An Economic Framework for Resource Allocation in Ad-hoc Grids

An Economic Framework for Resource Allocation in Ad-hoc Grids

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof. dr. ir. J.T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen

op donderdag 26 november 2009 om 12:30 uur

door

Behnaz Pourebrahimi

Master of Science in Computer Engineering -
Machine Intelligence and Robotics,
Shiraz University, Shiraz, Iran
geboren te Kerman, Iran

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. K.G.W. Goossens

Copromotor: Dr. K.L.M. Bertels

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof. dr. K.G.W. Goossens	Technische Universiteit Delft, promotor
Dr. K.L.M. Bertels	Technische Universiteit Delft, copromotor
Prof. dr. Magnus Boman	KTH Royal Institute of Technology
Dr. Luc Onana Alima	Université de Mons-Hainaut
Prof. dr. Jan Broeckhove	Universiteit Antwerpen
Prof. dr. Frances Brazier	Vrij Universiteit Amsterdam
Prof. dr. Cees Witteveen	Technische Universiteit Delft
Prof. dr. John Long, reservelid	Technische Universiteit Delft

Behnaz Pourebrahimi
An Economic Framework for Resource Allocation in Ad-hoc Grids
Ph.D Thesis, Delft University of Technology

Keywords: resource allocation, ad-hoc Grids, market-based mechanisms, price, network condition, design choices.

ISBN: 978-90-72298-04-1

Cover page: ATOC Design (http://www.atocdesigns.com/)

Printed by Sieca Repro, Delft, the Netherlands

In memory of my parents

An Economic Framework for Resource Allocation in Ad-hoc Grids

Behnaz Pourebrahimi

Abstract

In this dissertation, we present an economic framework to study and develop different market-based mechanisms for resource allocation in an ad-hoc Grid. Such an economic framework helps to understand the impact of certain choices and explores what are the suitable mechanisms from Grid user/owner perspectives under given circumstances. We focus on resource allocation in a Grid-based environment in the case where some resources are lying idle and could be linked with overloaded nodes in a network. In such networks, the resources are neither necessarily dedicated nor have predictable availability at any point in time. We call such networks ad-hoc Grids. Self-interested nodes in ad-hoc Grids are considered as consumers (buyers) and producers (sellers) of resources within the economic framework. Consumers and producers of resources are autonomous agents that cooperate through a simple, single metric namely the price that summarizes the global state of a network in a number. The price represents all the available information that may reside at the level of the individual nodes and that is not necessarily shared among them. A middle agent, named the matchmaker, sets up a mutual agreement between consumer and producer agents based on the price by employing economic mechanisms such as auctions. The transaction is established when the consumer and producer constraints such as resource quantity, time, and budget are met. In this dissertation, we study market-based resource allocation mechanisms at macro and micro levels. Macroeconomics addresses the behavior of an economy at the aggregate level and microeconomics describes the individual behavior. At the macro level, we compare different economic models as the matchmaking mechanisms. We study the impact of choosing particular auction mechanisms in the framework. At the micro-level, we study different pricing mechanisms and investigate the effect of introducing money and budget constraints. Furthermore, we analyze different bidding strategies that help agents to better achieve their objectives under varying constraints.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Problem Definition

Back in 1960s and 1970s, computers were rare and expensive and CPU time was scheduled and billed by millisecond. These days, CPU time is so cheap that simple screen saver programs, which do not involve computationally complex tasks, are only stirring up pixels on the display screen. Such programs probably consume most of the world's computational capacity compared to any other kind of software. At the same time, the digital age allows to collect terabytes of data on various aspects of our personal lives as well as on all kinds of business transactions.This thesis looks at the ways to match the idle computing resources with the increasing need for data mining for which more computation power is required than can be provided by a single computer. More specifically, we look at the problem of resource allocation in a Grid based environment and study how the resources can be found for specific tasks. We focus on LANs or WANs that can be found in any kind of organization and that can be restructured to form a computing Grid. In such networks, the resources are neither necessarily dedicated nor it is predictable how many resources will be available at a certain time. We call such networks **ad-hoc Grids**. The ad-hoc Grid is a type of Grid that aims to harness unused computational resources inside or across organizations. In an ad-hoc Grid, any node in the network can spontaneously arise as a resource consumer or a resource producer at any time. Participating nodes in an ad-hoc Grid may have different objectives and preferences and they may provide a variety of resources to the Grid or may have a variety of tasks to perform.

Conventional resource allocation schemes are based on relatively static models where a centralized controller manages jobs and resources. In fact, they focus on efficient allocation schedules which can optimize given performance metrics such as allocation time, resource utilization or system throughput. These resource allocation mechanisms may work well where resources are known in advance. However, they may fail to work in dynamic networks where jobs need to be executed by computing resources whose availability is difficult to predict. Due to the dynamic nature of ad-hoc Grids, mechanisms that are based on a system-wide performance metric to allocate resources, are not suitable. The status of resources and tasks are not known or predictable in an ad-hoc Grid. Therefore, resource allocation in an ad-hoc Grid needs mechanisms that are both system-centric as well as user-centric and can adapt to variations in supply and demand. Market-based mechanisms provide promising directions for building such a resource allocation mechanism. One of the promises, taken from economic theory, is that the fulfillment of individual self-interest automatically or through an unspecified mechanism called the *Invisible Hand* (proposed by Adam Smith [69]), leads to maximal generation of utility for the entire community. When transposed to the Grid environment, one can obtain the overall goal, which is the processing of data. This implies that as long as individual nodes look after themselves, by buying or selling resources, the overall goal, namely to execute tasks, is also satisfied.

This thesis is not only intended to define a specific resource allocation mechanism but also aims to study different design choices for resource allocation in ad-hoc Grids considering the existing challenges. We tend to make it clear that under given conditions, which particular market-based mechanism suits the best for a Grid user/owner. To accomplish this, we present an economic framework where we can study different resource allocation mechanisms in various conditions.

1.1 Research Challenges

A resource allocation mechanism for ad-hoc Grids should address the following challenges:

1.1.1 Resource Discovery

A resource discovery mechanism defines how a node finds appropriate resources to perform its tasks and how it requests for additional tasks when its task queue is getting empty. In an ad-hoc Grid, resource discovery mechanisms are needed to be scalable and to be able to deal with unpredictable availability of resources. Fully centralized or fully decentralized resource management mechanisms are considered as the two extremes of an architectural spectrum and each have their own advantages and disadvantages. Centralized methods with a central server have good throughput for a certain population size and guarantee that a task finds required resource if it exists and they are able to find the best match system-wide. At the same time, they have low scalability because of the bottleneck associated with the central server and also they are not robust, since the whole system may fail when the central server is no longer available for whatever reason. On the other hand, a completely decentralized and unstructured mechanism is robust but lacks manageability and may cause a communication bottleneck [72]. Moreover, it does not guarantee finding a match even if it exists in the network.

Fully decentralized and fully centralized systems are often considered as alternatives to each other, residing on both ends of an infrastructural spectrum. We consider them to be part of a continuum where the system should be capable of restructuring itself in either of these states or any intermediate state between the two extremes. The system should organize itself to adapt to changing the circumstances such as system workload. We look for mechanisms that can provide the system level information to facilitate such self-organization.

Economics can be used as a way to provide the restructuring information. In economics, price can be defined as a simple, single metric that summarizes the global state of a network in a number.

1.1.2 Matchmaking

A matchmaking mechanism that provides fair access to resources for everyone is another challenge. Matchmaking is the process of finding an appropriate resource for a task request through a middle agent (matchmaker). The main goal of matchmaking is to establish a mutual agreement between a resource producer and a resource consumer by which a producer agrees to sup-

ply a capability that can be used to perform some tasks on behalf of a consumer.

Economic models can be employed for the matchmaking of consumers and producers of resources. In such mechanisms price is the main element to settle an agreement between a consumer and a producer.

1.1.3 Self-interested Nodes

Nodes are considered to be self-interested in ad-hoc Grids. Self-interested nodes attempt to increase their profit without considering the global profit of the network. According to the term *Invisible Hand* by Adam Smith, the collective interest is served through individual self-interested behavior. Therefore, the objective of an ad-hoc Grid which is to process as many tasks as possible, can be achieved by allowing each node to only consider its own local state (to be self-interested). One way to provide such a facility is to use market-based mechanisms. In these mechanisms, decision on resource allocation or task assignment can be made by providing the consumers/producers with a monetary system, and modeling them as buyers and sellers of resources. Each consumer is endowed with money that it uses to purchase the required resources. Each producer owns a set of resources and charges consumers for the use of its resources.

1.1.4 Self-organization

Self-organization denotes a self-adaptation mechanism that can automatically initiate a modification according to changing circumstances within a system. An ad-hoc Grid in which the resources are not dedicated and their availability may change frequently, cannot be managed by a single controlling authority. In fact, a monitoring system or a centralized server to collect all network information is not feasible in such sporadic environments. Therefore, embedding self-organizing mechanisms inside ad-hoc Grids is necessary.

A self-organizing mechanism in an ad-hoc Grid can embrace two levels of adaptation namely system level adaptation and node level adaption. In system level adaptation, the system organizes its structure according to the changing circumstance such as growing and shrinking the population. For instance, a

self-adaptation mechanism at system level can organize the system structure in the continuum between fully centralized to fully decentralized by introducing more/less central servers. In the node level adaptation, individuals adapt to the variations in the network condition such as supply and demand for the resources. Self-adaptation at the node level can enable for the system level adaptation since the system level state is provided by state of the individual nodes. A way of providing self-adaptation at the node level is to learn network condition through interactions with the environment. Learning from interactions is a basic idea behind nearly all theories of learning and intelligence [89]. We look for such mechanisms that provide node level adaptation and through which system level information are obtained.

Microeconomic mechanisms can be used to implement self-adaptation. In these mechanisms, individuals are able to make decisions based on their own preferences and according to the condition of the market. Price is the main concept in these mechanisms based on which decisions are made. Therefore, a pricing mechanism that can learn the changes in an environment is required to perform adaptation.

1.2 Research Questions

In this thesis, we aim to study market-based resource allocation mechanisms in ad-hoc Grids. Considering the above mentioned challenges, the following research questions have to be addressed:

What is the system architecture based on which a resource allocation mechanism can be built?

Peer-to-peer systems are able to deal with decentralized and dynamic nature of ad-hoc Grids. Resource discovery in peer-to-peer systems can be conducted from fully centralized to fully decentralized. Different peer-to-peer architectures based on which a resource allocation mechanism can be built, are studied in Chapter 2.

Which market-based mechanisms are suitable for resource allocation in an ad-hoc Grid?

Spontaneous, heterogenous, and dynamic nature of ad-hoc Grids requires mechanisms that deal with these challenges. We study economic models in macro and micro levels and select mechanisms that meet the requirements of an ad-hoc Grid. These issues are addressed in Chapter 3.

How can the self-interested nodes be presented in an ad-hoc Grid?

The self-interested nodes can be presented by autonomous agents. The autonomous agents make their own decisions according to their budgets, capabilities, goals, and local knowledge without considering the global good of the entire ad-hoc Grid. The proposed economic framework in Chapter 4, models consumers and producers of resources as self-interested and autonomous agents. In this framework, money and pricing are introduced as a technique for coordinating the selfish behavior of the agents.

How do the self-interested nodes in an ad-hoc Grid adapt to varying availability of resources and tasks?

Due to highly dynamic nature of an ad-hoc Grid, providing global information about the status of the network is not feasible. Therefore, we look for simple mechanisms that need the least global information. For this purpose, we use price as an indicator of global status of the network. Dynamic and learning pricing mechanisms are therefore required where price of a resource reflects the resource supply as well as the resource demand. Chapters 6 and 7 present self-adaptation using such a pricing mechanism.

How can different design choices for market-based resource allocation in ad-hoc Grids be evaluated?

An economic framework allows us to study different resource allocation mechanisms. To implement and evaluate these mechanisms, we develop an experimental simulation platform. In this platform, we compare the performance of the mechanisms from the Grid user and owner perspectives in dif-

ferent network conditions. Chapters 5 and 8 are devoted to the evaluation of different design choices.

1.3 Thesis Overview

This dissertation is organized as follows:

- Chapter 2 presents the global context of the research with studying Grid and peer-to-peer systems. In this chapter, we study resource discovery mechanisms that are based on peer-to-peer architectures. A self-organizing resource discovery mechanism for ad-hoc Grids that can restructure itself based on variability of workloads and resources is discussed. In this chapter, we give an overview of the related market-based research and highlight the differences of our work with the previous research.
- Chapter 3 discusses economic concepts and models. Behavior of an economy in the individual level is named microeconomics, and in the system level is named macroeconomics. In this chapter, we study market-based mechanisms in the micro and macro levels and discuss the mechanisms that are suitable for resource allocation in an ad-hoc Grid.
- Chapter 4 introduces the economic framework. Different framework components, the interaction between them and their attributes are discussed in this chapter. The framework is used to implement and study different resource allocation mechanisms. The evaluation of different mechanisms is performed through simulation. In this chapter, we describe the experimental platform by showing the setup and conditions in which the experiments are conducted.
- Chapter 5 studies the use of different market-based matchmaking mechanisms within the framework. Among different economic models, we select those that fulfil the requirements of an ad-hoc Grid. The performance of different mechanisms are evaluated in accordance with the criteria that are important to both user and owner of the ad-hoc Grid.
- Chapter 6 proposes a learning and adaptive pricing mechanism for resource allocation in an ad-hoc Grid. The performance of the pricing

mechanism is evaluated in different network conditions. The parameter regime of pricing mechanism is studied, and the usefulness of a budget constraint in the pricing mechanism is investigated.

- Chapter 7 introduces two aggressive and conservative bidding strategies in the pricing mechanism presented in Chapter 6. Based on these bidding strategies, agents become aggressive or conservative to increase their utilization of the ad-hoc Grid. We study how these strategies are adopted by agents as the condition of the network changes.
- Chapter 8 studies the impact of different choices for the pricing mechanisms in an ad-hoc Grid. Four pricing mechanisms are implemented and studied in the framework. Three pricing mechanisms are selected from the literature and the fourth is one we propose.
- Chapter 9 concludes the thesis by summarizing the chapters and discussing future research directions.

Chapter 2

Background & Related Research

Computer network systems can be classified into centralized and distributed systems. A distributed system is one in which components located at networked computers, communicate and coordinate their actions only by passing messages. The motivation for constructing and using distributed systems stems from a desire to share resources. The term *resource* is a rather abstract one, but it best characterizes the range of peripherals that can be usefully shared in a networked computer system. It extends from hardware components such as storage disks, processing power, and printers to software-defined entities such as files, databases and data objects of all kinds. Resources may be managed by servers and accessed by clients or they may be encapsulated as objects and accessed by other client objects.

Distributed systems can be classified into client-server and peer-to-peer models (Figure 2.1). A client-sever model can be flat where all clients only communicate with a single server (possibly replicated for improved reliability), or it can be hierarchical for improved scalability. In a hierarchical model, the servers of one level act as the clients to higher level servers. In the client-server model, the server is the central registering unit, as well as the only provider of content and services. A client only requests content or the execution of services, without sharing any of its own services. A peer-to-peer network is a distributed network composed of a large number of distributed, heterogeneous, autonomous, and highly dynamic peers. In peer-to-peer networks, participants share a part of their own resources such as processing

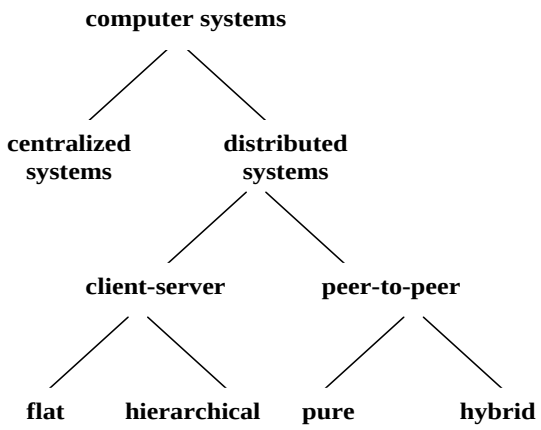

Figure 2.1: *Classification of computer network systems.*

power, storage capacity, softwares, or files contents. The participants in the peer-to-peer network can act as a server and a client at the same time. They are accessible by other nodes directly, without passing through intermediary entities. The peer-to-peer models can be pure or hybrid [81]. In a pure peer-to-peer model, any single, arbitrary chosen terminal entity can be removed from the network without suffering any loss of network service. Hybrid peer-to-peer model allows the existence of central entities in its network to provide parts of the offered network services.

2.1 Grid Systems

The Grid is a type of distributed system that enables sharing, selection, and aggregation of geographically distributed autonomous resources dynamically at runtime depending on their availability, capability, performance, cost, and users' quality-of-service requirements [26]. Grid systems interconnect computer clusters, storage systems, instruments, and in general, available infrastructure of large scientific computing centers. Grid enables sharing of existing resources such as CPU time, storage, equipment, data, and software applications. Most Grid systems are of moderate-size; centrally or hierarchically administered and there are strict rules governing the availability of the participating resources (i.e., a large percentage of the CPU time of a participating

cluster should be dedicated for Grid use 24 hours a day). Grids are used for complex scientific applications which are time critical and restricted by Quality of Service (QoS) rules. By using Grids, users can reduce computation time, access large databases, access special equipment, or collaborate with other users. According to user needs, Grid systems are classified into the followings:

- **Distributed supercomputing**: These systems use Grids to solve very large problems that cannot be solved on a single system and need lots of processing time, memory, etc. Depending on the use of Grid, these aggregated resources might comprise of the majority of the supercomputers in a country or simply all workstations within a company. One example of such systems is Distributed Interactive Simulation (DIS) which is a technique used for training and planning in the military [23]. Another example is the accurate simulation of complex physical processes such as cosmology which can employ coupled supercomputers to overcome resolution barriers [71].
- **High-throughput computing**: In high-throughput computing, the Grid is used to schedule large numbers of loosely coupled or independent tasks with the goal of utilizing unused processor cycles (often from idle workstations). Condor [65] is an instance of such kind of systems that manages pools of hundreds of workstations around the world and allows the use of idle CPU cycles among them.
- **On-demand computing**: On-demand computing uses Grid capabilities to meet short-term requirements for resources that cannot be cost-effective or conveniently located locally. These resources may be computation, software, data repositories, specialized sensors, and so on. In contrast to distributed supercomputing, on-demand computing is often driven by cost-performance concerns rather than absolute performance. An example is Netsolve [30] which is a client-server system that enables users to solve complex scientific problems remotely. This system allows users to access both hardware and software computational resources distributed across a network. Cloud computing is also on-demand computing in which dynamically scalable and often virtualized resources are provided on-demand as a service over the Internet [49].
- **Data-intensive computing**: In data-intensive computing, the focus is on synthesizing new information from data that is maintained in geo-

graphically distributed repositories, digital libraries, and databases. A good example of a data-intensive computing is the problem of assimilating remote satellite observations into a comprehensive data set that describes the weather over the entire globe [54].

- **Collaborative computing**: A Grid may be considered as a set of distributed services that are accessed securely by various remote clients working in collaboration with each other. Example of such services include secure remote command execution, file management, database access, and access to queuing systems.

2.2 Grid Applications

Grids are used mostly for complex scientific applications that are computational intensive, data intensive, or require distributed collaborations. These applications are deployed in astronomy, biomedicine, climate modeling and earthquake prediction. There are several projects on Grid that support a wide range of applications, namely:

- **International Grid projects**: These projects create international resources and power global science using global computing. EGEE (Enabling Grids for E-sciencE) [5] is the largest multi-disciplinary Grid infrastructure in the world, bringing together more than 120 organizations to provide scientific computing resources to the European and global research community. OGF (Open Grid Forum) [13] is an open community committed to driving the rapid evolution and adoption of applied distributed computing. The Open Grid Forum is a community-initiated forum of people interested in distributed computing and Grid technologies.
- **National Grid projects**: These projects combine national computing resources to create powerful Grid computing resources such as D-Grid [7]. D-Grid started with the goal of developing a distributed, integrated resource platform for high-performance computing and related services to enable the processing of large amounts of scientific data and information. DutchGrid [4] is a platform for Grid computing and technology in the Netherlands open to all institutions for research and test-bed activities; it aims to coordinate various Grid deployment efforts and to offer

a forum for exchange of experiences on Grid technologies.

- **Field-specific Grid projects**: These projects have been created to tackle specific scientific problems. AstroGrid [1] enables astronomers to explore and bookmark resources from around the world, find data, store and share files, query databases, plot and manipulate tables, cross-match catalogues, build and run scripts to automate sequences of tasks. ESG [6] aims to enable the next generation of climate research, to power new ways of analyzing and developing knowledge from global earth system models. The Biomedical Informatics Research Network (BIRN) [2] is a geographically distributed virtual community aiming to advance the diagnosis and treatment of human diseases.
- **Volunteer computing projects**: These projects use the CPU scavenging model. CPU-scavenging or cycle stealing creates a Grid from the unused resources in a network of participants (whether worldwide or internal to an organization). Typically this technique uses desktop computer instruction cycles that would otherwise be wasted at night, during lunch, or even in the scattered seconds throughout the day when the computer is waiting for user input or slow devices. Any computer can donate resources and one or a small number of applications can use such resources. For example, SETI@home [14] which uses volunteer computers to search for intelligent life outside earth. It is run from the University of California-Berkeley in the USA. Likewise Climateprediction.net [3] uses idle processor cycles contributed by volunteers to investigate state-of-the-art climate models; it studies the response of climate to slight tweaks in the models.
- **Middleware projects**: These projects provide tools for distributed application management to use distributed resources. A Middleware offers services such as remote process management, co-allocation of resources, storage access, information registration and discovery, security, and aspects of QoS such as resource reservation and trading. Globus project [8], the UNIform access to COmputing REsources (UNICORE) [18] are instances of the middleware projects.

2.3 Peer-to-Peer Applications & Architectures

The most popular service provided by peer-to-peer systems is file sharing. File sharing applications [20, 64, 77] focus on storing information on and retrieving information from various peers in the network. Both Grid and peer-to-peer technologies have the same final objectives, pooling and coordinating the use of large set of distributed resources, but are based on different communities and focus on different requirements. Grid computing addresses infrastructure but not yet failure, whereas peer-to-peer addresses failure but not yet infrastructure [41]. It has been argued in the literature that Grid and peer-to-peer systems will eventually converge [91]. The QoS constraints that currently govern most Grid applications will loosen up as Grids will move towards more popular and diverse application scenarios. Strict resource participation rules will be relaxed because participating organizations may need to have their infrastructure for own use at certain periods and for Grid jobs at other times, and the use of commodity hardware will be allowed. On the other hand, peer-to-peer systems will open up to more sophisticated applications and they will have to support more complex queries and different QoS levels [91].

Decentralization is one of the major concepts of peer-to-peer systems. Based on the degree of decentralization in a peer-to-peer system, we can classify them into two categories:

2.3.1 Purely Decentralized Architecture

A pure peer-to-peer system is a distributed system without any centralized control. In such systems all nodes are equivalent in functionality. In such networks, nodes are named *SERVENT* (SERVer+cliENT). The term servent represents the capability of the nodes in a peer-to-peer network for acting both as server as well as client. Freenet [32], Chord [86] and CAN [75] are instances of such systems.

Pure peer-to-peer systems are inherently scalable. Scalability in a system is usually restricted by the amount of centralized operations necessary, and such systems largely avoid central instances or servers. Peer-to-peer systems are inherently fault-tolerant since there is no single point of failure and the loss of a peer or even a number of peers can easily be compensated. Pure peer-to-

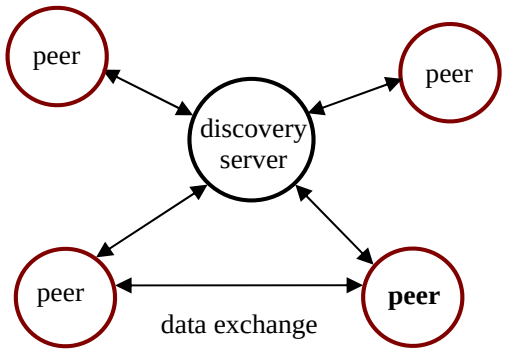

Figure 2.2: *Centralized Indexing.*

peer systems lack manageability since every peer is its own controller. Also, due to the lack of a global view at the system level, it is difficult to predict the system behavior. On the other hand, these systems present slow information discovery and there is no guarantee on quality of services. Such systems also tend to be insecure in the sense that it is easy for a node to join the network and start publishing malicious data into the network.

2.3.2 Hybrid Architecture

In hybrid peer-to-peer systems [100], there is a central server that maintains directories of information about registered users to the network in the form of meta-data. The end-to-end interaction (data exchange) is between two peer clients. There are two kinds of hybrid systems namely centralized indexing and decentralized indexing. In centralized indexing, a central server maintains an index of the data or files that are currently being shared by active peers (see Figure 2.2). Each peer maintains a connection to the central server through which queries are sent. This architecture is used by Napster[12]. Such systems are simple, and they operate efficiently for information discovery as searches are comprehensive and guaranties provided. On the other hand, centralized indexing systems may have a single point of failure. Moreover, they are not inherently scalable because a central server can become a bottleneck that limits scalability.

In decentralized indexing, there can be more than one central server that fa-

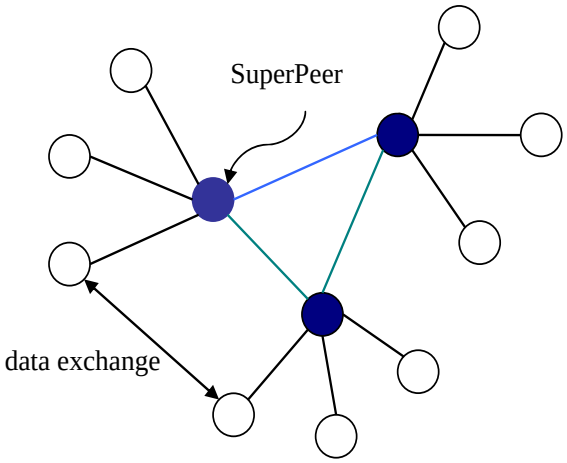

Figure 2.3: *Distributed Indexing.*

cilitate the discovery process. In these systems, some of the nodes assume a more important role than the rest of the nodes, and they are called *SuperPeers* or *SuperNodes* [100] (see Figure 2.3). SuperPeers maintain the central indexes for information shared by local peers connected to them, and proxy search requests on behalf of these peers. Queries are therefore sent to the SuperPeers, not to other peers. Kazaa [10, 46] and Morpheus [11] are two similar decentralized indexing systems. In such systems, peers are automatically elected to become SuperPeers if they have sufficient bandwidth and processing power, while a central server provides new peers with a list of one or more SuperPeers with which they can connect.

More recent architectures such as Gnutella [9] also use the concept of SuperPeers. When a node with a certain threshold CPU power joins the network, it immediately becomes a SuperPeer and establishes connections with other SuperPeers thereby forming a flat unstructured network of SuperPeers. SuperPeer sets the number of clients required for it to remain a SuperPeer. If it receives at least the required number of connections to client nodes within a specified time, it remains a SuperPeer otherwise it turns into a regular client node. If no SuperPeer is available, it tries to become a SuperPeer again for another probation period.

Decentralized indexing systems reduce the workload on central server but present slower information discovery in comparison with centralized index-

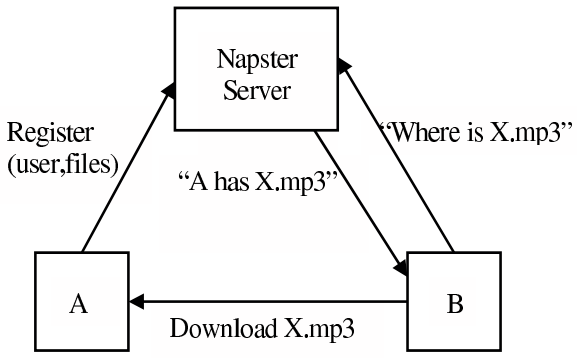

Figure 2.4: *Resource discovery in Napster.*

ing systems. Distributed indexing systems alleviate the drawbacks associated with centralized indexing such as scalability and single point of failure. Hybrid peer-to-peer systems (centralized & decentralized indexing) solve the manageability problem of pure peer-to-peer systems such that the control server(s) acts as a monitoring agent for all the other peers and ensures information coherence.

2.4 Peer-to-Peer Discovery Mechanisms

Peer-to-peer systems often require a discovery mechanism to locate specific data within the system. Discovery mechanisms in peer-to-peer systems have evolved from the first generation centralized structures to the second generation flooding-based, and then the third generation based on distributed hash tables [58]:

- **Centralized indexes and repositories**

  This mechanism is used in hybrid architecture. Based on this mechanism, the peers of the community connect to a centralized directory server which stores all information regarding location and usage of resources. Upon request from a peer, the central server matches the request with the best peer in its directory that meets the request requirements. The best peer could be the one that is cheapest, fastest, nearest, or most available, depending on the user needs; then the data exchange

will occur directly between the two peers. Napster uses this mechanism [20]. In Napster, a central directory server maintains an index with meta data (file name, time of creation etc.) of all files in the network, a table of registered user connection information (IP addresses, connection speeds etc.), and a table listing the files that each user holds and shares in the network. In the beginning, clients contact the central server and report a list with the files they maintain. When the server receives a query from a user, it searches for matches in its index and returns a list of users that hold the matching file. The user then opens a direct connection with the peer that holds the requested file and downloads the file (see figure 2.4).

- **Flooding broadcast of queries**

  This mechanism is applied in a pure peer-to-peer architecture. In this mechanism, no peer maintains any central directory about the shared contents in the peer-to-peer network. Since no single peer knows about all resources, peers in need for resources flood an overlay network queries to discover a resource. Each request from a peer is flooded (broadcasted) to directly connected peers, which themselves flood it to their peers, until the request is answered or a maximum number of flooding steps occurs. Flooding based search networks are built in an ad-hoc manner, without restricting a priori which nodes can connect or what type of information they can exchange [36]. Different broadcast policies have been implemented to improve search in peer-to-peer networks [101, 88, 92]. Although the flooding mechanism might give optimal results in a network with a small to average number of peers, it does not scale well. Blind flooding search limits scalability, since a large number of exchange messages is needed in the large scale systems. Furthermore, accurate discovery of peers is not guaranteed in flooding mechanisms.

