저작자표시•비영리•변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer
Thesis for the Degree of Doctor of Philosophy

Approximation and Matching Game Approaches for Network Service Chain Placement on Cloud Datacenters

Pham Chuan

Department of Computer Science and Engineering
Graduate School
Kyung Hee University
Seoul, Korea

August, 2017
Approximation and Matching Game Approaches for Network Service Chain Placement on Cloud Datacenters

Pham Chuan

Department of Computer Science and Engineering
Graduate School
Kyung Hee University
Seoul, Korea

August, 2017
Approximation and Matching Game Approaches for Network Service Chain Placement on Cloud Datacenters

by

Pham Chuan

Advised by

Professor Choong Seon Hong, Ph.D.

Submitted to the Department of Computer Science and Engineering
and the Faculty of the Graduate School of
Kyung Hee University in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

Dissertation Committee:

Professor Eui-Nam Huh, Ph.D (Committee Head),
Professor Sung-Won Lee, Ph.D,
Professor Hyun-Dong Shin, Ph.D
Professor Ki-Hyung Kim, Ph.D
Professor Choong Seon Hong, Ph.D
Acknowledgment

First and foremost, I am truly grateful to my supervisor, Prof. Choong Seon Hong, Networking Lab, Department of Computer Science and Engineering, Kyung Hee University, (KHU), who has supported, advised and helped me throughout my Ph.D study. During my Ph.D. research works, I simply appreciate the inspiration and freedom, which Professor Choong Seon Hong has given to me. Without his advice and guidance, I would not be able to pursue my study to the end. Besides, supervisor’s continuous guidance, encouragement, support and friendly professional attitude, he always played an important role in my entire doctoral study and would also inspiration for my future. I am also grateful to him for inviting my family in South Korea and giving me the support of living and studying.

In addition, I really appreciate the dissertation committee members for their valuable and constructive comments, positive criticism and insightful suggestions, which surely enriches the quality of my dissertation.

My special thanks and gratitude goes to Prof. Tran Hoang Nguyen for his valuable suggestion, guidance, supports, which encouraged me a lot to make my research more wealthy during my academic life at KHU. I also pay sincere thanks to Prof. Walid Saad for his invaluable comments and support; Prof. Shaolei Ren for his thoughtful comments and recommendations, which helped a lot to enrich my research quality.

During four years, I spend almost my time in the Networking Laboratory and live in sweet relationship of all members. I would like to lengthen my thanks to all members of the Networking Laboratory for their support to my personal and academic life at KHU. Especially, my colleagues Dr. Md. Golam Rabiul Alam, Dr. Le Tuan Anh, Dr. Thantzin Oo, Dr. Dang Ngoc Minh Duc, Mr. Dang Le Dinh Trang, Mr. Nguyen Huu Minh Nhat, Mr. Kyithar, Ms. Tran Thi Kim Oanh, Ms. Le Thi Huong Tra, Mr. Nguyen Van Dung deserve special thanks because they helped me a lot in research and in my life.

Playing an important role for my life in Korea, I would like to express my gratitude to all the Vietnamese students in KHU for the community feelings; especially, for their long-lasting supportive hand for me throughout my stay period in Korea. The sweet memories during the
staying at KHU are unforgettable.

Last but not least, I cannot express by any word about my gratitude to my family for their endless support and encouragement; especially, my wife, Mrs. Mai Thi Hai, who is always beside me, supports me and gives all her love, all her sacrifice without waiting any response. I would like to acknowledge and show my deepest gratitude to my parents, my parent in law, my sisters for their unconditional love.

There are numerous well wishers too who have shown me their constant support and friendship in various ways to my life. The short time in KHU, Korea, will be the sweet duration in my life. I will put them deeply in my heart and hope to find a more appropriate place to acknowledge them in the future.

Pham Chuan
Seoul, Korea
August, 2017
Abstract

Recent years, network function virtualization (NFV) has been regarded as an incredible and innovative technology in telecommunications services provisioning in both industry and academia. Many network service providers have been seen NFV as the heart of their infrastructures with profound impacts. The important terminology in NFV is virtual network functions (vNFs), an abstract concept that implies to virtualize many dedicated physical network functions (such as, a network address translator, a firewall, an anti-virus) as software. These virtual network functions can run and share resources on commodity servers based on virtual machines. In legacy network, they are basically are implemented on dedicated hardware devices, or even on middleboxes. However, such physical hardware and middleboxes are quite expensive and inflexible to implement, to scale and to upgrade the network system. Considering a customer, who is owning his local network with multiple hardware devices, when he has to move his company to another place, it is difficult to migrate his company network, including switches, routers, firewalls, etc. The inconvenience also increases when upgrading the network system or scaling up the network, which are normal issues to fulfill the density requirements of users.

Releasing dedicated hardware platforms based on software implementations in a virtualized network environment, NFV supports flexible and agile platforms for cloud network providers in deployment. Underlying NFV platform, the network providers can consolidate many network devices onto the rich resource pool of datacenters (commodity servers). NFV removes the limitation of implementation since it can combine and orchestrate from different domains, different Point of Presences (PoPs). It leads to significantly reduce operating expenses (OPEX) and capital expenses (CAPEX) in implementation. It also makes the deployment of network services flexibly
and agilely in time-to-value.

In its infancy period, NFV still faces several challenges in order to adapt to the diverse network appliances. There are multiple open source platforms, which enable virtualization technologies. NFV requirements from ETSI can bring them to sit together in the network implementation. NFV components still need to enhance the performance, such as efficient algorithms to manage the resource pool, mechanism to make a schedule for executing virtual network functions (VNFs), security mechanisms for network protection. Especially, we focus on improving the performance of manager component in NFV-Management and Orchestrator (NFV-MANO).

