as in the path of the option (ii) we have the Mobile Client 8 which is a PDA.

The mathematical model:

Assume that in the above system a Mobile Client MC_0 is observed for certain time interval (say ζ). It is being assumed that during this time interval ζ the MC_0 is not connected with any of the Access Point in its proximity.

Let us assume the following:

The No. of Data Block request made by the user of MC₀ be = N

The field corresponding to the Client-Power of pth Return-Reply packet be = κ_{MCP, p}

Thus, the mean ($\mu_{\text{Client-Power}}$) of the Client-Power of the N paths that the MC₀ accessed during the time interval ζ is given by:

$$\mu_{\text{Client-Power}} = \frac{1}{N} \left\{ \sum_{j=1}^N \kappa_{\text{MCP}, j} \right\}$$

And, hence the standard Deviation $\sigma_{\text{Client-Power}}$ of the corresponding will be:

$$\sigma_{\text{Client-Power}} = \sqrt{\left[\frac{1}{N} \left\{ \sum_{j=1}^N (\kappa_{\text{MCP}, j} - \mu_{\text{Client-Power}})^2 \right\} \right]}$$

We have already presumed that the access of Data Block by a user follows the Poisson Process; now let's further assume that the disconnection of a MC from its Access Point also follows Poisson Process.

The arrival rate of the access request of the pth data block = $\lambda_{W,P}$

The rate of the pth data block being available from the cache at the MC = $\lambda_{K,P}$

The rate of disconnection of the MC from the Access Point when the pth Data Block is accessed = $\lambda_{X,P}$

We have already considered the case of the Data Block being available with the MC cache in the *section (i)* and the corresponding probability was given by:

$$\begin{aligned} P_K &= \int_0^{\infty} e^{-\lambda_{K,P} \cdot t} \lambda_{W,P} e^{-\lambda_{X,P} \cdot t} dt \\ &= \lambda_{W,P} / (\lambda_{W,P} + \lambda_{X,P}) \end{aligned}$$

Hence, the probability P_{Ad-Hoc} of the event that the MC is disconnected from its Access Point and the p^{th} Data Block is being asked by the user of the MC (by the time t) is given by:

$$\begin{aligned}
 P_{Ad-Hoc} &= \int_0^{\infty} (1 - e^{-\lambda_{x,p} \cdot t}) (1 - e^{-\lambda_{k,p} \cdot t}) \lambda_{w,p} e^{-\lambda_{w,p} \cdot t} dt \\
 &= 1 / \{ \lambda_{x,p} + \lambda_{k,p} + \lambda_{w,p} \} + 1 / \lambda_{w,p} - 1 / \{ \lambda_{x,p} + \lambda_{w,p} \} - 1 / \{ \lambda_{k,p} + \lambda_{w,p} \}
 \end{aligned}$$

This is how the cache co-operation among Mobile Client will improve the performance of the IP Storages over the wireless networks.

Encryption & Decryption of Data Blocks

The inspiration of this section is derived from the fact that the iSCSI allows transfer of sensitive block storage data over the TCP/IP. The inherited nature of the TCP/IP networks makes it vulnerable to malicious attacks from unauthorised persons or intruders. A Denial-of-Service attack may be also brought into place by sending TCP reset by the intruder. This results into compromise to the integrity and confidentiality of the corporate data many of which could be of mission critical importance. The Hierarchical caching of the Data Blocks (as proposed by us in the *section (i)* of our work) makes the Data even more vulnerable to such attacks from intruders.

The iSCSI Protocol (Login Authentication & Security of Data):

As per the IETF draft [1], in an iSCSI-based storage solution, the initiator (user, Mobile Client) and the target (Remote Storage Device) are connected by TCP/IP networks. The Mobile Client and the target may be physically separate and the network used for the connection may be the public network available. Thus, authentication of both the initiator and the target both is a must. The iSCSI protocol provides authentication in two phases:-

- (i) The target listens the network on the well known TCP ports and other ports as set by the administrator. A Mobile Client, which wants to establish a connection to the target, begins the login process by connecting to one of these ports available. In the iSCSI login phase, both the target and the Mobile Client exchange information to authenticate each other. During this phase the session's parameters, security association protocol etc are also exchanged. In order to protect this session the IPsec SA could be*

used. The IETF recommends Secure Remote Password (SRP) and Challenge Handshake Authentication Protocol (CHAP).

(ii) After this phase the Full Feature Phase begins in which the Mobile Client may send the SCSI commands to the target.

