Achieving Optimal Capacity Partitioning in P2P Networks using Game and Control Theoretic Approaches

A thesis submitted in fulfilment of the requirements

for the degree of PhD

by

Nitin Singh Singha

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

March 2018

Certificate

It is certified that the work contained in this thesis entitled "Achieving Optimal Capacity Partitioning in P2P Networks using Game and Control Theoretic Approaches" by "Nitin Singh Singha" has been carried out under my supervision and that it has not been submitted elsewhere for a degree.

Dr. Yatindra Nath Singh

Professor

Department of Electrical Engineering

Indian Institute of Technology Kanpur

14 *March 2018*

Synopsis

Name of the student: **Nitin Singh Singha** Roll No: **13104180**

Degree for which submitted: **PhD** Department: **Electrical Engineering**

Thesis title: **Achieving Optimal Capacity Partitioning in P2P Networks using Game and Control Theoretic Approaches**

Thesis supervisor: **Dr. Yatindra Nath Singh**

Month and year of thesis submission: **March 2018**

Traditionally Peer-to-Peer (P2P) network users use wired network technology like LAN for accessing the Internet. However nowadays, wireless technology can provide high speed data service at lower costs. This has led to widespread use of wireless technology in office, home and hotels to provide Internet connectivity. Further, wireless technology is also being used to provide Internet services in public places like airports, railway stations and restaurants etc. Therefore, many P2P users are now using wireless networks like WiFi, WLAN, LTE and WiMAX (in time division duplex (TDD) mode), where nodes are connected to the backbone network through an access link. The uplink and downlink data flows through the same access link. The access link capacity is fixed, but its partitioning between uplink and downlink can be varied by a user such that increasing upload capacity reduces download capacity and vice-versa. In a P2P network, since users act both as a server and a client, so they need to simultaneously utilize their upload and download capacities, optimally. In order to maximize the utilization of resources allocated from the network, users would like to dedicate their entire link

capacity for download. However, incentive mechanism (e.g. Reputation System) employed by a P2P system forces members to maintain certain minimum level of contribution (*i.e.* upload), to receive resources from the network. The amount of resources received by a user is proportional to its contribution level. Therefore in the aforementioned network, there is a need to efficiently divide the link capacity such that a user allocates just enough link capacity for upload, so as to receive resources equal to its current download capacity. This point of capacity partitioning will be referred as optimal capacity partitioning point. During optimal partitioning, the link capacity allocated by any user i for the upload is minimum such that resources allocated by other members to i become equal to its download capacity. If user i further increases its upload capacity, resources allocated to it will increase. However, it will not have enough download capacity to receive all the allocated resources. On the other hand, reducing upload capacity decreases the contribution level thereby reducing resources allocated to the user i w.r.t the optimal point. Therefore, during optimal point partitioning, a user will receive maximum resources from the network.

In earlier P2P networks users were connected to each other through wired network technology like LAN, where capacity assigned to uplink and downlink is fixed. Hence, until now, capacity partitioning was not a major issue in P2P networks. Consequently, not much literature is available dealing with this problem in P2P networks. Further, existing mechanisms on adjusting capacity partitioning used by the other networks cannot be applied to P2P scenario because of the following reasons.

- Majority of existing partitioning mechanisms seek to maximize link capacity utilization based upon the network traffic, *i.e.*, if currently download requests are more then such partitioning schemes increases capacity allocated for download and vice-versa. However, users in P2P network have altogether different requirements where partitioning mechanism should ensure minimal upload to maintain just enough incentive level for a user to completely utilize its download capacity.

- Owing to absence of central controlling authority in P2P network, users require a distributed capacity partitioning mechanism which can be operated independently by each user. However, most of the prevalent techniques used for capacity partitioning require a central authority like network administrator to modify the link capacity.

Therefore, there is a need for investigating the capacity partitioning problem in P2P network. Chapter 1 of this dissertation presents a general overview about P2P networks along with the major research challenges being faced in their implementation. Chapter 1 also states and explain the research problem being investigated in this thesis along with the existing solutions available for this problem.

This thesis deals first with the computation of optimal point of access link capacity partition between upload and download. Deviation from this optimal point can be used as metric to compare the efficiency of various algorithms used to partition the access link capacity. In Chapter 2, we model capacity partitioning as a game for homogeneous P2P network. In homogeneous network all the users have same access link capacity. P2P networks consisting of users connected to the same WiFi or wireless LAN network can be considered as homogeneous network. Under the Nash equilibrium (NE), all users equally partition the link capacity between upload and download. Further, this *NE* comes out to be socially optimal and provides maximum possible download to the users in the network. Therefore, during optimal point operation, users are equally dividing their link capacity between upload and download.

In Chapter 3, we have game theoretically established that resource distribution only on the basis of cooperation level is unfairly biased towards high capacity users. We show that high capacity users can strategically manipulate their upload capacity, so that the resources received by them is much higher than the amount they upload. These extra resources are drawn from the share of genuine low capacity users. In some cases, the low capacity users are unable to receive any resource even if they are uploading with their maximum capacity. We also prove that for fair resource distribution, the resources should be distributed according to ratio of contribution to consumption of resources by the requesters. Further, we extend our game theoretic model to a more generalized model of heterogeneous users in chapter 4. We establish that if resources are distributed in decreasing order of the ratio of the contribution to the consumption of resources by the requesters, then optimal partitioning corresponds to the network condition where each node is equally dividing its link capacity between upload and download.

Secondly, in this dissertation, we have proposed a mechanism which helps users to operate at optimal partitioning level in the P2P network. In Chapter 5, we have modeled capacity partitioning as a feedback control problem, where resources received by a user act as a feedback, which decides its output, *i.e.*, the amount of resources that the user uploads back to the network. The proposed control system implemented in the form of adaptive step size (ASZ) algorithm strives to take the current partitioning level to an optimal partitioning level. ASZ considers many aspects of real time P2P network and automatically adjusts capacity partitioning at the user when a new user enters or an existing user leaves the network. In addition, ASZ can be easily integrated with an existing P2P network, where some users are using partitioning scheme other than ASZ.

Finally, in Chapter 6, we compare ASZ with Reputation-Based Resource Allocation Policy (RRA), an existing capacity partitioning mechanism for P2P networks. RRA uses fixed size for change in capacity partitioning, so capacity partitioning at user never settles down at the optimal value. It is either more or less than what is the optimal value. This oscillatory behavior results in over or under allocation of upload bandwidth leading to wastage of resources. Unlike RRA, ASZ uses variable step size to take care of this inconsistency such that the step size tends to 0 as the current sharing level approaches an optimal value. Thus, total capacity partitioning stabilizes around optimal point leading to enhanced efficiency. We have also compared our proposed scheme with BitTorrent. Due to a more strict resource distribution, ASZ is fairer than BitTorrent, where lesser resources get awarded to free-riders. Further, in chapter 6, using simulation results we have established that the users employing the ASZ are able to operate near the optimal partitioning level in the network. In totality, the present work can provide a very efficient and fair sharing of network resources in P2P network containing.

Acknowledgements

First of all, I extend my sincerest gratitude towards my supervisor Prof. Yatindra Nath Singh, for providing me complete freedom to pursue my research. I also thank him for the invaluable suggestions that he provided me when things got out of my hand. He taught me how to analyse any research problem and manage my tasks on the priority basis. His support went beyond academics. I have learnt from him to be persistent and patient for accomplishing any goal in life.

I would also like to thank Prof. Ruchir Gupta for his guidance and motivation to pursue my research in peer-to-peer networks. Thanks are also due to Gopal Parashari for insightful discussions on game theory. I am particularly indebted to Kalpant Pathak and Chirag Kumar for their valuable support and constructive suggestions during the course of my PhD.

I also express my gratitude towards Prof. A. K. Jagannatham, Prof. R. K. Bansal, Prof. Govind Sharma, Prof. Adrish Banerjee and Prof. Abhishek Gupta for enhancing my knowledge of core subjects.

I express my heartiest thanks to Prof. A. R. Harish, Head, Department of Electrical Engineering, IIT Kanpur for providing me facilities to pursue my research. I am also thankful to all the faculty and staff at IIT Kanpur for their encouragement and support.

I am thankful to my lab members/friends, Ramesh Nath Tripathi,V Sateeshkrishna Dhuli, Sateesh Awasthi, Pallavi Athe, Kumar Gaurav, Anupam Soni, Amit Munjal, Varsha Lohani, Amit Vishwakarma, Rashmi Kushwaha and Anjali Sharma. I am also grateful to other students at IIT Kanpur Mohan Bansal, Anuj Nandanwar, Om Prakash, Sandeep Kumar, Amrit Singh Bedi, Sandeep Reddy, Suresh Sharma, Arti Singh, Saumya Dwivedi and Javed Akhtar for making my stay enjoyable and memorable at IIT Kanpur.

I would also like to thank Mr. Nagender and Brishaspati team for keeping lab devices free from technical glitches.

I am extremely grateful to my parents Dr. A. S. Singha and Mrs. Bhagirathi Singha for their never ending inspiration, unconditional support and invaluable advice throughout my life. They have been source of my strength when the going gets tough. Thanks is also due to my brother Kshitij Singha for his invaluable insights into problems that I encounter on daily basis. A special thought is devoted to my in laws Mr. Sunil and Mrs. Vandana Bhalaik for their support. Finally, I am indebted towards my wife Swati, for her patience, support and understanding without which this dissertation would not have been completed.

(Nitin Singh Singha)

Contents

List of Figures

List of Tables

Abbreviations

P2P	**P**eer **t**o **P**eer
DHT	**D**istributed **H**ash **T**able
ADSL	**A**symmetric **D**igital **S**ubscriber **L**ine
TDD	**T**ime **D**ivision **D**uplex
LAN	**L**ocal **A**rea **N**etwork
WLAN	**W**ireless **L**ocal **A**rea **N**etwork
LTE	**L**ong **T**erm **E**volution
WiMAX	**W**orldwide **I**nteroperability for **M**icrowave **A**ccess
MAC	**M**edium **A**ccess **C**ontrol
TCP	**T**ransport **C**ontrol **P**rotocol
RTT	**R**ound **T**rip **T**ime
NE	**N**ash **E**quilibrium
BR	**B**est **R**esponse
PI	**P**roportional–**I**ntegral

PID	**P**roportional–**I**ntegral–**D**erivative
RRA	**R**eputation-Based **R**esource **A**llocation **P**olicy
ASZ	**A**daptive **S**tep **S**ize **A**lgorithm

Symbols

Notation	Description
R_i	Reputation of node i.
t_{ji}	Trust of i as measured by j.
T_{ji}	Bandwidth received by j from i.
a_{ji}	Bandwidth allocated by i to j.
B_{ji}	Bandwidth demanded by j from i.
d_i	The current download capacity of the node i.
s_i	The current shared capacity of the node i.
C_i	The access link capacity at the node i.
T_i	Total resources node i is currently receiving from network.
l_i	No. of requests catered by the node i.
g_i	No. of requests generated by the node i.
B_{ji}^{fes}	The minimum bandwidth required to support feasible data rate possible on the link between i and j.

B_{min}^{fes}	The minimum value of the B_{ji}^{fes}, among all the possible values of i and j in the network.
$k_{ij_{ovd}}$	Proportionality constant for handling change in bandwidth received by i due to overloading at j.
$\mathcal{A}_i$	Set of serving nodes for the node i.
$\mathcal{Z}_i$	Set of nodes that request resources from node i since the last reputation estimation.
R_{min}	Reputation threshold below which node does not receive any service from network.
R_{in}^{max}	Maximum initial reputation that can be assigned to newcomers.
R_{in}	Initial reputation assigned to newcomers.
U_i	Level of optimality at node i.
U_{ref}	Reference level, signifying optimal partitioning.

Dedicated to my mother, Mrs. Bhagirathi Singha.

Chapter 1

Introduction and Literature Review

1.1 Peer-to-Peer (P2P) Network as a Resource Sharing System

Evolution of the human race requires that information gathered by a person be passed on to the other members of the society. Internet has acted as a widespread medium for information exchange, where a user can easily receive information from any other user in the world. Traditionally, Internet was based on a client-server model where a dedicated server provided content to the requesting clients. This is shown in Fig. 1.1. However, the finite bandwidth available at the server acts as a bottleneck, restricting the scalability of server, to serve only a limited number of clients. At the same time, the client-server model also suffers from the problem of single point of failure.

The problems state above can be solved by using Peer-to-peer networks. Peer-to-peer network, in this dissertation, will also be referred to as P2P network. P2P network consists of a large number of nodes (eg. Computers, PDAs , Smart phones etc.) where nodes act as both server and client simultaneously. A generic P2P network model is shown in Fig. 1.2. P2P network is implemented as a virtual overlay network on top of an existing network. In this overlay network, all the peers are interconnected with each other. In P2P network, as peers are consumers as well as resource providers, therefore the number of resource providers increases

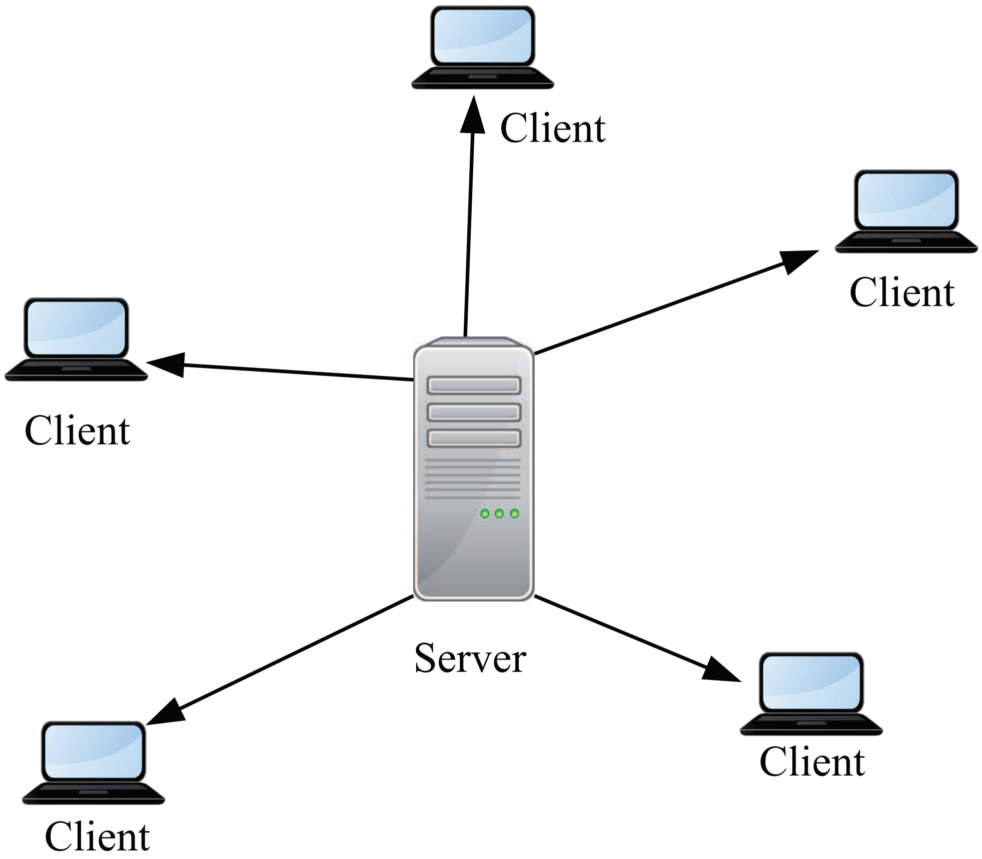

FIGURE 1.1: The Client-Server Model

with increase in the members of the P2P network. This reduces average load at each member in the network, thus making the P2P network highly scalable. As P2P network is scalable it can easily handle increase in number of users in the network. Another feature of P2P network is that it is a distributed network, *i.e.*, it lacks any central authority for controlling, supervising and maintaining the network. These functions are divided among the members of the network. Since a P2P network does not rely on dedicated servers, they exhibit high level of reliability, fault tolerance and resilience from a single point of failure.

P2P networks are used in various applications like content-sharing (especially files) [1], data lookout [2, 3], sharing disk space for storing files [4] and grid computing [5]. Wide applications of P2P networks are possible because of their ability to build a highly resource rich network by aggregating the resources of its members. Consequently P2P networks are able to provide many services traditionally provided by centralized systems at relatively lower cost. This has

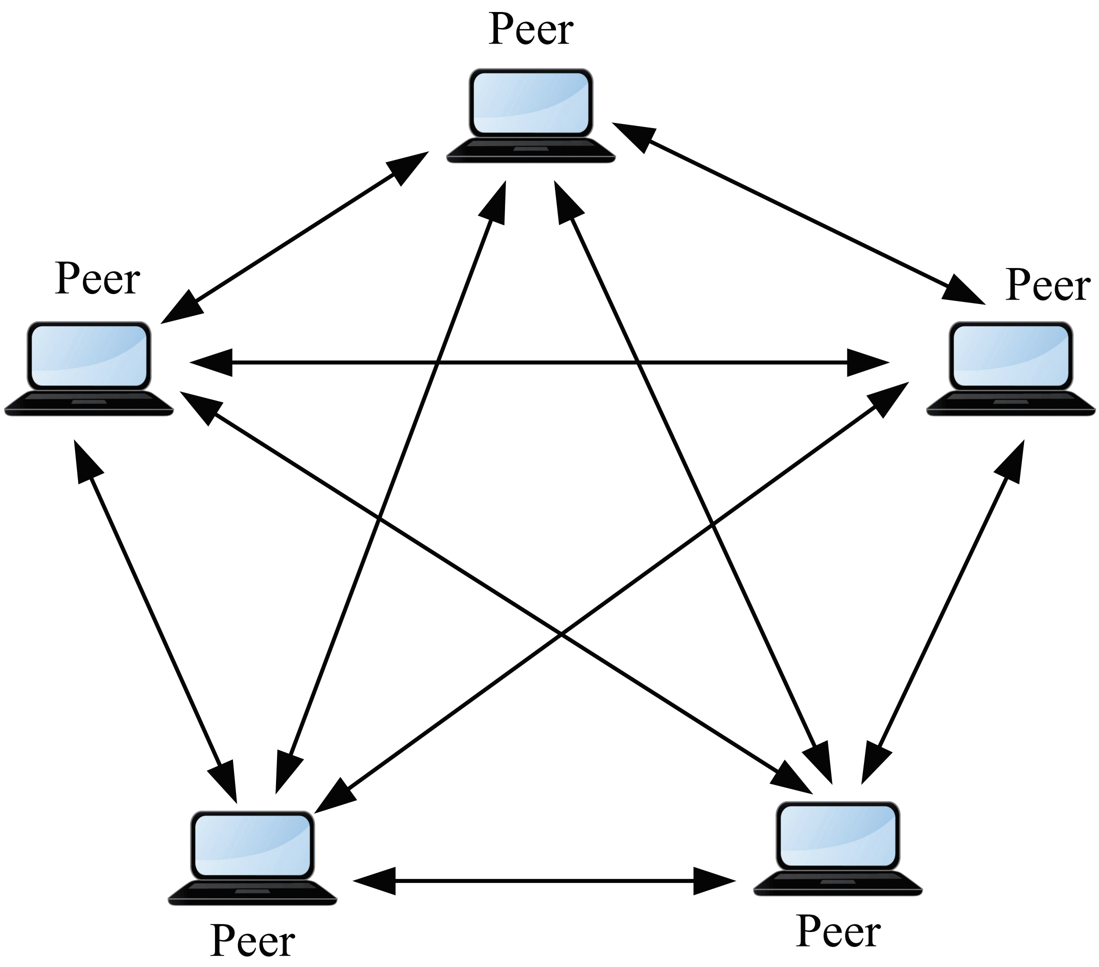

FIGURE 1.2: The P2P Network Model

lead to a surge in the popularity of P2P networks. Subsequently in this thesis, the terms node, peer and member will be used interchangeably to refer to the same thing, *i.e.*, member elements of the P2P network.

1.2 Evolution of P2P Networks

File sharing was initially responsible for the widespread popularity of P2P networks. We, therefore, discuss here the important P2P file sharing networks along with their history. Over time, P2P networks have evolved from a centralized content look out approach (e.g. Napster [6]) to a distributed object query approach (e.g. Gnutella [7]). Based upon the degree of network centralization [8], P2P file sharing networks can be divided into 3 types

1. Centralized P2P network.
2. Decentralized P2P network.
3. Hybrid P2P network.

These networks along with their examples are described next.

1.2.1 Centralized P2P Network

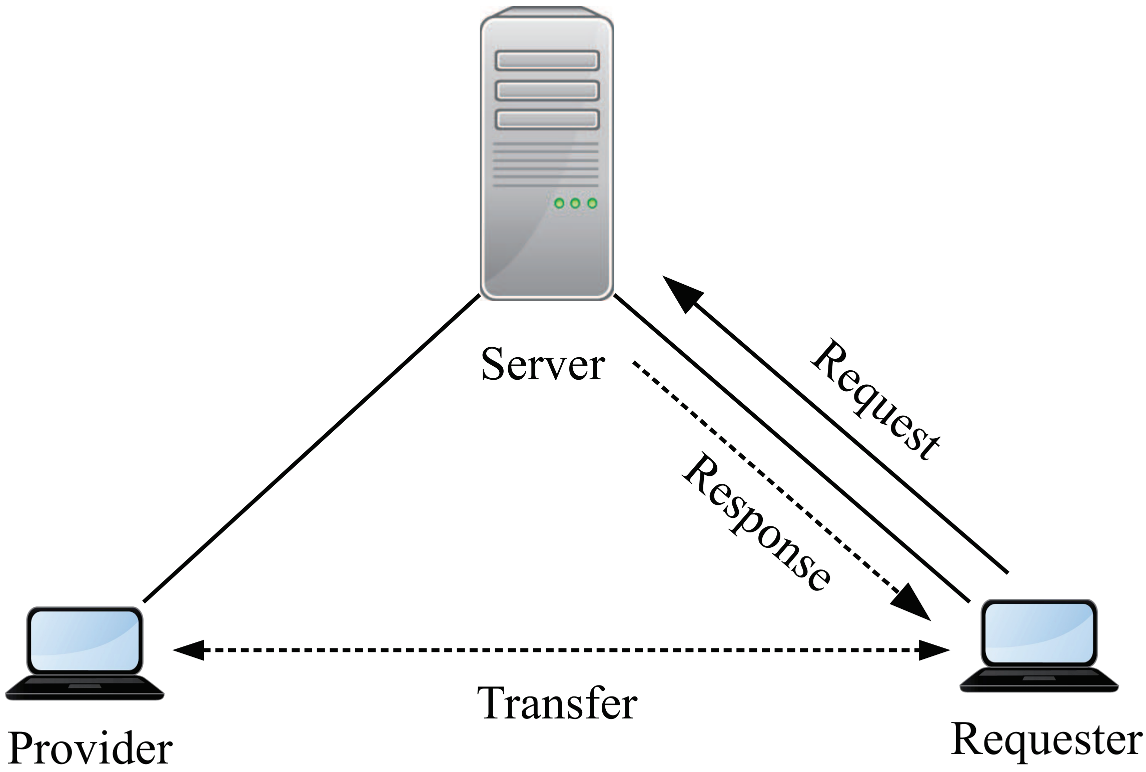

FIGURE 1.3: Centralized P2P Network

Although P2P networks have been seen as an alternative to a centralized client-server model, the first generation P2P networks were based upon the concept of centralization. Unlike client-server architecture, central entity in such P2P network did not provided any content to the users. The central entity use was limited to finding other members and content discovery across the network. The content exchange process in centralized P2P network is presented in Fig. 1.3. A peer first contacts the central server for location of the content. Once content gets located, requesting peer directly connects with the content provider. Central server dependency leads to a single point of failure in such kind of network.

Example : NAPSTER [6], BitTorrent [9] (before version 4.2.0) .

1.2.2 Decentralized P2P Network

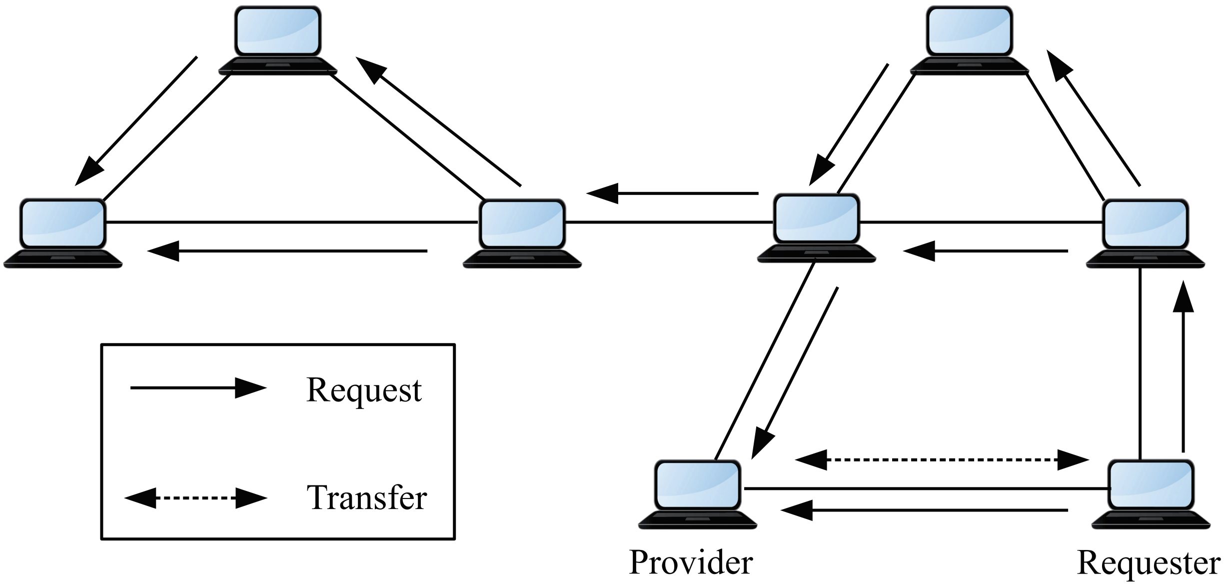

FIGURE 1.4: Decentralized P2P Network

The centralized network suffers from the problem of scalability, single point of failure and legal issues. To overcome these issues, decentralized networks were developed which do not rely on a central server for meta-data information. Peers in such kind of networks dynamically discover other members. Content discovery is usually achieved through flooding in the decentralized network. Once resource gets located, the requesting node directly connects to the serving node as shown in Fig . 1.4. However, flooding leads to huge network traffic. At the same time, in decentralized kind of architecture relatively less number of peers can be discovered. Therefore, resource lookout becomes a difficult task in such networks.

Example : Gnutella [7], Freenet [10].

1.2.3 Hybrid P2P Network

To mitigate the problems of the centralized and decentralized network discussed earlier, hybrid P2P networks as shown in Fig. 1.5, have become popular in recent times. Such a network divides its users into two types, namely super users and users. Super users are assigned additional responsibility of being indexing servers in the network. Users ask super users for the location of the content. Super users initially search for the content locally in the nodes which are assigned to it. However, if content is not available locally, it communicates with other super users for the content address. Super users provide the content location and thereafter the user can directly connect to the content provider for transfer of content. Generally, nodes with high computational power are made super users in the P2P network.

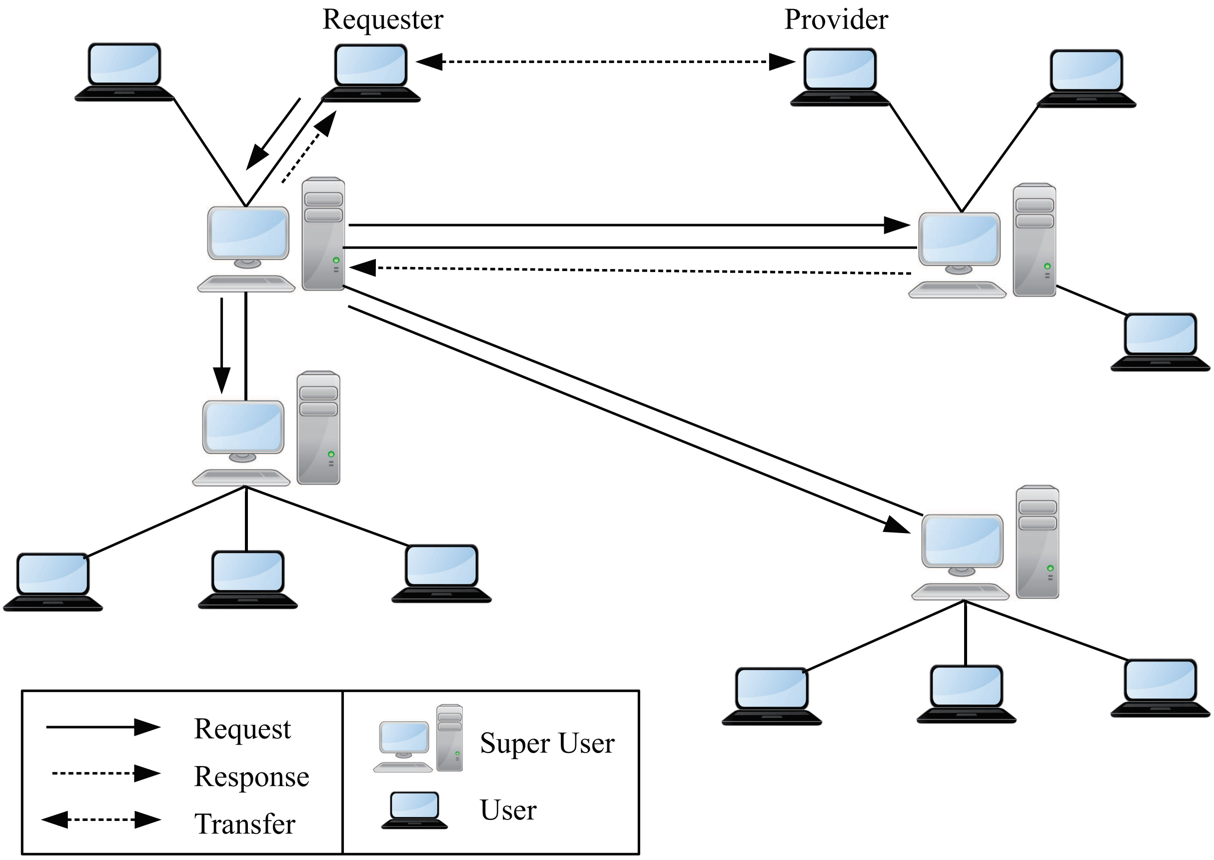

FIGURE 1.5: Hybrid P2P Network

Example : Kazaa [11].

1.2.4 Characteristics of P2P Networks

The following characteristics are generally associated with the P2P networks.

1. *Decentralization :* There is no central controlling authority in a P2P network. The collective actions of the peers determine the overall working of the network. However, certain cases like management of overlay network [12] and monetization of certain operations [13] may require centralized elements. Nevertheless, a central element will never exclusively provide resources in the P2P network.

2. *Symmetry :* All the nodes are assigned equal operational roles, *i.e.*, they act as both server and client. However, in some P2P network this condition is relaxed, and certain nodes are also assigned additional roles [14], e.g. maintaining location table for files. These nodes are referred as super nodes.

3. *Autonomy :* There is no central administrating node which determines the participation of nodes in the network. In P2P network, a node's participation is decided locally.

4. *Anonymity :* P2P networks can also be used to provide anonymous communication between the devices, so that physical location and actual identity of participant is hidden from other members in the network. Interest in anonymous P2P communication has increased in recent years as it avoids litigation issues arising due to distribution of copyright material over the network.

5. *Resource Sharing :* A P2P network thrives on contributions by its member nodes. Ideally, nodes should share in proportion to what they demand. However, there are also nodes who free-ride [15], *i.e.*, they utilize network resources without any appropriate contribution back to the network. In Section 1.5.2, we discuss several incentive mechanisms, which can be used to discourage free-riding in P2P network.

6. *Connectivity :* All the nodes in a P2P network are connected to each other (see Fig. 1.2) and can directly exchange resources with each other within a finite hop count.

7. *Stability :* The P2P network is able to provide service even when existing nodes leave and new nodes enter the network. This implies that, although the topology of the network changes, the nodes are still able to communicate with other members.

8. *Scalability :* The performance of the P2P network does not degrade with increase in the number of nodes in the network. The increase in number of nodes gets compensated by corresponding increase in the count of resource providers available across the network.

1.3 Major Research Issues in P2P Network

The unique characteristics of the P2P network discussed in the previous section, present many challenges in their practical implementation. These challenges need to be addressed to utilize the full potential of P2P technology. The major research challenges that are being faced by P2P network designer are listed below.

1. *Content Identification and Distribution :* The network scale and anonymity of members of the P2P network make it very difficult to identify and distribute resources across the network [16]. Distributed Hash Tables (DHT) [17] are used to store and retrieve content indexes in the P2P network. However, overheads required to maintain DHT tables and inconsistency in resolution of the queries, can be further reduced to improve the content search in the P2P network.

2. *Malicious Peers :* Malicious peers present a serious threat because of anonymous nature of communication in a P2P network. The member peers may receive fake content due to the presence of malicious peers [18, 19] in the network. Malicious peers may also provide false feedback about genuine peers, thereby reducing their reputation. The malicious peers need to be segregated and isolated from normal peers so that they do not affect the working of the P2P network.

