Service-Driven Networking

by

Jin Xiao

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
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.

Abstract

This thesis presents our research on service-driven networking, which is a general design framework for service quality assurance and integrated network and service management in large scale multi-domain networks. The philosophy is to facilitate bi-party open participation among the users and the providers of network services in order to bring about better service customization and quality assurance, without sacrificing the autonomy and objectives of the individual entities. Three primary research topics are documented: service composition and adaptation, self-stabilization in uncoordinated environment, and service quality modeling. The work involves theoretical analysis, algorithm design, and simulations as evaluation methodology.

Acknowledgements

To dear grandparents, for watching over me from land far away, from stars far above; this is for you.

To parents, for eternal love and unwavering support, through the small hours of dawn.

To loved ones, for lessons in humanity, humility, optimism and kinsmanship.

To Prof. Raouf Boutaba, for a journey no less arduous as it is glorious.

To friends, there is always one more seat at the beer table.

"To strive, to achieve, and not to yield."

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Current IP network design has revolved around best-effort data delivery in large volumes. In essence, it is the business of transporting data through the network while ensuring the network operates correctly and the network resources are utilized efficiently according to objectives set forth by the network operators. This view is reflected in the principle of network management [1]: *the aim of network management is to analyze, control and manage network infrastructure in order to ensure its configuration correctness, robustness, performance quality and security.* Little attention has been paid to the users and the applications that owns the data, and the network at its core is operated around the concept of packets rather than end-to-end flows. As shown in the past decades, this design is highly scalable and efficient, and has contributed to the rapid expansion of the Internet to date.

In recent years, we have seen a phenomenal proliferation of new networking concepts (e.g. Overlays, Peer-to-Peer, virtual networks, etc.) and Internet-based applications (e.g. Internet media such as YouTube, IP telephony, IPTV, massive multi-player online games, social networks, etc.). Many of them demand much stronger end-to-end Quality of Service (QoS) support than what the current Internet can offer and are advocating for end-user/application involvement in the traditional networking operations. As a result, there is a shift in the networking and distributed computing world towards a strongly service-centric view of the Internet [2][3], suggesting that the Internet should evolve from a data transportation medium to a service hosting platform. As a consequence, service management in the networking context has gained increased attention [4] and automated management designs are strongly favored [5][6] to deal with the rising volume, complexity, and diversity of this service-centric networking world.

1.1 Service-driven Networking

In its generic form, a **service** is a function offered by a provider to its customer at some agreed quality. In the networking context, a service is generally associated with an end-to-end flow of packets belonging to some user/application (the "customer"). Thus an end-to-end service flow must be provided at some guaranteed quality. The principle objective of service management is then to guarantee the assured delivery of a service correctly, efficiently, and securely. I term this problem as the **assured service delivery** problem. As this thesis will show, even if we only focus on the issue of service quality guarantee, it is challenging to obtain an efficient and effective mechanism under today's Internet infrastructure. In addition, it can be argued that the stated conception of service management is myopic, in that only the concerns of individual services are taken into account. What about the efficient and correct operation of the service infrastructure as a whole? How would the service operational concerns impact the objectives of traditional network management? Indeed, this myopic conception of service is reflected by current service oriented approaches, ranging from service overlays to P2P networks. Therefore, it is important to establish that service management must ensure the service environment as a whole is properly managed, and both the service concerns and the network management objectives are consolidated. This is the **management consolidation** problem. In this thesis, the aspects of stability and optimality are the key focus. Thus, **service management** to encompass both the assured service delivery problem and the management consolidation problem. In the proceeding paragraphs, we discuss the key obstacles to overcome when addressing these problems.

On the issue of assured service delivery, today's IP networks are not designed to operate at the end-to-end flow level. More precisely, it is generally the case that the default

Internet routing does not provide quality of service assurance and very little path diversity. To its advantage, such routing mechanism is efficient, scalable, and relatively easy to manage and control. In the past, the Internet Engineering Task Force (IETF) has proposed two QoS assurance schemes, namely the Integrated Service (IntServ) [7] protocol and the Differentiated Service (DiffServ) [8] protocol. The IntServ protocol employs the Resource Reservation Setup Protocol (RSVP) to setup a QoS assured path end-to-end. It is a two-pass protocol with soft state, meaning that during the forward setup, each router along the path must keep states. This in principle violates the Internet's state-less design, and thus scalability and resilience concerns are raised: the routers need to hold many soft-states for many traffic flows, there are frequent needs to refresh soft-states, recovery from router crashes is difficult, etc. To date, the IntServ protocol is not widely accepted with no real deployments in the networks. The DiffServ protocol allows for relative service differentiations through service class labelling by the service providers. Although the scheme is efficient and does not require state keeping inside the network, it does have a number of unresolved issues. Foremost, DiffServ does not provide per-flow QoS guarantees. Furthermore, because the service class specifications are done manually by the service providers, end-to-end network path crossing multiple providers would require manual SLA setup and service class classification across provider boundaries. Despite these shortcomings, more and more DiffServ capable routers are being deployed in the networks today. Another challenge presents itself at the inter-domain level. Inter-domain routing is heavily influenced by service providers' routing policies, which are local decisions made based on business objectives and other non-technical factors. Thus inter-domain routing today does not provide any path diversity, and does not reflect the requirements of the customers. Furthermore, inter-domain route conflicts and path instabilities are largely unresolved [9][10]. In a nutshell, today's Internet infrastructure does not guarantee service quality. Consequently, we have seen application level "patches" to the problem. Efforts such as overlay networks (e.g. [11][12]) have attempted to provide some QoS assurance across the Internet. Generally, a network of overlay nodes is setup and end-to-end routing is conducted as a traversal on the overlay graph such that user/application specific QoS requirements are satisfied. Work such as Service Overlay Networks (SON) [13] provide algorithms and protocols for dynamic QoS-aware service path computation. Although overlay networks provide an effective method of supporting per-flow routing and QoS assurance on the Internet, it is not a long-term solution because 1) application level routing does not respect underlay network traffic engineering and management practices and thus interferes with service provider's network planning and traffic engineering operations and causes network instabilities [14]; 2) overlay routes must use application level relay nodes, which are typically located at the edge of the network and hence the resulting end-to-end paths are likely to be "bounced" around a number of edge relays before reaching the destination. This is contrary to the expected edge-core-edge network path traversal, and thus inefficiency arises.

The issue of management consolidation requires careful analysis and design. As it stands today, end users do not have control over their service flows, and their concerns are not reflected in the traditional network management objectives. One cannot simply circumvent the issue at the application level oblivious of the traditional networking concerns. Overlay networking is a good case study where numerous investigations have shown the adverse effect it has on the performance and stability of the network layer [14][15]. Foremost is the issue of how to achieve management goals in a service-oriented environment, where the behaviors of the services are selfish and uncoordinated. This is in direct contrast to the traditional network management approach where the network resources and their management are tightly controlled and centrally planned by their respective network operators. Secondly, consolidating the management objective of the services and the networks is not trivial as the two substrates are fundamentally different and sometimes conflicting in their objectives: satisfying the service requirements of the customer vs. achieving optimal network performance.

Clearly, a rethinking in management design is required wherein the goals of service management, assured service delivery and management consolidation, can be achieved. I term this shift in paradigm as "**service-driven networking**". Our earlier work on end-to-end service provisioning with QoS-assurance [16] have postulated one such possible design. Some recent works also expressed similar service-driven view on the future of the Internet [2][17][18]. The analysis and solutions presented in this thesis are specifically designed to realize the service-driven networking paradigm.

1.2 Research Objectives

This thesis work is grounded on the following observations: 1) service management objectives such as service quality assurance require dedicated service management functions, at the same time the objectives of the network management must be satisfied. 2) Network service as an end-to-end construct requires open bi-party participation of the application and the network. It cannot be sustained at a single layer. 3) Increasing network and service complexity, scale, dynamicity, and mobility favours distributed self-managing solutions. To this end, our work is focused on providing quality assured end-to-end service delivery in multi-domain large scale networks, and to facilitate consolidated service and network management in this distributed setting. More specifically, three related topics are investigated:

Service composition with QoS assurance: We consider a large scale network consisting of independently operated network domains. Service composition addresses the issue of how does a customer obtain an end-to-end QoS assured service path. Furthermore, service adaptation mechanism enables the service path to self-configure and self-optimize as to alleviate runtime QoS degradation while minimizing the cost of change. Algorithmic solutions are sought for the composition and adaptation problem, and comparative study and analysis are conducted to evaluate the performance of the solutions under various network settings. The details of this work are presented in Chapter 3.

Self-stabilization in uncoordinated environment: In an environment of uncoordinated and distributed control, the service functions are implemented by individual service/network providers with selfish objectives. It is questionable whether such an environment can be stable and at what loss of system optimality? Furthermore, if there exist stable states, can the environment always gravitate towards such a stable state and retains it (i.e. *self-stabilization*) from an arbitrary state? We seek theoretical answers to these questions in the context of overlay (*service substrate*) and underlay (*network substrate*) interactions. This contemporary problem setting allows us to leverage and compare our findings to existing results and conduct concrete analysis. As detailed in Chapter 4, the results shed interesting insight on how service-driven networking could be realized in practice and suggest a set of operation and design guidelines that should be followed by the network operators and the service providers.

The meaning and implication of service quality: service quality is an important concept in service-driven networking. Contrary to traditional network operations wherein QoS performance metrics are well established and understood, few works have been done on the computation and correlation of service quality metrics and their relation with network performance and business revenue in the network service context. Based on the classic models of service quality and customer satisfaction from economics and marketing science, a comprehensive modeling framework is developed for analyzing the impact of network performance and service quality on customer satisfaction, and consequently on the provider's business revenue. To date, many of these aforementioned aspects are obtained by empirical means that are subjective and difficult to capture mathematically. Evidence from marketing science show that for services with clearly defined performance metrics and evaluation methods, which majority of the network services are, there is a strong and objective linkage. The details of this work are presented in Chapter 5.

1.3 Thesis Contribution

First, a service composition and adaptation algorithm is developed for QoS assurance of end-to-end service paths. Through graph abstraction, this problem can be reduced to the classic k-MCOP (k multi-constrained optimal path) problem. In analyzing existing k-MCOP solutions, we discover their inefficiencies when applied to the service-driven networking context and derive a new heuristic accordingly. The proposed solution enables the self-configuration, self-optimization and self-adaptation of end-to-end services and can provide hard QoS guarantees over independent network domains with relative QoS differentiations. Through in-depth simulations, the performance of our solution is compared with classic k-MCOP solutions and its effectiveness is demonstrated. Our solution supports open bi-party collaboration between the network providers and the customers while maintaining intra-domain management integrity. This design approach is gaining popularity in the network research community recently.

Second, we analyze the interactions of uncoordinated and selfish overlays and underlays in distributed environment. We examine whether there exist practical conditions under which the interactions of independent service overlays and network underlays can result in stable system state? To this end, we first motivate and show a new way to model the overlay interactions as a congestion avoidance game with pure Nash equilibrium and bound its convergence and price of anarchy under a novel desirability concept. A number of practical implementation issues are also addressed. Then we explore how the interactions among underlays and overlays can be modeled as a long term game where a stable and unique equilibrium is reachable. A number of preconditions must be satisfied to achieve this system stability. Simulation results confirm with the theoretical findings.

Third, we examine the crucial linkage between network performance, customer satisfaction and profitability of network service, and present an analytical modeling approach from market science perspective. A generalized forecasting model is developed to project service profitability from the underlying network service infrastructure and the subscriber population. Through simulation studies and analysis, we demonstrate the effectiveness of this approach in capturing key factors and trends that influence service profitability, and in improving current network planning and upgrade processes. Little work has been carried out to address this aspect of service management in the networking community, and none as encompassing in scope.

1.4 Thesis Outline

The remainder of this thesis is organized as follows: Chapter 2 presents general background on network and service management, and discuss our perspectives on the current state of service management. Chapter 3 presents the result of the investigation on the QoS-assured path composition and adaptation problems, and the resulting heuristic design; Chapter 4 documents the analysis of the overlay-underlay interaction, the lessons learned, and discuss a number of implementation issues in practice; Chapter 5 presents the service quality modeling framework from a market science perspective; and Chapter 6 concludes this thesis and outlines additional problems of interest for future research.

Chapter 2

Background and Perspectives

In this chapter, we present the background literature on network and service management, and discuss their influence on service-driven networking. We first introduce the general concepts of network and service management, and briefly describe a number of prominent service and business management frameworks. In particular, we examine the impact of web service and Service Oriented Architecture (SOA) on service design and management. The principles of autonomic networks are then presented as well as existing approaches to autonomic system design. Finally we discuss the challenges of service management that are the rationale and motivator of our research.

2.1 Principles of Network and Service Management

The traditional service offerings in telecommunication are few and primarily oriented around flow-based voice communication, which tended to be defined and uniform. In contrast, the best-effort delivery paradigm of the Internet eschews flow-based design in favour of simplicity and scalability. Thus, when examining the principles of network and service management, we track their roots back to traditional telecommunication management.

The network management concept originated in the late 1980s, the International Telegraph and Telephone Consultative Committee (CCITT) - now the International Telecommunication Union-Telecommunication Standardisation Sector (ITU-T) and the International Organization for Standardization (ISO) jointly published a set of network and system management standards under the X.700 series [19][20][21]. This has since been known as the Open Systems Interconnection (OSI) Reference Model for network and service management. A number of concepts introduced in the OSI Reference Model are fundamental to the development and structuring of network and service management to date. It is also where a definition of network management is given: a network management system is an application. Its aim is to analyze, control and manage network and service infrastructure in order to ensure its configuration correctness, robustness, performance quality and security.

Two types of entities are defined in the management scope: the managed objects and the manager. The managed objects are the network resources and end systems being managed, while the manager is responsible for all of the management activities. The managed objects generally have an information view pertaining to their management information and management capacities. For example, a router could have processing speed, maximum buffer size, and number of input/output (I/O) interfaces as part of its management information, as well as the set of configurable parameters as part of its management capacity. This separation of information and capacities gives rise to the dual view of a managed object in terms of what is functionally present and what is manageable. Considering the diverse set of resources and devices present in networks and systems today, it is immediately apparent that some sort of common information model is necessary to present such information in a consistent manner. This is well understood even outside the telecommunication sector, as evident in the prominent Management Information Base (MIB) [22] standardized through the Internet Engineering Task Force (IETF), whose principle interest lies in the operation and management of the Internet.

From a functional perspective, the OSI reference model defines five functional areas of management: configuration, fault, performance, accounting, and security (FCAPS).

- *Configuration management* is concerned with resource configuration such as network path setup, resource provisioning, device configuration, etc. In telecommunication networks, configuration may also include user terminal configuration, user profiling and service personalization.

- *Fault management* deals with fault detection, identification, isolation, recovery, path protection, etc. The immediate cause and effect relationship between fault and service disruption makes fault management a highly regarded area of research. With the increasing distributed-ness of systems and resources, as well as the growing size and complexity of the networks, it is extremely difficult to identify network faults and be able to address them efficiently.

- *Performance management* is concerned with the quality of service delivery of network services, the monitoring of traffic, traffic control techniques, and resource management. With the introduction of telecommunication services in IP networks and the desire to support multimedia services spanning across both wireless and wired networks, the need for effective performance management has increased significantly.

- *Accounting management* focuses on the charging and accounting of user traffics. This area of management has not been a major focus of network management research. However, works on pricing and charging in networks directly impact the way accounting management could be done. Furthermore, as more and more application service providers and usage-based premium services (e.g. real-time broadcasting) are being made available as part of the telecommunication service offering, effective accounting management becomes a necessity.

- *Security management* is about aspects such as integrity of traffic, authentication of parties, authorization, security auditing, access control, etc. Although security management is an important area of network management, it has received less attention compared with the other areas in the past. This can be attributed to the fact that the traditional telecommunication network was a tightly managed private network separate from the packet-switching Internet. With the convergence of communication infrastructures, and the increased severity of security attacks on networks and services, security management has become the management problem to be addressed.

In implementing the OSI Reference Model, the Telecommunication Management Network (TMN) [1] framework was perhaps the most influential and prominent architecture. It adopted a layered view of management in the same spirit as the layered network protocol stack. Each layer builds on the functions of the layer below it and is considered an abstraction of the lower layer. In TMN, four management layers are defined:

- *Element management*: is concerned with the management of individual or a collection of network components (e.g. a switch, a link, etc.). It also mediates data between the network element and the network manager. It is device and technology-specific.
- *Network management*: provides an end-to-end network view of the managed resources and devices. It is device neutral.
- *Service management*: is concerned with service orders and their translation into the establishment of a circuit. It includes contracting with customers and service providers, service order fulfillment, quality of service assurances, billing information and troubleshooting.
- *Business management*: is concerned with the area of business and human resource planning. It is focused on the services as a whole and a company's financial concerns.

Figure 2.1 presents a combined model of telecommunication network management according to the TMN and OSI models [23]. This layered management view has become prevalent in structuring management system research and development. However, as the networking infrastructures and functions continue to evolve, some problems have emerged, especially regarding the service management layer. In particular, the definition of services and their scope has changed drastically. The traditional telecommunication services consisted mainly of telephony features and their supporting business processes, such as installation, call setup, session management, customer care, billing, etc. In contrast, the types of services today are far more diverse and often include generic software applications and processes. They encompass a wide range of applications and features: text messaging, Web services, wired and wireless voice communications, video on demand, location-dependent services, etc. It is becoming increasingly difficult to maintain a uniform management view

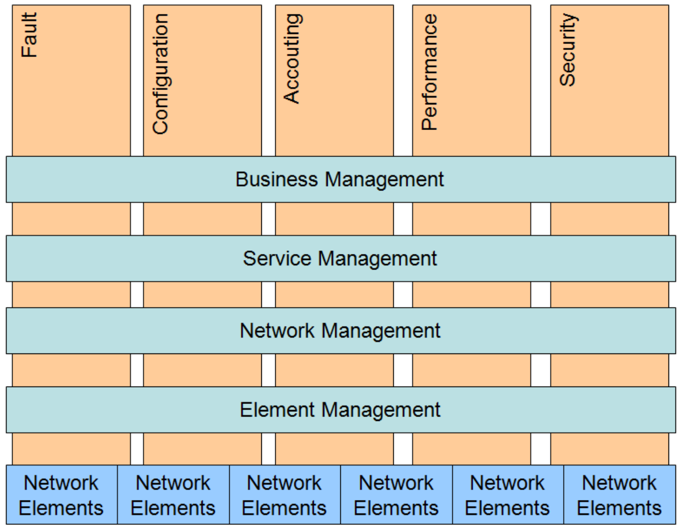


Figure 2.1: Telecommunication Network Management (TMN) Overview

of these diverse services in a single layer (e.g. the service management concerns of a telephone call are quite different from that of a web application). Furthermore, the TMN model lacks the explicit representation of the customers. With the growing service differentiation and customization, there is an increasing need for the formal representation and structuring of customer-oriented processes in the management framework.

In a generic sense, we consider that: *a service is a process or functionality (or set of) that is offered by a service provider to the customer based on a common agreement of service quality and price*. The important aspects of a service with respect to management are: the service specification, a Service Level Agreement (SLA), and a service life cycle model. The service specification is service provider-oriented; it details the function of the service, its service access point (or interface), capacities, expected input and expected output. The SLA is customer-oriented: it contains a general service description as it relates to the customer, the servicing terms, technical quality guarantees in the form of Service Level Specifications (SLSes), and monetary clauses.

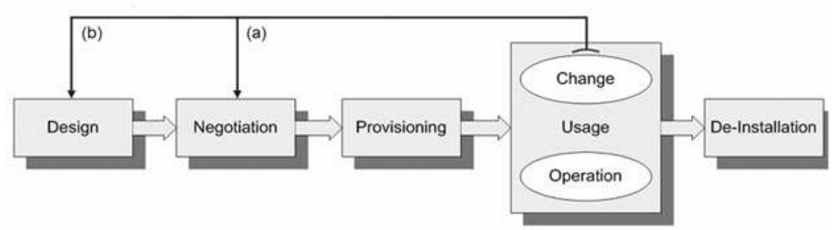


Figure 2.2: The Service Life Cycle

The service life cycle describes the various stages of a service, from design to de-installation. Compared with network components, services are much more transient and dynamic, and this is reflected in the service life cycle flow: a service could be instantiated and accessed at the request of a user and quickly torn down thereafter or the features and characteristic of a service could change during runtime due to changes in the environment or user requirements. The general service life cycle is depicted in Figure 2.2. It comprises the following phases: design, negotiation, provisioning, usage and de-installation. The design phase is concerned with the design and implementation of a service; the negotiation phase deals with the contract between the service provider and the user, including agreements on service quality; the provisioning phase deals with service configuration and resource reservation. The usage phase comprises operation and change. Change covers modifications to the existing service because of changes in network conditions or user requirements and may induce a renegotiation of the service contract or even the redesign of the service itself. The de-installation deals with service termination and contract conclusion. Our work focuses on the provisioning and usage phase of the service lifecycle.

2.2 Service and Business Management Frameworks

In this section, we present a number of current initiatives aimed at providing a structured and integrated view of management, encompassing network, service and business operations. Three frameworks are discussed: the Next Generation Networks (NGN), the Next Generation Operational Support Systems (NGOSS), and IT Infrastructure Library (ITIL).

Both the NGN and NGOSS originate from the telecommunication sector, while the NGN describes a management framework to enable network and service management, it covers very little non-architectural aspects of service management. The NGOSS specifies operational guidelines that covers all aspects of telecommunication service and business. The ITIL offers a good contrast to NGN and NGOSS as it is rooted in the IT service sector. Finally we summarize the impact of web service and service oriented architecture on service management.

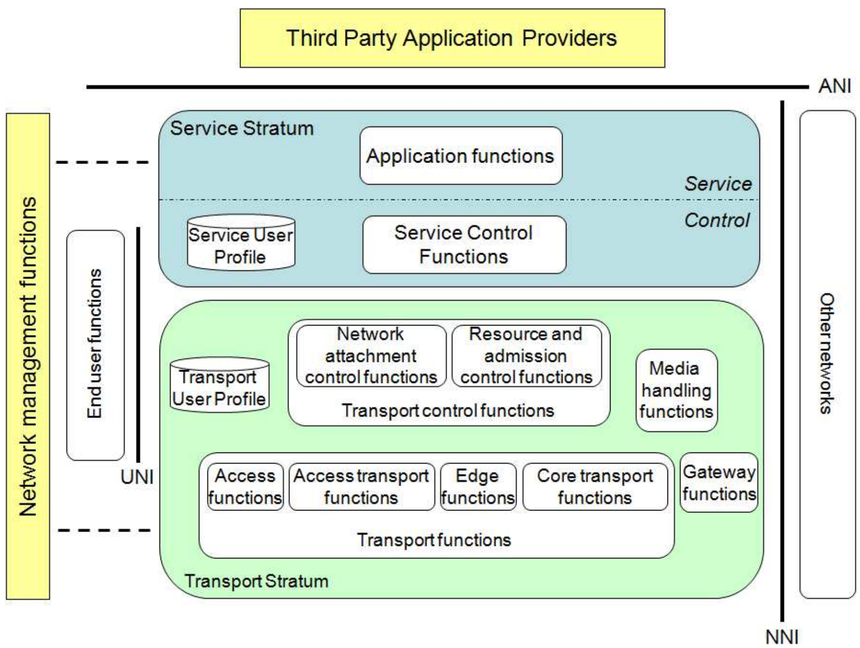


Figure 2.3: Overview of the NGN Model

2.2.1 Next generation networks (NGN)

With the convergence of Telecommunication networks and IP networks, the International Telecommunication Union Telecommunication-Telecommunication Standardization Sector (ITU-T) has already begun the process of creating a new standard network and service model for Next Generation Networks (NGN) [24][25]. The primary aim of this initiative is to create a manageable packet-based network capable of providing telecommunication services and using QoS-enabled transport technologies. The NGN model has a strong focus on services. To address the issues of mobility and network heterogeneity, a technology neutral transport layer is proposed that also support generalized mobility. Figure 2.3 illustrates the NGN model. The NGN network model consists of a transport stratum and a service stratum. The transport stratum is IP-based, with additional access functions to inter-work with different access technologies and transport QoS controls. The service stratum supports both session-based and non-session based services, functions such as session control, subscribe/notify, instant message exchange, presence information, user profiling, application plug-in APIs are included. The service stratum also provides public switched telephone network capabilities and inter-working through PSTN (Public Switched Telephone Network)/ISDN (Integrated Services Digital Network) emulation. Management functions are distributed within both stratums, but the specifics of the management components and models are yet to be defined. User control of NGN networks is supported both at the transport and the service stratums through User-to-Network Interfaces (UNI); Application interaction with the service stratum is provided through Application-to-Network Interfaces (ANI); Inter-working with other networks is supported through Network-to-Network Interfaces (NNI). Compared with the traditional TCP/IP layered model, the addition of service stratum allows for much stronger service-oriented control and management from the provider side. The type of services NGN network model is aimed at supporting are session-based services such as IP telephony and video conferences, non-session-based services such as video streaming, and public switched telephone network functions (through software emulation). Additional considerations are made for multimedia support within the network through IP Multimedia Subsystem (IMS) consisting of core network functional entities and interfaces that can be used by network service providers to offer SIP-based services to subscribers. Much of the call session intelligence of the telecommunication services is to be supported through IMS. However, services are not explicitly modeled and represented in NGN and thus there is no structured support for the various stages of the service life cycle.

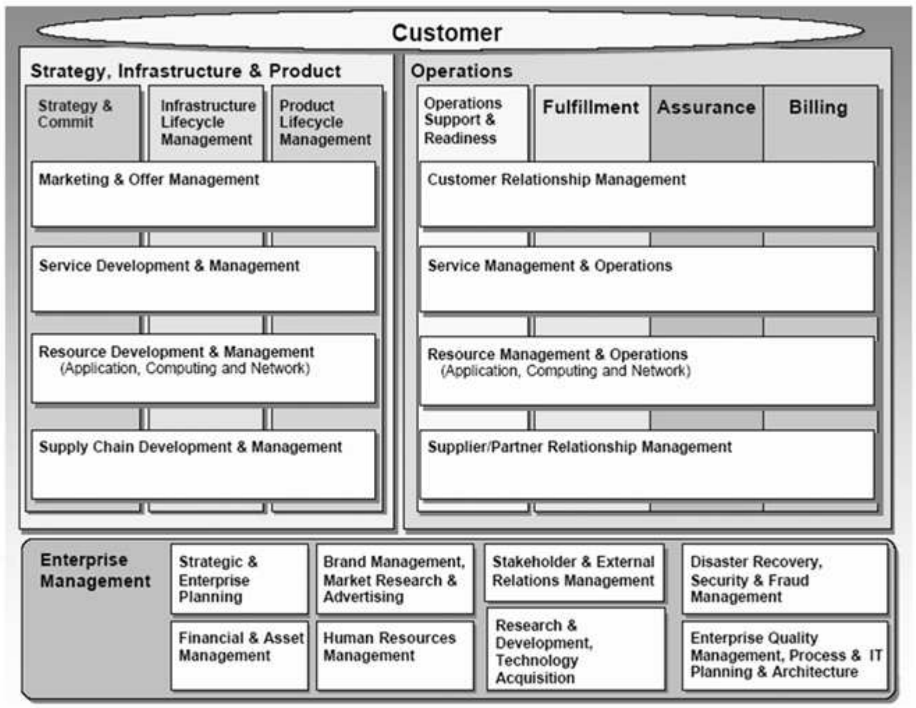


Figure 2.4: Enhanced Telecom Operations Map (eTOM) top level process map

2.2.2 Next generation operation support systems (NGOSS)

The Next Generation Operation Support Systems (NGOSS) [26] is an ongoing effort in the Telecommunication Management Forum (TMF) to provide a new framework for the design and development of telecommunication business and service management systems. Compared with the traditional OSS systems of telecommunication, which are largely founded on the TMN model, the NGOSS system places strong emphasis on the integration of business and service processes in telecommunication management systems. The NGOSS effort is composed of four interrelated activities: design of technology neutral platform architecture, business process model specification, definition of a shared data model and creation of compliance programs. Technology Neutral Architecture (TNA) is a new NGOSS service platform that integrates traditional "OSS silos" (i.e. independent OSS systems for different parts of the telecommunication network) under a single distributed architecture. The

Enhanced Telecom Operations Map (eTOM) is the new NGOSS business process model that seeks to replace the old TMN layered management model. The map defines multiple levels of processes with each below level further refines and realizes the level above. Figure 2.4 shows the top process level which gives a general view of the NGOSS organization and management areas. It includes fulfillment, assurance, billing and operations support and readiness as the top level processes and each of which is broken down into level two sub processes. The lower levels consist of activities in support of each sub-process. Four functional areas are defined spanning across these processes. A Shared Information and Data model (SID) captures the business and system entity definitions, relations and management abstractions. It is defined with the Unified Modeling Language (UML), a standard language for architectural specification in software development. A Compliance Program consists of a NGOSS system testing strategy and a suite of test cases for validating NGOSS systems. An implementation of NGOSS is being developed under the TM Forum program OSS/J: the OSS through Java initiative. The goal is to deliver standards-based NGOSS interfaces design and implementation guidelines for the development of component-based OSS.

In comparison to NGN, NGOSS has a much stronger focus on the service and business processes in telecommunication, including many non-technical aspects such as customer relation management, marketing, etc. In some sense, NGN and NGOSS are complimentary in that both are telecommunication service- and business-oriented: NGN focuses primarily on the integration of the IP protocol in the telecommunication infrastructure, operations and service offers while the scope of NGOSS is the service and business support at the operational level. Both NGN and NGOSS do not consider the management of service life cycles. This is because telecommunication services tend to be manually designed, their deployment is pre-planned, and the services have long lifetime, which made automated service life cycle management less crucial than in the IT sector. However, with the convergence of IT and Telecommunications, NGOSS has introduced Telecom Application Map (TAM) [27] to specifically address this issue.

2.2.3 IT infrastructure library (ITIL)

The IT Infrastructure Library (ITIL) [28] is a collection of international best practices for IT service management. Its primary purpose is to provide service support and service delivery guidelines for IT business. Service support addresses issues that arise in service operations and maintenance, such as the role of a service helpdesk in effective incident management and in managing customer problem reports. Service delivery relates to service provisioning and maintenance to meet business requirements. Important service factors being considered are: Change Management for impact assessment and scheduling of service change; Release Management for coordination, control and physical introduction of different service versions such as incremental application upgrades; Configuration Management for ensuring service component information, relationships, and documentations are comprehensive and accurate; Availability Management to ascertain component reliability and recovery procedures; Capacity Management for efficient resource and infrastructure usage; and Financial Management for effective business budgeting and accounting. It is apparent that although ITIL is deeply rooted in the IT industry and is largely software service oriented, it shares much of the same management concerns as the TMF management areas, such as configuration, fault and performance. In 2005, ITIL v3 formally introduced the ITIL service life cycle guidelines. It includes explicit support for three service life cycle stages: service design, service transition and service operation. Service design deals with the creation or change of services and service management processes. Service transition deals with the validation of services utility (a measure of a service's performance, similar to what QoS is to networks except that it is much broader and more general) and to guide their transiting into service operation. Service operation deals with providing the services effectively and efficiently. This roughly corresponds to the general service life cycle stages depicted in Figure 2.2. Service transition encompasses service negotiation and provisioning. Continual service improvement maps to the service change stage, and deals with service adjustment to satisfy changing business needs. Furthermore, an overarching service strategy provides the general service guidelines and how a service should be designed, configured and operated. Compared to NGOSS and NGN, ITIL views services as a set of software components that are dynamically designed and deployed then evolve to suit the changing business needs of IT industry.

2.2.4 Relating to service-driven networking

As evident from the description of NGN, NGOSS and ITIL, the existing service management initiatives are "specification heavy". They establish the essential components, structure, processes, and interfaces for service and business management at large, but do not provide concrete enabling mechanisms or address the issues of management consolidation. In contrast, this thesis work focuses on specific enabling mechanisms for service quality assurance and analyze key system properties such as stability and optimality.

Furthermore, The current initiatives are realized from a operator-centric perspective, meaning that the management framework is focused on the specific concerns and operations of a single network or service operator, while service-driven networking proposes a distributed multi-service environment where multiple operators co-exist. In this context, works on Web service management offers some unique perspectives. Some of these aspects are discussed in the proceeding subsection.

2.2.5 Web service management and the influence of SOA

The increasing presence of web services as the leading technology for distributed applications and services has given rise to a new service design paradigm the Service-Oriented Architecture (SOA). This approach is founded on loosely coupled web service components with clear defined interfaces, functionalities, and access behavior. This is similar to the idea of having standard interfaces in telecommunication to interconnect equipment from different vendors. SOA is a computing model that strives for the creation of software or applications based on a set of independent service components that are loosely coupled. The only obligation among these components is to provide a set of promised service features through defined service interfaces (this is a strict definition, as the service interfaces could potentially be discovered dynamically as well). In this sense, web service composition (the interconnection of web services to form a single application) could be viewed as an example of the service-oriented architecture model.

The concept of a web service is underpinned by a set of XML (Extensible Markup Language)-based technologies. It allows for data representation, access and processing. Standardized interface definition and access is supported through the Web Service Definition Language (WSDL) and a Uniform Resource Identifier. Customizable information transport and object invocation/access is performed through the Simple Object Access Protocol (SOAP). Open service specification and discovery is standardized through Universal Definition, Discovery and Integration (UDDI). Questions are raised on how to utilize web service as a management technology for networks and services and how to manage Web services effectively? In 2004, OASIS (Organization for the Advancement of Structured Information Standards) released a set of standards for Web Services Distributed Management (WSDM) [29]. Some of the WSDM design goals include unified management through web services irrespective of how manageable resources are organized and composed and what information is modeled by the resource, and whether a web service could be composed with others and under what preconditions.

WSDM consists of two standard sets, Management Using Web Services (MUWS) [30][31] and Management of Web Services (MOWS) [32] that address the previous two questions respectively. In this section, we first present the framework of WSDM and its two sub-topics. Then we introduce the web service orchestration and choreography tools that are essential to facilitate the automation of web service composition, components configuration and runtime behavior.

2.2.6 Web service distributed management (WSDM)

Web Services Distributed Management (WSDM) defines a framework to access and receive management information from manageable resources with Web services. Using a standardized set of XML (Extensible Markup Language) specifications and Web service technology, the WSDM framework is an attempt to bridge resource heterogeneity and to integrate existing management products. In essence, the WSDM framework proposes to wrap all manageable resources and management software with Web service-based interfaces. In the WSDM context, the term manageable resource covers network resources, end user devices, software components, and Web services. These manageable resources can be accessed through Web service endpoint references (EPRs). In addition, manageable resources must support one or more standardized manageability capabilities, such as identity, description, state, configuration, advertisement etc. MUWS defines the representation of and the access to the manageability interfaces of Web services. A manageability interface is a Web service based management interface to a Web service. MUWS provides a set of basic and interoperable manageability specifications for management system design. It defines manageability capabilities as a composeable set of properties, operations, events, metadata, etc., that supports management functions. A set of standard manageable resource definitions are described in order to achieve integration between WSDM management systems and non-Web service managed resources e.g., SNMP agents. Some capabilities defined in the base model represent generic management information such as resource metrics, configuration parameters, characteristics of the manageability interfaces, operational status, properties, etc. Furthermore, relationships among resources and management components are described with a web service data representation such that it is possible to query about these relationships at runtime. MUWS also provides some support for manageable resource discovery and advertisement through advertisement, relationships, and registration events. Management messages and events are defined through the WSDM Event Format, organized as event reporter, event source and situation data. The advertisement event allows a newly created resource to advertise its manageability (i.e. a publish/subscribe mechanism). The relationships event allows the discovery of relationships among manageable resources. The registration event allows a manageable resource to expose its manageability via some registries (i.e. a lookup mechanism).

Management of Web Services (MOWS) is an extension and application of MUWS by specifying how Web services could be managed within the WSDM framework. More specifically, MOWS defines the following set of Manageability Capabilities for managing Web services:

- Identity: a unique MUWS capability identifier for the Web service.

- Identification: a reference for the Web service being managed.
- Metrics: a set of MOWS basic metrics such as NumberOfRequests, NumberOfFailedRequests, ServiceTime, MaxResponseTime, etc.
- Operational state: the MOWS operational status of the Web service being either UP (Busy or idle) or DOWN (stopped, crashed or saturated).
- Operational status: A MUWS equivalent of the MOWS operational states, except in more generic terms: available, unavailable, and partially available.
- Request processing state: it defines a request state diagram and provides mechanisms to define events to be sent when request processing states change.

Thus MOWS and MUWS are complimentary in WSDM, in that one specifies how management should be structured and management information modeled, while the other specifies how management functions and information could be accessed and understood. Much of the WSDM framework is focused on representing and managing software components. Although this is a fairly narrow view of service management, it occupies a much needed niche in today's network and service management spectrum.

2.2.7 Web service orchestration and choreography

A formal methodology to specify the interactions among components and runtime behavior is needed to support SOA. In traditional software engineering practices, entity-relation diagrams and sequential diagrams describe the interactions among various software blocks and participants; similarly, operational state diagrams are used to track the operational status of the software. SOA promotes component-based interactions and runtime compositions, which are far more dynamic and unpredictable than traditional distributed software design. For instance, the individual components of a Web service can be developed by different service providers, independently and at different times, and maybe even with different design requirements. Thus it is indeed a challenge to know what components can inter-work, how, and with what configurations.

Web service, as a technology that favors SOA design, can provide a set of mechanisms to address the issues discussed. In particular, there are two promising technologies: Web service orchestration and Web service choreography. Web service orchestration is concerned with the business protocols of the Web services as they interact to fulfill the business logic. It describes how these services collectively perform a business execution. For example, the actions a recipient Web service will execute in response to a service request. Thus, Web service orchestration provides a map for the interaction of the services at the message level (i.e. their order of execution, the timing of message exchanges among the Web services, etc.), and defines the ownership of the processes by the service providers (i.e. which process belongs to which service provider). Thus orchestration is concerned with the execution of a single Web service business process. In contrast, Web service choreography details the external observable exchanges that are conducted among interacting Web services. Hence choreography is more collaborative in the sense that the overall message exchange is the combined interplay of multiple Web service business processes.

The Business Process Execution Language for Web Services (BPEL4WS) [33] is a standard for describing Web service orchestration. It was proposed by Microsoft, IBM, Siebel Systems, BEA and SAP to OASIS, which adopted it as a standard way for describing control logics and Web service process coordination using WSDL. Primitives in BPEL4WS include sequential and parallel activities, conditional looping, dynamic branching, etc., commonly supported by process flow languages. Dynamic branching is a form of conditional branching in that the direction of process flow is guided by the particular runtime state of a process (i.e. the branching process). A partner is a participant in the business protocol whose role is clearly defined by a BPEL4WS role specification. The concept of "partner" is therefore essential in BPEL4WS because of its business execution-oriented view point.

Web Service Choreography Interface (WSCI) [34] is an extension of WSDL to implement Web service choreography. It was initially proposed by Sun, SAP, BEA and Intalio to the W3C (World Wide Web Consortium). WSCI specifies the stream of WSDL messages that the Web services exchange and does not describe any component behavior. WSCI uses tags to differentiate various activities. The $< action >$ tag denotes a request or response message, with each $< action >$ specifying the operation involved and the role played by the participant described with WSDL. The $< call >$ tag is used to invoke external services. The $< all >$ tag indicates that the associated messages must all be performed but in any order. Similar to BPEL4WS primitives, all of these activities could be sequential, parallel executions, or conditional loops. The dynamic branching primitive from BPEL4WS is not present in WSCI because it is dependent on the state of the component which is outside the scope of WSCI. A short comparison between Web service orchestration and choreography is given in [35] which also highlights the need for both in Web service operations.

Web service technology is a flexible enabler of service-driven networking. It allows independent service logics to be developed separately and only requires agreement on common open access interfaces. Our proposed bi-party open participation approach to service composition naturally fits in the service-oriented architecture approach.

2.3 Autonomic Networking

Autonomic networking is an promising enabler of future network and service designs. In this section, the concept of autonomic networks is briefly presented as well as the theories and approaches that have been proposed in this area.

2.3.1 The Autonomic Property

IBM research [5] has coin-phrased the term "autonomic computing" as: a distributed system where a set of software/network components that can regulate and manage themselves in areas of configuration, fault, performance and security to achieve some common user defined objectives. The word "autonomic" originates from the autonomic nerves system that acts as the primary conduit of self-regulation and control in human bodies. Four self-* properties of autonomic computing are defined:

- Self-configuration: the entities can automate system configuration following high level specifications, and can self-organize into desirable structures and/or patterns.
- Self-optimization: the entities constantly seek improvement to their performance and efficiency, and able to adapt to changing environment without direct human input.
- Self-healing: the entities can automatically detect, diagnose and recover from faults as the result of internal errors or external inconsistencies.
- Self-protection: the entities can automatically defend against malicious attacks or isolate the attacks to prevent system wide failures.

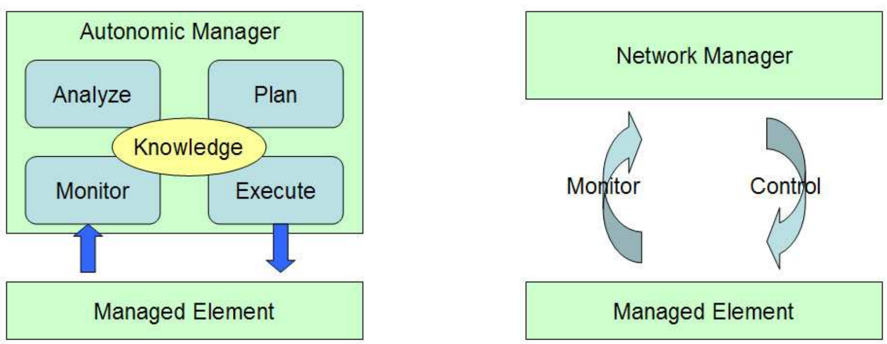


Figure 2.5: Autonomic Computing and The Network Management Loop

Although the autonomic computing concept was first proposed for distributed systems and software components, it is equally relevant to self-managing networks, especially with the increasing associations between networks and distributed applications today and the extensive use of network management applications. In general, we consider a self-managing network as a particular type of autonomic system. Hence, the term autonomic systems and self-managing networks are used interchangeably in our discussion. The left side of Figure 2.5 shows the proposed anatomy of an autonomic component whereby the autonomic manager interacts with the managed elements and its surroundings by taking inputs from the external environment, applying analysis and reasoning logic, generating corresponding actions, and executing these actions as output. This work flow fits well with the classic monitor and control loop of network management (right side of Figure 2.5), where monitored data from the networks are used to determine the appropriate management decisions and then translated into corresponding control actions. What is added is the ability for the autonomic management system to take part in the management decision making that is currently an extensive human administered task.

