Dynamic Internet Pricing with Service Level Agreements for Clients with Multi-ISP Connections

A Thesis Submitted
for the award of the degree of

DOCTOR OF PHILOSOPHY

by
Rohit Tripathi

Under the guidance of
Prof. Gautam Barua

Department of Computer Science and Engineering
Indian Institute of Technology
Guwahati-781039, Assam, India
April 2016

Statement

I do hereby declare that the matter embodied in this thesis is the result of investigations carried out by me in the department of Computer Science and Engineering, Indian Institute of Technology Guwahati, India, under the supervision of Prof. Gautam Barua.

In keeping with the general practice of reporting scientific observations, due acknowledgments have been made wherever the work described is based on the finding of other investigators.

Rohit Tripathi
Research Scholar,
Department of Computer Science and Engineering,
Indian Institute of Technology Guwahati

April, 2016
Guwahati
Assam, India

Certificate

*This is to certify that the work described in this thesis entitled "**Dynamic Internet Pricing with Service Level Agreements for Clients with Multi-ISP Connections**" submitted by **Rohit Tripathi**, a research scholar in the department of Computer Science and Engineering, Indian Institute of Technology Guwahati, for the award of the degree of **Doctor of Philosophy**, is a record of an original research work carried out by him under my supervision and guidance. The thesis has fulfilled all requirement as per the regulations of the institute. The results embodied in this thesis have not been submitted to any other university or institute for the award of any degree or diploma.*

April, 2016
Guwahati
Assam, India

Gautam Barua
Professor,
Department of Computer Science and Engineering,
Indian Institute of Technology Guwahati

Acknowledgment

I thank everyone who has helped or influenced me during the PhD duration. It is however not possible to mention everyone in this limited space. I thank my PhD supervisor, Prof. Gautam Barua for providing guidance and advises. He suggested good research ideas; improvements and extensions of my research ideas; and provided guidance in every phase of PhD.

I thank my institute, IIT Guwahati for providing the necessary resources for doing research. The institute provided me access to online research libraries such as IEEE, ACM, Springer, etc. I was able to download almost all the research papers that I needed for my research. Computer Science and Engineering (CSE) Department, IIT Guwahati provided me access to high performance server for running simulations and this saved a lot of time.

I thank funding agencies for providing me financial support during the PhD duration. MHRD provided me scholarship for five years. After the completion of the scholarship duration, our CSE department arranged scholarship for some months and then my supervisor arranged funding for remaining duration from one of his projects.

I thank my parents for their support and patience during the PhD duration. I thank my friends for making the PhD duration comfortable.

Abstract

One of the most important inventions which has revolutionized the way of living and working of people is the internet. The increase in internet coverage and the decrease in internet access price has resulted in demand for a good internet service. Clients want some guarantee in internet access quality. In this thesis, we present a model in which clients are guaranteed connection and bandwidth and if clients do not get the service they request, the service provider pays a penalty to the clients. We consider a system where each client has access to multiple internet service providers (ISP) and can choose one of them to connect to based on the prices being offered by the ISPs. When a client arrives, an ISP has to decide whether to accept the client, and the price to charge from the client for the duration of its connection. Rejection of a client results in a penalty and delay in getting requested bandwidth while connected also incurs a penalty. We assume a Poisson arrival process with the rate of arrival sensitive to the price being charged. While connected, a client sometimes remain idle and sometimes consumes bandwidth; and both these durations are exponentially distributed. A service provider tries to maximize its income by charging appropriate prices based on its current state and deciding whether to accept more clients or not. Since penalties are imposed, such solutions also automatically balance load among service providers, and so the quality of service to clients improves. We first present a solution to obtain Nash equilibrium between two ISPs. We then present a solution that maximizes the steady state income of service providers. As the computational complexity of this solution is high, we propose two approximate solutions. The solutions are then compared using simulation. Simulation results show that our solutions, including the approximate solutions, significantly improve quality of service of clients and increase the income of service providers as compared to a simple heuristic based solution that otherwise could to be used.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

One of the most important inventions which has revolutionized the way of living and working of people is the internet. It has brought many useful information and services in just a few clicks. The internet provides us useful services like gathering information, payment of bills, social networking, watching videos etc. In recent years, the price of internet access has declined and the number of people using the internet has increased. In fact, many people have become highly dependent on the internet.

Clients have started demanding good, reliable, and predictable internet services. Clients want to have internet access with guaranteed quality of service (QoS). When a client connects to a service provider, he wants that his connection request should be accepted immediately. When a client requests for bandwidth, he should get the requested bandwidth immediately. He should remain connected as long as he wants. The bandwidth given to a client should be taken back only after the client himself releases the bandwidth.

The number of people with internet access and the internet access speeds are different in different parts of the world [2, 3, 1]. In some places high bandwidth internet is available and in some places only limited internet bandwidth is available. Where there is adequate bandwidth available, a flat pricing model is used as it is simple to implement. There are no QoS guarantees in such situations and the need is not felt by users either. However, many countries, especially developing countries, face shortage of internet bandwidth. There is a need to provide methods of access that can properly utilize the limited bandwidth and there is a need to provide QoS.

The growth in the internet demand is faster than the growth in internet bandwidth capacity [54, 4, 33]. So even if places where bandwidth is now in plenty, there will be congestion at times. The growth of Internet of Things means that all devices will be connect to the internet [37]. A lot of research has been done on providing

QoS as real-time services do require QoS [20, 12, 11, 31, 38, 63]. Bandwidth pricing schemes are also used for providing QoS. In [30], the authors have shown that a service provider should charge a price that should be greater than or equal to some limit because if it charges low prices, it will increase internet bandwidth requests and the user experience will go down. The clients want some assurance regarding the connection guarantee. When a service provider charges a low price, the demand for service increases and it increases congestion. The result is that users may not be able to get service when they want and their service experience goes down.

In this thesis, we present a scheme in which multihomed clients are given bandwidth guarantees and dynamic pricing schemes are used by the internet service providers. In the internet, access ISPs have limited bandwidth and with this limited bandwidth they provide service to clients. The details of the architecture are described in section 1.1. We consider multihomed clients and multihoming is explained in section 1.2. The research idea of balancing load by dynamic pricing is given in section 1.3. Our research involves bandwidth pricing and service level agreements. The contribution of the thesis is given in section 1.4. The organization of the thesis is given in section 1.5.

1.1 Internet Architecture

The internet is a large interconnected computer network. The internet consists of multiple autonomous systems [65, 19, 43]. An autonomous system is a network or a set of networks which are owned by a single administration entity such as a business division or a university. An internet service provider either owns one or more autonomous systems or buys bandwidth from autonomous systems to provide service to clients. The service providers from which clients directly buy bandwidth are called access ISPs. In figure 1.1, the big dotted circle represent autonomous systems. The figure consists of vertices (small circle) and edges. The vertices represent routers and edges represent connection between routers.

The internet can be depicted in hierarchical form as shown in figure 1.2 [52]. Based on the services the internet provides, the internet service providers can be divided into tier 1, tier 2, and tier 3 providers. Tier 1 ISPs have access to all major internet regions either directly or by agreement with other tier 1 ISPs. Tier 1 network provides networking service to tier 2 and tier 3 networks. Tier 2 networks take service from tier 1 networks for transferring data and it may as well be connected directly to other tier 2 networks. A tier 3 network, also known as a local ISP (Internet Service Provider) or access ISP, takes services from tier 2 or tier 1 networks and provides service to internet users.

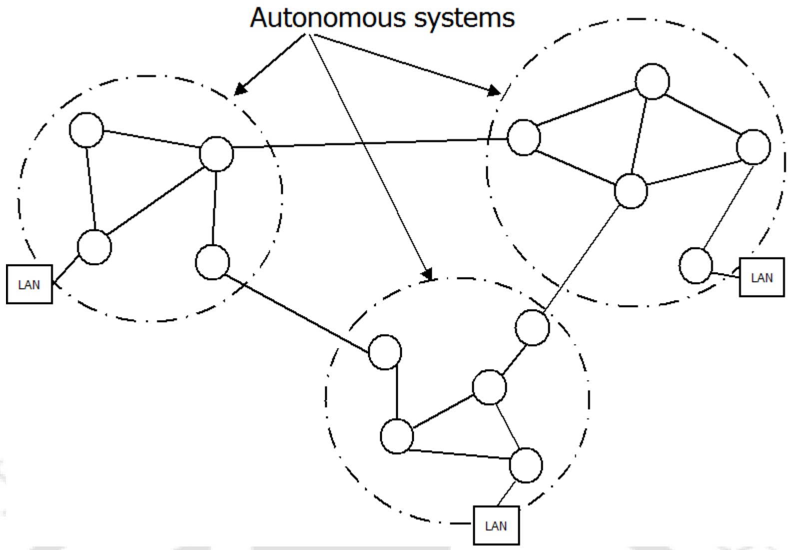

Figure 1.1: Internet autonomous systems

Access ISPs purchase bandwidth from higher level ISPs (tier 1 or tier 2) and provide services to clients. They have limited bandwidth and when their bandwidth is fully consumed, they get congested. Clients connect to the internet randomly and due to this randomness sometimes an ISP has shortage of available bandwidth and sometimes there is excess of available bandwidth. In our model, we consider access ISPs with limited bandwidth that provide service to clients.

1.2 Multihomed clients

A client accesses the internet by connecting a device to an internet service provider or any other network. This connection can be made in multiple ways. A client can use WiFi, cellular network, wired network etc. When a client is connected to multiple networks at the same time, the phenomenon is called multihoming. Multihoming gives multiple advantages to a client. A client can make a choice between multiple service providers for accessing the internet.

Because of decline of internet access prices, a client can subscribe to multiple service providers. Many wireless internet access technologies such as 3G and 4G internet have emerged. The use of smart phones and tablets for accessing the internet has given users the flexibility of choosing among one or more service providers. These devices are now being equipped with multiple "SIM" cards allowing subscription

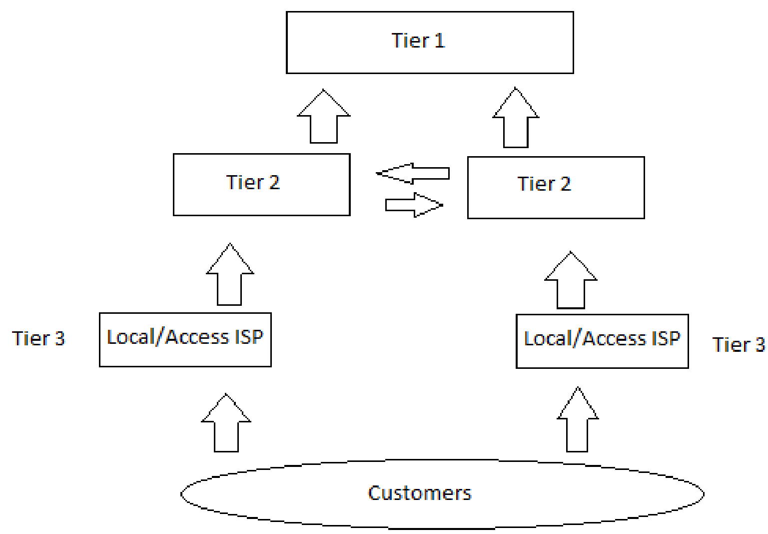

Figure 1.2: Multi-tier internet

to services from more than one provider. Many clients have become multihomed [24, 14]. A client can have wired and wireless internet access. A mobile user can be multihomed because of multiple SIM cards and / or other methods to access network such as WiFi.

There are many researches that present schemes that take advantage of multi homing [51]. In [56], authors focus on managing multihoming clients. In [10], authors mention multihoming architecture for mobile IPv6. In [66], auction based pricing for internet access is mentioned. This auction based pricing also considers situations in which there are multihomed clients.

Our solution also takes advantage of multihoming in which clients connect to appropriate service providers. When a client accesses the internet and he has multiple network interfaces, he can use any of those network interfaces to connect. However, many times one internet service provider may be congested and another service provider may have sufficient available bandwidth. If a client chooses an appropriate network interface, new traffic automatically gets diverted to less loaded service providers. Our research uses this concept and presents solutions which make it possible to divert new clients to less loaded service providers.

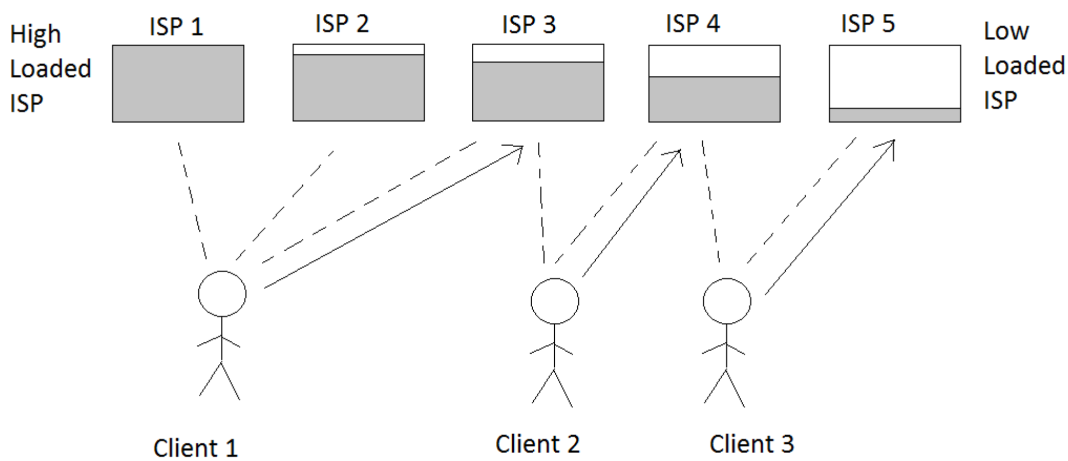

Figure 1.3: Load balancing by Dynamic Pricing

1.3 Motivation

Consider a system of multiple internet service providers and multihomed clients. A client can connect to multiple service providers. Clients randomly request internet access, take service and disconnect. The level of congestion in internet service providers is different at different points of time. Consider a situation in which there are few service providers with different congestion levels. This situation is depicted in Figure 1.3. In the figure, there are five service service providers written as ISP 1 to ISP 5 and these are in descending order of congestion. ISP 1 is the most congested and ISP 5 is the least. As shown in the figure, when client 1 wants to access the internet he can connect to three ISPs. If client 1 connects to ISP 3, client 2 connects to ISP 4 and client 3 connects to ISP 5, this will shift some load from highly loaded ISPs to less loaded ISPs. Our research problem is to present a scheme and solution that enables new arriving clients to connect to appropriate ISPs. The methods involve dynamic pricing and penalties.

1.4 Contribution of the thesis

The main objectives of the thesis are to present solutions that provide quality of service to users accessing the internet by a scheme of dynamic pricing and penalties. A client wants a good internet service and wants to pay less for the same service.

If a client is given the same service by two or more service service providers at different prices, he is likely to take service from a service provider charging less price. We have introduced dynamic pricing in our model. The next thing is to present a mechanism such that a service provider charges high price when it is congested and low price when it has high available bandwidth. Therefore, we introduce penalties in our model. A service provider has to provide good service to clients and if it fails, it has to pay penalties to the clients. If a service provider gets congested, it loses money in the form of penalties. Therefore, a service provider will take appropriate steps before congestion takes place. This is done by increasing the price to reduce arrival rate of clients before and during congestion. Therefore, a consequence of our solutions is that load among the ISPs also get balanced.

We have tried to make our model realistic. We assume Poisson arrival and exponential service times. We assume that a connected client is sometimes idle and sometimes consumes bandwidth. When he is idle, he is connected but not consuming bandwidth. When a client consumes bandwidth, he is termed as being in session. The duration for which a client remains in idle state or in session are exponentially distributed. After some time a client disconnects. We have introduced two guarantees: connection guarantees and bandwidth guarantees. When a client requests for connection, and the service provider refuses the request, it pays a connection penalty. Once a connection is refused, no further requests are entertained to prevent a client getting multiple penalties. When a client requests for bandwidth and has to wait for the bandwidth, the service provider has to pay a penalty for the duration he has to wait and it is the product of the penalty value and the duration for which the client has to wait.

Initially we explored the problem using game theory. We developed a game theoretic solution based on obtaining Nash equilibrium. We found that the solution had high space and time complexities and therefore, we also developed an approximate Nash equilibrium solution. The approximate Nash equilibrium has relatively low complexities but still the complexities were high. In addition the complexities increase by increasing the number of service providers. To handle these limitations, we considered non game theoretic solutions.

The non game theoretic solutions have relatively lower time and space complexities. We developed an accurate solution. But the complexity of an accurate solution is still very high if we assume that a client can request for more than one unit bandwidth. Therefore, we developed approximate solutions that have lower complexities. The solutions have been compared using simulations. We could not find any existing solution with which we could compare our work. Therefore, we have compared our solutions with a simple heuristic. The comparison observes income of service providers and quality of service provided to clients.

1.5 Organization of the thesis

The rest of the thesis is organized as follows.

Chapter 2 In this chapter, we present a survey of existing research in our research area.

Chapter 3 This chapter presents our game theory model and solutions. The objective is to find a Nash equilibrium solution between two service providers. We present an accurate and an approximate solution and these are compared using simulation.

Chapter 4 In this chapter we present our non game theoretic model, describe the problem and present the solution.

Chapter 5 This chapter presents approximate solutions that have lower space and time complexities than the accurate solution (Chapter 4). These are later compared using simulations.

Chapter 6 In this chapter, we compare our non game-theoretic solutions with each other and with a simple heuristic using simulations.

Chapter 7 In this chapter we conclude the thesis and present some future research directions.

Chapter 2

Literature Review

This thesis has internet pricing and service level agreements. We have presented game theory solutions and non game theory solutions. There are a lot of existing researches that are related to our work. In this chapter, we present an overview of these researches. We have presented major existing internet pricing schemes and these are given in section 2.1. The current prices charged by ISPs are given in section 2.2. These pricing schemes alone are insufficient to provide good service to clients. Therefore, service level agreements are introduced and we present existing related researches that are on service level agreements. These are given in section 2.3.

2.1 Different Internet Pricing Schemes

A service provider wants to maximize its profit and a client wants good internet service by paying less money. Internet pricing effects both these needs. An appropriate pricing scheme can satisfy both clients and service providers. This section presents an overview of major pricing schemes.

Existing pricing schemes mainly belong to three types of pricing models [55]. These include static non-competitive pricing models, dynamic non-competitive pricing models and dynamic competitive pricing models. In static non-competitive pricing models, a seller decides prices for his products and clients buys a product based on the price. The price once decided remains static and the clients do not have any other seller as an option on this connection. In dynamic non-competitive pricing model, there is only one seller and the seller charges different prices depending on the load on the network. It is assumed that the arrival rate of clients' is a function of the price charged and it is called the demand function. Different papers assume different demand functions. In a dynamic competitive pricing model, there are multiple sellers and clients' arrival rate is a function of price charged as well as some other parameters. The demand functions have these additional parameters.

A simple pricing scheme is to fix some price and clients pay the price for service. This type of pricing is called static pricing scheme. Static pricing is a pricing scheme that is fixed for a large duration. This pricing scheme does not change dynamically. However, a service provider may observe traffic for a long duration (for example few months) and then decide appropriate prices. Few research works in static pricing are given in section 2.1.1. In static pricing, price does not change when clients' arrival rate randomly changes. Therefore, this type of pricing cannot handle congestion that is due to random change in arrival rate of clients. In addition, when clients' arrival rate increases or decreases, service providers cannot increase or decrease price and earn more money.

The limitations of static pricing can be handled to some extent by dynamic pricing. Dynamic pricing is a pricing scheme in which price can be changed at any time by service providers. Few research works in dynamic pricing are given in section 2.1.2. Few of the good survey papers in this area are [58, 23, 29, 61, 50, 17].

2.1.1 Static Pricing

Few Static Pricing schemes are: Flat Pricing [53], Cumulus Pricing [28], Usage Based Pricing [27], Time of the Day Pricing [26] and Priority pricing schemes [16].

2.1.1.1 Flat Pricing

A simple static pricing scheme is to charge some price for a large duration (for example a month) and allow clients to access unlimited service for this duration. This type of pricing is called a flat pricing scheme [53]. This type of pricing is useful when the service provider has large amount of bandwidth to provide unlimited service to users. Flat pricing gives freedom to users to have unlimited access to service. However, when service providers have limited bandwidth, this type of pricing is not suitable to users.

2.1.1.2 Cumulus Pricing

In cumulus pricing [28], there are three steps:

1. Initially a service provider charges a flat price based on the expected resource consumption by the client. A contract is decided between the service provider and the client that consists of the resource requirement specification of the client and the flat price. This flat price is charged for a specified time period.

2. During the above time period, resource consumption of the client is observed. This is reported to clients as Cumulus Points that indicate when the client has violated the resource requirement specifications.

3. If the overall Cumulus Points exceed some predefined limit, the contract is re-negotiated.

Cumulus pricing assumes that clients will have some mean data requirement and it will not significantly fluctuate. For example, if a client consumes approximately 1 gigabyte of internet data per month and this trend continues, cumulus pricing performs well. However, if a client's data requirement has significant fluctuation such as consuming 500 MB data for the first month, 2 GB in the second month, 100KB in thethird month etc., cumulus pricing will not perform well.

2.1.1.3 Usage Based Pricing

The above pricing schemes are not properly able to handle dynamic internet requests by clients. In addition, a client gets unlimited access by paying some fixed price. However, when multiple clients get unlimited access, there are chances that they will consume a lot of resources. As internet access demand increase, it is not possible to give unlimited access to all the internet clients and this is especially true in the developing countries. A service provider can serve more clients if in some way it is able to reduce the resource requirement of existing clients. This can be done by charging clients based on resource consumption. When clients are charged based on resource consumption, they will consume less resources and therefore pay less. A client may be charged based on the actual time for which he uses service or based on the data transferred. This type of pricing is called usage based pricing [27]. Because of usage based pricing, a service provider can serve many clients and each client will access service only when that access is of sufficient use. In [27] authors have mentioned profit maximization strategies of monopoly ISPs. They have mentioned ways to find optimal prices for bandwidth after considering time varying user demand for bandwidth. Clients have time varying utility for accessing the internet. If their utility is more than the cost of accessing the internet, they will use the service. The paper mentions the relationship between flat price, usage price and customer's utility and how the ISPs should charge prices to maximize income.

2.1.1.4 Time of the day Pricing

Usually, the rate at which clients access the internet changes during a day. Sometimes the resource requirement of clients is very high and sometimes it is very low. Some

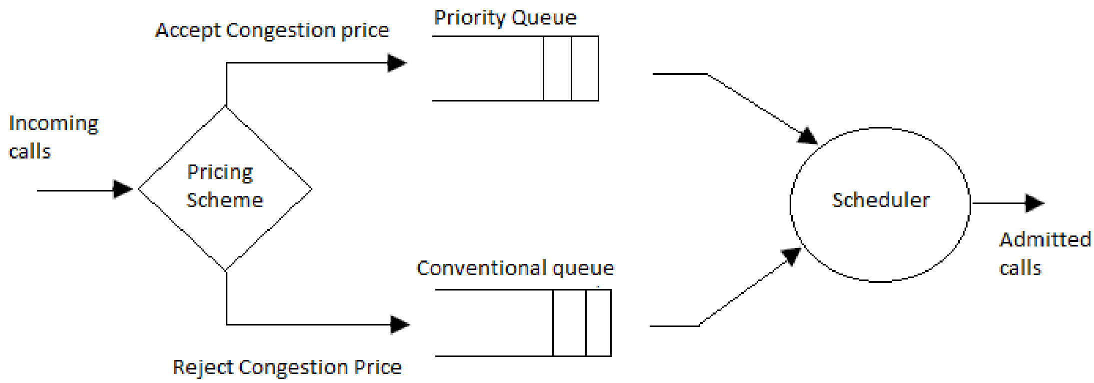

Figure 2.1: Dynamic Pricing with Alternatives

authors, have suggested pricing based on time [26]. In [26], authors mention that the peak demand of users concentrate nearly at the same time. Existing pricing schemes were unable to handle it. They present both static and dynamic time dependent pricing. They propose a framework for time dependent pricing that can be used by ISPs. They argue that their framework is computationally tractable.

2.1.1.5 Priority Pricing Scheme

The urgency of internet access is different for different people at different times. Priority pricing schemes give clients access to service when clients are in urgent need for internet access and they have to pay more. In priority pricing schemes, a service provider maintains classes of service. A client can choose the class of service that he wants. A client may get good service by paying a high price or get poorer service during congestion by paying a low price.

In [16], a priority pricing scheme, Paris metro pricing, is proposed. The scheme provides multiple classes of service to clients. There will be different prices for different classes. Most people will prefer to go for the cheaper classes. People who pay more money go for the higher classes. Since there are fewer clients in the higher class, clients of the higher class get better QoS. If a higher class and a lower class are allotted the same bandwidth, clients in the higher class get better quality of service because fewer clients are there in the higher class. A similar pricing scheme is mentioned in [62] for mobile calls. It is given in figure 2.1. In this scheme there are two connection queues. The first queue is the conventional queue and the second queue is a priority queue. When a user dials a number, if the network is not congested, his request will be placed in the conventional queue. If the network is congested and a user dials a number, he will be notified the approximate time for which he has to wait and if the user wishes, his request may be placed in the priority queue but the user has to pay more. Based on the preference, the user may decide to place his call in the priority queue or wait in the conventional queue or make a call after some time.

2.1.2 Dynamic Pricing

There are many types of dynamic pricing schemes. Few classifications are given in [50, 23]. Our research is on dynamic congestion pricing. Few Dynamic Pricing Schemes are Priority Pricing [40], Auction Based Pricing [47], Congestion Pricing [42] and Game Theoretic Pricing 2.1.2.4.

2.1.2.1 Priority Pricing

Dynamic pricing can also have priorities. The purpose of priority pricing is the same as already mentioned in the static priority pricing schemes in that these schemes give clients access to service when clients are in urgent need for internet access and they have to pay more. A service provider announces different priority classes to a client. Users can use any priority class according to their requirement. These priority classes can be modified depending on networks conditions and bandwidth demand. In [40] authors have mentioned a scheme in which a service provider specifies service classes to clients. These service classes have their own service quality and a client can specify his preference. The paper mentions three priority classes and these classes have prices and QoS guarantee when the service provider gets congested. The prices of priority classes can be modified by a service provider at any time.

2.1.2.2 Auction Based Pricing

In auction based pricing, buyers or sellers can bid for price of a product and based on the bid an appropriate buyer or seller is chosen. Auctions are especially useful when the price of a product is not decided and auctions are used to find the appropriate price for the product. It is assumed that when a product is sold using auctions, the buyers are ready to pay a price that is less than or equal to the actual value of the product according to them. Therefore, the product will be sold at a high price. Auctions are used to sell wireless spectrum to service providers [18]. One important advantage of auctioning is that the bidder that has the highest value for the product gets the product. This means that the buyer who is likely produce maximum profit from the spectrum gets the spectrum and in this way spectrum is likely to be appropriately utilized.

[41] provides description of different types of auctions. Auction is also applied for bandwidth pricing [66, 47]. In [47], a customer is given resources through bidding. Clients have a choice to connect to multiple service providers and the service providers do bidding. There can be negotiations between service providers and clients. Auctions cannot be held very frequently and so they are used in scenarios where prices, once decided through an auction, remain static over a relatively long period of time.

2.1.2.3 Congestion Pricing

In congestion pricing, prices of bandwidth is based on the congestion level of service providers. When high price is charged, some clients will not access service and this reduces congestion and gives higher profit to service providers during congestion. Similarly when congestion is not there, low price is charged and it increases clients to access service. Many researchers have presented schemes with congestion pricing and some of these are given below.

In [42, 64], authors have assumed that a client can connect to only one service provider. The mean arrival rate of clients is a function of price charged and a high price discourages a client to connect. There is dynamic pricing and dynamic quality of service. In [64], when a client wants to access service he has to make a choice between multiple service classes. A service provider dynamically decides service classes based on available resources. Each service class has two components: bandwidth and cost. If a client does not find any service class suitable, he does not connect. When a client does not choose any service class a penalty is incurred by the service provider. A service provider has to maximize its income by attracting more clients and offering appropriate service classes to clients depending on the current available bandwidth. The solution method uses Bellman equations.

In [42], authors have considered a model in which a service provider has limited bandwidth and clients make service requests. Calls belong to a fixed number of M classes. Calls arrive according to Poisson processes and connection durations are exponentially distributed. Mean connection duration for the ith class is $\frac{1}{\mu_i}$. There is a known demand function $\lambda_i(u_i)$ which represents the mean arrival rate of class i and it is a function of price charged u_i. If service provider increases price, the mean arrival rate declines. A similar model is mentioned in [64]. Similarly in [46], dynamic pricing scheme is applied for multimedia networks.

In [15], authors have considered a system with a single server queue and arrivals belonging to two classes. The service provider has to do dynamic pricing to maximize profit. When arrivals wait, there is holding cost which is incurred by the queue owner. The holding cost depends on the time for which arrivals have to wait. They have used Markov decision processes to find properties of an optimal pricing policy.

There are many congestion pricing schemes to handle congestion in wireless cellular networks [9]. In [39] a scheme is mentioned where a service provider divides users into k quality of service(qos) classes. The difference between qos classes is call admission probability. The purpose of the scheme is to provide the same quality of service to users even at the time of congestion. The service provider provides dynamic pricing to the qos classes to maintain the call admission probability. When congestion occurs, the price is increased to discourage clients from connecting and the clients who connect still get the same quality of service.

Another method to provide quality of service to clients is to provide congestion discounts. In [34], the author present a scheme in which service providers try to shift traffic from busy periods to non busy periods. When a client tries to take service and there is congestion, the service provider gives him an offer called a congestion discount. A client may accept the offer by deciding to access service with the discount when there is no congestion or reject the offer by continuing service. A service provider tries to minimize congestion discounts and also tries to satisfy all clients.

2.1.2.4 Game Theory in pricing

In access networks, there are usually two types of entities. These are service providers and clients. They have their own objectives. Their interaction has been also modelled using game theory [59]. Game theory is a study of decision making in which there are a set of players trying to achieve some objective or reward, and strategy of a player affects the reward of other players. The objective of game theory is to find an appropriate strategy for a player.

Many papers have considered this as game theory problems [58]. Based on players, game theory approaches are classified into three broad categories: service providers vs service providers, clients vs service providers, and clients vs clients. The games are further classified as cooperative or non-cooperative. In a cooperative game, the players may cooperate with other players to achieve their objective. In a non-cooperative game, there is no cooperative between the players and each player decides his strategy independently.

An example of a game between ISP and client is given in [32]. The scheme mentioned in [32] is that the complete day will be divided into time slots and users have their own preference of using the internet in the time slots. The service provider has limited capacity and if more users access the internet in some time slot, he will increase the price for the time slot. Users have different utility in accessing the internet in different time slots. The paper talks about a game theory approach in which the users and the service provider try to maximize their revenue and reach equilibrium. A user knows the prices charged by the ISP and chooses proper time slots to maximize his utility. As already mentioned, a client wants a good service in terms of quality, etc. by paying least possible money and a service provider tries to maximize its income. In the above papers it is assumed that a client will decide not to access internet service in particular time slot when price is high. The problem is that price is affecting a client's decision to access the internet. This scheme does not mention any load balancing among ISPs.

In many cases a client has the option to connect to multiple service providers. He has to make a choice to find the best service provider. In [44, 45], a non-cooperative game between service providers and clients is described. A client has the option to connect to multiple service providers. A service provider tries to maximize his income using dynamic pricing. A client uses multiple criteria to choose an appropriate service provider. The criteria can be price, service provider's reputation, mobility etc.

