# Service Negotiation and Contracting in Virtual Network Environment

by

Fida-E Zaheer

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2010

## AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Fida-E Zaheer

# Abstract

The current Internet presents a high barrier to entry for new service providers, due to its inability to accommodate new protocols and technologies, and lack of competition among the network providers. Recently, network virtualization has gained considerable attention as a possible solution, as it enables multiple networks to concurrently run over a shared substrate. It allows for deploying diverse network protocols and technologies customized for specific networked services and applications. Moreover, any party can take on the role of a network provider by simply offering his virtual network infrastructure to customers, increasing competition in the market. However, the first challenge in realizing a fair and competitive market in a virtual network environment is to have a service negotiation and contracting mechanism in place, that will allow (i) multiple infrastructure providers to participate in a fair and faithful competition, and (ii) a service provider to negotiate the price and quality of service with the providers.

In this thesis, we present V-Mart, an open market model and enabling framework for automated service negotiation and contracting in a virtual network environment. To the infrastructure providers, V-Mart fosters an open and fair competition realized by a two stage auction. The V-Mart auction model ensures that bidders (infrastructure providers) bid truthfully, have the flexibility to apply diverse pricing policies, and still gain profit from hosting customers' virtual resources. To the service providers, V-Mart offers virtual network partitioning algorithms that allow them to divide their virtual networks among competing infrastructure providers while minimizing the total cost. V-Mart offers two types of algorithms to suit different market scenarios. The algorithms not only consider virtual resource hosting price but also the service provider's preference for resource co-location and the high cost of inter-provider communication. Through extensive simulation experiments we show the efficiency and effectiveness of the algorithms under various market conditions.

# Acknowledgements

First and foremost I thank Allah for giving me the strength to complete this task.

I express profound gratitude to my supervisor, Professor Raouf Boutaba. He has been a true mentor and a major source of support, encouragement and courage over the last couple of years. I also thank the readers of this thesis, Professor Ken Salem and Professor Richard Trefler, for their invaluable feedbacks.

I am most grateful to my parents; but for their tremendous support and inspiration, I would not have been to able to pursue this, or any of my endeavors.

I am immensely grateful to my elder brother, Shama, without whose sacrifice and support I could not have completed this task. I cannot appreciate my wife, Rizwana, enough for constantly being there to support and inspire me. I, also, thank my brother, Naba, for his constant encouragement.

I convey my deep gratitude to Jin, my good friend and collaborator on the *V-Mart* project, for his continuous cooperation and invaluable inputs. I am greatly appreciative of Mosharaf for taking me on board the *iMark* project in my first term. I am also honored to be able to work with Qi, Muntasir and all the members of *NouVeau: University of Waterloo Network Virtualization Project*. I am thankful to all my friends and colleagues.

Finally, I would like to thank the University of Waterloo for providing me financial, technical support, and, most importantly, an excellent academic atmosphere throughout the program.

*To my father, my biggest inspiration in life.*

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Although the Internet has been stunningly successful from its inception, its architecture and business model pose a high barrier to entry for new and innovative service providers. The Internet architecture, developed decades ago, has proven its worth by the wide variety of applications that run on it and the heterogeneity of technologies over which it currently runs. Nonetheless, many applications and services often find the architecture ill-suited for their purposes [15, 57, 30], while some others could benefit from having more control over the underlying architecture parameters (e.g, packet formats, routing protocols, forwarding mechanisms and other control and management protocols). But, such flexibility and control is hard to imagine. Changes to the Internet architecture are limited to mere makeshift solutions and patches to temporarily handle problems. Anything more disruptive in nature is next to impossible, as it requires a consensus among multiple stakeholders with different goals and policies. This is evident from the painstakingly slow and still incomplete deployment of IPv6.

From a business perspective, a major concern is the lack of competition among the network providers. The role of network provider in the Internet has become the prerogative of a handful of large companies with big pockets. Service providers have no option but to select from pre-specified levels of services at a price fixed by the providers. But, ideally, a service provider would want to negotiate with the network providers for a price based on the demand, the utility he will derive from it, priority and available budget. Thus, virtual network hosting service should be rendered at a price that maintains a balance between the desires of the service provider and the network or infrastructure providers.

Network virtualization has gained considerable attention [15, 57, 28] as a possible solution for this stalemate, as it provides the means to concurrently run multiple virtual networks, each customized for a particular use, on a shared physical substrate. In a Virtual Network Environment (VNE), the basic entity is a virtual network (VN), which is a logical topology composed of virtual nodes and virtual links. Provisioning a VN involves the mapping/embedding of virtual nodes onto physical ones and virtual links onto physical links or paths. Once provisioned, a VN has the semblance of an actual physical network. The key to a VNE's flexibility is the splitting of the traditional Internet Service Provider (ISP)'s role into two: *infrastructure providers (InPs)*, who are responsible for deploying and managing the substrate networks, i.e, the underlying physical routers and links, and *service providers (SPs)*, who synthesize virtual networks by aggregating resources from multiple infrastructure providers and use them to deploy their services. The decoupling of the ISP's role enables service providers to deploy a customized network for his services without the need to build his own expensive physical infrastructure. Moreover, the role of an InP is no longer the prerogative of expensive infrastructure owners; a SP can serve as an InP and lease out virtual resources spawned from his own virtual network. Thus, a VNE offers an overall open and competitive market, where service providers have a wide variety of infrastructure providers to lease virtual networks from, while enjoying the flexibility to customize their virtual networks to best suit their services.

Many research projects have acknowledged the need for virtual network environments [56, 51, 3, 6, 30, 49]. And, as major router vendors are beginning to oblige with support for router virtualization [2] and customized protocols [5, 10], the time is imminent, when hosted virtual networks will be offered as a service, much like computing facilities are offered in a cloud computing environment.

## 1.1 Challenges

The realization of a Virtual Network Environment (VNE) is associated with a number of new technical, service and network management challenges [28].

From the network management perspective, the most significant challenge is an effective virtual network embedding, which deals with the efficient mapping of virtual resources to underlying resources of an InP. Other network management aspects like interfacing, signaling, bootstrapping, failure handling, performance monitoring and security also have new requirements and complexities in this environment.

From a service management perspective, it is important to ensure a market environment where successful customer-provider relations are established between SPs and InPs. A customer-provider business relation is successful only when service is rendered at the level of quality desired by the customer and at a price that satisfies the objectives of both parties. An effective service negotiation and contracting mechanism is required to achieve such an equilibrium. This is especially true for a VNE, where great business flexibility exists in terms of which providers (InPs) a SP can contract with. Although it may be possible for a small virtual network to be fully embedded in a single InP's infrastructure, it is much less likely for large inter-continental VNs. Indeed, VNs (e.g, VPNs, overlays) that are spread geographically are often provisioned among multiple network providers today. With VNE, we can expect a rather large number of InPs in the market, ranging from traditional underlay network providers to new 3rd party virtual network providers. Under the current inter-network business model, a group of collaborative InPs negotiate among themselves to jointly host such a VN; however, such a business arrangement is not customer-driven, as the VN assignment (among InPs) does not involve the customer in the negotiation process. Furthermore, it does not exhibit fair market properties, as it lacks free market competition and does not ensure price minimization for the customer. But, the aspect of service negotiation and contracting in a multi-provider scenario remain untouched in this context.

The concept of Network Virtualization is not entirely new. Concepts like P2P services, VPN, Overlay Networks, Grid Computing, and Cloud computing bear many similarities with it, as they all allow sharing of physical resources from across the world using virtualization techniques. Therefore, one might look to these, especially hosted cloud computing for solutions, given its tremendous recent growth. But, even in cloud computing, service management and network management in a multiple provider setting are yet to be explored, because

(i) the concept is fairly new and, therefore, the number of companies offering hosted cloud computing services is still limited and

(ii) critical IT infrastructures are yet to migrate to these cloud services. So, many management issues have not been pushed to the forefront.

However, we feel that the issue cannot be ignored for VNE.

## 1.2 Contributions

In this thesis, we present *V-Mart* [61], an open market model and enabling framework for automated service negotiation and contracting in VNE. To the InPs, V-Mart fosters an open and fair competition environment through auctioning; and to the SPs, it offers a customer-driven virtual network partitioning and contracting engine.

With V-Mart, a SP is not required to select from a set of pre-defined levels of services, rather he can specify his requirements and InPs can respond with offers based on them. Any willing InP may participate in a two-stage auctioning process, through which the SP ultimately decides who to contract with. The InPs need not disclose sensitive information (e.g, their pricing models) to participate in the auction. The first stage of the auction, which uses the *Vickrey Truth Serum*, elicits true estimates of virtual resources hosting costs in order to avoid price manipulations from the InPs. The second stage of the auction uses the results of the first and finally resolves the auction to decide the winners.

V-Mart offers two types of partitioning algorithms: one deterministic algorithm based on mathematical formulation of the VN partitioning problem and the other a greedy local search. The algorithms not only consider virtual node hosting and intra-InP data transfer price estimates but also the SP's preference for resource co-location and the high cost of inter-InP communication. Furthermore, V-Mart's partitioning algorithms does not impose a particular pricing model on the InPs, but on the contrary, supports diverse InP pricing policies ranging from resource-wise pricing to full network package pricing. Through extensive simulation experiments, we show that the algorithms are fast, efficient, and suited to handle heterogeneous market conditions and InP pricing models.

Although, V-Mart is designed for the VNE context, it is flexible enough to be applied to other distributed multi-provider service environments such as Cloud computing, Service-Oriented Architecture (SOA) infrastructures and Business Process Outsourcing.

## 1.3 Thesis Structure

The remainder of the thesis is organized as follows. Chapter 2 presents the concepts that are pertinent to this thesis and related work. Chapter 3 describes the proposed framework. In chapter 4, we evaluate the performance of V-Mart's VN partitioning algorithms through simulations. We conclude in chapter 5 with a summary of contributions and future plans.

# Chapter 2

# Background and Related Work

In this chapter we present the concepts that are relevant to the work presented in this thesis, and describe related work in literature. We begin in section 2.1 with an introduction to the virtual network environment, where we delve further into the VNE business model and identify important business actors and relations that define its economic and market models. We then explore, in section 2.2, existing work in literature that address service negotiation and contracting in this context. We conclude this chapter in section 2.3 with a discussion of service negotiation and contracting in areas that have resemblances with VNE.

## 2.1 The Virtual Network Environment

In a virtual network environment, the basic entity is a virtual network (VN), which is a logical topology composed of virtual nodes and virtual links. Provisioning a VN means mapping/embedding virtual nodes onto physical ones and virtual links onto physical links or paths. Once provisioned, a VN has the semblance of an actual physical network[1].

### 2.1.1 The Reference Business Model

The VNE business model resembles that of a cloud computing environment, except the service on offer is hosting virtual networks, rather than offering virtual computing resources. In this section, we describe the business entities in a VNE and their relations. We focus

---

[1] Note that virtual routers are not fixed to the physical routers on which they are mapped; rather, they can move to other physical locations if need be [59]

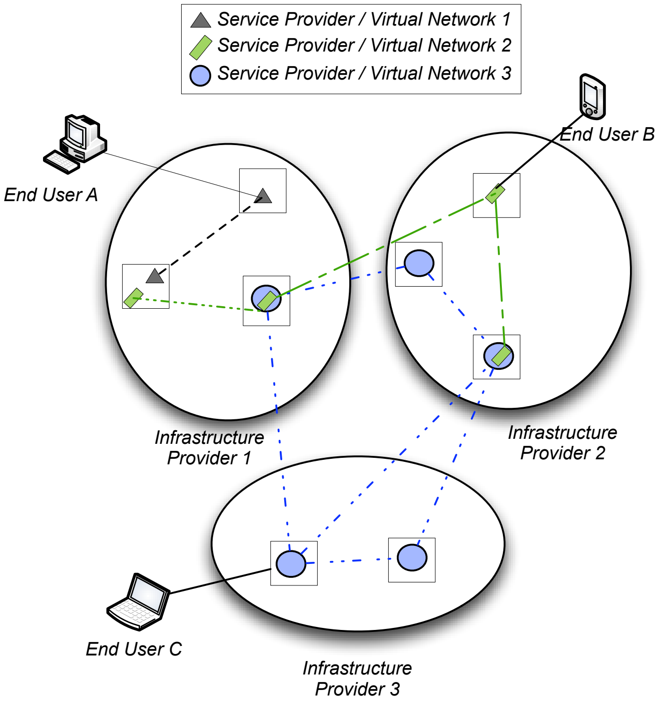


Figure 2.1: Network Virtualization Environment

primarily on the actors and the relations that are relevant in the design and implementation of a service negotiation and contracting framework; further details on the business model can be found in [27, 28].

### Actors

The VNE is primarily characterized by the decoupling of the traditional *Internet Service Providers' (ISP)* role. Here, the role of the ISP has been split into two [15, 57, 30, 43, 28]: *Infrastructure Providers*, and *Service Providers*. A sample VNE is shown in Figure 2.1, where the virtual networks 1,2, and 3 are created from physical resources provided by infrastructure providers 1, 2, and 3.

**Infrastructure Provider (InP):** Infrastructure providers own and manage physical networked resources, i.e, routers and links that connect the routers. By utilizing virtualization techniques, an InP can divide his physical resources into multiple logical/virtual ones. Depending on a customer's specifications, e.g, network topology, capacity of the routers and link bandwidth, an InP can set up a virtual network from a subset of his virtual resources and allow the customer full control over them.

**Service Provider (SP):** Service providers offer services to end users. Here, a SP is no longer faced with the daunting challenge of setting up an expensive infrastructure customized to suit his service(s). Rather, he can simply lease *virtual networks* from InP(s) for the job. A SP is free to customize his virtual network to his will and implement any routing, naming, control or management protocol or technology.

This decoupling of the ISP's role also helps modularize network management tasks and increase accountability at every layer of networking. InPs will be responsible for the management and operations of physical entities in the network. SPs, on the other hand, will only concern themselves with the management of the virtual networks and the services deployed on these VNs. Moreover, the separation of accountability will provide complete, end-to-end control over the VNs to the SPs, obviating the requirement of coordination across administrative boundaries as seen in the case of services that are deployed across multiple administrative domains.

Note that these role do not map one-to-one to the entities. VNE offers flexibility in who can take on these roles, and a single entity can take on multiple roles simultaneously. For example, a SP who leases virtual resources from the InPs can further divide his resources and offer them to other SPs. The recursive spawning of virtual resources, can thus, result in a *hierarchy* of roles.

The other roles encountered in a VNE are *end users* and *brokers*. These roles closely resemble their counterparts in any service environment deployed over the current Internet.

**End User:** End users are customers to service providers, whom they connect to through a local physical infrastructure provider's network. In this environment, an end user can choose from a number of service providers, and possibly subscribe to multiple services at the same time by simply connecting to multiple VNs [27].

**Broker:** Brokers act as mediators between InPs and SPs, to reconcile their different objectives. They can work *(a)* on the customers' behalf to gather information or act as customers to the infrastructure providers, *(b)* on the providers' behalf to perform resource scheduling, i.e., consolidate resources from multiple providers, manage these resources, and offer them to customers, or *(c)* as a neutral third party to manage negotiations between SPs and InPs, and to mediate contracts between them. As in any service environment, we encounter two different implementations of brokers in a VNE: (i) *Centralized*: where a single broker performs the task of mediation, or (ii) *Distributed*: where multiple brokers [39,41] work in conjunction under a single resource management fabric.

### Business Relations

Many business relations exist in a VNE between the actors. In this thesis, we are only interested in that part of the business model where the service in question is virtual network hosting, and the customers and the providers are SPs and InPs, respectively. End users have no impact or relevance, and a customer-centric broker appears as single customer in this context. Therefore, we explore them no further in the remainder of the thesis.

In this section we only focus on SP-to-InP and InP-to-InP relations. SP-to-SP relations are commonly established to provide end-to-end services to end users, and come at a stage following the virtual network service contracting. Therefore, we do not focus on them. Note that an in depth discussion of the business model is provided in [25].