The above phases are very much needed in the case of iSCSI as the security protocol must support the following:-

- (i) Data Integrity,
- (ii) Data confidentiality,
- (iii) Data origin authentication.

Of the above mentioned point the most important is the Data confidentiality, as it is possible that the traffic of iSCSI may flow through insecure public networks. Also, the security protocol must provide security on per packet basis. Though, iSCSI login do provides authentication of both the initiator and the target at the beginning of a session, it does not provide any per packet authentication, integrity or confidentiality. Thus, to achieve the above mentioned we have to rely on the features available with the TCP/IP networks.

We propose the implementation of the IPSec protocol as mentioned by *Chaitanya, et al* in [14] with certain changes to suite our proposed architecture and some improvement in the performance. In this work (as already mentioned in chapter 2 section) 'Lazy Decryption' and 'Lazy Authentication' has been proposed. Also, it has been proposed that at the target the crypto-processing of the Data Blocks is not done as at the target data is only stored and no processing of data takes place.

We propose the following changes to work perfectly with our proposed architecture:-

- (i) In addition to the crypto-processing of Data Blocks being avoided only at the target level, we propose that at each of the hierarchical caches no crypto-processing of the Data Block should be done and only the crypto-processing of the control data which is contained in the iSCSI header should be done. The concept is depicted in the following figure.