The main contributions of this thesis focuses on studying the resource allocation problem in NFV-MANO, where manages physical resources and allocates virtual resource to implement virtual network systems. In particular, we study mechanisms that can allocate physical resources for implementing a set of virtual network functions that chain in specific orders to run customer services (so-called network service chains). There are some use cases in NFV, here, we focus on one specific use case that the network provider who receives a set of virtual network functions for implementation. Given a physical network of the network provider, the controller needs efficient mechanisms to implement all network service chains onto the physical network in terms of reducing the system cost. There are multiple objects that can incur the cost when implementing these virtual network systems, such as the amount of energy consumption of active nodes, network delay, configuration. An effective mechanism in NFV-MANO needs to fulfill the user requirement in implementation also increase the revenue of the network provider. There are two approaches that are traditionally discussed in state-of-the-art, including increasing resource utilization (consolidation) and improving network overhead; those objectives are usually considered separately.

In this thesis, we study mechanism that can combine those traditional methods, called a joint mechanism, combining the consolidation policy and the network-aware policy, even though is not easy to find a solution. It is an NP-hard combinatorial optimization problem, and named the joint operational and network traffic cost (OPNET) problem.

To finding a solution, we solve OPNET by proposing an algorithm based on the Markov ap-
proximation and matching theory game techniques. This framework can provide a suitable solution for a set of combinatorial network optimization problems. It not only provides an optimality gap but also enables an simple implementation. Actually, this framework cannot find an optimal solution, thus, the original problem is applied the log-sum-exp function to approximate to a tractable convex optimization problem. In theory, it can find a close form optimal solution for the approximation problem, however, in practice, it is impossible. Based on this framework, we then design a randomized algorithm to approximate the optimal value based on the time-reversible Markov chain.

In deep research, we analyze the performance of Markov approximation in the network service chain placement to show the advantages and disadvantages in practice. Markov approximation framework can find a close optimal solution for our problem; however, it is then raised to a slow convergence. This happens in our system due to the huge state space, which are caused by combining multiple objectives. We find a solution to tackle this challenge by advocating the matching game theory, which is a good candidate in terms of practical solutions in the network resource allocation problem.

Recent years, matching game theory receives much attentions as a practical and promising approach for network resource allocation. It can eliminate some limitations of optimization, such as complicated formulation and calculation, unable implementation with a lot of theoretic assumption. The robustness of the matching game framework is proven by several matching applications in theory and also a Nobel-price winning framework in 2012.

In this work, we propose a novel approach, named as SAMA that is applied in the algorithm of Markov approximation, which can reduce the state space significantly. In particular, we formulate a network service chain placement underlying two sides of a many-to-one matching game with the main target is to find a stable allocation that can match vNFs into physical nodes. The advantages of this matching approach for the service chain placement problem include 1) covering multiple issues in the model of network service chain placement, 2) a suitable solution, in terms of stability and optimality and 3) an efficient algorithmic implementations that requires low complexity and fast convergence in implementation.
In summary, in this work, we deal with the network service chain placement problem in NFV, the problem belongs to the resource allocation area in NFV. We apply Markov approximation framework to solve this problem, however, the latter results shows the weak point in convergence, so we introduce matching theory game, and propose an algorithm that can reduce the complexity in Markov approximation. We make an analysis and simulation to illustrate the efficiencies of our proposed algorithms.
Contents

List of Figures x

List of Tables xii

1 Introduction 1
   1.1 Motivation ................................................................. 2
   1.2 Approaches to resource allocation in NFV .............................. 3
       1.2.1 Horizontal classification ......................................... 4
       1.2.2 Vertical classification ............................................ 5
       1.2.3 Problem statement for network service chain placement ........ 6
       1.2.4 Challenges in the service chain placement problem ............ 9
       1.2.5 Study goal and methodology ..................................... 9
       1.2.6 Contributions ...................................................... 10
       1.2.7 Structure of the dissertation .................................... 12

2 Background and Related Work 13
   2.1 Network service chain implementations: traditional and modern approaches . . 13
   2.2 Resource allocation in network function virtualization .................. 16
       2.2.1 Operational cost in the network service chain placement problem . . 18
       2.2.2 Network traffic cost in the network service placement problem .... 18
   2.3 Markov approximation-based method .................................... 19
       2.3.1 Combinatorial optimization problem ............................. 19

vii
## List of Figures

1.1 Possible CPE implementation with NFV. ............................... 2  
1.2 A toy example in service chain placement with different policies. ...... 8  
2.1 Traditional network service chain implementation. ...................... 14  
2.2 Network service chain implementation based on NFV platform. .......... 15  
2.3 Network service chain implementation based NFV platform. ............ 17  
2.4 The example of Markov chain [20]. .................................. 21  
2.5 The example of the one-to-many matching game. ....................... 24  
4.1 A simple network service chain placement with 3 vNFs and 3 physical nodes. 35  
4.2 Transition rates from $f \rightarrow f'$ and $f' \rightarrow f$. .................. 39  
5.1 The network topology in the cloud/datacenters. ....................... 54  
5.2 Setting of workload and electricity price. ........................... 55  
5.3 Measurement of the system cost with different weight parameter $\sigma$. . 56  
5.4 Evaluation of the convergence in our proposed approaches. .............. 57  
5.5 Evaluation of the execution time. .................................. 57  
5.6 Evaluation of bandwidth allocation. ................................ 58  
5.7 Evaluation of computing resource utilization. .......................... 58  
5.8 Evaluation of the system cost within 10 time slots. .................... 59  
5.9 Comparison between SAMA and matching approach. ................... 60  
5.10 Evaluation the performance in the long-run simulation. ............... 62
5.11 Measurement of wear-and-tear cost. .............................. 65
List of Tables