**Vertical Relations**: A vertical relation, or customer-provider relation, is established between a SP and an InP at the time of VN setup. The InP generates revenue through provisioning, operation and maintenance of virtual resources[2] belonging to the SP. Like most other customer-provider relations, these are also regulated by *Service Level Agreements* (SLAs) [9, 8, 42] between the parties, that specify, among other details, performance constraints on virtual resources, and how violations of the agreement can be resolved.

In the traditional inter-networking model, a service provider is usually tied to one network provider. The provider ensures end-to-end service delivery by establishing peering relations with other providers. A single InP can serve multiple customers (SPs), resulting in a many-to-one customer-provider relations. However, VNE allows the flexibility of a single SP forming relations with multiple InPs simultaneously by partitioning his VN and allocating them to different InPs, resulting in a one-to-many customer-providers relation.

**Horizontal Relations**: Traditionally termed peering relations, a horizontal relation is formed between two or more providers to facilitate end-to-end service delivery. In the context of VNE, it is hard to imagine that all VN requests can be satisfied using the resources of a single InP. Most often a VN has to be split among multiple InPs. Therefore, a horizontal relation is formed between two or more InPs where each host a portion of the SP's virtual network.

---

[2] Throughout the rest of the thesis, we use the term *virtual resources* to refer to both virtual nodes and virtual links

In this context, these relations can be of two types: *public relations* and *private relations*.

Public relations are formed under the direction of a market mechanism, where competing InPs find themselves co-hosting neighboring segments of a VN. As these relations are often formed reluctantly, they hardly ensure low prices on inter-InP connectivity and guarantees on overall network performance.

Private relations, on the contrary, are formed voluntarily among a group of InPs who decide to cooperate in private in order to better compete in the market. These groups are represented in the market by a provider-centric broker working under a management fabric that spans all the members' domains. Note, that private relations are beyond the boundaries of the market mechanism, and to the market a group of InPs in such a relation (or the broker) appears as a single provider.

## 2.2 Service Negotiation and Contracting in Virtual Network Environment

A customer-provider(s) (vertical) business relation is successful only when service is rendered at the level of quality desired by the customer and at a price that satisfies the objectives of both the providers and the customer. Effective *service negotiation* (negotiations for setting price and quality of service) and *contracting* (the selection of providers) are extremely important to reach such an equilibrium. These are of central importance in a VNE, where great business flexibility exists in terms of which providers (InPs) a SP will contract with. Although it may be possible for a small size virtual network to be fully embedded in a single InP's infrastructure, it is much less likely for VNs with high bandwidth and CPU constraints and/or a wide geographic spread. Indeed, inter-continental VNs (e.g, VPNs, overlays) are often provisioned among multiple network providers today. Under the current network model, a service provider has access to few local infrastructure providers who collaborate with other InPs, and negotiate privately among themselves, to jointly host such a VN. However, such a business arrangement is not customer-driven, as the VN assignment (among InPs) does not involve the customer in the negotiation process. Furthermore, it does not exhibit fair market properties, as it lacks free market competition. With VNE, we can expect a large number of InPs in the market, ranging from traditional underlay network providers to new virtual network providers.

However, to the best of our knowledge, the aspects of service negotiation and contracting in a multi-provider scenario remain untouched in this context. Service negotiation in VNE literature is restricted to the virtual network *embedding* or mapping problem, that deals with provisioning or mapping virtual resources of a virtual network onto an underlying substrate network. The aim is to allow a maximum number of VNs while reducing total embedding cost (the total amount of resources consumed) and increasing revenue for providers.

The virtual network embedding problem is divided into two phases: (i) virtual node mapping, followed by (ii) virtual link mapping. Many proposals exist in networking literature to solve this NP-Hard problem [14]. Some make simplifying assumptions about the nature of VNs or physical infrastructures; these assumptions include:

(i) all virtual network requests are known in advance [63],

(ii) infinite capacity of the underlay resources [63, 46], and

(iii) VNs can only be of some specific topologies [46]

No such simplifying assumptions are made in [60, 26]. However, [60] assumes support in the underlay for virtual node and link migration, as well as multi-path routing. The authors in [26] consider location requirements on nodes, in addition to previously considered requirements on virtual resources, and introduce coordination between the node mapping and the link mapping phases.

The VN embedding proposals assign VN requests to one substrate domain, i.e, consider one InP's infrastructure. Moreover, the existing proposals address the mapping from the InP's point of view, and do not consider the monetary cost (price). But, without the direct participation in the service negotiation process, it is extremely difficult for a SP to ensure fair market practices or minimum costs.

## 2.3 Service Negotiation and Contracting in Related Areas

The concept of multiple virtual networks cohabiting a shared physical substrate has appeared in different capacities both in network literature and the industry. Like VNE, many service environments exist that leverage virtualization techniques to offer resources as a service to customers. These services can be divided into five categories: (i) network experiment testbeds, (ii) virtual private networks (VPN), (iii) peer-to-peer services, (iv) grids, and (v) cloud computing environments . Here, we explore the service negotiation and contracting in these environments.

### 2.3.1 Network Testbeds

PlanetLab [12, 51, 55] is an overlay testbed designed to allow researchers and other users to design, deploy and experiment with network applications and services that benefit from distribution across a wide geographic area. PlanetLab concurrently hosts multiple network applications and services by allocating a *slice* of its network-wide hardware resources to each of them. Each slice, which resembles a virtual network, is composed of a set of lightweight virtual nodes or *Virtual Servers (VServer*) spawned from physical nodes and connected through the Internet.

PlanetLab's business model resembles the VNE one, where the infrastructure provider's role is taken by PlanetLab, as they offer slices as a service to network researchers.

Technically, a slice somewhat differs from a virtual network. Rather than having dedicated virtual links between VServers, PlanetLab uses the Internet's best effort data delivery. Moreover, as an application layer testbed, PlanetLab can afford little or no control over the lower layers of the network protocol stack to the users. Therefore, services and network experiments are often bound by constraints imposed by the underlying architecture.

The VINI Project [6,19,20] extends PlanetLab to reduce the gap between virtual networks and slices with the addition of dedicated virtual point-to-point connectivity, access for each virtual server to network interfaces and improved isolation between slices. In addition to this, the *Trellis* platform [20], allows each virtual network to define its own topology, control protocols and forwarding tables.

However, few effective network experiments with new protocols, especially routing protocols, can be done on PlanetLab, as the transport mechanism remains TCP over IP. The OpenFlow project [49] addresses this issue. It allows users to try out different routing mechanisms by allowing direct access to switch/router flow tables. It uses customized switches[3] that consist of three parts: (i) *A flow table* with an action associated with each flow entry to tell the switch how to process the flow, (ii) *A secure channel* connecting the switch to a remote control-process (called the *controller*), allowing packets and commands to be sent between the controller and the switch, and (iii) *the OpenFlow protocol*, which provides a way for a controller to communicate with the switch. Network researchers are allocated OpenFlow controllers or slices of them to conduct their experiments. Using the OpenFlow protocol, they can configure switch/router flow tables and determine the routes

---

[3] OpenFlow switches are already in operation in the Stanford University campus

their packets follow and the processing they receive. In this way, researchers have the flexibility to experiment with routing protocols, security models, addressing schemes.

Although these network testbeds resemble the virtual network environment in outlook, there are fundamental differences in the two business models. For example, in PlanetLab, the physical hosts/nodes on which the overlay testbed is deployed are mostly contributions by different research organizations. Contributing organizations relinquish complete control over to PlanetLab. Moreover, PlanetLab slices are offered to network researchers at no cost. Therefore, no competition exist between actors, obviating the need for price negotiation.

Slice management in PlanetLab is accomplished by a centralized authority called the *PlanetLab Central or PLC*. The PLC prefers *best-effort open access* over admission control; therefore, there is no room or need for negotiations for quality of service.

The PlanetLab architecture permits third-party brokers with the endorsement of the PLC. However, these brokers are only responsible for managing slices at the granularity of individual nodes, and therefore, have little significance in business activities.

### 2.3.2 Virtual Private Networks

A virtual private network [31] connects multiple geographically distributed sites using a non-private data network to carry traffic between them. A VPN establishes connectivity between its remote sites using various tunneling mechanisms [52] over the Internet. Each VPN site contains one or more Customer Premises Equipment (CPE). These CPEs are connected using direct tunnels between them, or through other VPN capable routers, known as provider equipment (PE), in the core non-private network.

Typically a VPN is provisioned and managed by a VPN service provider (SP) on the customers' behalf [16]. The SP negotiates with the infrastructure providers (owners of the PEs) to deploy a VPN. Deploying a VPN involves the selection of a set of providers, who will provide their PEs, and a layout for VPN tunnels, such that one or more *factors* are optimized. These factors are (i) total monetary cost for the customer, (ii) bandwidth of tunnels, (iii) survivability of tunnels and edges, and (iv) number of hops between source and destination CPEs.

The authors in [38] provide optimal and approximate algorithms for provisioning a VPN such that bandwidth on the tunnels can be optimized. That is, the objective is to minimize cost of connectivity for the customer by reserving as little bandwidth as necessary to support the expected communication, as well as to be flexible enough to support a wide range of communication (traffic matrix) among the CPEs.

The authors in [29], on the other hand, provides heuristic based approaches that minimizes total monetary cost of operation and maintenance of the VPN. The proposed algorithms consider the cost of setting up and maintaining the tunnels, as well as of using the PEs as tunnel endpoints. The algorithms first construct a CPE-based solution, where the CPEs are connected through direct tunnels, then tries to improve the solution by spending funds $\boldsymbol{F}$ on activating provider edges.

### 2.3.3 Peer-to-Peer Services

Peer-toPeer (P2P) networks [47] are overlay networks, composed of nodes or peers that allow each other direct access to their resources (e.g, processing power, disk storage or network bandwidth). The peers form self-organizing networks that are overlayed on the Internet Protocol (IP). P2P networks became popular through its extensive use as a medium for file sharing. Now they have reached far beyond, as they have become popular for deploying a variety of services, including distributed storage, multimedia streaming and distributed online games.

The business model for P2P services differs from other service environments in two ways:

(i) The customer and the service provider roles are symmetric; here, a customer is a peer of the provider and may be a provider itself.

(ii) Any host can join in the network and become a peer as long as it is on the Internet; and, it can also leave the network at will. Therefore, service provider to infrastructure provider(s) (whose physical network is used to establish the overlay) relations are very dynamic in nature, and most often SPs and InPs are oblivious of each other.

PeerMart [40, 39] proposes a distributed market mechanism for trading P2P services. PeerMart holds a two-sided auction, where service providers specify the minimum cost of their services, while customers specify the maximum price they are willing to pay for it. In PeerMart, A group of decentralized brokers work in combination to reconcile the providers' ask prices and the customers' bid prices, by running a matching strategy.

### 2.3.4 Grids

Grids [35] gained popularity in the past decade for solving large-scale problems and for hosting large-scale applications and services. They enable the creation of *virtual organizations*, a group of geographically distributed individuals and/or organizations that share their heterogenous resources, including computing resources, storage resources, softwares, and databases. The sharing of resources in such a distributed multi-organizational environment is enabled by running software systems, such as Globus [33], and interoperability is achieved by having a common protocol architecture [35,34], that defines the basic mechanisms by which sharing relations are negotiated, established and managed.

In [23,24], a business model is proposed for the Grid environment with two key players: *Grid Service Providers* (GSPs) and *Grid Resource Brokers* (GRBs) representing customers. GSPs make their resources available to customers and GRBs manage and schedule these resources on the customers' behalf. The interaction between GRBs and GSPs during the price negotiation process is mediated by a *Grid Market Directory* (GMD). The authors of [23,24] propose the use of real-world economic models for service negotiation in the grid environment. These include,

- *Commodity market model* (also known as supply-and-demand driven pricing model), where the resource owners set prices for their resources such that supply and demand equilibrium is maintained. These prices are published through the GMD service, and a GRB tries to identify resources that meet the customers' requirements at minimum cost.

- *Posted price model*, which is similar to the commodity market model, except, here the providers advertise special offers to attract more customers.

- *Bargaining model*, where a GRB can bargain with a GSP for lower access price to its resources. The negotiation continues until a price is agreed upon which satisfies the objectives of both.

- *Tender/contract-net model*; Here the GRB announces its requirements and invites bids from GSPs. Interested GSPs respond with their bids, and the contract is awarded to the most appropriate (decided by the GRB) one.

- *Auction models*, in which a GSP acts as an auctioneer and invites bids from consumers or from GRBs. Access is provided to the consumer who offers the highest price.

To ensure proper operation of the market, the authors also propose an infrastructure to support interaction protocols, allocation mechanisms, currency implementation, secure banking and enforcement services.

Note that the business model for a grid computing environment resembles the VNE's, as both separate the roles of infrastructure providers and service providers. However, only private InP-to-InP business relations can be formed in a grid where participating organizations agree to share resources. But in VNE, InP-to-InP relations can be formed under the direction of the market, when InPs (not in an existing peering relation) host different segments of the same VN. The nature of these relations are not known in advance to the customer, making service negotiation and contracting more challenging here than in the grid environment.

### 2.3.5 Cloud Computing Services

Cloud computing has recently become one of the most addressed topics in both academic research and industry. The authors of [17] describe cloud computing as the combination of the applications delivered as services and the infrastructure (datacenter hardware and management softwares) that enable those services. Here, the infrastructure providers deploy traditional utility services or web applications (e.g, Google AppEngine) on their infrastructure (cloud), or virtualize their physical resources (e.g, Amazon EC2, Microsoft Azure) and offer them to customers. Additionally, the InPs offer automatic and easy scalability and pay-per-use option with no long term commitment. The customers or the service providers can, therefore, utilize the virtual resources or the web applications on-demand to deploy their own applications and services without having to worry about scale or huge investments in infrastructure.

The cloud computing business model is analogous to the VNE's, except for the difference in the type of service on offer. But, service negotiation and contracting in a multi-provider cloud computing services market is still unexplored. We believe the reasons behind this are:

1. The concept is fairly new, and the number of companies in the market offering hosted cloud computing services is still limited. Therefore, the need for a multi-provider market is yet to be felt.

2. Critical IT infrastructures are yet to migrate to cloud environments. So, many economic and management issues are not pushed to the forefront yet.

Here, we discuss the pricing models of two cloud computing services, Amazon Elastic Compute Cloud (Amazon EC2) [1] and Windows Azure [7], that offer compute resources in the cloud to its customers.

Amazon EC2 uses virtualization techniques [18] to divide its physical hosts or datacenters into multiple virtual instances, and offers them to the service providers. The customers have complete control, from the kernel upwards, over these virtual instances. Currently, Amazon offers eight types of instances differing from each other in memory, cpu capacity and/or storage capacity. Note that amazon provides no guarantee on I/O performance of these instances.

Amazon primarily adopts two different pricing models:

1. *Per-resource, usage-based pricing*. Customers have no room for price negotiation as the pricing is fixed for guaranteed instances. Promotional per usage prices are offered with long term commitments.
2. *Dynamic auction-based pricing for unused instances*. A customer can bid with the maximum amount he is willing to pay for an unused instance. If the bid exceeds the current spot price (maximum current bid), the instance is granted to the customer until the spot price exceeds it's bid.

Microsoft Azure provides a platform for development, service hosting and service management through which users can develop applications and deploy them using on-demand compute and storage capacities from Microsoft's datacenters. Microsoft uses two pricing models for this service:

1. *a fixed (non-negotiable), usage-based pricing model*, and
2. *a flat discounted rate with long term commitments*. In this category, the usage is limited to a maximum amount, additional usage is charged according to the usage-based pricing model.

# Chapter 3

# A Framework for Service Negotiation and Contracting

In this chapter, we describe V-Mart, our proposed framework for service negotiation and contracting in virtual network environment. We commence with a sample business case, in section 3.1, used to illustrate the problem that is addressed in this thesis. In section 3.2, we present the important features of V-Mart and describe its workflow. In section 3.3, we provide a detailed description of V-Mart's auction model, the model for service negotiation. Finally, in section 3.4, we describe V-Mart's VN partitioning algorithms that can be used by a SP to determine how to best divide his VN among the bidders.