3. *Churn :* P2P network suffers from the problem of high churn rate, *i.e.*, nodes continuously joining and leaving the network. Studies [20, 21, 22] conducted

on real-life P2P networks have shown that nearly 50% of the peers get replaced within an hour. This high churn rate in the P2P network causes difficulty in storage and retrieval of information from the network.

4. *Free-riding :* Selfish behavior [23] of members prevent them from voluntarily sharing their resources, and thereby threatens the very existence of P2P network. To motivate peers to share resources, various incentive mechanisms have been proposed, which provide incentive in terms of services received by the node from the network [24, 25, 26, 27, 28]. However, the unique attributes of the P2P network present challenges in design and implementation of the incentive mechanisms in a P2P network. Some of the challenges are

   (a) Absence of central controlling authority.

   (b) Anonymity leading to hidden or untraceable actions of the members.

   (c) Highly dynamic membership, with large number of nodes continuously entering and leaving the network.

   (d) Collusive behavior and false feedback provided by the malicious peers.

   (e) Availability of cheap identities or pseudonyms.

5. *Whitewashing :* In whitewashing [29, 30, 31], users with low contribution level leave the network and rejoin it with new identities to avoid low contribution penalties and exploit the incentives provided by network to newcomers on joining the network. Whitewashing occurs because the members in a P2P network can easily change their identities. Solution to the whitewashing problem is tricky, as providing no incentive to newcomers will discourage peers to whitewash, but it will also deter genuine newcomers from joining the network.

6. *Interoperability :* With the advent of many P2P file sharing applications their interoperability has become a major issue [32]. Various file sharing applications should be able to exchange information with each other to fully utilize the potential of P2P technology.

7. *Load Balancing :* Continuously changing demand for the data items and skewed query patterns in a P2P network lead to overloading [33] of some

members and thereby increase overall response time. Therefore, effective load balancing becomes a necessity in P2P networks.

1.4 Cooperation as Vital Element in Success of P2P Networks

Mast of the current research on P2P technology is devoted to increasing the performance of various content look-out algorithms. However, because of the unique features of P2P network, elementary and challenging issues still remain in motivating peers to cooperate and exchange resources across a P2P network. Most of the traditional models are based on the assumption that members are altruistic, *i.e.*, they blindly share all their resources without thinking about the utility they receive in return. This assumption is not correct in practical P2P networks where members are rational. Rational users will prefer not to contribute resources because of the significant amount of communication and computation cost attached with the contribution, which diminishes the utility they derive from the network. Thus, individual rationality and social welfare are in direct conflict. This leads to the problem of free-riding where many users free ride, *i.e.*, they benefit from the resources and services of the network without contributing anything back to the network. Consequently, most of the resources and service requests are directed toward few P2P nodes in the network which are willing to contribute. This results in degradation in the overall performance of the P2P network. In the next subsection, we briefly discuss different studies conducted on various P2P networks to show the severity of the free-riding problem in such networks.

1.4.1 Experimental Proof of Free-Riding in P2P Networks

Several experimental studies have been conducted in the past which confirm the presence of free-riders in P2P networks. Adar *et al.* in the year 2000 [34], studied the presence of free-riders in Gnutella network. The important observations of their study are given below.

1. Free-riders were found to be uniformly distributed across the network.

2. 70% of users in Gnutella were free-riders.

3. Only 5% of peers uploaded 70% of the files available in the network, while the top 25% were responsible for 98% content in Gnutella.

4. Nearly half of the total file search responses in such network came from 1% of total members, causing congestion in these serving members.

5. 63% of peers did not answer a single query, although they did share some files. This can also be interpreted as free-riding behavior of the nodes, where peers deliberately put content for upload which is of no use to the other peers in the network .

Saroiu *et al.* [35], also studied Gnutella and Napster networks in year 2000. Following were the highlights of this study.

1. Peers indulge in a different type of non cooperative behavior, where they reported lesser bandwidth than the actual amount available with them.

2. In Napster, 40 to 60% of peers contributed to 5 to 20% of their total content.

3. Only 7% of the total peers were responsible for sharing the major portion of the total files in Gnutella.

4. 75% of the members in Gnutella shared upto 100 files only.

One more study was conducted by the Hughes *et al.* in year 2005 [36], which showed that free-riders percentage had further increased to 85% in Gnutella network. This study also reported that free-riding can exist in BitTorrent, a popular file sharing P2P application. In year 2008, Zghaibeh and Harmantzis [37], published a study related to free-riding in BitTorrent. The major results of their study were as follows.

1. Volume of free riders is increasing in BitTorrent.

2. 16.8% of the total peers in BitTorrent network free-ride.

3. The majority of free-riders upload only 5% of their downloaded data.
4. 8% of the total free-rider population upload nothing back to the network.

Cuevas *et al.* [38] also published a study on BitTorrent in 2013. It segregated content publishers in BitTorrent on the basis of their motivation to publish content. The content publishers were divided into 3 groups as discussed below.

- *Altruistic publishers :* These are genuine users who simultaneously consume and publish content in P2P network.
- *Fake publishers :* Fake publishers consist of anti-piracy agencies or malicious users. These publishers are responsible for publication of fake content across the P2P network.
- *Profit-driven publishers :* These publishers are the website owners who use BitTorrent as a platform for the advertisement of their website. These publishers display URL of their website to the user, during content download.

The study conducted by Cuevas *et al.* concluded that,

1. Content distribution across the network by publishers is very skewed, *i.e.*, a very small number of publishers account for significant amount of the published content. Statistically, only 3% of publishers are responsible for 67% of the contents and 75% of download.
2. Fake publishers are incessantly poisoning the content and are responsible for 30% of content, affecting 25% of download sessions.
3. Profit driven publishers, publish nearly 26% of the available content and account for 40% downloads in the BitTorrent.

In a nutshell, all the common P2P networks suffer from the free-riding problem where many peers end up not sharing any resources. The reason for free-riding or non-cooperative behavior can be understood by game theoretically analysing the behavior of the members in a P2P network.

1.5 Game Theoretic Explanation of Free-Riding

1.5.1 Brief Review of Game Theory

Game theory is a mathematical tool, used for decision making in competitive situations. Various competitive situations can be modeled as a game which has an outcome decided by the actions chosen by the participants. Moreover, the payoff received by the participants is determined by the game's outcome .

We assume that $\mathcal{N} = \{1, 2, \cdots, N\}$ represents the set of players (or participants) playing the game. A player i's strategy/action is represented by $s_i \in \mathcal{S}_i$. Player i will always choose a particular action to maximize its utility (u_i). Usually, a steady state or equilibrium arises in the game when no player has any incentive to change their current strategy. This steady state is the Nash equilibrium (NE) and is elaborated in the next subsection.

1.5.1.1 Nash Equilibrium (*NE*)

A strategy profile $\mathcal{S}^* = \{s_1^*,\ s_2^*, \cdots,\ s_i^*, \cdots,\ s_N^*\}$ consisting of the strategies of all the players in the game is a *NE*, if no player i can be better off by choosing a strategy different form s_i^*, provided that every other player $j \in \mathcal{N} \setminus \{i\}$, adheres to its strategy s_j^*. Therefore for $\mathcal{S}^*$ to be a *NE*,

$$u(s_i^*, \mathcal{S}_{-i}) \geq u(s_i, \mathcal{S}_{-i}),\ \forall i \in \mathcal{N}, \tag{1.1}$$

where $\mathcal{S}_{-i} = \{s_1^*,\ s_2^*, \cdots,\ s_{i-1}^*,\ s_{i+1}^*, \cdots,\ s_N^*\}$ is a set of strategies of all the players except i.

At *NE*, every player plays his best response against actions of the other players. The best responses are strategies which provide maximum payoff to a node i for a fixed $\mathcal{S}_{-i}$. It is a set, defined as

$$\mathcal{B}_i(\mathcal{S}_{-i}) = \{s_i | s_i = \arg\max_{\hat{s}_i \in \mathcal{S}_i} U_i(\hat{s}_i, \mathcal{S}_{-i})\} \tag{1.2}$$

Therefore, *NE* in term of best responses of the players is defined as

$$\mathcal{S}^* = \{s_i | s_i \in \mathcal{B}_i(\mathcal{S}_{-i}), \forall i \in \mathcal{N}\}. \tag{1.3}$$

In section 1.5.1.3, we model file sharing between the members of the P2P network as a non cooperative game. In non cooperative games, no cooperation or coordination exists between the players involved in the game. The *NE* of the file sharing game is members not sharing any resource, causing problem of free-riding in the P2P network. The game theoretical justification of free-riding phenomena in P2P network, is described in detail in the following subsection.

1.5.1.2 Prisoner's Dilemma Game

The *Prisoner's Dilemma* [39], is a famous game in game theory which describes rationale of being non cooperative even if cooperative behavior would have been beneficial for all the players (prisoners) involved. This game theoretic modeling of *Prisoner's Dilemma* is as follows.

1. **Players:** The two suspects X and Y.
2. **Action:** Each player's set of action is {cooperate (C), Defect (D)}. When a player cooperate it denies that the other player has committed the crime, whereas it admits involvement of other player in case of defection.
3. **Preferences:** The payoff (*i.e.* number of years for which a player is sentenced) received by the two players for their corresponding actions are listed in Table 1.1.
4. **Objective:** Each player desires to minimize its years of imprisonment.

Now we present detailed analysis of the *Prisoner's Dilemma* game. The game consists of two players/suspects X and Y. They have been apprehended by the police with evidence for some petty crime. Police also knows that they both are involved in bigger crime but lacks any evidence to prove their involvement. Therefore, police sets a trap, where it offers a prisoner (say X) to admit that other prisoner (say Y) has committed the crime. If X admits of Y's involvement in crime, then X will be set free if Y denies X's involvement in crime. However, if Y also admits that X is involved in the crime then X will get a 5 years imprisonment .

	cooperate(Y)	**Defect**(Y)
cooperate(X)	(1, 1)	(10, 0)
Defect(X)	(0, 10)	(5, 5)

TABLE 1.1: Payoff Matrix for Prisoner's Dilemma

	cooperate(Y)	**Defect**(Y)
cooperate(X)	(1, 1)	(10, 0)
Defect(X)	($\boxed{0}$, 10)	($\boxed{5}$, 5)

TABLE 1.2: Best Response of Player X

	cooperate(Y)	**Defect**(Y)
cooperate(X)	(1, 1)	(10, $\boxed{0}$)
Defect(X)	(0, 10)	(5, $\boxed{5}$)

TABLE 1.3: Best Response of Player Y

Alternatively if X denies Y's involvement, then if Y says X is not involved, then both X and Y will receive a lighter sentence of 1 year pertaining to lesser crime, but if Y says X is involved, Y will be set free and X receives a sentence of 10 years of imprisonment. On same line, symmetric arguments hold for player Y.

If X denies Y's involvement, we say X is cooperative, else he is defecting to Y. We have represented all possible scenarios in the game in Table 1.1.

For prisoner X, if he defects he can get imprisonment of 0 or 5 years, whereas if he cooperates, he gets 1 or 10 years. Thus out of the two options, defection is the best strategy for X (refer Table 1.2). The same is true for prisoner Y as shown in Table 1.3. In the above scenario, it is assumed that X and Y cannot communicate with each other. If they are allowed to communicate with each other before giving their statement to police, they both will cooperate. Even if they are allowed to communicate, but police asks from them separately, they both will defect because they will act selfishly in order to optimize for themselves. This is represented in Table 1.4.

	cooperate(Y)	**Defect**(Y)
cooperate(X)	(1, 1)	(10, 0)
Defect(X)	(0, 10)	($\boxed{5}$, $\boxed{5}$)

TABLE 1.4: Nash Equilibrium in Prisoner's Dilemma

	Share(Y)	**Not Share**(Y)
Share(X)	$(D - S, D - S)$	$(-S, D)$
Not Share(X)	$(D, -S)$	(0, 0)

TABLE 1.5: Payoff Matrix for File Sharing Game

We can observe these kinds of scenarios in daily life where optimality for the whole group is sacrificed in favor of self interest, leading to the scenario where individual actually loose. Similar kind of logic holds for file sharing in P2P network. The member nodes in network give preference to their individual benefit, (*i.e.*, they do not share) over social welfare, resulting in their own loss, as network has no resource available which can be downloaded. For better understanding of this situation, we model the interaction between two member as a non cooperative game in the subsequent subsection.

1.5.1.3 Modeling of File Sharing as Non Cooperative Game

The file sharing between two users in a P2P network is modeled as a non cooperative game[1]. The game is described as follows,

1. **Players:** Two member nodes X and Y.
2. **Action:** Each player's set of action is {Share, Not Share}.
3. **Preferences:** The payoff is the utility earned by downloading the resource (D) minus cost incurred (S) in sharing the resource. The payoff received for all possible set of actions is given in Table 1.5.

[1] In non cooperative games, no cooperation can exist between the players involved in the game

	Share(Y)	**Not Share**(Y)
Share(X)	$(D - S, D - S)$	$(-S, D)$
Not Share(X)	$(\boxed{D}, -S)$	$(\boxed{0}, 0)$

TABLE 1.6: Best response of Player X

	Share(Y)	**Not Share**(Y)
Share(X)	$(D - S, D - S)$	$(-S, \boxed{D})$
Not Share(X)	$(D, -S)$	$(0, \boxed{0})$

TABLE 1.7: Best response of Player Y

	Share(Y)	**Not Share**(Y)
Share(X)	$(D - S, D - S)$	$(-S, D)$
Not Share(X)	$(D, -S)$	$(\boxed{0}, \boxed{0})$

TABLE 1.8: Nash Equilibrium of File Sharing Game

From Table 1.6, it is clear that irrespective of the action of node Y, not sharing always earns highest payoff to X. Similarly, on referring Table 1.7, not sharing is most optimal strategy for Y also. When both the players do not share, then they are playing best response against actions of each other (refer Table 1.8). Therefore, the Nash equilibrium for the game is (Not Share, Not Share). This tendency of utilizing resources from the network and not sharing back is termed as free-riding [30]. In practice, if every member follows this strategy then the P2P network will have nothing to share and it will subsequently become defunct.

1.5.2 Solutions to the Free-riding Problem

To overcome free-riding, several mechanisms have been proposed to be introduced in the P2P network, which provide incentive [24, 25, 26, 27, 28, 40] for cooperation. An incentive can be in the form of preference during resource distribution or improvement in the quality of service provided to the contributing peers in the network. Based on the type of incentive received for cooperation, the incentive mechanisms can be further classified as follows.

1. ***Inherent generosity :*** Studies conducted in behavioral economics have shown that models based upon self-interest alone cannot completely explain the behavior of the members in society [41, 42]. Some users gain utility just by altruistically sharing their resources. Feldman *et al.* [43] developed a model which considers peer's generosity while studying free-riding phenomena in the P2P networks. The peers initially calculate contribution cost, which is inverse of the total contributing peers in the network. Every peer will have an inherent generous behavior. If the contribution cost is less than generosity than a node will contribute, otherwise it will free-ride.

2. ***Monetary payment based incentive schemes :*** In such type of incentive mechanism, the requesters pay to the resource provider for the resources consumed by them. Golle *et al.* [28] was the first to study the potential benefits of the payment based schemes in the P2P networks. Monetary payment schemes provide flexible mechanism for exchanging resources in terms of payments. However, there are many practical difficulties involved in the implementation of monetary payment based incentive mechanisms. The implementation difficulties are outlined below.

   (a) Monetary payment based incentive mechanism require dedicated systems for accounting and payments.

   (b) How to ensure that resource providers will reveal the real cost of the resource to the requesters [44].

   (c) How the payments from the requesters get delivered to the resource providers [45, 46].

3. ***Cryptography based incentive mechanism :*** In recent times, T-chain [47] a cryptography based incentive mechanism has been proposed, where a node x will always receive an encrypted file piece on download. In this mechanism, the decryption key corresponding to the received encrypted piece is the incentive. Node x will receives this decryption key when it uploads the received encrypted file piece to another peer in the network. In this way, T-chain coerces every peer to cooperate.

4. ***Direct reciprocation based incentive mechanism :*** In this type of incentive mechanism, decision of a node i serving node j will be entirely based on j's

past behavior toward i. It does not take into account the behavior of j towards the other nodes in the P2P network. "tit for tat" [1] employed in BitTorrent is a direct reciprocation based incentive mechanism. Andrade *et al.* [48] conducted a study which showed that "tit for tat" has appreciably reduced free-riding in BitTorrent. Laoutaris *et al.* [49] proposed BitMax algorithm as an improvement over *"tit for tat"* for improving uplink utilization in BitTorrent for asymmetric digital subscriber line (ADSL) links [50].

The direct reciprocity based incentive mechanism is suitable for those P2P networks, which have long session durations, so that nodes have ample opportunities to reciprocate for the resources provided by the other nodes [23]. In P2P applications, with high churn rate[2] along with infrequent repeat transactions between the peers, reputation based system are more efficient in enforcing cooperation among the users.

5. ***Reputation based incentive mechanism :*** In a reputation based P2P network, every peer maintains the history of its past cooperative behavior with all the other peers in the network. The cooperative behavior of a peer is termed as trust or reputation of the node. This reputation value is used in the decision making process during content distribution. The system which considers reputation during content distribution is termed as the Reputation Management System. Several reputation systems [14, 51, 52, 53, 54, 55, 56] have been proposed in the past, which differ in the reputation calculation mechanism and the subsequent strategies used for resource allocation.

   Eigen-Trust [54] uses the local reputation values calculated by peers to arrive at a global reputation. The local reputation values are weighted by the reputation of peers providing the feedback to prevent collusion by malicious peers. In Peer Trust [19] a peer gives more weightage to reputation feedback of a peer which shows similar trend in rating the other peers' reputation. In Fuzzy Trust [56], a peer calculates the reputation by considering three factors viz., peer's reputation, transaction date and amount. Power Trust [14] uses power nodes for reputation calculation. Power nodes are the most reputed nodes in the network. This mechanism further uses look ahead random walk for the reputation aggregation. By using power nodes

[2] Churn rate is the rate at which nodes join and leave the network.

and employing look ahead random walk strategy, power trust considerably improves the accuracy of the global reputation and reduces reputation aggregation time. Gossip Trust [53] uses gossiping algorithm for reputation aggregation to reduce reputation calculation overhead and fast dissemination of global reputation score across the network. Mengshu *et. al.* [57] used ratio of number of successful transactions to total number of transactions for evaluating the reputation of a node. On the basis of the service quality, PET [55] categorized the transactions into four types. Each type is assigned a different weight. Nodes calculate the average reputation by weighted average of different transactions based upon their type. Satsiou and Tassiulas [15] proposed that reputation should be evaluated as a ratio of resources received to resources demanded by a node. Each peer uses its copy of local reputation value to distribute resources among the requesting nodes. Usually, peers are awarded resources in descending order of their reputation. Sometimes, this may lead to isolation of lower reputation peers from the network. To overcome this problem and to provide a fair chance to lower reputation peers to improve their reputation, probabilistic resource allocation in [58] was proposed. In probabilistic resource allocation, the resources are allocated between requesters in such a manner that probability of a requesting node getting selected for the resource allocation increases with increase in its reputation. Since the reputation affects probability of allocation instead of actual allocation, the lower reputation nodes still have some finite chance of receiving resources. Sometimes a serving node may be cooperative but it is unable to deliver service because it is highly overloaded with requests. Again, due to underlying network congestion a node may not be able to provide resources. In such cases, although the node is cooperative but it receives bad reputation due to network uncertainties. Gupta *et al.* [59] put forward a reputation based system that takes into account various uncertainties at the serving node to arrive at a more accurate estimate of the reputation.

Many game theoretic analyses, have been conducted on reputation based systems to analyse their stability. Goswami *et. al.* [60] analysed the problem of reputation based resource allocation as a non cooperative game and established that cooperation among peers is the best strategy. Goswami *et. al.*

[61] also analysed the evolutionary stability of reputation systems in a P2P network. Their results showed that reputation calculation is not an evolutionarily stable strategy. However, if some initial entry fee for newcomers is introduced, which gets distributed among the existing users involved in reputation evaluation, then reputation calculation becomes evolutionary stable strategy. Hence, reputation based system can be used to mitigate free-riding in P2P network.

Although, reputation systems are one of the popular incentive mechanisms used to reduce free-riding, their implementation in the P2P network is a challenging task. Famous reputation based systems like e-bay [62] uses a central repository for calculating and storing reputation data of all the users. However, in P2P network, due to absence of central server, reputation has to be calculated and stored in a distributed manner. This leads to the following implementation issues.

(a) Reputation Systems are vulnerable to collusive behavior. The peers may give false feedback about the cooperative behavior of others to enhance their own reputation. The effect of collusive behavior gets augmented in network where peers can easily change their identities [63]. The misbehaving peer can create numerous fake identities and provide multiple false reputation reports to the other peers.

(b) How much reputation should be provided to the newcomers is a critical problem in the P2P network [30]. Ideally, for encouraging new peers to join the network, a high reputation value can be provided to them so that they can easily receive service from the network. However, it is easy for peers to change their identities, due to absence of a trusted central authority which can assign strong identities. So a low reputation peer can easily subvert the penalties imposed by the reputation system by regularly changing its identity.

1.5.2.1 Reputation Based Resource Allocation

Once, the reputation of a node gets estimated, a suitable mechanism is required to distribute resources among requesting peers on the basis of their reputation.

BitTorrent uses contribution levels, only for the selection of peers (from the pool of requesting peers), who will be awarded the resource. Thereafter, serving node equally distributes [1, 15] its resources among the selected peers. Ma *et al.* [64] put forward a modified version of the progressive water filling algorithm, where requesting nodes are provided resources at the rates proportional to their reputation. The authors theoretically proved that such kind of allocation maximizes the utility received by the node. Yan *et al.* [65] proposed a mechanism where peers' ranking and the utility were the basis for the resource allocation such that resource distribution achieves max-min fairness. Satsiou and Tassiulas [15] arranged peers in decreasing order of their reputation to demand ration for service delivery. This approach maximizes the satisfaction level of the nodes in the network.

In the next section, we study single capacity links and discuss how the free-riding problem becomes more severe in the networks, which contain such type of links.

1.6 Problem Under Investigation

In certain networks like WiFi, WLAN, LTE and WiMAX (in time division duplex (TDD) mode) [15, 66, 67] the nodes are connected to the backbone network through access link as shown in Fig. 1.6. The uplink and downlink flow shares the common access link. The partitioning of the access link capacity between uplink and downlink is not fixed and can be varied by user or network administrator [67, 68]. We will refer to such type of links as single capacity links.

There exists another type of link which is present in networks like local area network (LAN). In such type of link, fixed capacity is assigned to uplink and downlink, such that access link capacity rearrangement is not possible. The common example is asymmetric digital subscriber loop (ADSL) [50] link.

1.6.1 Single Capacity Links

The basic structure of single capacity links is shown in Fig. 1.6. The access link capacity (C_i) connecting any node i to the network is fixed but its partitioning

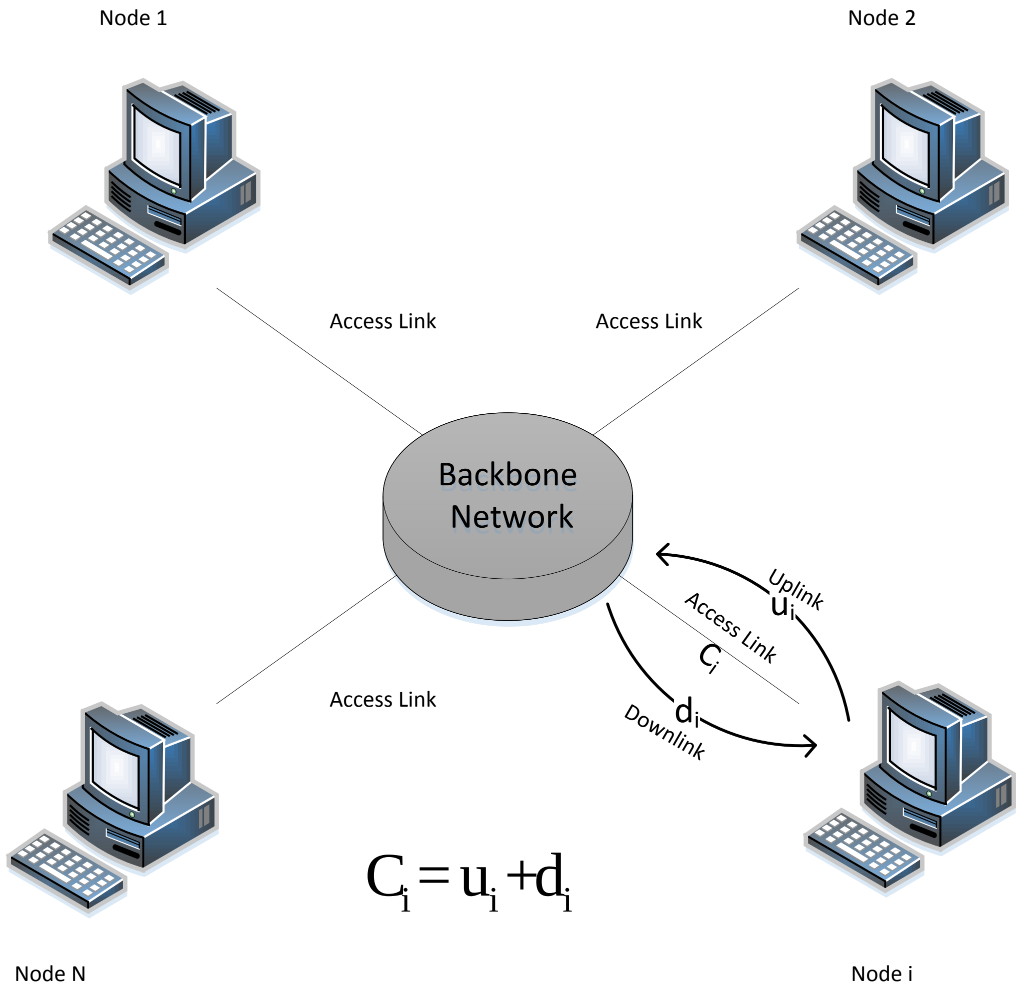

FIGURE 1.6: The P2P Network Model

between upload (s_i) and download (d_i) can be changed such that increasing s_i reduces d_i and vice-versa.

In this thesis, we have assumed that the access link capacity available with the nodes are independent of each other. We have undertaken this assumption to keep the problem tractable. In reality, when network is observed for the shorter duration, the link capacities available to a node will be dependent on the links available to the other nodes in the network. This will certainly happen when shared MACs such as WiFi networks or collision based ETHERNETs are used. However, when long-term averages of the capacities are undertaken, the access link capacities of nodes will become constant and can be assumed independent of each other. Therefore, all the analysis are undertaken in this thesis are applicable only to the scenarios where network observations are made for very long durations.

1.6.2 Capacity Partitioning Problem in P2P Networks

Most of the nodes in P2P networks use network like LAN for communication, where capacity assigned to uplink and downlink is fixed. Hence until now, capacity partitioning was not a major issue in P2P networks. Consequently, not much literature is available to solve capacity partitioning problem in P2P network.

However, with advancement in technology, wireless networks can provide high speed data service at lower costs. Therefore, nowadays there is rapid shift from wired to wireless communication. The wireless technology is being used in office, home and hotels to provide Internet services. The focus is shifting towards using wireless technology to provide Internet services in the public places like airports, railway stations, restaurants etc., where people might require uninterrupted Internet connectivity. Hence, many P2P nodes will be using wireless networks like WiFi, WLAN, LTE and WiMAX (in time division duplex (TDD) mode)[15, 66, 67], which contain single capacity links. In a P2P network, since nodes act as both server and client, so they need to simultaneously utilize their upload and download capacities. Therefore, in network with single capacity links, there arises a need to efficiently divide link capacity so that nodes can maximize utility from the network.

Peers being selfish [69] will always try to utilize their complete link capacity for download. However, incentive mechanism used by the P2P network forces them to upload. In such a scenario, a peer will like to fulfill the minimum contribution/upload requirement, so as to achieve just enough incentive to meet all of its download requirement. Such capacity partitioning will be referred to as optimal partitioning.

If a node operates below optimal point, it will not have enough contribution level to completely utilize its current downlink capacity. The other way around, if the node operates above optimal level its contribution level will increase but its current downlink capacity will reduce w.r.t. download capacity at optimal partitioning point[3]. Hence, resources received by node which can be utilized decreases when it deviates from optimal point of operation. The received resources which can be put into use will be referred to as node's utility. Clearly, at optimal partitioning point a node derives maximum utility from the network. In this dissertation, we propose a mechanism which helps nodes in maximizing their utility by making them operate at optimal partitioning point.

This capacity partitioning scenario is an extension of free-riding behavior of peers, where peers' basic tendency is to free-ride and allocate no capacity for upload. In this way, nodes can utilize their entire link capacity for download. However, the work discussed earlier for preventing free-riding in section 1.5.2 assumed that each peer has separate capacities for its upload and download, which cannot be changed by the peer. This is usually the case in ADSL [50] connections. It is relatively more difficult in single capacity limited network to persuade nodes to cooperate because allocating higher portion of capacity to uplink has negative effect on downlink performance. Therefore, incentive mechanisms discussed under section 1.5.2 cannot be applied directly in the single capacity link system. A capacity partitioning system which optimally divides the access link capacity is needed along with these incentive mechanisms for efficient operation of P2P network containing single capacity links.

In the past, many capacity partitioning mechanisms [66, 67] have been proposed to adjust uplink and downlink flows to enhance the utilization of access link capacity. However, we cannot use these schemes to achieve optimal partitioning in P2P network because of the following reasons.

1. These techniques require a central authority like network administrator to modify the link capacity partitioning according to the current traffic flow. However, P2P network lacks any central authority.

2. The objective of the above mentioned mechanisms is not to control free-riding. These mechanisms aim at maximizing link capacity utilization based

[3] Increasing uplink capacity reduces downlink capacity and vice-versa in single capacity links, as access link capacity is fixed

upon network traffic, *i.e.*, if currently download requests are more in the network, the above mechanism will increase capacity allocated for download and vice-versa. In a P2P network, users have a different requirement, where partitioning mechanism should ensure just enough reputation/incentive for a node so that it can completely utilize its download capacity.

P2P network requires a distributed mechanism which can be implemented independently at nodes such that they help nodes in achieving optimal partitioning. Till recently literature on capacity partitioning in P2P network is limited because most of the P2P nodes still use wired connections. However, with increase in popularity of wireless networks, optimal capacity partitioning will become a serious issue. Important mechanisms for capacity partitioning in P2P networks are presented as follows.

Meo and Milan [69] analysed capacity partitioning as a market. The nodes employ second price auction to purchase and sell to each other, their access link capacity. However, their analysis is restricted to a situation where the nodes are allowed only one upload and download in the network. This restriction limits maximum throughput that can be achieved by the nodes in the network. No node in the network can receive a download rate greater than the capacity of the lowest access link in the network, *i.e.*,

$$d_{\max} = \min_{i \in \mathcal{N}} \frac{C_i}{2},$$

where C_i is access link capacity of node i, $\mathcal{N}$ represents the set of all the network users and $d_{\max}$ is the maximum download rate achieved by any node in the network.

Iosifidis and Koutsopoulos [70] modeled capacity portioning between upload and download as utility maximization problem. However, authors did not provide any mechanism for complete utilization of the access link capacity.

Satsiou and Tassiulas [15], proposed an algorithm which dynamically adjusts the access link capacity between uplink and downlink so that a node receives maximum bandwidth from the network. This algorithm modifies the capacity partitioning at the node in fixed step sizes, so nodes usually overshoot or undershoot the optimal partitioning level. This oscillatory behavior results in reduced utilization of the network resources.

1.7 Aims and Objective of Dissertation

In a network containing single capacity links, there is a need to determine equilibrium/optimal state in the network, such that no node can increase its utility[4] by deviation from this state. This optimal state will act as a benchmark for various partitioning schemes which strive to maximize the utility received at the node by strategically partitioning the access link capacity between uplink and downlink. Ideally, capacity partitioning achieved by any partitioning scheme should finally converge to the optimal state. The greater the deviation from optimal state the lesser is the efficiency of that scheme.

There is a requirement of efficient capacity partitioning mechanism because existing schemes [15][69], are unable to completely utilize the network resources. This may arise due to the following reasons.

1. A limited system model is often considered while designing partitioning mechanism e.g., one method allows only a single upload and download at a time [69].
2. Modification in capacity partitioning may be fixed and independent of difference between current partitioning and optimal partitioning [15].

To overcome these issues, following objectives have been investigated in this dissertation.

1. To calculate an optimal state of capacity partitioning in the network, where multiple uploads and downloads are allowed at the node. At this optimal

[4]Utility is the benefit derived by the node from the network. In current context, it is the portion of received resources, which can be put into use by the node. The concept of utility will be dealt in detail, in the later part of this dissertation.

state, every node is able to receive maximum possible resources from the network.

2. To design an algorithm which will dynamically adjust partitioning of link capacity at the node such that change in capacity partitioning is proportional to the deviation of current partitioning from the optimal point. In this way, nodes will always operate close to the optimal partitioning level. This algorithm, does not require a central authority and thus can be implemented independently at each node.

Based upon these objectives, we have organized this dissertation into the following chapters.