2.2 Current Practice of ISPs

[49] presents a survey of past and current pricing schemes used by internet service providers. Traditionally, many ISPs have used flat pricing schemes. A client is charged some monthly fee and the client can transfer unlimited data and access the internet for unlimited time. Many variation of flat pricing is prevalent in the world. For example, a service provider may allow clients to access unlimited service up to some limit. This is called "flat up to a cap" pricing. Many internet service providers of India offer 3G internet schemes that are "flat up to a cap" schemes. They also offer usage based pricing for clients who are not subscribed to any 3G internet plan. Another internet plan that is popular is "Cap then metered" plan. In this internet scheme, a client is charged a flat price up to some limit of data transferred and after this limit, usage based price is charged. Another variation of service provided by many internet service providers in India is that a user can have limited access at high bandwidth up to a limit but once the limit has been reached, the bandwidth provided to the user will go down but there will be no extra charge. Another data plan is "shared data plan" in which a client is allowed to share its data across all his devices. Roger has been offering this plan in Canada [5]. A simple time of the day pricing is two period pricing in which day is divided into two parts and price in the two parts are different. BSNL offers unlimited download at timing 2 am to 8 am in

Table 2.1: Access plans of an ISP

Plan period	Data Limit	Price	Price beyond data limit
1 month	0	0	4p/KB
1 month	1GB	250	4p/KB
1 month	2GB	450	4p/KB
1 month	6 GB	950	4p/KB
1 month	10 GB	1500	4p/KB

many of its broadband plans. This is an example of time of the day pricing.

In many developing countries such as in India and Africa, service providers have used innovative methods to reduce cost. Some of these methods are mentioned in [6] and these apply only to voice calls. The African service provider MTN provides dynamic congestion pricing in which cost of a call is decided every hour based on the level of usage. This method reduces voice traffic during peak hours of a day and many users make voice calls when charges are low. Similarly, Uninor in India is providing location dependent pricing for voice calls [48]. In this scheme, a client is offered discounts that is based on the level of congestion in the region from where he is making a voice call.

Table 2.1 gives the access plans of an ISP in the Indian market in January 2015. As far as quality of service is concerned, their web site states: "Actual Internet speed would depend on multiple factors like device, web pages accessed, time of day, number of simultaneous users etc.". So there are no guarantees of service quality. There is no mention of even the maximum speed that their network supports making the user believe it is the maximum the technology being used supports (3G, 2G, etc.). This model, which is typical of models available in the Indian market gives an assured income to the ISP per client. The price beyond the data limit is very high (works out to Rs. 4000/- for 1 GB) and this is another source of income from unsuspecting and naive users. Users are therefore demanding service with service level agreements guaranteeing minimum levels of speed and delay.

2.3 Pricing with Service Level Agreements

A service provider has the objective to maximize its profit. It is expected that when a service provider has to make a choice between profit and quality of service for its client, a service provider will select profit. Therefore there should be some way to force a service provider to provide good service. This is done using service level agreements [36] between service providers and clients. A service level agreement is an agreement between service provider and clients that specifies the services the service provider will provide to clients. Service level agreements may define some penalties those are paid to clients if promised service is not provided. A service provider tries to minimize these penalties by providing good service.

A lot of research has been done in internet pricing. Dynamic pricing schemes can be used to divert new arriving clients to appropriate service providers. Our view is that dynamic pricing alone is not sufficient to provide good service to clients. An appropriate scheme should put some pressure on service providers to provide good service. Service level agreements with penalties should be used. The concept of service level agreements with penalties is not new. In [21], authors have proposed a scheme which has some features similar to our scheme. The quality of service being guaranteed is bandwidth and penalties are imposed when the guarantees cannot be met. A client negotiates a rate for some assured bandwidth. When the client does not want to use some of the assured bandwidth, he can request the service provider to reduce the assured bandwidth and thereby pay a discounted rate for the bandwidth used. When a client wants that guaranteed bandwidth back, the service provider has to give it to him. In case it is unable to do so, a penalty is imposed. Although they mention delay dependent penalty, they use a fixed penalty to reduce complexity. Since prices are fixed at admission time, and there are no connection guarantees, the policy is to do appropriate admission control to maximize income. The problem is then to find a suitable trunk reservation scheme where spare capacity is kept to handle bandwidth return requests, and admission is done if sufficient bandwidth can be reserved for the incoming client. They propose a heuristic to find such a reservation scheme. Their scheme is more suitable for leased connections or for VPNs, but not suitable for a retail environment like ours. Similarly, in [35], different penalty functions are mentioned but the the authors did not present any analytical solution.

Each of these proposals has a different model of user interaction and currently it is not clear what will be an acceptable user model. To the best of our knowledge, no proposal presents an appropriate solution when a user has the choice of connecting to competing providers and no proposal talks about providing connection guarantees to users with penalties in case a connection is refused. Our view is that a realistic but simple model should be used to cater to retail internet users. Such a model should have connection guarantees and bandwidth guarantees with penalties in case of failure to provide these guarantees. Because of the complexity of the problem, we restrict ourselves to only providing bandwidth guarantees and no other guarantees such as response time. Finally, users must have some degree of certainty in pricing and a completely dynamic pricing scheme is unlikely to be acceptable to users. So we propose a dynamic pricing scheme where the price is fixed at the time of admission into the system.

Chapter 3

Nash Equilibrium

Consider two internet service providers: service provider 1 and service provider 2. These service providers will try to maximize their profits. Service provider 1 will choose some strategy that maximizes its profit and then service provider 2 will respond by finding a strategy that maximizes its profit. Again service provider 1 will modify its strategy because service provider 2 has modified its strategy. This process continues till neither service provider 1 nor service provider 2 can increase their profit by modifying their strategies. Thus, we expect that when each service provider tries to maximize its profit, the service providers are likely to reach an equilibrium such that each service provider cannot increase its profit by modifying its strategy given that the other service providers have fixed their strategy. This situation is called Nash equilibrium. In this chapter, we present a Nash equilibrium solution for our research problem. The research problem and solution are described in details in section 3.1. Our Nash equilibrium solution has high space and time complexities. Therefore, we present an approximate Nash equilibrium solution in section 3.2. The accurate and the approximate solutions are compared by running the solutions in section 3.3. In section 3.4, we present a discussion of the existence of Nash equilibrium. In section 3.5, we present limitations of our Nash equilibrium solutions.

3.1 Research Problem and solution

We assume that there are two service providers and each client can connect to either of the two ISPs. Each client requires exactly one unit of bandwidth (this is to reduce the size of the solution space). As shown in Figure 3.1, when a client tries to connect, each of his service providers announces a price from a finite set of possible prices. The client connects to the ISP offering the lower price. If both offer the same price, then

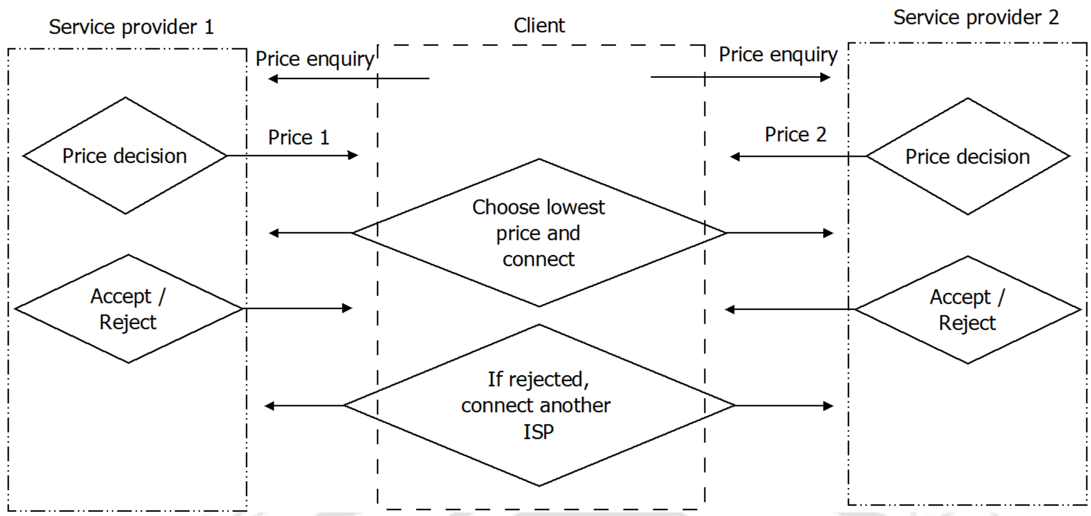

Figure 3.1: The connection process

the client randomly chooses one of the ISPs. If an ISP decides to reject a request, it offers the highest possible price. This is done to avoid paying a penalty without a client deciding to connect to it. So, if ISP 1 wants to reject and ISP 2 is offering the highest price, the client will see both offering the highest price and it is only with a 50 % probability that ISP 1 will have to pay a penalty as the client may connect to ISP 2 and get accepted. So only after an ISP is chosen, will the ISP accept or reject the connection. In the latter case, the ISP will pay a penalty to the client. The client will then connect to the second ISP. Here again there may be an acceptance or a rejection.

As already mentioned, we assume that a service provider can charge prices from a finite set. It is because the prices charged by internet service providers depend on market conditions. A client buys a sim card to subscribe to a service provider. If a service provider charges too high a price, its reputation decreases and its clients will then subscribe to other service providers. If a service provider charges too low a price, other service providers may also have to charge low prices and all the service providers will be in loss. Therefore, a service provider can only charge a price from some range. If two service providers charge slightly different prices, the difference will be too small to be considered by clients. Therefore if a service provider wants to attract (or repel) clients, a service provider has to charge a price which is sufficiently low as compared to the price charged by other service providers. Therefore, in our model we assume a finite set of discrete prices.

The arrival of clients is modelled as a Poisson process[8]. λ is the mean arrival rate

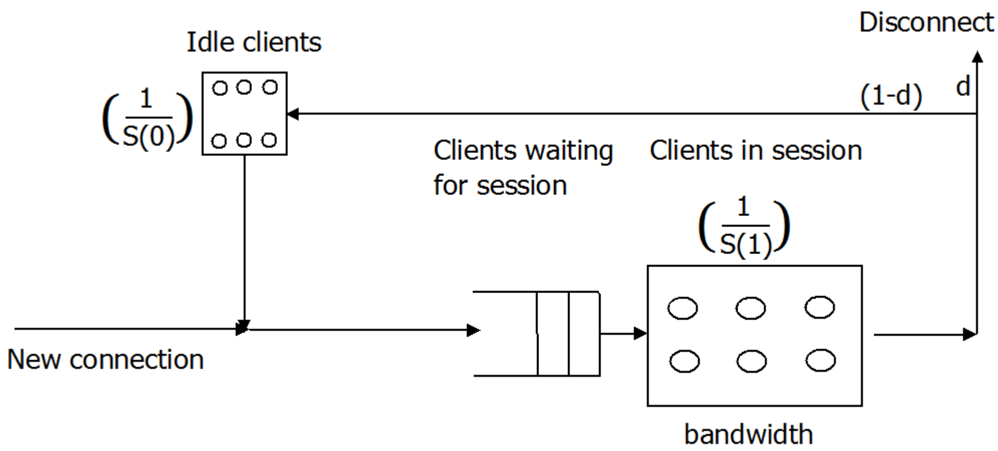

Figure 3.2: Connected clients

of clients. After connection, a client sometimes remains idle and sometimes consumes bandwidth and these durations are both exponentially distributed. When a client consumes bandwidth, it is termed as being in session. When a client requests for bandwidth but he has to wait, it is termed as waiting for session. When a connected client is neither in session nor waiting for session, he is in an idle state. A client moves out of idle state when he needs to enter the session state. He may have to wait to get into the session state. If so, he gets a penalty from the ISP for the duration he has to wait. He is then waiting for a session. Figure 3.2 illustrates the scheme. After a session ends, with probability d the client exits the system. Otherwise, he goes into the idle state.

3.1.1 Symbol Declaration

The state of a service provider is represented by a set of integers (m, n) where m is the number of clients connected and n is the sum of the number of clients in session and the number of clients waiting for a session. It is shortened to a single symbol s. The definitions of all the symbols used are given in Table 3.1.

Table 3.1: Symbol declaration

Symbol	Definition
$state$	It represents the state of a service provider which is the set (m, n). As a shorthand, it is represented by s, $s1$, or $s2$ below.
m	Number of connected clients of the service provider being considered.
n	Sum of the number of clients in session and the number of clients waiting for session of the service provider being considered.
m_{max}	Maximum possible value of m.
d	Probability that a client disconnects immediately after closing a session.
$\frac{1}{S(i)}$	If i is zero, it is the mean duration for which a client remains idle and then requests for bandwidth and if i is 1 it is the mean duration for which a client consumes 1 unit bandwidth and then releases bandwidth.
$price()$	An array consisting of possible prices a service provider can charge. The values are in ascending order. $price(0)$ is the least price.
$C(i, s)$	The decision service provider i takes when the state of a service provider is (s). When $C(i, s)$ returns zero or less, the client is accepted. When $C(i, s)$ returns 1, the client should be rejected. When $C(i, s)$ is zero or less a new arriving client is charged the price $price(-C(i, s))$.
$C_{checked}$	A set that is used to store collection of decisions $(C(i, s))$.
$Pr(s1, s2)$	Probability that the state of service provider 1 is $s1$ and the state of service provider 2 is $s2$.
$Pr_t(s1, s2, t)$	Probability that the state of service provider 1 is $s1$ and the state of service provider 2 is $s2$ at time t. Other things are the same as in $Pr()$.
λ	The mean arrival rate of clients.
B_1	Bandwidth of service provider 1.
B_2	Bandwidth of service provider 2.
T	Number of prices.
D	Expected total data transferred by an arriving client.
$I(i)$	Income per unit time at steady state of service provider i.
$P(0)$	Penalty per unit time paid to a client waiting for bandwidth.
$P(1)$	Penalty paid to the client being rejected.
$Pri(m, n)$	Finite population steady state probability when the population is m and $(m - n)$ clients are idle. In other words, n is the sum of the number of clients in session and the number of clients waiting for session
$E(i, m)$	The expected number of clients in session for service provider i when m clients are connected and so the population is finite
$Waiting(i, s)$	The expected number of clients waiting for session for service provider i when state of service provider is s

3.1.2 Nash Equilibrium solution method (Accurate Solution)

Our solution finds a Nash equilibrium if it exists. Our payoff function (or the criterion for deciding if a solution is the best) is the expected income of a service provider at steady state. We now show an example of our model in which Nash equilibrium exists. Consider two service providers who have large amounts of bandwidth. So there will be no rejections and so no penalties. If the mean arrival rate of clients and connection durations are very small, the service providers will always charge the lowest possible price from arriving clients. This is because charging a higher prices will result in clients going to another service provider who is charging a lower price. This is an example of Nash equilibrium.

Solution steps

The method of finding a Nash equilibrium has the following steps. We use variable x and y to denote the two service providers. To begin with, let x be service provider 1 and y be service provider 2.

1. Start with service provider x. The set of decisions it needs to take are the values in the array $C(x, s)$ as each element is indexed by the state of the system and the value is the price to be charged or a decision to reject the client. Assume some starting values for each entry of $C(x, s)$. Initialize $C_{checked}$ as an empty set.

2. Assume some values for each element of $C(y, s)$ for service provider y.

3. Find the income per unit time at steady state of service provider y, given the current values in $C(x, s)$ and $C(y, s)$. The method is given in section 3.1.3.

4. Repeat step 3 for all possible sets of values of $C(y, s)$.

5. Choose that version of $C(y, s)$ which gives the maximum expected income. Assume that service provider y takes this decision.

6. Now interchange the roles of x and y and let the decision of service provider y found in step 5 become the decision of service provider x in step 2.

7. Let $C_{checked}$ represent collection of decisions that have already been considered in step 2. Store $C(y, s)$ in $C_{checked}$ because $C(y, s)$ has already been tried.

8. Check if $C(x, s)$ already exists in $C_{checked}$. If it exists, choose a new $C(x, s)$ that does not exist in $C_{checked}$. If no new $C(x, s)$ exist, this means all possible decisions have been checked for Nash equilibrium and so Nash equilibrium does not exist. Repeat step 2 to step 8 till a Nash Equilibrium condition is satisfied or till all possible values of $C(y, s)$ have been considered. Nash Equilibrium will be reached when, given the decision of service provider 1 is a, the decision with maximum income per unit time of service provider 2 is b and given the decision of service provider 2 is b, the decision with maximum income per unit time of service provider 1 is a.

3.1.3 Finding income per unit time at steady state

Let D be the expected number of units of data transferred by an arriving client. We assume that a client does not disconnect when he is waiting for a session. Therefore the average data transferred by an arriving client does not depend on congestion. Let $Pr(s1, s2)$ be the steady state probability that service provider 1 is in state $(s1)$ and service provider 2 is in state $(s2)$. Let $I(1)$ be the income per unit time at steady state of service provider 1 and $I(2)$ be the income per unit time at steady state of service provider 2. The income per unit time at steady state is found by multiplying the steady state probability of being in a particular state by the income at that state and then this is added for every possible state. The method to find the steady state probability is shown in section 3.1.4.

The equation for finding $I(1)$ is given below as Equation 3.1 and its explanation is as follows. An arriving client connects to service provider 1 if it does not reject the client, and it charges less than 2 or 2 rejects the client (this is the first term in the first set). If provider 1 does not reject the client and it charges the same price as provider 2, then with half the probability the arriving client goes to service provider 1 (term 2 in first set). A penalty of $Waiting(1, s1) \times P(0)$ (the second term) has to be incurred and this will be non-zero if the number of clients in session (n) is more than B_1. The formula for $Waiting(1, s1)$ is given in section 3.1.5. Finally, a penalty of $P(1) \times \lambda$ has to be incurred if both providers reject the client and with probability half if provider 1 rejects the client and provider 2 charges the highest possible price.

$$
\begin{aligned}
I(1) \;=\; & \sum_{s1,s2} Pr(s1,s2) \times \{ \\
& \begin{cases} \lambda \times D \times price(-C(1,s1)) & C(1,s1) \leq 0 \\ & ,\{C(2,s2) < C(1,s1) \; or \; C(2,s2) = 1\} \\ \frac{\lambda}{2} \times D \times price(-C(1,s1)) & ,C(1,s1) \leq 0, C(2,s2) = C(1,s1) \\ 0 & ,otherwise \end{cases} \\
& -Waiting(1,s1) \times P(0) \\
& - \begin{cases} P(1) \times \lambda & ,C(1,s1) = 1, C(2,s2) = 1 \\ P(1) \times \frac{\lambda}{2} & ,C(1,s1) = 1, C(2,s2) = -(T-1) \,\} \\ 0 & ,otherwise \end{cases}
\end{aligned} \tag{3.1}
$$

In a similar way, the value of $I(2)$ can be found.

The method of finding D is as follows. As shown in figure 3.2, when a client connects, he requests for bandwidth for some duration. The expected time for which a client remains in session is $\frac{1}{S(1)}$. The probability that a client disconnects immediately after releasing bandwidth is d. The probability that a client again requests for bandwidth after remaining idle for some time is $(1-d)$. Therefore the total time that a client spends on consuming bandwidth is

$$D = \frac{1}{S(1)} \times \left(1 + (1-d) + (1-d)^2 + (1-d)^3 ...\right)$$

This give us

$$D = \frac{1}{S(1)} \times \left(\frac{1}{1-(1-d)}\right)$$

On simplification, the value of D is :

$$D = \frac{1}{d \times S(1)} \tag{3.2}$$

3.1.4 Finding steady state probability

Let $Pr_t(s1, s2, t)$ be the probability of service provider 1 in state $s1$ and service provider 2 in state $s2$ at time t. Let dt be an infinitely small time such that the probability of two or more events to take place in time dt is negligible. The method of finding $Pr_t(s1, s2, t+dt)$ is as follows. The probability of service providers being in some state at time t, $Pr_t(s3, s2, t)$ is multiplied by the probability of state transition in time dt to state $s1$ from state $s3$.It is similar for the case when the state is $Pr_t(s1, s3, t)$. As only one event can occur in time dt, we need to only consider single state changes. If the current stae is $Pr_t(s1, s2, t)$ then it is multiplied by the probability of no state change in time dt. Probability of no state change in time dt is (1-probability of state change in time dt). All these terms are summed for each possible value of $Pr_t(..., t)$ and the result is $Pr_t(s1, s2, t + dt)$. $Pr_t(s1, s2, t + dt)$ is written in terms of $Pr_t(..., t)$ as given below.

$$\begin{aligned} Pr_t(s1, s2, t + dt) = \\ \sum_{s3, s1 \neq s3} \{ Pr_t(s3, s2, t) \times \\ \text{(Probability of state change from s3 to s1 in time dt) } \} \\ + \sum_{s3, s2 \neq s3} \{ Pr_t(s1, s3, t) \times \\ \text{(Probability of state change from s3 to s2 in time dt)} \} \\ + Pr_t(s1, s2, t) \\ (1 - \sum_{s3, s1 \neq s3} \{ \\ \text{(Probability of state change from s1 to s3 in time dt) } \} \\ - \sum_{s3, s2 \neq s3} \{ \\ \text{(Probability of state change from s2 to s3 in time dt)} \}) \end{aligned} \tag{3.3}$$

Probability of state change from $s3$ to $s1$ and $s3$ to $s2$ depends on the event due to which the change took place and it is the rate of change multiplied by dt. If the number of events is two or more, the probability is zero because the time dt is so small that the probability of more than one event is negligible. If one event can change state from $s3$ to $s1$ or $s3$ to $s2$, this probability is the event rate multiplied by dt.

If the event is an arrival then it is $\lambda \times dt$ when all clients come to service provider 1 or $\frac{\lambda}{2} \times dt$ when half the clients come to service provider 1, otherwise it is zero. Which alternative will hold depends on the values of the decision matrix C(), as can be seen in the equation above. If $s3$ is $(m3, n3)$, $n3$ is less than B_1 and the event is a departure, the departure probability is $S(1) \times d \times n3 \times dt$. The probability of a client opening a session is $S(0) \times (m3 - n3) \times dt$ and the probability that a client closes a session and then becomes idle is $S(1) \times (1-d) \times n3 \times dt$. If $n3$ is greater than B_1, it is replaced by B_1. For example the departure probability becomes $S(1) \times d \times B_1 \times dt$. The same way $s3$ to $s2$ probabilities are written. Similarly probability of state change from $s1$ to $s3$ means the probability of state change from $s1$ to any other state. If $s1$ is (m, n), it is the sum of three terms: $\lambda \times dt$ or $\frac{\lambda}{2} \times dt$ or zero, $S(0) \times (m-n) \times dt$ and $S(1) \times n \times dt$ or $S(1) \times B_1 \times dt$.

The equation is rearranged such that $\frac{Pr_t(s1,s2,t+dt)-Pr_t(s1,s2,t)}{dt}$ comes to the left hand side. Probability of state change from $s1$ and $s2$ to $s1$ and $s2$ when divided by dt, becomes the rate of state change from $s1$ and $s2$ to $s1$ and $s2$. Since we are finding a solution for the steady state, the left hand side becomes zero and time is removed from the equation. The following equation is then obtained.

$$
\begin{aligned}
0 \quad = \quad & \sum_{s3, s3 \neq s1} \{Pr(s3, s2) \times (\text{Rate of change of} \\
& \text{state of service provider 1 from s3 to s1})\} \\
& + \sum_{s3, s3 \neq s2} \{Pr(s1, s3) \times (\text{Rate of change of} \\
& \text{state of service provider 2 from s3 to s2})\} \\
& + Pr(s1, s2) \times \\
& (- \sum_{s3, s3 \neq s1} \{\text{Rate of change of state of service} \\
& \ \text{provider 1 from s1 to s3}\} \\
& - \sum_{s3, s3 \neq s2} \{\text{Rate of change of state of service} \\
& \ \text{provider 2 from s2 to s3}\})
\end{aligned}
$$

The expanded equation is given below. Let state $s1$ be the state (m, n). And the state $s2$ be the state (m', n'). The expanded equation contains three sets of terms and these are given separately inside three sets of curly brackets: $\{\}$. The first set contains all the possible transitions of service provider 1 to the present state (m, n). The second set contains all the possible transitions of service provider 2 to the present state (m', n'). The third set represents the possibility of both the service provider being in the present state (m, n, m', n') and rate at which any service provider leaves its present state.

The explanation of the first set is as follows. $Pr(m-1, n-1, m', n')$ is the probability that service provider 1 is in state $(m-1, n-1)$ and service provider 2 is in state (m', n'). It is multiplied by the mean arrival rate with which an arrival takes place for service provider 1. This rate is λ when the arrival is accepted by service provider 1 (given by $C(1, m-1, n-1) \leq 0$) and the price charged by service provider 2 is more than the price charged by service provider 1 (given by $C(1, m-1, n-1) > C(2, m', n')$) or service provider 2 rejects an arriving client (given by $C(2, m', n') = 1$). This arrival rate is $\frac{\lambda}{2}$ when the arrival is accepted by service provider 1 (given by $C(1, m-1, n-1) \leq 0$) and the price charged by both the service providers are the same (given by $C(1, m-1, n-1) = C(2, m', n')$). Similarly, $(m+1, n+1, m', n')$ is the state at which service provider 1 had an extra connected client and the rate at which this client departs is $S(1) \times (n+1) \times d$ when $n+1$ is greater than the bandwidth B1 of service provider 1 or $S(1) \times B_1 \times d$ otherwise. $(m, n+1, m', n')$ represents the state where service provider 1 has one extra client in session and this client closes its session with the rate $S(1) \times (n+1) \times (1-d)$ when $n+1$ is greater than the bandwidth of service provider 1 or $S(1) \times B_1 \times (1-d)$ otherwise. $(m, n-1, m', n')$ represents the possibility that service provider 1 has one client less in session (is idle) and this client opens a session with the rate $S(0) \times (m-n+1)$.

The second set of terms are identical to the first set of terms and therefore, the explanation is the same. As already mentioned, the third set represents the possibility of both the service providers being in the present state (m, n, m', n') and rate at which any service provider leaves its present state. The rate at which an arrival takes place is λ when a service provider accepts an arriving client (given by $C(1, m, n) \leq 0$ or $C(2, m', n') \leq 0$). The rate at which an idle client opens a session for service provider 1 is $S(0) \times (m-n)$ and for service provider 2 is $S(0) \times (m'-n')$. The rate at which a client in session closes a session for service provider 1 is $S(1) \times \left(\begin{cases} n & , n \leq B_1 \\ B_1 & , otherwise \end{cases} \right)$ and for service provider 2 is $S(1) \times \left(\begin{cases} n' & , n' \leq B_2 \\ B_2 & , otherwise \end{cases} \right)$.

$$
\begin{aligned}
0 \;=\; & \{ \\
& \begin{cases} Pr(m-1,n-1,m',n') & ,m \neq 0, n \neq 0 \\ \\ 0 & ,otherwise \end{cases} \\
& \times \begin{cases} \lambda & C(1,m-1,n-1) \leq 0 \\ & ,\{C(1,m-1,n-1) > C(2,m',n') \\ & or\ C(2,m',n') = 1\} \\ & ,m \neq 0, n \neq 0 \\ \\ \frac{\lambda}{2} & ,C(1,m-1,n-1) \leq 0 \\ & ,C(1,m-1,n-1) = C(2,m',n') \\ & ,m \neq 0, n \neq 0 \\ \\ 0 & ,otherwise \end{cases} \\
& + \begin{cases} Pr(m+1,n+1,m',n') \times S(1) \times (n+1) \times d & ,m < m_{max} \\ & ,n+1 \leq B_1 \\ \\ Pr(m+1,n+1,m',n') \times S(1) \times B_1 \times d & ,m < m_{max} \\ & ,n+1 > B_1 \\ \\ 0 & ,otherwise \end{cases} \\
& + \begin{cases} Pr(m,n+1,m',n') \times S(1) \times (n+1) \times (1-d) & ,n < m \\ & ,n+1 \leq B_1 \\ \\ Pr(m,n+1,m',n') \times S(1) \times B_1 \times (1-d) & ,n < m \\ & ,n+1 > B_1 \\ \\ 0 & ,otherwise \end{cases} \\
& + \begin{cases} Pr(m,n-1,m',n') \times S(0) \times (m-n+1) & ,n > 0 \\ \\ 0 & ,otherwise \end{cases} \\
& \}
\end{aligned}
$$

$$+\{$$

$$\begin{cases} Pr(m,n,m'-1,n'-1) & ,m'\neq 0, n'\neq 0 \\ \\ 0 & ,otherwise \end{cases}$$

$$\begin{cases} \lambda & C(2,m'-1,n'-1)\leq 0 \\ & ,\{C(2,m'-1,n'-1) > C(1,m,n) \\ & or\ C(1,m,n)=1\} \\ & ,m'\neq 0, n'\neq 0 \\ \\ \frac{\lambda}{2} & ,C(2,m'-1,n'-1)\leq 0 \\ & ,C(2,m'-1,n'-1) = C(1,m,n) \\ & ,m'\neq 0, n'\neq 0 \\ \\ 0 & ,otherwise \end{cases}$$

$$+\begin{cases} Pr(m,n,m'+1,n'+1)\times S(1)\times (n'+1)\times d & ,m'<m_{max} \\ & ,n'+1\leq B_2 \\ \\ Pr(m,n,m'+1,n'+1)\times S(1)\times B_2\times d & ,m'<m_{max} \\ & ,n'+1 > B_2 \\ \\ 0 & ,otherwise \end{cases}$$

$$+\begin{cases} Pr(m,n,m',n'+1)\times S(1)\times (n'+1)\times (1-d) & ,n'<m' \\ & ,n'+1\leq B_2 \\ \\ Pr(m,n,m',n'+1)\times S(1)\times B_2\times (1-d) & ,n'<m' \\ & ,n'+1 > B_2 \\ \\ 0 & ,otherwise \end{cases}$$

$$+\begin{cases} Pr(m,n,m',n'-1)\times S(0)\times (m'-n'+1) & ,n'>0 \\ \\ 0 & ,otherwise \end{cases}$$

$$\}$$

$$
\begin{aligned}
&-\{Pr(m,n,m',n')\\
&\times(\left(\begin{cases}\lambda & ,C(1,m,n)\le 0\\ & or\ C(2,m',n')\le 0\\ 0 & ,otherwise\end{cases}\right)+S(0)\times(m-n)+S(0)\times(m'-n')\\
&+S(1)\times\left(\begin{cases}n & ,n\le B_1\\ B_1 & ,otherwise\end{cases}\right)+S(1)\times\left(\begin{cases}n' & ,n'\le B_2\\ B_2 & ,otherwise\end{cases}\right)\\
&)\\
&\}
\end{aligned}
\tag{3.4}
$$

Since the sum of probabilities is 1, another equation is :

$$\sum_{s1,s2} Pr(s1,s2) = 1 \tag{3.5}$$

The values of $Pr()$ are found by solving equations 3.4 and 3.5. The method used is LU Decomposition [7].

3.1.5 Finding $Waiting(i, s1)$

The value of the $Waiting()$ function is the number of clients waiting for a session for a given state. Given $s1$ is (m, n), there will be no waiting client if the n, is less than the bandwidth B_1 (remember, each client uses exactly one unit of bandwidth). Otherwise, it will be $n - B_1$, assuming a first come first serve queueing discipline. So we get,

$$Waiting(i,m,n) = \begin{cases}0 & ,n\le B_1\\ (n-B_1) & ,otherwise\end{cases} \tag{3.6}$$

3.1.6 Accurate Solution Complexities

The complexity analysis consists of three components. The first is the space required to store the decision matrix. It is the size of the result. The second is the time complexity of finding the decision matrix. The third is the space complexity of finding the decision matrix.

The decision matrix C stores an integer for every possible state. The number of possible values of (m, n) is the space requirement of storing C. For a given value of

m, the value of n ranges from 0 to m. The value of m ranges from 1 to m_{max}. By multiplying all of these values, the space required to store C comes to be $O({m_{max}}^2)$.