## 3.1 A Sample Business Case

Waterloo based company *W-VPN* is in the business of setting up VPNs for its customers on demand. Most of W-VPN's customers are multi-national companies who use VPNs to connect their offices that are distributed all over the globe. W-VPN establishes connectivity between the customers' offices using VPN tunneling mechanisms over the Internet. It has contracts with multiple network providers who offer their VPN-capable routers and switches as intermediate points for these tunnels.

W-VPN would benefit immensely from having its own virtual network infrastructure, consisting of virtual routers and virtual links. A VN would give it direct control over the intermediate PEs (Provider Equipment), and it can set up or modify VPN tunnels quickly and on-demand. Moreover, having its own dedicated infrastructure would allow W-VPN to deploy any tunneling mechanism in existence or even experiment with new and alternative mechanisms.

### 3.1.1 Requirements on Virtual Networks

W-VPN's virtual network has a topology and a set of performance constraints (e.g, [9,13,11]). Generally, the kind of services deployed over the virtual network, quality of service and geographic location distribution of the end user base determine the topology of a VN and performance constraints on it.

Constraints on a virtual network can be specified at different granularities, starting from each virtual resource to the entire VN. Typical constraints on a virtual node includes CPU capacity, Queue size, availability, mean time to repair (MTTR), etc. For virtual links, these include bandwidth, delay, latency, packet loss and availability. Also, end-to-end performance constraints are often imposed on the entire VN topology or on parts (subgraphs) of it. Threshold values for end-to-end latency, end-to-end delay, average jitter, maximum jitter and network availability are few such constraints. Figure 3.1.2 shows the topology of W-VPNs virtual network, the bandwidth requirements[1] are denoted beside each link and the CPU capacity constraints on the virtual nodes are shown in rectangles.

### 3.1.2 Co-Location Constraints

Different parts of a VN can be assigned to competing InPs. End-to-end performance guarantees on a subgraph (of the virtual network) that spans multiple such competing provider domains are hard to achieve, as a clear assignment of responsibility and accountability is not easy to establish. However, a SP may have strict performance constraints on a group of resources or a subgraph of its VN. For example, W-VPN requires that the maximum end-to-end delay be 2s in the subgraph used for the VPN of Company $G$, its most important customer. A guarantee on this constraint is imperative to ensure that $G$ is completely satisfied.

In situations like this, a SP can specify that these virtual resources be assigned to a single InP so that failures can be promptly handled, and the responsible party can be clearly identified and accordingly penalized. We term this requirement as *Co-Location Constraint*.

---

[1] We use the terms *performance constraints* and *performance requirements* interchangeably

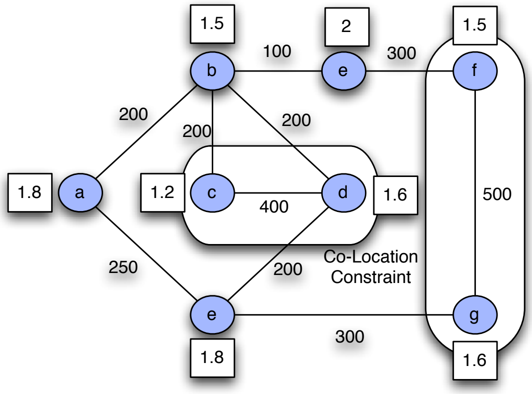


Figure 3.1: The W-VPN virtual network

The parts of the W-VPN's VN graph which have a co-location constraint, i.e, that have to be assigned to a single InP, are shown using white ovals in figure 3.1.2.

### 3.1.3 Service Negotiation and Contracting for W-VPN's VN

Although it may be possible for a small size virtual network to be fully embedded in a single InP's infrastructure, it is much less likely for large inter-continental VNs like W-VPN's. Also, an InP might only be willing to host parts of the VN simply because the rest does not appear profitable. Moreover, in case a single InP can be found who can and is willing to host the entire VN, it might still be cost minimizing for W-VPN to go with multiple InPs.

Given a market with multiple InPs competing to host its VN, the problem faced by W-VPN boils down to establishing contracts with a set of InP(s), who can host the VN while satisfying all the requirements at the least cost.

## 3.2 Overview of V-Mart

We now introduce V-Mart, an open market model and an enabling framework for automated service negotiation and contracting in VNE. To the InPs, V-Mart fosters an open and fair competition environment through auctioning; and to the SPs, it offers a customer-driven virtual network partitioning and contracting engine.

Although, V-Mart is designed for the VNE context, it is flexible enough to be applied to other distributed multi-provider service environment such as Cloud computing, Service-

Oriented Architecture (SOA) based services and vendor selection for Business Process Outsourcing.

### 3.2.1 Features of V-Mart

V-Mart is designed to have the following features:

- **Open Market:** Any willing InP may participate in V-Mart's two-stage auctioning process, through which the SP ultimately decides who to contract with and at what price.
- **Flexibility:** An InP can deploy any pricing mechanism in the second round auction and need not disclose it to the SP. In the first round, the InP can adjust his bidding strategy according to his pricing model to get an upper hand in the second. Furthermore, V-Mart's partitioning algorithms do not impose a particular pricing model on the InPs, but on the contrary, supports diverse InP pricing policies ranging from resource-wise pricing to full network package pricing.
- **Truthfulness:** The first stage of the auction uses the *Vickrey Truth Serum* to elicit truthful estimates on virtual resources hosting costs in order to avoid price manipulations from the InPs. The second stage of the auction uses the results of the first and finally resolves the auction to decide the winners.
- **Incentive Compatibility:** All parties in the market have enough incentive to gain profit. The auction model ensures that InPs can bid with a certain profit margin, and the VN Partitioning algorithms ensure that total cost of VN hosting for the SP is minimized.
- **Automated:** The negotiation and contracting process is automated in V-Mart, supporting quick on-demand VN setup. To this end, V-Mart offers the service providers multiple VN partitioning algorithms that can be used to automatically determine a contracting strategy that is cost minimizing. The algorithms not only consider virtual node hosting and intra-InP virtual link hosting price estimates but also the SP's preference for resource co-location and the high cost of inter-InP communication.
- **Effective and Efficient:** V-Mart's auction is designed to efficiently perform service negotiation. Also, V-Mart provides effective VN Partitioning algorithms for various

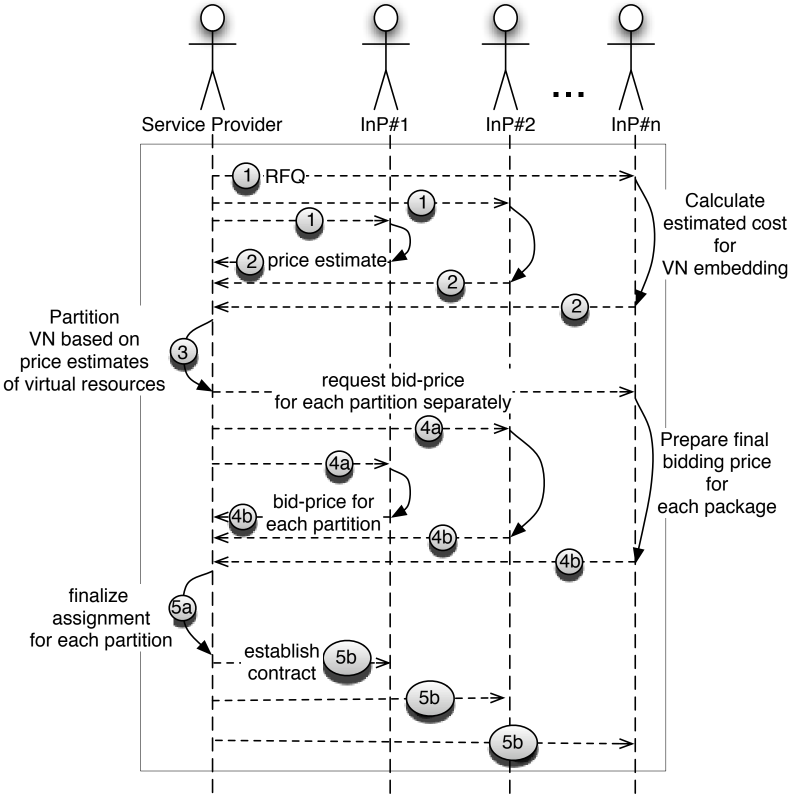


Figure 3.2: The V-Mart Workflow

market conditions, depending on the size of the VN topology, pricing models of the providers, and preferences of the providers and the customers.

### 3.2.2 V-Mart Workflow

We briefly overview the V-Mart operations here, the technical details are presented in sections 3.3 and 3.4. Figure 3.2 illustrates a V-Mart workflow example.

**Phase 1 - Request for Quotation (RFQ):** The SP formulates his VN request in the form an RFQ. The RFQ includes the virtual network topology, co-location constraints, and performance constraints. Figure 3.3 shows a sample xml-like RFQ for W-VPN's VN.

The RFQ is sent out to all interested InPs. This phase is shown as step 1 in Figure 3.2. RFQs can be disseminated by posting them on well known RFQ repositories. InP's can periodically check these repositories to find VNs that suit their niche. We refer to each interested InP as a *VN bidder*. Note that from the market's perspective, there is no distinction between a single InP and a representative of a group of collaborating InPs in a private relation. Therefore, in this context we refer to either as a VN bidder.

**Phase 2 - Resource Estimates:** Each VN Bidder is expected to indicate the virtual resources he is willing to host and the corresponding estimated price quotation under the Vickrey Model (section 3.3). This is shown as step 2 in Figure 3.2.

The VN Bidder performs embedding (mapping virtual resources to physical ones) of the VN to determine the actual hosting cost. However, the quotation is derived using a pricing mechanism (e.g, [45]). V-Mart allows the use of any pricing model, but it is quite important for a VN Bidder to adopt the proper bidding strategy (section 3.3.3).

**Phase 3 - VN Partitioning:** By the end of Phase 2, the SP obtains a set of price estimates for each virtual resource in his virtual network. Based on these estimates the SP partitions the VN into multiple segments and attach them to specific VN Bidders. V-Mart provides two partitioning algorithms that perform this task automatically for various market conditions, aimed at both minimizing total cost for VN hosting and satisfying the SP's co-location constraints. This is shown as step 3 in Figure 3.2.

**Phase 4 - Final Offer:** The list of segments obtained in Phase 3 is sent to all VN Bidders, as well as the winning VN Bidder of each segment and the winning Vickrey price. The VN Bidders make one final sealed bid that is upper bounded by the winning quote. This second and final stage of the auction determines a final winner for each segment. This phase corresponds to steps 4a and 4b in Figure 3.2.

**Phase 5 - Contracting:** The SP contacts the winning VN Bidder of each segment and performs the final contracting and SLA generation. This phase is steps 5a and 5b in Figure 3.2.

## 3.3 The V-Mart Auction Model

Infrastructure providers would naturally desire to get the highest price from the service provider for hosting his virtual network. A service provider, on the other hand, does not want to pay the highest price but wants to negotiate for the best price based on the demand, the utility he will derive from it, priority and available budget.

There are many approaches to set prices for networked resources in a multi-provider scenario. The simplest approach is to have fixed prices determined by a central authority.

```
<RFQ ID="01011101">
<Owner>
<Name>W-VPN</Name>
<Address>192.168.19.33</Address>
</Owner>
<Validity>2009-07-18T00:00:00.45+01:00</Validity>
<Auction Time>2009-01-18T00:00:00.45+01:00</Auction Time>
<Topology>
<Node ID=N0>
<Name>"a"</Name>
</Node>
<Node ID=N1>
<Name>"b"</Name>
</Node>
....
<Link ID=L0>
<EndNodes>"a","b"</EndNodes>
</Link>
<Link ID=L1>
<EndNodes>"b","c"</EndNodes>
</Link>
....
<V-Let ID=I0>
<NodeID>N2</NodeID>
<NodeID>N3</NodeID>
</V-Let>
</Topology>
<Constraints ID=R0, Type="Node", Target=N0>
<Constraint ID=R00, Type="CPU Capacity">1.8GHz</Constraint>
.....
</Constraints>
<Constraints ID=R1, Type="Link", Target=L1>
<Constraint ID=R10, Type="Bandwidth">200Mbps</Constraint>
.....
</Constraints>
<Constraints ID=R2, Type="V-Let", Target=I0>
<Constraint ID=R20, Type="Delay">200ms</Constraint>
.....
</Constraints>
</RFQ>
```

Figure 3.3: Sample RFQ for W-VPN's VN

While this removes most communication and processing overhead, it implies that the central authority have all information about the infrastructure providers' networks, pricing models and business policies, so it can determine the appropriate price. Moreover, it has to be trusted equally by all. Unfortunately, these assumptions are hardly practical, because InPs are reluctant to put complete faith on any party and share their sensitive internal information. Another option might be for service providers to only have access to a set of reputed infrastructure providers who aggregate resources from others to offer the VN as a package to the SP. However, this process does not foster free market competition and creates a strong barrier to entry for new and small InPs. Furthermore, it is impossible to guard against monopoly or collusion on part of these InPs.

Auction is an effective open negotiation mechanism for multiple competing buyers and sellers. It does not rely on price fixing by centralized authority and serves as a fairer alternative to provider-centric whole VN packaging. The first step is to employ an appropriate auction model for this context. The number and types of items being traded, the number of sellers and buyers, the preferences of the parties, and the form of private information participants have about preferences determine the best auction model for a particular environment.

A virtual network (the item being traded) is a set of virtual routers correlated with each other through virtual links. A VN bidder can bid on a per virtual resource basis. But, a common approach is to offer price discounts for hosting a certain volume or a combination/*package* of virtual resources. These discounted rates are calculated using a *discount function*.

Once the bids are received by the auctioneer, the problem of winner determination (who gets to host which virtual resources) involves finding a partitioning of the VN topology and an allocation of each partition or segment to bidders such that total cost is minimized. But:

(i) The discount functions are rarely shared with the auctioneer or customer, and

(ii) In case these discount functions are disclosed, considering diverse discount functions and pricing models makes winner determination extremely complex.

Our approach is to simplify the winner determination problem by splitting the auction into two stages. In the first stage, each bidder specifies an estimate of the actual price for hosting each virtual resource. These estimates are used to guide the VN partitioning process.

VN bidders compete to influence the partitioning so that one or more resulting segments suit their niche. The goal of the SP is to elicit truthful information on the cost of hosting the virtual resources in order to perform an effective partitioning. In the second stage, each partition or segment is auctioned individually and is assigned to the lowest VN bidder. The second stage auction is based on actual prices. Bidders can calculate their bids by applying any pricing model, including package pricing mechanisms without having to worry about a mismatch between the package and the segment configurations.

### 3.3.1 A Two-Stage Vickrey Auction Model

There exist many forms of auctions, such as the popular English auction with reservation, the Dutch auction, sealed high bid auction, Vickrey auction, etc. In general, a fair auction is *two-sided*, where customers submit bids and producers submit quotations, and a matching algorithm is run to produce the final result. When the customer or the producer cannot form an informed evaluation of the goods, *one-sided* auction is preferred. This is the case in VNE. Each infrastructure provider places different values on his virtual resources depending on multiple factors including the complexity involved in embedding and provisioning the virtual resources on a physical network, the amount of physical resources consumed, the amount of residual capacity, load on the physical network, lifetime of the virtual network, and his business goals and policies. The *valuation* is *private* and not disclosed to other infrastructure providers or the service providers. A SP can only formulate a vague upper bound on the cost. It is, therefore, very important to have an auction model that is free from price manipulations, i.e, that ensures the InPs will offer prices proportional to his valuation of the virtual resources. The Vickrey auction model [58] is such a truthful mechanism. Under the Vickrey auction model, VN Bidders will quote a price for hosting a virtual resource, and the bidder of the lowest quote would be given priority in the partitioning, but the SP will accept prices equal to the second lowest quote. The Vickrey model is strategy-proof, meaning that the only dominating strategy in this auction is for each VN Bidder to quote a price which is not too high compared to the actual valuation of the VN. We employ this model in the first stage of V-Mart's auction.