- **Routing**

  The routing mechanism adds structure to the way information about resources are stored using distributed hash tables. This mechanism provides a mapping between the resource identifier and location in the form of a distributed routing table so that queries can be efficiently routed to the node with the desired resource. This mechanism reduces the number of peer-to-peer hops that must be taken to locate a resource. The

look-up service is implemented by organizing the peers in a structured overlay network and by routing a message through the overlay to the responsible peer [42]. In structured networks, the topology is tightly controlled and data is placed at specific locations. Some instances of such systems that have implemented distributed peer-to-peer look-up services are: DKS [17], an infrastructure for building peer-to-peer applications based on routing tables; Freenet [32], a file-storage service; Chord [86], a decentralized peer-to-peer lookup mechanism that stores key/value pairs for distributed data items; CAN [75], a distributed infrastructure that provides hash table-like functionality on the internet like scales; and Pastry [76], an overlay and routing network for the implementation of a distributed hash table similar to Chord. These structured systems can solve scalability limitation. But, the disadvantage of these models is that it is hard to maintain the structure required for routing in a very transient node population, in which nodes join and leave at a high rate.

2.5 Resource Discovery Mechanisms in Grids

The limitations of client/server mechanisms for resource allocation have become evident in large scale distributed environments. In such systems, individual resources are concentrated on one or a small number of nodes. In order to provide access to resources with an acceptable response time, sophisticated load balancing and fault-tolerant algorithms have to be applied. Limitation on the network bandwidth adds to the bottleneck problem. These limitations have motivated researchers to suggest approaches to distribute processing loads and network bandwidth among all nodes participating in a distributed system. Peer-to-peer systems offer an alternative to traditional client/server systems that solve bottleneck problems and improve the Grid scalability.

In the literature, we can find several resource discovery mechanisms in Grids that adopt peer-to-peer architectures. In the following paragraphs, we provide a short overview of some of these mechanisms.

Iamnitchi et al. [53] propose a fully decentralized peer-to-peer architecture for resource discovery in Grid environments. In this architecture, every participant in a Virtual Organization (VO) publishes information on one or more

local servers, called nodes or peers, which store and provide access to local resource information. A node may provide information about one resource (e.g. itself) or multiple resources (e.g. all resources shared by an organization). Users send their requests to some known (typically local) node. The node responds with a matching resource description if it incorporates them, otherwise it forwards the requests to another node. Intermediate nodes forward a request until its Time To Live (TTL) expires or matching resources are found, whichever occurs first.

Mastroianni et al. [68] adopt the SuperPeer model to design a peer-to-peer based Grid information service. The SuperPeer model is proposed to achieve a balance between the inherent efficiency of centralized search, and the autonomy, load balancing and fault-tolerant features offered by distributed search [54]. A SuperPeer node acts as a centralized server for a number of regular peers, while SuperPeers connect to each other to form an overlay network that exploits peer-to-peer mechanisms at a higher level.

Puppin et al. [74] propose another Grid information service based on the SuperPeer model. In this work, Grid nodes are grouped into clusters where each cluster may include one or more SuperPeer nodes. The system defines two main components namely the Agent and the Aggregator. The Agent works as an OGSA (Open Grid Services Architecture) compliant Grid Service available at each network node. It publishes all information made available by the information providers. The information providers periodically query the resources and store the gathered information as Service Data Element (SDE). When a resource is published, the name of its Service Data is broadcasted to all the Aggregators in the cluster.

Marzolla et al. [67] propose a system for discovering Grid resources based on routing indexes. In this system, nodes are organized in a tree-structured overlay network, where each node maintains information about the set of resources that it manages directly and a condensed description of the resources present in the sub-trees rooted in each of its neighboring nodes. The data location algorithm exploits those indexes to route queries towards areas where matches can be found.

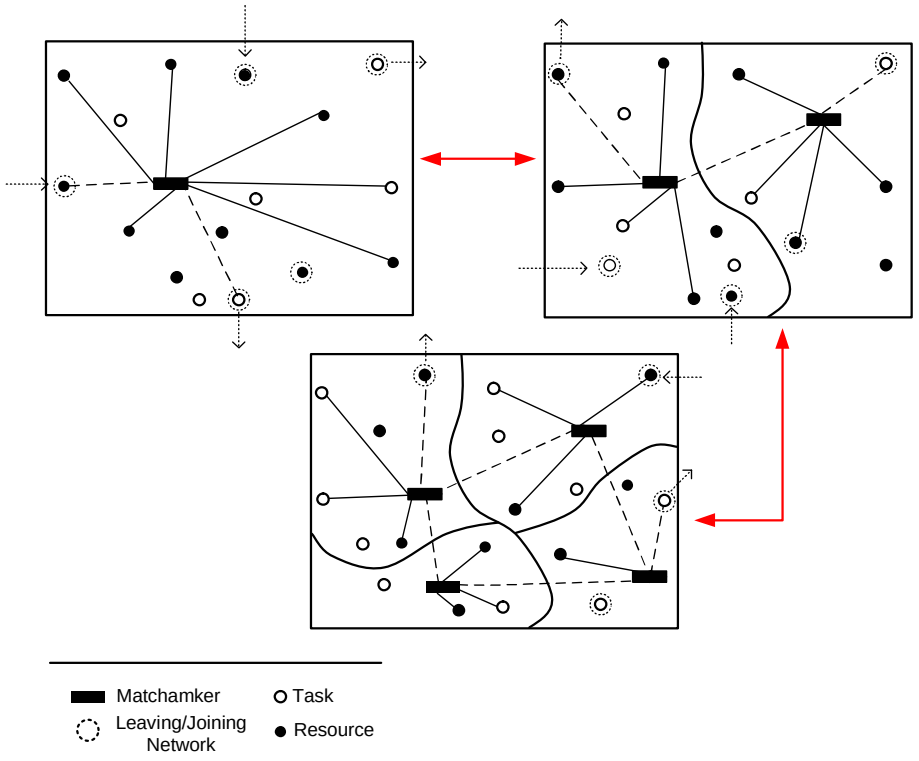

Figure 2.5: *A self-organizing mechanism for resource allocation in an Ad-hoc Grid*

2.6 Market-based Resource Allocation in Ad-hoc Grids

Resource allocation is the process of discovering and allocating resources to requested tasks in a way that satisfy both the application jobs and resource administrators. Different resource discovery mechanisms can be considered for ad-hoc Grids varying from fully centralized (centralized indexing) to fully decentralized ones (blind flooding). Fully centralized resource discovery mechanisms can be efficient for small scale systems and may take less time in finding a required resource. The searches are also comprehensive in these mechanisms. However, these mechanisms are not scalable and the centralized resource broker becomes a performance bottleneck. In contrast, fully decentralized resource discovery mechanisms do not have a single point of

failure and are scalable. The drawback is that fully decentralized mechanisms are computationally expensive and may take more time to find a resource. Fully decentralized mechanisms do not also guarantee finding a resource.

Fully centralized and fully decentralized can be considered as two ends of the infrastructural spectrum for resource discovery in ad-hoc Grids. A self-organizing mechanism capable of restructuring itself between these two ends, can be designed to overcome the shortcomings of theses two mechanisms. In Figure 2.5, an example of such self-organizing mechanism is depicted. When there is a relatively low workload, a single centralized matchmaker suffices. However, in case the workload increases, the load on the matchmaker will also increase and in order to ensure a timely and appropriate match, the matchmaker may decide to look for support by promoting other nodes to become matchmaker. This process may be repeated any number of times. In this way, the system organizes itself starting from a simple centralized state and evolving successively into increasingly distributed ones. One extreme situation, in the fully decentralized case, would be when every single node in the Grid becomes a matchmaker. The reverse could happen and nodes are demoted to turn into ordinary nodes when the workload decreases.

To understand appropriate mechanisms for self-organization in the range from fully decentralized to fully centralized, the question has to be how can the system determine what infrastructure, ranging from fully centralized to fully decentralized, is more appropriate given its current status which is defined in terms of available and requested resources. The challenge is to find a way to generate system wide information on the basis of the individual states of the participating nodes.

Economics may provide one way of doing so. The basic institution in the economy is the market where producers and consumers can meet. A transaction is completed when both parties reach an agreement on the quantity and the price of the goods (resources in our case). It is an accepted axiom in economic markets that all the available information which may reside at the level of the individual nodes and which is not necessarily shared among them, is consolidated into a simple global metric, named the price. The price is therefore an indicator on the scarcity of a particular good. Such that, if there is a shortage of resources, its price will go up and vice versa.

Moreover, the market-based mechanisms address the incentives of both users and suppliers to participate in Grid. As Buyya pointed out [28] , a market-

based resource allocation mechanism provides the following benefits:

- It helps in building large-scale computational Grid as it motivates resource owners to contribute their idle resources for others to use and take benefit from them.
- It provides a fair basis to access the Grid resources for every user. The resource owners charge the consumers according to the amount of resources they consume.
- It helps in regulating demand and supply in the Grid. The transactions are made in the price where supply and demand converge.
- It helps in building a scalable system as the decision-making process is distributed across all users and resource owners. Therefore, it removes the need for a central coordinator for negotiation.
- Resource allocation based on economics is both user-centric and system-centric. The system-centric mechanisms attempt to maximize overall throughput of the system. They ignore user-centric requirements. The market-based resource management support user-defined parameters such as deadlines and budget constraints.
- The dynamics of Grid performance is difficult to model. By formulating Grid resource usage in market terms, we are able to apply analytical research from economics for understanding of the behavior of the Grid.

2.7 Related Work

Resource allocation based on economics has been widely studied for distributed systems [33]. In computational Grids, some experiments have been done based on some popular economic mechanisms, such as auctions and commodity markets [26, 97, 98]. Several research have attempted to apply the concept of computational economy for information management, CPU cycles, storage, and network access. A few systems such as Java Market [19], Popcorn [70], and JaWS [63] have built market-oriented environments to harness the processing power of a small network of computers on the web-based systems. Spawn [94] is another example of a market-based system that harnesses idle processor time in a computer network. Tycoon [61] is a market-based distributed resource allocation system based on proportional share, where users

get resources in proportion to their pre-dened weight. Mungi [51] is a shared storage management system and Mariposa [87] is a distributed database management system. Some of these systems are based on computational clusters. For example Libra [82] is an economy-driven cluster scheduler. Nimrod-G [15, 25] is an economic based scheduler that uses the Globus toolkit services and can be extended to operate with any other emerging Grid middleware services. It supports economic models such as commodity market, spot market and contract-net. Wolski et al. [98] uses commodity market mechanism to allocate two types of resources (CPU and disk storage) in Grids.

In the literature, we can find several computational markets that use auctions for resource allocation [94, 70, 63]. Most of these works consider only one type of the auction and compare it with other economic and conventional models. Gomoluch et al. [45] investigate that under which circumstances a market-based resource allocation by continuous double auction and by the proportional share mechanism, outperforms a conventional Round-Robin approach. It is concluded that the continuous double auction performs best for a cluster of homogeneous resources. However, if the load is low, the differences between three mechanisms are small, and the computationally less expensive Round-Robin mechanism might be sufficient. For a situation where there is a choice of resources with different quality of load, as it is the case in a computational Grid, the results of Round-Robin is worse compared to the two other market-based mechanisms [45]. The continuous double auction performs best in most cases[45]. Weng et al. [96] present a periodic double auction mechanism with uniform price for resource allocation in Grids. In this work, auction takes place in rounds and all exchanges are performed with the same price. There are few research that compare different auction models. Kant et al. [57] compare three different variations of double auction mechanisms from both resource's and user's perspectives. Comparison parameters in [57] are resource utilization, resource profit, and consumed budget. It is concluded that continuous double auction mechanism outperforms the other two variations of double auctions regarding the evaluation criteria [57]. Grosu et al. [48] investigate three types of auction mechanisms; First-Price auction, Vickrey auction and double auction. Resource utilization, resource profit, and user payment are measured as the parameters for comparing these mechanisms. Simulation environment consists of limited number of resources with predefined capabilities, reservation price, and Risk Averse/Risk Neutral users. Their results show the First-price auction is better from resource's perspective

while Vickrey auction is better from user's perspective. Double Auction favors both resources and users [48]. Assuncao et al. [37] analyze the different auction models in terms of communication demand for resource allocation in Grid computing environments. The investigation is done on First-price, English, Dutch and continuous double auctions. Their experiments show that English auction presents higher communication requirements while continuous double auction presents the least demand of communications.

There are some issues that are not well addressed in the existing market-based research:

- Most of the market-based research have been done in the context of conventional Grids [15, 25, 26, 82, 97, 98]. In conventional Grids, resources are assumed to be dedicated with a fixed number of nodes which provide services. For instance, Ferguson [39] microeconomic algorithm is one of the earliest examples of applying market-based ideas to cluster management. In this algorithm, multiple sequential jobs compete for dedicated slices of CPU time on a collection of heterogeneous machines interconnected by point-to-point links.
- Many of the market-based resource management systems follow a single model for resource trading. In order to use some of these systems, applications have to be specifically developed for execution on those systems, which is generally discouraging. For instance, in systems like Popcorn [70], there is a need for highly specialized access modules. Popcorn requires each program to be written with the economy in mind using a special application programming interface.
- Another problem in the current systems is adaptability. There are several research efforts on Grid adaptability using a central monitoring system. For instance, [31] and [83] assume a centralized network architecture where a centralized server collects environmental conditions from each component to make replacement decisions. Gounaris et al. [47] address adaptive query processing on Grid in order to cope with evolving resource characteristics, such as machine load and availability. Their model supports partitioning and resource allocation in response to the changes in the resource performance and in the resource pool, respectively with the help of a monitoring information.
- Considering market-based resource allocation in Grid, although the

market-based resource allocation has attracted considerable attention [26, 97], but few research have addressed the problem of learning and adaptation. Most of these systems function just based on a predefined economic approach and there is no option to change the strategy in case of changing environment. Preist et al. [73] demonstrate the simple adaptive agents inspired by the ZIP (Zero Intelligence Plus) agent [35]. ZIP agent consists of a small number of heuristics and a simple learning rule. These agents learn to trade at an equilibrium price in a form of a periodic double auction marketplace. Bagnall et al. [21] propose an adaptation of two learning mechanisms (Zero Intelligence Plus & Gjerstad-Dickhaut [43]) for single sellers in First-price and Vickrey auctions.

In this thesis, we aim to address resource allocation in ad-hoc Grids based on market-based mechanisms. We look for the mechanisms that solve the existing problems in the previous research.

2.8 Conclusions

This chapter presented the research background by studying Grid and peer-to-peer systems and applications. We discussed different resource discovery mechanisms for Grids that are built based on peer-to-peer architectures. A pure peer-to-peer architecture lacks manageability but it is more fault-tolerant and has better scalability. A centralized indexing architecture provides manageability, but it is not scalable and has less fault-tolerant because of the single point of failure. Mechanisms that are based on decentralized indexing are manageable, scalable and fault-tolerant. In this chapter, we described how a resource discovery mechanism for an ad-hoc Grid can be designed based on architectures ranging from fully centralized to fully decentralized or having any intermediate state. We have argued for the use of economics to implement self-organization such that the Grid can adapt itself to changing circumstances and will have an infrastructure which is tuned towards a particular context. Especially the price of a resource is a crucial element in this view as the price is a simple metric that, in a synthetic way, represents information about the entire system's status. In the subsequent chapters we will present in more detail this approach.

Chapter 3

Economic Concepts and Mechanisms

In this chapter, we discuss economic concepts and mechanisms in the micro and macro levels. We want to provide a deeper understanding of the choices one can make as far as market-based mechanisms for resource allocation are concerned.

3.1 Microeconomics

Microeconomics is the branch of economics that deals with the behavior of individual buyers and sellers. It studies how individuals decide how much of a particular good they need and how much they are willing to pay for it. Microeconomics examines the impact of supply and demand for resources on price decisions and price behaviors. In turn, it also explains how prices influence supply and demand for good - resources in our case. The most important concepts in microeconomics are [62] [40]:

- **Price**: The concept of price is central to microeconomics where it is one of the most important variables in any economic decision. Pricing can be driven by demand and supply just like in real market environments. Pricing can also be driven by how much users value the resources. In this thesis, we will use three kinds of prices:
  - *Bid price (buyer price)* is the price offered by a buyer. Bid price can be a function of budget, deadline, task complexity, scheduling

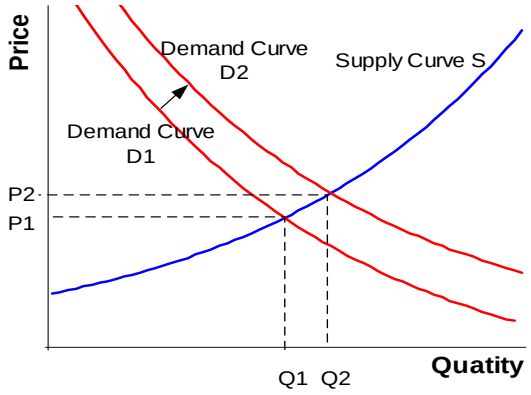

Figure 3.1: *Law of demand.*

strategy and resource requirements.

- *Ask price (seller price)* is the price offered by a seller. Ask price can be a function of current load, resource availability, and resource characteristics.
- *Transaction price* is the price at which the transaction is executed between a buyer and a seller. The transaction price can be also defined as the price at which the market is said *clear*, which is the point where supply and demand are equal to each other. Transaction price can be a function of buyer and seller prices or it can be set by resource or market owner.

- **Demand**: Demand is the expression of willingness or capability of a potential buyer to acquire certain quantities of a resource given certain prices.
  - **Law of demand**: The law of demand states that if supply is held constant, an increase in demand leads to an increase in transaction price, while a decrease in demand leads to a decrease in transaction price. As it is shown in Figure 3.1, for instance, when buyers increase the quantity demanded from $Q1$ to $Q2$, the price raises from $P1$ to $P2$. Accordingly, a *demand curve* depicts the relation between the price of a certain resource, and the resource quantity that buyers are willing or able to purchase at that given price.

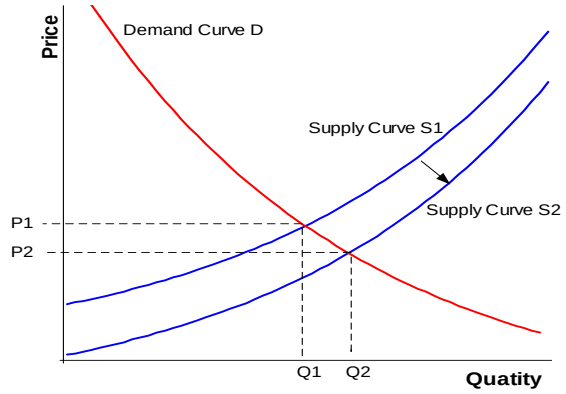

Figure 3.2: *Law of supply.*

A demand curve usually slopes downward because a lower price encourages the buyer to purchase more of a resource.

- **Supply**: Supply is the willingness or capability of sellers or suppliers to make available different possible quantities of a resource at all relevant prices.
  - **Law of supply**: The law of supply states that if demand is held constant, an increase in supply leads to a decrease in transaction price, while a decrease in supply leads to an increase in transaction price. Figure 3.2 depicts the law of supply. For instance, a decrease in supply from $Q1$ to $Q2$ causes decrease of price from $P1$ to $P2$. Accordingly, a *supply curve* presents the relation between the price of a certain resource and the resource quantity provided by the sellers. The supply curve usually slopes upward, since higher prices give the sellers an incentive to supply more in the hope of making greater revenue.
- **Equilibrium**: The price and quantity equilibrium refers to where demand and supply intersect. Figure 3.3 shows the an equilibrium point as demand and supply curves converge.
- **Utility Function**: utility functions are used to characterize the resource requirements and the degree of satisfaction of individual users. A utility function provides the maximal utility for buyers or sellers when faced

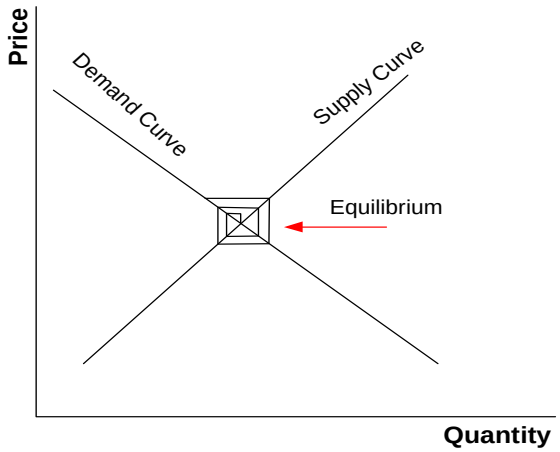

Figure 3.3: *Equilibrium price where supply and demand curves intersect.*

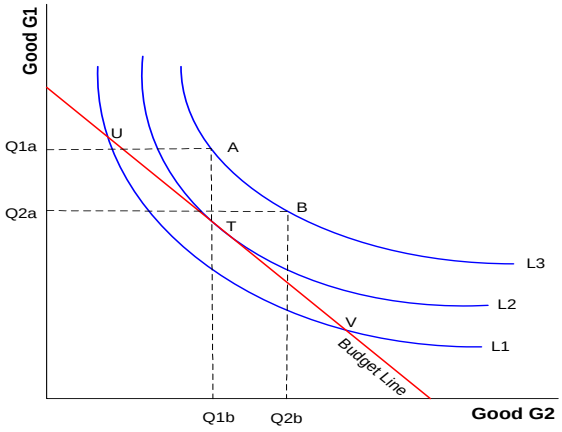

Figure 3.4: *Indifference curve & Budget Line.*

with a given price of the resource. For instance, in commodity markets, the utility is a function of the price. In these mechanisms, buyers and sellers update their demand and supply based on the new price decided by the market to maximize their utility. In auctions, the prices can be calculated as a function of utility in a way to increase the utility for a given demand/supply.

- **Indifference Curve & Budget Line**: In microeconomics, an indifference curve is a graph showing different combinations of goods, each measured by quantity, that yield the same level of utility for a consumer. That means at each point on the curve, the consumer has no preference for one bundle over another. Figure 3.4 shows three indifference curves which provide three different levels of utility ($L1$, $L2$, and $L3$) for a consumer. Any point on a higher indifference curve is preferred to any on a lower one. For instance, the point A provides a higher utility as compared to point V for a consumer. A budget line illustrates all the possible combinations of the goods that can be afforded by the consumer. In Figure 3.4, a consumer can afford all combination of goods $G1$ and $G2$ in the points below the budget line. The optimal combination of goods for a consumer is the point on the budget line where an indifference curve is tangent to the budger line. In Figure 3.4, the optimal combinations of goods $G1$ and $G2$ is shown as point T.

3.2 Macroeconomics

Macroeconomics is the branch of economics that studies the behavior of an economy at the aggregate level, as opposed to the level of a specific subgroups or individuals, which we discussed in the previous section. Macroeconomics investigates the interaction among individual components of the system. Based on macroeconomic mechanisms, we identify the following resource allocation mechanisms which can be classified as either non-price or price-based [52, 39]. Each of theses two categories is further classified into different models as Figure 3.5 shows. In the following, we discuss these economic models:

- **Non-price based mechanisms**: In these mechanisms, no price is involved and the mechanisms are either selfish or cooperative [50]. In selfish mechanisms that rely on the *game theory* [84], each user performs selfish optimization and has its own utility function, independent of the others [102, 103]. However, unlike selfish mechanisms, the cooperative mechanisms have a global utility function which is known to all nodes in the system. For instance, Kurose et al. [59] have provided decentralized algorithms to allocate resources (such as files) in a coop-

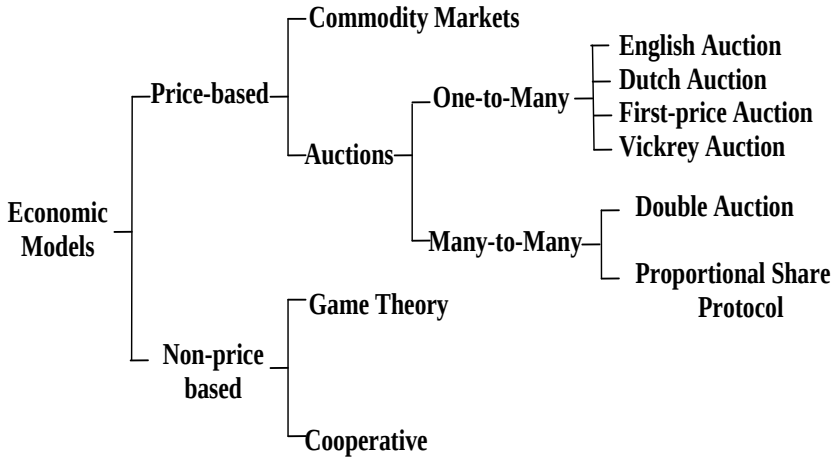

Figure 3.5: *Economic Models.*

erative and non-competitive manner among agents.

- **Price-based mechanisms**: In a price-based system, the resources are priced based on demand, supply, and wealth in the economic system. One approach is that buyers and sellers determine the quantity of resources they are willing to buy or sell. Then, the price-based mechanism computes the equilibrium price at which supply and demand converge and the market clears. An alternative but related mechanism is auction. In auctions, buyers and sellers set the price at which they want to buy or sell resources. The basic philosophy behind auction mechanisms is that the bidder with the highest price always gets the resource, and the current price for a resource is determined by the buyer and seller prices.

In price-based mechanisms, the actual price of resources can be a representative of supply and demand conditions in the market. Moreover, the individuals in a market can express their needs and preferences by the prices they bid. To build a market-based resource allocation mechanism in ad-hoc Grids, we tend to use the price as an indicator of the global status of the network as well as individual's status. Therefore, among the two mentioned mechanisms, we choose price-based mechanisms. There are two main classes of price-based mechanisms namely auctions and commodity markets.

3.2.1 Auction Mechanisms

Within auction mechanisms, pricing is driven by how much value resources owners place on the resources, and access to resources is won by the buyer whose valuation comes closest to that of the resource owner [27]. The most common auction mechanisms are *one-to-many* and *many-to-many* auctions. In one-to-many auctions, one agent initiates an auction and a number of other agents can make a bid. The *English auction*, *Dutch auction*, *First-price auction*, and *Vickrey auction (Second-price auction)* belong to this category. In many-to-many auctions, several agents initiate an auction and several other agents can bid. The *double auction* is the most known auction mechanism for many-to-many auctions. In this type of auction, buyers and sellers are treated symmetrically with buyers submitting bids and sellers submitting offers.

Auctions can also be classified into *open* or *close* auctions. In open auctions, bidders know the bid value of other bidders. In closed or sealed bid auctions, the bids are not disclosed to others.

One-to-Many Auctions

- **English Auction:** The English auction is a sequential bidding auction in which buyers take turns publicly to submit increasing bids. The seller openly announces a minimal price for the resource to be sold. Buyers decide on a price and quantity depending on the user-defined requirements (mainly deadline and budget that they are willing to invest for solving the problem). A bidder can have a strategy for a series of bids as a function of its private value and prior estimation of other bidder's valuations. A dominant strategy for a buyer is to bid always a small amount higher than the current highest bid and to stop when its private value is reached. The auction continues until only one potential buyer remains, so that the highest bidder wins the item at the price of its bid. English auctions often take many rounds to complete and thus can be inefficient for the time constrained resources.
- **Dutch Auction:** The Dutch auction is a sequential auction in which the seller starts with a high price and continuously lowers the price until a sale is confirmed by the first bidder to indicate acceptance of the price. The rate of reduction is up to the seller and it has a reservation price

below which the auction is stopped. The Dutch auction also takes place in rounds like the English auction and therefore, it is not efficient for time constrained resources.

- **First-price Auction:** The First-price auction is a simultaneous bidding auction in which bidders submit sealed bids. Each bidder submits one bid without knowing about the others' bids. The buyer with the highest bid wins and pays his or her own bid price. For instance, one strategy is to bid less than the true value of the good and one might still win the auction, but it all depends on what the others bid.
- **Vickrey Auction (Second-price Auction):** The Vickrey auction [93] involves submission of simultaneous sealed bids. Buyers privately submit their own valuations. The winner is the buyer submitting the highest bid, though the price of the resource is the highest losing bid. If there is no second-highest bidder, then the price of the good is the average of the minimum selling price and the buyer's bid price.

Many-to-Many Auctions

- **Double Auction**: Double auctions are referred to as two sided auctions permitting multiple buyers and sellers to bid to exchange a designated commodity. In double auctions, buy orders (requests) and sell orders (offers) may be submitted at anytime during the trading period. Buyers and sellers of resources put their requests or offers attached with a price into the market. There are two types of double auctions namely *Continuous Double Auction (CDA)* and *periodic double auction*. CDA matches buyers and sellers immediately on detection of compatible bids. In the other words, if at any time there are open requests and offers that match or are compatible in terms of price and requirements (e.g. quantity of resources or shares), a trade is executed immediately. In CDA, the transaction prices are computed using a *discriminatory pricing policy*. In a discriminatory policy, the transaction prices are set individually for each matched buyer-seller as a function of corresponding seller and buyer prices. A periodic version of the double auction instead collects bids over a specified interval of time, then clears the market at the expiration of the bidding interval [99][56]. Periodic double auction is sometimes called *SPUD* (Sealed bid, Uniform-price,

Double auction); because the bids are not visible to other participants, the auction is to be held at intervals and has to support multiple buyers and sellers at the same time. This type of auction uses a *uniform pricing policy*. In a uniform pricing policy, all transactions are concluded at the same price determined during the auction clearing stage.

- **Proportional Share Protocol (PSP)**: This mechanism is a similar protocol to CDA, as both use a centralized scheduling algorithm. In PSP, several tasks can execute on a server at a time. The amount of resources allocated to a task depends on its bid price in relation to the sum of bid prices of all tasks executing on that server. When a task query object arrives at the marketplace, all resource offers are checked in order to find the resource which meets the task's constraints. PSP is proposed for the scheduling of tasks in computational clusters [95]. Clustering involves connecting two or more computers together to take advantage of combined computational power and resources. An application of this protocol is Libra [82] which is an economy-driven cluster scheduler.

3.2.2 Commodity Markets Mechanisms

In this economic model, different resources are treated as commodities and buyers purchase from sellers by paying the cost of the resource. Pricing scheme in commodity markets can be based on flat fee, usage duration, or demand and supply based. In the flat fee scheme, buyers will pay a fixed amount for a certain period irrespective of the resource quality. The second scheme is based on the usage duration. If the resource is used for one hour then the buyer will pay for that one hour only. In the final scheme, the prices will change dynamically based on the supply and demand of the resource.

In the demand and supply based scheme, a buyer indicates a demand by requesting the quantity of a resource and a seller indicates a supply by offering the quantity of its free resource. The market calculates a price for the resource based on the aggregated demand and supply. Then, buyers and sellers update their demand and supply based on the new price. This process continuous until an equilibrium price is reached. The equilibrium is achieved when the demand for resources is equal to the supply of resources [66]. A unique equilibrium price is guaranteed to exist by the theorem of Debreu [38].