2.3.2 Representative approaches to self-management

Over the past years, works on self-managing networks and systems have flourished, with many proposed solutions covering virtually all aspects of networking. Here, a summary is provided on the representative approaches, focusing on principal theories and mechanisms that enable autonomic behaviors.

Design patterns

Design patterns offer an effective method of capturing expert knowledge on how to cope with specific problems from an architectural stand point. Applying appropriate design patterns in self-managing networks not only results in intelligent organization of autonomic components at runtime, but also provide a concrete guideline to component interactions. More importantly, autonomic systems developed using design patterns are guaranteed to yield desirable system output and correct system states. In a broad sense, design patterns could be general design practices, such as the patterns proposed by White et. al. [36]:

- Goal-driven self-assembly: this pattern is useful for self-configuration. System configuration decisions are made a priori and are assigned to the system components as goals. When the component joins an autonomic system, it knows how to contact a service registry to obtain resources and services based on its goal description.

- Self-regenerating cluster: two or more instances of a particular type of autonomic components are tied together in a cluster. They share the same input element and process external requests assigned by some scheduling algorithm. Instances in a cluster monitor others to ensure their proper operations.

- Market-control model: autonomic components compute the utility of candidate services or resources and make local purchase decisions for resource acquisition. A resource-arbiter element could be added to compute system-wide optimal allocation of resources among autonomic components.

Design patterns could also be tailored to solve specific problems and could be combined together to form larger patterns. For example, work of Wile [37] outlines some problem specific design patterns:

- Resource reallocation pattern: probes are associated with resource consumer to monitor resource usage and gauges are used to compute average or maximum resource usage, any violations of threshold is reported through an alarm.

- Model comparator pattern: two identical copies of an autonomic entity co-exists: an actual copy and a simulated copy. Environmental events that a component responds to are also copied to the simulator. A comparator gauge compares the output of the actual copy to the output of the simulated copy to determine any inconsistencies.

Similar to design patterns, architectural descriptions and models could be used to facilitate runtime binding of autonomic components and resources. Work on architectural prescriptions [38] captures the functionalities, constraints, resource requirements and operational states of an autonomic component as activities, roles and intents using architectural description language and state change models. This allows the resulting system to find at runtime suitable components it depends on and be able to reason about a component's ability to fulfill a particular task.

The primary problem with design patterns is their rigidity. Although design patterns are highly effective in addressing specific problems, they are only effective under the architect's envisioned context. When the environmental conditions change that render the old design obsolete, autonomic components that follow design patterns can not evolve to cope with the new environment and may even produce erroneous executions.

Policy-based management

Policy based control is an effective method of defining the behaviors of an autonomic component and can be used to drive the condition-response interactions between an autonomic component and its environment. The work of Bahati et. al. [39] proposes such a policy based self-management system that leverages the traditional policy-based management architecture. Extensive database of polices and their associated action sets are stored in a knowledge base. An event handler interprets monitored environmental variables to determine whether a particular enforced policy condition is met. The decision of applying specific polices is performed at the policy decision point, translated into executions at the policy enforcement point and carried out by the effectors. Event analyzer incorporating information processing and pattern recognition is suggested to help the policy decision point in deriving long term trends in observed environment variables. Although elaborate, such policy architecture may not be able to cope with environmental conditions that are not prescribed in its knowledge base. Other works explore various ways of dynamic policy generation. The Accord programming system [40] is designed to provide rule-based descriptions of a component's runtime behavior and the language also support rule combinations for component composition. Work of Samman and Karmouch [41] examines how high level user preferences and business goals could be translated into appropriate network level objectives. The work recognizes the disparity in quality of service representation across different layers (e.g. user, application, system, network, etc.) and seeks automated mapping of QoS attributes. The result of this mapping forms the basis for their assembly of policies and actions. A reassessment mechanism is also proposed to monitor the actual performance of the established policies through environmental feedbacks. On the subject of assessing runtime policy interactions, the work of Aib and Boutaba [42] proposed a mechanism for evaluating the optimality of policy interactions with respect to business profit. Although the work is not particularly focused on autonomic networks, the proposed solution fits well with the general autonomic framework. Work on self-optimization using policies [43] similarly uses business level policies to control autonomic system behavior. However a different approach is taken in that prescribed models are used to assist in the translation of policies to system behaviors. The work also suggests that reinforcement learning and statistical techniques could be useful in providing specialized models such as user behavior prediction and user attribute modeling.

AI techniques

As the anatomy of an autonomic component corresponds closely to the structure of an agent in multi-agent systems, research in multi-agent systems could find application in autonomic systems, particularly with respect to agent interactions and collaborations. The COUGAAR agent system is one such example [44]. COUGAAR agents are arranged into collaborative societies with common and specific goals. Communication between agents is unique in that agents do not directly communicate with each other. Rather a task is formed and agents capable of performing the task are associated with the task through an information channel. Thus, agents interact with each other for the purpose of achieving a specific task and complex composition of agents is possible depending on the complexity of the task. Lavinal et. al.'s work on multi-agent systems for automated network management [45] adapts management domain specific models to traditional agency concept and describes management specific agent interactions. In this work, not only are agents assigned roles (e.g. managed element, manager, etc.) but also all agent-to-agent interactions are typed according to their roles and task dependencies. Multi-agent systems offer a structured way of defining agent interactions and coordinating agent activities, but they do not guarantee system wide correctness or consistency as in the case of design patterns.

Reinforcement learning is effective in dealing with dynamic environment and uncertainties. The techniques do not require exact modeling of the system and they assume the environment is constantly changing and hence requires constant adaptation. These two factors make this technique quite valuable to autonomic computing and networking. However, the main drawbacks of reinforcement learning are: the learning process may be slow; large volume of consistent training data is required; and it may not be able to capture complex multi-variable dependencies in the environment. Work of Tesauro et. al. [46] remedies some of these drawbacks by seeking a hybrid approach where initial models are used to guide the learning process and in turn the models are refined overtime using reinforcement learning. Study of the hybrid approach suggests its effectiveness over pure reinforcement learning techniques. Collaborative reinforcement learning is used in a k-component model to generate self-adaptive distributed systems by Dowling and Cahill [47]. The k-component is a modeling framework for creating distributed components and specifying their interfaces. Collaborative reinforcement learning is a modification of the reinforcement learning approach by allowing the components to not only learn from environmental feedbacks but also from the experiences of neighboring components. Global optimization problems are thus tackled in a distributed manner by first having the individual component perform its local reasoning using reinforcement learning and then advertise the results to the neighbors. Based on the results, the neighbors may initiate another round of distributed optimization process. The Dynamic Control of Behavior based on Learning (DCBL) middleware [48] uses reinforcement learning for quality of service management. The middleware implements wide range of reinforcement learning algorithms.

Control theory

Control theory was first established as a general reference architecture for adjusting system behaviors to achieve desired objectives. Because adjustments are made incrementally based on environmental feedbacks, this process fits well with the execution flow of an autonomic component, and has thus been proposed as a methodology for developing autonomic components [49]. Figure 2.6 depicts the basic elements of a control system. The controller is the autonomic component and the target system is the environment. The controller affects the target system by feeding it control input. The output from the target system is then measured. Uncertainties in the environment and the monitoring devices are represented by disturbance and noise. The transducer smoothes the measured output to a form comparable to the desired input that the controller would like the measured output to match. The difference between the measurement and the objective produces a control error that serves as feedback to the controller.

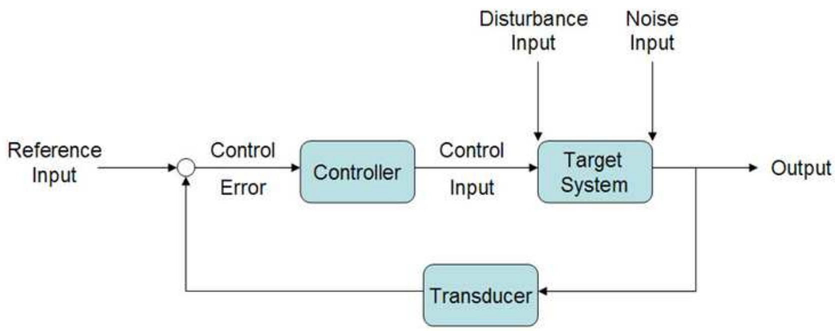


Figure 2.6: Block Diagram of the Control System

Stochastic adaptive control is a self-optimization theory from real-time control systems. Its adaptation to network self-optimization [50] also follows a similar execution flow of monitor, plan and execute. Measurements from the environment are used to build an optimization model. The model produces a series of optimization steps and only the immediate step is carried out, thus providing some degree of look ahead. The resulting changes in the environment are further monitored to refine the model. Due to the locality of the information on the environment and the possibility of multiple variables required for monitoring, the optimization model does not always yield global optimal solutions.

Although close loop controls are very well formed self-optimization mechanisms, they require clear and simple environment models (e.g. identifying the control input, understanding the dependencies among control variables, etc.) to be effective.

Economic models and game theory

Market based approaches to autonomic decision making and component interactions constitute another promising direction of research. Such a minimalist approach that allows the components to freely make their own self-maximizing decisions regardless of the overall system welfare results in significant design and control simplification. Two groups exist in the market: buyers and sellers. The sellers offer services at particular prices which may vary depending on the availability of the resources and the demand for them. The buyers purchase the services based on their specific needs at purchase time. The decision process itself is driven by the evaluation of utility functions. Price may be an input to the function or as a comparator to the utility. In the work of Wang and Li [51], such a market approach is applied to media streaming in overlay networks. The buyer maintains a list of high utility sellers of a desired service while the seller determines the amount of throughput to allocate to each buyer based on utility computation. The service prices are updated periodically based on utility maximization for the future. It is provable that game theoretical equilibrium of such system could be reached when seller/buyer has perfect information about the market and there exists some stable equilibrium points. In reality, a component may not have perfect information about the system and environment.

Game theory has been applied to many fields of networking. It has shown to be an effective method of analyzing system-wise properties (i.e. stability and optimality) of large distributed and often uncoordinated network environment. Unfortunately, the necessary simplifications that are placed on the game models often renders the resulting analysis artificial and non-applicable. Nevertheless, significant theoretical insights can be gained even from such simple models of the networks, and some of the work focus on implementing the game models in practice. An in-depth discussion on related works in game theory is presented in Chapter 4, as the prelude to the self-stabilization analysis.

Designs inspired by biological systems

Emergence from biology describes how simple local behaviors by entities without global knowledge result in global behavior. In Anthony's work [52], the emergence concept is used to design an emergent election algorithm. A key feature of emergence is that interactions are generally between the components and the environment (e.g. emitting pheromone trails), and the communications are one way and independent. Individual messages in emergent systems have low values on their own and the system behavior is non-deterministic.

These features of emergence make their designs extremely simple on the component level and render component validation tractable. Thus far, it is uncertain whether emergent designs could lead to complex and yet stable system behaviors in self-managing networks. Another related concept is stigmergy: insects coordinate their behaviors by using environmental modifications as cue. Work in collaborative construction [53] gives promising lessons in this direction, where swarms of robots are able to construct complex building structures from blocks by following simple rules and observe local environmental stimuli. The similarities between the collaborative construction and the self-organizing properties of autonomic systems suggest that indeed it is possible to achieve similar constructs, such as clustering and election, in networks using stigmergy concepts.

Biological systems are simple to implement and the resulting emergence property is highly desirable. However, it is often challenging to theoretically guarantee that emergence will occur in the system, and to deterministically bound the convergence span.

2.4 Challenges of Future Networks: a Service Management Perspective

The proliferation of new networking concepts (e.g. Overlays, Peer-to-Peer, Web services, virtual organizations, etc.) and Internet-based applications (e.g. Internet media such as YouTube, IP telephony, massive multi-player online games, social networks, etc.) has a pronounced effect on the network and service management processes, not only in terms of management scale and complexity, but also in terms of the service environment and business model. It is increasingly difficult to manage a rapidly growing set of services with centralized and manual-intensive management applications. Even at the network level, the Internet consists today of mixture of diverse networking concepts (e.g. virtual networks, Grid, etc.) and complex and programmable hardware resources, whose management is far more involved than the management of traditionally simple network resources and components. Centralized management solutions are ill-suited for the situation and it is highly unlikely that a unified set of management functionalities would emerge in the short- or mid-term future. Current trend on the design of self-managing networks and distributed management systems is a promising and fruitful avenue of development.

From the management viewpoint, the service-driven nature of future networks has significant ramifications on the design of the management infrastructures. First, the functional areas of network management (i.e. fault, configuration, performance, security and accounting) must be extended to the service layer. Second, with the focus shifting from networks to services, service assurance and customer support is an essential aspect of management. For instance, the configuration of a service must also consider its user-friendliness to the end users; the changes made at runtime must consider their impact on the user's perception of the service quality. (e.g. availability, service interruption penalties, etc.) Third, services mostly consist of software applications and processes, are more short lived and less reliable than network elements. Finally, the dynamics of services require much stronger runtime management support, as the capability or features of a service may very well change due to changes in the network environment, user requirements, etc.

Understanding service is a fundamental challenge in service-driven networking. What constitutes a service? How does service substrates interact among themselves and with the network substrates? The answer to these questions will form the fundamental assumptions and constraints that service management design is framed into. Furthermore, service quality is not as strictly defined and standardized as network QoS. It is highly application and user dependent, and thus to assure service quality, there needs to be a correct model of service quality not only from the service performance perspective but also account for the effect on user perception. For instance, quality of experience (QoE) is a term that is gaining popularity in the multimedia networks community. A measure of QoE encompasses network performance, application characteristics (e.g. video playback quality, audio jitter, etc.), and the end user's experience.

The administration and management of the Internet today is characterized by isolation and segregation. Each network domain and each service block has their own management tools and their own sets of policies and controls. However, a network service, typically end-to-end, necessitates the integration of multiple such segregates. There exists a discrepancy between the user view and network view of a service. More specifically, the user view of a service topology will be composed of a set of service components supported by clouds of interconnected physical networks and virtual networks, while the actual end-to-end physical connectivity among the service component is quite complex and hidden from the user. Network managers will have little control over the formation of the service topology, while the user will have little control over the routing and provisioning of the underlying physical resources. Thus there is a need for rethinking and re-engineering of network and service management designs. Formal analysis is required to understand the minimum degree of cooperation that must exist among the network operators and the service providers to realize service-driven networking.

Many of these aforementioned issues become the fundamental design constraints for our research. In particular, we stress on the importance of an automated service composition and delivery mechanism that understands underlay management segregation, and that is, accordingly, self-optimizing and self-adapting. We have examined the necessity of self-stabilization as a system property in such a service-driven environment and proposed theoretical bounds and implementation guidelines. Finally, we examine the meaning of service quality form multiple perspectives, integrating performance metrics with service characteristics and user perception, and establish customer satisfaction as a crucial linkage between network and business operations.

Chapter 3

Service Composition and Adaptation

3.1 Introduction

In the service-driven networking context, the goal of service quality assurance is to guarantee service quality, typically specified as a set of QoS requirements, over an end-to-end communication path from source to destination in a shared multi-domain networking infrastructure. With the increased complexity of networks and the rise in short term on-demand service offerings, the service quality assurance mechanism must be scalable and exhibits a high degree of autonomy. Furthermore, one must consider the requirements of the upper layer applications, who are the customers of a service and the characteristics of the underlay networks who are the providers of the service. From the customer side, a service path generally traverses multiple *domains*. In the context of this thesis, we use the term *domain* to mean administratively independent network domains or independent service providers. From the provider side, each domain is managed and controlled independently of each other and at a macro level (i.e. providers manage aggregate traffic flows rather than individual customers). Consequently, domains share little information with each other and generally no information about their respective networks and each domain independently conducts its own intra-domain provisioning and management according to its domain-wise objectives.

A customer's service flow is thus composed of a set of consecutive domains selected among a much larger set of interconnected domains and each of which offers different sets of QoS guarantees at varied cost. For example, the IETF DiffServ approach [8] uses class markings to facilitate differentiated services in a domain, and each domain is entitled to its own class specifications and associated QoS levels. An important question to answer when establishing an end-to-end network service is: what domains to involve and what service class to choose in each involved domain? We term this the *service composition* problem. From the problem specification point of view, its formulation is identical to traditional QoS routing. However, as we will show in this chapter, the particular characteristics of the input graph makes the problem unique and challenging. At the provider side, the dynamicity in network operations induces rapid changes in QoS conditions of domains. This issue is particularly pronounced for wireless networks. As a result, domains that promise specific QoS levels at configuration time may fail to honor these requirements at runtime. Therefore, a customer's service flow must be monitored and adjusted dynamically in order to ensure the required end-to-end QoS level is maintained. In the presence of mobility, as communicating applications/users roam across domains, path reconstruction may also become necessary. We term this the *service adaptation* problem. Together, the solutions to these two problems form the foundation of service quality assurance. More specifically, the composition should result in a feasible end-to-end QoS-assured path across multiple domains, and the adaptation must be performed to ensure the customer's requirements are upheld when network performance fluctuates.

In this chapter, we describe the formulation and analysis of the service composition and adaptation problems and present the design and evaluation of a practical solution. First, a generic service quality assurance framework is proposed based on the autonomic communication principle, covering a number of essential functions: domain discovery, domain reacheability, composition, cross-domain contracting, intra-domain provisioning, domain-wide monitoring and adaptation. We show that to achieve self-management, efficient methods for service composition and adaptation are required at the inter-domain level. Through service graph abstraction, the domain composition and adaptation problem can be reduced to the classic k-MCOP (Multi-Constrained Optimal Path) problem. However, in the context of service provisioning, existing k-MCOP solutions are inadequate and inefficient. Following this analysis, new heuristics are developed for service composition and adaptation. With high probability, our solution finds a feasible end-to-end service path composition with appropriate service class selection in each domain such that the customer's overall QoS requirements are satisfied. The solution also minimizes path establishment cost. As the network condition changes over time or as the user roams across domains, our adaptation heuristic ensures the QoS requirements of the communication path is respected as long as it is feasible to do so, while minimizing the cost of change. As we address the service provisioning problem at the domain level, the proposed solution can function over heterogeneous intra-domain provisioning mechanisms, and more importantly, provide hard end-to-end QoS guarantees over "soft" intra-domain QoS schemes (i.e. offered by service differentiation approaches). Through in-depth simulation studies with real network topologies, we compare the performance of our solution with well-known k-MCOP solutions and demonstrate the effectiveness of the approach.

The remainder of the chapter is organized as follows. Section 3.2 presents a sketch of the service provisioning framework and related works. In Section 3.3, we relate the composition and adaptation problems to k-MCOP through service graph abstraction. We analyze existing k-MCOP solutions and the unique characteristics of the service graph, illustrating why the service composition and adaptation problem cannot be solved by straightforward application of existing k-MCOP solutions. Section 3.4 presents our heuristics and their application to the composition and adaptation problems. Through simulation study in Section 3.5, we show the performance and effectiveness of the proposed solution under various graph characteristics.

3.2 Service Quality Assurance in Multi-domain Networks

First, we briefly discuss existing works on inter-domain service composition and provisioning. Raman et. al. [54] presented a general framework for service composition across multiple providers. They stress on the importance of inter-domain cooperation among service providers and the need for performance aware service composition. In their reference architecture, they envision the service composition to encompass both the application and the connectivity planes wherein "end-to-end network with desirable properties" should be constructed at the connectivity plane. Zeng et. al. [55] proposed a QoS-aware service composition scheme for web services. They are concerned with finding the optimal service execution plan among the set of candidate service components, while taking into account the QoS characteristics of these components. The QoS attributes investigated in this work are software quality oriented (e.g. execution duration, reputation, success rate, etc.). They pose the composition problem as a graph search problem, where the graph represents the execution states of the service components. An integer programming technique is proposed to solve this problem. Their simulation results show that the proposed solution do not scale well beyond 60 states. Furthermore, no QoS adaptation scheme is proposed at the software level. Similarly, Gu et. al. [13] proposed a QoS-assured service composition mechanism for Service Overlay Networks (SON). They attempt to find a feasible service component flow via a linear multi-constraint mapping function. However, the search heuristic does not perform well (Section 3.5) and only attempts to find a feasible path. They also proposed a simple localized recovery scheme to cope with QoS violations.

In the network service context, there lacks a guiding framework for service quality assurance over multi-domain networks. We present our view of an autonomic service quality assurance framework in this section and discuss its key functions.

Figure 3.1 illustrates the framework. We abstract the functional entities into two layers: the inter-domain service substrate (*InterDom*) and the intra-domain network substrate (*IntraDom*). The IntraDom encapsulates the functioning and management of a network domain, where we assume the ISP conducts its management operations and offers a set of essential functions to the service substrates. The IntraDom retains full control over its management operations. A service flow cannot interfere with the networking functions of a domain. However, the IntraDom does not have the necessary end-to-end flow information. Thus the network substrate exhibits "per-domain" behavior. The InterDom represents end-to-end flow management functions at the service substrate. It ensures end-to-end service quality is upheld for individual services. The InterDom is where the service composition

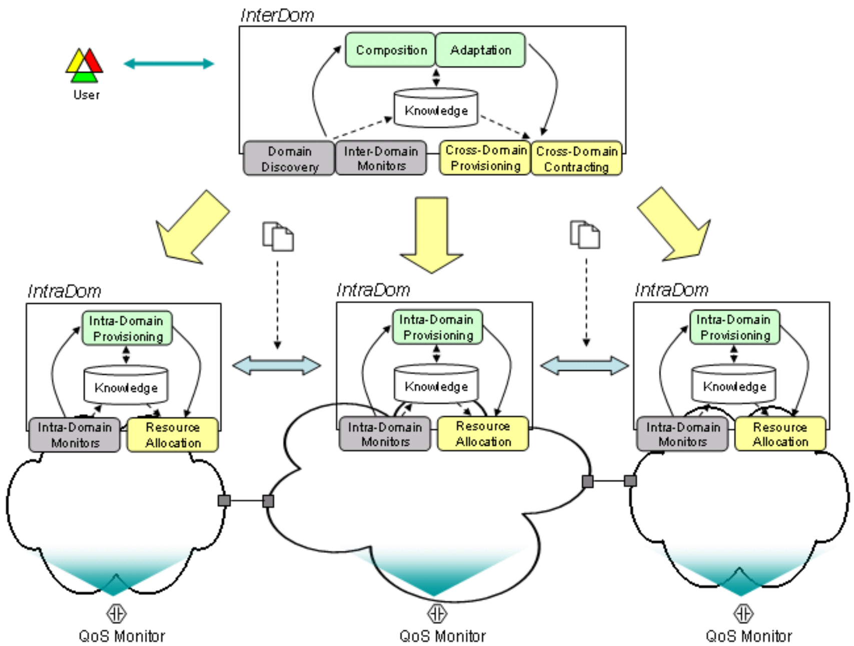


Figure 3.1: Autonomic Service Quality Assurance

and adaptation mechanisms reside. The said mechanisms operate on an end-to-end basis under the constraint of per-domain behavior.

Three general functions are prescribed for an autonomic component [5]: sensor, analyzer/planner, and actuator. The sensor function of the InterDom consists of two part: domain discovery and inter-domain monitoring. The domain discovery function is able to discover domain connectivities and QoS service class descriptions for domains. Such discovery could be facilitated either through large-scale discovery systems, such as Secure Service Discovery Service [56] and works on Web Service discovery [57][58], or through facilities in the existing network infrastructure, such as BGP [59]. Information on domain administrative policies and service class descriptions are gathered. The inter-domain monitor function is tasked with monitoring the cross domain QoS condition for the service classes a domain provides. This function can be facilitated via a set of distributed QoS monitors installed at domain borders, measuring the aggregate QoS condition of each service class in the domain. As InterDom works at the domain-level, for better scalability, only domain-wide QoS measurements are necessary. The monitors of InterDom should not rely on the monitoring functions offered by the IntraDom for the following reason: monitoring is an essential part of QoS enforcement, a domain could misrepresent its own QoS conditions for obvious economical reasons. Hence domain-level QoS monitoring should be conducted independent of the domain administrations. Furthermore, the sensor functions can be implemented as a shared service for all the customers.

The actuator function of InterDom is composed of the cross-domain contracting function and the cross-domain provisioning function. The cross-domain contracting function is responsible for establishing the required contracts with each domain for specific QoS classes and setup cross-domain traffic exchange with neighboring domains. Recent research works on service composition [13][54][55] and work flow languages, such as Web Service Flow Language (WSFL) [60] and Business Process Execution Language for Web Service (BPEL4WS) [61] could be leveraged to accomplish this task, especially when combined with works done on contract-based cross-domain management [62][63][64]. The inter-domain provisioning function interfaces with IntraDom to obtain a QoS-assured path segment across the domain at specific border points (as specified by the cross-domain contracting function), or from source/destination to the border point. Such domain provisioning independence grants each domain the autonomy in conducting its own resource management, QoS-based admission control, and pricing strategies. In practice, we expect most of these domains to employ DiffServ. Then a bandwidth broker like architecture can be applied to realize both the intra-domain provisioning and cross-domain negotiation [65].

Service composition and adaptation mechanisms are required for the InterDom. The composition function must utilize the domain connectivity and service class information to determine the best suited domain path from source to destination and their respective service classes. The adaptation function must react to changing QoS conditions along the path by modifying the existing composition to ensure end-to-end QoS requirements are satisfied. We envision an InterDom is dynamically created specifically for each communication path, utilizing the same underlying functions (e.g. IntraDoms, monitors, etc.). The management composition as determined by InterDom is dynamically adjusted whenever path adjustment takes place. More precisely, when a domain is selected/deselected from the communication path, the associated autonomic components of the domain are connected/disconnected from the InterDom.

Both the network providers (i.e. underlay) and the customers (i.e. the applications) have the freedom of conducting their own management and control actions in an open bi-party collaboration. More specifically, the customers have the ability to choose what domains to involve in its service path and what service classes to use, while the providers have the ability to determine the intra-domain routing and resource provisioning policies, as well as their inter-domain peering policies. Thus, efficient and effective service composition and adaptation logics are required to drive the autonomic behavior of the communication paths. In this scope, we deal with domain level information: domain connectivity, service classes and domain-wide QoS conditions.

Recently a number of works on multi-domain routing from a service perspective have expressed similar design principle as the framework we have presented. More specifically, Routing as a Service [17] proposes that each domain can establish itself as a service and thus facilitate end-to-end network path selection through user based service composition; a New Inter-domain Routing Architecture (NIRA) [18] allows for limited path selection (at the network domain level) by users and propose a network-level addressing and routing mechanism to realize such path selection; service oriented Internet (SOI) [2] argues for the need to establish a service layer for the Internet that is more structured than current application-level networking (e.g. Overlays) and supports virtual organizations, addressing and routing.

3.3 Service Graphs and Problem Analysis

In this section, we detail the creation of a service graph, which is an abstract representation of the domain connectivity and their service classes. Each service class is considered to have a set of QoS assurance and an associated cost. We focus on three common QoS factors in this chapter: delay, availability, and bandwidth. Jitter and other second order performance properties are not considered here. In doing so, the service composition and adaptation problems can be reduced to k-MCOP (k Multi-Constraint Optimal Path problem), which is known to be NP-Complete [66]. We then analyze the prominent solutions to the k-MCOP problem, and show their inadequacies in the service context. The result of this analysis motivates the development of a new service composition and adaptation solutions for service quality assurance.

3.3.1 Service Graph Creation

A domain exchanges traffic flows with its neighboring domains via border gateways. Each domain has a number of service classes, each with a set of QoS assurances and a price. For domains without QoS differentiations, they are assumed to provide a single set of QoS assurances and price. Figure 3.2 depicts a typical example of domain connectivity between two service components: S and D.

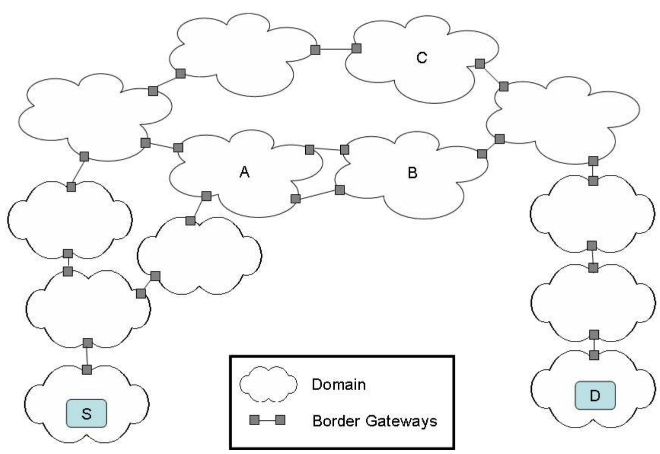


Figure 3.2: An Example of Domain Connectivity

We can abstract the domain connectivity information as an undirected graph, where the nodes of the graph represent the border gateway exchanges between neighboring domains and the edges of the graph represent the connectivity between border gateways in a domain. Figure 3.3 illustrates this transformation process. Inter-domain routing policies can be incorporated during this process. For example, domain A routes all traffics transiting from its south neighbor to its east neighbor through border gateway β. This policy is reflected in the graph (Figure 3.3). With such an abstraction, it is natural to represent the QoS assurance set and its associated price as a set of weights on each edge. For example, if domain C offers a QoS class with minimum bandwidth BW_C, minimum availability A_C, maximum delay D_C, and price C_C, then the edge connecting domain C's border gateways is assigned the weight set $\{D_C, C_C, A_C, BW_C\}$.

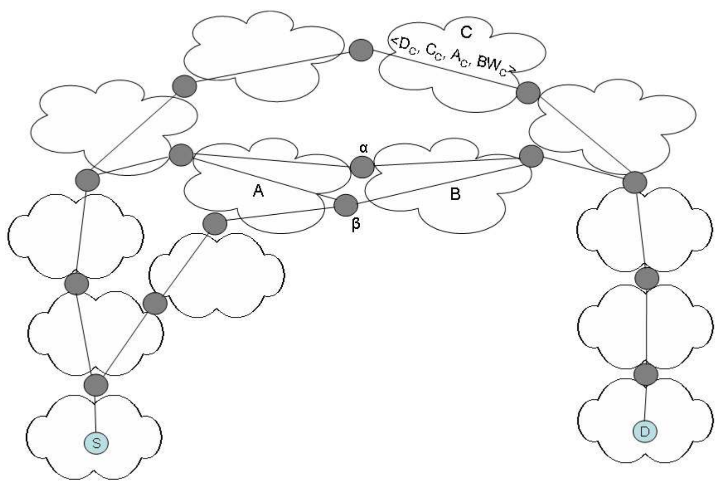


Figure 3.3: Graph Representation of Domain Connectivity

As mentioned before, a domain can have a set of service classes (e.g. domain C offers three service classes and domain B offers two). A domain can also have complex QoS class offerings based on its policies. For example, domain A offers two service classes ($\{D_{A1}, C_{A1}, A_{A1}, BW_{A1}\}$ and $\{D_{A2}, C_{A2}, A_{A2}, BW_{A2}\}$) to traffics coming from the domain to its west, and only offers one service class ($\{D_{A3}, C_{A3}, A_{A3}, BW_{A3}\}$) to traffics coming from the domain to its south. To incorporate these service classes onto the service graph, we first associate each edge of the graph with the set of possible service classes. Then, the edge is expanded by introducing a number of "service nodes", where one node of the original edge now connects to a service node via a new edge with a weight set representing one service class, and the other node of the original edge connects to the service node via a new edge with a *nil* weight set $\{0, 0, 1, \infty\}$. Then it is apparent that the number of service nodes introduced on such an edge is equal to the number of service classes associated with that edge. Figure 3.4 illustrates the edge expansions involving domains A, B and C.

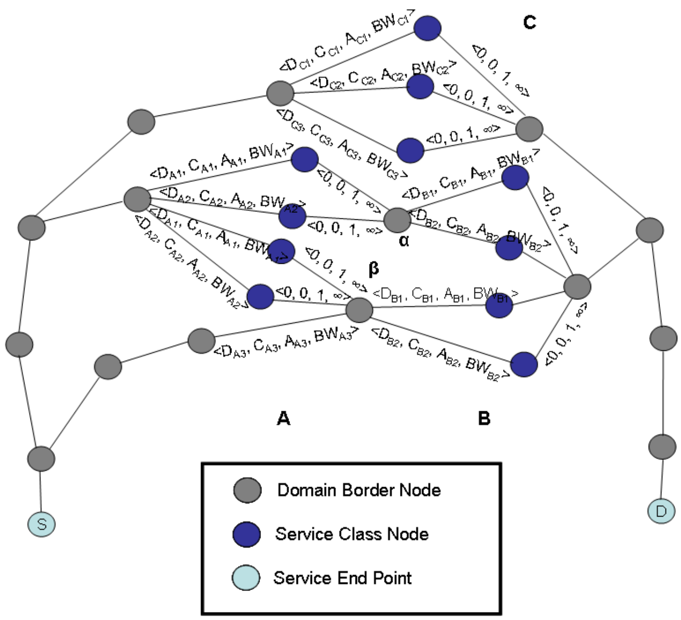


Figure 3.4: Service Graph with Service Class Expansion

A path from node S to node D on the service graph not only represents a possible sequence of interconnecting domains between the two end points, but also depicts a selection of respective service classes in these domains. By traversing through all possible paths between S and D, we can exhaustively search all possible service compositions between them. Thus, it is possible to formulate the composition and adaptation as a graph search problem.

Therefore, the composition and adaptation problems can be stated as: given the service graph $G(V, E)$, find a path $P = (\omega_1, ..., \omega_k, \omega \in E)$ from node v_s to node v_d such that the end-to-end delay $\sum_{i=1...k} D_i$ is below delay constraint κ_D, the end-to-end availability $\prod_{i=1...k} A_i$ is above availability constraint κ_A, the bandwidth BW of all edges in P is above bandwidth constraint κ_{BW}, and the cost $\sum_{i=1...k} C_i$ is below cost constraint κ_C. Such a path is termed a feasible path. Then, a minimal feasible path is a feasible path whose cost is minimal among all feasible paths.

Rather than dealing with heterogenous constraint conditions, we can rewrite the above constraints as:

$$\tau_1 = \frac{\sum_{i=1...k} D_i}{\kappa_D}, \tau_2 = \frac{\sum_{i=1...k} C_i}{\kappa_C} \tag{3.1}$$
$$\tau_3 = \frac{1 - \prod_{i=1...k} A_i}{1 - \kappa_A}, \tau_{4i} = \frac{\kappa_{BW}}{BW_i}$$

With respect to equations 3.1, the service composition problem can be formalized as:

Given an undirected graph G(V,E) and two nodes in V (v_s and v_d), where each edge $u \in E$ has weights $\{D, C, A, BW\}$, find a path P=(ω_1,...,ω_k, $\omega \in E$) connecting v_s and v_d such that $\tau_1, \tau_2, \tau_3 \leq 1$, $\tau_{4i} \leq 1$ for all ω, and $\sum_{i=1...k} C_i$ is minimal.

This is equivalent to the *k Multi-Constraint Optimal Path problem* (k-MCOP), which is known to be NP-Complete [66]. In the following subsection, we first present existing solutions to the k-MCOP problem, and then analyze the characteristics of the service graph.

3.3.2 Solutions to k-MCOP

The k-MCOP problem is well studied in the literature, particularly in the context of QoS routing. For purpose of analysis, we denote the number of nodes in the graph (including all node types) by N and the number of edges in the graph by E. Chen and Nahrstedt [67] proposed an approximation algorithm (*Chen*) for finding a feasible path. Their algorithm involves mapping $k-1$ real weights to $k-1$ integers in the range of 0 to x. A dynamic programming scheme is then used to obtain a feasible path. The runtime complexity of their algorithm is $O(x^2|N|^2)$. The probability of finding a solution with Chen's algorithm is directly related to the value space size of the weights. When a weight can take on a large set of real values, x must also increase proportionally such that there is an integer value in x very close to the weight values of a feasible path. Thus, the value of x must be very large in practice, especially when dealing with delay and availability metrics. Furthermore, Chen's algorithm does not seek optimal solution.

The limited path heuristic (LPH) [68] is constructed based on an extended Bellman-Ford algorithm. As the extended Bellman-Ford algorithm expands each node to keep track of all possible paths from source to the node, the overall runtime is exponential. The LPH heuristic attempts to obtain an approximation by limiting the number of paths stored at any node to X. They further prove that when $X = O(|N|^2 lg|N|)$, LPH can obtain near optimal solutions. However, to obtain this, the runtime of LPH ($O(X^2|N||E|)$) becomes very large for the purpose of adaptation. In addition, Bellman-Ford requires complete domain connectivity information at its initialization stage.

The TAMCRA [69] heuristic ($TAMCRA$) attempts to find a feasible path without optimization. The work proposes to use non-linear constraint mapping functions that maps k constraints to a single value. When designed carefully, such mapping function can produce the desirable effect of amplifying the resulting value when one of the weights is above constraint. The heuristic keeps track of K non-dominating paths at each node. A path is said to be non-dominating over another path if one of its weight values is better than the corresponding weight value of the other path. The runtime of TAMCRA can be quite large when K is high ($O(K|N|log(K|N|) + K^3k|E|)$). As TAMCRA is not optimized, a high value of K is necessary to obtain near optimal solution. An extension to the TAMCRA heuristic is later proposed [70], but the increased runtime complexity makes it impractical for dynamic service provisioning.

Korkmaz and Krunz [71] proposed a heuristic (H_MCOP) using a two pass Dijkstra's algorithm with look ahead. However, their look ahead heuristic is overly simplistic and their minimization step does not yield near optimal solutions. Nevertheless, H_MCOP is fast and does find feasible solutions when the constraint bound is loose.

Some investigations are conducted on the performance of these solutions [72]. TAMCRA and H_MCOP algorithms are illustrated to provide the best performance. However, we find both of these heuristics are inadequate for service provisioning due to the existence of many "two-hop loop" on a service graph. Figure 3.5 depicts such a loop. It is an abstraction of a domain with three service classes. Suppose the constraint set is {20,20}, both the TAMCRA and H_MCOP solutions will find the path highlighted in 3.5a), which is not a feasible path. 3.5b) highlights a feasible path. This problem arises because the non-linear mapping function only retains information on the highest aggregate weight at any node, and both solutions greedily explore the minimum of such aggregates. TAMCRA's limited backtracking capability allows it to sometimes recover from this condition, but when there are many two-hop loops on a graph, the heuristic could not perform well. This problem is formally analyzed in the proceeding subsection.

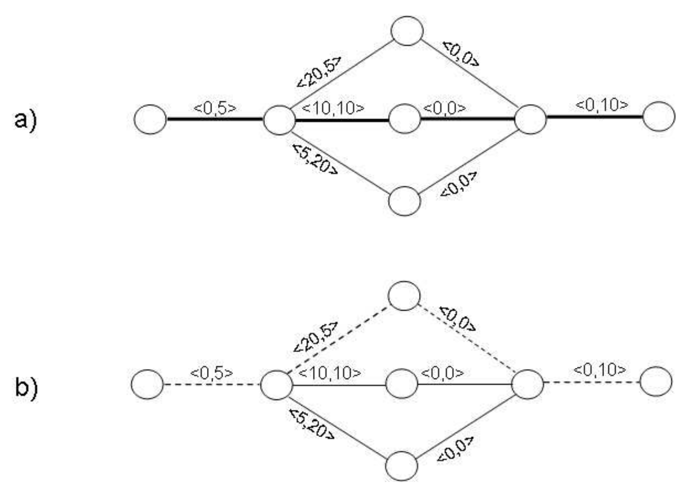


Figure 3.5: Example of the Two-hop Problem on Service Graph

3.3.3 The difficulty of service composition

We now provide a formal analysis of the service composition problem and show why the existence of many "two-hop loops" (tentatively call them "locks") makes the problem particularly difficult to solve.

In general, the k-MCOP problem is difficult because of the following characteristic: let node v be a node along the minimal feasible path P between source node s and destination node d. Then the sub-path of P from source s to node v may not be a minimal feasible path from s to v. Hence, greedy search algorithms such as Dijkstra's algorithm becomes ineffective in addressing the k-MCOP. To date, there exists many poly-time solvers to the k-MCOP problem. Recent performance study [72] shows that some of these heuristics perform well in the QoS routing problem domain. Further analysis [73] suggests that k-MCOP is not "strong" NP-Complete when applied to random graphs. The work further postulates that the tightness of the constraints and the correlations among the link weights could further influence the "hardness" of the problem. In our investigation [16], we have demonstrated similar results through simulation studies (reported in the simulation section), and more importantly our performance study has suggested the ineffectiveness of existing k-MCOP poly-time solvers in addressing the network service composition problem. We hereby analyze the characteristics of the network service composition problem and determine its "hardness".

Two techniques are commonly used in developing poly-time solutions to the k-MCOP problem. 1) use non-linear mapping function to map k weight values to a single variable onto which Dijkstra's algorithm is used to search out a feasible path. The H_MCOP heuristic [71] offers one such example. 2) Each node on the graph keeps track of n paths and thus attempts to capture a feasible path through clever path selection (such as keeping only dominant paths). The LPH heuristic [68] is one such example. Other solutions such as TAMCRA [69] uses a combination of both techniques.

If a nonlinear mapping function is to be used, ideally it should satisfy the following two requirements: a) the function produces smaller values for all feasible paths than the values it produces for infeasible paths. b) a path that minimizes such mapped value is also a minimal cost path. Consider the following function:

$$H(D, C, A) = \max(\frac{\sum_{i=1...k} D_i}{\kappa_D}, \frac{\sum_{i=1...k} C_i}{\kappa_C}, \frac{1 - \prod_{i=1...k} A_i}{1 - \kappa_A}) \tag{3.2}$$

$H(D, C, A)$ will be no greater than 1 when all of the weights are below constraints, thus satisfies requirement a). Furthermore, the value will always reflect the largest weight value in the set (i.e. closest to the constraint). This additional property is very useful in a greedy search strategy as it attempts to select paths with good overall QoS values. However requirement b) is not achievable with such nonlinear mapping functions.