The total time taken is the number of values of C for each service provider multiplied by the time taken to find the value of $Pr()$. The number of values of C decides the number of times steps 2 to 8 have to be executed in the solution method in section 3.1.2. In these steps, for a given decision matrix of service provider x, the best decision of service provider y is found. A simple method is to find all possible decisions of service provider y and then choose the best. It is assumed that if a service provider charges price p from an arriving client at state (m, n), he will charge price p or less at state (x, y) given $x \leq m$ and $y \leq n$. It is not expected that a service provider will charge low price at state (5,0) and high price at state (2,0). Therefore, for finding all possible decision matrix, the objective is to find the values in which the decision changes from low prices to high prices and on reaching the highest price, the decision becomes rejection.

Let $c(i, n)$ be a matrix which represents the states at which the decision changes. When i is zero, it is about a change in decision from lowest price to the second lowest price. $c(i, n)$ returns the value of m at which the decision is to charge $price(i)$ or higher price if $i \leq T - 1$ or reject if i is T (T is the number of distinct prices). Below this value of m the decision is to charge $price(i - 1)$ or less. The number of possible values of $c(i, n)$ is $O({m_{max}}^{T \times m_{max}})$. It is because for each (i, n), the values can range maximum from 0 to m_{max} and i can range from 0 to $T - 1$, and n depends on the value of m_{max}.

In the best case the solution runs for all possible C matrix for the smaller service provider and is $O({m_{max}}^{T \times m_{max}})$. In the worst case, it is for all possible values of the C matrix for each service provider and the number of possibilities of decision possibilities for two service providers is then $O(({m_{max}}^{T \times m_{max}})^2)$ and it becomes $O({m_{max}}^{2 \times T \times m_{max}})$. Finding income per unit time from equation 3.1 and finding the value of D from equation 3.2 does not take much time and space; and it can be ignored.

The next step is to find the time and space need to find the values of $Pr()$ for given $C()$ values. For finding $Pr()$, and we use LU decomposition method ([7]) to solve the equations. The number of possible equations is the number of possible values of $Pr()$ and it is $O({m_{max}}^4)$. It is possible to choose some $Pr()$ values and write every other $Pr()$ values in terms of the chosen $Pr()$ values. The chosen unknowns are the values of Pr() in which all connected clients are idle in service provider 1. Finally all the $Pr()$ are written in terms of the chosen $Pr()$ values. This reduces the number of equations to be solved by m_{max} and it comes to be $O({m_{max}}^3)$. The step which takes maximum time in LU decomposition method is triangular factorization[13]. In [13], authors have shown that if a time complexity to multiply two n by n matrix is O($x(n)$) then triangular factorization can also be done in O($x(n)$) time. In [60], the authors have shown it is possible to multiply n by n matrix in O($n^{2.3727}$) time. The number of possible values of equation to be solved is O(${m_{max}}^3$). Therefore the time complexity is O($({m_{max}}^3)^{2.3727}$). After multiplying the time taken to find the the value of $Pr()$ for each C combination by the number of C combinations, time complexity comes to be O($\{{m_{max}}^3\}^{2.3727} \times {m_{max}}^{T \times m_{max}}$) in best case and O($\{{m_{max}}^3\}^{2.3727} \times {m_{max}}^{2 \times T \times m_{max}}$) in worst case. The complexity becomes O(${m_{max}}^{7.12} \times {m_{max}}^{T \times m_{max}}$) in the best case and O(${m_{max}}^{7.12} \times {m_{max}}^{2 \times T \times m_{max}}$) in the worst case. Complexity can be reduced by refining the method of search of the solution space, but details are omitted as the complexity is high in any case.

The size of LU Matrix is O($({m_{max}}^3)^2$) which reduces to O(${m_{max}}^6$). In the method to find Nash equilibrium given in Section 3.1.2, multiple values of decisions have to be stored in $C_{checked}$. The space taken to store one decision to $C_{checked}$ is the size of result and is O(${m_{max}}^2$). In the best case, the number of such values is small and can be ignored. In the worst case, it is not small but in the worst case a simple algorithm is to check all possible decisions for Nash equilibrium. This brute force algorithm does not use $C_{checked}$ and so the space complexity does not increase. The space complexity is therefore the size of LU matrix and is O(${m_{max}}^6$).

Clearly the complexity is very high and even for moderate values of m_{max} (say 100), a solution cannot be found practically. So we need to consider simplifications to the problem.

3.2 Approximate solution

The process of finding the Nash Equilibrium using the above analysis requires high memory space and computation time. Therefore, we present an approximate solution which has low space and time complexity. This approximate solution is compared with the accurate solution by running the two solutions for small values of m_{max} and comparing the expected incomes.

In the approximate solution, the state of a service provider is represented by a single integer m, the number of clients connected to the service provider. We do not keep track of the number of clients that are in session or waiting for a session (n in the accurate solution). Instead, we estimate the number of clients that are in session. The method to find the approximate Nash equilibrium is similar to the method used in the accurate solution. There are three differences: the first difference is that the number of solution matrices to be consider is reduced because the state is represented by a single integer, the second difference is that the equations of $Pr()$ are different and the third difference is that the $Waiting()$ function's value is different.

3.2.1 Steady State Probability

The combined steady state probability $Pr(s1, s2)$ represents the probability that service provider 1 is in state $s1$ and service provider 2 is in state $s2$. Like the accurate solution, the method to find $Pr()$ is to consider $Pr_t()$ and then write equations 3.3. The next step is to remove time to obtain equation 3.4. In the approximate solution, the state of a service provider is represented by a single integer and therefore, there are some differences in the steady state probability formula. In the approximate solution, $E(i, m)$ is used to estimate the expected number of clients in session and this is used instead of the actual (unknown) value, as an approximation. Equation 3.4 is expanded for the approximate solution as equation 3.7.

$$
\begin{aligned}
0 \quad = \quad & Pr(m_1+1, m_2) \times \\
& \begin{cases} S(1) \times d \times E(1, m_1+1) & , m_1 \neq m_{max} \\ 0 & , otherwise \end{cases} \\
& + \left(\begin{cases} Pr(m_1-1, m_2) & , m_1 \neq 0 \\ 0 & , otherwise \end{cases} \right) \times \\
& \begin{cases} \frac{\lambda}{2} & , C(1, m_1-1) \leq 0 \\ & , C(1, m_1-1) = C(2, m_2) \\ \lambda & , C(1, m_1-1) \leq 0 \\ & , \{C(2, m_2) < C(1, m_1-1) \\ & \quad or\ C(2, m_2) = 1\} \end{cases} \\
& + Pr(m_1, m_2+1) \times \\
& \begin{cases} S(1) \times d \times E(2, m_2+1) & , m_2 \neq m_{max} \\ 0 & , otherwise \end{cases}
\end{aligned}
$$

$$+\left(\begin{cases} Pr(m_1, m_2 - 1) & , m_2 \neq 0 \\ 0 & , otherwise \end{cases}\right) \times$$
$$\begin{cases} \frac{\lambda}{2} & , C(2, m_2 - 1) \leq 0 \\ & , C(1, m_1) = C(2, m_2 - 1) \\ \lambda & , C(2, m_2 - 1) \leq 0 \\ & , \{C(1, m_1) < C(2, m_2 - 1) \\ & or\, C(1, m_1) = 1\} \end{cases}$$
$$-Pr(m_1, m_2) \times$$
$$\{\begin{cases} \lambda & , C(1, m_1) \leq 0\, or\, C(2, m_2) \leq 0 \\ 0 & , otherwise \end{cases}$$
$$+S(1) \times d \times E(1, m_1) + S(1) \times d \times E(2, m_2)\} \tag{3.7}$$

3.2.2 Finding expected clients in session and waiting for session

3.2.2.1 Finding $E(i, m)$

$E(i, m)$ is the expected number of clients in session when m clients are connected. The equation is given below.

$$E(i, m) = \sum_n \left(Pri(m, n) \times \begin{cases} B_i & , n > B_i \\ n & , otherwise \end{cases} \right) \tag{3.8}$$

where $Pri(m, n)$ is obtained using the finite source queueing theory formula[57]. This distribution applies as the service time is exponentially distributed,the arrival process is Poisson, and the idle time is also exponentially distributed. The formula for $Pri(m, n)$ is:

$$Pri(m, n) =$$
$$\begin{cases} \begin{pmatrix} m \\ n \end{pmatrix} \left(\frac{S(0)}{(1-d)\times S(1)}\right)^n Pr(m, 0) & , n < B_i \\ \frac{m!}{(m-n)!} \frac{1}{B_i! \times B_i^{n-B_i}} \left(\frac{S(0)}{(1-d)\times S(1)}\right)^n Pr(m, 0) & , n \geq B_i \end{cases} \tag{3.9}$$

and

$$Pri(m,0) = \{\sum_{n=0}^{B_i-1} \binom{m}{n} \times \left(\frac{S(0)}{(1-d)\times S(1)}\right)^n + \sum_{n=c}^{m} \frac{m!}{(m-n)!} \times \frac{1}{B_i! \times B_i^{n-B_i}} \times \left(\frac{S(0)}{(1-d)\times S(1)}\right)^n \}^{-1} \tag{3.10}$$

3.2.2.2 Finding $Waiting(i,s)$

As already mentioned, in the approximate solution the state of a service provider is represented by a single integer. Therefore, $Waiting(i,s)$ can be written as $Waiting(i,m)$. $Waiting(i,m)$ is the expected number of waiting clients when m clients are connected. The value of $Waiting(i,m)$ is found in the same way as $E(i,m)$. The equation is given below.

$$Waiting(i,m) = \sum_n \left(Pri(m,n) \begin{cases} n - B_i & ,n > B_i \\ 0 & ,otherwise \end{cases} \right) \tag{3.11}$$

3.2.3 Approximate Solution Complexities

The space and time complexity depend on the computational time and memory required to solve equation 3.7, equation 3.8 and equation 3.11. Other steps take less time and space and therefore are ignored.

Equation 3.7 has ${m_{max}}^2$ unknowns. It is possible to choose some unknowns and write everything else in terms of the chosen unknowns. The chosen unknowns are those values of $Pr()$ for which the first term is zero. The time taken to write all unknowns in terms of the chosen unknowns is $O({m_{max}}^3)$ and the space taken is $O({m_{max}}^2)$. This reduces the number of equations to be solved to m_{max}. The time taken to solve m_{max} equations is within $O({m_{max}}^3)$ and the space taken is $O({m_{max}}^2)$. The overall time complexity is $O({m_{max}}^3)$ and space complexity is $O({m_{max}}^2)$.

The number of possible values of $Pri()$ in equation 3.8 and equation 3.11, for a given value of m is $O(m)$. It is because $Pri(m,n)$ has two terms and for a given value of m, n ranges from 0 to m. m can range from 0 to m_{max} so the total number of values is $O({m_{max}}^2)$. The method of finding $Pri()$ does not have any space requirement. The only memory requirement is to store the values of $E(m)$ that can have m_{max} values. Therefore the time and space complexity to find $E()$ is $O({m_{max}}^2)$ and $O(m_{max})$ respectively.

The method of finding $Pr()$; and $E()$ has to be done separately. Therefore the time and space complexity to find $Pr()$ is the maximum of these; and the time complexity is O($m_{max}{}^3$) and space complexity is O($m_{max}{}^2$).

The method of choosing the decisions of service provider 1 and then finding the best decisions of service provider 2 and then finding best decisions of service provider 1 and so on requires finding of all possible values of $C()$. If there are T prices, the decision that can be taken in a state is $T+1$. It is assumed that if a service provider charges price p from an arriving client at state m, he will charge price p or less at state (x) given $x \leq m$. It is not expected that a service provider will charge low price at state 5 and high price at state 2. Therefore, for finding all possible decision matrix, the objective is to find the values in which the decision changes from low prices to high prices and on reaching the highest price, the decision becomes rejection.

Let $c(i)$ be a matrix which represents the states at which the decision changes. When i is zero, it is about a change in decision from lowest price to second lowest price. $c(i)$ returns the value of m at which the decision is to charge $price(i)$ or higher price if $i \leq T - 1$ or reject if i is T. Below this value of m the decision is to charge $price(i-1)$ or less. The number of possible values of $c(i)$ is $(m_{max})^T$. It is because for each i, the values can range maximum from 0 to m_{max} and i can range from 0 to $T-1$. Therefore, it takes O($m_{max}{}^T$) steps in the best case and O($m_{max}{}^{2\times T}$) steps when all possible $C()$ are chosen. Therefore the overall time complexity is O($m_{max}{}^{3+T}$) in the best case and O($m_{max}{}^{3+2\times T}$) in the worst case; and space complexity is O($m_{max}{}^2$) .

3.3 Comparison of the two solutions

We compare with two service providers. There are a number of comparisons and in each comparison bandwidths are different. Bandwidth of service providers range from 3 units to 5 units while the maximum number of users, m_{max}, is 8. The mean arrival rate of clients is 0.2 per unit time. The clients' mean idle time is 20 units, the mean session duration is 12 units of time, and bandwidth is consumed by every client at 1 unit per unit time. The value of d is 0.4. There are two prices and these are 0.1 and 0.12 per unit data transfer. Penalty for session delay is 0.2 per unit time and penalty for rejection is 1 per rejection. We find the values of the C arrays of the two providers at Nash equilibrium using the method given in section 3.1.2. We then calculate the expected incomes $I(1)$ and $I(2)$ using equation 3.1.

The result of the comparisons is given in Table 3.2. Each entry of the table contains two values and these values are separated by a comma. The first value is for service provider 1 and the second value is for service provider 2. Column 1 gives the expected income of the service providers when the accurate solution is used and

Table 3.2: Comparison between Accurate and Approximate solutions

Accurate solution income	Approximate solution income	Bandwidth
0.141, 0.141	0.120, 0.120	3,3
0.152, 0.228	0.132, 0.227	3,4
0.154, 0.306	0.138, 0.303	3,5
0.245, 0.245	0.237, 0.237	4,4
0.249, 0.292	0.230, 0.293	4,5
0.283, 0.283	0.283, 0.283	5,5

column two gives the expected income of the service providers when the approximate solution is used. The bandwidths of the service providers is given in column 3. The approximate solutions income is at most 15 % less (when the bandwidths are 3 for both; with maximum 8 users, the chances of penalties are high in this case) to almost no difference when the available bandwidth is high (in the case of bandwidths are 5 for both; the chances of penalties are low). So, these preliminary results show that, if highly congested situations can be avoided (which the ISPs will strive to do as otherwise penalties will increase), the approximate solution will perform as well as the accurate solution.

3.4 Existence of Nash Equilibrium

We have presented Nash equilibrium solutions. These solutions can find Nash equilibrium only if it exists. If any Nash equilibrium solution does not exist, our method cannot find a Nash equilibrium. Therefore, the next question that arises is whether Nash equilibrium exists in all cases. We do not have a proof regarding existence of Nash equilibrium. However, we argue that Nash equilibrium should exist in all the cases.

We assume that Nash equilibrium does not exist for some system configuration. Consider two service providers and assume that they have chosen their strategies. Each service provider will try to choose an appropriate strategy to maximize its profit. Suppose service provider 1 chooses strategy A. Based on this strategy, service provider 2 computes its best strategy and let it be strategy B. Given the strategy of service provider 2, service provider 1 finds its best strategy and let it be strategy C. The strategy C, should be different from its strategy A. If A and C are the same, this means that the strategies are in Nash equilibrium. Based on strategy C of service provider 1, service provider 2 recomputes its best strategy and let this strategy be strategy D. Similarly, if strategies B and D are the same, then this means that the service providers are in Nash equilibrium. Therefore, we assume that strategies B and D are different.

We assume that the main difference between strategies A and C is that there will either be an increase in price charged and / or increase in connection rejections; or a decrease in price charged and / or decrease in connection rejections. It may also be possible that in decision C service provider 1 increases prices in some states and decreases prices in some other states as compared to strategy A. However, we are considering the overall effect and one set of decisions is unlikely to have a major effect on the other service providers. Therefore, we assume that the main decisions that service provider takes in strategy C as compared to strategy A is either to increase prices and / or increase rejections; or decrease prices and / or decrease rejections. The same argument applies for strategies B and D.

Consider strategies A and C. When one service provider takes some decisions, its affect on the other service provider is that there is an increase or a decrease in arrival rate of clients. When congestion is low, a service provider is likely to charge low prices. When there is an increase in arrival rate, a service provider is likely to increase prices charged and / or reject connection requests of arriving clients. When one service provider increases price or increases number of rejections of arriving clients, it will increase arrival rate of the other service provider and therefore, the other service provider may also increase price or reject connection requests of arriving clients. If in strategy C, service provider 1 increases prices and / or increase the number of rejections of arriving clients as compared to strategy A, we expect that service provider 2 may also increase price and / or increase the number of rejections of arriving clients in strategy D as compared to strategy B. It is because if service provider 1 increases price or increase rejections, it will reduce its arrival rate and the arrivals for service provider 2 will increase and to reduce chances of congestion, service provider 2 may also increase price or increase rejections. Similarly if service provider 1 decreases prices or decreases rejections, service provider 2 may also decrease prices or decrease rejections. This means that decision of service provider 2 is likely to be in the same direction as the decision of service provider 1. Based on the strategy of service provider 2, service provider may again modify its strategy and this procedure continues. The number of states are finite and therefore the number of possible decisions are also finite. Since the change in decisions (from A to C and so on; and from B to D and so on) will be in the same direction, it should not go to an infinite loop and this change should stop somewhere and that is until Nash equilibrium is reached. Therefore, we feel that Nash equilibrium should exist.

3.5 Limitations of Nash Equilibrium

Our Nash equilibrium solutions have many limitations. Our Nash equilibrium solutions have high time and space complexities. We have presented an approximate solution but still the complexities are high. We have present Nash equilibrium for two ISPs. If the same method is used to find Nash equilibrium for more than two ISPs, the time and space requirements again increase. We have also put a restriction on the maximum bandwidth a client can request to one unit. If this is increases, the Nash equilibrium solution cannot handle it because of increased complexities. To handle these limitations, we present non game theoretic solutions in the coming chapters. The solutions have lower time and space complexities. The complexities are independent of the number of ISPs.

Chapter 4

Non Game Theoretic solution

In this chapter, we present a non game theoretic model and its solution that provides bandwidth and connection guarantees to clients. The solution has lower time and space complexities than our game theoretic solutions and the complexities are independent of the number of service providers. The system architecture and the model are explained in section 4.1 and 4.2. Our solution and its space and time complexities are described in section 4.3. This solution's correctness is proved in section 4.3.6.

4.1 The System Architecture

The interaction between a client and service providers is almost the same as in the Nash equilibrium solution and is already explained in Figure 3.1. The main differences are: a client can connect to any number of service providers and a client can request for bandwidth in the range 1 unit to e units. When a client wishes to connect, he requests a price for some bandwidth in the range of '1 to e' units (we assign probabilities $G(i)$ for i units in our model (see below)). The ISP then offers him a price (per unit of bandwidth consumed). The user obtains such prices from all the ISPs he can connect to and makes a connection request to the ISP offering the lowest price. In case more than one ISP is offering the lowest price, the client randomly chooses one of them. The price offered to a user remains the same during the entire duration the user is connected to the ISP. So this gives a connection of a fixed number of units of bandwidth at a fixed price to a client for the entire duration of his connection to the internet. When a connection request is made to an ISP, the ISP decides whether to accept the connection or not. If it refuses the connection, the client is paid a fixed penalty and no further requests are entertained from that client till the price offered changes. These steps prevent a client from taking advantage of a congested ISP by making repeated requests to collect penalties. So, when the price is offered, the ISP will offer the highest possible price if it is going to refuse connections. The client will not know if the connection is going to be refused till it actually makes a request and so it may make a request to this ISP only if all other ISPs are also offering the highest price. Once a connection is refused, no further requests are entertained to prevent a client getting multiple penalties. The ISP consults a table where, for each possible state of the ISP, the decisions are given (the decisions are a) whether to refuse or accept a connection, and b) if accepting, the price to offer). We need to populate this table so that the mean income of the ISP at steady state is maximised.

Once a client enters an ISP's system, it is assumed that he will require bursts of bandwidth (of the number of units he requested) and then he will be idle for a while and then he will again request for a burst. So a client can be in one of three states while connected: a) consuming bandwidth (he is said to be in session), b) idling (he is in state idle), or c) waiting for bandwidth to be allocated (he is waiting to enter a session). After consuming a burst of bandwidth, the client may leave the system or it may go into the idle state. When a connected client has to wait to enter a session, he earns a session delay penalty which is proportional to the time he is delayed. The client has to pay for the amount of bandwidth he consumes.

Our premise is that QoS can be effectively imposed by penalties. While we have penalties for delays, we also have penalties for rejection of connections as without them, an ISP can oversubscribe users ensuring a good load always without getting penalised for it. So when an ISP allows users to connect to it, it will use its past knowledge to decide whether to accept new clients or not. With this feature, a client gets "end-to-end" QoS guarantees, not merely when he succeeds in getting connected. As already mentioned, this scheme introduces competition among ISPs. So on the one hand, an ISP will try to attract as many clients as possible by lowering its price. However, as its available bandwidth decreases, the chance of paying delay penalties increases and so it will slow down the arrival of new clients by increasing the offered price. A stage will be reached when it will be cost-effective to pay a penalty and refuse a client's request to connect. So a lightly loaded ISP is likely to offer a low price, while a heavily loaded ISP is likely to offer a high price. As clients will connect to that ISP offering the lowest price, load balancing among ISPs is built-in in the scheme. From a client's point of view, an assurance that he will not get delayed due to want of bandwidth is given by the ISP in the scheme. To prevent "squatting" (a client remains in idle state for long periods as he got a very good price when connecting, but now the ISPs are all offering higher prices), connections will have to be dropped if idle times exceed a threshold a pre-defined number of times during a connection. The state of an ISP is represented by an array of integers $(m, n_1, n_2, ..., n_e, r_1, r_2, ..., r_e)$. For ease of exposition we group them and express it as (m, N, R) where m represents the number of clients connected. Since a service provider can reject an arriving client, the value of m is within some range from 0 to m_{max} where m_{max} is the maximum possible value of m. e is the size of the arrays N and R and it represents the maximum bandwidth which a client can request. It is assumed that bandwidth is requested in discrete units from 1 unit to e units. N is an array which represents the number of clients in session and its index ranges from 1 to e. So n_i represents the number of clients in session with i units of bandwidth. R is an array which represents the number of clients waiting for a session and its index also ranges from 1 to e. r_i represents the number of users who requested for i units of bandwidth but are queued. So we have a finite state space and our goal is to define the decisions to be taken on an arrival, for each state. So we have a decision matrix $C(m, N, R)$. When the value of $C(m, N, R)$ is zero or less, the client is to be accepted and the price to be charged is $price(-C(m, N, R))$, where price is an array containing the different prices that can be levied. When $C(m, N, R)$ is 1, the client is to be rejected. All these definitions and functions derived from them are described and shown in Table 4.1 and Table 4.2.

4.2 The Model

In order to find the values of the matrix, we introduce a simplified model that is amenable to analysis. Further, the size of $C()$ is very large for practical ISPs and so it is not feasible to implement. Our model is depicted in Figure 4.1. The model mirrors the architecture described above, except that it makes the following assumptions:

- Clients arrive at an ISP according to a Poisson process. The mean arrival rate is dependent on the price being offered. It increases with a decrease in price. These rates are assumed to be known a priori. In practice, they will be determined by actually observing the arrival rates and using these rates to make future decisions. The impact of prices charged by the other service provider is coming in the change is the mean arrival rates, λ, when different prices are charged. When price is increased, some clients may connect to other service providers and so the mean arrival rate decreases; similarly when price is decreased, the arrival rate increases.

- The service time of a client consuming bandwidth and the time spent in idle state by a client are both assumed to be distributed exponentially with fixed mean rates. All arrivals and departure processes are therefore "memoryless".

Table 4.1: Definition of Symbols

Symbol	Description
$\lambda(i)$	The mean arrival rate of clients at the service provider when $price(i)$ is being charged; i varies from 0 to $T-1$
$G(i)$	Probability of a client requesting i units of bandwidth when he initially has no bandwidth. Here i is an integer ranging from 1 to e.
$\frac{1}{S(i)}$	Mean session duration or idle duration for which a client consumes i units of bandwidth. If i is zero, it is the mean time a client remains idle. If i is non zero, it is the mean time for which a client remains in session consuming i units of bandwidth.
$price(i)$	$price(i)$ is an array which stores the possible prices per unit data that can be charged from an arriving client. The values stored in $price(i)$ are in ascending order. $price(0)$ is the least price and $price(T-1)$ is the maximum.
T	Number of prices
p	The price being charged from the client being considered.
m_{max}	It is the maximum possible value of the number of clients connected.
B	The total units of bandwidth available with the service provider
e	Maximum amount of bandwidth a client can request from his service provider. A client can request 1 to e units of bandwidth.
d	Probability that a client disconnects immediately leaving idle state.
$P(0)$	Penalty per unit time which is paid to a client when he waits for Session entry.
$P(1)$	Penalty which a client gets when his request to connect is rejected.
m	The number of clients connected. It ranges from 1 to m_{max}.
N	An array which represents the number of clients in session and its index ranges from 1 to e. So n_i represents the number of clients in session with i units of bandwidth.
R	An array which represents the number of clients waiting for a session and its index also ranges from 1 to e. r_i represents the number of users who requested for i units of bandwidth but are queued where i ranges from 1 to e.

Table 4.2: Function declaration

Function	Description
$C(m, N, R)$	The decision a service provider takes on the arrival of a client when the state of a service provider is (m, N, R). When the value of $C(m, N, R)$ is zero or less, the client is to be accepted and the price to be charged is $price(-C(m, N, R))$. When $C(m, N, R)$ is 1, the client is to be rejected. $C(m, N, R)$ is a function, but its values are calculated offline and stored in an array $C()$ and at run time, it is the array that is consulted.
$C_{new}(m, N, R)$	The decision which has maximum advantage based on the current values of the array $C()$. Other things are the same as in function $C(m, N, R)$
$C_{opt}(m, N, R)$	It represents the optimal set of decisions. When this function is used in decision making,the expected income is maximized. Other things are the same as in function $C(m, N, R)$
$D(m, N, R, K, p)$	The advantage of having an extra client (the net expected increase in income) when the state of the service provider is (m, N, R). K is an array with two terms: k_1 and k_2. If $k_1 = 0$ then the extra client is connected and not queued for session and k_2 is the bandwidth which the extra client asks for and is given. If $k_1 = 1$ then the extra client is connected and queued for session and k_2 is the bandwidth which the extra client has requested. p is the price charged from the extra client.
$D1(m, N, R, K, p)$	The expected income earned and lost in future due to session delays, from the arrival of an extra client when the state of the service provider is (m, N, R) and this client is represented by K. The definition of inputs to this function is the same as in the function $D()$.
$D1_t(m, N, R, K, p, t)$	The expected income earned and lost in future by the arrival of an extra client at time t when the state of the service provider is (m, N, R). The definition of inputs to this function is the same as in the function $D1()$.
$D2(m, N, R)$	The advantage of a service provider in being in state (m, N, R) instead of state $(0, 0, 0)$ (the state with no clients) due to income earned from future newly arriving clients.
$D2_t(m, N, R, t)$	The advantage of a service provider in being in state (m, N, R) instead of state $(0, 0, 0)$ at time t due to income earned from newly arriving clients.

- Among the connected clients the opening of and closing of sessions is modelled as a finite population queuing system. Clients queued for entering a session will in general have different bandwidth requirements and the total bandwidth available is fixed. Therefore, there may not be enough spare bandwidth to service the client first in the queue, but another client behind in the queue could be serviced. So, to ease analysis, the queue servicing discipline is assumed to be "least bandwidth required first". In actual practice, a first-come-first-serve policy with "queue-jumping" if service is not possible, should be used.

- Rather than associating a bandwidth requirement with each arriving client, in the model it is assumed that clients' bandwidth requirements are determined at the time of moving out of the idle state and the allotted bandwidth is i units with probability $G(i)$. This makes the state of a client also "memoryless" and assists in analysis.

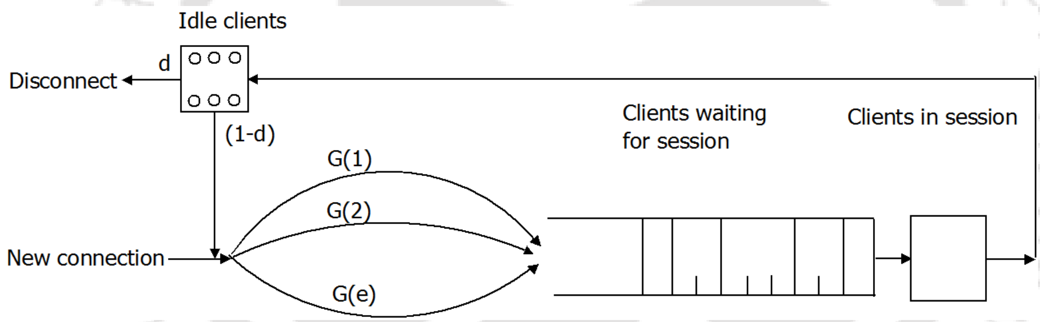

Figure 4.1: A service provider in our model

4.3 Accurate Solution

In this section, we present a solution to our model. In the coming chapters, we also present approximate solutions and to distinguish our non-approximate solution with the approximate solutions, we use the term "accurate solution" for the non-approximate solution. We present an algorithm below to find the accurate solution.

4.3.1 Solution Method

4.3.1.1 Continuous Time Markov Decision Process

Our scheme can be modelled as a Continuous Time Markov Decision Process (CTMDP). [22] mentions a bandwidth pricing and call admission control scheme and also models the problem as a CTMDP. A CTMDP has four properties [25]: a state space, an action space, transition rates and reward rates. These properties for our model are given below.

1. State space: The state space is given by (m, N, R). The number of states are the unique possible value of (m, N, R). The state space is finite.

2. The action space is $0, 1, ..., T$. which represents decision to be taken when a client arrives. When chosen action 'a' is less than T, price $price(a)$ is charged from the arriving client and the client is allowed to connect. Otherwise, $price(T-1)$, is advertised but the client is rejected.

3. Transition rates depend on the decision taken at state (m, N, R). In our model, there are four types of events: an arrival event; a client leaving idle state and then requesting for bandwidth; a session close and then departure; and a session close and then becoming idle. The rate at which the service provider moves from one state to another state is the transition rate. For example, if from state $s1$, an ISP can go to state $s2$ because of an arrival and the arrival requests for one unit bandwidth initially, the transition rate is the arrival rate multiplied by $G(1)$. If a single event cannot change state from $s1$ to $s2$, the transition rate is zero.

4. The Reward rate. The expected income earned per unit time when a service provider is in a given state when a particular action is taken is the reward rate. It is the sum of two components: the rate at which an arrival takes place multiplied by the income earned from the arrival and the expected penalty paid if any client(s) is/are waiting for bandwidth. When an arrival takes place, the expected income earned from an arrival can be written as $E_d \times$ <price charged from the arrival>, if arrival is rejected it is $-P(1)$ where E_d is the expected data to be consumed by an arriving client. The penalty to be paid is P(0) for all waiting clients. The waiting clients are r_i for i equal to 1 to e, where r_i are number of clients waiting for i units bandwidth. The reward rate at state (m, N, R) when action a is taken, is therefore

$$\text{Reward rate} = \left(\begin{cases} \lambda(a) \times E_d \times price(a) & , a < T \\ \lambda(T-1) \times (-P(1)) & , a = T \end{cases} \right) - \sum_{i=1}^{e} r_i \times P(0)$$

There are three optimality criteria in the CTMDP as given in [25]. In our case, the objective is to maximize the expected average reward. We use the policy iteration method to find a solution. The method of policy iteration is to pick an initial policy, improve the policy and then continue improvement till it converges. This method assumes some initial solution and then based on the solution calculates a better solution and this improvement continues. This algorithm is explained in Section 4.3.1.2. Later we prove in Section 4.3.6 that using this method we get an optimal solution.