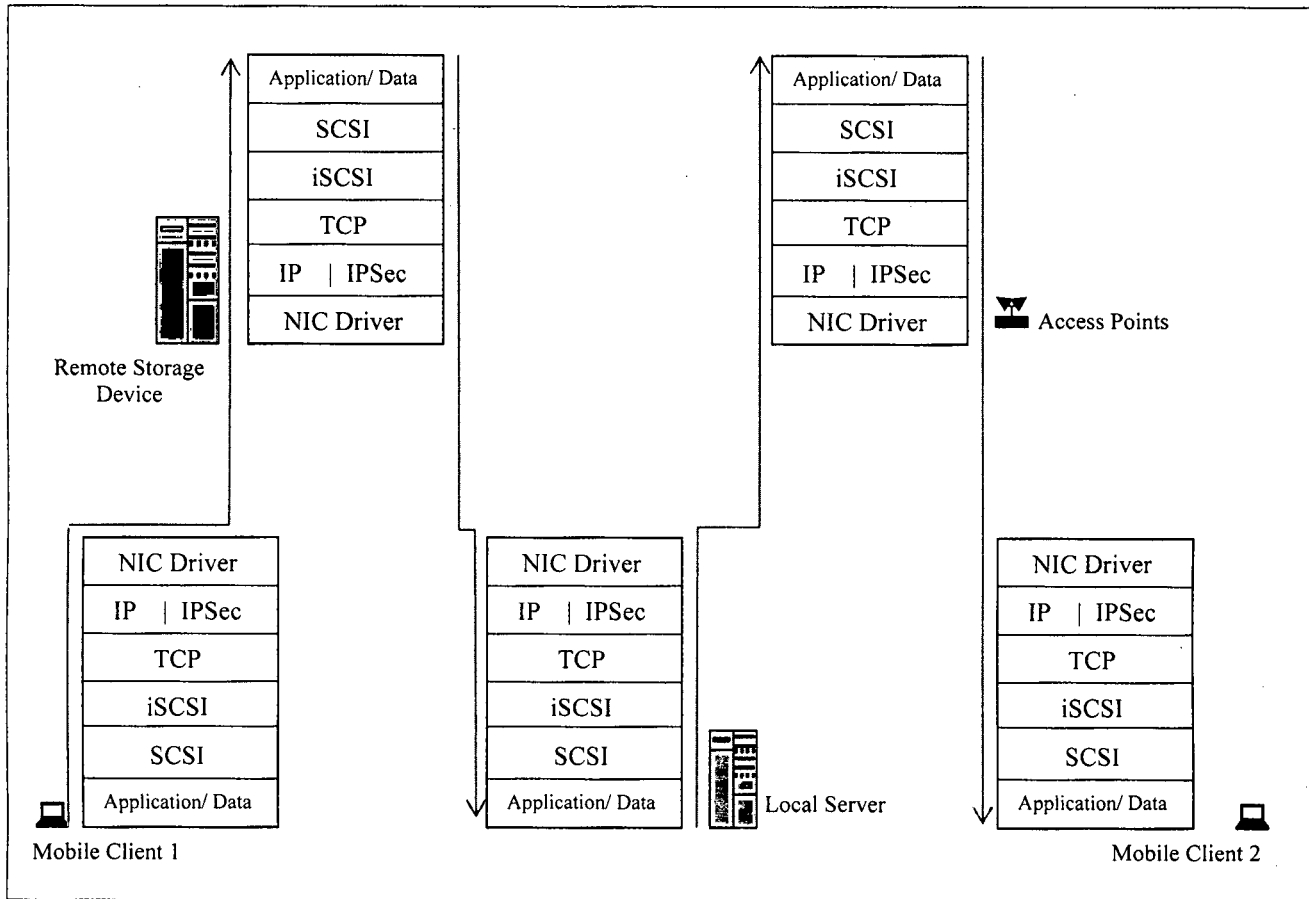


Figure No. 3.9: The improvised implementation of IPSec to enhance performance

- (ii) IPSec developed by IETF includes the Authentication Header (AH) and the Encapsulating Security Payload (ESP). The AH guarantees that the data being received by a receiver from a claimed sender is actually from the same sender and the data is also the same as it was sent by that sender. AH allows the receiver to identify the sender. It also ensures that the data received by a receiver has not been altered or tampered with. While the ESP provides confidentiality of the data being received by a receiver, connectionless integrity and anti-reply service. As the implementation of IPSec requires exchange of cryptographic algorithms and symmetric keys used by it between two connecting devices, the secrecy of such exchanges is of immense importance. IKE (Internet Key Exchange) protocol is one of the best

protocols to exchange the security associations (SAs) between the communicating devices. Thus, the proper functioning of IPSec depends upon the efficient implementation of IKE. We propose to use the efficient and secure IKE design as proposed in [12] by *Chang et al.*

Application Scenario:

In order to improve the performance we propose to use the following situation in our favour ‘ the data is processed only by the end user and no processing of a data is required at the Remote Storage Device, Local Server and at the Access Points’ .

The Mobile Client 1 (as shown in the figure 9) stores some Data Block, say X, in the Remote Storage Device. When the Data Block reaches at the Remote Storage Device only the iSCSI header is decrypted to know about the necessary control data. The rest of the data is stored in encrypted form with the Block Encryption codes of the respective Blocks stored in a separate file.

Now, when the Mobile Client 2 asks for the Data Block X, (presuming that it is not available at the caches of corresponding Access Point and Local server) the Data Block X is accessed from the Remote Storage Device along with the corresponding Block encryption keys and forwarded to the local server. At the Local Server level a copy of the Data Block X without any decryption is stored and the corresponding Block encryption keys will also be stored in a separate file. A similar task will be performed at the Access Point.

This greatly improves in the security of the Data Blocks as for example if an intruder gets the Access to any of the Hierarchical Caches, the intruder would not be able to induce any loss to the system as it would be accessing an encrypted data without the required decryption code.

This is how the encryption and decryption of the data blocks will improve the performance of the IP Storages over the wireless networks.

Simulation Results

We have simulated the *section (i)* of our proposed work. The code has been written in C ++, which is given in the appendix.

Simulation Results:

This code was run on Dev-Cpp beta version 4.9.9.2 [35] on a machine having the following configuration:-

- Processor Intel Pentium 4 CPU 2.80 GHz,
- 1 GB RAM,
- Intel 915 chipset Motherboard.
- Microsoft Windows XP service pack 2

We ran the simulation code several times in the above configuration with different seeds to the random function. This code has been also executed on a LINUX (Red Hat version 4 ES) machine couple of times. All the results were compared, in most of the cases the output was similar to the output we have presented below. In fact the result we put forward in the following pages was very near to the mean of all the results.

One important aspect witnessed by us was the percentage of number of Data Blocks that were not accessible in the first try by the Mobile Client, was never ever more than 4.6%.

The above said fact emphasises that the hierarchical caching of the Data Blocks has been really helpful in improving the access rate.

We now present the result of our simulation to the Hierarchical Caching of the Data Blocks.

No. of Data Blocks	Mobile Client cache	Access Point cache	Local Server cache	Remote Storage Device	Not Available
10000	1247	1064	1383	5880	426
25000	3004	2769	3574	14638	1015
50000	6251	5348	7016	29300	2085
75000	9388	8247	10335	43946	3084
100000	12471	10905	13716	58736	4172
125000	15493	13545	17040	73802	5120
150000	18573	16260	20658	88235	6274
200000	24901	21735	27436	117594	8334
250000	31052	27256	33994	147271	10427
300000	37114	32828	41102	176507	12449
500000	62419	54210	68388	294229	20754
750000	93618	81192	102422	441593	31175
1000000	124678	108817	136898	587959	41648
1250000	155853	135671	171531	734690	52255