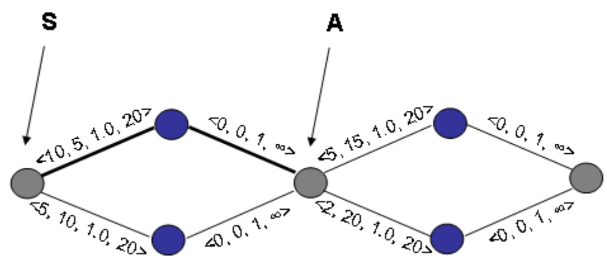


Figure 3.6: A Two Service Provider Service Graph

Unfortunately nonlinear mapping function does not provide effective solution to the network service composition problem. Consider the illustrated example in Figure 3.6. Suppose the constraint set is {15,15,0.9,10}. Then at node S, both the upper and the lower path yield identical $H(D, C, A)$ value and hence the upper link is chosen due to precedence. When the heuristic reaches node A, it is forced to commit to that upper path selection due to the lock topology and subsequently the heuristic will fail to return a feasible path even when there exists one (by taking the lower path of the first lock). This example may seem pathological and indeed it is. However, two general conditions make such occurrence likely. First, it is prone to happen in chain topologies where many such interconnected lock formations exist. Unfortunately, in examining the service graph construction, indeed such complex chain topologies frequently arise. Second, the illustrated case is much more likely to occur when some link weights are negatively correlated. Again, this is the case with service class descriptions where lower QoS metric values (e.g. delay, packet loss, etc.)

typically entails higher service cost.

The general technique of keeping multiple paths at each node during the search process appears to alleviate the above problem. Some heuristics, such as TAMCRA, uses the concept of path dominance. A path P kept at a node v is said to be a dominant path if for all other paths P_n kept at v, there exists ω_i of $P > \omega_i$ of P_n for some link weight ω_i, and there exists ω_j of $P < \omega_j$ of P_n for some link weight ω_j. Conversely, a path P of node v is said to be non-dominant if there exists some path P_n kept at v such that ω_i of $P > \omega_i$ of P_n for all i. Logically, a path is non-dominant when there exists another path that is strictly better than it. Hence it is clear each node only needs to keep track of dominant paths. Furthermore, if the total number of dominant paths remains relatively small throughout the search process, then keeping a fixed number of dominant paths at each node is an effective poly-time solver of k-MCOP. For the remainder of this subsection, We derive the bound on the number of dominant paths on a service graph.

Lemma 3.1: Searching a path on a service graph consists of searching a series of locks.

This is a direct result from the service graph construction. Given that each service provider is represented by a lock on the service graph, and all service providers are interconnected, one naturally obtain a graph of consecutive locks.

□

We extend the concept of path dominance to service class dominance. A service class K is said to be dominant if for all other service classes K_n in the same lock, there exists some weight ω_i of $K > \omega_i$ of K_n and some other weight ω_j of $K < \omega_j$ of K_n.

Lemma 3.2: A set of service classes containing negatively correlated weights are dominant service classes.

If two link weights ω of service classes are negatively correlated (e.g. delay and cost), then the resulting service classes are all dominant. In addition, the dominance is additive, in that when a path is extended through a lock containing L dominant service classes, it produces L dominant paths.

□

Theorem 3.1: The total number of non-dominant paths generated from exploring a lock is bounded by (n-1)(L-1), where n is the number of dominant paths before exploring the lock and L is the total number of service classes in the lock.

Case 1: n=1 or L=1. When L=1, exploring the lock is equivalent to adding the same weight set to each of n dominant paths. It is clear that such addition will not result in any non-dominant paths. When n=1, Lemma 3 guarantees that no non-dominant paths will be produced.

Case 2: n=2. Let the two dominant paths be A and B and let there be L service classes K_i. According to definition of dominant path, there must exist some ω of $A < \omega$ of B. Without the loss of generality, let:

$$\omega_1 \ \textit{of}\ A = x \tag{3.3}$$

$$\omega_1 \ \textit{of}\ B = x + \epsilon \tag{3.4}$$

Adding the same service class to two dominant paths yields two dominant paths. From Lemma 3.2, it follows that all paths resulting from adding the dominant service classes to A will produce a dominant set θ ={$A + K_1$,$A + K_2$,...}. This is also true for B, producing a dominant set η ={$B + K_1$,$B + K_2$,...}. Hence any non-dominant paths can only be the result of some path in η better than some paths in θ, or vice versa. Since the two cases are symmetrical, it is suffice to show one of them.

Assuming some paths in η of B is better than some paths in θ of A, a pathological case can be created by considering a list of dominant service class K with the following set of conditions:

$$\omega_1 \ \textit{of}\ K_j = \omega_1 \ \textit{of}\ K_{j+1} + \Delta \tag{3.5}$$

$$\omega_2 \ of \ K_j = \omega_2 \ of \ K_{j+1} - \Lambda \tag{3.6}$$

$$\omega_i \ of \ K_j = 0 \ where \ i > 2 \tag{3.7}$$

Thus for $A + K_j$ to be worse than $B + K_{j+1}$, and the following set of conditions must be true:

$$x + \Delta > x + \epsilon \tag{3.8}$$

$$\omega_2 \ of \ A > \omega_2 \ of \ B + \Lambda \tag{3.9}$$

$$\omega_i \ of \ A > \omega_i \ of \ B \ where \ i > 2 \tag{3.10}$$

These two sets of conditions are satisfiable, hence yielding $(L-1)$ non-dominant paths in the set θ. Because of the inequality condition placed on ω_2 of A and B from above, there could not exist any other non-dominant paths.

Case 3: n>2. The proof proceeds similar to Case 2. One can order the list of dominant paths by increasing ω_1 value $\{A, B, C, \ ...\}$ and could proceed to construct a pathological case where each path (e.g. B) when extended with the list of service classes could yield $(L-1)$ non-dominant paths due to being "worse" than that produced by extending the immediate successor (e.g. C) with service classes. Hence the bound on the number of non-dominant paths produced when exploring a lock is $(n-1)(L-1)$.

□

Theorem 3.2: The number of dominant paths on a service graph is strictly increasing during the path search process.

From Lemma 3.1, it follows that there are distinct steps to the path search process on the service graph, whereby each step involves the exploration of a lock. It is clear that exploring each lock produces nL paths where n is the number of dominant paths before exploring the lock and L is the number of service classes in the lock, and is strictly greater than the $(n-1)(L-1)$ upper bound number of non-dominant paths. Hence the number of dominant paths strictly increases during the path search process.

□

Although the bound from Theorem 3.1 is tight (i.e. one could construct such pathological cases that force the upper bound). The conditions placed on the weights are unusual. More specifically, the weight difference Δ between each consecutive service class must be strictly equal or increasing. For instance, the difference between ω of K_2 and ω of K_3 must be at least as large as the difference between ω of K_1 and ω of K_2. Furthermore, the difference ϵ between ω of two dominant paths must be less than Δ. In practice, such condition is not frequently encountered. However, running into a few of these problematic locks early on in the search process has significant impact on the success and optimality of the solution.

3.3.4 Design goals

An in-depth performance evaluation of these solutions is presented in Section 3.5. An effective service composition and adaptation solution must consider a number of factors. First, the solution must perform within reasonable time bound for adaptation to be effective (within hundreds of milliseconds). Although the actual efficiency of adaptation is also contingent on the efficiency of the information gathering facilities and the provisioning mechanisms, we believe the composition and adaptation solution should not become another significant factor to the overall runtime. Second, the cost and degree of disruption during adaptation should be minimized. Both of these factors are important for self-adaptation. Third, the solution should have a high probability of finding a feasible path when one exists and provide a near optimal solution with regard to cost. This factor directly impacts the effectiveness of self-configuration and self-optimization. Fourth, the solution should not introduce excessive communication overhead. Based on these guidelines, we develop new heuristics for the service composition and adaptation problem. In particular, we desire to find a solution that can effectively address the chain topology of the service graphs and provide a better near optimal solution compared with existing works. For analysis simplicity, we deal with undirected graphs in the investigation, while in practice the QoS conditions across two gateways of a domain could be different depending on the direction of the flow. Using directed service graph could account for this issue and since Dijkstra's algorithm could be applied to both directed and undirected graphs, our results also applies to directed graphs.

3.4 Composition and Adaptation for Autonomic Service Quality Assurance

In this section, we detail the development of new composition (*SComp*) and adaptation solutions for autonomic service quality assurance. We first introduce the construction of a non-linear mapping function and the service composition heuristic. Then we augment the solution to address the two-hop loop problem. Finally, the adaptation heuristic is presented.

3.4.1 Solution to the Service Composition Problem

Most of the existing heuristics for k-MCOP utilize the Bellman-Ford or Dijkstra's algorithm, since both of them are simple and fast, while still yield provable shortest path solutions for 1-MCOP. Our heuristic design favors Dijkstra's algorithm, as it relies on per-hop information in its search process. Two major issues are encountered in utilizing Dijkstra's algorithm to solve the k-MCOP problem. One, Dijkstra's algorithm uses a greedy search strategy without backtracking (hence the fast runtime bound). However, as the service graph analysis has shown, it is often the case that the minimal cost path will violate one or more constraints that forces the algorithm to backtrack. Two, if a mapping function is used to transform the k weights into a single value, it is difficult to ensure the following requirements: a) the function produces smaller values for all feasible paths than the values it produces for infeasible paths. b) a path that minimizes such mapped value is also a minimal cost path.

Let $H(D, C, A)$ be a mapping function that satisfies requirement a), our heuristic first runs Dijkstra's algorithm from v_d to v_s by minimizing $H(D, C, A)$. The purpose of this reverse search step is to determine whether there is a feasible path from every node v_j to v_d.

We denote this step as *MC_Search*. Then, we run the Dijkstra's algorithm from v_s to v_d by minimizing the cost. However, we include a node v_j on the shortest path only if the entire path from v_s to v_d through v_j is a feasible path. Such look ahead is possible as *MC_Search* provides this feasibility information from v_j to v_d. This prevents our heuristic from following a shortest path that would result in constraint violations at a later point along the path. We denote this cost minimization step as *MIN_Search*. *MIN_Search* also removes requirement b) from $H(D, C, A)$. For the remainder of this subsection, we first develop the mapping function $H(D, C, A)$, and then present the *MC_Search* and *MIN_Search* functions.

Consider the following non-linear function:

$$H^*(D, C, A) = \left(\frac{\sum_{i=1\ldots k} D_i}{\kappa_D}\right)^\lambda + \left(\frac{\sum_{i=1\ldots k} C_i}{\kappa_C}\right)^\lambda + \left(\frac{1 - \prod_{i=1\ldots k} A_i}{1 - \kappa_A}\right)^\lambda \tag{3.11}$$

When λ is set to a large constant value, $H^*(D, C, A)$ approximates very closely to Equation 3.2 we have presented during the analysis. Such a non-linear function is the basis for a number of k-MCOP heuristics [69][71]. Let $H(D, C, A)$ be such a maximization function. As we have discussed, $H(D, C, A)$ will be no greater than 1 when all of the weights are below constraints. Furthermore, the value will always reflect the largest weight value in the set (i.e. closest to the constraint). This additional property is very useful in a greedy search strategy as it attempts to select paths with good overall QoS values.

```
MC_Search(G=(V,E), vd, κD, κA, κC, κBW)
vd.r = 0, vd.wA = 1, vd.wD, vd.wC = 0;
P = { }, T = {vd};
while T ≠ { } do
    vc = min(vi.r), where vi are nodes in T;
    t_list = {neighbors of vc} \ P;
    for each node vi in t_list do
        if E(vi,vc).BW ≥ κBW then
            MC_ Search_Update(vi, vc, κD, κA, κC, E(vi,vc), T);
        end if
    end for
    remove vc from T and add vc to P;
end while
```

Figure 3.7: *MC_Search* function

The *MC_Search* function is presented in Figure 3.7. It tries to minimize the maximum weight from each node v_j to v_d. Each node keeps track of the following information: the maximum weight r of the minimal path from v_j to v_d, and the delay weight w_D, the availability weight w_A and the cost w_C of the path. The w weights are also used to compute complete path information in the *MIN_Search* function. Steps 1 and 2 are initialization steps. The list P contains nodes whose r values cannot be further improved. Steps 4 and 5 greedily select a node with the smallest r and obtain its improvable neighbors (i.e. either in T or newly discovered). *t_list* is a list of such neighbors. Steps 6 to 10 iterate through each member of *t_list* and update its r value if permissible. Figure 3.8 details *MC_Search_Update*. Steps 1 to 5 compute the new weights and r of a path from v_d to v_i via v_c, where v_i is the set of neighboring nodes of v_c that has not been added to P yet. If v_i is a newly discovered node, it is added to T (steps 6 to 8), otherwise v_i is updated iff. the new path has a smaller r value than v_i's old path (steps 9 to 11).

```
MC_Search_Update(vi, vc, KD, KA, KC, E(vi,vc), T)
 1  let vt be a temporary node;
 2  vt.wD = vc.wD + E(vi,vc).D;
 3  vt.wA = vc.wA * E(vi,vc).A;
 4  vt.wC = vc.wC + E(vi,vc).C;
 5  vt.r = max(vt.wD/KD, (1-vt.wA)/(1-KA), vt.wC/KC);
 6  if vi not in T then
 7      add vi to T;
 8      vi.r = vt.r, vi.wD = vt.wD, vi.wA = vt.wA, vi.wC = vt.wC;
 9  else if vt.r < vi.r then
10      vi.r = vt.r, vi.wD = vt.wD, vi.wA = vt.wA, vi.wC = vt.wC;
11  end if
```

Figure 3.8: *MC_Search_Update* function

Figure 3.9 illustrates the operation of *MC_Search*. In 3.9a, node α is the current best node. The value r of node α is obtained by taking the highlighted path. Three neighbors of node α are updated (in 3.9b) and node α is then added to P as it could not be further improved by the heuristic.

The *MIN_Search* function (Figure 3.10) is identical to the *MC_Search* function. Each

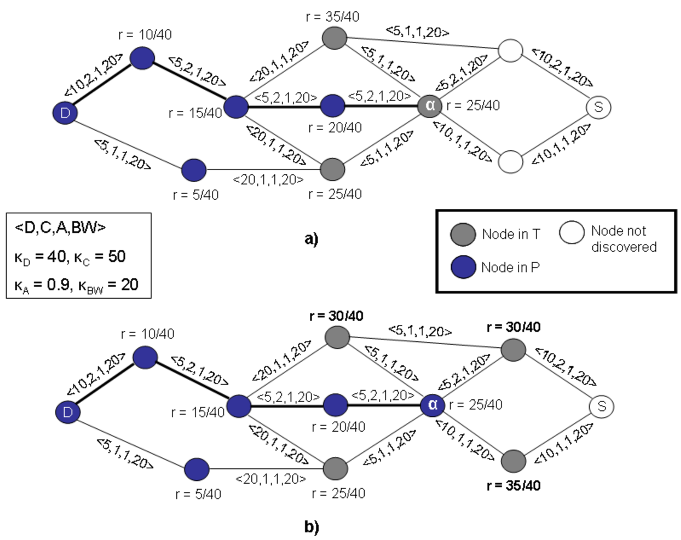


Figure 3.9: An Example of *MC_Search*

node v_j keeps track of the cost $v_j.f$ of a minimal feasible path from v_s to v_j, the predecessor $v_j.l$ of v_j on the said path, the delay $v_j.h_D$ of the path, and the availability $v_j.h_A$ of the path. The function tries to find the minimal feasible path from v_s to v_d.

```
MIN_Search(G=(V,E), vs, κD, κA, κC, κBW)
vs.l = nil, vs.f = 0, vs.hD = 0, vs.hA = 1;
P = { }, T = {vs};
while T ≠ { } do
   vc = min(vi.f), where vi are nodes in T;
   t_list = {neighbors of vc} \ P;
   for each node vi in t_list do
      if E(vi,vc).BW > κBW then
         MIN_ Search_Update(vi, vc, κD, κA, κC, E(vi,vc), T);
      end if
   end for
   remove vc from T and add vc to P;
end while
```

Figure 3.10: *MIN_Search* function

The *MIN_Search_Update* function (Figure 3.11) first computes the path cost and weights from v_s to v_i via v_c (steps 1 to 4). If no foreseeable feasible path exists from v_s to v_d via v_c and v_i, the function returns (steps 5 to 7). Otherwise, v_i is added to T if it is a newly discovered node (steps 8 to 10), or v_i is updated if the new path has lower cost than the old path (steps 11 to 13).

```
MIN_Search_Update(vi, vc, KD, KA, KC, E(vi,vc), T)
 1  let vt be a temporary node;
 2  vt.hD = vc.hD + E(vi,vc).D;
 3  vt.hA = vc.hA * E(vi,vc).A;
 4  vt.f = vc.fC + E(vi,vc).C;
 5  if (vt.hD + vi.wD) > KD or (vt.hA * vi.wA) < KA then
 6     return;
 7  end if
 8  if vi not in T then
 9     add vi to T;
10     vi.l = vc, vi.f = vt.f, vi.hD = vt.hD, vi.hA = vt.hA;
11  else if vt.f < vi.f then
12     vi.l = vc, vi.f = vt.f, vi.hD = vt.hD, vi.hA = vt.hA;
13  end if
```

Figure 3.11: *MIN_Search_Update* function

The operation of *MIN_Search* is illustrated in Figure 3.12a. Node α has three improvable neighbors β_1, β_2 and β_3. However, only β_2 is added to T, as following β_1 or β_3 does not lead to feasible paths. This look ahead property prevents the heuristic from examining lower cost paths (via β_1 or β_3 in this case) that may not be feasible.

Now, we present the service composition heuristic (Figure 3.13) that utilizes the *MC_Search* and *MIN_Search* functions. The heuristic terminates early if *MC_Search* does not return a feasible path. Otherwise, it will optimize on such a feasible path p using *MIN_Search*, which yields a feasible path p^* with cost at least as low as p. Figure 3.12c illustrates the result of the service composition heuristic, which is a minimal feasible path on the graph. Clearly, this path is not the minimal cost path (as generated in Figure 3.12b) which violates the delay constraint.

The service composition heuristic has double the runtime of Dijkstra's algorithm, hence

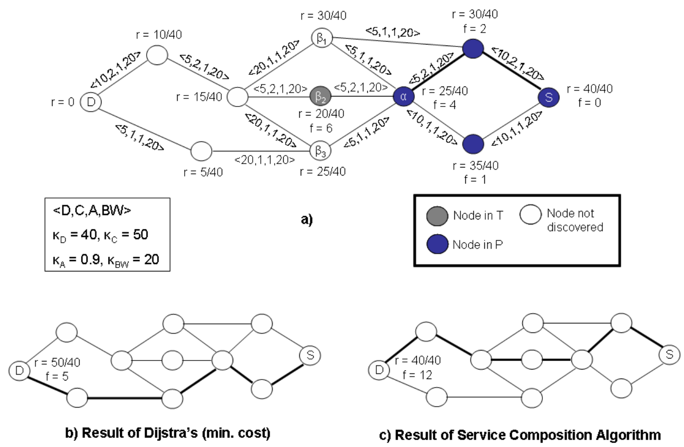


Figure 3.12: Example of the Service Composition Heuristic

```
Service_Composition(G=(V,E), vs, vd, κD, κA, κC, κBW)
  MC_Search(G=(V,E), vd, κD, κA, κC, κBW);
  if vs.r > 1 then
     return nil;  //no feasible path
  end if
  MIN_Search(G=(V,E), vs, κD, κA, κC, κBW);
  return the path by following vd.l;
```

Figure 3.13: The Service Composition Heuristic

the complexity O($2(|E|+|N|log|N|)$) where $|E|$ is the total number of edges on the service graph and $|N|$ is the total number of nodes (including service nodes) on the service graph. This is proportional to the total number of service classes in the domains interconnecting v_s and v_d.

3.4.2 The Hybrid Composition Heuristic

The service composition heuristic we have presented in the above subsection suffers from the two-hop loop problem similar to the TAMCRA and H_MCOP heuristics. To address this issue, we augment the service composition heuristic with an additional full path heuristic. During the backward feasibility search, in addition to performing *MC_Search*, we also perform Dijkstra's algorithm twice, once to find the minimal delay from any node to the source, and once to find the maximum availability from any node to the source. After this triple backward search step, each node will now hold not only the minimum non-linear mapping values to the source (and its path), but also the minimum delay and maximum availability paths. In general, these three paths are not the same. During the forward minimization step, in addition to performing *MIN_Search*, we also perform *MIN_Search_Full*, which instead of minimizing cost, attempts to minimize the full path cost projection based on the three paths recorded at each node. The hybrid heuristic returns the better result between *MIN_Search* and *MIN_Search_Full*. Figure 3.14 presents the hybrid service composition heuristic.

```
Service_Composition(G=(V,E), vs, vd, KD, KA, KC, KBW)
  MC_Search(G=(V,E), vd, KD, KA, KC, KBW);
  Dijkstra_Delay(G=(V,E), vd);
  Dijkstra_Avail(G=(V,E), vd);
  If vs.r <= 1 then
     MIN_Search(G=(V,E), vs, vd, KD, KA, KC, KBW);
  end if
  MIN_Search_Full(G=(V,E), vs, vd, KD, KA, KC, KBW);
  return the lowest cost path between MIN_Search
  and MIN_Search_Full if any
```

Figure 3.14: The Hybrid Service Composition Heuristic

MIN_Search_Full can recover some feasible paths that was deemed infeasible by the *MC_Search*. When the two-hop loop problem is encountered, although the non-linear mapping aggregate will lead to constraint violation, one of the single attribute minimizing path may help to keep such violating weight in check. Figure 3.15 and Figure 3.16 present *MIN_Search_Full* and *MIN_Search_Full_Update*. In *MIN_Search_Full_Update*, all three paths recorded by a node are evaluated (steps 4 to 25), and the node is evaluated based on the lowest full path cost projection instead of current path cost. The lowest full path cost is computed based on the current path cost to the node, plus the projected backward costs from destination to the node. In considering full path projections, the problem of greedily exploring lowest cost path that may lead to irrecoverable constraint violations (as in the case of two-hop loop problem) may be avoided. Furthermore, during the forward search, when a particular attribute grows near constraint, taking its minimal path may lead to feasible paths which SComp cannot find. Because *MIN_Search_Full* does not attempt to minimize cost, a path resulting from *MIN_Search_Full* could have higher cost than a path resulting from *MIN_Search*. The converse is also true, the full path cost projection can provide minimal solutions via lookahead, which a simple greedy search may not. Therefore, both the *MIN_Search* and the *MIN_Search_Full* functions are run in the hybrid heuristic.

```
MIN_Search_Full(G=(V,E), v_s, v_d, κ_D, κ_A, κ_C, κ_BW)
 v_s.l = nil, v_s.f = 0, v_s.h_D = 0, v_s.h_A = 1; v_s.h_C = 0;
 P = { }, T = {v_s};
 while T ≠ { } do
    v_c = min(v_j.f), where v_j are nodes in T;
    If v_c == v_d then
       return path;
    end if
    t_list = {neighbors of v_c} \ P;
    for each node v_i in t_list do
       if E(v_i,v_c).BW > κ_BW then
          MIN_ Search_Full_Update(v_i, v_c, κ_D, κ_A, κ_C, E(v_i,v_c), T);
       end if
    end for
    remove v_c from T and add v_c to P;
 end while
 return nil;
```

Figure 3.15: *MIN_Search_Full* function

Similar to *MIN_Search*, if *MC_Search* is able to find a feasible path, then *MIN_Search_Full* is guaranteed to return this path if it cannot find a lower cost path. This property holds because the full path projection of the path found through *MC_Search* will always be within constraints and *MIN_Search_Full* will follow another path of a lower cost if and only if that path is also a feasible solution (guaranteed by the full path projection). Figure 3.17 shows an example in which the hybrid service composition heuristic is able to recover a feasible path where the simple service composition heuristic cannot. After the three backward

```
MIN_Search_Full_Update(v_i, v_c, κ_D, κ_A, κ_C, E(v_i,v_c), T)
let v_t be a temporary node;
let v^kd, v^ka be the node v containing information from
    Dijkstra_Delay and Dijkstra_Avail respectively
v_t.f = ∞;
if (v_c.h_D + E(v_i,v_c).D + v_i.w_D) > κ_D
  or (v_c.h_A * E(v_i,v_c).A * v_i.w_A) < κ_A then
    v_t.h_D = v_c.h_D + E(v_i,v_c).D;
    v_t.h_A = v_c.h_A * E(v_i,v_c).A;
    v_t.h_C = v_c.h_C + E(v_i,v_c).C;
    v_t.f = v_c.h_C + v_i.w_C;
end if
if (v_c.h_D + E(v_i,v_c).D + v^kd_i.w_D) > κ_D
  or (v_c.h_A * E(v_i,v_c).A * v^kd_i.w_A) < κ_A then
    if (v_c.h_C + E(v_i,v_c).C + v^kd_i.w_C) < v_t.f then
        v_t.h_D = v_c.h_D + E(v_i,v_c).D;
        v_t.h_A = v_c.h_A * E(v_i,v_c).A;
        v_t.h_C = v_c.h_C + E(v_i,v_c).C;
        v_t.f = v_c.h_C + v^kd_i.w_C;
    end if
end if
if (v_c.h_D + E(v_i,v_c).D + v^ka_i.w_D) > κ_D
  or (v_c.h_A * E(v_i,v_c).A * v^ka_i.w_A) < κ_A then
    if (v_c.h_C + E(v_i,v_c).C + v^ka_i.w_C) < v_t.f then
        v_t.h_D = v_c.h_D + E(v_i,v_c).D;
        v_t.h_A = v_c.h_A * E(v_i,v_c).A;
        v_t.h_C = v_c.h_C + E(v_i,v_c).C;
        v_t.f = v_c.h_C + v^ka_i.w_C;
    end if
end if
If v_t.f == ∞ then
    return;
end if
if v_i not in T then
    add v_i to T;
    v_i.l = v_c, v_i.f = v_t.f, v_i.h_D = v_t.h_D, v_i.h_A = v_t.h_A;
else if v_t.f < v_i.f then
    v_i.l = v_c, v_i.f = v_t.f, v_i.h_D = v_t.h_D, v_i.h_A = v_t.h_A;
end if
```

Figure 3.16: *MIN_Search_Full_Update* function

searches, each node in the graph records the cumulative values of three paths: MC_Search, minimum delay, and maximum availability. For simplicity, only the results of MC_search and minimum delay are presented in Figure 3.17a. Since MC_Search does not return a feasible path (Figure 3.17b), the simple service composition heuristic fails. This is caused by the presence of a two-hop loop in the graph. The hybrid heuristic is able to recover a feasible path by using the path projected by the minimal delay search (Figure 3.17c).

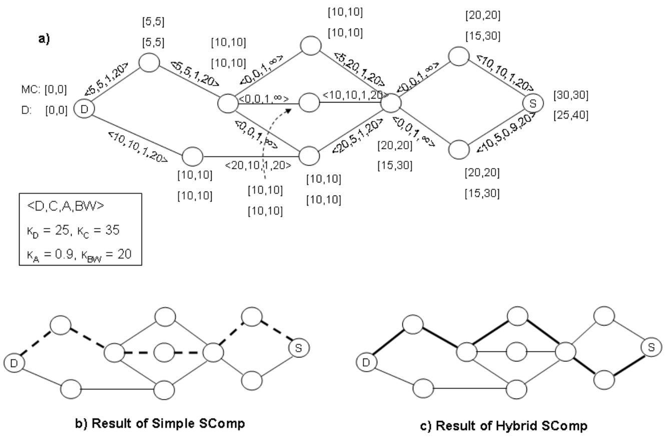


Figure 3.17: Example of the Hybrid Service Composition Heuristic

The runtime complexity of the hybrid composition heuristic is identical to that of the simple composition scheme, with a constant factor change of O($5(|E|+|N|log|N|)$) instead of O($2(|E|+|N|log|N|)$).

3.4.3 Network Adaptation and Mobility Support

Even with the best service composition solution, at runtime one or more domains carrying the service traffic may fail to deliver their promised QoS performance. When such an event occurs, self-adaptation should take place in seeking an alternative service path that satisfies the original QoS requirements, while causing as little service disturbance as possible. In this subsection, we present the network adaptation heuristic. The objective of the solution is to find a minimal cost alternative path p_{new} that can satisfy the end-to-end QoS requirement while utilizes as much of the old path p_{old} as possible.

The solution is based on the service composition heuristic. First, edge weights on the service graph are updated to reflect the new service conditions. Then, for each domain

```
SC_Adaptation(G=(V,E), p_old, v_s, v_d, κ_D, κ_A, κ_C, κ_BW)
  set cost of each edge in p_old to 0;
  for each domain travesed by p_old do
      set cost of other edges in the domain to the switching cost;
  end for
  Service_Composition(G=(V,E), v_s, v_d, κ_D, κ_A, κ_C, κ_BW);
```

Figure 3.18: The Network Adaptation Heuristic

traversed by p_{old}, set the cost of the edge in p_{old} (i.e. the chosen service class in the domain) to 0, and set the cost of the other edges (i.e. the other service classes in the domain) to the cost of switching to that class. Now the service composition heuristic can be run on this modified graph to obtain a new path. Figure 3.18 details the solution *SC_Adaptation*. More importantly, the cost minimization strategy also ensures that many parts of the old path is included in the new alternative path. With the proposed approach, the edges corresponding to the defective domains are not removed from the graph, but rather updated to reflect the new domain condition. Thus setting the costs of these edges to 0 does not prevent the heuristic from selecting a defective domain. It is designed in this way such that it is possible to obtain an alternative path that includes some/all of the defective domains. For example, when a domain in an existing service composition suffers QoS deterioration, it may be possible to raise the QoS service classes of its upstream and/or downstream domains in the existing composition and absorb the QoS deterioration. In this way, negotiating for new domain connections to bypass the defective domain is avoided, which is likely to be costly and disruptive. Figure 3.19a shows p_{old} in which two domains fail to deliver their promised delay requirements. The graph further shows two other domains that are willing to deliver better delay assurances at higher price. Figure 3.19b shows the result of running the *MC_Search* function which generated a feasible path. Figure 3.19c shows the new path p_{new} (returned by the heuristic) that improves on the cost of the path in 3.19b. The new path cost is the additional cost that must be absorbed by the violating domains in order to maintain the service.

In the case of mobility, the service endpoint will roam across domain boundaries, thus

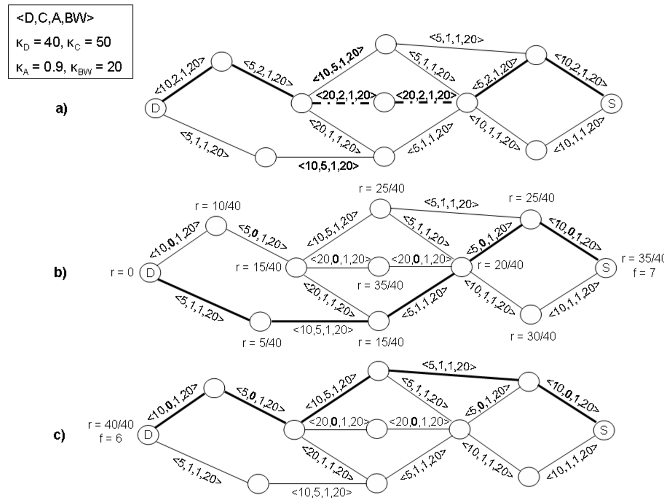


Figure 3.19: Example of the Network Adaptation Heuristic

triggering the self-adaptation process. This is in fact just a special case of the service adaptation process. First, the service graph is updated to include the endpoint's new locations, and then *SC_Adaptation* is run. Again, the heuristic will attempt to minimize the service disturbance by reusing parts of the old path. Figure 3.20 illustrates this scenario. Figure 3.20a shows the existing path from S to D before S moves. Figure 3.20b shows the result of running the adaptation heuristic after S moves. In this instance, the new path is a straightforward extension of the old path.

The adaptation heuristic can provide hard QoS guarantees over domains with relative service differentiations. The heuristic does not assume that each domain will always be able to fulfill its QoS promises, but rather relies on runtime adaptation as a safeguard against changes in network conditions. Moreover, the adaptation mechanism helps in network load balancing, in that when a domain experiences performance degradation, the adaptation process will lessens the traffic load on the domain or one of its service class by redistributing traffic to another service class or to other network domains.

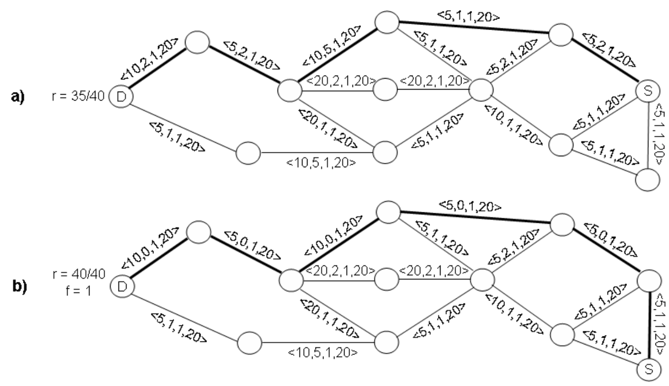


Figure 3.20: Example of Mobility Support

3.5 Simulation Study

In this section, we evaluate the runtime performance, success rate and effectiveness of the composition and adaptation solution against some well-known k-MCOP solutions in the literature. For completeness, we also include the SC-SON solution [13] proposed for service composition in Service Overlay Networks.

Two sets of service graphs are used in the study. The first service graph is constructed based on the ANSNET (ANSNET) topology (Figure 3.21), as presented in Chen's work [67]. The second service graph is constructed based on the Cable&Wireless (CNW) network topology in the US and UK (Figure 3.22). The CNW topology is of substantially larger size and contains significant number of service classes. In the illustrations, links with boxes depict domains with three service classes, while the other links have a single service class. The actual number of nodes on the graph is larger than shown since the service class nodes are omitted. At the start of each simulation run, the weight of each link is randomly distributed with relative differentiation between different service classes. A service class of higher cost offers better delay and availability values than a lower service class.

In each run, the graph weights are randomly generated as specified above and each solution is then asked to find the minimal cost feasible path between source (S) and desti-

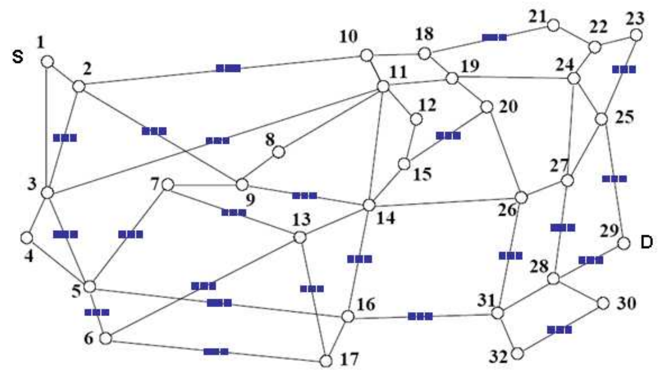


Figure 3.21: Service Graph of ANSNET Topology (ANSNET)

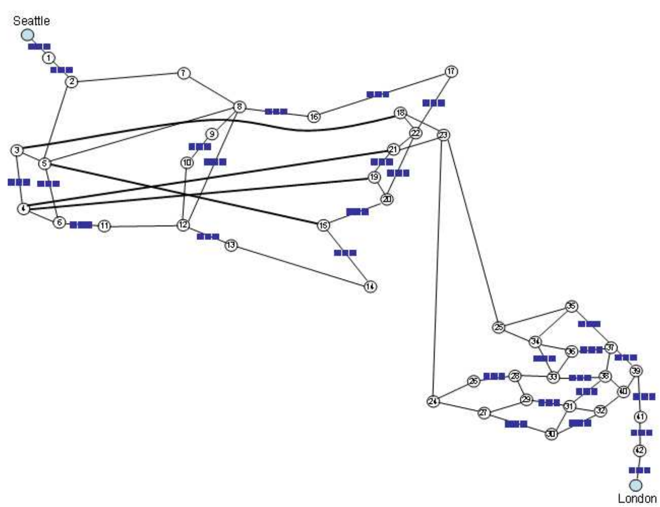


Figure 3.22: Service Graph of Cable&Wireless Topology (CNW)

nation (D) on ANSNET and between the node in Seattle and the node in London on CNW. As the base case, depth-first search is performed to find the minimal cost path on the graph.

The performance of each solution is evaluated using randomly generated graphs of size $|N|$. The path length between source and destination is at least 15 links. Figure 3.23 plot the average runtime of the solutions over 100 runs for each graph size. As shown through theoretical studies, the LPH and Chen solution performs significantly worse than the other solutions. The performance of our solution is roughly on par with TAMCRA when K is set to 3 (i.e. three non-dominate paths are kept at each node). As the service graphs of both the ANSNET and CNW topology has 100 or more nodes, we are particularly interested in solutions with acceptable runtime speed on graphs of this size. Excluding Chen and LPH, most of the solutions can return a result within 100 millisecond on graphs of size 100. The Chen's algorithm is able to return a result under 200 millisecond. For sake of comparison, depth first search algorithm takes on average 8 minutes with graphs of 50 nodes.

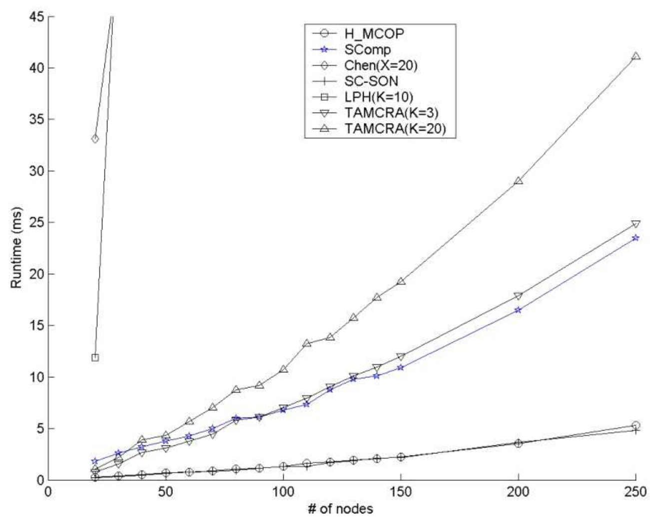


Figure 3.23: Runtime Performance on Random Graphs of Varied Size

The success rate of a k-MCOP solution is the percentage of time the heuristic could return a feasible path when feasible path(s) exist in a graph. To evaluate the success rate of k-MCOP solutions, most simulation studies in literature apply randomly generated QoS requirements on a graph with random weight initialization. Such construction does not accurately reflect the success rate of a solution, as there is no control over the constraint ratio. More precisely, when the delay requirement of a path is close to the minimum delay bound of the path (i.e. the shortest delay path), one can expect the success rate of a solution to drop, as there are few feasible paths. In the following study, we evaluate the success rate of each solution over different delay ratios, computed as:

$$DelayRatio = \frac{DelayRequirement}{Minimum\ Delay\ between\ src\ and\ dst} \tag{3.12}$$

Figure 3.24 and 3.25 present the success rate of these heuristics for ANSNET and CNW topology respectively. The difference between ANSNET and CNW topologies are the latter is significantly larger in size and also has many "two-hop loops". We also consider the impact of attribute correlations on the success rate. Three sets of correlations are evaluated: positive, negative and no correlations between delay and cost. For each delay ratio and attribute correlation, 50 runs are conducted. On the ANSNET topology, the SComp solution is able to achieve close to 100% success rate even under tight constraint bound. The TAMCRA solution also performs well. On the CNW topology, our hybrid solution again shows good performance, but as expected when constraint bound is tight the success rate is adversely affected, especially with negative correlation. The effect of the "two-hop loops" is observable in this study. When the delay ratio is tight (e.g. around 1.2), we see that all of the heuristics cannot obtain 100% success rate. The TAMCRA solution outperforms our heuristic under these tight conditions but only with a very high K value setting. In other words, TAMCRA at lower k settings (i.e. k=3), which is comparable to ours in terms of number of paths kept per node, does not resolve the loop issue nearly as effectively as our solution. The runtime of TAMCRA at K=20 is significantly higher than our hybrid service composition heuristic. In general, it is observed that with negative correlations which is the case in practice, our solution can fail to find feasible paths, but only under tight constraints.

The optimality of the solutions are demonstrated in Figure 3.26 and 3.27 over 50 runs. The advantage of the forward minimization step is apparent. Our hybrid composition heuristic often finds the optimal cost solution or near optimal cost solution. In comparison, the other solutions perform much worse. In this study, the percentage error ϵ (% error) is computed based on the marginal cost difference between the optimal cost solution C_{opt} (as obtained via depth-first search) and the solution obtained using one of the evaluated solutions C_{algo}:

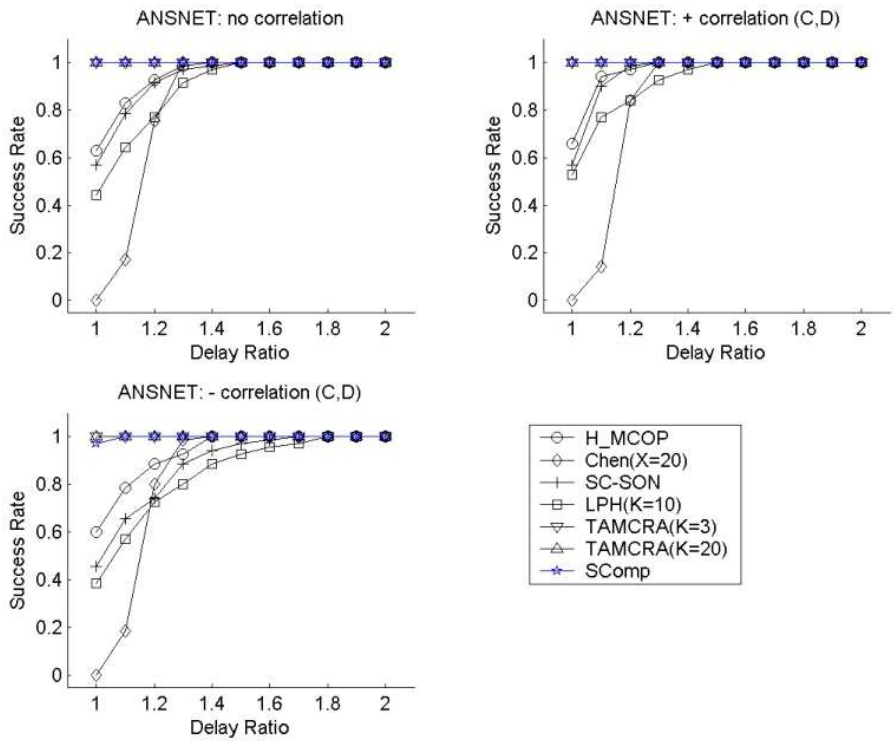


Figure 3.24: Success Ratio of Heuristics on ANSNET Topology

$$\epsilon = (C_{algo} - C_{opt})/C_{opt} \tag{3.13}$$

Figure 3.28 illustrates the percentage of times each solution is able to find the optimal solution on CNW topology. It further demonstrates the effectiveness of our solution in dealing with the chain topology of service graphs.