4.3.1.2 AAEC (Advantage of An Extra Client) Method

We start with some instance of the matrix $C()$ and then improve it. An instance gets improved, if, by changing some values in the matrix, the expected income increases. This improvement continues till no more improvement is possible. The steps to find the solution are given below.

1. Start with a random $C()$ matrix.

2. For every element of $C()$ (which corresponds to a state of the system), keeping all other elements as it is, find that value which improves the mean income the most (each value is a decision to be taken when in that state). It could be the existing value or some other value. The value is the price to be charged to an arriving client when the system is in that state or it could be the decision to reject an incoming client. To find the increase or decrease in the mean income due to a new value, we need to find the advantage of introducing an extra client. An arriving client causes three things: gives some income to the ISP, causes congestion to existing clients (incurring penalties) and makes future arriving clients congested (incurring penalties again). The net effect is the advantage. If the decision is to reject an arriving client, then the advantage is negative, being the penalty of refusing a connection. So that decision which gives the best advantage of an extra client is the decision whose value replaces the existing value of the current element. If two or more decisions produce the same maximum advantage and the existing decision is among them, choose the existing decision ($C()$); otherwise choose any of the decisions that gives the maximum advantage.

3. The changed $C()$ is $C_{new}()$.

4. If $C(state)$ and $C_{new}(state)$ are the same for every possible 'state', the optimal result is obtained. If not, then copy $C_{new}()$ to $C()$ and go to step 2.

4.3.1.3 Method to find advantage of a decision in terms of D()

When a service provider takes some decision, that decision affects the mean arrival rate of clients and therefore it effects state transitions. A service provider can find the advantage of a decision in terms of state transitions this decision produces. At steady state, a service provider has to maximize the rate at which it earns income. This rate means the product of mean arrival rate of clients and advantage of an arrival (also referred to as an extra client). Suppose a service provider is in state $s1$. If an arrival takes place in state $s1$, state becomes $s2(i)$ with probability $G(i)$, where i is an integer whose value ranges from 1 to e. The difference between states $s1$ and $s2(i)$ is that in state $s2(i)$, there is an extra client. The advantage of this extra client can be found by the $D()$ function defined in Table 4.2. The advantage of an extra client is given by $\sum_{i=1}^{e} G(i) \times D(s2(i), K(i), -C(s1))$, where $K(i)$ is the state of the extra client (see below) and $price(-C(s1))$ is the price charged from the extra client. Similarly if decision $C_{new}(s1)$ is taken and the decision is to charge $price(-C_{new}(s1))$ and accept the connection request of the client, then if an arrival takes place, its advantage is $\sum_{i=1}^{e} G(i) \times D(s2(i), K(i), -C_{new}(s1))$. The advantage of taking decision $C(s1)$ is $\lambda(-C(s1)) \times (\sum_{i=1}^{e} G(i) \times D(s2(i), K(i), -C(s1)))$ and that of decision $C_{new}(s1)$ is $\lambda(-C_{new}(s1)) \times (\sum_{i=1}^{e} G(i) \times D(s2(i), K(i), -C_{new}(s1)))$. If decision $C(s1)$ or $C_{new}(s1)$ is to reject the connection request of an arriving client, advantage of the decision is $\lambda(T-1) \times (-P(1))$. The method to find the value of $D()$ is given in section 4.3.3.

4.3.2 Finding the value of $C_{new}(m, N, R)$

We find the advantage of all possible decisions and choose the decision with the maximum advantage. The decisions that can be taken are a) to reject the client and pay a penalty resulting in a loss of $P(1)$, or, b) to accept the client with a price of $p = price(i)$, for some i, i in the range of 0 to $T-1$. The decision is stored as a number which is 1 if the decision is to reject the client and which is $-i$ if the decision is to accept the client with price $price(i)$.

The advantage of rejecting clients is $\lambda(T-1) \times (-P(1))$. Let $adv(m, N, R, p)$ be the advantage of an arrival at state (m, N, R) when price p is charged. The following equations describe how $C_{new}(m, N, R)$, the decision with maximum advantage, is calculated.

$$adv(m,N,R,p) = \sum_{i=1}^{e} G(i) \times \begin{cases} D(m+1, n_1, ..., n_i+1, ..., n_e, R, k_1 = 0, k_2 = i, p) & , \sum_{j=1}^{e} n_j \times j + i \leq B \\ D(m+1, N, r_1, ..., r_i+1, ..., r_e, k_1 = 1, k_2 = i, p) & , otherwise \end{cases} \tag{4.1}$$

The first term is for the case when there is bandwidth available out of the maximum B, and the second term is for when there is no bandwidth and the client has to wait.

$$\begin{aligned} Bp(m,N,R) \;=\; & i \in \{0,1,2,...,T-1\} : \forall j \in \{0,1,2,...,T-1\}, \\ & \lambda(i) \times adv(m,N,R,price(i)) \geq \lambda(j) \times adv(m,N,R,price(j)) \end{aligned} \tag{4.2}$$

Find that price i which gives the maximum advantage.

$$C_{new}(m,N,R) = \begin{cases} -Bp(m,N,R) & , \lambda(Bp(m,N,R)) \times adv(m,N,R,price(Bp(m,N,R))) \\ & \geq \lambda(T-1) \times (-P(1)) \\ 1 & , otherwise \end{cases} \tag{4.3}$$

The value is the negative of the price found, or 1 if the best advantage is negative and is worse than rejecting the incoming client (1 means reject the client). In equation 4.2, if there is more than one best price, any price is copied to $Bp(m,N,R)$. If the existing decision ($C(m,N,R)$) and the new decision ($C_{new}(m,N,R)$) produces the same advantage, the existing decision is copied to $C_{new}(m,N,R)$. This means that for a given state (m,N,R) if $C(m,N,R)$ and $C_{new}(m,N,R)$ are different, $C_{new}(m,N,R)$ produces more advantage than $C(m,N,R)$.

4.3.3 Finding the value of $D()$

The next step is to find the value of the function $D(m,N,R,K,p)$. In this subsection, $D()$ is written in terms of $D1()$ and $D2()$ and then the method of finding $D1()$ and

$D2()$ is mentioned using function $D1_t()$ and $D2_t()$. Their definitions are given in Table 4.2. The advantage of having an extra client is the difference between the expected income when the extra client is present and the expected income when the extra client is not present. This value has two parts. The first part is the difference between the expected income earned from all the clients who are connected when the extra client is present and the expected income earned from all the clients who are connected when the extra client is not present. The second part represents the effect of the extra client on the subsequent newly arriving clients and is the expected income difference between the income earned from the newly arriving clients when the extra client is present and the expected income earned from the newly arriving clients when the extra client is not present. The first part is in terms of the function $D1()$, which represents the expected income earned by a client. The second part represents the effect of the extra client on newly arriving clients and it is in terms of $D2()$.

For finding the first part, the expected income earned from all the connected clients has to be found. This is done by finding the expected income for each of the clients using $D1()$ and multiplying it by the number of such clients. For example, if five clients are connected and not in session, and the state of the service provider is (m, N, R), we can find the expected income earned by a client who is connected but not in session using $D1()$ and then multiply it by 5. This has to be done for all the clients. The first term on the right hand side is the sum of all such incomes. The second term handles the case of i as there are $n_i + 1$ clients. The third term is the income of all clients that are waiting for a session (and so r_j is used) and the fourth term is for the clients that are idle which is $(m - \sum_{j=1}^{e}(n_j + r_j))$. From the sum of all these terms is subtracted the income when the extra client is not present.

The second part can be represented by $D2(m, N, R) - D2(m', N', R')$ where (m, N, R) is the state when the extra client is present and (m', N', R') is the state when the extra client is not present. For example, if the state of the service provider is (m, N, R), and an extra client arrives and is given i units of bandwidth and there is no waiting, then the state will become $(m+1, n_1, ...n_i+1, ...n_e, R)$. $D2(m, N, R)$ has been artificially defined as the difference between the advantage (or disadvantage) due to newly arriving clients in future when in state (m, N, R) to the advantage had these clients arrived when the initial state was $(0, 0, 0)$ (this represents the initial state where m is zero and all elements of the arrays N and R are also 0). Since we are taking the difference $D2(m, N, R) - D2(m', N', R')$, the effect of state $(0, 0, 0)$ will cancel out. This has been done to reduce the complexity of $D2$. $D2$ would otherwise have to be written as $D2(state1, state2)$ and this would have increased the state space tremendously.

$D()$ is given below as equation 4.4.

$$
\begin{aligned}
&D(m+1, n_1, ..., n_i+1, ..., n_e, R, k_1=0, k_2=i, p) = \\
&[\left(\sum_{j=1}^{e, j \neq i} D1(m+1, n_1, ..., n_i+1, ..., n_e, R, k_1=0, k_2=j, p) \times n_j\right) \\
&+D1(m+1, n_1, ..., n_i+1, ..., n_e, R, k_1=0, k_2=i, p) \times (n_i+1) \\
&+\left(\sum_{j=1}^{e} D1(m+1, n_1, ..., n_i+1, ..., n_e, R, k_1=1, k_2=j, p) \times r_j\right) \\
&+D1(m+1, n_1, ..., n_i+1, ..., n_e, R, k_1=0, k_2=0, p) \times (m - \sum_{j=1}^{e}(n_j+r_j))] \\
&-[\sum_{j=1}^{e}\{D1(m, N, R, k_1=0, k_2=j, p) \times n_j \\
&+D1(m, N, R, k_1=1, k_2=j, p) \times r_j\} \\
&+D1(m, N, R, k_1=0, k_2=0, p) \times (m - \sum_{j=1}^{e}(n_j+r_j))] \\
&+D2(m+1, n_1, ..., n_i+1, ..., n_e, R) - D2(m, N, R) \qquad (4.4)
\end{aligned}
$$

The method of finding $D1()$ is as follows. Let $D1_t(state1, K, p, t)$ be expected income to be earned from a client at time t. Consider a very small time dt after time t such that the probability of two events taking place in time dt is negligible. The expected income after time t is the sum of two values and these are the expected income earned between time t and time $t+dt$, and the expected future income earned after time $t+dt$. The expected future income earned between time t and time $t+dt$ depends on the state of the client at time t. If the state of the client is idle, the expected income is zero. If the state is session, the expected income is <bandwidth consumed by extra client>$\times p \times dt$. If the state is waiting for session, the expected income is $-P(0) \times dt$. The expected future income after time $t+dt$ is in terms of $D1_t(state2, K', p, t+dt)$, where state2 is the state of service provider after time dt, K' is state of client after time dt.

$D1_t(state1, K, p, t)$ is written in terms of $D1_t(state2, K', p, t+dt)$ for all $state2$ and K' as given in equation 4.5. $D1_t(state1, K, p, t)$ is the sum of three sets of terms, which are separated by square brackets in the equation. The first set represents the new $D1_t()$ values because of a change in the state of the service provider from $state1$ to $state2$ in time dt and a change in the state of the client from K to K'. The second set represents no events taking place in time dt. The third set represents the income earned from the extra client in the small time dt.

$$
\begin{aligned}
&D1_t(state1, K, p, t) = \\
&[\sum_{state2} \{(\text{probability of state change from state1 to state2 in time dt}) \\
&\times \sum_{K'} (\text{probability of change in state of client from K to K' in time dt}) \\
&\times D1_t(state2, K', p, t + dt)\}] \\
&+[\{1 - \sum_{state2} (\text{probability of state change from state1 to state2 in time dt})\} \\
&\times D1_t(state1, K, p, t + dt)] \\
&+[\begin{cases} p \times k_2 dt & , k_1 = 0 \\ -P(0)dt & , k_1 = 1 \end{cases}]
\end{aligned} \tag{4.5}
$$

The probability of change in state from $state1$ to $state2$ depends on the event, which causes the change. If this change takes place with changes in more than one value (signifying two or more events), then the probability is zero. Otherwise, this probability is the rate of the event multiplied by dt. For example, if the change in state is caused by an arrival then it is the probability of an arrival in time dt, which is 'mean arrival rate' multiplied by dt. It is to be noted, that in state $state1$, this arrival rate is $\lambda(-C(state1))$ if $C(state1)$ is not 1. If it is one, then the arrival rate is assumed to be $\lambda(T-1)$, the lowest possible rate. If this change is caused by a departure then it is the mean departure rate multiplied by dt. If the change in state from $state1$ to $state2$ reduces the number of clients who were in the same state of the extra client, it might be possible that the state of the extra client would have changed; otherwise the state of the extra client after time dt is the same as before time dt. If the number of clients who were in the same state of the extra client was x and this is reduced to $(x-1)$ after time dt, then with $\frac{1}{x}$ probability, the state of the extra client changed and with $(1-\frac{1}{x})$ probability the state of the extra client did not change. The new state K' is the new state of the extra client if the state changed otherwise it is the same as K.

These cases are written in equation form below. This equation is later expanded in section 4.3.4.

$$
\begin{aligned}
& D1_t(state1, K, p, t) = \\
& [\sum_{state2} \{(\text{probability of an arrival in time dt at state1}) \\
& \times(\text{probability of state change from state1 to state2 due to an arrival}) \\
& \times \sum_{K'} (\text{probability of change in state of client from K to K' because of} \\
& \text{state change to state2}) \times D1_t(state2, K', p, t + dt)\} \\
& + \sum_{state2} \{(\text{probability that a client leaves idle state in time dt}) \\
& \times[d \times (\text{probability of state change from state1 to state2 due to a departure}) \\
& +(1 - d) \times (\text{probability of state change from state1 to state2 due to} \\
& \text{session open})] \times \sum_{K'} (\text{probability of change in state of client from K to K'} \\
& \text{because of state change to state2}) \times D1_t(state2, K', p, t + dt)\} \\
& + \sum_{state2} \{(\text{probability of a session close at state1 in time dt}) \\
& (\text{probability of state change from state1 to state2 due to the session close})\} \\
& \times \sum_{K'} (\text{probability of change in state of client from K to K' because of} \\
& \text{state change to state2}) \times D1_t(state2, K', p, t + dt)\}] \\
& +[\{1 - \sum_{state2} (\text{probability of state change from state1 to state2 in time dt})\} \\
& \times D1_t(state1, K, p, t + dt)] \\
& +[\begin{cases} p \times k_2 dt & , k_1 = 0 \\ -P(0)dt & , k_1 = 1 \end{cases}]
\end{aligned} \tag{4.6}
$$

The same way we find the value of the function $D2()$. We use a function $D2_t(state1, t)$, which represents the advantage of being in state $state1$ at time t because of income earned from newly arriving clients. $D2_t(state1, t)$ is written in terms of $D2_t(state2, t + dt)$ for some $state2$ as given below in equation 4.7. $D2_t(state1, t)$ is the sum of three sets of terms separated by square brackets in the equation. The first set is the difference between the expected income earned from an arrival in state $state1$ and the expected income earned from an arrival in state $(0, 0, 0)$ (this second term will be zero except for transitions to states $(1, x, y)$ as only one event can take place during dt. It is included for technical completeness). The second set of terms are new $D2()$ values if an event happens in state $state1$ or in state $(0, 0, 0)$. The third set of terms is the probability of no event taking place in time dt multiplied by $D2(state1, t + dt)$.

The probability of change in state from $state1$ to $state2$ from an arrival only considers the arrival events. The probability of an arrival in state $state1$ is $\lambda(-C(state1))$ or $\lambda(T - 1)$ (as already discussed) multiplied by dt.

$$
\begin{aligned}
&D2_t(state1, t) = \\
&[\begin{cases}
\{\text{Probability of an arrival in state state1 in time dt}\} \\
\times\{\text{-P(1)}\} & , C(state1) = 1 \\
\sum_{state2} \{\text{Probability of state change from state1 to state2} \\
\ \text{in time dt due to an arrival}\} \\
\times\{\text{D1(state2,K=state of new client,price(-C(state1)))}\} & , C(state1) \leq 0
\end{cases} \\
&-\sum_{state2} \{\text{(Probability of state change from (0,0,0) to state2 in time dt} \\
&\ \text{due to an arrival)} \\
&\times\text{D1(state2,K=state of new client,price(-C(0,0,0)))}\}] \\
&+[\sum_{state2} \{\text{(Probability of state change from state1 to state2 in time dt)} \\
&\times D2_t(state2, t + dt) \\
&-\text{(Probability of state change from (0,0,0) to state2 in time dt)} \\
&\times D2_t(state2, t + dt)\}] \\
&+[\{1 - \sum_{state2} \text{(probability of state change from state1 to state2 in time dt)}\} \\
&\times D2_t(state1, t + dt)]
\end{aligned} \tag{4.7}
$$

The next step is find the values of $D1()$ and $D2()$ from equations 4.6 and equation 4.7. The method to find $D1()$ is to rearrange everything in equation 4.6 such that the $\{D1_t(state1, K, p, t) - D1_t(state1, K, p, t + dt)\}\ /dt$ comes to the left hand side. At steady state, this rate of change of $D1_t(state1, K, p, t)$ becomes zero and time is removed from the equations. We remove time from the function $D1_t()$ and the function $D1_t(state, K, p, t)$ becomes $D1(state, K, p)$ for all values of $state$ and K. In this way we get equations of $D1()$. The same way from equation 4.7, we get equations of $D2()$.

4.3.4 Expanded Equations of D1() and D2()

The detailed equations of $D1()$ and $D2()$ are given below. First we have described equation of $D1()$ and then equation of $D2()$. The equations of $D1()$ and $D2()$ are picked from above and then the equations are expanded in details.

4.3.4.1 Expanded Equations of D1

We have got the following equation as equation 4.6.

$$\begin{aligned}
&D1_t(state1, K, p, t) = \\
&[\sum_{state2}\{(\text{probability of an arrival in time dt at state1}) \\
&\times(\text{probability of state change from state1 to state2 due to an arrival}) \\
&\times\sum_{K'}(\text{probability of change in state of client from K to K' because of} \\
&\text{state change to state2}) \times D1_t(state2, K', p, t+dt)\}] \\
&+[\sum_{state2}\{(\text{probability that a client leaves idle state in time dt}) \\
&\times\{d\times(\text{probability of state change from state1 to state2 due to a departure}) \\
&+(1-d)\times(\text{probability of state change from state1 to state2 due to a} \\
&\text{session open})\}\times\sum_{K'}(\text{probability of change in state of client from K to K'} \\
&\text{because of state change to state2}) \times D1_t(state2, K', p, t+dt)\}] \\
&+[\sum_{state2}\{(\text{probability of a session close at state1 in time dt})\times \\
&(\text{probability of state change from state1 to state2 due to the session close}) \\
&\times\sum_{K'}(\text{probability of change in state of client from K to K' because of} \\
&\text{state change to state2}) \times D1_t(state2, K', p, t+dt)\}] \\
&+[\{1-\sum_{state2}(\text{probability of state change from state1 to state2 in time dt})\} \\
&\times D1_t(state1, K, p, t+dt)] \\
&+[\begin{cases} p\times k_2 dt & ,k_1 = 0 \\ -P(0)dt & ,k_1 = 1\end{cases}]
\end{aligned} \tag{4.8}$$

The left hand side of the equation contains $D1_t(state1, K, p, t)$. The right hand side of the equation is grouped into five groups as given below. It is the sum of these five groups. These groups represent respectively, the arrival of a new client, a client leaving idle state, a client leaving session state, no change in state, and income earned at time dt.

1. Arrival

$$\sum_{state2} \{(\text{probability of an arrival in time dt at state1})$$
$$\times(\text{probability of state change from state1 to state2 due to an arrival})$$
$$\times \sum_{K'} (\text{probability of change in state of client from K to K' because of}$$
$$\text{state change to state2}) \times D1_t(state2, K', p, t + dt)\}$$

Probability of an arrival in time dt is $\lambda(-C(m, N, R))dt$ if an arrival is accepted and $\lambda(T - 1)dt$ if an arrival is rejected. If the arrival is accepted ($C(m, N, R)<1$), the arriving client will request for i unit of bandwidth with probability $G(i)$. If bandwidth is available it will increase the number of clients in session with i unit of bandwidth (n_i) and if bandwidth is not available it will increase the number of clients waiting for i unit of bandwidth (r_i). If the arrival is rejected ($C(m, N, R)=1$), there is no change in state. The function D1() does not store the penalty when a new arriving client is rejected and instead it is stored in function $D2()$ which will be explained later. Probability of change in state of client from K to K' is one when K' is K and zero otherwise. In other words, the state of the extra client will not change due to an arrival. The following expression is obtained.

$$\left(\begin{cases}\lambda(-C(m,N,R)) & ,C(m,N,R) \leq 0 \\ \lambda(T-1) & ,C(m,N,R) = 1\end{cases}\right) dt$$

$$\begin{cases}\sum_{i=1}^{e} G(i)\times \\ \begin{cases} D1_t(m+1, n_1, ..., n_i+1, ..., n_e, R, \\ K, p, t+dt) & , B - \sum i \times n_i \geq i \\ D1_t(m+1, N, r_1, ..., r_i+1, ..., r_e, \\ K, p, t+dt) & , otherwise \end{cases} & ,C(m,N,R) < 1 \\ \\ D1_t(m, N, R, K, p, t+dt) & ,C(m,N,R) = 1 \end{cases}$$

2. Leaving idle state

$+[\sum_{state2}$ {(probability that a client leaves idle state in time dt)
$\times\{d \times$ (probability of state change from state1 to state2 due to a departure)
$+(1-d) \times$ (probability of state change from state1 to state2 due to a session open)} $\times \sum_{K'}$ (probability of change in state of client from K to K' because of state change to state2) $\times D1_t(state2, K', p, t+dt)\}]$

When the difference between $state1$ and $state2$ is that $state2$, can be reached by a departure from $state1$, the term "probability of state change from $state1$ to $state2$ due to a departure" will be positive and the term "probability of state change from $state1$ to $state2$ due to a session open" will be zero. When the value of $state2$ is such that it can be reached by opening a session at $state1$, the term "probability of state change from $state1$ to $state2$ due to a session open" will be positive and the term "probability of state change from $state1$ to $state2$ due to a departure" will be zero. In other cases both the terms will be zero. The expression is expanded as given below:

$\sum_{state2}$ (probability that a client leaves idle state in time dt)
$\times\{d \times$ (probability of state change from state1 to state2 due to a departure)
$\times \sum_{K'}$ (probability of change in state of client from K to K'
because of state change to state2) $\times D1_t(state2, K, p, t + dt)$
$+(1 - d) \times$ (probability of state change from state1 to state2 due to session open) $\times \sum_{K'}$ (probability of change in state of client from K to K'
because of state change to state2) $\times D1_t(state2, K', p, t + dt)\}$

When a client leaves idle state, the client will disconnect with probability d or try to open a session with probability $(1 - d)$. If the extra client was in idle state and any other client disconnects, the new state will be $(m - 1, N, R)$ and the future income by the extra client is given by $D1_t(m - 1, N, R, K, p, t + dt)$. If the extra client disconnects, it is multiplied by zero and therefore the case when the extra client disconnects is not given below. If the extra client was not in idle state and then a client leaves, the state will become $(m-1, N, R)$ and the expected income by the extra client is given by $D1_t(m - 1, N, R, K, p, t + dt)$.

The other possibility is that the client tries to open a session with probability $(1 - d)$. If the extra client was in idle state then with some probability a client other than the extra client opens a session and with some probability the extra client opens a session. The difference between the two values is in the value of K. If the extra client opens a session, the value of K will change. If the extra client was not in idle state, any client opening a session will not change the value of K. The following expression is obtained.

(probability that a client leaves idle state in time dt)

$\times\{$

$$\times[d \times \begin{cases} \text{Probability that a client other} \\ \text{than the extra client leaves idle state} \\ \times D1_t(m-1, N, R, K, p, t+dt) & \text{,The extra client was idle} \\ D1_t(m-1, N, R, K, p, t+dt) & \text{, The extra client was not idle} \end{cases}$$

$+(1-d)\times$

$$\begin{cases} \text{Probability that a client other} \\ \text{than the extra client leaves idle state} \\ \times \sum_{i=1}^{e} G(i) \times \\ \left(\begin{cases} D1_t(m, n_1, n_2, .n_i+1.., n_e, R, \\ K, p, t+dt) & , B - \sum x \times n_x \geq i \\ D1_t(m, N, r_1.r_i+1..r_e, \\ K, p, t+dt) & , otherwise \end{cases} \right) \\ \text{+Probability that the extra} \\ \text{client leaves idle state} \\ \times \sum_{i=1}^{e} G(i) \times \\ \begin{cases} D1_t(m, n_1, n_2, .n_i+1.., n_e, R, \\ k_1 = 0, k_2 = i, p, t+dt) & , B - \sum x \times n_x \geq i \\ D1_t(m, N, r_1.r_i+1..r_e, \\ k_1 = 1, k_2 = i, p, t+dt) & , otherwise \end{cases} \} & \text{,The extra client was idle} \\ \times \sum_{i=1}^{e} G(i) \times \\ \left(\begin{cases} D1_t(m, n_1, n_2, .n_i+1.., n_e, R, \\ K, p, t+dt) & , B - \sum x \times n_x \geq i \\ D1_t(m, N, r_1.r_i+1..r_e, \\ K, p, t+dt) & , otherwise \end{cases} \right) & \text{, The extra client was not idle} \end{cases}$$

The probability that a client leaves idle state is $S(0) \times (m - \sum_{i=1}^{e}(n_i + r_i))dt$. If the extra client was in idle state, the probability that a client other than the extra client leaves idle state is $\frac{\text{number of idle clients -1}}{\text{number of idle clients}}$ and it becomes $\frac{m-\sum_{i=1}^{e}(n_i+r_i)-1}{m-\sum_{i=1}^{e}(n_i+r_i)}$; and the probability that the extra client leaves idle state is $\frac{1}{\text{number of idle clients}}$ and it becomes $\frac{1}{m-\sum_{i=1}^{e}(n_i+r_i)}$. If k_1 and k_2 are zero, it means that the extra client was idle and otherwise the extra client was not idle.

$$S(0) \times (m - \sum_{i=1}^{e}(n_i + r_i))dt \times$$

$$[d \times \begin{cases} \frac{m-\sum_{i=1}^{e}(n_i+r_i)-1}{m-\sum_{i=1}^{e}(n_i+r_i)} \times D1_t(m-1, N, R, \\ K, p, t+dt) & , k_1 = 0, k_2 = 0 \\ \\ D1_t(m-1, N, R, K, p, t+dt) & , k_1 = 0, k_2 \neq 0 \end{cases}$$

$$+(1-d) \times$$

$$
\begin{cases}
\frac{m-\sum_{i=1}^{e}(n_i+r_i)-1}{m-\sum_{i=1}^{e}(n_i+r_i)}\{\sum_{i=1}^{e} G(i)\times \\
\left(\begin{cases} D1_t(m, \\ n_1, n_2, .n_i+1.., n_e, \\ R, K, p, t+dt) & , B-\sum x \times n_x \geq i \\ D1_t(m, \\ N, r_1..r_i+1..r_e, \\ K, p, t+dt) & , otherwise \end{cases}\right) \\
\} + \frac{1}{m-\sum_{i=1}^{e}(n_i+r_i)} \sum_{i=1}^{e}\{G(i)\times \\
\left(\begin{cases} D1_t(m, \\ n_1, n_2, .n_i+1.., n_e, \\ R, k_1=0, k_2=i, \\ p, t+dt) & , B-\sum x \times n_x \geq i \\ D1_t(m, \\ N, r_1..r_i+1..r_e, \\ k_1=1, k_2=i, \\ p, t+dt) & , otherwise \end{cases}\right) \} & , m-\sum_{i=1}^{e}(n_i+r_i) \neq 0 \quad] \\
& , k_1=0, k_2=0 \\
\sum_{i=1}^{e} G(i)\times \\
\left(\begin{cases} D1_t(m, \\ n_1, n_2, .n_i+1.., n_e, \\ R, K, p, t+dt) & , B-\sum x \times n_x \geq i \\ D1_t(m, \\ N, r_1..r_i+1..r_e, \\ K, p, t+dt) & , otherwise \end{cases}\right) & , otherwise, \\
& , m-\sum_{i=1}^{e}(n_i+r_i) \neq 0 \\
0 & , m-\sum_{i=1}^{e}(n_i+r_i) = 0
\end{cases}
$$

3. Closing of session

$$\sum_{state2} \{(\text{probability of a session close at state1 in time dt})$$
$$(\text{probability of state change from state1 to state2 due to the session close})\}$$
$$\times \sum_{K'} (\text{probability of change in state of client from K to K' because of state change to state2}) \times D1_t(state2, K', p, t + dt)\}]$$

The probability of session close for a client in session with i unit bandwidth is $n_i \times S(i)dt$. It is multiplied by the $D1_t()$ value when the session is closed. However, when a session is closed, some bandwidth is released and therefore among the clients waiting for session, few may get bandwidth. We assume a temporary function $D1'_t()$ which represents the value of $D1_t()$ when some checking needs to be done for available bandwidth and if bandwidth is available, bandwidth may be given to the waiting clients.

If the extra client was in session (given by $k_1 = 0, k_2 \neq 0$), there are two possibilities: the extra client leaves the session or any other client leaves its session. If the extra client was in session, the extra client consumes k_2 units of bandwidth. The summation $\sum_{i=1}^{e,k_2 \neq i,n_i \neq 0}$ in the equation below represents the case when we consider that value of i which is different from the bandwidth used by the extra client and the number of clients in session with i unit bandwidth is not zero. The next term $(n_{k_2} - 1) \times S(k_2)dt \times D1'_t(m, n_1, n_2, .n_{k_2} - 1.., n_e, R, K, p, t + dt)$ is for the clients other than the extra client consuming the same bandwidth as the extra client. The next term $S(k_2)dt \times D1'_t(m, n_1, n_2, .n_{k_2} - 1.., n_e, R, k_1 = 0, k_2 = 0, p, t + dt)$ is for the extra client closing a session and therefore, the value of k_1 and k_2 changes. If the extra client was not in session, any client might close its session and the value of the expression will be $\sum_{i=1}^{e,n_i \neq 0} n_i \times S(i)dt \times D1'_t(m, n_1, n_2, .n_i - 1.., n_e, R, K, p, t + dt)$.

$$\begin{cases} \sum_{i=1}^{e,k_2\neq i,n_i\neq 0}\{n_i \times S(i)dt\times \\ \times D1'_t(m,n_1,n_2,.n_i-1..,n_e,R,K,p,t+dt)\}+ \\ (n_{k_2}-1)\times S(k_2)dt\times \\ D1'_t(m,n_1,n_2,.n_{k_2}-1..,n_e,R,K,p,t+dt) \\ +S(k_2)dt\times \\ D1'_t(m,n_1,n_2,.n_{k_2}-1..,n_e,R,k_1=0,k_2=0,p,t+dt) & ,k_1=0,k_2\neq 0 \\ \\ \sum_{i=1}^{e,n_i\neq 0} n_i\times S(i)dt\times \\ D1'_t(m,n_1,n_2,.n_i-1..,n_e,R,K,p,t+dt) & ,otherwise \end{cases}$$

where $D1'_t()$ in terms of $D1_t()$ is given below. As mentioned earlier, $D1'_t()$ is a temporary function that represents the value of $D1_t()$ when some checking needs to be done for available bandwidth and if available, it may be given to the waiting clients. The first possibility is checking for a client waiting for 1 unit bandwidth and whether this much bandwidth is available. This possibility is represented by $,B-\sum_{j=1}^{e} j\times n_j \geq 1, r_1\neq 0$. Similarly the possibility of checking for a client waiting for i unit bandwidth and whether this much bandwidth is available is represented by $,B-\sum j\times n_j \geq i, r_i\neq 0$. If for any value of i, the condition $B-\sum j\times n_j \geq i, r_i\neq 0$ is true, it means that bandwidth may be given to a client requesting for i unit bandwidth. After giving the bandwidth, the value of r_i decreases and the value of n_i increases. However, it is again possible that the extra client might have changed state and therefore we consider the possibilities of the extra client being in waiting state and going to session. If $k_1=1$ and $k_2=i$, the extra client is waiting for i unit bandwidth. In this case, with $\frac{r_1-1}{r_1}$ probability, a client other than the extra client gets bandwidth and with $\frac{1}{r_1}$ probability, the extra client gets bandwidth. If the extra client goes to session, the value of k_1 becomes 0 and the value of k_2 remains unchanged.