Table No. 4.1: The Location from where Data Block was accessed

The bar Graph of the above table is as follows:-

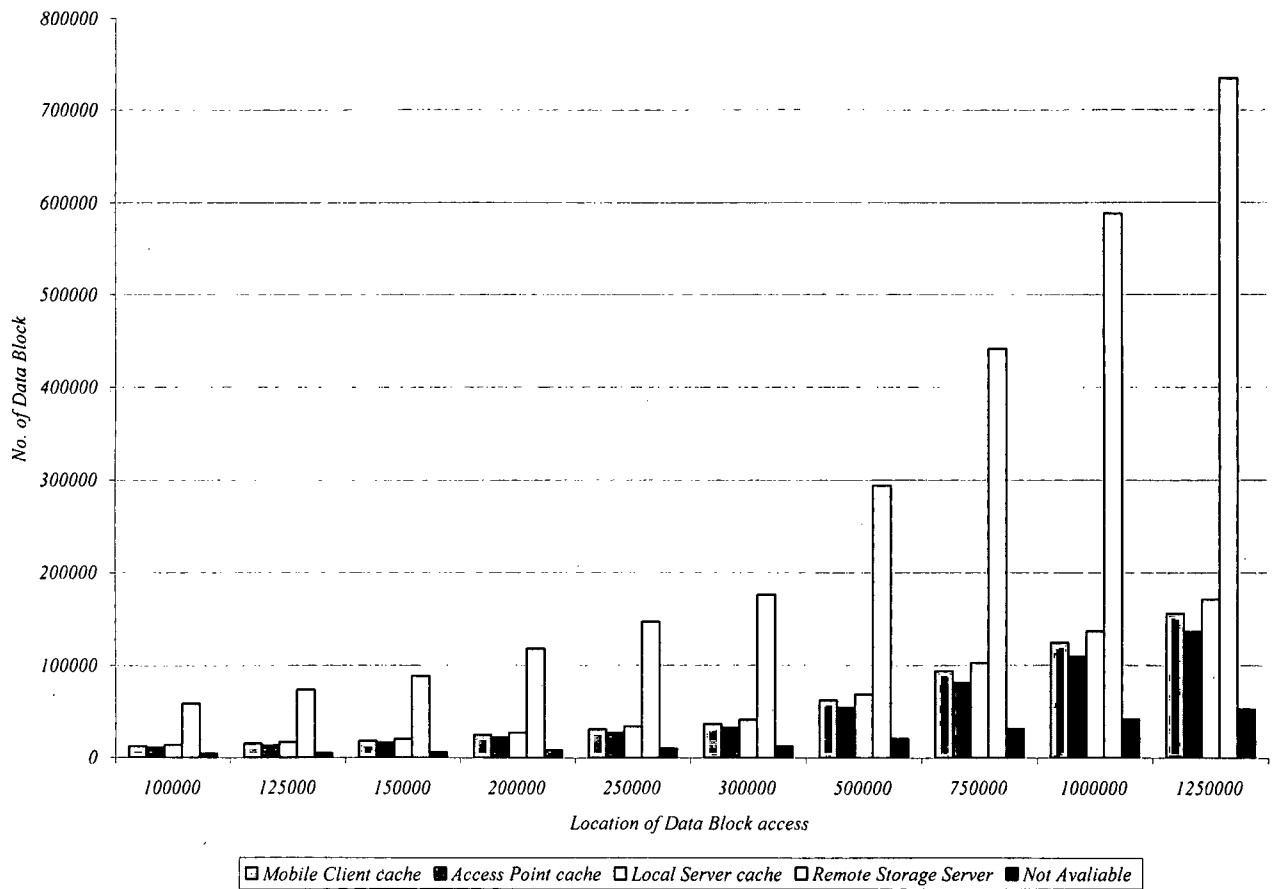


Figure No. 4.1: The Bar Graph of Data Block access from different locations.

Analysis of the result:

In the above figure, it is shown that the number of Data Blocks accessed from the particular location vs. the number of Data Blocks accessed. For example: in the case when 300,000 Data Blocks were accessed in total (6th Bar-Graph from the left), the following is the number of Data Blocks accessed from the corresponding location:-

- 37114 Data Blocks were accessed from the cache of the Mobile Client,
- 32828 Data Blocks were accessed from the cache at the Access Point,
- 41107 Data Blocks were accessed from the cache at the Local Server,
- 176507 Data Blocks were accessed from the Remote Storage Server.