5.1 Network traffic requirements ........................................ 61
5.2 vNF instances ....................................................... 63
Chapter 1

Introduction

Background

The concept of Network Function Virtualization (NFV) was born in October 2012, when many Telecommunication Service Providers (TSPs) (e.g., BT, Orange, AT&T, Deutsche Telekom, Telecom Italia, Telefonica) collaborate for both industrial and research actions. The European Telecommunications Standards Institute (ETSI) becomes the home of the Industry Specification Group for NFV (ETSI ISG NFV) [38]. NFV concept implies an architecture that entire network services are virtualized based virtualization technologies. Underlying virtualization layer, the network provider can operate, mange and orchestrate both virtual network function and physical resource flexibly, agilely.

Up to now, ETSI group has completed successfully release the first ETSI documents from October 2013 including an overview infrastructure, architecture framework, description of NFV. The ETSI also proposes many use cases for NFV that may be applied to Customer Premises Equipment (CPE) and Evolved Packet Core (EPC) networks. Fig. 1.1 represents a typical example of an implementation of an CPE, which is made up of dedicated network functions including: NAT, DHCP, firewall, etc. These network functions are virtualized as software and implements on physical resource pools of cloud/datacenters. Further, such network functions chain together in a

---

1In the rest of this thesis, the abbreviate ETSI is used instead of ETSI ISG NFV.
specific order of the service execution (called a network service chain) that needs to implement exactly on physical network. A dynamic placement scheme to allocate resources for network service chains is a critical issue in NFV in order to explore the huge potential of this architecture.

1.1 Motivation

The evolution of NFV are the compelling operational saving and capital expenditure efficiencies to be gained from moving traditionally proprietary hardware solutions to a more generic compute environment [4]. Virtualization is not a new technology, which enables to host multiple applications on a single server, to significantly reduce implementation cost. Virtualization opens the new trend of “anything-as-a-Service” (XaaS). The expectation of NFV does not stop at reducing implementation cost. Indeed, for service providers (SPs), NFV paradigm opens a potential trend that is able to leverage the flexible deployment of network services for SPs. Considering traditional SPs, such as Amazon [1], Rackspace [7], their models are now high automated, orchestrated, virtualized and distributed. What makes these SPs so attractive, innovative, that is the virtualization technologies, specifically, NFV and Software-defined network (SDN) [10]. Following the white-paper [4], we summary briefly the validation for SPs as follows:

- NFV enables to remove the need for the operator to bear the burden of proprietary hardware
platform and simplify the integration from multiple vendors in network implementation.

- NFV leaves SPs to hardware obsolescence across multiple proprietary system.
- NFV can minimize the operational cost from equipment upgrades, maintaining systems.
- NFV is able to enhance deployment for new network functions or changes in scale across existing network functions.
- NFV opens the evolution of elastic cloud based Infrastructure as a Service (IaaS) (either public, or their own private) wherein the many communication network may be elastically scaled as required based on actual live network use.

Currently, there are several approaches that develops resource allocation mechanisms applied in NFV. In this thesis, we classify and summary these approaches in the next Section.

1.2 Approaches to resource allocation in NFV

To enhance the revenue of network providers underlying NFV platform, resource allocation issues in NFV are the main objectives in literature and in practice. Concretely, the network providers prefer to improve efficiently the physical resource utilization while all the requirements of users are fulfilled. In this thesis, we consider the resource allocation in network service chain implementation, where a network service chain is a list of vNFs that are chained in specific order to run its service. There exists service chain deployment solutions that can achieve a sub-optimal resource allocation [50]. It is necessary to find and enhance allocation methods in NFV for making a decision, which physical nodes (servers) should host vNFs in terms of satisfying such objectives as recovery from failures, security, load balancing, energy saving, etc. In this thesis, we classify approaches of resource allocation in NFV in both horizontal and vertical point of views.
1.2.1 Horizontal classification

In this point of view, the resource allocation issue can be classified to two similar approaches: virtual machine (VM) placement in datacenters and virtual network embedding (VNE) with a particular objective.

Virtual machine placement-based approaches

First, some existing works [13, 46, 51] tried to map the resource allocation problem in NFV to VM placement problem since basically, network functions in NFV are deployed on physical servers as VMs. This approach can be applied heuristic algorithms in the set of bin-packing algorithms [29]. In this approach, the bandwidth requirement between vNFs can be considered as one type of resources similar to CPU, hence the requirement bandwidth of service chain is simplified and formulated underlying a link capacity constraint. However, the architecture of network in NFV mainly focuses on the network issues such as bandwidth, flows, etc., and the relationship between vNFs in a service chain is quite complicated represented with the order execution of vNF. These properties could not be ignored as be done in VM placement approach.