To evaluate the performance of the network adaptation heuristic, the path returned by the composition scheme is subjected to random quality defection. Each edge along a solution service path has a 10% independent probability of defection. On average, a path consists of 15 edges and each defective edge is assigned available bandwidth of 0Mb/s. The network adaptation heuristic is then performed in each run. Our studies show that when the number of defective domains is reasonable (i.e. up to 4 domains), the adaptation

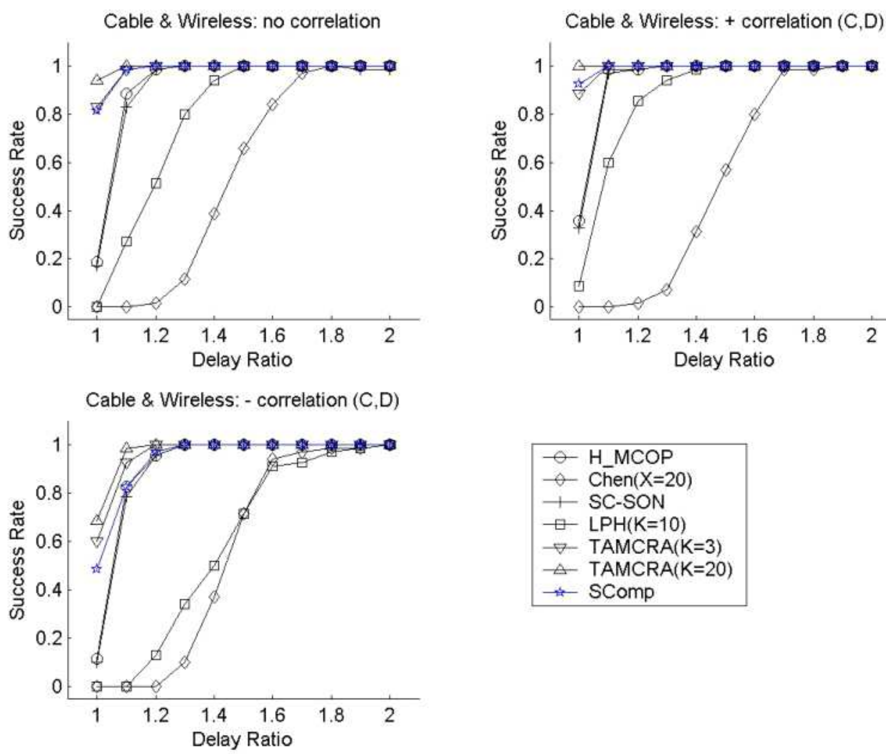


Figure 3.25: Success Ratio of Heuristics on CNW Topology

heuristic can reuse most of the old edges in the new path (69% or more). Even when subject to high defection rate (6 to 7 domains), the heuristic can still reuse 60% of the original path.

With fast runtime performance and good success rate compared to the classic k-MCOP solutions in the service composition context, the hybrid composition solution we have presented is able to find minimal cost feasible paths or near minimal cost paths. In particular, when subject to large scale networks with many service classes, our heuristic significantly outperforms the others in terms of computation time. Moreover, the network adaptation heuristic achieves good path reuse rate, while finding alternative feasible paths with low switching cost.

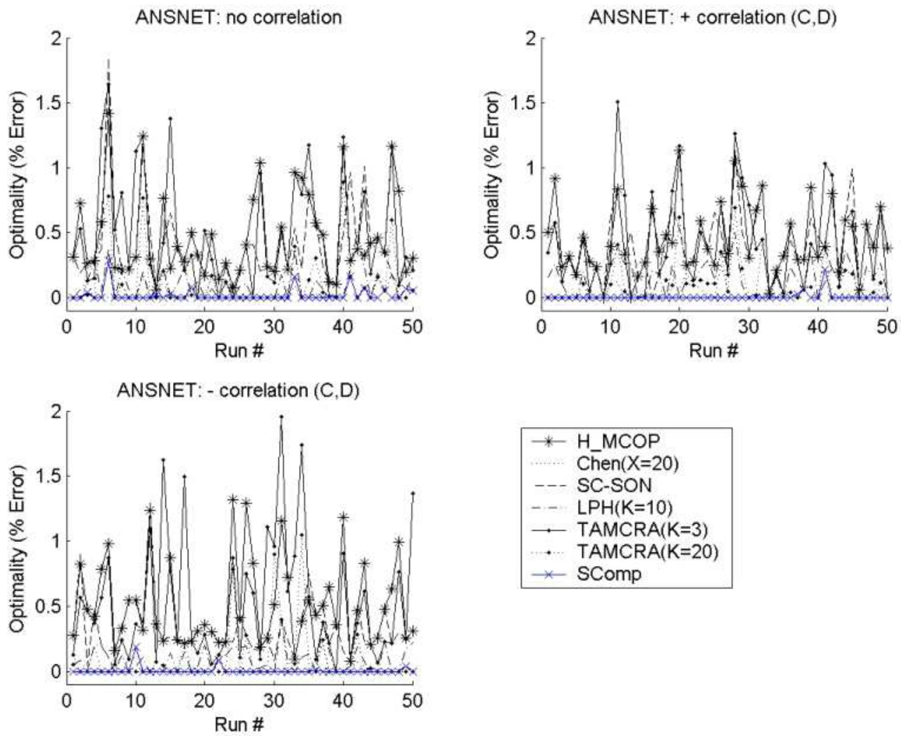


Figure 3.26: Relative optimality of Heuristics on ANSNET Topology

3.6 A Stability Study

In this section, we study the stabilization of the service composition and adaptation solutions through simulation. The Cable and Wireless network topology (Figure 3.22) is used. The QoS and cost metrics of each link and their respective service classes are set as a value range based on their relative length. At the start of a simulation run, the link weights are randomly generated from the specific value ranges. Each link on the graph has a QoS-assured capacity of 20 Mb/s and the two intercontinental links and the four cross US links (in bold) have QoS-assured capacity of 100 Mb/s. Our hybrid composition heuristic is used to study the system stability. The simulation uses simple QoS model based on congestion level. When the total load of a link is above a set threshold, QoS degradation is experienced in the respective domains, in which case adaptation is necessary for affected

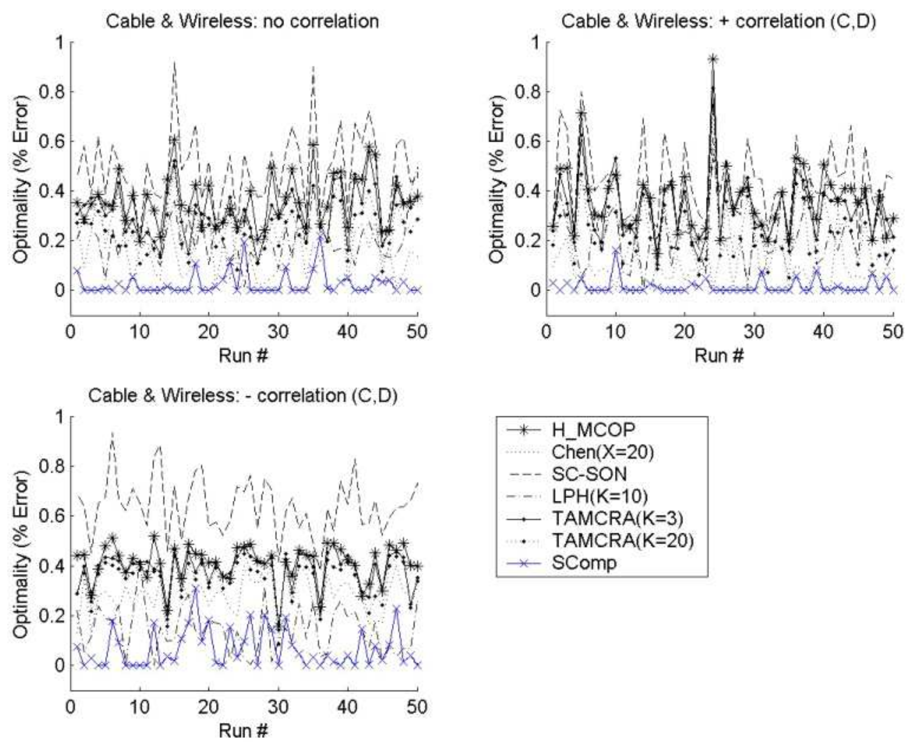


Figure 3.27: Relative optimality of Heuristics on CNW Topology

service flows.

Instead of having the service flows deterministically undergo path adaptation whenever QoS degradation is experienced, we introduce a parameter σ ($0 \leq \sigma \leq 1$), such that a service flow experiencing QoS degradation only performs path adaptation with probability σ. In other words, the adaptation process is stochastic. The rationale behind studying this process is as follows: consider a congested path shared by two service flows, it is often sufficient to shift one path away from the congested path rather than both, and we can obtain this effect stochastically with a σ setting of 0.5. Furthermore, this process has an observed effect on the stability of the system, as we document in the following case studies.

In the first simulation case, we study the ability of the system to return to normal state

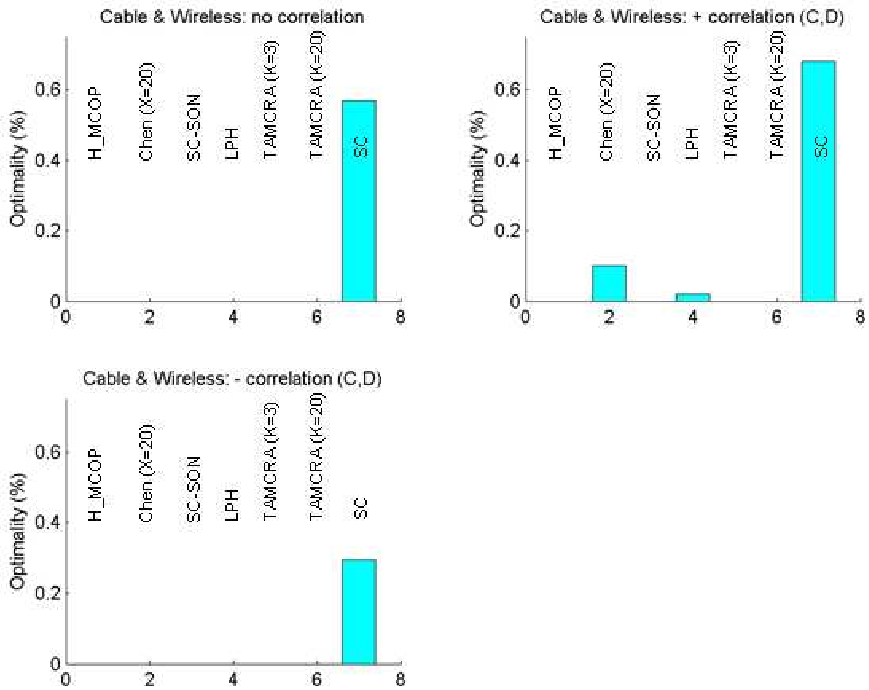


Figure 3.28: % optimality of Heuristics on CNW Topology

after disturbance. Severe congestions are induced in parts of the networks by using the data set from Table 3.1. The state of the system over time is illustrated in Figure 3.29. The circled line depicts the number of paths that require adaptation in a given iteration and the diamond line plots the number of paths that failed to obtain a new service path in that iteration. We see that the effect of σ was apparent. When service paths that are entailed adaptation do so 100% of the time, the system stays in oscillation. With a setting of $\sigma = 0.25$ the system is able to converge to a stable state within a short time period. In the last case, the system is severely overloaded with 25 service paths that interfere with each other and a cost modulation function is added that raises service cost based on the residual QoS-assured capacity of the link. It is observed that the system is able to return to stable state after some violent disturbances.

Src	Dst	BW	# of paths
0	106	5	4
0	40	5	2
66	106	5	1
11	90	5	4
14	48	5	2
90	63	5	2

Src	Dst	BW	# of paths
0	106	5	6
0	40	5	2
66	106	5	2
11	90	5	6
14	48	5	2
90	63	5	2

Table 3.1: Data Set 1

In the second simulation case, the effect of the heuristics on load balancing is shown. The simulation setup from the above experiment for 25 paths is used here. The network will undergo severe disturbance where parts of the network experiencing significant overload with this setup, and we examine the effect of the adaptation heuristic in balancing load. Figure 3.30 illustrates the link load conditions of the network at iteration 1, 5, 10 and 15 respectively. The x-axis depicts the load demand placed by service paths on a service link (in Mb/s) and the y-axis plots the number of links that have the specific load. As shown, in some cases the demand may exceed the link's QoS-assured capacity. The effect of load balancing is apparent, by iteration 5 most of the service paths have migrated to light load links and by iteration 15 the system has stabilized. This result is significant as the system was initiated in heavy load conditions with many overlapping service links as can be seen in the load condition of iteration 1. There are 6 links with load 35 and 40+ by iteration 15. These are 100 Mb/s high speed links in the network.

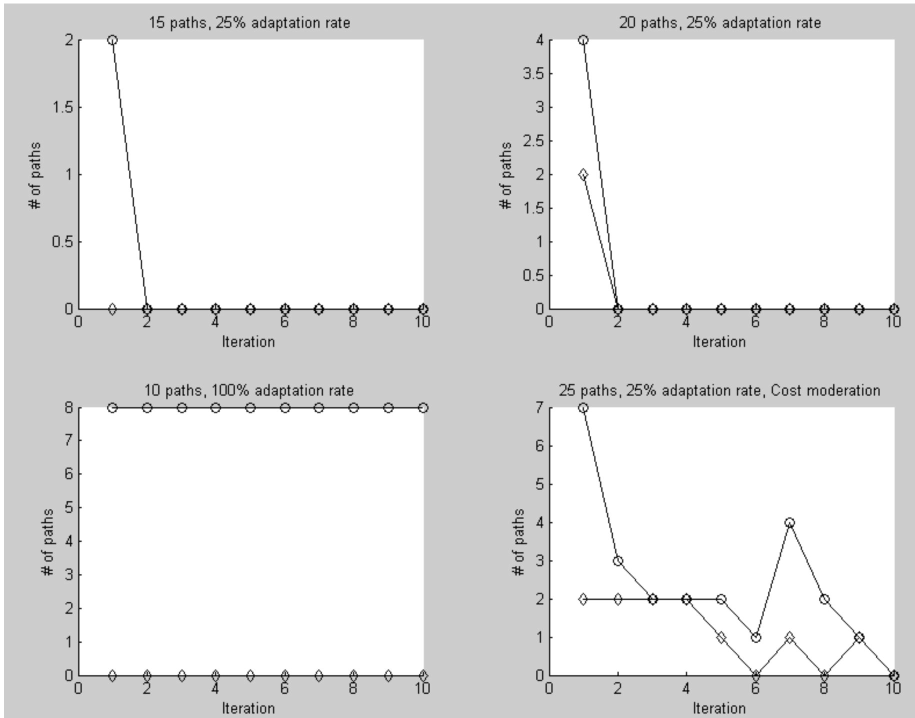


Figure 3.29: Adaptation Rate over Time

3.7 Conclusion

In this chapter, service composition and adaptation heuristics are developed for end-to-end service quality assurance. The proposed solution is established on the foundation of a bi-party collaborative framework consisting of the network and the service substrates. As this chapter has shown, it is possible to obtain end-to-end quality assurance at the service substrate based on the per-domain behavior of the network substrate. Although the theoretical problem can be reduced to the classic k-MCOP problem, as we have shown, the service composition and adaptation problem is particularly difficult due to the presence of "locks" in the service graph. Consequently existing k-MCOP heuristics cannot perform well in the service-driven networking context. We thus proposed new service composition and adaptation heuristics and demonstrated their effectiveness through comparative studies.

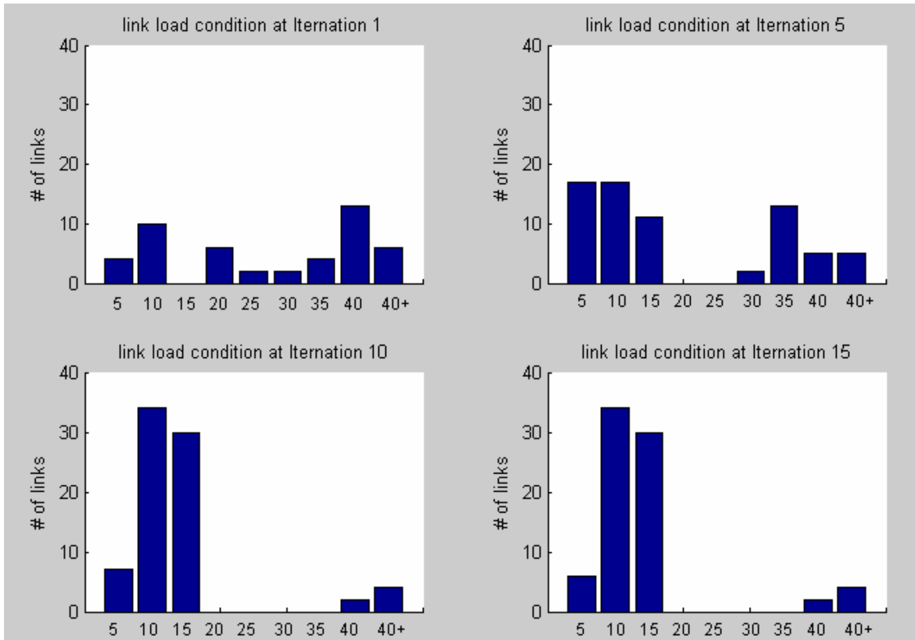


Figure 3.30: Link Load Conditions over Time

Although the simulation results are encouraging, there lacks a formal analysis of the system stability. This is an important issue in the service-driven networking context because the service flows are allowed to compose their own paths and to self-adapt in a shared resource infrastructure. The existence of underlay per-domain network management behavior further exacerbates the problem. The following chapter focuses on these issues.

Chapter 4

Self-Stabilization in Uncoordinated Environment

4.1 Introduction

In Chapter 3, we have described a mechanism for providing service quality assurance over multi-domain networks. Given the uncoordinated and self-motivated nature of the customers and the providers, there is a strong need to ascertain the stability and optimality of such an environment as a whole. In this chapter, we conduct a formal analysis on the stability and optimality of such an environment using interaction of overlays and underlays as a case study. The reason for choosing this particular case study is two fold. One, the construction and behavior of the overlay networks are quite similar to our description of a service (in fact we can view an overlay as a collection of service flows). Two, there are known literature results, both theoretical and experimental, on the interaction of overlays and underlays for me to build upon and compare to. This provides the necessary practical grounding on which the results of our analysis are evaluated.

The proliferation of overlay networks has a profound impact on the Internet operations. It is expected that many overlay networks will be deployed on the same network infrastructure across multiple network domains. Whereas traditional network management and route optimization were conducted exclusively at the underlays to ensure global network performance and stability, the presence of overlays deviates traffic from their underlay routes to achieve their application-level requirements (Figure 4.1). Thus, the global network behavior is the result of interactions among the overlay routing decisions and the underlay management actions. To truly understand this problem, one must deal with the non-cooperative behavior of the overlay interactions, the independent nature of the underlay domains, and the inherent disparity between the overlay and underlay operations, all of which are separate subjects of research. Thus it is difficult to assess the stability of the networks especially in multi-domain environment where many overlays and underlay providers influence the routing decisions based on their own (sometimes divergent) objectives. To date, several studies on overlay interactions have been conducted. A common approach is to model the overlay interactions as a game and establish the existence of pure Nash equilibrium (sometimes unique) in such a game (e.g. [74][75]). A particular type of game called *congestion game* has been quite useful in modeling overlay networks, due to the observation that when a congestion game exhibits a global potential, it guarantees the existence of a pure Nash equilibrium and in some cases convergence is the natural result of selfish game plays without knowing the strategies and payoffs of the other players. In the past, analysis has been conducted to establish the existence of pure Nash equilibrium in congestion games and to determine their complexity (e.g. [76][77][78]). In general, there is no guarantee that a pure Nash equilibrium exists in all congestion games [76], and when it does, the convergence on asymmetric games with polynomial cost function could be exponential in worst case [77]. On the subject of underlay and overlay interactions, existing works show that such interactions can easily lead to instability in the networks [14] and such self-serving behavior results in a performance tussle [15] between the overlay layer and the underlay layer.

In analyzing complex and largely distributed problems using abstracted game models, much of the physical complexity is often stripped away in favor of tractability. Thus one sometimes arrives at overlay game models that are gamey and no longer resemble the real world problem at hand. For instance, the early K-P model [79] is a single commodity model that assumes all players have a common source and destination, and uses a single link. In the often studied symmetrical congestion games, all players share the same strategy set (i.e.

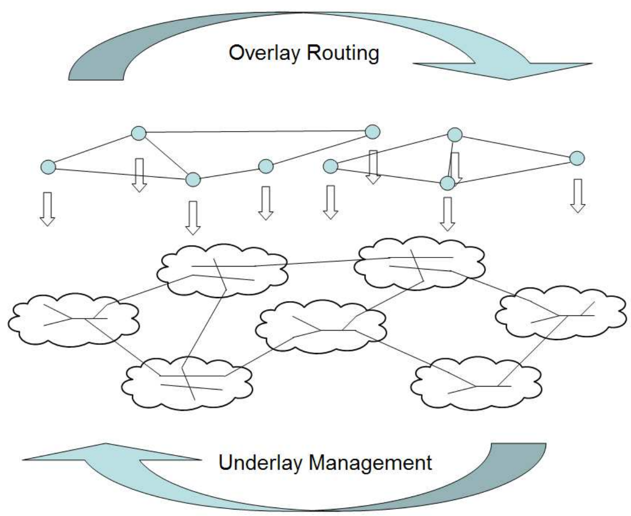


Figure 4.1: Overlay and underlay interactions in multi-domain networks

same source-destination and a common set of potential routes). Furthermore, even given a realistic game model, it is argued [80] that the resulting game solutions are often difficult to implement due to practical issues such as information sharing, privacy, cost-effectiveness, global synchronization, etc. Hence when conducting game theory analysis on a practical problem, it is important to make sure that the game model reflects the principle problem at hand and the solution has practical grounding and offers implementable guidelines or insights.

Also of particular interest to our research context, the results established from game theory analysis lend naturally to the concept of self-stabilization. In his seminal work [81], Dijkstra describes a self-stabilizing system as a distributed system where individual component behaviors are determined by a subset of global system state that is known to the component. The global system must exhibit the property of: "regardless of its initial state, it is guaranteed to arrive at a legitimate state in a finite number of steps". Two key properties are outlined: 1) the system can initialize in any state; 2) the system can always recover from transient faults. The concept of legitimate states is important to self-stabilization. A legitimate state denotes an operational state of a system that is within the design consideration of the system. Thus, as long as the system is in a legitimate state, its operation is expected or bounded. The self-stabilization concept recognizes that the set of all possible states of a system is rather large in the face of arbitrary faults and attacks, most of which are outside the system design. When discussing self-stabilization, the concept of closure and convergence [82] are often mentioned. The closure property describes that if the system is stable (i.e. in a legitimate state), after a number of executions it will remain stable unless perturbed by external force. The convergence property states that if the system is unstable (i.e. not in a legitimate state), it has a tendency to be stable over time. The length of time that must elapse for the system to move from an illegitimate state to a legitimate state is sometimes referred to as the convergence span. In essence, the properties of self-stabilization help in addressing many of the autonomic management properties. The concept of legitimate state is a strong motivation for our establishment of a desirability concept in games.

In this chapter, we study whether the interaction of non-cooperative overlays and underlays in multi-domain networks can result in stable global configurations? First, we motivate a new concept of desirability and construct a congestion avoidance game model of overlay interactions. We show such game converges to a pure Nash equilibrium. Then through a game transformation, we bound the convergence and show that its desirability ratio naturally leads to a price of anarchy bound. We also discuss important implementation issues. Secondly, we model the interaction of overlays and underlays as a dual-game consisting of a short-term overlay game and a long-term overlay-underlay game. This is motivated by the fact that the underlay network management actions occur at a much larger time granularity than the overlay routing decisions. We abstract the short term game as a single aggregate overlay player that plays against an aggregate underlay player. Both games are modeled as staged congestion games and we show that stability can be obtained when the following conditions are met: 1) the overlays can stabilize in a non-congested configuration within the time interval of underlay management actions; 2) both the overlays and the underlays have aligned objective functions as motivated by our desirability concept; 3) the overlays have control over domain traversal. We discuss the feasibility of these guidelines in practice and show why ad hoc practices such as traffic throttling are extremely harmful. Through extensive simulations, we study the overlay underlay interaction and compare the results with our theoretical findings.

The rest of the chapter is organized as follows. Section 4.2 presents related works. Section 4.3 analyzes the overlay interactions and discusses implementation techniques. Section 4.4 shows the long term overlay underlay interactions. Simulation study is presented in Section 4.5.

4.2 Related work

On the topic of overlay network routing, Liu et. al. [14] have studied the interaction between overlay routing and underlay traffic engineering. They modeled the selfish optimization of the overlay routing and the load balancing objective of the underlying traffic engineering as a two-player non-cooperative non-zero sum game. They found that although equilibrium exists in simple networks, oscillations and inefficiency arise due to the different objectives in general topologies. Seetharaman [15] et. al. further expanded along this direction in their study of the routing performance of underlay and overlay layers using a Stackelberg approach. They were able to show that performance gain could be obtained when one layer takes the leader role while the other player takes a follower strategy. They further noted that the degree of performance gain for each layer is a win-lose struggle determined by the selfishness of a layer. Using realistic topologies and traffic demands, Qiu et. al. [83] presented through simulation that contrary to theoretical worst case, selfish routing in overlay achieves close to optimal average latency. They also noted that some links in the network will have significantly increased congestion. Keralapura et. al. [84] showed that racing conditions exist between non-cooperative overlays due to inadvertent synchronization. Their result confirmed the observation that asymmetric congestion games in general cannot allow simultaneous moves. Zhang, Kurose and Towsley [85] showed that when underlay topology is rich, overlay can compensate underlay routing inefficiencies. To date, few work has considered the combined behavior of interacting overlays as well as overlay-underlay interactions in multi-domain setting.

Game theoretical analysis has been conducted in many fields of network research in the past (e.g. pricing, flow control, efficiency of wireless networks, etc.). Some works have examined the uniqueness of Nash equilibrium in non-cooperative user-based routing environment [74][75]. Orda, Rom and Shimkin [74] have shown that in two-node multi-link network topology, a unique Nash equilibrium exists. Altman et. al. [75] studied non-cooperative routing games under general topology network with polynomial cost function. They have shown the uniqueness of Nash equilibrium under bounded cost. Yaiche et. al. [86] modeled the bandwidth allocation problem as a Nash bargaining solution and devised a distributed optimization algorithm that is Pareto optimal. They are among the few earliest work that addressed game implementation and explicitly used social fairness as a performance criteria. Bottleneck routing game was investigated by Banner and Orda [87], in which the user attempts to minimize the load of its bottleneck link rather than to minimize the end-to-end cost. In a nutshell, the objective is a discontinuous MINMAX function. They have shown that in unsplittable bottleneck game, the worst case convergence bound is exponential $O(2^{|U|^2}|E|^U)$ and the price of anarchy is polynomial $O(|E|^p)$, where $|U|$ is the number of users, $|E|$ is number of links in the network, and p is a constant. In our formulation, we use an exponential objective function that is continuous and strictly convex, which exhibits a number of interesting properties compared with a strict MINMAX function. Furthermore, the formulation of our problem considers both the interactions among overlays and the overlays-underlays game.

Congestion games were first introduced by Rosenthal [88] and later formalized by Monderer and Shapley [78]. It's a class of games in which the cost of a resource is a non-decreasing function depending on the number of players sharing the resource. A game is called a potential game when there exists a potential function such that the increase in utility of a player (or drop in cost) causes a decrease in potential. Fabrikant, Papadimitriou and Talwar [77] have shown that the complexity of finding a pure Nash equilibrium in asymmetric congestion games is PLS-complete. On the topic of convergence to Nash equilibria in congestion games, the single-commodity K-P model [79] is often used. Milchtaich [76] has shown that polynomial time convergence exists for players with varied payoff functions. Goldberg [89] bounded the convergence in such games to polynomial time, and Even-Dar and Mansour [90] considered the case in which all players can move simultaneously according to a Nash rerouting policy and have found a polynomial time convergence bound. Because of its complexity, the study of convergence in general congestion games has been mainly focused on finding convergence bound to approximate solutions. Christodoulou, Mirrokni and Sidiropoulos [91] bounded the solution after one round of best-response walk by all players to Θ(n)-approximate in general case. Chien and Sinclair [92] showed that when the increase in cost of adding a player is bounded ("bounded jump" condition), convergence to ϵ-Nash occurs in polynomial time. The congestion game model we are interested is multi-commodity asymmetric games with polynomial cost function. It has a worst case exponential convergence bound [87][93]. Therefore, we have established a theoretical approximate bound (under desirability concept) on its convergence.

In bounding the optimality of an equilibrium, price of anarchy (PoA) is frequently used [94]. It defines the ratio between the social cost of an equilibrium and the social optimal, and PoA is the worst case bound. Related to this concept is the Price of Stability (PoS) which defines the best case bound on the ratio [95]. Some research (e.g. [96][97]) have been conducted on bounding the PoA ratio in congestion games and in some cases tight bounds are found. Roughgarden and Tardos [98] have shown that for non-atomic (i.e. each player controls negligible portion of the network) and splittable flows, the price of anarchy is exactly $\frac{4}{3}$. This result is extended to atomic unsplittable flow case in their subsequent work [99]. Awerbuch et. al. [97] have shown that for atomic unsplittable flows, the price of anarchy is exactly 2.5 for a pure strategy game over unweighed players with linear latency cost function. In the above game, if the cost function is a general polynomial cost function of degree d, the price of anarchy bound is at least $\Omega(d^{d/2})$ and at most $O(2^d d^{d+1})$. In our work, we consider a uniform polynomial cost function whose price of anarchy is 1 (this can be shown rather simply). Furthermore we introduce the desirability ratio as a related concept, where the ratio is a measurement of optimality loss when a game transformation is performed, and obtain the price of anarchy accordingly.

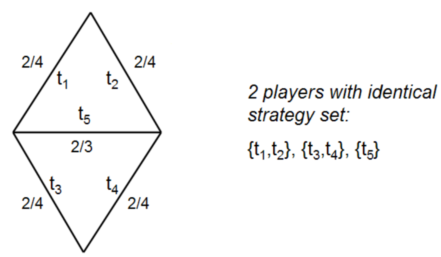
4.3 Interactions among overlays

In this section, we analyze the interactions among overlays. First we model the interactions as a congestion game with a congestion avoidance cost function. This is in contrast to the traditional delay minimization cost function common for overlays. Through a simple example, we show that the traditional delay minimization function easily leads to sub-optimal and sometimes non-working system equilibrium. Accordingly, we define desirability as an evaluation of a working system state and devise a congestion game whose equilibrium is always desirable. Our examination of the overlay interactions from a congestion avoidance perspective is further motivated by: one, network congestion is the major source of delay and transient network failures; two, the application-oriented nature of overlays means it is often sufficient to satisfy a delay constraint rather than minimizing delay; three, a congestion-minimization objective at the overlay layer is a necessary pre-condition for overall network stability as we will show in Section 4.4.

Utilizing a game transformation technique, we bound the convergence span of our game model. The resulting equilibrium is always desirable and we define the loss of optimality as a desirability ratio and from which price of anarchy is computed. In addition, we discuss implementation issues that arise when applying this game model in practice.

4.3.1 Interaction of overlays as a congestion game

For an overlay, we consider each source-destination pair in the overlay traffic matrix to have a set of candidate overlay routes. Each overlay route traverses through a consecutive list of underlay domains. This definition is consistent with the definition of service in our context and fits with the service quality assurance framework we described in Chapter 3. We represent each domain as one or more virtual resource links t. Details of this mapping and its implication on underlay network management is discussed in Section 4.4. We define a unit traffic κ as a discrete volume of traffic and consider each overlay source-destination pair is split into finite number of such unit traffic. We define the capacity of a resource link t_j as the amount of traffic it can serve and normalize its value according to κ. Thus, a resource link abstractly represents an underlay domain's capacity to host overlay traffic and is considered in working state unless the capacity is exceeded by overlay demand. Also, each unit traffic can be routed independently of each other and is considered a player in our game model. Accordingly, we define the overlay routing game as:



Player 1 \ Player 2	$\{t_1,t_2\}$	$\{t_3,t_4\}$	$\{t_5\}$
$\{t_1,t_2\}$	(2,2) **7.5**	(1.5,1.5) **7**	(1.5,1*) **6.5**
$\{t_3,t_4\}$	(1.5,1.5) **7**	(2,2) **7.5**	(1.5,1*) **6.5**
$\{t_5\}$	(1*,1.5) **6.5**	(1*,1.5) **6.5**	(1.3*,1.3*) **6.3**

Figure 4.2: An example of two-player Γ_B

Let $\Gamma_B = \langle N, \{Y_i\}_{i\in N}, \{u_i\}_{i\in N}\rangle$ be a game in strategic form. N is the finite set of players $\{1,\ldots,n\}$, Y_i is the finite set of strategies available to player i and $u_i : Y \rightarrow \Re_+$ where $Y = Y_1 \times Y_2 \ldots \times Y_n$ is the cost function of player i. Given a finite set of resources $T = \{t_1,\ldots,t_m\}$, define $Y_i \subset 2^T$. Let $A_i \in Y_i$ be a strategy of player i, $A \in Y$ be a strategy profile, c_j be the cost function of resource t_j, and l_j be the normalized serving capacity of t_j, then

$$u_i(A) = \textstyle\sum_{j\in A_i} c_j(A)$$
$$c_j(A) = \frac{x_j(A)}{l_j}$$
$$x_j(A) = \#\{i \in N : t_j \in A_i\}$$

Γ_B is a multi-commodity asymmetric unweighed game. The cost function c_j of resource t_j is a strictly increasing function of the number of players using t_j. This simple cost function exhibits similar properties as a delay function. The game is multi-commodity since a strategy includes more than one resource and the game is asymmetrical since each player may have different strategy set (e.g. different overlays have different routes). The game is unweighed in that each player has unit load κ. It is straightforward to show that Γ_B has an exact potential.

THEOREM 4.3.1: *A congestion game $\Gamma_B = \langle N, \{Y_i\}_{i\in N}, \{u_i,\}_{i\in N}\rangle$ has an exact potential.*

Define $\phi = \sum_{j=1}^{T} \sum_{k=0}^{x_j(A)} \frac{k}{l_j}$. Given player i changes strategy from A_i to A_i', the change in potential is:

$$
\begin{aligned}
\phi_i - \phi_i' &= \sum_{j\in(A_i\cup A_i')} \sum_{k=0}^{x_j(A)} \frac{k}{l_j} - \sum_{j\in(A_i'\cup A_i)} \sum_{k=0}^{x_j(A')} \frac{k}{l_j} \\
&= \sum_{j\in(A_i-A_i')} \sum_{k=0}^{x_j(A')} \frac{k}{l_j} + \sum_{j\in(A_i-A_i')} \frac{x_j(A)}{l_j} \\
&\quad + \sum_{j\in(A_i'-A_i)} \sum_{k=0}^{x_j(A)} \frac{k}{l_j} - \sum_{j\in(A_i'-A_i)} \sum_{k=0}^{x_j(A)} \frac{k}{l_j} \\
&\quad - \sum_{j\in(A_i'-A_i)} \frac{x_j(A')}{l_j} - \sum_{j\in(A_i-A_i')} \sum_{k=0}^{x_j(A')} \frac{k}{l_j} \\
&= \sum_{j\in(A_i-A_i')} \frac{x_j(A)}{l_j} - \sum_{j\in(A_i'-A_i)} \frac{x_j(A')}{l_j} \\
&= u_i(A) - u_i(A')
\end{aligned}
$$

□

Thus, ϕ is an exact potential function of Γ_B. Since a solution always exists when minimizing the value of ϕ over Y, there must exist a pure Nash equilibrium in Γ_B.

A congestion game with an exact potential not only has a pure Nash equilibrium, but also has the finite improvement property (FIP). Hence from an arbitrary state, following the best-reply path, the game is guaranteed to converge to the pure Nash equilibrium over time, albeit exponential in worst case. Figure 4.2 shows a two-player game of Γ_B. The graph is shown supposing each resource link has some fixed amount of underlay traffic. The matrix form underneath shows the player's cost given each strategy profile and the bold numbers are the system potential for the strategy profile as computed using the aforementioned potential function. A '*' symbol besides a cost indicates it is the player's best-response to the other player's particular strategy. It can be observed that indeed a Nash equilibrium between the two players corresponds to a system state in which the potential is minimal. Under this equilibrium state, resource t_5 is in an overloaded condition, yet no player is willing to play another strategy. This effect is in fact observed in simulation studies of overlay interaction [83]. The principle behind it is similar to the Braess paradox for network design: when a group of selfish users contend for a common resource pool, contention arises that inevitably leads to sub-optimal (in this case extremely harmful) system configurations. It is natural to avoid these system configurations since congestive resources lead to significant performance deterioration. (i.e. high delay and high packet loss rate). As we will show in Section 4.4, non-congestive configuration is also a pre-condition for the long term system stability.

Thus, we define a desirable equilibrium in the context of congestion game as:

DEFINITION 4.3.1: *Let $\Gamma = \langle N, \{Y_i\}_{i \in N}, \{u_i\}_{i \in N} \rangle$ be a game in strategic form with finite number of players over a finite collection of resources T=$\{t_1, \dots, t_m\}$, where l_j is the capacity of t_j. Let A be a strategy profile and A_i be the strategy of player i in A. A is a* ***desirable equilibrium*** *iff. the following conditions are true:*

$$u_i(A_i, A_{-i}) \leq u_i(A'_i, A_{-i}), \forall i \in N (\forall A'_i \in Y_i, A'_i \neq A_i)$$
$$\textit{and}$$
$$\frac{\#\{i \in N : t_j \in A_i\}}{l_j} \leq 1, \forall t_j \in T$$

This is a more strict form of the pure Nash equilibrium definition and therefore all desirable equilibrium are pure Nash equilibrium. As a weaker argument, we define desirable state as:

DEFINITION 4.3.2: *A strategy profile A is a **desirable state** of Γ iff. the following condition is true:*

$$\frac{\#\{i \in N : t_j \in A_i\}}{l_j} \leq 1, \forall t_j \in T \tag{4.1}$$

We will refer to Eq. 4.1 as the *desirability condition*. It is apparent that a desirable equilibrium is also a desirable state. In a nutshell, we are interested in exploring games with desirable equilibrium. One philosophy of achieving this is to consider cooperative game play in which players collaborate to achieve a desirable equilibrium. However, as Mahajan et. al. [80] rightly argue, such collaboration often gives rise to implementation difficulties. Issues such as information sharing, privacy, identity etc. are essential to cooperative game playing but difficult or expensive to realize in distributed setting. Furthermore, the players in our scenario do not have strong incentives to faithfully cooperate with each other as their objective functions are purely self-fulfilling rather than system agonistic. These concerns led to the exploration of another philosophy: to design a type of non-cooperative congestion game Γ_D that can guarantee a desirable equilibrium if there exist desirable states in the system. This property is satisfied by defining a cost function that heavily penalizes overloaded resources. We modify Γ_B to Γ_D with the following change to the cost function:

$$c_j(A) = (\frac{x_j(A)}{l_j})^{\lambda}, \lambda >> 1 \tag{4.2}$$

Γ_D is still an asymmetric unweighted congestion game. With an appropriate λ value, the cost function ensures an overloaded resource will have value far greater than 1 while any non-overloaded resource will have value far less that 1. Thus this cost function reflects the mentality of a congestion avoiding player by exhibiting a MINMAX property. A cost function modeled based on M/M/1 queuing is also considered, but without a tunable parameter such as *lambda* it is difficult guarantee the MINMAX property always hold. The exact MINMAX function as defined by Bottleneck congestion game [87] is also considered.

However as we will show, our particular form of objective function permits a Price of Anarchy of 1 rather than a polynomial PoA of the MINMAX.

LEMMA 4.3.1: *The congestion game Γ_D has an exact potential.*

Similar to Theorem 4.3.1, we can define $\phi = \sum_{j=1}^{T} \sum_{k=0}^{x_j(A)} (\frac{k}{l_j})^\lambda$ as the potential function.

□

THEOREM 4.3.2: Γ_D *has at least one desirable equilibrium if there exist desirable states in the system.*

Proof by contradiction. Assume this is not the case. Let the equilibrium be A^* and select an arbitrary desirable state A'. By our assumption, A^* contains at least one overloaded resource while A' does not. According to Lemma 4.3.1, the equilibrium state A^* corresponds to the minimization of the potential function, such that $\phi(A^*) \leq \phi(A)$, $\forall A \in Y$, $A \neq A^*$. Since A' contains no resource with load over the threshold, $\phi(A') = c$. where c is some small constant. Since A^* is not a desirable equilibrium and thus contains at least one resource with load exceeding the threshold, thus $\phi(A^*) >> c$. Therefore, $\phi(A^*) > \phi(A')$, $A^* \neq A'$. We arrive at a contradiction.

□

COROLLARY 4.3.2: Γ_D *has at least one desirable equilibrium if there exists desirable states in the system.*

The system potential $\phi = \sum_{j=1}^{T} \sum_{k=0}^{x_j(A)} (\frac{k}{l_j})^\lambda$ is strictly convex in the positive domain, thus permits a unique global minimal that is reachable through finite improvement [76]. By Theorem 4.3.2, it follows that this equilibrium is also desirable if there exists desirable states in the system. This desirable equilibrium is not guaranteed to be unique. For instance, consider two players with identical strategy set in a NE state. It follows that if the two players are to swap their strategy choices, we arrive at a new NE state. However, this set of NEs all exhibit the same unique global minimal.

□

COROLLARY 4.3.3: Γ_D *has Price of Anarchy of 1.*

This result directly follows from COROLLARY 4.3.2.

□

$\lambda = 8$

Player 1 \ Player 2	$\{t_1,t_2\}$	$\{t_3,t_4\}$	$\{t_5\}$
$\{t_1,t_2\}$	**2.05**	***0.44***	**1.21**
$\{t_3,t_4\}$	***0.44***	**2.05**	**1.21**
$\{t_5\}$	**1.21**	**1.21**	**10.00**

Figure 4.3: The potentials of 2-player example with $\lambda = 8$

Figure 4.3 shows the potential values of the two-player scenario depicted in Figure 4.2 in Γ_D game form. Compared with Γ_B, the system potential of Γ_D is minimal at a load balanced state of the system and avoids overloading resource t_5. The result is similar to that obtained through cooperative game playing.

It is important to set a proper value for λ in order to maintain the MINMAX property our game model desires. To this end, we first upper bound the value of λ in general with respect to the size of the network n and then show through game equivalence that this upper bound can be drastically reduced with respect to a player's path length p.