While the last bar in the graph denotes that the user couldn't access 12449 numbers of Data Blocks in the first go. The Data Blocks may not have been accessed in the first try do the following factors:-

- Non availability of the Data Block in the storage of the corporate,
- Congestion in the network (due to which the request packet may have been dropped),
- Denial of Access to some permission settings associated with the corresponding Data Block,
- The Mobile Client might have been got disconnected, during the reply phase, from its Access Point, etc.

The following pie-charts give a better explanation of the above said scenario.

Distribution of Data Block Access when No. of Data Blocks = 50000

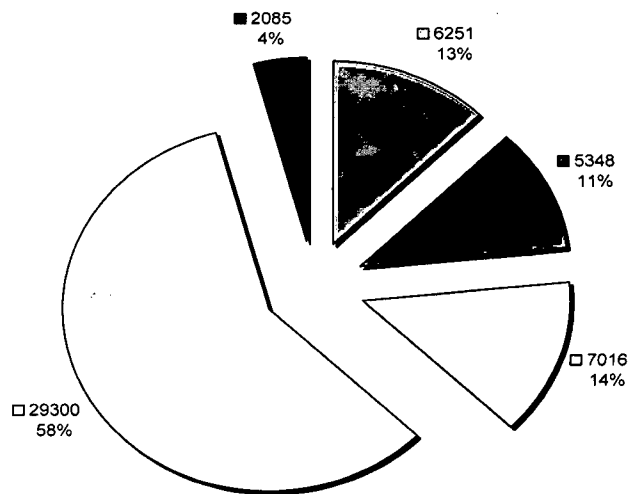


Figure No. 4.2: The pie-chart of location of Data Block Access when N=50000

Distribution of Data Block Access when No. of Data Blocks = 125000

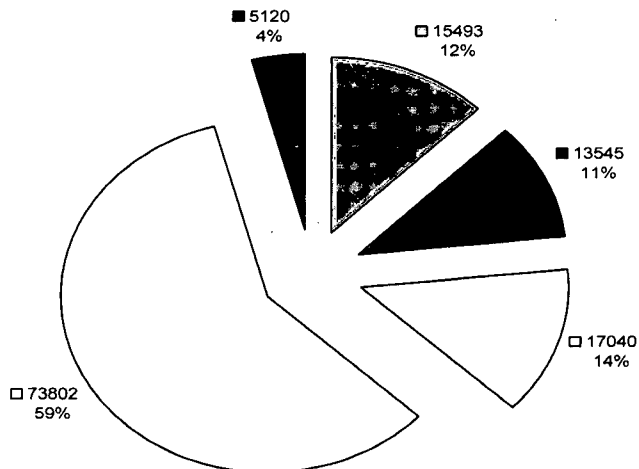


Figure No. 4.2: The pie-chart of location of Data Block Access when $N=125000$

The Pie-Charts of other distributions gave similar results. In all the runs of the simulation, the number of Data Blocks that were not accessible or available never ever increased 4.6% of the total number of the Data Blocks accessed. This percentage will be lowered to a lower level if the other sections of the proposed work are carried out. As, the disconnection from the Access Point is quite frequent in today's wireless communications, the number of such non accessibility of the Data Blocks will be definitely lowered if we successfully implement the *section(iii)* of our proposed work.

Conclusion & Future Work

Conclusion:

The simulation results of the *section (i)* of our proposed work has already shown that the 'Hierarchical Caching of the Data Blocks' has been helpful in improving the performance of iSCSI over the wireless networks. In a similar manner it can be shown that the 'Pre-fetching of the Data Blocks' will also contribute to improving the performance. With the advent of better algorithms in the arena of intelligent fetching and application of some sort of heuristic algorithm (computer science already have few very good algorithms in this area) the results could be further improved.

Another aspect that we chose to work on was that of the frequent disconnection of the Mobile Client from their respective Access Points, the solution proposed by us to this problem is quite worthy in nature as it doesn't demand any specific Hardware Requirements. The proposed solution utilises the presently available features in any Mobile Device configuration.

The problem related to the security of the Data Blocks when the public networks are being used. This was a challenging task as any compromise to the speed of access rate would have affected the entire performance. The solution proposed by us in this regard scales well with currently available features of the IP protocol.

Future Work:

The emergence of new technologies in the field of computer science is very rapid. The network is becoming more and more fast and reliable too, the computation power of the Mobile Clients is increasing day by day, and the new searching algorithms are even faster and more reliable. The new innovations in the field of Neural Networks, Artificial Intelligence, processing power of the CPUs, Graphics etc, all have contributed to the satisfy the ever increasing demand from the users of different classes.