Virtual network embedding-based approaches

Second, resource allocation in NFV problem is applied solutions in VNE due to the similar problem domain. The allocation of VNE deals with virtual resource nodes (hosted in substrate nodes) and links (embedded to substrate paths) [21, 23]. VNE problem belongs to the NP-hard set, which is impossibly found a solution in polynomial time. Indeed, most of works done in this area has focused on heuristic methods to reduce complexity in solving VNE problem. Even though proposed solutions in VNE can be applied acceptably to resource allocation in NFV, each problem owns different properties that cannot be merged easily. Specifically, the input of resource allocation in NFV is more diverse with precedence constraints of the set of vNFs. Furthermore, the relationship of vNFs in a service chain is also more complex compared to VNE model, where network traffic
may change according to the network function (e.g., network traffic is reduced after filtered by a firewall). Due to the intractable problem of NFV, existing works [35, 55] have to relax some specific issues to apply well-known approaches in VNE for finding solutions.

1.2.2 Vertical classification

Resource allocation approaches in NFV can be classified based on particular objectives since different objectives result in different resource allocation scheme solutions.

**Operational cost approach**

Resource allocation problem in NFV is often accounted for reducing the energy usage in implementation [15, 40]. In particular, the operational cost is often mentioned by the number of active nodes underlying different models such as energy consumption, resource utilization. Mathematically, this cost is formulated as an Integer Linear Program (ILP) to find a binary variable that is used to determine which vNF is placed on which node. The main objective targets to minimize the set of active nodes used to place network service chains.

In another way, the operational cost can be formulated by defining amount of money to handle an active nodes. The objective of this approach can be changed to optimize the utilization of all active nodes (called consolidation approach). However, this approach usually relaxes the complicated connection between vNFs of service chains that impacts directly to placement scheme [48]. As shown in [18, 41, 42], in order to improve the resource utilization, service chain placement solutions may result in network congestion in the physical network.

**Network traffic cost approach**

In a different point of views, the network service providers consider on the network traffic overhead in the network service chain implementation, such as in [13, 18, 47]. In particular, to deploy service chains on physical nodes, they prefer to choose solutions with minimum network traffic
cost [18]. For example, they would like to minimize the network delay, the amount of network traffic allocation in their implementation. However, if a placement scheme only optimize the network overhead, it may deduce the computing resource utilization (e.g., memory, CPU, storage), which can negative effect with the resource allocation in the first approach.

**Joint multiple objectives approach**

To increase the revenue of network providers, they would like to find an effective service chain deployment that can meet several key challenges. Moreover, in terms of heterogeneous network functions, the network providers need to determine tradeoffs between different objectives. The network operator can mainly enhance the latency of the network services or dominantly improve the physical resource utilization. In some senses, reducing network overhead and improving resource utilization seem conflicting because reducing the set of active nodes in the consolidation policy [48] without considering network traffic may lead to increase the network latency in the customer services [47]. In contrast, reducing the network overhead may spend more computing resources to deploy network service chains. Especially, in virtualization layer, it is simple to share computing resources, such as CPU, memory, while there is no efficient approach to share the network bandwidth. It is necessary to develop joint approaches for service chain placement in terms of considering multiple objectives, as shown in [42, 47].

1.2.3 **Problem statement for network service chain placement**

In this thesis, we analyze the network service chain placement problem in terms of optimizing multiple objectives, an approach is not well addressed in NFV.

To illustrate the system model and problem formulation of the network service chain placement problem in a joint consolidation and network traffic-aware model, we show a toy example in Fig. 1.2. We consider three possible policies of network service chain placements by deploying vNFs on physical nodes. The physical nodes in this example are homogeneous. For ease of illus-
Chapter 1. Introduction

In this section, we only consider one computing resource type (vCPU) of vNFs and nodes. For network connection, each network service chain requires a specific bandwidth. In this case, we consider that all network service chain requires the same network bandwidth. Suppose that the network provider will be charged $3/hour to operate one node (e.g., energy consumption) is $3/hour and the cost to handle a virtual link is $0.5. Hence, we can calculate the total system cost for deploying all network service chains as follows:

a) Case 1: In this case, the network provider uses the consolidation policy without considering network traffic-aware consideration. There exists one idle node that can be turned-off. However the system must carries out four interconnections between vNFs. Four interconnections are embedded on physical links. Thus, the network service provider needs to pay $11 to host all network service chains.

b) Case 2: In this case, the network provider uses the network traffic policy and no consolidation policy. The number of interconnections that need to be carried out can be reduced. The network provider only handles two virtual connections. However, all nodes have to be turned-on, which costs more energy consumption. The computing resource utilization in this case is the lowest. The cost that has to pay for implementation, is up to $13. In this case, it is the highest total cost compared to others.

c) Case 3: Underlying both consolidation and network traffic-aware policies, the system can reduce payment to $10.5. It can be seen as the smallest cost to implement all network service chains. Compared to two cases, it can reduce both the number of active nodes and the number of interconnections.

This toy example illustrates that joint consolidation and network traffic-aware policy can increase the efficiency of service chain placement, compared to separately handle as traditional approaches. In addition, the joint approach in NFV is not well-studied in literature [18, 51].
Chapter 1. Introduction

Figure 1.2: A toy example in service chain placement with different policies.
1.2.4 Challenges in the service chain placement problem

The main challenge of this research is to satisfy the network relationship between vNFs. In particular, the network relationship between vNFs is represented by the specific order depending on network service deployment. Each network service chain contains an ordered list of vNFs that are “stitched” together in the network implementation and operation. The order of vNFs in a service chain makes the barrier to apply well-known methods in virtual machine placement (e.g., bin-packing algorithms), where virtual machines are placed into physical servers independently.

Moreover, network service chain placement problem is NP-hard, so that it requires an efficient solution to reduce the complexity of the placement algorithm. Currently, there is no joint consideration of multiple objectives in the network service chain placement, which increases the complexity in computation. Thus, in this research, we analysis the network service chain placement problem with different approaches. We also compare the complexity and the performance that convince the efficiency of our proposed method.