LEMMA 4.4.1: *lambda is lower bounded by* $O(\frac{\log(n)}{\log(1+\frac{1}{L})})$ *in the general case.*

The bound can be obtained by investigating the worst case. Consider a set of resources $\{l_1, l_2, ...l_t, ...l_n\}$. According to the MINMAX property, given a strategy profile A in which $C_t(A)$ is the bottleneck link, then $u(A)$ must be strictly higher than any other strategy profile of $u(A_i)$ where the bottleneck link is of the condition: $C_i < C_t, \forall i$. This implies that $C_t(A)$ must be the dominating term in $u(A)$: $C_t(A) \geqslant \sum C_i(A)$ where $i \in n$ and $i \neq t$.

The worst case occurs at boundary condition $C_i(A) = 1$, where $1_\lambda = 1$ regardless the value of λ. Thus we obtain the following worst case assuming l_t has the largest capacity L in the network and all of the other links are at capacity $l_i = 1$:

$$(\frac{L+1}{L})^\lambda > \sum C_i > n$$

It follows that unless the above equality holds, there exists an alternative strategy profile A' such that arbitrarily moving a player from a link in C_i to C_t in fact reduces the system potential, and thus the MINMAX property is not guaranteed.

Solving the above inequality yield the necessary lower bound on λ:

$$\lambda > \frac{\log(n)}{\log(1+\frac{1}{L})}$$

□

THEOREM 4.4.1: *lambda is lower bounded by* $O(\frac{\log(p)}{\log(1+\frac{1}{L})})$ *where* p *is the maximum path length.*

The above general case analysis assumes a player's strategy choice is the entire network. In practice, an overlay has limited number of paths with finite length. It is sufficient to establish the proof of Lemma 4.4.1 over the set of candidate paths an overlay is interested in. It follows that the analysis of Lemma 4.4.1 holds over maximum path length p.

Proof by contradiction. Assume Theorem 4.4.1 is not true. Then let $\frac{\log(p)}{\log(1+\frac{1}{L})} < \lambda < \frac{\log(n)}{\log(1+\frac{1}{L})}$, and take a boundary case where all of $C_i = 1$ and $C_t > C_i, \forall i$. Assign a player k from C_i to C_t will result in a reduction in system potential given the value of λ. However, this move will not occur since the player k's evaluation of $C_t(A_k)$ will necessarily yield a higher potential than any other alternative as long as $\frac{\log(p_k)}{\log(1+\frac{1}{L})} < \lambda$ where p_k is the player k's path length. Thus we arrive at a contradiction.

□

In effect, $\lambda > \frac{\log(p)}{\log(1+\frac{1}{L})}$ is sufficient to guarantee that system-wise MINMAX property always holds.

4.3.2 Convergence to desirable equilibrium

Bounding the convergence of Γ_D is challenging due to the nature of the cost function and the variability in resource capacities. More specifically, to bound the convergence span of a potential game, one need to bound the minimal potential drop due to a player's move, which is the difference between the benefit obtained from reducing congestion on some links and the cost incurred from adding congestion on some other links. With varied resource capacity and a polynomial cost function, this difference can be arbitrarily small. For weighted max-congestion games, it is known that the complexity of pure Nash equilibria is PLS-complete [93], and for Bottleneck congestion games, of which our game model is a more general form, the convergence bound is known to be exponential $O(2^{|U|^2}|E|^U)$ with respect to number of users $|U|$ and the number of links $|E|$ [87]. Thus, we seek an approximate solution. We first transform the game Γ_D into a binary factor form Γ_D^T which has the interesting property that the difference in potential due to a player move is always a multiple of a common factor, and obtain a polynomial time convergence bound of $O(L^\lambda Cn)$. Furthermore, we establish the isomorphic in desirability property between Γ_D and Γ_D^T which helps in computing the desirability ratio of this game transformation and to bound the price of anarchy.

We now define the transformed game Γ_D^T as a congestion game in strategic form identical to Γ_D except for a transformed resource collection and a transformed strategy set. Let $L = [\{l_j\}_{j\in T}]_{MAX}$, construct the binary factor set $B = \{2^0, 2^1, \ldots, 2^{\lfloor \log_2 L \rfloor}\}$. For each resource t_j in T, associate a resource set $t_j^T \subset B$ in Γ_D^T, such that $\sum_{k\in t_j^T} l_k = l_j$. Hence, the set of resources in Γ_D^T is a binary factoring of the resources in Γ_D. For each strategy A_i of player $i \in N$, associate to Y_i^T the set of strategies $\prod_{t_j\in A_i} t_j^T$. Thus, each A_i of i in Γ_D is expanded to a set over the binary factoring of the resources in A_i.

THEOREM 4.3.3: *convergence in the transformed game Γ_D^T is bounded by $O(L^\lambda Cn)$.*

Γ_D^T has identical game form as Γ_D and hence it has a pure Nash equilibrium induced by a system potential (Lemma 4.3.1). When a player makes a move to a new state, it must be that the new state has a lower cost than the old state. Since Γ_D^T has an exact system potential, by its exact potential property, a drop in a player's cost must cause an equal drop in system potential. We claim that in Γ_D^T, the smallest drop in potential when a player makes a move is $2^{-\lambda\lfloor \log_2 L \rfloor}$. Suppose player i makes a move, let A be the state before the move and A' be the state after the move, using the system potential definition from Theorem 4.3.1, the drop in potential can be expressed as $\Delta\phi = \sum_{j\in(A_i - A'_i)} (\frac{x_j(A)}{l_j})^\lambda - \sum_{k\in(A'_i - A_i)} (\frac{x_k(A)+1}{l_k})^\lambda$. Since all l_j are factors of 2, there must exist a sequence of constants $a_1, a_2, \ldots, a_m$ such that for all l_j, $a_j l_j = 2^{\lfloor \log_2 L \rfloor}$. It then follows that $\Delta\phi$ is,

$$
\begin{aligned}
&= \sum_{j\in(A_i - A'_i)} (\frac{a_j x_j(A)}{a_j l_j})^\lambda - \sum_{k\in(A'_i - A_i)} (\frac{a_k(x_k(A)+1)}{a_k l_k})^\lambda \\
&= \sum_{j\in(A_i - A'_i)} \frac{(a_j x_j(A))^\lambda}{2^{\lambda\lfloor \log_2 L \rfloor}} - \sum_{k\in(A'_i - A_i)} \frac{(a_k(x_k(A)+1))^\lambda}{2^{\lambda\lfloor \log_2 L \rfloor}}
\end{aligned}
$$

Since all the terms in the above equation have a common denominator, the result of the arithmetic operations is guaranteed to be some multiple of $2^{-\lambda\lfloor \log_2 L \rfloor}$. Therefore, the smallest potential drop from a move in Γ_α^T is bounded by $2^{-\lambda\lfloor \log_2 L \rfloor}$. Let ϕ_{max} and ϕ_{min} be the upper and lower bounds on the potential values respectively, and C be the upper bound on the cost of any player, then the maximum number of steps to convergence is bounded by:

$$
\frac{\phi_{max} - \phi_{min}}{2^{-\lambda\lfloor \log_2 L \rfloor}} \leq \frac{nC}{2^{-\lambda\lfloor \log_2 L \rfloor}} = O(L^\lambda C n)
$$

□

THEOREM 4.3.4: *A desirable equilibrium exists in* Γ_D *iff. a desirable equilibrium exists in* Γ_D^T.

Given a desirable equilibrium A^* in Γ_D, it must be the case that for all $t_j \in T$, $x_j(A^*) \leq l_j$. By the method of transformation from t_j to the factored set t_j^T, the total capacity does not change: $\sum_{k\in t_j^T} l_k = l_j$. Therefore, $x_j(A^*)$ can easily be distributed among the factored resources in t_j^T such that none of the resources exceed threshold. In fact, there's an optimal distribution under the best-reply dynamic. Therefore the transformed state A^{*T} is a desirable state in Γ_D^T. By Theorem 4.3.2, it is then clear that Γ_D^T must have a desirable equilibrium.

Given a desirable equilibrium A^{*T} in Γ_D^T, it must be the case that for all $t_k \in T$, $x_k(A^{*T}) \leq l_k$. With similar reasoning as above, one can assign $\sum_{k \in t_j^T} x_k(A^{*T})$ to the corresponding resource t_j in Γ_D and it is guaranteed that the number of users of resource t_j will not exceed threshold. Hence the transformed state A^* is a desirable state in Γ_D. By Theorem 4.3.2, Γ_D then must have a desirable equilibrium.

□

PROPOSITION 4.3.5: *For every desirable state in Γ_D there is a corresponding desirable state in Γ_D^T and vice versa.*

This is a stronger claim that follows from the proof of Theorem 4.3.4. Given a desirable state in Γ_D, by applying the transformation technique, one will obtain a desirable state in Γ_D^T, and vice versa.

□

To formalize our findings, we define the following property concerning Γ_D and Γ_D^T:

DEFINITION 4.3.3: *given a congestion game Γ and a congestion game Γ'. We say Γ and Γ' are* ***isomorphic in desirability*** *if the following two conditions are true:*

- *There exist desirable equilibria in Γ and Γ'.*
- *There exist transformation functions ζ and ζ', such that for every desirable state A in Γ, $\zeta(A)$ is a desirable state in Γ'; and for every desirable state A' in Γ', $\zeta'(A')$ is a desirable state in Γ.*

It is clear that Γ_D and Γ_D^T are isomorphic in desirability. This implies that an equilibrium state reached in Γ_D^T is a desirable state in Γ_D. In other words, a finite set of players participating in a game Γ, can in fact obtain convergence to stability by playing an isomorphic (hopefully simpler) game Γ', and still arrive at a desirable state in Γ. This is in effect a trade-off between the quality of the game equilibrium and the length of the convergence span, with the added benefit of a useful lower bound (i.e. the guarantee of a desirable equilibrium).

Since Γ_D^T is transformed from Γ_D and result in a different desirable equilibrium, it is important to analyze the loss of optimality due to this transformation. Price of anarchy evaluates the difference in optimality between a system-wide optimum typically obtained through centralized planning, and the goodness of a Nash equilibrium. Similarly, we define the *desirability ratio* as the difference in optimality between the equilibriums of two games that are isomorphic in desirability. Under our design goal, the optimal state of the system is a load-balanced state where the resource congestion levels are minimized. More specifically, the desirability ratio between Γ_D and Γ_D^T can be expressed as $\frac{\Gamma_D^T}{\Gamma_D}$.

THEOREM 4.3.5: *The desirability ratio between Γ_D and Γ_D^T is 2.*

The difference in the equilibriums of Γ_D and Γ_D^T is caused by the binary factoring of resources in Γ_D. Because of the min-max nature of the cost function, the minimal system potential (i.e. the game equilibrium) is in essence the minimization on the maximum resource congestion in the game space. Each strategy A_i of a player i in Γ_D is expanded into a set of strategies in Γ_D^T over the binary factoring of the resources in A_i. This expansion can potentially cause a rise in the system potential of Γ_D^T compared with Γ_D. To illustrate this effect, consider the following simple example: a resource t has capacity 5 and three players are using this resource, yield the congestion level $\frac{3}{5}$. in Γ_D^T, t is transformed into a resource set of $\{t_1, t_2\}$ with capacities 1 and 4 respectively. Allocating three players according to min-max property results in congestion levels $\frac{3}{4}$ and $\frac{0}{1}$. Thus one arrive at two different system potentials. To bound this loss of optimality at equilibrium, it is sufficient to bound the loss of optimality for all mappings of resources between Γ_D and Γ_D^T. Given any strategy profile A in Γ_D, we examine the transformation process of any resource t_j in Γ_D:

Case 1: $\frac{x_j(A)}{l_j} < \frac{1}{2}$. t_j is transformed into its binary factors in Γ_D^T. Mapping players of resource t_j to t_j^T in Γ_D^T follows the min-max property, and the resulting congestion on any member of t_j^T is upper bounded by $\frac{1}{2}$. Furthermore, given any congestion level of $t_j < \frac{1}{2}$, there exists a satisfying mapping among the transformed resources with congestion level at most $2\frac{x_j(A)}{l_j}$ on any of the binary factors. This is due to the binary nature of the factors, whenever mapping a player on a binary resource would exceed $2\frac{x_j(A)}{l_j}$, the player is mapped to the resource of the next binary grade. For example, a congestion level of $\frac{3}{7}$ in Γ_D is mapped to $\{\frac{2}{4}, \frac{1}{2}, \frac{0}{1}\}$ in Γ_D^T. This assertion always hold as long as $\frac{x_j(A)}{l_j} < \frac{1}{2}$.

Case 2: $1 \geq \frac{x_j(A)}{l_j} \geq \frac{1}{2}$. This is the simple case. By the desirability property (Theorem 3.3.4), mapping players of resource t_j to t_j^T in Γ_D^T is guaranteed to be bounded by 1 on any of the transformed resources. Thus the difference in congestion level of t_j^T in Γ_D^T compared to t_j in Γ_D is at most $2\frac{x_j(A)}{l_j}$.

The above analysis is conducted assuming there exists desirable states in the system. The lack of desirable states indicate a resource starved network and is outside the context of the study. Since the above cases are exhaustive, the desirability ratio is then: $\frac{\Gamma_D^T}{\Gamma_D} = 2$.

$\square$

The minimization of system potential in Γ_D naturally leads to an optimal configuration of balanced load in the system. Thus with the definition of price of anarchy given as:

$$\frac{\Gamma_D}{[(\frac{x_j(A)}{l_j})_{j \in T}]_{max}}$$

The price of anarchy for Γ_D is 1 by Corollary 4.3.3. It follows then the price of anarchy for Γ_D^T is 2.

In fact this game transformation technique is recursive. Γ_D^T could be further transformed by transforming $2^{\lfloor \log_2 L \rfloor}$ into two resources of $2^{\lfloor \log_2 L \rfloor - 1}$, and yield a convergence bound of $O((\frac{L}{2})^\lambda Cn)$ with price of anarchy 4. More generally, let α be the number of times this transformation is conducted, the convergence bound is $O((\frac{L}{2^{\alpha-1}})^\lambda Cn)$ with price of anarchy 2^α.

4.3.3 Implementation issues

A number of issues arise when implementing our game model due to the nature of congestion games in general. In this section, we first discuss how information sharing could be conducted in practice and then present techniques for realizing game moves in distributed environment.

Information sharing

Information sharing is a major obstacle in implementing game models [80]. Here we discuss the three necessary directions of information flow in our game: overlay-overlay exchange, overlay-domain exchange, domain-domain exchange.

- Overlay-overlay: overlays are represented by players in the game model. For a player to strategize a move, it must know the effects of other players' strategies on its own payoffs. This often requires overlay players identify themselves be able to exchange information about the decisions they have made in the past. Fortunately this process is greatly simplified in congestion games, since the collective influence of other player's strategies on a given player is represented by the congestion level of the resources a given player is interested in. Thus, a player only needs to obtain information about the resources rather than explicitly knowing the strategy choices of the other players.

- Overlay-domain: domains are represented by resources in the game model. Each domain provides a pre-determined capacity similar to how ISP services are offered today. In addition, the domains are only required to keep track of the aggregate congestion level of their service and report to the overlays interested in the service. No individual state keeping of each overlay player is needed.

- Domain-domain: since domains are represented as independent resources in the overlay game, there is no information exchange among the domains beyond adjacency reporting. In long overlay-underlay game, we will show that domains can operate independent of each other as well.

Simultaneous moves among overlays

Throughout our analysis, we have considered a system where only one player moves at a time. This is known as the Elementary Step System [74], and it is a necessary precondition for the system to guarantee convergence under the finite improvement property (FIP). Although there are special cases in single-commodity congestion games [90] that allow for simultaneous player moves and still converge in polynomial time, it is not the case in general. Consider the game presented in Figure 4.2, if the two players are allowed to move simultaneously under the best-reply dynamic, they may both choose the strategy set $\{t_1, t_2\}$ in round 1 and then both deterministically switch to the strategy set $\{t_3, t_4\}$ in round 2, and then back to strategy set $\{t_1, t_2\}$ in round 3, and so forth. Hence even with the existence of pure Nash equilibria, the system is not guaranteed to converge to an equilibrium. A more thorough study of this phenomenon is termed inadvertent synchronization and has been shown to be unavoidable [84].

Thus far, research on pure congestion games has used various methods for facilitating the sequential movement mechanism, such as random selection, round-robin, or highest improvement first. Here we consider an approach to enable partial simultaneous moves. Assuming each player's strategy set is a small subset of the common resource collection, let T_i be the set of resources used by player i's strategy set Y_i (i.e. $T_i = \{t_j : t_j \in A_i\}$). Then, define the *neighborhood* of player i as

$$NB_i = \{k : T_k \cap T_i \neq \varnothing\}_{k \in N}$$

We claim that if a player k is not in the neighborhood of player i or vice versa, then players i and k may move simultaneously in the system and the resulting potential change is as if they have moved in sequence, without loss of generality, say i moves then k moves.

THEOREM 4.3.5: *In Γ_D, if player i is not a neighbor of player k, then the potential change of players i and k moving simultaneously is equal to the total potential change of player i and player k moving in sequence.*

By Theorem 4.3.1, it suffices to study the change in player costs. Let A be the state before players i and k move, A' be the state after player i has moved but before k moves, and A'' be the state after both i and k have moved, the total change in potential in state A'' is:

$$\begin{aligned}&(u_i(A) - u_i(A')) + (u_k(A') - u_k(A''))\\ &= \left(\sum_{t_j \in A_i} (\frac{x_j(A)}{l_j})^\lambda - \sum_{t_j \in A'_i} (\frac{x_j(A')}{l_j})^\lambda\right)\\ &+ \left(\sum_{t_j \in A'_k} (\frac{x_j(A')}{l_j})^\lambda - \sum_{t_j \in A''_k} (\frac{x_j(A'')}{l_j})^\lambda\right)\end{aligned}$$

It is apparent that $A_k = A'_k$ and $A'_i = A''_i$. Since $T_i \cap T_k = \varnothing$, then $A^\Diamond_i - A^\Diamond_k = A^\Diamond_i$ and $A^\Diamond_k - A^\Diamond_i = A^\Diamond_k$ for all $A^\Diamond \in Y$. A move made by i would not have any impact on the load level of the resources used by k, and vice versa. The following equalities hold,

$$\begin{aligned}\textstyle\sum_{t_j \in A'_i} (\frac{x_j(A')}{l_j})^\lambda &= \textstyle\sum_{t_j \in A''_i} (\frac{x_j(A'')}{l_j})^\lambda\\ \textstyle\sum_{t_j \in A'_k} (\frac{x_j(A')}{l_j})^\lambda &= \textstyle\sum_{t_j \in A_k} (\frac{x_j(A)}{l_j})^\lambda\end{aligned}$$

Thus this is indeed the potential change for players i and k moving simultaneously.

□

Now consider a neighborhood graph $G = \langle N, E \rangle$. G is undirected and an edge exists between two players iff. they are in the neighborhood of each other. The nodes (i.e. players) in G can be partitioned into a number of sets, where in each set, there does not exist an edge connecting any two nodes in the set. It follows from Theorem 4.3.5 that all nodes in a set can move simultaneously. Therefore the minimum number of iterations it takes for all players to have a chance to move is equal to the minimum number of colors needed to color a graph. This is a straightforward reduction to the graph color problem. Since optimality is not strictly required in our context, any distributed graph coloring heuristic (e.g. local greedy strategy) can be used.

It is important to note that if a resource is present in all of a player's candidate strategy, it should not be considered in the neighborhood graph construction of that player. This is because no matter what move a player makes, it bears no influence on the congestion level of this common resource. This is particularly relevant to multi-domain networks, where the domain organization is mostly hierarchically. At the top level the number of transit domains are rather limited, and thus it is likely that a few of them (e.g. transcontinental links) are used by all the candidate routes of an overlay player.

Sequential moves in distributed environment

Although we have discussed a technique for enabling partial simultaneous moves, there still needs to be a mechanism for conducting sequential moves in distributed setting. This mechanism should be simple, robust and efficient (i.e. low communication overhead), without the need for global synchronization. A simple token ring is suitable for this purpose. Token ring was established as an effective mechanism to enable distributed computing since the 1960s [100]. Figure 4.4 illustrates its general structure. The ring is formed among the overlay players. Each player only needs to be aware of its two-hop neighbors along the ring (e.g. node A in Figure 4.4). A token is passed along the ring and the player holding the token may execute a move and then pass the token along to its clock-wise neighbor. Four types of messages are necessary:

- Token_Pass: this message pass the token to a player's clock-wise neighbor after the player has used the token to conduct a move.
- Move_Complete: this message is sent to a player's counterclock-wise neighbor after the player has sent Token_Pass message.

- Keep_Alive: this message is sent periodically by each player to its immediate neighbors. The time length is set to be slightly larger than the operation time of a player move.
- NB_Update: this message is sent to update a player's neighborhood when a node is added/deleted from the ring.

Thus the token is passed along the ring and allows sequential moves on a round-robin schedule. If a player receiving the token does not want to make a move, it simply passes the token to the clock-wise neighbor.

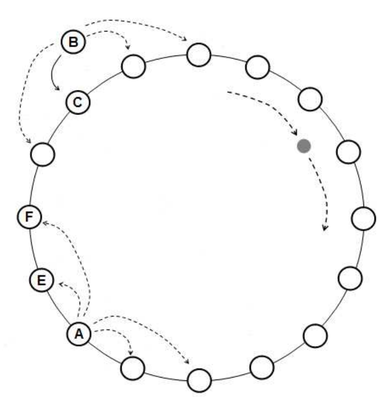


Figure 4.4: An Example Token Ring

Add operation is invoked when an overlay wants to join the ring (e.g. node B in Figure 4.4). It only needs to contact any node already in the ring (e.g. node C) and construct its neighborhood list accordingly. Then a NB_Update message is sent to all of its neighbors.

Delete operation is invoked when either an overlay terminates or exits due to fault. Normal termination can be handled by the neighbors easily, but fault cases are more troublesome. The distributed token ring we have described here can handle non-cascade failures of overlays (i.e. nodes consecutive along the ring does not fail simultaneously). Although cascade failures are common among network links, since the token ring neighborhood is virtual and arbitrarily formed, the occurrence of cascade failures should not be high. The non-cascade node failures falls under two categories:

- During token operation: a player (e.g. E in Figure 4.4) can fail while holding the token or after it has sent Token_Pass message but not Move_Complete message. Both of these cases are detected by the player's counterclock-wise neighbor (e.g. A) since a Move_Complete message is not sent before the next Keep_Alive probe. A will contact F to notify about E's failure. If E has not sent Token_Pass message to F, then the token is lost and a new token is generated. The neighborhoods of E's neighbors are updated by A and F sending NB_Update messages.

- Outside token operation: a player (e.g. E) can fail while not participating in a token operation. This is detected either by Keep_Alive message of E's neighbors or when F tries to send E Move_Complete message. In the prior case, A and F simply update the token ring by sending NB_Update message. In the latter case, F sends Move_Complete message to A and then the NB_Update messages are sent.

Overall, the number of messages sent to maintain and operate the token ring are small and the communications are exclusively among the two-hop neighbors of each overlay, without the need for global synchronization. If partial simultaneous move technique is enabled, then one token ring is required per color group.

4.4 Interaction between overlays and underlays

In this section, we model the long term interaction between overlays and underlays. It is a game played between an aggregate overlay player and an aggregate underlay player. By abstracting the underlay domains as virtual resource links and map the overlay and underlay traffic onto these links, we show that the resulting scenario can be modeled as a two-player staged congestion game and the congestion avoidance objectives of the overlays (shown in Section 4.3) agree with the resource optimization objective of the underlays and hence lead to a stable and optimal solution of the game. We discuss the pre-conditions of this game and argue that they are achievable in practice. Finally we show domain-wise management actions that arbitrarily limit overlay traffic (e.g. traffic throttling) are harmful to the performance and stability of the global network.

4.4.1 Interaction between overlays and underlays as a congestion game

We consider that domain-wise management actions are carried out periodically by underlay providers to perform independent intra-domain route optimization with the objective of minimizing the maximum resource usage (i.e. load balancing) of both the overlay and underlay traffic. We also consider that underlay aggregate traffic in a domain is stable in semi-long term (e.g. weekly). However, changes in overlay traffic distribution inevitably cause network management actions to take place.

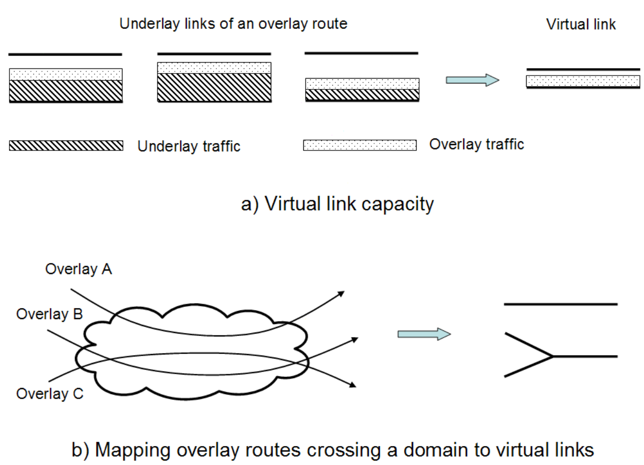


Figure 4.5: Underlay routing of overlay routes is mapped to virtual links

Conversely, the intra-domain routing decisions of the underlay determine the underlay links that are used to realize the overlay routes. In examining the traffic distribution of the overlays in a underlay domain, two general cases arise: a) the overlays do not share any underlay links; b) the overlays do share some underlay links. In the case of a), the traffic of the overlays do not affect each other and hence are mapped to disjoint virtual links; in the case of b), the overlays are mapped to a disjoint virtual link and a shared virtual link. Figure 4.5 illustrates the mapping. Figure 4.5a shows the underlay links of an overlay route are mapped to a virtual link where the bottleneck overlay capacity is considered as the virtual link capacity. Figure 4.5b shows the mapping of multiple overlay routes in a domain to virtual links. The shared underlay links of Overlay B and C are mapped to a shared virtual link and the remaining disjoint underlay links are mapped to a disjoint virtual link. The result of this mapping produces a graph of virtual links on which each strategy of the overlays is realized by an adjoining sequence of virtual links. For such a mapping to be valid end-to-end across multiple domains, it is important for the networks to be: in a congestion-free configuration, such that the workload of an overlay is experienced by all virtual links of its route; and the overlay must have control over the network domains it traverses. The former condition is satisfied in Γ_D and the latter condition, as we will discuss shortly, can be obtained by employing some of the emerging network architecture proposals (e.g. [18] and [17]). We also note that when an overlay route changes, it generally induces a corresponding change in its underlay route mapping, resulting in a new set of virtual links. This in turn affects the routing of the overlays. Hence their interaction affects the network stability.

Let Γ_L be a two player game where Player 1 is the aggregate overlay player and Player 2 is the aggregate underlay player. In this game, Player 1 always moves first then followed by Player 2. The strategy set of Player 1 consists of the set of all possible overlay traffic distributions Y. The strategy set of Player 2 consists of all possible mappings of $H(Y,Z): Y,Z \rightarrow T$ where $T = \{t_i\}$ is the set of virtual links and Z is the underlay traffic matrix. The capacity of each virtual link l_j is defined by the bottleneck capacity as presented in Figure 4.5a. Let L_j be the capacity of a underlay link and Z_j be the volume of underlay traffic routed through this underlay link, then the cost function is defined as:

$$u(A, Z) = \sum_{j \in A} c_j(A, Z)$$
$$c_j(A, Z) = (\frac{x_j(A)}{l_j})^\lambda, \lambda >> 1$$
$$l_j = MIN\{L_j - Z_j\}$$
$$x_j(A) = \#\{i \in N : t_j \in A_i\}$$

The cost function is similar to the overlay game except that the cost is defined over all virtual links in use rather than the links used by an individual overlay player, and the virtual link capacity is defined based on the mapping function $H(Y, Z)$. Hence the objectives of Player 1 and Player 2 are aligned such that they both minimize the same cost function (through the mapping of $H(Y, Z)$), with Player 1 having control over A and Player 2 having control over T and l. In other words, where as the strategies of an overlay player depicts a collection of virtual resources the overlay route uses, the strategies of the underlay player depicts the collection of physical resources it employs to support overlay traffic in its domain. Since there exists an exact mapping of the two resources (i.e. $H(Y, Z)$), a consistent and common cost function is obtainable. Now we show that the collective behaviors of the individual overlays can be represented by an aggregate overlay player and the collective behaviors of the individual underlay domain provider can be represented by an aggregate underlay player. Furthermore, we show that Γ_L is a potential game with $u(A, Z)$ being the system potential.

LEMMA 4.4.1: Each move by the aggregate overlay player results in a drop in $u(A, Z)$.

The role of the aggregate overlay player is taken on by the short-term overlay game as a whole. Thus, a move in the long-term game is in fact the equilibrium configuration of the short-term overlay game. According to Lemma 4.3.1, the convergence of the short term overlay game necessitates a decrease in $u(A, Z)$ given fixed T and Z. In fact, when the minimum potential of the short-term overlay game is obtained, it is a best response move by the aggregate overlay player.

□

LEMMA 4.4.2: Each move by the aggregate underlay player results in a drop in $u(A, Z)$.

The role of the aggregate underlay player is the natural effect of the distributed intra-domain optimization of each underlay domain. Recall that we consider the underlay traffic of a domain to be stable in semi-long term, then the traffic matrix of each underlay domain is stable if one can guarantee that the overlay traffic routing is stable. This leads to the pre-condition that short-term overlay game must converge before a staged move in the long-term game occurs. With this pre-condition satisfied, there exists an optimal $H(Y, Z)$ that minimizes $u(A, Z)$. Since the inter-domain traversal of an overlay is not controlled by the underlay (i.e. underlay network management action cannot change the traffic demand of each domain) and the overlay configurations are desirable (i.e. each domain an overlay player uses experience the same workload from that overlay player), this optimal $H(Y, Z)$ global mapping is achievable through the optimal $H(Y, Z)$ mapping in each domain. This is a best response move by the aggregate underlay player.

□

THEOREM 4.4.1: Γ_L has a pure Nash equilibrium.

From Lemma 4.4.1 and Lemma 4.4.2, it follows that Γ_L is a potential game and thus the minimization of the potential $u(A, Z)$ brings the game to an equilibrium under the finite improvement property.

□

THEOREM 4.4.2: Γ_L has a minimal potential equilibrium when the overlay game minimizes its potential and the network management solutions are domain-wise optimal.

As discussed in Lemma 4.4.1, since the overlay game minimizes its potential (result of Corollary 4.3.3), the aggregate overlay player effectively conducts a best response move. Similarly in Lemma 4.4.2, when each network domain minimizes its congestion, it corresponds to a best response move by the aggregate underlay player. Thus, in a two player potential game, following the best response moves naturally lead to a game solution that minimizes the potential $u(A, Z)$ [76]. This is the optimal system configuration that minimizes the congestion level of the networks.

$\square$

For the remainder of this section, we will discuss the practical implications of the preconditions that must be satisfied for the long-term overlay-underlay game to be stable and show through a simple example why popular techniques such as throttling is extremely harmful to the stability of the network in the long term.

4.4.2 Establishing the pre-conditions

The three pre-conditions that must be satisfied are: i) the objectives of the overlays and the underlays must be aligned; ii) the domain traversal of an overlay route must be controlled by the overlay; and iii) the short-term overlay game must stabilize before a move is made in the long-term overlay underlay game.

Pre-condition i) has partly motivated me in examining overlay interactions from a congestion-avoidance point of view. As we have discussed in above sections, delay minimization may not be a strict requirement in practice because the supported applications general have a upper delay bound that need to be satisfied. Furthermore, congestion is the leading cause of delay in networks. These observations naturally lead to the use of congestion game as a common basis of objective alignment between the overlays and the underlays. We have thus devised a min-max objective function and defined the concept of desirability to ensure a congestion-free system state. The properties exhibited by our overlay game are aligned with the practical concerns of current network designs.

Pre-condition ii) is motivated by the observation that system stability could not be guaranteed when underlay domain providers independently decide on how inter-domain routing should occur, without explicit knowledge of the routing policy and network infrastructure of the other domains, and without consensus among the domains. Most of the inter-domain routing issues today (e.g. route inconsistency, sub-optimal performance, routing conflict, etc.) can be attributed to this observation. As a direct result, new network architecture proposals has emerged in the past years (e.g. [18] and [17]) and advocate for the underlays to relinquish control over inter-domain routing. However, that is not to say that underlay providers should have no involvement in inter-domain routing policies. First, each underlay domain can have jurisdiction over what domains it is willing to exchange traffic with, just not control exactly what domains an overlay route must traverse. Secondly, each underlay domain is not required to disclose how intra-domain network operations are conducted, but only need to report its performance aggregate (e.g. congestion level in our context). Thus pre-condition ii) can be made feasible in practice. As discussed in Chapter 3, our service quality assurance framework advocates a similar philosophy.

Pre-condition iii) relates to the convergence of overlay games. This is in fact a major motivator of the overlay game design as we have presented in this chapter. Not only it is important to bound the convergence span of the game model from a theoretical perspective, as we have shown through game transformation and approximation, but also it is important to consider implementation techniques, such as partial simultaneous moves. Another aspect of this direction of exploration is the definition of desirability. As we have shown in the long term game analysis, it is fundamental for the overlay interactions to arrive at a desirable state (i.e. congestion-free) and convergence to such a state can be accomplished in much faster time than convergence to system optimal. We have derived an approximation bound on this convergence through the computation of the desirability ratio.

4.4.3 Limiting overlay traffic

As we have discussed in the previous sections, given the right conditions, the interaction of overlays and underlays can lead to stable system configurations over time. However, it presents an environment in which underlay providers have no control over the amount of overlay traffic a domain carries. The underlay providers may attempt to limit the overlay traffic through techniques such as throttling, a popular mechanism today to limit P2P network activities. Through a simple example, we show the side effect of throttling on global stability and performance.

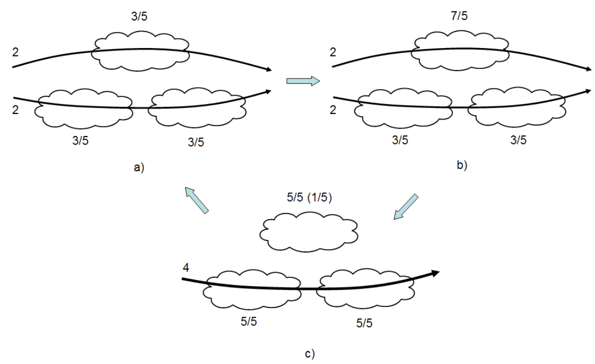


Figure 4.6: An example of throttling overlay traffic in a domain

Figure 4.6a depicts an overlay distributing its traffic evenly on two possible paths (a working and congestion free configuration). The domain provider (top domain in the figure) throttles the overlay traffic, artificially inflating its congestion level (Figure 4.6b). The overlay responds by shifting its traffic away onto the other path (Figure 4.6c). Thus, a sub-optimal traffic distribution arises where two domains are bearing the burden of excess traffic pushed out by the throttling domain. This in turn causes a cascade of overlay reroutes and corresponding underlay management actions in the other domains. Furthermore, there is no longer a guarantee of the overlays converging to a desirable state since the congestion potentials in the overlay games are no longer accurate. If the throttling is only employed as a temporary measure, then when it deactivates, the resulting downshift in congestion level causes a state transition back to 4.6a from 4.6c, resulting in perpetual oscillation. Therefore, it is important for the underlays and overlays to form a mutual agreement over how overlay traffic should be routed in the networks as unilateral actions can easily result in global destabilization.

4.5 Simulation Studies

To study the interaction of overlays and underlays in multi-domain networks, we construct a connected network consisting of 210 domains. 100 domains form a randomly connected graph called the *source region*, and 100 domains form the *destination region*. The two regions are connected to a *transit region* of 10 domains to form a two-tier multi-domain network. 50 pairs of source-destination are randomly selected to represent overlays with the source nodes linked to domains in the source region and the destination nodes linked to domains in the destination region. Thus, each overlay route has a minimum length of 3 domains. Each overlay will generate 4 units of traffic that can be routed independently (i.e. 200 overlay players in total) over four disjoint shortest paths. Initially one unit of traffic is assigned on each path. Because all the overlays must travel through the transit region which has limited number of domains, sufficient interaction among the overlays are ensured and we can control the overall congestion level of the system by modifying the capacities of the transit domains. Each domain is modeled with a simple underlay topology as shown in Figure 4.7, with four possible paths through the domain. Although the topology is simplistic, it allows for a large number of different underlay-overlay distribution strategies to be simulated and is still small enough to be computation light (considering that hundreds of such domains are simulated in the study). The domain is randomly assigned a fixed underlay traffic of 8 to 12 units. A domain planner computes the optimal distribution of traffic (overlay and underlay) in the domains and creates the virtual links for the overlay game. The underlay link capacities are shown in Figure 4.7. Links in transit domains have higher capacity ($< 10, 20, 10 >$).

Two sets of simulation studies are performed. In the first set, we study the interaction of the overlays when the underlay paths do not change. In the second set, we study the interaction of the overlays and the underlays. 100 simulation runs are conducted for each study. Time is broken into discrete steps, where a *step* is the time it takes for an overlay player to change its route. A *round* is the time it takes for all players to make a move if required.

Figure 4.8 shows the total number of steps it takes for the overlays to stabilize. We

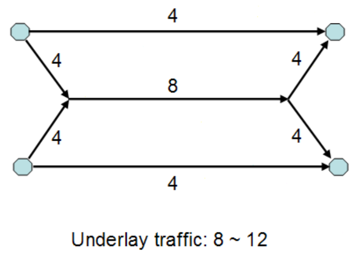


Figure 4.7: Simulation setup for intra-domain underlay topology

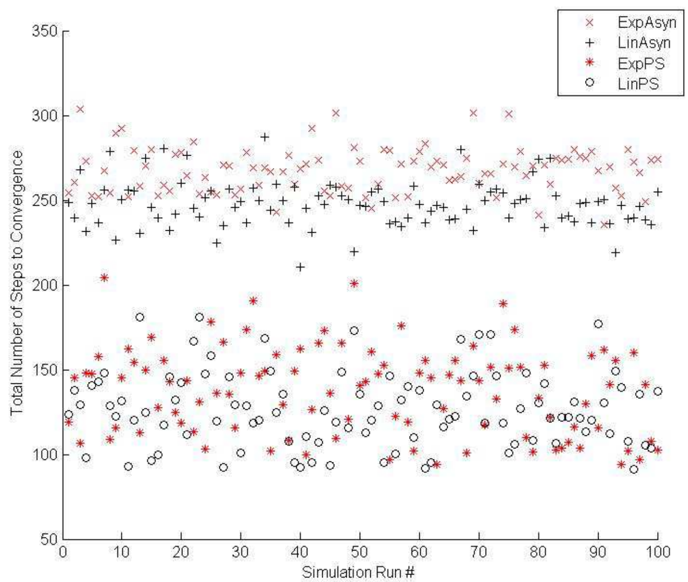


Figure 4.8: Overlays convergence time

study both the exponential cost function (Exp) of our work and the traditional linear cost function (Lin). Furthermore, we want to see the effect of allowing partial simultaneous moves (PS) compared with asynchronous moves (Asyn). For each run, a set of random overlay and underlay traffic is generated and these varied techniques are applied on the same input. Observe that exponential cost function results in slower convergence (higher

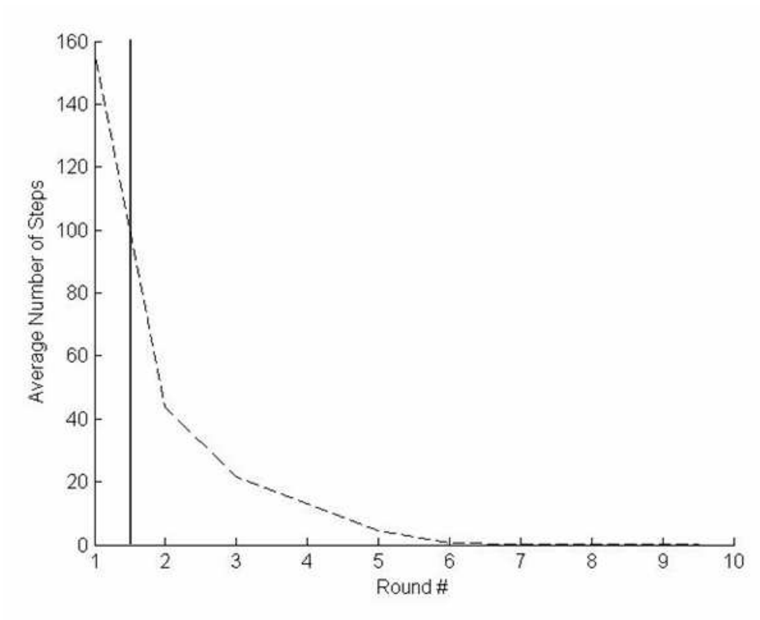


Figure 4.9: Number of rounds to convergence

total number of steps) than linear cost function, but in general the increase is not as significant as the theoretical worst case. Partial simultaneous moves achieve faster convergence but the percentage of improvement varies greatly depending on the overlay topologies as expected. In fact, the total number of player moves in PS is the same as the asynchronous case under the same cost function. The difference is that a step in PS allows multiple overlays to move at the same time while a step in asynchronous case only allows one overlay to move.

The length of each round varies depending on the number of steps in the round. Therefore, as the overlay game converges to equilibrium, the number of steps in a round should also diminish. This is shown in Figure 4.9. The average number of steps (over 100 runs) in each round is plotted on the graph and the overlays stabilize within 10 rounds. The vertical line on the graph shows that the average number of rounds for the system to reach a desirable state is only 1.5 round, which is much smaller than the game's natural convergence span. This is encouraging since it is possible to bring the game to a stable desirable state using *isomorphic game transformation* to achieve much faster convergence, at the expense of optimality.

