Heterogeneous Peers Configuration to Enhance Cooperation in P2P Overlay Networks

By
Saira Aslam
2008-NUST-MS PhD-CSE (E)-19
MS-8 (SE)

Submitted to the Department of Computer Engineering
In fulfillment of the requirements for the degree of

Master of Science
In
Computer Software Engineering

Thesis Advisor
Brig. Dr. Muhammad Younus Javed

College of Electrical & Mechanical Engineering
National University of Sciences & Technology

2011

In the name of Allah, the most Merciful and

the most Beneficent

DECLARATION

I hereby declare that I have developed this thesis entirely on the basis of personal efforts under the sincere guidance of my supervisor Brig. Dr. Muhammad Younus Javed. All the sources used in this thesis have been cited. No portion of the work presented in this thesis has been submitted in support of any application for any other degree of qualification to this or any other university or institute of learning.

Saira Aslam

ACKNOWLEDGEMENTS

A research work is always an ongoing process, it has no end, but it definitely has a start. This start depends on the environment around us. I would really like to thank Allah that he has given me the chance to study in one of the most prestigious universities of my country. Studying in NUST has been the toughest, but the most precious experience of my life. I have been provided with all the academic support from the extremely qualified, helpful and inspirational faculty. This thesis work is an essence of all the help and guidance I got from the following people.

I offer my sincerest gratitude to my supervisor, Brig. Dr. Muhammad Younus Javed, who has supported me throughout my thesis with his competence and knowledge. I attribute the level of my Masters degree to his encouragement and effort and without his support this thesis would not have been completed. One simply could not wish for a better and friendlier supervisor.

I am heartily thankful to Col. Dr. Khalid Iqbal, whose encouragement, guidance and support enabled me to develop an understanding of the subject. I also owe my deepest gratitude to Dr. Aasia Khanum and to Dr. Saad Rehman for their help and encouragement which was always a source of motivation and confidence.

I would also like to thank my husband and parents for supporting me throughout all my studies at University. Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of the project.

DEDICATED TO MY PARENTS, TEACHERS,

HUSBAND AND CHILDREN

.

Abstract

This research work is based on the enhancement of Quality of Service (QoS) in Peer-to-Peer (P2P) overlay networks. It reduces free riding by enhancement of cooperation among heterogeneous peers for the fair share of resources, as well as elevates the efficiency of information retrieval regarding a particular peer and its resources available and demanded for and from the network respectively. These resources could be peer's physical attributes such as free storage space; RAM, processor cycles, or it could be any other application based resources, like media files etc. An efficient behavior of the peers in the network is retrieved by introducing cooperating groups (CG) in the network, where cooperating peers are associated to a particular CG. CG-Identifier (ID) is assigned to those peers who share resources on the basis of give and take rule. Through CGs it would be far more efficient to search an idol peer waiting for another peer so that both could share each other's resources. Heterogeneous Cooperating Group-based Newscast Protocol (HCGNP) has been introduced in the network which generates cooperating peers in the network. Simulations and results show that the probability of cooperation between the peers, who remain part of the network, is very high as compared to other cooperation algorithms. Hence the current research introduces an entirely different and an efficient way of enhancing cooperation between the peers in P2P overlay networks.

TABLE OF CONTENTS

Contents **Page No**

CHAPTER 3: IMPLEMENTATION OF HETEROGENEOUS COOPERATING GROUP-BASED NEWSCAST PROTOCOL (HCGNP)

CHAPTER 4: SIMULATIONS AND RESULTS

CHAPTER 5: CONCLUSIONS & FUTURE WORK

REFERENCES

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction to Heterogeneous Peer-to-Peer Networks

1.1 Motivation

The idea of resource sharing in digital communication is although not very new but it requires a continuous study and research. The most promising fact is that resources associated to individual machines should become available for other machines that require those resources. Peer-to-Peer (P2P) computing, hence proposed many different ways to design architectures, where resources of different peers can be utilized, minimizing the wastage factor associated to any resource. Apart from digital communication, every other development associated to science and engineering requires different resources to function. To provide resources to other machines, every machine or technology requires another resource. The resource sharing procedure continues only through this give and take rule. If this rule is not followed the cycle of resource sharing will never continue, which in return would shatter the procedural and functional requirements of every other development.

However, to implement a trend for sharing resources is more towards playing with the psychology of the resource owner. There is a need to grow an environment, where the excessive resources available should be shared among others instead of being wasted. Moving on to P2P computing, there is a vast research going on to enhance the efficiency of information retrieval regarding resources available at various peers (machine owners). The idea behind is not only the sharing of resources but also to induce a sense of cooperation among the peers. Numerous techniques have been implemented which enhance cooperation among the peers to share several resources. These resources are mostly for sharing application level data, such as media files. But another important aspect of P2P networks is to share physical resources such as, free storage space, RAM or processor cycles, in a heterogeneous environment where peers connected to each other have different operating systems, processors, RAM and secondary storage. There is a need of an architecture which not only handles the heterogeneous environment of the network but also maintains a quality of service (QoS) by enhancing cooperation, security, network management and information retrieval through resource searching proficiently.

1.2 Background

One of the many magical elements of the Internet is that every computer connected to it is also connected to every other computer. There is no central switching office as with the telephone system. Some of the computers on the internet are servers, providing huge amounts of information and transactions, but most of the computers are home and office PCs operated by the peers (individuals). When one of these peers connects with another one, it is called a peer-to-peer connection. Like most technologies that have gained attention on the Internet, peer-to-peer is not a new idea. Peer-to-peer went main stream during the dot com era of the late 1990s when a teenager named Shawn Fenning appeared on the cover of *Time* magazine after having founded a company called Napster. Napster devised a technology for using peer-to-peer connections to exchange compressed music files (MP3s). Because MP3 music downloaded from the Net sounds the same as music from a CD, and because there are millions of college students with fast Internet connections, the peer-to-peer phenomenon experienced a meteoric growth in popularity. The recording industry should have anticipated music sharing but instead found itself on the defense and then resorted to legal action to stem the tide.

However, it must be understood that scope of P2P networks is not limited to some application rather it accelerates information sharing which includes many fields of knowledge, such as business collaboration, governance tools, medicines and academics. It must not be viewed as person to person contact for sharing of fun material; rather it should be taken in perspective of network of millions of people from various fields of life with huge knowledge base. Another vital aspect which should not be neglected that it is not just sharing of information but it also guarantees combining the computing power of huge number of computers which may surely assist in finding the solutions of many hindrances that still affect human life. However its aspect that is effecting music industry cannot be set aside. The concern of major enterprises who banned P2P tools, considering the threat of illegal use of intellectual property must be given due consideration. Hence P2P must not be banned categorically. Its horizon should be thoroughly understood and exploited for its merits while policy makers should evolve the procedures which deal with its legal and societal issues. If the benefits of P2P network are thoroughly recognized in comparison of its demerits, it will find its true place in the communication world.

1.3 Methodology

This research work is based on utilization of resources in Peer-to-Peer (P2P) overlay networks. The formation of P2P overlay architecture is based on sharing resources among different peers. These peers (nodes) join the network in order to gain benefits from the network in the form of resources like, different files, applications, bandwidth, extra storage, RAM and processor cycles. When the nodes with variant properties and physical resources join a network, such a network is known as heterogeneous network. Sometimes a node demands a resource, which it has shared with other nodes previously [1].

The interconnected nodes in P2P overlay networks possess multiple physical parameters which are shared among the participating peers. Apart from data and applications, physical parameters include RAM, extra storage space and processor speed. The interconnection of nodes largely depends on the contents which are shared among the nodes in the network. This resource sharing process is often deteriorated as after searching the relevant resource in the network, the node detects that it is not capable enough to share that resource. The efficiency of the network decreases when the network is unaware of the node's physical capabilities. The current research emphasizes on creating method for efficient collaboration of nodes for sharing physical resources like storage and processor cycles. This collaboration should have the ability to enhance cooperation among the nodes. The method currently discussed is based on development of cooperating groups, which enhances information retrieval between the nodes. These cooperating groups are formed on the basis of the cooperation level associated to it. By cooperation level it means weather the node in the network requires or demands a particular resource from the other node. The cooperating group formation is an on-demand procedure depending on weather the node is cooperating or not. The following conditions need to be work out to enhance cooperation in the cooperating groups:

(a) Check the capability level of the physical resource of the node to be shared.
(b) Check if the capability level of the node is available or demanded.
(c) By available capability level it means the physical resource to be shared with other node.
(d) By demanded capability level is means the resource that is required by a node from another node in the network.

(e) Capability levels assigned to the nodes are from 5 to 1.
(f) 5 is the highest level, showing maximum availability of a particular physical resource.
(g) 1 is the lowest level showing demanded resource from a node.

The method for assigning the cooperating group ID (CG-ID) to a node should check the capability level (available/demanded) of every node, and assign cooperating group ID only if a node shares and gains a resource at the same time. CG-ID would not be assigned if a node only shares its resource with the other node. The node is compelled to gain any other resource from the node with which it is sharing its resource. Hence every cooperating group contains pair of nodes that are sharing and gaining resources from one another at the same time.

1.4 Scope of the Project

This research deals in the evaluation of the trend of cooperation among the peers in P2P networks. Different algorithms like SLACER (Selfish Link-based Adaptation for Cooperation, Excluding Rewiring) and Social Network Architecture (SNA) have been studied to observe the cooperation behavior among the nodes in P2P overlay networks. Following areas are covered in this research:

(a) The protocol implemented is based on Newscast computing which generates the nodes in the network randomly, where every node keeps the information of its neighboring nodes. On the top of newscast protocol, a cooperating grouped approach will be introduced, to modify newscast protocol to HCGNP (heterogeneous cooperating group-based newscast protocol).
(b) The implementation of the protocol is based on the data structures assigned to the peers (nodes).
(c) The trend of cooperation among the peers after the implementation of HCGNP would be observed.
(d) The benefit of cooperating group approach to enhance cooperation is also to be marked.
(e) In order to accelerate the nodes to cooperate, the rewarding factors are assigned according to the network capabilities.

(f) The probability of cooperation between any two nodes at a particular time is compared with SLACER and SNA algorithms.

1.5 P2P Networks- An Introduction

Peer-to-Peer (P2P) has emerged as one of the most vibrant technique in information technology. P2P refers to the concept of simultaneous (spontaneous) collaboration in a network of two or more individuals (peers) using specified information and communication systems. A vital aspect of such network is that they do not require any central coordination. In comparison to client/server networks, P2P networking is a promising technique that offers improved scalability and fault tolerance, reduced ownership cost, decentralized coordination of underutilized resources and enhanced support for developing ad hoc networks. P2P networks also extend support to incorporate new client scenarios, which is a cumbersome task for conventional approaches [1]. All the peers in the network have portion of their resources directly available for the other peers. The basic characteristics of P2P networks are:

(a) Sharing of distributed resources and services :

P2P networks support the functioning of each node in a dual role of client and server. A node (Peer) may either act as a service provider or a consumer of resources such as storage, bandwidth, processor cycles, files and other required information. These nodes are also called as servants, a terminology deduced from client and server technique.

(b) Decentralization:

In P2P networks, a central coordinating authority to manage the setup and sequence aspects is not required. This technique negates the supremacy of one node over the others. The network is self organized, resource sharing and communication among the nodes is established directly. The P2P networks are further categorized as pure and hybrid networks. In P2P networks each node shares equal rights and functions, thus avoiding supremacy of any entity within the decentralized structure. On the other hand, hybrid P2P network amalgamates the principles of P2P and client server technique within the architecture by assigning various functions including indexing and authentication to a certain subset of nodes who act as coordinating entity. P2P networks have attracted wide attention and opportunities due to rapid internet growth and reducing costs for bandwidth, storage and processor cycles. Recent past has brought enormous increase in P2P applications along with controversies regarding its performance, limits, socioeconomic bounding and legal implication of these networks.

Figure 1.1 represents a multi level (three-level) model containing P2P infrastructure, application and communities [19]. It defines and clarifies the currently existing terminologies in theory and practice.

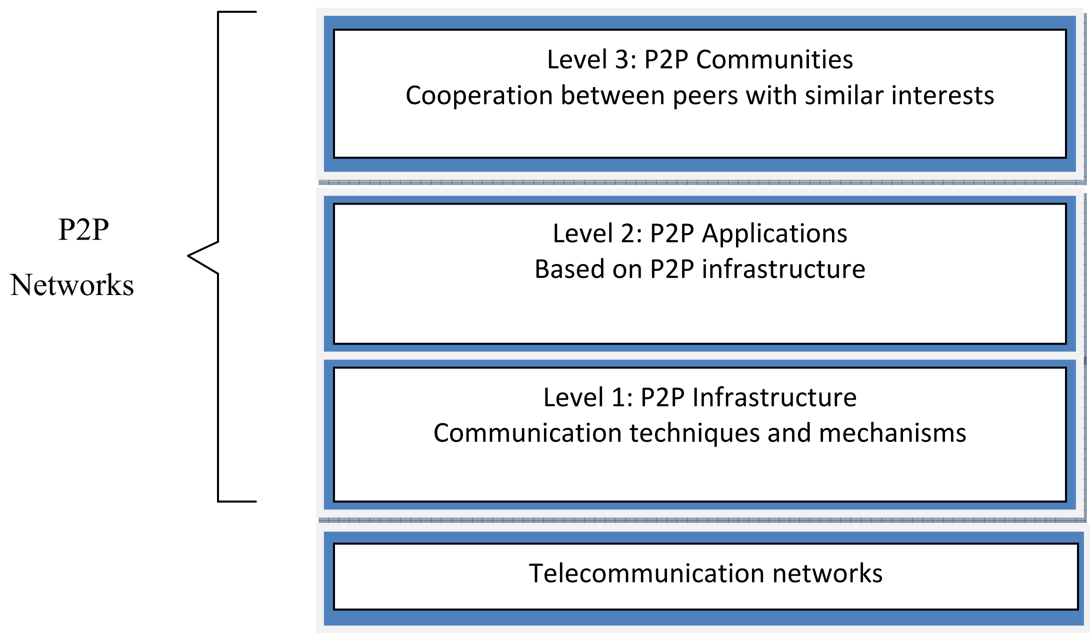


Figure 1.1: Levels of P2P networks

Level 1 in above flow diagram illustrates P2P infrastructure, existing above telecommunication networks. These telecommunication networks provide a base for the constituent levels. Communication, integration and translation functions among various information technology components are established by P2P infrastructure. It provides differentiated services to the peers of the network for communication, identification, usage and resource sharing. It also helps in authentication and authorization to activate the security process.

Level 2 enlists P2P application based on services provided by P2P infrastructure. They are designed in a fashion, such that various entities of the network may collaborate and communicate without any central control. While level-3 emphasizes on development of various communities and their interconnection dynamics. Level-1 and level-2 deal with the term 'Peer' purely as technical entity. On the other hand level-3 handles non technical and social aspects of this terminology

P2P networks architecture is contrary to traditional client/server system. Instead of designating separate machines for server or client, the nodes in P2P environment are designed to perform dual role. P2P technology has led to various applications with huge networks, thus demanding the development of various schemes which may handle the organization and location of resources of participating nodes.

1.6 P2P Overlay Networks

P2P overlay networks represent a distributed system, functioning without any central control or some specific hierarchical organization [20]. The peers from these networks are overlayed over internet protocol (IP) network. These peers need various characteristics to perform efficiently in a network such as selection of neighboring peers, hierarchical naming, authentication, anonymity, robust wide area routing structure, efficient date search, fault tolerance, utilization of redundant storage and consistency and massive scalability. The services of P2P overlay networks include assignment of dual role of client and server to a peer and deliberate resource sharing by other systems. Resource access and sharing in P2P overlay network is assisted by self organization, massive scalability and fault tolerance characteristics. The P2P networks are independent of collaboration demand among interconnected groups, while it is basic requirement for establishment of a network in grid systems. Similarly, peers from P2P networks may collaborate and establish resource sharing even with less reliable set of resources. P2P overlay network models cover a wide range of communication frame work specified by a completely distributed and cooperative network design, where the peers establish a self organizing system. Figure 1.2 [2] represents various components of overlay communication frame work in a conceptual P2P overlay architecture.

The network communication layer specifies the properties of computer machines, linked through internet. These machines can also be interlinked through a wireless local area network in an ad hoc environment. Challenges are faced by communication infrastructure due to dynamic nature of peers in an overlay network. The overlay node management layer describes peer's management including their discovery and implementation of optimized routing algorithm. The main task of features management layer is to maintain the robustness of P2P network systems. This layer mainly negotiates the reliability, security, fault resilience and resource availability issues. The service specific layer provides the tools to support core P2P infrastructure and components that are purely application based. Meta-data deals with the stored content and location information of peers across the P2P network. The application level layer handles the tools, applications, services and data. These components are implemented with particular programming techniques and methods.

The P2P overlay network layer lies between the network management layer and overlay node management layer. The idea of decentralization is such that the network topology is maintained at the overlay management layer which can implement protocols that need centralization like authentication of a peer, node management issues, node history to be maintained, etc. At this level, all the nodes are connected directly to each other on the basis of virtual links. The virtual link setup is the key property of P2P overlay networks, which lacks the presence of any centralized authority only at overlay network layer. Hence the implementation of security protocols which require centralized authority like authentication needs to be implemented at overlay node management layer. P2P overlay networks are broadly categorized in two classes, namely structured and unstructured overlay networks [2]. The details of these classes are explained below.

1.6.1 Structured P2P Overlay Networks

Structured P2P overlay networks may technically be defined as a tightly controlled topology [21]. The contents are placed at specific locations in order to maintain the efficient handling of subsequent queries, instead of distributing the contents randomly with participating peers.

These P2P overlay networks utilize the Distributed Hash Table (DHT) to maintain record of such locations [22]. The location information of data content is deterministically placed at the peers along with certain identifiers which correspond to the data content's unique key. The DHT based systems help in assigning a uniform random node-ID to the set of peers among a vast space for identifiers. A unique identifier, which is assigned to certain data content (object) and selected from above mentioned identifier space, is known as a key. Keys reflect the sequential mapping between overlay network protocol and a distinct active peer in that network. Figure 1.3 [2] indicates the functioning of P2P overlay networks for sealable storage and subsequent retrieval of designated key pairs.

DHT based systems are designed to ensure that any data object may be located in a small *O(log M)* overlay which proceeds on an average, while *'M'* depicts the quantity (number) of peers in that system. The path for a DHT based overlay network significantly differs from underlying network path between two peers. Therefore, the latency i.e. delay caused by look up tables in a DHT based P2P overlay network can rise high, thus subsequently degrading the performance of certain applications running over it.

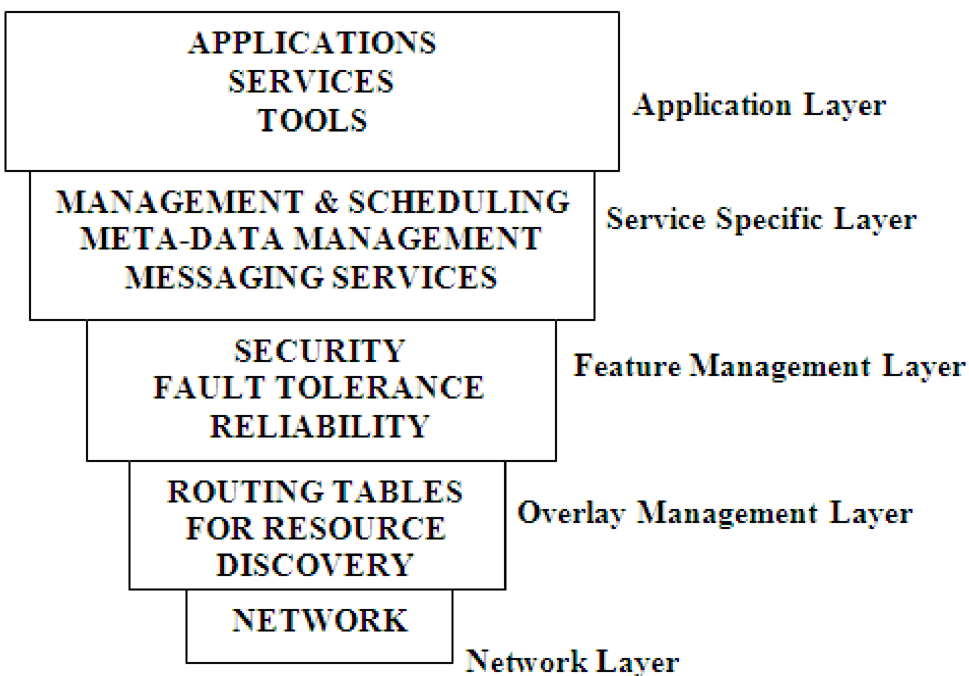


Figure 1.2: An abstract P2P overlay architecture

The search for a particular data object in a reduced circle of overlay nodes is guaranteed by DHT-based systems. This is done in a hop by hop manner. The average time period of a single hop is taken as $O(logM)$, where 'M' is the number of nodes in the overlay network. A normal routing path between the nodes in an unstructured P2P overlay network is quite different from the hop by hop methodology of DHT-based systems in structured P2P overlay networks. The look up latency to search a data in a hop by hop path is very prominent, which deteriorates the performance of any protocol running at the application level.

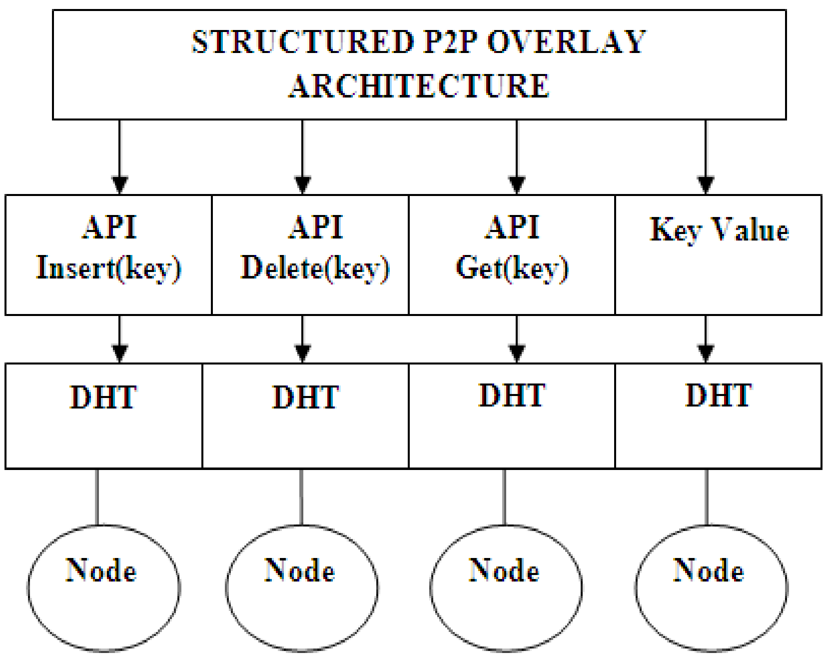


Figure 1.3: Application interface for structured DHT-based P2P overlay systems

1.6.2 Unstructured P2P Overlay Networks

The unstructured network is a class of overlay networks which randomly organizes the peers in a hierarchical view (e.g. super-peers layer) and utilizes time-to-live (TTL) or flooding search techniques in orders to response a query regarding any stored content by participating peers. This process is illustrated in Figure 1.4 [2]. It is worth mentioning here that each visited peer locally evaluates the query from its own content and subsequently assists while responding to complex queries.

The unstructured P2P centralized overlay model was initially introduced by Napster [23]. The model demands a designated infrastructure i.e. a directory server and poses certain scalability limits. Gnutella [24] is a flooding requests model for decentralized P2P overlay system in which every peer maintains a user-driven neighbor table. This technique is sufficiently effective to identify popular data objects, but it may lead to excessive consumption of network's bandwidth. Look up capability limit imposed by TTL, unearths another shortcoming in accessing the unpopular or remote data objects. KAAZA [25] is another unstructured P2P architecture. It is like fast track [26] architecture and works as a decentralized data sharing system based on meta-data searching technique.

Figure 1.5 shows super peers architecture in which participating peers possess high bandwidth, processing power and larger disk space. Such peers are volunteered to be elected in order to facilitate meta-data searching. This architecture is a structured overlay of peers and enhances the search efficiency of the system. The functioning of system is streamlined by use of super peers. All queries are transmitted to them, where as ordinary peers also transmit the meta-data of shared content to these super peers. Then this highly selected overlay network of super peers performs a Gnutella type broadcast based search. Although, super peers technique consumes a considerable amount of bandwidth resources in order to maintain indexing at super peers on behalf of other ordinary connected peers, still its performance in responding queries is outstanding. One the other hand, systems without super peers cause worse query delay and enhance latency of the system. Free-net [27] is another technique in which a combination of structured and unstructured P2P overlay architecture is used to develop an adaptive P2P network. This network of participating peers initially stores the queries and resultantly retrieves the data identified by location independent keys. It may be termed as loosely structured decentralized P2P network with anonymity based data placement. Each peer in this network maintains a dynamic routing table containing addresses of data keys and other participating peers.