1.8 Dissertation Organization

In Chapter 1:, we briefly discussed the the history of P2P network along with free-riding problem. We also stated and explored the problem in detail, which is further used to define the objectives of this dissertation.
In Chapter 2, we model the capacity partitioning of access link between uplink and downlink as a non cooperative game[5] for a homogeneous network, where all users have the same access link capacity. We calculate Nash equilibrium (*NE*) of this game and also establish that this *NE* is the optimal state of capacity partitioning in the network. This chapter also contains simulation results of various P2P networks e.g., BitTorrent, which further substantiates our claim that *NE* of the current game is indeed the optimal partitioning point of the access link capacity.
In Chapter 3, we establish game theoretically that, if resources are distributed on the basis of contribution level only, then resource allocation process gets dictated by high capacity peers. We further prove that for fair resource allocation, the incentive mechanism should consider both resources consumed and resources contributed by the requesting nodes for the resource distribution.
Based on the incentive parameter for fair resource allocation discussed in Chapter

[5]In non cooperative games, no cooperation can exist between the players involved in the game

3, we extended our game theoretic model to heterogeneous P2P network in **Chapter 4**. We evaluate the game's *NE* and also prove that this *NE* is socially optimal.
In Chapter 5, we model the problem of capacity partitioning as a control feedback problem. Using control theoretic analysis, we present an adaptive step size (ASZ) algorithm which helps nodes to achieve optimal partitioning level. This chapter also contains simulation results, which indicate various characteristics of ASZ, e.g. free-riding control and its adaptability towards random arrival and departure of nodes in the network.
In Chapter 6,we present the comparisons of ASZ with the existing schemes. This chapter also contains simulation results which establish that the capacity partitioning mechanisms mentioned in the related work, converge to *NE* evaluated in Chapters 2 and 4.
Finally, **Chapter 7**, presents the main conclusion of this dissertation along with possible future work.

Chapter 2

Optimal Capacity Partitioning in Homogeneous P2P Network

2.1 Introduction

In a Peer-to-Peer (P2P) network, the nodes need to simultaneously utilize their uplink and downlink capacities because they act both as servers as well as clients. Unlike asymmetric digital subscriber line (ADSL) [50] links, where there is a strict separation between uplink and downlink capacities, in networks like WLAN and Wi-Fi, the uplink and downlink traffic flows through common access link before entering the backbone network. The capacity division of the access link between upload and download can be modified at the nodes [15, 68]. The nodes being selfish will always try to maximize their download from the network by attempting to allocate their entire capacity for download. However, the incentive mechanisms like the reputation system [15, 71] used in the P2P networks force them to maintain a certain level of contribution (upload). In such a scenario, an optimal point is expected to exist, where nodes share minimal resources which are just enough to fulfill all of their download requirements.

In this chapter, we model the capacity partitioning as a game and calculate its Nash Equilibrium (henceforth *NE*) point, where no node has any incentive of unilaterally changing its behavior. In the current context, the point of optimal sharing corresponds to *NE* as node operating at this point, receive maximum bandwidth from the network. To simplify the analysis, we consider a homogeneous P2P network where all the nodes have same link capacity to access the network. Networks like Wi-Fi, and wireless LANs can be modeled as a homogeneous P2P network, where bandwidth is shared among multiple nodes. In such networks, the underlying medium access control (MAC) protocol allows fair and equal distribution of bandwidth among all the nodes.

2.2 Motivation

Many game theoretic analyses [60, 64], have been conducted in the past to analyse the cooperative behavior of the members in the P2P networks. However, most of them do not deal with the division of the link capacity between the upload and download. There is limited literature available on the analysis [69] of capacity partitioning using game theory. Analysis in [69] is based upon a restricted P2P model, where a node is allowed maximum one upload and one download at a time. Most of the literature on capacity partitioning [15, 71], only deals with design of partitioning mechanism for maximizing the resources received, without exploring the point of optimal sharing.

We evaluate *NE* for a more generalized P2P scenario, where nodes can simultaneously perform multiple uploads and downloads. We also prove that this *NE* is also socially optimal, *i.e* the aggregate resources received by the nodes is equal to total resources available across the network. Since, current *NE* efficiently utilizes the network resources, so the capacity partitioning achieved by any algorithm or strategy should finally converge to this *NE*. Deviation from *NE* can be used as metric to compare the efficiency of various capacity partitioning algorithms.

2.3 System Model

We consider a homogeneous P2P network of N nodes, where all the nodes have same access link capacity (C_i), *i.e.*, $C_i = C$, $\forall i \in \mathcal{N}$, where C is a constant. C_i can be divided between upload (s_i) and download (d_i) capacities such that $C_i = s_i + d_i$. Therefore, increase in s_i, decreases d_i and vice-versa.

Similar to P2P models in [15, 69], it is presumed that member nodes employ resource lookup algorithm like distributed hash tables (DHT) [17], flooding etc., to identify the resource providing nodes across the network. After identification, nodes can directly communicate with each other. Further, every node in the network is capable of simultaneously performing multiple uploads and downloads. We assume that nodes randomly request other nodes with equal probability to avoid unnecessary overloading of nodes with higher contribution level. This is a valid assumption which is also true for popular P2P file sharing protocols like BitTorrent. In BitTorrent, a single file gets further divided into large number of chunks [1]. In addition, "rarest first" strategy employed in BitTorrent compels nodes to download rarest chunks first. This ensures that each node usually has data chunks which other downloaders in the network require. Thus, a random request will always deliver some resource of interest to a node. Due to random resource requests which are uniformly directed across the whole network, the average requests received at the node should be equal to the average request generated by a node.

The resources are distributed among the requesting nodes based upon the requesters' contribution levels (*i.e.*, allocated upload capacity). The amount of resource (T_{ij}), a requesting node i receives from a serving node j, is

$$T_{ij} = s_j \frac{s_i}{\sum\limits_{k \in \mathcal{Z}_j} s_k}, \tag{2.1}$$

where s_j is upload capacity of j and $\frac{s_i}{\sum\limits_{k \in \mathcal{Z}_j} s_k}$ denotes the contribution level of i *w.r.t.* all the nodes requesting from j. $\mathcal{Z}_j$ is the set of nodes who are currently requesting resources from the node j. We denote the set of nodes requested by node i as $\mathcal{A}_i$. Let average number of requests made by a node is l, so expected cardinality of set $\mathcal{A}_i$ is l. As discussed earlier, under steady state condition, cardinality of $\mathcal{Z}_j$ is also expected to be l. Thus, the total resources T_i, currently being received by the node

i is

$$T_i = \sum_{j \in \mathcal{A}_i} s_j \frac{s_i}{\sum\limits_{k \in \mathcal{Z}_j} s_k}. \tag{2.2}$$

Utility of a node is equal to the portion of the total received resources, which can be utilized. Let $\mathbb{E}(T_i)$ represent the expected[1] amount of resources that any node i is being allocated from the network. This is the utility provided by the network to the node. However in certain cases, the node may not have enough download capacity d_i to utilize all the received resources, *i.e.*, $\mathbb{E}[T_i] > d_i$. In such a scenario, the node will be able to utilize resources equal to d_i and the remaining resources will be wasted. Therefore, utility for node i is defined as,

$$u_i = \min \{\mathbb{E}[T_i], d_i\}. \tag{2.3}$$

Every node in the network strives to maximize its utility. Based on this objective, we formulate capacity partitioning as a non co-operative game[2] in the next section.

2.4 Capacity Partitioning as Non-Cooperative Game

Let $\mathcal{G} = [\mathcal{N}, \{\mathcal{S}_i\}, u_i]$ represents the capacity partitioning game. $\mathcal{N} = \{1, 2, \cdots, N\}$ represents the set of players which are the nodes in the P2P network. Every player strategically allocates the capacity for upload ($s_i \in \mathcal{S}_i = [0, C_i]$, $\forall i \in \mathcal{N}$) so as to maximize its payoff or utility (u_i).

Using (2.3), payoff/utility for node i is given by,

$$u_i(s_i, \mathcal{S}_{-i}) = \min \{\mathbb{E}[T_i], d_i\}, \tag{2.4}$$

where $\mathcal{S}_{-i} = \{s_1, s_2, \cdots, s_{i-1}, s_{i+1}, \cdots, s_N\}$ represent strategies of all players except player i. In subsequent sections, utility received will also be referred to as *usable*

[1] We consider expected value of the resources received (T_i), because value of T_i keeps on fluctuating with time even when node's contribution level remain unchanged. This happens due to change in network dynamics caused by many factors like congestion in backbone network, variation in number of requests arriving at the serving nodes etc.

[2] In a non co-operative game no co-operation can exist between the players involved in the game.

bandwidth received by the node.

2.4.1 Nash Equilibrium *(NE)* Analysis

The strategy profile[3] ($\mathcal{S}^*$) is a *NE*, when no player can be better off by opting for any unilateral deviation *i.e.*

$$u_i(s_i^*, \mathcal{S}_{-i}^*) \geq u_i(s_i, \mathcal{S}_{-i}^*) \ \forall i \in \mathcal{N} \text{ with } s_i \in \mathcal{S}_i. \tag{2.5}$$

At *NE*, every player plays his best response against actions of other players. The best response are strategies which provide maximum payoff to a node i for a fixed $\mathcal{S}_{-i}$. It is a set, defined as

$$\mathcal{B}_i(\mathcal{S}_{-i}) = \{s_i | s_i = \arg\max_{\hat{s}_i \in \mathcal{S}_i} u_i(\hat{s}_i, \mathcal{S}_{-i})\} \tag{2.6}$$

Therefore, *NE* is a point when every player is playing his best response against each other.

$$\mathcal{S}^* = \{s_i | s_i \in \mathcal{B}_i(\mathcal{S}_{-i}), \forall i \in \mathcal{N}\}. \tag{2.7}$$

Theorem 2.1. *For "capacity partitioning game" in a homogeneous P2P networks, following NEs exist*

1. *All the nodes allocating their entire capacity for download i.e.* $\mathcal{S}_0^* = \{s_i | s_i = 0, \forall i \in \mathcal{N}\}$.
2. *All the nodes equally divide their link capacity between upload and download, i.e.,* $\mathcal{S}_{\frac{C}{2}}^* = \left\{s_i | s_i = \frac{C}{2}, \forall i \in \mathcal{N}\right\}$.

Proof. Let $S_{\text{total}} = \sum_{j \in \mathcal{N}} s_j$ and $D_{\text{total}} = \sum_{j \in \mathcal{N}} d_j$ represent the total resources available and demanded across the network respectively. To determine all the *NE* states feasible in the network, we analyse all the possible combinations of S_{total} and D_{total} in the network.

Case 1. $\mathbf{S_{\text{total}} < D_{\text{total}}}$

Two sub cases arise here.

[3]Strategy profile is a set consisting of the strategies played by all the players.

- **Sub Case I :-** $S_{\text{total}} = 0$.
  $\mathcal{S}_0^* = \{s_i | s_i = 0, \forall i \in \mathcal{N}\}$ represents the strategy profile of players when $S_{\text{total}} = 0$. As $S_{\text{total}} = 0$, so the utility received by any node i is

$$u_i(s_i, \mathcal{S}_{-i}) = 0 \; \forall i \in \mathcal{N}.$$

  If i starts increasing its s_i, it will still not receive any resource from network as remaining nodes do not upload. Therefore, node's utility remains unchanged, with no incentive for it to deviate. Hence, $\boldsymbol{\mathcal{S}_0^*}$ **is a *NE***.

- **Sub Case II :-** $S_{\text{total}} > 0$.
  Due to scarcity of shared resources ($S_{\text{total}} < D_{\text{total}}$), there is at least one node in network, which is receiving resources less than its download capacity. This node will start uploading more to enhance its contribution level and thereby increase its chances of getting more resources from the network. Hence, some of the nodes in the network have incentive to deviate. Consequently, this network state is not *NE*.

Case 2. $\boldsymbol{S_{\text{total}} = D_{\text{total}}}$
It can be further divided into two sub cases.

- **Sub Case I :-** $s_i = d_i, \; \forall i \in \mathcal{N}$.
  Here, all the nodes allocate equal portion of their capacity for upload and download. Therefore, strategy profile is given by $\mathcal{S}_{\frac{C}{2}}^* = \left\{s_i | s_i = \frac{C}{2}, \forall i \in \mathcal{N}\right\}$, where C is the access link capacity. From (2.2), the expected amount of resources received by the node i is

$$\mathbb{E}[T_i] = \mathbb{E}\left(\sum_{j \in \mathcal{A}_i} s_j \frac{s_i}{\sum\limits_{k \in \mathcal{Z}_j} s_k}\right). \tag{2.8}$$

  To ascertain stability of $\mathcal{S}_{\frac{C}{2}}^*$, we analyse change in the utility received by i due to its unilateral deviation *i.e.* strategies of all the nodes except i will remain fixed, when the node i attempts to deviate. To simplify the analysis we divide node $i's$ region of operation into 3 regions as shown below

$$u_i(s_i, \mathcal{S}_{-i}) \in \begin{cases} u_i^1 & \text{when } S_i = \frac{C}{2}, \\ u_i^2 & \text{when } S_i < \frac{C}{2}, \\ u_i^3 & \text{when } S_i > \frac{C}{2}, \end{cases}$$

where, u_i^1, u_i^2 and u_i^3 corresponds to utility of node i for these three regions.

• u_i^1 *Calculation:* Here, $s_j = \frac{C}{2}$, $\forall j \in \mathcal{N}$ and the cardinality of the set $\mathcal{Z}_j$ is l (refer to section 2.3), so $\mathbb{E}\left(\frac{s_i}{\sum_{k\in\mathcal{Z}_j} s_k}\right) = \frac{1}{l}$. In addition, node generates l requests, so expected amount of resources $\left(\sum_{j\in\mathcal{A}_i} s_j\right)$ available for download are $l.\frac{C}{2}$. Therefore, (2.8) reduces to

$$\mathbb{E}[T_i] = l.\frac{C}{2}.\frac{1}{l} = \frac{C}{2} \tag{2.9}$$

The current download capacity ($d_i = C - s_i$), is also equal to the resources available. Using (2.4), the payoff of i is

$$u_i\left(\frac{C}{2}, \frac{C}{2}_{-1}\right) = u_i^1 = \frac{C}{2}. \tag{2.10}$$

where $\frac{C}{2}_{-1} = \left\{s_j | s_j = \frac{C}{2}, \forall j \in \mathcal{N} \backslash \{i\}\right\}$.

• u_i^2 *Calculation:* In this region, the node i reduces its upload capacity such that $s_i = \frac{C}{2} - \Delta_2$, for $\Delta_2 > 0$. Hence,

$$\mathbb{E}\left(\frac{s_i}{\sum_{k\in\mathcal{Z}_j} s_k}\right) = \frac{\frac{C}{2} - \Delta_2}{\frac{lC}{2} - \Delta_2}$$

$$\mathbb{E}\left(\frac{s_i}{\sum_{k\in\mathcal{Z}_j} s_k}\right) = \frac{1}{l}\left(1 - \frac{2\Delta_2}{C}\right)\left(1 - \frac{2\Delta_2}{lC}\right)^{-1}$$

Using binomial expansion, $\mathbb{E}\left(\frac{s_i}{\sum\limits_{k\in\mathcal{Z}_j} s_k}\right)$ can be approximated as

$$\mathbb{E}\left(\frac{s_i}{\sum\limits_{k\in\mathcal{Z}_j} s_k}\right) \approx \frac{1}{l}\left(1-\frac{2\Delta_2}{C}\right)\left(1+\frac{2\Delta_2}{lC}\right) \approx \frac{1}{l}-\frac{2\Delta_2}{C}\left(\frac{l-1}{l^2}\right).$$

Putting this in (2.8), the expected amount of resources received from the network is

$$\mathbb{E}[T_i] \approx l.\frac{C}{2}\left[\frac{1}{l}-\frac{2\Delta_2}{C}\left(\frac{l-1}{l^2}\right)\right]=\frac{C}{2}-\Delta_2\left(\frac{l-1}{l}\right).$$

Hence payoff received by node i (refer (2.4)) is

$$u_i\left(\frac{C}{2}-\Delta_2,\frac{C}{2}_{-1}\right)=u_i^2 \approx \frac{C}{2}-\Delta_2\left(\frac{l-1}{l}\right).$$

Comparing u_i^2 with u_i^1 from (2.10), we get $u_i^2 < u_i^1$. Therefore, all the strategies with $s_i < \frac{C}{2}$, are strictly dominated by the strategy $s_i = \frac{C}{2}$. A node never plays strictly dominated strategy in equilibrium because it always yields a lesser payoff.

• u_i^3 *Calculation:* This region of operation includes all the strategies of i with $s_i > \frac{C}{2}$, *i.e.*, $s_i = \frac{C}{2}+\Delta_3$ for $\Delta_3 > 0$. Consequently,

$$\mathbb{E}\left(\frac{s_i}{\sum\limits_{k\in\mathcal{Z}_j} s_k}\right)=\frac{\frac{C}{2}+\Delta_3}{\frac{lC}{2}+\Delta_3}$$

$$\mathbb{E}\left(\frac{s_i}{\sum\limits_{k\in\mathcal{Z}_j} s_k}\right)=\frac{1}{l}\left(1+\frac{2\Delta_3}{C}\right)\left(1+\frac{2\Delta_3}{lC}\right)^{-1}$$

Using binomial expansion, $\mathbb{E}\left(\frac{s_i}{\sum\limits_{k\in\mathcal{Z}_j} s_k}\right)$ can be approximated as

$$\mathbb{E}\left(\frac{s_i}{\sum\limits_{k\in\mathcal{Z}_j} s_k}\right) \approx \frac{1}{l}\left(1+\frac{2\Delta_3}{C}\right)\left(1-\frac{2\Delta_3}{lC}\right) \approx \frac{1}{l}+\frac{2\Delta_3}{C}\left(\frac{l-1}{l^2}\right).$$

Putting this in (2.8), the expected resources received by i is

$$\mathbb{E}[T_i] \approx l.\frac{C}{2}\left[\frac{1}{l} + \frac{2\Delta_3}{C}\left(\frac{l-1}{l^2}\right)\right] = \frac{C}{2} + \Delta_3\left(\frac{l-1}{l}\right).$$

Thus, received resources are higher, than earlier two cases. However, i does not have sufficient download capacity to utilize these resources. Available download capacity is $\frac{C}{2} - \Delta_3$, so using (2.4), the payoff of i is

$$u_i\left(\frac{C}{2} + \Delta_3, \frac{C}{2}_{-1}\right) = u_i^3 = \frac{C}{2} - \Delta_3. \tag{2.11}$$

Comparing (2.10) and (2.11), we get $u_i^3 < u_i^1$. As $s_i = \frac{C}{2}$, strictly dominates all the strategies where $s_i > \frac{C}{2}$, so nodes will never play $s_i > \frac{C}{2}$ strategies in the equilibrium.

Clearly, strategy $s_i = \frac{C}{2}$, $\forall i \in \mathcal{N}$, delivers the maximum payoff, so $\boldsymbol{S}^*_{\frac{C}{2}}$ **is a** ***NE***.

– **Sub Case II :-** $s_i \neq d_i$, $\exists i \in \mathcal{N}$.
  Here, network will contain at least one node i whose $s_i > d_i$ and one node j whose $s_j < d_j$, to maintain $S_{\text{total}} = D_{\text{total}}$. As $S_{\text{total}} = D_{total}$, the download requirements of almost all the nodes will be met. Nodes like i, with higher contribution level can further enhance their utility by increasing their download capacity. The extra resources received by i will be at the expense of reduction in resource received by node with lowest upload capacity in the network. As node i, can increase its utility by changing its strategy, so current network state is not *NE*.

Case 3. $\boldsymbol{S_{\text{total}} > D_{\text{total}}}$
In this case, there will be $(S_{\text{total}} - D_{\text{total}})$ amount of unused resources available for download, after meeting the download requirement of all the member nodes. Any node can utilize these unused resources by increasing its download capacity. Hence, $S_{\text{total}} > D_{\text{total}}$ is not a *NE*. □

Only two *NE* states, $\mathcal{S}^*_0$ and $\mathcal{S}^*_{\frac{C}{2}}$ are possible in *capacity partitioning game*. However in practice, network will never operate in $\mathcal{S}^*_0$ state. Nodes join the P2P network with the motive of receiving resources. If the nodes are unable to receive any resource they will leave the network. Therefore, before network reaches $\mathcal{S}_0^*$ state, nodes would have already left the network.

For maximizing download, nodes try to minimize their upload capacity usage but will never reduce it to 0. Hence, in a network employing incentive mechanism (like reputation system), where resources are distributed based upon requester's contribution level, the capacity partitioning converges to $\mathcal{S}_{\frac{C}{2}}^* = \left\{s_i | s_i = \frac{C}{2}, \forall i \in \mathcal{N}\right\}$. Further, at $\mathcal{S}_{\frac{C}{2}}^*$, nodes receive maximum utility. In order to verify the above hypothesis, we resorted to simulation. The observations indicate that the hypothesis of *NE* at $\mathcal{S}_{\frac{C}{2}}^*$ is true. Next section presents the results of simulations.

2.5 Simulation Results

We have developed an event-driven simulator in MATLAB® to evaluate performance of various partitioning strategies employed by the nodes. We simulated a P2P network containing 100 peers of same access link capacity (18 Mb/s) as in [15, 71]. In this network, peers share bandwidth for file sharing. Peers generate bandwidth request and randomly direct them to the other peers in the network. The resources are distributed by the serving node among the requesting nodes based on equation (2.2). We assume that the network is operating in $\mathcal{S}_{\frac{C}{2}}^*$ state, *i.e.*, the link capacity allocated by all the nodes for upload and download are 9 and 9 Mb/s respectively. To demonstrate that this state is indeed a *NE*, a node i is randomly chosen and the partitioning of its link capacity is varied, to determine whether there exists any other strategy which can provide higher utility to i than *NE* strategy. The utility/usable bandwidth observations are presented in Fig. 2.1. For the remaining nodes, the link capacity partitioning is fixed with $s_j = d_j, \forall j \in \mathcal{N} \backslash \{i\}$.

Fig. 2.1, displays the usable bandwidth received when the node i follows x-y partitioning strategy, *i.e.*, it allocates x and y amount of capacities for upload and download respectively. When node equally partitions its access link capacity between upload and download (*i.e.* $x = y = 9$ Mb/s) it derives maximum usable bandwidth from the network. If a node uploads less than 9 Mb/s, it receives lesser resources due to lower contribution level. Other way around, when node uploads

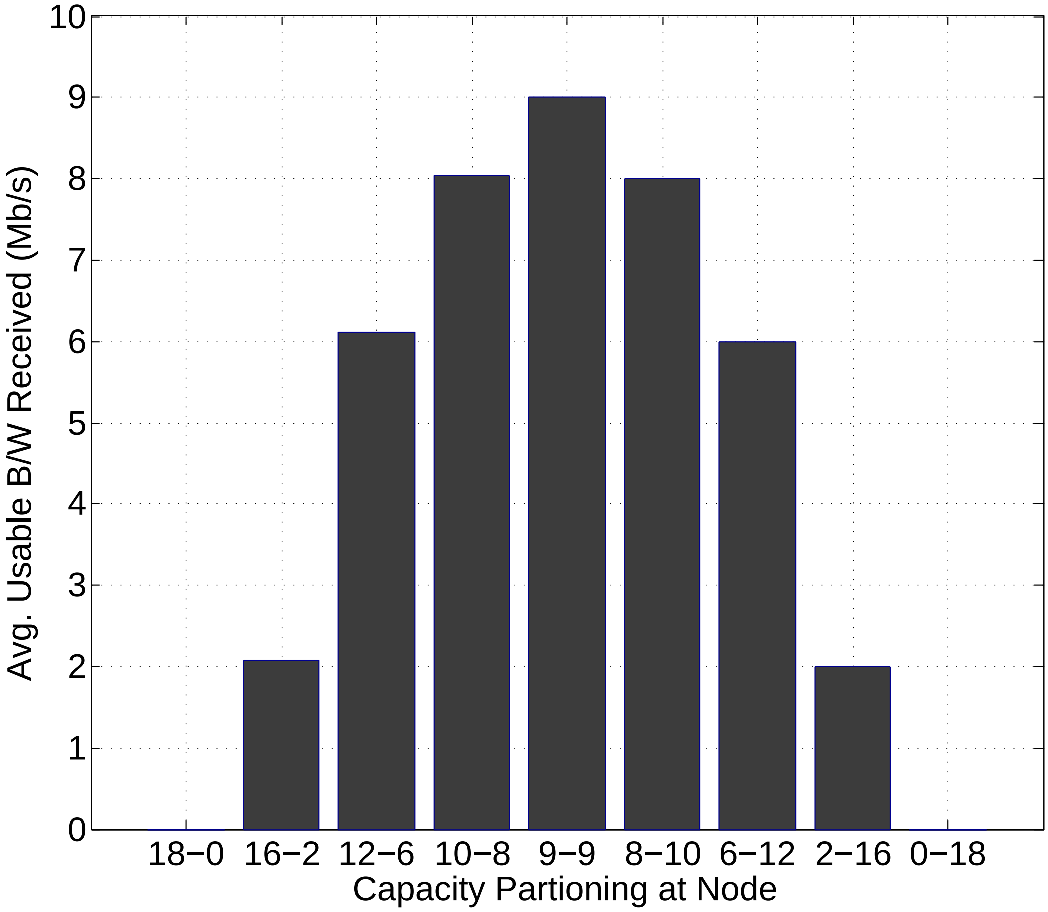

FIGURE 2.1: Change in Received Usable Bandwidth, with Variation in Capacity Partitioning at Node.

more than 9 Mb/s, although the resources received start increasing but node does not have enough download capacity to utilize all the received resources. Hence utility received by node starts decreasing as we move to the right of (9-9) point in Fig. 2.1.

Therefore, a node receives maximum payoff at (9-9) point, which corresponds to strategy employed in $\mathcal{S}^*_{\frac{C}{2}}$. Since, node has no incentive to deviate from (9-9) partitioning strategy, so $\mathcal{S}^*_{\frac{C}{2}}$ is indeed a *NE*.

2.6 BitTorrent Simulation

We simulated 100 member model of BitTorrent [1, 15], based upon Azures, a popular BitTorrent Client. The access link capacity of every member is 18 Mb/s. We discretized the time into periods such that at the starting of a period, any node i, implements *tit for tat* strategy [1] where top four nodes providing the highest download rate to i are selected for being served. To find better service providers, every node i employs a random unchoke policy at the end of every third period, where i randomly selects a node from the whole network for resource distribution. After finalizing the requesters, the upload capacity is allocated equally among the requesting nodes.

BitTorrent lacks algorithm for optimal division of total capacity between uplink and down link [15]. Therefore, we fix capacity partition arrangement at the nodes and analyse the partition that gives the best payoff. We divide different nodes in the network into 9 groups on the basis of their partitioning arrangement. A group x-y denotes that the nodes have allocated x and y units of its total capacity for download and upload respectively, as shown in Fig. 2.2.

From Fig. 2.2, it is clear that point of maximum payoff (*i.e.* 9-9) coincides with the Nash Equilibrium evaluated using Theorem 2.1. The reason for a nodes' on the left side of (9-9) in Fig. 2.2, receiving lower payoff is due to their lower contribution level, whereas nodes on the right side of (9-9) receive lesser payoff due to insufficient download capacity to utilize the received resources. Since, nodes always strive to receive maximum payoff so network will finally reach *NE* state $\mathcal{S}^*_{\frac{C}{2}}$.

2.7 Socially Optimal Partitioning Strategies

NE only considers the node's individual benefit and may not be socially optimal *i.e.* the overall network resources may not get efficiently distributed among the member nodes. In the present game, efficiency will reduce whenever resource wastage happens due to imbalance between net upload and download bandwidth available across the network. We determine the maximum achievable network

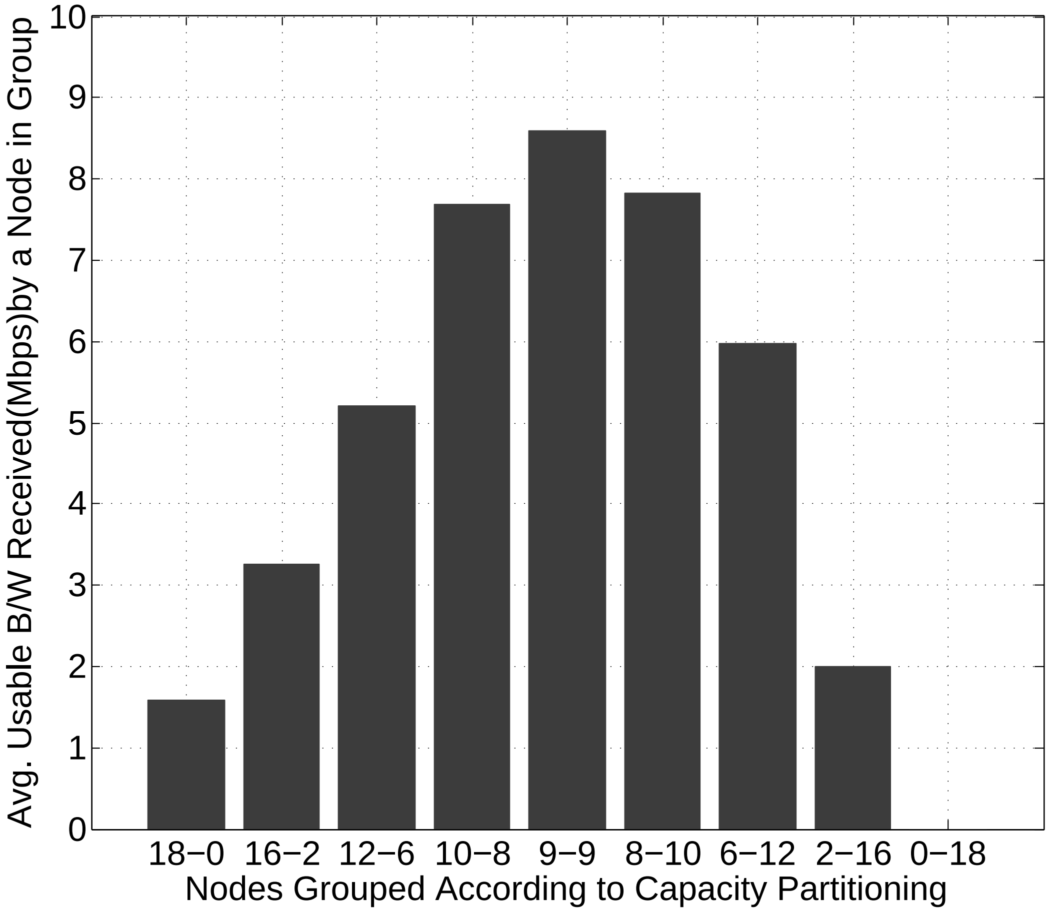

FIGURE 2.2: Change in Received Usable Bandwidth when a Node Changes its Capacity Partitioning in BitTorrent.

utility and compare it with aggregate utility achieved in *NE* state to analyse the social optimality of *NE*.

Let $C_{\text{nwk}}\left(=\sum_{j\in\mathcal{N}} C_j\right)$, $S_{\text{nwk}}\left(=\sum_{j\in\mathcal{N}} s_j\right)$ and $D_{\text{nwk}}\left(=\sum_{j\in\mathcal{N}} d_j\right)$ denote the total access link capacity, resources available and demanded respectively across the network. Sum of upload and download is equal to sum of link capacities of all the nodes in the network. Therefore,

$$S_{\text{nwk}} = C_{\text{nwk}} - D_{\text{nwk}} = NC - D_{\text{nwk}}, \tag{2.12}$$

where C is the individual node's link capacity in network of N nodes. $u_{\text{nwk}}\left(=\sum_{j\in\mathcal{N}} u_j\right)$ is the utility derived by all the nodes from the network and is dependent on total resources shared across the network *i.e.* S_{nwk}. However, if network does not have enough download capacity to utilize shared resources, then extra resources $S_{\text{nwk}} - D_{\text{nwk}}$ get wasted. Therefore,

$$u_{\text{nwk}} = \min\{S_{\text{nwk}}, D_{\text{nwk}}\}. \tag{2.13}$$

We now determine the maximum achievable u_{nwk} during the network operation. Three cases can arise

Case 1. $S_{\text{nwk}} = D_{\text{nwk}}$. From (2.12), the value of $S_{\text{nwk}} = \frac{NC}{2}$. Using (2.13), achieved utility is

$$u_{\text{nwk}} = \frac{NC}{2}. \tag{2.14}$$

Case 2. $S_{\text{nwk}} < D_{\text{nwk}}$. Using (2.12), the value of $S_{\text{nwk}} < \frac{NC}{2}$. Therefore, the achieved utility (refer (2.13)) is

$$u_{\text{nwk}} < \frac{NC}{2}. \tag{2.15}$$

Case 3. $S_{\text{nwk}} > D_{\text{nwk}}$. Using (2.12), S_{nwk} will be greater than $\frac{NC}{2}$ but $D_{\text{nwk}} < \frac{NC}{2}$. From (2.13), the utility is

$$u_{\text{nwk}} < \frac{NC}{2}. \tag{2.16}$$

On comparing (2.14), (2.15) and (2.16), the maximum achievable utility is

$$u_{\text{nwk}}^{\max} = \frac{NC}{2}. \tag{2.17}$$

The variation in the network utility with change in s_{nwk} is also plotted in Fig. 2.3. Initially u_{nwk} is 0. The value of u_{nwk} then starts increasing with value of s_{nwk} and reaches maximum value $N \cdot \frac{C}{2}$ at $s_{\text{nwk}} = N \cdot \frac{C}{2}$. Thereafter, value of u_{nwk} starts decreasing with increase in s_{nwk} because network does not have enough download capacity, to utilize all the shared resources *i.e.*, $d_{\text{nwk}} < N \cdot \frac{C}{2}\ \forall\ s_{\text{nwk}} > N \cdot \frac{C}{2}$. For a *NE* state to be socially optimal, the net utility received by all the nodes should be equal to maximum possible utility $\left(N \cdot \frac{C}{2}\right)$ in the network. We now analyse social optimality of all the *NE* states in capacity partitioning game.