1.2.5 Study goal and methodology

The objective in this research is to find an efficient solution for joint operational and network traffic cost in vNF placement problem, where the objective function is considered with multiple aspects, such as energy cost, wear-and-tear cost, and network traffic cost.

To derive the solution for the network service chain placement, we advocate the Markov approximation framework, a well-known framework to solve the combinatorial optimization problem in network [20]. It also enable a simple implementation in terms of low computation cost. Due to NP-hard, the problem formulation is transformed to the tractable convex optimization problem by using the log-sum-exp function. Thus, the close-form solution can be found in mathematical theory. But in practice, it is impossible due to the large state space of Markov chain. We design an algorithm based on the stationary probability distribution of the time-reversible Markov chain to treat the close optimal solution when the system converges to the stationary distribution. The
design algorithm can start at any state of the Markov chain to reach the close-optimal solution when it converges.

There are many applications similar to network service chain placement that are applied successfully Markov approximation framework [20,41]. In this problem, Markov approximation also can find the close optimal solution, which is explored in our research. However due to the complexity of NFV, it faces long convergence when finding the optimal solution in a huge feasible solution set. To tackle the long convergence, we propose a combination between the Markov approximation framework and the matching game.

Matching game approach appears in recent years, which a huge consideration, especially in network resource allocation. It also remove the complexity of the optimization and game theory in both problem formulation and practical implementation [26]. The promising of matching game theory is proven by a Nobel-prize 2012 winning framework. Matching theory is one of suitable solutions that is owning a stability, optimality outcome and fast implementation. The combining of these powerful approaches, the Markov approximation and the matching game, brings a new efficient framework to solve the service chain placement problem.

1.2.6 Contributions

In summary, the major contributions of this thesis is summarized as follows:

- Multiple objectives in network service chain placement problem is formulated as a combinatorial optimization problem (named OPNET). NFV forces virtual service functions and their execution as software deployed on a physical resource pool. Hence, the implementation of vNFs in service chains requires multiple issues including: a) where to instantiate vNFs, b) how to enhance the physical resource utilization, c) how to mitigate network overhead in implementation, and d) how to reduce the configuration cost in the system. These issues are explored and formulated in our work for finding an efficient solution of service chain placement.
Markov approximation based solution. Markov approximation framework is a promising candidate for the set of combinatorial optimization problems. For the network service chain placement problem, we develop a mechanism based on this approximation framework. To treat the close optimal solution, we design an algorithm that converges as the stationary probability distribution of the time-reversible Markov chain.

Furthermore, we analyze some methods to enhance the convergence of Markov approximation approach, such as controlling the weight parameter and designing a combination model between the Markov approximation and the matching game approaches, named the SAMA algorithm. SAMA is executed by involving two steps: i) exploring the active nodes to deploy network service chains, and ii) exploiting step that is to place vNFs onto physical nodes in terms of minimizing the total system cost.

Considering the exploitation step of SAMA, we can be reduced the computational cost by proposing a many-to-one matching game algorithm. In that game, vNFs and nodes can act as the matching game players, such as students and colleges. We design an matching algorithm based on many-to-one matching game for vNF placement. The implementation can be applied on both the centralized and distributed ways.

Finally, to evaluate our methods, we compare our methods with current approaches. We also use real trace dataset to build the test model. The results in a specific time slot and in a long term show that SAMA achieve the superior system cost in both . Furthermore, compared to centralized approaches (unable to implement in practice), SAMA converges quickly to the close optimal solution. That is an useful impact in practice implementation in terms of solving the NP-hard problem. Furthermore, SAMA outperforms others terms of reducing the total system cost, enhancing both resource utilization and network traffic overhead.
Chapter 1. Introduction

1.2.7 Structure of the dissertation

The content of thesis are organized into six chapters, as given below.

- Chapter 1 has introduced a brief scenario of the network service chain placement problem in traditional network and NFV. It discussed the motivation of the thesis, challenges of finding an efficient solution, the objective of the study and contributions of the thesis research.

- Chapter 2 has discussed the literature review and different existing approaches of the service chain placement problem. The limitations and drawbacks of existing approaches are also presented in this section.

- Chapter 3 provided the detail of the resource allocation problem in NFV, which is formulated as a combinatorial problem. In this problem, we formulate the resource constraints and multiple cost model to place network service chain.

- Chapter 4 proposed solution based on Markov approximation framework. The mathematical analysis, Markov chain design are proposed in this section for solving the service chain placement problem. We also show the limitation of this approach when applying directly on the service chain placement problem.

- Chapter 5 introduced the novel of the combination between Markov approximation framework and matching game method. The designed algorithms and the complexity analysis are presented in this Section.

- Chapter 6 concluded the thesis with the significant contributions of this research and future directions in this research area.
Chapter 2

Background and Related Work

This chapter presents the background related to our study, which is essential to understand the rest of the thesis. We also review the existing solutions for complete coverage problem of service chain placement in NFV.