Free net is characterized by its security features against malicious activities of other peers and the ability to maintain data in accordance with maximum disk space specified by the network operator.

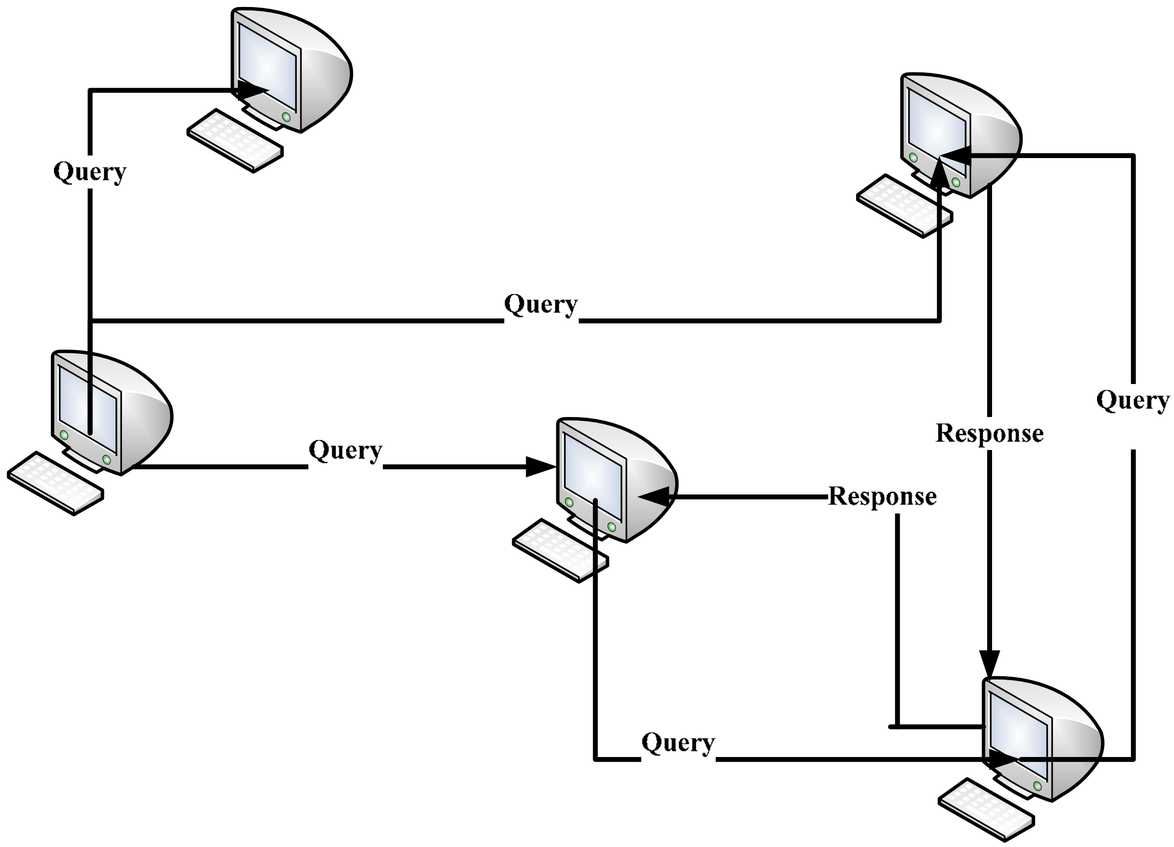


Figure 1.4: Gnutella-A decentralized architecture- document location and retrieval

The nodes in free-net architecture, broadcast the query request messages to acquire different key values. These key values are obtained by sending object-requests to the neighboring nodes. These object-requests are based on the decisions made by each node in the architecture in which the node decides the location to send the object-request, implicitly. These object-requests function exactly the way nodes send query-requests in internet routing protocol (see Figure 1.6 [2]). Free-net also facilitates peers to share free storage space which provides a logical extension to peer's own physical resources.

The basic drawback of DHT-based key assigning process is that it delays the process of file sharing and makes the network quite inefficient. DHT-based systems do not implement the protocols that can create communication for file sharing among the peers. Although the content-based file searching through DHTs always targets the resources effectively, but the look-up latency through hop by hop procedure is not a desirable situation. Hence structured P2P overlay network using DHTs is not proven to be an ideal environment for content-based file searching. A lot of research is going on to improve the routing algorithms of unstructured P2P infrastructure in terms of time and scalability.

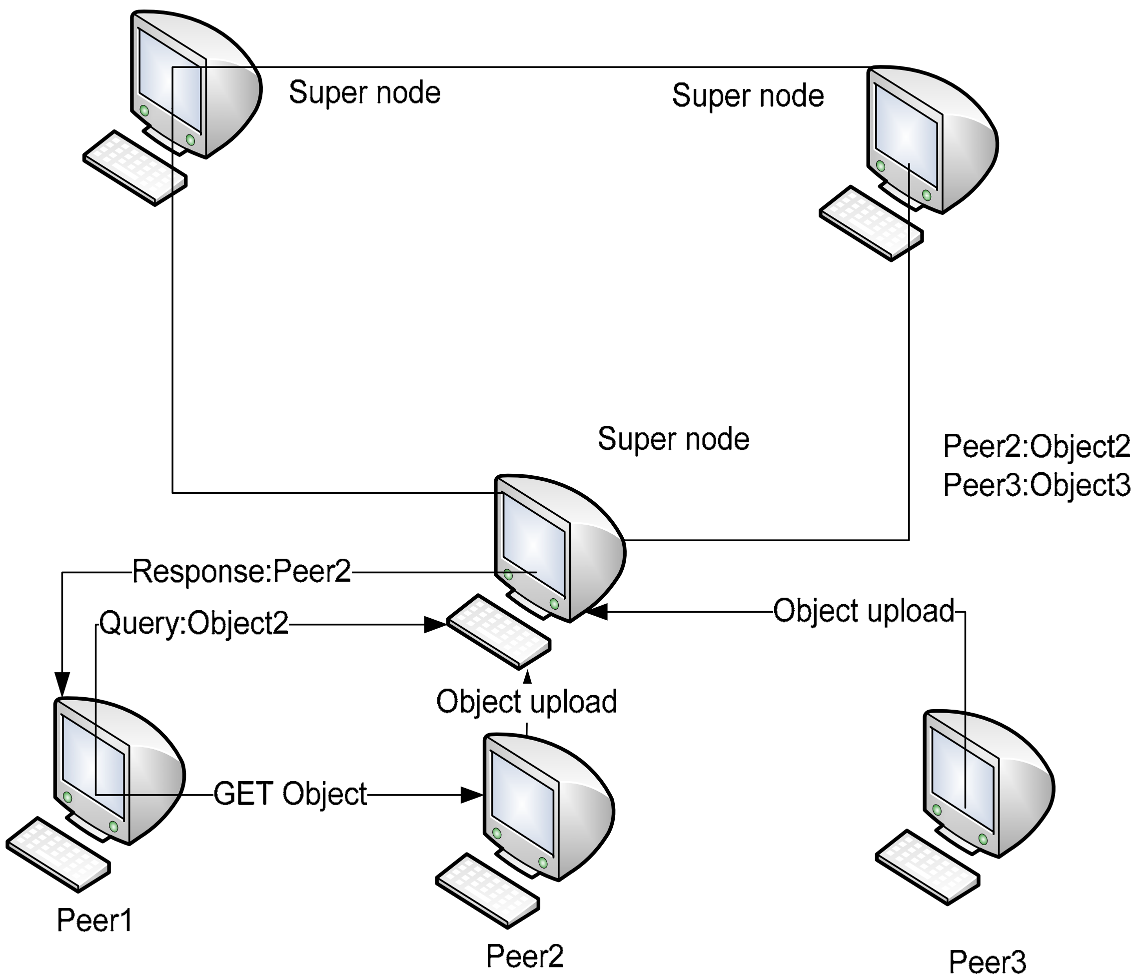


Figure 1.5: Fast-Track peers connected to super peers

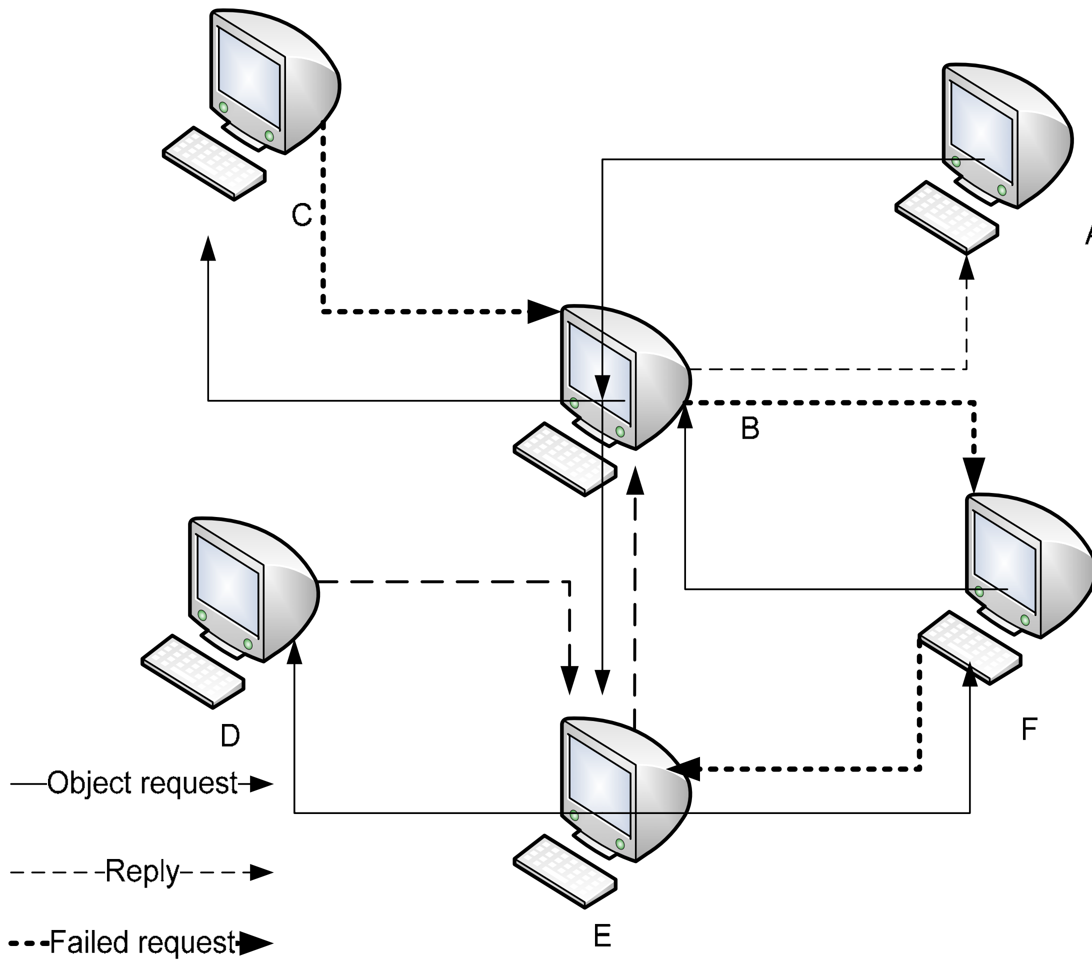


Figure 1.6: A typical request sequence in Free-net

1.7 Problems related to P2P Overlay Networks

As P2P overlay networks handle the type of networks which needs an extensive enhancement of features like management of peers according to network capacity in the network, where the peers are joining to share or retrieve resources from other peers. If any peer in the network has some extra resources to share with other peers, then it is quite difficult to create an environment of cooperation between the peers. In P2P networks, generally, the request or the availability message for a particular file or service is broadcasted to other peers in the network, which incorporates not only a load on the network (as every node has to pass on these messages to other nodes in the network), but also makes the network vulnerable to free riding, a behavior in which nodes receive messages, but do not want to invest their resources for forwarding them to neighbors [3].

1.7.1 Free riding

Optimum utilization of resources in P2P networking yields collectively desired results. This aspect is achieved by a simple technique of replicating a file downloaded by a particular peer, to the database of file sharing community. This technique is vulnerable to free riders who can deny access to a particular file by immediately moving it to some other location after download, thus they avoid increasing size of shared database. Free riders are the participating peers in a network who utilize the available resources, yet they do not provide or share their available resources. This behavior creates a sizeable hindrance in development of a P2P network to its full capacity. Free riding reduces the availability of information and degrades the network performance.

1.7.2 Lack of Cooperation in Peers

The research and development in the area of P2P computing is a continuous and ongoing process as it carries a lot of importance, regarding resource sharing aspects of a network. When peers in the network are not free to make their resources available, they incorporate a lack of cooperation among the peers, as every peer in the network desires to use the network and other peer’s resources, but most of them do not want to share their files and services (resources) with other peers. This lack of cooperation spreads a behavior of free riding in the peers, as discussed earlier. Hence lack of cooperation threatens the overall ideology and objectives of P2P computing.

1.7.3 Trust and Security

P2P systems require different entities to decide how, and in which conditions one peer interacts with other peers in the network. Such decisions could be based on security and trust factors associated during the phase of communication between two peers. The peer should itself allow particular types of interaction, or to allow particular choices about interaction. Within many P2P systems, a peer needs to know whether it can "trust" another peer within a particular entity. "Trust" is a word that is used very casually in English, but a concept that reflects the thoughts of anyone thinking about security in a network or computer system, particularly when that system is distributed, and even more so when it is a P2P system.

1.7.4 Bandwidth Congestion

The most highlighted problem with P2P file sharing programs is bandwidth congestion. As thousands of nodes (peers) are part of a P2P network, there are several procedures to search a particular file or service, mainly related to broadcasting queries or advertising messages. All the nodes in the network are sending or passing on these query messages, hence many of these messages are broadcasted unnecessarily. This phenomenon increases the response time for internal users and e-business customers. Virtual Private Networks (VPNs) [28] have been established by various organizations to interconnect their distant offices or mobile users through internet link. But the performance of the VPNs suffers due to increase in non business file sharing traffic in comparison to legitimate e-business communication.

1.8 Summary

In Peer-to-Peer overlay networks the dynamics affecting the effectiveness of a network are cooperating behavior of a node, security related issues, network capabilities like link bandwidth etc, hence a stable and manageable infrastructure is required to enhance these areas in P2P computing. The current research intends to achieve a level of cooperation where nodes are compelled to cooperate in order to remain part of a network. To implement the algorithm in order to achieve the desired level of cooperation, a Heterogeneous Cooperating Group-based Newscast Protocol (HCGNP) is introduced where peers become the part of a network by attaining a cooperating-group ID, and the rate of change of a particular cooperating-group ID assigned to a peer shows the pace of cooperating behavior of a node. Peers are provided with some benefit from the network depending on their cooperative behavior. This benefit depends upon the network capabilities.

The implementation has been carried out in PEERSIM simulator, using java (jdk1.6.0) as the backend supporting language. The cooperation level associated to the peers in different cycles of simulations, has been observed and calculated. These observations prove that the current approach provides much better platform to achieve cooperation efficiently, without wasting network and machine resources.

P2P networks refer to the communication between two or more individuals (peers) at the same time without any centralized authority. Such P2P systems are termed as decentralized P2P systems. But there are architectures which are considered as the part of P2P networks with some centralized authorities. Pure P2P networks, without centralized authority are commonly known as P2P overlay networks. P2P overlay networks could be structured or unstructured. In structured P2P overlays, the required resources are searched through distributed hash tables (DHTs), where file search is more efficient as the relevant file is targeted easily through DHTs, with the help of keys associated to the different required files. This method requires extra resources and network capabilities to maintain a database related to different resources in the DHTs. However, in unstructured P2P overlays, although file search is not very efficient, as query messages are broadcasted to all the nodes in the network for a required resource, it is quite easier to manage the overlay architecture as it does not require extra efficient network capabilities and maintenance. Ongoing research in P2P overlay network emphasizes more towards enhancing the efficiency in unstructured P2P overlays. The areas like handling free riding, enhancing cooperation and security with maximum congestion control and network management problems require a lot of emphasis in the future research.

Chapter: 2 Enhancement of Cooperation in P2P Overlay Networks

2.1 Introduction

P2P Overlay Networks require a platform where different peers collaborate with each other in order to share each other's resources. It is an obvious fact that the phenomenon of resource sharing is not possible until a particular peer is ready to share its resources like files and services. Sharing these resources require a cooperating behavior which should be indulged into the peer configuration or network architecture so that the problem of free riding could be eliminated by enhancing cooperation among the peers.

2.2 Previous Research

To enhance cooperation in P2P overlay networks, an extensive research has been carried out. To maintain this trend, algorithms like SLACER [4] and SNA [5], both using incentive mechanism, have been implemented. SLACER implemented Prisoner's Dilemma game to improve the rate of cooperating nodes in the overlay network where bandwidth link is taken as a constant and a realistic value. Similarly there is a Social Network Architecture (SNA) suggested by W. Wang, L. Zhao and R. Yuan, which enhanced cooperation by introducing group of peers formed on the basis of similar interest of the peers. Peers in each group adopt cooperating behavior when they are provided with some incentives. The current research also emphasizes on enhancing cooperation in peers by implementing Newscast Protocol, hence the detailed discussion on Newscast computing, SLACER algorithm and Prisoner's Dilemma Protocol is elaborated below.

2.2.1 Newscast Protocol

A gossip-based protocol that develops and sustains a random and dynamic overlay is known as newscast protocol [1]. It results in a stable topology which ensures a fault resilient connectivity. Various P2P protocols such as broadcast [29] and aggregation [30] have been successfully implemented utilizing this protocol. A fixed 'C' size node descriptor containing node's address and time stamp is used to represent the newscast state [1].

In this protocol, the neighbor selection process is executed randomly using the SELECTPEER ( ) method. The cache of every node is traversed and the cache list is transmitted to the neighboring nodes through SENDSTATE ( ) method. After receiving a message transmitted by a node through SENDSTATE ( ), the UPDATE ( ) procedure mixes this received view with the node's current state view. This merged view is trimmed by the Newscast to obtain the predefined *'C'* size. Figure 2.1 illustrates the procedure adopted by newscast protocol. It must be noted that the most 'old' node descriptor is discarded in each cycle, thus continuously altering the node descriptor hold for each node view. This technique updates the overlay defined by a set of all node views.

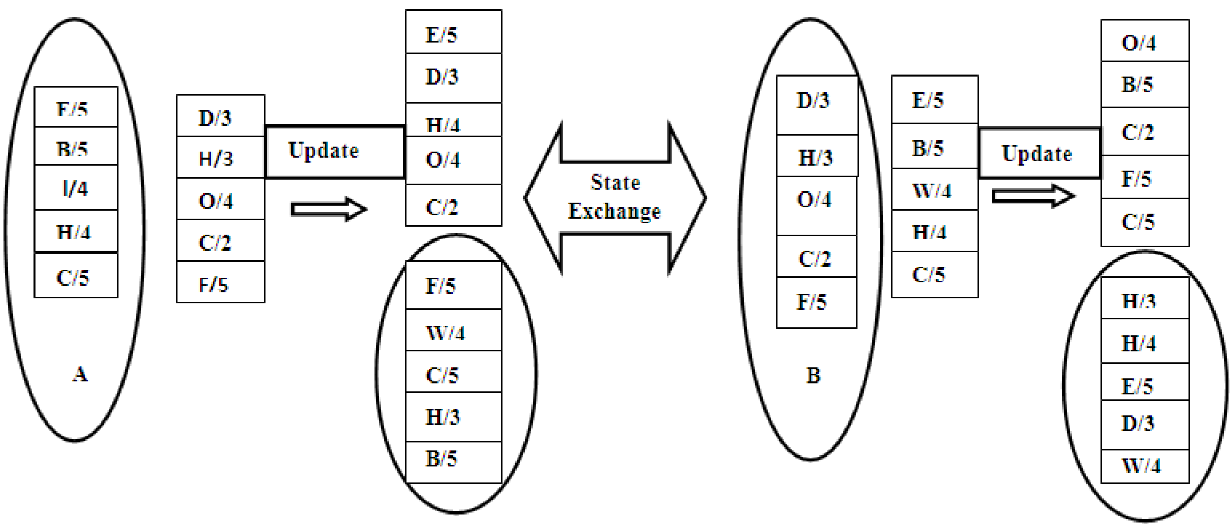


Figure 2.1: Newscast maintaining the freshest list of nodes

2.2.2 SLACER Algorithm

SLACER (Selfish Link-based Adaptation for Cooperation, Excluding Rewiring) algorithm [4] was introduced in 2005 to optimize the cooperation among participating nodes in a P2P overlay network. This algorithm considers the link bandwidth as a realistic value. In this algorithm, the participating nodes are at liberty to alter their behavior and establish connectivity with desired nodes. Peers can modify the request dispatch procedure and data handling technique with other nodes. The participating nodes are also delegated the ability to randomly choose other nodes from the network.

Therefore the nodes try to utilize their abilities to selfishly enhance own utility in a greedy and adoptive manner i.e. if altering some behavior increase utility then nodes tend to adopt it [4]. The algorithm is based on selfish link and behavior adaptation to enhance cooperation among nodes. The salient features of the algorithm are enumerated below:

(a) After a specific amount of time the nodes get busy in any activity.

(b) After that node makes some measurements of the utility U.

(c) Utility refers to the amount of data downloaded depending on the particular set of nodes encircled in a domain.

(d) In a fixed time interval, every node (n) compares itself with node (m), which is generated in a random manner.

(e) The node selection is also a random procedure.

(f) If utility of node (n) 'U_n' < utility of node (m) 'U_m', node (n) cancels all its connections and copies all node (m) connections.

(g) Using mutation process every node copies its connections in randomized manner implementing a random change in the behavior of nodes.

2.2.3 Prisoner's Dilemma Game

The prisoner's dilemma is considered as a basic problem in game theory that explains the reasons for non cooperation among two people even if it is in their best interest to cooperate. Merrit Flood and Melvin Dresher (1950) initially introduced this theory [31]. Albert W. Tucker formalized the game with prison sentence pay offs and suggested its name as Prisoner's dilemma. The "Prisoner's Dilemma" (PD) in classical form is illustrated below:

(a) The police have arrested two suspects.
(b) Due to insufficient evidence for conviction, both of the prisoners are separated by police and they are offered with the same deal in isolation.

(c) If one of the prisoners defects from other and testifies for prosecution, while the other prisoner cooperates with his partner by maintaining silence. The first one is set free and silent accomplice is punished with 10 years sentence.

(d) Both prisoners are sentenced to jail for six months if they remain silent.

(e) If both the prisoner betray each other, they both are sentenced to jail for five years.

(f) Each prisoner may either remain silent or is supposed to betray the other one.

(g) Before the end of investigation, none of the prisoner can come to know, if other has betrayed.

It is pertinent to note here that basically cooperation is strictly dominated by defecting in this game. The best possible balance is achieved in this game if all players defect. Any player can obtain larger pay off only by defecting, irrespective of other player's response. It is obvious that playing defect *'D'* pays more dividend rather than cooperating *'C'*. So this situation forces all rational players to defect. SLACER implemented prisoner's dilemma game by giving the following payoff values in Table. 2.1:

	C	**D**
C	**R,R**	**S,T**
D	**T,S**	**P,P**

Table: 2.1- Payoff values given to nodes according to PD game

Here, the payoff values assigned are exactly given the following values:

- T=1
- R=0.8
- P=0.1
- S=0

Hence, a node replicates by mutating itself from neighboring nodes to enhance its performance. The behavior of the node is similar to the neighboring node and moves from lower to higher utility area.

2.2.4 Social Network Algorithm (SNA)

To enhance cooperation in P2P overlay networks, SNA proposed an incentive mechanism which maintains the history of the peers by keeping a track of good and bad peers. Good peers refer to those nodes that have cooperated and bad peers refer to those that have shown free riding. This approach allows peers to maintain their own group of friends. By friends it means those peers who have common resources to be shared among each other. Each peer maintains the history of the transaction which takes place between himself and his friend (another peer in the same group). If any resource is not available at the neighboring peers, the request is transferred to the next level by increasing the TTL (time to live) value of the peer. This algorithm elevates the efficiency of resource retrieval by selecting the shortest path between two peers.