The Vickrey auction model can be implemented as either an open or sealed auction. The *open* auction has two issues.

1. It is price minimizing with respect to large number of bidders. The dominant strategy

for a VN Bidder is to quote a price exactly equal to cost, and thus gaining zero profit. This model is effective and strategy-proof, but does not offer fair market value to the InPs.

2. The number of auction iterations is large. In an attempt to optimize profit, each VN Bidder will decrease his bid $\epsilon$-small from the current winning quote during each iteration until he wins the bid or has a profit margin of zero. This is commonly known as the *shilling effect*.

On the contrary, a *sealed* Vickrey auction is a single round auction that has all VN Bidders bid a price in secrecy. This model does not suffer from the price minimization issue due to the psychological effect of incomplete information.

Thus we arrive at the one-sided sealed Vickrey auction model for V-Mart. Each VN Bidder receives a RFQ from the SP and submits a price quote for each resource he is willing to host. The SP takes all the quotes for each virtual resource and modifies the bids to equal the immediately higher bid. This is done to all bids, except the highest. For example, for the quotes on a virtual node, $\{2, 3, 4\}$ from bidders $A, B$ and $C$ respectively, the Vickrey quote is $\{3, 4, 4\}$. The V-Mart model is strategy-proof, as we can see that a bidder has no incentive to quote a price much higher than the actual cost, as it will only benefit his competitor.

The result of the first stage Vickrey auction serves as the basis for our VN partitioning algorithms that minimize cost for the SP. As the result, we obtain a set of VN partitions/segments with its associated total price and VN Bidder. In the second stage auction, all the segment topologies, their total costs (termed *maximum reservation price*), and the identities of the associated VN Bidders (termed *owner of the insured bid*) are sent to all VN Bidders in a single round sealed auction. Each VN Bidder is asked to provide a final price quote on each segment (if the VN Bidder is willing and/or able to do so), and this price quote is upper-bounded by the maximum reservation price.

The SP takes all the bids and awards the contract for each VN segment to the lowest VN Bidder for that segment. If the final price quote matches the maximum reservation price, the owner of the insured bid receives the contract. This stage is a one-sided sealed auction with maximum reservation. This auction model is selected for the same reasons as before.

Our two-stage auction model is flexible in dealing with heterogenous correlated commodities, which conventional auction models such as the Vickrey model are unable to. The strategy-proof first stage auction provides the necessary faithful cost information upon which a SP can effectively perform a cost minimizing partitioning. The resulting contracts are then processed between the SP and the winning VN Bidders. We note some issues:

1. In case one or more virtual nodes cannot be hosted by any bidder in the market, V-Mart will fail to find a solution, i.e., fail to determine a contracting strategy for the SP. However, in the presence of a large number of InPs in the market, we can expect this to rarely occur.

2. For the SP to be faithful, the first round bids should be secured either by audit via a trusted 3rd party, or by an open audit procedure at the end of the auction.

3. It is essential for the second-stage auction to have a maximum reservation price to ensure the validity of the first stage Vickrey auction, and to prevent VN bidders from quoting too low prices to influence the partitioning.

4. We do not study long term strategy emergence here, as typically a SP would not provision the same VN repeatedly. Furthermore, obtaining the strategy of an InP through observation over long term is not easy unless the physical resource topology and the pricing model of the InP is known, both of which are private information to the InPs.

### 3.3.2 The Pricing Models of a VN Bidder

Thus far, we have assumed that a VN Bidder is able to provide a per virtual resource price estimate in the first stage auction. Although there is great flexibility and advantages to per resource pricing model as implemented by a great majority of cloud computing infrastructures (e.g. Amazon EC2, Google Apps, Microsoft Azure, etc.), the pricing models of VNE are strongly influenced by the traditional network provider business model that operates quite differently from the application provider's or the cloud computing provider's. In this subsection, we show two common pricing strategies: *volume discount* and *package pricing*. Both of these strategies are commonly used by sellers of multiple commodities.

Volume discount is a standard economic practice where the seller is willing to offer a discounted per unit price when a large enough volume of the commodity is purchased (e.g. wholesale). The discount is typically described as a discount function and reported to the customer. In the VNE context, it is difficult to report this discount function to the SP. One reason is that the VN Bidder may not be willing to disclose his exact charging function to the buyer in order to guard against a competitor who can masquerade as a buyer and poach for his pricing model. Secondly, incorporating discount function in the VN partition problem would increase the complexity of an already challenging problem. Instead, a VN Bidder's volume discount function can be applied in the second stage of the V-Mart auction model, where the VN has already been segmented and the exact resource count in each segment is public information. Hence in the first stage auction, the VN Bidder may quote a resource-wise price at non-discounted rate, while in the second stage auction quote a segment-wise price as modified by his discount function.

Package pricing is a common practice in heterogenous multi-commodity market, where a seller wishes to sell a package of commodities together at lower price (e.g. a whole dinning room set is cheaper than the sum of its individual parts). For the first stage of V-Mart auction, the VN Bidder would calculate the price for a package of virtual resources and then map this package price to per resource price. This can be done by either computing the average price based on number of resources in the package, or by computing a weighted average of the resources based on the proportional cost of supporting each resource in the package. Although the latter method is preferred by V-Mart, it is understandable that a VN Bidder may not wish to disclose the exact cost of hosting any specific resource within a package. In the second stage of the auction, the VN Bidder will quote the segment price based on his package price model. It is apparent that a mismatch in the VN Bidder's original package and the SP's segment partition will be problematic. We discuss how such mismatch can be avoided by adopting the right bidding strategy.

### 3.3.3 Bidding Strategy

We now discuss the aspect of strategy game play by VN Bidders under different pricing models. To this end, the strategic move a VN Bidder makes in the first stage of V-Mart auction is critical as it strongly influences how the VN segments are generated by the partitioning algorithms. At first glance, this assertion appears to be contradictory to the concept of strategy-proof as we have stated: the one-sided sealed Vickrey auction model has a single dominate strategy which is to quote a price proportional to the cost. We now observe that both the volume discount and the package pricing models can only provide a truthful cost estimate in the first stage of the auction. Or to put it another way, the exact cost of hosting cannot be known until after VN partitioning.

In our discussion below, we use the term *best strategy* (not *dominant strategy*) for a bidder to denote an approach or a strategy that is best under specific conditions, specified by the bidder's pricing model and its competitors' pricing models and bidding strategies.

First we examine the possible strategy moves of a VN Bidder under volume discount pricing. In the first round of auction, the VN Bidder can adopt a *risk-averse* stance or a *risk-seeking* stance. A risk-averse VN Bidder will estimate the cost of the resources at their non-discounted price and thus risk no chance of negative profit after VN partitioning. A risk-seeking VN Bidder will estimate the cost of the resources already at a discounted price by assuming some of the final VN segments will contain at least the expected number of resources. Thus a negative cost could be incurred when this VN Bidder is the owner of an insured bid with smaller than expected segment size. On the other hand, a risk-seeking strategy is the *best strategy* move when the VN Bidder considers itself to be offering low discounted price in the market. Effectively, it can corner the demand market especially when the other VN Bidders are risk-averse or do not provide equivalent price discounts. However, when all VN Bidders are risk-seeking, we arrive at an inefficient system state where the sellers are selling at negative profit. Although such a state is cost minimizing for the SP, it is not fair to the InPs.

The best strategy for a VN bidder who adopts a package pricing model is also a risk-seeking approach: to bid for large packages and to compute per resource cost estimates based on the discounted package rate rather than non-packaged resource-wise estimates. It is also greatly important that the discounted package price be evenly distributed among the virtual resources, i.e, island prices should be comparable to that of the intra-domain virtual links. Effectively, this produces high inter-InP link weight compared with island weight. We show in section 4.4 that this strategy will give the bidder the best chance to get one or more segments/partitions in the second stage auction that match its desired packages.

## 3.4 VN Partitioning among Multiple Providers

The VN partitioning among multiple VN bidders is done after the first stage of the auction, and on the basis of price estimates. The actual price of hosting the VN is only determined after the second stage auction. However, the partitioning has great importance in achieving a balance between the interests of the SP and the InPs. A partitioning that appears to be cost reducing for the SP, might turn out to be undesirable to the VN bidders. For example, a partitioning that produces many small sized partitions can appear cost reducing based on the estimates, but it does not guarantee best offers in the second stage auction from the bidders who adopt volume based discounts.

Therefore, V-Mart's VN partitioning algorithms consider:

1. Total VN hosting cost,
2. High price of inter-domain virtual link cost,
3. SP's preference for virtual resource co-location, and
4. Bidders' preferences on the size of the partitions based on their pricing models.

In this section we describe V-Mart's VN partitioning algorithms. V-Mart proposes two partitioning algorithms. For the first algorithm, we formulated the VN partitioning problem mathematically as a mixed integer program. We then relaxed the integer constraints to obtain a relaxed linear program. Finally, we applied a deterministic rounding technique to obtain the assignment for each virtual node to a bidder. In the second type of algorithm, we used a greedy local search method. The algorithm starts from an initial mapping and greedily performs local improvements to find a local optima. We present two variants of this greedy search algorithm: G-MinIslandCost and G-MinCutSize, depending on the starting configurations that they use.

### 3.4.1 Virtual Network Model and Problem Description

We start with a formal description of the problem.

#### Virtual Network Request

We denote a virtual network request by an undirected weighted graph, $\boldsymbol{G^V = (N^V, L^V)}$, such that $\boldsymbol{N^V}$ and $\boldsymbol{L^V}$ are the set of virtual nodes and the set of virtual links, respectively. The performance constraints on a virtual resource are represented using a set of type/value pairs. For example, the constraints on link $\boldsymbol{l^V}$ are represented as $\boldsymbol{R(l^v) = \{(\text{"}bandwidth\text{"}, 200), (\text{"}MaxDelay\text{"}, 200ms)\}}$.

**Co-location Constraint:** Co-Location constraints on pairs of VN nodes are expressed using a *Binary Co-Location Matrix*, $\boldsymbol{Col}$, where $\boldsymbol{col(a,b) = \{0,1\} \forall a, b \epsilon N^V}$ such that, if,

$col(a, b) = 1$, then both $a$ and $b$ have to be assigned to the same bidder. Note, that the co-location constraint is symmetric and transitive. Therefore, the co-location matrix divides the VN into a set of *islands*, where each island is composed of VN nodes, that form a chain of co-location constraints, and the virtual links that connect these nodes. For example, if $col(a, x_1) = col(x_1, x_2) = \ldots = col(x_n, b) = 1$, then nodes $a, x_1, \ldots, x_n$ and $b$, along with the links that connect any two of these nodes, belong to the same island.

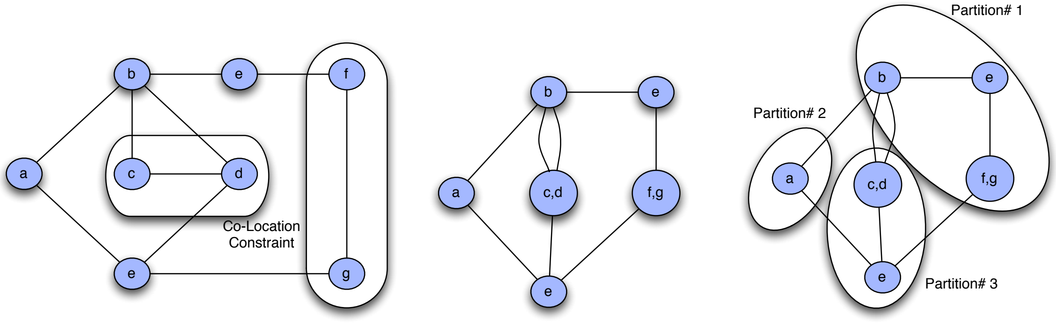
### Estimates of Virtual Resource Hosting Price

Assume there are $K$ VN bidders competing for the virtual network. Each bidder quotes the estimated price for hosting VN, $G^V = (N^V, L^V)$. The estimates are specified on a per-resource basis. Note that a VN bidder might not be able or willing to host an entire VN. Therefore, he specifies which virtual resources he is willing to host along with the price estimate. We represent the estimates on each virtual node $n^V \epsilon N^V$ with a vector, $C^N(n^V) = \{c_k^N(n^V) | k = 1, \ldots, K\}$, where $c_k^N(n^V)$ is the estimated price for node $n^V$ quoted by bidder $k$. And, for all $a \epsilon N^V$ such that bidder $k$ is unwilling to host $a$, the value of $c_k^N(a)$ is set to infinity.

We consider two types of prices for each virtual link $l^V$:

1. **Intra-domain:** A VN bidder specifies an intra-domain link hosting price for link $l^V$ if he also quotes prices for both of its endpoints. We represent the intra-domain link price estimates using a vector, $C^L(l^V) = \{c_k^L(l^V) | k = 1, \ldots, K\}$, where $c_k^L(l^V)$ is the quoted price for link $l^V$ by bidder $k$.

2. **Inter-domain:** If the end-nodes of link $l^V$ are assigned to two different bidders, say $k_1$ and $k_2$, the intra-domain link price specified by either bear no sense. The actual price depends on the horizontal relation between $k_1$ and $k_2$, and is not disclosed by either to the SP. Therefore, a SP is left with no choice, but to assume an estimate on an empirical basis. As, most often $k_1$ and $k_2$ are in competition, the prices of hosting such inter-domain links are generally higher than intra-domain links of similar capacity. In fact, industry trends suggest that inter-domain communication cost is a magnitude higher than intra-domain. For example, Amazon EC2 [1] charges at most \$0.01 for one GB of intra-provider data transfers and a minimum \$0.10 for inter-provider. Therefore, for each virtual link, V-Mart assigns a constant inter-domain cost $\acute{C}^L$, valued at an order of magnitude higher than the highest intra-domain link estimate.



(a) (b) (c)

Figure 3.4: V-Mart VN partitioning process (a)Virtual network topology of W-VPN, (b)Meta-graph formed by coalescing nodes with colocation constraints into islands (c) Formation of packages using V-Mart partitioning algorithm

### VN Partitioning Problem

The problem of partitioning a VN among $\boldsymbol{k}$ VN bidders can be divided into two steps:

1. Coalesce virtual nodes with Co-Location constraints, forming a meta-graph of islands, and
2. Partition the meta-graph into $\boldsymbol{P <= K}$ partitions.

The process is shown in figure 3.4.

**Meta-Graph Formation:** Each island in the VN graph has to be assigned to the same partition. Therefore, as the first step of partitioning, V-Mart forms a meta-graph, $\boldsymbol{G^M = (I^M, L^M)}$, from the VN request graph, where $\boldsymbol{I^M}$ is the set of islands in the VN and $\boldsymbol{L^M}$ is the set of virtual links between two virtual nodes in different islands. The price estimates for an island $\boldsymbol{i^M \epsilon I^M}$ are equal to the total of all the virtual resources, virtual nodes and links that compose the island, we represent the estimate as, $\boldsymbol{C^I(i^M) = \{c^I_k(i^M) | k = 1, ..., K\}}$, where $\boldsymbol{c^I_k(i^M) = \sum_{n^V \epsilon i^M} c^N_k(n^V) + \sum_{a,b \epsilon i^M} c^L_k(a,b)}$. Note that $\boldsymbol{L^M}$ is a subset of $\boldsymbol{L^V}$, so we do not modify the representation for the price estimate on these links, we only alter the end points of the links from virtual nodes to the islands that those nodes belong to. The meta-graph formation process is shown in figure 3.4(b).

**Meta-Graph Partitioning Problem:** The meta-graph partitioning problem can be expressed as the mapping function,

$$\mathcal{M}_N : I^M \rightarrow \{bidder|bidder = 1, 2, ..., K\} \tag{3.1}$$

from islands to VN bidders.