3.2.3 Comparing Auctions and Commodity Markets

Two main classes of economic models used for Grid resource allocation are commodity markets and auctions. In the following, we discuss similarities and differences between theses two models from different aspects:

- In both formulations, buyers and sellers appeal to a trusted third party to mediate the necessary transactions. The third party is referred as *market* in a commodity market setting and is referred as *auctioneer* in auctions.
- In commodity markets, the market sets a price for a resource and then queries both buyers and sellers for a willingness to buy and sell respectively at that price. Those wishing to participate agree to transact business at the given price point and an exchange of currency for resource takes place. The market observes the unsatisfied supply or demand and uses that information (as well as other inputs) to set a new price. Price setting and transactions may occur in distinct stages, or may be concurrent or asynchronous. Alternatively in auctions, auctioneer collects resources requests and offers along with the bid and ask prices from buyers and sellers. The strategy is to grant resources to the buyers that bid the highest prices.
- In commodity markets, an attempt is made to satisfy multiple buyers and sellers at a given price. On the other hand, in auctions, usually one buyer and one seller are satisfied at a given price.
- Commodity markets treat equivalent resources as interchangeable. A buyer purchases one of those that are available from a pool of equivalent choices without the ability to specify which resource will be exactly purchased. Alternatively, in one-to-many auctions, every buyer bids for a specific resource provided by a seller. In many-to-many auctions resources are also interchangeable and buyers purchase resources from any seller that satisfies their requirements.
- In Commodity markets, the complexity in implementation is high since the market has to calculate a price based on the supply and demand functions which are the result of the aggregate behavior of all the buyers and sellers. Auctions require little or no global information, and therefore are easier to be implemented.

3.3 Conclusions

In this chapter, we studied the behavior of an economy at individual and aggregate levels. Resource allocation in ad-hoc Grids is a complex undertaking as resources are distributed, heterogeneous in nature, owned by different individuals or organizations with their own policies, have different access and cost models. Moreover, in ad-hoc Grids, buyers and sellers may arise spontaneously at any time with various requirements and availabilities. Considering these features of ad-hoc Grids, the following issues have to be considered when designing a resource allocation mechanism. A microeconomic mechanism should support the laws of supply and demand where the supply and demand in the network are neither known in advance nor predictable. To avoid the need for global information in such dynamic networks, the price can be used as a single entity that indicates the global state of supply and demand. A macroeconomic mechanism should support simultaneous participation of buyers/sellers and observe time constraints for resources and tasks. It should accommodate the variations in resources availabilities and avoid global information such as aggregate supply/demand or aggregate utility. According to the mentioned issues, we can conclude that auction mechanisms are better choice for market-based resource allocation in ad-hoc Grids. In Chapter 5, the choice of auction mechanisms is discussed in more detail.

Chapter 4

Economic Framework

There are many design choices for market-based resource allocation in Grids that use different matchmaking and pricing mechanisms such as commodity markets and auctions. In this chapter, we present an economic framework which we will use throughout the thesis to experiment with certain design choices. The subsequent chapters are devoted to show that economic mechanisms are well suited to deal with the highly dynamic nature of ad-hoc Grids. Therefore, we first need to introduce the necessary concepts and mechanisms that will be explored later in the following chapters.

Throughout the thesis, we will use the term *market* to refer to that environment where buyers and sellers meet. Nodes having a surplus in any computing resource can act as the sellers and nodes in need of finding additional resources are the buyers. This implies that any node in the ad-hoc Grid can have two different roles, namely that of a *consumer* and a *producer* of resources. Some nodes can also play another role, namely that of *matchmaker*. The matchmaker receives all transaction requests from the nodes in the Grid. A transaction is an agreement between two parties to buy/sell a certain quantity of resources. In this chapter, we present the different components that allow nodes to become either a consumer or a producer of resources and show how matchmakers are different from consumer and producer nodes. We also describe the experimental platform that was used for all our experiments.

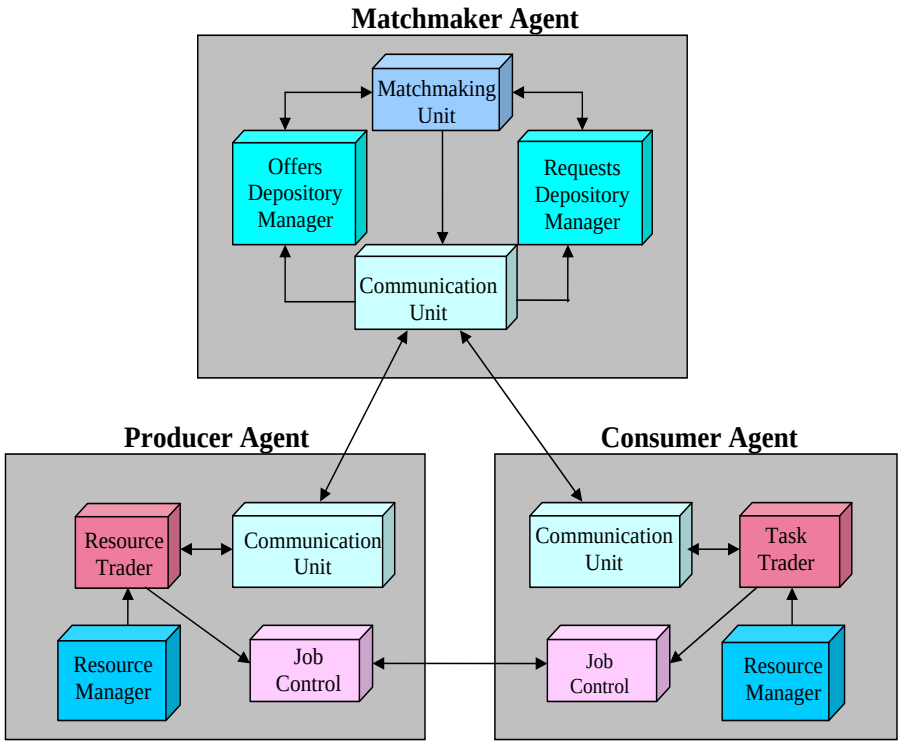

Figure 4.1: *Framework Components.*

4.1 Framework Components

The framework is composed of three types of agents as shown in Figure 4.1, namely *Consumer*, *Producer* and *Matchmaker*. The framework components are explained as follows:

4.1.1 Consumer/Producer Agent

Every node in the ad-hoc Grid can play two different roles either as a consumer or as a producer of resources. A node is a consumer whenever it requests some resources from the Grid, and it is a producer whenever it offers some resources to the Grid. In the framework, a consumer/producer agent is a representative of a consumer/producer node. Therefore, there are one con-

sumer agent and one producer agent per node. A consumer/producer agent controls the process of buying/selling resources. It estimates the execution time for tasks or availability time for resources, computes the quantity of required or available resources, and calculates the price that a consumer is willing to pay or a producer is willing to receive for a unit of resource. Based on this information, it generates and submits a request/offer.

Matchmaker Agent

The matchmaker agent is a middle agent between consumer and producer agents. It controls assigning resources to tasks in the network using a matchmaking mechanism. A matchmaking mechanism works in the following simple manner: consumers and producers announce their desire to buy or sell resources to the matchmaker. The Matchmaker keeps requests and offers in its depositories and finds matches between them. A request is matched with an offer if the offered resource meets the request requirements such as quantity, time, and price constraints.

Agent Attributes

Every agent consists of some components as depicted in Figure 4.1, some of which are common to all agents and some are specific to a particular agent. The common component in all three agents is:

- *Communication Unit*: This unit is responsible for exchanging messages between consumers, producers, and matchmaker agents.

Common attributes between consumer and producer agents are:

- *Resource Manager*: This component decides whether a node needs additional resources or it has free resources. The decision is made considering task queue and available resources of the node.
- *Job Controller*: This component is responsible for transferring tasks between consumer and producer agents. Shepherding tasks through the network, accepting, deploying, and launching tasks are controlled by this component.

The only attribute which is different when considering a consumer or a producer agent, is:

- *Task/Resource Trader*: This component decides the price in which a consumer or a producer buys or sells a unit of resource.

A matchmaker agent consists of the following components:

- *Requests/Offers Depository Manager*: This component is responsible for managing all requests/offers received from consumer/producer agents. Updating, sorting, inserting messages, and deleting expired ones are the responsibilities of this component. Requests and offers are kept in the depositories while their Time To Live (TTL) is not expired.
- *Matchmaking Unit*: This unit is the core unit of a matchmaker that finds matches between requests and offers. The matchmaker discovers the best available match based on a matchmaking mechanism. Whenever the matchmaker finds a match between a request and an offer, it informs the corresponding consumer and producer agents. An offer from a producer is matched with a request from a consumer if the following conditions are met:
  - The price bided by the consumer is higher than or equal with the price offered by the producer.
  - The producer has the quantity of the resource requested by the consumer.
  - The producer is able to provide the resource within the time specified by the consumer.

In this thesis, we do not consider task exchange between consumer and producer agents. Therefore, a job control component is not implemented. The interaction between the different components is depicted in Figure 4.2. In this collaboration diagram, the messages labeled by *a*, *b*, and *c* are independent and can be sent in parallel. Consumer and producer agents send the task or resource attributes to the *task/resource trader* component through the *resource manager* component. The *task/resource trader* component computes a price for requested/offered resource, and sends a request/offer message to the matchmaker through the *communication unit* component. Request messages are sent to the *requests depository manager* component and offer messages are sent to the *offers depository manager* component. These two components are responsible for inserting and updating messages as well as providing messages to *matchmaking unit* component. The *matchmaking unit* component finds the matches between request and offer messages, and sends the response

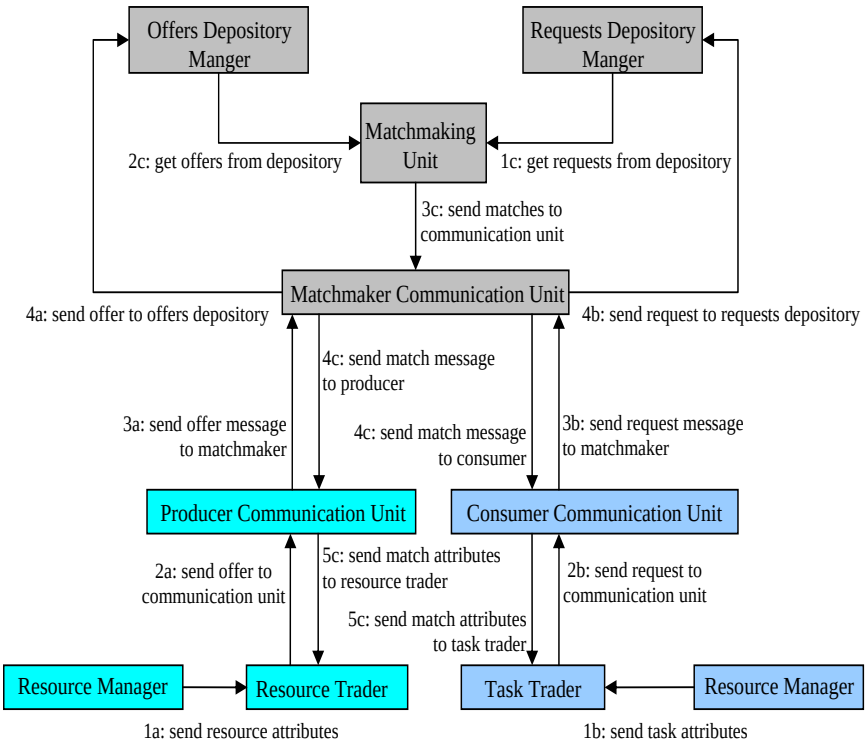

Figure 4.2: *Collaboration Diagram.*

messages back to the consumer and producer agents through the *communication unit* component.

4.2 Message Specifications

Each request/offer submitted by consumer/producer agents contains the consumer/producer identifications, resource requirements and attributes, and time and cost constraints. The messages submitted from the matchmaker to a consumer/producer agent hold the information of the matched pairs such as identification, deadline, and cost. There are four type of messages in the framework. In the following section, the message specifications are discussed.

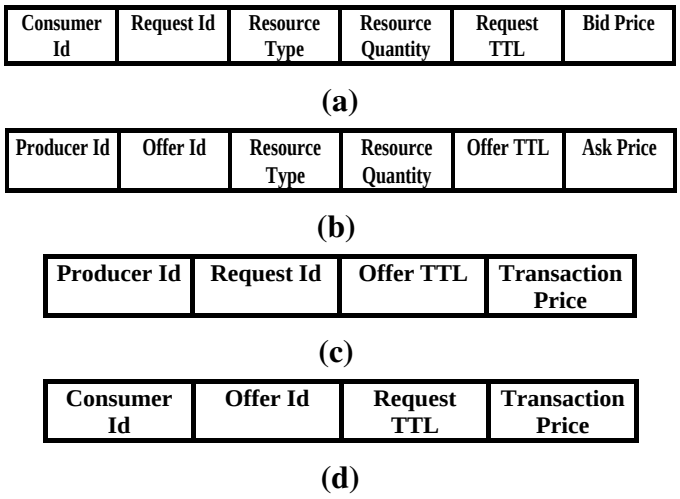

Figure 4.3: *(a)Request message (b)Offer message (c)Response message to consumer (d)Response message to producer.*

4.2.1 Request Message

This message is sent from a consumer agent to the matchmaker agent (Figure 4.3(a)).

- *Consumer Id* identifies the consumer node (e.g. IP address).
- *Request Id* is a unique number for each request to distinguish between different requests from a consumer.
- *Resource Type* is the type of resource that is required such as CPU time (cycle), memory, and storage.
- *Resource Quantity* is the quantity of the requested resource.
- *Request TTL* is the time-to-live for the request. For instance, the time in which the task has to be completed.
- *Bid Price* is the maximum price that the consumer is willing to pay for each unit of the resource.

4.2.2 Offer Message

This message is sent from a producer agent to the matchmaker agent (Figure 4.3(b)).

- *Producer Id* identifies the producer node (e.g. IP address).
- *Offer Id* is a unique number for each submitted offer to distinguish between different offers from one producer.
- *Resource Type* is the type of resource that is offered such as CPU time (cycle), memory, storage.
- *Resource Quantity* is the quantity of the offered resource.
- *Offer TTL* is the time during which the offer is valid. In other words, it is the time that the resource is available.
- *Ask Price* is the minimum price that the producer is willing to receive for each unit of the resource.

4.2.3 Response Message to Consumer

This message is sent from the matchmaker agent to a consumer agent (Figure 4.3(c)).

- *Producer Id* marks the producer from which to buy the requested resource.
- *Request Id* serves to match a response with the corresponding request that was previously submitted.
- *Offer TTL* is the remaining time during which the offered resource is available.
- *Transaction Price* is the price that the consumer has to pay to the producer for each unit of the resource.

4.2.4 Response Message to Producer

This message is sent from the matchmaker agent to a producer agent (Figure 4.3(d)).

- *Consumer Id* is used by the receiving producer to establish a contact with the consumer to sell its resource.
- *Offer Id* serves to establish correspondence between the offer and a matching request.
- *Request TTL* is the remaining time by which the request has to be executed. For instance, it can be the remaining time to complete a task.
- *Transaction Price* is the price that the producer receives for each unit of the resource.

4.3 Experimental Platform

The framework allows us to study different pricing and matchmaking mechanisms. These mechanisms have to be evaluated experimentally to investigate their efficiency. Evaluation of any proposed mechanism can be empirical or analytical. For complex systems like Grids, a theoretical analysis is almost infeasible. Empirical evaluation can be performed by real-world experiments or by simulation. The real-world experiments guarantee that there exist all factors that impact the performance and behavior of the system. However, these models have some disadvantages such as the fact that in the real-world experiments, researchers do not have control over the system setting. Therefore, it is difficult to study the system in different settings and compare different mechanisms with each other. The simulation model that imitates the operation of a real-world system deals with the mentioned disadvantages of real-world experiments. Simulation models are more flexible in this regard and they can compress time and allow faster execution of experiments. However, they can not resemble the real-world completely and some assumptions have to be made. Over the years, modeling and simulation have emerged as important disciplines in Grid systems and many simulation platforms have been built such as Bricks [16], MicroGrid [85], GridSim [24], jCase [79], and so on. A small number of the existing simulation platforms supports market-based resource management systems. GridSim [24] and jCase [79] are two instances of these simulators. The GridSim Toolkit is used to create a resource broker that simulates Nimrod-G [15] for the design and evaluation of deadline and budget constrained scheduling algorithms with cost and time optimizations. It is also used to simulate a market-based cluster scheduling

system in a cooperative economy environment. jCase is a tool for evaluating combinatorial auctions [80]. These toolkits model the resources as they have to be defined with specific availability and capability. None of these toolkits support simulation of tasks and resources in a dynamic ad-hoc Grid.

For our purposes, we constructed a simulation platform in which consumer and producer nodes interact with each other. We chose the simulation approach because we want to study the impact of making certain design choices when all other things are equal. This ceteris paribus approach allows to isolate and understand the effect of the choices one makes in constructing a market-based resource allocation mechanism. To this end, we implemented a Java-based platform that considers the LAN as the Grid thereby assuming that multiple nodes can be mapped on a single computer in the network. We developed our test-bed using J2EE and Enterprise Java beans. J2EE was used to program the client side architecture, and Enterprise Java beans was used to program the sever side architecture. A JBOSS application server was used to implement a matchmaker. JBOSS acts as an application server which hosts the server side beans and communicates with the clients. Producers and consumers are the clients in our market place. The clients and the server communicate with each other using JMS (Java Message Service). For inspection and analysis purpose of all experimental data, we set up a database to store all the results of the simulations using MySQL.

4.3.1 Assumptions

The following simplifying assumptions are made:

- Each consumer/producer agent is directly connected to the matchmaker with the same average latency. The cost for this connection is the same for all agents.
- Tasks are atomic by nature and cannot be divided.
- Consumer/producer agents are honest. Therefore, we are not concerned about security issues.
- Every node is assigned a limited budget when joining the network. No budget is injected during a simulation period. It should be taken into account that in a node, when the consumer agent buys a resource, it spends from the node's budget, and when the producer agent sells a

resource, it increases the amount of the same budget. The scenarios of considering a limited budget or an unlimited budget (with or without budget constraint) will be studied in Chapter 6.

4.3.2 Experimental Setup

In the experimental environment, there are N nodes each having one consumer agent and one producer agent. Therefore, there exist $2 * N$ agents. Some of these agents, called consumers, have tasks to perform for which they are looking for additional resources while others called producers have resources to sell. In our experiments, N is considered a number between 40 and 60, and durations of the experiments are considered between 20 to 40 minutes. During the experiments, every node creates a number of requests and offers in a random order. A node either creates a task request and activates a consumer agent or creates a resource offer and activates a producer agent depending on its workload or its idle resources. For each request or offer, a message is sent to the matchmaker. On the other hand, a message is sent to a consumer/producer agent from the matchmaker whenever a match is found. We already discussed the message specifications in Section 4.2. Here, we present the types and values of different fields in the messages. The values are generated by a uniform random distribution function between a maximum and a minimum with a step value. The maximum, the minimum, and the step values for each message field are shown in Table 4.1.

Every node has a unique *IP Address* which is used to specify the corresponding consumer/producer agent in that node. CPU time is considered as the resource in our experiments. For simplicity, we consider a reference 1.4GHz CPU based on that each consumer/producer agent indicates the quantity of its required/offered resource. This is the same as presenting resources based on CPU cycles. The experiments are performed in an environment with nodes having various CPU speeds. *Resource type* filed in an offer/request determines the CPU speed. Consumer/producer agents assign a unique number as *Id number* to every request/offer when generating them. To distinguish between requests and offers, requests are assigned positive Id numbers whereas offers are assigned negative Id numbers. For a request, *resource quantity* is indicated in terms of CPU time required to execute a task on the reference CPU. *Request TTL* is the time by which task has to be executed after submitting a

	Type	Min.	Max.	Step
Consumer/Producer Id	IP Address	-	-	-
Request Id	Id number	1	-	1
Offer Id	Id number	-	-1	-1
Resource Type	CPU speed	700 MHz	4 GHz	100 MHz
Res. Quantity (Request)	Task execution time	1000	4000	100
Res. Quantity (Offer)	Available CPU time	10000	15000	100
Request TTL	Request deadline	6000	10000	100
Offer TTL	Available CPU time	10000	15000	100
Budget	Grid$	-	-	-
Bid/Ask Price	Grid$	-	-	-

Table 4.1: *Experimental Setup.*

request. For an offer, *resource quantity* and *offer TTL* are the same and they represent the time during which the CPU is available. In case of considering budget, all nodes in the network are assigned a limited and equal amount of *budget*, for instance 240000 Grid$. *Bid and ask prices* and their maximum and minimum values are determined by the pricing mechanism adopted by consumers and producers.

4.3.3 Network Conditions

We consider three different network conditions under which experiments are performed. These conditions are defined based on the distribution of tasks and resources generated in the network. These conditions are:

- *Balanced Network (BN)*: This is the type of a network where there are more or less an equal number of tasks and resources. In this condition, tasks and resources are generated with a 50%-50% probability.
- *Task Intensive Network (TIN)*: This is a network where there are more tasks than resources. The probability of task and resource generation in such condition is 80%-20%.
- *Resource Intensive Network (RIN)*: This is a network where there are more resources than tasks. Tasks and resources are generated with a 20%-80% probability.

4.4 Conclusions

In this chapter, we introduced an economic framework by describing the framework components and interaction between them. The proposed framework allows us to study different matchmaking and pricing mechanisms. The evaluation of the mechanisms is considered to be performed through simulation. To this purpose, we presented an experimental platform. We also highlighted the simplifying assumptions we have made for executing the empirical validation. The following chapters study the performance of different matchmaking and pricing mechanisms within the framework under different network conditions.

Chapter 5

Matchmaking Mechanisms

This chapter is devoted to the study of the basic mechanisms that can be used for matchmaking between consumer and producer agents in an ad-hoc Grid. As explained earlier, we are looking at market-based mechanisms where nodes in the Grid can find the resources they want to sell or buy. This way, we want to exploit the *Invisible Hand* principle which states that through local, individual (and possibly selfish) behavior, the system interest is also maximally taken into account.

We first present the different possible market-based mechanisms that we will study, and we define a set of criteria to evaluate these mechanisms. On the basis of the criteria, we experimentally compare different market mechanisms given different workloads. This study helps us to assess the usefulness of any particular mechanism under different network conditions from a Grid user/owner perspective. We also compare a market-based mechanism with a non-market mechanism in which no price is involved. We investigate the impact of using a market-based mechanism on the performance and also on the fairness of access to tasks/resources in an ad-hoc Grid in compared to a non-market mechanism.

5.1 Market-based Matchmaking

For matchmaking in an ad-hoc Grid, we need a mechanism that supports simultaneous participation of producers/consumers, observes resource/request deadlines, and can accommodate variations in resource availability. As discussed in Chapter 3, two main types of the economic mechanisms are commodity markets and auctions. In commodity markets, resource prices are decided from the overall demand and supply in the network. The market gathers all the supply and demand information about the commodities from consumers and producers to generate a set of equilibrium prices. This type of market attempts to satisfy consumers and producers in a given price (equilibrium price) within a given time frame. In commodity markets, a single price (equilibrium price) is supposed to take into account all the preferences from the individual nodes. However, even though the concept is very appealing, they assume perfect information and are essentially static in nature. Given the highly dynamic nature of the ad-hoc Grids, this approach is not chosen. Moreover, the complexity of implementing a centralized market which relies on the aggregate supply and demand becomes very high in a dynamic ad-hoc Grid.

The competitive and decentralized nature of auction models make them an appropriate choice for resource allocation in ad-hoc Grids. Examples of such auction models are Dutch auction, English auction, First-price auction, Vickrey auction, and Continuous Double Auction (CDA) which we already studied in Chapter 3. English auction and Dutch auction are sequential and are based on open-cry. In open-cry auctions, each bid has to be broadcasted to all participants. This becomes a considerable communication overhead in the context of ad-hoc Grids. Moreover, these types of auctions are sequential. Therefore, they are not able to observe time deadlines, and they do not support simultaneous participation of producers and consumers. Hence, English auction and Dutch auction are not suitable for resource allocation in ad-hoc Grids.

First-price auction, Vickrey auction, and CDA are simultaneous and close bid auctions. We choose and implement these auctions as the matchmaking mechanisms in the framework. To support simultaneous participation of both consumer and producer agents in an ad-hoc Grid, these mechanisms have to be implemented in a many-to-many method where both consumer and producer agents can initiate an auction. In an auction mechanism, the main com-

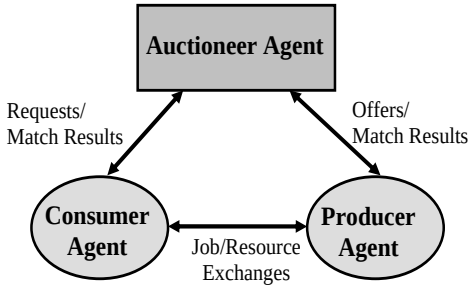

Figure 5.1: *Main components in an auction mechanism.*

ponents are consumers, producers, and a third party that acts as a mediator between consumers and producers called auctioneer. Figure 5.1 shows these components in an abstract form of the framework. The auctioneer plays the role of the matchmaker. In the following, we explain how the three auction mechanisms are implemented:

- **First-price Auction:** In this mechanism, producers submit their offers for a resource along with a minimum price below which they do not sell their resources. Consumers submit their requests together with a bid price. Consumer agents decide their bid prices using a dynamic pricing mechanism (to be introduced in Chapters 6 and 7) and producer agents have a fixed minimum ask price. The auctioneer sets up a First-price auction for each received offer. The requests are kept sorted from the highest price to lowest price. The request with the highest bid price wins the auction if the bid price is bigger than the producer's minimum price and offered resource satisfies task's constraints (such as resource quantity, price, and time frame). In this mechanism, transaction price in which a trade is executed, is equal to the bid price.
- **Vickrey Auction (Second-price Auction):** This mechanism is similar to the First-price auction except that the transaction price is the second highest bid price. In fact, the winner pays bid price of the first looser. If there is no second highest bidder then the transaction price will be the average of bid and ask prices. Similar to the First-price auction, consumer agents use a dynamic pricing mechanism (to be introduced in Chapters 6 and 7) to define their bid prices and producer agents have a fixed minimum ask price.

- **Continuous Double Auction (CDA):** In this mechanism, consumer and producer agents announce their desire to buy or sell resources to the auctioneer along with the bid and ask prices. Requests are sorted from the highest price to the lowest price and offers are sorted from the lowest price to the highest price by the auctioneer. The auctioneer finds matches between consumers and producers by matching requests (starting with highest price and moving down) with offers (starting with lowest price and moving up). When a task request arrives at the market place, the mechanism searches all available resource offers and returns the best match which satisfies the task's constraints. When a resource becomes available and several tasks are waiting, the one with the highest bid price is processed first. The transaction price is calculated as the average of ask price and bid price. In CDA, both consumer and producer agents decide their bid/ask prices using a dynamic pricing mechanism (to be introduced in Chapters 6 and 7).

For all the auction mechanisms, we assumed that nodes are given an equal amount of budget when joining the network. Therefore, consumer agents can not bid beyond their limited budget.

5.2 Performance Evaluation

In this section, we study the performance of the three auction mechanisms in three types of network conditions namely balanced, task intensive, and resource intensive. First, we define the evaluation criteria which is used to compare the different auctions.

5.2.1 Criteria to Evaluate Matchmaking Mechanisms

To evaluate the performance of different auction mechanisms, we consider the criteria that represent Grid owner and user objectives. We study the overall performance of the network in each mechanism by considering the individual performances using the following criteria:

- Throughput: Throughput is defined in terms of task and resource utilizations. Task utilization is the ratio of allocated tasks to all submitted

resource requests, and resource utilization is the ratio of allocated resources to all submitted resource offers in a network.

$$TaskUtilization = \frac{P}{N_r} \tag{5.1}$$

$$ResourceUtilization = \frac{S}{N_o} \tag{5.2}$$

where N_r and N_o are respectively the total number of submitted requests and offers. The number of purchased and sold resources are shown by P and S respectively.

- Consumer/producer deadline satisfaction: Consumer/producer deadline satisfaction is a measure of satisfaction for consumers/producers regarding the time by which they receive a response after submitting a request/offer. It shows how fast consumer/producer agents find the matches within their deadline. The deadline for consumer/producer agents is the TTL (Time To Live) in which a request or offer is valid. We define the deadline satisfaction for a consumer/producer agent as follows:

$$DedLinSaf = \sum_{i=0}^{N}((TTL_i - ResponseTime_i)/TTLi)/N \tag{5.3}$$

where $TTLi$ is the deadline for i^{th} submitted request/offer. N is the total number of submitted requests/offers at the consumer/producer agent. $ResponseTime$ is the time taken to receive a response from the matchmaker after submitting a request/offer. For unmatched request/offers, the level of satisfaction is zero. Because in this case the response time is equal to or greater than the TTL. In the ideal case, the highest deadline satisfaction is obtained when response time is zero. The average consumer/producer deadline satisfaction in a network is calculated by the averaging the values of deadline satisfactions in the individual agents in the network.

- Consumer surplus: Consumer surplus represents the amount that a consumer benefits by buying a resource at the price lower than the consumer bid price. Consumer surplus is calculated as the difference be-

tween the price that consumers are willing to pay (bid price) and the actual price (transaction price):

$$ConsumerSurplus = p_b - tp \tag{5.4}$$

where p_b is the bid price of the consumer agent and tp is the price in which the deal is made.

- Producer surplus: Producer surplus is the amount that a producer benefits by selling a resource at the price higher than the price a producer is willing to sell. The producer surplus is calculated as the difference between transaction price and ask price.

$$ProducerSurplus = tp - p_a \tag{5.5}$$

where p_a is the ask price of the producer agent and tp is the price in which the deal is made.