2.1 Network service chain implementations: traditional and modern approaches

In the past, the network implementation based on specialized network hardware, such as middleboxes or hardware appliances. However, a lot of hardware appliances are wasted when the administrator upgrades the system, even though some hardware devices are really expensive. The physical hardware-based implementation providing network services suffers of several shortcomings, such as:

- They are really expensive within their short life-cycles.
- They require high energy costs in operation and also the facility space in equipment.
- It is difficult to upgrade the system or even migrate the network.
Chapter 2. Background and Related Work

The network service implementation now is evolving towards the network function virtualization paradigm, where network functions are virtualized and delivered as software based on virtual machines (called virtual network functions, vNFs). These virtual network functions can share the resource pool of commodity servers. The Internet Service Providers (ISPs) now is strongly support this approach by supporting the so-called NFV infrastructures (NFVI). They also can deploy these network services on the resource pool of cloud/datacenters. then, customers now can buy network functions as virtual instances on the network provider to implement their systems without the need of install specific equipment on their local network, as shown in Fig. 2.2.

It is applicable to deploy customers’ network services based on virtual network functions, where cloud network providers now support multiple network functions such as firewalling, caching, load balancing, network address translating, etc. Underlying the NFV technology, cloud network providers now equip the NFV platform infrastructure for their network systems. Network service chain placement based on NFV platform of cloud datacenters requires efficient algorithm to determine on which nodes, vNFs are placed to reduce CAPEX and OPEX. Fundamentally, in the NFV architecture, the component NFV Management and Orchestrator is responsible to performs network service chain placement. Fig. 2.3 shows a picture of the network service chain placement architecture, where the orchestrator manages network service chains through the virtualized

Figure 2.1: Traditional network service chain implementation.
The architecture of NFV fundamentally includes three main components: Network service chains, NFV infrastructure and NFV Management and Orchestration (NFV-MANO), as illustrated in Fig. 2.3. We describe each component as follows.

- **Network service chain**: A network service chain concept implies a list of vNFs that can be chained together with a specific order to tailor for different use cases. vNFs can be run on virtual machines or dedicated hardware and are managed by hypervisors, such as Element Management System (EMS), which can create, configure, monitor, etc. The important issue in this component is the network service chain placement that is to place vNFs of network service chains onto physical resource pool in terms of satisfying those specific orders.

- **NFV infrastructure (NFVI)**. This component includes physical resource layer, virtualization layer and virtual resources layer, which covers all physical and virtual resources. NFVI component here enables to manage network connectivity between different locations such as between datacenters. The physical resources include storage, computing, network bandwidth for vNFs allocated via the virtualization layer.
• NFV-Management and Orchestration (NFV-MANO): This component is management and orchestration layer the NFV orchestrator, VNF manager and Virtualized Infrastructure Managers (VIM). First, VIM manages NFVI resources in one NFV infrastructure domain that includes physical computing resources, network resources, virtual resources. VIM can manage life-cycle of virtual resources, tear down virtual machines into physical resources, keep the association between virtual machines and physical resources, perform and manage fault. Second, VNF manager can manage life-cycle of VNFs. It is also responsible for fault, configurations, performance, security, etc. Finally, NFV orchestrator is the function used to coordinate, authorize, release, engage NFVI resources among different domains.

Fig. 2.3 illustrates a overview scenario, where the orchestrator manages virtual network functions, virtual resources, and physical resources through the VNF manager and VIM. Before allocation, the orchestrator component validates all conditions including physical resource status, vNF life cycle, virtual resources [28].

When the cloud network provider receives implementation requests from customers, the orchestrator manages vNFs requests through the VNF manager. It will evaluates the resource conditions through the physical resource management to perform the network service chain placement on physical resources. The orchestration module provides the functionality required for creating, managing, deleting vNFs. It also include databases that are used to manage network service chains and resources.

### 2.2 Resource allocation in network function virtualization

There are some important issues on resource allocation based on NFV architecture that are widely studied in literature [13, 39, 40]. In these works, the resource allocation in NFV deals with the minimization of the operational cost and the network latency in implementation.
Figure 2.3: Network service chain implementation based NFV platform.
2.2.1 Operational cost in the network service chain placement problem

In NFV, physical hardware-based network functions are virtualized as software and run on virtual machines to share pool of resources. Hence, the critical issue is how to improve the resource utilization of physical servers in the implementation. In this issue, the objective is often formulated as the operational cost that can be derived as controlling the set of active nodes. This issue originally begins from the middlebox implementation where the resource cost of middleboxes is really expensive. Some other works name this approach as the server consolidation method, which implies to consolidate the physical resource in these middleboxes in terms of improving the resource utilization.

In datacenters, sharing computing resources (e.g., vCPU, memory and storage) is explicit in virtualization layer. The network operator can easily monitor, manage and consolidate computing resources by using resource management functions. Therefore, there exist several consolidation policies [17, 18, 48] that often neglect connection between vNFs in the same network service chain. With the same objective, other studies [17, 56], and [27] considered the network service chain placement problem with the goal that is to control the set of active nodes based on typical approaches proposed in the VM placement problem.

2.2.2 Network traffic cost in the network service placement problem

In different points of views, there exists another important paradigm of service chain placement problem that is the network traffic cost. These approaches in this objective focus on reducing the network overhead in the implementation, such as [13, 16, 35, 52]. Depending the problem formulation, network traffic cost has been considered differently. The authors in [13] and [52] focus only on the network latency, while the consolidation policy is neglected. The authors in [35] and [16] consider the link utilization when deploying vNFs. However, a lesser-known fact that considering network traffic cost without involving the operational cost will degrade the physical computing resource utilization (i.e., increasing the number of active nodes), and vice-versa only
consolidating the physical computing resource may result in increasing the network traffic cost [18, 52].

In the next section, we introduce the basic knowledge of the Markov approximation and matching game methods. The approaches are used to solve our problem in this thesis.