In S_0^*, no node in the network is uploading, so utility received by every node is 0 , *i.e.*, $u_i^0 = 0,\ \forall i \in \mathcal{N}$. Therefore, the total utility $\left(u_{\text{nwk}}^0\right)$ derived across the network for *NE* state S_0^* is given as,

$$u_{\text{nwk}}^0 = \sum_{j \in \mathcal{N}} u_j = 0.$$

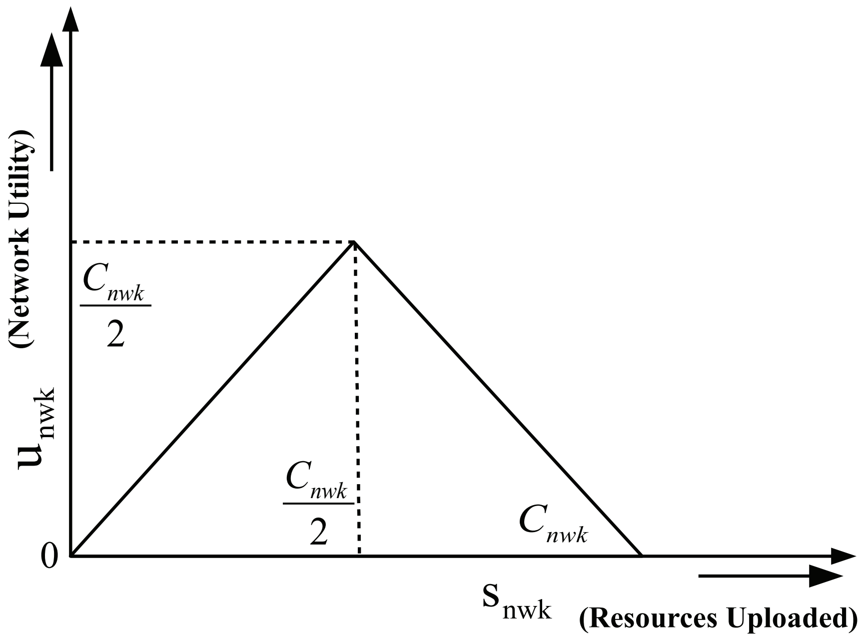

FIGURE 2.3: Aggregate Network Utility.

Clearly S_0^*, is not socially optimal, as utility received in this state is less than maximum possible utility in the network *i.e.* $N \cdot \frac{C}{2}$.

In *NE* state $S^*_{\frac{C}{2}}$, the utility received by every node is (refer (2.10)),

$$u_i^{\frac{C}{2}} = \frac{C}{2}, \ \forall i \in \mathcal{N}. \tag{2.18}$$

Therefore, utility derived by all the nodes in $S^*_{\frac{C}{2}}$ is

$$u_{\text{nwk}}^{\frac{C}{2}} = \sum_{j \in \mathcal{N}} \frac{C}{2} = N \cdot \frac{C}{2}. \tag{2.19}$$

The total utility earned in $S^*_{\frac{C}{2}}$ state is equal to maximum possible utility in the network, *i.e.*, $N \cdot \frac{C}{2}$. Therefore, *NE* state $S^*_{\frac{C}{2}}$ is socially optimal.

2.8 Conclusion

We have established game theoretically that in a in a homogeneous P2P network, where nodes are connected to backbone network through access links of constant capacity, equal partitioning of access link capacity between upload and download, maximizes download at a node. On game theoretically analysis this problem, it was found that equal partitioning strategy is also the Nash Equilibrium (*NE*) of capacity partitioning game. We also proved that this strategy is socially optimal and results in maximum possible utility across the network. To substantiate this claim, a homogeneous P2P network was simulated for different strategies of node. Simulation results demonstrate that node received maximum payoff, when it employs equal partitioning strategy. We also simulated BitTorrent and results indicate that nodes, which adhere to strategy profile during *NE*, receive maximum resources from the network. Thus, equal partitioning strategy, ensures maximum resource utilization both at the node as well as the network level.

Chapter 3

Game Theoretic Analysis of Incentive Parameters for Capacity Partitioning

3.1 Introduction

To motivate users to contribute in a P2P network, Feldman *et al.* [30] suggested that resources should be distributed in proportion to the contribution/upload of resources by the requesters. Contribution based resource allocation leads to fair resource allocation[1] in homogeneous P2P networks (refer chapter 2 for details). However, contribution level is not sufficient to enforce fair contribution in the network containing nodes of different capacities. In the latter part of this chapter we show that in such a network, it is possible for higher capacity nodes to receive greater resources than their contribution level. These extra resources are taken out from the share of genuine low capacity nodes. Therefore, for fair resources distribution an appropriate choice of incentive parameter is required.

In this chapter, we use game theory to show that in a network containing single capacity links, where resources are distributed only on the basis of contribution level, the resource allocation process gets dictated by few higher capacity users. In such scenario, some lower capacity users will receive virtually no resource,

[1]Fair allocation implies that nodes receive resources in proportion to what they contribute

even if they offer maximum possible capacity for upload. Consequently, these low capacity nodes will be forced to leave the network.

An incentive system has already been proposed in [15], where resources are allocated based upon the the ratio of resources contributed to the resources consumed by the requesters. Such allocation leads to fairness but no formal proof exists. We provide game theoretical justification that such type of allocation cannot be adversely influenced by the higher capacity peers.

3.2 System Model

A network $\mathcal{N}$ with $\{1, 2, \cdots, N\}$ users is considered. The access link capacity (C_i) of any user i is divided between upload (s_i) and download (d_i) capacities such that $C_i = s_i + d_i$.

If download capacity (d_i) of node i becomes 0, then it will be unable to receive any resource. We further can safely assume that any node i will never reduce its d_i below a threshold level Δ, such that,

$$d_i \geq \Delta \text{ with } \Delta > 0, \ \forall i \in \mathcal{N}.$$

We consider expected value of resources allocated $\mathbb{E}(r_i)$ for utility calculation because resources allocated (r_i) to any user i fluctuates with time even for the same contribution level. Payoff/utility (u_i) received by any node i is equal to the portion of the allocated resources that can be actually received. Therefore u_i can never be greater than the download capacity of the node i. Thus, utility received by i is given by

$$u_i = \min \{\mathbb{E}[r_i], d_i\}. \tag{3.1}$$

3.3 Contribution Level Based Capacity Partitioning

We game theoretically establish in next subsection, that incentive mechanisms which consider only the contribution level for resource distribution are unfair toward the low capacity users.

3.3.1 Game Formulation

Let $G_1 = [\mathcal{N}, \{\mathcal{S}_i\}, u_i]$ represents the capacity partitioning game, with $\mathcal{N} = \{1, 2, \cdots, N\}$ players/users. A user i, receives utility u_i for playing a strategy $s_i \in S_i$. Strategy is the portion of link capacity (C_i) that i allocates toward upload (s_i). $s_i \in S_i = [0, C_i - \Delta]$, $\forall i \in \mathcal{N}$

3.3.1.1 Nash Equilibrium (*NE*)

A strategy profile ($\mathcal{S}^* = \{s_i^* | s_i^* \in \mathcal{S}_i,\ \forall i \in \mathcal{N}\}$), is a *NE* if no user can increase its utility through unilateral deviation, *i.e.*,

$$u(s_i^*, \mathcal{S}_{-i}) \geq u(s_i, \mathcal{S}_{-i}),\ \forall i \in \mathcal{N}, \tag{3.2}$$

where $\mathcal{S}_{-i} = \{s_1^*, s_2^*, \cdots, s_{i-1}^*, s_{i+1}^*, \cdots, s_N^*\}$ is a set of strategies of all the users except i.

3.3.1.2 Game Analysis

The resources are allocated among the N users in the decreasing order of their upload capacity, such that resource requirement of requester with highest upload capacity is fulfilled first. Let $\hat{1}, \hat{2}, \cdots, \hat{N}$ be the competing users, sorted in decreasing order of their capacity. The *NE* of capacity partitioning game will be following.

Proposition 1. Strategy profile $\hat{\mathcal{S}} = \{s_{\hat{i}} | s_{\hat{i}} \in \mathcal{S}_i,\ \forall \hat{i} \in \hat{\mathcal{N}}\}$, such that

$$\begin{gathered} s_{\hat{1}} = s_{\hat{2}} = \cdots = s_{\hat{l}-1} > C_{\hat{l}} - \Delta \geq C_{\hat{l}+1} - \Delta \cdots \geq C_{\hat{N}} - \Delta,\ \text{for any } \hat{l} \leq N, \\ \text{with, } \sum_{j=0}^{\hat{l}-1} d_j \leq \sum_{j \in \hat{\mathcal{N}}} s_j \text{ and } \sum_{j=0}^{\hat{l}} d_j > \sum_{j \in \hat{\mathcal{N}}} s_j. \end{gathered} \tag{3.3}$$

is a *NE*. $\hat{\mathcal{S}}$ results in $u_i = d_i$, $\forall i \in \{\hat{1}, \hat{2}, \cdots, \hat{l} - 1\}$ and $u_{\hat{l}} = \sum_{j \in \hat{\mathcal{N}}} S_j - \sum_{j=\hat{1}}^{\hat{l}-1} d_j$. However, if there are more than one user having their upload capacity equal to $C_{\hat{l}} - \Delta$, then

$\sum_{j\in\hat{\mathcal{N}}} S_j - \sum_{j=1}^{\hat{l}-1} d_j$, will get equally divided between these users. Remaining users, sharing less than $C_{\hat{l}} - \Delta$ will receive 0 utility from the network.

Proof. Initially a user has no idea about the capacity of the other members in the network. Therefore, users would upload in proportion to their demand and would divide their link capacity equally between upload and download. Hence,

$$s_j = d_j = \frac{C_j}{2},\ \forall j \in \hat{\mathcal{N}},\ \text{with} \sum_{j\in\hat{\mathcal{N}}} d_j = \sum_{j\in\hat{\mathcal{N}}} s_j.$$

After some time, users estimate[2] the access capacity of other members and high capacity users start decreasing their upload in order to maximize their utility from the network. Therefore, $s_i < \frac{C_i}{2}$, for some higher capacity users such that

$$\begin{gathered} s_{\hat{1}} \geq s_{\hat{2}} \geq \cdots \geq s_{\hat{m}-1} > s_{\hat{m}} \geq s_{\hat{m}+1} \cdots \geq s_{\hat{N}},\ \text{for any } \hat{m} \leq N, \\ \text{with} \sum_{j=0}^{\hat{m}-1} d_j \leq \sum_{j\in\hat{\mathcal{N}}} s_j \text{ and } \sum_{j=0}^{\hat{m}} d_j > \sum_{j\in\hat{\mathcal{N}}} s_j. \end{gathered} \tag{3.4}$$

Users with index higher than $\hat{m}\in\hat{\mathcal{N}}$, will hardly receive any resource. Therefore, they will start increasing their upload capacities. However, there would be some low capacity users (from $\hat{l}$ to $\hat{N}$), who would share less than high capacity peers even when they allocate their entire link capacity for upload. Mathematically it can be represented as:

$$\begin{gathered} s_{\hat{1}} \geq s_{\hat{2}} \geq \cdots \geq s_{\hat{l}-1} > C_{\hat{l}} - \Delta \geq C_{\hat{l}+1} - \Delta \cdots \geq C_{\hat{N}} - \Delta,\ \hat{l} \leq N \\ \text{such that} \sum_{j=0}^{\hat{l}-1} d_j \leq \sum_{j\in\hat{\mathcal{N}}} s_j \text{ and } \sum_{j=0}^{\hat{l}} d_j > \sum_{j\in\hat{\mathcal{N}}} s_j. \end{gathered} \tag{3.5}$$

First, $l-1$ users will always receive resources equal to their download capacity. To further increase their utility (described in (3.1)), these users will start reducing

[2] Usually users demand resources in proportion to their link capacity. Therefore, a serving node estimates the capacity of requester from their demand profile.

their upload capacity until:

$$s_{\hat{1}} = s_{\hat{2}} = \cdots = s_{\hat{l}-1} > C_{\hat{l}} - \Delta \geq C_{\hat{l}+1} - \Delta \cdots \geq C_{\hat{N}} - \Delta,\ \hat{l} \leq N$$
$$\text{such that } \sum_{j=0}^{\hat{l}-1} d_j \leq \sum_{j \in \hat{\mathcal{N}}} s_j \text{ and } \sum_{j=0}^{\hat{l}} d_j > \sum_{j \in \hat{\mathcal{N}}} s_j. \tag{3.6}$$

In this network state, the first $l-1$ users in set $\hat{\mathcal{N}}$ will receive desired amount of resources from network, whereas remaining resources, *i.e.*, $\sum_{j \in \hat{\mathcal{N}}} s_j - \sum_{j=\hat{1}}^{\hat{l}-1} d_j$ will be distributed equally among the users with upload capacity $C_{\hat{l}} - \Delta$. Following argument establishes this is a *NE*.

- If a user $i \in \{\hat{1}, \hat{2}, \cdots, \hat{l}-1\}$, decreases its upload capacity then it will receive lesser resources due to decline in its contribution level and may not remain in top $\hat{l}-1$ users as per upload capacity. Similarly, increment in upload capacity reduces the download capacity and thereby the utility u_i, received by the user i. So there is no incentive for the first $l-1$ users in $\hat{\mathcal{N}}$ to deviate.
- If any user from $\hat{l}$ to $\hat{N}$, decreases its upload capacity, then it will loose any chance of getting resource in future, when one of the users among first $\hat{l}-1$ user leave the network. In addition, even on reducing their upload capacities, these users will hardly get any increment in their utility. So there is no incentive for them to deviate.

In this *NE*, low capacity users (say k) with $s_k < C_{\hat{l}} - \Delta$, will be unable to receive any resource despite offering maximum possible resources for upload, *i.e.*, $(C_k - \Delta)$. This unfairness in resource allocation can be explained more through the following numerical example. □

3.3.2 Problem Illustration using Numerical Example

Let us consider a distributed network of 5 users, $\vec{n} = [1, 2, 3, 4, 5]$ with corresponding access link capacities $\vec{c} = [6, 4, 2, 2, 2]$. Further, we assume a threshold download capacity of 0.1 (*i.e.* $\Delta = 0.1$). We deliberately choose these parameter values to illustrate unfairness in resource allocation. Initially, the users will divide their link capacity equally between upload and download, so upload and download capacities allocated by users are $\vec{s} = [3, 2, 1, 1, 1]$ and $\vec{d} = [3, 2, 1, 1, 1]$ respectively. As their is supply demand balance, the resources received by the nodes is $\vec{t} = [3, 2, 1, 1, 1]$. The corresponding utility received by users will be $\vec{u} = [3, 2, 1, 1, 1]$. To increase its utility, user 1 will further decrease its upload capacity such that new capacity distributions becomes $\vec{s} = [2, 2, 1, 1, 1]$ and $\vec{d} = [4, 2, 1, 1, 1]$. As the resources are distributed in the decreasing order of contribution level of requesters, new $\vec{t} = [4, 2, 0.33, 0.33, 0.33]$ (refer (2.2) for details). The utility corresponding to new $\vec{t}$ is $\vec{u} = [4, 2, 0.33, 0.33, 0.33]$ (refer (2.3)). Users with low capacity, *i.e.*, 3, 4 and 5 increase their contribution level (upload) to enhance their utility. This results in network equilibrium profile with upload $\vec{s} = [2, 2, 1.9, 1.9, 1.9]$ and download $\vec{d} = [4, 2, 0.1, 0.1, 0.1]$. Consequently, resources allocated to the nodes with their corresponding utility will be $\vec{t} = [4, 2, 0.1, 0.1, 0.1]$ and $\vec{u} = [4, 2, 0.1, 0.1, 0.1]$ respectively.

Clearly, high capacity user can misuse this resource distribution mechanism to receive far greater resources than their actual contribution. These extra resources are acquired from the share of deserving low capacity peers.

3.4 Capacity Partitioning Based on Resource Contribution and Consumption

Let $G_2 = [\hat{\mathcal{N}}, \{\mathcal{S}_i\}, \{\mathcal{U}_i\}]$ denote the capacity partitioning game where $\hat{\mathcal{N}} = \{1, 2, \cdots, \hat{N}\}$ represents network users, sorted in decreasing order of their link capacities. The capacity shared by player/user i, *i.e.*, $s_i \in \mathcal{S}_i = [0, C_i]$, $\forall i \in \mathcal{N}$ represent i's strategy. C_i is the total access link capacity of i and $u_i \in \mathcal{U}_i$ denotes utility earned by i (refer (3.1) for utility details).

3.4.1 Incentive level

Incentive level (I_i) of node i, considered for resource allocation is defined as ratio of $i's$ upload to download capacity and is given by,

$$I_i = \frac{s_i}{d_i}. \tag{3.7}$$

In this section, we game theoretically establish that for fair resource allocation, which safeguards interest of low capacity users, the resources should be distributed among requesters in decreasing order of their incentive level (I_i). If more than one user have same value of incentive level, then resources are equally distributed among them when the residual resources available at server are less than resources demanded by these users.

The serving user estimates the download capacity of requesters from the amount of resources demanded by them. A user request in proportion to its download capacity because demanding more than download capacity will reduce its incentive level whereas a user's download capacity remains underutilized if it demands less.

3.4.1.1 Nash equilibrium *(NE)*

The *NE* of capacity partitioning game is given by,

Proposition 2. Strategy profile $\mathcal{S}^* = \{s_i^* | s_i^* \in \mathcal{S}_i,\ \forall i \in \hat{\mathcal{N}}\}$, such that

$$s_i^* = d_i^* = \frac{C_i}{2},\ \forall i \in \mathcal{N},$$
$$\text{leading to } \sum_{i \in \hat{\mathcal{N}}} d_i = \sum_{i \in \hat{\mathcal{N}}} s_i, \tag{3.8}$$

where C_i and d_i are the access link and download capacity of node i. The network state $\mathcal{S}^*$ results in $u_i = \frac{C_i}{2},\ \forall i \in \hat{\mathcal{N}}$.

Proof. When users would upload in proportion to their demand they would divide their link capacity equally between upload and download. This distribution results in network attaining $\mathcal{S}^*$ state with,

$$s_j = d_j = \frac{C_j}{2},\ \forall j \in \hat{\mathcal{N}},\ \text{with} \sum_{j \in \hat{\mathcal{N}}} d_j = \sum_{j \in \hat{\mathcal{N}}} s_j \text{ and } I_j = 1\ \forall j \in \hat{\mathcal{N}}.$$

Following argument shows that strategy profile $\mathcal{S}^*$ is a *NE*.

- If any node decreases its upload capacity then its incentive parameter will become less than 1, due to which node will receive lesser resources.
- Conversely, if any node increases its upload capacity, its incentive level will become greater than 1 but its download capacity decreases. Consequently, overall utility received by the node will decrease.

Since, any node i in the network, doe not have incentive to deviate from strategy $s_i = \frac{C_i}{2}$, so strategy profile $\mathcal{S}^*$ is a *NE*. Resources received (s_j) by a node are equal to resources contributed (d_j), such that, node which is downloading more, has to upload more in comparison to the node downloading less to maintain same incentive level. Therefore, allocation based on incentive parameter $I_j = \frac{s_j}{d_j}$, leads to fair resource distribution. □

3.4.2 Problem Illustration using Numerical Example

We again consider a P2P network of 5 users represented by $\vec{n} = [1, 2, 3, 4, 5]$ with access link capacities $\vec{c} = [6, 4, 2, 2, 2]$. We assume a threshold download capacity of 0.1 (*i.e.*, $\Delta = 0.1$). Initially, the nodes will divide their link capacity equally between upload and download, which results in upload and download capacity distribution among the nodes as $\vec{s} = [3, 2, 1, 1, 1]$ and $\vec{d} = [3, 2, 1, 1, 1]$ respectively. Resources will be awarded among the nodes in decreasing order of the incentive level given by $\vec{I} = \{1, 1, 1, 1, 1\}$ (refer (3.7) for $\vec{I}$ calculation). Therefore, utility received by nodes will be $\vec{u} = [3, 2, 1, 1, 1]$. Let node 1 decrease its upload capacity such that new capacity partitioning becomes $\vec{s} = [2, 2, 1, 1, 1]$ and $\vec{d} = [4, 2, 1, 1, 1]$. The incentive level of nodes for the new partitioning arrangement is given by $\vec{I} = [0.5, 1, 1, 1, 1]$, which results in $\vec{u} = [2, 2, 1, 1, 1]$. The lower incentive level of 1 causes decrease in the utility received by 1. Other way around, let node 1 increases its upload capacity such that capacity partitioning across the network becomes $\vec{s} = [4, 2, 1, 1, 1]$ and $\vec{d} = [2, 2, 1, 1, 1]$. This results in $\vec{I} = [2, 1, 1, 1, 1]$. Although, 1 has higher incentive level than the other nodes in the network, its download capacity decreases. Consequently, $\vec{u} = [2, 2, 1, 1, 1]$.

Clearly, higher capacity nodes will be receiving less resources if they do not contribute in proportion to what they consume. Hence, if resources are distributed on the basis of ratio of resources contributed and demanded, high capacity nodes are unable to manipulate the resource distribution mechanism. All the nodes will receive fair amount of resources in proportion to their contribution back to the network.

3.5 Conclusion

In a P2P network with constant access link capacity, the cooperative behavior enforced by incentive mechanisms is analyzed using game theory. It was established that in the resource distribution based on contribution level only, few higher link capacity users consume most of the resources. Consequently, most of lower link capacity users will starve, even if they offer their full link capacity for upload. However, for unbiased resource distribution, resources are allocated among requesters in the decreasing order of their ratio of contribution and consumption of resources. In such resource allocation, every user will receive resources in proportion to their contribution to the network. Therefore, this allocation is better than earlier one in enforcing cooperation among users.

Chapter 4

Optimal Capacity Partitioning in Heterogeneous P2P Network

4.1 Introduction

In this chapter, we extend the game theoretic model for capacity partitioning in homogeneous networks (proposed in chapter 2) to heterogeneous networks, *i.e.*, we remove the limitation of all the nodes having the same access link capacity. For heterogeneous network analysis, resource distribution on the basis of cooperation level is not sufficient to ensure fair resource distribution (refer Chapter 3). Therefore to model capacity partitioning in heterogeneous network, we consider ratio of the contribution and the consumption of the resources by the requester for resource allocation. The brief details about the need for capacity partitioning in network containing single capacity links is discussed below.

In networks like WiFi, WLAN, LTE and WiMAX (in time division duplex (TDD) mode) [15, 66, 67] the uplink and downlink traffic flow through common access link, before entering the backbone network via a local hub. The access link of every node has an overall finite capacity which gets divided between upload and download, so that increasing upload decreases download and vice-versa. The capacity division [68] of the access link between uplink and downlink can be altered by the node/member. To maximize the resources received, node would like to allocate its entire link capacity for the download but the incentive mechanism will force it to maintain a certain level of contribution (*i.e.*, sharing/upload). Therefore, node seeks an optimal partitioning of link capacity where it maintains a minimal level of upload to manage just enough resources from network, to completely utilize its download capacity. In this chapter, we model capacity partitioning as a game and determine its Nash equilibrium (*NE*).

The Game theory has been already employed in [69] to model the capacity partitioning. However, this effort considers a restricted scenario where members are allowed only single upload and download at a time. This limits the maximum download rate achieved by any node, to half of the capacity of the slowest access link in the network, *i.e.*,

$$d_{\max} = \min_{i \in \mathcal{N}} \frac{C_i}{2} \; \forall i \in \mathcal{N}, \tag{4.1}$$

where C_i represents the link capacity of i^{th} node, $\mathcal{N}$ is a set containing all the nodes in the network and $d_{\max}$ is the maximum download rate achieved by any node in the network. To eliminate the bottleneck of the slowest link, we consider a more generalized model here, where nodes can simultaneously perform multiple uploads and downloads, enabling them to receive data rate in proportion to their contribution levels. Other works on capacity partitioning in [15, 71], dealt with the design of partitioning algorithm to achieve optimal partitioning, without exploring any optimal state of capacity partitioning in the network. We have found this optimal state to be the *NE* in capacity partitioning game.

Using game theory, we also show that *NE* state derived for the current game results in efficient distribution of resources across the network. Therefore *NE*, obtained for capacity partitioning process in heterogeneous network, simultaneously maximizes individual as well as social welfare. Theoretically, the partitioning of link capacity achieved by any algorithm should converge to the optimal partitioning point, *i.e.*, *NE*. In practical scenarios, output of partitioning algorithms will deviate from the *NE*. Therefore *NE* can be used as a benchmark by computing the deviation. Greater is the deviation from the *NE*, poorer is the performance of the algorithm.

4.2 System Model

A network $\mathcal{N}$ with $\{1, 2, \cdots, N\}$ nodes is considered. The access link capacity (C_i) of any node i is divided between upload (s_i) and download (d_i) capacities such that $C_i = s_i + d_i$. In line with existing models [15, 69], we assume that the nodes employ resource discovery algorithms like distributed hash table (DHT)[17], flooding etc., to find resource providers across the network. Thereafter, the nodes directly communicate with each other for resource transfer.

Resources received (r_i) by any node i changes with time even for the same contribution level. These fluctuations in r_i arise due to changing network conditions like change in the data traffic in the backbone network, variation in number of resource requests received by a serving member etc. Therefore, we consider expected value of resources received $\mathbb{E}(r_i)$ for utility calculation. Payoff or utility (u_i) received by any node i is equal to the portion of the received resources that can be utilized. In some cases, i may lack the download capacity (d_i) to use all the allocated resources, *i.e.*, $\mathbb{E}[r_i] > d_i$. The node can only utilize resources equal to d_i and leftover resources would be wasted. Thus, utility received by i is

$$u_i = \min \{\mathbb{E}[r_i], d_i\}. \tag{4.2}$$

In the subsequent section, we model capacity partitioning in heterogeneous network as game and calculate its Nash equilibrium.

4.3 Capacity Partitioning Based on Contribution and Resource Consumption

4.3.1 Modeling of Capacity Partitioning Game

Let $G_2 = [\mathcal{N}, \{\mathcal{S}_i\}, u_i]$ denote the capacity partitioning game where $\mathcal{N}$ represents network nodes. The capacity shared by player/node i, *i.e.*, $s_i \in \mathcal{S}_i = [0, C_i]$, $\forall i \in \mathcal{N}$ represent i's strategy. C_i is the total access link capacity of i and u_i denotes utility earned by i.

$$u_i(s_i, \mathcal{S}_{-i}) = \min\{\mathbb{E}[r_i], d_i\}, \tag{4.3}$$

where $\mathcal{S}_{-i} = \{s_1, s_2, \cdots, s_{i-1}, s_{i+1}, \cdots, s_N\}$ represents strategies of all the nodes except i. Utility will also be referred as the *usable bandwidth* received by the node. To simplify Nash equilibrium analysis, we consider download threshold, $\Delta = 0$ (refer section 3.2 for details about Δ), as it will not affect final results.

4.3.2 Game Analysis

Every node is assigned an incentive level. The incentive level of node i is the ratio of i's upload and download capacity and is given by,

$$I_i = \frac{s_i}{d_i}. \tag{4.4}$$

As discussed in chapter 3, the serving node estimates the download capacity of requesters from their request profile. The resources are allocated among the requesters in the decreasing order of their incentive parameter, such that the resource requirement of requester with highest incentive parameter is fulfilled first.

A strategy profile ($\mathcal{S}^* = \{s_j^* | s_j^* \in \mathcal{S}_j,\ \forall j \in \mathcal{N}\}$) is a *NE*, if $u(s_j^*, \mathcal{S}_{-j}) \geq u(s_j, \mathcal{S}_{-j}),\ j \in \mathcal{N}$, where $\mathcal{S}_{-j} = \{s_k^* | s_k^* \in \mathcal{S}_k,\ \forall k \in \mathcal{N} \backslash \{j\}\}$. We claim the following about *NE*.

Theorem 4.1. *Two NE's are possible in the game.*

1. *No node in the network contribute towards upload, i.e.,* $\mathcal{S}_1^* = \left\{s_j | s_j = 0, \forall j \in \mathcal{N}\right\}$.
2. *The link capacity is equally divided between upload and download, i.e.,* $\mathcal{S}_2^* = \left\{s_j | s_j = \frac{C_j}{2}, \forall j \in \mathcal{N}\right\}$.

Proof. Let S_{nwk} and D_{nwk} denote total upload and download capacities across the network, *i.e.*, $S_{\text{nwk}} = \sum\limits_{j \in \mathcal{N}} s_j$ and $D_{\text{nwk}} = \sum\limits_{j \in \mathcal{N}} d_j$. Based upon S_{nwk} and D_{nwk} comparison, three cases arise.

1. $\boldsymbol{S_{\text{nwk}} < D_{\text{nwk}}}$ **:** Two sub cases arise.

(a) "$\mathcal{S}_1^* = \left\{s_j | s_j = 0, \forall j \in \mathcal{N}\right\}$" **is a NE.**
Since no node uploads, the resources available across the network is 0. Therefore,

$$u_i(s_i, \mathcal{S}_{-i}) = 0, \ \forall i \in \mathcal{N}.$$

Even if any node unilaterally deviates and starts uploading, it will be unable to receive any resource as the remaining members in network are not uploading. As node has no incentive to deviate, so $\mathcal{S}_1^*$ is a NE.

(b) "$S_{\text{nwk}} \neq 0$" **is not a NE.**
Due to shortage of shared resources ($S_{\text{nwk}} < D_{nwk}$), at least one node in network will be receiving resources less than its download capacity. This node can increase its utility by increasing its upload capacity. Hence, current network state cannot be *NE*.

2. $\boldsymbol{S_{\text{nwk}} = D_{\text{nwk}}}$ **:** Here, two sub cases arise.

(a) "$\mathcal{S}_2^* = \left\{s_j | s_j = \frac{C_j}{2}, \forall j \in \mathcal{N}\right\}$" **is a NE.**
To determine the stability of $\mathcal{S}_2^*$, we calculate the variation in the received utility when a node unilaterally changes its strategy. To simplify the analysis we divide node *i's* region of operation into 3 regions as shown below.

$$U_i(S_i, \mathcal{S}_{-i}) = \begin{cases} U_i^1 & \text{when } S_i = \frac{C_i}{2}, \\ U_i^2 & \text{when } S_i > \frac{C_i}{2}, \\ U_i^3 & \text{when } S_i < \frac{C_i}{2}. \end{cases}$$

The strategy profile of remaining players remain fixed and is given by $\frac{C}{2}_{-i} = \{s_j | s_j = \frac{C_j}{2}, \ \forall j \in \mathcal{N} \setminus \{i\}\}$. The u_i is calculated as follows.

i. u_i^1 *Calculation:* As every node has divided its link capacity equally between upload (s_j) and download (s_j), following network conditions arise:

A. $s_j = d_j = \frac{C_j}{2}, \forall j \in \mathcal{N}$.

B. $S_{\text{nwk}} = D_{\text{nwk}} = \sum_{j \in \mathcal{N}} \frac{C_j}{2}$.

C. $I_j = 1, \forall j \in \mathcal{N}$.

Every node gets equal priority during resource distribution, because they have the same incentive level. In addition, as there is supply demand balance, *i.e.*, $S_{\text{nwk}} = D_{\text{nwk}}$, so expected received resources by every node will be equal to its download capacity, *i.e.*, $\mathbb{E}[r_j] = d_j,\ \forall j \in \mathcal{N}$. Using (4.3) utility obtained by i is given as

$$u_i^1 = u_i\left(\frac{C_i}{2}, \frac{C_i}{2}_{-1}\right) = \frac{C_i}{2}. \tag{4.5}$$

ii. u_i^2 *Calculation:* Capacity distribution and incentive level of nodes across network, when i plays strategy $s_i > \frac{C_i}{2}$ are :

A. $s_i = \frac{C_i}{2} + \Delta_2$ and $d_i = \frac{C_i}{2} - \Delta_2, \Delta_2 > 0$

B. $s_j = d_j = \frac{C_j}{2}, \forall j \in \mathcal{N} \backslash \{i\}$

C. $S_{\text{nwk}} = \sum_{j \in \mathcal{N}} \frac{C_j}{2} + \Delta_2$ and $D_{\text{nwk}} = \sum_{j \in \mathcal{N}} \frac{C_j}{2} - \Delta_2$.