Once this change is done, still the function $D1'_t()$ does not become $D1_t()$. It is because this checking is done recursively until it is not possible to give bandwidth to any waiting client. In that case, $D1'_t(m,N,R,K,p,t)$ becomes $D1_t(m,N,R,K,p,t)$.

$$D1'_t(m,N,R,K,p,t) =$$

$$\begin{cases}
\left(\begin{cases}
D1'_t(m,n_1+1,n_2,...,n_e, \\
r_1-1...r_e, \\
k_1=0,k_2=1,p,t)\times\frac{1}{r_i} \\
+D1'_t(m,n_1+1,n_2,...,n_e, \\
r_1-1...r_e,K,p,t)\times\frac{r_1-1}{r_1} & ,k_1=1 \\
 & ,k_2=1 \\
D1'_t(m,n_1+1,n_2,...,n_e, \\
r_1-1...r_e,K,p,t) & ,otherwise
\end{cases}\right) & ,B-\sum_{j=1}^{e} j\times n_j \geq 1 \\
 & ,r_1\neq 0 \\
... \\
\left(\begin{cases}
D1'_t(m,n_1,n_2,..n_i+1.,n_e, \\
r_1.r_i-1..r_e, \\
k_1=0,k_2=i,p,t)\times\frac{1}{r_i} \\
+D1'_t(m,n_1,n_2,..n_i+1.,n_e, \\
r_1.r_i-1..r_e,K,p,t)\times\frac{r_i-1}{r_i} & ,k_1=1 \\
 & ,k_2=i \\
D1'_t(m,n_1,n_2,..n_i+1.,n_e, \\
r_1.r_i-1..r_e,K,p,t) & ,otherwise
\end{cases}\right) & ,otherwise \\
 & ,B-\sum j\times n_j \geq i \\
 & ,r_i\neq 0 \\
... & . \\
\left(\begin{cases}
D1'_t(m,n_1,n_2,...,n_e+1 \\
,r_1...r_e-1, \\
k_1=0,k_2=e,p,t)\times\frac{1}{r_i} \\
+D1'_t(m,n_1,n_2,...,n_e+1, \\
r_1...r_e-1,K,p,t)\times\frac{r_e-1}{r_e} & ,k_1=1 \\
 & ,k_2=e \\
D1'_t(m,n_1,n_2,...,n_e+1, \\
r_1...r_e-1,K,p,t) & ,otherwise
\end{cases}\right) & ,otherwise \\
 & ,B-\sum j\times n_j \geq e \\
 & ,r_e\neq 0 \\
D1_t(m,N,R,K,p,t) & ,otherwise
\end{cases}$$

4. No change in state

$$[\{1-\sum_{state2}\text{(probability of state change from state1 to state2 in time dt)}\}$$
$$\times D1_t(state1, K, p, t+dt)]$$

After substituting values, it becomes.

$$(1-\left(\begin{cases}\lambda(-C(m,N,R)) & ,C(m,N,R)\leq 0\\ \lambda(T-1) & ,C(m,N,R)=1\end{cases}\right)dt - S(0)\times(m-\sum_{i=1}^{e}(n_i+r_i))dt$$
$$-\sum_{i=1}^{e}n_i\times S(i)dt)\times D1_t(m,N,R,K,p,t+dt)$$

5. Income or loss during time dt

$$\begin{cases}p\times k_2 dt & ,k_1=0\\ -P(0)dt & ,k_1=1\end{cases}$$

The equations are rearranged to bring make the left hand side become the expression: $\frac{D1_t(m,N,R,K,p,t)-D1_t(m,N,R,K,p,t+dt)}{dt}$. This is done by bringing the $D1_t(m,N,R,K,p,t+dt)$ term from the forth part 'No change in state' to the left hand side of the equation and then dividing everything by dt.

$\frac{D1_t(m,N,R,K,p,t)-D1_t(m,N,R,K,p,t+dt)}{dt}$ is equal to the sum of the following terms.

1. Arrival

$$\left(\begin{cases}\lambda(-C(m,N,R)) & ,C(m,N,R)\leq 0\\ \lambda(T-1) & ,C(m,N,R)=1\end{cases}\right)$$

$$\begin{cases}\sum_{i=1}^{e} G(i)\times & \\ \begin{cases} D1_t(m+1,n_1,.n_i+1..,n_e,R & \\ ,K,p,t+dt) & ,B-\sum i\times n_i\geq i\\ D1_t(m+1,N,r_1.r_i+1..r_e & \\ ,K,p,t+dt) & ,otherwise\end{cases} & ,C(m,N,R)<1\\ D1_t(m,N,R,K,p,t+dt) & ,otherwise\\ & ,C(m,N,R)=1\end{cases}$$

2. Leaving idle state

$$S(0)\times(m-\sum_{i=1}^{e}(n_i+r_i))\times$$

$$[d\times\begin{cases}\frac{m-\sum_{i=1}^{e}(n_i+r_i)-1}{m-\sum_{i=1}^{e}(n_i+r_i)}\times D1_t(m-1,N,R,K,p,t+dt) & ,k_1=0,k_2=0\\ D1_t(m-1,N,R,K,p,t+dt) & ,k_1=0,k_2\neq 0\end{cases}$$

$$+(1-d)\times$$

$$
\begin{cases}
\frac{m-\sum_{i=1}^{e}(n_i+r_i)-1}{m-\sum_{i=1}^{e}(n_i+r_i)}\{\sum_{i=1}^{e} G(i)\times \\
\left(\begin{cases} D1_t(m, \\ n_1, n_2, .n_i+1.., n_e, \\ R, K, p, t+dt) & , B-\sum x\times n_x \geq i \\ D1_t(m, N, r_1.r_i+1..r_e, \\ K, p, t+dt) & , otherwise \end{cases}\right) \\
\} + \frac{1}{m-\sum_{i=1}^{e}(n_i+r_i)}\sum_{i=1}^{e}\{G(i)\times \\
\left(\begin{cases} D1_t(m, \\ n_1, n_2, .n_i+1.., n_e, \\ R, k_1=0, k_2=i, \\ p, t+dt) & , B-\sum x\times n_x \geq i \\ D1_t(m, N, r_1.r_i+1..r_e, \\ k_1=1, k_2=i, p, t+dt) & , otherwise \end{cases}\right)\} & , m-\sum_{i=1}^{e}(n_i+r_i)\neq 0 \quad] \\
 & , k_1=0, k_2=0 \\
\sum_{i=1}^{e} G(i)\times \\
\left(\begin{cases} D1_t(m, \\ n_1, n_2, .n_i+1.., n_e, \\ R, K, p, t+dt) & , B-\sum x\times n_x \geq i \\ D1_t(m, N, \\ r_1.r_i+1..r_e, \\ K, p, t+dt) & , otherwise \end{cases}\right) & , otherwise, \\
 & , m-\sum_{i=1}^{e}(n_i+r_i)\neq 0 \\
0 & , m-\sum_{i=1}^{e}(n_i+r_i)=0
\end{cases}
$$

3. Closing of session

$$\begin{cases} \sum_{i=1}^{e,k_2\neq i,n_i\neq 0}\{n_i \times S(i)\times \\ \times D1'_t(m,n_1,n_2,.n_i-1..,n_e,R,K,p,t+dt)\}+ \\ (n_{k_2}-1)\times S(k_2)\times \\ D1'_t(m,n_1,n_2,.n_{k_2}-1..,n_e,R,K,p,t+dt) \\ +S(k_2)\times \\ D1'_t(m,n_1,n_2,.n_{k_2}-1..,n_e,R,k_1=0,k_2=0,p,t+dt) & ,k_1=0,k_2\neq 0 \\ \\ \sum_{i=1}^{e,n_i\neq 0} n_i \times S(i)\times \\ D1'_t(m,n_1,n_2,.n_i-1..,n_e,R,K,p,t+dt) & ,otherwise \end{cases}$$

where $D1'_t()$ is remains the same and therefore, we have not written it again.

4. No change in state

$$-(\left(\begin{cases}\lambda(-C(m,N,R)) & ,C(m,N,R)\leq 0 \\ \lambda(T-1) & ,C(m,N,R)=1\end{cases}\right) + S(0)\times(m-\sum_{i=1}^{e}(n_i+r_i)) \\ +\sum_{i=1}^{e} n_i \times S(i)) \times D1_t(m,N,R,K,p,t+dt)$$

5. Income or loss during time dt

$$\begin{cases} p\times k_2 & ,k_1=0 \\ -P(0) & ,k_1=1 \end{cases}]$$

At steady state the left hand side becomes zero and time is removed from the equations. $D1_t()$ becomes $D1()$ (and $D1'_t()$ becomes $D1'()$). Therefore the sum of the following terms is zero for every valid value of (m,N,R,K,p).

1. Arrival

$$\left(\begin{cases}\lambda(-C(m,N,R)) & ,C(m,N,R)\leq 0\\ \lambda(T-1) & ,C(m,N,R)=1\end{cases}\right)$$

$$\begin{cases}\sum_{i=1}^{e} G(i)\times \\ \begin{cases}D1(m+1,n_1,.n_i+1..,n_e,R,K,p) & ,B-\sum i\times n_i\geq i\\ D1(m+1,N,r_1.r_i+1..r_e,K,p) & ,otherwise\end{cases} & ,C(m,N,R)<1\\ \\ D1(m,N,R,K,p) & ,otherwise\\ & ,C(m,N,R)=1\end{cases}$$

2. Leaving idle state

$$S(0)\times(m-\sum_{i=1}^{e}(n_i+r_i))\times$$

$$[d\times\begin{cases}\frac{m-\sum_{i=1}^{e}(n_i+r_i)-1}{m-\sum_{i=1}^{e}(n_i+r_i)}\times D1(m-1,N,R,K,p) & ,k_1=0,k_2=0\\ \\ D1(m-1,N,R,K,p) & ,k_1=0,k_2\neq 0\end{cases}$$

$$+(1-d)\times$$

$$
\begin{cases}
\frac{m-\sum_{i=1}^{e}(n_i+r_i)-1}{m-\sum_{i=1}^{e}(n_i+r_i)}\{\sum_{i=1}^{e} G(i)\times \\
\left(\begin{cases} D1(m, \\ n_1, n_2, .n_i+1.., n_e, R, \\ K, p) & , B-\sum x\times n_x \geq i \\ D1(m, \\ N, r_1.r_i+1..r_e, \\ K, p) & , otherwise \end{cases}\right) \\
\} + \frac{1}{m-\sum_{i=1}^{e}(n_i+r_i)}\sum_{i=1}^{e}\{G(i)\times \\
\left(\begin{cases} D1(m, \\ n_1, n_2, .n_i+1.., n_e, R, \\ k_1=0, k_2=i, p) & , B-\sum x\times n_x \geq i \\ D1(m, \\ N, r_1.r_i+1..r_e, \\ k_1=1, k_2=i, p) & , otherwise \end{cases}\right)\} & , m-\sum_{i=1}^{e}(n_i+r_i)\neq 0 \\
 & , k_1=0, k_2=0 \\
\sum_{i=1}^{e} G(i)\times \\
\left(\begin{cases} D1(m, \\ n_1, n_2, .n_i+1.., n_e, R, \\ K, p) & , B-\sum x\times n_x \geq i \\ D1(m, \\ N, r_1.r_i+1..r_e, \\ K, p) & , otherwise \end{cases}\right) & , otherwise, \\
 & , m-\sum_{i=1}^{e}(n_i+r_i)\neq 0 \\
0 & , m-\sum_{i=1}^{e}(n_i+r_i)=0
\end{cases}
]
$$

3. Closing of session

$$\begin{cases} \sum_{i=1}^{e,k_2\neq i,n_i\neq 0}\{n_i \times S(i)\times \\ \times D1'(m,n_1,n_2,.n_i-1..,n_e,R,K,p)\}+ \\ (n_{k_2}-1)\times S(k_2)\times D1'(m,n_1,n_2,.n_{k_2}-1..,n_e,R,K,p) \\ +S(k_2)\times D1'(m,n_1,n_2,.n_{k_2}-1..,n_e,R,k_1=0,k_2=0,p) & ,k_1=0,k_2\neq 0 \\ \\ \sum_{i=1}^{e,n_i\neq 0} n_i\times S(i)\times D1'(m,n_1,n_2,.n_i-1..,n_e,R,K,p) & ,otherwise \end{cases}$$

where $D1'()$ is given below

$$D1'(m,N,R,K,p) =$$

$$\begin{cases}
\left(\begin{cases} D1'(m,n_1+1,n_2,...,n_e, \\ r_1-1...r_e, \\ k_1=0,k_2=1,p)\times\frac{1}{r_i} \\ +D1'(m,n_1+1,n_2,...,n_e, \\ r_1-1...r_e,K,p)\times\frac{r_1-1}{r_1} & ,k_1=1 \\ & ,k_2=1 \\ D1'(m,n_1+1,n_2,...,n_e, \\ r_1-1...r_e,K,p) & ,otherwise \end{cases}\right) & ,B-\sum_{j=1}^{e} j\times n_j \geq 1 \\ & ,r_1 \neq 0 \\
... & . \\
\left(\begin{cases} D1'(m,n_1,n_2,..n_i+1.,n_e, \\ r_1.r_i-1..r_e, \\ k_1=0,k_2=i,p)\times\frac{1}{r_i} \\ +D1'(m,n_1,n_2,..n_i+1.,n_e, \\ r_1.r_i-1..r_e,K,p)\times\frac{r_i-1}{r_i} & ,k_1=1 \\ & ,k_2=i \\ D1'(m,n_1,n_2,..n_i+1.,n_e, \\ r_1.r_i-1..r_e,K,p) & ,otherwise \end{cases}\right) & ,otherwise \\
... & ,B-\sum j\times n_j \geq i \\ & ,r_i \neq 0 \\
. & . \\
\left(\begin{cases} D1'(m,n_1,n_2,...,n_e+1, \\ r_1...r_e-1, \\ k_1=0,k_2=e,p)\times\frac{1}{r_i} \\ +D1'(m,n_1,n_2,...,n_e+1, \\ r_1...r_e-1,K,p)\times\frac{r_e-1}{r_e} & ,k_1=1 \\ & ,k_2=e \\ D1'(m,n_1,n_2,...,n_e+1, \\ r_1...r_e-1,K,p) & ,otherwise \end{cases}\right) & ,otherwise \\
 & ,B-\sum j\times n_j \geq e \\ & ,r_e \neq 0 \\
D1(m,N,R,K,p) & ,otherwise
\end{cases}$$

4. No change in state

$$-(\left(\begin{cases}\lambda(-C(m,N,R)) & ,C(m,N,R)\leq 0\\ \lambda(T-1) & ,C(m,N,R)=1\end{cases}\right)+S(0)\times(m-\sum_{i=1}^{e}(n_i+r_i)) + \sum_{i=1}^{e} n_i\times S(i))\times D1(m,N,R,K,p)$$

5. Income or loss

$$[\begin{cases}p\times k_2 & ,k_1=0\\ -P(0) & ,k_1=1\end{cases}]$$

4.3.4.2 Expanded equation of D2

We have got the following equation as equation 4.7.

$$\begin{aligned}
&D2_t(state1,t)=\\
&[\begin{cases}\{\text{Probability of an arrival in state state1 in time dt}\}\\ \times\{\text{-P(1)}\} & ,C(state1)=1\\ \sum_{state2}\{\text{Probability of state change from state1 to state2}\\ \text{ in time dt due to an arrival}\}\\ \times\{\text{D1(state2,K=state of new client,price(-C(state1)))}\} & ,C(state1)\leq 0\end{cases}\\
&-\sum_{state2}\{(\text{Probability of state change from (0,0,0) to state2 in time dt}\\
&\text{ due to an arrival})\\
&\times\text{D1(state2,K=state of new client,price(-C(0,0,0)))}\}]\\
&+[\sum_{state2}\{(\text{Probability of state change from state1 to state2 in time dt})\\
&\times D2_t(state2,t+dt)\\
&-(\text{Probability of state change from (0,0,0) to state2 in time dt})\\
&\times D2_t(state2,t+dt)\}]\\
&+[\{1-\sum_{state2}(\text{probability of state change from state1 to state2 in time dt})\}\\
&\times D2_t(state1,t+dt)]
\end{aligned} \tag{4.9}$$

It is expanded by separating the multiple ways by which state can change (like in equation 4.8). The multiple ways are arrival, leaving idle state and then opening a session or departing, and session close. The expanded equation is given below.

$$
\begin{aligned}
&D2_t(state1, t) = \\
&[\begin{cases} \text{\{Probability of an arrival in state state1 in time dt\}} & \\ \times\text{\{-P(1)\}} & , C(state1) = 1 \\ \sum_{state2} \text{\{Probability of state change from state1 to state2} & \\ \text{ in time dt due to an arrival\}} & \\ \times\text{\{D1(state2,K=state of new client,price(-C(state1)))\}} & , C(state1) \leq 0 \end{cases} \\
&- \sum_{state2} \text{\{(Probability of state change from (0,0,0) to state2 in time dt} \\
&\text{ due to an arrival)} \\
&\times\text{D1(state2,K=state of new client,price(-C(0,0,0)))\}]} \\
&+[\sum_{state2} \text{\{(Probability of an arrival that chages state from state1 to state2} \\
&\text{in time dt)} \times D2_t(state2, t + dt) \\
&-\text{(Probability of an arrival that changes state from (0,0,0) to state2 in time dt)} \\
&\times D2_t(state2, t + dt)\}] \\
&+[\sum_{state2} \text{\{(Probability of a client leaving idle state that} \\
&\text{changes state from state1 to state2 in time dt)} \\
&\times D2_t(state2, t + dt)] \\
&+[\sum_{state2} \text{\{(Probability of a client leaving session state that} \\
&\text{changes state from state1 to state2 in time dt)} \\
&\times D2_t(state2, t + dt)] \\
&+[\{1 - \sum_{state2} \text{(probability of state change from state1 to state2 in time dt)\}} \\
&\times D2_t(state1, t + dt)]
\end{aligned}
\tag{4.10}
$$

During time dt many state changes are possible. The right hand side of the equation is grouped into four groups. These groups respectively the arrival of a new client, a client leaving idle state, a client leaving session state, and no change in state.

The function $D2_t(m, N, R)$ represents the advantage of a service provider in being in state (m, N, R) instead of state $(0, 0, 0)$ at time t due to income earned from newly arriving clients. Therefore, we also consider the state change at state $(0, 0, 0)$. At state $(0, 0, 0)$, only an arrival can take place. Since no client is connected at state $(0, 0, 0)$, state changes such as a client leaving idle state and a client closing a session are not possible. If no state change takes place at state $(0, 0, 0)$, its effect is also zero because it is the product of two terms: <probability of no state change> and $-D2(0, 0, 0, t + dt)$. $D2(0, 0, 0, t + dt)$ is advantage of being in state $(0, 0, 0)$ as compared to state $(0, 0, 0)$ at time $t + dt$ and therefore it is zero. Therefore, we use state $(0, 0, 0)$ only when we consider arrival of a new client.

1. Arrival

$$
\left[\begin{cases} \text{\{Probability of an arrival in state state1 in time dt\}} & \\ \times\text{\{-P(1)\}} & , C(state1) = 1 \\ \sum_{state2} \text{\{Probability of state change from state1 to} & \\ \text{state2 in time dt due to an arrival\}} & \\ \times\text{\{D1(state2,K=state of new client,price(-C(state1)))\}} & , C(state1) \leq 0 \end{cases}\right.
$$

$$
- \sum_{state2} \text{\{(Probability of state change from (0,0,0) to state2 in time dt}
$$
$$
\text{due to an arrival)}
$$
$$
\times\text{D1(state2,K=state of new client,price(-C(0,0,0)))\}]}
$$
$$
+[\sum_{state2} \text{\{(Probability of an arrival that chages state from state1 to state2}
$$
$$
\text{in time dt)} \times D2_t(state2, t + dt)
$$
$$
-\text{(Probability of an arrival that changes state from (0,0,0) to state2}
$$
$$
\text{in time dt)} \times D2_t(state2, t + dt)\}]
$$

As already mentioned, probability of an arrival at state (m, N, R) in time dt is $\lambda(-C(m, N, R))dt$ if an arrival is accepted and $\lambda(T - 1)dt$ if an arrival is rejected. If the arrival is accepted ($C(m, N, R){<}1$), the arriving client will request for i unit of bandwidth with probability $G(i)$. If bandwidth is available it will increase the number of clients in session with i unit of bandwidth (n_i) and if bandwidth is not available it will increase the number of clients waiting for i unit of bandwidth (r_i). If the arrival is rejected ($C(m, N, R){=}1$), there is no change in state. Similarly, the probability of an arrival at state $(0, 0, 0)$ in time dt is $\lambda(-C(0,0,0))dt$. We assume that at state $(0,0,0)$, an arriving client is not rejected.

As already mentioned, the function $D2_t(m,N,R)$ represents the advantage of a service provider in being in state (m,N,R) instead of state $(0,0,0)$ at time t due to income earned from newly arriving clients. When an arrival takes place in time dt, its effect is the sum of two set of terms: the expected income earned from the arrival and the expected future congestion increase due to the arrival. When an arrival takes place at state (m,N,R), the expected income earned due to the arrival is calculated using function $D1()$. It is the difference between the expected income earned at state (m,N,R) when an arrival takes place at time dt and the expected income earned at state $(0,0,0)$ when an arrival takes place at time dt. We assume that the value of $D1()$ is calculated before calculating $D2()$ and its value is available. The expected loss of income from clients arriving in time dt is in terms of $D2()$. It is the difference between the new $D2()$ value when an arrival takes place at state (m,N,R) in time dt and the new $D2()$ value when an arrival takes place at state $(0,0,0)$ in time dt.

$$\left(\begin{cases}\lambda(-C(m,N,R)) & ,C(m,N,R)\le 0\\ \lambda(T-1) & ,C(m,N,R)=1\end{cases}\right) dt \times$$

$$\begin{cases} \sum_{i=1}^{e} G(i)\times & \\ \begin{cases} D1(m+1,n_1,.n_i+1..,n_e,R, & \\ k_1=0,k_2=i,price(-C(m,N,R))) & ,B-\sum i\times n_i \ge i \\ D1(m+1,N,r_1.r_i+1..r_e, & \\ k_1=1,k_2=i,price(-C(m,N,R))) & ,otherwise\end{cases} & ,C(m,N,R)<1 \\ -P(1) & ,otherwise \\ & ,C(m,N,R)=1\end{cases}$$

$$-\lambda(-C(0,0,0))dt\Big(\sum_{i=1}^{e} G(i)\times D1(m=1,n_1=0,.n_i=1..,n_e=0,R=0,$$

$$k_1=0,k_2=i,price(-C(0,0,0)))\Big)$$

$$+\left(\begin{cases}\lambda(-C(m,N,R)) & ,C(m,N,R)\leq 0\\ \lambda(T-1) & ,C(m,N,R)=1\end{cases}\right)dt\times$$

$$\begin{cases}\sum_{i=1}^{e}G(i)\times & \\ \begin{cases}D2_t(m+1,n_1,.n_i+1..,n_e,R, & \\ t+dt) & ,B-\sum i\times n_i\geq i\\ D2_t(m+1,N,r_1.r_i+1..r_e, & \\ t+dt) & ,otherwise\end{cases} & ,C(m,N,R)<1\\ 0 & ,otherwise\end{cases}$$

$$C(m,N,R)=1$$

$$-\lambda(-C(0,0,0))dt(\sum_{i=1}^{e}G(i)\times D2_t(m=1,n_1=0,.n_i=1..,n_e=0,R=0,t+dt))$$

2. Leaving idle state

$$+[\sum_{state2}\{(\text{Probability of a client leaving idle state that changes state from state1 to state2 in time dt})\times D2_t(state2,t+dt)]$$

As mentioned earlier, the probability that a client leaves idle state is $S(0)\times(m-\sum_{i=1}^{e}(n_i+r_i))dt$. When a client leaves idle state, the client will disconnect with probability d or try to open a session with probability $(1-d)$. The following expression is obtained.

$$S(0) \times (m - \sum_{i=1}^{e}(n_i + r_i))dt \times [$$
$$d \times D2_t(m-1, N, R, t+dt)$$
$$+(1-d) \times \sum_{i=1}^{e}\{G(i) \times$$
$$\begin{cases} D2_t(m, n_1, n_2, .n_i + 1.., n_e, R, t+dt) & , B - \sum x \times n_x \geq i \\ D2_t(m, N, r_1.r_i + 1..r_e, t+dt) & , otherwise \end{cases} \}]$$

3. Closing of a session

$$+[\sum_{state2}\{(\text{Probability of a client leaving session state that}$$
$$\text{changes state from state1 to state2 in time dt})$$
$$\times D2_t(state2, t+dt)]$$

The probability of a session close for a client in session with i unit bandwidth is $n_i \times S(i)dt$. It is multiplied by the $D2_t()$ value when the session is closed. However, when a session is closed, some bandwidth is released and therefore among the clients waiting for session, few may get bandwidth. We assume a temporary function $D2'_t()$ which represents the value of $D2_t()$ when some checking needs to be done for available bandwidth and if available, bandwidth may be given to the waiting clients.

$$\sum_{i=1}^{e, n_i \neq 0} D2'_t(m, n_1, n_2, .n_i - 1.., n_e, R, t+dt) \times n_i \times S(i)dt$$

where $D2'_t()$ in terms of $D2_t()$ is given below. As mentioned earlier, $D2'_t()$ is a temporary function. The first possibility is checking for a client waiting for 1 unit bandwidth and whether this much bandwidth is available. This possibility is represented by $, B - \sum_{j=1}^{e} j \times n_j \geq 1, r_1 \neq 0$. Similarly the possibility of checking for a client waiting for i unit bandwidth and whether this much bandwidth is available is represented by $, B - \sum j \times n_j \geq i, r_i \neq 0$. If for any value of i, the condition $B-\sum j\times n_j \geq i, r_i \neq 0$ is true, it means that bandwidth may be given to a client requesting for i unit bandwidth. After giving the bandwidth, the value of r_i decreases and the value of n_i increases. Once this change is done, still the function $D2'_t()$ does not become $D2_t()$. It is because this checking is done recursively until it is not possible to give bandwidth to any waiting client. In that case, $D2'_t(m, N, R, t)$ becomes $D2_t(m, N, R, t)$.

$$D2'_t(m, N, R, t) =$$

$$\begin{cases} D2'_t(m, n_1+1, n_2, ..., n_e, r_1-1...r_e, t) & , B-\sum i\times n_i \geq 1 \\ & , r_1 \neq 0 \\ ... & . \\ D2'_t(m, n_1, n_2, ..n_i+1., n_e, r_1.r_i-1..r_e, t) & , otherwise \\ & , B-\sum j\times n_j \geq i \\ & , r_i \neq 0 \\ ... & . \\ D2'_t(m, n_1, n_2, ..., n_e+1, r_1...r_e-1, t) & , otherwise \\ & , B-\sum j\times n_j \geq e \\ & , r_e \neq 0 \\ D2_t(m, N, R, t) & , otherwise \end{cases}$$

4. No change in state

$$[\{1-\sum_{state2}(\text{probability of state change from state1 to state2 in time dt})\} \times D2_t(state1, t+dt)]$$

It becomes

$$(1-\left(\begin{cases}\lambda(-C(m,N,R)) & ,C(m,N,R)\leq 0 \\ \lambda(T-1) & ,C(m,N,R)=1\end{cases}\right)dt - S(0)\times(m-\sum_{i=1}^{e}(n_i+r_i))dt - \sum_{i=1}^{e} n_i\times S(i)dt)\times D2_t(m,N,R,t+dt)$$

The equations are rearranged to bring make the left hand side become the expression: $\frac{D2_t(m,N,R,t)-D2_t(m,N,R,t+dt)}{dt}$. This is done by bringing the $D2_t(m,N,R,t+dt)$ term from the forth part 'No change in state' and then everything is divided by dt.

$\frac{D2_t(m,N,R,t)-D2_t(m,N,R,t+dt)}{dt}$ is equal to the sum of the following terms.

1. Arrival

$$
\left(\begin{cases}\lambda(-C(m,N,R)) & ,C(m,N,R)\leq 0\\ \lambda(T-1) & ,C(m,N,R)=1\end{cases}\right)\times
$$

$$
\begin{cases}\sum_{i=1}^{e} G(i)\times \begin{cases} D1(m+1,n_1,.n_i+1..,n_e,R, \\ k_1=0,k_2=i,price(-C(m,N,R))) & ,B-\sum i\times n_i\geq i \\ D1(m+1,N,r_1.r_i+1..r_e, \\ k_1=1,k_2=i,price(-C(m,N,R))) & ,otherwise\end{cases} & ,C(m,N,R)<1 \\ -P(1) & ,otherwise \\ & ,C(m,N,R)=1\end{cases}
$$

$$
-\lambda(-C(0,0,0))(\sum_{i=1}^{e} G(i)\times D1(m=1,n_1=0,.n_i=1..,n_e=0,R=0, k_1=0,k_2=i,price(-C(0,0,0))))
$$

$$
+\left(\begin{cases}\lambda(-C(m,N,R)) & ,C(m,N,R)\leq 0\\ \lambda(T-1) & ,C(m,N,R)=1\end{cases}\right)\times
$$

$$
\begin{cases}\sum_{i=1}^{e} G(i)\times \begin{cases} D2_t(m+1,n_1,.n_i+1..,n_e,R,t+dt) & ,B-\sum i\times n_i\geq i \\ D2_t(m+1,N,r_1.r_i+1..r_e,t+dt) & ,otherwise\end{cases} & ,C(m,N,R)<1 \\ 0 & ,otherwise \\ & ,C(m,N,R)=1\end{cases}
$$

$$
-\lambda(-C(0,0,0))(\sum_{i=1}^{e} G(i)\times D2_t(m=1,n_1=0,.n_i=1..,n_e=0,R=0,t+dt))
$$

2. Leaving idle state

$$S(0) \times (m - \sum_{i=1}^{e}(n_i + r_i)) \times [$$
$$d \times D2_t(m-1, N, R, t+dt)$$
$$+(1-d) \times \sum_{i=1}^{e}\{G(i) \times$$
$$\begin{cases} D2_t(m, n_1, n_2, .n_i + 1.., n_e, R, t+dt) & , B - \sum x \times n_x \geq i \\ D2_t(m, N, r_1.r_i + 1..r_e, t+dt) & , otherwise \end{cases} \}]$$

3. Closing of session

$$\sum_{i=1}^{e, n_i \neq 0} D2'_t(m, n_1, n_2, .n_i - 1.., n_e, R, t+dt) \times n_i \times S(i)$$

where $D2'_t()$ is written as

$$D2'_t(m, N, R, t) =$$
$$\begin{cases} D2'_t(m, n_1+1, n_2, ..., n_e, r_1 - 1...r_e, t) & , B - \sum i \times n_i \geq 1 \\ & , r_1 \neq 0 \\ ... & . \\ D2'_t(m, n_1, n_2, ..n_i + 1., n_e, r_1.r_i - 1..r_e, t) & , otherwise \\ & , B - \sum j \times n_j \geq i \\ & , r_i \neq 0 \\ ... & . \\ D2'_t(m, n_1, n_2, ..., n_e + 1, r_1...r_e - 1, t) & , otherwise \\ & , B - \sum j \times n_j \geq e \\ & , r_e \neq 0 \\ D2_t(m, N, R, t) & , otherwise \end{cases}$$

4. No change in state

$$-(\left(\begin{cases}\lambda(-C(m,N,R)) & ,C(m,N,R)\le 0\\ \lambda(T-1) & ,C(m,N,R)=1\end{cases}\right)dt+S(0)\times(m-\sum_{i=1}^{e}(n_i+r_i))dt$$
$$+\sum_{i=1}^{e}n_i\times S(i)dt)\times D2_t(m,N,R,t+dt)$$

At steady state the left hand side becomes zero and time is removed from the equations. $D2_t()$ becomes $D2()$ (and $D2'_t()$ becomes $D2'()$). Therefore the sum of the following terms is zero for every valid value of (m, N, R).