The objective is to minimize the total cost,

$$\mathbb{C}(G^M) = \sum_{i^M \epsilon I^M} c^I_{\mathcal{M}(i^M)}(i^M) + \sum_{(a,b)\epsilon L^M, \mathcal{M}(a)=\mathcal{M}(b)} c^L_{\mathcal{M}(a)}(a, b) + \sum_{(a,b)\epsilon L^M, \mathcal{M}(a)\neq\mathcal{M}(b)} \acute{C}^L \tag{3.2}$$

We represent the virtual links as a tuple of the end-points, e.g, $l^M$ is represented as $(a, b)$, such that islands $a$ and $b$ are the end points of $l^M$.

The VN meta-graph partitioning problem is a complex optimization problem. Each island and link in the meta-graph is associated with a $K$-dimensional cost vector. Also, the placement of an island is strongly correlated with that of its neighbor islands. Moreover, the partitioning has to consider the preferences of VN Bidders. In a market where most bidders adopt volume based pricing models, larger partition sizes are preferable to the InPs. The SP also benefits from having partitions that are appealing to the InP. This ensures that the SP will get low price offers from the bidders in the second stage. But, having large partitions carry lesser importance to bidders who adopt a resource wise pricing model.

### 3.4.2 Linear Programming Relaxation and Deterministic Rounding

As our first stab at this problem we formulate the VN partitioning problem as a mixed integer program. We then relax the integer constraints and apply a deterministic rounding technique to obtain a polynomial-time algorithm, LP-D, for finding a mapping function 3.1.

#### Mixed Integer Problem Formulation

The following mixed integer program represent the VN meta-graph partitioning problem.
**Variables:**

- $x_{i,j}$: A binary variable which has the value of $1$ if island $i$ is assigned to bidder $j$
- $y_{i,j}$: A binary variable that has the value of $1$ if both end-nodes of meta-link $i$ are assigned to the bidder $j$.

**Objective:**

*minimize*

$$\sum_{i\epsilon I^M}\sum_{k\epsilon\{1,2,\ldots,K\}} x_{i,k}c_k^I(i) + \sum_{l\epsilon L^M}\sum_{k\epsilon\{1,2,\ldots,K\}} y_{l,k}(c_k^L(l) - \acute{C}^L) + |L^M|\acute{C}^L \tag{3.3}$$

**Constraints:**

-Unique Assignment Constraint

$$\sum_{k\epsilon\{1,2,\ldots,K\}} x_{i,k} = 1, \forall i\epsilon I^M \tag{3.4}$$

-Link Assignment Consistency Constraint

$$y_{l,k} \leq x_{n,k} \forall k\epsilon\{1,2,\ldots,K\}, l = (n1,n2), \forall n\epsilon\{n1,n2\} \tag{3.5}$$

-Domain Constraints

$$x_{i,k}\epsilon\{0,1\}, \forall i\epsilon I^M, \forall k\epsilon\{1,2,\ldots,K\} \tag{3.6}$$

$$y_{l,k}\epsilon\{0,1\}, \forall l\epsilon L^M, \forall k\epsilon\{1,2,\ldots,K\} \tag{3.7}$$

**Remarks:**

- The objective function (3.3) of the Mixed Integer Program (MIP) tries to minimize the total cost of hosting the VN. The first term in the objective function calculates the total cost of hosting the virtual nodes, while the second and the third terms calculate the cost of hosting the virtual links.
- The unique assignment constraint (3.4) ensures that each island is assigned to exactly one VN bidder.
- The Link Assignment Consistency Constraint (3.5) ensures that a link is assigned to the VN bidder that hosts both its endpoints.

### Linear Program Relaxation and Deterministic Rounding Algorithm

Solving the mixed-integer program for VN partitioning is computationally intractable [54]. Hence, we relaxed the integer constraints (3.6) and (3.7) to obtain the following linear program (LP_RELAX),

**Relaxed Linear Program, LP_RELAX**

**Variables:**

- $x_{i,j}$: A binary variable which has the value of $1$ if island $i$ is assigned to bidder $j$
- $y_{i,j}$: A binary variable that has the value of $1$ if both end-nodes of meta-link $i$ are assigned to bidder $j$.

**Objective:**

*minimize*

$$\sum_{i\epsilon I^M}\sum_{k\epsilon\{1,2,...,K\}} x_{i,k}c_k^I(i) + \sum_{l\epsilon L^M}\sum_{k\epsilon\{1,2,...,K\}} y_{l,k}(c_k^L(l) - \acute{C}^L) + |L^M|\acute{C}^L$$

**Constraints:**

-Unique Assignment Constraint

$$\sum_{k\epsilon 1,2,...,K} x_{i,k} = 1, \forall i\epsilon I^M$$

-Link Assignment Consistency Constraint

$$y_{l,k} \le x_{n,k} \forall k\epsilon\{1,2,...,K\}, l = (n1, n2), \forall n\epsilon\{n1, n2\}$$

-Domain Constraints

$$0 \le x_{i,k} \le 1, \forall i\epsilon I^M, \forall k\epsilon\{1,2,...,K\} \tag{3.8}$$

$$0 \le y_{l,k} \le 1, \forall l\epsilon L^M, \forall k\epsilon\{1,2,...,K\} \tag{3.9}$$

**Deterministic Rounding-Based VN Partitioning Algorithm (LP-D):** Once we have the fractional solution (shown in line 1 of algorithm 1) of the relaxed linear program, LP_RELAX, we apply our deterministic rounding technique to obtain integer values ($0$ or $1$) for the variable $x$, i.e, determine the mapping for each island to a VN bidder.

For each island $i$, the maximum fractional value is rounded to 1, i.e, $i$ is assigned to bidder $k$, such that the value of $x_{i,k}$ is maximum. Ties are broken towards the bidder who offers the lower price for hosting the island (line 7).

The solution for the relaxed linear program, LP_RELAX, can be reached in polynomial

**Algorithm 1**: Linear Programming Relaxation and Deterministic Rounding

```
Procedure: LP-D($G^M = (I^M, L^M)$)
solve LP_RELAX;
foreach $i \epsilon I^M$ do
    max ← 0.0;
    max_mapping ← 0
    foreach $k \epsilon \{1, 2, ..., K\}$ do
        if (max = 0.0 || $x_{i,k} \geq$ max) then
            if (max ≥ 0.0 & $x_{i,k}$ = max & $c^I_k(i) < c^I_{max\_mapping}(i)$) then
                max ← $x_{i,k}$
                max_mapping ← k
            end
            else if (max ≥ 0.0 & $x_{i,k}$ > max) then
                max ← $x_{i,k}$
                max_mapping ← k
            end
        end
    end
    $\mathcal{M}_N(i)$ ← max_mapping
end
```

time. Therefore, it is easy to see that LP-D runs in polynomial time.

### 3.4.3 Greedy Local Search

The VN partitioning algorithm is a complex optimization problem. Many simplifying assumptions could be made to make it tractable. At one end of the spectrum lies a partitioning problem that only considers link costs and tries to minimize total cost by simply minimizing the *cut-size*, the number of links that cross domain borders. Unfortunately, this way we only reduce the problem to the NP-Hard $k$-cut problem [37]. At the other end of the spectrum, lies a greedy assignment of islands to bidders who quotes the lowest price, without considering link costs.

We present two variants, G-MinIslandCost and G-MinCutSize, of the greedy local search algorithm (outlined in algorithm 2) that start from the following two greedily formed configurations or mapping functions that resemble the extreme cases discussed above. The algorithm then performs local improvements to obtain a mapping function (or assignment of the islands) that minimizes the total VN hosting cost.

1. **MinIslandCost:** Each island is assigned greedily to the bidder with lowest quoted price, resulting in the initial mapping function,

$$\mathcal{M}_{N}(\boldsymbol{i^M}) = \boldsymbol{argmin_k}\{\boldsymbol{c_k^I}(\boldsymbol{i^M})\}, \forall \boldsymbol{i^M} \epsilon \boldsymbol{I^M}$$

2. **MinCutSize:** Each VN bidder bids for a part or a subgraph of the VN topology. The size of such a *desired subgraph* (desired by the bidder) can range from a single island to the entire VN. The MinCutSize forms its initial mapping by the following iterative method: In each iteration, it finds the largest desired subgraph (with the highest number of islands) that exclusively consists of islands that are yet to be assigned, and assigns each island of the subgraph to the bidder of that subgraph. In case of ties (equal subgraph sizes), the islands are assigned to the bidder who quotes the lowest total price (for the islands and the links) for the subgraph. This process is continued until no desired subgraph can be found that consists of only unassigned islands. The remaining islands are then greedily assigned using the MinIslandCost approach.

Note that the two variants of the algorithm only differ in the starting configurations. The algorithm starts by forming a starting configuration, in line 1. It then performs an iterative local search for a mapping function that yields the least total estimated cost. During an iteration, each island is considered separately to find the best alternative mapping (bidder) for it. Candidate bidders are compared using the *Gain* metric[2]. For an island $\boldsymbol{i^M}$, the value of gain corresponding to bidder $\boldsymbol{B}$ is proportional to the cost reduction that can be achieved if the assignment of $\boldsymbol{i^M}$ were to be changed to $\boldsymbol{B}$ (from its current assignment, $\mathcal{M}_{N}(\boldsymbol{i^M})$). It is calculated in line 6 using the formula, $\boldsymbol{Gain_B}(\boldsymbol{i^M}) =$

$$\beta[\sum_{a \epsilon A, \mathcal{M}_N(a)=\mathcal{M}_N(i^M)} c^L_{\mathcal{M}_N(i^M)}(i^M, a) + \sum_{a \epsilon A, \mathcal{M}_N(a) \neq \mathcal{M}_N(i^M)} \acute{C}^L - \sum_{a \epsilon A, \mathcal{M}_N(a)=B} c^L_B(i^M, a) - \sum_{a \epsilon A, \mathcal{M}_N(a) \neq B} \acute{C}^L] \tag{3.10}$$

$$+\alpha[c^I_{\mathcal{M}_N(i^M)}(i^M) - c^I_B(i^M)] \tag{3.11}$$

---

[2] This heuristic is inspired by the proposals, [44] and [32], for solving the graph bipartitioning problem

**Algorithm 2**: Greedy Local Search Based VN Partitioning

```
1  Form Initial Mappings;
2  for iteration ← 1 to m do
3      terminate ← 1
4      foreach $i^M \epsilon I^M$ do
5          foreach Bidder $B$, such that $\mathcal{M}_N(i^M) \neq B$ do
6              $Gain_B(i^M) \leftarrow$ calculateGain($i^M, B$)
           end
7          $B_{max} \leftarrow argmax_B\{Gain_B(i^M)\}$
8          if $Gain_{B_{max}}(i^M) > 0$ then
9              $\mathcal{M}_N(i^M) \leftarrow B_{max}$
10             terminate ← 0
           end
       end
11     if terminate = 1 then
12         return
       end
   end
```

where, $A = \{a \mid a \text{ is a neighbor of } i^M\}$

The first part of the formula (equation 3.10) calculates the decrease in estimated link (both intra and inter-domain) hosting cost, while the second (equation 3.11) calculates the decrease in estimated island provisioning cost. $0 \leq \alpha, \beta \leq 1$ are tuning parameters, that can be used to reflect other SP preferences than monetary cost minimization. We explore this in the next section.

Each island is assigned to the bidder that offers the highest positive gain (line 9). If no positive gain move is found for any island during an iteration, the algorithm is terminated (lines 10-12).

The algorithm runs for at most $m$ iterations. During each iteration, all islands are considered for new mapping. For each island, the algorithm calculates gains for all bidders, except that to which it is currently assigned. The gain calculation for a bidder considers all the virtual links from an island. Therefore, the worst case time complexity of the algorithm is $O(m|I^M|K|L^M|)$

### 3.4.4 Bidder Reputation in VN Partitioning

Not all infrastructure providers in a VNE are equally reputed. Reputation of a bidder is built over time based on many factors including quality of service, price, scale of the physical infrastructure, and commitment to customers. The VNE marketplace will consist of a mixture of reputed and non-reputed new InPs, and offering low prices is often a technique adopted by new and non-reputed enterprises to get entry into a market. Therefore, the SP has to make a tradeoff between provider reputation and cost.

As an extension of our work, we explore the impact of bidder reputation on the VN partitioning algorithms. Note that mechanisms used to measure and manage the reputation of providers are beyond the scope of our work. We assume that the knowledge about a bidder's reputation is available in the form of a *reputation factor* $\mathcal{R}_k$, $\forall k \epsilon \{1, 2, ..., K\}$.

#### Bidder Reputation in LP-D:

In order to consider the bidders' reputation, we modified the objective function of LP_RELAX (equation 3.3) as follows:

$$\sum_{i \epsilon I^M} \sum_{k \epsilon \{1,2,...,K\}} x_{i,k} \frac{\gamma c_k^I(i)}{\mathcal{R}_k} + \sum_{l \epsilon L^M} \sum_{k \epsilon \{1,2,...,K\}} y_{l,k} \left( \frac{\gamma c_k^L(l)}{\mathcal{R}_k} - \acute{C}^L \right) + |L^M| \acute{C}^L \tag{3.12}$$

By dividing the quoted prices for hosting both the islands and the links, we ensured that the partitioning process will have a bias towards more reputed bidders. $0 < \gamma <= 1$ is a tuning parameter that can be used to control this bias, i.e, express the weight of an SP's preference for reputed bidders relative to that for minimum total cost.

#### Bidder Reputation in the Greedy Local Search Algorithm:

To introduce bidder reputation to the local search based algorithm, we modified tuning parameters $\alpha$ and $\beta$ for gain calculation in equations 3.10 and 3.11 as:

$$\beta = \gamma \times \acute{L} \times (\mathcal{R}_B - \mathcal{R}_{\mathcal{M}_N(i^M)}) + 1 \tag{3.13}$$

$$\alpha = \gamma \times (\mathcal{R}_B - \mathcal{R}_{\mathcal{M}_N(i^M)}) + 1 \tag{3.14}$$

where, $\boldsymbol{A = \{a \mid a}$ is a neighbor of $\boldsymbol{i^M\}}$, $\boldsymbol{\acute{L} = \sum_{a \epsilon A, \mathcal{M}_N(a)=B} 1 - \sum_{a \epsilon A, \mathcal{M}_N(a)=\mathcal{M}_N(i^M)} 1}$ = the increase in the number of intra-domain links if the mapping for $\boldsymbol{i^M}$ were to change to bidder $\boldsymbol{B}$, and $\boldsymbol{0 < \gamma <= 1}$ is a tuning parameter that can be used to express the importance of reputation relative to cost. By multiplying the two factors of the gain (equations 3.10 and 3.11) with the difference in reputation between the current and candidate bidders, we ensure that each local move in our heuristic based local search will have a tendency, proportional to the reputation difference, to go towards the bidder with higher reputation.

# Chapter 4

# Performance Evaluation

In this chapter, we evaluate the performance of V-Mart's VN graph partitioning algorithms through simulation experiments. Our focus is to exhibit that these algorithms can be applied to a wide variety of virtual environments. We measure total cost, execution times, observed difference with the base case solution and average partition sizes for the algorithms. We vary the parameters of the VNE operational and market model to show that the algorithms perform well under various market conditions. We also study the impact of incorporating the bidders' reputation into the algorithms.

## 4.1 Compared Algorithms

We explore six algorithms in our experiments. These are summarized in table 4.1.

An exhaustive search for the mapping function that yields the minimum total cost has a complexity of $\boldsymbol{O(N^K)}$, where $\boldsymbol{N}$ is the total number of islands in the VN meta-graph and $\boldsymbol{K}$ is the number of bidders in the market. Therefore, even for a very small VN topology (say, with 30 nodes) and a small number of bidders (say, 10) finding the optimal solution takes hours. Therefore, we compare our algorithms, G-MinIslandCost, G-MinCutSize and LP-D with the relaxed linear program, LP_RELAX. The fractional solution (the achieved optimal value for the objective function 3.3) produced by relaxed linear program is guaranteed to be less than or equal to the minimum total cost yielded by the mixed-integer program. We use the GNU Linear Programming Kit [4] to solve LP_RELAX.