## 2.3 Markov approximation-based method

Recent years, Markov approximation framework has been considered as the promising solution of the combinatorial optimization problem, which cannot be solved in a polynomial time, even in a centralized manner. In practice, many network designs and resource allocation problems are formulated as combinatorial optimization problems, such as the peer-to-peer streaming system [20], the VM placement problem [53] and the maximum weighted independent set (MWIS) problem. Furthermore, this framework enables to design efficient solutions in terms of both centralized and distributed implements.

### 2.3.1 Combinatorial optimization problem

We now consider a simple network combinatorial optimization problem with a set \( R \) of users, and a set \( F \) of configurations. An individual user configuration \( f \in F \) can achieve certain performance \( \lambda_r(f) \). The optimization problem can be formulated to maximize the aggregation user performance as follows:

\[
\max_{f \in F} \sum_{r \in R} \lambda_r(f). \tag{2.1}
\]

The above problem is equivalent to

\[
\max_{f \in F} \sum_{f \in F} p_f \sum_{r \in R} \lambda_r(f), \tag{2.2}
\]

s.t

\[
\sum_{f \in F} p_f = 1, \tag{2.3}
\]
where \( p_f \) is the weight to use \( f \). Hence, the problem (2.1) is called maximum weighted configuration (MWC) problem [20].

### 2.3.2 Log-sum-exp approximation

The new problem is still unsolvable, which can be approximated by applying log-sum-exp approximation function. The first differentiable function is defined as follows:

\[
g_\theta(\lambda) := \frac{1}{\theta} \log \left( \sum_{f \in \mathcal{F}} \exp \left( \theta \sum_{r \in \mathcal{R}} \lambda_r(f) \right) \right),
\]

where \( \theta \) is a constant and positive.

As proven in [20], \( g_\theta(\lambda) \) is the same as the problem

\[
\max_{f \in \mathcal{F}} \sum_{f \in \mathcal{F}} p_f \sum_{r \in \mathcal{R}} \lambda_r(f) - \frac{1}{\theta} \sum_{f \in \mathcal{F}} p_f \log p_f,
\]

s.t.

\[
\sum_{f \in \mathcal{F}} p_f = 1.
\]

Note that \( g_\theta(\lambda) \) is a convex function. By solving the Karush-Kuhn-Tucker (KKT) conditions, we can obtain the close-form solution as follows:

\[
p_f^*(\lambda) = \frac{\exp(\theta \sum_{f \in \mathcal{F}} \lambda_r(f))}{\sum_{f' \in \mathcal{F}} \exp(\theta \sum_{f \in \mathcal{F}} \lambda_r(f'))}, \forall f \in \mathcal{F}.
\]

An useful insight is that by time-sharing among different configurations \( f \) according to their portions \( p_f^*(\lambda) \), we can solve the approximation problem MWC, and hence the problem MWC, approximately with the optimality gap bounded by \( \frac{1}{\theta} \log |\mathcal{F}| \). Then a key to creating a new algorithm designs is to treat \( p_f^*(\lambda), f \in \mathcal{F} \) as the stationary distribution of a time reversible Markov chain. Time-reversible Markov chains usually have structures that allow distributed implementation. As the Markov chain converges to its stationary distribution, which can approach \( p_f^*(\lambda) \).
2.3.3 Algorithm design based on Markov chain

To construct a time-reversible Markov chain with its stationary distribution $p^*_f(\lambda)$, we let $f \in \mathcal{F}$ be the state of the Markov chain, and denote $q_{f \rightarrow f'}$, as the transition rate between two states $f$ and $f'$. The design algorithm needs to satisfy:

- the resulting Markov chain is irreducible, i.e., any two states are reachable from each other,
- and the Markov chain balance equation is satisfied,

$$\exp \left( \theta \sum_{r \in R} \lambda_r(f) \right) q_{f \rightarrow f'} = \exp \left( \theta \sum_{r \in R} \lambda_r(f') \right) q_{f' \rightarrow f}. \quad (2.8)$$

We show an example with the 4-state Markov chain assumed as in Fig. 2.4. If $f$ and $f'$ that have direct transitions, then we can use four options of choosing a new state based on $q_{f \rightarrow f'}$ and $q_{f' \rightarrow f}$ as follows:

Option 1: If $q_{f \rightarrow f'}$ is negative, the system will select the configuration $f$ based on

$$q_{f \rightarrow f'} = \alpha \left[ \exp \left( \theta \sum_{r \in R} \lambda_r(f) \right) \right]^{-1}. \quad (2.9)$$

Option 2: If $q_{f \rightarrow f'} > 0$, the system will select the configuration $f'$ based on

$$q_{f \rightarrow f'} = \alpha \exp \left( \theta \sum_{r \in R} \lambda_r(f') \right). \quad (2.10)$$
Option 3: If $q_{f \rightarrow f'} > 0$, the system will uniformly select $f$ or $f'$:

$$q_{f \rightarrow f'} = \alpha \exp \left( \frac{1}{2} \sum_{r \in R} (\lambda_r(f) - \lambda_r(f')) \right).$$  \hspace{1cm} (2.11)

Option 4: If $q_{f \rightarrow f'} > 0$, the system will uniformly select $f$ or $f'$:

$$q_{f \rightarrow f'} = \alpha \exp \left( \frac{1}{2} \sum_{r \in R} (\lambda_r(f) - \lambda_r(f')) \right).$$  \hspace{1cm} (2.12)

Based on these options, the algorithm is designed, where the system is spent more time on a better configuration (i.e., higher performance, or better in terms of cost). In summary, Markov approximation framework can approximate any combinatorial optimization problem to the tractable problem using the log-sum-exp approximation function.