1. Arrival

$$\left(\begin{cases}\lambda(-C(m,N,R)) & ,C(m,N,R)\le 0\\ \lambda(T-1) & ,C(m,N,R)=1\end{cases}\right)\times$$
$$\begin{cases}\sum_{i=1}^{e}G(i)\times\begin{cases}D1(m+1,n_1,.n_i+1..,n_e,R,\\ k_1=0,k_2=i,price(-C(m,N,R))) & ,B-\sum i\times n_i\ge i\\ D1(m+1,N,r_1.r_i+1..r_e,\\ k_1=1,k_2=i,price(-C(m,N,R))) & ,otherwise\end{cases} & ,C(m,N,R)<1\\ -P(1) & ,otherwise\\ & ,C(m,N,R)=1\end{cases}$$

$$-\lambda(-C(0,0,0))(\sum_{i=1}^{e} G(i)\times D1(m=1,n_1=0,.n_i=1..,n_e=0,R=0,$$
$$k_1=0,k_2=i,price(-C(0,0,0))))$$

$$+\left(\begin{cases}\lambda(-C(m,N,R)) & ,C(m,N,R)\le 0\\ \lambda(T-1) & ,C(m,N,R)=1\end{cases}\right)\times$$
$$\begin{cases}\sum_{i=1}^{e} G(i)\times \begin{cases}D2(m+1,n_1,.n_i+1..,n_e,R) & ,B-\sum i\times n_i\ge i\\ D2(m+1,N,r_1.r_i+1..r_e) & ,otherwise\end{cases} & ,C(m,N,R)<1\\ 0 & ,otherwise\end{cases}$$
$$,C(m,N,R)=1$$
$$-\lambda(-C(0,0,0))(\sum_{i=1}^{e} G(i)\times D2(m=1,n_1=0,.n_i=1..,n_e=0,R=0))$$

2. Leaving idle state

$$S(0)\times(m-\sum_{i=1}^{e}(n_i+r_i))\times[$$
$$d\times D2(m-1,N,R)$$
$$+(1-d)\times\sum_{i=1}^{e}\{G(i)\times\begin{cases}D2(m,n_1,n_2,.n_i+1..,n_e,R) & ,B-\sum x\times n_x\ge i\\ D2(m,N,r_1.r_i+1..r_e) & ,otherwise\end{cases}\}]$$

3. Closing of session

$$\sum_{i=1}^{e,n_i\neq 0} D2'(m,n_1,n_2,.n_i-1..,n_e,R)\times n_i\times S(i)$$

where $D2'()$ is written as

$$D2'(m,N,R) =$$

$$\begin{cases} D2'(m, n_1+1, n_2, ..., n_e, r_1-1...r_e) & , B - \sum i \times n_i \geq 1 \\ & , r_1 \neq 0 \\ ... & . \\ D2'(m, n_1, n_2, ..n_i+1., n_e, r_1.r_i-1..r_e) & , otherwise \\ & , B - \sum j \times n_j \geq i \\ & , r_i \neq 0 \\ ... & . \\ D2'(m, n_1, n_2, ..., n_e+1, r_1...r_e-1) & , otherwise \\ & , B - \sum j \times n_j \geq e \\ & , r_e \neq 0 \\ D2(m,N,R) & , otherwise \end{cases}$$

4. No change in state

$$-(\left(\begin{cases} \lambda(-C(m,N,R)) & , C(m,N,R) \leq 0 \\ \lambda(T-1) & , C(m,N,R) = 1 \end{cases}\right) + S(0) \times (m - \sum_{i=1}^{e}(n_i + r_i)) + \sum_{i=1}^{e} n_i \times S(i)) \times D2(m,N,R)$$

4.3.5 Solution and Complexity analysis

In this subsection, we present the complexity analysis of our solution.

4.3.5.1 Solution and Complexity analysis

We need to estimate the time and space required to get a correct decision matrix C. Further, we need to estimate the amount of space that will be required to store C. The C array stores an integer for each value of state. The space required to store C is the number of possible values of state (m, N, R). The number of service classes is e. As m becomes large, the number of possible values of n_i for every i will depend on the bandwidth B and not on m. So the number of possible values of N becomes $O(B^e)$. But the possible values of r_i for every i will depend on m. Therefore, the number of possible values of R becomes $O({m_{max}}^e)$. The number of possible values of (m, N, R) comes to be $O({m_{max}}^{e+1} \times B^e)$. Therefore, the space required to store C is $O({m_{max}}^{e+1} \times B^e)$.

The total time taken is the number of possible values of C being checked multiplied by the time taken to verify the correctness of C values. The total time taken is the time taken to find $D()$ from $C()$ multiplied by the number of $C()$ values to be checked for correctness. As already shown, to find $D()$ from $C()$, some equations have to be solved and the number of equations are the number of possible values of (m, N, R, K) divided by m_{max}. The other steps take less time and space and can be ignored. There are many equation in $D1()$ and $D2()$ and these equations have to be solved. The number of equations of $D1()$ is the number of unique possibilities of (m, N, R, K, p). The number of equations of $D2()$ is the number of unique possibilities of (m, N, R). The method to find $D1()$ and $D2()$ is the same. $D1()$ consists of more unknowns and therefore the time and space taken to find $D1()$ is more than the time and space taken to find $D2()$. Therefore, we ignore the time and space taken to find $D2()$.

We consider each price separately so the number of possibilities now are (m, N, R, K). To solve these equations, we choose some unknowns, write every other unknown in terms of those chosen unknowns, and then solve for those chosen unknowns. The chosen unknowns are those values of $D1()$ and $D2()$ for which m is m_{max}. This reduces the number of possible equations by m_{max}. The time taken to write every unknowns in terms of the chosen unknowns is the product of the number of chosen unknowns and the number of unknowns. It is $O({m_{max}}^e \times B^e)$ multiplied by $O({m_{max}}^{e+1} \times B^e)$. It becomes $O({m_{max}}^{2e+1} \times B^{2e})$. This value is usually small as compared to the time taken to solve the equation we get below $(O(m_{max} \times B)^{2.3727 \times e})$ and can be ignored. The space taken is the square of the chosen unknowns and it does not affect the overall complexity so it can also be ignored. The main time and space taken to find $D1()$ and $D2()$ is the time to solve the equations containing chosen unknowns.

We use LU Decomposition method to solve the equations. In the method of solving LU decomposition[7], the step which takes maximum time is triangular factorization[13]. In [13], authors have shown that if an algorithm to multiply two n by n matrix has some time complexity then triangular factorization can also be done with the same time complexity. In [60], the authors have shown that multiplication of n by n matrix can be done in $O(n^{2.3727})$ time. Therefore the time complexity reduces to $\{(\text{number of possible values of } (m, N, R, K))/m_{max}\}^{2.3727} \times (\text{number of prices})$. The space complexity is of order $\{(\text{number of possible values of } (m, N, R, K))/m_{max}\}^2$ because this is the size of matrix in LU decomposition method.

We find the complexity in terms of m_{max}, e and B by finding the number of possible values of (m, N, R, K). m ranges from 1 to m_{max} in $D1()$ function. As shown above, space taken to store all the possible values of (m, N, R) is O($m_{max}{}^{e+1} \times B^e$). K represents the state of the extra client and it is of the order of the number of possible states of the extra client. A client can be idle, in session or waiting for session. Therefore, the number of possible states is $1 + 2 \times e$ and therefore K is of order e.

The number of possible values of $D1()$ for a given price p is ($m_{max}{}^{e+1} \times B^e$) $\times e/m_{max}$. It is of order O(($m_{max}{}^{e} \times B^e$) $\times$ e). This makes the time complexity of finding $D()$ from $C()$ and then finding $C_{new}()$ from $D()$ equal to O(($m_{max}{}^{e} \times B^e$) $\times e)^{2.3727}$ $\times$(number of prices)), which reduces to O($(m_{max} \times B)^{2.3747 \times e} \times T$) and space complexity equal to O((($m_{max}{}^{e} \times B^e$) $\times e)^2$), which reduces to O($(m_{max} \times B)^{2 \times e}$).

Assume that x is the number of different values of $C()$ which have to be checked for correctness. As shown above, the time taken to find $C_{new}()$ from $C()$ is of O((($m_{max}{}^{e} \times B^e$) $\times e)^{2.3727}$ $\times T$). The overall time taken is of order O((($m_{max}{}^{e} \times B^e$) $\times e)^{2.3727}$ $\times T$ $\times x$). The value of x is not known. In the best case, x will be very small and can be ignored. So the complexity reduces to O($(m_{max} \times B)^{2.4 \times e} \times T$). In the worst case, x can be very large. The number of possible values of (m, N, R) are exponentially large. In each state, '$T + 1$' decisions can be taken and therefore there can be $(T + 1)^{(m_{max})^{e+1} \times B^e}$ decisions. However, it is possible to reduce the number of possible decisions that are valid. As the connected clients increase or number of clients in session increase, a client is being charged a higher price instead of a low price or a client is rejected instead of being charged the higher price. It will not happen that a client was being rejected at lower value of m and is being accepted at higher value of m when N and R remain the same. Therefore, the objective is to find those states at which the best decision of a service provider changes. This reduced the number of possible states to be checked but still it is quite large.

So we obtain a time and space complexity of calculating the C matrix to be O($(m_{max} \times B)^{2.4 \times e} \times T$) and O($(m_{max} \times B)^{2 \times e}$) respectively. The size of the C matrix is O($(m_{max} \times B)^{e}$). Unless e is 1, it is not possible to calculate the C values except for small values of m_{max} and B. So with a maximum of 1000 users, 200 Mbps of bandwidth, two values for e and two prices, it will take approximately 1767 million years to calculate C with a one nanosecond per operation time. The size of C will be 400 MB.

4.3.5.2 Reduced complexities when $e = 1$

The time and the space complexities of the accurate solution is not too high when e is 1. When e is 1, the state of a service provider can be represented by just two integers (m, n) where m is the number of connected clients and n is sum of the number of clients in session and the number of clients waiting for session. The solution matrix takes two inputs m and n where m ranges from 0 to m_{max} and for each value of m, n ranges from 0 to m. Therefore, the space required to store $C()$ matrix is $O({m_{max}}^2)$.

When e becomes 1, just one integer is used to store the number of clients in session and the number of clients waiting for session. This removes the bandwidth term from the above complexities when e can take any value. Therefore, the time complexity $O((m_{max} \times B)^{2.4 \times e} \times T)$ becomes $O((m_{max})^{2.4} \times T)$; and space complexity $O((m_{max} \times B)^{2 \times e})$ becomes $O(m_{max})^2$.

4.3.6 Proof that the AAEC method always produces an optimal solution

We assume that a service provider considers a solution $C()$ and then using the AAEC method obtains $C_{new}()$. The first theorem is to show that when $C()$ and $C_{new}()$ differ in some values, $C_{new}()$ produces more income than $C()$ at steady state. The second theorem proves that starting from any $C()$, using AAEC steps, it is guaranteed in finite steps the solution converges. The third theorem proves that when $C()$ and $C_{new}()$ do not differ in any value, $C()$ is an optimal solution.

Theorem 1. *If $C()$ and $C_{new}()$ are different, at steady state, $C_{new}()$ produces more income than $C()$.*

Proof. Let Income($X()$) be income produced per unit time at steady state when solution X is used. Assume that $C()$ and $C_{new}()$ are different. We need to prove that Income($C_{new}()$) is greater than Income($C()$). Suppose at run time, solution $C()$ is being used. Consider a state $s1$ for which $C(s1)$ and $C_{new}(s1)$ are different. When a service provider reaches state $s1$, taking decision $C_{new}(s1)$ instead of $C(s1)$ increases income according to our equations that are used to calculate $C_{new}()$. Therefore, the decision at state $s1$ may be replaced by $C_{new}(s1)$. Now let the service provider reach any other state $s2$ for which $C(s2)$ and $C_{new}(s2)$ are different. All the state changes in our model have exponential distribution and this distribution has memoryless property. According to this property, the future state depends on the present state and is independent of the past. Therefore, the best decision to be taken in a state also does not depend on the decisions that were taken in the past. Therefore, the decision taken in state $s1$ does not affect the decision to be taken in state $s2$. It means that now the decision state state $s2$ may be modified to $C_{new}(s2)$ and this increases income. Therefore, in every state si for which $C(si)$ and $C_{new}(si)$ are different, taking decision $C_{new}(si)$ instead of $C(si)$ increases income because of the memoryless property of the system. Therefore, $C()$ may be replaced by $C_{new}()$ at runtime and expected future income increases. Therefore, Income($C_{new}()$) is greater than Income($C()$). □

Theorem 2. *The AAEC method is guaranteed to converge to a state where $C()$ is equal to $C_{new}()$.*

Proof. According to theorem 1, at each iteration of the AAEC method, the steady state income increases. m_{max} represents the maximum number of connected clients. When a service provider reaches this capacity, an arriving client is rejected. This makes the state space finite. Because the state space is finite, the number of possible solutions, $C()$ are also finite. At each AAEC iteration, among all possible solutions, an improved solution is found. Due to the memoryless property of the system, this means that a solution once reached cannot be reached again. Therefore, the AAEC method is guaranteed to give a solution which will be a maximum and therefore $C()$ will be equal to $C_{new}()$ □

Theorem 3. *$C()$ and $C_{new}()$ are the same, if and only if $C()$ is a globally optimal solution.*

Proof. While there may be more than one globally optimal solution, the method adopted keeps $C()$ if $C()$ is found to be equal to $C_{new}()$. We can therefore prove the reverse direction by definition: since $C()$ is an optimal solution, any solution derived from $C()$ must be the same as $C()$, and so $C()$ must be equal to $C_{new}()$. For the forward direction, suppose there is a solution $C()$ such that $C()$ is equal to $C_{new}()$ (that is they have the same value for every state). Suppose there exists a globally optimal solution $C_{opt}()$ which is different from $C()$. Suppose at run time, solution $C()$ is being used. Consider a state $s1$ for which $C(s1)$ and $C_{opt}(s1)$ are different. When a service provider reaches state $s1$, taking decision $C_{opt}(s1)$ instead of $C(s1)$ will not increase income because, if that were so, then it means that $C(s1)$ would have been replaced by $C_{opt}(s1)$ during the AAEC method. But $C(s1)$ is equal to $C_{new}(s1)$. Therefore, if the decision at state $s1$ is replaced by $C_{opt}(s1)$ income will not increase. Now suppose the service provider reaches some other state $s2$ for which $C(s2)$ and $C_{opt}(s2)$ are different. As already mentioned, because of the memoryless property of our system, past decisions do not effect the present decisions. Therefore, a decision taken in state $s1$ does not affect any decision taken in state $s2$. It means that now if the decision state state $s2$ is modified to $C_{opt}(s2)$, this will not increase income. Therefore in every state si for which $C(si)$ and $C_{opt}(si)$ are different, taking decision $C_{opt}(si)$ instead of $C(si)$ will not increase income. This means Income($C_{opt}()$) is less than or equal to Income($C()$). Because $C_{opt}()$ is an optimal solution, income produced by $C_{opt}()$ cannot be less than income produced by any other solution. Therefore, both $C()$ and $C_{opt}()$ produce the same income at steady state. Therefore $C()$ is a globally optimal solution. □

Chapter 5

Approximate Solutions

We present two approximate solutions. The first is the session approximate solution. The second is the grouped approximate solution. It is possible to combine the two approximate solutions and in that case it is called the session grouped approximate solution.

5.1 Definition of Symbols

Let $\lambda(i)$ be the mean arrival rate of clients that connect to the service provider when the price being charged is $price(i)$. As the price increases, the arrival rate goes down. The actual correlation between the prices and the mean arrival rate is left as inputs to the model. In practice, it is assumed that the mean arrival rates and prices will be set periodically based on historical data. The arrival of clients is modeled as a Poisson process. The session duration for a client consuming i units of bandwidth is modeled by an exponential distribution with a mean service time of $\frac{1}{S(i)}$. The idle duration is also modeled by an exponential duration with mean $\frac{1}{S(0)}$. After the idle duration ends, a client disconnects with probability d or again requests for bandwidth with probability $(1-d)$.

As already mentioned, the state of a service provider is represented as (m, N, R) where m represents the number of clients connected. All the symbols and their definitions are described and shown in Table 5.1 and Table 5.2. The definition of those symbols whose values are taken as input are given in Table 5.1 and the definition of symbols whose values are calculated are given in Table 5.2. The goal is therefore to find the values of the decision matrix that gives an optimal solution. These values can be found off-line, beforehand.

At steady state, a service provider earns some income per unit time. The decisions

Table 5.1: Definition of Symbols

Symbol	Description
$\lambda(i)$	The mean arrival rate of clients at the service provider when $price(i)$ is being charged; i varies from 0 to $T-1$ (unit: number of clients/time)
$G(i)$	Probability of a client requesting i units of bandwidth when he initially has no bandwidth. Here i is an integer ranging from 1 to e.
$\frac{1}{S(i)}$	Mean session duration or idle duration for which a client consumes i units of bandwidth. If i is zero, it is the mean time a client remains idle. If i is non zero, it is the mean time for which a client remains in session consuming i units of bandwidth. (time)
$price(i)$	$price(i)$ is an array which stores the possible prices per unit data that can be charged from an arriving client. The values stored in $price(i)$ are in ascending order. $price(0)$ is the least price and $price(T-1)$ is the maximum. (rupee/data-unit)
T	Number of prices
m_{max}	It is the maximum possible value of the number of clients connected.
B	The total units of bandwidth available with the service provider (data-units / time)
e	Maximum number of units of bandwidth a client can request from his service provider. A client can request 1 to e units of bandwidth (data-units / time; also used as an index to arrays).
d	Probability that a client disconnects immediately after leaving the idle state.
$P(0)$	Penalty per unit time which is paid to a client when he waits for Session entry. (rupee/time)
$P(1)$	Penalty which a client gets when his request to connect is rejected. (rupee)
m	The number of clients connected. It ranges from 1 to m_{max}.
N	An array which represents the number of clients in session and its index ranges from 1 to e. So n_i represents the number of clients in session with i units of bandwidth.
R	An array which represents the number of clients waiting for a session and its index also ranges from 1 to e. r_i represents the number of users who requested for i units of bandwidth but are queued where i ranges from 1 to e.

Table 5.2: Definition of Symbols whose values are calculated

Symbol	Description
$C(m, N, R)$	The decision a service provider takes on the arrival of a client when the state of a service provider is (m, N, R). When the value of $C(m, N, R)$ is zero or less, the client is to be accepted and the price to be charged is $price(-C(m, N, R))$. When $C(m, N, R)$ is 1, the client is to be rejected. $C(m, N, R)$ is a function in the analysis, but its values are calculated offline and stored in an array C and at run time, it is the array that is consulted. (an integer)
$C_a(m)$	It stores the decisions to be taken at state m in the simplified model. The values of $C(m, N, R)$ are obtained as given below (an integer) $$C(m, N, R) = C_a(m), \forall N, R$$.
$Income$	Expected income per unit time. (rupee/time)
$Pri(m)$	Steady state probability of m clients being connected when arriving clients have infinite population
$Pr(m, n_1, ..n_e, r_1, ..r_e)$	The steady state probability that n_i clients in session consuming i units of bandwidth and r_i clients queued for session for i units of bandwidth where i is an integer ranging from 1 to e and the total population is finite and is m.
$Waiting(m)$	Expected number of clients waiting for bandwidth when m clients are connected
$I(m)$	Expected number of idle clients when m clients are connected
E_d	It is the expected data transferred by a newly arriving client (data-unit)

that produces the maximum steady state income per unit time are optimal decisions. We do not assume that an optimal decision is unique. There may be multiple optimal decisions and each of those decisions may produce the same income at steady state. The objective is to get any optimal decision. To make the analysis tractable, we consider only the number of clients connected in taking any decision. For a given value of the number of connected clients, the number that will be idle and the number that will be waiting for a session are estimated. So we find the optimal set of values for the array $C_a(m)$ and then map them to the array $C(m, N, R)$.

5.2 Session Approximate Solution

In the session approximate solution, the state of a service provider is represented by an integer m, which is the number of connected clients. This reduces the state space considerably. The decision a service provider takes at state m is given by $C_a(m)$. Out of the m clients the expected number in session and the expected number in idle state are estimated as shown below in the analysis. The problem is to find that value of $C_a(m)$, for every m, which maximises the average income of the service provider. Once the solution is obtained, $C_a(m)$ is then used for all $C(m, N, R)$ as given below.

$$C(m, N, R) = C_a(m), \forall N, R$$

5.2.1 Finding solution $C_a()$

The method is to iterate through all possible values of $C_a()$ and in each case find the expected income. The solution is that $C_a()$ which provides the maximum expected income. It is assumed that a service provider charges a low price when no client is connected and increases its price when more clients are connected. Therefore, the number of possible permutations of $C_a()$ which have to be considered is not too high if the number of possible prices are limited. The details of the number of possible permutations of $C_a()$ to be considered is given in section 5.2.6.

5.2.2 Finding Income

The expected income per unit time is found by multiplying the probability of a service provider being in state m by the income per unit time earned at state m for every possible state m and summing the results. The expected income earned at state m consists of two terms as shown in the equation below: the first term is the loss because of clients waiting for a session. There are waiting(m) clients expected to be waiting and the penalty rate is P(0). The first part of the second term is the expected income from arriving clients. This is the rate at which clients arrive when in state m times the expected number of data units each client requests times the price per data unit when in this state. The second part of the second term is to handle the case when the client is rejected, and this is a loss and is the arrival rate at the highest price times the penalty to be paid.

$$\begin{aligned} Income &= \sum_{m=0}^{m_{max}} Pri(m) \\ &\times \left[-Waiting(m) \times P(0) + \left(\begin{cases} \lambda(-C_a(m)) \times \\ E_d \times price(-C_a(m)) & , C_a(m) \leq 0 \\ -\lambda(T-1) \times P(1) & , otherwise \end{cases} \right) \right] \end{aligned} \tag{5.1}$$

As can be seen, the expected income depends on the decision taken which is the value of $C_a(m)$. So to find the best expected income rate, we have to find the proper values of $C_a(m)$.

5.2.3 Finding $Waiting()$

As the state space has been reduced, the number of clients waiting for bandwidth (for a session) is approximated by calculating the expected number of clients waiting when m clients are connected and this is denoted by $Waiting(m)$. The method to find $Waiting(m)$ is as follows. The expected number of clients waiting when the state is $(m, n_1, n_2, ..., n_e, r_1,..., r_e)$ is multiplied by the probability that the state is $(m, n_1, n_2,..., n_e, r_1,..., r_e)$ when m clients are connected and this is added for every possible value of n_i and r_i, for all i. When m clients are connected, the probability that the state is $(m, n_1, n_2,..., n_e, r_1,..., r_e)$ is given by $Pr(m, n_1, n_2, ..., n_e, r_1, ..., r_e)$. The number of clients waiting when state is $(m, n_1, n_2, ..n_e, r_1, ..., r_e)$ is by definition $\sum_{i=1}^{e} r_i$. So

$$Waiting(m) = \sum_{(n_1,...,n_e,r_1,...,r_e)} \left(Pr(m, n_1, n_2, ..n_e, r_1, ..., r_e) \times \sum_{i=1}^{e} r_i \right) \tag{5.2}$$

When a client is connected, he may be in idle state, in session or waiting for a session. When m clients are connected, the probability that n_i clients are in session and r_i clients are waiting for a session for i ranging from 1 to e is given by $Pr(m, n_1, n_2, ..n_e, r_1, ..r_e)$. If the value of e is 1 (all clients request exactly one unit of bandwidth), Pr() can be obtained using the finite source queueing theory formula[57].

$$Pr(m, n_1, r_1) = \frac{m!}{(m - n_1 - r_1)! \times n_1! \times B^{r_1}} \left\{ \left(\frac{S(0)}{S(1)} \right)^{n_1+r_1} \right\} \times Pr(m, 0, 0) \quad (5.3)$$

and

$$Pr(m, 0, 0) = \frac{1}{\sum_{n_1, r_1} \frac{m!}{(m-n_1-r_1)! \times n_1! \times B^{r_1}} \left\{ \left(\frac{S(0)}{S(1)} \right)^{n_1+r_1} \right\}}$$

For values of $e > 1$, we expand the terms of the above equation in a "natural" manner (we are unable to prove that this is an accurate formula), and, for m clients, we use the following expression for steady state probability.

$$Pr(m, n_1, n_2, ..n_e, r_1, ..., r_e) = \frac{m!}{(m - \sum_{i=1}^{e}\{n_i + r_i\})! \times (\sum_{i=1}^{e} n_i)! \times \prod_{i=1}^{e} \left(\frac{B}{i}\right)^{r_i}} \times \prod_{i=1}^{e} \left\{ \left(\frac{S(0) \times G(i)}{S(i)} \right)^{n_i+r_i} \right\} \times Pr(m, 0, 0, ..., 0) \quad (5.4)$$

The sum of probabilities is 1. Therefore, another equation is

$$\sum_{(n_1,...,n_e,r_1,...,r_e)} Pr(m, n_1, n_2, ..n_e, r_1, ..., r_e) = 1$$

On substituting the value of $Pr(m, n_1, n_2, ..n_e, r_1, ..., r_e)$ from equation 5.4, we get

$$\sum_{(n_1,...,n_e,r_1,...,r_e)} [\frac{m!}{(m - \sum_{i=1}^{e}\{n_i + r_i\})! \times (\sum_{i=1}^{e} n_i)! \times \prod_{i=1}^{e} \left(\frac{B}{i}\right)^{r_i}} \times \prod_{i=1}^{e} \left\{ \left(\frac{S(0) \times G(i)}{S(i)} \right)^{n_i+r_i} \right\} \times Pr(m, 0, 0, ..., 0)] = 1$$

From this the value of $Pr(m, 0, 0...0)$ is

$$Pr(m,0,0,...,0) = \Big(\sum_{(n_1,...,n_e,r_1,...,r_e)} \Big\{ \frac{m!}{(m - \sum_{i=1}^{e}\{n_i + r_i\})! \times (\sum_{i=1}^{e} n_i)! \times \prod_{i=1}^{e} \left(\frac{B}{i}\right)^{r_i}} \prod_{i=1}^{e} \left(\frac{S(0) \times G(i)}{S(i)} \right)^{n_i + r_i} \Big\} \Big)^{-1} \tag{5.5}$$

5.2.4 Finding $Pri()$

$I(m)$, the expected number of idle clients when there are m clients in the system can be found the same way as $Waiting(m)$. So, the expression is

$$I(m) = \sum_{(n_1,...,n_e,r_1,...,r_e)} \left\{ Pr(m, n_1, n_2, ..n_e, r_1, ..., r_e) \times (m - \sum_{i=1}^{e}\{n_i + r_i\})! \right\} \tag{5.6}$$

Now, the departure rate from the system is $I(m) \times d \times S(0)$ by definition of the model. If we consider this departure process to be memoryless and if we ignore the rejection of clients at connection time (making the arrival process memoryless), then the system can be modelled as a Markov Chain and the global balancing equation can be applied:

$$Pri(m) \times departure\ rate = Pri(m-1) \times arrival\ rate \tag{5.7}$$

equation where $Pri(m)$ is the steady state probability of m connected clients. The mean arrival rate when there are $m-1$ clients is λ_{m-1}, and this depends on the price being charged in state $m-1$. By our definition given in Table 5.1 this is $\lambda(-C_a(m-1))$. So we get,

$$Pri(m) \times I(m) \times d \times S(0) = Pri(m-1) \times \lambda(-C_a(m-1)) \tag{5.8}$$

This can be re-written as

$$Pri(m) = \frac{1}{I(m)} \left(\frac{\lambda(-C_a(m-1))}{d \times S(0)} \right) \times Pri(m-1) \tag{5.9}$$

Further, the following will hold and the two equations can be used to solve for $Pri(m)$.

$$1 = \sum_{i=0}^{m_{max}} Pri(i) \tag{5.10}$$

5.2.5 Finding E_d

The method to find the approximate data consumption by an arriving client, E_d is given below. When a client connects, he consumes i units of bandwidth for a period with mean $\frac{1}{S(i)}$ with probability $G(i)$ and then becomes idle for a period with mean $\frac{1}{S(0)}$. At the end of this idle period he disconnects with probability d. Otherwise, he again requests for bandwidth with probability $(1-d)$ and therefore again transfers E_d units of data. So we get,

$$E_d = \sum_{i=1}^{e} G(i) \times \frac{1}{S(i)} \times i + (1-d) \times E_d$$

This can be re-written as

$$E_d = \sum_{i=1}^{e} \left(\frac{G(i)}{d \times S(i)} \times i \right) \tag{5.11}$$

.

5.2.6 Complexity Analysis

The values of $Pr()$ and $Waiting(m)$ have to be calculated only once because these do not depend on the values of $C_a()$. The number of times $Pr()$ has to be computed depends on the number of unique possible values of $Pr(m, n_1, ..n_e, r_1, ..r_e)$. m ranges from 0 to m_{max}. When m_{max} is large, n_1 to n_e depend on B and not on m_{max} so they are of order (B). r_1 to r_e are of order (m_{max}). By multiplying all of them, the number of possible values of $Pr()$ for m_{max} connected clients is O$({m_{max}}^e \times B^e)$. Since m can range from 0 to m_{max}, this complexity has to be multiplied by m_{max} to get the number of possible values of $Pr()$. For a given value of m, the value of $Pr()$ is computed by using equations 5.4 and 5.5. Their complexity is the same as the number of values computed. The time complexity is therefore given by O$({m_{max}}^{e+1} \times B^e)$. For example, if e is 2, m_{max} is 100 and B is 20, the time taken is O$(100^{2+1} \times 20^2)$ (O(4×10^8)).

As can be seen from equation 5.4, the values of $Pr(m, n_1, ..., n_e, r_1, ..r_e)$ are in terms of $Pr(m, 0, 0, ..., 0)$. For finding $Pr(m, 0, 0, ..., 0)$, equation 5.5 is solved which has constant space requirement. Once $Pr(m, 0, 0, .., 0)$ is computed, every other value can be computed by using equation 5.4. Therefore, the space requirement to compute all values of $Pr()$ is a constant.

Once $Pr()$ is found, $Waiting()$ is computed and its value has to be stored. Therefore, the space complexity depends on the value of $Waiting()$. The space complexity is O(m_{max}).

The time taken to find income and $Pri(m)$ from equations 5.9 and 5.10 is O(m_{max}).

The next step is to find the number of possible values of $C_a()$. As already mentioned, a service provider has to consider all possible values of $C_a()$ and to choose that $C_a()$ which maximizes the expected income. It is asserted without proof (although it is intuitively obvious if the model is examined) that, in any optimal solution, the price charged when there are k users will always be greater than or equal to the price charged when there are j users when k>j. So all permutations of the array $C_a()$ do not have to be considered. We have a case of permutations of length m_{max} (the length of $C_a()$) with elements ranging from 1 to T with repetitions allowed, but with all numbers in a permutation in non-decreasing order left to right. These permutations can be generated by T nested loops with indices $i1$, $i2$ ...iT and each index ranging from $i(j) = i(j-1)$ to m_{max} , with i1 starting from 0. The number of instances of $C_a()$ that will be generated will be of the order of ${m_{max}}^T$. For each instance the time to find the income has already been shown to be O(m_{max}). So the time complexity of the method is O(${m_{max}}^{T+1}$). This complexity of O(${m_{max}}^{T+1}$) is under the assumption that the value of $Pr()$, $Waiting()$, E_d are already known. The overall time complexity is the maximum of the two and it is O(${m_{max}}^{e+1} \times B^e + {m_{max}}^{T+1}$). If the solution is recomputed after modifying only the mean arrival rate $\lambda()$, the values of $Pr()$, $Waiting()$, E_d need not be recomputed and therefore the time taken will be O(${m_{max}}^{T+1}$). When the solution is implemented, a service provider observes the mean arrival rate of clients and based on this it calculates the output. If the mean arrival rate of client changes, it may recalculate the solution.