For large input graphs we compare our algorithms with an implementation of a simple

Table 4.1: Summary of Compared VN Partitioning Algorithms

| **Algorthm Name** | **Description** |
|---|---|
| G-MinIslandCost | Greedy Local Search Based VN Partitioning starting from a greedy assignment of islands to bidders with the lowest bid |
| G-MinCutSize | Greedy Local Search Based VN Partitioning starting from a greedy assignment of subgraphs based on its size |
| LP-D | Linear Program relaxation with deterministic rounding |
| LP_RELAX | Relaxed Linear Program of the Mixed-Integer Program for VN Partitioning |
| GA-Vanilla | A simple genetic algorithm for VN partitioning |
| G-GA | Greedy Local Search Based VN Partitioning starting from multiple starting configurations, selected through the application of genetic algorithm, GA-Vanilla |

genetic algorithm [36] for VN Partitioning, called GA-Vanilla (appendix A). The successful use of genetic algorithms (GA) as tools for solving many complex optimization problems [36,50], specifically graph partitioning [22] and VLSI circuit partitioning [48], inspired our choice. We also compare the algorithms with a third version of our greedy local search algorithm, G-GA. Rather than starting from a single starting configuration, the G-GA algorithm uses multiple starting points and takes the best solution after performing a local search from each. The starting configurations are selected through an application of GA-Vanilla.

## 4.2 Metrics

We use the following five metrics to evaluate the performance or the algorithms.

1. **Total Cost:** The total cost for hosting the entire virtual network is calculated using equation 3.2. This value is not the amount the SP has to pay, rather it is based on price estimates quoted by the VN bidders in the first round auction. However, as the actual cost for each segment of the VN, which is fixed after the second round auction, is upper-bounded by the costs that are calculated during the VN partitioning, the SP wants to minimize this value nonetheless.

2. **Observed Approximation Ratio:** We define the observed approximation ratio for an algorithm as the ratio between the total cost produced by it and that by LP_RELAX. The fractional solution produced by LP_RELAX is guaranteed to be better than or equal to the minimum total cost. Therefore, a high observed approximation ratio for an algorithm indicates that it produces a total cost that has a high difference with the minimum. The optimal value (one that is associated with the minimum total cost) of the observed approximation ratio for an algorithm is 1.

3. **Execution Time:** We calculate total time (in milliseconds) each algorithm takes to calculate the minimum cost assignment.

4. **Average Partition Sizes:** We measure the average sizes of the resulting partitions. Our main objective is to minimize the total price that the SP has to pay. So, first and foremost, the VN partitioning process has to minimize the total estimated price. But, it also has to consider the bidders' preferences. A partitioning that produces segments that are not consistent with the volume or configuration desired by the bidders, does not ensure good offers/bids in the second round auction. Larger average partition sizes preferred by bidders who adopt volume based pricing; but it is not as important to one who adopts resource-wise pricing.

5. **Percentage Increase in Total Cost:** We also measure the percentage increase in total cost that results from incorporating the bidders' reputation factors into an algorithm. Note that the total costs for both versions (with or without reputation factors) of all the algorithms are calculated in the same way (using equation 3.2), expect LP_RELAX, which calculates the fractional solution using equation 3.12. We measure the percentage increase in total cost to exhibit the impact of incorporating bidders' reputation factors into the algorithms.

## 4.3 Experiment Settings

We evaluate the performance of the V-Mart algorithms using four sets of experiments to cover a wide variety of market and operational models. The virtual network topologies, used in all of these experiments, are random, flat and connected graphs generated with the GT-ITM tool [62], where each pair of nodes in a graph are connected with a probability of

0.05. We ran each experiment on *five* different random input graphs, and displayed the average values taken after removing the highest and the lowest. We opt for random graphs to represent virtual network topologies for two reasons. *First*, as the concept of virtual network environment is fairly new, a concrete operational model, that specifies the nature of the virtual network topologies, is yet to be established. And *second*, we believe that such a fixed model is not likely to emerge, as VN topologies are expected to be more diverse than the current underlays. Note that we consider the input graphs as the meta-graphs; therefore each node in the graph represents an island.

The sets of experiments that we conducted are:

1. **Varying the size of the virtual network topology:** In this set of experiments, we explore the impact of the size of a virtual network topology on each of the algorithms. These experiments are divided into two parts. The first part deals with relatively smaller input graphs, where the number of nodes vary from 5 to 150. The number of bidders in the market is set to 20. The estimated prices of virtual resources are uniformly distributed between 10 and 100. We compare the G-MinIslandCost, G-MinCutSize, LP-D and LP_RELAX and show the results in figure 4.1. In the second part of this experiment, we examine the performance trends over very large (up to 1000 nodes) virtual network topologies. The number of bidders for these cases is set to one-fifth of the number of nodes in the graph. We show total costs and execution times in figure 4.2. The average size of partitions grow in a trend similar to that of figure 4.1(d) (which we do not show here).

2. **Varying the Acceptance Percentage:** We define the *acceptance percentage* of a virtual island as the percentage of bidders who want to host it, i.e, a 70% acceptance percentage for an island denotes that 70% bidders quoted a price for it. This value depends on a number of factors, including, performance constraints on the resource, its location, load on the providers' networks, or simply a bidder's business preference. For the previous set of experiments we fixed this value to 70% for all islands. In this set, we vary the value from 10 to 80 percent. We use input graphs with 70 nodes and consider 20 bidders. The quotes on virtual resources by the bidders are, again, taken from a uniform distribution with maximum and minimum values of 100 and 10, respectively. The results are shown figure 4.3.

3. **Varying the ratio between the price estimates for virtual islands and for virtual links:** In the previous experiments we picked the virtual link and island price quotes from the same distribution, representing a small class of pricing models (e.g, volume based pricing). However, in a resource-wise pricing model, virtual links and islands may not have similar prices. In this set of experiments, we vary the price ratio between islands and the links, from 1:10 to 50:1, to explore the applicability of the algorithms under different provider pricing models. We fix the size of the input graphs to 70 islands and the number of bidders to 20. The results are shown figure 4.4.

4. **Bidders' Reputation:** Thus far we have assumed that the customers are only concerned with the total cost and not the reputation when selecting the bidders. Therefore, we set the values of the tuning parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ to one. In the final set of experiments we introduce the bidders reputation into the algorithms as described in section 3.4.4.

   For this part, we do not assume any relation between a bidder's reputation and his pricing mechanism. Neither do we make any assumption on how an SP's preference for reputed bidders and that for low total costs are related. We only show how the inclusion of reputation affects the performance of the algorithms. The results are shown in figure 4.5. For these experiments, the number of islands in the input graph vary from 5 to 150. The number of bidders is set to 20, and each bidder is assigned a reputation factor between 1 and 10 at random. The acceptance probability of an island is set to 70% probability. The quotes on virtual resources are uniformly distributed between 10 and 100.

## 4.4 Observations

In this section we present our observations from the experiments.

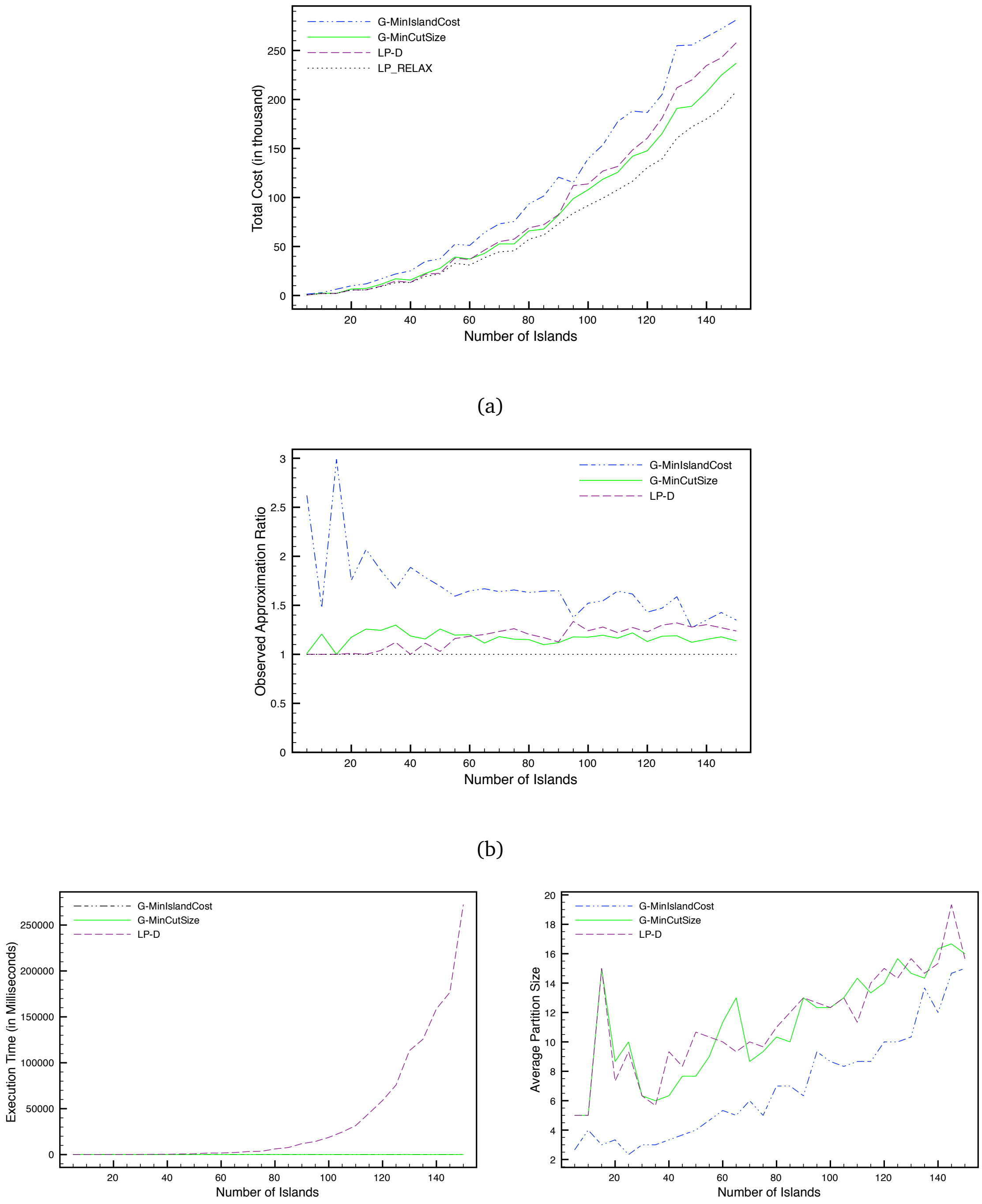
(i) **The proposed algorithms produce near optimal results.** It can be seen from figure 4.1(a) that all three proposed algorithms minimize the total cost of hosting a virtual network. For small input graphs (up to 60 nodes) the LP-D algorithm produces results that are very close to the optimal, while the G-MinIslandCost yields the highest cost. Also, the G-MinIslandCost produces partitions which are, on the average, much smaller compared to the remaining algorithms (figure 4.1(d)). Both results indicate a larger cut-size in G-MinIslandCost's solution compared to the other algorithms. But, as the graph size increases a high number of inter-domain links becomes an inevitable part of the final solution for all algorithms. Therefore, the differences between the total costs and the average package sizes reduce. The G-MinCutSize outperforms the rest because it looks up a solution in the vicinity of the largest average partition size and, thus, the lowest cut-size. Overall, for a sizable input graph all three algorithms achieve total costs that are close to the optimal (figure 4.1(b)), with the G-MinCutSize settling within 1.2 times, LP-D within 1.3 and G-MinIslandCost within 1.5 times the fractional solution.

(ii) **The heuristic-based algorithms are highly scalable, but LP-D is not.** The execution time of LP-D grows rapidly with the size of the topology ( figure 4.1(c)), making it unsuitable for very large virtual network topologies. In contrast, the heuristic based algorithms are highly scalable; this is apparent from the steady rise in execution times (figure 4.2(b)) and the total costs (figure 4.2(a)). We make three observations while comparing the performance of the algorithms for large virtual networks.

   (a) The G-GA outperforms both G-MinIslandCost and G-MinCutSize in terms of total cost. But, the difference is negligible. Moreover, it takes much longer to complete (figure 4.2(b)).
   (b) G-MinCutSize performs better than G-MinIslandCost for all sizes of the VN topology. But, the difference becomes quite small for very large VN topologies.
   (c) All three local search algorithms yield total costs that are less than half of that by GA-Vanilla.

   Under the assumption of high inter-domain link costs, the optimal solution lies in the locality of the minimum cut configuration. Therefore, the G-MinCutSize and G-GA, which perform a local search starting from a cut minimizing mapping, perform the best. In this context, a genetic algorithm is not a suitable local search tool; because, even if a cut-minimizing configuration is part of a parent population, the crossover and mutation operators are likely to produce a solution that is far away from it.

(iii) **The algorithms converge for low acceptance percentage.** A *feasible assignment* contains no mapping from an island to a bidder that has not quoted a price for it. For



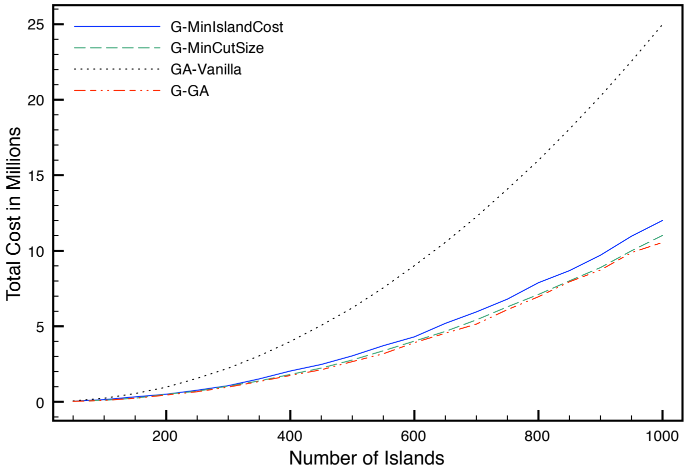
(a)

(b)

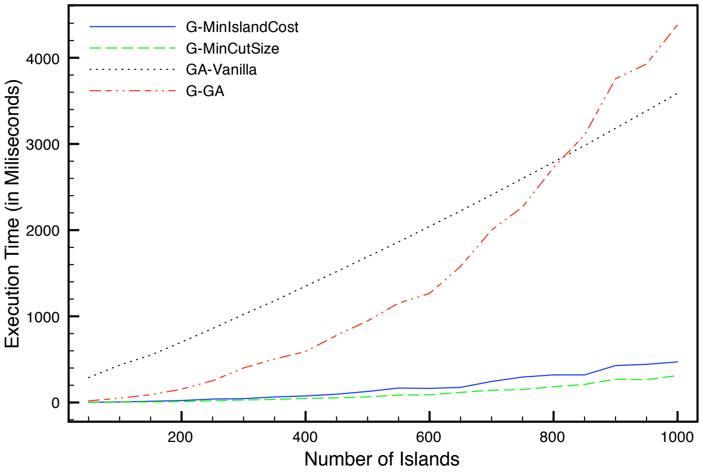
(c)

(d)

Figure 4.1: Varying the size of the input graphs, (a) Total Cost, (b) Observed Approximation Ratios, (c) Execution times, (d) Average Partition Sizes