- Price volatility: Price volatility is a measure of uncertainty in a market. This metric is inspired by the need of economic agents to be able to understand the future evolution of a price (of any good) in order to make a rational decision about whether or not to make an offer. Especially in financial markets, such a measure is important as the expected gain depends on the future price. Even though one could envision that at some point the nodes in our network would adopt a similar, speculative behavior and start anticipating future price evolutions for resources, we assume that in our current setting, we prefer stable and more predictable price evolutions rather than highly uncertain ones. What we aim for is to allow nodes to find the resources/tasks they need and not financial gain in itself - which would be the same as maximizing the budget. In the economic literature, this uncertainty or price volatility is measured through the standard deviation of transaction prices which is given in Equation 5.6:

$$PriceVolatility = \sqrt{\sum_{i=1}^{N}(mean - tp_i)^2/N} \tag{5.6}$$

where $mean$ is the average of transaction prices in the network which

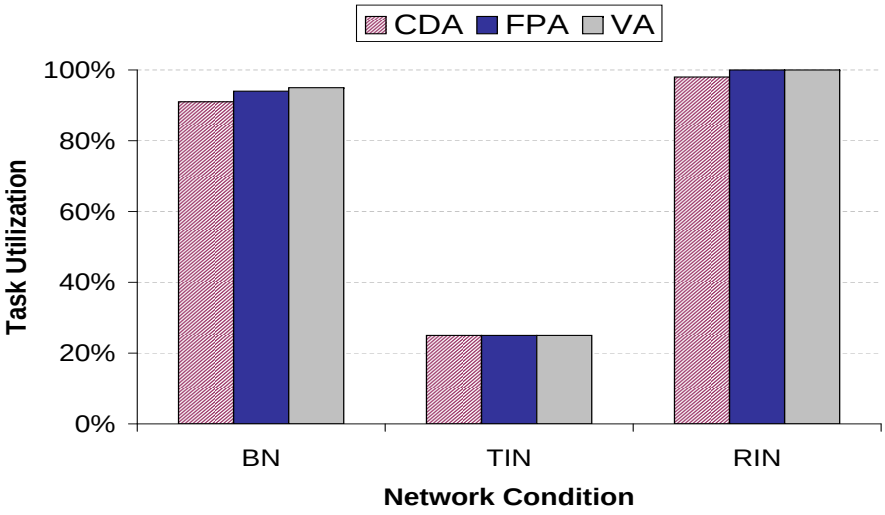

Figure 5.2: *Task utilization for three auction mechanisms in different network conditions.*

is defined as:

$$mean = \sum_{i=1}^{N} tp_i/N \tag{5.7}$$

N is the total number of transactions in the network and tp_i is the i^{th} transaction price.

5.3 Experimental Results

The experiments are performed in a local ad-hoc Grid with 60 nodes given different network conditions. The following abbreviations will be used: we refer to a balanced network condition as BN, a resource intensive network condition as RIN, and a task intensive network condition as TIN. We also use the abbreviation terms CDA and FPA and VA respectively for Continuous Double Auction, First-price auction, and Vickrey auction.

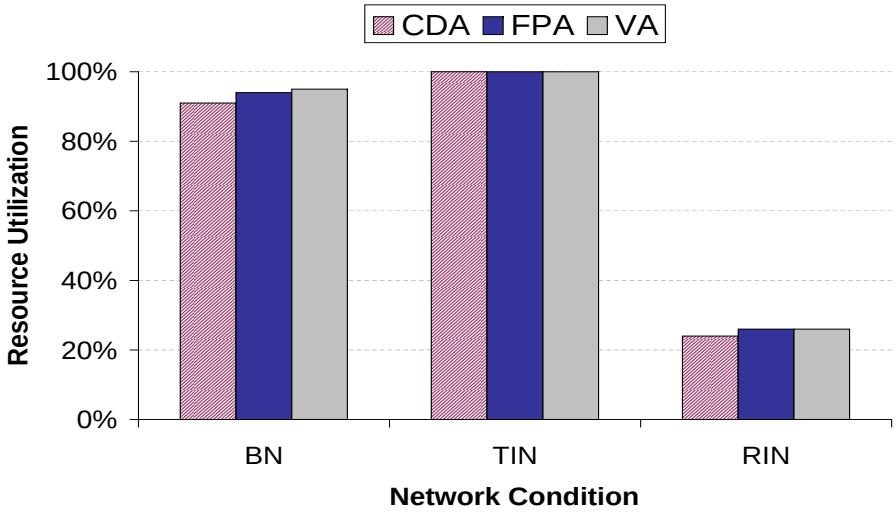

Figure 5.3: *Resource utilization for three auction mechanisms in different network conditions.*

Throughput

Throughput of the network is measured in terms of task and resource utilizations. We measure task and resource utilizations for each auction mechanism in different network conditions. The results are shown in Figures 5.2 and 5.3 in the form of percentages. We can observe that the three auction mechanisms show more or less the same level of task/resource utilization in different conditions. In the balanced condition, the Vickrey and First-price auctions show around 3% to 4% more throughput over the CDA. This can perhaps be the side effect of the simulation. Based on these observations, we can conclude that three auction mechanisms are interchangeable concerning the overall throughput of the network in any network condition.

Consumer/Producer Deadline Satisfaction

Figures 5.4 and 5.5 depict the average deadline satisfaction respectively for consumer and producer agents in different network conditions in the form of percentages. The results show that the level of satisfaction for consumer agents is more or less equal in the three auction mechanisms at any network

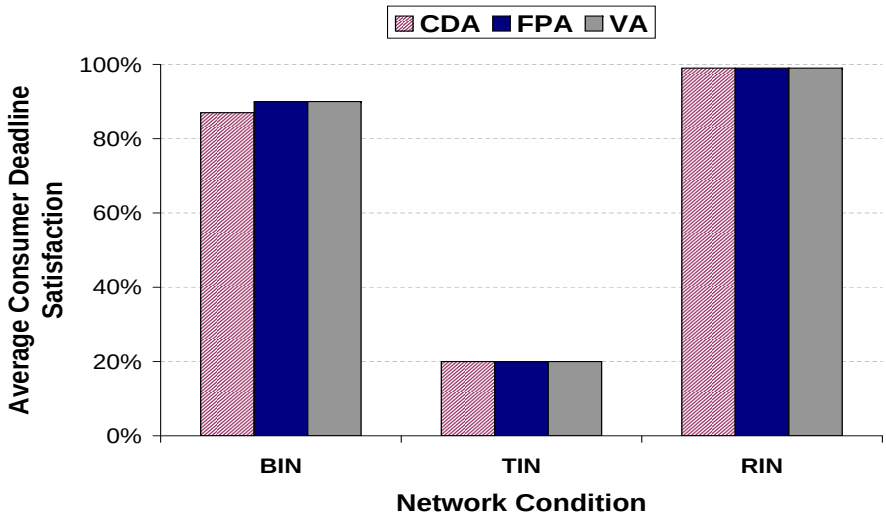

Figure 5.4: *Consumer deadline satisfaction for three auction mechanisms in different network conditions.*

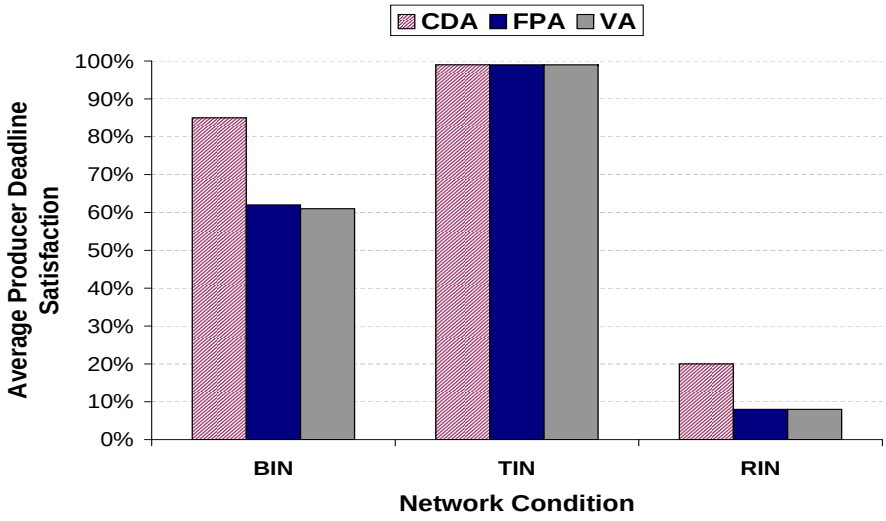

Figure 5.5: *Producer deadline satisfaction for three auction mechanisms in different network conditions.*

condition. This observation is due to the fact that the auctioneer responds to consumers in the same way in the three auction mechanisms, and that is based on giving the highest priority to the request with the highest bid price. For producer agents, the level of deadline satisfaction is higher in CDA compared to First-price auction and Vickrey auction in the balanced and resource intensive conditions. This implies that producer agents find their matches faster during the lifetime of their offer (TTL) when CDA mechanism is used compared to when First-price auction or Vickrey auction is used. This is due to the way that offers are processed in different auction mechanisms. In CDA, an offer with the lowest ask price has the highest priority. In First-price auction or Vickrey auction, offers are processed based on the time they are received by the auctioneer. The first one to arrive has the highest priority to be processed. By giving priority to the offers with lower ask prices in CDA, the offers which have more chance in finding a match, are processed first. Therefore, the speed of finding matches is increased and in consequence there will be shorter response time and higher deadline satisfaction for producers. In a task intensive condition, demand is high and producers find their matches very quickly in this condition. Therefore, the consumer deadline satisfaction has a high value (around 99%) in all the three auction mechanisms.

From these observations, we can conclude that CDA is the best choice for producers regarding deadline satisfaction in any network condition. From a consumer point of view, the three mechanisms are interchangeable.

Consumer Surplus

The average consumer surplus is calculated for matched consumers in the three mechanisms within different network conditions (Figure 5.6). The matched consumers are the consumers which have found matches for their requests. As expected, the consumer surplus for First-price auction is equal to zero. Because in this auction, the transaction price is equal to the consumer price (bid price). In CDA and Vickrey auction, consumers pay less than what they bid. Therefore, these auctions favor the consumers. Figure 5.6 shows that consumer surplus is higher in CDA compared to Vickrey auction. Higher consumer surplus in CDA is because of the higher difference between bid prices and transaction prices in CDA compared to the other auction mechanisms. In CDA and Vickrey auction, consumer surplus has its highest value

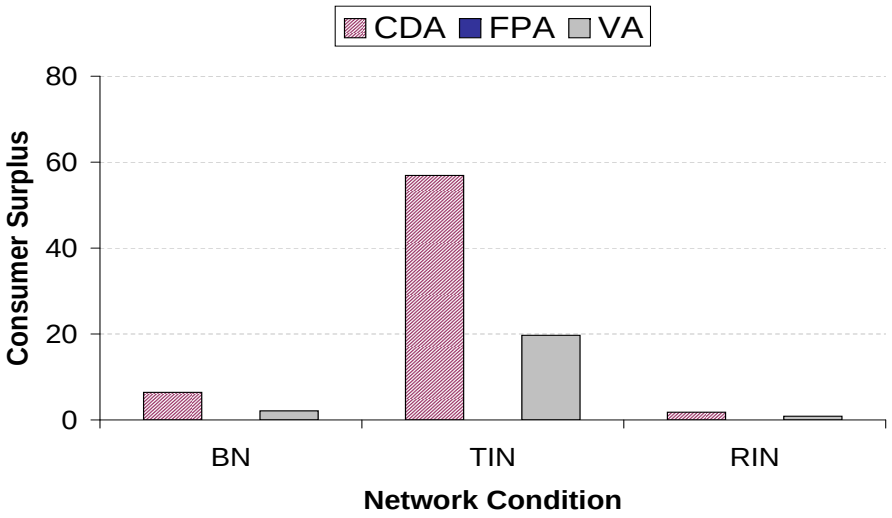

Figure 5.6: *Consumer surplus for three auction mechanisms in different network conditions.*

in the task intensive condition and its lowest value in the resource intensive condition. This observation is explained as follows. In the task intensive condition, consumers increase their bid prices to outbid their competitors and in consequence the difference between transaction prices and bid prices increases. On the other hand, due to the abundance of resources in a resource intensive condition, consumers bid the low prices which are close to the ask prices.

By comparing the three auction mechanisms regarding consumer surplus, we can observe that FPA provides no surplus for consumers. Between VA and CDA mechanisms, CDA provides a higher surplus for consumers in a task intensive condition compared to VA. In the balanced and resource intensive conditions, CDA and VA are interchangeable when consumer surplus is concerned.

Producer Surplus

We compute the average producer surplus for matched producers in the three mechanisms within different network conditions. The matched producers are the producers which have found matches for their offers. The results depicted

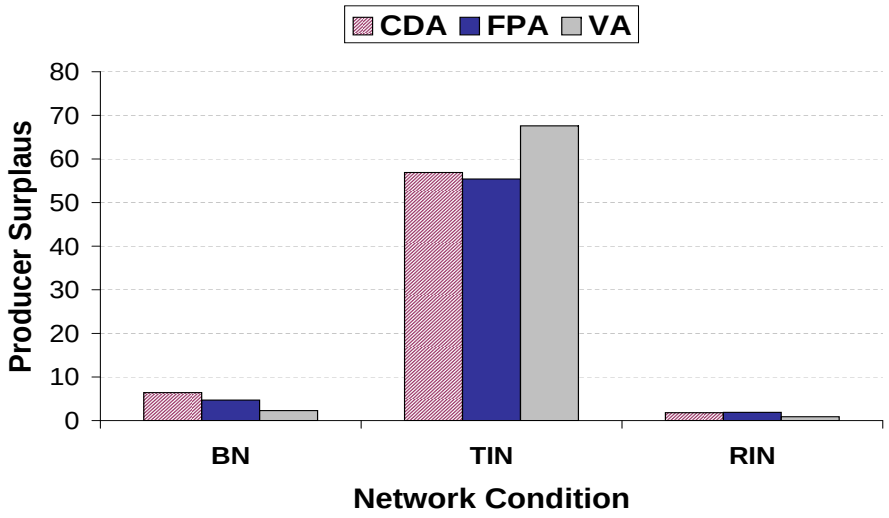

Figure 5.7: *Producer surplus for three auction mechanisms in different network conditions.*

in Figure 5.7 show that the Vickrey auction provides a higher producer surplus as compared to the other two auction mechanisms in the task intensive condition. In the resource intensive and balanced network conditions, producer surplus is approximately equivalent in the three auction mechanisms. The higher producer surplus in the case of a task intensive condition and lower producer surplus values in the other two network conditions can be explained with the same reasoning as we already discussed for consumer surplus.

As consumers pay the first highest price in the First-price auction and pay the second highest price in the Vickrey auction, it can be expected that producer surplus should be higher in the First-price auction as compared to the Vickrey auction. This interpretation can be justified in the following way. Consumers pay higher prices in the First-price auction. Therefore, they may run out of their budget more quickly compared to the Vickrey auction in a task intensive condition.

We conclude that when the producer surplus is the selection criterion, the three auction mechanisms are interchangeable in the balanced and resource intensive conditions. Under a task intensive conditions, Vickrey auction is the best option.

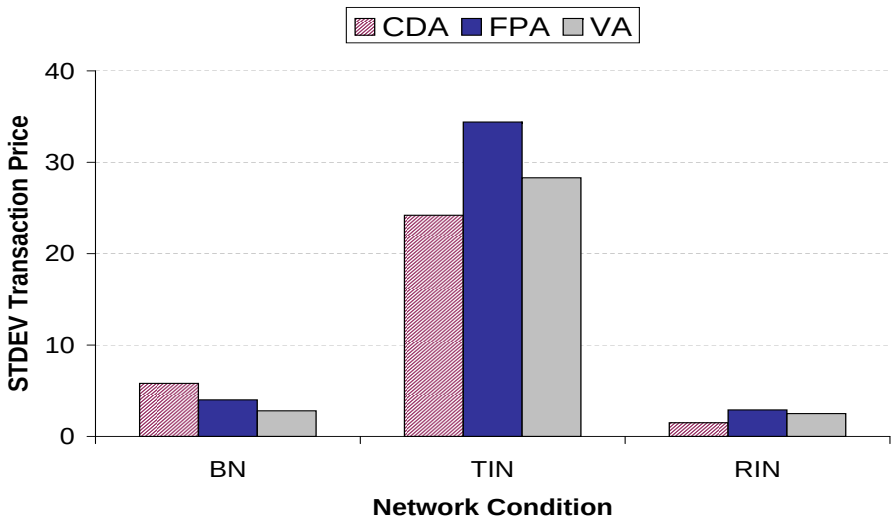

Figure 5.8: *Price volatility for three auction mechanisms in different network conditions.*

Price Volatility

We consider price volatility in our analysis as it is a measure of uncertainty with respect to the likelihood to conclude a transaction in the market. The price volatility is measured by the standard deviation of transaction prices. Figure 5.8 depicts the price volatility in the three auction mechanisms under different network conditions. From Figure 5.8, we can make the following observations. The volatility is highest in the task intensive case which is explainable by the fact that nodes will have to make more aggressive bids in order to obtain the resources they need. It is lowest in the resource intensive case where automatically the price for a resource is low. For both resource and task intensive conditions, CDA always has the lowest volatility. The only exception is the balanced case where CDA has a higher volatility compared to FPA and VA. This can be explained as follows. In CDA, both consumer and producer prices are dynamic (they are defined through a dynamic pricing mechanism) while in FPA and VA, consumer prices are dynamic but producer prices are static (every producer has a fixed minimum price). Therefore, we observe slightly higher variation in transaction prices when CDA is used in a balanced condition where there is no competition between agents.

	Throughput	Consumer Surplus	Producer Surplus	Price volatility	Con. Deadline Satisfaction	Pro. Deadline Satisfaction
BN	Interchangeable	CDA, VA	Interchangeable	VA	Interchangeable	CDA
TIN	Interchangeable	CDA	VA	CDA	Interchangeable	Interchangeable
RIN	Interchangeable	CDA, VA	Interchangeable	CDA	Interchangeable	CDA

Table 5.1: *Design choices.*

In the task and resource intensive conditions, CDA shows the lowest price volatility despite the dynamic bid and ask prices in this mechanism. We can conclude that CDA mechanism is a better choice when resources/tasks are scarce since a node can be more certain about the acceptance of its price and in consequence finding a task or a resource. In a balanced condition, VA can perform slightly better in this concern.

5.4 Results Discussion

One of the main objectives of this research is to assist Grid users/owners to decide which auction mechanism is most suitable for a particular network condition. We consider different Grid and user objectives namely throughput, deadline satisfaction, consumer surplus, producer surplus, and price volatility to evaluate various auction mechanisms.

The results show that the network throughput in terms of task and resource utilizations is approximately the same in the three auction mechanisms. As a result, these mechanisms are interchangeable in this regard. Considering deadline satisfaction, CDA favors producers with providing lower response time in any network condition compared to the other two mechanisms. Any mechanism can be chosen considering consumer deadline satisfaction.

CDA increases the economic benefit of the Grid consumers in a task intensive condition. On the other hand, Vickrey auction favors producers in the task intensive condition as it gives higher producer surplus. In the balanced and resource intensive conditions, CDA and Vickrey auction are interchangeable regarding consumer surplus. Regarding producer surplus, any mechanism can be selected in the balanced and resource intensive conditions.

We also observe that VA can be chosen in a balanced condition when the price

volatility is concerned. In the task and resource intensive conditions, CDA presents the lowest price volatility as compared to the other two mechanisms. Therefore, CDA mechanism is the best choice in this concern since a node can be more certain about the acceptance of its price and in consequence finding a task or a resource when they are scarce. Table 5.1 summarizes the choices of auction mechanisms in the three network conditions.

5.5 A Non-market Matchmaking Mechanism

The economic framework presented in this thesis can be also used for implementing non-market mechanisms. In this section, we study the performance of a non-market mechanism for resource allocation in ad-hoc Grids. Among different non-market mechanisms, we choose First-Come-First-Served (FCFS) mechanism as it has shown a good performance compared to the matchmaking mechanisms such as STF (Shorter Tasks First) and Min Distance [22]. FCFS is a non-market mechanism in which there is no price constraint for matchmaking between requests and offers. In this mechanism, the matchmaking is accomplished on the basis of the time of arrival and no sorting or any other manipulations are done to find the matches.

We compare the performance of FCFS mechanism with CDA mechanism in three network conditions namely balanced, task intensive, and resource intensive. Comparison is done based on the criteria task/resource utilization, deadline satisfaction, and also two new criteria namely load balancing and resource balancing. We introduce the new criteria since in FCFS no price is involved, and we can not evaluate it by price-based or budget-based criteria. On the other hand, the fairness of market-based mechanisms in providing access to resources and tasks is one of the claims we made when discussing the benefits of these mechanisms. To prove this claim, we measure load/resource balancing. Load and resource balancing assess the distribution of tasks among producers of resources and the distribution of resources among consumers of resources respectively.

	Average Task/Resource Util.	Resource Balancing	Load Balancing	Average cons. Deadline Satisf.	Average prod. Deadline Satisf.
CDA	92%	96%	98%	90%	81%
FCFS	95%	98%	98%	87%	65%

Table 5.2: *Comparing FCFS and CDA in a balanced network.*

Load/resource Balancing

Load balancing shows how evenly tasks are distributed among producers of resources. This can be detected by measuring the variation in resource utilizations among individual producers. The lower this variation, the higher load balancing is. On the other hand, resource balancing is a measure of fairness for access to resource by consumers. Variation in task utilizations among individual consumers shows the level of resource balancing. We use Relative Standard Deviation (RSD) to measure the variation in task/resource utilizations. The percentage of RSD is defined as:

$$RSD = (stdev/mean) * 100\% \tag{5.8}$$

where $mean$ is the average task/resource utilization in the network and $stdev$ is the standard deviation of task/resource utilization among different consumer/producer agents. Now, we define the level of resource/load balancing using RSD as follows [29]:

$$\eta = (100\% - RSD) \tag{5.9}$$

The most effective resource/load balancing is achieved when $\eta = 100\%$ that is obtained when $stdev = 0$.

In the following, we study the performance of FCFS in different network conditions and compare it with CDA.

5.5.1 Balanced Network Condition

Table 5.2 presents the results of comparing FCFS and CDA in a balanced condition. The results show that the market-based mechanism is as efficient as a blind and simple FCFS mechanism in terms of task/resource utilization. Regarding load/resource balancing both mechanisms present more or less the

	Average Task Util.	Average Res. Util.	Resource Balancing	Load Balancing	Average cons. Deadline Satisf.	Average prod. Deadline Satisf.
CDA	24%	95%	82%	94%	19%	94%
FCFS	25%	100%	58%	100%	6%	99%

Table 5.3: *Comparing FCFS and CDA in a task intensive network.*

	Average Task Util.	Average Res. Util.	Resource Balancing	Load Balancing	Average cons. Deadline Satisf.	Average prod. Deadline Satisf.
CDA	99%	26%	96%	81%	99%	18%
FCFS	100%	26%	100%	62%	99%	20%

Table 5.4: *Comparing FCFS and CDA in a resource intensive network.*

same performance in a balanced condition. Therefore, both mechanisms provide a fair access to resources and tasks in a balanced condition. The average deadline satisfaction for consumers and producers is higher in CDA mechanism compared to FCFS mechanism.

5.5.2 Task/Resource Intensive Network Condition

Table 5.3 presents results of comparing the two mechanisms in a task intensive network condition. The results show when resources are scarce, both mechanisms give the same throughput regarding the task and resource utilizations. Considering load balancing, both mechanisms show a high level of load balancing. Because in a task intensive condition that resources are scarce, tasks are distributed evenly among the few resources. While considering resource balancing, CDA mechanism shows a higher level of resource balancing compared to FCFS in a task intensive condition. This implies that CDA provides a fair access to resources for consumer agents when the resources are scarce. Average consumer deadline satisfaction is higher in CDA compared to FCFS mechanism. For producers, average deadline satisfaction is high in both mechanisms because of abundant tasks in this condition.

Results of comparison of two mechanisms in a resource intensive condition are shown in Table 5.4. In this condition, the two mechanisms perform more or less similar regarding all criteria expect for load balancing. CDA mechanism shows higher level of load balancing compared to FCFS mechanism.

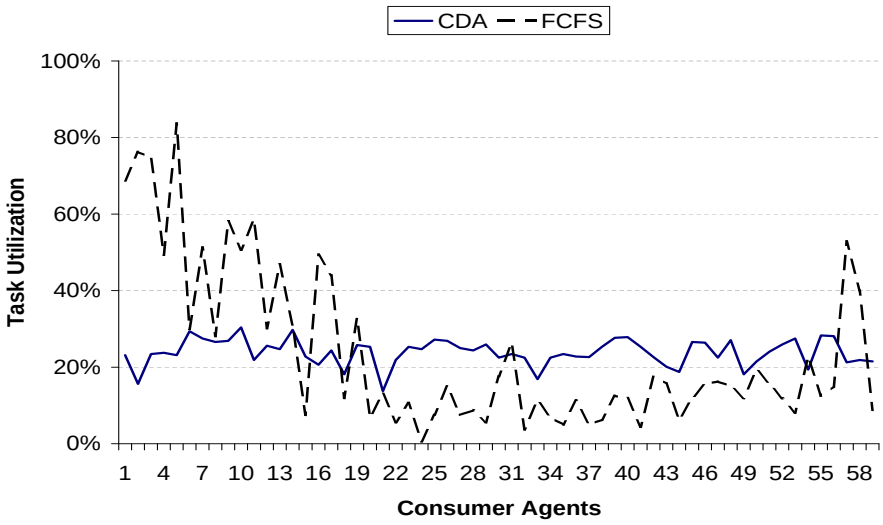

Figure 5.9: *Task utilization at individual consumer agents under a task intensive condition in FCFS and CDA mechanisms.*

This shows that a CDA mechanism distributes tasks among producer agents more evenly compared to a FCFS mechanism when tasks are scarce.

Furthermore, we also measure task utilization in individual consumer agents to explain the resource balancing in a task intensive condition. Figure 5.9 depicts the task utilizations in 60 consumer agents. As it can be seen from the figure, the task utilization fluctuates much less in CDA mechanism compared to FCFS mechanism. It is observed that in FCFS, some consumer agents receive a high level of task utilization while some others receive a very low utilization. In CDA mechanism, the task utilization is spread almost equally between all agents. This implies that access to the Grid resources is fairly distributed among all consumer agents when CDA is used.

5.6 Conclusions

The market-based framework enabled us to analyze different auction mechanisms (First-price auction, Vickrey auction, and CDA) as the matchmaking mechanisms. We considered the assessment criteria based on system as well as user preferences such as throughput, deadline satisfaction, consumer sur-

plus, producer surplus, and price volatility. These criteria were measured under three different network conditions. Based on our experimental results, different design choices were discussed. According to the results presented in Table 5.1 and the discussion in Section 5.4, we conclude that CDA seems to be the best choice given the different proposed evaluation criteria.

We also compared CDA mechanism with a non-market FCFS mechanism. The result showed that CDA and FCFS are similar in performance despite the FCFS has no price constraint. Nevertheless, CDA offers a number of additional advantages compared to FCFS. CDA with a high level of resource balancing provides a fair basis for access to Grid resources for the consumer agents in a task intensive condition while FCFS gives a low level of resource balancing in this condition. CDA also provides a higher level of load balancing compared to FCFS in a resource intensive condition. Load/resource balancing indicates how evenly the tasks/resources are distributed among producer/consumer agents. Therefore, we can conclude that CDA as a market-based mechanism provides a fair access to resources and tasks for consumer and producer agents in comparison to a non-market FCFS mechanism. Furthermore, a market-based mechanism allows to have the condensed global system information into one metric namely the price. In the next chapters, we will study pricing mechanisms and show how the price indicates this information.

Chapter 6

A Learning and Adaptive Pricing Mechanism

In the previous chapter, we studied the use of both market and non-market matchmaking mechanisms. As explained in that chapter, the difference between the two approaches is the use of money, or put differently, the existence of a price for a resource. As stated before, the price for a resource is a very synthetic metric summarizing a global system state aspect, the need for a particular resource. Through the use of the mechanisms explained in the previous chapter, the different nodes in the Grid contribute to this price by expressing their needs at the individual level. Moreover, money and budget are also interesting concepts for real world commercially exploited Grids where real money will have to be used.

In a dynamic ad-hoc Grid, we expect that it will be difficult to manually set prices that lead to a stable and healthy economy. Therefore, a dynamic pricing strategy that allows adapting to changes of supply and demand in the network is important. Moreover, setting prices based on gathering global information about supply and demand is not feasible in such dynamic networks. A way to learn about the supply/demand condition is through interaction with the environment. Learning from the interaction is a fundamental idea underlying nearly all theories of learning and intelligence [89]. As depicted in Figure 6.1, agents can learn the condition of the network using the reward they get from the environment for subsequently taking a proper action. For instance,

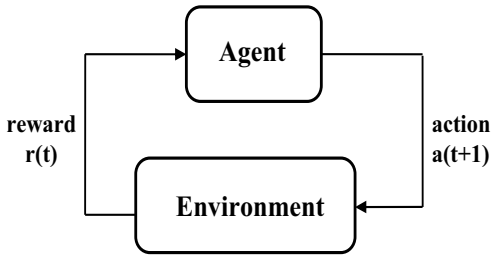

Figure 6.1: *The agent - environment interaction in a learning mechanism [89].*

the reward can be their utilization from the Grid, and the action can be setting the new prices.

In this chapter, we study how exactly nodes can make a bid, which means determining the price that nodes are willing to pay or want to receive in return for resources. We propose a dynamic and learning pricing mechanism for resource allocation in ad-hoc Grids. We name our pricing mechanism λ that stands for *Learning and Adaptive Mechanism for Bidding Agents*. This pricing mechanism is inspired from the pricing functions presented by Kuwabara et al. [60]. The essence of our λ pricing mechanism is that it takes into account how successful an agent has been in closing a transaction. When an agent is systematically bidding too low a price for resources, it will not succeed in actually finding the wanted resources. The λ pricing mechanism is constructed in such a way that the next bid will take this experience into account. A similar explanation holds for the case where a producer would systematically ask too high a price for its resources. We study the behavior of our λ pricing mechanism and its parameter regime under different network conditions. We will also introduce a budget constraint to avoid exuberant prices being asked or paid for. By applying the budget constraint, we want to study the performance of our pricing mechanism when creating a scenario of real markets. In the real markets, the buyers are always restricted by their budget when bidding a price. The pricing mechanism will therefore be extended to take such a constraint into account.

6.1 Consumer/Producer Pricing Algorithm

In a market, the objective of a seller is to maximize its earning as much as possible and the objective of a buyer is to spend as little money as possible. Based on these objectives, and following the laws of supply and demand, the strategy of producers of resources is to raise the price when the demand for the associated resource is high and to lower the price when the demand is low. On the contrary, the strategy of consumers of the resources is to lower the price when the supply is high and to raise the price when the supply is low. Based on these strategies, we define consumer and producer pricing functions.