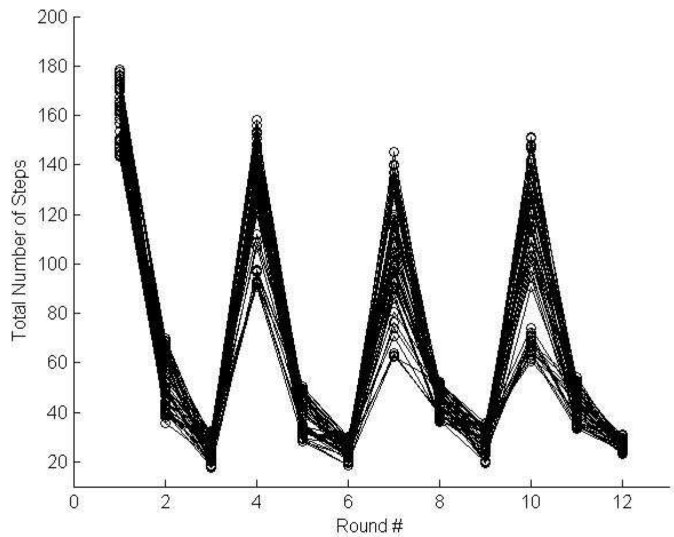


Figure 4.10: Overlays and underlays interaction in short cycles

Figure 4.10 and Figure 4.11 show our simulation results in studying the long term overlay underlay interaction. The underlays perform domain-wise optimization at fixed time intervals and the overlays attempt to optimize their routes in between these intervals. Again, 100 runs of randomized traffic setting are used. We only plot the results of the first 50 runs here so as not to obscure the graphs. In Figure 4.10, the underlay management actions are performed in intervals of 3 rounds which is not sufficient for the overlays to stabilize. The resulting oscillation of the overlays is evident. Also observe a spike in overlay re-routing activities immediately following an underlay optimization. The simulation result as depicted in Figure 4.11 shows the result of a 10-round underlay management interval. As the overlays are allowed to reach equilibrium before each underlay management activities, a clear convergence is observed over the long term among overlays and underlays.

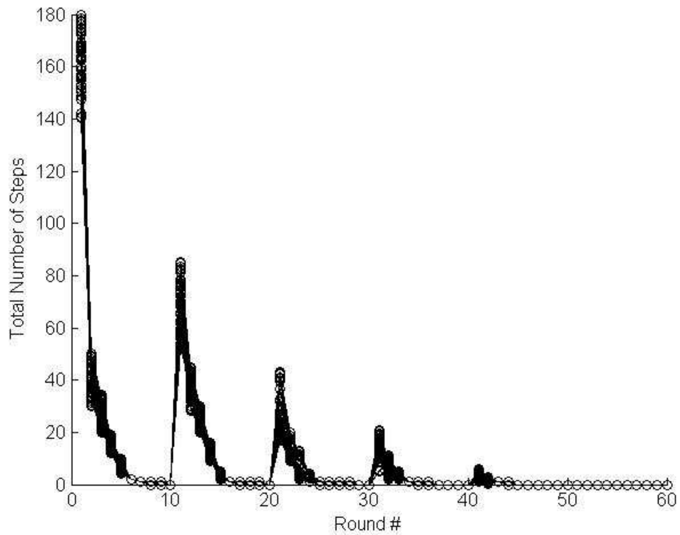


Figure 4.11: Overlays and underlays interaction in long cycles

The graph illustrates pictorially the dual game nature of overlay and underlay interactions.

Finally, we study the level of underlay link congestion when the overlay equilibrium has been reached. Figure 4.12 shows that our cost function results in desirable system configurations and in general the maximum link congestion rate is much lower than that of the linear cost function. The results appear to be layered due to the discrete workload and link capacity assignment in the simulation setup.

4.6 Conclusion

In this chapter, a formal analysis of the overlay-underlay interaction problem under the congestion game model is presented. Although congestion games have been used to study this type of network problems before, as we have shown, the studies performed in the past over-simplify the problem such that the results obtained are not sufficient to guide network

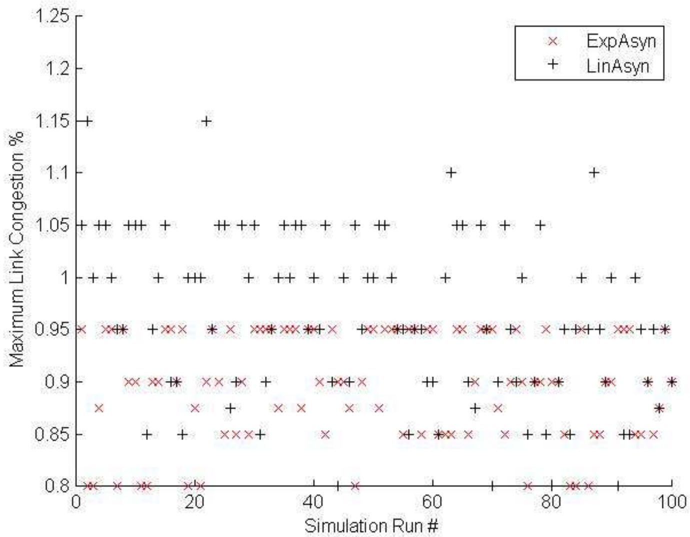


Figure 4.12: Maximum link congestion level

and service designs in practice.

Furthermore, desirability has been proposed as a network operation's concept and a congestion game model is formulated accordingly. For interacting overlays, we prove the existence of pure NE, its convergence, and price of anarchy. By modeling the overlay-underlay interaction game as a two-stage congestion game, pre-conditions are obtained which can serve as the basic design and management guidelines in service-driven networks. The simulation results confirm our theoretical findings. In addition, some implementation issues are addressed.

Chapter 5

The Meaning and Implication of Service Quality

5.1 Introduction

The prevalence of networked applications in business operations and daily lives creates enormous earning potential for network service providers (SPs) and at the same time, fierce competition. Facing a consumer market with rising demands for quality and impending saturation, SPs are struggling to keep their customers satisfied and their businesses profitable. In essence, the SPs must execute strategies that maximize service profitability. The process of network planning and upgrades are essential facilitators of this objective.

In the network service industry, network planning and upgrades are regularly exercised. The practice is mostly ad hoc, where investment decisions are made based on past experiences and "rule of thumb" estimations. The lack of formal methodology can be attributed to the large process gap between the network planners and the business analysts. From the network point of view, the network planners strive to improve network performance via fine tuning and optimizing upgrade decisions. Very little concern is given to the profitability of the resulting investments. From the business point of view, the business analysts have a very coarse understanding of how improved network performance can lead to future revenue generation. Better network performance directly translates to more profit is a common assumption. Considering the intricate relations among network operations, customer behaviors, and market dynamics that jointly influence service profitability, such an assumption is overly naive. A general and comprehensive analytical model linking these three factors to service profitability is then extremely beneficial and timely. In surveying literatures, we find that very little is done in studying the causes of customer behavior and its effects on network service profitability. Research in network planning and upgrades often assumes direct and simple relationship between improved network performance and revenue. As the customers are the central source of revenue in the network service industry, they should be a key focus of analysis when investments in network infrastructures are made. Research in market science and economics presents many insightful observations and empirical studies on service utility, customer behavior, and profitability, but remains descriptive and incomplete. This lack of formalization prevents the integration of key customer and market factors in network planning and upgrade analysis and produces ineffective network upgrade decisions that do not reflect customer behaviors and service dynamics, and do not give good service profitability estimates. In service-driven networking context, this "missing mile" among network, service and customer metrics lead to the inability to formulate implementable service logic that fits the customer's requirements and produces ineffective service planning and operational strategies.

In this chapter, we establish an analytical modelling approach relating the performance delivered by a network service infrastructure to the satisfaction of its customers and consequently to the network service provider's profit. We show that network upgrade and planning strategies should be made in accordance to their influence on customer satisfaction and the resulting changes in customer behavior. The ability of a service provider to retain and grow its customer population over time has an immediate impact on service revenue and is an invaluable indicator for network upgrade and planning operations. Based on a number of influential theories in economics and market science, we show that there is a strong ground for the derivation of well-behaved mathematical models linking the network service performance, the customer behavior and the market dynamics to profit. Based on this approach, we construct a generalized model, with meaningful parameters to reflect varieties of network service characteristics, customer attributes, and market conditions. Through analysis and simulation, we demonstrate how this model can capture SP service trends and customer behaviors, and its application to network upgrade decision processes. As our analysis shows, the effectiveness of a network upgrade and planning strategy is highly dependent on the customers' access behaviors, QoS sensitivities, service expectations, past experiences, service competitiveness and market growth trends. For the service providers, ensuring that service quality meets customers' expectation is of paramount importance, and service differentiation may improve customer retention rate resulting in better revenue generation even without additional customer charges. Although the benefit of our approach demonstrated in ISP network planning upgrade scenarios, the model is general and equally important to many aspects of service-driven networking such as service demand forecasting, network and service infrastructure analysis, and others.

The rest of this chapter is organized as follows: Section 5.2 presents a summary of current industry practices and academic research. Section 5.3 presents our modelling approach and its rationale, while Section 5.4 details the construction of a forecasting model following our approach. Section 5.5 analyzes the forms of our perception function and the impact of model parameters, followed by case studies and simulations in Section 5.6.

5.2 Prospectives and Literature Works

In conducting formal analysis of investment decisions, it is well understood that the soundness of a decision is dependent on the soundness of the analytical model and the value of the analyzed data. In the context of network services, one is presented with a rich reservoir of network information, ranging from statistical information gathered from Management Information Base [22] (e.g. via SNMP [101]) to active end-to-end measurements (e.g. pinging). Advanced tools, such as Cisco NetFlow [102], are even capable of tracking individual traffic flows. Traditional customer management processes (e.g. customer relation management) gather vast amount of customer information in the form of customer surveys, service usage, trouble-ticket logs, etc.

For a network service provider, its customers, the sole source of revenue, are the crucial link between network performance and service profitability. Hence the willingness of the customers to repurchase services should be the focus of analysis. In the context of network service operations, the satisfaction of a customer is strongly influenced by the service performance he/she receives from the underlying network infrastructure. It is then apparent that correlating network performance and customer information in an analytical process can provide crucial guidance to the effect of network improvements on customer satisfaction, which ultimately influences the customer's intention to repurchase. A number of market studies on Telecom service operators world-wide have confirmed the existence of these relationships [103][104][105].

Some works in network research [106][107] have recognized the importance of analyzing both the customer profile and the network information in a business decision process. However, the means of correlating the two aspects are missing [106] and there is no method for mapping network performance to service utility [107]. Using real option pricing, d'Halluin et al. [108] present a method for determining best investment time for link capacity upgrades. Their work evaluates profitability as a function of network usage, where customer dissatisfaction is modelled with a simple discount factor. Similarly, Jagannathan et al. [109] propose a revenue-based optimization for network upgrades. The profitability of a component is assessed based on the amount of customer traffic it supports, assuming previously unsatisfied customers are satisfied after upgrades. In these works, little effort is made in modelling the actual customer behaviors induced by their perceived service utilities, or on the subsequent shifting of consumer market dynamics. Bouch et al. [110] have studied the user's perception on quality of service, in particular regarding audio and video perceptions. They have devised studies under which users' cognitive and perceptual processes regarding network QoS can be obtained. In our past work [111], we have presented a customer-centric framework for network upgrade optimization. The work projects network QoS performance onto customer satisfaction and then linearly maps to future revenue. Some attention is paid to market dynamics in terms of new market growth. Although the framework is sound, the relationships are overly simplistic to capture the complexities of customer behavior and market competitions.

In the area of network charging and pricing, service profit maximization is often the aim of investigation based on which various charging schemes are proposed and analyzed. Mitra et al. [112] consider pricing and routing as a joint optimization problem in multi-service networks where revenue maximization could be achieved by not only charging traffic based on usage but also routing traffic through low cost routes. Works on usage based charging, such as [113][114], conduct service charging based on the volume of customer traffic and access time. The pricing schemes may be variable such that a customer is charged based on fluctuating demand. Shakkottai and Srikant study the effect of multi-ISP competitions on service pricing [115]. Modelling the competitors as a non-cooperative game, they are able to draw insightful conclusions about the pricing strategy in both local and transit ISP markets. Pricing is an important factor of service profit because it maximizes the monetary benefit a service provider can draw from its customers. Our work investigates another important factor of profit: customer population. By studying the cause and effect of customer satisfaction, we bring focus and structure to some of the key factors influencing customer retention and growth.

Our work is also complementary to works on bandwidth provisioning and network dimensioning. The goal of bandwidth provisioning and network dimensioning research is to find the optimal resource allocation that maximizes network performance and profit. Our model can aid in this maximization process by providing the mathematically means to evaluate the various methods of resource allocation in terms of their impact on customer satisfaction and hence the resulting customer population. The model is generic and applicable to provisioning problems in other network infrastructures as well. For example, Duan et. al.[12] consider the bandwidth provisioning and SLA assurance problem in Service Overlay Networks (SONs). Their analysis is conducted on customer flows with respect to specific QoS bounds. Our computation of QoS sensitive service utility takes a similar approach. However, in our utility computation, we also consider network fluctuations as a cause of customer dissatisfaction. Their work evaluates revenue generation as a function of link traffic volume, access time and levels of QoS. They consider variable traffic demand in daily cycles similar to what we have constructed in our simulation setup. Rather than computing revenue directly from network performance and usage as in their work, our model evaluates the impact of service performance in terms of customer satisfaction which influences the customer's intention to repurchase the service. This could provide an alternative evaluator for their algorithm in determining the best bandwidth provisioning plan in SONs.

Customer relations and profitability have been the subject of significant research in the fields of market science and economics. The well known expectancy-disconfirmation theory [116][117] relates service utility to customer satisfaction, based on the classic adaptation theory from psychology. The work views expectation as an adapted reference point for the customers, upon which satisfaction is the result of customer value judgement on expectation and perception. Later finding [118] suggests strong relationships among satisfaction, perceived quality, and disconformation. Anderson and Sullivan [119] follow up on these works with a descriptive model relating service quality to customer repurchase intention. However, their work remains qualitative and does not address the issue of expectation adjustment and market dynamics. Bolton [120] proposes a dynamic model for the duration of provider-customer relationships in continuous services. Through an iterative expectation update process, the work formalizes the influence of customer satisfaction on customer retention and increased sales volume. The model is refined and tested over a 22-month period with cellular customers. The linkage between expectancy and customer retention is coarsely treated in this work and the impact of divergent service quality on customer experience is not considered. To obtain service utility, SERVQUAL [121][122][123] is the most-used and proven model in market science. It categorizes service utility into five aspects: tangibles, empathy, assurance, responsiveness and reliability. The model is focused primarily on service industries and relies on consumer survey based data collection. In a recent study [104], the SERVQUAL model has been shown to capture customer's quality perception of China's Telecommunication services. Also in this work, network quality is found to be an additional SERVQUAL aspect for network services.

With the vast amount of existing conceptual and empirical results from market science and economics, we believe there is a strong foundation for deriving an analytical forecasting model for network service operations. We propose a methodology for formalizing the relationships between network performance, customer satisfaction, and service profitability. The approach covers the computation of utility for network services, the derivation of customer satisfaction based on service utility, and the projection on service profitability from customer repurchase intentions and market dynamics. Following this approach, we construct a network service specific forecasting model capable of forecasting service profitability induced by network infrastructure improvements. In the context of this paper, we make the following assumptions: the network service market is an open market with multiple competitors; The customer is rational in his/her purchase decisions and does not exit the service market; All competitors of the network service market charge similar price, have identical technology attractiveness from the customer's perspective and employ similar advertisement strategies; the pricing for the service is flat rate subscription based. The computation of our network utility functions is theoretical; It relies on end-to-end QoS measurements of the customers and knowing the access behavior (traffic flow and access time) and the QoS requirements of the customer. Our prior work [111] addresses the issue of how such measurement could be conducted and computed in practice.

5.3 A Market Science Methodology

In this section, we show how network performance, customer satisfaction and service profitability are related in market science research, and present our modelling methodology. A key driver of our approach is the well-established expectancy-disconfirmation theory [116][117] which relates expectation, perceived quality and disconfirmation to customer satisfaction. The perceived quality refers to the service utility a customer obtains from service usage, while expectation represents the expected utility a customer formulates before using the service. Disconfirmation is then the discrepancy between the expectation and the perceived quality. Anderson and Sullivan [119] refined this theory in a customer satisfaction framework (Figure 5.1).

They consider disconfirmation to have a positive and a negative component that are influenced by expectation and perceived quality. The customer satisfaction is then a function of perceived service quality and both components of disconfirmation. The perceived quality

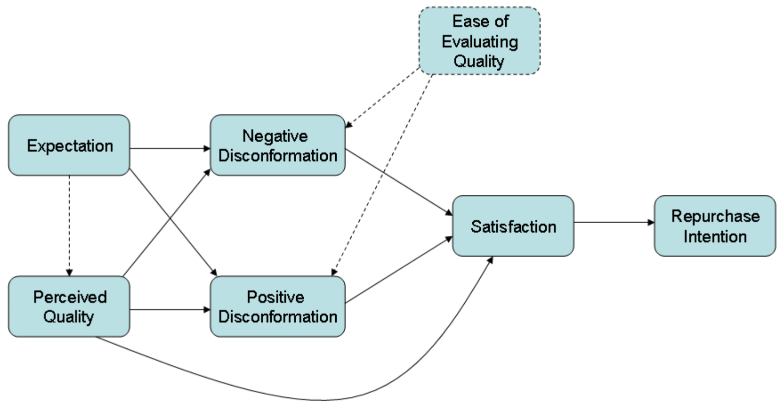


Figure 5.1: Customer Satisfaction Model (Anderson and Sullivan)

is affected by expectation based on the observation: when the difference between expectation and perceived quality is small, customer tends to equate perception to expectation. Furthermore, the level of disconfirmation is positively related to ease of evaluating quality. For network services, the ease of evaluating quality is high as service quality can be readily measured based on the network performance and the application requirements. Hence there is very little ambiguity in customer's perception of quality, and we simplify away this factor in our customer satisfaction relationships. Furthermore, their claim that expectation influences perception is controversial as a number of important findings [120][124][125][126] support the theory that perceived quality influences expectation via a dynamic update process. Based on these works, we reverse Anderson and Sullivan's expectation and perception relationship and formulate an expectation update process.

Figure 5.2 presents our modelling approach. Our view of the customer satisfaction model (CSAT) is a modified Anderson-Sullivan model. The expectation model updates a customer's future service expectation based on past expectations and current utility perception through a recurrent process in our expectation update model. The CSAT model takes as input the service utility, referred to as the "antecedent" of customer satisfaction [119]. It is computed through a utility model that operates on network performance and service attributes. Specialized from SERVQUAL model, we consider three aspects of net-

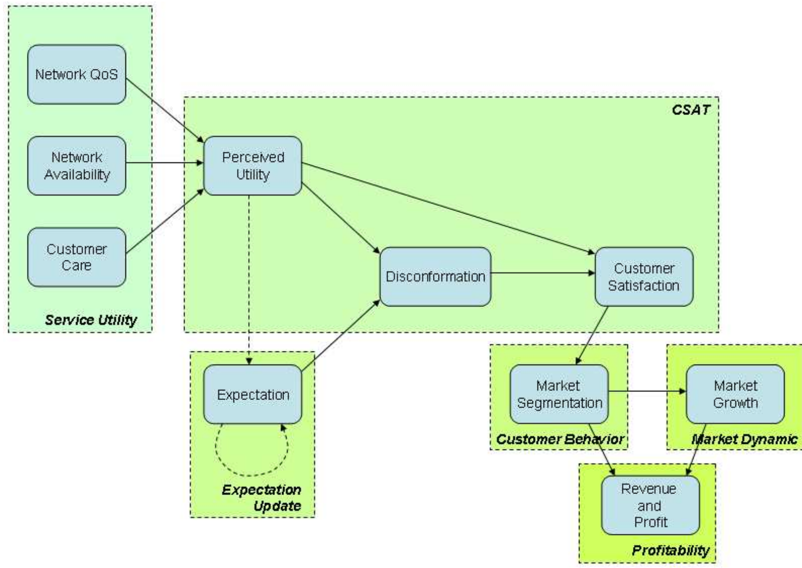


Figure 5.2: Modelling Service Profitability

work services: service quality, service availability and customer care. The "consequence" of customer satisfaction is customer's intention to repurchase [119]. It is captured in our customer behavior model, with regard to market competition and customer desire, to assess subscriber population change via a Bayesian decision process. The output of the customer behavior model is an estimation of the market segmentation: service provider retention, competitor retention, churn, and turnover from competitors. In our market dynamic model, the attractiveness of the service to new entry customers is projected using the Bass growth model [127]. The service profitability is then computed based on the revenue generating potential, derived from the market segments, and the service cost. Since network services are continuous where customers make periodic repurchase decisions (e.g. monthly for xDSL services), the entire process can be iterated through consecutive decision periods, providing long term profitability forecasts of network service operations.

A forecasting model developed from this market science methodology provides significantly better assessment on the impact of network performance on network service profitability, compared with the simple linear models used in network planning research today. In the following section, we detail the construction of such an analytical forecasting model following our methodology and show how it can capture important market trends and customer behavior in later sections.

5.4 Forecasting Service Profitability

In this section, we detail the construction of our analytical model. We first introduce the computation of utility based on network service performance (Section 5.4.1) and then construct the customer satisfaction model (Section 5.4.2), followed by a formalization of the expectation update process (Section 5.4.3). Formulating the outcome as a decision problem, we estimate the market segmentation in customer behavior model (Section 5.4.4) and then deduce the growth of new entry customers in market dynamics model (Section 5.4.5). Finally, service profitability is computed as a function of revenue and cost (Section 5.4.6). Table 5.1 presents a list of the model parameters.

5.4.1 Service Utility and Perceived Utility

As noted by Dabholkar [128], customer satisfaction and utility are not the same construct. Satisfaction is a customer's subjective evaluation of the service performance, while utility is its objective measurable quantification. There are two concepts of utility presented in this section: service utility and perceived utility. Service utility denotes a set of service related performance metrics that are measurable or observable. Together, they yield a single quantitative evaluation of utility: the perceived utility. We first discuss service utility and its computation.

The SERVQUAL model [122] identified tangibles, empathy, assurance, responsiveness and reliability as the five major aspects of service quality. In the context of network services, tangibles, empathy, assurance and responsiveness can be grouped together under customer care, including helpdesk support, installation, troubleshooting, billing service, on-call technical support, etc. Reliability is readily mapped to network service availability, often regarded as an essential factor in network service contract. Empirical studies done

Parameter	Appears in	Definition
γ_1	Service utility	Customer preference for downstream throughput
γ_2	Service utility	Customer preference for upstream throughput
α_1	Service utility	Customer preference for service quality
α_2	Service utility	Customer preference for service availability
α_3	Service utility	Customer preference for customer care quality
μ_1	CSAT	Perception function concavity control
μ_2	CSAT	Perception function concavity control
μ_3	CSAT	Impact of negative disconformation (convexity)
ω_p	CSAT	Maximum value of perception function
ω_{dp}	CSAT	Maximum value of disconformation function
ω_{dn}	CSAT	Minimum value of disconformation function
κ_a	Expectation update	Assimilation factor
κ_r	Customer behavior	Resistance factor
β_G	Expectation update	Positive disconformation modifier
β_L	Expectation update	Negative disconformation modifier
β_M	Expectation update	Memory factor
m	Expectation update	Memory length
p	Market dynamic	Innovator factor
q	Market dynamic	Imitator factor
S	Market dynamic	Maximum market potential (i.e. saturation point)

Table 5.1: List of Model Parameters

in Telecom services from Germany, US, and China [104][105] confirm the applicability of SERVQUAL to network services and suggest network quality as an additional aspect of SERVQUAL. In accordance, we consider service utility U as consisting of three basic aspects: service quality, service availability, and customer care. Service quality captures the network quality aspect by considering customer observed network QoS performance. Service availability represents the network availability experienced by the customer. The three service aspects are further documented by the TeleManagement Forum (TMF) in its SLA handbook suite [129].

To compute service quality Q, we consider factors related to network QoS of the customer's traffic flows, the application requirements, and the customer's own preferences. Let a *service path* denote an end-to-end network path carrying a customer's service traffic running a particular application. For a service path j of customer i, the service quality Q_{ij} is computed by considering a networked application to belong to one of two categories:

QoS-sensitive services and QoS-insensitive services. QoS-sensitive services are applications whose satisfactory performance is contingent on fulfilling certain QoS requirements. For example, a multimedia stream has stringent minimum throughput and maximum round-trip delay bounds, while a web browsing session requirement is more tolerant. On the other hand, QoS-insensitive services are applications that do not have specific QoS requirements. FTP and P2P applications are good examples of this. Their performance is best computed based on an overall measurement of throughput quality.

For QoS-sensitive services, we model Q_{ij} based on the concept of *defective service instances* (DSI). Define a defective service instance experienced by a customer i on a service path j, denoted by D_{ij}, as a series of consecutive network QoS measurements whose values are below the QoS requirements of the supported application. One can observe that during the course of a network trouble, the QoS measurements may fluctuate wildly above and below the QoS requirements. Hence a network flux parameter is introduced to account for this fluctuation. We consider the first observed QoS measurement below the QoS requirements to signal the onset of a D_{ij} instance, which ends when up to network flux number of consecutive QoS measurements are recorded as satisfying the QoS requirements. Let A_{ij} be the total access time of customer i on service path j, and $l(D_{ij})$ be the time length of D_{ij}, then Q_{ij} takes on the following form:

$$Q_{ij} = \frac{A_{ij} - \sum l(D_{ij})}{A_{ij}} \tag{5.1}$$

for QoS-sensitive services

For QoS-insensitive services, we model Q_{ij} based on the average throughput (P^a_{uij} for upload and P^a_{dij} for download) and the maximum bandwidth (P^o_{uij} and P^o_{dij} respectively). The maximum bandwidth is the capability limit of the customer's service offering (e.g. 2Mb/s customer download ceiling for xDSL service). Let γ_1 and γ_2 represent the download and upload performance preferences of the customer, then Q_{ij} is computed as:

$$Q_{ij} = \gamma_1 \frac{P^a_{dij}}{P^o_{dij}} + \gamma_2 \frac{P^a_{uij}}{P^o_{uij}} \tag{5.2}$$

$\gamma_1 + \gamma_2 = 1$, *for QoS-insensitive services*

Given the above formulation of service utility. We now define the computation of perceived utility. Let W_{ij} denote the percentage of time a service path j of customer i is deemed available, C_i be a scalar rating (between 0 and 1) of customer care service for customer i, SP_i be a set of service paths customer i uses, and α_1, α_2, and α_3 be customer i's weight preferences for service quality, service availability, and customer care respectively, the perceived utility for customer i is expressed as:

$$\begin{aligned} U_i = \quad & \frac{\alpha_1 \sum_j (Q_{ij} \times A_{ij})}{\sum_j A_{ij}} \\ & + \frac{\alpha_2 \sum_j (W_{ij} \times A_{ij})}{\sum_j A_{ij}} + \alpha_3 C_i \\ \textit{where } j \in SP_i & \textit{ and } \alpha_1 + \alpha_2 + \alpha_3 = 1 \end{aligned} \tag{5.3}$$

Taking as input the network and service performance of customer i's service paths, Equations 5.1, 5.2 and 5.3 yield the perceived utility of customer i, normalized between 0 and 1. This is a unified quantification of the service utility according to customer's service preference and serve as the input to the perception and the disconfirmation functions, described in Section 5.4.2. The presented utility model is theoretical. Our earlier work [111] gives a pragmatic framework on how the above computations can be performed in practice. Furthermore, Section 5.6 also provides some demonstration on how such computation could be carried out in regional networks.

5.4.2 Customer Satisfaction (CSAT)

Customer satisfaction can be modelled through the interaction between perceived utility and expectation [116][117], expressed as a linear combination of a perception function and a disconfirmation function. Let f_1 be the perception function, f_2 be the disconfirmation function, U_{pi} and U_{ei} be the perceived utility and expected utility (i.e. expectation) of customer i, then the general form of customer satisfaction Γ_i for a customer i is given in [119] as:

$$\Gamma_i = f_1(U_{pi}) + f_2(U_{pi} - U_{ei}) \tag{5.4}$$

The perception function gives the baseline customer satisfaction obtained from service utility, while the disconfirmation function modifies this satisfaction value based on the discrepancy between perceived utility and expectation (i.e. disconfirmation). The initial value of U_{ei} for new entry customers of a service can be computed using expected network service performance derived from service contract terms. This perception-disconfirmation theory for customer satisfaction has been confirmed in many empirical market research over the years, including very recent studies done in the Telecom sectors [103][104][105]. In the subsections below, we derive general mathematical forms for the perception and the disconfirmation functions.

The perception function

The perception function f_1 is a mapping between perceived utility and baseline customer satisfaction. It is described in [119] as an increasing concave function starting at the origin (i.e. $f_1(0) = 0$). Its general shape is conceived based on the observation that as the utility increases, the customer becomes less sensitive to changes in utility. The rate of change of the perception function can be expressed as:

$$f_1''(x) = \mu_2 x - \mu_1 \tag{5.5}$$
$$where\ x = U_{pi},\ and\ \mu_1,\mu_2 \geq 0$$

The parameters μ_1 and μ_2 control the concavity of the perception function. In Section 5.5, we will discuss our choice of this particular form f_1''. Integrating Equation 5.5 yields:

$$f_1'(x) = \int f_1''(x)dx = \frac{\mu_2}{2}x^2 - \mu_1 x + \Psi \tag{5.6}$$
$$where\ x = U_{pi},\ and\ \mu_1,\mu_2 \geq 0$$

The constant Ψ is a weight that ensures $f_1'(x)$ remains positive (i.e. f_1 is an increasing function) for all possible values of x. The perception function then takes on the following form:

$$f_1(x) = \int f_1'(x)dx = \frac{\mu_2}{6}x^3 - \frac{\mu_1}{2}x^2 + \Psi x + C \tag{5.7}$$
$$where\ x = U_{pi},\ and\ \mu_1,\mu_2 \geq 0$$

The constraint $f_1(0) = 0$ yields $C = 0$.

Observe that the domain of f_1 is bounded between 0 and 1. Moreover, we would like $f_1''(x)$ to be non-positive and $f_1'(x)$ to be non-negative for all possible values of x, and control the maximum value of f_1 via parameter ω_p (i.e. $f_1(1) = \omega_p$). Thus the following set of constraints on parameters of f_1 arises:

$$\begin{cases} \mu_1 \geq \mu_2 & \text{from } f_1''(x) \leq 0 \\ \Psi \geq \mu_1 - \frac{\mu_2}{2} & \text{from } f_1'(x) \geq 0 \\ \Psi = \frac{\mu_1}{2} - \frac{\mu_2}{6} + \omega_p & \text{from } f_1(1) = \omega_p \\ \mu_1, \mu_2 \geq 0, \omega_p > 0 & \end{cases} \tag{5.8}$$

Satisfying the constraint set of 5.8 entails solving the following inequality:

$$\frac{\mu_1}{2} - \frac{\mu_2}{6} + \omega_p \geq \mu_1 - \frac{\mu_2}{2} \tag{5.9}$$
$$\textit{where } \mu_1 \geq \mu_2 \textit{ and } \mu_1,\mu_2 \geq 0,\ \omega_p > 0$$

By solving Inequality 5.9, the constraint on μ_1 can be expressed as:

$$0 \leq \mu_1 \leq 2\omega_p + \frac{2}{3}\mu_2 \tag{5.10}$$

The constraint of 5.10 suggests that the upper bound of μ_1 is positively related to the upper bound of μ_2. As we would like the concavity parameter μ_1 to have the largest possible value range, and given the constraint $\mu_1 \geq \mu_2$, then the maximum value range of μ_1 is obtained when $\mu_1 = \mu_2$. This leads to a desirable simplification of f_1. Figure 5.3 demonstrates the general characteristics of the perception function. In summary, f_1 is a function of perceived utility, with the following form:

$$f_1(x) = \frac{\mu_1}{6}x^3 - \frac{\mu_1}{2}x^2 + \left(\frac{\mu_1}{3} + \omega_p\right)x \tag{5.11}$$
$$\textit{where } \mu_1 \leq 6\omega_p,\ \mu_1 \geq 0, \textit{ and } \omega_p > 0$$

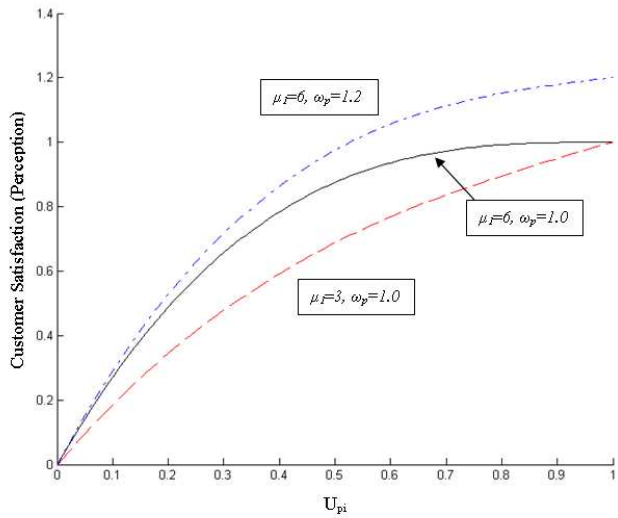


Figure 5.3: The Perception Function

The disconfirmation function

The disconfirmation function f_2 accounts for the subjectivity of customer evaluation given a reference point (i.e. expectation). Tversky and Kahneman [130] found that "losses relative to a reference value looms larger than gains". Grounded on this psychological theory, Anderson and Sullivan [119] suggest that customer satisfaction is mildly increasing when perceived utility exceeds expectation and is significantly reduced when perceived utility falls below expectation. We formalize this interaction as a two-piece increasing function:

$$f_2(x) = \begin{cases} \omega_{dp}x & x \geq 0 \\ \omega_{dn}(x+1)^{\mu_3} - \omega_{dn} & x \leq 0 \end{cases} \tag{5.12}$$

$$\textit{where } x = U_{pi} - UTIL_{ei},\ \mu_3 \geq 0,\ \textit{and } \omega_{dp},\omega_{dn} > 0$$

The domain of f_2 is bounded between -1 and 1. The function is continuous (i.e. the two piece-wise functions converge at $x = 0$). The parameter ω_{dp} controls the maximum value of f_2 (i.e. maximum positive disconformation) while ω_{dn} controls the minimum value of f_2 (i.e. maximum negative disconformation). μ_3 regulates the impact of negative disconformation on customer satisfaction. The general characteristics of the disconformation function is illustrated in Figure 5.4.

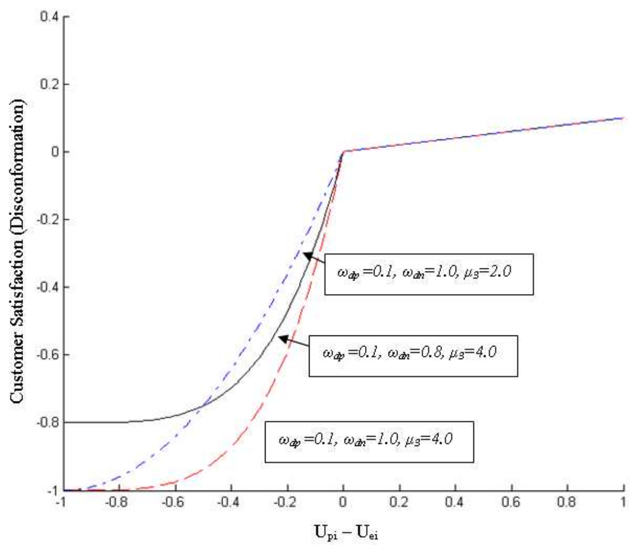


Figure 5.4: The Disconfirmation Function

Customer satisfaction

From Equation 5.4, 5.11 and 5.12, one observe that Γ_i is bounded between $-\omega_{dn}$ and $\omega_p + \omega_{dp}$. In general, the choice of ω parameters should follow: $\omega_p \geq \omega_{dn} > \omega_{dp}$. ω should be fixed for all customers of a service and ω_{dp} should be small compared to ω_p.

As shown in Figure 5.5, the rate of change in customer satisfaction differs significantly when perceived utility falls below and exceeds expectation. The rate and severity of dissatisfaction (controlled by ω_{dn} and μ_3 respectively) reflect different customer's tolerance to negative disconfirmation. Our formalization of the customer satisfaction fits a rational customer's subjective evaluation of the service utility, and conforms to empirical findings [116][117][119][130]. In addition, the model offers a set of well-defined control parameters to fit different service characteristics, and individual customer's preferences and sensitivities.

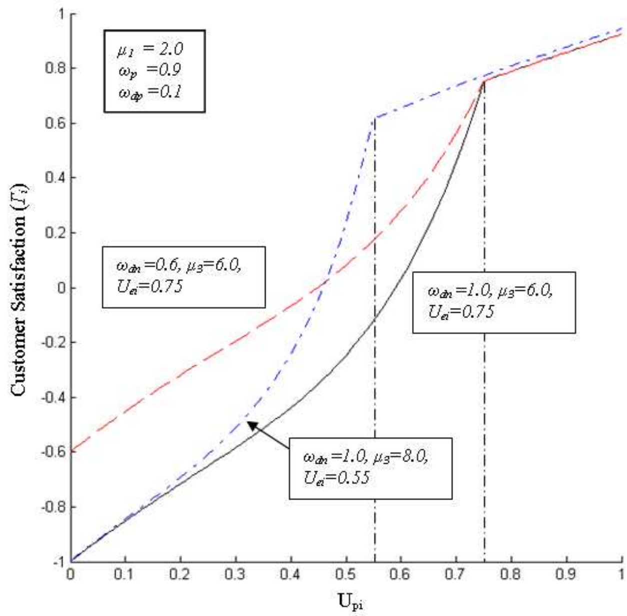


Figure 5.5: The Customer Satisfaction Function

5.4.3 Expectation Update

Empirical studies [126][131] suggest that a customer adjusts his/her future expectation of service utility based on current expectation and perception. The studies also find favorable disconfirmation increases future expectation while unfavorable disconfirmation has the opposite effect.

Through an expectation update process, we deduce a customer's future expectation as a function of the customer's current expectation and disconfirmation, subject to two psychological factors: assimilation and experience. When the relative level of disconfirmation is small, a customer tends to equate the perceived utility to the expected utility, due to assimilation effect [120]. Furthermore, as a customer perceives consistent service utility over time, he/she gains experience with the service, and consequently is less sensitive to short term utility fluctuations [126]. In other words, the customer gradually establishes long term reputation of the service.

Let κ_a be the assimilation factor, a customer i's future expectation U^*_{ei} has the following form:

$$U^*_{ei} = \begin{cases} U_{ei} & \left|\frac{U_{pi}-U_{ei}}{U_{ei}}\right| \leq \kappa_a \\ \\ \begin{array}{l} h(U_{ei})[U_{pi} - U_{ei}] \\ +U_{ei} \end{array} & \text{otherwise} \end{cases} \tag{5.13}$$
$$U^*_{ei} \textit{ is constrained to } 0 \leq U^*_{ei} \leq 1$$

The parameter κ_a is constrained ($0 \leq \kappa_a \leq 1$) and should be a very small value (e.g. 0.01). The function $h(U_{ei})$ adjusts the expectation as a factor of the disconfirmation. Our general form of Equation 5.13 is established based on assimilation theory of economics [125], where new information are assimilated as an aggregate quantity over time. According to literatures, Equation 5.13 should exhibit three characteristics. First, given the same expectation, a negative disconfirmation is weighed much more heavily than a positive disconfirmation [130]. This effect is similarly reflected in the construct of disconfirmation function. Second, a positive disconfirmation has a greater impact on U^*_{ei} as U_{ei} decreases, and conversely a negative disconfirmation has a greater impact on U^*_{ei} as U_{ei} increases [120]. Third, the longer a customer experiences consistent utility, the less impact on expectation should a short term utility fluctuation have [126]. Based on these characteristics, $h(U_{ei})$ is constructed as such:

$$h(U_{ei}) = \begin{cases} \beta_G \beta_M^m (1 - U_{ei}) & \frac{U_{pi}-U_{ei}}{U_{ei}} > \kappa_a \\ \\ \beta_L \beta_M^m U_{ei} & \frac{U_{pi}-U_{ei}}{U_{ei}} < -\kappa_a \end{cases} \tag{5.14}$$
$$\textit{where } 0 \leq m \leq \Upsilon,\ 0 \leq \beta_M \leq 1,\ \textit{and } 1 \leq \beta_G < \beta_L$$

β_G and β_L are the positive and negative disconfirmation factors respectively. β_M is the memory factor and m the memory length. The term β_M^m controls the significance of the new information (i.e. current disconfirmation) on the aggregate (i.e. expectation). As m increases, β_M^m decreases. We use integer values for m, representing the number of repurchase evaluations the customer underwent while using the service. The constant Υ

represents the maximum memory length a customer keeps track of. The value of m is updated (m^*) based on the following equation:

$$m^* = \begin{cases} m+1 & \left|\frac{U_{pi}-U_{ei}}{U_{ei}}\right| \leq \kappa_a \\ \\ m-1 & \text{otherwise} \end{cases} \tag{5.15}$$
$$m^* \textit{ is constrained to } 0 \leq m^* \leq \Upsilon$$

The initial value of m is set to 0. One can infer from Equation 5.15 that as customer regularly experiences consistent service performance, he/she is more insulated from short term performance fluctuations. Conversely, when performance significantly fluctuates over time, the customer is unable to make an experienced evaluation of the service, and hence his/her reliance on new information does not diminish with time (i.e. the divergence effect).

In this section, we formally constructed the process of expectation update, with consideration for assimilation and experience effects. The value of expectation is bounded between 0 and 1, and is adjusted based on perceived utilities in an iterative fashion.

5.4.4 Customer Behavior: Repurchase Decision and Market Segmentation

Repurchase intention is the direct consequence of customer satisfaction [119]. Researches in inter-temporal planning (e.g. [124]) state that customers re-estimate purchase decisions periodically based on previous estimates and new information. Furthermore, there exists a strong linkage among customer satisfaction, future expectation, and repurchase intention [119][120]. We formulate the customer's repurchase intention as a decision problem, subject to the following assumptions: the customer is rational in his/her purchase choice and does not exit the service market (i.e. the customer seeks maximization of future satisfaction); furthermore all competitors in the same service market charge similar price, have identical technology attractiveness from the customer perspective, and employ similar advertisement strategies. Although the above factors could be included in our analysis, they are discounted for sake of simplicity. In practice, with the fierce competitions exist among

SPs, these assumptions often hold. The customer's decision to use a service from a particular service provider is primarily influenced by the customer's current level of satisfaction and expected future utility. The finding of [131] suggests that when a customer chose a service brand that meets his/her desire, he/she is likely to choose the same service brand again regardless if the brand has the highest expected performance in the market or not. Therefore, we consider a customer i will stay with a service provider if his/her customer satisfaction at the end of the current service period is above such a desire threshold Γ_i^D. If below Γ_i^D, the action of choosing a new service provider is a decision problem in which the customer attempts to maximize his/her future satisfaction based on his/her future expectations of similar services. Let U_{eiv} be the future expectation of service v estimated by a customer i, let k be the service customer i has just used, let Φ be the set of all similar services, and let κ_r be the resistance factor of customer i, then the decision problem can be expressed as:

$$max \{f_1(U_{eik}) + \kappa_r, f_1(U_{eiv})\} \tag{5.16}$$
$$\forall(v \in \Phi, v \neq k)$$

The parameter κ_r is a small satisfaction modifier representing the extra effort (e.g. service switching time, etc.) customer i has to spend in order to switch service provider. Equation 5.16 relies on precise knowledge of a customer's future expectations. In practice, a customer's expectation of services he/she has not used can at best be estimated from service reputation with some uncertainty. Hence, Equation 5.16 can be reformulated as a Bayesian decision problem [132]. Let $F_{ev}(\mu_{ev}, \sigma_{ev})$ be the probability distribution of expectation of service v, with mean μ_{ev} and standard deviation σ_{ev}, the decision problem can be expressed as:

$$max \left\{ f_1(U_{eik}) + \kappa_r, \int f_1(\mu_{ev}) dF_{ev}(\mu_{ev}, \sigma_{ev}) \right\} \tag{5.17}$$
$$\forall(v \in \Phi, v \neq k)$$

In Equation 5.17, $f_1(\mu_{ev})$ is the loss function and $F_{ev}(\mu_{ev}, \sigma_{ev})$ is the prior distribution. Given overall customer turnover rate in the market, Equation 5.17 could also be used in a random sampling process to forecast service switching decisions of customers from other competitors. Ultimately, applying this decision process to all consumers in the market classifies the consumer population into three disjoint partitions: the set of customers with intention to repurchase the same service k (Ω_R), the set of customers choosing not to use service k (Ω_N), and the set of customers switching to service k from another service provider (Ω_P).