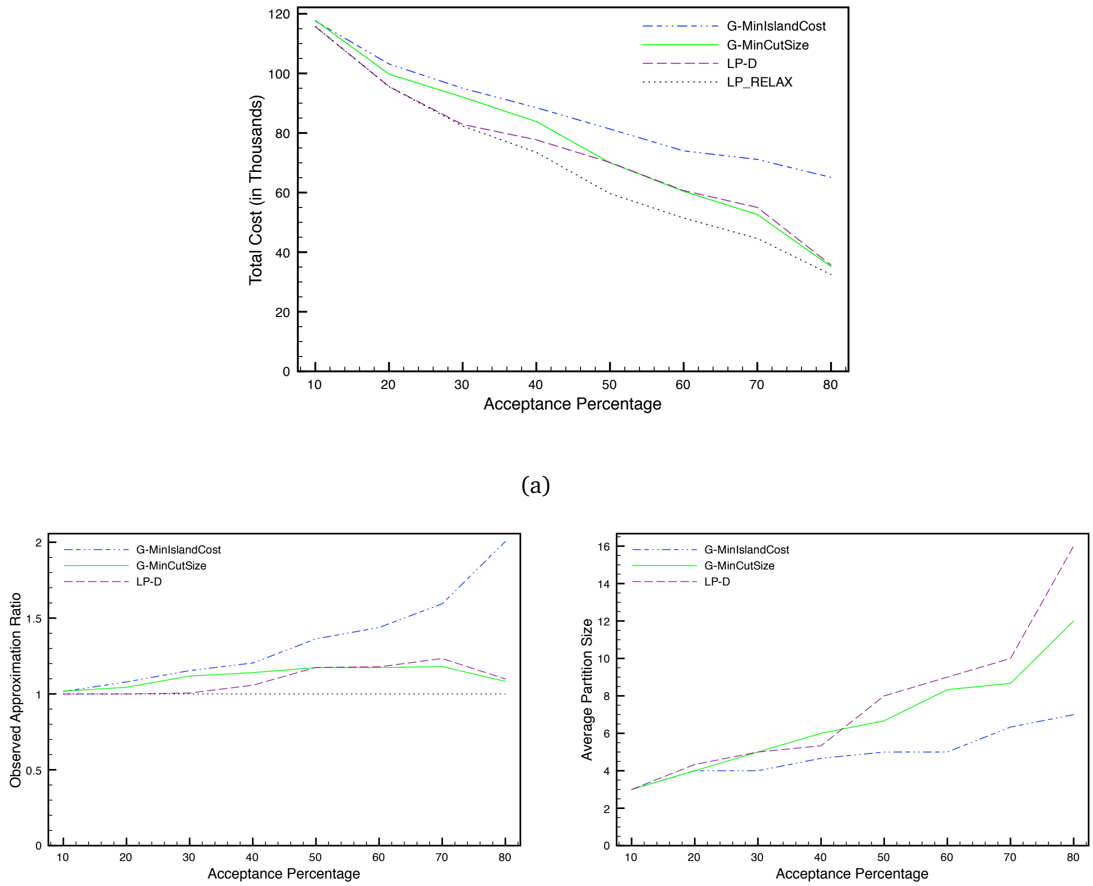
(a)



(b)

Figure 4.2: Scalability (a)Total Cost, (b) Execution Times

low acceptance percentages, few feasible assignments exist. So, all the algorithms converge to similar final configurations. Therefore, the total costs and the resulting average partition sizes produced by the algorithms are close to each other, as well as, to the optimal solution (figures 4.3(a), 4.3(b), and 4.3(c)). As the percentage goes higher, more and more bidders bid for hosting large portions of the virtual network.



(a)

(b)

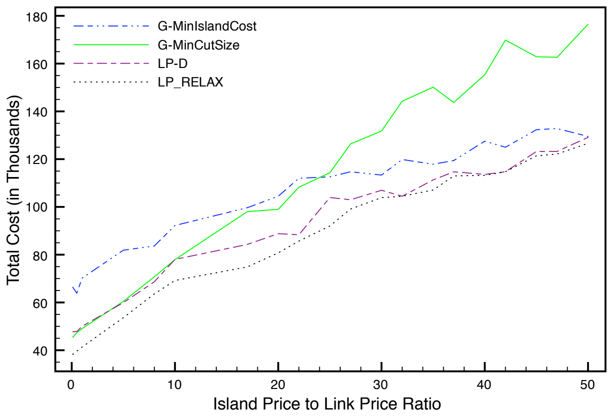
(c)

Figure 4.3: Varying the Probability that an InP will be willing to host a given Island (a) Total cost, (b) Approximation Ratios, (c) Average Partition Sizes

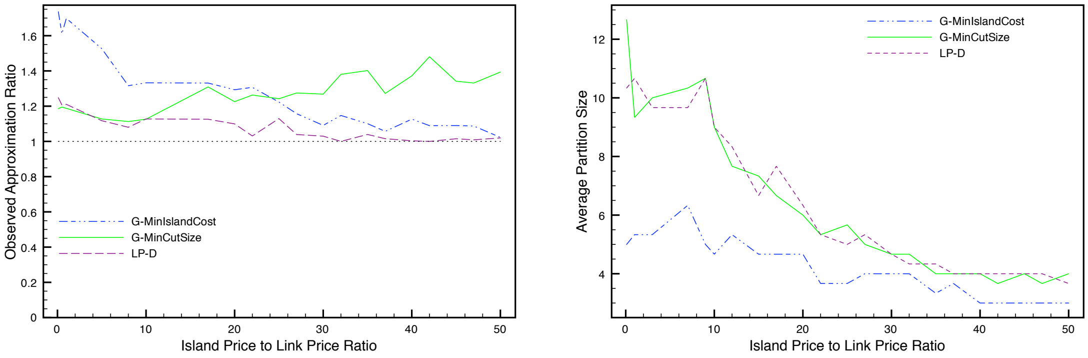
G-MinCutSize and LP-D perform better with higher acceptance percentages as they have a bias for larger partitions. The G-MinIslandCost soon parts company with the others because of its bias towards lower priced islands.

(iv) **Diverse pricing models are supported by the V-Mart's partitioning algorithms.** We divide our observations into two parts:

- A resource-wise pricing model is best represented by a island price to link price ratio of greater than or lesser than 1. The LP-D performs consistently over different island price to link price ratios (figure 4.4(a)). As the ratio rises, the impact of the



(a)



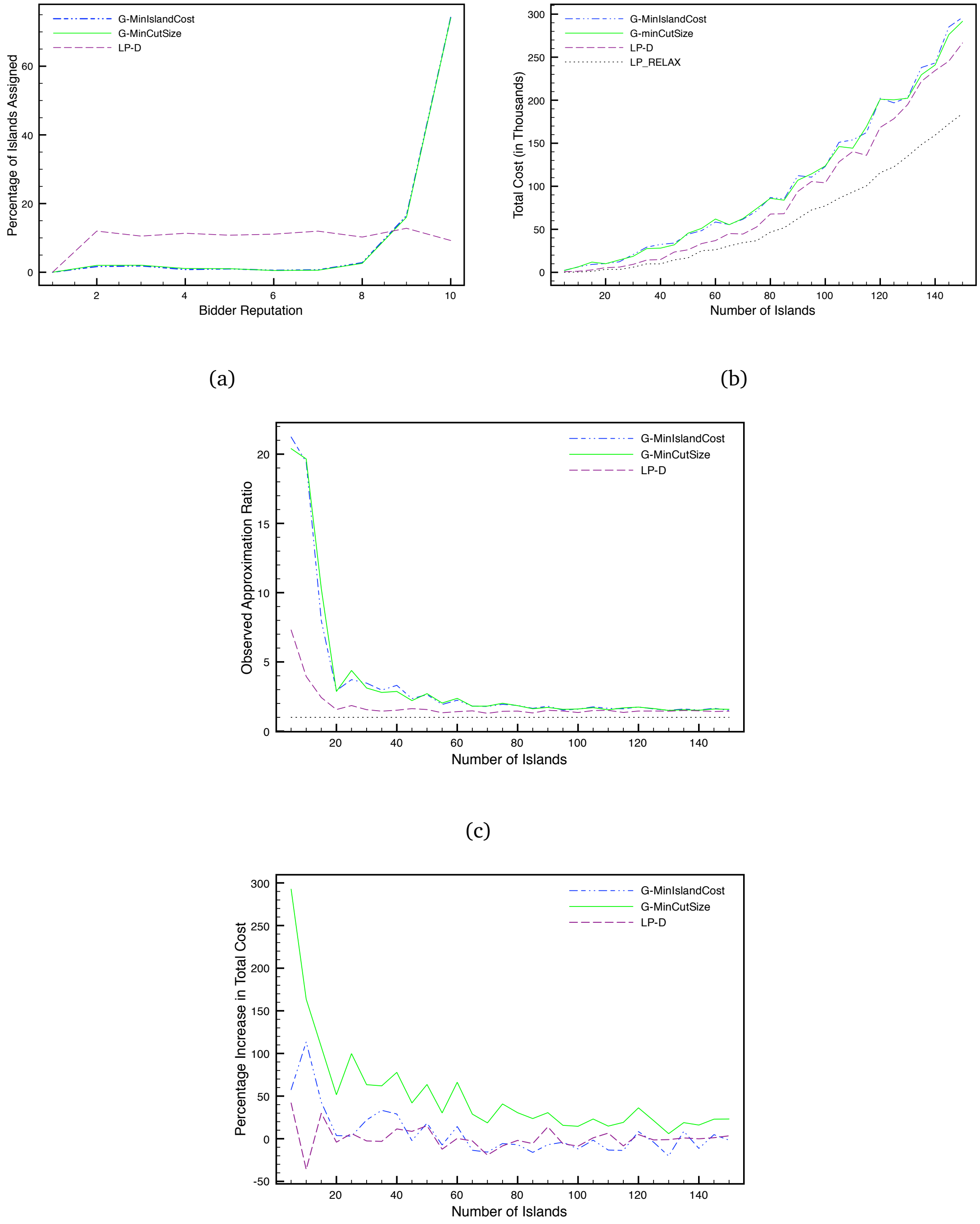
(b)

(c)

Figure 4.4: Varying The Ratio between Island Hosting cost and Link Hosting Cost (a) Total Cost, (b) Approximation Ratios (c)Average Partition Sizes

cut-size on the total cost begin to diminish and the impact of island hosting prices become more prominent. Therefore, the G-MinCutSize begins to deteriorate and G-MinIslandCost starts to improve. Finally G-MinIslandCost outperforms its local search based companion as the ratio reaches around 25:1. After this point the reduction in island cost by a move can be large enough to offset an increase in inter-domain link cost. This also results in the reduction in average package sizes for all the algorithms (figure 4.4(c)).

- In general, volume-based pricing models, e.g, package pricing and volume dis-



(a)

(b)

(c)

(d)

Figure 4.5: Preference for Reputed Bidders, (b) Total Cost (c) Approximation Ratio (a) Percentage of Islands Assigned, (d) Percentage Increase in Total Cost

counts, are better supported by G-MinCutSize and LP-D because of their bias towards larger partitions (figures 4.1(d), 4.3(c) and 4.4(c)). A volume-based pricing model is best represented by an island price to link price ratio of close to 1.Effectively, this introduces a high difference between the inter-domain link costs and the island hosting costs. G-MinCutSize and LP-D both perform well for these values (figures 4.4(a) and 4.4(b)). For package-pricing models, G-MinCutSize is the most suitable algorithm, as it assigns packages, starting with the largest, to the lowest price bidders. G-MinCutSize considers moving only one island at a time; therefore, a package is only broken if the reduction in price for moving an island is large enough to offset the increase in cut-size. This is highly unlikely for island price to link price ratios close to 1. Therefore, G-MinCutSize ensures the preservation of packages for bidders who bid low prices for large packages.

(v) **The local search algorithms display a bias towards more reputed bidders.** Inclusion of the bidders' reputation factors in the tuning parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ of the heuristic-based algorithms forces most of the islands to be assigned to bidders with high reputation (figure 4.5(a)). Here, G-MinCutSize's bias towards larger partitions is overridden by that for reputed bidders; because, the decrease in island hosting cost, multiplied by the increase in bidders' reputation, easily overcomes the increase in inter-domain link cost. Therefore, the G-MinIslandCost and the G-MinCutSize converge to similar mapping functions (figure 4.5(b)). The change in bias for G-MinCutSize contributes to a high increase in total cost from the initial version (settles to around 35%, as shown in figure 4.5(d)). However, the increase is lesser (settles to less than 20%) for G-MinIslandCost, as both versions assign similar importance/weights to reducing the number of inter-domain cuts. LP-D shows a more moderate bias, and, therefore, a much smaller increase in total costs from the initial version (figure 4.5(d)). All three algorithms settle to within 1.6 times the optimal solution (figure 4.5(c)). Note that the optimal value of the objective function of this version of LP_RELAX is lesser than the previous version's. Because, each virtual resource price is divided by the bidder's reputation. This is the reason behind the initial spike in the observed approximation ratios for all the algorithms. But, as the number of islands grows, the inter-domain link costs become the most significant part of the total cost, which are not affected by the reputation factors; so the observed approximation ratio settles to a smaller value.

## Chapter 5

# Conclusion

Network virtualization environment (VNE) affords great business and technological flexibilities to service providers and infrastructure providers. Under the current network model, a service provider has access to a few local infrastructure providers who offer end-to-end service delivery by peering with other InPs in private. Such a business arrangement is not always fair, as (i) service providers have no involvement in fixing the price for the deployment of their services, and (ii) the market is dominated by the *big players* and new infrastructure providers find it difficult to gain entry in the market. But, VNE allows any InP to participate in a fair competition and SPs to enjoy great flexibility in which InPs they can contract with.

With a view to enabling such flexibility, we presented V-Mart, a framework for automated service negotiation and contracting in this environment. There are two major parts of V-Mart: (i) an auction-based market where SPs and InPs can participate in open negotiations, and (ii) a VN partitioning tool which allows the SP to resolve the competition, and contract with the best set of InPs. We conclude our discussion in this chapter with a summary of V-Mart's contributions, and our plans for future.

## 5.1 Summary of Contributions

The contributions of the proposed framework are summarized below:

- **V-Mart is an open market**. The local infrastructure providers are not the only participants in V-Mart's auction. Rather, any provider (InP) has the option to compete in the V-Mart auction, and all bidders have the opportunity to win parts of a virtual network.

- **V-Mart promotes truthfulness, with selective information sharing**. V-Mart, through its two stage strategy-proof auction mechanism, ensures that the SP is be free from price manipulation from the InPs. However, it also ensures that the InPs are able to participate without disclosing sensitive information (e.g., physical network topology, physical resource capacity and configurations and pricing model).

- **The framework is flexible**. A SP enjoys the flexibility to specify the levels of services he would accept (through RFQs), not select from a pre-specified list. An InP, on the other hand, is able to employ any business policy (e.g., pricing mechanism) that best suits his purpose.

- **It is incentive compatible for both parties**. V-Mart's auction model and the VN partitioning algorithms attempt to minimize virtual network provisioning cost for the SP, but the two-stage, sealed auction model is chosen to guard against cases where the InPs gain no profit, therefore has no incentive to participate in the market. Moreover, the VN partitioning algorithms are designed to suit various InP preferences.

- **Service Negotiation and Contracting processes are automatic and efficient**. The negotiation process (auction) and the winner determination or contracting decision process (partitioning algorithm) are automatic and efficient, to allow fast and dynamic virtual network creation.

- **V-Mart is applicable to a wide variety of markets**. Although it is designed in the VNE context, V-Mart's graph representation, auction processes and partitioning mechanisms can equally be applied elsewhere. It is suitable for vendor/provider selection in many service markets including business process outsourcing, cloud computing, VPNs, P2P services and grids in a multi-providers setting. Practically it can handle any service which is a combination of other correlated services and the correlation can be represented as edges in a graph.

## 5.2 Future Plans

We have the following future plans for this framework.