2.2.5 Problems Related to SLACER and SNA Algorithms

SLACER implemented Prisoner’s Dilemma Protocol (PD-Protocol) and utilized network capabilities and resources in order to gain a trend of cooperation among the nodes. These network capabilities are the payoff values assigned to the nodes which do not tend to cooperate in the initial cycles of simulations. The trend of cooperation can be seen by expanding network resources, which can over load the network. Apart from the fact of overburdening the network, the nodes tend to show cooperating behavior which does not guarantee cooperation among the nodes. Hence SLACER developed an environment in which the nodes can behave cooperatively, but assurance of cooperation is not considered. On the other hand SNA keeps track of the nodes which show cooperation by maintaining the node’s history, but again overburdens the network when the path between any two nodes is very long. In other words the long TTL value affects the performance of the mechanism.

2.3 Proposed Solutions –Heterogeneous Cooperating Group-based Newscast Protocol (HCGNP)

Since P2P overlay architecture is a layered approach, there is a need to maintain an architecture which should utilize network resources with minimum load over the network. Although P2P overlays require an extra workload on the architectural area of implementation, but once a fine architecture has been created, the throughput of the networks accelerates with the passage of time.

After a thorough analysis of previous research the following facts could be extracted for proposing a well defined solution:

(a) There is a need of an architecture which should control the nodes according to the network capacity.

(b) The node search should be more efficient.

(c) The criteria for giving extra benefits to the nodes should depend upon the type of the network and its capacity.

(d) Only those nodes should become part of the network, which show cooperating behavior.

(e) Nodes that do not cooperate should be discarded from the network, which would automatically decrease the network load.

(f) The behavior of a particular node should depend upon the capability levels of the node.

(g) These capability levels should be according to the node's physical parameters, like availability of extra secondary storage, RAM and processing power.

(h) The diversified behavior of a node, i.e. it may require a resource as demanded capability level and another resource as available capability level, should be there to achieve certain level of cooperation with other nodes.

(i) This diversified nature of the node also reflects the heterogeneous nature of the network, which exists with variable values of physical parameters.

(j) Thus, the overlay architecture should be capable of incorporating cooperation among the nodes, along with the ability to handle the network with heterogeneous capabilities.

(k) The nodes which are discarded from the network could join the network again, with some different capability parameters, to have a fair share and gain of other's (nodes) resources.

(l) Whenever a node joins a network, it becomes part of cooperating group, a group of nodes where every node is cooperating with the other node.

(m) The approach for development of cooperating groups should be implemented to enhance the efficiency of node search procedure. When the nodes search the other nodes according to the cooperating group IDs assigned to them, the size limit of CG shows weather the nodes are idle or waiting for other nodes in the network.

2.3.1 Groups or Clusters in P2P Overlay Networks

The clustering approaches are introduced in a network to share the computer resources of the nodes in the network. A cluster is used to incorporate resources of multiple independent nodes in the network in order to perform a common task or meet a mutual objective.

Conceptually all the web servers which are interconnected through a network and have an access to the data on a particular site, perform as a cluster. However, it must be understood that network clustering has been evolved with lot of research and development and various types of clustering techniques exist with a historical background. Up till now, many clustering approaches have been implemented to create a multi-sharing environment in the network. Through clustering it is much easier to increase the availability of multiple resources, residing in a very small quantity, anywhere in the network [6].

Mostly grouping or clustering approaches have a group or cluster head, which is known as the super peer of that group or cluster. However, there are approaches in which this hierarchy is not followed. In case of P2P overlay networks, non hierarchical groups can be developed, which do not have any super peer concept in the network. As pure P2P overlays are totally decentralized, the non-hierarchical concept in the groups or clusters compel the nodes to participate and share the network load equally. Grouping of the nodes is also helpful to organize P2P overlay network so that the requests are routed more efficiently. Hence the nodes grouped together to share resources in such a way that every node gains a fair share of demanded resources from other peers. It could be an ideal way to eliminate nodal search extended to every single node in the network. The efficiency of the network to retrieve information regarding the files and services available on the other nodes, is one of the most required and demanded property of a particular P2P network. The implementation of groups not only minimizes the time to search the required resources, but can also give a free hand to manage the nodes on the application level in order to generate security protocols like authentication and to maintain a network data and node level information for further developments.

2.3.2 Heterogeneous Cluster-based Newscast Protocol (HCNP)

The current emphasis of this research is on designing an architecture which develops cooperating groups (CGs). The previous research work [18] done in the same direction proposed a protocol named Heterogeneous Cluster-based Newscast Protocol (HCNP) which lacks the concept of cooperation among the nodes, but utilized the network resources by forming clusters in the P2P overlay network. The clusters developed are based on multiple physical resources, termed as physical parameters, such as empty storage capacity, processor cycles, RAM etc.

Each cluster is comprised of heterogeneous nodes having diversified physical parameters to provide maximum utility out of each node's capability. Initially, each node maintains its capabilities that can be useful for other nodes. Then the capability level (L) of one node is compared with the capability level of the other nodes as shown in Figure 2.2. On the basis of those capability levels, the nodes become part of a cluster and start comparing these capability levels. The size of the cluster is configurable and depends on network size. The maximum number of nodes that can join the cluster depends on the size of network as well the size of the cache associated to each node. The cluster is configured in such a way that fixed number of nodes with same capability levels can join the cluster at one time. Through this configuration maximum utilization of the network resources is possible as each cluster contains nodes with heterogeneous capability levels. This variation maintains a justifiable amount of nodes in a particular cluster.

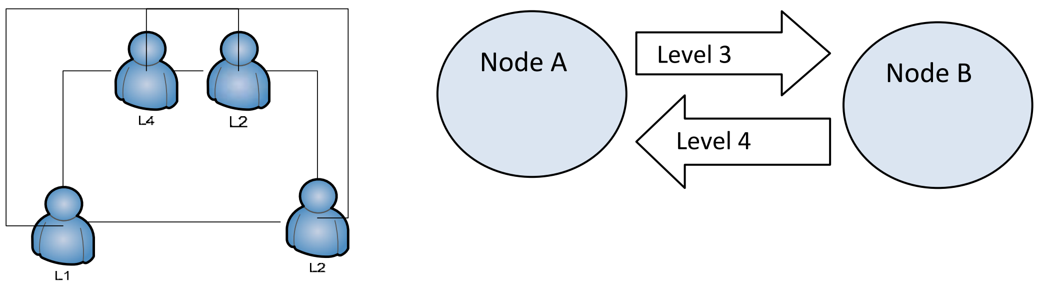


Figure 2.2: Clusters with nodes of different exchanging capability levels

HCNP is confined to the following two components:

(a) Peers Capability exchange and Capability level assignment.

(b) Clusters based peers configuration.

These two components of HCNP deal with measuring or calculating node's capability level values and assigning cluster id according to that capability level. Therefore, each node will have to maintain the data structure shown in Figure 2.3.

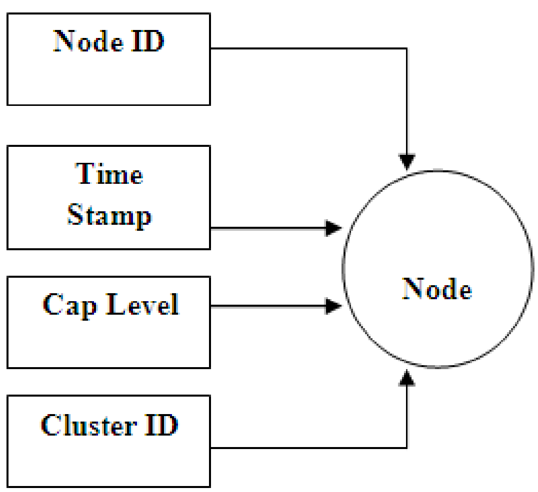


Figure 2.3 Data structure assigned to HCNP

On the basis of capability measurement and capability level assignment algorithm, each node (peer) can be configured to become part of a particular cluster. The following steps are followed to assign cluster ID to a node:

(a) A node will be configured of its neighboring nodes on the basis of predefined capability levels arranged in ascending and descending order. A cluster will be composed of heterogeneous capability levels. The size *(s)* of a cluster is fixed initially but can be configured dynamically according to the network size.
(b) The cluster size is less than or equal to the cache size to avoid cache-miss (occurs when a node is not found in the cache), conditionally the degree [2] (size of views exchanged between the peers) is equal to the cache size. As *peer-sampling-service* (PSS) is used in NP, the cluster size is restricted by one sample size.
(c) A Cluster ID (0 for non-participant and an integral value for participant node) is assigned to the node showing its participation to a cluster. This is maintained by each node's data structure and will be accessible to every node in the suburbs of a node.
(d) Each node maintains only one Cluster ID in its data structure at a time to avoid multiple cluster participation.
(e) The most important point is that one cluster can never contain all the nodes of same capability level, thus assuring heterogeneity.
(f) Each node will refresh its neighbor's list as well as its capability level after each cycle.
(g) The overheads (Cluster ID & Capability level) are handled implicitly by maintaining the freshest data structure in each cycle.

After performing the configuration on each node according to the above mentioned steps, a set of clusters is developed. The number of clusters depends on the size of the network and the length of the cache associated to it. The cache length can be configured according to the network size. The rate of change of cluster ID associated to a node depends on the rate of change in the capability levels, associated to each physical parameter of node. The simulations further explain that the total number of clusters developed at any instant of time in one cycle depends on the size of cluster, that has been configured according to the network size and length of the cache associated to a node. In Figure 2.4 it can be seen that in a network of 1000 node the total number of clusters developed at any instant of time (t) is 25 approximately. The number of clusters increases gradually with the increase in network size, i.e. when network size reaches 12000 nodes, the number of clusters developed is 147. When the network size reaches 16000 nodes the number clusters increases to 200. Mathematically when the network size increases with the power of 2, the number of clusters increases with the multiple of 2 when the cluster size is configured as 30 nodes, hence the number of clusters formed in a network at any instance of time *t,* is given as,

No. of clusters (t) = network size/cluster size

It can also be seen that the implementation of cluster-based protocol is worth and use full when the network size is at least thousand nodes. As implementation of any protocol at any layer of the network requires extra expenditures and resources, the tradeoffs should be kept in mind before implementing any protocol in the network.

Simulation results [18] also reveal that the efficiency of HCNP performing over different clustering models for content-based searching is very efficient. It can be seen that the cluster accuracy rate in terms of time to search the relevant resource is very fast. This efficiency is gained while running HCNP in parallel to selfless clustering algorithm and selfish clustering approaches proposed in Schelling model [7]. Hence HCNP improves the clustering accuracy rate twice the accuracy rate of selfless and selfish clustering models performing independently. The performance of selfless and selfish clustering models in the presence of HCNP can be seen in Figure 2.5. Here, when selfless clustering algorithm runs independently, the maximum accuracy rate to search for the desired neighbors is approximately 17%. On the other hand, when selfless clustering algorithm runs with HCNP on the top, the maximum accuracy rate of the protocol is increased to 24%. This results in a better performance of selfless clustering algorithm in the presence of HCNP. But it can also be seen that the overall performance of selfless clustering algorithm in the presence of HCNP is not as efficient as the performance of selfish clustering algorithm running independently.

The maximum accuracy rate of selfish clustering algorithm to search the desired resources in the network is approximately 50%. This accuracy rate is achieved when selfish clustering algorithm is running independently. The gain in the accuracy rate of selfish clustering algorithm can be seen when HCNP runs on its top. Hence the overall gain in the accuracy rate is approximately 90%, which is a very desirable approach to implement clusters in the presence of selfish clustering algorithm as well as HCNP.

Both selfless and selfish clustering algorithms are based on content-based searching i.e. discovering the neighbors with required resources like files or services. In the presence of HCNP, selfless and selfish clustering algorithms become very efficient, as prior to the search for a specific file, the network detects the capability level of the node. Hence HCNP can report the selfless and selfish clustering algorithms, that whether the specific node is physically capable of holding a large file or data.

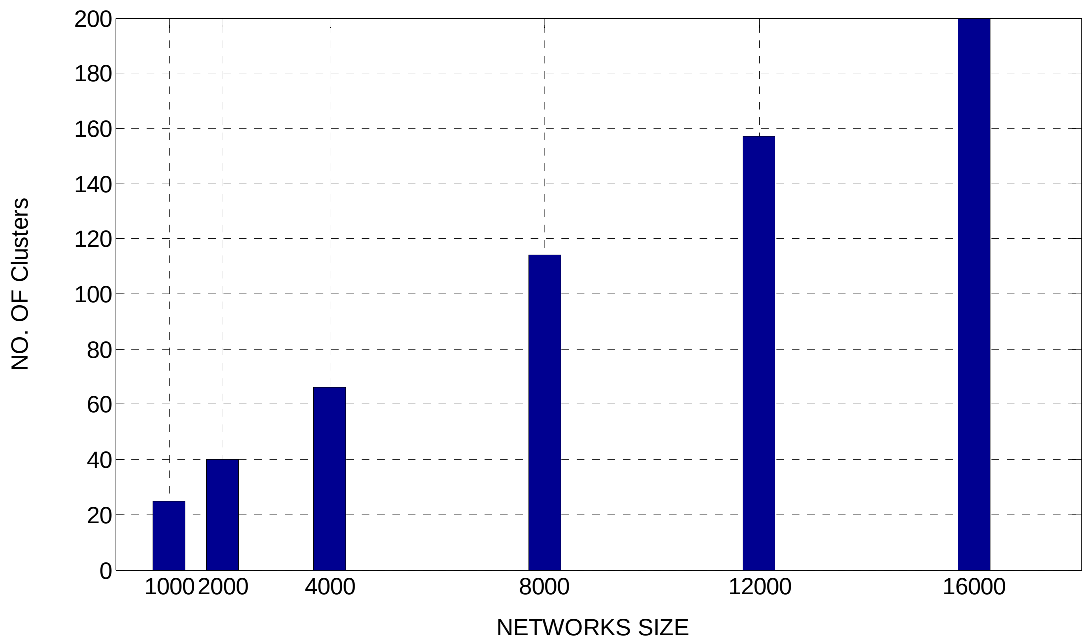


Figure 2.4: Number of clusters formed in a network at any instant of time (t)

2.3.3 Problems Related to HCNP

The previous research deals in single physical parameter, i.e. storage capacity of a node. Some new and different results can be observed if clusters are formed using multiple physical parameters, such as identification of a cluster carrying maximum nodes of a particular physical parameter. Hence the areas need to be handled are:

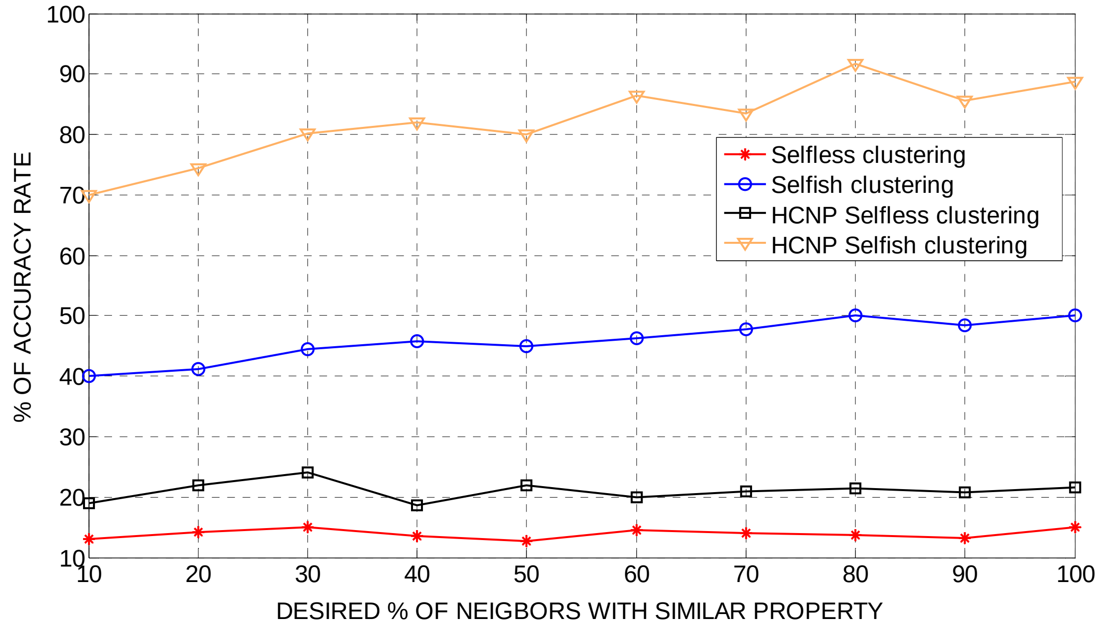


Figure 2.5: Percentage of accuracy rate of selfless and selfish algorithms with HCNP

(a) Cooperation among the peers

As discussed earlier the basic objective of P2P computing is to distribute the network resources among the peers, without any decentralization. Thus, peers are required to cooperate with each other in the network. The behavior of free riding in which the peers stay in the network to gain benefit from the other nodes or the network, incorporates a trend of non-cooperation among the nodes. The implementation of HCNP shows much better performance of the nodes in the network for resource searching procedures. But the lack of cooperation can make the implementation incomplete, as it lacks one of the most important objectives of P2P networks.

(b) Secure transmission and retrieval of resource information

Security and trust is another problem which is not handled in HCNP. There is a need of a security protocol at the architectural level, which should be embedded in the data structure assigned to each node in the network. This is required because when a peer communicates with the other peer, it can access the resources and other information of the peer. This may lead the peers to perform some malicious activity which is obviously not a desirable situation for a network. HCNP is not capable of detecting any malicious activity in the network as the information regarding the physical parameters of any particular node is neither encrypted, nor any hashed procedures have been implemented. Besides, the authentication protocols are also required to be implemented in order to maintain a trust worthy and secure environment in the network.

(c) Interoperability between the nodes

Interoperability is related to the way of communication between the nodes. The ease in node search and data transfer phase enhances the interoperability between any two nodes. Although HCNP provides a view to the peers related to the information about other peers, it lacks a platform through which nodes could communicate with the transport layer protocol on the basis of virtual links between them.

(d) Node management (maintaining the nodal states)

HCNP does not maintain a nodal state at a particular time instance. The time of a node to join a network, its capability level and the cluster ID assigned to each node at a particular instant of time, are required to be maintained in order to retrieve information regarding a particular node.

(e) Overlay network management at each layer

Like node management, HCNP also needs to maintain the network. For instance, if network administrator requires a state of a network at a specific time instance, then it needs to maintain the history of each and every node in the network, e.g. when a particular node joins a network, what cluster ID was assigned to that node. Hence node and network management procedures are almost similar to each other, but in network management the overall scenario of a network needs to be handled.

Another very important reason to implement clustered approach is that the nodes can be compelled to cooperate in the network as the capability levels of the nodes are extracted implicitly. Clustering approach enhances the quality of service (QoS) of a network in a particular environment and further implements protocols that can handle security, multicasting and network and node management data.

2.4 Summary

Enhancement of cooperation in P2P overlay networks is prominent and desirable phenomenon that needs to be handled by keeping in view the network capabilities. There has been a lot of research going on to enhance cooperation among the peers in P2P overlay networks. The protocol based on enhancing cooperation is developed while implementing Prisoner's Dilemma game on newscast protocol. This research implemented algorithms like SLACER or SNA which incorporate cooperation among the peers with the payoff values given to the peers according to Prisoner's Dilemma game rules. The setback of these protocols was extra utilization of network capabilities in order to give payoffs to the nodes, to achieve cooperation. The proposed solution to this problem is the implementation of Heterogeneous Cooperating Group-based Newscast Protocol (HCGNP). This protocol instead of running on top of newscast protocol is embedded in the newscast protocol. The implementation of HCGNP is based on Heterogeneous Cluster-based Newscast Protocol (HCNP). This protocol develops clusters on the basis of physical resources of the peers in P2P overlay network. Through the development of clusters the nodal capacity to download a file or service is pre determined which improvises the node search procedure. Besides this, HCNP also deals with heterogeneous environment of a P2P overlay network, by developing clusters on the basis of physical parameters of the peers through which peers not only share files and services but also share the physical resources like RAM, free storage space and processor cycles. The drawbacks related to HCNP are the lack of cooperation, security, node and network management and handling the interconnectivity among the nodes.

Chapter 3: Implementation of Heterogeneous Cooperating Group-based Newscast Protocol (HCGNP)

3.1 Introduction

In a heterogeneous P2P overlay network environment the peers possess a heterogeneous nature. The heterogeneous network is such that it has peers with different types of operating systems, memories and storage spaces. Such environment in the network needs a challenging architecture to be implemented. Implementation of cooperating behavior, security and network management issues are required to be handled in P2P overlay networks. The objective of P2P overlay network is to share resources and in the real time networking environment, the nodes (peers) with the heterogeneous properties need to collaborate with each other. The architecture handles the network heterogeneity and also incorporates quality of service. This chapter explains the existence of heterogeneity of the peers in the P2P overlay network and discusses different behavioral aspects of the peers accordingly. The role of cooperating nodes in the heterogeneous environment is discussed in detail where nodes cooperate on the basis of some benefit given to them in the network.

It has also been discussed that how cooperation can be enhanced by development of cooperating groups in a P2P overlay architecture. The nodes become part of a P2P overlay network by joining a specific group with no super peer, which means that there is no hierarchy in the network. Formation of groups enhances cooperation by identifying the node with its cooperating group ID (CG-ID). The node (peer) shows cooperation by the number of times it changes the CG-ID. In this case a payoff value is assigned to those nodes which show maximum cooperation. Through CG-ID it is also possible to enhance the node search where nodes with the required resources can be searched more efficiently.

3.2 Cooperation in Heterogeneous Peers (HP)

Heterogeneous peers (HP) as discussed earlier, have very diversified nature in the network. All the peers are virtually connected to other peers in the overlay network where they have difference in their operating systems, network connections, processors, RAM and memory. Hence the important aspect is to maintain an architecture in which peers can collaborate with each other without incorporating free riding, malicious threats and overload to the network. P2P systems consist of nodes that are able to interact with each other and for sharing resources such as files, contents, processor cycles, storage space and bandwidth [6]. Major application areas of these systems include distributed and scalable computing, internet service support, database systems, content distribution, communication and collaboration.

To support content transfer applications in heterogeneous P2P networks there is a need of following services:

- A resource discovery service is needed to search and locate resources.
- A content distribution service is required to reliably transfer the requested content (located via the resource discovery service) to a set of peers.

Up till now the resource discovery mechanisms developed are mostly for a semi-centralized P2P architecture, where the super nodes receive the resource request from its connected nodes and transfer it to the other super nodes [10]. To maintain a resource discovery mechanism in pure P2P overlay networks with heterogeneous peers where there is no centralization authority there has not been a precise and efficient mechanism developed. Many experimentations and techniques to enhance cooperation suggest that heterogeneity hinders efficiency, i.e. it reduces the overall collaboration between the nodes [11].

Clustering (grouping) the nodes together is one of the methods to enhance resource discovery. Bayesian statistic analysis proposed a level of trust between the nodes for file sharing [12]. Here the criterion for changing a cluster is just based on the relevant file required by the node. Mostly clusters are formed in a hierarchical manner. This hierarchy is maintained by introducing the concept of super-peers in the network. The network topology forms layers of nodes in a hierarchical way starting from super-peer to a single node. Many overheads and extra protocols are required to manage the nodes hierarchically, and the resource sharing is content based particularly for multimedia files. A. Yonezawa et.al [16] proposed a clustered approach over multicast overlay networks by introducing a cluster coefficient which is maximum physical number of hops between any two nodes. But this approach fails when the cluster size increases.

Data can be searched by building an overlay network on the basis of node's potential [17]. In this technique node has to search the relevant file across the entire network, giving room to an inefficient environment in the network. All the approaches for the formation of clusters work under some limitations and yet all of these approaches give different ideas to search and share the desired file in a purely hierarchical manner. Hence the content based cluster formation requires a strong architecture where content based file sharing could be enhanced on the basis of node's physical parameters. These physical parameters explain the existence of heterogeneous environment in P2P overlay network which needs to be handled before the start of file sharing procedure.

3.3 Cooperating Group Formation for Resource Sharing in HP

Grouping the peers together in P2P architecture is not a unique concept as there has been an extensive study going on in this direction. Most of the work that has been done is related to the phenomenon of grouping the same peers together. This similarity is mainly based on the same content that a peer has to share. This content is related to a particular file or service and the resources that are required to be shared are normally application based. In many different researches this grouping of peers together is termed as cluster formation. But this research is based on development of cooperating groups of peers with heterogeneous physical parameters in order to share physical resources as well as to improve the efficiency of node carrying relevant resource.