The price is defined as the value of each unit of resource in which the consumer and producer agents are willing to buy or sell. Consumer/producer agents join the network with an initial bid/ask price and dynamically update it over the time using the pricing functions presented below. Ask and bid prices are defined respectively for producers and consumers as follows:

$$p_a(t) = p_a(t-1) + \Delta p_a(t) \tag{6.1}$$

$$p_b(t) = p_b(t-1) + \Delta p_b(t) \tag{6.2}$$

Where $p(t)$ is the new price and $p(t-1)$ is the previous price. The value of Δp determines whether the price is increasing or decreasing. The agents perceive the supply and demand of the resources through their previous resource and task utilizations and set the Δp accordingly. Δp is calculated for a seller and a buyer respectively based on Equations 6.3 and 6.4 as seen below:

$$\Delta p_a(t) = \alpha(ru(t) - u_{thR})p_a(t-1) \tag{6.3}$$

$$\Delta p_b(t) = \beta(u_{thT} - tu(t))p_b(t-1) \tag{6.4}$$

so that,

$$\alpha, \beta \in [0,1] \qquad and \qquad u_{thT}, u_{thR} \in [0,1]$$

α and β are the coefficients that control the rate of changing the price. In this version of the pricing mechanism, α and β are constant values set by producer and consumer agents respectively. Producers and consumers can

speed up or slow down the rate of increasing/decreasing their ask and bid prices by modifying the values of α and β.

ru(t) and *tu*(t) are respectively the resource utilization and the task utilization at the time t. Each consumer or producer agent keeps a history of its previous experiences in buying or selling resource. Task/resource utilization represent this history. For a given agent, we define the task utilization as the ratio of allocated tasks to all submitted requests, and the resource utilization as the ratio of allocated resources to all submitted offers. The resource utilization is formally defined by Equation (6.5), and the task utilization is specified by Equation (6.6).

$$\textit{ru}(t) = \frac{S(t)}{N_o} \tag{6.5}$$

$$\textit{tu}(t) = \frac{P(t)}{N_r} \tag{6.6}$$

where $S(t)$ and $P(t)$ are respectively the numbers of sold and purchased resources at the time t. N_o and N_r show respectively the number of submitted offers and requests. N_o and N_r represent the length of the history of the pricing mechanism. For instance, if $N_r = 10$, it means that we have considered 10 recent submitted requests and then $P(t)$ is the number of purchased resources out of 10 submitted requests. Based on these definitions, ru and tu have always the values between 0 and 1.

u_{thT} and u_{thR} are the threshold values below which respectively the task utilization and the resource utilization should not go. These values are constant and set by the consumer and producer agents. u_{thT} and u_{thR} could be interpreted as the degree of an agent activity. Low activity implies that the agent is satisfied with a low usage of its resources or a low completion rate of its tasks. High activity means the agent is more demanding by imposing higher satisfaction thresholds.

In the current version of our λ pricing algorithm, we do not consider any budget constraint. It means that buyers can spend as much money as they want. Therefore, there is no upper limit for prices. Of course, the lower limit for prices is zero.

6.2 Transaction Price

The transaction price is the price in which a deal is made. We use a discriminatory pricing policy in Continuous Double Auction (CDA) mechanism to set transaction prices (see Chapter 3, Section 3.2.1). In a discriminatory pricing policy, the transaction price can be defined using the common k-double auction pricing rule [78] as follows:

$$p = k.p_a + (1 - k)p_b \tag{6.7}$$

where p_a and p_b are ask and bid prices respectively. Considering $k = 0.5$, the transaction price is the average of bid and ask prices.

6.3 Performance Evaluation

To evaluate the proposed pricing mechanism, we use CDA with the discriminatory pricing policy for matchmaking between consumers and producers. The CDA mechanism is selected based on the results of the previous chapter where we studied different matchmaking mechanisms. From the previous discussion, we observed that CDA could be the best choice for a matchmaking mechanism under different network conditions. In CDA, consumers and producers submit their prices along with the quantities of the requested or offered resources to the auctioneer. The auctioneer finds the requests and offers that are matched in terms of price, quantity, and TTL. A trade between each matched pair is made at the average of the corresponding consumer and producer prices (bid and ask prices). In this section, we study the pricing behavior under different network conditions with varying supply and demand. Three network conditions namely balanced, task intensive, and resource intensive are considered for performance evaluation. We study the parameter regime of the pricing functions and show the impact of different pricing parameters on the price values, the throughput, and the response time in different network conditions. At the end, we apply budget constraint to our pricing mechanism and study its influence.

In the experiments presented in this chapter, we have considered 50 nodes. Therefore there are 50 consumer agents and 50 producer agents in the network. Consumer and producer agents start in the ad-hoc Grid with an initial

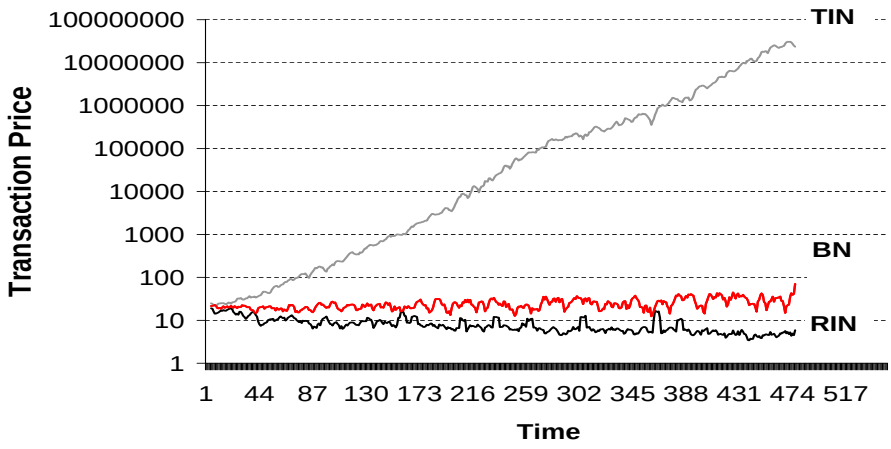

Figure 6.2: *Transaction price evolution in a Task Intensive Network (TIN), a Resource Intensive Network (RIN), and a Balanced Network (BN) with the values of* $\alpha = 0.8$ *and* $\beta = 0.8$ *(Logarithmic Y-scale).*

random price between 10 and 30. In these experiments, we assume that agents consider a history with the length equal to 10 that means the agents compute the prices considering a history of 10 previous submitted requests or offers.

6.3.1 Pricing Behavior in Different Network Conditions

The pricing behavior is studied by presenting the transaction price evolution in different network conditions. For this set of experiments, the following parameter values have been considered : $u_{thR} = u_{thT} = 0.9$ and $\alpha = \beta = 0.8$. In this section, we aim to show the pricing behavior, and the parameters values are not of our concern. Later in this chapter, we will examine the pricing parameters with different values. Figure 6.2 shows the transaction price in three network conditions. By examining the three trends in each network condition, the following observations have been made:

- Balanced Network: This condition is similar to a balanced market where supply and demand are equal. In this type of market, we do not expect to see any noticeable upward or downward trend of prices. As observed in Figure 6.2, the price indeed fluctuates around a level of 25, which is the average price chosen as the starting price for the agents.

It should be noticed that an agent's initial price is between 10 and 30.

- Task Intensive Network: This type of network is similar to what is called a *sellers market* which has more buyers than sellers. High prices in this market, are the result of exceedance of demand over supply. Buyers compete with each other in order to obtain scarce resources by increasing their bid prices. When resources are scarce, sellers are also willing to increase their ask prices to earn more. This creates an upward trend of the transaction price. It should be noticed that there is no upper limitation for the buyer price in this set of experiments.
- Resource Intensive Network: The third scenario involves a *buyers market*. In a buyers market, there are more sellers than buyers and low prices result from the exceedance of supply over demand. Similar to the sellers market, sellers compete in order to find tasks for their resources by decreasing their ask prices. Under this condition, there is hardly any competition between buyers, and buyers also decrease their bid prices to spend less money. A downward trend of the transaction price is apparent as a result.

6.3.2 Parameter Regime Analysis

The values of the pricing parameters α, β, u_{thT}, and u_{thR} can be chosen by consumer and producer agents. It is important to understand the impact of such choices on the overall system behavior. The parameters α and β are used to define the rate of changing ask and bid prices. The parameters u_{thT} and u_{thR} define critical values for task or resource utilization which determine the activity of consumer/producer agents by their satisfaction level. We study the parameter regime of the pricing mechanism in three network conditions.

α and β parameters

In this section, we study transaction prices when different values of α and β are applied. We also measure throughput of the network in terms of task and resource utilizations; and we compute the average time of finding matches for consumer/producer agents with varying values of α and β. To study α and β parameters, we consider fixed values for parameters u_{thT} and u_{thR} to avoid the impacts of changing their values on these experiments. Parameters u_{thT} and

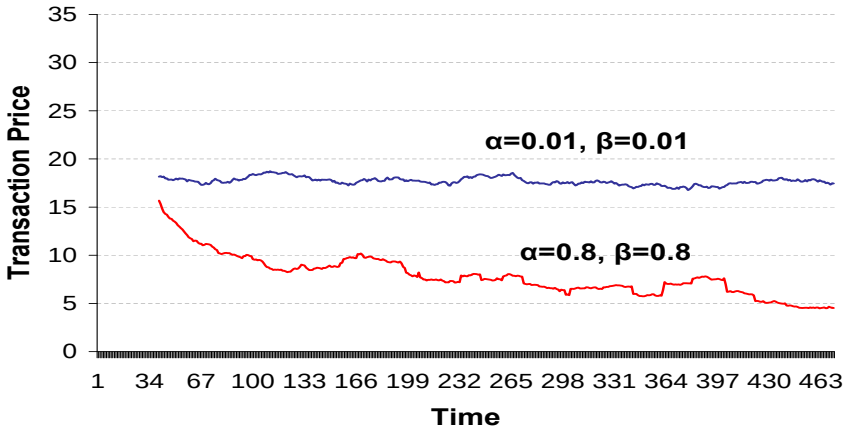

Figure 6.3: *Transaction price trends with different values for α and β in a resource intensive network.*

u_{thR} can have a value between 0 and 1. we set the values $u_{thT} = u_{thR} = 0.9$ for all agents which give them a high satisfaction threshold.

- **Transaction price:** In the previous section, we studied the transaction price trends in three network conditions. In this section, we show the impact of the parameters α and β on the transaction price. Producers increase or decrease the ask prices at a rate of α, and consumers increase or decrease the bid prices at a rate of β. According to Equations 6.3 and 6.4, the higher the value for α and β, the higher Δp is. And consequently, a higher Δp leads to higher price changes. Considering for instance a resource intensive condition, we expect a downward trend of prices. Figure 6.3 shows the transaction price trend in a resource intensive network with different values for α and β. Once, we consider $\alpha = 0.01$ and $\beta = 0.01$ and another time we consider $\alpha = 0.8$ and $\beta = 0.8$. In a resource intensive condition, higher values of α and β speed up the rate of decreasing ask and bid prices and in consequence it speeds up the rate of decreasing the transaction prices. As seen from Figure 6.3, the rate of decreasing transaction prices is higher when the values of α and β are higher.

  To study different cases, the average transaction price is measured in three network conditions with various values for α and β. As it can be seen in Figure 6.4, the lowest prices in the resource intensive condition

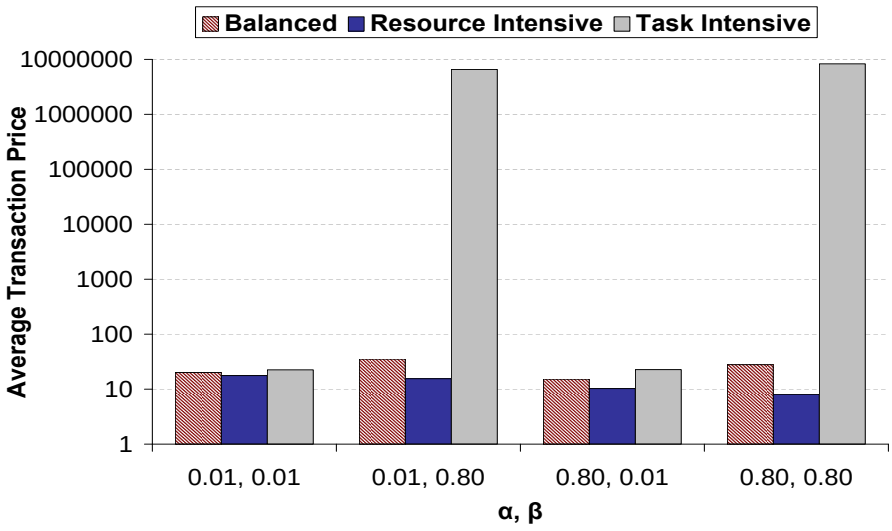

Figure 6.4: *Average transaction price with various values for α and β in different network conditions (Logarithmic Y-scale).*

and the highest prices in the task intensive condition are observed. In a balanced condition, high or low prices are not expected. The average transaction price in the task intensive condition is higher when β is higher. On the other hand, the average transaction price in the resource intensive condition is lower when α is higher. These observations are expected, as in the task intensive condition, resources are scarce and consumers increase their bid prices in a competitive way with the rate of β. Therefore, high values for β speed up the price increase and lead to higher transaction prices. The same reasoning holds for the resource intensive condition when the producers decrease their prices at a rate of α.

- **Task/Resource utilization**: Task/resource utilization is defined as the ratio of allocated tasks/resources to all submitted resource requests/offers in the network. Figures 6.5 and 6.6 show the task and resource utilizations under different network conditions with different values for α and β. Task and resource utilizations are shown in the form of percentages. As Figures 6.5 and 6.6 show, task and resource utilizations in a balanced condition for all values of α and β are around 90% except when α and β have very low values ($\alpha = 0.01$ and $\beta = 0.01$).

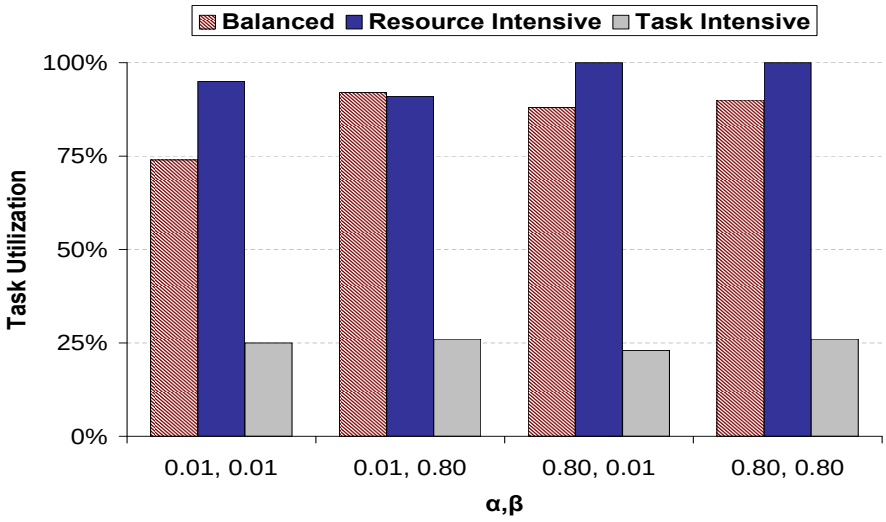

Figure 6.5: *Task utilization with various values for α and β in different network conditions.*

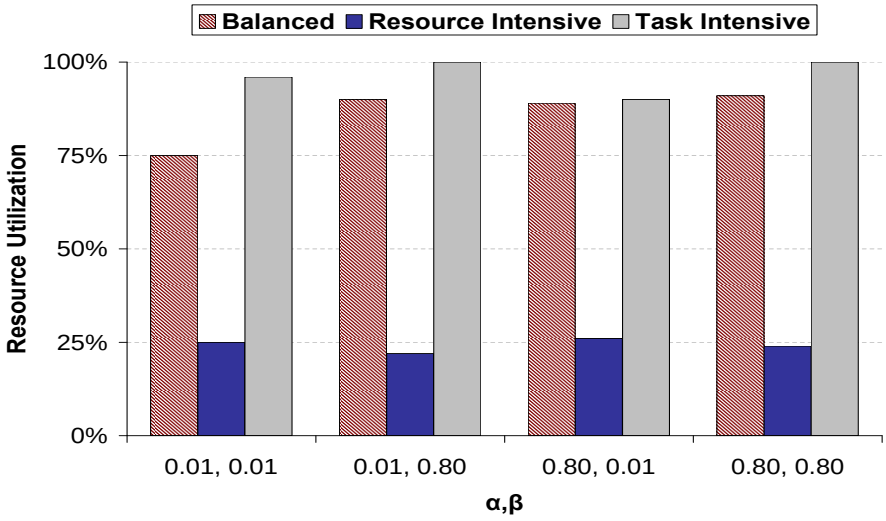

Figure 6.6: *Resource utilization with various values for α and β in different network conditions.*

In the case of $\alpha = 0.01$ and $\beta = 0.01$, task and resource utilizations are around 75% in the balanced condition. With low values for α and β, producer and consumer update their prices at a slower rate that leads to a lower utilization of tasks and resources. In a resource intensive condition, a global observation is apparent from Figures 6.5 and 6.6 as the task completion is close to 100% in most cases while only around 25% of the available resources are used. This is expected since we are now looking at an ad-hoc Grid condition where there are abundant resources. In such a condition, the highest task/resource utilization is obtained when $\alpha = 0.8$ and, $\beta = 0.01$ or $\beta = 0.8$. Since in a resource intensive condition there is a high competition between producers, decreasing ask prices at a higher rate gives the producers more chance in selling their resources. A global view on the task/resource utilization in a task intensive condition determines a utilization close to 100% for resources and 25% for allocated tasks. These results are a consequence of the higher number of tasks compared to resources. The highest task/resource utilization in this condition is obtained when $\beta = 0.8$ and, $\alpha = 0.01$ or $\alpha = 0.8$. A higher rate in increasing bid price helps competitive consumers to find more matches which leads to more task/resource utilization.

- **Average time of finding matches**: We measure the average time that it takes for consumers and producers to find their required matches after submitting their requests or offers. In CDA, auctioneer sorts requests in a decreasing and offers in an increasing price order, and finds the matches by comparing requests with available offers regarding resource quantity, time, and price. The search in offer list is stopped when reaching a ask price that is higher than the bid price.

  Figures 6.7 and 6.8 show the average time of finding matches with different values for α and β under different network conditions. In a resource intensive condition where the resources are abundant, we expect that consumers find matches for their requests in a short time. As shown in Figure 6.7, we can observe that the average time of finding matches for consumers is much higher when $\alpha = 0.01$ compared to when $\alpha = 0.8$ in a resource intensive condition. This can be explained as follows. When α is low in a resource intensive condition, the producers reduce their prices in a low rate and in consequence it takes more

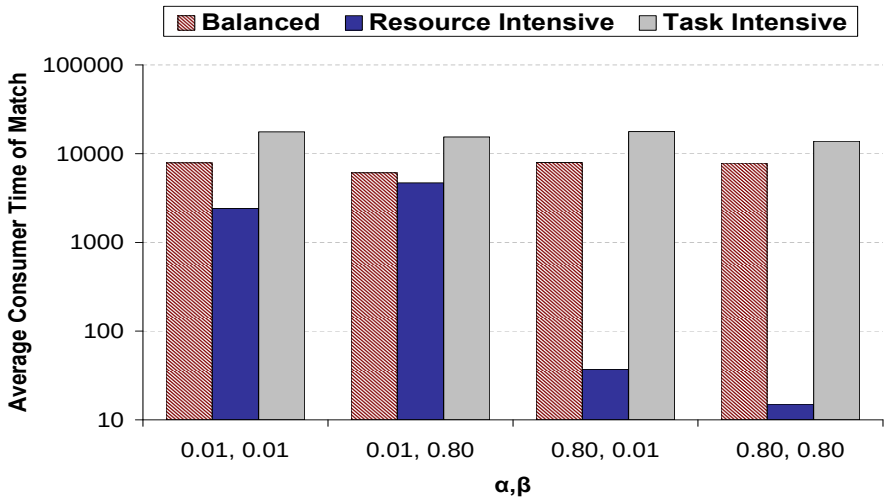

Figure 6.7: *Average time of finding matches for consumers with various values of α and β in different network conditions (Logarithmic Y-scale).*

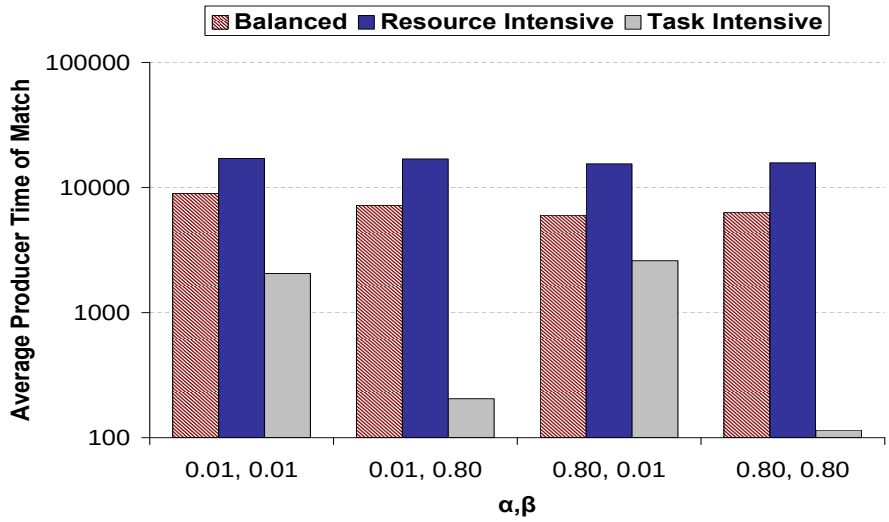

Figure 6.8: *Average time of finding matches for producers with various values of α and β in different network conditions (Logarithmic Y-scale).*

time for consumers to find a resource that has an ask price lower than or equal to their bid price. The same reasoning is applied for producers in a task intensive condition when β has a low value (Figure 6.8). In that case, consumers increase their bid prices in a low rate and it causes producers to spend more time on finding a consumer that bids a price higher than or equal to their ask price. The lowest consumer matching time in a resource intensive condition is obtained when $\alpha = 0.8$; and the lowest producer matching time in a task intensive condition is obtained when $\beta = 0.8$.

In a balanced condition, we do not see much difference in the average matching time of consumers and producers. However, the matching time is slightly higher for both consumers and producers when $\alpha = 0.01$ and $\beta = 0.01$. This is again because of a slower rate of updating producer and consumer prices that results in a longer time for finding proper matches.

u_{thT} and u_{thR} parameters

u_{thT},u_{thR} determine task and resource utilizations thresholds respectively for consumer and producer agents. These parameters determine the activeness of the agents in an ad-hoc Grid. If u_{thT} or u_{thR} is low, it implies that the consumer/producer agent is contributing with a low usage of its resources or is demanding a low completion rate of its tasks. If it is high, the consumer/producer agent is contributing to the Grid by offering more resources or is demanding more resources from the Grid. We consider the agents with a low value of task/resource utilization threshold as lazy agents and the agents with a high value of task/resource utilization threshold as active agents.

In this set of experiments, we study the impact of varying u_{thT} and u_{thR} on the task/resource utilization and the average time of finding matches. From previous discussion on parameters α and β, we observed that a higher value of task/resource utilization and also a lower average time of finding matches are obtained when these parameters have higher values. Therefore, we consider a high value of 0.8 for both α and β ($\alpha = \beta = 0.8$) when studying u_{thT} and u_{thR} parameters. We also assume that the task utilization threshold u_{thT} is equal to the resource utilization threshold u_{thR} for all consumer and producer agents (u_{th} =u_{thT} = u_{thR}). The experiments are performed in a balanced

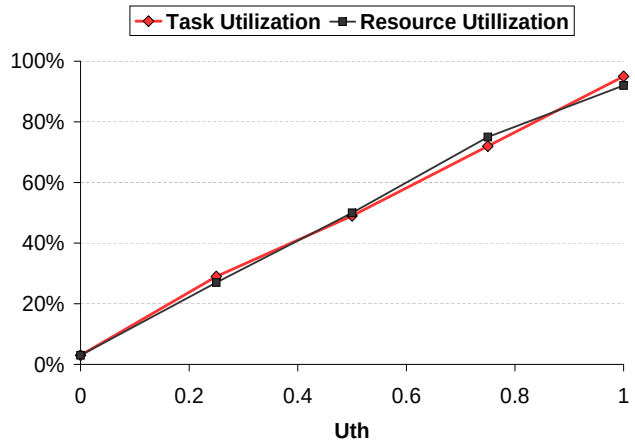

Figure 6.9: *Task/resource utilization (with different values of u_{th} in a balanced network (u_{th}=u_{thT}=u_{thR}).*

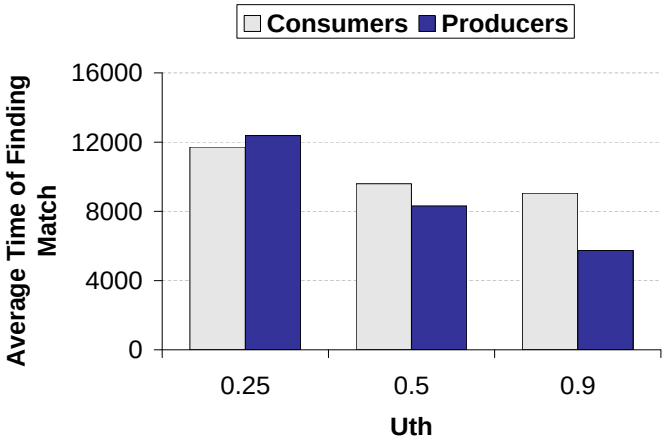

Figure 6.10: *Average time of finding matches with different values of u_{th} in a balanced network (u_{th}=u_{thT}=u_{thR}).*

condition.

- **Task/resource utilization:** We measure the task and resource utilizations considering different values for u_{th}. The results are shown in Figure 6.9. It can be deduced from the figure that the task and resource utilizations are linearly proportional to the threshold value u_{th}. As it can be seen in Figure 6.9, the tasks and resource utilizations increase

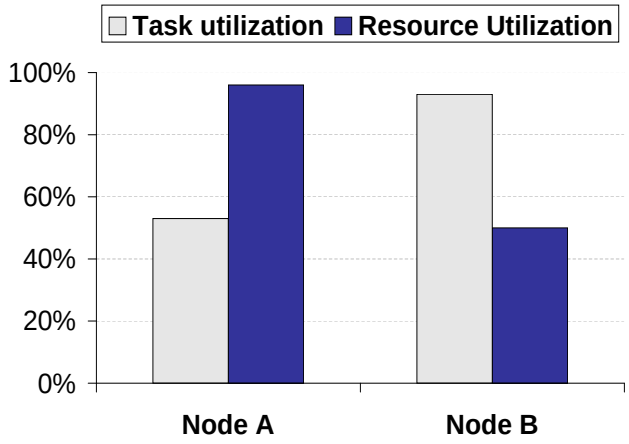

Figure 6.11: *Task/resources utilization for lazy/active agents. Node type A:* $u_{thR} = 0.9$ *and* $u_{thT} = 0.25$*; Node type B:* $u_{thR} = 0.25$ *and* $u_{thT} = 0.9$*.*

when u_{th} is increased since in this case the agents become more active.

- **Average time of finding matches:** In the same experiments, the average time taken to find a match is measured for both consumer and producer agents. With increasing u_{th}, agents become more active in ad-hoc Grid and in consequence the time to find a match decreases. As seen in Figure 6.10, consumer and producer agents with higher values of u_{th} spend less time to find matches compared to the agents that have lower values of u_{th} .

Lazy/active agents

To show in a Grid how consumer and producer agents can become lazy or active by modifying u_{thT} and u_{thR} parameters, we undertake one other experiment. Assume that in a Grid some nodes have heavy workloads and need extra resources. These kinds of nodes prefer to complete their tasks rather than offering their resources, so these nodes can be regarded as lazy producers but active consumers. Other nodes are more willing to contribute their resources as they have idle resources or low workloads. That means that these nodes are active producers but lazy consumers. In this experiment, we consider 40 nodes in the network. The values of $u_{thR} = 0.9$ and $u_{thT} = 0.25$

are set for 20 nodes which are assumed to be active producers and lazy consumers. Other half of the nodes have values that are set to $u_{thR} = 0.25$ and $u_{thT} = 0.9$ to create active consumers and lazy producers. We consider a balanced condition where each node generates more or less the same number of tasks and resources during the experiment. We study the task and resource utilizations of the individual nodes from these two categories. Figure 6.11 shows the average of resource and task utilizations of two typical nodes from each category. Node A is an instance of the first category with low workloads which has more idle resources, and Node B is an instance of the nodes with high workloads. As seen in Figure 6.11, nodes of type A contribute more as producers than as consumers and Grid utilizes more resources (93%) from this group compared to the tasks (56%). On the other hand, more tasks are utilized from the nodes of type B compared to the resources, which is 96% for the tasks and 59% for the resources. These nodes contribute to Grid more as consumers than as producers. Therefore, consumer and producer agents can decide on their task usage and resource contribution to Grid individually by setting the parameters u_{thT} and u_{thR}.

6.4 Budget Constraint

In absence of a budget constraint for bidding agents, the bid prices and in consequence the transaction prices may grow infinitely. We observed very high prices in a task intensive condition when studying pricing behavior in Section 6.3.1. If we want to apply real money in a resource allocation mechanism then we have to avoid such excessively high prices, since these prices do not exist in real markets. We limit bid prices by applying a budget constraint in the pricing mechanism. To compute the bid prices with a budget constraint, we change Equation 6.4 as below:

$$p_b(t) \leftarrow \min\{max_b, \beta(u_{thT} - tu(t))p_b(t-1)\} \tag{6.8}$$

where max_b is the maximum bid price that a consumer can afford with its limited budget. This means that consumers are subject to a budget limitation and can not bid a price beyond their budget. A minimum price can also be considered for ask prices below which producers do not sell their resources.

Hence, Equation 6.3 is changed to:

$$p_a(t) \leftarrow \max\{min_a, \alpha(u(t) - u_{thR})p_a(t-1)\} \tag{6.9}$$

In this section, we study the impact of budget constraint on the boundness of transaction prices and we also study its impact on the overall throughput of an ad-hoc Grid. Throughput is computed based on the task and resource utilizations.

Furthermore, we are going to investigate the case where consumer agents restrict their budget for each transaction. This means that a consumer may consider the whole amount of their budget or a part of it for buying a resource. As maximum bid prices are limited by the budget (Equation 6.8), therefore we expect higher upper limits for bid prices when spending higher budget. We study the influence on transaction prices and throughput when a consumer considers different amount of its budget for one transaction.

6.4.1 Budget influence

We study the budget influence using the following criteria:

- Boundness of transaction price: The price at which a transaction is concluded is computed as the average of consumer and producer prices. The boundness of transaction price defines the upper limit for a transaction price.
- Throughput: Throughput is calculated based on the overall task and resource utilizations. Task utilization is the ratio of allocated tasks to all submitted requests and resource utilization is the ratio of allocated resources to all submitted offers.

Figures 6.12 and 6.2 show the boundness of transaction price in different network conditions respectively with and without budget constraint. As we already observed in Figure 6.2, the transaction prices are growing infinitely in a task intensive condition when there is no budget constraint. Applying budget constraint prevents infinite prices and puts an upper limit for consumer prices. We can observe in Figure 6.12 that prices are limited in the case of task intensive condition. In the balanced and resource intensive conditions, the prices are not growing to high values.