D. $I_i > 1$ and $I_j = 1, \forall j \in \mathcal{N} \backslash \{i\}$.

Due to highest incentive level, i will get priority over other nodes during resource allocation. The expected resources received by i will be $\mathbb{E}(r_i) = \frac{C_i}{2} + \Delta_2$. However, i does not have enough download capacity ($d_i = \frac{C_i}{2} - \Delta_2$) to utilize these resources. Hence, using (4.3), i's utility is given as

$$u_i^2 = u_i\left(\frac{C_i}{2} + \Delta_2, \frac{C_i}{2}_{-1}\right) = \frac{C_i}{2} - \Delta_2. \tag{4.6}$$

Comparing (4.5) and (4.6) we get $u_i^2 < u_i^1$. Therefore, $s_i = \frac{C_i}{2}$ strictly dominates all the strategies where $s_i > \frac{C_i}{2}$. Node i will never play strictly dominated strategies during equilibrium, because they provide lower payoff.

iii. u_i^3 *Calculation:* Network conditions when node i employs strategy $s_i < \frac{C}{2}$, are given as:

A. $s_i = \frac{C_i}{2} - \Delta_3$ and $d_i = \frac{C_i}{2} + \Delta_3, \Delta_3 > 0$

B. $s_j = d_j = \frac{C_j}{2}, \forall j \in \mathcal{N} \backslash \{i\}$

C. $S_{\text{nwk}} = \sum_{j \in \mathcal{N}} \frac{C_j}{2} - \Delta_3$ and $D_{\text{nwk}} = \sum_{j \in \mathcal{N}} \frac{C_j}{2} + \Delta_3$.

D. $I_i < 1$ and $I_j = 1, \forall j \in \mathcal{N} \backslash \{i\}$.

As $I_i < I_j, \forall j \in \mathcal{N} \backslash \{i\}$, therefore node i is given lowest priority during resource allocation. The resources consumed by nodes with priority higher than i are given by,

$$D_{nwk-\{i\}} = \min \left\{ S_{\text{nwk}}, \sum_{j \in \mathcal{N} \backslash \{i\}} d_j \right\} = \sum_{j \in \mathcal{N} \backslash \{i\}} \frac{C_j}{2}. \tag{4.7}$$

Residual resources available to i for download are,

$$\mathbb{E}[r_i] = S_{\text{nwk}} - D_{\text{nwk}-\{\text{i}\}} = \frac{C_i}{2} - \Delta_3. \tag{4.8}$$

Hence utility received by i (refer (4.3)) is

$$u_i^3 = u_i \left(\frac{C_i}{2} - \Delta_3, \frac{C_i}{2}_{-1} \right) = \frac{C_i}{2} - \Delta_3. \tag{4.9}$$

Comparing (4.5) and (4.9) we get $u_i^3 < u_i^1$. As the strategy $s_i = \frac{C_i}{2}$, strictly dominates all the strategies with $s_i < \frac{C_i}{2}$, so i has no incentive to deviate from strategy $s_i = \frac{C_i}{2}$. Therefore, $\mathcal{S}_2^* = \left\{ s_j | s_j = \frac{C_j}{2}, \forall j \in \mathcal{N} \right\}$ is indeed a NE.

(b) "$s_i \neq d_i, \exists i \in \mathcal{N}$" **is not a NE.**

In this case, network will contain at least one node j whose $s_j < d_j$ and a node i with $s_i > d_i$ to maintain $S_{\text{nwk}} = D_{\text{nwk}}$. As $S_{\text{nwk}} = D_{\text{nwk}}$, the resource requirement of almost every node is satisfied. Nodes like i, with higher incentive level can further increase their utility by incrementing their download capacity. These additional resources are drawn from the share of resources allocated to node with lowest incentive level.

3. $\boldsymbol{S_{\text{nwk}} > D_{\text{nwk}}}$ **:** In this case, "$S_{\text{nwk}} - D_{\text{nwk}}$" amount of resources remain unutilized after fulfilling the download requirements of every node. Any node can receive these unutilized resources by increasing its download capacity. As nodes have incentive to change their strategy, so "$S_{\text{nwk}} > D_{\text{nwk}}$" is not a *NE*.

□

Link Capacity (*Mbps*)	**Upload Capacity** (*Mbps*)	**Download Capacity** (*Mbps*)
4	2.0	2.0
5	2.5	2.5
6	3.0	3.0
7	3.5	3.5
8	4.0	4.0

TABLE 4.1: Nash Equilibrium Partitioning of Access Capacities

Hence, in capacity partitioning game there exists two *NE*'s, namely $\mathcal{S}_1^*$ and $\mathcal{S}_2^*$. In $\mathcal{S}_1^*$ state, as every node receives 0 utility, so nodes will start leaving the network because their motive of downloading resources remains unfulfilled. Therefore, in practice, network will operate in $\mathcal{S}_2^*$ state, where the nodes have to upload (s_i) in proportion to their consumption (d_i) of resources, *i.e.*, $s_i = d_i = \frac{C_i}{2}$, $\forall i \in \mathcal{N}$. The *NE* distribution of capacities across the network is further illustrated through the numerical example presented in next subsection.

4.3.3 Nash equilibrium (NE) Illustration using Numerical Example

We consider a network of 100 users. These users are distributed in equal proportion into 5 groups of different link capacities viz. 4, 5, 6, 7 and 8 Mb/s, such that each group consists of 20 nodes. *NE* state is given by $\mathcal{S}_2^* = \left\{s_j | s_j = \frac{C_j}{2}, \forall j \in \mathcal{N}\right\}$". Therefore, the link capacity will get equally divided between upload and download capacity during *NE*. The link capacity division, across the network during *NE*, is presented in Table 4.1. The numerical example is presented for understating partitioning under *NE*. To further substantiate authors claim, the network simulation results will be discussed in section 5.7.

4.4 Socially Optimal Partitioning Strategies

Nash equilibrium (*NE*) only takes into account users' individual benefit. It may not be always socially optimum, *i.e.*, the resources available across the network might get wasted due to mismatch between demand and the availability of resources. To evaluate the efficiency of a given strategy profile, we compare the sum of utilities achieved by all the users while implementing that profile with the maximum achievable utility in the network.

Let $S_{\text{nwk}} \left(= \sum_{i\in\mathcal{N}} s_i\right)$ and $D_{\text{nwk}} \left(= \sum_{i\in\mathcal{N}} d_i\right)$ denote the total resources available and demanded respectively in the network. The total access link capacity available across network is $C_{\text{nwk}} \left(= \sum_{i\in\mathcal{N}} C_i\right)$, where C_i is the node i's link capacity . Sum of upload and download is equal to sum of link capacities of all the nodes in the network. Therefore,

$$S_{\text{nwk}} = C_{\text{nwk}} - D_{\text{nwk}} = \sum_{i\in\mathcal{N}} C_i - D_{\text{nwk}}. \tag{4.10}$$

$u_{\text{nwk}} \left(= \sum_{i\in\mathcal{N}} u_i\right)$ is the utility derived by all the nodes from the network and is dependent on total resources shared across the network *i.e.* S_{nwk}. However, if network does not have enough download capacity to utilize shared resources, then extra resources $S_{\text{nwk}} - D_{\text{nwk}}$ get wasted. Therefore,

$$u_{\text{nwk}} = \min\left\{S_{\text{nwk}}, D_{\text{nwk}}\right\}. \tag{4.11}$$

We now determine the maximum achievable u_{nwk} during the network operation. Three cases arise,

Case 1. $S_{\text{nwk}} = D_{\text{nwk}}$. From (4.10), the value of $S_{\text{nwk}} = \frac{\sum_{i\in\mathcal{N}} C_i}{2}$. Using (4.11), achieved utility is

$$u_{\text{nwk}} = \frac{\sum_{i\in\mathcal{N}} C_i}{2}. \tag{4.12}$$

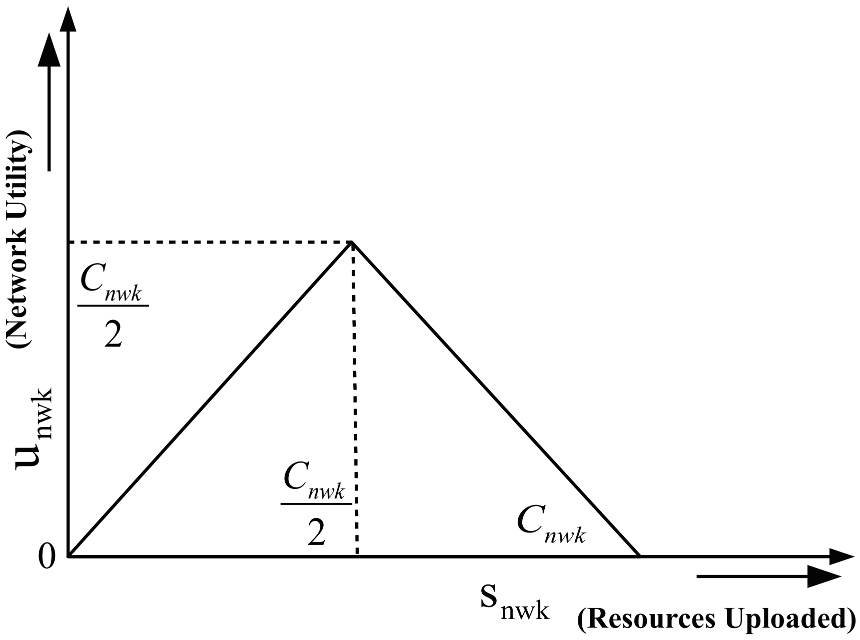

FIGURE 4.1: Aggregate Network Utility.

Case 2. $S_{\text{nwk}} < D_{\text{nwk}}$. Using (4.10), the value of $S_{\text{nwk}} < \frac{\sum_{i \in \mathcal{N}} C_i}{2}$. Therefore, the achieved utility (refer (4.11)) is

$$u_{\text{nwk}} < \frac{\sum_{i \in \mathcal{N}} C_i}{2}. \tag{4.13}$$

Case 3. $S_{\text{nwk}} > D_{\text{nwk}}$. Using (4.10), S_{nwk} will be greater than $\frac{\sum_{i \in \mathcal{N}} C_i}{2}$ but $D_{\text{nwk}} < \frac{\sum_{i \in \mathcal{N}} C_i}{2}$. From (4.11), the utility is

$$u_{\text{nwk}} < \frac{\sum_{i \in \mathcal{N}} C_i}{2}. \tag{4.14}$$

On comparing (4.12), (4.13) and (4.14), the maximum achievable utility is

$$u_{\text{nwk}}^{\max} = \frac{\sum_{i \in \mathcal{N}} C_i}{2} = \frac{C_{\text{nwk}}}{2}. \tag{4.15}$$

The variation in the network utility with change in s_{nwk} is also plotted in Fig. 4.1. Initially u_{nwk} is 0. The value of u_{nwk} then starts increasing with value of s_{nwk} and reaches maximum value $\frac{C_{\text{nwk}}}{2}$ at $s_{\text{nwk}} = \frac{C_{\text{nwk}}}{2}$. Thereafter, value of u_{nwk} starts decreasing with increase in s_{nwk} because network does not have enough download capacity to utilize all the shared resources, *i.e.*, $d_{\text{nwk}} < \frac{C_{\text{nwk}}}{2}\ \forall\ s_{\text{nwk}} > \frac{C_{\text{nwk}}}{2}$. For a

NE state to be socially optimal, the net utility received by all the nodes should be equal to maximum possible utility $\left(\frac{C_{\text{nwk}}}{2}\right)$ in the network. We now analyse social optimality of all the *NE* states in capacity partitioning game.

In S_1^*, no node in the network is uploading, so utility received by every node is 0 , *i.e.*, $u_i^1 = 0, \ \forall i \in \mathcal{N}$. Therefore, the total utility $\left(u_{\text{nwk}}^1\right)$ derived across the network for *NE* state S_1^* is given as,

$$u_{\text{nwk}}^1 = \sum_{i \in \mathcal{N}} u_i^1 = 0.$$

Clearly S_1^* is not socially optimal, as utility received in this state is less than maximum possible utility in the network *i.e.* $\frac{C_{\text{nwk}}}{2}$.

In *NE* state S_2^*, the utility received by every node is (refer (4.5)),

$$u_i^2 = \frac{C_i}{2}, \ \forall i \in \mathcal{N}. \tag{4.16}$$

Therefore, utility derived by all the nodes in S_2^* is

$$u_{\text{nwk}}^2 = \sum_{i \in \mathcal{N}} u_i^2 = \frac{1}{2} \cdot \sum_{i \in \mathcal{N}} C_i = \frac{C_{\text{nwk}}}{2}. \tag{4.17}$$

The total utility earned in S_2^* state is equal to maximum possible utility in the network. Therefore, *NE* state S_2^* is socially optimal.

4.5 Conclusion

In this chapter, we have successfully extended game theoretic model for capacity partitioning to heterogeneous network. The resources are distributed in the decreasing order of the ratio of contribution and consumption of the resources by the requesters. We proved that for such resource allocation, the strategy of equal division of link capacity between uplink and downlink is the Nash equilibrium of capacity partitioning game. This Nash equilibrium maximizes the utility received by a user. Finally, we established that equal partitioning strategy is also socially optimal. Hence, such strategy ensures individual as well as social welfare.

Chapter 5

Adaptive Capacity Allocation in P2P Networks: A Control Theoretic Perspective

5.1 Introduction

The strength of a P2P system lies in the cooperative behavior of its members. However, members are usually reluctant to share resources because there are inherent costs associated with sharing, e.g., financial cost involved in usage of bandwidth for sharing. Therefore, they tend to become free-riders, *i.e.*, they consume resources without contributing back to the network. Many solutions such as micro payment based schemes [13], game theoretic based approaches [30][64] and trust or reputation based approaches [15][54] have been proposed in the past to provide incentives to the cooperating members, thereby forcing all nodes to cooperate. Trust or reputation based system, being simpler and easier to implement, have been used in this chapter to estimate the cooperation level of different members in the form of their reputation. The members are given preference in service according to their reputation. The terms member, node and peer are used interchangeably within this chapter.

In networks like WiFi, WLAN, LTE and WiMAX (in time division duplex (TDD) mode), [15, 66, 67], the nodes are connected to the backbone network through a common access link, as shown in Fig. 1.6. The capacity division of access link between uplink and downlink can be modified by the node. The nodes being selfish, want to use their entire capacity for download but a reputation system forces them to upload as well. Therefore, a mechanism which optimally divides the total capacity at node i between upload and download is needed, such that the capacity allocated for upload should be just enough to maintain node i's reputation close to a minimum level, such that all of i's download requirements are fulfilled.

Reputation-Based Resource Allocation Policy (RRA) [15] strives to achieve optimal partitioning by updating the upload bandwidth in fixed step sizes. Under stable condition, the upload bandwidth oscillates around the optimum due to fixed sized step. We can allow upload bandwidth to change by any value, so that with suitable control mechanism, an optimal operation without oscillatory behavior can be achieved.

In this chapter, we model total capacity distribution between uplink and downlink as a feedback control problem. In this control system, resources that a peer receives from the network act as a feedback, which decides its output, *i.e.*, the resources that the peer uploads back to the network. The control system seeks to take the system to an optimal partitioning level where the bandwidth or capacity received is equal to download capacity available for the current level of upload. Unlike existing schemes, we employ a PI controller to make size of capacity increment or decrement adaptive such that the step size tends to 0 as node's sharing level approaches the optimal value. Thus, total capacity partitioning stabilizes around the optimal point and the resource wastage is reduced leading to enhanced efficiency compared to existing schemes.

Main highlights of this chapter are listed below.

1. We model the resource allocation mechanism as control system, which can be implemented in distributed fashion across the network. This design is in line with basic structure of the P2P network, which lacks any central authority for coordination among members. We also derive a generic transfer function,

which can be used to study various resource allocation algorithms in a control theoretic framework.

2. We also propose 'level of optimality'(U), a metric to measure the optimality level at a node. A node operates at the optimal level when it receives desired amount of resources with minimal sharing (i.e expense). This definition corresponds to selfish behavior of members where they seek to derive maximum benefit at the minimum expense [69].

3. Proportional Integral (PI) controller (discussed later in section 5.5.2.1) has been employed for devising a robust capacity partitioning process. The control system based capacity partitioning is adaptive to network dynamics, caused by nodes entering or leaving the network or some of them starting to free ride. In addition, the proposed control system can be easily integrated with the existing nodes in the network.

4. Unlike earlier works, the proposed system is more efficient in reducing resource wastage by making the nodes operate at optimal capacity distribution. This makes it more difficult for free-riders to get resources as optimal operation ensures minimum resource wastage, which results in higher penalty on the free-riders. Free-riding nodes who could have survived on such unused resource do not get anything from the network, which compels them to cooperate.

Notation: A symbol written in calligraphic font represents a set (e.g. $\mathcal{A}$). $\mathbb{R}^+$ represents the interval $[0, \infty)$. Other notations used in this chapter are mentioned in Table 5.1.

5.2 Related Work

In single capacity links, the total capacity available at the node should be optimally divided between upload and download capacities, so that node can get maximum resources from the network at minimum sharing. Meo *et al.* [69] contemplated network as market and proved that if peers are greedy and employ second price auction then network reaches an equilibrium stage, where the download capacity converges to a specific value. For a network with homogeneous single capacity links, the download capacity converges to half of the total capacity. In the model proposed by Meo *et al.* [69] , it was proved that any peer in the network cannot maintain a download rate higher than $\min_{i \in \mathcal{N}} \frac{C_i}{2}$, where $\mathcal{N}$ corresponds to set of all the member nodes and C_i is the total capacity of the node i. Therefore the slowest link present in the network acts as bottleneck. Another limitation of this model is that a member node can perform only a single upload and download at a given time. Limitations in [69] were alleviated by Reputation based Resource Allocation (RRA) mechanism [15]. RRA dynamically divides the total capacity between upload and download in the single capacity link. RRA strives to attain optimal point of resource sharing by modifying a peer's upload capacities in fixed step sizes. Due to fixed step size, total capacity distribution never settles down and keeps on oscillating around the optimal point. Iosifidis and Koutsopoulos [70] have modeled total capacity partitioning between uplink and downlink as a utility maximization problem. However, authors did not provide any formal framework to utilize the complete link capacity available at the node.

Once the total capacity partition between upload and download is decided, we need a mechanism to award the resources to nodes according to their cooperation level, so that they do not free-ride. Ma *et al.* [64] put forward a modified version of the progressive water filling algorithm where nodes are provided resources at the rates proportional to their reputation. The authors theoretically proved that such kind of allocation maximizes the marginal utility. Yan *et al.* [65] proposed a mechanism where peers' ranking and the utility were the basis for the resource allocation such that resource distribution achieves max-min fairness.

Co-operation level needs to be estimated at every node in the network so that resources can be awarded to requesting nodes based on their contribution. Various works have suggested an array of techniques to determine the cooperation level. Satsiou and Tassiulas in [15] estimated cooperative behavior from the ratio of the resources received and demanded by the node. Banerjee *et al.* [72] proposed that serving peer will compute the expected utility function of the requesting nodes, which will be the basis for the resource allocation.

BitTorent [1] was designed for links (eg. ADSL), where there is strict segregation between upload and download capacities. Hence, it does not provide any mechanism for capacity partitioning. However, BitTorent employs *tit for tat* [73] for resource sharing where service provided by a node i to a node j solely depends on the service j has provided to i in the past, independent of j's behavior with other members. Current incentive mechanism used by BitTorent does not motivate peers to stay in the system, once their download requirement gets fulfilled.

Contemporary P2P networks have large population size with high churn rate [74] and infrequent repeat transactions. In such kind of networks, reputation based systems are very effective. We use a modified version of RRA's reputation system for evaluating reputation. Instead of storing last 10 transaction values, the proposed model uses exponential moving average between present and past values to reduce the memory usage at a node.

5.3 P2P Network Model

We consider a P2P network where files are the resources being shared and consumed by its members. Bandwidth is required to transfer files between various members across the network. Members are reluctant to share bandwidth as there is cost involved in sharing. Therefore, to motivate members to share, they are provided incentive based upon amount of bandwidth shared by them. Hence, in this chapter, we analyze and compare various resource allocation strategies in terms of bandwidth shared by its members. Considering bandwidth as a commodity for consumption and sharing, the P2P network model can be described as follows.

5.3.1 Resource Discovery and Connection Setup

Each peer in the network is connected to a backbone network via an access link (refer Fig. 1.6). In a P2P network, there can be two mechanisms possible for resource discovery. In the first mechanism, distribute hash table (DHT) [17] based overlay is created and in the second mechanism, unstructured neighborhood is

Notation	**Description**
R_i	Reputation of node i.
t_{ji}	Trust of i measured by j.
T_{ji}	Bandwidth received by j from i.
a_{ji}	Resource allocated by i to j
B_{ji}	Bandwidth demanded by j from i.
d_i	The current download capacity of the node i.
s_i	The current shared/upload capacity of the node i.
C_i	The access link capacity at the node i.
T_i	Total resources node i is currently receiving from network.
l_i	No. of requests catered by the node i.
g_i	No. of requests generated by the node i.
B_{ji}^{fes}	The minimum bandwidth required to support feasible data rate possible on the link between i and j.
B_{min}^{fes}	The minimum value of the B_{ji}^{fes}, among all the possible values of i and j in the network.
$k_{ij_{ovd}}$	Proportionality constant for handling change in bandwidth received by i due to overloading at j.
$\mathcal{A}_i$	Set of serving nodes for the node i.
$\mathcal{Z}_i$	Set of nodes that request resources from node i since the last reputation estimation.
R_{min}	Reputation threshold below which a node does not receive any service from network
R_{in}^{max}	Maximum initial reputation that can be assigned to newcomers
R_{in}	Initial reputation assigned to newcomers.
U_i	Level of optimality at node i.
U_{ref}	Reference level, signifying optimal partitioning.

Table 5.1: Important Notations

maintained. Any query for resource is forwarded in an organized fashion in the structured DHT based P2P system while in unstructured network, the query is forwarded to all neighbors except from the one from which query is received. The response of query once found, traverses the same path in reverse, as taken by the query.

As in most of existing P2P resource allocation models [15] [58][69], problem of the resource allocation has been isolated from query propagation and resources discovery across the network. It is assumed that resource lookup algorithm employed by P2P application is ideal, such that every peer has complete knowledge of the nodes who hold the required resource. This assumption is in line with practical P2P system like BitTorrent where members in swarm[1] know about the chunks available with the other peers through tracker[1]. Various messages for link setup and chunk exchange are assumed to be taken care by the P2P application. Once the resource location is received in response, the node directly connects with the peer having the resource. In this sense, for resource provisioning P2P network is fully connected mesh.

5.3.2 Chunk Exchange

Large size resources can be assumed to be consisting of smaller chunks, which are again uniformly distributed across the network. These assumptions are reasonable and close to real life P2P networks,due to following reasons.

1. In popular file sharing networks like BitTorrent, a single file is further divided into large number of chunks [75]. A peer is usually interested in downloading more than one file. Therefore, it needs to download very large number of chunks from the network. Peers first download those chunks which are rare in the swarm. This technique called 'rarest first' [1] ensures that each peer usually has data chunks which the other downloaders in the swarm desire. A typical swarm consists of small number of peers (usually 100) with a requester requiring large number of chunks and other peers containing different chunks due to rarest first technique. All these factors in

[1]The set of peers actively uploading and downloading data [1].

conjunction lead to a situation where for most of the network life time, a peer will always have something of interest for other peers across the swarm.

2. Although in this chapter a user is interested in a file as a resource, there can be P2P system, where members may also be interested in other kind of resources like CPU cycles or storage space. Such type of resources are not specific like a data file pertaining to a particular movie. A user watches movie based upon its interest and mood and therefore, it will not randomly download any movie file. However, there is no interest involved in generic resources of type CPU cycle or storage space. Therefore in such scenario, a requester can request any member of P2P network for the resource.

Hence, similar to P2P network models in extant literature for resource allocation [15, 58, 69], it is assumed that at least one of the many chunks that the requester is interested in, can be found at any node except the requester in the network.

5.3.3 Query Generation Profile

While designing the proposed model we have considered operational overloading [15][69], *i.e.*, there is always a pending request to download data and corresponding content is always available for download. This signifies worst possible operating condition where infinite number of chunks of the same file are spread across the network. If a system can withstand such extreme condition, then it will work efficiently for normal operating conditions too.

5.3.4 Access Link Type

This chapter deals with the links where the total capacity[2] available with a user is constant but there is no strict separation between uplink and downlink capacities. A user can adjust between upstream and downstream flows using certain implementation tools [68]. Such kind of links exist in WiFi, WLAN etc. In line

[2] Total capacity is sum of uplink and downlink capacities.

with the nomenclature in the existing literature [15], we will refer to such type of links as single capacity links.

Based upon above discussed system model specifications, we study the resource allocation process across the P2P network in the next section.

5.4 Resource Allocation across P2P Network

Rational peers will adopt a strategy to derive maximum benefit from the network with minimum cost [69]. This is the basic principle driving the resource allocation. A peer has to pay the cost for both upload and download bandwidth usage. It is assumed that the cost incurred in download is less than the utility gained by downloading the resource from network. Hence, a peer should be concerned about its upload bandwidth usage cost only. To minimize this cost, it will avoid sharing or contributing resources. However, the incentive techniques e.g., reputation system forces peers to upload, by distributing the resources among requesters in proportion to their contribution level. This ensures that the cost incurred while sharing a resource gets transferred to gain in the download. In such cases, peers minimize the cost by sharing just enough so that all of its download requirements are met. The proposed resource allocation system takes care of all of these constraints. We expect this resource allocation system to ensure the optimal operation of P2P system while acting as deterrent to free-riding. A resource allocation system can be divided into three components as described below.

1. A mechanism to achieve optimal distribution of total capacity between upload and download, such that a node is able to receive maximum download while maintaining minimum upload.

2. A mechanism to distribute the upload capacity allocated by the first component, among the requesting nodes based upon requesters' contribution level.

3. Finally, a reputation system to estimate the contribution level of various nodes in the network.

Our main focus is to design a control system which can achieve optimal distribution of total capacity between upload and download bandwidth. To derive transfer function for this control system, we require a mathematical framework of reputation based resource allocation process among requesters. Therefore, we first discuss the reputation system and resource allocation based upon it.

5.4.1 Reputation Calculation System for Member nodes

To make the proposed control system compatible with the prevalent reputation systems, we use a modified version of an existing reputation system [15]. Even if in future, somebody proposes altogether a new reputation evaluation system, then also the approach to build the overall control system will remain the same. Thus, with manageable changes, control system can work with any underlying reputation system. We now describe in detail reputation metric being used to estimate the cooperative behavior.

The trust is a measure of the cooperative behavior of a node. The trust (t_{ji}) calculated by the requesting node j for the serving node i for any transaction is defined as the ratio of the bandwidth received by the node j (T_{ji}) and what it has actually demanded from the node i (B_{ji}) during that transaction, *i.e.*,

$$t_{ji} = \frac{T_{ji}}{B_{ji}}. \tag{5.1}$$

Reputation of the node i at any time instant is calculated using the exponential moving average [76] of old reputation value with the average of current trust values as

$$R_i = (\alpha)R_i^{old} + (1-\alpha)\frac{\sum\limits_{j\in\mathcal{Z}_i} t_{ji}}{\sum\limits_{j\in\mathcal{Z}_i} 1}, \tag{5.2}$$

where R_i is the reputation of the node i and R_i^{old} is the last reputation value of node i available in the network. $\mathcal{Z}_i$ is the set of nodes which have requested the node i for resources after R_i^{old} was calculated. $\mathcal{Z}_i$ is reset every time the reputation R_i of node i is calculated. The cardinality of the set $\mathcal{Z}_i$ i.e $\left(\sum\limits_{j\in\mathcal{Z}_i} 1\right)$ is equal to l_i, which is the number of requests received by the node i after its last reputation evaluation. The proposed system uses global reputation, which is available to every peer in the network. The reputation aggregation (for calculating global reputation) can be achieved using gossiping [77] which has very low complexity in terms of memory, space, and time. Hence without much overhead, global reputation can be maintained by the members of a P2P network.

In (5.2), α determines how the exponential average forgets the contribution of the past values. The α always lies in the interval $[0, 1]$. A lower α causes the older observations to decay faster and $\alpha = 0$ completely ignores the older values. We have taken the value of α as $\frac{1}{2}$ in (5.2) so that equal [3] priority is given to the present as well as past reputation values. Thus, the reputation of the node i is

$$R_i = \frac{R_i^{old} + \frac{\sum_{j \in \mathcal{Z}_i} t_{ji}}{l_i}}{2}. \tag{5.3}$$

We now derive a generic resource distribution mechanism among the requesting nodes based upon their reputation.

5.4.2 Reputation Based Resource Distribution Among Requesters

To design a control system independent of underlying reputation based resource distribution technique among requesters, we present a generalized mathematical analysis for the resource distribution mechanism. Thus, every node in the network is free to use its own reputation based resource distribution independent of what other nodes are using. This generic framework estimates the amount of bandwidth available to a node from the network. The bandwidth allocated to any requesting node i from j is proportional to its reputation R_i, bandwidth demanded B_{ij} from j and a proportionality constant $k_{ij_{ovd}}$ which caters for change in received bandwidth due to overloading[4] at j. Thus the bandwidth received by i from j is

[3] In this way nodes are discouraged to abruptly change their behavior during the current time period to gain advantage in the reputation for the future transactions. The nodes need to be cooperative over a period of time for getting better quality of services than the existing services.

[4] Overloading represents the situation when overall bandwidth demanded by all the requesters from the serving node j is more than what j has shared.

given as

$$T_{ij} = k_{ij_{ovd}} \times R_i \times B_{ij}. \tag{5.4}$$

The value of $k_{ij_{ovd}}$ lies in the interval $\left[0, \frac{1}{R_i}\right]$. $k_{ij_{ovd}} = 0$, denotes the worst case of overloading at serving node j, where the receiving node i will not receive any bandwidth. When $k_{ij_{ovd}} = \frac{1}{R_i}$, the bandwidth requirement of node i will be completely satisfied by the node j. The value of proportionality constant $k_{ij_{ovd}}$ varies in each round depending upon the degree of overloading at the serving node. At the same time, for another requester say x requesting the same serving node, the value of $k_{xj_{ovd}}$ will vary depending upon its reputations. The mechanism to estimate $k_{ij_{ovd}}$ is out of scope of this thesis. We just need to ensure that control system remains stable, even for maximum possible value of $k_{ij_{ovd}}$.

The total bandwidth (T_i) received by the node i during the period, after last estimate of R_i is given by

$$T_i = R_i \times \sum_{j \in \mathcal{A}_i} k_{ij_{ovd}} B_{ij}, \tag{5.5}$$

where $\mathcal{A}_i$ is the set of the nodes from which node i has demanded the resources.

5.5 Modeling of Capacity Partition as Control System

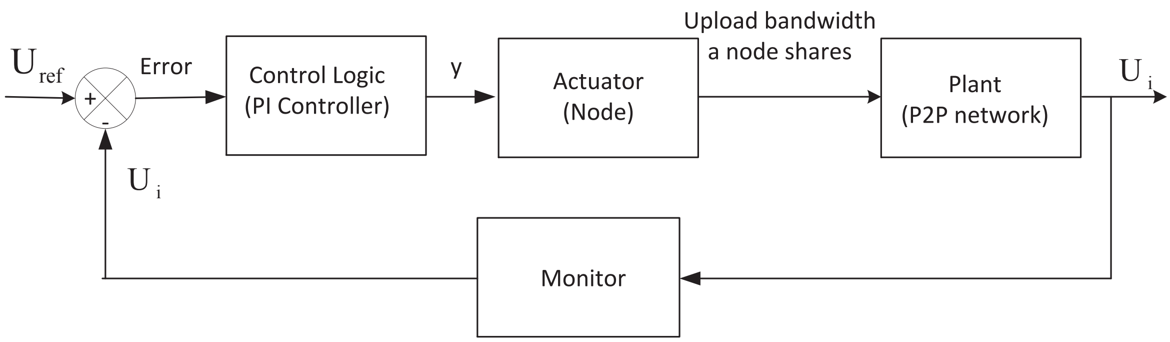

Figure 5.1: Proposed Control System

Dividing the total capacity between upload and download is a tricky problem in a P2P network. If a node keeps all of its capacity for download, then its cooperation level or reputation decreases and it will start receiving lesser resources from the network. After some time, a node's reputation will become 0 and it will not receive any resource from the network. On the contrary, if a node keeps all of its resources for sharing, it will not be able to download anything from the network. In both the scenarios, the utility or satisfaction derived from the network finally becomes 0. Therefore, a node needs to operate at an optimal point where it allocates minimal capacity for upload, so as to manage just enough reputation to fulfill its maximum possible download requirement from the network.

The problem of optimal partitioning of bandwidth into upload and download at a node, is modeled as a classical feedback control system, which follows a set or reference point. In our case, the reference point signifies the optimal point of bandwidth partitioning. The control system in Fig. 5.1, always strives to make the output approach the reference point based upon the feedback. The difference between the observed and the reference output is error. The error changes the upload bandwidth offered by the node, which subsequently modifies the node's reputation, thereby changing the effective bandwidth offered back to it by the network. The change in the output is always directed towards minimizing the error. This results in nodes working at optimal operating point. In practice, the control system dynamically adjusts the total capacity in such a way that a node shares minimal amount of upload capacity so that all of the remaining capacity left for download is completely utilized. Thus for most of the time, a node is able to derive maximum benefit from the network at minimal cost. The existing algorithm in [15], also seeks to optimize the total capacity, but they used fixed step size Δ to change the capacity distribution between upload and download. Due to fixed Δ, distribution never stabilizes but oscillates around the optimal point with an error proportional to Δ. Hence, the node either receives more or less than the optimal capacity allocated for upload and download, which results in wastage of overall network resource. The control system makes the step size say y proportional to error such that $y \rightarrow 0$ as the node approaches the optimal point. This minimizes the resource wastage due to over-allocation and the received resource shortage due to under-allocation of the upload bandwidth. Hence, due to variable step size y, the control system based resource allocation is more efficient than the existing techniques.