5.3 Grouped approximate solution

When a service provider becomes very large, a single client will be too small to be considered for taking a decision. Therefore, a service provider, instead of considering each client, may consider him in groups. A service provider does not consider exactly how many clients are present but considers the approximate value. The approximation depends on group size. For example, suppose the group size is 100. The input to the solution is not exactly how many clients are present but is some range. For example, the input may be that 500 to 600 clients are connected but not the exact number. When arrival takes place, with $\frac{1}{100}$ probability the number of clients, change to '600 to 700' from '500 to 600'.

For finding the solution, the values of m_{max}, B, $\lambda()$ and connected clients are divided by the group size and the new values are the input to the problem. The result is applied to the actual scenario. Other things remain the same.

This grouping reduces the space and time complexity of the solution. For finding the space and time complexity, the value of m_{max} and B is divided by group size and everything else is the same. A service provider can trade off accuracy against time complexity by varying the group size.

It is possible to have a combination of session approximate and grouped approximate solution. In that case it is termed as session grouped approximate solution.

5.4 A Simple Heuristic

We need a basis of comparing our solutions to other possible solutions. We cannot compare our results with flat pricing schemes as the QoS requirements do not exist there. In this section, we present a simple heuristic which fixes the price based on an estimate of how loaded the system is and which pays penalties like our methods. In this method, a service provider estimates the bandwidth requirement by connected clients. If it is greater than or equal to the actual bandwidth, a newly arriving client is rejected. Otherwise, an arriving client is charged an appropriate price and is allowed to connect. The appropriate price is given below. An arriving client is charged price $price(i)$ if the following two conditions are satisfied.

$$i \leq \left(\frac{\text{$<$estimated bandwidth requirement$>$}}{\text{$<$total bandwidth$>$}} \times T \right)$$

and

$$i + 1 > \left(\frac{\text{$<$estimated bandwidth requirement$>$}}{\text{$<$total bandwidth$>$}} \times T \right)$$

where T is the number of prices. The value of total bandwidth and number of prices is known. The only thing to be found is the bandwidth required by connected clients. The method to find the estimated bandwidth requirement by connected clients is as follows. Let there be m connected clients. With burst times much less than inter-arrival times of new clients, this can be considered a finite population system in steady state. A connected client sometimes remains idle and sometimes consumes bandwidth. Let x be the estimated number of idle clients. These clients open sessions needing i units of bandwidth at the rate $S(0) \times x \times (1-d) \times G(i)$. The estimated number of clients consuming i units of bandwidth is n_i. The clients who are in session with i units of bandwidth complete their sessions at the rate of $S(i) \times n_i$. Both the rates are equal at steady state and so $S(0) \times x \times (1-d) \times G(i) = S(i) \times n_i$

Therefore,

$$\frac{x}{n_i} = \frac{S(i)}{G(i) \times (1-d) \times S(0)}$$

Rearranging we get,

$$\frac{n_i}{x} = \frac{G(i) \times (1-d) \times S(0)}{S(i)} \tag{5.12}$$

Therefore,

$$n_i = x \times \frac{G(i) \times (1-d) \times S(0)}{S(i)} \tag{5.13}$$

The total number of clients connected is m. As a simplification for estimation purposes, we assume that no client waits for bandwidth. Therefore, m is the sum of the estimated number of idle clients and the sum of the estimated number of clients consuming bandwidth:

$$m = x + \sum_{i=1}^{e} n_i$$

With this the following terms are obtained.

$$m = x + x \times \sum_{i=1}^{e} \frac{G(i) \times (1-d) \times S(0)}{S(i)}$$

The ratio of the number of connected clients and the estimated number of idle clients is obtained by the following steps.

$$\frac{m}{x} = 1 + \sum_{i=1}^{e} \frac{G(i) \times (1-d) \times S(0)}{S(i)}$$

$$\frac{x}{m} = \frac{1}{1 + \sum_{i=1}^{e} \frac{G(i) \times (1-d) \times S(0)}{S(i)}} \tag{5.14}$$

Therefore,

$$\frac{n_i}{m} = \frac{n_i}{x} \times \frac{x}{m}$$

After substituting the value of $\frac{n_i}{x}$ from equation 5.12 and the value of $\frac{x}{m}$ from equation 5.14, we get

$$\frac{n_i}{m} = \frac{G(i) \times (1-d) \times S(0)}{S(i)} \times \frac{1}{1 + \sum_{i=1}^{e} \frac{G(i) \times (1-d) \times S(0)}{S(i)}}$$

$$n_i = m \times \frac{G(i) \times (1-d) \times S(0)}{S(i)} \times \frac{1}{1 + \sum_{i=1}^{e} \frac{G(i) \times (1-d) \times S(0)}{S(i)}}$$

Estimated bandwidth consumed is

$$\sum_{i=1}^{e} i \times n_i$$

After substituting the value of n_i, the estimated bandwidth consumed is found to be

$$\sum_{i=1}^{e} i \times m \times \frac{G(i) \times (1-d) \times S(0)}{S(i)} \times \frac{1}{1 + \sum_{i=1}^{e} \frac{G(i) \times (1-d) \times S(0)}{S(i)}}$$

Chapter 6

Simulations

We have presented an accurate solution, approximate solutions and a simple heuristic solution. The accurate solution has high time and space complexities and is not suitable for large sized service providers. Even for medium sized service providers it works only when we apply a restriction that a client can request for only one unit of bandwidth (when e=1). Therefore, we have presented solutions with lower time and space complexities. These solutions however are approximate and need to be evaluated. In this chapter, we evaluate the performance of our approximate solutions and our simple heuristic solution. We simulate the clients and the observe the income of service providers and quality of service provided to clients.

For each simulation, the performance of the solutions is in terms of income of a service provider when a given solutions is used, total delay for which connected clients have to wait for bandwidth and the total number of connection requests rejected. In each comparison, service providers are identical (have same bandwidth, m_{max}, etc. and may differ only based on the solution they use) and this is done for fair comparison of our solutions. These simulations are categorized into various categories based on their purpose. These are given below.

1. **Comparison of our accurate solution, our session approximate solution and the simple heuristic solutions.** As already mentioned, the main objective of our simulations is to evaluate our approximate solutions and our simple heuristic solution. In these simulations, we directly compare our approximate solutions and simple heuristic solution with our accurate solution. However, these simulations have limitations. Because of high time and space complexities of our accurate solution, the solution generally works for small sized service providers. Only in the specific case when the maximum bandwidth that may be requested by a client is restricted to one unit, it works for

medium sized service providers. These simulations are given in Section 6.1. The simulations show that the session approximate solution works well and the average decline in income when the session approximate solution is used instead of the accurate solution is approximately within 10 percent.

2. **Simulating solutions that work for medium and large service providers.** We have done another set of simulations in which we compare our solutions that work for medium and large sized service providers. As already mentioned, our accurate solution generally works for only small sized service providers and therefore we have compared our non accurate solutions with each other in these simulations. We compare our solutions by varying the important simulation parameters and observe the performance difference. These simulations are given in Section 6.2. The simulations show that our approximate solutions perform well for medium and large service providers. Our solutions provide better income and quality of service as compared to the simple heuristic solution.

3. **Simulation to find which approximate solution is better.** After doing the above simulations, we found that one of our approximate solutions performs well for some configuration and another performs well for some other configuration. Therefore, we have done another set of simulations to find the appropriate solution to be used for different configurations. These simulations are given in Section 6.3.

4. **Checking the performance of our simple heuristic solution.** In most of the above simulations, our accurate and approximate solutions perform better than the simple heuristic solution. This raises a question on the performance of our simple heuristic solution. Therefore, we have also compared our simple heuristic with a simpler heuristic and a random solution to find out whether our chosen heuristic is good enough. These simulations are given in Section 6.4. Our simple heuristic generates a better income as compared to the simpler heuristic and the random solution.

5. **Simulation for realistic schemes.** In the above simulations, arrival of client is based on the price charged by service providers. The mean arrival rate of clients is $\lambda(i)$ when a service provider charges price $price(i)$. However, in the simulations for realistic schemes, the mean arrival rate of clients is fixed and a service provider observes the rate at which clients connect to it. These simulations are given in Section 6.5. Our solutions perform well in the realistic schemes in most of the cases.

In simulations 6.1, 6.2, 6.3 and 6.4, for each simulation specification, the system is simulated two or three times and each time with a different solution. This comparison helps in finding the effect of changing a solution on the output. These simulations run for one hour. In 6.5, only one simulation is done for each simulation specification. Each simulation consists of multiple identical competing service providers and these service provider may differ only based on the solution they are using. The simulations run for three hours. For the first two hours, service providers observe the arrival rate of clients and in the performance of a solution in the third hour is the simulation output. The simulation result does not contain the performance of a solution for the first two hours.

6.1 Comparison with our accurate solution

We have presented five solutions: the accurate solution, the session approximate solution, the grouped approximate solution, the session grouped approximate solution and the simple heuristic solution. The grouped approximate solution is an approximation of the accurate solution and the session approximate solution is an approximation of the session approximate solution. Therefore, we have not included the grouped approximate solution and the session grouped approximate solution in this section. These solutions are included in the next section. In this section, our session approximate solution and the simple heuristic solutions are compared with the accurate solution.

There are three groups of simulations. In the first group (Section 6.1.1), the size of service providers is small. There are multiple sets of simulations and in each set, the mean arrival rate of clients (λ) is different. In the second and third group (Section 6.1.2 and Section 6.1.3), the size of service provider is medium. In the second group, the multiple set of simulations differ in the bandwidth of service providers and in the third group, simulation sets differ in the penalty function.

6.1.1 Small ISP Simulation and variation of mean arrival rates

Simulation specifications are given in Table 6.1. There are three prices and therefore there are three mean arrival rates that are based on the price charged. The value of $\lambda(2)$ takes two values 0.075 and 0.15. For each value of $\lambda(2)$, $\lambda(1)$ takes two values, $\lambda(2) + 0.075$ and $\lambda(2) + 0.15$. For each combination of $\lambda(2)$ and $\lambda(1)$, $\lambda(0)$ takes two values, $\lambda(1) + 0.075$ and $\lambda(1) + 0.15$. The total number of combinations are $2\times2\times2=8$. For each combination, the system is simulated thrice.

Table 6.1: Simulation Details: Accurate-approximate-heuristic comparison 1

Variable	Value
d	0.4
B	5
m_{max}	14
e	2
T	3
$price(0)$	0.1 per unit data
$price(1)$	0.12 per unit data
$price(2)$	0.15 per unit data
$P(0)$	0.4 per second
$P(1)$	5
Mean idle time ($\frac{1}{S(0)}$)	20
Mean session duration ($\frac{1}{S(1)}$, $\frac{1}{S(2)}$)	$\frac{\text{Mean idle time}}{5}$
$G(1)$	0.3
$G(2)$	0.7

Simulation results are given in Table 6.2. The value of mean arrival rates for each of the simulations is given in the table. For finding the average performance of the three solutions, the incomes, delays and rejections are added for each of the simulations. Simulation results show that income produced by the accurate solution and the session approximate solution is almost three times as compared to the simple heuristic solution. The difference between incomes of the accurate solution and the session approximate solution is approximately within 10% of the income in the accurate solution. The quality of service is improved in both the accurate solution and the session approximate solutions. The accurate solution improves delay by approximately 67 % and number of rejections by approximately 15 %. The session approximate solution improves delay by approximately 57 % and number of rejections by approximately 72 %. This shows that the session approximate solution and the accurate solution each significantly improves income and quality of service.

6.1.2 Variation of bandwidth

When the value of e is 1, the symbol $G(i)$ is removed. Simulation specifications are given in Table 6.3. We considered multiple values of bandwidth and compared the results. The value of bandwidth takes five values 20 to 40 in multiple of 5 (20, 25,

Table 6.2: Simulation Result: Accurate-approximate-heuristic comparison 1

λ(0)	λ(1)	λ(2)	Accurate Solution			Session Approximate Solution			Simple Heuristic		
			Income	Delay	Reject	Income	Delay	Reject	Income	Delay	Reject
0.225	0.15	0.075	724	776	0	634	746	0	393	1669	2
0.3	0.15	0.075	699	938	3	603	1027	0	382	2126	0
0.3	0.225	0.075	800	674	2	759	951	3	337	2356	5
0.375	0.225	0.075	747	1031	3	725	944	3	406	2167	4
0.3	0.225	0.15	842	741	5	848	1181	3	99	2939	9
0.375	0.225	0.15	849	820	5	747	1400	0	187	2812	7
0.375	0.3	0.15	915	862	9	884	1346	2	9	3319	16
0.45	0.3	0.15	823	976	12	875	1258	2	163	3029	3
Total			6399	6818	39	6075	8853	13	1976	20417	46
Percentage of deviation from values in the simple heuristic solution			+224%	-67%	-15%	+207%	-57%	-72%	-	-	-

30, 35, 40). Therefore, there are five sets of simulations.

Simulation results are given in Table 6.4. For each of the five sets of simulations, the value of bandwidth is given in the table. Simulation result shows that income produced by the accurate solution and the session approximate solution is approximately 50% more as compared to the simple heuristic solution. The difference between incomes of the accurate solution and the session approximate solution is approximately within 10 % of the income in the accurate solution. The accurate solution improves delay by approximately 89 % but the number of rejections increase by approximately 44 %. The session approximate solution improves delay by approximately 63 % and number of rejections by approximately 7 %. This shows that the session approximate solution and the accurate solution each improves income and quality of service on average.

6.1.3 Variation of Penalties

In order to assess how the methods perform when the penalties $P(0)$ (delay penalty per second) and $P(1)$ (penalty per rejection) are varied, a set of simulations were carried with different values of $P(0)$ and $P(1)$. The value of $P(0)$ ranged from 0 to 0.6 in multiples of 0.2. The value of $P(1)$ ranged from 0 to 3. The simulation was run

Table 6.3: Simulation Details: Accurate-approximate-heuristic comparison 2

Variable	Value
d	0.3
B	20 to 40
m_{max}	130
e	1
$\lambda(0)$	2.5
$\lambda(1)$	2
$\lambda(2)$	1.5
T	3
$price(0)$	0.1 per unit data
$price(1)$	0.2 per unit data
$price(2)$	0.3 per unit data
$P(0)$	0.3 per second
$P(1)$	3
Mean idle time ($\frac{1}{S(0)}$)	15
Mean session duration ($\frac{1}{S(1)}$)	5

Table 6.4: Simulation Result: Accurate-approximate-heuristic comparison 2

Bandwidth	Accurate Solution			Session Approximate Solution			Simple Heuristic		
	Income	Delay	Reject	Income	Delay	Reject	Income	Delay	Reject
20	13,959	4,631	1,427	11,041	21,896	1,036	-5,296	79,747	893
25	20,841	5,901	561	17,065	25,650	262	10,716	48,498	87
30	26,302	4,016	18	25,403	3,089	0	22,794	7,289	30
35	26,779	316	0	26,531	338	0	22,390	3,234	62
40	27,258	8	0	26,569	12	0	20,729	371	320
Total	115,140	14,871	2,006	106,609	50,983	1,298	71,332	139,140	1,392
Percentage of deviation from values in the simple heuristic solution	+61%	-89%	+44%	+49%	-63%	-7%	-	-	-

Table 6.5: Simulation Details: Accurate-approximate-heuristic comparison 3

Variable	Value
d	0.3
B	25
m_{max}	130
e	1
T	3
$price(0)$	0.1 per unit data
$price(1)$	0.2 per unit data
$price(2)$	0.3 per unit data
$\lambda(0)$	2.5
$\lambda(1)$	2
$\lambda(2)$	1.5
$P(0)$	0 to 0.6 per second
$P(1)$	0 to 3
Mean idle time ($\frac{1}{S(0)}$)	15
$\frac{1}{S(1)}$	5

with m_{max} as 130 and e=1 to allow the accurate solution to also be compared. All the parameters of the simulation are given in Table 6.5. There are sixteen simulations and in each simulation penalties are different.

The results are given in Table 6.6. First of all, as is to be expected, the simple heuristic is not sensitive to change in the penalty values as penalties are not used in the decision making process, only the load is. Secondly, it is seen that except for the case when there are no penalties - when all three methods perform similarly - both the accurate and session approximate methods perform better than the simple heuristic in all three parameters. So we can infer that our earlier results are not sensitive to the penalty values. Both the solutions are more sensitive to the delay penalty than the rejection penalty. For the same rejection penalty, as the delay penalty value increases, the total delay penalty decreases. On the other hand, for the same delay penalty, increasing the rejection penalty does not have that much of an affect on the total number of rejections. So, the delay penalty is playing a large role in controlling both the total delay as well as the total number of rejections. Intuitively this makes sense because below a certain load level delays will be few and so increasing rejections to further reduce the load will not accrue any benefit. So reducing the penalty for rejection may not result in an increase in rejections.

Table 6.6: Simulation Result: Accurate-approximate-heuristic comparison 3

Penalties		Accurate solution			Session Approximate Solution			Simple Heuristic		
P(0)	P(1)	Income	Delay	Reject	Income	Delay	Reject	Income	Delay	Reject
0	0	26254	59096	181	25981	50843	151	25622	48857	88
0.2	0	23193	5713	578	20833	23092	236	16654	44251	93
0.4	0	22429	2536	770	20266	11239	382	6039	48778	139
0.6	0	22054	1480	804	19742	6987	548	1418	40249	69
0	1	26072	59096	181	25829	50843	151	25533	48857	88
0.2	1	22588	6386	540	19814	29585	172	16473	43919	82
0.4	1	21748	2718	659	18925	14042	366	7635	45004	81
0.6	1	21317	1826	826	18740	8264	489	-1795	45138	89
0	2	25890	59096	181	25677	50843	151	25444	48857	88
0.2	2	22092	7914	530	19096	31948	165	16555	42999	65
0.4	2	20974	3806	700	17597	16822	289	9408	40064	84
0.6	2	20430	2238	803	17423	10244	445	34	41912	125
0	3	25708	59096	181	25525	50843	151	25355	48857	88
0.2	3	21551	8988	467	17309	39631	138	16876	42589	56
0.4	3	20406	4461	668	17580	16823	332	6961	45606	78
0.6	3	19805	2601	745	16064	12638	377	-246	42364	95

6.2 Solutions for medium and large service providers

The grouped approximate solution, the session approximate solution and the simple heuristic solution can work for medium sized service providers. For large sized service providers, the grouped approximate can work by further increasing the group size; and the session grouped approximate solution replaces the session approximate solution. In this section, we compare these three solution with the simple heuristic solution.

There are multiple groups of simulations. In the first group (Section 6.2.1), the size of service providers is medium. There are multiple sets of simulations and in each set, the mean arrival rate of clients (λ) is different. In the second group (Section 6.2.2), the size of service provider is large and again there is variation in mean arrival rates. In the third group (Section 6.2.3), the size of the service providers is medium and multiple set of simulations differ in bandwidth of service provider.

Table 6.7: Simulation Details: Approximate-heuristic comparison 1

Variable	Value
d	0.4
B	60
m_{max}	200
e	2
T	3
$price(0)$	0.1 per unit data
$price(1)$	0.12 per unit data
$price(2)$	0.15 per unit data
$P(0)$	0.4 per second
$P(1)$	5
Mean idle time ($\frac{1}{S(0)}$)	20
Mean session duration ($\frac{1}{S(1)}$, $\frac{1}{S(2)}$)	$\frac{\text{Mean idle time}}{5}$
$G(1)$	0.3
$G(2)$	0.7

6.2.1 Variation of mean arrival rates in Medium ISP

In these simulations, we compare the three solutions: the grouped approximate solution, the session approximate solution, and the simple heuristic solution. In the grouped approximate solution, the group size is 20. There are few differences in simulation specifications as compared to the simulation given in Section 6.1.1. The value of m_{max} is 200 and the value of B is 60. There are three prices and therefore there are three mean arrival rates that are based on the price charged. The value of $\lambda(2)$ takes two values 2 and 4. For each value of $\lambda(2)$, $\lambda(1)$ takes two values $\lambda(2)+2$ and $\lambda(2)+4$. For each combination of $\lambda(2)$ and $\lambda(1)$, $\lambda(0)$ takes two values $\lambda(1)+2$ and $\lambda(1)+4$. The total number of combinations are $2\times2\times2=8$. Therefore, there are 8 simulations. The parameters of simulation are given in Table 6.7.

Simulation results are given in Table 6.8. There are eight simulations and the value of mean arrival rates for each of the simulation is given. Simulation results show that session approximate solution improves income by 35 % as compared to the simple heuristic solution. The grouped approximate solution produces almost no delay and the session approximate solution reduces the delay by approximately 36 %. The improvement in number of rejections in the grouped approximate solution is 31 % and in the session approximate solution is 72 %. The simulation shows that both the approximate solutions improve income and quality of service.

Table 6.8: Simulation Result: Approximate-heuristic comparison 1

λ(0)	λ(1)	λ(2)	Grouped Approximate Solution (group size = 20)			Session Approximate Solution			Simple Heuristic		
			Income	Delay	Reject	Income	Delay	Reject	Income	Delay	Reject
6	4	2	17268	0	19	20725	9847	13	18436	12708	64
8	4	2	17241	0	11	20847	9330	12	18472	12502	54
8	6	2	17441	0	9	21923	9773	12	17054	15780	164
10	6	2	17323	0	12	21981	9500	20	16562	16919	171
8	6	4	17013	0	199	24980	10121	70	16810	16557	148
10	6	4	17105	0	195	24874	10314	57	16514	17264	155
10	8	4	16995	0	191	25005	9900	66	16704	16086	208
12	8	4	17011	0	175	24988	10031	83	16726	15804	210
Total			137397	0	811	185323	78816	333	137278	123620	1174
Percentage of deviation from values in the simple heuristic solution			0%	-100%	-31%	+35%	-36%	-72%	-	-	-

6.2.2 Variation of mean arrival rates in large ISPs

In these simulations, we compare the three solutions: the grouped approximate solution, the session grouped approximate solution, and the simple heuristic solution. In the grouped approximate , the group size is 200. In the session grouped approximate, the group size is 10. There are a few differences in simulation specifications as compared to the above simulation. The value of m_{max} is 2000 and the value of B is 600. There are three prices and therefore there are three mean arrival rates that are based on the price charged. The value of $\lambda(2)$ takes two values 20 and 40. For each value of $\lambda(2)$, $\lambda(1)$ takes two values $\lambda(2) + 20$ and $\lambda(2) + 40$. For each combination of $\lambda(2)$ and $\lambda(1)$, $\lambda(0)$ takes two values $\lambda(1) + 20$ and $\lambda(1) + 40$. The total number of combinations are 2×2×2=8. Therefore, there are 8 simulations.

Simulation results are given in Table 6.9. The format of the result is the same as that of earlier simulations. Simulation results show that the grouped approximate solution improves income by 11 % and the session grouped approximate solution improves income by 35 % as compared to the simple heuristic solution. The grouped approximate solution produces almost no delay and the session approximate solution reduces the delay be approximately 59 %. The improvement in number of rejections

Table 6.9: Simulation Result: Approximate-heuristic comparison 2

$\lambda(0)$	$\lambda(1)$	$\lambda(2)$	Grouped Approximate Solution (group size = 200)			Session Grouped Approximate Solution (group size = 10)			Simple Heuristic		
			Income	Delay	Reject	Income	Delay	Reject	Income	Delay	Reject
60	40	20	362926	0	0	438612	2070	0	397056	5674	567
80	40	20	364771	0	0	438798	1872	0	397614	5435	557
80	60	20	363290	0	0	476592	3437	0	392201	10490	1545
100	60	20	365552	0	0	475994	3837	0	392784	11701	1535
80	60	40	508268	0	1082	600082	4648	581	392230	12192	1516
100	60	40	507770	0	1113	599820	5013	586	392828	10763	1518
100	80	40	508685	0	1003	598951	7308	515	387481	17733	2055
120	80	40	508758	0	1039	598186	9501	548	387669	17418	2029
Total			3490020	0	4237	4227035	37686	2230	3139863	91406	11322
Percentage of deviation from values in the simple heuristic solution			+11%	-100%	-63%	+35%	-59%	-80%	-	-	-

in the grouped approximate solution is 63 % and in the session grouped approximate solution is 80 %. The simulation shows that both the approximate solutions improve income and quality of service.

6.2.3 Variation of Bandwidth

In these simulations, we compare the three solutions: the grouped approximate solution, the session approximate solution, and the simple heuristic solution. We run multiple simulations and divide them in groups and the difference in each group is only the bandwidth. We take the average performance of each group. The simulation specifications are given in Table 6.10. The value of mean idle time takes two values 15 and 20. For each value of mean idle time $\frac{1}{S(1)}$ takes two values 3 and 5. For each combinations of mean idle time and $\frac{1}{S(1)}$, the value of $\frac{1}{S(2)}$ takes two values 3 and 5. There are 8 combinations of $\frac{1}{S()}$. For each of the combinations the value of bandwidth ranges from 50 to 70 units in multiple of 2.

Simulation results are given in Table 6.11. Simulation results show that each of our two approximate solutions perform well in different simulation specifications.

Table 6.10: Simulation Details: Approximate-heuristic comparison 3

Variable	Value
d	0.5
B	50 to 70
m_{max}	200
e	2
T	3
$price(0)$	0.1 per unit data
$price(1)$	0.2 per unit data
$price(2)$	0.3 per unit data
$\lambda(0)$	8
$\lambda(1)$	6
$\lambda(2)$	4
$P(0)$	0.2 per second
$P(1)$	2
Mean idle time ($\frac{1}{S(0)}$)	15, 20
Mean session duration ($\frac{1}{S(1)}$, $\frac{1}{S(2)}$)	3, 5
$G(1)$	0.5
$G(2)$	0.5

Therefore, later we also run simulations to find which approximate solutions should be used when. Simulations also show that when bandwidth is low, the difference in performance of our solutions in large and when bandwidth is high, the difference in performance in our simulations is low.

6.3 Which approximate solution is better?

The above simulations show that both the approximate solutions perform well for different simulation specifications. In the session approximate solution, a service provider takes decision based on the number of connected clients. A connected client can be in the following states: idle state, session state and waiting for session state. We generally assume that a client is expected to be in idle state for more time as compared to session state or waiting state. If mean idle time for a client is much larger than the mean non idle time (the sum of the session duration and the waiting time), this means that among the connected clients, large number of clients are idle. A client will open session, remain in session for a small amount of time and then

Table 6.11: Simulation Result: Approximate-heuristic comparison 3

Bandwidth	Grouped Approximate Solution (group size = 20)			Session Approximate Solution			Simple Heuristic		
	Income	Delay	Reject	Income	Delay	Reject	Income	Delay	Reject
50	328,756	86,802	10,489	267,018	515,113	7,390	110,449	696,576	27,807
52	337,608	50,877	9,744	295,121	424,548	6,059	148,304	559,839	25,909
54	341,939	27,358	9,878	324,954	311,076	4,811	174,041	470,973	24,551
56	345,565	13,748	9,540	349,831	221,024	4,028	192,665	389,131	24,207
58	347,832	6,587	9,849	368,175	153,860	3,251	200,555	310,472	25,394
60	383,394	40,375	2,660	379,542	107,907	2,996	216,128	247,025	24,905
62	388,842	31,465	2,400	392,187	67,983	2,622	219,393	202,872	25,453
64	391,678	19,106	2,204	398,815	38,773	2,421	200,326	163,305	29,424
66	393,907	13,781	2,050	402,696	27,788	2,407	213,390	119,741	28,584
68	395,383	9,168	1,931	404,660	15,490	2,411	222,838	90,181	27,982
70	396,350	4,359	1,858	406,076	7,504	2,371	229,073	67,558	27,724
Total	4,051,255	303,625	62,603	3,989,074	1,891,066	40,767	2,127,164	3,317,673	291,940
Percentage of deviation from values in the simple heuristic solution	+90%	-91%	-79%	+88%	-43%	-86%	-	-	-

become idle.

When a client is expected to be in more than one state for large amount of time, the state of the client becomes more unpredictable. When the ratio of the mean idle time to mean non idle time is more, the probability that a connected client is idle will be more. For example, if with 99 percent probability, the actual state of a client is idle, the decision taken just by looking at the number of connected clients is likely to perform well. When a client can be in more than one state for more time, the present state of the client cannot be so accurately predicted. For example, with 60% probability the client is idle and with 40% probability the client is in session. In this case, the decision taken just by looking at the number of connected clients may not work well. In the session approximate solution, decisions are just based on the number of connected clients. Therefore, we expect it not to perform well if the ratio of the mean idle time to mean non idle time is small and otherwise we expect it to perform well.

In Section 6.3.1, we run multiple sets of simulations and in each set we vary the mean idle time. In Section 6.3.2, we run multiple sets of simulations and in each set we vary the bandwidth of the service providers. In Section 6.3.3, we present the method to find an appropriate approximate solution that is likely to perform better for a given simulation specification.

6.3.1 Effect of varying mean idle time on the two solutions

Simulation specifications are given in Table 6.12. There are 9 simulations, and in each simulation the value of mean idle time is different. The mean idle time is in the range 4 to 20 in multiples of 2. We compare grouped approximate solution and session approximate solution. The group size in grouped approximate solution is 20.

Simulation result is given in Table 6.13. As shown, when the mean idle time is low, the grouped approximate solution performs better and when it is is high, the session approximate solution performs better. This is because when the mean idle time is low, the ratio of the mean idle time to mean non idle time is low and when the mean idle time is high, the ratio of the mean idle time to mean non idle time is high.

6.3.2 Effect of congestion on the two solutions

In the above simulations, when the ratio of the mean idle time to the mean non idle time is low, the grouped approximate solution performs better and when the ratio is high, the session approximate solution performs better. When there is congestion,

Table 6.12: Simulation Details: Search for an appropriate approximate solution 1

Variable	Value
d	0.3
B	50
m_{max}	250
e	2
T	3
$price(0)$	0.1 per unit data
$price(1)$	0.2 per unit data
$price(2)$	0.3 per unit data
$\lambda(0)$	10
$\lambda(1)$	8
$\lambda(2)$	6
$P(0)$	0.1 per second
$P(1)$	1
Mean idle time ($\frac{1}{S(0)}$)	4 to 20
Mean session duration ($\frac{1}{S(1)}$, $\frac{1}{S(2)}$)	2
$G(1)$	0.5
$G(2)$	0.5

Table 6.13: Simulation Result: Search for an appropriate approximate solution 1

Mean idle time	Grouped Approximate			Session Approximate		
	Income	Delay	Rejects	Income	Delay	Rejects
4	43126	63652	3957	28618	214032	3535
6	44189	48646	4086	21545	283871	3539
8	44613	40681	3989	21832	279748	3643
10	44466	44177	3982	33735	160646	3553
12	44614	24189	4595	43192	54988	4037
14	40564	5294	5771	42417	10718	5254
16	34809	1454	7280	37098	2844	7088
18	29196	255	8764	31576	527	8386
20	25176	49	9640	26723	108	9433

Table 6.14: Simulation Details: Search for an appropriate approximate solution 2

Variable	Value
d	0.3
B	40 to 60
m_{max}	250
e	2
T	3
$price(0)$	0.1 per unit data
$price(1)$	0.2 per unit data
$price(2)$	0.3 per unit data
$\lambda(0)$	10
$\lambda(1)$	8
$\lambda(2)$	6
$P(0)$	0.1 per second
$P(1)$	1
Mean idle time ($\frac{1}{S(0)}$)	12
Mean session duration ($\frac{1}{S(1)}$, $\frac{1}{S(2)}$)	2
$G(1)$	0.5
$G(2)$	0.5

a client will spend more time in waiting state. This mean that the time spent by a client in non idle state will increase. Therefore, we expect congestion to improve the performance of the grouped approximate solution as compared to the session approximate solution.