1. **Theoretical Analysis**: V-Mart establishes most of its features through quantitative

or qualitative analysis. As part of our future work, we intend to take a theoretical approach to further assess V-Mart. These include

   (a) An outline of the proof that the proposed auction model is strategy-proof is provided in section 3.3. We intend to provide a formal proof of this important property.
   (b) We present the observed approximation factors of V-Mart's VN partitioning algorithms in chapter 4. In future, we intend to establish theoretical approximation factors and bounds on the performance of the partitioning algorithms
   (c) We plan to evaluate the framework using the following criteria [53]:
       i. Social welfare and/or Pareto efficiency: measure the *global good* of the SP and the VN bidders in the market.
       ii. Individual rationality: evaluate whether each individual (SP or InP) has a rationale to participate in V-Mart.
       iii. Distribution and Communication efficiency: evaluate the overhead of V-Mart.
   (d) As another alternative algorithm for the VN partitioning problem, we also intend to explore available approaches used to directly solve mixed integer programs.

2. **Observe real-world performance**. The framework was evaluated in an in-house simulated setting. We plan to evaluate V-Mart in a much larger setting. A possible option is to simulate the market on PlanetLab. By logically dividing a slice into multiple domains (InPs), we can simulate a multi-domain VN hosting market. This way we can evaluate V-Mart under real network load, apply VN embedding algorithms to generate quotes, and observe the behavior patterns under different pricing models.

# APPENDICES

# Appendix A

# Simple Genetic Algorithms for VN Partitioning

Here we describe our implementation of a *simple* genetic algorithm, GA-Vanilla. The basic steps of a simple genetic algorithm are outlined in algorithm 3). The operations in the simple genetic algorithm: selection, crossover and mutation, are outlined in the following.

### Representation Scheme and Fitness Function:

The first step in designing a genetic algorithm is to devise a representation scheme, i.e, a way to represent individuals in the population, that suits the problem. As a representation scheme we use a $\boldsymbol{K}$-ary string of length $|\boldsymbol{I^M}|$ that denotes a mapping function $\mathcal{M}_{\boldsymbol{N}}$, i.e., individual $\boldsymbol{S = s_1 s_2 ... s_{|I^M|}}$ such that, $\boldsymbol{s_i} = \mathcal{M}_{\boldsymbol{N}}(\boldsymbol{i}) =$ bidder $\boldsymbol{k \epsilon \{1, 2, ..., K\}}$ to which $\boldsymbol{i}$ is assigned.

As the fitness of an individual, we use the inverse of the total cost for the individual or mapping function, i.e, $\boldsymbol{fitness(S)} = \frac{\boldsymbol{1}}{\mathbb{C}(\boldsymbol{G^M})}$, $\mathbb{C}(\boldsymbol{G^M})$ is defined as equation 3.2.

### Initial Population:

We randomly select *feasible* individuals (in line 2 of algorithm 3) to form the initial population, $\boldsymbol{Pop_0 = \{S_j | j \epsilon 1, ..., popSize\}}$. Note that a bidder may not always bid for all the islands in the meta-graph, therefore, any assignment for an island is not guaranteed to be feasible. A feasible string or individual is one that contains no mapping from an island to a bidder that has not quoted a price for it. We also insert the two extreme points (MinIslandCost

**Algorithm 3**: A GA for Virtual Network Partitioning

```
1  generation ← 0;
2  old_population ← Generate_Initial_Population;
3  while $generation < MAX\_GENERATION$ do
4      $generation \leftarrow generation + 1$
5      $j \leftarrow 1$;
6      while $j <= popSize$ do
7          selected_population = select reproducing population from old_population;
8          parent1 ← random_select( selected_population );
9          parent2 ← random_select( seleted_population );
10         crossover( parent1, parent2, child1, child2, probCrossover );
11         child1 ← mutation(child1, probMutation, probRepair);
12         child2 ← mutation(child2, probMutation, probRepair);
13         add child1 and child2 to new_population;
14         j = j + 2;
       end
15     old_population ← new_population;
   end
```

and MinCutSize) from our heuristic-based algorithm into the initial population, as they are proven to produce good final results.

### Reproducing Population Selection Schemes

The parent selection process (line 7) selects individuals from a population who will create offspring individuals in the following generation. We use the *deterministic sampling* scheme [21] for our implementation. In this scheme, each individual has an expected number of offsprings, calculated as $\boldsymbol{e_{S_j} = popSize} \times \frac{fitness(S_j)}{\sum_{j\epsilon\{1,2,\ldots,popSize\}} fitness(S_j)}$, in the following generation. Each individual is allocated samples equal to the integer part of its expectation. The population is then sorted using the fractional part of $\boldsymbol{e_{S_j}}$, and the remaining individuals needed to fill the population are taken from the top of the list. We select this scheme because it produces a population that is a mix of the fittest individuals from the past generation and individuals selected randomly.

### Crossover and Mutation

Pairs of parents from the selected population produce two offsprings in the next generation through crossover and mutation. The parents are paired randomly in lines 8 and 9. The representation that we adopted allows a wide range of standard GA crossover and mutation operators. In our implementation, we used the one-point crossover [36] operator with crossover probability *probCrosover*, and a random crossover point.

Clearly, the crossover operator may produce individuals that are not feasible. Therefore, we probabilistically repaired the infeasible strings in the mutation operator (lines 11 and 12). We use two different probabilities, *probMutation* and *probRepair*. An infeasible mapping of an island is corrected to a randomly selected feasible one with probability probRepair. We set high values for this to ensure that the resulting population has a very small section of infeasible individuals. *probMutation*, is the probability with which a mapping or character in the string is changed to an alternative feasible mapping, selected at random.

# Bibliography

[1] Amazon Elastic Compute Cloud (Amazon EC2). `http://aws.amazon.com/ec2/`.

[2] Cisco Application eXtension Platform Overview. `http://www.cisco.com/en/US/prod/collateral/routers/ps9701/white_paper_c11_459082.html`.

[3] GENI: Global Environment for Network Innovations. `http://www.geni.net/`.

[4] GNU Linear Programming Kit. `http://www.gnu.org/software/glpk/`.

[5] Juniper Partner Solution Development Platform. `http://www.juniper.net/us/en/products-services/nos/junos/psdp`.

[6] VINI: A Virtual Network Infrastructure. `http://www.vini-veritas.net/`.

[7] Windows Azure. `http://www.microsoft.com/windowsazure/`.

[8] Amazon EC2 Service Level Agreement. `http://aws.amazon.com/ec2-sla/`, 2009.

[9] AT&T Managed Internet Service (MIS). `http://new.serviceguide.att.com/mis.htm`, 2009.

[10] Juniper Networks brings Virtualization to the Core with Industry's Most Flexible Multi-Chassis Routing System. `http://www.juniper.net/company/presscenter/pr/2009/pr_2009_02_02-17_19.html`, 2009.

[11] NTT America-Customer Support-Service Level Agreements. `http://www.us.ntt.net/support/sla/`, 2009.

[12] PlanetLab: An Open Platform for Developing, Deploying, and Accessing Planetary-Scale Service. `http://www.planet-lab.org/`, 2009.

[13] Sprint NEXTEL Service Level Agreements. http://www.sprint.com/business/support/serviceLevelAgreements.html, 2009.

[14] David G. Andersen. Theoretical Approaches to Node Assignment. Unpublished Manuscript, 2002.

[15] Thomas Anderson, Larry Peterson, Scott Shenker, and Jonathan Turner. Overcoming the Internet Impasse through Virtualization. *Computer*, 38(4):34–41, 2005.

[16] L. Andersson and T. Madsen. Provider Provisioned Virtual Private Network (VPN) Terminology. RFC 4026, March 2005.

[17] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Above the Clouds: A Berkeley View of Cloud Computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.

[18] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the Art of Virtualization. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 164–177, New York, NY, USA, 2003. ACM.

[19] Andy Bavier, Nick Feamster, Mark Huang, Larry Peterson, and Jennifer Rexford. In VINI Veritas: Realistic and Controlled Network Experimentation. In *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 3–14, New York, NY, USA, 2006. ACM.

[20] Sapan Bhatia, Murtaza Motiwala, Wolfgang Muhlbauer, Yogesh Mundada, Vytautas Valancius, Andy Bavier, Nick Feamster, Larry Peterson, and Jennifer Rexford. Trellis: A Platform for building Flexible, Fast Virtual Networks on Commodity Hardware. In *CoNEXT '08: Proceedings of the 2008 ACM CoNEXT Conference*, pages 1–6, New York, NY, USA, 2008. ACM.

[21] A. Brindle. Genetic Algorithms for Function Optimization. Unpublished Doctoral Dissertation, University of Alberta, Edmunton, 1981.

[22] Thang Nguyen Bui and Byung Ro Moon. Genetic Algorithm and Graph Partitioning. *IEEE Trans. Comput.*, 45(7):841–855, 1996.

[23] Rajkumar Buyya, David Abramson, and Jonathan Giddy. A Case for Economy Grid Architecture for Service Oriented Grid Computing. In *IPDPS '01: Proceedings of the 10th Heterogeneous Computing Workshop â"" HCW 2001 (Workshop 1)*, page 20083.1, Washington, DC, USA, 2001. IEEE Computer Society.

[24] Rajkumar Buyya, David Abramson, Jonathan Giddy, and Heinz Stockinger. Economic models for resource management and scheduling in Grid computing. pages 1507–1542. Wiley Press, 2002.

[25] Mosharaf Chowdhury and Raouf Boutaba. A Survey of Network Virtualization. Technical Report CS-2008-25, University of Waterloo, 2008.

[26] Mosharaf Kabir Chowdhury, Muntasir Raihan Rahman, and Raouf Boutaba. Virtual Network Embedding with Coordinated Node and Link Mapping. In *28th Conference on Computer Communications (IEEE INFOCOM)*, Rio de Janeiro, Brazil, April, 2009.

[27] Mosharaf Kabir Chowdhury, Fida-E Zaheer, and Raouf Boutaba. iMark: An Identity Management Framework for Network Virtualization Environment. In *The 11th IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2009.

[28] N. M. Mosharaf Kabir Chowdhury and Raouf Boutaba. Network Virtualization: State of the Art and Research Challenges. *IEEE Communications Magazine*, 47(7):20–26, July 2009.

[29] Reuven Cohen and Gideon Kaempfer. On the Cost of Virtual Private Networks. *IEEE/ACM Trans. Netw.*, 8(6):775–784, 2000.

[30] Nick Feamster, Lixin Gao, and Jennifer Rexford. How to Lease the Internet in Your Spare Time. *SIGCOMM CCR*, 37(1):61–64, 2007.

[31] Paul Ferguson and Geoff Huston. What is a VPN?, 1998.

[32] C. M. Fiduccia and R. M. Mattheyses. A Linear-Time Heuristic for Improving Network Partitions. In *25 years of DAC: Papers on Twenty-five years of electronic design automation*, pages 241–247, New York, NY, USA, 1988. ACM.

[33] Ian Foster. Globus Toolkit Version 4: Software for Service-Oriented Systems. In *IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779*, pages 2–13, 2005.

[34] Ian Foster, Carl Kesselman, Jeffrey M. Nick, and Steven Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. 2002.

[35] Ian T. Foster. The anatomy of the grid: Enabling scalable virtual organizations. In *Euro-Par '01: Proceedings of the 7th International Euro-Par Conference Manchester on Parallel Processing*, pages 1–4, London, UK, 2001. Springer-Verlag.

[36] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[37] Olivier Goldschmidt and Dorit S. Hochbaum. A Polynomial Algorithm for the k-cut Problem for Fixed k. *Math. Oper. Res.*, 19(1):24–37, 1994.

[38] Anupam Gupta, Jon Kleinberg, Amit Kumar, Rajeev Rastogi, and Bulent Yener. Provisioning a Virtual Private Network: A Network Design Problem for Multicommodity Flow. In *In Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 389–398, 2001.

[39] D. Hausheer and B. Stiller. PeerMart: The Technology for a Distributed Auction-based Market for Peer-to-Peer Services. In *Proceedings of IEEE ICC'2005*, pages 1583–1587, 2005.

[40] David Hausheer and Burkhard Stiller. Auctions for Virtual Network Environments. In *Workshop on Management of Network Virtualisation*, 2007.

[41] David E. Irwin, Jeffrey S. Chase, Laura E. Grit, Aydan R. Yumerefendi, David Becker, and Ken Yocum. Sharing networked resources with brokered leases. In *USENIX Annual Technical Conference, General Track*, pages 199–212, 2006.

[42] Alexander Keller and Heiko Ludwig. The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. *J. Network Syst. Manage.*, 11(1), 2003.

[43] Eric Keller, Ruby Lee, and Jennifer rexford. Accountability in Hosted Virtual Networks. In *ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures*, 2009.

[44] B W Kernighan and S Lin. An Efficient Heuristic Procedure for Partitioning Graphs. *Bell System Technical Journal*, (49):291–307, 1970.

[45] Tianshu Li, Youssef Iraqi, and Raouf Boutaba. Pricing and Admission Control for QoS-Enabled Internet. *Computer Networks*, 46(1):87–110, 2004.

[46] Jing Lu and Jonathan Turner. Efficient mapping of virtual networks onto a shared substrate. Technical Report WUCSE-2006-35, Washington University, 2006.

[47] Eng Keong Lua, Jon Crowcroft, Marcelo Pias, Ravi Sharma, and Steven Lim. A Survey and Comparison of Peer-to-Peer Overlay Network Schemes. *IEEE Communications Surveys and Tutorials*, 7:72–93, 2005.

[48] Pinaki Mazumder and Elizabeth M. Rudnick. *Genetic Algorithms for VLSI Design, Layout & Test Automation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.

[49] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. OpenFlow: Enabling Innovation in Campus Networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, 2008.

[50] Martin Pelikan, David E. Goldberg, and Fernando G. Lobo. A Survey of Optimization by Building and Using Probabilistic Models. *Comput. Optim. Appl.*, 21(1):5–20, 2002.

[51] Larry L. Peterson, Thomas E. Anderson, David E. Culler, and Timothy Roscoe. A Blueprint for Introducing Disruptive Technology into the Internet. *Computer Communication Review,* 33(1):59–64, 2003.

[52] E. Rosen and Y. Rekhter. BGP/MPLS VPNs, 1999.

[53] Tuomas W. Sandholm. Distributed rational decision making. pages 201–258, 1999.

[54] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1986.

[55] Neil Spring, Larry Peterson, Andy Bavier, and Vivek Pai. Using PlanetLab for Network Research: Myths, Realities, and Best Practices. *SIGOPS Oper. Syst. Rev.*, 40(1):17–24, 2006.

[56] Joe Touch and Steve Hotz. The X-Bone. In *Proceedings of the Third Global Internet Mini-Conference at GLOBECOM'98*, pages 44–52, 1998.

[57] J.S. Turner and D.E. Taylor. Diversifying the Internet. In *GLOBECOM'05*, volume 2, 2005.

[58] William Vickrey. Counterspeculation, Auctions, and Competitive Sealed Tenders. *Journal of Finance*, 16:8–37, 1961.

[59] Yi Wang, Eric Keller, Brian Biskeborn, Jacobus van der Merwe, and Jennifer Rexford. Virtual Routers on the Move: Live Router Migration as a Network-Management Primitive. In *ACM SIGCOMM*, pages 231–242, 2008.

[60] Minlan Yu, Yung Yi, Jennifer Rexford, and Mung Chiang. Rethinking Virtual Network Embedding: Substrate Support for Path Splitting and Migration. *Computer Communication Review*, 38(2):17–29, 2008.

[61] Fida-E Zaheer, Jin Xiao, and Raouf Boutaba. Multi-Provider Service Negotiation and Contracting in Network Virtualization. *To appear in the Proceedings of NOMS 2010, Osaka, Japan*, April 19-23, 2010.

[62] Ellen W. Zegura, Kenneth L. Calvert, and Samrat Bhattacharjee. How to Model an Internetwork. In *INFOCOM*, pages 594–602, 1996.

[63] Y. Zhu and M. Ammar. Algorithms for Assigning Substrate Network Resources to Virtual Network Components. In *Proceedings of the IEEE INFOCOM'06*, 2006.