The study on the issues likes secure clustering and grouping of peers in P2P networks has also elevated the importance of grouping in P2P networks. In 2007 S. Wang and Y. Zhang presented a reliable self clustering methodology by evaluating the level of trust between the nodes which can maintain the autonomous behavior of the nodes by publishing partial knowledge of the resources [13]. Similarly in 2008 M. Amad and A. Meddahi presented an optimized flooding and clustering based approach for increasing the efficiency of content-based file search in the network [14]. This approach is also based on enhancing the search of a peer according to the required resource, where the formation of clusters is hierarchical and message to acquire a particular resource is flooded in the network with nodes having certain TTL values. This technique is scalable and practical as it deals with problem of disconnection during any session of a P2P communication. The condition for grouping the peers non-hierarchically is that each peer (node) should participate equally and there is no super peer who has the entire maintenance load to manage the group. Some cluster formation methods like Schelling model

[15] for data sharing implements sociological protocol. Nodes after becoming the part of a cluster, decide whether to stay in the network, on the basis of relevant resource discovery. Schelling model implements selfless and selfish clustering algorithms. In selfless clustering algorithm, node gets disconnected with the neighboring node if the neighbor is connected with the other nodes. While in selfish clustering node gets disconnected from the neighboring node even if it has no other connected node.

3.4 Heterogeneous Cooperating Group-based Newscast Protocol (HCGNP)

The current research is an enhancement of our previous research based on formation of clusters in P2P overlay networks [18]. These clusters were formed on the basis of physical parameters of the nodes, i.e. storage space, RAM, processor speed etc, in such a way that cluster ID is assigned to different nodes on the basis of capability levels. These capability levels are different metrics values assigned to a single physical parameter. The current research enhances this formation of clusters by adding an environment of cooperation between the nodes in a P2P overlay network. This environment is provided to each node by adding an idea of on-demand peer configuration over heterogeneous cluster-based newscast protocol (HCNP). On-demand peer configuration to achieve cooperation is a procedure in which nodes join or leave a particular cluster or a group on the basis of its available capability level (ACL) and demanded capability level (DCL). The values of available and demanded capability levels decide nodes which are going to cooperate. Each node maintains a cache in which it holds the information of its neighboring nodes. In every cycle these caches update themselves to maintain the freshest list of the neighboring peers. This improvises the idea of maintaining cooperation among the nodes in a network where nodes continuously join or leave the network.

The basic objectives of the current research are:

(a) To implement a protocol that develops non-hierarchical cooperating groups.

(b) Enhancement of cooperation by developing **cooperating groups (CG)** in Peer-to-Peer (P2P) overlay networks in a heterogeneous environment.

(c) The idea of resource sharing is related to the sharing of files, services and physical resources, such as RAM, free storage capacity, processor cycles, etc.

(d) Nodes that become part of a cooperating group have to cooperate with the other nodes in that cooperating group.

(e) If a peer finds its compatible peer such that both the peers share each other's resources (storage, processor cycles, etc) simultaneously, Cooperating Group Identifier (CG-ID) is assigned to both the peers.
(f) To find the cooperation level of peer by the rate of change of its CG-ID
(g) To identify the cooperating group with maximum cooperation.
(h) To observe the rate of cooperation without giving any benefit 'ß' to the node.
(i) Categorize the nodes with maximum rate of cooperation for rewarding additional benefit.
(j) After rewarding additional benefit, rate of cooperation is again observed.
(k) To observe ß-Cooperation (cooperation level achieved after giving benefit to the nodes) between the nodes
(l) To mark Tradeoffs between ß-cooperation & cooperation according to the network capacity.
(m) To compare simulation results with other researchers work in the same direction.

The data structure associated to each node, the algorithm of the protocol, data flow diagram, details of each iteration and the simulation model of the algorithm are discussed in the sections below.

3.4.1 Data Structure

The data structures assigned to each peer is according to the newscast protocol with various modifications. In newscast protocol the node has been assigned a "node-ID" and a "Time-stamp" value which is just like TTL values associated to any message segment. Each node maintains the information of the neighboring nodes in a cache associated to it. The size of the cache is according to the network size and is configurable through configuration file. In each cycle during the simulation of newscast protocol the nodes with minimum time-stamp values are replaced with new nodes in the network, as in the real time situations the nodes randomly join or leave the network. The data structure associated to each node has been implemented accordingly. To incorporate a clustered (grouped) approach in a P2P overlay network along with the achievement of cooperation among the nodes, the data structure assigned to each node in HCGNP is shown in Figure 3.1.

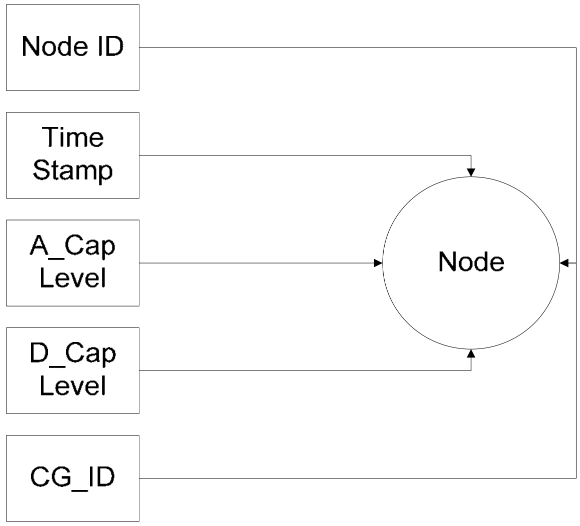


Figure 3.1: Data structure assigned to the node in HCGNP

In HCGNP:

(a) Physical parameters to be shared among the peers could be:

- (i) Node's free storage space
- (ii) RAM
- (iii) Processor cycles, etc.

(b) All the physical parameters are extracted implicitly from the node in order to avoid false publishing of these parameters.

(c) Available or demanded capability levels are assigned to each physical parameter according to its capability.

(d) A_Cap_Level: Physical parameter available to share with other node

(e) D_Cap_Level: Physical parameter demanded by the node

(f) Cooperating-Group Identifier (CG-ID) is assigned on the basis of node's physical parameters.

(g) Each cluster has nodes with different levels of physical parameters.

(h) Physical parameters reflect the heterogeneous nature of the network.

Capability level refers to the value of level assigned to a node on the basis of particular physical parameter, e.g. a node with free secondary storage of 40 GB (out of 110 GB of total space) is assigned max capability level: "5". As discussed earlier, that through node data structure we can extract capability level of the physical parameter of a node. On the basis of which Cooperating Group-ID is assigned to the node.

ACL is that capability level of a particular physical parameter which is available in the current node for sharing and DCL is the demanded capability level of the same current node which shows the required level of node for a particular physical parameter. If there are two nodes A and B then to compel them for cooperation, ACL of node A should be greater than DCL of node B (considering the ACL and DCL belong to same physical parameter), and ACL of node B is greater than DCL of node A, i.e.,

$$\text{ACL (A)} \geq \text{DCL (B)} \ \& \ \text{ACL (B)} \geq \text{DCL (A)}$$

If these two conditions are true for any two nodes then these nodes are assigned a particular Cooperating Group-ID (CG-ID). Otherwise node B will wait for another compatible node in a temporary array before the start of a next cycle and node A will check the same condition with another node, let say node C, in the next iteration. This process continues until a specified number of iterations are executed, which is equal to the cache size specified in the configuration file. After comparisons between the nodes in the cache, the comparisons between the ACL and DCL values of nodes are made in the temporary array. If the nodes form cooperating pairs, they become part of a particular CG, otherwise they are discarded from the network. CG also enhances efficiency and scalability by quickly locating a compatible node in a particular CG after some change in its ACL or DCL (as the values of ACL and DCL changes very frequently with the passage of time). As soon as the node finds another compatible node it changes its CG-ID accordingly as shown in Figure 3.2. The architecture of the overlay network is configured to enhance cooperation because if a node does not require a resource then it should not become part of a CG. The rate of change in CG-ID of a particular node can make it easier to estimate the percentage of cooperation in a particular node. If the percentage of cooperation for a node is very high it could be given some benefits "ß" in the network. These benefits could be related to the physical links of the node. The algorithm for assigning CG-ID is explained below.

3.4.2 Algorithm of HCGNP

Node node, node in temp

Node neighbor, neighbor in temp

If
A_Cap_Level of node ≥ D_Cap_Level of neighbor

AND

A_Cap_Level of neighbor ≥ D_Cap_Level of node

Assign.CG_ID to node & neighbor

else

Send node to temporary array (temp)

If
A_Cap_Level of node in temp ≥ D_Cap_Level of neighbor in temp

AND

A_Cap_Level of neighbor in temp ≥ D_Cap_Level of node in temp

Assign.CG_ID to node in temp & neighbor in temp

else

Discard node in temp and neighbor in temp

This algorithm develops cooperating-groups on the basis of conditions discussed above. At every iteration of each cycle, the number of nodes developed, are assigned with some data structures. HCGNP checks the cache of every node and compares the ACL and DCL values of each node with the neighboring node. Hence CG-ID is assigned to only those nodes which follow the give and take rule of HCGNP, either in the cache of the node or in the temporary location. The nodes are discarded from the network if they do not satisfy the condition for cooperation. In the next cycle, when the same discarded node joins the network again with some different ACL and DCL values, it can become part of a CG by satisfying the above mentioned conditions.

3.4.3 Algorithm (Data Flow Diagram)

The data flow diagram, shown in Figure 3.2 starts with the generation of node through HCGNP embedded in newscast protocol. Following steps show the complete explanation of this data flow diagram.

(a) Through methods like Create_Peer(), the nodes are generated in the network.

(b) Each node is considered as an object while implementing the protocol.

(c) As soon as the node joins the network, the data structure values like Node-ID, time stamp, available and demanded capability levels and cooperating-group ID.

(d) The time stamp is the Time-to-Live (TTL) value. TTL is the time slot allotted to each node at the time of its creation.

(e) The value of TTL decreases with the passage of time which determines the life of a particular node in the network.

(f) It should be kept in mind that the initial value of CG-ID is zero.

(g) The next step of the protocol is to assign cache to each node. This cache is a set of predefined array which holds all the data structure values of the nodes.

(h) The implementation of the protocol is such that the cache of each node holds the information of other neighboring nodes in the network.

(i) The size of the cache depends on the number of neighboring nodes and the network size. Hence cache size is configurable according to the network size through configuration file associated to each source code.

(j) The configuration file plays the role of a middle-man, through which the source code is assessable.

(k) At each iteration the ACL and DCL values, lying at the first index of each cache are compared with the values in the rest of the cache. As at first index place, the node holds its own ACL and DCL values.

(l) At each iteration it compares its ACL and DCL values with the ACL and DCL values of the neighboring nodes. This process continues till the condition for cooperation becomes true and CG-ID is assigned to the node.

(m) If the condition for cooperation becomes false, the neighboring node is sent to the temporary array, where the process is repeated.

(n) If in temporary array the condition for cooperation becomes false, the neighboring node is discarded.

(o) In the temporary array the nodes are compared with the same condition, they have been compared in the cache.

(p) The size of each cooperating group also depends upon the network size.

(q) The current implementation keeps the cache size and the CG size same, and the cache size is approximately one-tenth of the network size.

(r) After the comparison of ACL and DCL values between the nodes, the decision of joining a particular CG is based on the size of the group.

(s) In a larger network, large cooperating groups are formed and every node after joining the network checks the status of initially developed cooperating groups.

(t) If the initially developed cooperating-groups have the capacity, then the upcoming node will compare its ACL and DCL values with the nodes in that particular group.

(u) If there is no capacity in a group, the upcoming node will wait for new nodes to compare its resources and form CGs.

(v) The running of each iteration can be seen in the next section, where each iteration is based on the data structure values residing at single cell of a cache.

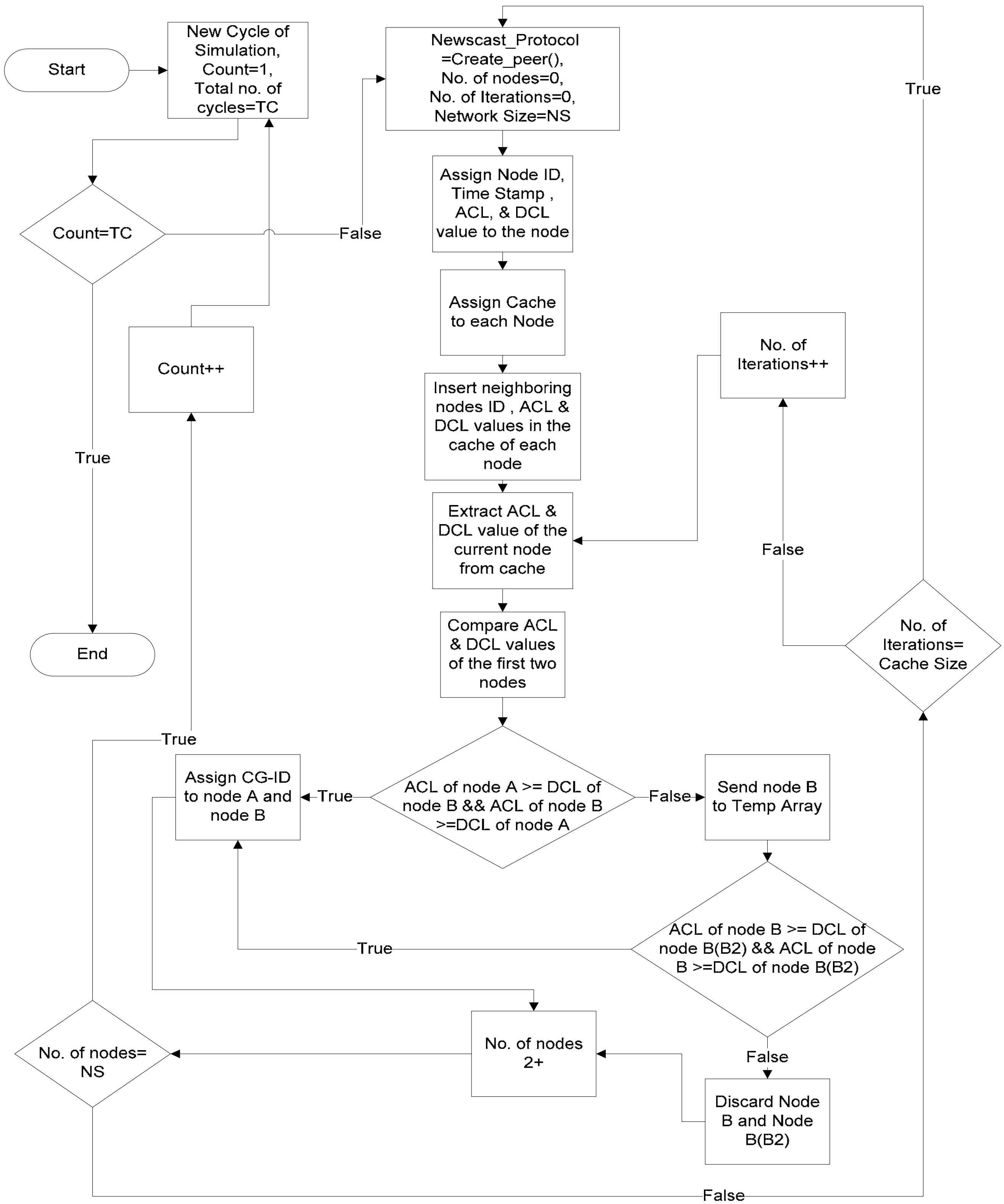


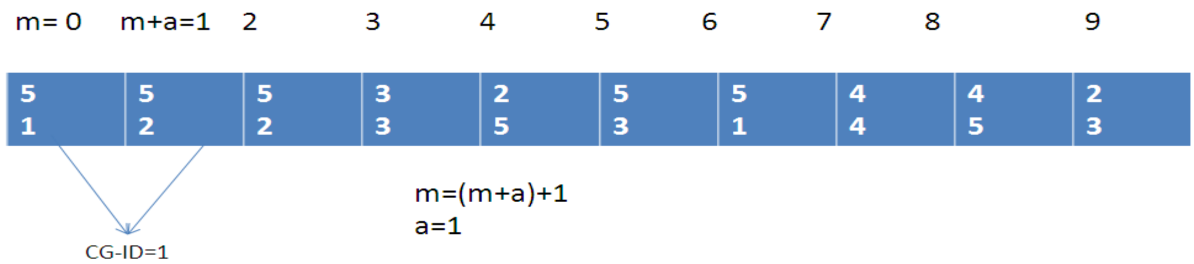
Figure 3.2: Data flow diagram based on algorithm to implement HCGNP

3.4.4 Algorithm (Dry Run):

(a) 1st iteration:

When m=0, and a =1
m+a =1, so values at index m and m+a are compared every time,

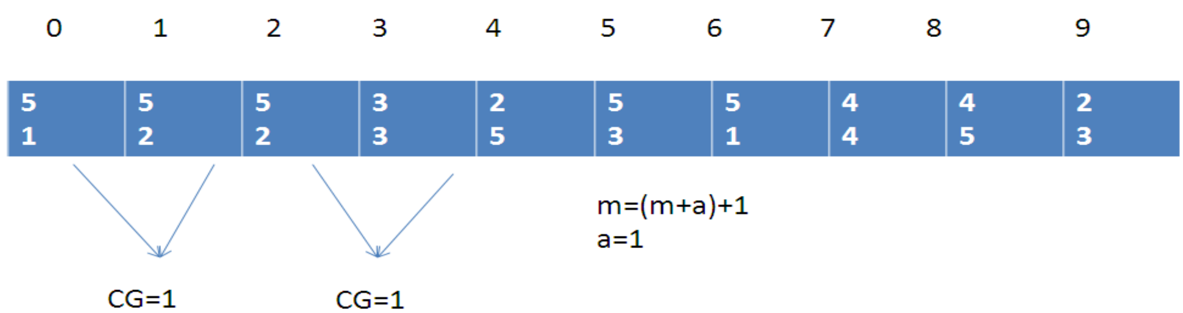
cache:



(b) 2nd Iteration:

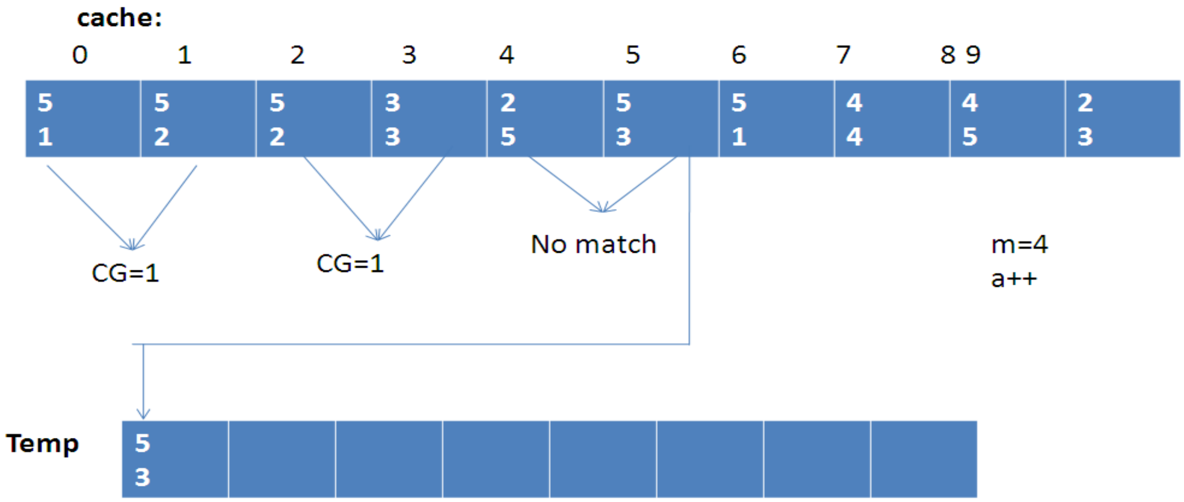
Now m=2, and a=1
m+a =2+1=3

cache:



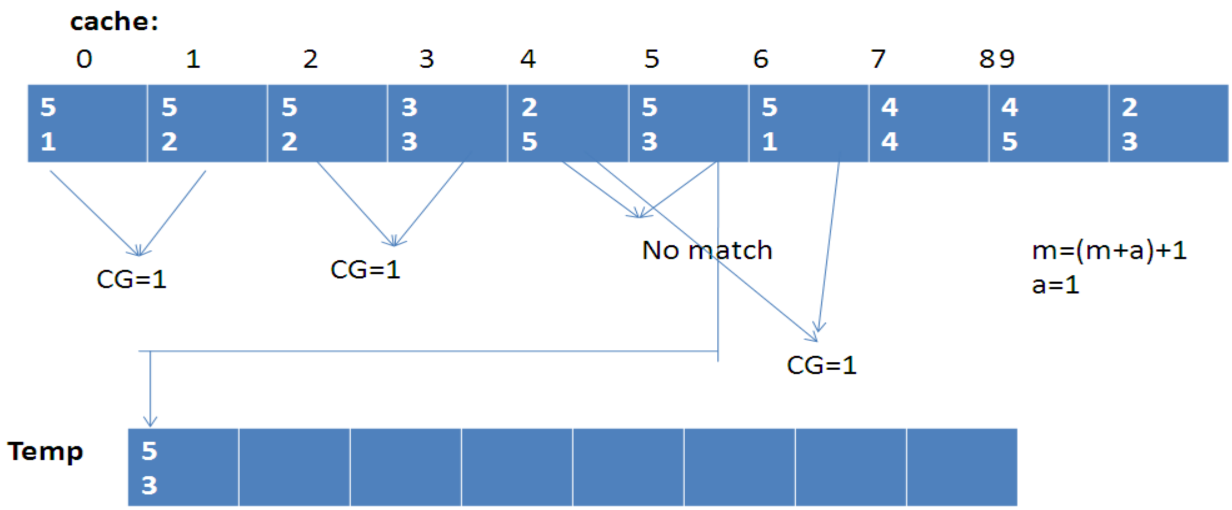
(c) 3rd Iteration:

Now,
m = 4, a =1
So m+a =5

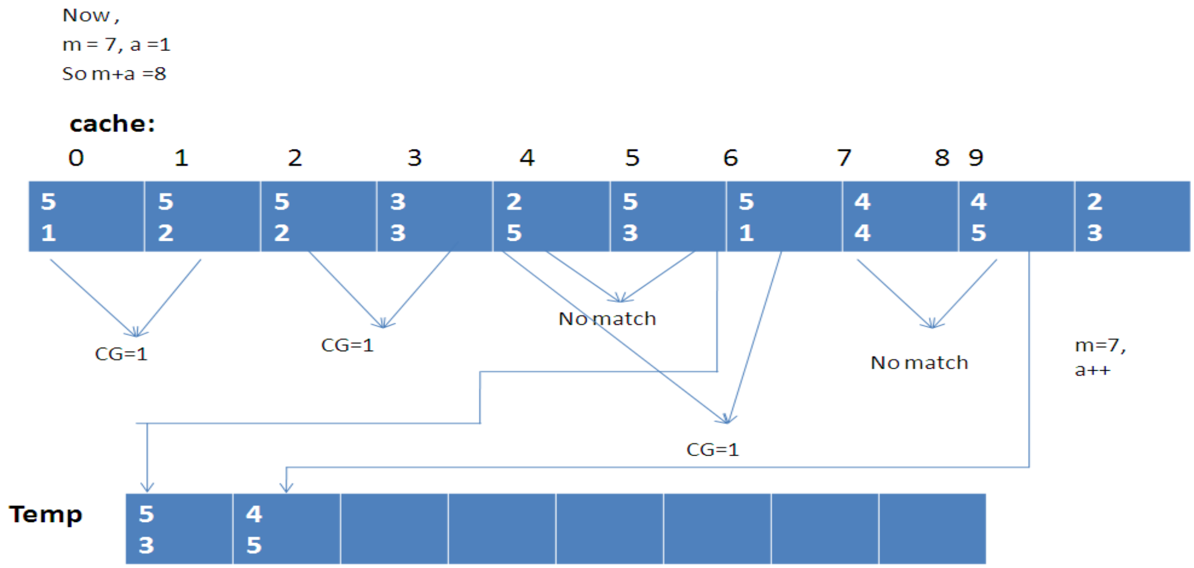


(d) 4th Iteration

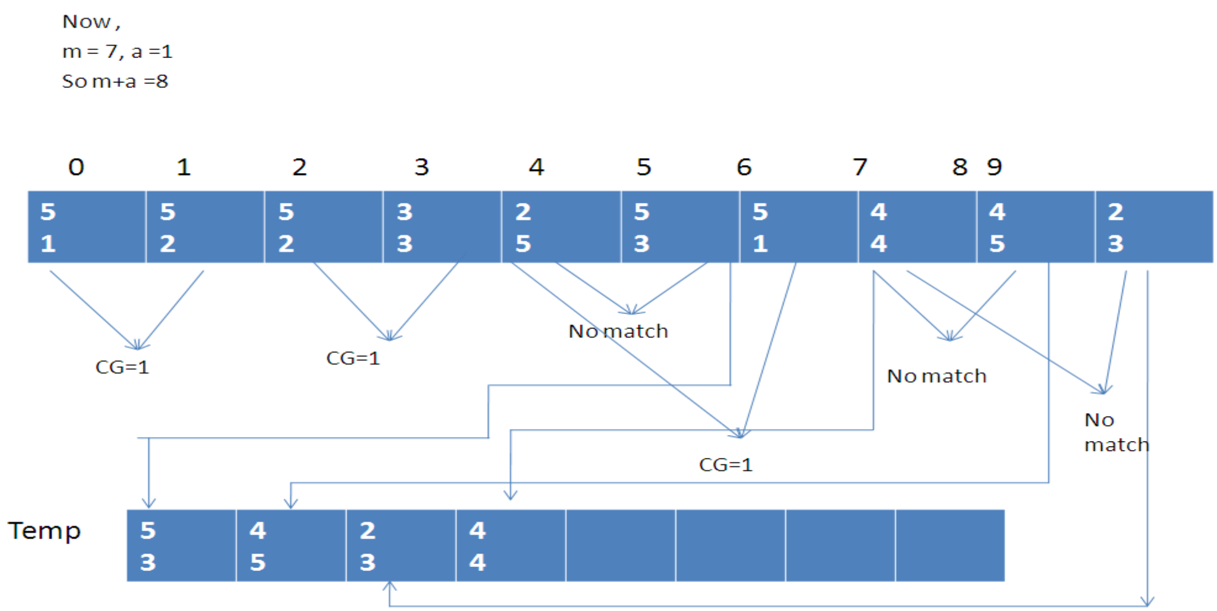
Now,
m = 4, a =2
So m+a =6



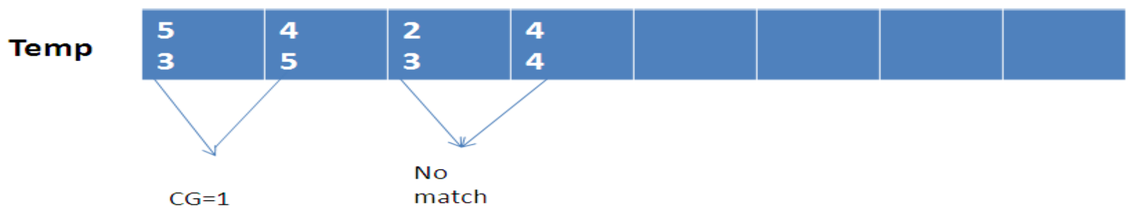
(e) 5th Iteration:



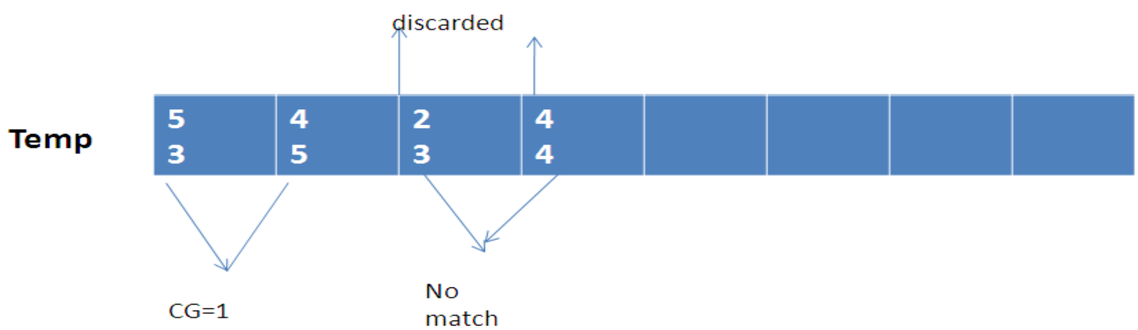
(f) 6th Iteration:



(g) 7th Iteration:



(h) 8th Iteration:



At every iteration the values of ‘m’ and ‘a’ are checked, as the index of the cache has to move accordingly. After the completion of all the iterations, the protocol executes the next cycle. If it finds the node with higher rate of CG-ID change, it increases the level of that node by assigning a benefit ‘ß’. This is how the rate of cooperation between the nodes gradually increases. As discussed earlier the benefited value ‘ß’ assigned to each node is according to the network capacity ‘x’. Hence the value of ‘ß’ is directly proportional to ‘x’.

3.4.5 Simulation Model

When HCGNP runs on top of gossip-based newscast protocol, the formation of nodes with different node identifiers can be seen in figures below, where every node joins the network with a particular value of available and demanded capability levels.

Figure 3.3 shows that node number 7 joins the network with ACL = 2 and DCL = 3, while node number 4 joins the network with ACL = 5 and DCL = 1. When the cross condition for cooperation is checked between node 7 and 4, it is found that both the nodes are compatible and can cooperate with each other. Hence a common cooperating-group ID is allotted to node 7 and node 4.

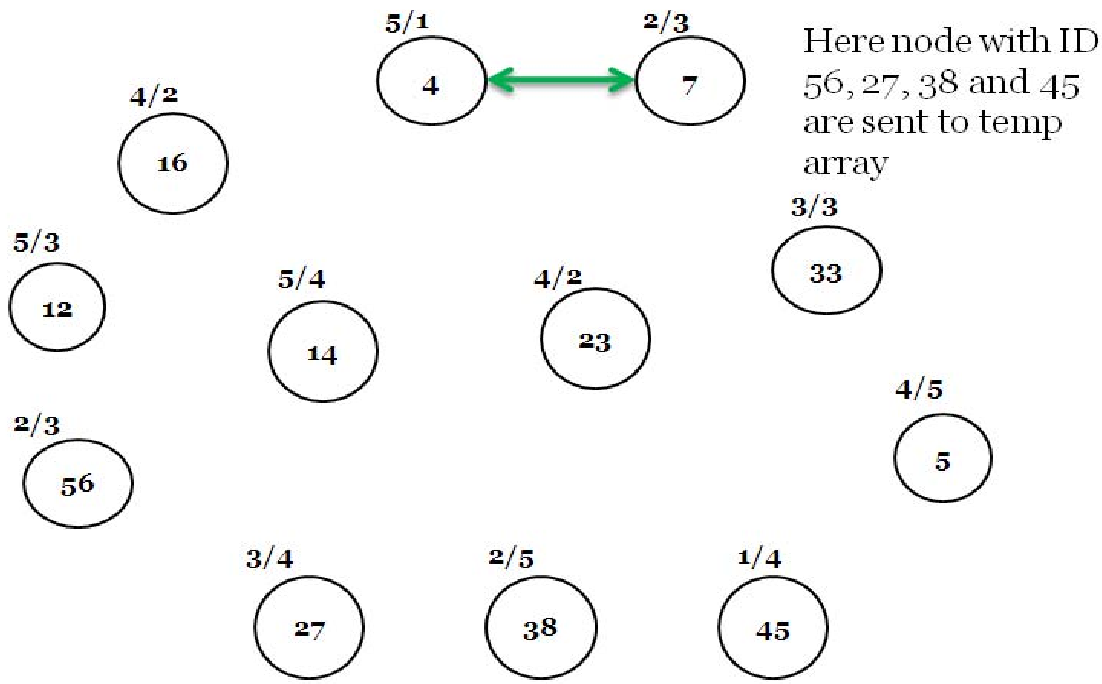


Figure 3.3 Node 4 & 7 making a cooperating pair

The same situation of cooperating pairs can be seen in Figure 3.4 and Figure 3.5. Here the nodes become part of a cooperating-group as they fairly satisfy the conditions for cooperation. Through this condition the sharing of node's physical resources is extremely fair, without wasting the network resources. The node identifiers shown in these figures are randomly generated through gossip-based protocol. These identifiers, ACL and DCL values and cooperating-group IDs are the data structure values assigned to each node whenever it is generated and becomes part of the network.

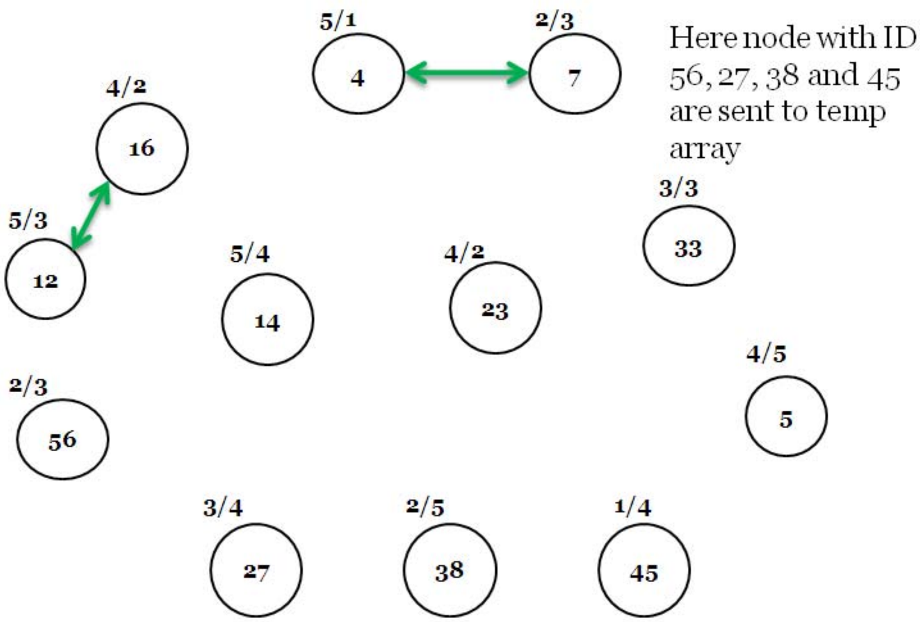


Figure 3.4 Node 12 & 16 making a cooperating pair

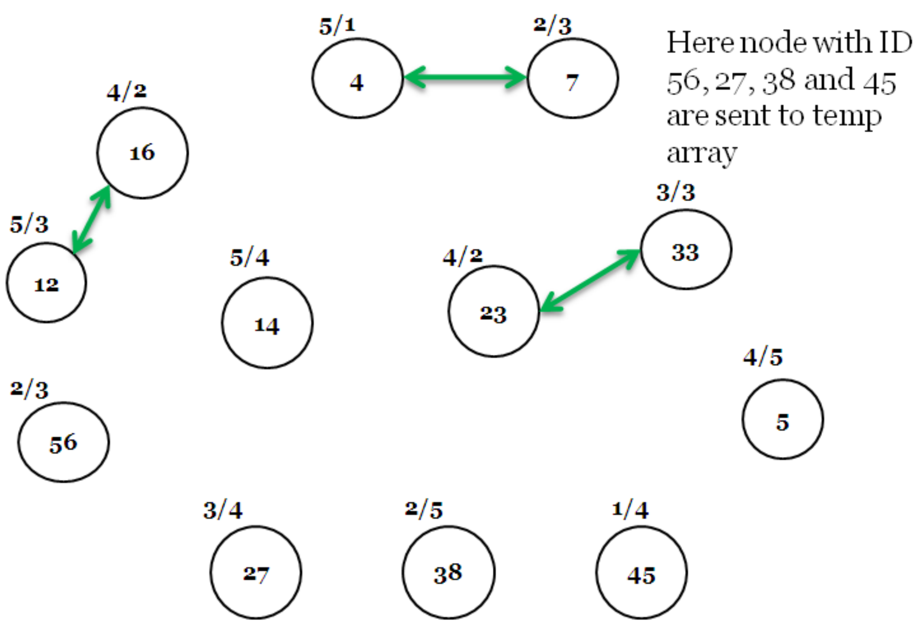


Figure 3.5 Node 23 & 33 making a cooperating pair

During the node generation process in the network the ACL and DCL values are also randomized. This randomization is for the achievement of heterogeneous P2P network. During the allotment phase of ACL and DCL values, when ACL value of one node does not fulfill the DCL value of its neighboring node, the cooperating pair is not developed. Nodes which do not form the cooperating pairs do not become part of cooperating-group. Figure 3.6 shows that node 56 and 14 do not become part of any cooperating-group as the ACL value of any of the two nodes is less than the DCL value of other neighboring node. Thus node 56 and 14 are sent to the temporary locations for further checks. This shows that only that node becomes part of cooperating-group that has a physical resource to share with other nodes in the network.

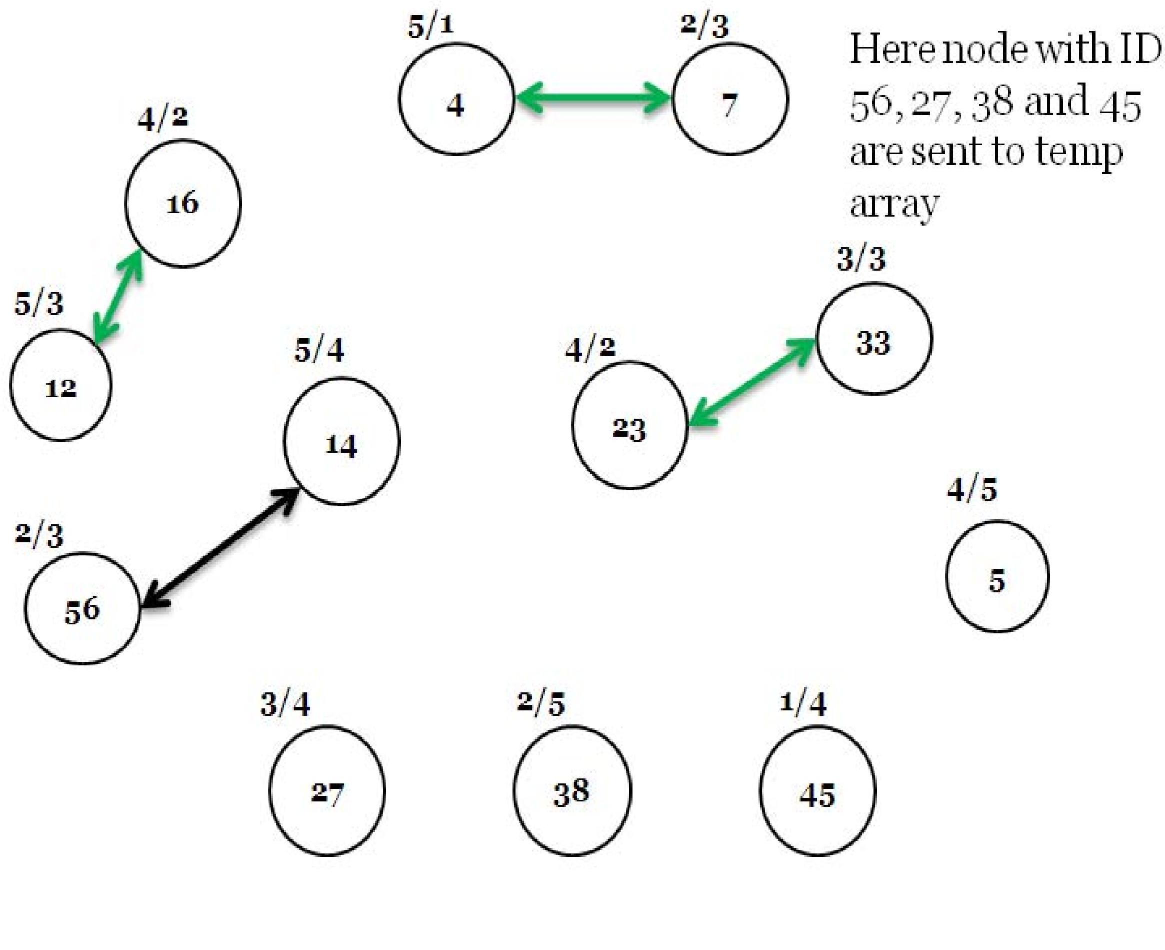


Figure 3.6 Node 56 & 14 showing no cooperation

Similarly during the comparison phase of node 14 with node 27, 38 and 45, as shown in Figures 3.7, 3.8 and 3.9, it can be seen that cooperating-pairs are not formed. This is because the ACL value of node 14 is less than the DCL values of node 27, 38 and 45. This condition is implemented for fair distribution of resources between the nodes. Nodes that possess low values of available capabilities cannot share their available resources. Hence the demanded resources of such nodes are not fulfilled. As discussed earlier, nodes which have low ACL values are sent to temporary locations. Every time when the ACL value is compared, the node with lower value of ACL is sent to temporary location, while the node with greater value of ACL remains in the cache list.

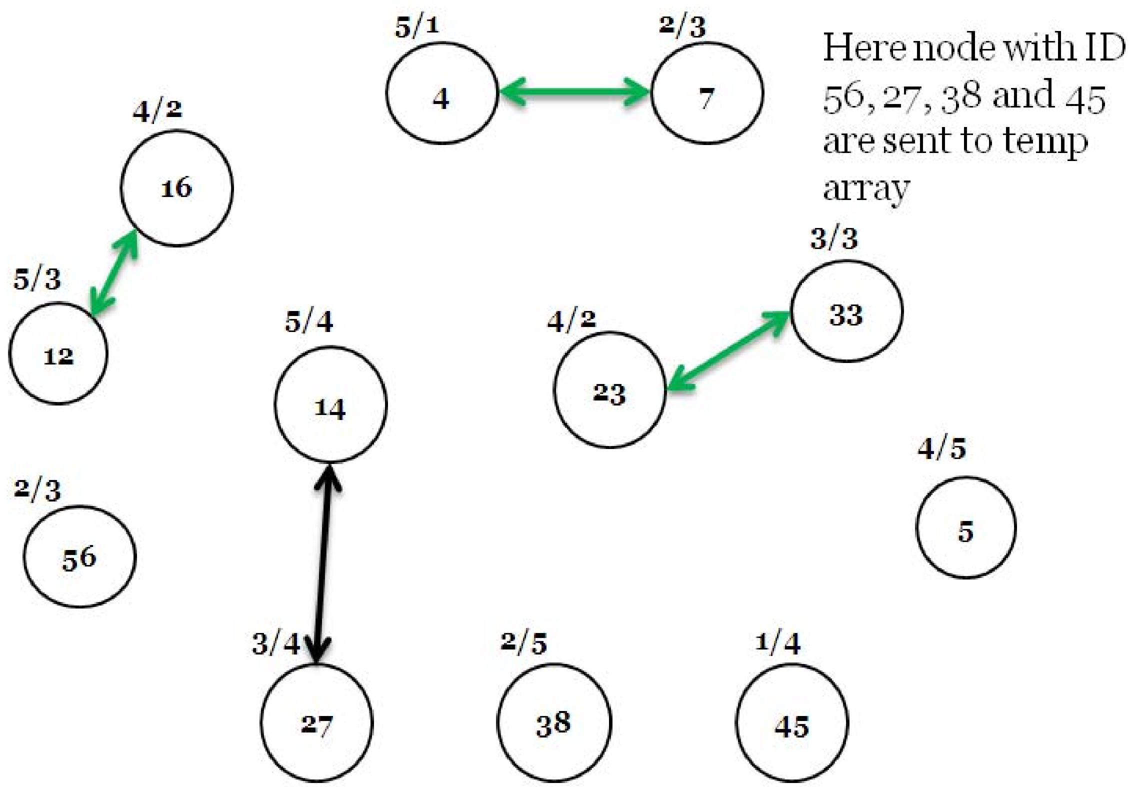


Figure 3.7 Node 27 & 14 showing no cooperation

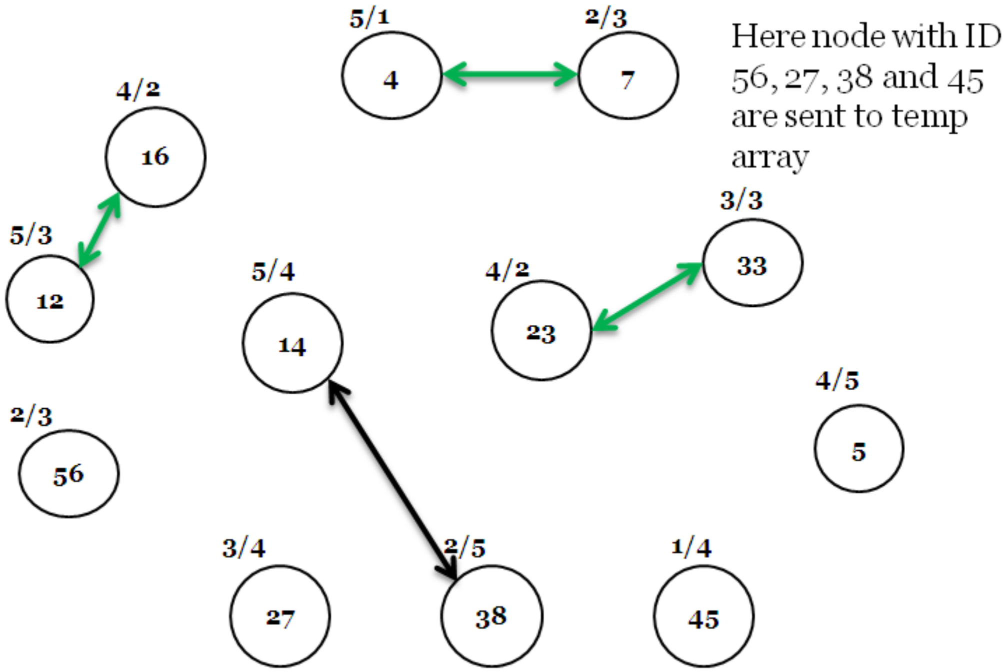


Figure 3.8 Node 38 & 14 showing no cooperation

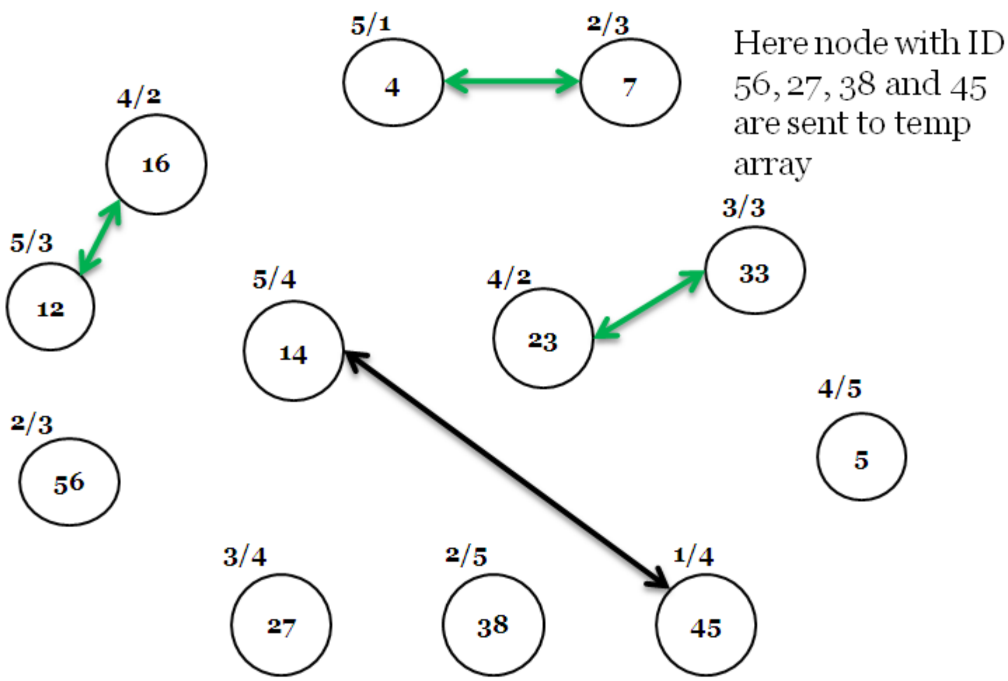


Figure 3.9 Node 45 & 14 showing no cooperation

The behavior of cooperation between the nodes is only achieved when both the nodes possess a specific capability level to fulfill the demand of another node in the network. The network model shown in Figure 3.10 shows that after comparing itself with node 56, 27, 38 and 45, node 14 finds its compatibility with node 5. Hence an adequate level of resources is required to build an environment of cooperation in the network by the assignment of cooperating-group ID to the node. Through these cooperating-groups the node search for a particular resource becomes quite efficient, as the number of nodes for a particular group remains constant. Whenever there is a vacancy for a cooperating pair in the group, the upcoming nodes become part of that particular group.