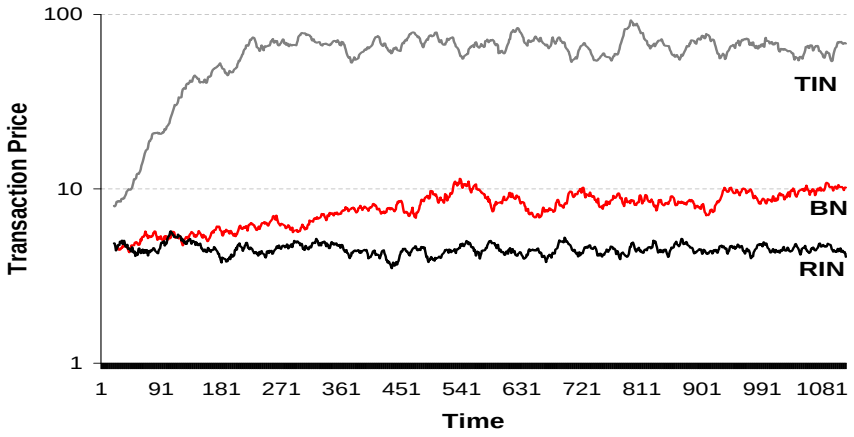

Figure 6.12: *Transaction price evolution in a Task Intensive network (TIN), a Resource Intensive network (RIN), and a Balanced Network (BN) with budget constraint.*

We also study the throughput of the network in three network conditions with and without budget constraint. The throughput of the network in different network conditions when a budget constraint is applied is presented in Table 6.1. Our experiments without a budget constraint show the same throughput in the three network conditions. To avoid repetition of the results, we do not present these results.

Furthermore, we study the impact of restriction budget for each transaction. When a budget constraint is applied, consumers are restricted to a price to spend below or equal to their budget in each transaction. A consumer may consider the whole amount of its budget or a part of it for a transaction. We want to know what is the affect of considering different percentages of the budget. We study two cases: one is when consumers consider their entire budget and another is when consumers consider 50% of their budget for buying a resource in each transaction. It should be noticed that the maximum budget is spent only when a bid price reaches its maximum value. For each of the two cases, throughput and average transaction price are measured in the balanced and task intensive conditions. We do not take resource intensive condition into account since in this condition, bid prices are low and they are far below their maximum values. The results of the two cases are shown in Table 6.1. As we can see from the results, the average transaction price in

	Task Utili.	Resource Util.	Ave. trans. Price
BN, 100% budget	93%	93%	12
BN, 50% budget	92%	92%	11
TIN, 100% budget	24%	95%	99
TIN, 50% budget	24%	95%	52

Table 6.1: *Budget influence on task/resource utilization and average transaction price in a balanced network (BN) and a task intensive network (TIN).*

a balanced condition is more or less the same in the two cases. This is because the prices are more stable and hardly reach their maximum values in a balanced condition. In a task intensive condition, the average transaction price decreases to almost half when 50% of the budget is considered. In this condition, prices grow and they may reach their maximum values. Therefore, considering a lower budget in this condition lowers the maximum prices.

Regarding throughput of the network, it can be seen from the results that the task and resource utilizations do not depend on the amount of budget that a consumer considers in each transaction. It could be expected that in case of considering total budget, a consumer may spend its entire budget in one transaction and in consequence lower task/resource utilization may occur. This is explained as follows. In a balanced condition that the prices are not growing and hardly reach their maximum values, the probability of spending the total budget in one transaction is low. This reasoning also applies to the resource intensive condition. As in this condition, the prices are even lower than in the balanced condition. In a task intensive condition which is a competitive condition for consumer agents, the prices may reach their maximum values. In such a condition, there is a tradeoff between bidding higher prices (spending more budget) and higher chance of getting resources. The agents that bid higher prices have more chance of winning. Remembering the fact that a node can switch its role between consumer agent and producer agent, then even if bidding higher prices reduces the consumer budget but also increases the producer income which can be a consumer budget at some other time.

6.5 Conclusions

In this chapter, we introduced a dynamic pricing mechanism, so called λ, which enables the nodes to make sensible bids. The λ pricing mechanism defines a price purely on the basis of local information but taking into account how successful one has been in the previous requests. By empirical evaluation, we showed that this mechanism helps in regulating the supply and demand in the ad-hoc Grid. For instance, if the demand for resources is high, prices start to increase as to discourage consumers from demanding the resources and to maintain equilibrium of supply and demand of resources. One important conclusion we can draw from the experiments discussed in this chapter is that the price indeed reflects the scarcity of a resource in the Grid. There will be a clear trend in the price evolution given different network conditions, which is the result of individual and local decision making and not of any global entity processing all available information.

We studied the parameter regime of the pricing functions under different network conditions. There are four parameters that can be manipulated by the consumer and producer agents: α, β, u_{thT} and u_{thR}. The α and β are parameters through which the rate of increasing/decreasing ask and bid prices are defined. The u_{thT} and u_{thR} determine the activeness of the agents in the ad-hoc Grid. Our experiments showed that a resource intensive network is more influenced by α parameter while a task intensive network is more influenced by β. In a balanced network, both α and β parameters have the same affect. Moreover, producers and consumers can change the degree of their activity in the ad-hoc Grid using u_{thR} and u_{thT} parameters. In fact, producers and consumers can decide on their level of contribution to the ad-hoc Grid considering their capabilities and their workloads. They become less/more active by decreasing/increasing these values.

We also observed that prices may grow infinitely when there is no upper limit for bid prices. To avoid unlimited prices and to be consistent with real markets, we applied a budget constraint for consumer agents. Higher/lower the considered budget for each transaction, higher/lower is the maximum price that a consumer is able to pay in one transaction.

Chapter 7

Bidding Strategies

In an ad-hoc Grid where tasks and resources are not dedicated and their availability changes frequently, it is not feasible to have a central monitoring entity that assembles all available information. The question is then how can an agent recognize the current network condition for taking a proper action. In this chapter, we will show that an agent can sense the condition through the way its utilization of the ad-hoc Grid is evolving. For instance, when the utilization is decreasing, an agent has to adapt its bidding strategy to become more aggressive. On the other hand, when the utilization is increasing, the agent continues its way of bidding conservatively.

In Chapter 6, we introduced λ pricing mechanism in which agents learn from their previous utilizations of an ad-hoc Grid and accordingly define their prices. This chapter is complementary to Chapter 6 and introduces a new version of the λ pricing mechanism. In this mechanism, two aggressive and conservative bidding strategies are applied by adjusting pricing parameters automatically as the network condition changes.

7.1 Aggressive Bidding Strategy

Aggressive bidding is defined as a strategy where agents increase or decrease their prices in a high rate. Aggressive bidding strategy is adopted by consumer agents when there is a high competition between these agents due to the high

demand for resources in a network. On the other hand, producers adopt an aggressive bidding strategy when there is a high competition between producers due to a high supply in a network. By adopting aggressive bidding, consumers and producers respectively speed up increasing or decreasing their prices to outbid their competitors.

7.2 Conservative Bidding Strategy

A bidding is called conservative when agents change their prices in a low rate. Consumers use a conservative bidding strategy when the supply of resources in the network is high. In this case, consumers decrease their bid prices to reduce their spending. They decrease the bid prices in a low rate conservatively not to lose the competition. In the same way, when demand for resources is high, producers increase their ask prices to increase their credits. They increase their ask prices conservatively in a low rate not to be surpassed by other producers. The aggressive and conservative bidding strategies are taken when there is no balance between supply and demand in the network. In such a condition, when one of the consumers or producers parties employs an aggressive or a conservative strategy, the other party reacts in the other way by taking respectively a conservative or an aggressive strategy.

In the following section, we present an extension of the pricing algorithm introduced in Chapter 6, Section 6.1 in which aggressive and conservative bidding strategies are introduced.

7.3 Extension of the λ Pricing Algorithm

The agents in an ad-hoc Grid choose a bidding strategy through the way that their utilization of Grid changes. For instance, when the task utilization is decreasing, a consumer agent discovers that the resources are becoming scarce. Therefore, it has to adapt its bidding strategy to become more aggressive in order to increase its bid price faster. Similarly, when the resource utilization is decreasing, it shows a low demand for resources. Then, a producer agent bidding strategy should switch to an aggressive strategy to decrease its ask price rapidly. The faster decreasing the task or resource utilization, the

more aggressive a strategy is required. On the other hand, increment in the task or resource utilization shows respectively supply or demand is high. In this case, consumers should decrease their bid prices and producers should increase their ask prices which is called a conservative strategy. The faster the task or resource utilization increases, the less conservative the bidding strategy needs to be. This method of adaptation is applied in the following proposed pricing mechanism which is an extension of the pricing mechanism presented in Chapter 6. In this version, the rate of increasing or decreasing prices is adjusted automatically and there is no need for setting these parameters manually.

We consider a budget constraint and assume that nodes are assigned a limited budget when joining the network. Consumers spend their budget when buying resources and producers earn credits when devoting their idle computational resources.

Let us denote by $p_b(t)$ the bid price of a consumer agent and by $p_a(t)$ the ask price of a producer agent at time t. We assume that:

1. Each consumer agent has a maximum bid price, denoted by max_b, for a resource unit. This maximum price is limited by the node's budget.
2. Each producer agent has a minimum ask price, denoted by min_a, below which the producer is not willing to sell a unit of resource.

More formally, we have:

$$\begin{aligned} \forall t, \quad & p_a(t) \geq min_a \text{ and} \\ & p_b(t) \leq max_b \end{aligned} \tag{7.1}$$

In our pricing algorithm, agents update their ask (respectively bid) prices using the utilizations they gained from Grid in their previous experiences.

Informally, the idea is as follows. If an agent has been unsuccessful in buying resources, it updates its bid price in a way that tends to increase its chance to buy resources in the future. On the other hand, if an agent has been successful, it conservatively continues to bid in a similar way that ensures its chance of buying resources in the future.

A seller agent behaves in a similar manner. If it has not been successful in selling resources, it updates its ask price in a way that increases its chance of selling its resources in the future. Otherwise, it behaves in a conservative manner.

Formally, the ask price of a producer agent at time t is computed according to Equation 7.2, while the bid price for a consumer agent is given by Equation 7.3.

$$p_a(t) \leftarrow \max\{min_a, p_a(t-1) + \alpha.p_a(t-1)\} \tag{7.2}$$

$$p_b(t) \leftarrow \min\{max_b, p_b(t-1) + \beta.p_b(t-1)\} \tag{7.3}$$

where α and β are coefficients that determine the rate at which the price is increasing or decreasing. We name these parameters *reinforcement parameters*, because they reinforce the learning and apply two aggressive and conservative bidding strategies. These parameters are set according to variations in the task or resource utilization at each individual agent over a specific time period.

For a given agent, we define the task utilization as the ratio of allocated tasks to all submitted requests and the resource utilization as the ratio of allocated resources to all submitted offers. *ru(t)* and *tu(t)* represent respectively the resource and task utilizations at the time t. We already defined these values in Chapter 6, Section 6.1.

We now define variations in resource and task utilizations. To this end, let $T_1 = [s_1, e_1]$ and $T_2 = [s_2, e_2]$ be two consecutive time periods such that $e_1 = s_2$ where e_2 is the current time.

We capture variations in resource (respectively task) utilization from period T_1 to T_2 by the following equations.

$$\Delta ru_{(e_1 \rightarrow e_2)} = ru(e_2) - ru(e_1) \tag{7.4}$$

$$\Delta tu_{(e_1 \rightarrow e_2)} = tu(e_2) - tu(e_1) \tag{7.5}$$

If we consider the length of the history in our pricing mechanism N, then T_1 and T_2 are the time periods in each $N/2$ requests or offers are submitted. We now define the reinforcement parameters α and β as follows:

$$\alpha = \begin{cases} -(K - (ru(e_2))^2)^2 & if\ \Delta ru_{(e_1 \rightarrow e_2)} \leq 0 \\ L * (ru(e_2))^2 & if\ \Delta ru_{(e_1 \rightarrow e_2)} > 0 \end{cases} \tag{7.6}$$

$$\beta = \begin{cases} (K - (tu(e_2))^2)^2 & if\ \Delta tu_{(e_1 \rightarrow e_2)} \leq 0 \\ -L * (tu(e_2))^2 & if\ \Delta tu_{(e_1 \rightarrow e_2)} > 0 \end{cases} \tag{7.7}$$

K and L have positive values that define respectively the maximum rate of aggressive and conservative bidding strategies. According to Equation 7.6, when resource utilization decreases ($\Delta ru \leq 0$), α becomes negative. A negative α results in decreasing the ask price according to Equation 7.2. The smaller the value of current resource utilization (ru), the more negative α and the faster decreasing the ask price is. On the other hand, when resource utilization increases ($\Delta ru > 0$), α becomes positive that causes increasing the ask price. The higher the value of current resource utilization (ru), the higher α and the faster increasing the ask price is. The same behavior is seen for bid price in Equations 7.7 except that the increasing the task utilization ($\Delta tu > 0$,) decreases the bid price and decreasing the task utilization ($\Delta tu > 0$), increases the bid price.

An aggressive bidding is adopted by consumers or producers when they respectively increase or decrease their bid or ask prices. On the other hand, a conservative bidding is applied when consumer or producer agents respectively decrease or increase their bid or ask prices. In an aggressive bidding, the maximum rate of increasing price by consumers or decreasing price by producers is K. While in a conservative bidding, the maximum rate of increasing price by producers or decreasing price by consumers is L. The values of K and L can be modified individually by nodes based on user preferences such as budget consumption, urgent tasks. For instance, if a node has a very limited budget, it has to choose a lower level for aggressive bidding and avoid very high bid prices. On the other hand, in case of very urgent tasks, nodes can always increase the level of their aggressive or conservative bidding.

Figure 7.1 presents the aggressive and conservative bidding strategies by depicting reinforcement parameters α and β as the task/resource utilization changes. We consider the values $K = 1$ and $L = 0.2$. The aggressive and conservative bidding strategies are explained according to Figure 7.1 as follows:

- **Aggressive Bidding**: When the resource or task utilization is decreasing, producer or consumer agents employ an aggressive bidding strategy. In an aggressive bidding, the rate of increasing price by consumers (β) and the rate of decreasing price by producers (α) depends on the value of recent task/resource utilization. An aggressive bidding is depicted in Figure 7.1 at the time period [t_1,t_2]. As it can be seen in the figure, lower rates of chaining the price are observed when the

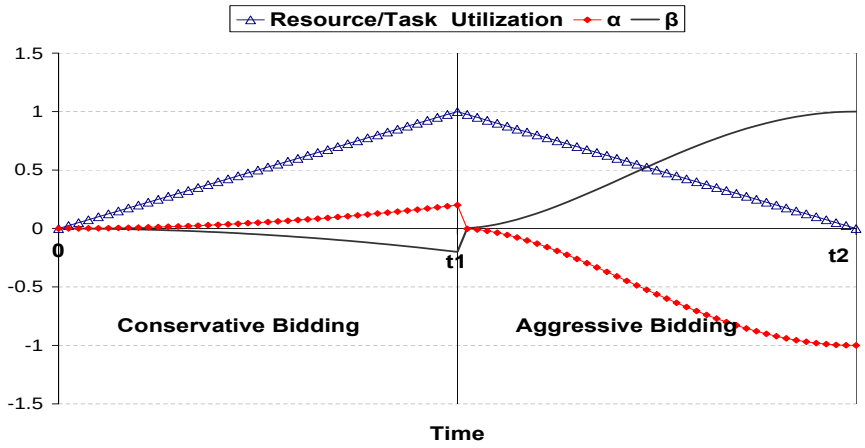

Figure 7.1: *Reinforcement parameters (α and β) and task/resource utilization.*

task/resource utilization is higher, and higher rates of chaining the price are observed when the task/resource utilization is lower. Here, the maximum rate of increasing or decreasing the price is 1 ($K = 1$) when the task or resource utilization has its minimum value ($ru(T) = 0$ or $tu(T) = 0$).

- **Conservative Bidding**: Producer or consumer agents bid with conservative strategy when the resource or task utilization is increasing. In a conservative bidding, the rate of increasing price by producers (α) and the rate of decreasing price by consumers (β) are proportional to the value of recent task/resource utilization. An aggressive bidding is depicted in Figure 7.1 at the time period [0,t_1]. We can observe from the figure, lower rates of chaining the price when the task/resource utilization is lower, and higher rates of chaining the price when the task/resource utilization is higher. The maximum rate of increasing or decreasing price is 0.2 ($L = 0.2$) when the task or resource utilization has its maximum value ($ru(T) = 1$ or $tu(T) = 1$).

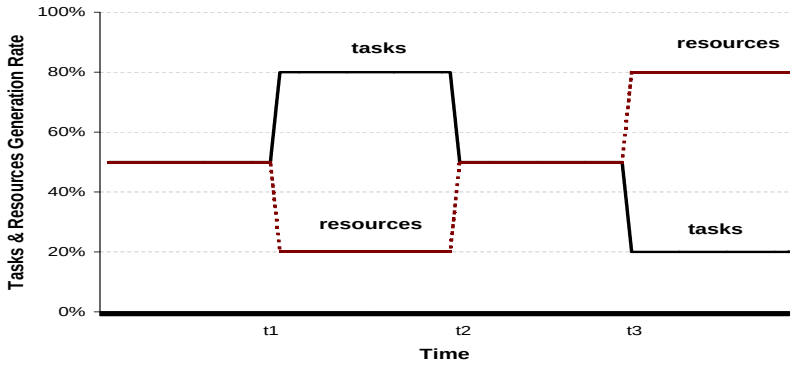

Figure 7.2: *The generation rate of tasks and resources.*

7.4 Performance Evaluation

In this section, we study the aggressive and conservative bidding strategies experimentally. We show how agents take a proper strategy according to the current availability of resources and tasks in the network. In these experiments, we have considered a history of length 10 in the pricing mechanism. This means that the agents compute the prices considering a history of 10 latest submitted requests or offers.

7.4.1 Experimental Condition

A dynamic network condition is considered for running the experiments. In this condition, supply and demand vary during four time periods as the number of tasks/resources in the network changes. Resources and tasks are generated with the ratios 20%, 50%, and 80% in different time periods. Figure 7.2 depicts the network status evolving over time. In time period [0,t_1], a balanced condition with more or less an equal number of tasks and resources is considered. From the time $t1$ to $t2$, the number of tasks increases. Time period [t_1,t_2] is a task intensive condition which can represent the busy hours during the working days. From the time $t2$ onward, tasks start to decrease and again we have a balanced condition. Resources start to increase from the time $t3$ onward and a resource intensive condition is generated which could simulate the situations with low workloads and high idle resources during night

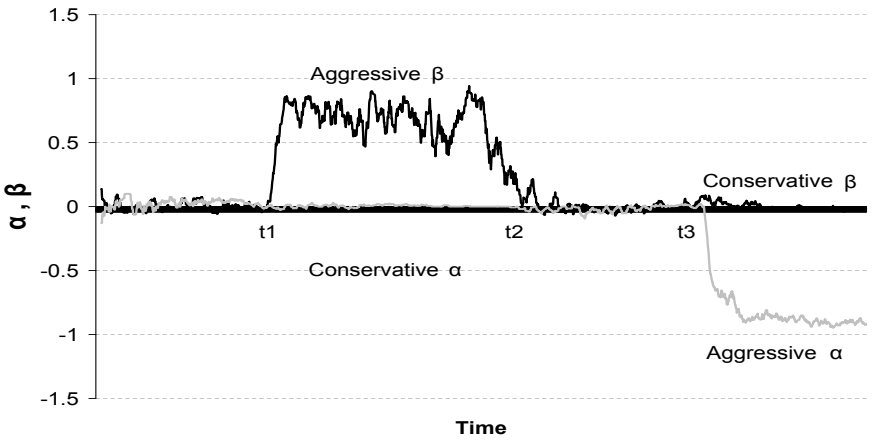

Figure 7.3: *Reinforcement parameters evolving as the network condition changes.*

or weekend. In the experiments shown here, each time period is set to 10 minutes.

7.4.2 Reinforcement Parameters & Transaction Prices

We analyzed the parameters α and β theoretically in Section 7.3. In this section, we study these two parameters practically when running experiments in the dynamic condition described above. We also present the transaction price evolution in the dynamic network condition. Figures 7.3 and 7.4 depict respectively the reinforcement parameters and transaction prices evolving during the time in the dynamic condition of the network. As it can be observed from Figure 7.3, in the time periods [0, t_1] and [$t2$,$t3$], where there is a balanced condition between tasks and resources, the consumer and producer agents do not take any aggressive bidding. Instead, they increase or decrease their prices more conservatively in this condition. In the time period [$t1$, $t2$], where there is a task intensive condition, the consumer agents take an aggressive bidding strategy and on the other side producers employ a conservative bidding strategy. As in such a network condition, consumers are abundant and they have to increase their prices aggressively to win the competition (aggressive β). In a task intensive network, producers increase their prices conservatively to increase their credits (conservative α). As seen

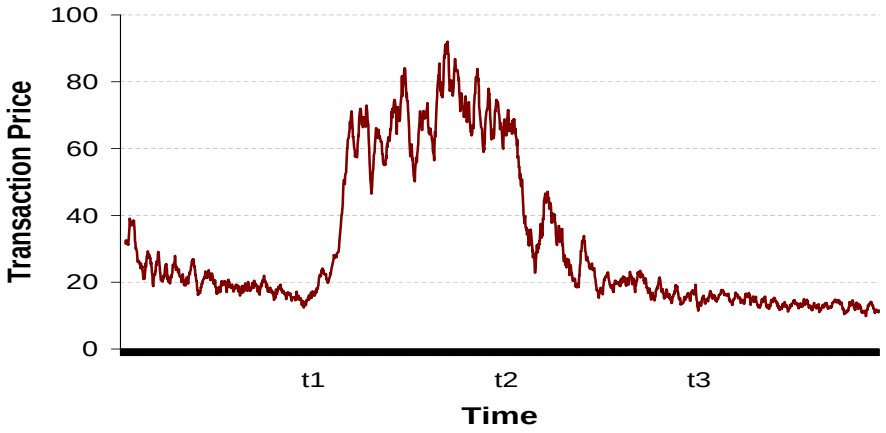

Figure 7.4: *Transaction price evolution in the dynamic network condition.*

in Figure 7.4, an upward trend of the transaction prices in this condition is observed. When a resource intensive condition is created from the time period $t3$ onward, resources become abundant. In this condition, as seen in Figure 7.3, producers take an aggressive strategy to decrease their prices (aggressive α) and consumers take a conservative strategy to decrease their prices (conservative β). Therefore, transaction price start to decrease in such a condition (see Figure 7.4).

As discussed in Section 7.3, the maximum rate for aggressive and conservative biddings can be set by each agent individually. This is feasible by modifying the constant coefficients K and L provided in Equations 7.6 and 7.7. These values can be modified according to consumer/producer preferences. It can be concluded that the λ mechanism besides providing adaptation to the network condition, also provides autonomy for consumer and producer agents to decide their level of the activity in the ad-hoc Grid. In Chapter 6, we showed that consumer and producer agents could determine their level of the activity using utilization thresholds. In the new version of λ pricing mechanism, the activeness of the agents can be decided by being more/less aggressive or conservative.

7.5 Conclusions

Based on the results that we obtained by studying the parameter regime of the pricing mechanism in chapter 4, we proposed an extension of λ pricing mechanism in this chapter. In λ pricing mechanism, consumer and producer agents can adapt to the new condition of the network by learning from their previous trades. They accordingly adopt a proper bidding strategy and modify their ask and bid prices. The two aggressive and conservative bidding strategies are applied by reinforcement parameters α and β at the agent level. These parameters are set dynamically as the resource and task utilizations evolve at the corresponding agents. For consumers, aggressive bidding strategy speeds up adaptation to the low-supply/high-demand condition by forcing consumers to bid higher prices. In the same way, an aggressive bidding strategy indicates that producers should expect less when the demand is decreasing and the supply is increasing. Therefore, producers are forced to offer lower prices. In a conservative bidding strategy, the decreasing consumer price or increasing producer price is achieved conservatively to keep the chance of winning the competition. A conservative bidding strategy encourages consumers to save their money by paying less when there is enough supply. Producers in a conservative bidding strategy are encouraged to ask more when there is abundant demand in the network.

Chapter 8

Pricing Mechanisms

In the literature, we can find several pricing mechanisms for consumer and producer agents in a Grid [34, 55, 90]. In this chapter, we evaluate the performance of four pricing mechanisms in an ad-hoc Grid. The evaluations are based on the criteria which are important to Grid users and owners. We selected three pricing mechanisms from the current state-of-art namely Zero Intelligence (ZI) [44], Zero Intelligence Plus (ZIP) [35], and Gjerstad and Dickhaut (GD) [43]. We also implemented our pricing mechanism, λ, and compared its performance against the others. The selected pricing mechanisms are dynamic and are well suited to be used in auctions. The three ZIP, GD, and λ are all learning pricing mechanisms in the sense that the price is decided based on the agent's previous experiences. ZI is a non-learning mechanism as its name implies. In the ZI, ZIP, and λ mechanisms, consumer and producer agents decide prices just based on their local knowledge. In the GD mechanism, the consumer and producer agents need to know about the bid and ask prices of the recent transactions in the network for setting their prices. Despite our aim to avoid broadcasting information in the network, we nevertheless decided to include the GD pricing in our comparison as it is a well known mechanism used in auctions. In the following sections, we first study the pricing mechanisms in more detail and then present the performance evaluation of the mechanisms in an ad-hoc Grid.

8.1 Zero Intelligence (ZI)

ZI mechanism [44] uses a simple strategy in which agents generate random order prices, without taking into account the state of the network. There are two types of Zero Intelligence pricing strategies, ZI-U (Zero Intelligence-Unconstraint) and ZI-C (Zero Intelligence-Constraint). In ZI-U, the traders announce a price for a unit of commodity using a uniform distribution of probabilities across a specific range for a minimum and a maximum price. In a ZI-C, traders are subject to a budget constraint. ZI-C sellers can only make offers in the range between their limit price and a maximum price, and ZI-C buyers can only make bids in the range from a minimum up to their limit price which is restricted by their budget [34]. ZI-C market prevents the traders from engaging in transactions that they cannot settle. Traders have no intelligence in ZI-U and ZI-C markets.

8.2 Zero Intelligence Plus (ZIP)

In ZIP mechanism [35], each agent maintains a scalar variable denoting its desired profit margin, and it combines this with a unit's limit price to compute a bid or ask price. At the start of trading, the price is initialized to a random positive value (a private value), and it is adjusted after every successful or failed trade. For a successful trade, price is adjusted towards the transaction price. For a failed bid price, buyers adjust price in the direction of beating the failed bid. Sellers behave similarly for failed ask prices. Bid function for a ZIP agent a at the time t is as follows [21]:

$$p_a(t) = x_a * (1 - \mu_a(t)) \tag{8.1}$$

Where μ_a represents the fraction above or below its value at which an agent bids and x_a is a private value. In this mechanism, the problem of learning an optimal strategy is reduced to learning optimal margins. The margin $\mu_a(t)$ is adjusted to $\mu_a(t+1)$ using the rule:

$$\mu_a(t+1) = \mu_a(t) + \Delta_a(t) \tag{8.2}$$

and,

$$\Delta_a(t) = \eta(d_a(t) - \mu_a(t)) \tag{8.3}$$

$d_a(t)$ is a desired margin which is defined as:

$$d_a(t) = 1 - o_a(t)/x_a(t) \tag{8.4}$$

where o_a is an optimal bid. In selecting an optimal bid, the ZIP agent considers any bid which could have increased their reward. Hence, adaptive agents are restricted to updating their values in two situations:

1. When an agent wins: This situation is characterized as the agent being greedy. Hence, a buyer agent increases its margin by estimating an optimal bid to be lower than the current bid. On the other side, a seller agent reduces its margin by estimating an optimal bid to be greater than the current bid.

2. When an agent looses: In this situation the agent is fearful and becomes more cautious. A buyer agent reduces its margin by estimating an optimal bid to be greater than the current bid. In this situation, a seller agent increases its margin by estimating the optimal bid to be lower than the current bid.

In order to smooth the update variable μ_a, ZIP agents employ a momentum coefficient Γ_a. Δ_a is replaced by Γ_a in the Equation 8.2. Γ_a is determined by the formula:

$$\Gamma_a(t+1) = \gamma_a \Gamma_a(t) + (1 - \gamma_a)\Delta_a(t) \tag{8.5}$$

where $\gamma_A \in [0, 1]$. Larger values of γ results in greater smoothing. The update on the margin then becomes:

$$\mu_a(t+1) = \mu_a(t) + \Gamma_a(t) \tag{8.6}$$

ZIP agents select their optimal bid by randomly sampling a range of values given by:

$$o_a(t) = b_a(t).R_a + A_a \tag{8.7}$$

where R and A are observations of independent random variables with a uniform distribution. Based on the two situations mentioned above, adaptive agents update their prices as follows. When a buyer ZIP agent wins, we consider $A \in [A_{min}, 0]$ and $R \in [R_{min}, 1]$. When a buyer agent looses, we consider $A \in [0, A_{max}]$ and $R \in [1, R_{max}]$ [21]. For seller agents selecting variables A and R is opposite, as sellers and buyers update their prices in inverse directions. Therefore, when a seller ZIP agent wins, $A \in [0, A_{max}]$

and $R \in [1, R_{max}]$ and when it looses, $A \in [A_{min}, 0]$ and $R \in [R_{min}, 1]$ are considered[21].

In our experiments, learning rate $\eta = 0.1$, momentum coefficient $\gamma = 0.7$, $R_{min} = 0.95$, $R_{max} = 1.05$, $A_{min} = -0.05$ and $A_{max} = 0.05$ are considered [21].