We have done simulations by varying the value of bandwidth. Simulation specifications are given in Table 6.14. There are 11 simulations and in each simulation bandwidth is different. The value of bandwidth ranges from 40 to 60 in multiples of 2. Simulation result are given in Table 6.15. The simulation shows that congestion decreases the performance of the session approximate solution. When the bandwidth is low, the grouped approximate solution performs better and when the bandwidth is high, the session approximate solution performs better.

6.3.3 Which solution should be used when?

The above simulations show that the ratio of mean idle time to mean non idle time seems to be the criteria for the better performance of one approximate solution over

Table 6.15: Simulation Result: Search for an appropriate approximate solution 2

Bandwidth	Grouped Approximate			Session Approximate		
	Income	Delay	Rejection	Income	Delay	Rejection
40	28410	71262	7182	13740	216415	7380
42	31902	63434	6564	20141	181593	6573
44	35230	55620	6033	26191	146808	6019
46	38676	48947	5288	31935	116907	5446
48	42090	35981	4760	38298	79578	4705
50	44614	24189	4595	43192	54988	4037
52	46696	14940	3958	46843	32965	3691
54	47464	8003	4111	49595	19227	3448
56	48217	4005	3961	50771	11790	3270
58	48136	2269	4185	51745	7569	3117
60	49942	2532	3630	52448	4221	3091

the other. In this subsection, we estimate this ratio. This is done by estimating the mean idle time, mean session time and mean waiting time.

Let λ represent the mean arrival rate to a service provider. It excludes rejection. As shown in Figure 6.1 (copy of Figure 4.1), an arriving client after arrival requests i units of bandwidth with probability $G(i)$. He remains in session for approximately $\frac{1}{S(i)}$ time and then becomes idle. He then departs with probability d and again requests for bandwidth with probability $(1-d)$.

Therefore, the mean session time is

$$\sum_{i=1}^{e} G(i) \times \frac{1}{S(i)} \times (1 + (1-d) + (1-d)^2 ...)$$
$$= \sum_{i=1}^{e} G(i) \times \frac{1}{S(i)} \times \frac{1}{1-(1-d)}$$
$$= \sum_{i=1}^{e} G(i) \times \frac{1}{d \times S(i)}$$

The mean time spend in idle state is

$$\frac{1}{S(0)}(1 + (1-d) + (1-d)^2 ...)$$
$$= \frac{1}{S(0)} \times \frac{1}{1-(1-d)}$$
$$= \frac{1}{d \times S(0)}$$

The mean time spent in waiting for one client is unknown. It can be found by observing it. It is the total delay divided by the number of clients arrived. The number of clients arrived is <mean arrival rate> multiplied by the time duration.

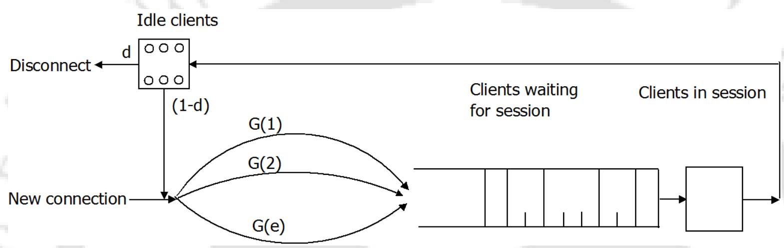

Figure 6.1: A service provider in our model

After calculating the ratio of mean idle time and mean non idle time, the following results are obtained. Table 6.13 is re written below as Table 6.16. Similarly, Table 6.15 is rewritten below as Table 6.17. The results show that when the ratio of mean idle time to mean non idle time is 6 or more, the performance of the session approximate solution is better and when it is less, the performance of the grouped approximate solution is better.

6.4 Evaluation of our simple heuristic solution

We have used a simple heuristic. In these simulations we check the performance our simple heuristic to check if it is a reasonably good heuristic. We compare our simple heuristic solutions with a simpler heuristic and a random solution. The simpler heuristic and the random solution are described below.

Simpler heuristic

In this method, a service provider finds the bandwidth consumed by connected clients. If it is equal to the actual bandwidth, a newly arriving client is rejected.

Table 6.16: Simulation Result: Search for an appropriate approximate solution 1 analyzed

Mean idle time	Grouped Approximate		Session Approximate	
	Income	Ratio of mean idle time to mean non idle time	Income	Ratio of mean idle time of mean non idle time
4	43126	2	28618	1
6	44189	2	21545	1
8	44613	3	21832	2
10	44466	4	33735	3
12	44614	5	43192	5
14	40564	7	42417	7
16	34809	8	37098	8
18	29196	9	31576	9
20	25176	10	26723	10

Table 6.17: Simulation Result: Search for an appropriate approximate solution 2 analyzed

Bandwidth	Grouped Approximate		Session Approximate	
	Income	Ratio of mean idle time to mean non idle time	Income	Ratio of mean idle time to mean non idle time
40	28410	4	13740	3
42	31902	5	20141	3
44	35230	5	26191	3
46	38676	5	31935	4
48	42090	5	38298	4
50	44614	5	43192	5
52	46696	6	46843	5
54	47464	6	49595	5
56	48217	6	50771	6
58	48136	6	51745	6
60	49942	6	52448	6

Otherwise, an arriving client is charged an appropriate price and is allowed to connect. The appropriate price is given below. An arriving client is charged price $price(i)$ if the following two conditions are satisfied.

$$i \leq \left(\frac{<\text{bandwidth consumed}>}{<\text{total bandwidth}>} \times T \right)$$

and

$$i + 1 > \left(\frac{<\text{bandwidth consumed}>}{<\text{total bandwidth}>} \times T \right)$$

where T is the number of prices. The value of total bandwidth and number of prices is known.

Random Solution

In this solution a service provider randomly takes any decision with equal probability. The number of possible decisions are $T+1$. A service provider takes a decision with $\frac{1}{T+1}$ probability.

Simulation

We have done forty simulations and taken the average of all the simulations. There are ten values of bandwidth in range 51 to 60. For each value of bandwidth, d takes four values from 0.2 to 0.8 in multiples of 0.2. Therefore, there are 40 simulations. Simulation details are given in table 6.18. Simulation results are given in Table 6.19. The result shows that our simple heuristic performs much better than the simpler heuristic. The random solution fails to produce any net income. This shows that our simple heuristic is good enough for comparison with our solutions.

6.5 Realistic schemes

In these simulations, we simulate our solution in scenarios that seems to be realistic. We compare the three solutions: the grouped approximate solution, the session approximate solution and the simple heuristic. Each simulation consists of six identical competing service providers and these service provider may differ only based on the solution they are using. Among the six service providers, two use the grouped approximate solution, two use the session approximate solution and two use the simple

Table 6.18: Simulation Details: Evaluation of the simple heuristic

Variable	Value
d	0.2 to 0.8
B	51 to 60
m_{max}	200
e	2
T	3
$price(0)$	0.1 per unit data
$price(1)$	0.2 per unit data
$price(2)$	0.3 per unit data
$\lambda(0)$	4
$\lambda(1)$	3
$\lambda(2)$	2
$P(0)$	0.3 per second
$P(1)$	3
Mean idle time ($\frac{1}{S(0)}$)	20
Mean session duration ($\frac{1}{S(1)}$, $\frac{1}{S(2)}$)	5
$G(1)$	0.5
$G(2)$	0.5

Table 6.19: Simulation Result: Evaluation of the simple heuristic

	Simple Heuristic	Simpler Heuristic	Random Solution
Total Income	1,052,687	99,035	-843,227
Total delay	771,896	1,374,773	0
Total rejections	19,582	64,145	281,038

heuristic. In the grouped approximate solution, the group size is 20. The simulations run for three hours. For the first two hours, service providers observe the arrival rate of clients and the simulation result contains the performance of a solution in the third hour. The simulation result does not contain the performance of a solution for the first two hours. In few of our simulations results, income of one or more service providers were negative. In actual scenarios, a service provider is likely to earn a positive income and to increase income, it will choose a solution. Therefore, we feel that the simulations with negative income are not realistic. In almost all of the simulations with negative income, the simple heuristic solution produced negative income. We are comparing our solutions with the simple heuristic and we should choose those simulation specifications in which the simple heuristic should produce a positive income. Therefore, we have removed simulations with negative income.

We consider three schemes in which our solutions can be applied. In each of these schemes, service providers observe arrival rate of clients when a given price is charged. The first scheme assumes a system of six service providers and multihomed internet clients. Each client can connect to two service providers. This scheme is given in Section 6.5.1. The second scheme assumes that each client has access to all the six service providers. When a clients connects, he observes the prices charged by all the service providers and chooses the service provider who charges the least price. This scheme is given in Section 6.5.2. The third schemes is a modification of the first scheme and in this scheme a client decides the bandwidth he will use at the beginning of the connection (instead of a decision taken everytime he moves to session based on $G(i)$). This scheme is given in Section 6.5.3.

6.5.1 Scheme 1: Each client has access to two service providers

As already mentioned, we assume that there are six service providers. Each client has access to two service providers. Based on prices advertised by the service providers, an arriving client chooses the service provider oferring the lower price. Simulation specifications are given in Table 6.20. The mean arrival rate of clients common to every pair of service providers is 1.4 clients per second. Thus each service provider sees arrivals at the rate of 7 (5×1.4) clients per second and for each arrival there is a single competing service provider. The value of m_{max} is 200. The value of bandwidth ranges from 40 to 70 in multiples of 2 (40, 42, 44,...70). For each of the bandwidth, there are four combinations of penalties: P(0) and P(1) where P(0) takes two values: 0.05 and 0.1 and P(1) takes two values 0.5 and 1. The total number of combinations are 16×4=64. Therefore, there are 64 simulations. As already mentioned. the simulations in which a service provider earns negative income are removed.

Table 6.20: Simulation Details: Realistic Scheme 1

Variable	Value
d	0.5
B	40 to 70
m_{max}	200
e	2
T	3
$price(0)$	0.1
$price(1)$	0.12
$price(2)$	0.15
$P(0)$	0.05 and 0.1 per second
$P(1)$	0.5 and 1
Mean idle time ($\frac{1}{S(0)}$)	22
Mean session duration ($\frac{1}{S(1)}$, $\frac{1}{S(2)}$)	6
$G(1)$	0.5
$G(2)$	0.5

Simulation results are given in Table 6.21. The incomes, delays and rejections are added for each of the simulations and these are displayed in the table. Simulation result shows that the grouped approximate solution produces maximum income among the three solutions and the improvement is 56% as compared to the simple heuristic solution.

6.5.2 Scheme 2: Each client has access to all service providers

In these simulations, we assume that each client has access to all the six service providers. The mean arrival rate of clients is 20 per second. The other specifications are the same as in the above simulation.

Simulation result is given in Table 6.22. The incomes, delays and rejections are added for each of the simulations and these are displayed in the table. Simulation result shows that both of our solutions improve income and quality of service as compared to the simple heuristics. The grouped approximate solution produces maximum income among the three solutions and the improvement is 45% as compared to the simple heuristic solution. Both the approximate solutions improves the quality of service.

Table 6.21: Simulation Result: Realistic Scheme 1

	Grouped Approximate Solution (group size = 20)	Session Approximate Solution	Simple Heuristic
Total Income	2,404,237	2,142,351	1,538,885
Percentage change in total income as compared to total income in the simple heuristic solution	+56%	+39%	-
Total delay	4,421,361	9,246,264	9,214,609
Percentage change in total delay as compared to total delay in the simple heuristic solution	-52%	+0%	-
Total rejections	478,772	472,824	778,484
Percentage change in total rejection as compared to total rejection in the simple heuristic solution	-38%	-39%	-

Table 6.22: Simulation Result: Realistic Scheme 2

	Grouped Approximate Solution (group size = 20)	Session Approximate Solution	Simple Heuristic
Total Income	1,608,396	1,570,629	1,111,706
Percentage change in total income as compared to total income in the simple heuristic solution	+45%	+41%	
Total delay	2,023,252	4,368,201	4,526,906
Percentage change in total delay as compared to total delay in the simple heuristic solution	-55%	-4%	
Total rejections	555,545	517,853	838,303
Percentage change in total rejection as compared to total rejection in the simple heuristic solution	-34%	-38%	

Table 6.23: Simulation Result: Realistic Scheme 3

	Grouped Approximate Solution (group size = 20)	Session Approximate Solution	Simple Heuristic
Total Income	2,242,191	1,962,629	1,662,017
Percentage change in total income as compared to total income in the simple heuristic solution	+35%	+18%	-
Total delay	4,102,986	10,140,375	9,493,557
Percentage change in total delay as compared to total delay in the simple heuristic solution	-57%	+7%	-
Total rejections	194,573	209,716	284,293
Percentage change in total rejection as compared to total rejection in the simple heuristic solution	-32%	-26%	-

6.5.3 Scheme 3: A client consumed fixed bandwidth for complete connection duration

In this scheme, a client can connect to two service providers (like in the simulation given in Section 6.5.1). The bandwidth a client can request and consume is decided at the beginning of his connection. A client after connection decides the bandwidth he wants to consume and he consumes that much bandwidth every time he goes into session. The mean arrival rate for clients common to each pair of service providers is 2 per seconds. Other things are the same as in the above simulation.

Simulation results are given in Table 6.23. The incomes, delays and rejections are added for each of the simulations and these are displayed in the table. Simulation result shows that both of our solutions improve income and quality of service as compared to the simple heuristics. The grouped approximate solution produces maximum income among the three solutions and the improvement is 35% as compared to the simple heuristic solution.

Chapter 7

Conclusion and Future Work

The internet has revolutionized the working and living of people. Clients want to have a good, reliable, and predictable internet service. The number of people with internet access and the internet access speeds are different in different parts of the world. Many regions of the world face shortage of bandwidth. We have presented a scheme and solutions that contains assurance from service providers to clients regarding good internet service and if clients do not get assured service, service providers pay penalties to clients. As in such schemes with penalties, a side effect is to balance load among service providers.

7.1 Conclusion

Initially we explored the problem using game theory. We developed a game theoretic solution based on obtaining Nash equilibrium. This solution assumes that there are just two ISPs and each client can request just one unit of bandwidth. The algorithmic complexities of the solutions is shown in Table 7.1. We have compared the two solutions by running them and observed the expected income when both the service providers used the accurate solution and when both the service providers used the approximate solutions. There were six comparison with multiple variations of bandwidth. The result shows that when both the service providers used the approximate Nash equilibrium instead of the accurate Nash equilibrium solution, the performance decline is at most 15% when the congestion is high to almost no difference when the congestion is low.

The Nash equilibrium solutions have many limitations. These have high time and space complexities. In addition to that, if the same method to find the Nash equilibrium is extended for more than two service providers, the complexities increase

Table 7.1: Complexities of our Nash Equilibrium solutions

	Size of solution	Time complexity	Space complexity
Accurate Nash Equilibrium	O($m_{max}{}^2$)	Best case: O($m_{max}{}^{7.12} \times m_{max}{}^{T \times m_{max}}$) Worst case: O($m_{max}{}^{7.12} \times m_{max}{}^{2 \times T \times m_{max}}$)	O($m_{max}{}^6$)
Approximate Nash Equilibrium	O(m_{max})	Best case: O($(m_{max})^{3+T}$) Worst case: O($(m_{max})^{3+2\times T}$)	O$(m_{max})^2$

exponentially. We do not have a proof that a Nash equilibrium will always exist. Therefore, we have also presented non-Nash equilibrium solutions. We presented a model in which an arriving client can request for more than one unit bandwidth and there is no restriction on the number of service providers.

We presented an accurate solution for the non-game theoretic model. The method to find the accurate solution is to start with a solution and then improve it; and this improvement continues till no more improvement is possible. The accurate solutions has significantly lower time and space complexities than the accurate Nash equilibrium solution. However, as the size of service providers increase, there was a need for solutions with lower complexities. Therefore, we presented approximate solutions: the session approximate solution, the grouped approximate solution and the session grouped approximate solution. The complexities of the accurate solution and the approximate solutions are given in Table 7.2. The accurate solution can run for very small sized service providers (when $m_{max} \simeq 10$ and e=2) and for medium sized service providers only when the maximum bandwidth that can be requested by a client is 1 (when $m_{max} \simeq 100$ and e=1). The session approximate solution can run for medium sized service providers (when $m_{max} \simeq 100$ and e=2) but cannot run for large sized service providers (when $m_{max} > 1000$ and $e \geq 2$). The grouped approximate solution and the session grouped approximate solution can run for large sized service providers.

The session approximate solution, the grouped approximate solution and the session grouped approximate solution are approximate and we have evaluated these solutions by simulating them. As already mentioned, we compared these solutions with a simple heuristic. The accurate solution and the session approximate solution were compared with the simple heuristic solution for small and medium ISPs. The decline in income when the session approximate solution is used instead of the accurate solution is approximately 10% and both performed significantly better than

Table 7.2: Complexities of our Non Game-theoretic solutions

	Size of solution	Time complexity	Space complexity
Accurate Solution	$O({m_{max}}^{e+1} \times B^e)$	Best case: When $e \neq 1$: $O((m_{max} \times B)^{2.4\times e} \times T)$ When $e=1$: $O((m_{max})^{2.4} \times T)$	$O((m_{max} \times B)^{2\times e})$
Session Approximate Solution	$O(m_{max})$	Finding solution for the first time $O({m_{max}}^{e+1} \times B^e + {m_{max}}^{T+1})$ Updating solution if just the mean arrival rates change $O({m_{max}}^{T+1})$	$O(m_{max})$
Grouped Approximate	Replace m_{max} with $\frac{m_{max}}{\text{group size}}$ and B with $\frac{B}{\text{group size}}$ in the accurate solution complexities		
Session Grouped Approximate Solution	Replace m_{max} with $\frac{m_{max}}{\text{group size}}$ and B with $\frac{B}{\text{group size}}$ in the session approximate solution complexities		

the simple heuristic. We also did simulations for medium and large sized service providers and found that there is an improvement in income and quality of service in our accurate and approximate solutions (the session approximate solution, the grouped approximate solution and the session grouped approximate solution) as compared to the simple heuristic in almost all the simulations. We also did simulations for real scenarios in which there are multihomed clients and service providers observe arrival rate of clients instead of being told arrival rates when a given price is charged. The simulations show that our accurate and approximate solutions perform better as compared to the simple heuristic solution in most of our simulations.

7.2 Future Work

The thesis opens many future research directions. These are given below.

7.2.1 Game Theory Solution

We have presented a Nash Equilibrium solution for two service providers. We were unable to prove the existence of a Nash equilibrium solution in all cases. One future direction is to present Nash equilibrium solutions for more than two service providers. Apart from Nash equilibrium solutions, there are many other game theoretic models and solutions that can be explored. For example, the game between service providers can be cooperative or non cooperative. There may be coalition among service providers.

7.2.2 Improvement of time and space complexities of our accurate solutions

Our accurate solutions (Both Nash equilibrium and non game theoretic solutions) have high time and space complexities and therefore, one direction is to try and reduce the time or the space complexities of our accurate solutions. We tried to get a closed form solution but we could not get it and so this is also a future research direction. The other option is to present a simpler model whose solution has lower time and space complexities.

7.2.3 More general model

Our model can be made more general in multiple directions. For example, we assume that all the clients are identical. In some cases, clients may not be identical. Different clients use the internet for different purpose. This additional knowledge can help a service provider in deciding what he should do when a new client arrived.

7.2.4 Inclusion of other QoS parameters

In our research problem, we considered only bandwidth guarantees. The research can be extended to include other QoS parameters such as packet delay, jitter, packet loss, etc.

7.3 Publications

Published papers

1. R. Tripathi and G. Barua. "Dynamic Internet Pricing and Bandwidth Guarantees with Nash Equilibrium". In *16th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2014.

2. R. Tripathi and G. Barua. "Pricing with Bandwidth Guarantees for Clients with multi-ISP Connections". In *16th International Conference on Distributed Computing and Networking (ICDCN)*, 2015.

Papers accepted for publication

1. R. Tripathi and G. Barua. "Dynamic internet Pricing with service level agreements for multihomed clients". *NETNOMICS: Economic Research and Electronic Networking*, Springer.

Bibliography

[1] Internet World Stats. http://www.internetworldstats.com/stats.htm.

[2] Netindex. http://netindex.com.

[3] Speedtest.net. http://speedtest.net.

[4] The World in 2014: ICT Facts and Figures - ITU. "http://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx".

[5] –. Rogers intros data sharing for 6GB plans with hefty premium. *Electronista*, February 8 2011.

[6] –. The mother of invention: Network operators in the poor world are cutting costs and increasing access in innovative ways. *Special Report. The Economist*, September 24 2009.

[7] S. Abbasbandy, R. Ezzati, and A. Jafarian. LU decomposition method for solving fuzzy system of linear equations. *Applied Mathematics and Computation*, 172(1):633 – 643, 2006.

[8] I. J. B. F. Adan and J. A. C. Resing. *Queueing theory. Lecture notes.* Eindhoven University of Technology, ftp://ftp.win.tue.nl/pub/stoch-or/queueing.pdf., 2001.

[9] B. Al-Manthari, N. Nasser, and H. Hassanein. Congestion Pricing in Wireless Cellular Networks. *IEEE Communications Surveys and Tutorials*, 13(3):358 –371, quarter 2011.

[10] M. Bagnulo, A. Garcia-Martinez, and A. Azcorra. Ipv6 multihoming support in the mobile internet. *IEEE Wireless Communications*, 14(5):92–98, 2007.

[11] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Service. *RFC Editor*, 1998.

[12] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: An Overview. *RFC Editor*, 1994.

[13] J. Bunch and J. Hopcroft. Triangular factorization and inversion by fast matrix multiplication. *Mathematics of Computation*, 28(125):231–236, 1974.

[14] D. Cavalcanti, D. Agrawal, C. Cordeiro, B. Xie, and A. Kumar. Issues in integrating cellular networks WLANs, AND MANETs: A futuristic heterogeneous wireless network. *IEEE Wireless Communications*, 12(3):30–41, June 2005.

[15] E. B. Çil, F. Karaesmen, and E. L. Örmeci. Dynamic Pricing and Scheduling in a Multi-class Single-server Queueing System. *Queueing Systems Theory Applications*, 67(4):305–331, April 2011.

[16] C. K. Chau, Q. Wang, and D. M. Chiu. On the Viability of Paris Metro Pricing for Communication and Service Networks. In *IEEE INFOCOM*, pages 1–9, March 2010.

[17] M. Chiang, P. Hande, H. Kim, S. Ha, and R. Calderbank. Pricing broadband: Survey and open problems. In *2nd International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 303–308, 2010.

[18] P. Cramton. Spectrum auctions. In *Handbook of Telecommunications Economics*. Elsevier Science, 2002.

[19] B. Donnet. *Data Traffic Monitoring and Analysis*. Springer Berlin Heidelberg, 2013.

[20] V. Firoiu, J. Y. Le Boudec, D. Towsley, and Z. L. Zhang. Theories and models for Internet quality of service. *Proceedings of the IEEE*, 90(9):1565–1591, Sep 2002.

[21] R. Garg, H. Saran, R. S. Randhawa, and M. Singh. A SLA framework for QoS provisioning and dynamic capacity allocation. In *10th International Workshop on Quality of Service (IWQoS)*, pages 129–137, 2002.

[22] A. Giloni, Y. L. Koçağa, and P. Troy. State dependent pricing policies: Differentiating customers through valuations and waiting costs. *Journal of Revenue and Pricing Management*, 12:139–161, 2013.

[23] C. A. Gizelis and D. D. Vergados. A survey of pricing schemes in wireless networks. *IEEE Communications Surveys and Tutorials*, 13(1):126–145, 2011.

[24] A. Gladisch, R. Daher, and D. Tavangarian. Survey on Mobility and Multihoming in Future Internet. *Wireless Personal Communications*, 74(1):45–81, 2014.

[25] X. Guo, O. Hernández-Lerma, T. Prieto-Rumeau, X. R. Cao, J. Zhang, Q. Hu, M. Lewis, and R. Vélez. A survey of recent results on continuous-time Markov decision processes. *TOP*, 14(2):177–261, 2006.

[26] S. Ha, C. Joe-Wong, S. Sen, and M. Chiang. Pricing by timing: Innovating broadband data plans. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 8282, 2012.

[27] P. Hande, M. Chiang, R. Calderbank, and J. Zhang. Pricing under constraints in access networks: Revenue maximization and congestion management. In *Proceedings - IEEE INFOCOM*, 2010.

[28] Y. Hayel and B. Tuffin. A Mathematical Analysis of the Cumulus Pricing Scheme. *Computer Networks*, 47(6):907–921, April 2005.

[29] H. He, K. Xu, and Y. Liu. Internet resource pricing models, mechanisms, and methods. *Networking Science*, 1(1-4):48–66, 2012.

[30] J. Hou, J. Yang, and S. Papavassiliou. Integration of pricing with call admission control to meet QoS requirements in cellular networks. *IEEE Transactions on Parallel and Distributed Systems*, 13(9):898–910, 2002.

[31] L. Hua, Y. Wei, and L. Qingchen. An adaptive control strategy for QoS in wireless multimedia network. In *WIT Transactions on Information and Communication Technologies*, volume 48, pages 373–379, 2014.

[32] L. Jiang, S. Parekh, and J. Walrand. Time-dependent network pricing and bandwidth trading. In *IEEE Network Operations and Management Symposium Workshops (NOMS)*, pages 193–200, 2008.

[33] W. Jianping, L. Hewu, S. Wenqi, W. Qian, J. Zhuo, and Z. Wei. Technology trends and architecture research for future mobile internet. *China Communications*, 10(6):14–27, June 2013.

[34] N. Keon and G. Anandalingam. A new pricing model for competitive telecommunications services using congestion discounts. *INFORMS Journal on Computing*, 17(2):248–262, 2005.

[35] J. Kosinski, D. Radziszowski, K. Zielinski, S. Zielinski, G. Przybylski, and P. Niedziela. Definition and Evaluation of Penalty Functions in SLA Management Framework. In *4th International Conference on Networking and Services (ICNS)*, pages 176–181, 2008.

[36] K. Kritikos, B. Pernici, P. Plebani, C. Cappiello, M. Comuzzi, S. Benrernou, I. Brandic, A. Kertész, M. Parkin, and M. Carro. A Survey on Service Quality Description. *ACM Computing Surveys*, 46(1):1:1–1:58, July 2013.

[37] S. Li, L. Xu, and S. Zhao. The internet of things: a survey. *Information Systems Frontiers*, pages 1–17, 2014.

[38] H. Luo and M. L. Shyu. Quality of service provision in mobile multimedia - a survey. *Human-centric Computing and Information Sciences*, 1(1):1–15, 2011.

[39] S. Mandal, D. Saha, and A. Mahanti. A technique to support dynamic pricing strategy for differentiated cellular mobile services. In *IEEE Global Telecommunications Conference (GLOBECOM)*, volume 6, pages 3388–3392, 2005.

[40] V. G. Ozianyi, N. Ventura, and E. Golovins. A novel pricing approach to support QoS in 3G networks. *Computer Networks*, 52(7):1433–1450, 2008.

[41] S. Parsons, J. A. Rodriguez-Aguilar, and M. Klein. Auctions and Bidding: A Guide for Computer Scientists. *ACM Computing Surveys*, 43(2):10:1–10:59, February 2011.

[42] I. C. Paschalidis and J. N. Tsitsiklis. Congestion-dependent pricing of network services. *IEEE/ACM Transactions on Networking*, 8(2):171–184, 2000.

[43] R. Pastor-Satorras and A. Vespignani. *Evolution and Structure of the Internet: A Statistical Physics Approach.* Cambridge University Press, New York, USA, 2004.

[44] H. Pervaiz. A Multi-Criteria Decision Making (MCDM) network selection model providing enhanced QoS differentiation to customers. In *International Conference on Multimedia Computing and Information Technology (MCIT)*, pages 49–52, 2010.

[45] H. Pervaiz and J. Bigham. Game Theoretical Formulation of Network Selection in Competing Wireless Networks: An Analytic Hierarchy Process Model. In *Third International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST)*, pages 292–297, 2009.

[46] S. Ren, F. Fu, and M. Van Der Schaar. Traffic-dependent pricing for delay-sensitive multimedia networks. In *IEEE Global Telecommunications Conference (GLOBECOM)*, 2011.

[47] P. L. Scandizzo and M. Ventura. Bids for the UMTS system: An empirical evaluation of the Italian case. *Telecommunications Policy*, 30(10-11):533–551, 2006.

[48] S. Sen. Bare-knuckled Wireless. *Business Today*, March 10 2011.

[49] S. Sen, C. Joe-Wong, S. Ha, and M. Chiang. Pricing data: A look at past proposals, current plans, and future trends. *Technical Report, Princeton University*, 2012.

[50] S. Sen, C. Joe-Wong, S. Ha, and M. Chiang. A survey of smart data pricing: Past proposals, current plans, and future trends. *ACM Computing Surveys*, 46(2), 2013.

[51] S. Shakkottai, E. Altman, and A. Kumar. Multihoming of Users to Access Points in WLANs: A Population Game Perspective. *IEEE Journal on Selected Areas in Communications*, 25(6):1207–1215, August 2007.

[52] S. Shakkottai and R. Srikant. Economics of network pricing with multiple ISPs. *IEEE/ACM Transactions on Networking (TON)*, 14(6):1233–1245, 2006.

[53] S. Shakkottai, R. Srikant, A. Ozdaglar, and D. Acemoglu. The price of simplicity. *IEEE Journal on Selected Areas in Communications*, 26(7):1269–1276, 2008.

[54] R. J. Shapiro. The Internet's Capacity to Handle Fast-Rising Demand for Bandwidth. US Internet Industry Association, 2007.

[55] W. Soon. A review of multi-product pricing models. *Applied Mathematics and Computation*, 217(21):8149–8165, 2011.

[56] B. M. Sousa, K. Pentikousis, and M. Curado. Multihoming Management for Future Networks. *Mobile Networks and Applications*, 16(4):505–517, August 2011.

[57] J. Sztrik. *Basic Queueing Theory*. University of Debrecen: Faculty of Informatics, 2011.

[58] R. Trestian, O. Ormond, and G. M. Muntean. Game Theory-Based Network Selection: Solutions and Challenges. *IEEE Communications Surveys and Tutorials*, 14(4):1212–1231, 2012.

[59] J. Watson. *Strategy: An Introduction to Game Theory*. W.W. Norton, 2013.

[60] V. V. Williams. Breaking the coppersmith-winograd barrier. *Unpublished manuscript, November*, 2011.

[61] K. Xu, Y. Zhong, and H. He. *Internet Resource Pricing Models*. Springer Publishing Company, Incorporated, 2013.

[62] S. Yaipairoj and F. C. Harmantzis. Dynamic pricing with "alternatives" for mobile networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, volume 2, pages 671–676, 2004.

[63] M. Yampolskiy, W. Fritz, and W. Hommel. Managing network quality of service in current and future internet. *Network and Traffic Engineering in Emerging Distributed Computing Applications*, pages 179–223, 2012.

[64] G. Zachariadis and J. A. Barria. Dynamic pricing and resource allocation using revenue management for multiservice networks. *IEEE Transactions on Network and Service Management*, 5(4):215–226, 2008.

[65] B. Zhang, R. Liu, D. Massey, and L. Zhang. Collecting the Internet AS-level Topology. *SIGCOMM Computer Communication Review*, 35(1):53–61, January 2005.

[66] Y. Zhang, C. Lee, D. Niyato, and P. Wang. Auction approaches for resource allocation in wireless systems: A survey. *IEEE Communications Surveys and Tutorials*, 15(3):1020–1041, 2013.