The scope of improvement in our proposed work is immense; we consider a few of them out here. The possibilities of the future work are as follows:-

- In the *section (i)* of our proposed work we have put forward the concept of 'Hierarchical caching of the Data Blocks'. We have used Cache Discs at the Local Server level and the NVRAMs at the Access Points, with the emergence of the new technologies (and algorithms) in faster retrieving the concerned documents from the secondary memory sources the cache at the Local Server may be extended to the secondary memory storage available. This will result in greater storage capacity at the nearest level and hence will result in improved performance of the IP Storages.
- In the *section (ii)* of our proposed work we have given the concept of 'Pre-fetching of Data Blocks'. In this section there is even more possibility of enhancement in the algorithm of pre-fetching. The use of Artificial Intelligence, Neural Networks and Data Mining tools will greatly improve the pre-fetching technique and hence result in greater hits registered from any end user. As if the association rule (of the Data Mining) if implemented in a nice manner may result in predicting the next 'Data Block required' in more efficient manner than the present used method. Similarly, the Artificial Intelligence could be used for better result. This will in turn improve the performance of IP Storages.
- In *section (iii)* we have discussed about the 'Cache co-operation among the Mobile Clients' using the Ad-Hoc networks. The throughput in this case can be further

increased by developing separate Ad-Hoc routing protocol which can address to the required situation in an even better manner than what we have proposed. Also, as the power of the Mobile Clients is ever increasing, we may even route the Data Blocks required through a path containing PDAs. Also the pervasive architecture as mentioned by *Srinivasan* in [7] could be used to improve the performance of the IP Storages over the wireless networks.

- In our final section of the proposed work *section (iv)*, the ‘encryption and decryption of Data Blocks’ policy has been put forward. In this there lies a lot of scope as the algorithms are becoming better and better. A huge lot of options are already available and more suitable tailor made for this purpose could be designed.
- Another area of development is related to the mobility of the Mobile Clients. Since the high mobility of such devices results into frequent hand-offs and sometimes may lead to isolated node situation, this could be a challenging field to work on.
- The development of more powerful wireless LAN and the upcoming IEEE 802.11n like protocol would further help in improving the performance.

Simulation Code

```
#include<iostream>
#include<conio.h>
#include<stdlib.h>
#include<limits.h>
#include<time.h>
#include<math.h>
#include<stdio.h>

using namespace std;

float store[8];
int record[5];
int N;
void recorder(void);
void display();
void rander();

class linklist
{
private:
    static int s;
    struct node{
        int d;
        float data[8];
        struct node* link;
    }*p;
public:
    linklist();
    void append();
    void display_list();
    ~linklist();
};

linklist::linklist()
{
    p=NULL;
}
```

```

int linklist :: s;

void linklist :: append( )
{
    node *temp, *r;

    temp=new node;
    if(temp==NULL)
    {
        cout << "No further allocation possible Press Any Key for EXIT";
        getch( );
        exit(0);
    }

    else{
        temp->d = s++;
        for(int z=0; z<8;z++)
            temp->data[z]=store[z];

        temp->link=NULL;

        if(p==NULL)
            p=temp;
        else{
            r=p;
            while(r->link)
                r=r->link;
            r->link=temp;
        }
    }
}

```

```

void linklist :: display_list( )
{
    int f;
    node *temp;
    cout << "\n\n\tEnter the Data Block No. you wish to see: ";
    fflush(stdin);
    cin >> f;

    if((f<1)||f>N)
    {
        cout << "\n\n\t Invalid Choice";
        return;
    }
    temp = p;

    while(temp->d != f)

```

```

temp = temp->link;

if(temp->data[0]< temp->data[4])
{
    cout << "\n\t The " << f << "th Data Block was Accessed from cache of Mobile Client";
    return;
}
else if(temp->data[1]< temp->data[5])
{
    cout << "\n\t The " << f << "th Data Block was Accessed from cache at the Access Point";
    return;
}
else if(temp->data[2]< temp->data[6])
{
    cout << "\n\t The " << f << "th Data Block was Accessed from cache at the Local Server";
    return;
}
else if(temp->data[3]< temp->data[7])
{
    cout << "\n\t The " << f << "th Data Block was Accessed from the Remote Storage Device";
    return;
}
else
{
    cout << "\n\t The " << f << "th Data Block was not Accessible";
    return;
}
}

```

```

linklist :: ~linklist( )
{
    node *r;
    while(p)
    {
        r=p;
        p=p->link;
        delete r;
    }
}