8.3 Gjerstad and Dickhaut (GD)

The GD trader algorithm for Continuous Double Auction (CDA) is a memory based agent architecture described in [43] and refined in [90]. GD traders have a strategy for announcing a price based on maximizing expected profit. The maximization of expected profit relies on the GD trader forming a belief function. GD agents [43] use the history H_M of recent market activity (the bids and asks leading to the last M trades) to calculate a belief function $f(p)$ estimating the probability for a bid or ask at price p to be accepted. For a seller, $f(p)$ is defined as follows:

$$f(p) = \frac{AAG(p) + BG(p)}{AAG(p) + BG(p) + UAL(p)} \tag{8.8}$$

where $AAG(p)$ is the number of accepted asks in H_M with $price \geq p$, $BG(p)$ is the number of bids in H_M with $price \geq p$, and $UAL(p)$ is the number of unaccepted asks in H_M with $price \leq p$. For a buyer $f(p)$ calculated as:

$$f(p) = \frac{TBL(p) + AL(p)}{TBL(p) + AL(p) + RBG(p)} \tag{8.9}$$

where $TBL(p)$ is the number of accepted bids in H_M with $price \leq p$, $AL(p)$ is the number of asks in H_M with $price \leq p$, and $RBG(p)$ is the number of unaccepted bids in H_M with $price \geq p$. The GD algorithm requires some alternation for sealed bid traders in Double Auction since a different set of information is available. Suppose an agent is able to record the price paid by the winner (tp) and the winning bid price (b_w) in a history H_M. With this historical information, it can be asserted that if a price p_w was a winning price, then a bid price $p \geq p_w$ would also have won. An agent belief that a price p will be accepted is:

$$f(p) = \frac{T_p}{M} \tag{8.10}$$

where T_p is the number of times in which p would have been a winning bid price and M is the length of the history. The agent then estimates the gain from the trade at price tp if it was the winner. The gain for the seller is:

$$r_p = tp - x \tag{8.11}$$

and for the buyer:

$$r_p = x - tp \tag{8.12}$$

where x is the private value of the agent. The expected profit is then the product of probability of winning and estimate of the profit assuming the agent won:

$$E_p = f(p) * r_p \tag{8.13}$$

The agent selects the bid price that maximizes the expected profit. In the cases when the maximum expected profit is less than or equal to zero, the agent chooses a bidding margin such as in ZIP mechanism.

8.4 Learning and Adaptive Mechanism for Bidding Agents (λ)

λ is the latest version of the proposed pricing mechanism presented in Chapter 7, Section 7.3. In this pricing mechanism, agents update their ask (respectively bid) prices using the utilizations they gained in their previous experiences from the ad-hoc Grid. Based on this mechanism, ask price of a producer agent at time t is computed according to Equation 8.14, while the bid price for a consumer agent is given by Equation 8.15.

$$p_a(t) \leftarrow \max\{min_a, p_a(t-1) + \alpha.p_a(t-1)\} \tag{8.14}$$

$$p_b(t) \leftarrow \min\{max_b, p_b(t-1) + \beta.p_b(t-1)\} \tag{8.15}$$

where min_a is the minimum ask price and max_b is the maximum bid price limited by a budget. α and β are the coefficients that determine the rate at which the price is increasing or decreasing. These parameters indicate the aggressive and conservative bidding strategies. More detail explanation is given in Chapter 7, Section 7.3.

8.5 Performance Evaluation

We implement and study the performance of the four pricing mechanisms. λ, ZI, and ZIP mechanisms are easier to be implemented compared to GD mechanism since in these mechanisms, agents define the new prices just based on their local knowledge. Consumer/producer agents in λ mechanism need a history of their N previous submitted requests/offers. In our implementation, we have considered $N = 10$ for λ mechanism. In ZI, agents need to keep just their previous request or offer ($N = 1$). No history information is required in ZI mechanism. Among the four mechanisms, GD is more complicated as it requires broadcasting price information to agents for any transaction made in the market. The pricing mechanisms are studied in two different ways: a homogenous and a heterogenous pricing mechanisms under three different network conditions. In the homogenous case, the entire population of agents adopt the same pricing mechanism and in a heterogenous case, different pricing mechanisms are adopted by different agents. For this study, we consider the same three different network conditions as used throughout this thesis.

The Continuous Double Auction (CDA) with discriminatory pricing strategy is used as the matchmaking mechanism. The matchmaking between requests and offers are subject to resource quantity, deadlines, price and budget constraints. In all pricing mechanisms, the agents are assigned an equal amount of budget when joining the network. Each agent starts with an initial price and then update its price using a pricing mechanism. The initial prices for all agents is generated randomly in a similar range.

8.5.1 Criteria to Evaluate Pricing Mechanisms

In this section, we present the criteria considered for evaluating the pricing mechanisms.

- Throughput: Throughput is measured in terms of task and resource utilizations. Task/resource utilization are defined as presented in Chapter 5, Section 5.2.1.
- Load/resource balancing: Load/resource balancing studies how evenly tasks are distributed among producer agents (load balancing) and on the other hand, how evenly resources are distributed among consumer

agents (resource balancing). In Chapter 5, Section 5.5, we have explained how to measure load/resource balancing.

- Consumer/producer deadline satisfaction: Deadline satisfaction says how fast consumer/producer agents find the matches within their deadline (TTL). Consumer/producer deadline satisfaction are measured as we already showed in Chapter 5, Section 5.2.1.
- Price volatility: price volatility can be a measure of uncertainty in a network. This criterion which shows the variation in transaction prices, is computed based on the standard deviation. In Chapter 5, Section 5.2.1, we have given the description of how this metric is measured.
- Average transaction price: Average transaction price shows the average value of the actual prices of resources in a network. Low prices are desired in a market as they imply that more resources can be purchased using a given amount of budget.

8.6 Experimental Results

In this section, we present the results of applying different pricing mechanisms in the framework. The pricing mechanisms are evaluated considering the above mentioned criteria in the cases of homogenous pricing and heterogenous pricing under different network conditions. The results for task/resource utilizations, load/resource balancing, and average deadline satisfactions are shown in the form of percentages. The experiments are performed in a local ad-hoc Grid with 60 nodes. Therefore, there are 60 consumer agents and 60 producer agents participating in the ad-hoc Grid.

8.6.1 Homogenous Pricing

In the homogenous pricing, all agents in the network apply the same pricing mechanism. The performance of each pricing mechanism is studied separately in the three different network conditions.

	Average TPrice	Price Volatility	Task/Res. Utilization	Resource Balancing	Load Balancing	Average consumer Deadline Satisfaction	Average producer Deadline Satisfaction
λ	15	6.7	92%	96%	98%	87%	85%
ZI	47	34.2	80%	93%	95%	75%	72%
ZIP	26	10.2	61%	45%	30%	56%	56%
GD	30	1.06	50%	24%	4%	40%	45%

Table 8.1: *Performance of different mechanisms in a homogenous pricing under a balanced network condition.*

Balanced network

Table 8.1 shows the evaluation results of the four pricing mechanisms in a balanced condition regarding the mentioned criteria. Stable prices are expected in a market when supply and demand are equal (a balanced condition). Price volatility measurements in Table 8.1 show that GD provides the most stable prices among the four mechanisms while ZI presents a high volatility in prices. Unstable prices in ZI are due to random generation of bid and ask prices in this mechanism. Our λ mechanism also has a low price volatility in this condition. We also measure the average transaction price in each mechanism. By comparing these values in Table 8.1, we observe that our λ mechanism gives the lowest transaction prices while ZI presents the highest among the four mechanisms. Figure 8.1 depicts transaction prices evolution during the simulation time in each mechanism. As it can also be observed from the figure, ZI mechanism presents the highest price fluctuation and also the highest price values, GD shows the most stable prices, and λ mechanism gives the lowest price values with a low fluctuation in prices.

For each mechanism, we measure throughput of the network in terms of the task and resource utilizations. In a balanced condition, the task utilization in the network is equal to the resource utilization since the total number of tasks and resources are equal in this condition. The values of task/resource utilization presented in Table 8.1 show that λ mechanism gives the highest throughput compared to the other mechanisms. The simple and non intelligence ZI mechanism provides higher throughput compared to the ZIP and GD mechanisms.

The values of load/resource balancing presented in Table 8.1 show that λ mechanism provides a fair distribution of tasks/resources among pro-

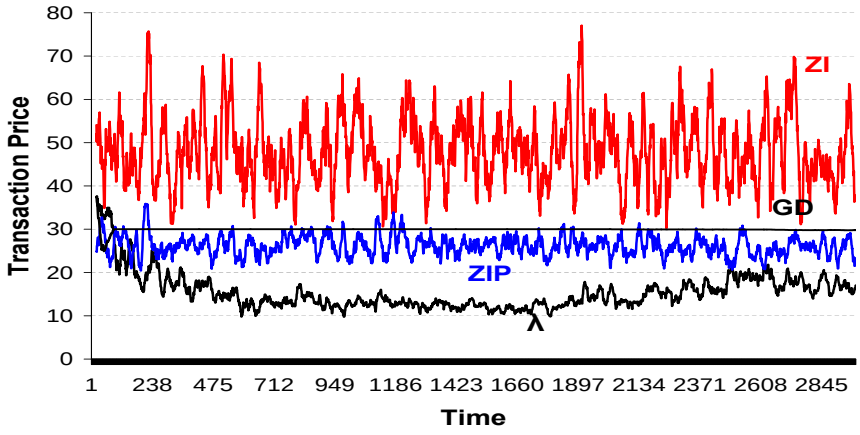

Figure 8.1: *Transaction prices in the homogeneous pricing for different pricing mechanisms in a balanced network.*

ducer/consumer agents compared to the other three mechanisms. Among ZI, ZIP, and GD mechanisms, ZI shows better performance compared to the other two mechanisms regarding load/resource balancing. ZIP and GD present a low level of load/resource balancing. The lowest level of load/resource balancing is observed in GD mechanism.

To explain load or resource balancing more clearly, we show resource distribution among the individual consumer agents by measuring task utilization for each consumer agent. Figure 8.2 depicts the task utilization for each of the 60 consumer agents. As it can be seen from the figure, the task utilization is almost equally distributed among different agents in λ mechanism. Task utilization of the agents varies between 82% and 99% in this mechanism. Among the other three mechanisms, ZI shows more balance in task utilizations compared to the ZIP and GD mechanisms. Variation in task utilization is between 62% and 91% in ZI, between 100% and 0% in ZIP, and between 87% and 0% in GD. We observe that some agents obtain a high and some others a zero task utilization in ZIP and GD mechanisms. This implies that these mechanisms do not provide a fair access to resource for consumer agents.

Deadline satisfactions for consumers and producers is correlated with task/resource utilization. When the deadline satisfaction is low, it means either some tasks/resources are matched very close to their deadline or they remain unmatched due to their deadline expiry. As the results show, our λ

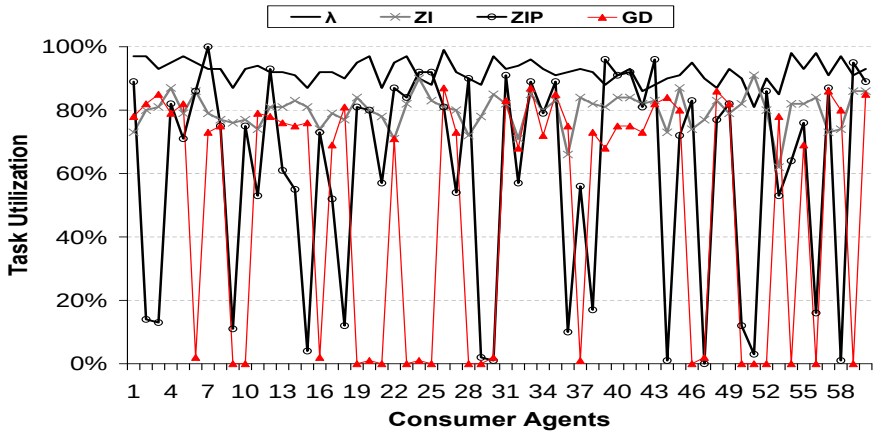

Figure 8.2: *Task utilizations at individual consumer agents in the homogeneous pricing for different pricing mechanisms in a balanced network.*

	Average TPrice	Price Volatility	Task Utilization	Resource Utilization	Resource Balancing	Load Balancing	Average cons. Deadline Satisf.	Average prod. Deadline Satisf.
λ	101	31.5	24%	95%	82%	94%	19%	94%
ZI	104	41.3	22%	90%	80%	93%	18%	90%
ZIP	74	21.9	18%	73%	20%	54%	15%	68%
GD	86	4.6	8%	31%	-36%	39%	8%	15%

Table 8.2: *Performance of different mechanisms in a homogenous pricing under a task intensive network condition.*

mechanism gives the highest satisfaction for both consumers and producers compared to the other three mechanisms.

Task/resource intensive network

Table 8.2 shows the results in a task intensive network. It can be observed that λ and ZI mechanisms perform much better in a task intensive network compared to ZIP and GD mechanisms regarding all criteria except for price volatility and average transaction price. Although ZIP and GD show lower price volatility and lower average transaction price, but they do not present a good throughput. It can be explained that in a competitive condition like task intensive network, the agents need to make more aggressive bids to get their

	Average TPrice	Price Volatility	Task Utilization	Resource Utilization	Resource Balancing	Load Balancing	Average cons. Deadline Satisf.	Average prod. Deadline Satisf.
λ	27	12.8	99%	26%	96%	81%	99%	18%
ZI	38	21.4	94%	23%	93%	82%	91%	19%
ZIP	40	15.9	84%	21%	60%	-67%	83%	17%
GD	41	19.1	95%	23%	90%	-58%	93%	17%

Table 8.3: *Performance of different mechanisms in a homogenous pricing under a resource intensive network condition.*

required resources. Therefore, the stable and low prices do not necessarily lead to a good performance in such a condition. However, ZIP mechanism performs better than GD in a task intensive condition. At the same time, λ mechanism slightly outperforms ZI mechanism. The results in Table 8.2 presents negative values for resource balancing in GD mechanism. Based on the definition of load/resource balancing (see Chapter 5, Section 5.5), a negative value for a resource or load balancing implies that the standard deviation of the resource or task utilization is greater than its mean value. This infers that in GD mechanism, distribution of resources among consumer agents is very uneven.

Table 8.3 shows the performance of the four pricing mechanisms in a resource intensive network. In this condition, λ, ZI, and GD perform more or less similar with respect to the task/resource utilization, resource balancing and consumer/producer deadline satisfaction while GD shows a low level of load balancing as compared to λ and ZI mechanisms. ZIP presents the lowest level of performance compared to the other three mechanisms regarding all criteria. Our λ mechanism shows the lowest price volatility and average transaction price among the four mechanisms.

Based on the observations in different network conditions, we can conclude that ZIP and GD mechanisms do not show good performance compared to ZI and λ mechanisms in a homogeneous pricing regarding the most evaluation criteria. Therefore, these pricing mechanisms are not good choices to be applied in an ad-hoc Grid. Among λ and ZI mechanisms, λ mechanism slightly outperform ZI mechanism.

	Task Utilization	Resource Utilization	Average consumer Deadline Satisf.	Average Producer Deadline Satisf.
λ	80%	98%	73%	95%
ZI	72%	67%	68%	61%
ZIP	72%	65%	64%	57%
GD	78%	71%	71%	64%

Table 8.4: *Performance of different mechanisms in a heterogenous pricing under a balanced network condition.*

	Task Utilization	Resource Utilization.	Average consumer Deadline Satisf.	Average Producer Deadline Satisf.
λ	25%	98%	20%	95%
ZI	24%	87%	20%	84%
ZIP	21%	92%	17%	86%
GD	23%	88%	18%	82%

Table 8.5: *Performance of different mechanisms in a heterogenous pricing under a task intensive network condition.*

8.6.2 Heterogeneous Pricing

In this section, we study the performance of the four pricing mechanisms in a heterogeneous pricing. In this case, different agents in the network apply different pricing mechanisms. To perform the experimental evaluations, we divided 60 nodes in the network to the four groups of 15 nodes where every group applies a different pricing mechanism. The performance of different mechanisms are compared based on task/resource utilization and consumer/producer deadline satisfaction. We measure the average of task/resource utilization and deadline satisfaction for each group of 15 nodes in three network conditions. Load/resource balancing in the network is affected by all pricing mechanisms in this case, and it can not show the performance of a particular mechanism in this regard. Therefore, we do not consider load/resource balancing criteria in case of the heterogeneous pricing.

	Task Utilization	Resource Utilization.	Average consumer Deadline Satisf.	Average Producer Deadline Satisf.
λ	99%	82%	98%	63%
ZI	99%	7%	99%	6%
ZIP	99%	4%	98%	3%
GD	99%	5%	98%	4%

Table 8.6: *Performance of different mechanisms in a heterogenous pricing under a resource intensive network condition.*

Balanced network

Table 8.4 shows the evaluation results in a balanced condition. It can be seen from the results that the consumer/producer agents that apply λ pricing mechanism, obtain the highest task and resource utilizations. The consumer/producer agents that use λ mechanism, also achieve the highest deadline satisfactions. Higher deadline satisfaction means that the agents have been more successful in meeting their deadline which corresponds to high task/resource utilizations. Among the other three mechanisms, GD performs slightly better compared to ZIP and ZI mechanisms in a balanced condition.

Task/resource intensive network

The results of experiments in the task and resource intensive networks are presented respectively in Table 8.5 and Table 8.6. In both task and resource intensive conditions, task utilization and consumer deadline satisfaction are more or less the same for different groups of consumer agents that apply different pricing mechanisms. Therefore, there is no prominent mechanism choice for consumers in these conditions. On the other hand, producer agents that employ λ mechanism obtain the highest resource utilization and deadline satisfaction as compared to the other mechanisms in both task and resource intensive conditions. λ mechanism favors producers specially in case of a resource intensive networks more than other three mechanisms.

Het	Average TPrice	Price Volatility	Task Utilization	Resource Utilization	Resource Balancing	Load Balancing	Average cons. Deadline Satisf.	Average prod. Deadline Satisf.
BN	27	19.3	75%	75%	77%	76%	69%	69%
TIN	75	29	23%	91%	42%	87%	19%	87%
RIN	33	17.5	99%	25%	98%	-38%	98%	19%

Hom λ	Average TPrice	Price Volatility	Task Utilization	Resource Utilization	Resource Balancing	Load Balancing	Average cons. Deadline Satisf.	Average prod. Deadline Satisf.
BN	15	6.7	92%	92%	96%	98%	87%	85%
TIN	101	31.5	24%	95%	82%	94%	19%	94%
RIN	27	12.8	99%	26%	96%	81%	99%	18%

Hom ZI	Average TPrice	Price Volatility	Task Utilization	Resource Utilization	Resource Balancing	Load Balancing	Average cons. Deadline Satisf.	Average prod. Deadline Satisf.
BN	47	34.2	80%	80%	93%	95%	75%	72%
TIN	104	41.3	22%	90%	80%	93%	18%	90%
RIN	38	21.4	94%	23%	93%	82%	91%	19%

Hom ZIP	Average TPrice	Price Volatility	Task Utilization	Resource Utilization	Resource Balancing	Load Balancing	Average cons. Deadline Satisf.	Average prod. Deadline Satisf.
BN	26	10.2	61%	61%	45%	30%	56%	56%
TIN	74	21.9	18%	73%	20%	54%	15%	68%
RIN	40	15.9	84%	21%	60%	-67%	83%	17%

Hom GD	Average TPrice	Price Volatility	Task Utilization	Resource Utilization	Resource Balancing	Load Balancing	Average cons. Deadline Satisf.	Average prod. Deadline Satisf.
BN	30	1.06	50%	50%	24%	4%	40%	45%
TIN	86	4.6	8%	31%	-36%	39%	8%	15%
RIN	41	19.1	95%	23%	90%	-58%	93%	17%

Table 8.7: *Performances of heterogenous pricing (Het) and homogeneous pricing (Hom, using λ, ZI, ZIP, and GD) under the Balanced Network (BN), Task Intensive Network (TIN), and Resource Intensive Network (RIN).*

8.7 Comparing Homogeneous & Heterogenous Pricings

In this section, we want to investigate which choice is better to make between homogeneous and heterogeneous pricing. We compare the performance of

the heterogeneous and homogeneous pricing based on the criteria already described in Section 8.5.1. To make the comparison easier, we repeated the results of homogeneous pricing in Section 8.6.1 for each mechanism in the separate sub-tables and presented them in Table 8.7. In the heterogeneous case, we consider the performance of the whole network rather than the individual mechanisms. Table 8.7 shows all the results under the three network conditions.

By looking at the results in Table 8.7, we can observe that a homogeneous pricing using λ mechanism outperforms all other cases in almost all aspects. Following right after the λ mechanism, ZI in a homogeneous pricing performs better than a heterogeneous pricing as well as the ZIP and GD mechanisms in a homogeneous pricing. A heterogeneous pricing outperforms a homogeneous pricing only when the homogeneous pricing applies ZIP or GD mechanism.

These observations seem to indicate that the best performance under any network condition is achieved when all agents apply λ mechanism. On the other hand, the results in Section 8.6.2 also showed that in case of heterogenous pricing, λ mechanism is the most efficient mechanism. Therefore, it can be concluded that λ mechanism is the best choice to be made whether the agents apply the same pricing mechanism or different agents apply different pricing mechanisms.

8.8 Conclusions

In this chapter, we discussed different pricing mechanisms for resource allocation in ad-hoc Grids. The performance of λ pricing mechanism was compared against that of the Zero Intelligence (ZI), Zero Intelligence Plus (ZIP), and Gjerstad Dickhaut (GD). The evaluation criteria are throughput, deadline satisfaction, load/resource balancing, and price volatility. We studied the performance of each pricing mechanism separately when all the agents in the network apply similar pricing mechanism. We named this situation as a homogeneous pricing. An overall view on the result shows that λ mechanism performs better than the other three mechanisms according to the most criteria. After λ mechanism, ZI is performing better than ZIP and GD in a homogeneous pricing. Regarding price volatility criterion, the most stable prices

in a balanced condition is provided by GD mechanism, but this mechanism does not show a good throughput in this condition. GD and ZIP mechanisms also show inefficiency considering load and resource balancing in most cases. This implies that these mechanisms are not able to provide a fair distribution of resources and tasks among consumer and producer agents in an ad-hoc Grid.

We also studied the performance of the pricing mechanisms in a heterogeneous pricing. In this situation different agents in the network apply different pricing mechanisms. In the case of a heterogeneous pricing, λ mechanism outperforms the other three mechanisms in any network condition. ZI, ZIP and GD perform similarly in this case.

By comparing a homogeneous pricing with a heterogeneous pricing, we observed that a homogeneous case when λ mechanism is used, outperforms a heterogeneous case regarding all criteria under any network condition. Considering all the results in the homogeneous and heterogeneous cases, we can conclude that λ mechanism is a good candidate for pricing mechanism in an ad-hoc Grid irrespective of which pricing mechanisms are used by others.

Chapter 9

Conclusions and Future Work

In this dissertation, we presented an economic framework to examine different market-based mechanisms for resource allocation in an ad-hoc Grid. The framework represents an ad-hoc Grid as a market with consumers and producers of resources as buyers and sellers. Consumers and producers of resources are considered to be self-interested agents that can enter the market at any time and request or offer a resource. The matchmaking between consumer and producer agents is done through an intermediate agent namely the matchmaker. By using market mechanisms for resource allocation in such an environment, aggregated utility of the entire network is obtained through the self-interested behavior of individual agents. Individual decision making is achieved by allowing individual nodes to make decisions with respect to resources on the basis of their local state. All the proposed transactions by the nodes in the network condense, in turn, all that information into a simple metric, the price. In this dissertation, we explored different pricing and matchmaking mechanisms that can be employed by consumer, producer, and matchmaker agents in an ad-hoc Grid. Although the use of market-based mechanisms is not novel in itself, we investigated in a systematic way how to construct such an infrastructure and what is the impact of certain choices.

9.1 Summary

The work presented in this thesis is summarized as follows:

In Chapter 2, we provided the necessary background regarding Grid and P2P computing. We studied resource discovery mechanisms that are based on peer-to-peer architectures. We explained how these architectures can be used for building resource allocation mechanisms in an ad-hoc Grid. Furthermore, we emphasized the advantages of using market-based mechanisms for resource allocation, and highlighted the shortcomings of the previous research.

In Chapter 3, we introduced the necessary economic concepts at both the micro and macro economics levels. At the micro level, the individual behavior and at the macro level, the system behavior were studied. We described two main classes of the economic mechanisms namely auctions and commodity markets. The economic mechanisms were compared considering the aspects that are important when implementing a resource allocation mechanism in an ad-hoc Grid such as capability of providing simultaneous participation of consumers and producers, policies in setting transaction prices, and the ease of the implementation.

In Chapter 4, we presented the economic framework. The main components of the framework (consumer, producer, and matchmaker), their attributes, and interaction between them were described in this chapter. Furthermore, we described our experimental platform that we used for all our experiments regarding resource allocation. We sketched the experimental settings, assumptions, and the network conditions in which the experiments were performed.

In Chapter 5, we studied different instances of the framework considering different matchmaking mechanisms. Three auction mechanisms namely First-price auction, Vickrey auction, and Continuous Double Auction (CDA) were implemented and compared. The performances of these mechanisms were evaluated under different network conditions based on the Grid user and owner objectives such as task/resource utilization, deadline satisfaction, consumer/producer surplus, and price volatility. The experimental results showed that CDA is the most efficient mechanism regarding the evaluation criteria under different network conditions. We also compared CDA with the non-market FCFS mechanism. The results of this comparison showed that CDA performs as good as a FCFS mechanism which does not have price constraint. Moreover, CDA provides a fair access to Grid resources and tasks when they

are scarce while FCFS shows a lower fairness in these conditions.

In Chapter 6, we proposed a learning and adaptive pricing mechanism. In this mechanism, called λ, the consumer/producer agents learn the condition of the network based on their previous utilization of the ad-hoc Grid and adjust their prices accordingly. We examined the pricing mechanism with varying parameters values in different network conditions. By modifying the utilization threshold, agents can change their satisfaction level, and in consequence their level of contribution or demanding for resources in the Grid. By increasing or decreasing the rate at which bid/ask prices are changing, agents can speed up or speed down their adaptation to the new network condition. We also observed that when resources are scarce, the bid prices and in consequence transaction prices may grow infinitely. To prevent infinite prices and in coherence with real markets, we applied the budget constraint to the pricing mechanism.

In Chapter 7, we studied the use of different kinds of bidding strategies and explained under what conditions they could be useful. The results obtained from studying pricing parameters in Chapter 6, led us to extend the λ pricing mechanism. In this mechanism, agents decide to change their bidding strategy to aggressive or conservative based on the way that their utilization from an ad-hoc Grid is evolving. By taking aggressive/conservative biding, agents increase/decrease the speed of changing their prices to adapt to the new condition. We studied the two bidding strategies in a dynamic network condition where the supply and demand for resources varies over the time. Consumer/producer agents become aggressive when their task/resource utilization from the Grid is decreasing. The faster decreasing the utilization, the more aggressive the agents are. In the same way, consumer/producer agents become conservative when their task/resource utilization from the Grid is increasing. The faster increasing their utilization, the less conservative the agents are. The consumer/producer agents are capable of changing their level of aggressiveness/conservativeness individually.

In Chapter 8, we compared different pricing mechanisms using our experimental framework and platform. Four pricing mechanisms were examined namely Zero Intelligence (ZI), Zero Intelligence Plus (ZIP), and Gjerstad Dickhaut (GD), and λ pricing mechanisms. We studied the performance of each pricing mechanism in a homogeneous pricing when all the agents in the network adopt the same pricing mechanism as well as in a heterogeneous pric-

ing when different agents in the network apply different pricing mechanisms. Task/resource utilization, throughput, deadline satisfaction, load/resource balancing, and price volatility were the criteria to compare the pricing mechanisms. The evaluation results showed that λ and ZI outperform other two mechanisms in a homogeneous pricing case. However, λ performs slightly better than ZI mechanism. GD and ZIP mechanisms show a low performance especially when a fair access to tasks/resources is concerned. In a heterogeneous pricing, λ mechanism outperforms the other three mechanisms under any network condition. ZI, ZIP, and GD perform more or less the same in this case. An overview of the results in the homogeneous and heterogeneous cases under different network conditions showed that λ mechanism is the best choice whether all agents in the network use the same pricing mechanism or they use different pricing mechanisms.

9.2 Contributions

The main contributions of this work are:

- We studied market-based resource allocation mechanisms in an ad-hoc Grid rather than a conventional Grid. Ad-hoc Grids are distinguished from conventional Grids in a sense that in ad-hoc Grids, neither resources nor tasks are dedicated and their availabilities vary during the time. This implies that any kind of central controller is difficult to implement due to the highly dynamic nature of the Grid.
- We presented an economic framework which allowed us to study different market-based mechanisms for resource allocation in ad-hoc Grids. We implemented different matchmaking and pricing mechanisms and compared their performances experimentally given various workloads. By this study, we discovered what are more suitable choices for Grid user/owner under different network conditions.
- We explored the market-based mechanisms that can deal with the dynamic and spontaneous nature of ad-hoc Grids. We studied different auctions as they can provide the necessary matchmaking mechanisms for resource allocation where multiple participants will submit simultaneously their transaction offers. Moreover, there is no global price for

all transaction requests in these mechanisms; rather there are multiple micro transactions, each with their own price/quantity. In this way, the individual's preferences can be presented by their own prices instead of a global price that has to take into account all preferences.

- We proposed a pricing mechanism that takes into account the past performance of the node in the market. Crucial is that no global but only local information is needed for any decision. This pricing mechanisms can adapt to the dynamic condition of the ad-hoc Grid. Adaptation is achieved by the individuals through adjusting their prices in function of the direction in which their utilization of the Grid (and thus supply and demand) is evolving.
- We demonstrated how the price information can be used to understand the global state of the Grid and how a node can adapt to that by changing its bidding strategy to optimize its local utility. For instance, an upward trend or downward trend of transaction prices detects an intensive condition respectively for tasks or resources. Therefore, one can understand what is the current condition by studying the direction in which the transaction prices are evolving.
- We suggested two aggressive and conservative bidding strategies for bidding agents in an ad-hoc Grid. By aggressive bidding strategy, the consumer/producer agents that have gained low utilization from the ad-hoc Grid, are encouraged to demand more with increasing/decreasing their bid/ask prices. The lower utilization from the ad-hoc Grid, the more aggressive bidding is concluded. On the other hand, the consumer/producer agents which have had sufficient utilization from the ad-hoc Grid, expect less by taking a conservative bidding strategy. The aggressive and conservative bidding strategies provide agents a fair access to Grid tasks/resources by encouraging them for more demanding or more contributing if they have not gained enough before.

9.3 Future Research Directions

We suggest the following directions for the future research:

- Self-organizing ad-hoc Grid: A self-organizing mechanism at the sys-

tem level can restructure the system architecture to make the system scalable. For instance, a single matchmaker limits scalability due to bottleneck associated with a single computer to respond to a large number of queries. As we discussed in Chapter 2, Section 2.6, a self-organizing architecture can address the problem of scalability. A scalable ad-hoc Grid is provided by promoting and demoting ordinary nodes to matchmakers or vise versa whenever the workload on the matchmaker(s) respectively increases or decreases. This could be done on the basis of, for instance, the cost of transaction between producers and consumers. In case of an overload of such requests, and following the market logic, the price for finding a match will increase. The matchmaker may be constructed in such a way that once this price goes above a certain level, another matchmaker is initiated. On the other hand, whenever the price goes again below a certain level, the reverse could happen and nodes are demoted to become again ordinary nodes. Self-organization can be also defined with introducing reconfigurable services which provide, for instance, hardware specialization in Grid.