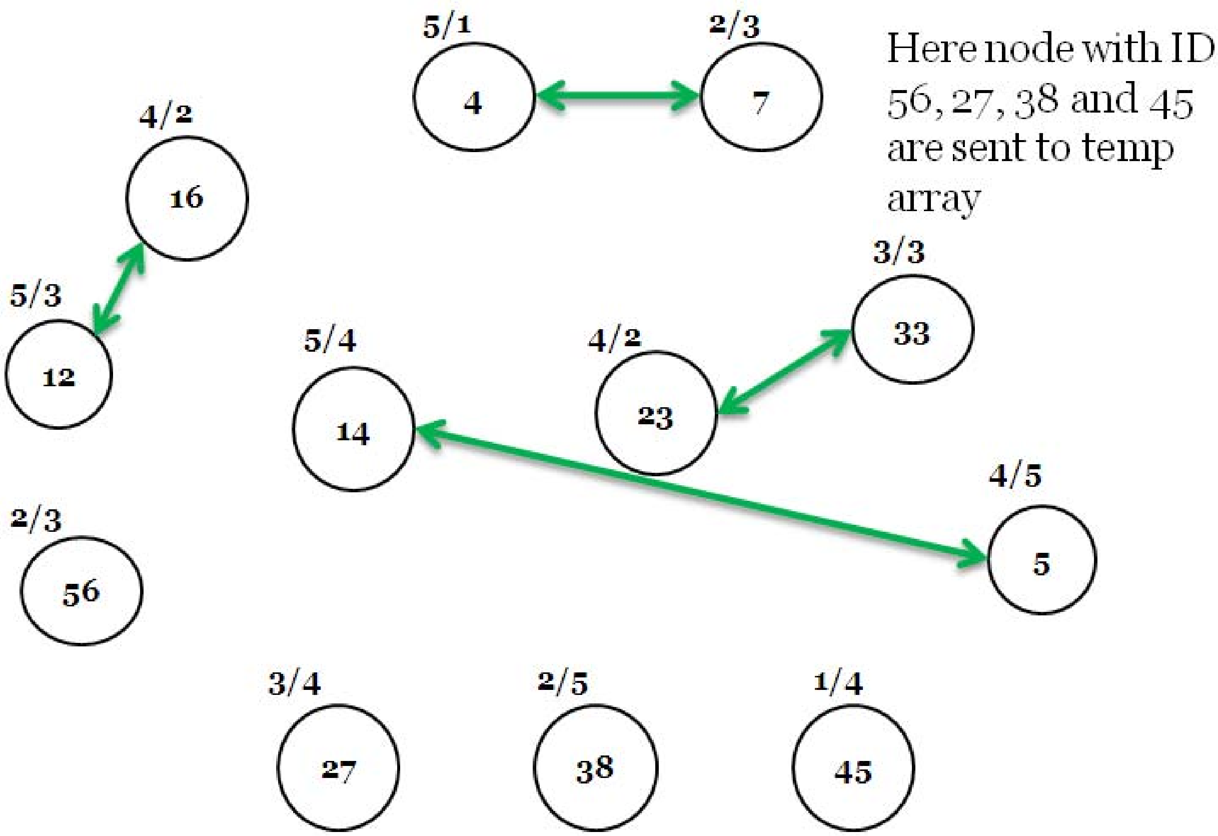


Figure 3.10 Node 5 & 14 making a cooperating pair

The implementation of HCGNP is such that during each cyclic phase the number of iterations is equal to the cache size associated to each node. At every iteration the comparison for ACL and DCL values between the nodes is carried out twice. First comparison is made in the cache and the second one is made in the temporary location. Figure 3.11 shows that when node 14 compares its ACL and DCL values with node 56, its condition for developing cooperating pairs becomes false and node 56 is sent to temporary location. In the next iteration, node 14 compares itself with the next neighboring node, i.e. node 27 (in the current model). The next time when the condition for cooperation becomes false, the next neighboring node is sent to the temporary location and so on.

Each cooperating-group forms pair of nodes which cooperate with each other by sharing their resources. In Figure 3.11 it can be seen that the neighboring nodes, such as, node 56, 27, 38 and 45 that have low values of capability levels go to the temporary location. The nodes in the temporary locations are again compared for the development of cooperating-pairs. Each pair can become part of any cooperating-group. These pairs can become part of an old cooperating-group or can be configured for a new cooperating-group by assigning a new identifier to the nodes. When the nodes in the temporary locations have not enough resources, i.e. with very low capability levels, then the nodes are marked to be discarded from the network. Figure 3.12 shows that the nodes that do not show cooperation are discarded from the network. Such nodes, when join the network again with different ACL and DCL values can become part of a particular cooperating-group.

The whole idea behind the implementation of HCGNP is to gain a certain level of cooperation by compelling the nodes for sharing their resources in the network. Without cooperation, it is useless for a node to become part of a P2P network. The basic idea behind P2P computing is the utilization of resources. Through HCGNP every node can become part of a P2P network by increasing any of its capability level before joining the network. Heterogeneous nodes with varying capabilities can share and gain benefits from the network without wasting network resources, which can be utilized for other important tasks.

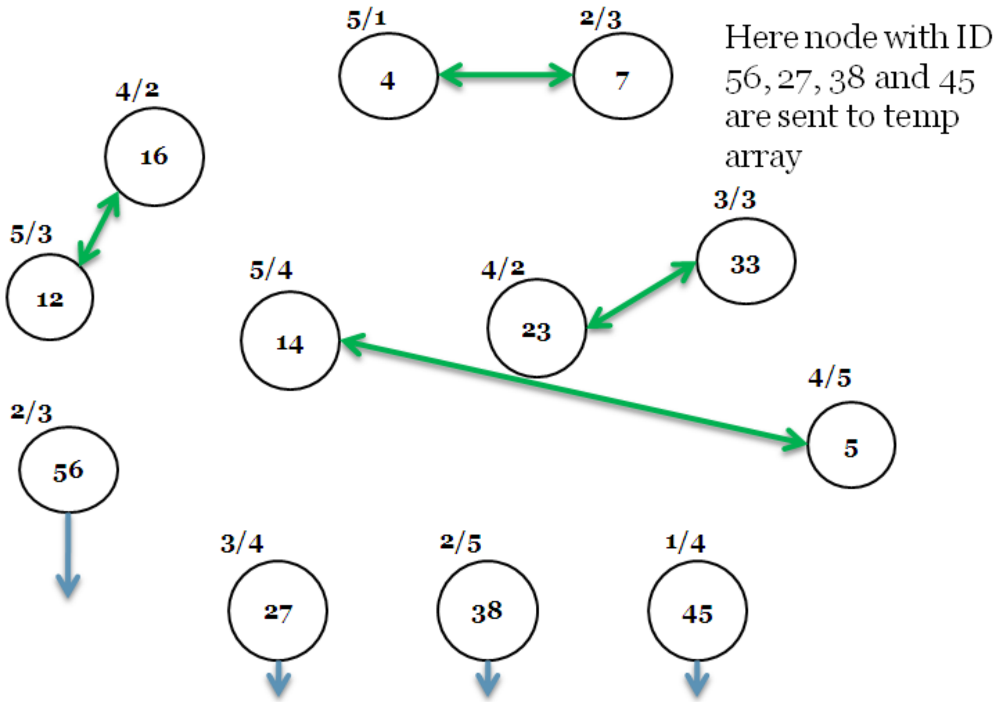


Figure 3.11 Node 56, 27, 38 & 45 are sent to temporary location

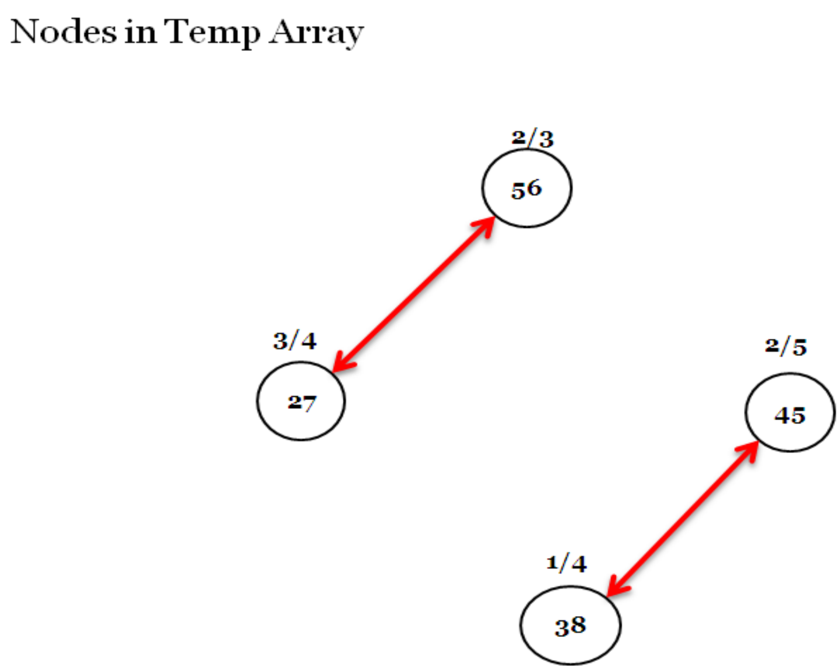


Figure 3.12 Node 56, 27, 38 and 45 are discarded on showing no cooperation

3.5 Test bed –PEERSIM SIMULATOR

P2P networks are comprised of millions of nodes and to handle the simulations for such a huge network, a scalable simulator is required. PeerSim is comprised of different pluggable building blocks implemented in Java. Through these building blocks it is easier to prototype a protocol. These blocks can be easily replaced by other blocks, by implementing the same interface. There are two simulation models supported by PeerSim. One is cycle-based model and the other is event-based model. The current research is based on evaluation of PeerSim with cyclic-based modeling of different protocols. In PeerSim, the protocol once implemented, can be controlled by the relevant configuration file. This file is a plain ASCII text file comprised of key-value pairs representing java.util.properties. Such file is very helpful in supplying a running network and the configuration parameters to study different behaviors of nodes.

Initially PeerSim was developed as a tool for the researchers at University of Bologna and Trento Italy, later it was released under LGPL open source license to make it available for other research projects.

Several other simulators used in P2P environment like Oversim, used only event-based modeling for designing any architecture which is applicable for both structured and unstructured P2P networks. PeerSim provides an independent handling of protocols in a pure unstructured environment. Similarly simulator D-P2PSim is purely used for a distributed environment in P2P networks [3], implements a hierarchical based architecture.

The surveys conducted on peer-to-peer simulators e.g. in [32] and [34] enlist different peer to peer simulators e.g. NS-2 , PlanetSim, NAM, OMNet+, OverSim, GPS, Agent and P2PSim[33]. All these are used in different scenarios in peer-to-peer overlay networks but none of them explains or thoroughly implements the non hierarchical and decentralized architectures with high scalability and efficiency. All the simulations and experiments done previously in PeerSim as in [35], [36] and [37], are very specifically designed and do not give an idea about the implementation of cluster based heterogeneous P2P network protocols. So the current research gives an overview of PeerSim usage for such environment.

3.6 Programming Techniques

Running the simulations on PEERSIM platform requires a Java platform to support the backend implementation of the protocol. The version of java that is required to compile and extract the physical parameters of any node is jdk1.6.0. The compatibility of PEERSIM with java is such that it requires simulating the class file, which has been created after compiling the java file, with in the same path directory. Programming techniques required to implement the protocol include arrays, loops and conditions. The most important factor is to implement a java code which is compatible with the configuration file, in order to avoid conflicts and to gain error free results. The sample configuration file is shown in the figure below (Figure 3.13). In this file we can see that single protocol has been used. If another protocol needs to be implemented on top of the Newscast Protocol (basic protocol of PEERSIM to generate nodes), then it has to be mentioned in the configuration file, class file and the JAR files (discussed below). Moreover, in the configuration file the size of the network, cache length and degree (view) of the node's cache is mentioned. As the number of protocols embedded in the configuration file increases, the time to process them also increases. Through configuration file, the protocols can be tested with different values of network sizes, in order to see the enhancement in the required value of the parameter desired.

```
# PEERSIM EXAMPLE 1
random.seed 1234567890
simulation.cycles 4
simulation.shuffle

overlay.size 1000
overlay.maxsize 2000

protocol.0 example.SimpleNewscast.SimpleNewscast
protocol.0.cache 40

init.0 peersim.dynamics.WireRegularRandom
init.0.protocol 0
init.0.degree 20
```

Figure 3.13: Configuration file to simulate java class file in PEERSIM

The steps to simulate java class and configuration file are discussed below:

(a) **How to compile:**

- Compile java file from the command window.
- Place the class file (x) (created after compiling java file) in the main folder having following executables:
  - Peersim folder (built-in class files to design topology)
  - META-INF (provides a run time environment for peersim)
  - Create a jar file of class file, peersim and META-INF through following command:
  - jar cf peersim.jar x peersim META-INF

(b) PEERSIM simulator is invoked by running the following components:

- peersim.jar
- jep-2.24.jar (a Java API for parsing and evaluating mathematical expressions)
- Peersim.Simulator (invokes simulator class)
- Configuration file (maintaining the network size, cache size, degree size and no of cycles)
- Command to run peersim simulator is:
  - java –cp "peersim.jar;jep-2.24.jar" peersim. Simulator config/config1.txt

If the implementation of protocols in the configuration file is layered, i.e. if at the top of node formation protocol different protocols like security protocol, cooperation protocol, etc. are implemented then the computational complexity of the architecture will increase. Figure 3.14 shows the implementation of HCGNP in layered architecture.

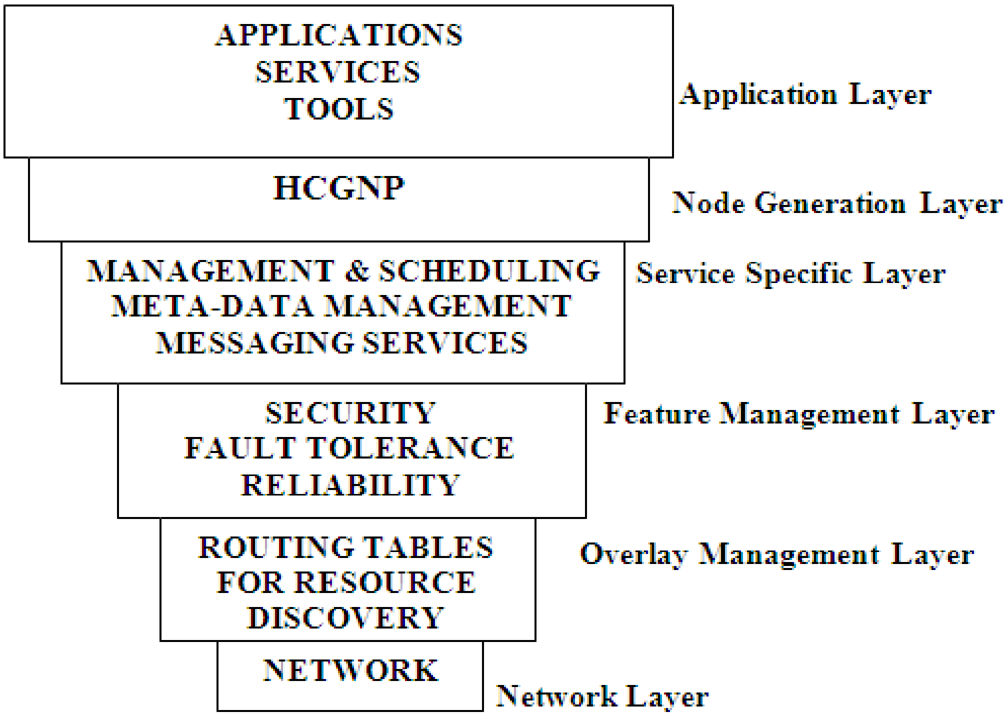


Figure 3.14: Layered architecture showing implementation area of HCGNP

3.7 Summary

This chapter discusses the implementation procedures of the protocols in PEERSIM simulator. The implementation of Heterogeneous Cooperating Group-based Newscast Protocol (HCGNP) is based on the formation of non-hierarchical groups in P2P overlay networks. This formation of groups is based on clusters that were developed, in the previous research, to share physical resources but the nodes in those clusters lacked the behavior of cooperation. The implementation of HCGNP required Java (jdk1.6.0) at the backend. PEERSIM supports Java APIs to simulate different protocols. The protocol is first implemented in Java, and then requires a PEERSIM configuration file to simulate it further through the Simulator class files. If the implementation of protocols in the configuration file is layered, i.e. if at the top of node formation protocol, different protocols like security protocol, cooperation protocol, etc. are implemented, then the computational complexity of the architecture will be increased. This shows that handling the configuration file in the context of computational complexity and time as well as the compatibility of the configuration file with the java source code should be handled according to the simulation environment.

Chapter 4: Simulations and Results

4.1 Introduction

This chapter is based on the output and results generated after simulation of HCGNP. The results are based on the development of cooperating groups in P2P Overlay Networks. To enhance cooperation, some comparisons between different algorithms have been carried out. These comparisons are based on the tradeoff values of different parameters like network capacity, bandwidth utilization etc. The basic criterion to simulate a protocol in PEERSIM is to invoke a mechanism which ensures the compatibility of the java source file and the configuration file. In PEERSIM the basic protocol to generate node is Newscast Protocol.

4.2 Protocol Simulations

The simulations of the protocol is based on the configuration file which defines the P2P overlay networks size, the protocols used to generate the nodes and the number of cycles for which the protocol is to be simulated. The formation of nodes in the network is such that at each cycle new nodes join the network and the older nodes leave the network. This happens by updating the cache list associated to each node. As mentioned earlier, HCGNP has been embedded in the Newscast Protocol which implicitly incorporates the cooperating behavior in the nodes when the nodes are joining the network. This cooperating behavior is based on the nodal resources and the physical parameters associated to the nodes.

The simulations are also based on the benefit value "ß" assigned to the nodes according to the network capacity. The enhanced cooperating behavior is observed after the implementation of HCGNP. The cooperation level of nodes is compared with PD-Protocol that has implemented SLACER on the top of Newscast protocol. The level of cooperation has also been observed in the simulations when there is no benefit "ß" given to the node. Moreover the overall performance of HCGNP over SLACER and SNA has also been observed. The simulation graphs have been plotted in MATLAB after generating the desired results in PEERSIM, using JAVA at the backend. The following sections will elaborate the findings of these observations.

4.2.1 Parameters Measured

The implementation of HCGNP is based on achieving a level of cooperation among the nodes and to improve the nodal search in order to retrieve information about the relevant resources. The parameters on the basis of which HCGNP has been evaluated are discussed below:

(a) Cooperation

To achieve a level of cooperation in the nodes there should be a criterion to judge the node's behavior, i.e. it can cooperate or not. Every single node participating in P2P overlay networks possesses some resource. This resource could be a file, a service, free storage to share with other node, RAM, processor cycles etc. As there is no centralized authority in P2P overlay networks, the task of incorporating cooperation among the peers depends upon the capability parameters assigned to the nodes according to its available and demanded resources. The rate of cooperation in the network is elevated if and only if two nodes cooperate with each other simultaneously. This conditional communication between the nodes in the network compels them to cooperate, if they have to remain part of the network. Thus the percentage and probability of cooperation in the nodes communicating with each other is found to be maximum. The cooperating behavior of the nodes while implementing HCGNP and SLACER has been discussed in detail in the later sections.

(b) Node Search

The importance of nodal search in order to acquire the information of the desired resources is one of the most important aspects that have been studied by several researchers. In HCGNP node search procedure has also been observed as the protocol is also required to be efficient in locating the nodes carrying desired resources.

(c) Computational Complexity

Another important feature which plays a vital role in evaluating protocols is the computational complexity associated to a particular protocol. The current research implements a protocol that has been embedded in Newscast protocol to generate cooperating nodes. Protocol like PD-Protocol that has implemented SLACER on the top of Newscast Protocol proves to be less efficient which has been discussed in section below. The computational complexity associated to any implementation plays a vital role as it proves whether the achieved results are worth to be implemented in a real time environment.

4.2.2 Output Generated

The simulation results can be seen in the output file generated by configuring several parameters of the configuration file. These output values can be seen in Table 5.1 and are elaborated below:

(a) It can be seen that the available capability level (ACL) and demanded capability level (DCL) of each node (A) is compared with the ACL and DCL value of the neighboring node (B).

(b) Cooperating Group-ID (CG-ID) is assigned to A and B, if:

   (i) the ACL of (A) ≥ to the DCL of (B) and

   (ii) the ACL of (B) ≥ to the DCL of (A)

(c) If any of the above two conditions are untrue, the neighboring node (B) is sent to the temporary array and becomes node (B1, B2, B3….BN).

(d) Its ACL value is sent to ATEMP array and DCL value is sent to DTEMP array.

(e) Node (A) will be compared with next neighboring node (B) in the same cache.

(f) After the completion of the comparisons from the cache, the comparisons are going to take place in the temporary array.

(g) CG-ID is assigned to the nodes B1 and B2 in temporary array, if:

   (i) the ACL of (B1) ≥ to the DCL of (B2) and

   (ii) the ACL of (B2) ≥ to the DCL of (B1)

(h) If any of the above two conditions are untrue, node (B1) and node (B2) are discarded from the network.

(i) The discarded nodes can become part of a CG in the next cycle with different values of ACL and DCL.

ACL(A)	DCL(B)	ACL(B)	DCL(A)	ACL(A) >=DCL (B)	ACL(B) >=DCL (A)	ATEMP []- =ACL(B)	DTEM P[] =DCL(B)	ATEMP [ACL(B 1) &ACL(B2)]	DTEM P[DCL(B1) &DCL(B2)]	ACL(B 1)>= DCL(B 2)	ACL(B 2)>=D CL(B1)	DISCARD	ASSIG N CG- ID
5	2	5	1	TRUE	TRUE	–	–			–	–	–	1
5	3	3	2	TRUE	TRUE	–	–			–	–	–	1
2	3	5	5	FALSE	TRUE	5	3			–	–	–	–
2	1	5	5	TRUE	TRUE	–	–			–	–	–	1
4	5	4	4	FALSE	TRUE	4	5			–	–	–	–
4	3	2	4	TRUE	FALSE	2	3			–	–	–	–
4	5	4	4	FALSE	TRUE	–	–	–	–	–	–	–	–
3	5	4	1	FALSE	TRUE	4	5						
								5 & 4	3 & 5	(5>=5)TRU E	(4>=3)TRUE	–	1
								2 & 4	3 & 5	(2>=5)FALS E	(4>=3) TRUE	1	–

Table 4.1: The output gathered from the simulations

The number of cooperating-groups (CGs) in the network shows the cooperation rate of the nodes in a P2P overlay environment where each node possesses different physical resources. On the basis of these resources the capability level is assigned to that node. The formation of CGs not only shows the rate of cooperation in the network but also improves the nodal search in order to acquire the required resource. The protocols developed earlier only enhanced cooperation among the nodes without considering the network capacity and network heterogeneity, whereas HCGNP has been implemented for the network in a heterogeneous environment. The level of cooperation achieved while implementing HCGNP is according to the network capacity. The benefit 'ß' is rewarded to the nodes which show cooperating behavior. However the value of 'ß' should not overload the network.

4.3 Results and Comparisons

Formation of CGs is one of the most important tasks, which is achieved after creating an environment of cooperation in the network. The results are accumulated by making changes in the configuration file. The following values have been changed in the configuration file to observe enhancement in cooperation between the nodes in P2P overlay network:

- Number of cycles to simulate HCGNP.
- Network size
- Cache length associated to each node.
- Degree size on the basis of which the cache would be refreshed in the next cycle of simulation.
- Benefit value (ß) given to the cooperating node at every cycle according to the network capacity (x).

The observations of the simulations are observed in the graphs below. The number of CGs at any time instance (t) with different network sizes can be seen in Figure 4.1, where the increase in the number of CGs according to the network size can be observed. We see that as the size of the network increases the number of CGs also increases. When network size is just 1000 nodes, the number of CGs formed is minimum, i.e. not more than 25 groups. This network size of 1000 nodes is the minimal value required to generate acceptable number of CGs in the network. Network size taken below 1000 nodes does not fulfill the objective behind development of groups in the network. When the network size grows to 2000 nodes, the number of groups developed is 40. When the network size is doubled again, the number of groups developed is 68. It can be seen that the number of CGs is almost doubled when the network size is increased exponentially. As the network size reaches to 16000 nodes, the number of CGs approaches to 200. This shows that cooperating-groups are worth to be implemented with larger network sizes rather than smaller ones.

Here, the number of CGs, taken at a specific time instance (t), can be expressed mathematically by following relationship:

Number of CGs (t) = Network size/ cache length (associated to each node)

The nodes that join the networks become part of a CG according to its available and demanded capability levels. At every cycle the new nodes join the network and the older nodes leave the network, hence the nodes which do not become part of CG in one (current) cycle can become part of a CG in the next cycle with different values of capability levels.