The following subsections discuss the level of optimality metric, reference point determination and controller action to achieve it. These subsections also deliberate about design of various components of control system shown in Fig. 5.1. The design involves determining the corresponding equivalent components for controller, actuator, plant and monitoring unit in a P2P network. Finally, we also derive the transfer function for this control system model.

5.5.1 Reference or Set Point in Control System

5.5.1.1 Level of Optimality U

To quantify the optimality level of resource sharing at a node, we propose a metric called 'level of optimality' which is denoted by U. For a node i, level of optimality U_i is defined as

$$U_i = \frac{T_i}{d_i}, \tag{5.6}$$

where T_i is the total data rate given by all the peers currently serving the requesting node i and d_i is the current[5] download capacity of the node i. U_i implicitly signifies benefit or the satisfaction achieved by the node i from the network.

At the outset, U may resemble the popular concept of utility [78] in computer networks but it is quite different. The utility corresponds to the degree of the satisfaction received by the node from the network, whereas U is dependent upon the total resources that are available to the node from the network. Sometimes, the node's capacity may not be sufficient to utilize all the available resources, therefore these extra resources which are considered in U are useless for the node and don't contribute towards the utility derived by the node. For example, if a node i is offered bandwidth T_i, such that $T_i > d_i$, the current download capacity of i. This scenario is possible because peers usually demand more[6] [15] than their download capacity to increase their chances of getting resources from the network. The node i can only utilize the bandwidth equal to its current download capacity d_i and the rest of the offered bandwidth $(T_i - d_i)$ gets wasted. The U_i takes T_i into account, whereas utility is dependent on d_i only. We now try to determine

[5]For single capacity link, d_i may vary in each round depending upon the capacity i has kept for upload.

[6]Demanding in excess is usually beneficial when network is overloaded with requests.

the value of U_i which will be optimal point of operation such that a node gets maximum benefit at minimum sharing.

5.5.1.2 Reference Point Determination

A control system always strives to bring the system to the reference point or a set point. When the requesting node i increases its upload capacity s_i, its reputation increases. This makes i eligible for greater amount of download T_i from the network. However, download capacity d_i of node i limits maximum download and consequently received resources, which can be utilized by the node. Hence, we need to find an optimal point where a node gets maximum resources from the network at the minimum cost. This optimal point corresponds to the reference point of the proposed control system. To determine the reference point (refer (5.6)), we divide the feasible region $\mathbb{R}^+$ of variable U_i into three subsets $\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3$ as follows.

$$U_i \in \begin{cases} \mathcal{U}_1 = [0, 1), & \text{when } T_i < d_i, \\ \mathcal{U}_2 = \{1\}, & \text{when } T_i = d_i, \\ \mathcal{U}_3 = (1, \infty), & \text{when } T_i > d_i. \end{cases}$$

1. ($U_i \in \mathcal{U}_1$):- During this state, both satisfaction and U_i derived by the peer i from the network increases with increase in T_i. Therefore, a node always has scope of increasing its utility or satisfaction in this region of operation. Hence, reference point doesn't lie in set $\mathcal{U}_1$, because it doesn't contain point of maximum satisfaction level.

2. ($U_i \in \mathcal{U}_3$):- Increasing s_i in this region, results in more resources (T_i) from network. However increased T_i will not be beneficial for the node because it exceeds the maximum data rate d_i a node can handle. Therefore, when ($U_i \in \mathcal{U}_3$), with increase in s_i, U_i increases but the satisfaction derived by the node decreases. The decrease in satisfaction level occurs because d_i decreases as s_i increases. Hence, node will be able to process lesser resources from the network resulting in decreased satisfaction from the network. Consequently,

node i is at loss when its U_i increases beyond 1, as i has to pay more in terms of the greater upload capacity. Hence, $\mathcal{U}_3$ will not contain reference point .

3. ($U_i \in \mathcal{U}_2$):- This point corresponds to minimum s_i with which a node can receive resources T_i equal to its present download capacity d_i. Therefore, maximum satisfaction is achieved at $U_i = 1$ with minimum cost.

Hence, $U_i = 1$ corresponds to the reference point of control system.

5.5.1.3 Achieving Reference Level

The feedback monitor in the loop (see Fig. 5.1) will estimate function U_i. This value will be compared with the reference point ($U_{\text{ref}} = 1$) and the error is fed to the controller. In other words, upload capacity available at node i is adjusted so that it is adequate enough to make the U_i at the node i equal to the U_{ref} of the control system.

5.5.2 Control System Components

The various components of the proposed control system model as shown in Fig. 5.1 are described below

5.5.2.1 Controller

Controller [79] stabilizes the system output to a particular value called reference or set point. Based upon the difference between the reference point (U_{ref}) and the current level of optimality (U), the controller drives the actuator to regulate the output. The proportional action in a controller helps the node in reaching optimal value of resource sharing faster, whereas integral action reduces the steady state error[7] [79]. When the P2P network reaches the steady state, nodes generally share in proportion to their requirement so as to preserve their reputation. Therefore

[7]Steady state error is the perturbations occurring after a node has reached optimal point of operation, *i.e.*, $U = U_{\text{ref}}$.

total shared capacity in network does not change abruptly and so does U as consequence. The differential action used to counter sudden changes in output (U in our case), is therefore not required in our system. Thus the PI controller with transfer function[79]

$$G(s) = K_p \left(1 + \frac{k_i}{s}\right) \tag{5.7}$$

is sufficient to model the control system, where K_p and K_i are the proportional and integral gain respectively of the PI controller. The value of these gain parameters are calculated in section 5.5.4.

In the proposed model, controller modifies the shared capacity on the basis of the error ($U_{\text{ref}} - U_i$). As the output of controller is proportional to error, therefore the size (y) through which upload capacity gets modified is adaptive. As the error becomes 0, so does the y, implying that the shared capacity stabilizes around the optimal point.

5.5.2.2 Actuator

The role of the actuator [79] in the control loop is to update physical entity based upon the controller output. In the model under consideration, the actuator modifies the upload capacity of the node in response to the controller output y. The actuator changes the shared capacity of the node by y units in the next round. The change in the upload capacity by the actuator modifies the reputation of the node, thereby adjusting the download bandwidth, the node receives from the network.

5.5.2.3 Plant

Plant represents the P2P network, which decides the amount of resources available to a node from the network on the basis of the capacity shared by that node. The output of the plant corresponding to a node i is the level of optimality U_i.

5.5.2.4 Monitor

The function of the monitor or the observer at the node i is to sense the output of the system. Output is then compared with the reference point to calculate the error and drive the controller. In the proposed system, monitor gain of any node i is 1 as the system output, *i.e.*, U_i can be compared directly with reference point U_{ref}. Let $C(s)$ denote the transfer function of controlled software system (which

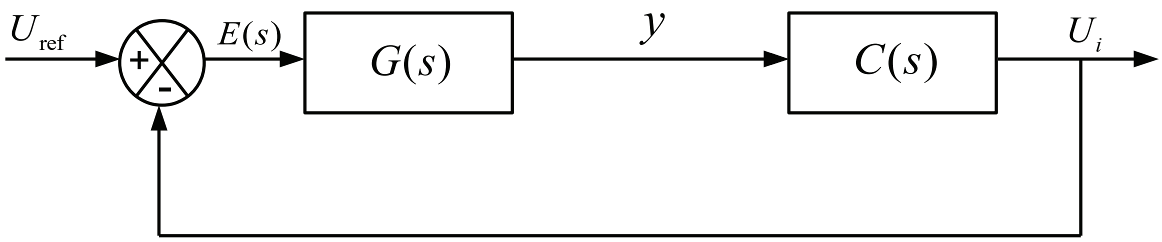

FIGURE 5.2: Transfer Function representation of Bock Diagram

includes actuator and P2P network) and $G(s)$ represents the transfer function of controller. Representing various blocks in Fig. 5.1 with their corresponding transfer functions, we obtain Fig. 5.2. To complete the design of proposed control system, we further derive the transfer function of various components in the subsequent sub section.

5.5.3 Controlled P2P Software System Model

For the derivation of transfer function $C(s)$ for controlled P2P software System model (actuator and plant), we need to first calculate its output, *i.e.*, level of optimality (U).

5.5.3.1 Evaluating Level of Optimality (U_i) at Node i

Using (5.1) and (5.3) the reputation of the node i is given by

$$R_i = \frac{R_i^{old} + \frac{\sum_{j \in Z_i} \frac{T_{ji}}{B_{ji}}}{l_i}}{2}, \tag{5.8}$$

where R_i^{old} is the last reputation value of the node available in the network. T_{ji} and B_{ji} are the bandwidth received and demanded respectively by the node j from i. $\mathcal{Z}_i$ denotes the set of nodes requesting service from i while l_i is the number of requests catered by i. The total bandwidth (T_i) received by the node is calculated using (5.5) as

$$T_i = R_i \times \sum_{k \in \mathcal{A}_i} k_{ik_{ovd}} \times B_{ik} = \frac{R_i^{old} + \frac{\sum_{j \in \mathcal{Z}_i} \frac{T_{ji}}{B_{ji}}}{l_i}}{2} \times \sum_{k \in \mathcal{A}_i} k_{ik_{ovd}} \times B_{ik}, \tag{5.9}$$

where $k_{ik_{ovd}}$ is a constant which takes the overloading at the serving node into account.

The serving peers can estimate the feasible capacity of the link to the requesting peer and use it in place of B_{ji} in (5.9). Feasible capacity is the minimum amount of the bandwidth required to support the feasible service rate across the link. Feasible service rate is the maximum achievable throughput via underlying path in the network with packet loss probability p. It can be estimated from the expression of effective rate for an end-end TCP connection using TCP Reno congestion control algorithm [80] as

$$R(p) \approx \frac{M}{RTT \cdot \sqrt{\frac{2bp}{3}} + T_0 \cdot \min\left(1, 3\sqrt{\frac{3bp}{8}}\right) p(1 + 32p^2)}. \tag{5.10}$$

Here $R(p)$ is the feasible service rate which is the function of the packet loss probability p, M is the maximum transmit window that the receiver indicates to the sender and RTT is the round trip time between the two nodes. T_0 is the retransmission timeout in seconds and b is the number of packets acknowledged by each acknowledgment message. On substituting B_{ji} by feasible bandwidth B_{ji}^{fes} in (5.9), the total bandwidth received by a node is given by,

$$T_i = \frac{R_i^{old} + \frac{\sum_{j \in \mathcal{Z}_i} \frac{T_{ji}}{B_{ji}^{fes}}}{l_i}}{2} \times \sum_{k \in \mathcal{A}_i} k_{ik_{ovd}} \times B_{ik}, \tag{5.11}$$

where B_{ji}^{fes} is the minimum amount of bandwidth required to support a feasible data rate from node i to j. B_{ji}^{fes} can be estimated using (5.10). Since the aim of the dissertation is to optimally partition link capacity between upload and download, TCP Reno congestion control algorithm [80] (in (5.10)), is not simulated in this dissertation. We assume that serving nodes know the value of B_{ji}^{fes}.

As, nodes will estimate the capacity using TCP Reno congestion control algorithm, serving node will never provide resources greater than the capacity at the disposal of the requesting node. Therefore, node i will demand resources equal to its current download capacity (d_i), *i.e.*, $B_{ik} = d_i$. Consequently, (5.11) becomes

$$T_i = \frac{R_i^{old} + \frac{\sum_{j \in \mathcal{Z}_i} \frac{T_{ji}}{B_{ji}^{fes}}}{l_i}}{2} \times d_i \sum_{k \in \mathcal{A}_i} k_{ik_{ovd}}.$$

Using (5.6), the measured variable U_i corresponding to the node i is given by

$$U_i = \frac{T_i}{d_i} = \frac{R_i^{old} + \frac{\sum_{j \in \mathcal{Z}_i} \frac{T_{ji}}{B_{ji}^{fes}}}{l_i}}{2} \times \sum_{k \in \mathcal{A}_i} k_{ik_{ovd}}. \tag{5.12}$$

5.5.3.2 Deriving Transfer Function for Controlled P2P Software System

For ease of analysis, we linearize the non-liner model[8] of the process which optimally partitions total capacity between uplink and downlink. Any non-linear system can be linearized if it works under narrow operating range [79]. The proposed capacity partitioning process can be linearized because a node operates within a tapered operating range around the optimal point, *i.e.*, $U = U_{\text{ref}}$ for maximum period of its lifetime due to the following reasons:

1. A new entrant is able to swiftly achieve $U = U_{\text{ref}}$ due to the proportional action of controller.

[8]Many complex nonlinear systems like web servers, servomotor, tachometer, synchros have been studied using linearization.

2. Once $U = U_{\text{ref}}$ is achieved, the integral action maintains U's value close to U_{ref} during perturbations arising due to change in network dynamics. For details refer to section 5.6.3.

3. Later in section 5.6.2, we prove that it is more profitable for new entrants to initially share its total capacity equally between upload and download. This approach also helps new entrants to operate near U_{ref} as soon as they enter the network.

In addition to the above stated arguments, the accuracy of linearized model is further substantiated by simulation results in section 5.6.2, which show that the linearized version of the proposed system closely follows the output of actual nonlinear P2P system. Hence, proposed system can be modeled as a linear system. We futher derive the transfer functions $C(s)$ and $G(s)$ for this linear system.

As discussed in 5.6.2, capacity partitioning already starts near the optimal point, so we can neglect the process dynamics[9], and the controlled software system's transfer function $C(s)$ can be modeled as static gain c. On linearizing this gain c around the reference point ($U_{\text{ref}} = 1$), c can be obtained as the derivative of the output (U) with respect to the input (y). The gain c^i corresponding to the node i is given as

$$c^i = \frac{dU_i}{dy} = \frac{d\left[\left(\frac{R_i^{old} + \frac{\sum_{j \in \mathcal{Z}_i} \frac{T_{ji}}{B_{ji}^{fes}}}{l_i}}{2}\right) \times \sum_{k \in \mathcal{A}_i} k_{ik_{ovd}}\right]}{dy}. \tag{5.13}$$

The c^i is also called the process gain corresponding to the combined gain of actuator plant and the monitor[10]. It signifies system's sensitivity and is defined as relative distance process variable (U) travels in response to change in controller output (Y). The c^i is subsequently used for controller design. Tuning the controller for maximum process gain ensures the stability of the system for all controller gain values [81, 82, 83]. Therefore, to design a robust controller, we derive the transfer function corresponding to maximum process gain. The maximum gain c^i_{max} occurs when $B_{ji}^{fes} = B_{min}^{fes}$. Where, $B_{min}^{fes} = \min_{i,j} \; B_{ji}^{fes}$.

[9] Process dynamics play significant part during transient phase ($U \neq 1$) of the system. The transient state analysis is an interesting problem and will be taken up by the authors in future.

[10] For the proposed system monitor gain is unity

Another factor $k_{ik_{ovd}}$ as discussed in section 5.4.2 attains its maximum value when reputation of node is minimum. R_{min} is the minimum value of the reputation at which a node is eligible for resource allocation in the P2P network. Therefore, the maximum possible value of ($k_{ik_{ovd}}$) such that node can receive data from network is $\frac{1}{R_{min}}$. The total capacity $\sum_{j \in \mathcal{Z}_i} T_{ji}$ currently allocated by node i is equal to its previous shared capacity s_i^{old} plus the the amount y by which it gets currently modified. y is the current controller output. Hence the (5.13) can be used to determine the maximum gain c^i_{max}.

$$c^i_{max} = \frac{d\left[\left(\frac{R_i^{old} + \frac{(s_i^{old}+y)}{l_i \times B_{min}^{fes}}}{2}\right) \times \frac{1}{R_{min}} \times \sum_{k \in \mathcal{A}_i} 1\right]}{dy}, \tag{5.14}$$

where $\sum_{k \in \mathcal{A}_i} 1$ represents the cardinality of the set $\mathcal{A}_i$ and is equal to g_i, the number of the nodes from which node i is currently requesting for the resources. In a network, the average generated (g_i) and received (l_i) requests by any node i settles down to the same value after some time, *i.e.*, $\frac{g_i}{l_i} \to 1$. This happens because, a network cannot store data packets. Therefore average upload and download should be equal to maintain the network balance. In addition, reputation system forces nodes to download in proportion to their contribution or upload. Consequently the requests generated for average upload and download by a node in the network should be the same. This claim has been further verified through simulations. On substituting $\frac{g_i}{l_i} = 1$ in (5.14), we get

$$c^i_{max} = \frac{d\left(\frac{l_i \times R_i^{old} + \frac{(s_i^{old}+y)}{B_{min}^{fes}}}{2} \times \frac{1}{R_{min}}\right)}{dy} = \frac{1}{2R_{min}B_{min}^{fes}}. \tag{5.15}$$

Change in the upload capacity induced by actuator is observed in the next period. This results in dead time (delay) of one period (T). Further, (5.15) does not contain any term specific to the node i. Therefore, superscript i from c^i_{max} can be dropped to obtain a generic expression applicable to any node in the network. Hence, the overall transfer function of the controlled software system for any member node can be given as

$$C(s) = c_{max} \cdot e^{-sT} = \left(\frac{1}{2R_{min}B_{min}^{fes}} \right) e^{-sT}. \tag{5.16}$$

The overall transfer function of the control loop as shown in Fig. 5.2 is given by

$$T(s) = C(s)G(s) = c_{max} \cdot e^{-sT} K_p \left(1 + \frac{k_i}{s} \right). \tag{5.17}$$

For ease of analysis we convert above equation in into Fourier form as follows (Refer APPENDIX A for details).

$$T(w) = c_{max} \cdot e^{-jwT} K_p \left(1 + \frac{K_i}{jw} \right). \tag{5.18}$$

This $T(w)$ is used in the subsection to calculate the gain parameters K_p and K_i of the PI controller.

5.5.4 Tuning PI Controller

The proposed PI controller is tuned w.r.t $C(s)$ on the basis of gain margin specification. The gain margin [84] is the amount of increase or decrease in gain required to make the loop gain $T(s)$ equals to 1 at the phase crossover frequency (w_p), where phase angle ($\angle T(w_p)$) is $-\pi$.

From the definition [84], the phase and gain of the system at w_p are $-\pi$ and inverse of gain margin (G) respectively, *i.e.*,

$$\angle T(w_p) = -w_p T - tan^{-1} \left(\frac{K_i}{w_p} \right) = -\pi, \tag{5.19}$$

and

$$|T(w_p)| = |c_{max}| \left| (e^{-jw_p T})(K_p) \left(1 + \frac{K_i}{jw_p} \right) \right| = \frac{1}{G}. \tag{5.20}$$

In the industrial PI controllers, it is common practice to tune the controller phase, *i.e* ., $-tan^{-1}\left(\frac{K_i}{w_p}\right)$ to $-\frac{\pi}{6}$ [85][81], thus

$$tan^{-1}\left(\frac{K_i}{w_p}\right) = \frac{\pi}{6}. \tag{5.21}$$

The parameter G and T are set by the designer and c_{max} is calculated using (5.15). Using (5.21) in (5.19) we get the value of phase crossover frequency as $w_p = \frac{5\pi}{6T} = \frac{2.618}{T}$ rad/s.

The parameter K_i of controller is calculated by substituting above computed value of w_p in (5.21). The simple algebraic manipulations give

$$K_i = \frac{1.512}{T}. \tag{5.22}$$

The above computed values of w_p and K_i are used in (5.20) for obtaining the expression for K_p as

$$K_p = \frac{0.866}{c_{max}G}. \tag{5.23}$$

Equations (5.22) and (5.23) provide the generic value of the model parameters. The actual parameter values used during simulations are calculated in the next section. All the component values namely U_{ref}, $G(s)$ and $C(s)$ constituting the proposed control system in Fig. 5.2, have been evaluated. The subsequent section provides details about practical realization of the proposed system.

5.6 Performance Evaluation

To analyse the performance of proposed model, we simulate a discrete time P2P network. In the current scenario, discrete time simulation is suitable because parameter (bandwidth) used for performance analysis changes in discrete steps at discrete time instants. Various parameter values used for modeling the control system in the discrete simulation model are calculated from the transfer function in continuous domain. The conversion from continuous to discrete domain is carried out by converting various input and output relationship representations

Parameter	Description	Value
N	Number of the nodes in network	100
K_p	Proportional Gain	0.00577
K_i	Integral Gain	0.151149
C_i	The access link capacity at the node i	$4Mbps$
G	Gain Margin	3
α	Exponential moving average constant	$\frac{1}{2}$
U_{ref}	Level of optimality which any node in the network, wants to achieve	1
R_{in}	Initial reputation assigned to newcomers	0.07
R_{min}	Reputation threshold below which node does not receive any service	0.01
B_{min}^{fes}	The minimum bandwidth required to support feasible data rate possible on the link between nodes i and j	$2Mbps$

TABLE 5.2: Simulation Parameters along with Values

in integro-differential equations[11] to their corresponding discrete counterparts, *i.e.*, summation-difference equations. The time step for the summation-difference equations is taken to be equal to a round. Each round signifies a discrete time slot[12]. The nodes send requests at the starting of round, and if they get selected for the service allocation their requests get fulfilled within the same round. For the next round, the whole process is repeated again. The simulation results presented in section 5.6.2 show that the output obtained from the discrete time model closely follow continuous system response. As there is no significant loss of data or introduction of spurious information at output after discrete modeling of the system, continuous to discrete time conversion is considered to be acceptable. A P2P network is simulated for 1000 rounds and this simulation is repeated 10

[11]It is an equation containing both integral and derivative functions.
[12]We divided time into discrete time slots for discrete implementation.

times. The final results presented in this thesis are averaged out over these 10 simulations to eliminate any random coincidence.

In [1, 15], simulation of small sized P2P network, consisting of 50 or 100 peers was carried out. However, to demonstrate that proposed system will work fine even for higher number of peers, simulation of network containing 1000 peers is carried out. The new entrant in a network is provided with initial reputation, $R_{in} = 0.07$ for its survival, while any node whose reputation falls below threshold reputation, $R_{min} = 0.01$ is rendered ineligible for receiving services. Each round is assumed to last for 10 *sec*. As changes initiated by controller appears at output with delay of 1 time period, therefore dead time (T) value is taken as 10 *sec*, in the current simulation. The above mentioned parameter values are in accordance with the reputation model of [15]. The value of gain margin is set to $G = 3$ [84]. The remaining parameter values are taken from existing P2P models in [1, 15], as it allows us to easily compare our results with existing literature [1, 15]. To facilitate readability, various simulation parameters along with their values are listed under Table 5.2. The design parameters of PI controller, $K_p = 0.00577$ and $K_i = 0.151149$ are evaluated from (5.22) and (5.23), respectively, using previously discussed parameters.

We assume that resources required at the requesting node is generally available at all the members except the requesting node (refer section 5.3.2). Therefore, in the simulation model under consideration, a node randomly requests other members with equal probability for resource download. After slight modification, the proposed model is also applicable to scenarios where service providers to be requested are selected on the basis of their reputation. However, such a selection mechanism will overload high reputation nodes with huge amount of download requests, resulting in degradation of service quality provided by them. Hence the random approach for server selection is investigated during the simulations.

Above mentioned specifications, along with mechanism for resource discovery and exchange discussed under section III, have been used to develop a customized simulator for P2P network. In addition, various control system components e.g., controller, used in the simulation were simulated as per the mathematical model discussed under section 5.5. We now present simulation results to demonstrate the performance of the proposed model.

5.6.1 Real Time Implementation of Proposed Control System

The proposed control system in practical implementation is an algorithm named adaptive step size (ASZ), being run at each node. ASZ dynamically adjusts the upload (s_i^p) and consequently the download capacity ($d_i^p = C_i - s_i^p$) at any node i during time period p, such that i operates around optimal point of resource sharing, *i.e.*, U_{ref}. We will show in next subsection 5.6.2 that, if any new entrant i initially allocates half of its capacity for sharing, *i.e.*, $s_i^0 = \frac{C_i}{2}$ then it results in better reception of resources from the network. At the starting of each round, the level of optimality U_i^p at node i is calculated by taking the ratio of the bandwidth received to the current download bandwidth at a node. In each iteration, the upload capacity (s_i^p) gets modified by amount y_i^p (output of the PI controller) which pushes the node i's U_i^p to U_{ref}. For any time period p, y_i^p is calculated [79] from error ($E_i^p = U_{\text{ref}} - U_i^p$), which have occurred in the system till time period $p - 1$. To maintain a minimum level of satisfaction from the network, the download bandwidth should not decrease beyond a threshold level Δ_{thr} as in [15]. If the download bandwidth (d_i^p) reduce below threshold value, then the download and the upload capacity are set to values Δ_{thr} and $C_i - \Delta_{thr}$, respectively. Finally, the new capacity partition values, *i.e.*, s_i^{p+1} and d_i^{p+1} are updated for the next period $p + 1$. This process is repeated as long as the node remains in the network. The ASZ algorithm is presented in Algorithm 1. In the subsequent sections, we study the resource allocation process using ASZ algorithm.

5.6.2 Study of Control System Performance

We simulate the linear model of resource allocation process in P2P network using SIMULINK in MATLAB. The block diagram model used in SIMULINK is based on Fig. 5.2. The components values used in the model, *i.e.*, transfer functions $G(s)$ and $C(s)$ are evaluated from (5.7) and (5.16) respectively. The parameter values used in these equations are listed under Table 5.2. The output of the linear model is represented by the label $U_{\text{Transfer Function}}$ in Fig. 5.3. In the simulation model of resource distribution in actual P2P system, Algorithm 1 is used to decide the distribution of total capacity between upload and download. At the outset, in

Algorithm 1 ASZ: Bandwidth Allocation Algorithm

Initialization:
Initialize k_p ,k_i and C_i from Table 5.2
Set $s_i^0 = \frac{C_i}{2}$, $\Delta_{thr} = \frac{C_i}{10}$, $E_i^0 = 0$, $TE_i^0 = 0$ and $U_{\text{ref}} = 1$
$p \leftarrow 0$
Shared Capacity Evaluation:
repeat
  Compute U_i^p as in equation (5.6) and update
  $E_i^p \leftarrow U_{\text{ref}} - U_i^p$
  $y_i^p \leftarrow K_p \times E_i^p + K_i^p \times (E_i^p + TE_i^p)$
  $TE_i^{p+1} \leftarrow TE_i^p + E_i^p$
  $S_i^{p+1} \leftarrow S_i^p + y_i^p$
  if $C_i - S_i^{p+1} \leq \Delta_{thr}$ **then**
    $S_i^{p+1} \leftarrow C_i - \Delta_{thr}$
  end if
  $d_i^{p+1} \leftarrow C_i - S_i^{p+1}$
  $p \leftarrow p + 1$
  return S_i^{p+1} and d_i^{p+1}
until Node i is in the network

P2P Nwk 1, a node keeps all of its capacity for download, while in P2P Nwk 2 a node initially allocates half of its capacity for upload. Thereafter, the nodes in the network adjust their capacity based upon Algorithm 1.

Fig. 5.3 demonstrates how closely the results calculated using transfer function follow the simulation results of both types of P2P networks. This validates the accuracy of the transfer function and the PI controller tuning parameters (K_p and K_i), which were derived in section 5.5.3 by linearizing the actual system. Initially, there is slight deviations between the three systems however, for most part of node's life time, these systems follow each other, hence these deviations can be tolerated. Variable step size (y) is used to modify the upload capacity and consequently the total capacity distribution between upload and download. The step size y, is proportional to the error between current U and desired U, *i.e.*, U_{ref}, as mentioned in Algorithm 1. As the system reaches steady state, error$\rightarrow$ 0 so does the step size. Due to dynamically adjusting step size, the upload capacity and thereby the system, stabilizes around the optimal point (U_{ref}), unlike existing algorithms where fixed step size causes shared capacity to oscillate around optimal point. This will be discussed in more detail under section 6.3.1.

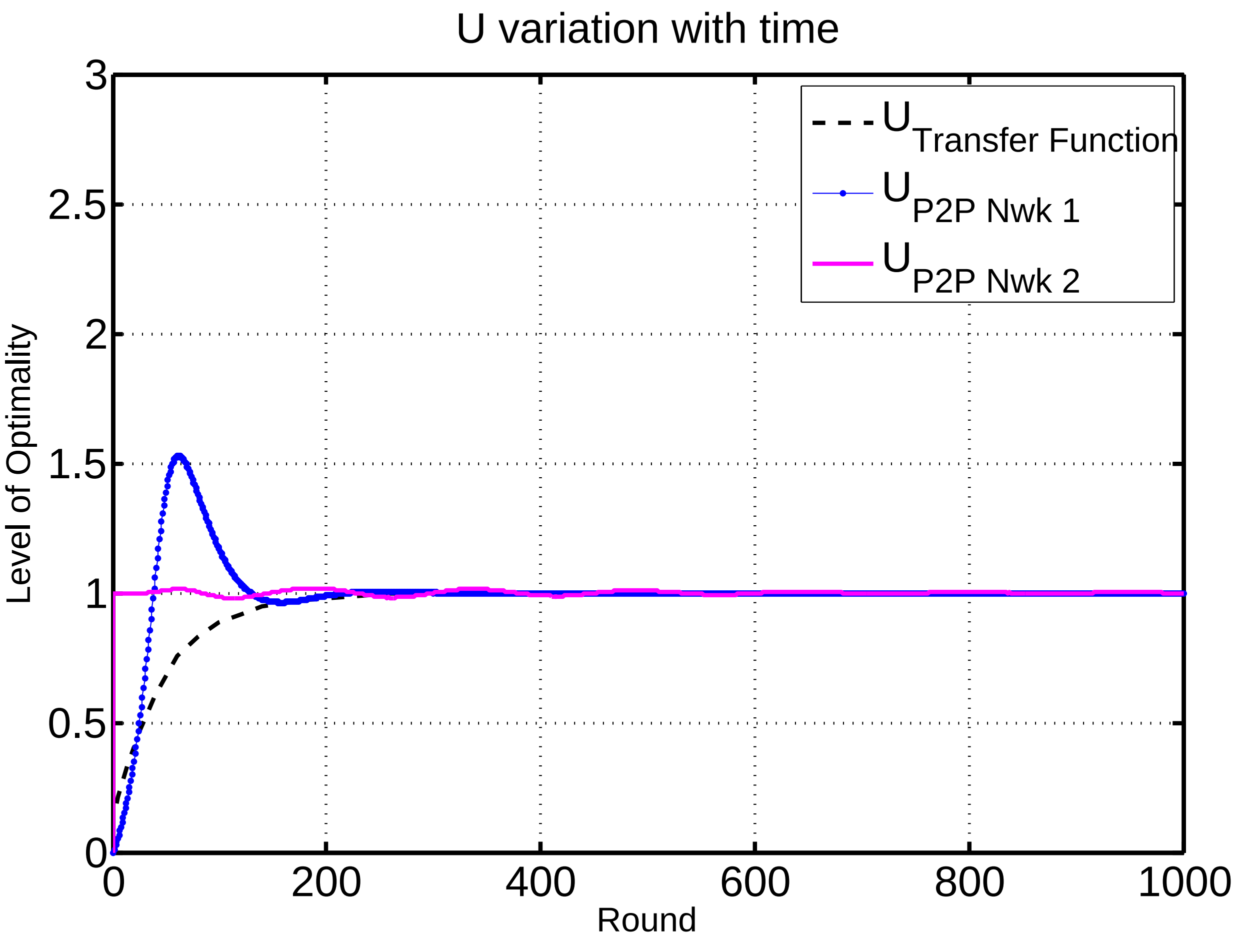

FIGURE 5.3: Transfer Function Comparison with Actual Network

While linearizing the system model for driving the transfer function, it was assumed that node operates around $U_{\text{ref}} = 1$. When there is large deviation of U_i of node i from U_{ref}, then it take some time for system to settle down. This happens in P2P Nwk 1 and transfer function implementation, where node starts with $U_i = 0$ as node initially share nothing. This deviation is more prominent in P2P Nwk 1, where during start of network, a node's output overshoots the (U_{ref}) in Fig. 5.3. This large deviation happens because initially all the nodes share nothing, so any node i receives 0 resources (T_i) from the network. The error input ($U_{\text{ref}} - U_i$) to controller becomes very high. This causes PI output to reach very large value for the next period, which leads to node i sharing large amount of s_i. After some time, there is abundance of resources across network as all nodes are sharing large proportion of their capacity due to initial high error. So U_i of node i overshoots U_{ref}. The PI controller again readjusts its output to modify the shared capacity so that U_i stabilizes around U_{ref}. This scenario is analogous to supply and price of a commodity in market. Initially (T_i), commodity available in market is scarce so its price in terms of the upload capacity (s_i) of a node is very high. As all the nodes start sharing, the T_i available to a node i increases, so its cost in terms of s_i decreases. However, s_i values are updated in next round, therefore, U_i overshoots U_{ref}. Node i readjusts its price s_i based upon current availability of T_i in market and finally, commodity (T_i) attains a stable value in market corresponding to $U_i \rightarrow U_{\text{ref}}$.

In P2P Nwk 2, any node i initially shares its capacity equally between upload and download. Therefore its U_i is close to U_{ref} when node joins the network. Consequently, all the members always work around optimal point of operation. Fig. 5.3 shows that nodes are at loss if they initially share nothing. Nodes will prefer to be members of P2P Nwk 2 over P2P Nwk 1. Therefore, P2P Nwk2 will be used for all further simulation and comparison in this thesis. Thus in Algorithm 1, we have set initial upload capacity s_i as half of total capacity $\frac{C_i}{2}$. This also justifies our assumption while deriving transfer function, that system had already attained the steady state, *i.e.*, working close to U_{ref}.