5.4.5 Market Dynamic

Up to this point, we have considered the partitioning of the existing consumer market. The entry of new consumers in the market could be described by the Bass growth model [127]. This model is applicable to network service industry, as suggested by the techno-economic studies on European xDSL market penetration [133]. The Bass model categorizes new consumers that enter the market into two categories: innovators and imitators. The innovators enter the market without any incentives and they are the main consumer faction during the inception of the market; the imitators are attracted to the market by the innovators and they are the main consumer faction as the market matures. The hazard function of the Bass model, describing the conditional probability of new consumers entering the market, is formally expressed [127] as:

$$\frac{f(T)}{1-F(T)} = p + qF(T) \tag{5.18}$$
$$\textit{where } 0 < p < q < 1 \textit{ and } p + q = 1$$

$f(T)$ is the probability density function over time T, while $F(T)$ is the cumulative function over T. The parameter p is the coefficient of innovators and q is the coefficient of imitators. In general, p is much smaller than q. Rewriting and integrating Equation 5.18 yield:

$$\int \frac{dF}{p + (q-p)F - qF^2} = \int dT \tag{5.19}$$

The solution to Equation 5.19 yields the cumulative function $F(T)$ [127]:

$$F(T) = \frac{1 - e^{-(p+q)T}}{1 + \frac{q}{p}e^{-(p+q)T}} \tag{5.20}$$

and the density function $f(T)$ [127]:

$$f(T) = \frac{\frac{(p+q)^2}{p} e^{-(p+q)T}}{\left(1 + \frac{q}{p} e^{-(p+q)T}\right)^2} \tag{5.21}$$

Figure 5.6 illustrates the characteristics of $f(T)$. The values of $f(T)$ when $T < 0$ has no practical meaning since $T = 0$ indicates the inception of the market. For ease of comparison, the p and q parameters in Figure 5.6 do not conform to the constraint $p + q = 1$. The parameter p specifies the initial consumer population size (i.e. $f(0) = p$), and q affects the arrival probabilities of the imitators.

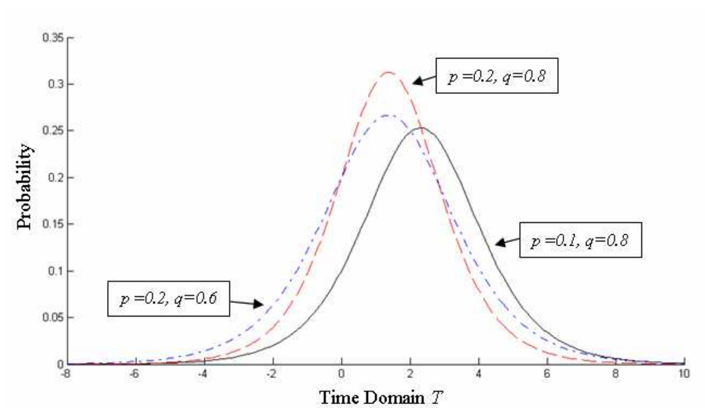


Figure 5.6: Probability Density Function of Bass

Let S be the market potential (i.e. the maximum number of consumers), and $L(t)$ be the mapping function that maps real time t to time domain T of the Bass model, then the number of entry customers that choose service k can be represented as:

$$\Omega_N = \frac{\Omega_R + \Omega_P}{\Omega_R + \Omega_P + \Omega_N} S[F(L(t_{c+1})) - F(L(t_c))] \tag{5.22}$$

The time value t_c denotes the end of current service period (i.e. current evaluation time), while t_{c+1} denotes the time of next evaluation. The last term in Equation 5.22 is the cumulative probability of new consumers entering the market from current time to next evaluation time. The model estimates that a fraction of them will choose service k based on the competitiveness of service k at current time. This is represented by the first term of Equation 5.22.

In the above discussion, we have considered a single service market. For SPs that has multiple service offerings, a market dynamic should be established per service. We will show an example of this in Section 5.6.

5.4.6 Service Profitability

From our forecast of consumer market segmentations at time t_c, the revenue generating potential R_k of service k in $[t_c, t_{c+1}]$ time interval is:

$$\begin{aligned} R_k = \quad & (\Omega_N + \Omega_P) \times \xi_N \\ & + \Omega_R \times \xi_R \end{aligned} \tag{5.23}$$

The parameters ξ represent the price of service k to new customers ξ_N and old customers ξ_R in time interval $[t_c, t_{c+1}]$. Follow from Equation 5.23, the profitability of service k is then:

$$PROF_k = R_k - COST_k - PEN_k \tag{5.24}$$

The parameters $COST_k$ and PEN_k are the cost of running service k and the estimated monetary penalties (e.g. due to contract violation, etc.) from time t_c to time t_{c+1}.

5.5 Model Analysis

In Section 5.4.2, we have formalized the customer satisfaction function based on a number of literatures. In the context of network services, we now discuss the particular choice of the perception function and analyze the impact of the parameters in the model.

5.5.1 Choice of the Perception Function

In constructing the perception function, we also considered two other simple equation forms (Equations 5.25 and 5.26). Both of them are concave increasing functions in the domain of 0 to 1.

$$f_1(x) = \omega_p x^{\mu_1} \tag{5.25}$$

$$\textit{where } 0 \leq \mu_1 \leq 1,\ \omega_p \geq 0,\ \textit{and } 0 \leq x \leq 1$$

$$\textit{or}$$

$$f_1(x) = 1 - e^{-\mu_1 x} + (e^{-\mu_1} - 1 + \omega_p)x \tag{5.26}$$

$$\textit{where } \mu_1,\omega_p \geq 0,\ \textit{and } 0 \leq x \leq 1$$

Similar to Equation 5.11, the μ_1 parameter controls the concavity and $f_1(1) = \omega_p$. Figure 5.7 illustrates the characteristics of Equations 5.11, 5.25 and 5.26.

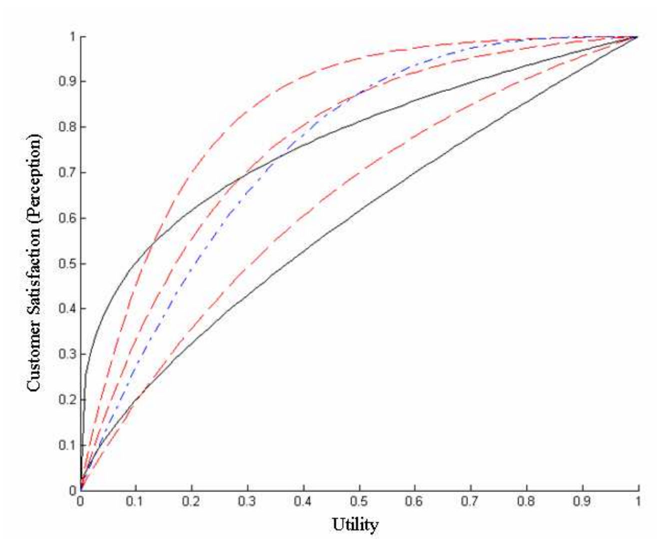


Figure 5.7: Forms of Perception Functions

The solid curves are the forms of Equation 5.25 with varied concavities. These forms are useful in modelling services that have high customer satisfaction even when utility is low, and the effect of desensitization is not significant when utility is high. The dot slash curve is the perception function of Equation 5.11 with maximum concavity. The dash curves are the forms of Equation 5.26 with varied concavities. Unlike Equation 5.11, the forms of Equation 5.26 do not place constraint on μ_1. However, for curves with similar concavity, the forms generated by Equation 5.11 delay the severity of desensitization until higher utility level. In our work, Equation 5.11 is chosen because it appears to fit what one can expect from network services best: the increase in customer satisfaction will be approximately linear to increase in utility when utility level is low and the effect of desensitization does not become very significant until utility level is high (i.e. over 0.7). In addition, the concavity factor of Equation 5.11 is more meaningful to analysis (i.e. f'' is in linear form). Higher orders of polynomials are also considered, but they do not add significant control to concavity. In practice, the choice of a best form should be network service specific and be determined based on empirical data gathered for the analyzed service.

5.5.2 Impact of The Perception and Disconfirmation parameters

The parameters ω_p, ω_{dp} and ω_{dn} define the range of customer satisfaction values. The maximum disconformation parameter ω_{dp} should be much smaller than the maximum perception parameter ω_p as utility above expectation does not induce significant satisfaction improvement from customers. The combination of ω_p and ω_{dp} gives the maximum ceiling value of customer satisfaction Γ. A value above 1 is not meaningful as Γ is bounded between 0 and 1. However, a value of below 1 is quite feasible, as Γ may be influenced by non-service related factors (e.g. a chronical complainer is unlikely to be fully satisfiable regardless of delivered service utility). The ω_{dn} controls the maximum impact a negative disconfirmation has on perception. When ω_{dn} is large, the degree of negative disconfirmation is also large. As Γ is non-negative, ω_{dn} should be at most as large as ω_p.

When the service utility is fixed, the parameter μ_1 of the perception function is linearly related to customer satisfaction. A higher μ_1 value results in a higher customer satisfaction value. Figure 5.8 illustrates the interactions among μ_1, utility, and customer satisfaction.

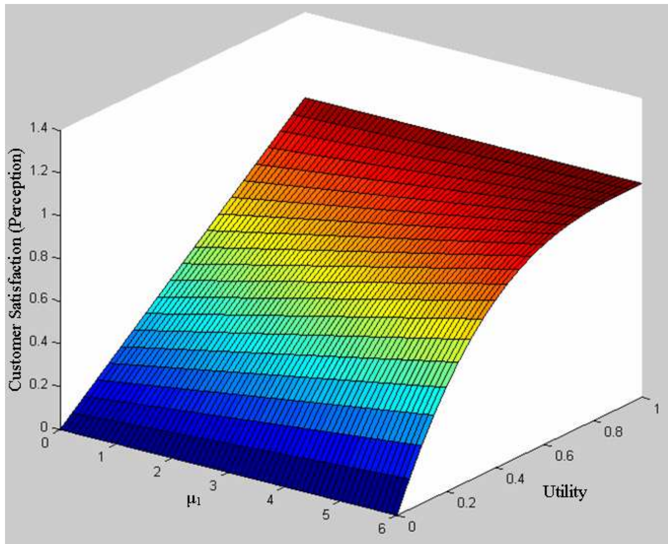


Figure 5.8: Effect of μ_1 on Perception

Figure 5.8 suggests that customer satisfaction is particularly sensitive to the choice of μ_1 when the utility value is moderate (i.e. $0.4 \sim 0.8$). For instance, given a utility value of 0.6, the customer satisfaction is as low as 0.6 when $\mu_1 = 0$, and as high as 0.95 when $\mu_1 = 6$. With high μ_1 values (i.e. $\mu_1 \geq 3$), the customer satisfaction increases much more rapidly when utility is below 0.5. One can infer from Figure 5.8 that when μ_1 is high (i.e. $\mu_1 \geq 3$), it is more beneficial for the network service provider to keep service utility at a moderate range (i.e. $U \approx 0.8$). However, this inference holds only if the effect of disconfirmation is low (i.e. the customer expectation is met or the customer has high tolerance to negative disconfirmation).

When perceived quality falls below expectation, the impact of negative disconfirmation on customer satisfaction could be significant. Parameter μ_3 controls the rate of this reduction. When utility is fixed, an increase in μ_3 exponentially decreases customer satisfaction (Figure 5.9). However when negative disconfirmation is very low (i.e. below 0.1), increases in μ_3 approximately result in a linear reduction of customer satisfaction. One can thus infer that network service providers should always ensure that the perceived utility of a customer does not fall below his/her expectation.

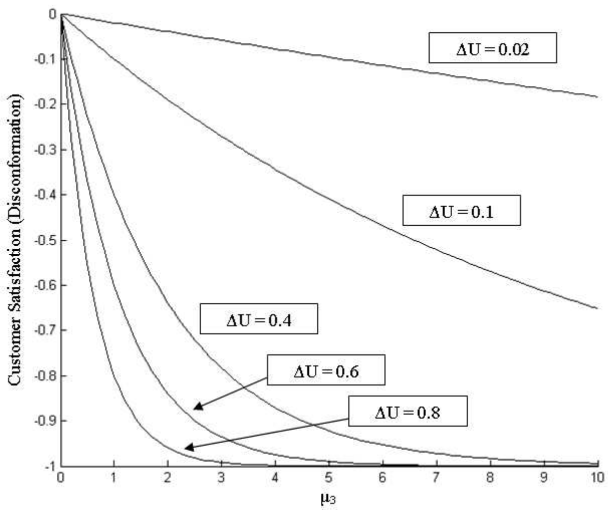


Figure 5.9: Effect of μ_3 on Disconfirmation

In summary, for network services, where both μ_1 and the customer expectation are high (i.e. $\mu_1 \geq 3$ and $U_{ei} \geq 0.7$), customer satisfaction does not differ significantly when perceived utility exceeds expectation. However, when expectation is not met, negative disconfirmation will have a significant impact on customer satisfaction, depending on the value of μ_3. Hence, to retain customers, it is sufficient for a network service provider to deliver service at a quality level matching the expectations of the customers, without maximizing their perceived utilities. However, this observation holds only if the customer does not have a low expectation of the service. Low service expectation yields overall customer satisfactions below the desire threshold Γ_i^D and drives customers to evaluate alternatives. In practice, the parameters of Γ should be acquired through data fitting techniques. The network performance of customers could be obtained in conjunction with customer satisfaction surveys overtime. The computed service utility should be plotted against customer satisfaction and then use best-fit techniques to determine the most appropriate function parameters.

5.5.3 Impact of other model parameters

Service utility parameters for QoS-insensitive services influence a customer's preference proportioning between upstream throughput performance γ_1 and downstream throughput performance γ_2. This proportioning is very application and user dependent. For P2P applications, one typically expect γ_2 to be much higher than γ_1, whereas for FTP-based applications, the proportioning depends on customer's access behavior. For customer preference parameters α_1, α_2 and α_3, the proportioning fundamentally influences the degree of impact each aspects of service utility has on the overall customer satisfaction, and therefore service profitability. For example, when customer care preference α_3 is high, enhancing network infrastructure does not constitute good investment strategy. Hence it is important for a forecasting model to identify such customers as they influence the profit margin of network upgrades. All of the service utility parameters should be acquired via structured SERVQUAL customer survey. Survey methods used in Telecom customer satisfaction studies [104][105] could serve as guides.

In the expectation update process, the assimilation factor κ_a controls the likelihood of the assimilation effect. The perception of difference does not differ significantly between humans and should be small [125] (e.g. $\kappa_a = 0.01$). The parameters β_G and β_L are modifiers of disconfirmation. Consider $U_{ei} = 0.5$, the impact of disconfirmation on expectation is differentiated by β_G and β_L. The value of β_G and β_L should be equal to or greater than 1, with β_L larger than β_G. The parameter β_M considers the effect of past experience. A larger β_M causes current disconfirmation to be evaluated more significantly on expectation with regard to experience, and the dissipation of this impact is slower as experience accumulates. More specifically, a customer without prior experience is not influenced by β_M (i.e. β_M^0). The cumulation of experience (i.e. $m \geq 1$) rapidly lessens the impact of current disconfirmation on expectation, represented by β_M^m. Furthermore experience is accumulated with consistent performance, whether good or bad, and can be destroyed by inconsistency.

The Bass model parameters p, q and S governs the general market growth pattern in a service market. The parameter values are determined via techno-economic studies (e.g. [133]) or growth analysis of similar service markets in the past.

5.6 Case Studies and Simulation

In this section, we demonstrate the effectiveness and practicality of our approach through two sets of case studies and simulations. First, we show how our models could help in a network upgrade decision process and illustrate how key economic, customer and market factors that influence network service planning are captured by the models. Then we analyze the performance of a typical regional ISP network through simulation and show that by representative flow tracking, our model can be applied to WANs. A comparative analysis of the network infrastructure is conducted from three different perspectives: network utilization, customer traffic flow, and customer satisfaction.

As the basis of our first discussion, we simulate a network infrastructure and customer population that is representative of a real world network planning scenario, onto which we offer three equally promising upgrade strategies. We show that a sound upgrade decision could not be made by only considering network performance, even with the ability to track customer's access behaviors. We then present a step by step application of our model incorporating the customer and economic factors and show that service profitability derived from customer satisfaction could be an effective decision indicator for network planning and upgrades. We further demonstrates how our expectation model captures the effect and durability of customer loyalty over long term and how different market dynamics play a crucial role in network service planning. Through this discussion, we also show how the model formally relate and explain some important market observations on customer behaviors and growth.

The network infrastructure as depicted in Figure 5.10 is realized in NS2 [134]. The link

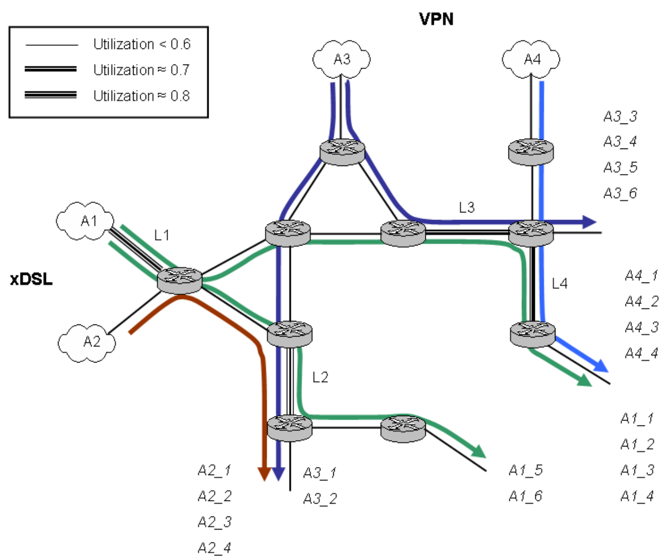


Figure 5.10: Simulation Topology

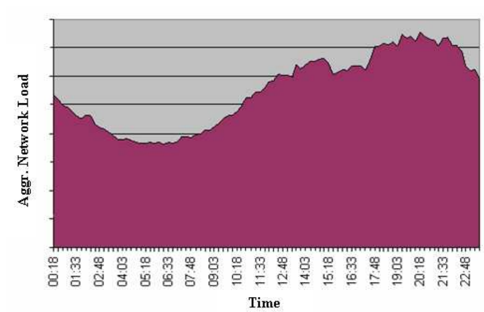


Figure 5.11: Daily Aggregate Load of A Regional ISP Network

capacity is set to 24 Mbps. Two regions of xDSL customers ($A1$ and $A2$) and two regions of VPN customers ($A3$ and $A4$) generate a total of 20 service paths (e.g. $A1\_1$, $A2\_1$, etc.). Each service path is used by 5 customers with similar application characteristics, QoS requirements, and access time (Table 5.2). However, each customer varies in performance preferences and service expectations. The routing in this setup is static, such that all customer flows will follow the pre-established paths. As the customers in regions $A1$ and $A2$

Path ID	Traffic Type	Requirements / Opt. BW	Start Time	End Time
A1_1	FTP	800 Kb/s Max.	18:00	23:00
A1_2	FTP	800 Kb/s Max.	16:00	23:00
A1_3	FTP	800 Kb/s Max.	6:00	24:00
A1_4	FTP	800 Kb/s Max.	0:00	24:00
A1_5	HTTP	120Kb/s, 100ms	18:00	23:00
A1_6	HTTP	120Kb/s, 100ms	16:00	23:00
A2_1	HTTP	120Kb/s, 100ms	16:00	23:00
A2_2	HTTP	120Kb/s, 100ms	18:00	23:00
A2_3	FTP	800 Kb/s Max.	6:00	24:00
A2_4	FTP	800 Kb/s Max.	0:00	24:00
A3_1	CBR	450Kb/s, 100ms	6:00	24:00
A3_2	CBR	450Kb/s, 100ms	0:00	24:00
A3_3	CBR	450Kb/s, 100ms	9:00	17:00
A3_4	CBR	450Kb/s, 100ms	11:00	19:00
A3_5	CBR	450Kb/s, 100ms	6:00	24:00
A3_6	CBR	450Kb/s, 100ms	0:00	24:00
A4_1	CBR	450Kb/s, 100ms	9:00	17:00
A4_2	CBR	450Kb/s, 100ms	11:00	19:00
A4_3	CBR	450Kb/s, 100ms	6:00	24:00
A4_4	CBR	450Kb/s, 100ms	0:00	24:00

Table 5.2: Customer Service Paths

represents xDSL customers. We have chosen the traffic mix to be primarily a mix of web based light traffic (web browsing, email, etc.) and regular data transfer occurring in prime time after work. We also disperse sources of semi-permanent and permanent FTP traffic to represent P2P and other constant data transfer traffic often present in the SP networks. For the VPN customers, CBR traffics are used to represent their constant resource demand. For realism, the simulated network infrastructure and customer behaviors are designed to facilitate traffic intermixing among customers of different service classes and introduce a varied mix of customer access behaviors. Additional background aggregate traffics modelled as Pareto flows establish a daily cyclic pattern (Figure 5.11). These aggregate traffic are introduced to obtain the desired average link utilizations (Figure 5.10), yet still reflect the typical daily network load observed by a SP. Details on the setup of aggregate traffics are presented in the latter simulation case. In our simulation studies, the 24 hr. daily cycle is mapped to 120 min. simulation time so that each simulation run is within reasonable time bound. With this setup being the network conditions and customer population at the deployment time of the network upgrades, we consider four upgrade options: no upgrade (base case), upgrade links L1 and L2 to 48 Mbps (option 1), upgrade links L2 and L3 to 48 Mbps (option 2), and upgrade links L2 and L4 to 48 Mbps (option 3). All three upgrade options have the same cost. Option 1 is an aggressive upgrade strategy aimed at pleasing xDSL customers, while option 2 and 3 are more balanced strategies. Each of the upgrade option is simulated multiple times in NS2 through a 24 hr. day (i.e. 120 min. simulation time) and the QoS measurements (i.e. delay and throughput) are taken at 1 min. intervals. The average measurements of each interval across runs are used to compute network performance for the customers.

When taking into account the customer access behaviors (i.e. customers' service path and service time), we could compute the perceived utility of each upgrade option. For simplicity, we assume the customers only care about service quality (i.e. α_1=1, α_2, α_3=0). It then implies that the perceived utility is analogous to the ratio of average throughput to optimum bandwidth for FTP traffic, and to the percentage of non-defective service time for the other traffic. The solid lines in Figure 5.12 present the perceived utility of the customers under each upgrade option. In each region, the customers are ordered on the graphs by their path IDs. In the base case of no upgrades, notice that xDSL customers accessing path $A1\_3$ and $A1\_4$ have significantly better perceived utilities compared to other xDSL customers because they do not access link L1 and are not "prime time" traffic. Option 1 significantly increases the service performance of the xDSL customers at the expense of some moderate performance drop from the VPN customers, which seems quite acceptable. The perceived utility of some customers from region $A1$ remains low due to L3 link load. Option 2 and 3 are roughly equivalent as they improve the service performance of region $A1\_3$ and $A1\_4$ respectively. Although informative from a network performance point of view, this analysis provides few insight into which upgrade option is in fact the most beneficial, as all of the options improve performance for some customers. Without considering the economic and customer factors, there is little else that could be done. We now carry the computation of these options through the rest of our model.

For this scenario, let the expectation of the customers be normal distributions with means of 0.65, 0.7, 0.8 and 0.9 respectively for $A1$, $A2$, $A3$ and $A4$, and a standard deviation of 0.05. This distribution is used to reflect the diversity in customer expectations and the relative differentiation between the xDSL customers and the VPN customers. Figure 5.12 illustrates the perceived utilities, expectations, and customer satisfactions of the customers under each upgrade option. The choice of customer satisfaction parameters are taken so that the customer satisfaction function exhibits its general form as observed in empirical studies. In the base case, the xDSL customers have significant negative disconfirmations and consequently low satisfactions. Option 1 significantly improves the satisfactions of xDSL customers, although some of the customers from region $A1$ are still dissatisfied due to L3 link load. Interestingly, the additional influx of $A1$ traffics on L3 and L4 reduced the perceived utilities of VPN customers, just enough to make their perceived utilities to fall below expectations. Therefore, the computations of Γ indicate option 1 may not be a good upgrade option. Regarding option 2 and option 3, our model suggests that negative disconfirmations are eliminated in region $A3$ and $A4$ respectively in each option. When factoring in the Γ^D level, one further observe that option 2 seems to generate more dissatisfied customers than option 3 due to the high expectation rating in region $A4$. To accentuate our case, consider a saturated xDSL and VPN market where the network service provider does not offer the best service. This market condition could be represented in our model with zero market growth and a 0.0 market turnover rate (from other competitors). According to our market segmentation model, this implies that more customers will leave the provider in option 2 compared with option 3. By determining the number of customers remaining with the service after each upgrade option, we can compute the retention rate and project the future profitability from our model as presented in Figure 5.13. With the cost of upgrades being equal, the outcome of our analytical model indicates option 3 is the more profitable option as long as the service charge for VPN is higher than the service charge for xDSL. By following our model and factor in the customer's expectations, satisfaction conditions and market dynamics, we are able to arrive at a much more informed upgrade decision using service profitability as an indicator.

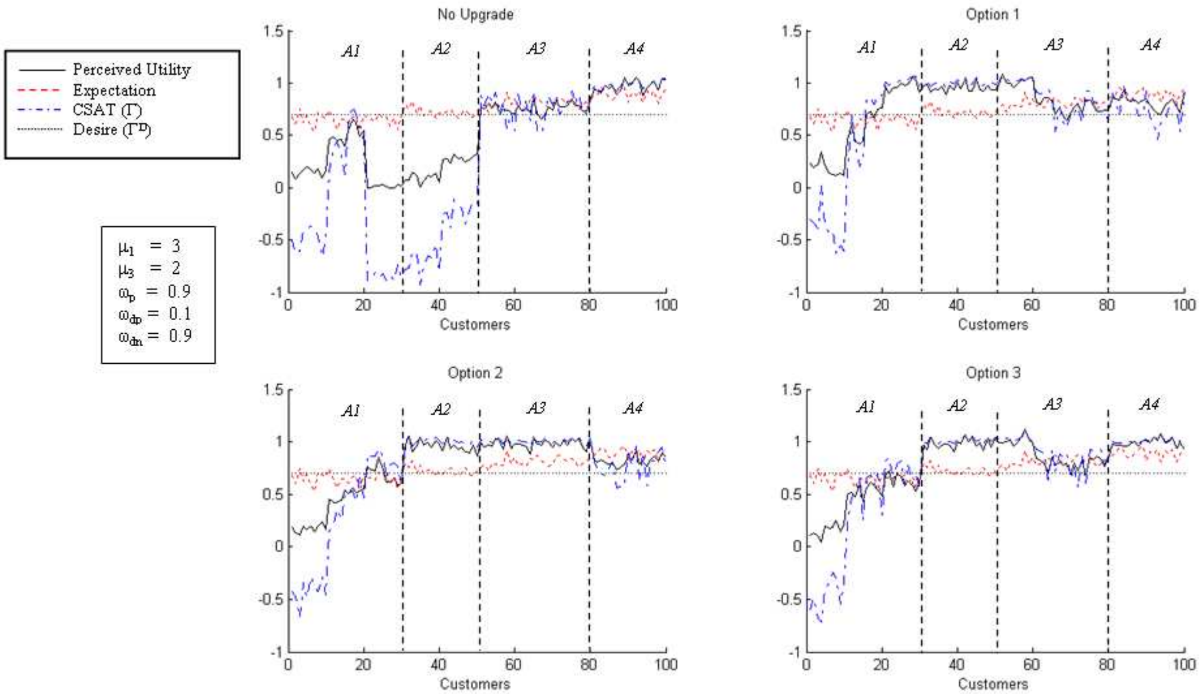


Figure 5.12: Customer Satisfaction of Network Upgrades

A commonly observed principle in market science states that service profitability is maximized with respect to customer's satisfiability. Rather than satisfy each customer, a business should strive to satisfy each satisfiable customer, and only if it is profitable to do so [135]. This phenomenon is manifested in our model, as illustrated in Figure 5.14. The top graph shows the computation of Γ for upgrade option 2, while the bottom graph shows the same computation except with raised ω_p value for the VPN customers. It suggests that if the VPN customers are difficult to satisfy (i.e. $\omega_p = 0.75$), then an upgrade option maybe ineffective despite improved service utility.

The long term interactions among service performance, customer experience and future expectation is a well studied topic in market science and economic psychology. Our model effectively captures many of their key observations. Consider a customer who has stayed

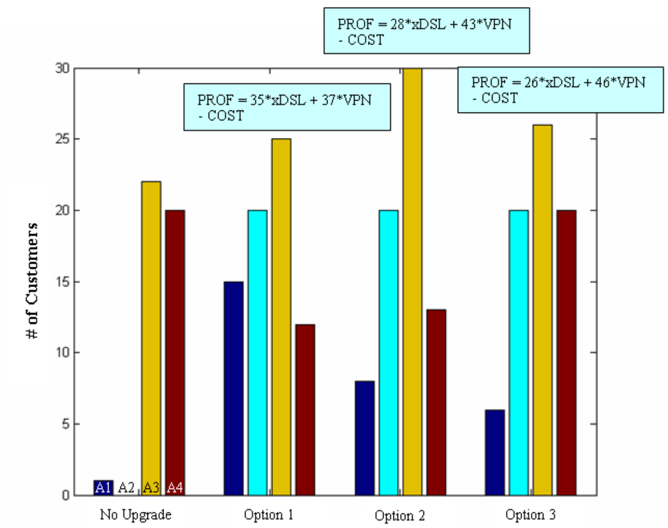


Figure 5.13: Customer Retention of Different Upgrade Options

with the service provider for 20 evaluation periods and has experienced consistent service performance, we subject the customer to low and inconsistent service performance for the next 50 evaluation periods and trace his/her service expectations (Figure 5.15) obtained from our expectation update process. The parameters β_L and β_G are general values taken based on our discussion in Section 5.5.3, and κ_a is set to 0 (i.e. no assimilation) for simplicity. As shown in Figure 5.15a, in the short term (first 7 iterations), the customer's future expectations are not significantly influenced by perceived utilities as the customer has experience with consistent service delivery in the past. However, the customer gradually loses confidence with the service (iterations 8 to 17) and expectations become heavily dependent on short term perceived utilities. This trend confirms with the observations on expectation and customer experience [136][137]. When a customer is dissatisfied due to poor service performance in the short term, an experienced customer (whose future expectation is not significantly reduced) is more likely to be loyal than an inexperienced customer. The works on expectation further suggest that when customer perceives disconfirmation, the degree of adjustment to expectation is determined by the uniqueness of the event and the strength of previous expectation. In the first few iterations of our illustrated

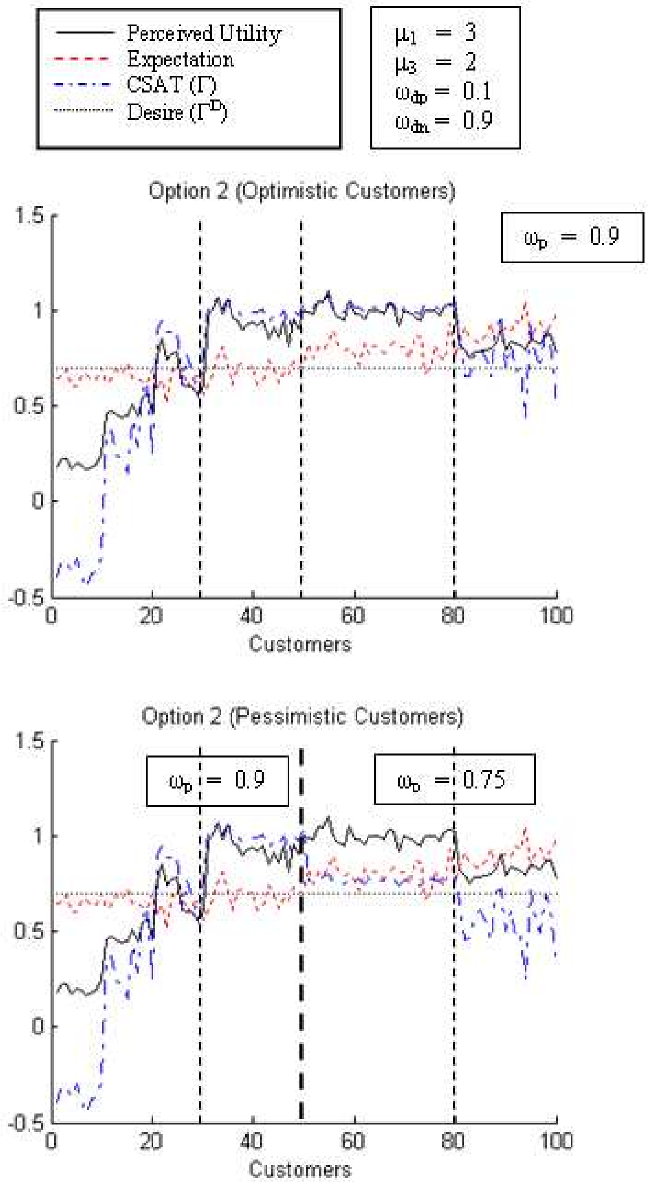


Figure 5.14: Effect of Varied ω_p

case (Figure 5.15a), the impact of disconfirmation on expectation adjustment is low. As the occurrence of disconfirmation increases, its impact is significantly more severe. The parameter β_M controls the weight of current disconfirmation on expectation. A higher β_M indicates a lower strength of the past expectation. In Figure 5.15b where β_M is higher, the impact of disconfirmation on expectation is significantly more severe even in the presence of long past experience. It is apparent that the interaction among expectation, performance, and customer satisfaction is a significant factor influencing the service profitability of SP operations and should be considered in the network upgrade decision process. The trends captured by our model integrate such factors in the network upgrade decision process.

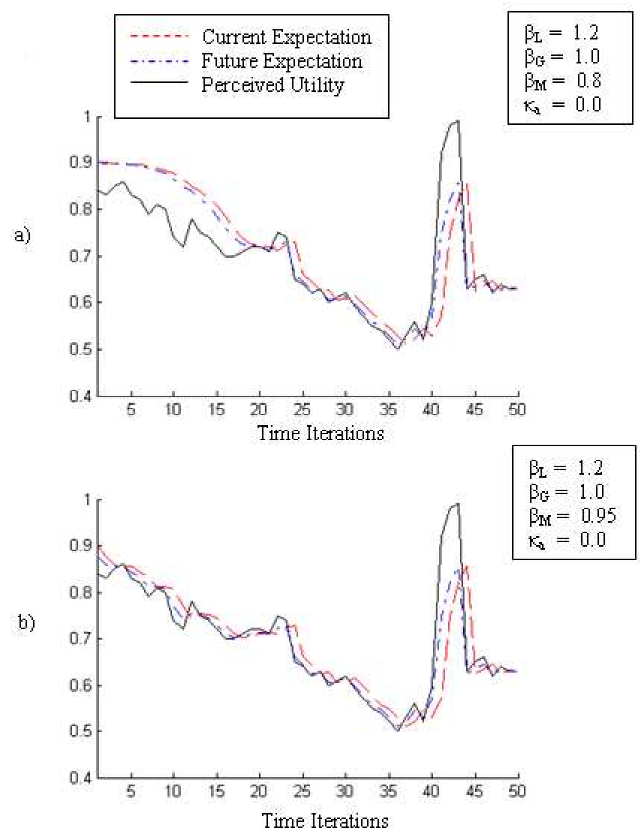


Figure 5.15: Experience and Long Term Service Expectation

Finally, we examine how varying market conditions can affect service profitability as presented in our model. Regarding the aforementioned three upgrade options, suppose the current consumer market size is 100 each for the xDSL and the VPN service. Furthermore, suppose the VPN market is fully saturated while the xDSL market is estimated to grow by 80 customers (in practice, this value is projected by the Bass model). Figure 5.16 illustrates the customer populations and service profitabilities for the three upgrade options. Compared with Figure 5.13, our model shows that the aggressive xDSL strategy (option 1) is able to attract more xDSL customers by pushing for better service performance, and hence better future expectation. Thus, depending on the actual earning difference of VPN service over xDSL service, our model may evaluate option 1 as the more profitable option.

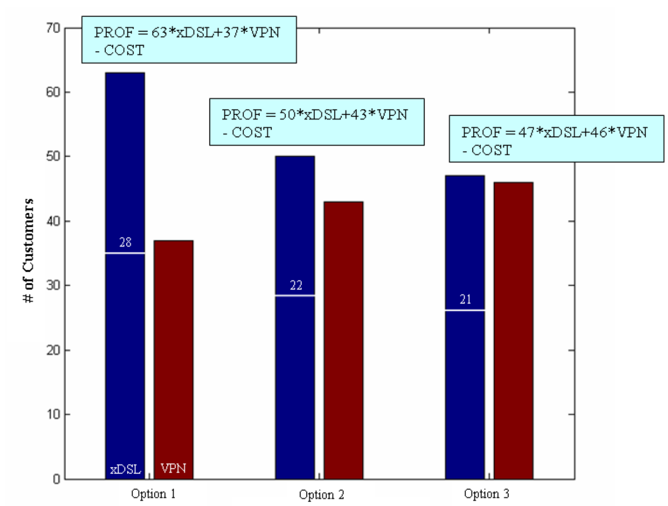


Figure 5.16: Customer Population for Different Upgrade Options

In the second simulation case, we conduct a more detailed discussion on how network performance influences customer satisfaction and show an example of how our model could be used in practice. The simulation setup depicts a typical regional service provider network. Three comparative performance analysis of the network infrastructure are presented, each from a different view: link utilization, QoS performance of customer flows, and customer satisfaction. Figure 5.17 shows the regional service provider network, simulated in NS2. The typical access, transit and core network topology is recreated. The links in this

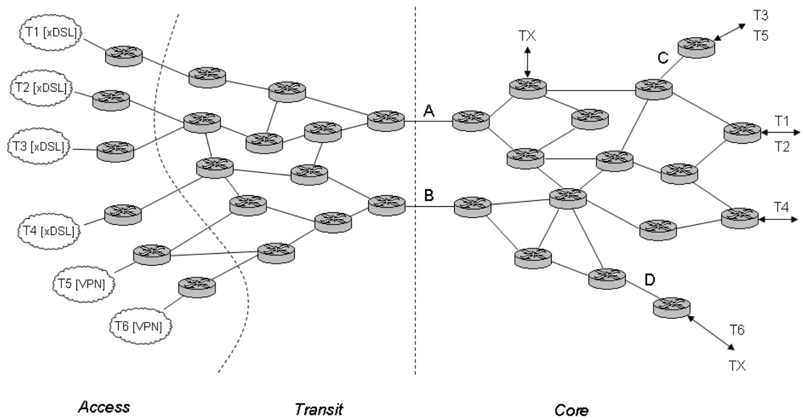


Figure 5.17: Simulated topology of a regional service provider network

simulated network are identical in characteristic to the first simulation study. Six customer groups and one transit traffic from a peer provider are studied. Each customer group has a mixture of service types and customer access times with daily traffic shapes similar to Figure 5.11, we track a representative customer flow from each population assuming they are a subscriber of either the xDSL or VPN service (noted between brackets in Figure 5.17) and analyze their behaviors under various conditions. The traffic exchange between the transit and core network is facilitated with two network links A and B. The customer flows from each service population are modelled as an aggregate Pareto flow from customer access to their respective traffic exchange point at the edge of the core network. The flows are routed through least joint paths. As in real world networks, the flows tend to merge around the various traffic exchange points, forming potential bottleneck links. In this simulation study, four such bottleneck links exist: A, B, C and D. Figure 5.18 illustrates the utilization of each link over 24 hr. period where link utilization measures are taken every minute. The average link utilizations of the entire day are also presented. In practice, link utilization is often used as an indicator for link upgrades or traffic re-engineering. However, as shown on Figure 5.18, it conveys no information as to the impact of congestive links on the performance of customer flows.