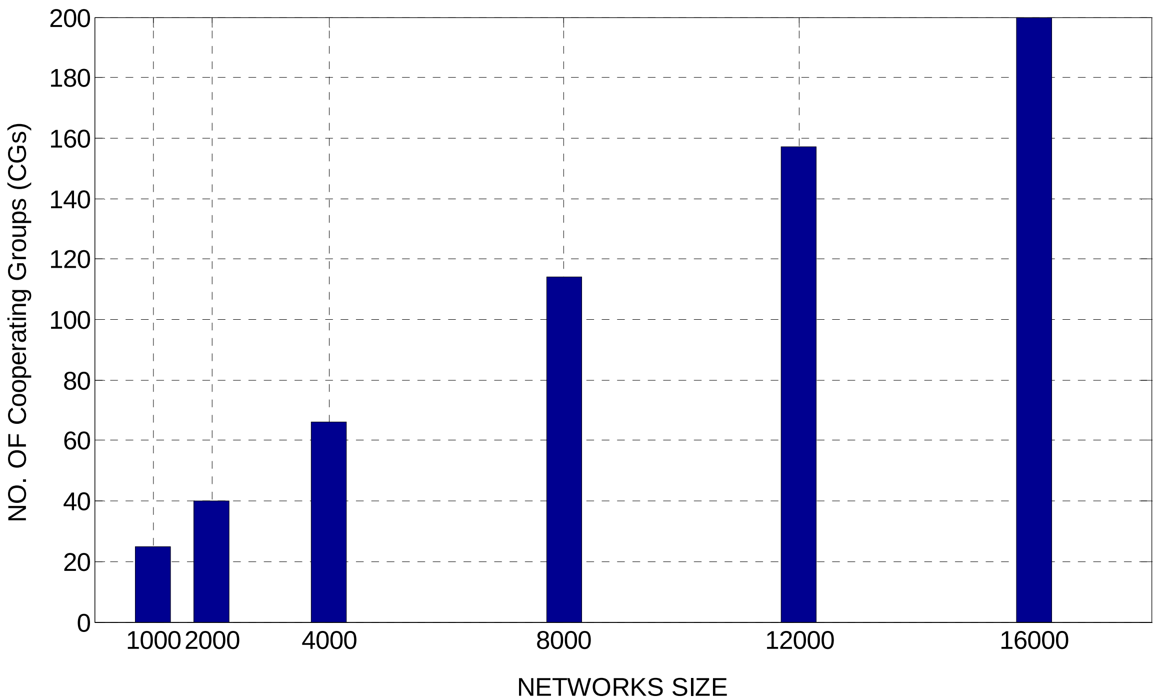


Figure 4.1: Number of CGs at any instant of time (t) with different network sizes

In Figure 4.2 it can be seen that each CG carries number of nodes with different levels of cooperating nodes (CN) and non-cooperating nodes (NCN). Here we can see that:

(a) The percentage of CNs in CG-ID=8 is approximately 85% and the percentage of NCNs in the same CG is approximately 15%.

(b) Similarly, the percentage of CNs in CG-ID=46 is approximately 35% and the percentage of NCNs in the same CG is approximately 65%.

(c) Thus, it can be seen that as the percentage of CNs in a CG increases, the percentage of NCNs decreases and vice versa.

(d) In CGs with IDs 9, 32, 44, 46 and 64, it can be seen that more than 50% of nodes are NCNs of the group.

(e) These nodes are termed as NCNs because they have not developed a cooperating pair with any node in the group.

(f) Whenever new nodes join the network, first they compare their ACL and DCL values with these NCNs.

(g) If NCNs are not able to develop a cooperating pair in the entire cycle, they are discarded from the network.

(h) This result is taken when no benefit is given to the nodes in the network which shows that the average percentage of CNs is greater than the average percentage of NCNs even when there is no benefit or payoff value rewarded to the nodes.

(i) This shows that an acceptable condition for cooperation is achieved without overloading the network.

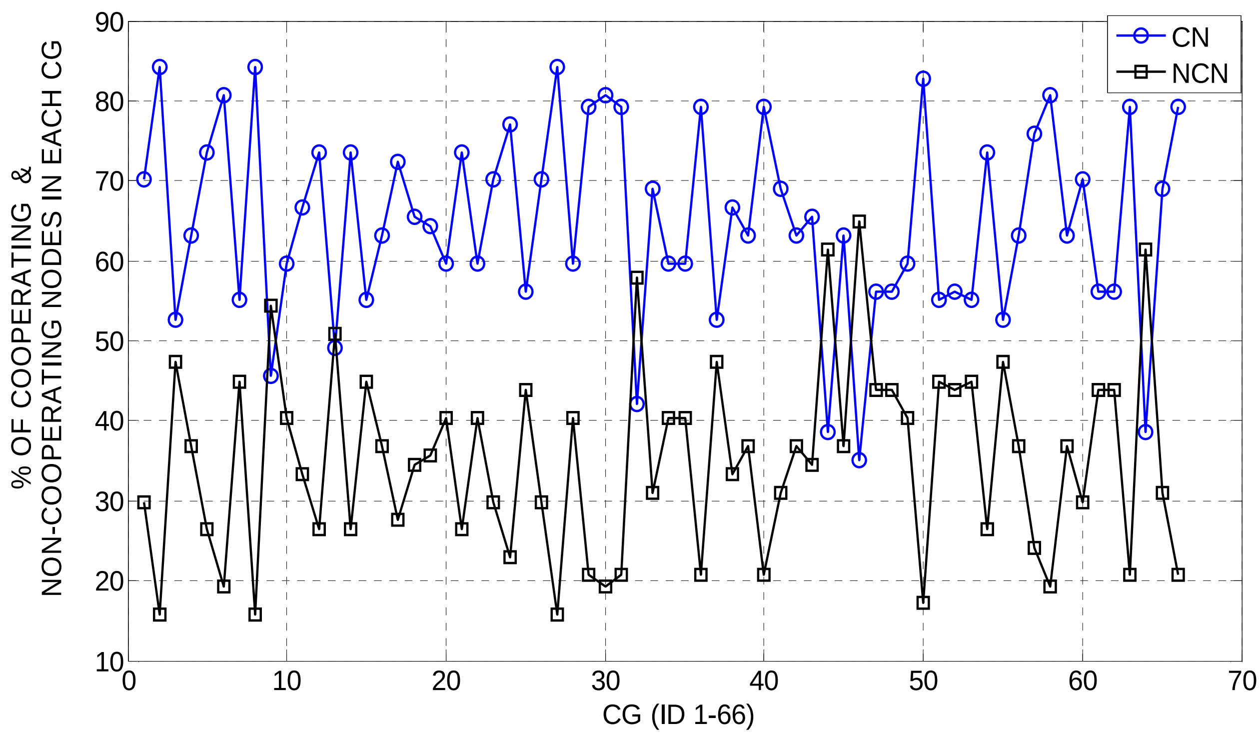


Figure 4.2: Percentage of cooperating nodes and non-cooperating nodes in each CG with network size = 4000 nodes

The number of CNs in later iterations gradually increases as the new nodes find their compatible cooperating (old) nodes. These old nodes when change their capability levels, find their compatible new nodes in later iterations. The gradual increase in the CNs in the same cycle with multiple iterations can be seen in Figure 4.3. Here, it is observed that:

(a) During initial iterations, i.e. from iteration 1 to iteration 10, the maximum percentage of CNs in a CG is approximately 84%, but the stability in trend of cooperation is very low.

(b) Percentage of cooperation is dropped to 38% in the 7th iteration of a cycle. This is because during the initial iterations of a cycle, the network size is in the process of growing to its maximum limit. Hence the nodes, that join the network in initial iterations, have to wait for their compatible nodes till later iterations. Therefore the stability in cooperation does not stay for a longer period in initial iterations.

(c) In the later iterations, i.e. from iteration 11 to iteration 30, a small increase in stability of cooperation can be observed. In 29th iteration the drop level of cooperation is not less than 48%, which shows that with the passage of time, the trend in cooperation is increased. This is because of the reason that in later iterations, the size of the network grows and the NCNs find their compatible nodes with desired values of ACL and DCL.

(d) During 50th iteration the cooperating trend drops down to almost 36%, which is very low, but the gain in cooperation till 55th iteration reaches 84% again. It also shows that during these 5 iterations, i.e. between iteration 50 to 55, the percentage of CNs constantly increases.

(e) The nodes that have been discarded from the CGs in previous iterations also try to become part of a CG in the next iterations, before the start of the next cycle by implicitly changing their available and demanded capability levels. This increases the overall percentage of cooperation within a cooperating group iteratively in each cycle.

The overall network state in terms of cooperation and non-cooperation among the nodes can be observed in Figure 4.4. Without investing any network resources in rewarding benefit, it can be seen that the percentage of CNs as compared to NCNs is always greater. Irrespective of network size, percentage of optimized cooperation achieved is higher than the percentage of non-cooperation. Although the number of non-cooperating nodes in the overlay network is not minimum, it is worth retrieving such cooperating behavior with no impact on network capacity. Network capacity being the vital attribute required during the communication phase of the nodes in the network, should not be wasted in rewarding as an “extra benefit value” to the CNs. But if network capacity allows such implementations, it may be the best option.

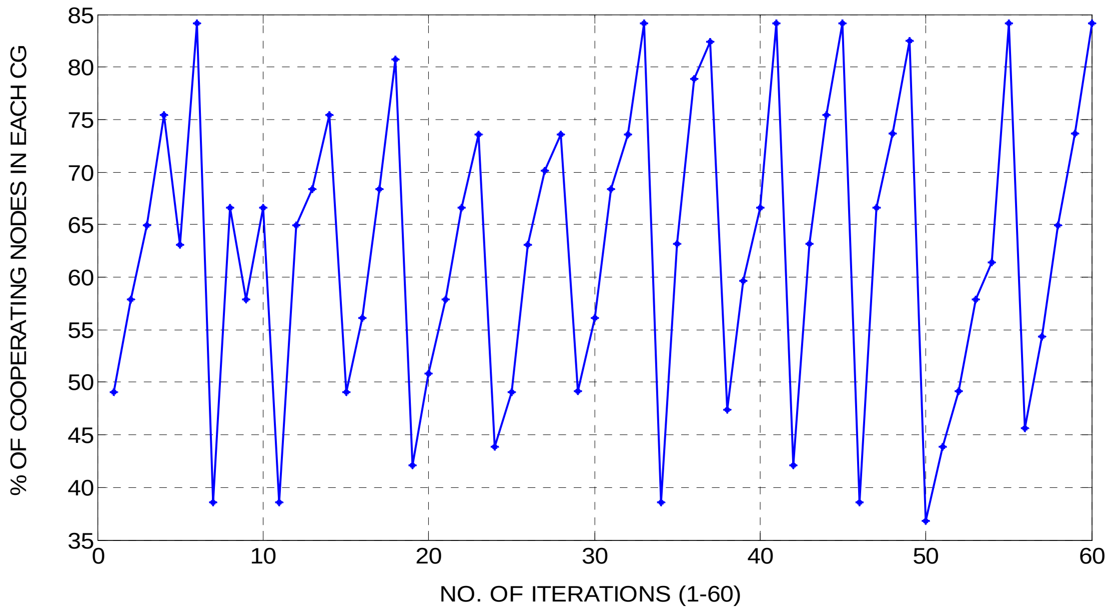


Figure 4.3: Iterative increase in percentage of cooperating nodes in each cycle

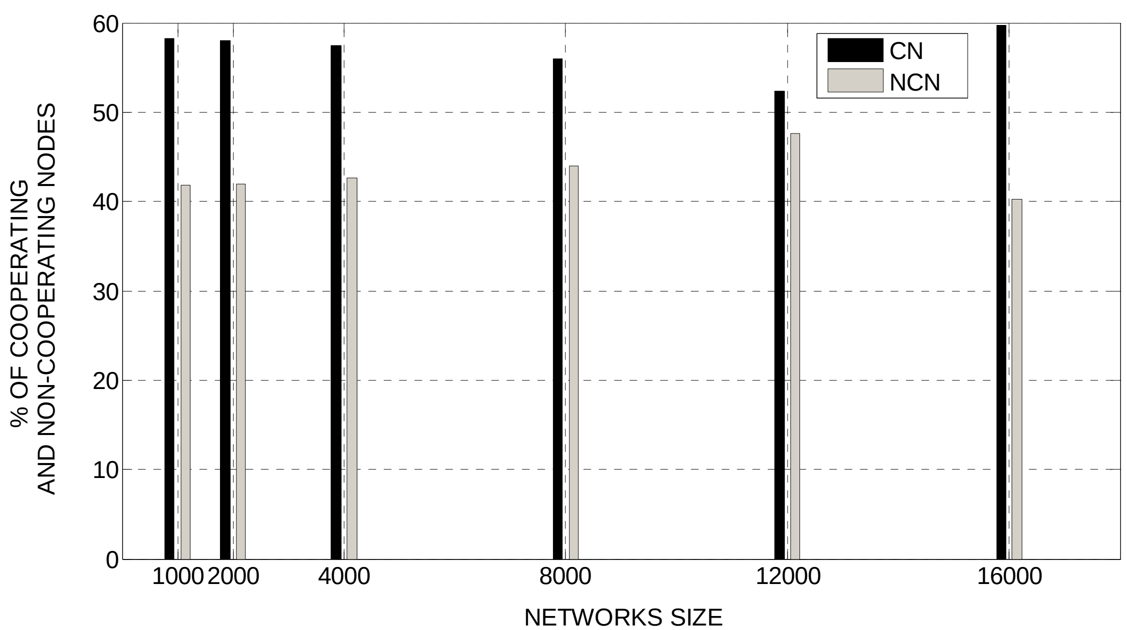


Figure 4.4: Percentage of cooperating and non-cooperating nodes in each network

After the implementation of SLACER and HCGNP (ß) with the provision of rewarded benefit value at each cycle to the CNs, it can be seen in Figure 4.5 that HCGNP performs far better than SLACER. This improved behavior is attributed to the factor that maximum percentage of cooperation in HCGNP (ß) is achieved within 10 cycles of simulations. While in SLACER the maximum percentage of cooperation is achieved after 100 cycles of simulations. It is observed that during first four cycles, the percentage of CNs in SLACER is 9%, while the percentage of CNs in HCGNP (ß) is 30%, for network size (NS) of 1000 nodes. This trend in cooperation is doubled within next two cycles in HCGNP, i.e. 60% cooperation is achieved, while SLACER increased its level of cooperation to 11% at that stage. Hence SLACER reached at maximum level of cooperation in 100 cycles. The observation adds to the fact that the computational complexity while implementing SLACER is far greater than implementing HCGNP, as to achieve the targeted value of cooperation wastes several cycles of simulations in SLACER. But another very important fact is the wastage of network resources in rewarding extra benefit to the CNs in SLACER as well as in HCGNP. It demands a thoughtful approach to find the tradeoffs between the two protocols.

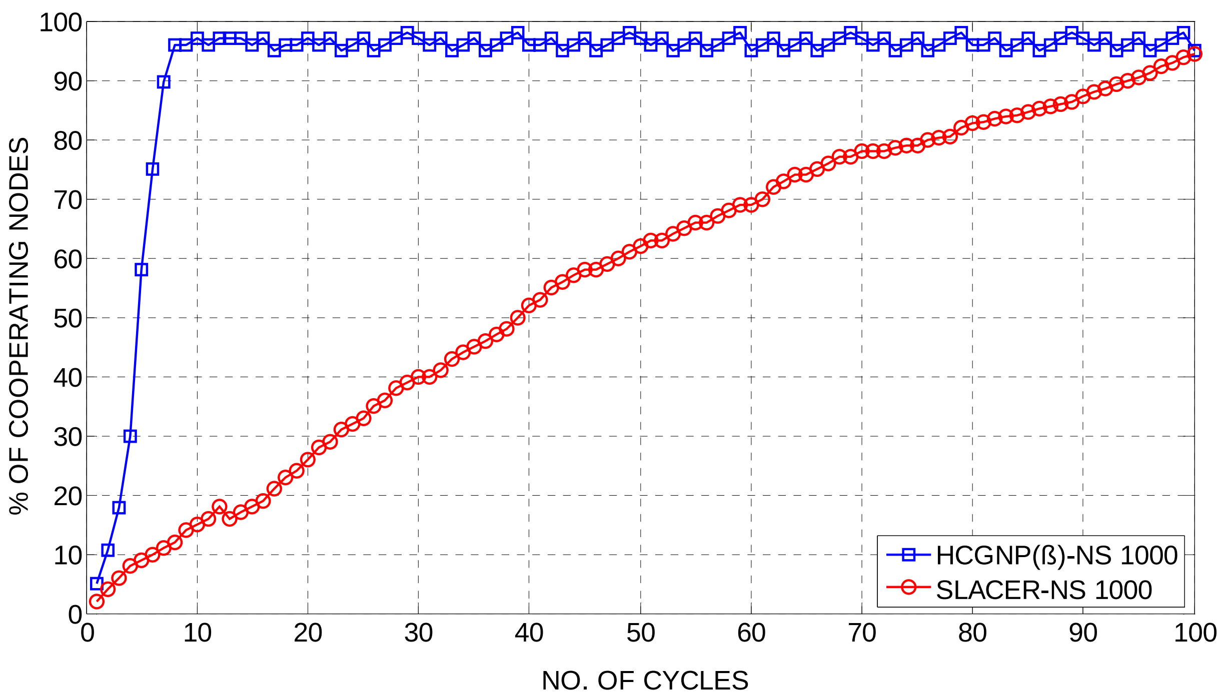


Figure 4.5: Percentage of cooperation/cycle in HCGNP (ß) and SLACER

This difference in the level of cooperation between SLACER and HCGNP is due to the simplified nature of HCGNP. The node is forced to cooperate as soon as it joins the network through HCGNP. While in SLACER, two protocols run one after the other, i.e. at the start newscast protocol is configured to generate nodes with Node ID and TTL value. After the formation of whole network, PD-protocol is implemented on each node. In SLACER the delay in the gain of cooperation is because of bi-level protocol handling. The rule of cooperation implemented in HCGNP is embedded in newscast protocol. Therefore the gain in cooperation is very rapid as the node is configured using single protocol. At the time of node generation, the available and demanded capability levels are also assigned to the nodes along with the Node-ID and TTL value. Hence the protocol handling of the nodes is very efficient and resourceful.

Now if we assume that there is no extra benefit given to a node in HCGNP then according to Figure 4.6 we can see that 50% of the nodes in the entire network are cooperating. This percentage of cooperation is achieved within the first ten cycles of simulation. Although the percentage of cooperating nodes is not more than 50%, the behavior of the nodes is such that they are compelled to cooperate after joining the network. This may be viewed as absolute cooperating behavior of the nodes in the network. This kind of behavior of nodes elevates the purpose of P2P networks, i.e. to utilize maximum resources of the network without increasing its workload. Hence, there is a requirement to decide the rewarding value of the benefit (ß) to the nodes according to the network capacity.

Besides comparing HCGNP with SLACER, another technique that is studied to evaluate the current research is the implementation of Social Network Architecture (SNA). As discussed in earlier chapters, this technique also paid incentives to the nodes after they have shown cooperating behavior in the network. As compared to SLACER this technique is far more adoptable. The incentives are given to the nodes after they have shown cooperating behavior, thus assuring the cooperation. But this technique does not discuss the effects of giving incentives on the network load. It is fair to say that this technique is better than HCGNP only when it provides incentives to the nodes. Figure 4.7 shows that the level of cooperation in social network architecture lies between HCGNP (ß) and HCGNP (without awarding benefit). After 10 cycles, the percentage of cooperation among the nodes with HCGNP (without awarding benefit) is 50%, the percentage of cooperation among the nodes with SNA is 70% and the gain in cooperation with HCGNP (ß) is 94%. The maximum percentage of cooperation gained by SNA is 82% after 30 cycles of simulation. When SNA is compared with HCGNP (without awarding benefit), it is observed that the networks implementing SNA have larger percentage of cooperating nodes. While comparing SNA with HCGNP (ß), it can be seen that SNA does not yield more than 82% of cooperating nodes whereas HCGNP (ß) shows 94% cooperation. Here it must be noted that HCGNP (ß) can provide maximum benefit with minimum utilization of processor cycles. As the adaptability of a particular protocol in any environment is much dependant on its efficiency and computational complexity, HCGNP (ß) is far better than SNA because it saves the processing power of the nodes.

On the other hand, HCGNP (without rewarding benefit) provides incentives or benefits to the nodes on the basis of network capacity, taken as a variable value "x". While implementing a particular protocol in any network, the most important aspect which should always be considered is the capacity and capability of that network. In a P2P environment the idea of bandwidth sharing depends upon the number of nodes in the network. The affordable network capacity permits different protocols to reward benefits to the nodes from the network. HCGNP configures nodes to cooperate in the network by giving incentives to the nodes according to these network capacities.

If we consider that the network capacity is a variable "x" we can see different levels of percentage of cooperating nodes according to network capacities. Figure 4.8 shows different levels of cooperation among the nodes by rewarding a benefit (ß) to the node in the network, on the basis of showing cooperation with other nodes in the network. The procedure of giving reward to a cooperating node in HCGNP justifies the utilization of benefit, contrary to SLACER where payoff values have to be assigned to the nodes which do not cooperate. This creates an unjustifiable situation in the network. Similarly in SNA the benefit is given to only those nodes which show cooperation, which is again a waste of network resources. Therefore in HCGNP the value of "x" is linearly increased. When network capacity is taken as 'x' the percentage of cooperation is 50%. When it is taken as '2x' the percentage if cooperation is increased to 68%. And finally when the network capacity is increased to 5x, the gain in cooperation is maximum (i.e. 94%). Figure 4.8 also shows that the highest level of cooperation achieved for each value of network capacity takes not more than ten cycles to achieve cooperation. This also decreases the computational complexity and the overall percentage of cooperation remains approximately constant after achieving its highest value.

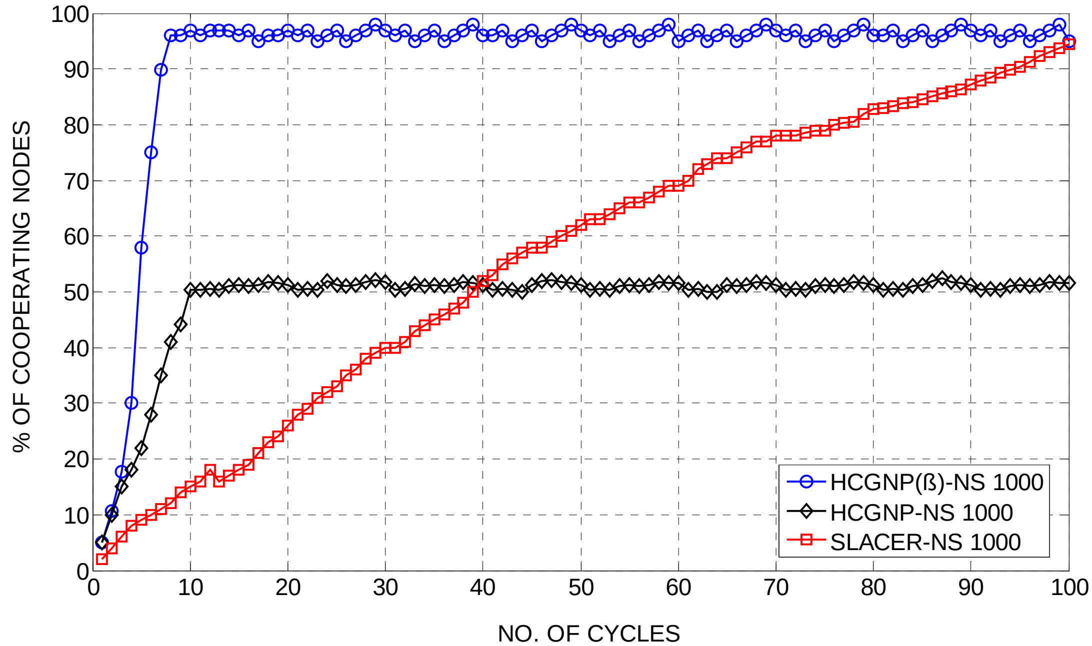


Figure 4.6: Percentage of cooperation/cycle in HCGNP (ß), HCGNP (without benefit value) and SLACER

In real-time environment, the status of network capacity needs to be maintained at the application layer level. Prior to the joining process of the nodes in the network, the status of the network should be determined. This can be achieved by summing-up the bandwidth allocated to each node that has previously joined the network. The current state of HCGNP does not maintain the status of network capacity. It is not necessary that the benefit value should be an extra bandwidth provided to the node. The node can also be provided with the CG-ID even with the false condition of cooperation, if it has shown cooperation frequently. The decision on categorizing the type of benefit depends upon the type of network and the type of resources shared among the nodes.