- Secure ad-hoc Grid: In this thesis, we assumed that the agents are honest and there is no malicious activity in the network. However, this cannot always be guaranteed. In a market, buyers must trust that sellers will provide the services they advertise. On the other hand, sellers must trust that the buyers are able to pay for services. This is essential in an ad-hoc Grid where market-based mechanisms are applied for allocating resources. Consumers and producers of resources in the ad-hoc Grid must trust each other like buyers and sellers in a market. A way of providing security in the ad-hoc Grid can be through identifying malicious agents by matchmaker and preventing them from participating in the market.

- Ad-hoc Grid accounting service: In this thesis, we considered virtual money for trading between agents. One of the assumptions that we made was assigning a limited budget to each node. We also assumed no budget injection to the network during each simulation. An ad-hoc Grid accounting service still has to designed. This service will be responsible for managing payments, assigning budgets, and determining the legibility of the transactions.

- Advanced resource reservation: Resource reservation guarantees to obtain a resource in future. Advanced resource reservation in a market-based Grid can be done on the basis of price prediction mechanisms. A way to predict the price of a resource can be through studying the price volatility in a market. For instance, the less volatility in prices, the more confident the prediction is.

Bibliography

[1] Astrogrid, http://www.astrogrid.org.

[2] Birn (biomedical informatics research network), http://www.nbirn.net/.

[3] Climateprediction.net, http://www.climateprediction.net.

[4] Dutch grid, http://www.dutchgrid.nl.

[5] Enabling grids for e-science (egee), http://www.eu-egee.org.

[6] Esg (earth system grid), http://www.earthsystemgrid.org.

[7] The german grid initiative (d-grid), http://www.d-grid.de.

[8] Globus, http://www.globus.org.

[9] Gnutella2, http://www.gnutella2.com.

[10] Kazaa, http://www.kazaa.com.

[11] Morpheus, http://www.morpheus.com.

[12] Napster, http://www.napster.com.

[13] Ogf (open grid forum), http://www.gridforum.org.

[14] Seti@home, http://setiathome.ssl.berkeley.edu.

[15] David Abramson, Jon Giddy, and Lew Kotler. High performance parametric modeling with nimrod/g: Killer application for the global grid?

In *IPDPS '00: Proceedings of the 14th International Symposium on Parallel and Distributed Processing*, page 520, Washington, DC, USA, 2000. IEEE Computer Society.

[16] Kento Aida, Atsuko Takefusa, Hidemoto Nakada, Satoshi Matsuoka, Satoshi Sekiguchi, and Umpei Nagashima. Performance evaluation model for scheduling in global computing systems. *Int. J. High Perform. Comput. Appl.*, 14(3):268–279, 2000.

[17] Luc Onana Alima, Sameh El-Ansary, Per Brand, and Seif Haridi. Dks(n, k, f): a family of low communication, scalable and fault-tolerant infrastructures for p2p applications. In *Proceedings of Proceedings of the 3rd International Symposium on Cluster Computing and the Grid*, page 15, Tokyo, Japan, 2003.

[18] Jim Almond and Dave Snelling. Unicore: uniform access to supercomputing as an element of electronic commerce. *Future Gener. Comput. Syst.*, 15(5-6):539–548, 1999.

[19] Y. Amir, B. Awerbuch, and R. Borgstrom. The java market: Transforming the internet into a metacomputer, 1998.

[20] S. Androutsellis-Theotokis and D. Spinellis. A survey of peer-to-peer file sharing technologies. Technical report, Athens University of Economics and Business, 2002.

[21] Anthony J. Bagnall and Iain E. Toft. Zero intelligence plus and gjerstad-dickhaut agents for sealed bid auctions. In *Workshop on Trading Agent Design and Analysis*, pages 59–64, 2004.

[22] K. Bertels, N. Panchanathan, S. Vassiliadis, and B. Pourebrahimi. Centralized matchmaking for minimal agents. In *Proceedings of the Conference on Parallel and Distributed Computer Systems*, page 9, November 2004.

[23] Sharon Brunett, Dan Davis, Thomas Gottschalk, Paul Messina, and Carl Kesselman. Implementing distributed synthetic forces simulations in metacomputing environments. In *HCW '98: Proceedings of the Seventh Heterogeneous Computing Workshop*, page 29, Washington, DC, USA, 1998. IEEE Computer Society.

[24] Murshed M. Buyya, R. Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. *Concurrency and Computation: Practice and Experience*, 14(13-15):1175–1220, 2002.

[25] R. Buyya, D. Abramson, and J. Giddy. Nimrod/g: an architecture for a resource management and scheduling system in a global computational grid. In *Proceedings of The Fourth International Conference on High Performance Computing in Asia-Pacific Region (HPC Asia 2000)*, USA, 2000. IEEE Computer Society Press.

[26] R. Buyya, D. Abramson, Jonathan Giddy, and Heinz Stockinger. Economic models for resource management and scheduling in grid computing. *Concurrency and Computation: Practice and Experience*, 14(13-15):1507–1542, 2002.

[27] R. Buyya, D. Abramson, and S. Venugopal. The grid economy. *Special Issue on Grid Computing, Proceedings of the IEEE*, 93:698–714, 2005.

[28] Rajkumar Buyya, David Abramson, and Jonathan Giddy. An economy driven resource management architecture for global computational power grids. In *PDPTA*, 2000.

[29] Junwei Cao, Daniel P. Spooner, Stephen A. Jarvis, and Graham R. Nudd. Grid load balancing using intelligent agents. *Future Gener. Comput. Syst.*, 21(1):135–149, 2005.

[30] H. Casanova and J. Dongarra. Netsolve: A network server for solving computational science problems. Technical report, University of Tennessee, Knoxville, TN, USA, 1995.

[31] Shang-Wen Cheng, David Garlan, Bradley Schmerl, Peter Steenkiste, and Ningning Hu. Software architecture-based adaptation for grid computing. In *Proceedings of the 11th IEEE Int. Symposium on High Performance Distributed Computing*, page 389, Washington, DC, USA, 2002. IEEE Computer Society.

[32] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science*, 2009:46+, 2001.

[33] Scott H. Clearwater, editor. *Market-based control: a paradigm for distributed resource allocation*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1996.

[34] Bruten J. Cliff, D. Zero is not enough: On the lower limit of agent intelligence for continuous double auction markets. Technical Report 97-141, HP, 1997.

[35] D. Cliff. Minimal-intelligence agents for bargaining behaviors in market-based environments. Technical Report 97-91, HP, 1997.

[36] B. F. Cooper and H. Garcia-Molina. Ad hoc, self-supervising peer-to-peer search networks. Technical report, Computer Science Dept., Stanford University, 2003.

[37] M. Dias de Assuncao and R. Buyya. An evaluation of communication demand of auction protocols in grid environments. In *Proceedings of the 3rd International Workshop on Grid Economics & Business (GECON 2006)*. World Scientific Press, May 2006.

[38] G. Debreu. *Theory of Value*. Yale University Press, 1959.

[39] Y. Yemini D.F. Ferguson, J. Sairamesh and C. Nikolaou. Economic models for allocating resources in computer systems. In Scott H. Clearwater, editor, *Market-Based Control*, pages 156–183. World Scientific Publishing Co. Pte. Ltd., 1996.

[40] M. Fogiel. *The Economics problem solver*. Research and Education Association, 1980.

[41] Ian Foster and Adriana Iamnitchi. On death, taxes, and the convergence of peer-to-peer and grid computing. In *2nd International Workshop on Peer-to-Peer Systems (IPTPS'03), February 2003, Berkeley, CA*, 2003.

[42] Luis Garces-Erice, Ernst W. Biersack, Keith W. Ross, Pascal A. Felber, and Guillaume Urvoy-Keller. Hierarchical p2p systems. In *Proceedings of ACM/IFIP International Conference on Parallel and Distributed Computing (Euro-Par)*, Klagenfurt, Austria, 2003.

[43] Steven Gjerstad and John Dickhaut. Price formation in double auctions. *Games and Economic Behavior*, 22:1–29, 1998.

[44] D.K. Gode and S. Sunder. Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality. *The Journal of Political Economy*, 101(1):119–137, 1993.

[45] Jacek Gomoluch and Michael Schroeder. Market-based resource allocation for grid computing: A model and simulation. In *Proceedings of the First International Workshop on Middleware for Grid Computing. Rio de*, pages 211–218, 2003.

[46] Nathaniel S. Good and Aaron Krekelberg. Usability and privacy: a study of kazaa p2p file-sharing. In *CHI '03: Proceedings of the conference on Human factors in computing systems*, pages 137–144. ACM Press, 2003.

[47] Anastasios Gounaris, Norman W. Paton, Rizos Sakellariou, Alvaro A. A. Fernandes, Jim Smith, and Paul Watson. Practical adaptation to changing resources in grid query processing. In *ICDE '06: Proceedings of the 22nd Int. Conf. on Data Engineering*, page 165, USA, 2006. IEEE Computer Society.

[48] D. Grosu and A. Das. Auction-based resource allocation protocols in grids. In *Proceedings of the 16th IASTED International Conference on Parallel and Distributed Computing and Systems*, pages 20–27, November 2004.

[49] Galen Gruman and Eric Knorr. What cloud computing really means. In *InfoWorld, http://www.infoworld.com/d/cloud-computing/what-cloud-computing-really-means-031?page=0,1*.

[50] G M Heal. Planning without prices. *Review of Economic Studies*, 36(107):347–62, 1969.

[51] Gernot Heiser, Fondy Lam, and Stephen Russell. Resource management in the mungi single-address-space operating system. In *In Proceedings of the 21 st Australasian Computer Science Conference (ACSC)*, pages 417–428, Perth, Australia, February 1998.

[52] Leonid Hurwicz. The design of mechanisms for resource allocation. *American Economic Review*, 63(2):1–30, 1973. available at http://ideas.repec.org/a/aea/aecrev/v63y1973i2p1-30.html.

[53] Adriana Iamnitchi, Ian Foster, and Daniel C. Nurmi. A peer-to-peer approach to resource location in grid environments. In *HPDC '02: Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing*, page 419, Washington, DC, USA, 2002. IEEE Computer Society.

[54] Pfaendtner J., S. Bloom, D. Lamich, M. Seablom, M. Sienkiewicz, J. Stobie, and A. da Silva. Documentation of the goddard earth observing system (geos) data assimilation system ? version 1, 1995.

[55] J. Miller J. Rust and R. Palmer. Behavior of trading automata in a computerized double auction market. *The Double Auction Market: Institutions, Theories, and Evidenc*, 1991.

[56] S. J. Rassenti K. A. McCabe and V. L. Smith. Auction institutional design: Theory and behavior of simultaneous multiple-unit generalizations of dutch and english auctions. *American Economic Review*, 80(5):1276–1283, 1990.

[57] U. Kant and D. Grosu. Double auction protocols for resource allocation in grids. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)*, pages 366–371, 2005.

[58] Mandar Kelaskar, Vincent Matossian, Preeti Mehra, Dennis Paul, and Manish Parashar. A study of discovery mechanisms for peer-to-peer applications. In *CCGRID*, pages 444–445, 2002.

[59] J. F. Kurose and R. Simha. A microeconomic approach to optimal resource allocation in distributed computer systems. *IEEE Transactions on computers*, pages 705–717, May 1989.

[60] K. Kuwabara and T. Ishida. Equilibratory approach to distributed resource allocation: Toward coordinated balancing. In *Proceedings. Artificial Sociality; MAAMAW'92, Lecture Notes in Artificial Intelligence 830, Springer-Verlag*, pages 133–146. 1994.

[61] Kevin Lai, Bernardo A. Huberman, and Leslie R. Fine. Tycoon: A distributed market-based resource allocation system. *CoRR*, cs.DC/0404013, 2004.

[62] David Laidler. *Introduction to Microeconomics*. Philip Allan, 1974.

[63] Spyros Lalis and Alexandros Karipidis. Jaws: An open market-based framework for distributed computing over the internet. In *GRID '00: Proceedings of the First IEEE/ACM International Workshop on Grid Computing*, pages 36–46, London, UK, 2000.

[64] Ulrike Lechner. Peer-to-peer beyond file sharing. In *IICS '02: Proceedings of the Second International Workshop on Innovative Internet Computing Systems*, pages 229–249, London, UK, 2002. Springer-Verlag.

[65] Michael Litzkow, Miron Livny, and Matthew Mutka. Condor - a hunter of idle workstations. In *Proceedings of the 8th International Conference of Distributed Computing Systems*, June 1988.

[66] L.Walras. *Elements of pure economics; or, the theory of social wealth*. Allen and Unwin, 1954.

[67] Moreno Marzolla, Matteo Mordacchini, and Salvatore Orl. Resource discovery in a dynamic grid environment. In *Proceedings DEXA05*, pages 256–260. IEEE Press, 2005.

[68] Carlo Mastroianni, Domenico Talia, and Oreste Verta. A super-peer model for building resource discovery services in grids: Design and simulation analysis. In *EGC*, pages 132–143, 2005.

[69] Peter Minowitz. Adam smiths invisible hands. *Econ Journal Watch*, 1(3):381–412, December 2004.

[70] N. Nisan, S. London, O. Regev, and N. Camiel. Globally distributed computation over the internet - the popcorn project. In *ICDCS '98: Proceedings of the The 18th International Conference on Distributed Computing Systems*, page 592. IEEE Computer Society, 1998.

[71] Michael L. Norman, Peter Beckman, Greg Bryan, John Dubinski, Dennis Gannon, Lars Hernquist, Kete Keahey, Jeremiah P. Ostriker, John Shalf, Joel Welling, and Shelby Yang. Galaxies collide on the I-WAY: An example of heterogeneous wide-area collaborative supercomputing. 10(2/3):132–144, Summer/Fall 1996.

[72] B Pourebrahimi, K.L.M. Bertels, and S. Vassiliadis. A survey of peer-to-peer networks. In *Proceedings of the 16th Annual Workshop on Circuits, Systems and Signal Processing, ProRisc 2005*, November 2005.

[73] M. Preist C., Van Tol. Adaptive agents in a persistent shout double auction. In *Proceedings of 1st International Conference on the Internet Computing and Economics*, pages 11–17, 1998.

[74] Diego Puppin, Stefano Moncelli, Ranieri Baraglia, Nicola Tonellotto, and Fabrizio Silvestri. A grid information service based on peer-to-peer. In *Euro-Par 2005 Parallel Processing*, pages 454–464, 2005.

[75] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Schenker. A scalable content-addressable network. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 161–172, New York, NY, USA, 2001. ACM.

[76] Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, 2001.

[77] Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proceedings of Multimedia Computing and Networking 2002 (MMCN '02)*, San Jose, CA, USA, January 2002.

[78] M.A. Satterthwaite and S.R. Williams. The Bayesian theory of the k-double auction. *The Double Auction Market: Institutions, Theories and Evidence, Santa Fe Institute Studies in the Sciences of Complexity*, pages 99–123.

[79] B. Schnizler. Resource allocation in the grid - a market engineering approach. Technical report, PhD thesis, university of Karlsruh, 2007.

[80] Björn Schnizler and Dirk Neumann. Combinatorial exchanges for coordinating grid services. *SIGecom Exch.*, 7(1):65–68, 2007.

[81] Rüdiger Schollmeier. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In *Peer-to-Peer Computing*, pages 101–102, 2001.

[82] Jahanzeb Sherwani, Nosheen Ali, Nausheen Lotia, Zahra Hayat, and Rajkumar Buyya. Libra: An economy driven job scheduling system for clusters. In *Proceedings of the 6th Intl. Conf. on High Performance Computing in AsiaPacific Region (HPC Asia 2002*, pages 16–19, 2002.

[83] Ken'ichiro Shirose, Satoshi Matsuoka, Hidemoto Nakada, and Hirotaka Ogawa. Autonomous con.guration of grid monitoring systems. In *Proceedings of the 2004 Symposium on Applications and the Internet-Workshops*, page 651, USA, 2004. IEEE Computer Society.

[84] M. Shubik. *Game Theory in Social Sciences*. The MIT Press, Cambridge, Mass, 1983.

[85] H. J. Song, X. Liu, D. Jakobsen, R. Bhagwan, X. Zhang, K. Taura, and A. Chien. The microgrid: A scientific tool for modeling computational grids. *Sci. Program.*, 8(3):127–141, 2000.

[86] Ion Stoica, Robert Morris, David Karger, Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable Peer-To-Peer lookup service for internet applications. In *Proceedings of the 2001 ACM SIGCOMM Conference*, pages 149–160, 2001.

[87] Michael Stonebraker, Paul M. Aoki, Witold Litwin, Avi Pfeffer, Adam Sah, Jeff Sidell, Carl Staelin, and Andrew Yu. Mariposa: a wide-area distributed database system. *The VLDB Journal*, 5(1):048–063, 1996.

[88] In suk Kim, Yong hyeog Kang, and Young Ik Eom. An efficient contents discovery mechanism in pure p2p environments. In *GCC (1)*, pages 420–427, 2003.

[89] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[90] Gerald Tesauro and Rajarshi Das. High-performance bidding agents for the continuous double auction. In *EC '01: Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 206–209, New York, NY, USA, 2001. ACM.

[91] P. Trunfio, D. Talia, H. Papadakis, P. Fragopoulou, M. Mordacchini, M. Pennanen, K. Popov, V. Vlassov, and S. Haridi. Peer-to-peer resource discovery in grids: Models and systems. *Future Gener. Comput. Syst.*, 23(7):864–878, 2007.

[92] Dimitrios Tsoumakos and Nick Roussopoulos. A comparison of peer-to-peer search methods. In *WebDB*, pages 61–66, 2003.

[93] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

[94] Carl A. Waldspurger, Tad Hogg, Bernardo A. Huberman, Jeffrey O. Kephart, and W. Scott Stornetta. Spawn: A distributed computational economy. *Software Engineering*, 18(2):103–117, 1992.

[95] Carl A. Waldspurger and William E. Weihl. Lottery scheduling: Flexible proportional-share resource management. In *Operating Systems Design and Implementation*, pages 1–11, 1994.

[96] Chuliang Weng, Xinda Lu, Guangtao Xue, Qianni Deng, and Minglu Li. A double auction mechanism for resource allocation on grid computing systems. In *GCC*, page 269, 2004.

[97] R. Wolski, J. Brevik, J. S. Plank, and T. Bryan. Grid resource allocation and control using computational economies. In *Grid Computing: Making The Global Infrastructure a Reality*, pages 747–772. John Wiley & Sons, 2003.

[98] Rich Wolski, James Plank, John Brevik, and Todd Bryan. G-commerce: Market formulations controlling resource allocation on the computational grid. In *Proceedings International parallel and Distributed Processing Symposium (IPDPS)*, April 2001.

[99] P. Wurman, W. Walsh, and M. Wellman. Flexible double auctions for electronic commerce: Theory and implementation. *Decision Support Systems*, 24:17–27, 1998.

[100] Beverly Yang and Hector Garcia-Molina. Comparing hybrid peer-to-peer systems. In *The VLDB Journal*, pages 561–570, sep 2001.

[101] Beverly Yang and Hector Garcia-Molina. Improving search in peer-to-peer networks. In *ICDCS '02: Proceedings of the 22 nd International*

Conference on Distributed Computing Systems (ICDCS'02), page 5. IEEE Computer Society, 2002.

[102] Y. Yemini. Selfish optimization in computer networks. In *Proceeding of the 20th IEEE Conf. on Decision and Control*, pages 281–285, San Diego, December 1981.

[103] Y. Yemini and L.Kleinrock. On a general rule for access control, or silence is golden. In *Proceeding of the International Conf. on Flow Control in Computer Networks*, pages 335–347, North Holland Press Amsterdam, Versailles, February 1979.

List of Publications

International Conferences

1. T. Abdullah, L. Mhamdi, **B. Pourebrahimi**, K.L.M. Bertels, "Resource Discovery with Dynamic Matchmakers in Ad Hoc Grid", The Fourth International Conference on Systems, March 2009.

2. **B. Pourebrahimi**, K.L.M. Bertels, "Auction Protocols for Resource Allocations in Ad-hoc Grids", In Proceedings of 14th International Euro-Par Conference, pp. 520-533, Las Palmas de Gran Canaria, Spain, August 2008.

3. **B. Pourebrahimi**, K.L.M. Bertels,"Adaptation to Dynamic Resource Availability in Ad-hoc Grids through a Learning Mechanism", CSE08: Proceedings of the 2008 11th IEEE International Conference on Computational Science and Engineering, pp. 171-178, Sao Paulo, Brazil, July 2008.

4. T. Abdullah, V. Sokolov, **B. Pourebrahimi**, K.L.M. Bertels, "Self-Organizing Dynamic Ad Hoc Grids", In Proceedings of 2nd IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops (SASOW 2008), Venice, October 2008.

5. **B. Pourebrahimi**, L.O. Alima, K.L.M. Bertels, "Market Formulation for Resources Allocation in an Ad-hoc Grid", In Proceedings of 2nd IEEE International Conference on Self-Adaptive and Self-Organizing

Systems Workshops (SASOW 2008), Venice, October 2008.

6. **B. Pourebrahimi**, K.L.M. Bertels, S. Vassiliadis, L.O. Alima, "A Dynamic Pricing and Bidding Strategy for Autonomous Agents in Grids", In Proceedings of Sixth International Workshop on Agents and Peer-to-Peer Computing (AP2PC 2007), Honolulu, Hawaii, May 2007.

7. **B. Pourebrahimi**, S. A. Ostadzadeh, K.L.M. Bertels, "Resource Allocation in Market-based Grids Using a History-based Pricing Mechanism", In Proceedings of SCSS (1) 2007, pp. 97-100, December 2007.

8. **B. Pourebrahimi**, K.L.M. Bertels, "Fair Access to Scarce Resources in Ad-hoc Grids using an Economic-based Approach", MGC07: Proceedings of the 5th international workshop on Middleware for grid computing, pp. 1-5, Newport Beach, California, November 2007.

9. **B. Pourebrahimi**, K.L.M. Bertels, G Kandru, S. Vassiliadis, "Market-based Resource Allocation in Grids", In Proceedings of second IEEE International Conference on e-Science and Grid Computing, pp. 80, Amsterdam, The Netherlands, December 2006.

10. K Sigdel, K.L.M. Bertels, **B. Pourebrahimi**, S. Vassiliadis, L.S Shuai, "A framework for Adaptive Matchmaking in Distributed Computing", In Proceeding of GRID Workshop Cracow-04, Cracow, Poland, December 2004 .

11. K.L.M. Bertels, N. Panchanathan, S. Vassiliadis, **B. Pourebrahimi**, "Centralized Matchmaking for Minimal Agents", In Proceedings of the Conference on Parallel and Distributed Computer Systems, pp. 9, November 2004.

Local Conferences

1. **B. Pourebrahimi**, K.L.M. Bertels, "Matchmaking through Economic-based Approaches in Ad-hoc Grids", In Proceedings of the 18th Annual Workshop on Circuits, Systems and Signal Processing (ProRisc 2007), Nov. 2007, The Netherlands.

2. **B. Pourebrahimi**, K.L.M. Bertels, S. Vassiliadis, "A Survey of Peer-to-Peer Networks", In Proceedings of the 16th Annual Workshop on Circuits, Systems and Signal Processing (ProRISC2005), Nov. 2005, The Netherlands.

3. K Sigdel, Shuai, **B. Pourebrahimi**, K.L.M. Bertels, S. Vassiliadis, "Centralized Matchmaking, An Empirical Study", In Proceedings of the 16th Annual Workshop on Circuits, Systems and Signal Processing (ProRISC2005), Nov. 2005, The Netherlands.

4. **B. Pourebrahimi**, K.L.M. Bertels, S. Vassiliadis, K Sigdel, "Matchmaking within Multi-Agent Systems", In Proceedings of the 15th Annual Workshop on Circuits, Systems and Signal Processing (ProRISC2004), Nov. 2004, The Netherlands.

Non-related Publications

1. **B. Pourebrahimi**, J.C.A. Lubbe, "A Novel Approach for Noise Reduction in the Gabor Time-frequency Domain", In Proceedings of the Fourth International Conference, on Computer Vision Theory and Applications (VISAPP 2009), Lisboa, Portugal, vol. 2, pp. 22-27, February 2009.

2. **B. Pourebrahimi**, "Edge Detection using Gabor Coefficients through Neural Network", ASCI 2004 conference, Port Zelande, The Netherlands, June 2004.

Samenvatting

In deze dissertatie presenteren we een economisch raamwerk om verschillende op-markt-gebaseerde mechanismen voor resource allocation in een ad-hoc grid te bestuderen en te ontwikkelen. Een dergelijk economisch raamwerk helpt het effect van bepaalde keuzes te begrijpen en verkent welke mechanismen toepasselijk zijn onder bepaalde omstandigheden vanuit het perspectief van de gebruiker/eigenaar. We richten ons op resource allocation in een omgeving gebaseerd op grids in het geval dat enige resources ongebruikt blijven en verbonden zouden kunnen worden met overbelaste knopen in een netwerk. In dergelijke netwerken zijn resources noch noodzakelijkerwijs gereserveerd, noch hebben zij op ieder moment een voorspelbare beschikbaarheid. We noemen zulke netwerken ad-hoc grids. Zelfzuchtige knopen in ad-hoc grids worden beschouwd als consumenten (kopers) en producenten (verkopers) van resources binnen het economische raamwerk. Consumenten en producenten van resources zijn zelfstandige agenten die samenwerken door een enkele eenvoudige metriek, namelijk de prijs die de globale status van een netwerk samenvat in n waarde. De prijs vertegenwoordigd alle beschikbare informatie die aanwezig is op het niveau van de individuele knopen en niet noodzakelijk onderling gedeeld wordt. Een ondersteunende agent, genaamd de koppelaar, brengt een wederzijdse overeenkomst tot stand tussen de consument- en producent-agenten op basis van de prijs door het toepassen van economische mechanismen zoals veilingen. De transactie is voltooid wanneer aan de voorwaarden van de consument- en producentagent - zoals hoeveelheid, tijd, en budget - is voldaan. In deze dissertatie bestuderen we op-markt-gebaseerde mechanismen voor resource allocation op macro- en microniveau. Macro-economie behandelt het gedrag van een economie op samengesteld niveau en micro-economie beschrijft het individuele gedrag. Op het macroniveau vergelijken we verschillende economische modellen als koppelmechanismen. We onderzoeken welk effect de keuze van bepaalde veilingsmechanismen heeft in het raamwerk. Op het microniveau bestuderen we verschillende prijsmechanismen en gaan we na wat het effect is van het introduceren van geld- en budgetvoorwaarden. Daarenboven analyseren we verschillende biedstrategien die agenten helpen hun doelstellingen te verwezenlijken onder wisselende voorwaarden.

Acknowledgments

During the years that I was doing my Ph.D. in Computer Engineering group, I was fortunate to come across many people whom I benefited from their help, encouragement, and criticism. I express my sincere gratitude to all of them.

First, I wish to acknowledge my supervisor, Dr. Koen Bertels, for his support and guidance through the hardest challenges I faced in the course of my research, and for his inputs which influenced considerably the writing of this dissertation. Over the years, I have come to know and appreciate Koen's gentle and generous attitude, alongside his immense scientific knowledge.

I would like to express my sincere gratitude to late Prof. Stamatis Vassiliadis for giving me the opportunity to study in the Computer Engineering group. It was a privilege knowing him and his memory will stay in my heart forever.

I am thankful to Prof. Kees Goossens as my promoter, and I am grateful for his valuable comments on this dissertation. A special thank you to Dr. Luc Onana Alima for his wonderful ideas, and the useful discussions that we had. I wish to thank Dr. Jan van der Lubbe, the supervisor of my current research group, for his kind understanding at the time I was busy with writing this dissertation and unable to devote my full attention towards my current research work.

During my Ph.D. research, I was surrounded by many wonderful people. And although it is not possible for me to name all of them here, I wish to thank them for their friendship and support. I received help from several people while editing this dissertation, and to them I extend my appreciation. I would

like to thank Tariq Abdullah, my colleague in this project, for all the fruitful discussions we had, and for his comments on our research. It was an enjoyable experience working with him. I wish to thank Mahesh Kandru and Li Shui, master's students with whom I worked during the preliminary phase of this project. My warmest thanks to Kamana Sigdel for being such a wonderful friend and colleague, and for being supportive over these years. I enjoyed working with her during her master's thesis as well. I wish to express my sincere thanks to Sandra Irobi for reading my thesis. I am also thankful to Roel Meeuws for translating the abstract and propositions into Dutch.

Many thanks are due to Arash Ostadzadeh and Mojtaba Sabeghi for their valuable suggestions, discussions, and comments regarding the work, and for their support in different issues. I would also like to thank my colleagues and friends Elena Moscu Panainte, Faisal Nadeem, Asad Shahbahrami, Carlo Galuzzi, and Yana Yankova for all their support and open discussions. I am grateful to Lidwina Tromp for her excellent administration. Many thanks to Bert Meijs for his technical support whenever I was in need.

I consider myself lucky to be in a university with many Iranian students around. I am grateful to all my friends for being there when I needed them the most during my stay in the Netherlands. A special thank you goes to Azadeh Firouzian for her valuable friendship during these years. I would also like to thank Mina Danesh, Nasrin Esfandiari, Azadeh Arjomand, Jawad Qureshi, Pedram Khalili, Omid Noroozian, and all my Iranian friends in Delft.

Last but certainly not least, I would like to thank my family for all their support and encouragement. I am deeply indebted to my sister Sharareh for her immense support and care during all these years. I would also like to thank Mansoureh and my brother-in-law Reza for being there whenever I needed help. I sincerely thank my sisters Mahnaz, Minoo, Nahid and my brothers Hamid and Reza for all their kindness, support and encouragement. I honor the memory of my dear sister, Shahin. Many thanks to my dears Golnoosh, Parastou and Arash, and my nephews and nieces in Iran for bringing hope and cheer into my life in all these years that I was far from home. I am forever indebted to my mother and my father for their unconditional love and endless support. This dissertation is dedicated to the memory of my parents.

Delft,
November 2009

Behnaz Pourebrahimi

About the Author

Behnaz Pourebrahimi was born in Kerman, Iran on 21^{st} of July 1970. In 1993, she received a B.Sc. degree in Computer Engineering (Hardware) from Amirkabir University of Technology, Tehran, Iran. She worked as a software programmer for three years until 1996. In 1999, she received her M.Sc. degree in Machine Intelligence and Robotics from Shiraz university, Shiraz, Iran. During the last year of her M.Sc. studies, she was working at Azad university of Kerman as a part time lecturer. After her graduation, she was appointed as a full time faculty member in Azad University of Kerman. She worked there for four years. She was also a part-time lecturer in Bahonar University of Kerman at the same time.

In 2004, she joined the Computer Engineering group in Delft University of Technology, the Netherlands as a Ph.D. candidate. She conducted her Ph.D research under the supervision of Dr. Koen Bertels. Since November 2007, she has been working as a research associate in the Information and Communication Theory (ICT) group in the field of image processing at the same university.