```

```
linklist list;
```

```

void recorder(void)
{
    if(store[0]<store[4])
    {
        record[0]++;
        return;
    }
}

```

```

    }
    else if (store[1]<store[5])
    {
        record[1]++;
        return;
    }
    else if (store[2]<store[6])
    {
        record[2]++;
        return;
    }
    else if (store[3]<store[7])
    {
        record[4]++;
        return;
    }
    else
        record[3]++;
    return;
}

```

```

int factorial (int n)
{
    int f=1;
    if ((n==1)|| (n==0))
        return 1;
    else
    {
        while(n>1)
        {
            f*=n;
            n--;
        }
        return f;
    }
}

```

```

float powermaker(float x, int y)
{
    float w = 1.0;
    if(y==0)
        return 1;
    else if (y==1)
        return x;
    else{

```

```

        for(int j=0; j<y; j++)
        {
            w = x*w;
        }
        return w;
    }
}

```

```

void rander( )
{
    int u,v,y,z;
    float a,b,c,d;
    time_t t;
    srand((unsigned)time(&t));
    for(int i=0; i<N; i++)
    {
        for(int k=0; k<4;k++)
        {
            y = rand( )%10;
            z = rand( )%10;
            a = rand( )/47;
            b = rand( )/37;
            c = powermaker(a,y);
            d = powermaker(b,z);
            u = factorial(y);
            v = factorial(z);
            store[k] = (c*(exp(-a)))/u;
            store[k+4] = (d*(exp(-b)))/v;
        }
        recorder( );
        list.append( );
    }
}

```

```

void display( )
{
    cout << "\n\tTotal No. of Data Blocks Accessed: " << N;
    cout << "\n The No. of Data Blocks Accessed from cache of Mobile Client: " << record[0];
    cout << "\n The No. of Data Blocks Accessed from cache at Access Point: " << record[1];
    cout << "\n The No. of Data Blocks Accessed from cache at Local Server: " << record[2];
    cout << "\n The No. of Data Blocks Accessed from the Remote Storage Device: " << record[4];
    cout << "\n The No. of Data Blocks that couldn't be Accessed: " << record[3];
}

```

```

int main(void)
{
    char c;
    cout << "\n\nEnter the number of Data Blocks to be Accessed \t\t";
    cin >> N;

    for(int i=0; i<8; i++)
        store[i] = 0.0;

    for(int j=0; j<5; j++)
        record[j] = 0;

    if(N<0)
    {
        cout << "\n\n\tImproper Choice Press Any Key to EXIT";
        getch( );
        exit(0);
    }
    rander( );
    display( );

    cout << "\n\n Do you want info about the Data Block Access if yes PRESS y or Y ";
    fflush(stdin);
    cin >> c;
    if((c=='y')||(c=='Y'))
        list.display_list();
    cout << "\n\n\t\tPress Any Key To Exit";
    getch( );
    return 0;
}

```

References

1. J. Satran, K. Meth, C. Sapuntzakis, M. Chadalpaka and E. Zeidner , RFC 3720 “ Internet Small Computer Systems Interface,” in Standard Track, April 2004, <http://www.ietf.org>
2. C. Boulton, “iSCSI becomes official storage standard,” <http://www.internetnews.com/>
3. X. He, Q. Yang, and M. Zhang, “Introducing SCSI-To-IP cache for Storage Area Networks,” in *Proceedings of the 2002 International Conference on Parallel Processing*, Vancouver, Canada, August 2002.
4. Y. Hu and Q. Yang, “DCD–disk caching disc: A new approach for boosting I/O performance,” in *Proceedings of the 23rd International Symposium on Computer Architecture*, Philadelphia, Pennsylvania, May 1996.
5. W. Sun, J. Shu, W. Zheng, “Dynamic File Allocation in Storage Area Networks with Neural Network Prediction,” LNCS, Springer-Verlag 2004.
6. S. Deering and R. Hinden , RFC 2460 “Internet Protocol, Version 6 (IPv6) specification ” in Standard Track, December 1998, <http://www.ietf.org>
7. S. H. Srinivasan, “Pervasive wireless grid architecture,” in WONS, 2005.
8. M. He, T. D. Todd, D. Zhao and V. Kezys, “Ad Hoc Assisted Handoff for Real-time Voice in IEEE 802.11 Infrastructure WLANs,” WCNC 2004, IEEE communications society.
9. A. Iwata, C. Chaing, G. Pei, M. Gerla and T. Chen, “Scalable Routing Strategies for Ad Hoc wireless Networks,” *IEEE journals on selected areas in communications*, vol. 17 No.– 8, August 1999.