The ACL and DCL values are the resources that nodes share among each other. For deciding the benefit type, the determination of types of resources is another important task that should be taken care of. HCGNP can handle multiple capability levels showing multiple resources in the network. It is quite easier to handle resource types by assigning a tag value to each resource type and giving benefits accordingly.

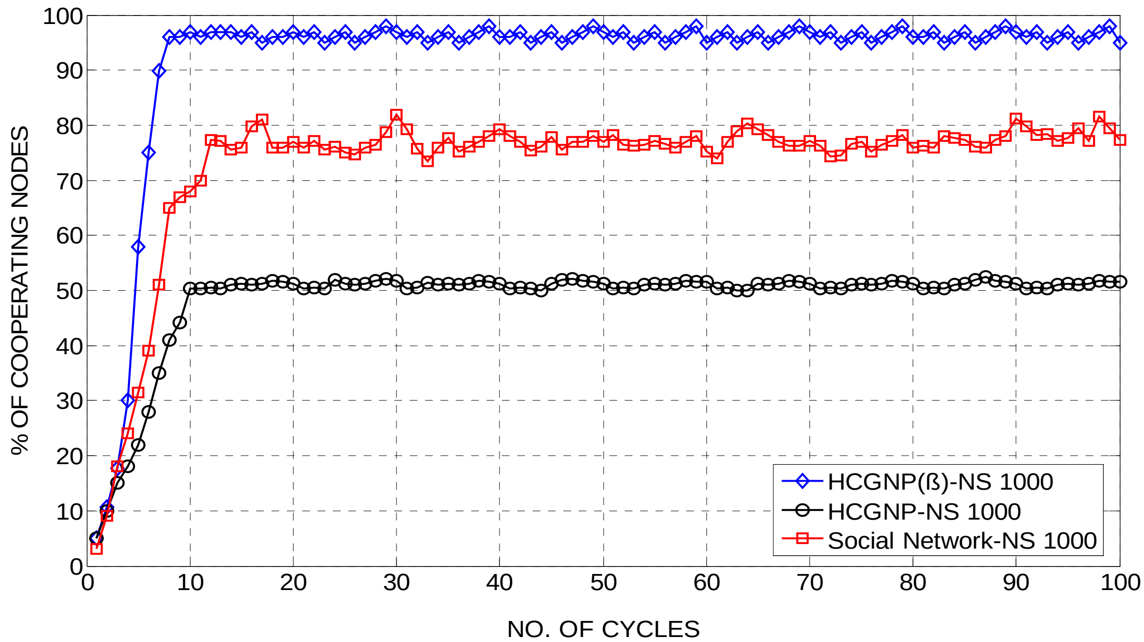


Figure 4.7: Percentage of cooperation/cycle in HCGNP (ß), HCGNP (without benefit value) and SNA

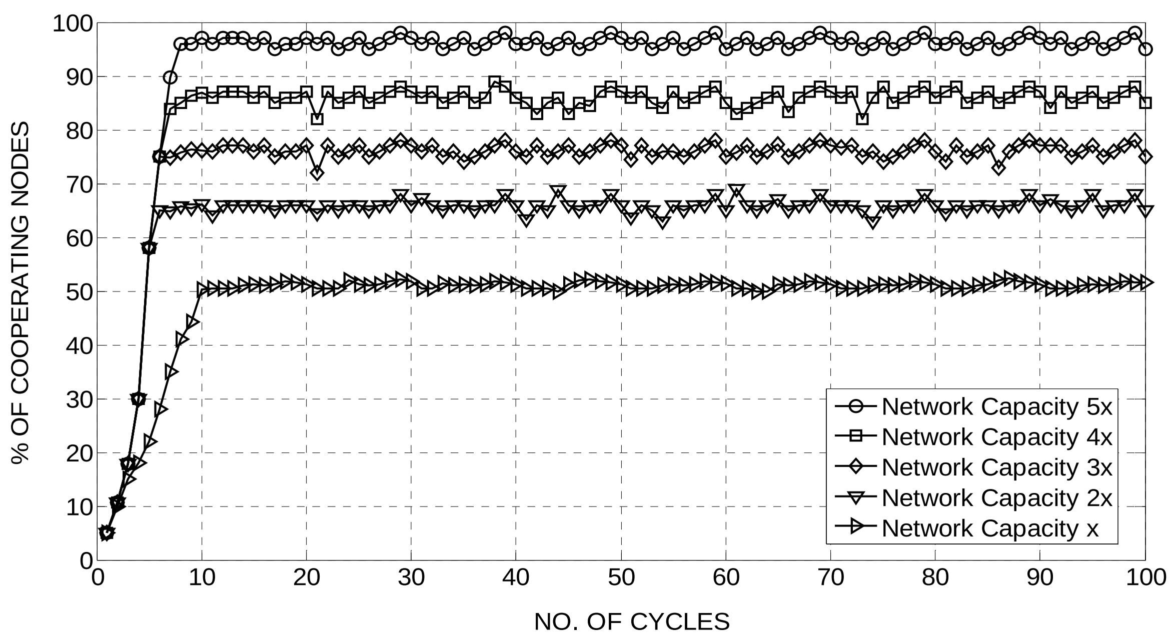


Figure 4.8: Percentage of cooperation/cycle in HCGNP with different values of network capacity (x)

Running HCGNP on top of different clustering algorithms for content-based sharing can also be very useful. By running cooperating algorithm on top of clustering algorithm, we can implement not only clusters for content-based sharing of resources but may also force the nodes to cooperate in the network. Besides cooperation, the nodes are grouped together on the basis of physical resources. The accuracy rate of clustering algorithms like selfless and selfish algorithm increases as the physical capability of the node to share and download a particular file or resources is predetermined.

Figure 4.9 shows different percentages of accuracy rates, each related to a different algorithm. The maximum accuracy rate of selfless clustering algorithm is 15%, but when HCGNP is implemented on top of it, the maximum accuracy rate increases to 24%. This is because HCGNP determines the node's capacity before that node searches for any resource. Hence it can be predetermined, whether the node is capable enough to exchange the resources. The accuracy rate rapidly increases when the selfish clustering algorithm is implemented on the nodes, i.e. 50% of the nodes managed to collaborate with other nodes having desired resources. This algorithm performs even better in the presence of HCGNP, as the accuracy rate reaches 92%. Hence running such an algorithm which deals with the physical parameters of the nodes is always favorable for the network as desired routing algorithms can be implemented easily.

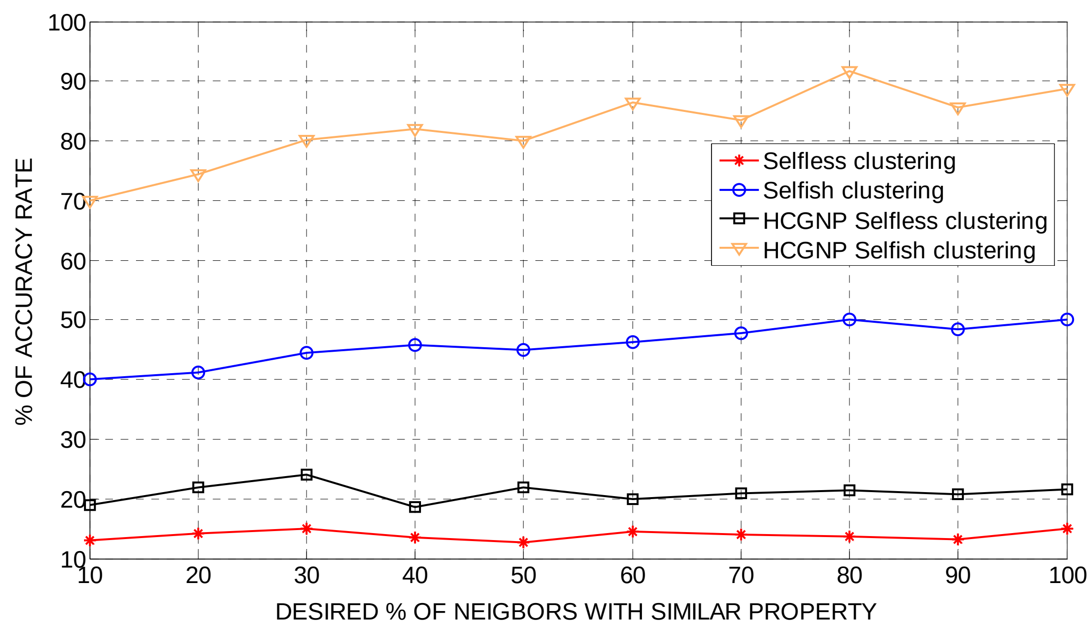


Figure 4.9: Percentage of accuracy rate of selfless and selfish algorithms with HCGNP

4.4 Computational Complexity of HCGNP

After the evaluation of HCGNP on the basis of its computational time, it is revealed that HCGNP is far more efficient than SLACER. The main reason behind this fact is that in SLACER, two protocols i.e. Newscast protocol and PD-Protocol are layered in the configuration file. Newscast is implemented to generate random nodes, while PD-Protocol enhances level of cooperation among the nodes. On the other hand, HCGNP is not a layered approach and is embedded in the newscast protocol. The time a node takes to join the network and compares it capability levels with other nodes, is extremely short. Hence the assignment of CG-ID to the node, showing cooperation of two nodes simultaneously is much efficient than implementing PD-Protocol in a predefined network.

As discussed earlier, SLACER generates nodes with newscast protocol. These nodes are mutated, replicated and reproduced through PD-Protocol to enhance cooperation in the network. The process of node mutation, node replication and node reproduction does not run in parallel. Hence the simultaneous execution of the protocols and node creation methods is not considered in SLACER. This leads to the gain of cooperation with extended amount of time, as the number of cycles required to reach at maximum level of cooperation is very high.

Figure 4.10 clearly shows the difference between the execution time both the protocols take to complete 100 cycles. For example, it can be seen that during 9 cycles (which takes 54 msecs) HCNP gives 94% cooperation while SLACER has achieved only 18% cooperation level. Hence SLACER takes approximately 100 cycles to show maximum percentage of cooperation while HCGNP takes only 9 cycles to show maximum percentage of cooperation. The main reason to achieve this difference in computational complexity is:

- The parallel handling of multiple nodes with multiple physical resources.
- At any instant of time (t), there is always a pair of nodes which show complete cooperating behavior.
- The implementation of HCGNP is based on a single-layered protocol execution.

The difference in execution time of each cycle in SLACER and HCGNP is very marginal. The gain in cooperation within few cycles enhanced the importance of HCGNP for gaining maximum cooperation with minimum utilization of processor cycles and network resources. Thus, HCGNP is a far better approach to enhance cooperation in the network, with reduced computational complexity.

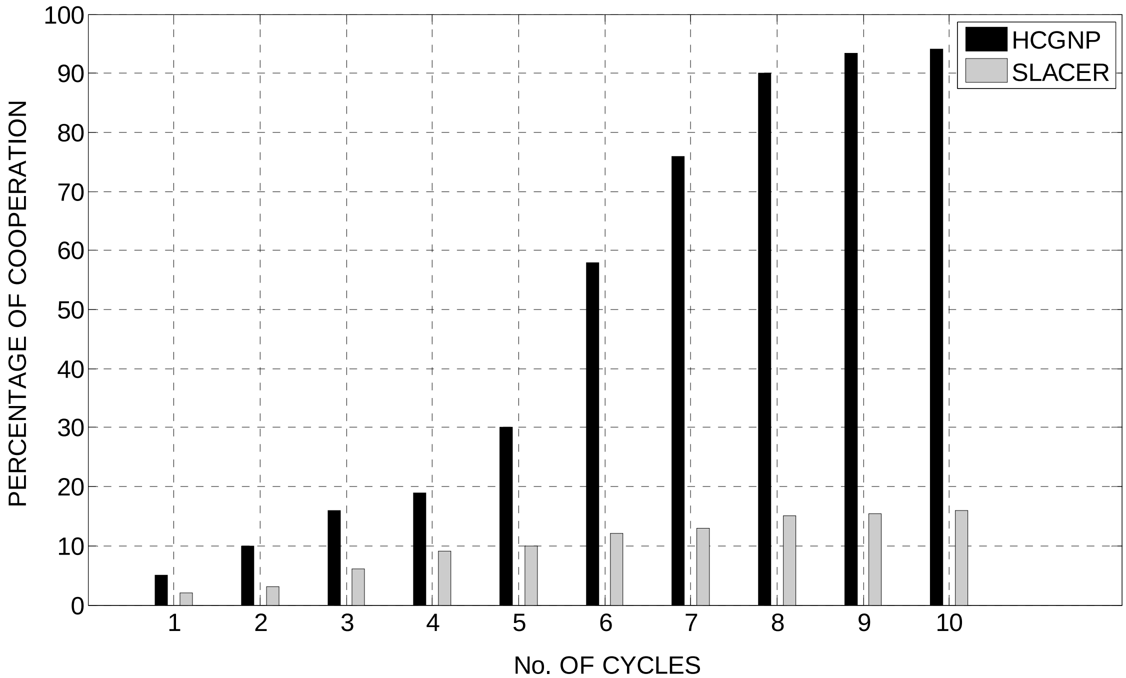


Figure 4.10: Execution time of HCGNP and SLACER to achieve maximum cooperation in 100 cycles

4.5 Summary

The implementation of HCGNP on heterogeneous nodes in a P2P overlay network generates cooperating-groups in the network. Cooperating Group-ID is assigned to those nodes which find a compatible node to exchange resources. This procedure of exchanging resources reveals that nodes are cooperating in the network. The nodes that show cooperation, are rewarded the benefit value which is dependent on the network capacity. The simulation results show that the performance of HCGNP with benefit value (ß) is much better than the performance of SLACER and SNA algorithms. The value of (ß) depends on the network capacity taken as a variable (x). Thus increasing network capacity helps in achieving maximum percentage of cooperation by rewarding more benefits. HCGNP also provides an additional advantage in terms of computational complexity as compared to SLACER algorithm.

Chapter 5: Conclusions & Future Work

5.1 Conclusions

SLACER and SNA show that the increase in the percentage of cooperation among the nodes is achieved after a linear increase in the number of cycles. On the other hand, HCGNP with some benefit given to the cooperating nodes at every cycle shows a rapid increase in the percentage of cooperating nodes, i.e. within 10 cycles where more than 90% of the nodes in the network show cooperating behavior.

SLACER, SNA and HCGNP, all give some extra benefit to the nodes in order to achieve cooperation. This phenomenon seems to create an ideal situation in P2P overlay networks to eliminate free riding and enhance cooperation among the nodes. But realistically, while giving some extra benefit to the nodes the network becomes overloaded. This load is obviously the extra share of bandwidth provided to the cooperating nodes in form of a benefit value. The observations discussed in the previous chapter show that, to enhance the behavior of cooperation in a P2P overlay network the nodes have to maintain some capability level. This capability level should be such that it drives the node to gain its desired resource, without creating any congestion in the network. The results also show that the level of cooperation among the nodes in any network depends upon the network capacity for support. Hence this parameter proves to be the most important, as the desired values of cooperation are totally dependent on it. However the minimum level of cooperation achieved after implementing HCGNP is also more than 50% in which the overall probability of the nodes to cooperate in the network is highest. Hence we can say that HCGNP proves to be more suitable protocol that elevates the percentage of cooperation among the nodes in a P2P overlay environment without over burdening the network and without wasting network capacity. Thus the major advantage of cooperating-groups in a heterogeneous P2P overlay network is that nodes having varying resources can cooperate well if designed on the basis of heterogeneity. Quality of service (QoS) like security protocols, multicasting and network management protocols like tracking the current state of a network, can also be implemented using heterogeneous capabilities of nodes.

The benefit value rewarded to the nodes is the extra bandwidth share. But the type of benefit value may vary according to the network size, type of the network and type of the network resources shared. The criteria to reward a benefit to the node not only depend upon the network capacity, but also depend upon the nodal capacity and nodal state. The most important point needs to be highlighted is the maintenance of network and node state. HCGNP requires an application level enhancement of certain features. Some important features that are required to be deliberated in future are discussed in next section.

5.2 Future Work

One of the fundamental problems, which need to be handled in the current state of HCGNP, is the implementation of security protocols that can be embedded in HCGNP or on top of it. The use of implicit extraction of physical parameters compels the nodes for cooperation. The major threat a node can encounter from the neighboring node in the network is the unaware access of physical resources. A well maintained authentication protocol is required which authenticates the nodes before they join the network. Other security protocols, like encryption of node's cache especially the information related to capability levels, need to be studied and implemented in future.

HCGNP also requires enhancements to maintain a network state at application level. The number of times a node joins a particular cooperating-group should be maintained extensively to accommodate upcoming pair of cooperating nodes in existing cooperating groups. This is required to gain as much cooperation as possible from the existing cooperating-groups rather than developing new cooperating-groups. Another very important enhancement required in HCGNP is the improvement in its dynamic nature in heterogeneous environment. This can be done by expanding the concept of multiple physical parameters under the consideration of different network connections and different network types.

5.3 Summary

The implementation of Heterogeneous Cooperating Group-based Newscast Protocol (HCGNP) is based on the development of cooperating-groups or clusters in P2P overlay networks. These clusters are developed to activate a trend of cooperation among the nodes. Nodes become part of a particular cooperating-group by exchanging its physical parameters and compatibility with the neighboring nodes. This protocol is useful because it does not overburden the network by giving benefits to the nodes for showing cooperative behavior. The nodes after joining the network exchange their capability levels and form cooperating pairs. During the whole procedure, nodes cooperate on the basis of their available and demanded capability levels. No benefit is provided to the nodes to compel it for cooperation, which on the other hand is extremely suitable for those networks which have less network capacities. Although a desired level of cooperation can be gained by implementation of HCGNP, it lacks many other properties. To make it more applicable in a real time environment, it requires implementation of security protocols both at application and transport layer level and, network and node management protocols at application layer level. Thus the efficient handling of multimedia files through multicasting can also be implemented by using grouping or clustering approach.

REFERENCES

[1] R. Subramanin, B. D. Goodman, “Peer to Peer computing: The evolution of a Disruptive Technology”, Idea Group Publishing, 2005.

[2] Eng. K. Lua, J. Crowcroft, M. Pias, R. Sharma and S. Lim, “A Survey and Comparison of Peer-to Peer Overlay Network schemes”, IEEE Communications Survey and Tutorial, March 2004.

[3] S. Arteconi, D. Hales and O. Babaoglu, “Broadcasting at Critical Threshold in Peer-to-Peer Networks”, Department of Computer Science, University of Bologna, Technical Report UBLCS-2006

[4] D. Hales, G. P. Jesi, S. Arteconi and O. Babaoglu, “SLACER: Randomness to Cooperation in Peer-to-Peer Networks”, Department of computer science, University of Bologna, Italy, IEEE-2005.

[5] W. Wang, L. Zhao, R. Yuan, “Improving Cooperation in Peer-to-Peer Systems Using Social Networks”, Tsinghua University Beijing, China, IEEE 2006.

[6]http://www.cisco.com/global/EMEA/cds/corporate_marketing/HA_Clusters_White_Paper.pdf

[7] Atul Singh, Mads Haahr, “Decentralizing clustering in pure P2P overlay networks using Schelling’s model”, ICC’2007, pp. 1860-1866, 2007.

[8] T. K Madson, Q. Zang, F. Fitzek, M. Katz, “Exploiting Cooperation for Performance Enhancements and High Data Rates”, Journal of Communications, VOL. 4, NO. 3, April 2009.

[9] G. Exarchakos, J. Salter, “Semantic Cooperation and Node Sharing Among P2P Networks”, Department of Computing, University of Surrey, Guildford, Surrey, United Kingdom.

[10] Victor O. K. Li, Li Cui, Q. Liu, G. H. Yang, Z. Zhao, “A Heterogeneous Peer-to-Peer Network Testbed”, ICUFN, IEEE, 2009.\

[11] S. Schosser, K. Bohm, B. Vogt, “Competition and Cooperation in Heterogeneous Structured P2P Systems-Are they Mutually Exclusive”, University of Karlsruhe, Germany, 2007.

[12] S. Wang , Y. Zhang , “Reliable Self-Clustering P2P Overlay Networks”, Department of Computer Science and Technology, Tongi University, China, COMPSAC-IEEE-2007.

[13] M. Amad, A. Meddahi, “DV-Flood”: An Optimized Flooding and Clustering based Approach for Lookup Acceleration in P2P networks”, GET/Telecom Lille 1, France, 2008-IEEE.

[14] Atul Singh, Mads Haahr, “Decentralizing clustering in pure P2P overlay networks using Schelling’s model”, ICC’2007, pp. 1860-1866, 2007.

[15] Mouna Kacimi, Kokou Yétongnon, Yinghua Ma, Richard Chbeir, “HON-P2P: A Cluster-based Hybrid Overlay Network for Multimedia Object Management”, ICPADS’2005, University of Bourgogne, 2005.

[16] Akinori Yonezawa, Khaled Ragab, “A Self-organized Clustering-based Overlay Network for Application Level Multicast”, Journal of Networks, April 2009.

[17] James Z. Wang, Matti A. Vanninen, "A Novel Self-Configuration Mechanism for Heterogeneous P2P Networks," Intelligent Agent Technology, IEEE / WIC / ACM International Conference on, pp. 281-287, 2004 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04), 2004

[18] Irum Kazmi, Saira Aslam, & M. Y. Javed, “Cluster Based Peers Configuration with using HCNP in Peer-to-Peer Overlay Networks”, CICSYN 2010, IEEE 2010.

[19] Jing Wu and Michel Savoie, “Peer-to-Peer Network Architecture”, Communication Research Center (CRC), Canada.

[20] Diego Doval and Donal O’Mahony, “Overlay Networks: A Scalable Alternative for P2P”, Trinity College, Dublin, IEEE 2003.

[21] Miguel Castro, Peter Druchel, Ayalvadi Ganesh, Antony Rowstron and Dan S. Wallach, “Secure Routing for Structured Peer-to-Peer Overlay networks”, 5th Usenix Symposium on Operating System Design and Implementation, Boston, MA, December 2002.

[22] Eric Rescorla, “Introduction to Distributed Hash Tables”, Network Resonance, IAB Plenary, IETF 65.

[23] Susan Crosse, Elaine Wilson, AnnMarie Walsh, David Coen, Charles Smith, “P2P Networks: Napster”, TCD 4BA2 Project 2002/2003.

[24] M. Ripeanu, "Peer-to-Peer Architecture Case Study: Gnutella Network", Peer-to-Peer Computing, 2001. Proceedings: First International Conference on (27-29 August 2001), pp. 99-100. Key: citeulike: 785523.

[25] Jun Shi, Jian Liang and Jinyuan You, "Measurements and Understanding of KaZaA P2P Network, Department of Computer and Engineering, Shanghai Jiao Tong University, China.

[26] Erik Veerman, "An Introduction to Fast Track Data Warehouse Architectures", SQL Server 2008, February 2009.

[27] Siu Man Lui and Sai Ho Kwok, "Interoperability of Peer-to-Peer File Sharing Protocols", ACM SIGecom Exchanges, Vol. 3, No. 3, August 2002, Pages 25-33.

[28] Raj Jain, "Virtual Private Networks (A tutorial)", The Ohio State University, Ohio, USA.

[29] Sameh El-Ansary, Luc Onana Alima, Per Brand and Seif Haridi, "Efficient Broadcast in Efficient P2P Networks", Swedish Institute of Computer Science, Kista, Sweden.

[30] Keno Albrecht, Ruedi Arnold, Michael Gahwiler and Roger Wattenhofer, "Aggregation Information in Peer-to-Peer Systems for Improved Join and Leave", Swiss Federal Institute of Technology, Department of Computer Science, Zurich, Switzerland.

[31] http://faculty.lebow.drexel.edu/McCainR/top/eco/game/dilemma.html

[32] Mark Baker and Rahim Lakhoo, "Peer-toPeer Simulators", Technical Report ,University of Reading, UK, 2007.

[33] Naicken. S., Basu. A., Livingston. B., and Rodhetbhai. S., "A Survey of Peer-to- Peer Network Simulators", Proc The Seventh Annual Postgraduate Symposium, Liverpool, UK, (2006).

[34] P2PSim, http://project-iris.net/.

[35] Jelasity. M., Babaoglu. O., "T-Man: Fast gossip-based construction of largescale overlay topologies". Technical Report UBLCS-2004-7, University of Bologna, Department of Computer Science, Bologna, Italy, May 2004.

[36] Montresor. A., Jelasity. M., Babaoglu. O., "Robust Aggregation Protocols for Large-Scale Overlay Networks", IEEE, DSN'04, p 19-24, (2004).

[37] Jelasity. M. Montresor. A., "Epidemic-Style Proactive Aggregation in Large Overlay Networks", IEEE, ICDCS'04, p 102-109, (2004)