5.6.3 Control system's adaptiveness to Change in Network Dynamics

In this section, we study the adaptiveness of the proposed system to the changing network dynamics. Dynamic changes occur due to the nodes entering or the existing nodes either leaving the network or becoming free-riders.

A node becoming free-rider is even worse than node leaving the network because departing nodes will neither contribute nor consume resources, whereas free-riders in spite of no contribution, try to consume resources. Therefore, system behaving appropriately for free-riders will also be robust for nodes leaving the network. Thus, we check system's adaptability for only the free-riding nodes. In the simulation model, the percentage of free-riders is gradually increased from 0% to 99.99% of the total nodes. Network bootstraps with 1000 node with no free-rides. At the end of rounds, which are multiple of 100, there is increment

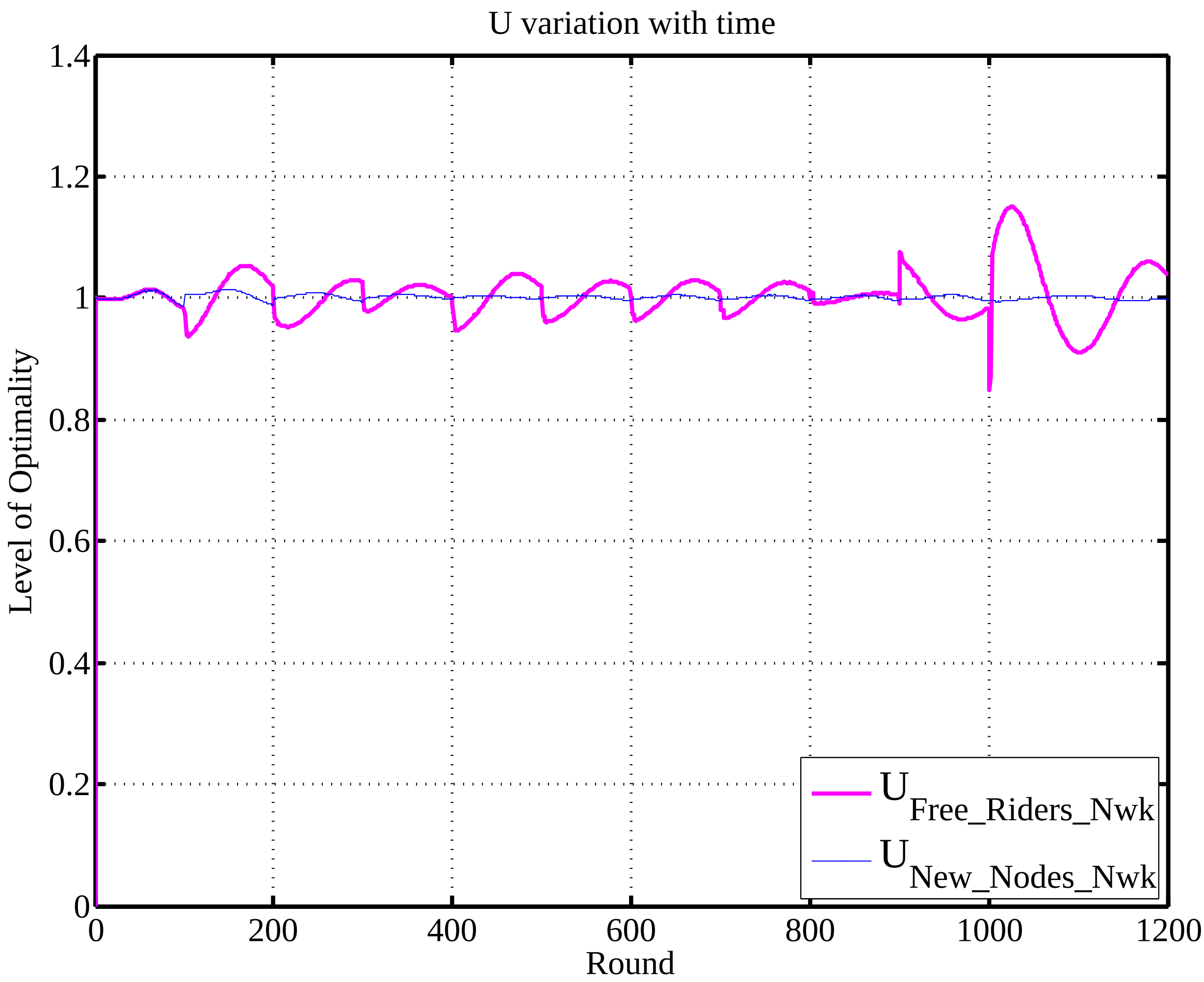

FIGURE 5.4: System Performance under Changing Network Dynamics

in free-riders percentage by 10% of the initial number of nodes. Thus, after 100th round 10% contributing nodes become free-riders, this percentage increases to 20% after 200th round and so on. At the end of 1000 rounds, 90% of the nodes free-ride. Finally, to model worse case scenario, after 1050 rounds, 99.99% of the total nodes in the network are free-riding.

As shown in Fig. 5.4, contributing nodes also referred as genuine nodes are able to maintain their level of optimality ($U_{\text{Free\_Riders\_Nwk}}$), around optimal value, *i.e.*, U_{ref}, even when other nodes become free-riders. There is slight deviation in $U_{\text{Free\_Riders\_Nwk}}$ from U_{ref} because there arise scarcity of resources when contributing nodes become free-riders. This condition arises because, their is more demand for resources across the network and less resources available, as new free-riders stop contributing but they are still eligible to receive resources. Free-riders will continue to receive resources for some time, till their reputation falls below threshold value. Therefore, in Fig. 5.4, $U_{\text{Free\_Riders\_Nwk}}$ initially falls down. As genuine nodes start operating below U_{ref}, they start sharing more to achieve U_{ref}. At the same time the reputation of free-riders also start decreasing because of their non cooperative behavior and after some time it reduces below threshold value. Once all the nodes who have just started free-riders get barred from resource reception, the amount of resources available for download becomes greater than the net resource demanded across the network, because of extra resources put in by genuine nodes. Hence, $U_{\text{Free\_Riders\_Nwk}}$ overshoots U_{ref}. The proposed control system automatically adjusts the shared capacity of member nodes to bring back $U_{\text{Free\_Riders\_Nwk}}$ to U_{ref}. The amplitude of perturbations in $U_{\text{Free\_Riders\_Nwk}}$ from U_{ref}, increases with increase in percentage of free-riders as there are more free-riding nodes and less genuine nodes to compensate for the reduced resources. Thus, greater fall in $U_{\text{Free\_Riders\_Nwk}}$ value is encountered when more nodes become free-riders. The genuine nodes have to contribute more than earlier scenario which had lesser percentage of free-riders. Hence, when new free-riders get debarred from service, there are greater amount of extra resources w.r.t. resources demanded in the network. Consequently, increase in $U_{\text{Free\_Riders\_Nwk}}$ from U_{ref} is more. However, amplitude of perturbations is not very large to cause any major concerns and proposed system is able to bring back $U_{\text{Free\_Riders\_Nwk}}$ to U_{ref} after some time. Hence, the system is robust to free-riding behavior of nodes.

We now ascertain the robustness of the proposed system in handling new entrants in the network. Initially there are 100 nodes is the network. After end of the rounds, which are multiple of 100, 100 new nodes arrive in the network. It implies that when 100^{th} and 200^{th} round ends, the total number of nodes in the network become 200 and 300, respectively. This process continues till 900^{th} round, where total count of members in network reaches 1000. The new entrants start contributing as soon they enter the network. Therefore, the net resources demanded is equal to resources available across the network. Consequently, demand and request equilibrium is maintained so there is no appreciable change in ($U_{\text{New\_Nodes\_Nwk}}$), as depicted in Fig. 5.4. Thus new nodes entering the system do not disturb the optimal point operation of the other nodes in the network.

Hence, simulation result clearly demonstrate that the proposed control system is adaptive to changing network dynamics.

5.6.4 Compatibility Analysis of Control System

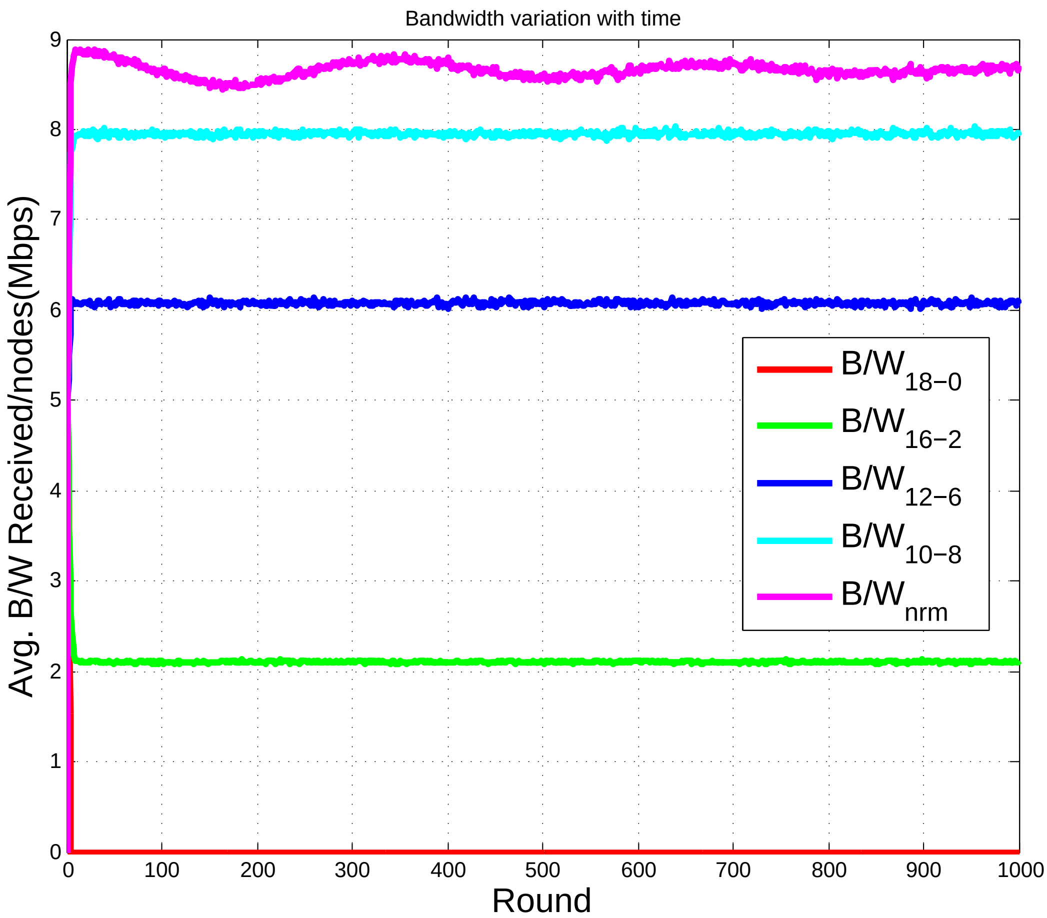

FIGURE 5.5: Comparability Analysis of Control Systems.

We now analyze the effectiveness of control system when it is deployed together with existing system, *i.e.*, portion of peers are using control system and remaining peers are using some other partitioning strategy.

The control system is implemented at each node in such a way that it is independent of partitioning strategy employed by the other nodes. It will optimize on the basis of resources received from the other nodes in the network. Therefore, analysing performance of ASZ w.r.t. efficient partitioning strategies[13] [15, 69] is almost similar to analysis, where all the nodes are employing ASZ. In such scenario, all the nodes will be able to easily achieve and subsequently maintain their level of optimality around 1 (refer section 5.6.1). The real performance issue lies in more adverse scenario where net resources available across network is less than resources demanded. Therefore we consider a scenario, where there is scarcity of resources across the network, *i.e.*, other nodes are either free-riders or employ inefficient strategies, such that it is always allocating greater part of its link capacity for download, independent of the resources received by the node. The simulation model is described as follows.

Every node is connected to the network with a single capacity link of 18 *Mbps* as in [15]. In the network, 20% of peers are normal peers, who share according to the proposed control system. Their corresponding received bandwidth is denoted by BW_{nrm}. The nodes employing inefficient partitioning strategies are modeled as follows. 20% of peers completely free-ride, *i.e.*, they use their entire link capacity of 18 *Mbps* for download, other 20% peers will contribute 2 *Mbps* capacity at max for uploading, while other 20% will provide maximum 6 *Mbps* for upload and the remaining 20% peers left, will provide maximum 8 *Mbps* of their link capacity for upload. The corresponding bandwidth received back from network is denoted by BW_{18-0}, BW_{16-2}, BW_{12-6}, and BW_{10-8} respectively. These peers are designated as free-riders, as they request more and share less than what is prescribed by the control system.

From Fig. 5.5, it is evident that normal peers who follow the proposed control system receive maximum bandwidth from the network. The amount of bandwidth received by them, does get slightly affected as there is scarcity of resources across the networks arising due to high level of free-riding (80% of the peers free-ride in some form). While linearizing system in section 5.5.3.2, it was assumed that node's operational range is close to the optimal partitioning. To achieve this, nodes initially allocate half of their total capacity for upload (refer section (5.6.2) for details). However due to resource scarcity, sharing half of total capacity initially is not enough to make the system operate close to optimal partitioning.

[13]The strategies in which average bandwidth allocated for upload and download are close to optimal partition are referred as efficient partition strategy

Hence, an initial oscillatory behavior is observed in received bandwidth for normal nodes (refer Fig. 5.5), which settles down at optimal partitioning after some time. This is in line with output of P2P Nwk 1 in Fig. 5.3, where nodes initially operating in the region far away from optimal partitioning. The other peers are not using control system to adapt their link capacity to maximize their output. Therefore they do not show any oscillatory behavior. However, such peers are at loss as the bandwidth received by the nodes decreases with decrease in capacity allocation for upload, even though they have more download bandwidth available at their disposal. This happens because of decrease in contribution level with the decline in upload capacity. Thus, there is no advantage for nodes in allocating more portion of their total capacity for download. As nodes employing ASZ receive highest payoff, so the free-riding nodes in network will gradually start implementing ASZ algorithm.

Hence, without any modification in a network, new node employing ASZ can be deployed. In addition exiting nodes can also employ ASZ as partitioning strategy so as to receive higher resources from the network.

The capacity partitioning algorithms [15, 71] are run in a distributed manner at every node to maximize node's utility/usable bandwidth. This objective is in line with the definition of Nash equilibrium (NE) (refer (1.3)), where each player is playing its best response *i.e.* maximizing its usable bandwidth. Closer the capacity partitioning is to the *NE*, higher the usable bandwidth received by the node, attaining the maximum value at *NE*. Therefore, *NE* can be used to ascertain which algorithm provides higher usable bandwidth and therefore, should be preferred.

5.7 ASZ's Equilibrium analysis

We analyse the efficiency of ASZ in utilization of the network resources by comparing the partitioning of access link capacity achieved by it with the partitioning during Nash equilibrium (NE) state evaluated in Chapter 4. When network is in *NE* state, every node divides its access link capacity equally between upload and

User Capacity (*Mbps*)	**Upload Capacity** (*Mbps*) ASZ(Nash)	**Download Capacity** (*Mbps*) ASZ(Nash)
4	1.97(2.0)	2.03(2.0)
5	2.49(2.5)	2.51(2.5)
6	2.99(3.0)	3.01(3.0)
7	3.51(3.5)	3.49(3.5)
8	4.03(4.0)	3.97(4.0)

TABLE 5.3: ASZ Comparison with Nash Equilibrium.

download. As discussed earlier in Chapter 4, greater the deviation of partitioning achieved by any algorithm w.r.t. partitioning in *NE* state, lesser its efficiency.

Based on existing models [15, 71], we consider a network of 100 nodes. Nodes are distributed in equal proportion into 5 groups of different link capacities viz. $4, 5, 6, 7$ and 8 Mb/s. The partition of link capacity is carried out using ASZ, described in algorithm 1. Table 5.3 shows that capacity partitioning achieved by ASZ closely follows the *NE*. Hence, ASZ efficiently utilizes the network resources.

The partitioning algorithm is run independently at each node to maximize its utility. This is equivalent to every node playing its best response, *i.e.*, playing strategies which maximize its payoff. At *NE*, no player can increase its utility by unilateral deviation, so every player is also playing its best response at *NE*. Therefore, objective of partitioning algorithm shows congruence with the definition of *NE*. Hence, an efficient partitioning algorithm will finally lead the network to *NE* state.

5.8 Conclusions and Future Work

The authors successfully modeled the partitioning of a node's total capacity between upload and download as a feedback control problem. Based on this modeling, we proposed adaptive step size (ASZ) algorithm which make the nodes to operate at optimal partitioning level, *i.e.*, $U = 1$. When $U = 1$, a node can fulfill its download requirement with minimum upload. The proposed control system is also adaptive to changing network dynamics, as nodes are able to maintain their U around 1, even during arrivals and exits of the nodes from the network. At the same time, ASZ can be easily deployed together with the existing schemes to help nodes achieve maximum utility from the network. Finally, we also compared partitioning achieved by ASZ with link capacity partitioning during Nash equilibrium. Results indicate that capacity partition achieved by ASZ, converges towards Nash equilibrium, thereby establishing that ASZ efficiently partitions access link capacity.

In future, we plan to do the transient analysis of the P2P system in more detail, using non linear tools like state space analysis.

Chapter 6

Control System Comparison with Existing Schemes

6.1 Introduction

Until now, we have evaluated our proposed capacity partitioning system in general setup for P2P networks and explored its effectiveness in optimally partitioning the access link capacity. In this section, we compare the proposed model with existing schemes, *i.e.*, BitTorrent [1] and Reputation-Based Allocation Policy (RRA) [15] to determine the improvement achieved in terms of efficiency and fairness w.r.t. these schemes. Efficiency signifies how effectively the available resources across the network are exploited, whereas fairness implies that the resources received should be in proportion to their cooperative behavior. We also derive control theoretic model of RRA to show that capacity partitioning achieved by it will be oscillatory. Our claim about RRA gets subsequently verified by simulation results.

A typical swarm[1] in BitTorrent consists of 50 peers [15][75], whereas Satsiou and Tassiulas [15] simulated a network of 100 peers for analyzing RRA. For ease of analysis, same sized P2P network is used for comparison purpose. In the subsequent section, we provide details of simulation setup along with the

[1]The set of peers actively uploading and downloading data [1]

comparison results. We also discuss the limitations of earlier models and describe how the proposed model is able to overcome them.

6.2 BitTorrent

In year 2001, Brahm Cohen developed BitTorrent, a communication protocol to help users share files in a P2P network [1]. The files typically consist of videos, audio, software etc. After its initial launch, BitTorrent has gained enormous popularity among Internet users to share content.

Unlike most of the P2P networks, BitTorrent uses a central repository (called tracker) which contains address of the peers possessing a particular file. For sending and receiving files using BitTorrent, the user should have BitTorrent client installed in his device. BitTorrent client is a software which implements BitTorrent protocol. It communicates with tracker to find out other users in the P2P network, which have file (or portion of file) of our interest. A file is divided into large number of chunks and most of peers in network have portion of file, *i.e.*, some of the chunks of the required file. Peers trade chunks among themselves to get complete copy of file. This trading is accomplished using *"tit for tat"* strategy.

In *"tit for tat"* strategy, peers trade chunks with those peers who provide them with highest download rate. In addition, peers also randomly try some new peers in the network for downloading file. Random selection of peers for service, allows newcomers in the network to receive chunks and it also helps a peer to find out peers, who could provide better download rate than existing service providers. In this manner, *"tit for tat"* strategy induces cooperation among members of BitTorrent application.

BitTorrent was developed during the time when most of the peers were using wired connection, where there is strict separation between upload and download capacities, e.g., asymmetric digital subscriber line (ADSL) [50] links. Therefore, capacity partitioning mechanism to optimally divide link capacity between upload and download is absent in BitTorrent[15]. In this section, we compare our proposed approach with BitTorrent and illustrate improvements achieved in terms of fairness.

6.2.0.1 Simulation Setup

We have simulated elementary version of BitTorrent [1, 15] based upon Azures, a popular BT Client. All the peers are connected by single capacity link. As the main focus of the thesis is on evaluating performance of allocation strategy, we overlook block selection algorithm and assume that a peer is always interested in blocks of its neighbor. Section 5.3.2 describes, how this assumption will not lead to appreciable deviation in results *w.r.t.* original network. The time in BitTorrent model is broken into periods, each lasting for 10 seconds. Each peer say x, implements choking algorithm [1] for its resource distribution. Top four peers who have provided highest download rate to x receive upload from the peer x. After every third period, peer x randomly selects a peer for resource allocation regardless of its download performance, thereby allowing new entrants to obtain initial chunks and trying out other peers to find out better service providers. Once requesters are finalized by serving node x, the upload bandwidth is allocated equally among the requesting peers.

6.2.0.2 BitTorrent Modeling as Control System

The BitTorrent protocol lacks capacity adaptation algorithm [15] which can dynamically adapt upload and download capacities in single capacity link to optimize the overall performance. Therefore we cannot design a control system model for capacity devision in the system employing BitTorrent protocol.

6.2.1 Comparison with BitTorrent

In the simulation model, peers are grouped on the basis of maximum upload capacity they are willing to share. In Fig. 6.1, $18-0$, $16-2$, $12-6$, $10-8$ and $9-9$ represents group of peers sharing upload bandwidth up to 0, 2, 6, 8 and

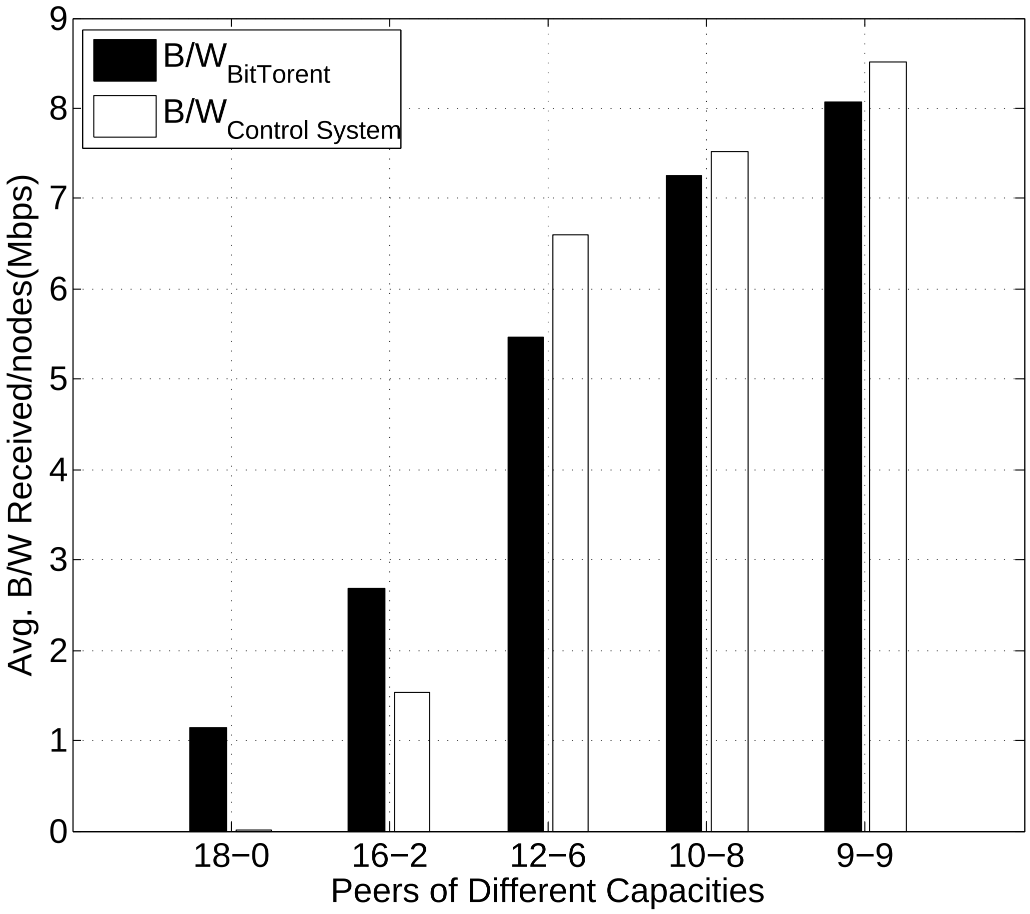

FIGURE 6.1: BitTorrent Comparison

9*Mbps*, respectively. The last three groups consisting of nodes sharing upload bandwidth up to 6, 8 and 9*Mbps*, receive greater resources when they implement control system instead of BitTorrent for bandwidth division between upstream and downstream. The optimistic unchoke slots[2] and round robin schemes[3] of seeders in BitTorrent [75] provide some chances for free-riders to download. Free-riders get these undue share of resources at the expense of reduced resources awarded to the deserving high reputation peers, thereby compromising fairness in BitTorrent protocol. Unlike BitTorrent, the proposed model heavily penalizes the free-riders giving them no opportunity to download once their reputation falls below a threshold value. This leads to better performance of our system

[2]Each peer periodically unchoke a neighboring peer, transferring some resources to it for free.

[3]When seeder has nothing to download, they distributes resources independent of what requesters have shared. In fact some torrent protocols [1] in such scenario, distribute resources on the basis of download capacity of requesters, which unintentionally benefits free-riders.

over BitTorrent as shown in Fig. 6.1, where nodes with low cooperation level, *i.e.*, sharing up to 2*Mbps*, receive meager amount of resources.

6.3 Reputation based Resource Allocation (RRA) Policy

As peers are rational, they always seek to maximize their utility derived from the network. In current context utility derived is the fraction of received resources that can be utilized (refer (2.3)). Therefore, Satsiou and Tassiulas in [15], assumed that initially every peer will allocate their entire link capacity for download, in expectation of utilizing greater amount of received resources from the network. However, this reduces contribution of a peer and consequently its reputation. Therefore, peers will start receiving less resources from the network. Thus reputation system employed by P2P network, forces each peer to allocate some portion of its access link capacity for upload, in order to receive resources from the network. In such scenario, member peers will seek to maximize their utility with minimum possible contributions.

RRA [15] strives to divide peer $i's$ total capacity between upload and download in such a way that a peer is able to derive maximum utility from network with least possible contribution. Let C_i, s_i^p and d_i^p represent total, upload and the download capacity of the peer i during time period p. Peers initially upload nothing and then, after each round change their upload capacity in fixed step size $\Delta = \frac{C_i}{10}$, until they get desired bandwidth from the network or it doesn't exceed their total capacity. As discussed earlier, increase in upload capacity increases peer's chances of getting more resources from the network. However, when a peer receives as much bandwidth as its current downlink can handle, it will further try to maximize its utility by incrementing its download capacity by step size Δ at the expense of decreased upload bandwidth. This process is represented by Algorithm 2. After adjusting its shared capacity, a serving peer i distributes its resources among requesters in decreasing order of their reputation to demand ratio.

Algorithm 2 RRA Bandwidth Allocation Algorithm

Initialization:
Initialize C_i from Table 5.2
Set $s_i^0 = 0$, $d_i^0 = C_i$ and $\Delta = \frac{C_i}{10}$
$p \leftarrow 0$
Shared Capacity Evaluation:
repeat
  Compute T_i^p by adding all the resources received by i, during period p
  if $(T_i^p < d_i^p)$ AND $(S_i^p < C_i - \Delta)$ **then**
    $S_i^{p+1} \leftarrow S_i^p + \Delta$
  else if $(T_i^p \geq d_i^p)$ AND $(S_i^p \geq \Delta)$ **then**
    $S_i^{p+1} \leftarrow S_i^p - \Delta$
  end if
  $d_i^{p+1} \leftarrow C_i - s_i^{p+1}$
  $p \leftarrow p + 1$
  return s_i^{p+1} and d_i^{p+1}
until Node i is in the network

6.3.1 Comparison with Reputation-Based Allocation Policy (RRA)

6.3.1.1 RRA Modeling as Control System

The total capacity distribution between download and upload capacity can be modeled as feedback control problem, where based upon the feedback (amount of download (T_i)), a user is receiving from the network, it keeps on adjusting its upload capacity (s_i) and consequently its download capacity (d_i). To simplify the modeling of RRA algorithm as a control system, "If" conditions in original *RRA*, *i.e.*, Algorithm 2, are re-written in terms of level of optimality U. The modified RRA is represented by Algorithm 3. The RRA algorithm at node i will strive to achieve $U_i = 1$, *i.e.*, optimal point of resource sharing. The output of the RRA algorithm is the amount by which s_i will be modified in the subsequent round. As RRA permits the modification of the upload capacity (s_i) in fixed step size Δ, the output of controller used to implement RRA can take only two values $+\Delta$ or Δ. Input for the RRA controller is the error signal, *i.e.*, $U_{\text{ref}} - U_i$. According to RRA strategy, s_i is to be increased by amount Δ, when the feedback (*i.e.*, amount of download (T_i)) a node receives from the network is less than its current download capacity (d_i). As $U_i = \frac{T_i}{d_i}$, therefore incrementing shared capacity by Δ corresponds

Algorithm 3 RRA Algorithm Modeled as Control System

Initialization:
Initialize C_i from Table 5.2
Set $s_i^0 = 0$, $\Delta = \frac{C_i}{10}$ and $U_{\text{ref}} = 1$
$p \leftarrow 0$
Shared Capacity Evaluation:
repeat
  Compute U_i^p as in equation (5.6) and update
  if $(U_i^p < 1)$ AND $(S_i^p < C_i - \Delta)$ **then**
    $S_i^{p+1} \leftarrow S_i^p + \Delta$
  else if $(U_i^p \geq 1)$ AND $(S_i^p > \Delta)$ **then**
    $S_i^{p+1} \leftarrow S_i^p - \Delta$
  end if
  $d_i^{p+1} \leftarrow C_i - s_i^{p+1}$
  $p \leftarrow p + 1$
  return s_i^{p+1} and d_i^{p+1}
until Node i is in the network

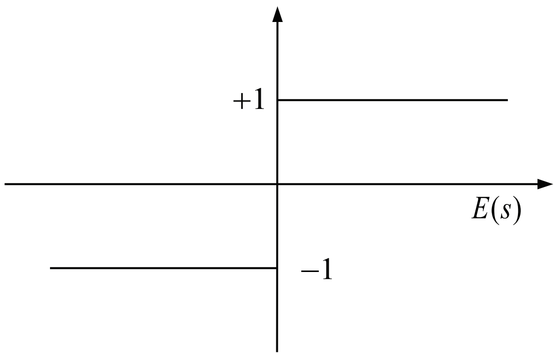

FIGURE 6.2: I/P and O/P of Controller in RRA.

to state when error is positive. Similarly, other way around, the output of RRA controller is $-\Delta$ for the negative error. Hence $G_{RRA}(s)$, the transfer function of controller used to implement RRA is ($\Delta \times$ signum function) shown in Fig. 6.3. Laplace transform of signum function is given by $\frac{2}{s}$ [86] (refer Fig. 6.2). Therefore

$$G_{RRA}(s) = \frac{2\Delta}{s}. \tag{6.1}$$

The control system model of resource allocation process using RRA is presented in Fig. 6.3. The transfer function $C(s)$, derived in the section 5.5.3.2 is reused to represent actuator and plant. The controller's transfer function, which is specific

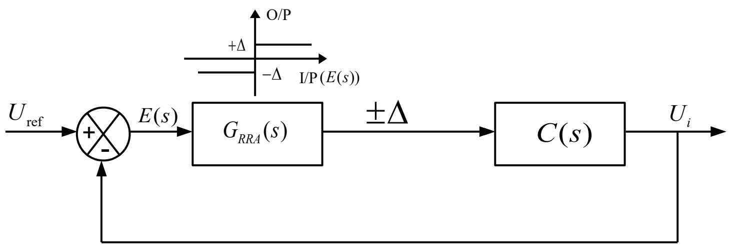

FIGURE 6.3: Control Loop at Node i for RRA Implementation.

to the partitioning algorithm is derived as follows. Since the derived controller is not tuned keeping system's stability under consideration, therefore simulation of the linear model ($T(s) = G_{RRA}(s) \times C(s)$) of the system using Simulink, leads to unbounded output U, oscillating between $(-\infty, \infty)$. In actual network simulation, as the link capacities are finite so U does not reach ∞ but it saturates and keeps on oscillating with a finite amplitude. Hence, resource allocation using RRA when compared with ASZ will be less efficient. Results along with discussion are elaborated in the next subsection

6.3.1.2 RRA Result Comparison and Limitation

The level of optimality (U_{RRA}) received by nodes in a network employing RRA is calculated using equation (5.6). Let $U_{CONTROL}$ denote the level of optimality received by the nodes employing the proposed control system. The comparison between the RRA and control system is demonstrated in Fig. 6.4 and 6.5. Fig. 6.4 shows level of optimality (U) observed at a single node whereas Fig. 6.5 plots the U which is averaged out across all the nodes in the network.