10. X. Hong, K. Xu and M. Gerla, "Scalable Routing Protocols for Mobile Ad Hoc Networks," IEEE Network July/ August 2002.
11. G. Pei, M. Gerla, T. Chen, "Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks," *Proceedings in the ICC 2000*, New Orleans, LA, June 2000.
12. M. Su and J. Chang, "An Efficient and Secured Internet Key Exchange Protocol Design," *Fifth Annual Conference on Communication Networks and Services Research(CNSR '07)*
13. X. He, Q. Yang and M. Zhang, "A Caching Strategy to improve iSCSI Performance," *Proceedings of the 27th Annual IEEE Conference on Local Computer Networks (LCN '02)*.
14. S. Chaitanya, K. Butler, A. Sivasubramaniam, P. McDaniel and M. Vilayannur, "Design, Implementation and Evaluation of Security in iSCSI-based Network Storage Systems," *StorageSS '06*, ACM, October 2006, Alexandria, Virginia, USA.
15. D. He and D. H. C. Du, "An Efficient Data Sharing Scheme for iSCSI-Based File Systems," *DTC intelligent consortium at UMN*.
16. Z. Wang, M. Kumar, S. K. Das and H. Shen, "Investigation of Cache Maintenance Strategies for Multi-Cell Environments," MDM 2003, LNCS 2574, Springer-Verlag.
17. Storage Networks Industry Association, <http://www.snia.org>.
18. C. Monia, R. Mullendore, F. Travostino, W. Jeong, M. Edwards, RFC 4172 "Internet Fibre Channel Protocol," in Standard Track, April 2004, <http://www.ietf.org>.
19. J. Kubaiatowicz, et al, "OceanStore: An Architecture for Global-Scale Persistent Storage," *Proceedings of the international conference on Architectural support for programming languages and operating systems (ASPLOS' 2000)*, 2000.
20. "iSCSI for Storage Networking," SNIA IP Storage Forum White Paper, <http://www.snia.org>.

21. "iSCSI Technical White Paper," SNIA IP Storage Forum White Paper, <http://www.snia.org>.
22. "Connecting SANs Over Metropolitan and Wide Area Networks," Brocade White Paper, <http://www.brocade.com>.
23. D. V. Anidi and S. Nujeerallee, "Storage area networking – an introduction and future development trends," BT Technology Joournal, Vol 20 No. 4, October 2002.
24. "Brocade SAN Solutions: A More Effective Approach to Information Storage and Management," White Paper, Brocade, <http://www.brocade.com>.
25. "Comparing Storage Area Networks and Networks Attached Storage," White Paper, Brocade, <http://www.brocade.com>.
26. Tapan K. Lala, "Storage Area Networking," Guest Editorial, IEEE Communications Magazine, August 2003.
27. Eric Sheppard, "Migrating to Storage Area Networks: A Report from the Front Lines," An IDC White Paper, <http://www.idc.com>.
28. S. Aiken, D. Grunwald, Andrew R. Pleszkun, J. Willeke, "A Performance Analysis of the iSCSI Protocol," *Proceedings of the 20th IEEE / 11th NASA Goddard Conference on Mass Storage Systems and Technologies (MSS'03)*.
29. J. Menon, D. A. Pease, R. Rees, L. Duyanovich and B. Hillsberg, "IBM Storage Tanks – A Heterogeneous Scalable SAN File Systems," IBM SYSTEMS JOURNAL, VOL 42, NO. 2, 2003.
30. G. Orenstein and J. Shurteff, "Integration Scenarios for iSCSI and Fibre Channel," SNIA IP Storage Forum White Paper, <http://www.snia.org>.
31. "iSCSI Building Blocks for IP Storage Networking," SNIA IP Storage Forum White Paper, <http://www.snia.org>.

32. "IP Storage Customer Deployments – Where IP Storage Fits: The Emergence of IP-Based SAN Solutions," issue 20 November 2004, <http://www.snseurope.com/>
33. J. Sco, H. Shin and M. Park, "Optimising iSCSI Parameters for Improving the Performance of iSCSI based Mobile Appliance in Wireless Network," *Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies(ICNICONSMCL'06)*, 2006
34. Y. Lu and D. H. C. Du, "Performance Study of iSCSI- Based Storage Subsystems," *IEEE Communications Magazine*, August 2003.
35. <http://www.bloodshed.net>
36. K. Z. Meth and J. Satran, "Features of the iSCSI Protocol," *IEEE Communications Magazine*, August 2003.