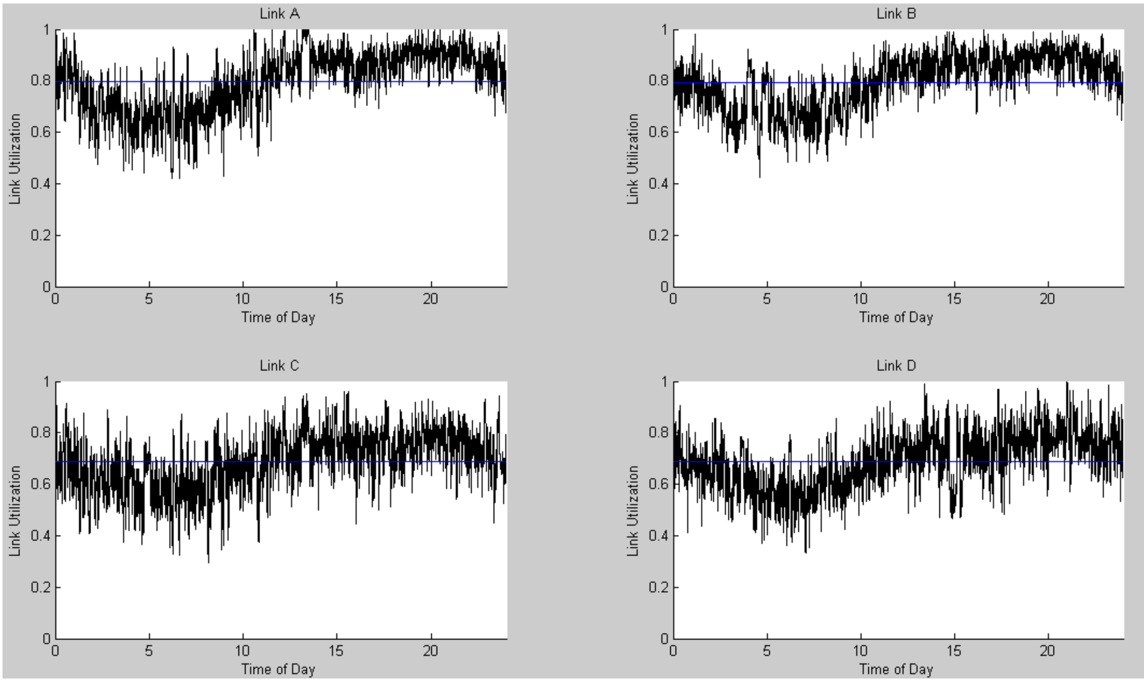


Figure 5.18: Link Utilization over 24 hr. Period

To analyze the impact of the network utilization on customer flows, representative customer flow tracking is conducted. In this case, we trace a representative customer flow from each customer population. A representative xDSL customer is traced in each of the population T1 to T4 and a representative VPN customer is traced in each of the population population T5 and T6. The xDSL customers are offered 330 Kbps (maximum throughput) service while the VPN customers are offered 680 Kbps service. Each traced flow is modelled as FTP over TCP in the simulation and Figure 5.19 shows the application level throughput measured over a 24 hr. period at five minutes sampling intervals. Delay is not monitored in this case because round trip delays within regional network seldom exceed application requirements. From the throughput trace, it is apparent that the congestion at link A and B during prime time of the day causes significant impact on customer flows. Comparing the throughput of T1 to T4, T4 seems the least impacted because link A is the only bottleneck link along the flow and trace from T4 has the least shared path with other

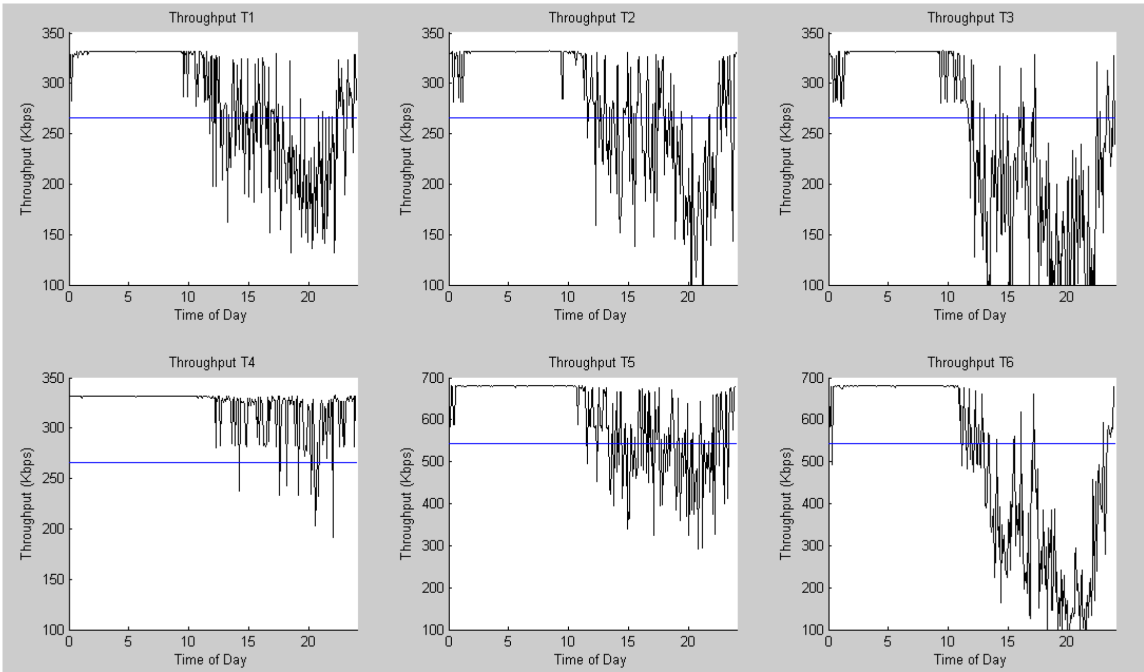


Figure 5.19: Throughput Performance of Representative Customer Flows

flows. Comparing T5 and T6, T6 fares significantly worse since in addition to the bottleneck at link B, its traffic also shares link D with transit traffic TX from a peer provider. Upgrade link A and link B appears to be imminent from this analysis. Figure 5.20 presents the throughput trace after link A and B are upgraded.

The link upgrades produce significant improvement over all the customer traces. However, the traces from T3, T5 and T6 still indicate potential problems especially if T5 and T6 have strong mix of VPN customers over xDSL customers. Besides conducting link upgrades which are cost prohibitive, resource provisioning mechanisms such as service differentiation and network dimensioning could be conducted. In this case, we create service differentiation across link C and D into premium and standard classes (60% and 40% of the link capacity is dimensioned for each class respectively). Traffic from T5 and T6 is thus given precedence over traffic from T3 and TX. Figure 5.21 shows the result of such

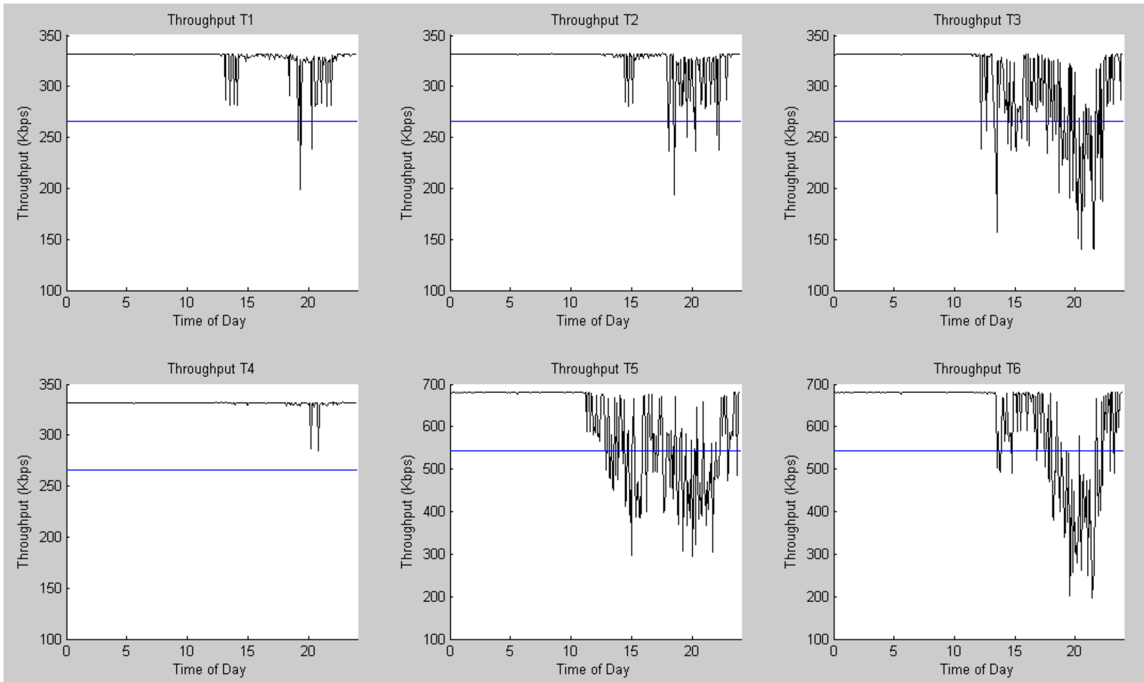


Figure 5.20: Throughput Performance After Link A and B Upgrade

dimensioning. It is apparent that the analysis resulting from tracing customer flows yields sensible network upgrade and planning strategies that maximize the performance of the customer flows that hopefully lead to better revenue generation.

As demonstrated in this chapter, improvement in network performance does not produce proportional improvement in customer satisfaction. In analyzing the customer satisfaction of the above trace traffics under varied customer conditions, we study the effect of these enhancement on customers. Figure 5.22 presents the customer satisfaction under different customer access patterns and QoS sensitivity. For QoS-insensitive traffic (e.g. FTP and P2P) the perceived utility is computed as the ratio between obtained throughput over maximum throughput. For QoS-sensitive traffic (e.g. multimedia traffic), throughput of 266 kpbs (for xDSL) and 544 kpbs (for VPN) are used as the defective thresholds, corresponding to roughly 80% of the maximum throughput. These thresholds are also depicted

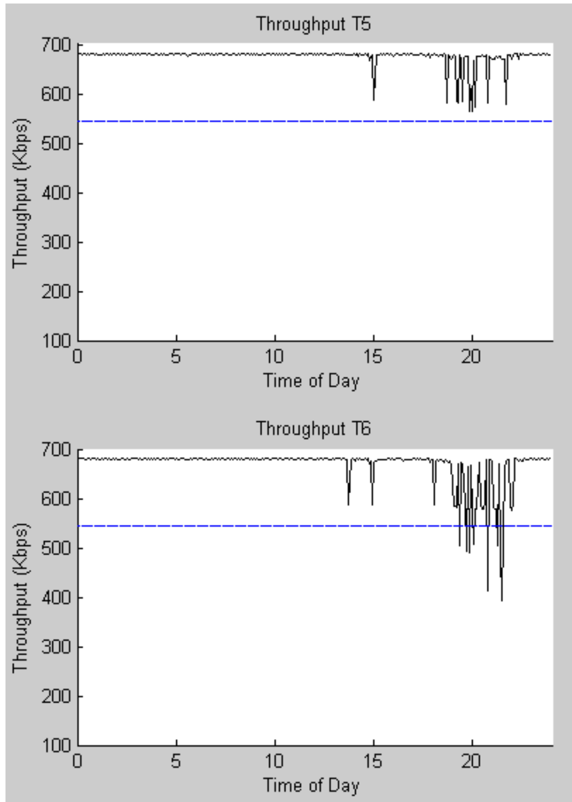


Figure 5.21: Throughput Performance After Link C and D Dimensioning

in Figures 5.19, 5.20 and 5.21. Customer satisfaction is computed with the same modelling parameters as used in the previous simulation setup with customer expectation set at 0.8. As illustrated in Figure 5.22, the raw computation of customer satisfaction could yield negative values. In practice, these negative values should be set to 0 to obtain the normalized value of Γ, nevertheless they are left here for comparison. The first three set of graphs consider xDSL customers from population T1 to T4. Observe that for customers that are QoS-insensitive and access the network 24 hours a day (representation of the permanent P2P population often prevalent in xDSL service), performing link upgrades is of little consequence. This category of customers is satisfied as long as their achievable daily average throughput remains reasonable. However, for the other xDSL users that are the

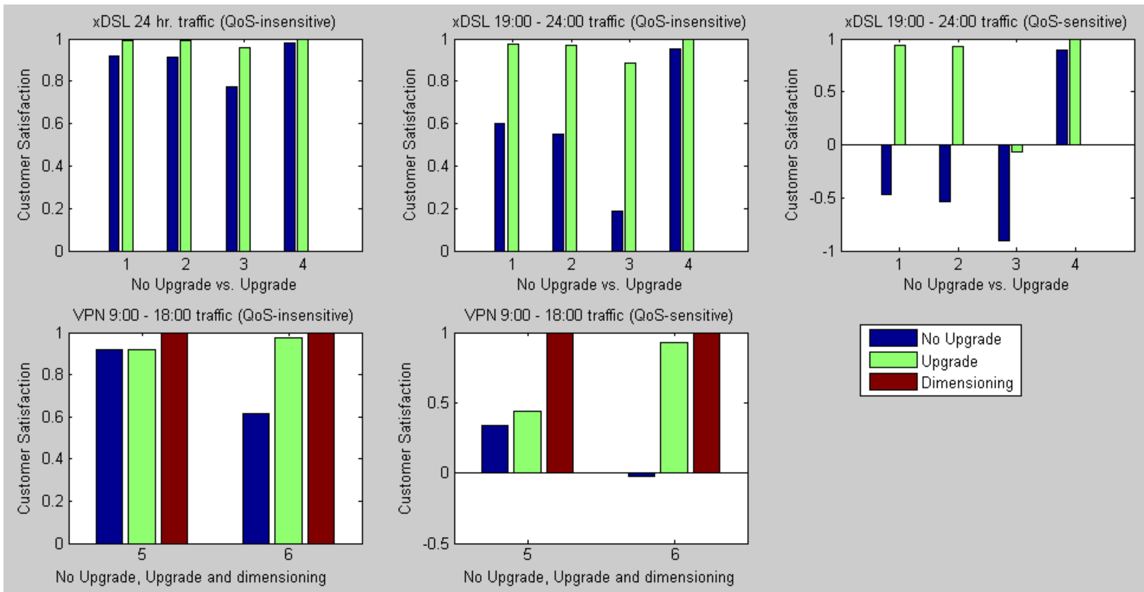


Figure 5.22: Effect of Upgrades and Dimensioning on Customer Satisfaction

bulk of "prime time" traffic, their satisfaction is severely impacted by link congestion and hence they benefit the most from link upgrade. Furthermore note that because T4 does not access the same transit-core link as T1 to T3. It was not significantly impacted by prime time traffic as the others. Hence from the analysis of the xDSL customer satisfaction, it seems that upgrade link A is quite effective given a large mix of prime time xDSL users in population T1 to T3 (which should be the case in practice). For the VPN customers, their access times are generally during business hours. For QoS-insensitive customers, upgrade link B only improves the performance of T6 somewhat, while service differentiation does not yield any visible result. For QoS-sensitive customers, The link upgrade and service differentiation strategies creates very different customer responses. It illuminates a prevailing theory in our model: customer satisfaction is a subjective, comparative evaluation between perception and expectation. In the case of T5, performing network upgrade alone does not raise the customer's received performance to a level that meets the customer's expectation and hence despite the actual increase in performance, the customer perceives very little improvement in satisfaction. In the case of T6, the improvement over performance as the result of link upgrade already meets the customer's expectation, conducing service differentiation in addition does not significantly influence the customer's opinion of the service. Our analysis illustrates that service and network planning strategies should be made with respect to the particularities of network infrastructure, service environment and the customer characteristics.

5.7 Conclusion

In this chapter, a comprehensive framework has been presented that can relate network performance to customer satisfaction and business profitability in service-driven networks. As we have shown, there exists a wealth of established models from economics and market science which are appropriate for analyzing the service performance and its effect on customer satisfaction. The models we have obtained under the proposed framework are general and can be adapted to the service specifics. Accordingly, two issues arise: how to choose the right parameter settings and how to fit the models in practice? We briefly discuss our perspectives below.

Given the intricacies among the modeling parameters, it is imperative to conduct validation and tuning over time in real world service operations. In market science, when faced with complex models and hypotheses, most of the validation work is carried out over large data sets across long periods of time, where statistical analysis is often helpful in deducing trends and linkages among metrics. A similar approach is appropriate to the tuning of our model, in conjunction with simulation studies and numerical analysis. Simulation studies and numerical analysis can shed some light on the sensitivity of the modeling parameters especially those closely related to network performance. However, market data such as customer satisfaction and service turnover rate is recorded during regular business operations and hence difficult to obtain. Carrying out this validation exercise is invaluable to the service providers. Experimentation with the model in the network planning and upgrade processes not only provides additional forecasting capability to the planners but also yields outputs (e.g. customer turnover rate) that are comparable with future data. Through an iterative validation and parameter tuning process, the model and its parameters can be adjusted and refined over time to suite the particular customer base, service condition, and market environment of the service provider.

The validity of a model and its derivable results are inherently dependent on the availability of its input data and parameters, and the correctness of the theories that underpin the model. Care was taken in constructing the models only based on parameters and input data that are tractable, and in many cases known to be available to the service providers. Among the many theories available in market science, we only included the most fundamental and tried ones. The applicability of these theories is confirmed in market studies across many service industries over the past decades. The mathematical forms that we have derived from these theories are intentionally designed to be simple, with flexible parameters to ensure that the model is tunable to the particularities of the service provider and market scenario.

Chapter 6

Conclusion

In this thesis, we have put forth a perspective on service management in current and future networks. We framed the research scope under the notion of service-driven networking and defined services as end-to-end networking constructs with specific quality requirements. This view is similar to the circuit-based service view of the traditional telecommunication networks, except that its realization in the IP world poses significant challenges. Thus, we have directed the investigation on two fundamental issues: service quality assurance and management integration. As we have shown, the management concerns in service-driven networks must be addressed from both the service perspective and the network perspective, resulting in necessary mutual collaboration and inherently distributed management designs. The following sections summarize the three topics presented in this thesis:

Autonomic service composition and adaptation in multi-domain networks: the problem of service path composition and adaptation is formalized based on domain graph construction. The objective is to ensure that the end-to-end performance requirements are satisfied (and maintained during runtime) and the cost of the composition is minimum. The problem is reduced to k-MCOP and we have shown the inadequacies of classic k-MCOP solutions in the service-driven networking context. New service composition and adaptation heuristics were thus developed and simulation results showed that the proposed solution has very high probability of finding the optimal or near optimal solution with path cost much lower than those generated by the known k-MCOP solutions. Furthermore, our solution supports the open bi-participation of service-driven networking, wherein both the customers and the providers have control over the service composition process. It also supports quality assurance over best-effort or service differentiation based networks and is hence suitable for deployment in the current Internet. Two interesting problems related to this topic are left open. One, the problem of path composition becomes quite difficult in the context of specific applications such as web service composites, where an end-to-end QoS assured path must be found traversing a set of specific application-level service components in specific order. Two, the path composition and adaptation requires knowledge of the domain graph as well as the timely reporting of QoS information. The question of how such knowledge can be supplied, updated and obtained are unresolved today.

Self-stabilization: System stability is a fundamental issue in large uncoordinated multi-domain environments consisting of participants with divergent objectives (i.e. the application and the network providers). We have analyzed this problem from a game theory perspective on the interaction among overlays and underlays. By modeling the problem as a congestion game, we have shown that stability is possible when: 1) underlay management intervals are long enough to permit overlay convergence to a desirable stable state; 2) overlays have control over domain traversal; 3) overlay and underlay have similar objectives of congestion avoidance. As we have discussed, these conditions are realizable in practice. Through the concept of desirability and desirable transformation, we were able to provide a polynomial convergence bound on a class of congestion games that are PLS-complete, and have obtained a tunable price of anarchy. Furthermore, we have discussed some implementation issues and have shown some promising results through simulation studies. As a future work, it is interesting to apply the proposed model to specific network applications (e.g. IPTV) where multiple quality criteria must be considered. This would require nontrivial extensions to our game model, including the establishment of a mapping function from multi-criteria objectives to single potential value, and the introduction of player-specific preferences in the cost functions.

Service quality modeling: we have presented a market science approach to model the antecedent and consequence of service quality in service-driven networks. Our framework captures the intricate relationships among network performance, customer behavior, and market dynamics and is founded on theoretical and empirical studies from market science and economics. It involves a set of meaningful control parameters to model wide varieties of network service characteristics, customer attributes, and market conditions. Many of these complex relationships are captured and reflected in the models and are key determinants of service profitability. Through extensive simulation studies, we have shown how the proposed framework is used to analyze practical problems in regional networks through representative flow tracing and customer satisfaction analysis. As a future work, we would like to further validate and refine our approach based on real world service provider operations and market data. Such a study can be a long term process but would be highly beneficial to the network service industry.

Throughout this thesis we have shown that integrated network and service management is fundamental to the development of quality-assured services. This is a major challenge as there does not exist a single authority capable of and has the incentives to pursue both the network management objectives and the end-to-end service needs. There is, instead, an increasing number of stakeholders with intricate interactions both horizontally (peer service providers sharing the same network infrastructure, or jointly supporting a service), and vertically (network providers offering services in return for profit). The key principle in service-driven networking lies in understanding the characteristics of these interactions and providing the correct incentives to facilitate win-win collaborations.

Bibliography

[1] "Principles for a telecommunications management network (TMN)," CCITT M.3010, 1996.

[2] J. Chandrashekar, Z.-L. Zhang, Z. Duan, and Y. Hou, "Towards a service oriented internet," *IEICE Transactions on Communication*, vol. E89-B, no. 9, pp. 2292–2298, September 2006.

[3] M. P. Singh and M. N. Huhns, *Service-Oriented Computing*. John Wiley and Sons, 2005.

[4] R. Sterritt and M. Hinchey, "Why computer-based systems should be autonomic," in *Proceedings of IEEE International Conference and Workshops on the Engineering of Computer-Based Systems*, April 2005.

[5] J. Kephart and D. Chess, "The vision of autonomic computing," *IEEE Computer*, no. 1, pp. 41–50, 2003.

[6] Q. Mahmoud, Ed., *Cognitive Networks: Towards Self-Aware Networks*. John Wiley and Sons, September, 2007.

[7] J. Wroclawski, "The use of RSVP with IETF integrated services," RFC2210, September 1997.

[8] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," RFC2475, December 1998.

[9] T. Griffin, F. Shepherd, and G. Wilfong, “Stable paths problem and interdomain routing,” *IEEE/ACM Transactions on Networking*, vol. 10, no. 2, pp. 232–243, 2002.

[10] L. Subramanian, M. Caesar, C. Ee, M. Handley, M. Mao, S. Shenker, and I. Stoica, “HLP: A next generation inter-domain routing protocol,” in *ACM SIGCOMM'05*, August 2005, pp. 13–24.

[11] R. Morris, F. Kasshoek, H. Balakrishman, and D. Anderson, “Resilient overlay networks,” in *In Proceedings of Symposium of Operating System Principles (SOSP'01)*, 2001.

[12] Z. Duan, Z. Zhang, and Y. Hou, “Service overlay networks: SLAs, QoS, and bandwidth provisioning,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 6, 2003.

[13] X. Gu, K. Nahrstedt, R. N. Chang, and C. Ward, “QoS-assured service composition in managed service overlay networks,” in *Proceedings of the 23rd International Conference on Distributed Computing Systems*. ACM, May 2003.

[14] Y. Liu, H. Zhang, W. Gong, and D. Towsley, “On the interaction between overlay routing and underlay routing,” in *IEEE INFOCOM 2005*, vol. 4, 2005, pp. 2543–2553.

[15] S. Seetharaman, V. Hilt, M. Hofmann, and M. Ammar, “Preemptive strategies to improve routing performance of native and overlay layers,” in *IEEE INFOCOM*, 2007.

[16] J. Xiao and R. Boutaba, “Qos-aware service composition in large scale multi-domain networks,” in *Proceedings of the IEEE/IFIP Symposium on Integrated Network and System Management (IM'05)*, 2005.

[17] K. Lakshminarayanan, I. Stoica, and S. Shenker, “Routing as a service,” Technical Report, UCB/CSD-04-1327, UC Berkley, 2004.

[18] X. Yang, D. Clark, and A. Berger, “NIRA: A new inter-domain routing architecture,” *IEEE Transactions on Networking*, vol. 15, no. 4, pp. 775–787, December 2007.

[19] "X.701: Information technology - open system interconnection - system management overview," CCITT, Geneva, 1989.

[20] "X.710: Common management information service definition for CCITT applications," CCITT, Geneva, 1991.

[21] "X.720: Information technology - open systems interconnection - structure of management information: management information model," CCITT, Geneva, 1992.

[22] "Concise MIB definitions," March 1991, rFC 1212.

[23] R. Boutaba and J. Xiao, "Telecommunication network management," *Encyclopedia on Life Support Systems*, 2007.

[24] "Y.2001: General overview of NGN," ITU-T, Geneva, 2004.

[25] "Y.2011: General principles and general reference model for next generation network," ITU-T, Geneva, 2004.

[26] "Next generation operation support systems (NGOSS)," TM Forum, http://www.tmforum.org/TechnicalPrograms/NGOSS/1911/Home.html.

[27] "Telecom application map (TAM)," TM Forum, http://www.tmforum.org/BusinessSolutions/TelecomApplications/2322/Home.html.

[28] "IT infrastructure library," ITIL, http://www.itil-officialsite.com/home/home.asp.

[29] "Web service distributed management (WSDM), v.1.1," OASIS Standard, 2006.

[30] "Web services distributed management: management using web services (MUWS 1.1) part 1," OASIS Standard, 2006.

[31] "Web services distributed management: management using web services (MUWS 1.1) part 2," OASIS Standard, 2006.

[32] "Web service distributed management: management of web services (MOWS), v.1.1," OASIS Standard, 2006.

[33] "Business process execution language for web services v1.1," IBM, http://www.ibm.com/developerworks/library/specification/ws-bpel/.

[34] "Web service choreography interface (WSCI) v1.0," W3C, http://www.w3.org/TR/wsci/.

[35] C. Peltz, "Web service orchestration and choreography," *IEEE Computer*, vol. 36, no. 10, pp. 46–52, 2003.

[36] S. White, J. Hanson, and I. Whalley, "An architectural approach to autonomic computing," in *IEEE 1st International Conference on Autonomic Computing*, May 2004.

[37] D. Wile, "Patterns of self-management," in *ACM Workshop on Self-Healing Systems*, October 2004.

[38] M. Hawthorne and D. Perry, "Exploiting architectural prescriptions for self-managing, self-adaptive systems: A position paper," in *ACM Workshop on Self-Healing Systems*, October 2004.

[39] R. Bahati, M. Bauer, and E. Vieira, "Using policies to drive autonomic management," in *IEEE International Symposium on a WOrld of WIreless Mobile and Multimedia Networks*, June 2006.

[40] H. Liu and M. Parashar, "Accord: A programming framework for autonomic applications," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 36, no. 3, pp. 341–352, 2006.

[41] N. Samman and A. Karmouch, "An automated policy-based management framework for differentiated communication systems," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 12, pp. 2236–2247, December 2005.

[42] I. Aib and R. Boutaba, "On leveraging policy-based management for maximizing business profit," *IEEE Transactions on Network and Service Management*, vol. 4, no. 3, pp. 163–176, December 2007.

[43] S. Aiber, D. Gilat, and A. Landau, "Autonomic self-optimization according to business objectives," in *IEEE International Conference on Autonomic Computing*, May 2004.

[44] D. Gracanin, S. Bohner, and M. Hinchey, "Towards a model-driven architecture for autonomic systems," in *IEEE International Conference and Workshops on the Engineering of Computer-Based Systems*, May 2004.

[45] E. Lavinal, T. Desprats, and Y. Raynaud, "A generic multi-agent conceptual framework towards self-management," in *IEEE/IFIP Network Operations and Management Symposium*, April 2006.

[46] G. Tesauro, N. Jong, and R. Das, "A hybrid reinforcement learning approach to autonomic resource allocation," in *IEEE International Conference on Autonomic Computing*, June 2006.

[47] J. Dowling and V. Cahill, "Self-managed decentralised systems using k-components and collaborative reinforcement learning," in *ACM Workshop on Self-Healing Systems*, October 2004.

[48] P. Vienne and J. Sourrouille, "A middleware for autonomic QoS management based on learning," in *ACM Software Engineering and Middleware*, September 2005.

[49] Y. Diao, J. Hellerstein, and S. Parekh, "A control theory foundation for self-managing computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 12, pp. 2213–2222, 2005.

[50] H. Tianfield, "Multi-agent based autonomic architecture for network management," in *IEEE International Conference on Industrial Informatics*, August 2003.

[51] W. Wang and B. Li, "Market-based self-optimization for autonomic service overlay networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 12, pp. 2320–2332, December 2005.

[52] R. Anthony, "Emergence: a paradigm for robust and scalable distributed applications," in *IEEE International Conference on Autonomic Computing*, May 2004.

[53] J. Werfel and R. Nagpal, "Extended stigmergy in collective construction," *IEEE Intelligent Systems*, vol. 21, no. 2, pp. 20–28, March 2006.

[54] B. Raman, S. Agarwal, Y. Chen, M. Caesar, W. Cui, P. Johansson, K. Lai, T. Lavian, S. Machiraju, Z. Morley-Mao, G. Porter, T. Roscoe, M. Seshadri, J. S. Shih, K. Sklower, L. Subramanian, T. Suzuki, S. Zhuang, A. D. Joseph, R. H. Katz, and I. Stoica, "The SAHARA model for service composition across multiple providers," in *Proceedings of the First International Conference on Pervasive Computing*. ACM, August 2002.

[55] L. Zeng, B. Benatallah, A. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-aware middleware for web service composition," *IEEE Transactions on Software Engineering*, vol. 30, no. 5, pp. 311–327, May 2004.

[56] S. E. Czerwinski, B. Y. Zhao, T. D. Hodes, A. D. Joseph, and R. H. Katz, "An architecture for a secure service discovery service," in *Proceedings of Fifth Annual International Conference on Mobile Computing and Networks (MobiCom '99)*. IEEE, August 1999.

[57] "Universal description, discovery and integration technical white paper," UDDI.org, September 2000.

[58] W. Hoschek, "The web service discovery architecture," in *Proceedings of ACM/IEEE Conference on Supercomputing 2002*, 2002.

[59] Y. Rekhter and P. Gross, "Application of the border gateway protocol in the internet," RFC1772, March 1995.

[60] F. Leymann, "Web service flow language (WSFL 1.0)," IBM Software Group, May 2001, http://www-3.ibm.com/software/ solutions/webservices/pdf/WSFL.pdf.

[61] T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana, "Business process execution language for web service v1.1," IBM Technical White Paper, May 2003, http://www.ibm.com/developerworks/library/ws-bpel/.

[62] A. Keller, G. Kar, H. Ludwig, A. Dan, and J. L. Hellerstein, "Managing dynamic services: A contract based approach to a conceptual architecture," in *Proceedings of IEEE/IFIP Network Operations and Management Symposium 2002 (NOMS2002).* IEEE/IFIP, April 2002.

[63] M. Salle and C. Bartolini, "Management by contract," in *Proceedings of IEEE/IFIP Network Operations and Management Symposium 2004 (NOMS2004).* IEEE/IFIP, April 2004.

[64] G. Piccinelli, C. Preist, and C. Bartolini, "E-service composition: Supporting dynamic definition of process-oriented negotiation parameters," in *Proceedings of 12th International Workshop on Database and Expert Systems Applications.* IEEE, September 2001.

[65] W. Rhee, J. Lee, M. Yang, I. Lee, J. Yu, and S. Kim, "Dynamic provisioning mechanism for heterogeneous QoS guarantee in differentiated service networks," in *Proceedings of IEEE International Conference on Communications 2003 (ICC03)*, vol. 3. IEEE, May 2003, pp. 1912–1916.

[66] Z. Wang and J. Crowcroft, "QoS routing for supporting resource reservation," *IEEE Journal on Selected Areas in Communication*, 1996.

[67] S. Chen and K. Nahrstedt, "On finding multi-constrained paths," in *Proceedings of IEEE International Conference on Communications (ICC98)*, vol. 2. IEEE, June 1998, pp. 874–879.

[68] X. Yuan, "Heuristic algorithms for multiconstrained quality-of-service routing," *IEEE/ACM Transactions on Networking*, vol. 10, no. 2, April 2002.

[69] H. D. Neve and P. V. Mieghem, "A multiple quality of service routing algorithm for PNNI," in *Proceedings of the ATM Workshop.* IEEE, May 1998, pp. 324–328.

[70] P. V. Mieghem, H. D. Neve, and F. Kuipers, "Hop-by-hop quality of service routing," *Computer Networks*, no. 37, pp. 407–423, 2001.

[71] T. Korkmaz and M. Krunz, "Multi-constrained optimal path selection," in *Proceedings of IEEE INFOCOM 2001*, vol. 2. IEEE, April 2001, pp. 834–843.

[72] F. A. Kuipers, T. Korkmaz, M. Krunz, and P. V. Meighem, "Performance evaluation of constraint-based path selection," *IEEE Network*, vol. 18, no. 5, pp. 16–23, September/October 2004.

[73] F. A. Kuipers and P. F. A. V. Mieghem, "Conditions that impact the complexity of QoS routing," *IEEE/ACM Transactions on Networking*, vol. 13, no. 4, August 2005.

[74] A. Orda, R. Rom, and N. Shimkin, "Competitive routing in multiuser communication networks," *IEEE/ACM Transactions on Networking*, vol. 1, no. 5, pp. 510–522, October 1993.

[75] E. Altman, T. Basar, T. Jimenez, and N. Shimkin, "Competitive routing in networks with polynomial cost," in *IEEE INFOCOM 2000*, vol. 1586-1593, 2000.

[76] I. Milchtaich, "Congestion games with player-specific payoff functions," *Games and Economic Behavior*, vol. 13, pp. 111–124, 1996.

[77] A. Fabrikant, C. Papadimitriou, and K. Talwar, "The complexity of pure Nash equilibria," in *36th ACM Symposium on Theory of Computing (STOC)*, 2004, pp. 604–612.

[78] D. Monderer and L. Shapley, "Potential games," *Games and Economics Behavior*, vol. 14, pp. 124–143, 1996.

[79] E. Koutsoupias and C. Papadimitriou, "Worst-case equilibria," in *Symposium on Theoretical Aspects in Computer Science (STACS)*, 1999.

[80] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Experiences applying game theory to system design," in *ACM SIGCOMM Workshops*, 2004.

[81] E. Dijkstra, "Self-stablization in spite of distributed control," *Communications of the ACM*, vol. 17, no. 11, pp. 643–644, 1974.

[82] M. Schneider, "Self-stabilization," *ACM Computing Surveys*, vol. 25, no. 1, pp. 45–67, March 1993.

[83] L. Qiu, Y. R. Yang, Y. Zhang, and S. Shenker, "On selfish routing in internet-like environments," *IEEE/ACM Transactions on Networking*, vol. 14, no. 4, pp. 725–739, August 2006.

[84] R. Keralapura, C.-N. Chuah, N. Taft, and G. Iannaccone, "Race conditions in coexisting overlay networks," *IEEE Transactions on Networking*, vol. 16, no. 1, February 2008.

[85] H. Zhang, J. Kurose, and D. Towsley, "Can an overlay compensate for a careless underlay?" in *IEEE INFOCOM 2006*, April 2006.

[86] H. Yaiche, R. R. Mazumdar, and C. Rosenberg, "A game theoretic framework for bandwidth allocation and pricing in broadband networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 667–678, October 2000.

[87] R. Banner and A. Orda, "Bottleneck routing games in communication networks," in *IEEE INFOCOM 2006*, April 2006.

[88] R. Rosenthal, "A class of games possessing pure-strategy Nash equilibria," *International Journal of Game Theory*, vol. 2, pp. 65–67, 1973.

[89] P. Goldberg, "Bounds for the convergence rate of randomized local search in a multiplayer load-balancing game," in *25th ACM Symposium on Principles of Distributed Computing (PODC)*, 2004, pp. 131–140.

[90] E. Even-Dar and Y. Mansour, "Fast convergence of selfish rerouting," in *16th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2005, pp. 772–781.

[91] G. Christodoulou, V. Mirrokni, and A. Sidiropoulos, "Convergence and approximation in potential games," in *Symposium on Theoretical Aspects in Computer Science (STACS)*, 2006.

[92] S. Chien and A. Sinclair, "Convergence to approximate Nash equilibria in congestion games," in *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007.

[93] C. Alvarez and G. Frances, "The complexity of pure nash equilibria in weighted max-congestion games," Technical Report, Universitat Politecnica de Catalunya, http://hdl.handle.net/2099.1/5589, 2008.

[94] C. Papadimitriou, "Algorithms, games and the Internet," in *33rd ACM Symposium on Theory of Computing (STOC)*, 2001, pp. 749–753.

[95] E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden, "The price of stability for network design with fair cost allocation," in *IEEE Symposium on Foundations of Computer Science*, 2004.

[96] T. Roughgarden and E. Tardos, "Bounding the inefficiency of equilibria in nonatomic congestion games," *Games and Economic Behavior*, vol. 47, pp. 389–403, 2004.

[97] B. Awerbuch, Y. Azar, and A. Epstein, "The price of routing unsplittable flow," in *37th ACM Symposium on Theory of Computing (STOC)*, 2005, pp. 57–66.

[98] T. Roughgarden and E. Tardos, "How bad is selfish routing?" in *41st Annual Symposium on Foundations of Computer Science*, November 2000.

[99] T. Roughgarden, "Selfish routing with atomic players," in *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2005.

[100] D. J. Farber and K. C. Larson, "The system architecture of the distributed computing system - the communications network," in *Symposium on Computer Networks*, 1972.

[101] "A simple network management protocol (SNMP)," May 1990, rFC 1157.

[102] "Cisco netflow," http://www.cisco.com/warp/public/732/Tech/nmp/netflow/index.shtml.

[103] L. L. Tung, "Service quality and perceived value's impact on satisfaction, intention and usage of short message service (SMS)," *Information Systems Frontiers*, vol. 6, no. 4, pp. 353–368, 2004.

[104] Y. Wang, H. P. Lo, and Y. Yang, "An integrated framework for service quality, customer value, satisfaction: Evidence from China's telecommunication industry," *Information Systems Frontiers*, vol. 6, no. 4, pp. 325–340, 2004.

[105] B. Leisen and C. Vance, "Cross-national assessment of service quality in the telecommunication industry: Evidence from the USA and Germany," *Managing Service Quality*, vol. 11, no. 5, pp. 307–317, 2001.

[106] P. Flynn, K. Curran, and T. Lunney, "A decision support system for telecommunications," *International Journal of Network Management*, no. 12, 2002.

[107] S. Khan, K. F. Li, and E. G. Manning, "The utility model for adaptive multimedia systems," in *International Conference on Multimedia Modeling*, Nov. 1997.

[108] Y. d'Halluin, P. Forsyth, and K. R. Vetzal, "Managing capacity for telecommunications networks under uncertainty," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, August 2002.

[109] S. Jagannathan, J. Altmann, and L. Rhodes, "A revenue-based model for making resource investment decisions in IP networks," in *IEEE/IFIP Symposium on Integrated Network and System Management (IM)*, 2003.

[110] A. Bouch, M. Sasse, and H. G. DeMeer, "Of packets and people: A user-centered approach to quality of service," in *Proceedings of IWQoS*, June 2000.

[111] J. Xiao and R. Boutaba, "Customer-centric network upgrade strategy: Maximizing investment benefits for enhanced service quality," in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2004.

[112] D. Mitra, K. G. Ramakrishnan, and Q. Wang, "Combined economic modeling and traffic engineering: Joint optimization of pricing and routing in multi-service networks," in *17th International Teletraffic Congress (ITC-7)*, 2001.

[113] V. A. Siris, D. J. Songhurst, G. D. Stamoulis, and M. Stoer, "Usage-based charging using effective bandwidths: Studies and reality," in *16th International Teletraffic Congress (ITC-16)*, 1999.

[114] C. Courcoubetis, F. P. Kelly, V. A. Siris, and R. Weber, "A study of simple usage-based charging schemes for broadband networks," *Telecommunication Systems*, vol. 15, no. 3-4, pp. 323–343, 2000.

[115] S. Shakkottai and R. Srikant, "Economics of network pricing with multiple ISPs," in *IEEE INFOCOM*, 2005.

[116] R. L. Oliver, "Effects of expectations and disconformation on postexposure product evaluations," *Journal of Applied Psychology*, vol. 62, April 1977.

[117] ——, "A cognitive model of the antecedents and conseuqences of satisfaction decisions," *Journal of Marketing Research*, vol. 17, November 1980.

[118] R. L. Oliver and W. DeSarbo, "Response determinants in satisfaction judgements," *Journal of Consumer Research*, vol. 14, March 1988.

[119] E. W. Anderson and M. W. Sullivan, "The antecedents and consequences of customer satisfaction of firms," *Marketing Science*, vol. 12, no. 2, 1993.

[120] R. N. Bolton, "A dynamic model for the duration of the customer's relationship with a continuous service provider: The role of satisfaction," *Marketing Science*, vol. 17, no. 1, 1998.

[121] A. Parasuraman, V. A. Zeithaml, and L. L. Berry, "A conceptual model of service quality and its implications for future research," *Journal of Marketing*, vol. 49, pp. 41–50, 1985.

[122] ——, "SERVQUAL: A multiple-item scale for measuring consumer perceptions of service quality," *Journal of Retailing*, vol. 64, no. 1, pp. 12–40, 1988.

[123] ——, "Refinement and reassessment of the SERVQUAL scale," *Journal of Retailing*, vol. 67, no. 4, pp. 420–450, 1991.

[124] A. Deaton and J. Muellbauer, *Economics and Consumer Behavior.* Cambridge University Press, Cambridge, England, 1980.

[125] P. Slovic and S. Lichenstein, "Comparison of Bayesian and regression approaches to the study of information processing in judgement," *Organizational Behavior and Human Performance*, vol. 6, pp. 649–744, 1971.

[126] W. Boulding, R. Staelin, A. Kalra, and V. Ziethaml, "A dynamic process model of service quality: From expectations to behavioral intentions," *Journal of Marketing Research*, pp. 7–27, Feburary 1993.

[127] F. M. Bass, "New product growth model for consumer durables," *Management Science*, vol. 15, p. 215, 1969.

[128] P. A. Dabholkar, "Customer satisfaction and service quality: Two constructs or one?" in *Enchancing Knowledge Development in Marketing*, D. W. Cravens and P. R. Dickson, Eds. American Marketing Association, Chicago, IL, 1993, pp. 10–18.

[129] TeleManagement Forum, "SLA handbook solution suite v2.0 volume 2: Concepts and principles," 2005.

[130] A. Tversky and D. Kahneman, "Availability: A heuristic for judging frequency and probability," *Cognitive Psychology*, vol. 4, pp. 207–232, 1973.

[131] R. T. Rust, J. Inman, and J. Jia, "Customer expectation updating: A dynamic model, theoretical implications and empirical evidence," Vanderbilt University, Nashville, TN," Working Paper, 1997.

[132] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis*, 2nd ed. Springer-Verlag, New York, NY, 1985.

[133] "Implementation strategies for advanced access networks," EUROSCOM Project P614 Main Report, 1999.

[134] "The network simulator NS2," http://www.isi.edu/nsnam/ns/.

[135] R. Hallowell, "The relationships of customer satisfaction, customer loyalty, and profitability: An emprirical study," *International Journal of Service Industry Management*, vol. 7, no. 4, pp. 27–42, 1996.

[136] R. B. Woodruff, E. R. Cadotte, and R. L. Jenkins, "Modeling consumer satisfaction processes using experience-based norms," *Journal of Marketing Research*, vol. 20, no. 3, 1983.

[137] R. L. Oliver and R. S. Winer, "A framework for the formation and structure of consumer expectations: Review and propositions," *Journal of Economic Psychology*, vol. 8, no. 4, 1987.