As RRA uses a fixed step size Δ for modification in capacity partitioning, the level of optimality (U) received by any node will not settle down and it will keep on oscillating around $U_{\text{ref}} = 1$ with the amplitude proportional to the step size Δ. To demonstrate this problem, we observed bandwidth allocation at one particular node selected randomly from the network (Fig. 6.4). It clearly shows that U_{RRA} never becomes equal to U_{ref}, but oscillates around it. However in control system based approach, due to the inherent integral action [79] of the

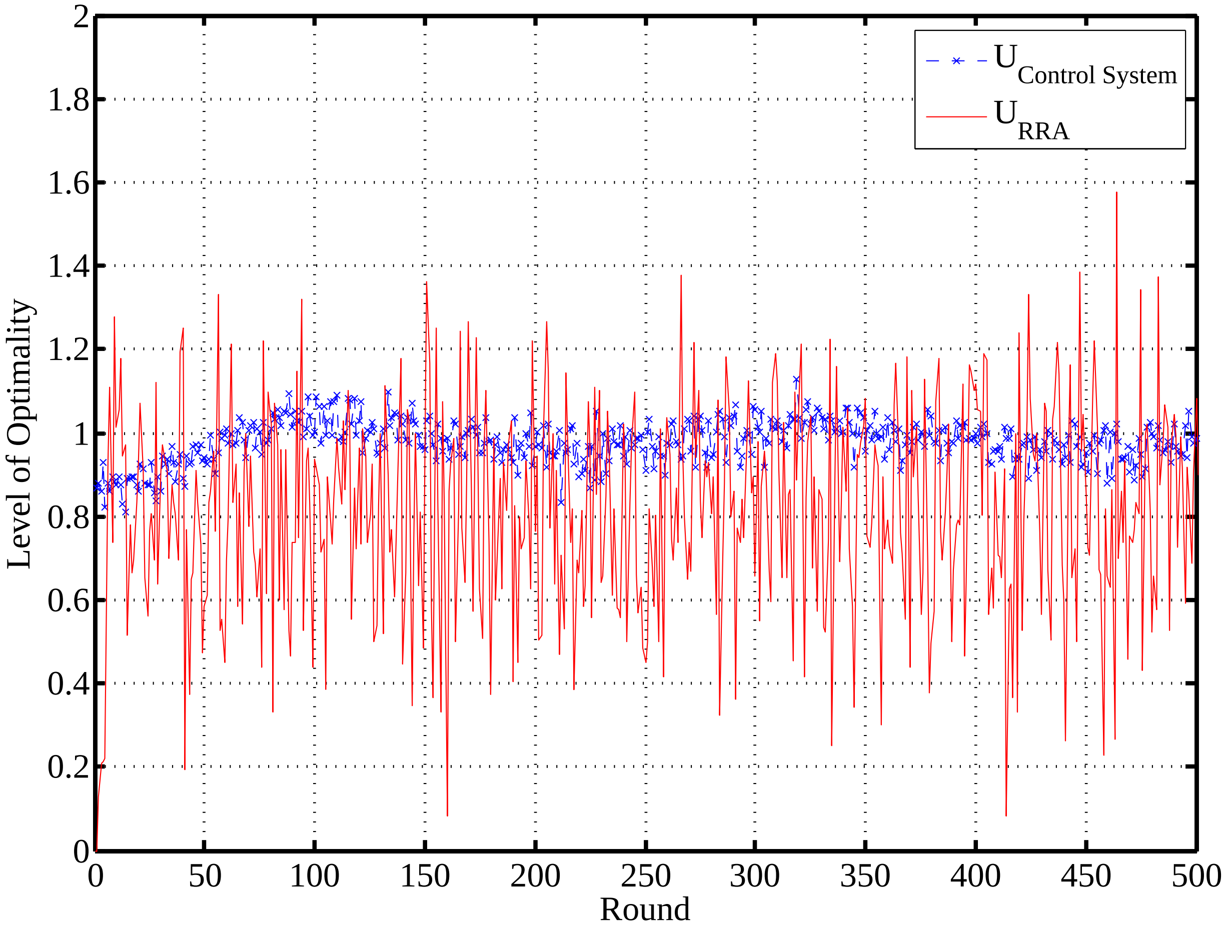

FIGURE 6.4: RRA Comparison at Single Node

PI controller, the steady state error[4] gets reduced and node level of optimality ($U_{CONTROL}$) follows desired U_{ref} (Fig. 6.4). Satsiou and Tassiulas [15] who proposed RRA, studied average bandwidth allocation for the overall network, *i.e.*, authors took the sum of the bandwidth received at every node and then averaged it out. However this method of averaging is unable to show the unsettled behavior in the network because it gives mean and not the higher order moments like variance. It physically implies that the average across the overall network gives the false impression that bandwidth has stabilized, because some of the nodes whose capacities are getting increased in current round are being compensated by the nodes whose capacities are decreasing. When the level of optimality (U_{RRA}) achieved by using RRA algorithm is average out, U_{RRA} does stabilizes but at a

[4] Steady state error refers to the difference between U_{ref} and actual level of optimality ($U_{CONTROL}$) received, when system is already working close to U_{ref}.

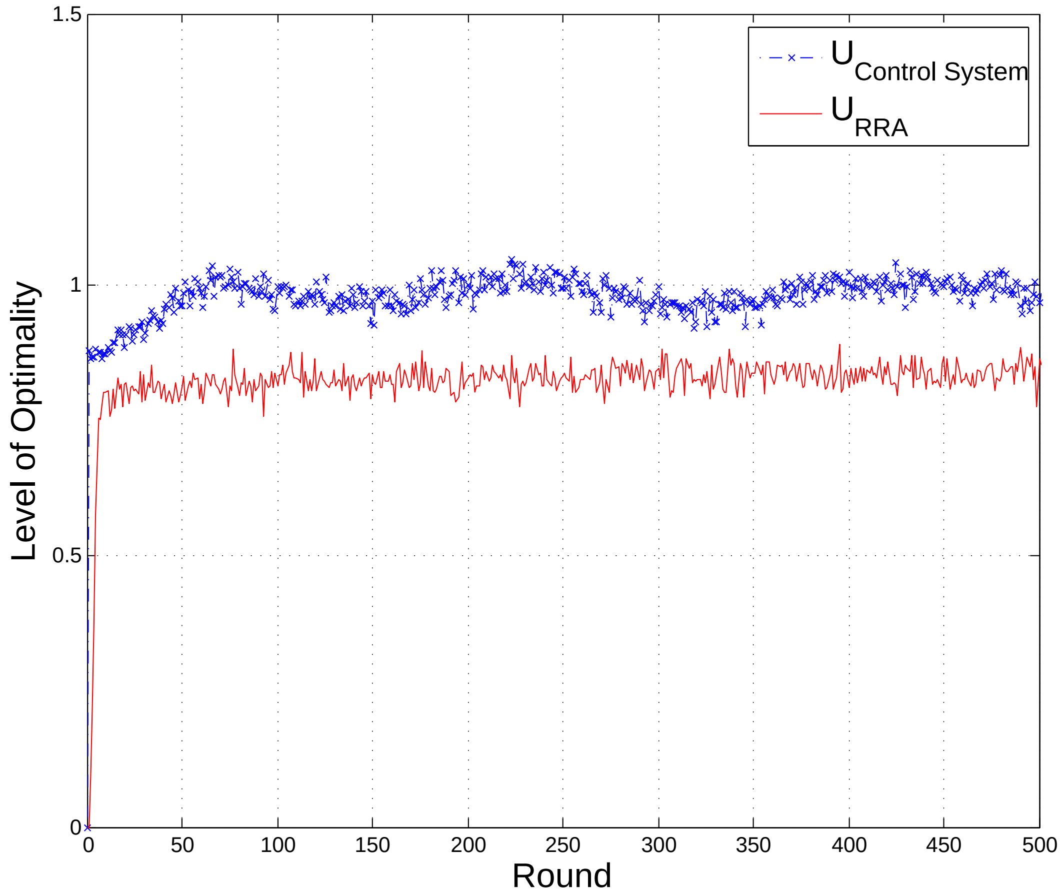

FIGURE 6.5: RRA Comparison Averaged Over Network

point lower than $U_{\text{ref}} = 1$, as shown in Fig. 6.5. Averaged U_{RRA} is unable to achieve U_{ref} because there is greater wastage of resources in the network implementing RRA. Bandwidth wastage occurs when a node is getting more than its current download capacity. A node can not utilize this additional capacity. At the same time a needy node, who receives less than its current capacity will be deprived of the resource. In RRA, this wastage is proportional to fixed step size (Δ), while the control system minimizes bandwidth wastage by adjusting step size according to error, *i.e.*, ($U_{\text{ref}} - U_{CONTROL}$). $U_{CONTROL}$ denotes average efficiency of node implementing ASZ algorithm. Fig. 6.5 substantiates this claim where nodes implementing proposed control system are able to achieve $U = 1$.

Hence the proposed control system is a better resource allocation mechanism than RRA in terms of efficient allocation of resources

6.3.1.3 Complexity Comparison

The PI parameters calculation is one time process, therefore its time complexity is $O(1)$. Although the number of arithmetic operations and storage variables required in ASZ are more than in RRA, their asymptotic bound is same, which results in same time ($O(n)$) and space ($O(n)$) complexity for both the algorithms.

Hence, the proposed ASZ algorithm gives better performance than RRA, without any significant increase in the system's complexity.

6.4 Conclusion

We have compared the proposed control system with the existing schemes like BitTorrent and RRA. The proposed system when compared with BitTorent, is more stringent in adhering to contribution level while distributing resources among the requesters. Therefore, lesser resources get awarded to the free-riders in proposed scheme w.r.t. BitTorrent. We have also carried out the control theoretic modeling of Reputation-Based Allocation Policy (RRA) and demonstrated that RRA is unable to maintain node's U close to 1, thereby reducing its efficiency w.r.t. the proposed model in terms of received bandwidth. When nodes implement the proposed model they are able to receive greater bandwidth for the same contribution level. This improvement is obtained by using the PI controllers to generate adaptive sizes for modifying the total capacity partitioning.

Chapter 7

Conclusions and Future Directions

With advancements in wireless technology, high data rate can be provided to users at low cost. Hence, usage of wireless communication technology is becoming popular in the P2P network. In wireless technologies like WiFi, WLAN, LTE and WiMAX (in time division duplex (TDD) mode), nodes are connected to the backbone network via an access link through which uplink and downlink data flow. The partitioning of access link capacity between uplink and downlink can be altered by the users. In order to seek maximum resources from the network, a user will try to allocate the entire link capacity for download. However, incentive mechanisms (eg. Reputation System) force users to maintain certain level of contribution (in the form of upload) to continue receiving resources from the network. Therefore, the best strategy for a node is to maintain a minimal level of upload, such that it receives resources equal to its current download capacity. The capacity partitioning corresponding to minimal upload level to achieve maximum download is referred to as optimal point of partitioning.

7.1 Conclusions

In this dissertation, we have evaluated the optimal point of partitioning, where nodes are able to receive maximum download with minimum upload. We also proposed a mechanism/algorithm which helps nodes in the network to operate at optimal partitioning level. This algorithm is implemented in a distributed fashion, in line with the basic structure of P2P networks which lack any central authority.

In Chapter 1, we have presented an overview about P2P network along with the major research challenges in their implementation. This chapter also details some basic concepts of game theory, which will be helpful in understanding game theoretic analysis in the later part of dissertation. Further, we have provided a detailed explanation about the problem of capacity partitioning. Finally, we conclude this chapter with the discussion about existing state of art to solve this problem.

Chapter 2, deals with the study of optimal capacity partitioning in homogeneous P2P networks. P2P networks consisting of users using the same WiFi or wireless LAN network can be considered as homogeneous network, *i.e.* one where all nodes have the same access link capacity. We have modeled partitioning of access link capacity as a game and evaluated its Nash equilibrium (NE). The strategy of equal partitioning of access link capacity between uplink and downlink is found to be *NE*. In addition, this *NE* comes out to be socially optimal. Hence, equal partitioning strategy ensures maximum resource utilization at a node. The theoretical analysis on homogeneous network has been further verified using the simulation results.

Most of the incentive mechanisms used to prevent free-riding consider only contribution level of the node for resource distribution. In Chapter 3, we have established that, if such incentive mechanism is used for capacity partitioning, then high capacity nodes can easily manipulate resource distribution process such that they receive resources which should have been allocated to the lower capacity nodes. Thus some lower capacity nodes will receive no resource, even if they allocate the maximum possible link capacity for upload. Using game theoretic analysis we prove that for unbiased resource distribution, the serving node should distribute resources in decreasing order of the ratio of the contribution to the consumption of the resources by requesters.

In Chapter 4, we have extended our game theoretic model to the P2P networks containing nodes with heterogeneous access link capacity and evaluated the optimal capacity partitioning for heterogeneous P2P networks. When resources are distributed in the decreasing order of the ratio of the contribution to the consumption of resources by requesters, then the strategy of equal division of link capacity between uplink and downlink is the *NE* of capacity partitioning game. Beside being the *NE*, strategy of equal partitioning is also socially optimal. Therefore, this strategy maximizes the resource utilization at a node.

To help nodes in the P2P network to operate at optimal partitioning level, we have proposed adaptive step size (ASZ) algorithm in Chapter 5. ASZ considers many aspects of real time P2P system. This algorithm dynamically adjusts capacity partitioning at the node when new nodes enter or existing nodes leave the network. Further ASZ can be easily integrated in the network, where other nodes may not partition their link capacity according to the proposed mechanism. In addition, if the network contains free-riding nodes, then the nodes implementing ASZ receive maximum possible download from the network. We have also provided simulation results to verify the above mentioned claims.

Finally in Chapter 6, we have compared the proposed ASZ algorithm with existing state of art. Reputation-Based Resource Allocation Policy (RRA) uses fixed step size while ASZ uses variable step size to change capacity partitioning between upload and download. The step size in ASZ is directly proportional to difference in current partitioning level and the optimal partitioning. Hence, ASZ outperforms RRA in efficient distribution of resources across the network. In addition, through simulation we demonstrate that ASZ is fairer than BitTorrent in distribution of resources across the network. Further, using simulation results we have established that the nodes employing ASZ for capacity partitioning are able to operate near the optimal partitioning level.

7.2 Future Directions

This dissertation is mainly focused on study and design of optimal partitioning of the link capacity (between uplink and downlink) for efficient content sharing across the network. However, as discussed earlier in Chapter 1, content sharing between nodes across P2P networks involves many issues. In this thesis, we have investigated one of the issues which is very critical in efficient performance of P2P network containing wireless links. Other issues in capacity partitioning can be part of future study. Some possible extension of our work in future could be the following.

- In this dissertation, we have considered incentive mechanisms which distribute resources on the basis of cooperation level of a node. However, there are malicious users which publish fake content. Even if such peers are cooperative, the fake content is of no use to the requesting node. To counter this problem, a new incentive mechanism may be devised which considers content relevance along with cooperation level for determining the incentive level of a node. It will be interesting to analyse whether the Nash equilibrium in the capacity partitioning game based upon the new incentive mechanism, shifts or remains the same. In addition, a modified version of ASZ algorithm will be required to maintain a node's operation around the new optimal point.

- The game theoretic analysis can be extended to the whitewashing problem. An efficient incentive mechanism needs to be designed which makes whitewashing unprofitable for the nodes in the network.

- As we have already discussed in Chapter 5, we have approximated P2P system to be a linear system for simplifying the mathematical analysis. We can use control theoretic tools like state space analysis to perform non linear analysis of the P2P network. Non linear analysis is closer to real life scenario and should result in more efficient partitioning of link capacity w.r.t. ASZ algorithm.

- The control theoretic analysis can be extended beyond single capacity links. Such kind of links (e.g. ADSL links) will have fixed partitioning between uplink and downlink. The control theoretic analysis can be used by the nodes to share minimum resources corresponding to a fixed download capacity. In this way nodes can receive resources at the minimum cost/upload from the network.

- Finally, verification of the proposed algorithm in real life networks is very important. It would be interesting to see how the ASZ performs when it is deployed together with the existing system, *i.e.*, when a portion of the peers use state of art mechanism for capacity partitioning and the remaining nodes use ASZ strategy.

Appendix A

Laplace to Fourier Conversion of Transfer Function

The Laplace transform of any function $f(t)$ is $\int_0^t f(t)e^{-st}dt = \int_0^t f(t)e^{-(\sigma+jw)t}dt$, where $s = \sigma + jw$. The real part σ adds to the term $e^{-\sigma t}$. This term decays to zero during steady state ($t \to \infty$) and only the jw part which gives the sinusoidal steady state response i.e. $e^{-jwt} = cos(wt) - jsin(wt)$ remains. Life time of every P2P systems is very large, compared with the initial bootstrapping period. Hence, it can be assumed that P2P network reaches steady state. Therefore, σ can be neglected and $s = jw$ can be substituted in the overall transfer function of the control loop.

Bibliography

[1] B. Cohen, "Incentives build robustness in bittorrent," in *Proceedings of the Workshop on Economics of Peer-to-Peer Systems*, 5-6 June 2003.

[2] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Proceedings of International Conference on Distributed Systems Platforms, November*, 2001, pp. 329–350.

[3] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2001, pp. 149–160.

[4] R. Hasan, Z. Anwar, W. Yurcik, L. Brumbaugh, and R. Campbell, "A survey of peer-to-peer storage techniques for distributed file systems," in *Proceedings of International Conference on Information Technology: Coding and Computing (ITCC'05)*, 2005, pp. 205–213 Vol. 2.

[5] I. Foster and A. Iamnitchi, "On death, taxes, and the convergence of peer-to-peer and grid computing," in *2nd International Workshop on Peer-to-Peer Systems*, 2003, pp. 118–128.

[6] Napster, "Peer-to-peer music focused online services," [Available Online] http://www.us.napster.com.

[7] Gnutella, "Gnutella protocol specification," [Available Online] https://www.rfc-gnutella.sourceforge.net/index.html.

[8] J. Risson and T. Moors, "Survey of research towards robust peer-to-peer networks: Search methods," *Comput. Netw.*, vol. 50, no. 17, pp. 3485–3521, 2006.

[9] BitTorrent, “Peer-to-peer file sharing client,” [Available Online] http://www.bittorrent.com.

[10] Freenet, “Peer-to-peer platform for communication and publication,” [Available Online] https://freenetproject.org/index.html.

[11] Kazaa, “Peer-to-peer file sharing client,” [Available Online] http://www.kazaa.com.

[12] J. F. Buford, “Management of peer to peer overlays,” *Int. J. Internet Protoc. Technol.*, vol. 3, no. 1, pp. 2–12, Jul. 2008.

[13] V. Vishnumurthy, S. Chandrakumar, and E. G. Sirer, “KARMA : A Secure Economic Framework for P2P Resource Sharing,” in *Workshop on Economics of Peer-to-Peer Systems*, Jun. 2003.

[14] R. Zhou and K. Hwang, “Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 4, pp. 460–473, 2007.

[15] A. Satsiou and L. Tassiulas, “Reputation-based resource allocation in P2P systems of rational users,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, pp. 466–479, April 2010.

[16] M. Bawa, B. F. Cooper, A. Crespo, N. Daswani, P. Ganesan, H. Garcia-Molina, S. Kamvar, S. Marti, M. Schlosser, Q. Sun, P. Vinograd, and B. Yang, “Peer-to-peer research at stanford,” *SIGMOD Record*, vol. 32, no. 3, pp. 23–28, Sep. 2003.

[17] H. Zhang, Y. Wen, H. Xie, and N. Yu, *Distributed Hash Table: Theory, Platforms and Applications*. Springer Publishing Company, Incorporated, 2013.

[18] S. Lee, S. Zhu, Y. Kim, and J. Chang, “Analysis on Malicious Peer’s Behavior of the P2P Trust Resource Chain Model,” *Software Engineering, Artificial Intelligence, Networking and Parallel/distributed Computing*, vol. 149, 2008.

[19] L. Xiong and L. Liu, “Peertrust: supporting reputation-based trust for peer-to-peer electronic communities,” *IEEE Trans. Knowl. Data Eng.*, vol. 16, pp. 843–857, 2004.

[20] J. Augustine, A. R. Molla, E. Morsy, G. Pandurangan, P. Robinson, and E. Upfal, “Storage and search in dynamic peer-to-peer networks,” in *Proceedings of the Twenty-fifth Annual ACM Symposium on Parallelism in Algorithms and Architectures*, 2013, pp. 53–62.

[21] S. Sen and J. Wang, “Analyzing peer-to-peer traffic across large networks,” *IEEE/ACM Trans. Netw.*, vol. 12, no. 2, pp. 219–232, 2004.

[22] P. K. Gummadi, S. Saroiu, and S. D. Gribble, “A measurement study of napster and gnutella as examples of peer-to-peer file sharing systems,” *SIGCOMM Computer Communication Review*, vol. 32, no. 1, pp. 82–82, 2002.

[23] M. Feldman and J. Chuang, “Overcoming free-riding behavior in peer-to-peer systems,” *SIGecom Exchange.*, vol. 5, no. 4, pp. 41–50, 2005.

[24] C. Buragohain, D. Agrawal, and S. Suri, “A game theoretic framework for incentives in P2P systems,” in *Proceedings of Internation Conference on Peer-to-Peer Computing*, 2003, pp. 48–56.

[25] H. Kung and C. H. Wu, “Differentiated admission for peer-to-peer systems: Incentivizing peers to contribute their resources,” in *Proceedings of Workshop on Economics of Peer-to-Peer System*, 2003.

[26] K. Lai, M. Feldman, I. Stoica, and J. Chuang, “Incentives for cooperation in peer-to-peer networks,” in *Proceedings of Workshop on Economics of Peer-to-Peer System*, 2003.

[27] T. G. Papaioannou and G. D. Stamoulis, “An incentives’ mechanism promoting truthful feedback in peer-to-peer systems,” in *Proceedings of the IEEE International Symposium on Cluster Computing and the Grid*, 2005, pp. 275–283.

[28] P. Golle, K. Leyton-Brown, and I. Mironov, “Incentives for sharing in peer-to-peer networks,” in *Proceedings of the 3rd Conference on Electronic Commerce*, 2001, pp. 264–267.

[29] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, “Secure routing for structured peer-to-peer overlay networks,” *SIGOPS Oper. Syst. Rev.*, vol. 36, pp. 299–314, 2002.

[30] M. Feldman, K. Lai, I. Stoica, and J. Chuang, "Robust incentive techniques for peer-to-peer networks," in *Proceedings of the Conference on Electronic Commerce*, 2004, pp. 102–111.

[31] R. Gupta and Y. N. Singh, "Avoiding whitewashing in unstructured peer-to-peer resource sharing network," *CoRR*, vol. abs/1307.5057, 2013.

[32] S. M. Lui and S. H. Kwok, "Interoperability of peer-to-peer file sharing protocols," *SIGecom Exchanges*, vol. 3, no. 3, pp. 25–33, 2002.

[33] A. Mondal, K. Goda, and M. Kitsuregawa, "Effective load-balancing of peer-to-peer systems," in *Data Engineering Workshop*, 2003.

[34] E. Adar and B. A. Huberman, "Free riding on gnutella," *First Monday*, vol. 5, 2000.

[35] S. Saroiu, K. P. Gummadi, and S. D. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems," in *Proceedings of Multimedia Computing and Networking (MMCN)*, January 2002.

[36] D. Hughes, G. Coulson, and J. Walkerdine, "Free riding on gnutella revisited: the bell tolls," *IEEE Distributed Systems Online*, vol. 6, June 2005.

[37] M. Zghaibeh and F. C. Harmantzis, "Revisiting free riding and the tit-for-tat in bittorrent: A measurement study," *Peer-to-Peer Networking and Applications*, vol. 1, pp. 162–173, Sep 2008.

[38] R. Cuevas, M. Kryczka, A. Cuevas, S. Kaune, C. Guerrero, and R. Rejaie, "Unveiling the incentives for content publishing in popular bittorrent portals," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1421–1435, 2013.

[39] S. Kuhn, "Prisoner's dilemma," in *The Stanford Encyclopedia of Philosophy*, 2017.

[40] X. Kang and Y. Wu, "Incentive mechanism design for heterogeneous peer-to-peer networks: A stackelberg game approach," *IEEE Trans. Mobile Comput.*, vol. 14, pp. 1018–1030, May 2015.

[41] J. Andreoni, "Giving with impure altruism: Applications to charity and ricardian equivalence," *Journal of Political Economy*, vol. 97, pp. 1447–1458, 1989.

[42] C. F. Camerer, *behavioural game theory*. Palgrave Macmillan UK, 2010, pp. 42–50.

[43] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica, "Free-riding and whitewashing in peer-to-peer systems," *IEEE J. Sel. Areas Commun.*, vol. 24, pp. 1010–1019, May 2006.

[44] S. Zhong, Y. Yang, and J. Chen, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *Proceedings of IEEE Infocom*, 2003.

[45] J. Markus, H. Jean-Pierre, and L. Buttyán, "A micro-payment scheme encouraging collaboration in multi-hop cellular networks." Financial Cryptography: International Conference, FC 2003,, pp. 15–33.

[46] L. Buttyán and J.-P. Hubaux, "Enforcing service availability in mobile ad-hoc wans," in *Proceedings of the International Symposium on Mobile Ad Hoc Networking & Computing*, 2000, pp. 87–96.

[47] K. Shin, C. Joe-Wong, S. Ha, Y. Yi, I. Rhee, and D. S. Reeves, "T-chain: A general incentive scheme for cooperative computing," *IEEE/ACM Trans. Netw.*, vol. 25, pp. 2122–2137, Aug 2017.

[48] N. Andrade, M. Mowbray, A. Lima, G. Wagner, and M. Ripeanu, "Influences on cooperation in bittorrent communities," in *Proceedings of the SIGCOMM Workshop on Economics of Peer-to-peer Systems*, 2005, pp. 111–115.

[49] N. Laoutaris, D. Carra, and P. Michiardi, "Uplink allocation beyond choke/unchoke: Or how to divide and conquer best," in *Proceedings of the ACM CoNEXT*, 2008, pp. 18:1–18:12.

[50] A. T1.413-1998, "Network and customer installation interfaces – asymmetric digital subscriber line (ADSL) metallic interface," *American National Standards Institute*, 1998.

[51] M. Gupta, P. Judge, and M. Ammar, "A reputation system for peer-to-peer networks," in *Proceedings of the International Workshop on Network and Operating Systems Support for Digital Audio and Video*, 2003, pp. 144–152.

[52] B. Yu and M. P. Singh, "A social mechanism of reputation management in electronic communities," in *Proceedings of the 4th International Workshop on Cooperative Information Agents IV, The Future of Information Agents in Cyberspace*, 2000, pp. 154–165.

[53] R. Zhou, K. Hwang, and M. Cai, "Gossip trust for fast reputation aggregation in peer-to-peer networks," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 9, pp. 1282–1295, 2008.

[54] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in P2P networks," in *Proceedings of the International Conference on World Wide Web*, 2003, pp. 640–651.

[55] Z. Liang and W. Shi, "Pet: A personalized trust model with reputation and risk evaluation for P2P resource sharing," in *Proceedings of Hawaii International Conference on System Sciences*, vol. 7, 2005.

[56] R. Zhou, S. Song, K. Hwang, and Y.-K. Kwok, "Trusted P2P transactions with fuzzy reputation aggregation," *IEEE Internet Computing*, vol. 9, pp. 24–34, 2005.

[57] H. Mengshu, L. Xianliang, Z. Xu, and Z. Chuan, "A trust model of p2p system based on confirmation theory," *SIGOPS Operating Systems Review*, vol. 39, pp. 56–62, 2005.

[58] R. Gupta, N. Singha, and Y. N. Singh, "Reputation based probabilistic resource allocation for avoiding free riding and formation of common interest groups in unstructured P2P networks," *Peer-to-Peer Networking and Applications*, vol. 9, pp. 1101–1113, Nov 2016.

[59] R. Gupta and Y. N. Singh, "Trust estimation in peer-to-peer network using blue," *CoRR*, vol. abs/1304.1649, 2013. [Online]. Available: http://arxiv.org/abs/1304.1649

[60] A. Goswami, R. Gupta, and G. S. Parashari, “Reputation-based resource allocation in P2P systems: A game theoretic perspective,” *IEEE Commun. Lett.*, vol. 21, p. 1273, 2017.

[61] A. Goswami, G. S. Parashari, and R. Gupta, “Evolutionary stability of reputation based incentive mechanisms in P2P systems,” *IEEE Commun. Lett.*, no. 99, 2017.

[62] eBay, “e-commerce portal,” [Available Online] http://www.ebay.com.

[63] J. R. Douceur, “The sybil attack,” in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, 2002, pp. 251–260.

[64] R. T. B. Ma, S. C. M. Lee, J. C. S. Lui, and D. K. Y. Yau, “A game theoretic approach to provide incentive and service differentiation in P2P networks,” in *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, 2004, pp. 189–198.

[65] Y. Yan, A. El-Atawy, and E. Al-Shaer, “Ranking-based optimal resource allocation in peer-to-peer networks,” in *Proceedings of IEEE INFOCOM 2007*, May 2007, pp. 1100–1108.

[66] C. H. Chiang, W. Liao, and T. Liu, “Adaptive downlink/uplink bandwidth allocation in ieee 802.16 (wimax) wireless networks: A cross-layer approach,” in *IEEE GLOBECOM 2007 - IEEE Global Telecommunications Conference*, Nov 2007, pp. 4775–4779.

[67] D. G. Jeong and W. S. Jeon, “CDMA/TDD system for wireless multimedia services with traffic unbalance between uplink and downlink,” *IEEE J. Sel. Areas Commun.*, vol. 17, no. 5, pp. 939–946, May 1999.

[68] B. H. et al, “Linux advanced routing and traffic control howto,” [Available Online] http://www.lartc.org/.

[69] M. Meo and F. Milan, “A rational model for service rate allocation in peer-to-peer networks,” in *Proc. of IEEE INFOCOM.*, April 2006, pp. 1–5.

[70] G. Iosifidis and I. Koutsopoulos, “Reputation-assisted utility maximization algorithms for peer-to-peer networks,” in *Interntional Workshop on Quality of Service*, June 2008, pp. 20–29.

[71] N. Singha, R. Gupta, and Y. N. Singh, "Resource allocation in peer-to-peer networks: A control-theoretical perspective," [Available Online] https://arxiv.org/abs/1509.07989, 2015.

[72] D. Banerjee, S. Saha, S. Sen, and P. Dasgupta, "Reciprocal resource sharing in P2P environments," in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, 2005, pp. 853–859.

[73] D. Peng, W. Liu, C. Lin, Z. Chen, and X. Peng, "Enhancing tit-for-tat strategy to cope with free-riding in unreliable P2P networks," in *Proceedings of International Conference on Internet and Web Applications and Services, 2008.*, June 2008, pp. 336–341.

[74] D. Stutzbach and R. Rejaie, "Understanding churn in peer-to-peer networks," in *Proceedings of the Conference on Internet Measurement*, 2006, pp. 189–202.

[75] T. Locher, P. Moor, S. Schmid, and R. Wattenhofer, "Free Riding in BitTorrent is Cheap," in *Fifth Workshop on Hot Topics in Networks*, Nov. 2006, pp. 85–90.

[76] Newcastle University, "Exponential moving average," [Available Online] http://www.lorien.ncl.ac.uk/ming/filter/filewma.htm.

[77] R. Gupta and Y. N. Singh, "Reputation aggregation in peer-to-peer network using differential gossip algorithm," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 10, pp. 2812–2823, Oct 2015.

[78] S. Shenker, "Fundamental design issues for the future internet," *IEEE J. Sel. Areas Commun*, vol. 13, no. 7, pp. 1176–1188, Sept 1995.

[79] K. Ogata, *Modern Control Engineering*, 5th ed. Prentice Hall PTR, 2011.

[80] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling tcp throughput: A simple model and its empirical validation," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, 1998, pp. 303–314.

[81] T. F. Abdelzaher, K. G. Shin, and N. Bhatti, "Performance guarantees for web server end-systems: a control-theoretical approach," *IEEE Trans. Parallel Distrib. Syst.*, vol. 13, no. 1, pp. 80–96, Jan 2002.

[82] T. Blevins and M. Nixon, *Control Loop Foundation: Batch and Continuous Processes*. International Society of Automation, 2010.

[83] R. Berber and C. Kravaris, *Nonlinear Model Based Process Control*, ser. Applied Sciences Series. Kluwer Academic, 1998.

[84] W. K. Ho, C. C. Hang, and L. S. Cao, "Tuning of pid controllers based on gain and phase margin specifications," *Automatica*, vol. 31, no. 3, pp. 497–502, Mar. 1995.

[85] F.G.Shinskey, *Process Control Systems: Application, Design and Tuning*. McGraw-Hill Inc., New York, 1996.

[86] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals &Amp; Systems (2nd Ed.)*. Prentice-Hall, Inc., 1996.

[87] M. Osborne, *An Introduction to Game Theory*. Oxford University Press, 2009.

[88] A. C. V. Gummalla and J. O. Limb, "Wireless medium access control protocols," *IEEE Commun. Surveys Tut.*, vol. 3, no. 2, pp. 2–15, Second 2000.

[89] Direct Connect, "Peer-to-peer file sharing client," [Available Online] http://www.dcplusplus.sourceforge.net/tutorial.html.

[90] E. Cerqueira, E. Lee, J.-T. Weng, J.-H. Lim, J. Joy, and M. Gerla, "Recent advances and challenges in human-centric multimedia mobile cloud computing," in *2014 International Conference on Computing, Networking and Communications (ICNC)*, Feb 2014.

[91] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, Oct. 2014.

[92] J. Berry, M. Collins, A. Kearns, C. A. Phillips, J. Saia, and R. Smith, "Cooperative computing for autonomous data centers," in *IEEE International Parallel and Distributed Processing Symposium*, May 2015, pp. 38–47.