The approximation problem is the convex problem that can find a close form solution; hence, the approximation gap is guaranteed lower by $\frac{1}{\theta} \log |F|$ \cite{20}. Even though Markov approximation method is a promising approach for resource allocation problem, little work exploits this study on NFV as well as analyzes the challenges when applying to solve a service chain placement problem.

### 2.4 Matching game approach to the network service chain placement problem

In the set of solutions to NP-hard combinatorial problems, matching theory has been risen from 2012 with a Nobel-prize winning framework. Matching game not only provides mathematically tractable solutions but also simple implementations for the combinatorial problem. It is one of suitable solutions for many resource allocation problems in terms of stability, optimality and fast implementation. In VM placement problem, matching theory is also an applicable solution that is done successfully in \cite{41,53,54}. It also leverages us for exploring the study of matching on the service chain placement problem.
Chapter 2. Background and Related Work

2.4.1 Background of the stable matching game

We briefly introduce the basic stable matching game theory in the one-to-one stable marriage model as the first step to deeply research on the matching game theory. Here, the stable matching game consists of two disjoint sets, men and women, defined by $I = \{i_1, i_2, ..., i_n\}$ and $J = \{j_1, j_2, ..., j_p\}$, respectively. For each agent, it has a transitive preference over individuals on the other side, and the possibility of being unmatched [24]. To match agents between two side, each agent needs a preferences that can be represented as ranking list of other agents in the opposite side, $p(i_1) = \{j_1, j_3, ..., j_i\}$, meaning that man $i_1$’s has the first choice is $j_1$, the second choice is $j_3$ and so on. We use $a \succ_i b$ to denote the agent $i$ that prefers $a$ to $b$.

**Definition 2.1.** A stable matching outcome $\mu: I \times J \times \emptyset \rightarrow I \times J \times \emptyset$ such that $\mu(i) = j$ if and only if $\mu(j) = i$ and $\mu(i) \in J \cup \emptyset \forall i, j$.

**Definition 2.2.** A matching is stable when all agent are matched. In another word, a matching is individual rational to all agents, if and only if there does not exists an agent $i$ who prefers being unmatched to being matched with $\mu(i)$.

**Definition 2.3.** A stable matching does not exist any blocking pair. A matching $\mu$ is blocked by a pair of agents $(i, j)$ if they each prefer each other to the partner in their matching $\mu$. It means that $j \succ_i \mu(i)$ and $i \succ_j \mu(j)$. Such a pair is called a blocking pair in stable matching.

**Definition 2.4.** A stable matching $\mu$ is individual rational and is not blocked by any pair of agents. Further, all agent have to be matched in the matching $\mu$.

The classic deferred acceptance algorithm (DA), or the Gale-Shapley algorithm [24] works with both side of the market, where men propose to the women, according to their preferences. A man will propose to the woman, who is most preferred in his list and has not yet rejected him before. For a women, if she is available, she will hold the best proposed proposal depending on her preference. In case, she is not available, she can reject the old one and hold the better proposal.
Many-to-one matching algorithm

Figure 2.5: The example of the one-to-many matching game.

if he comes. This iteration until no proposal can be made, where all men are matched to women.
It can be readily seen as the stable outcome of the deferred acceptance algorithm.

2.4.2 Many-to-one matching game for network service chain placement

In a cloud/datacenters, the resources on node (server) are slicing and allocated to vNF. Placing
service chain on physical nodes, can be cast as a one-to-many matching game [24], where multiple
vNFs of service chains can be placed in the same node.

In particular, the many-to-one matching game is an approach that can be used to solve the
college admission problem [24]. In that game, one college has a limitation of the number of
accepted students. Each student also has his private preference to colleges, he then applies his
application to the most preferred. Each college will select a list of students depending on its
ranking, and reject when it comes to the quota. The context here looks similar to the service chain
placement, however, this cannot be directly applied due to the different size and the relationship
between vNFs. To due with this issue, we propose the matching game algorithm to place vNFs on
physical nodes (explained in details in Section 5).
2.5 Summary

Service chain placement are traditionally formulated as Mixed Integer Linear Problem (MILP) in terms of reducing operational cost or network load overhead. Without joint multiple objectives, existing works address the service chain placement to minimize only the operational cost or network traffic cost. Each objective leads to different network service chain placement schemes in the result. By showing dependent and interaction between vNFs, it is significant to consider a joint operational and network traffic cost problem in the network service chain placement (called OPNET). It is impossible to solve OPNET in a polynomial time with a large scale system since OPNET still belongs to the combinatorial optimization problem set, which is NP-hard. Based on the powerful and robust Markov approximation framework, we propose a new mechanism to place vNFs satisfying service chain requirements. We also illustrate the drawback of Markov approximation approach in service chain placement that leads to a new algorithm that combines the matching game theory and Markov approximation framework to reduce the complexity during exploring the huge feasible states of Markov chain.
Chapter 3

Network Service Chain Placement

Problem Statement

3.1 Introduction

The network service chain placement is the method of mapping vNFs to physical nodes (servers). In other words, a placement scheme is a way of choosing the foremost appropriate node for executing a given set of vNFs. Due to the precedent order in execution of a specific network service chain, the method may involve different objectives. Here, we mainly focus on reducing both the operational cost and the network traffic cost for deploying the network service chains in NFV.

The chapter is organized as follows. In section 3.2 the general system model of the service chain placement is presented. We present the problem formulation of network service chain placement in section 3.3, where we describe the system constraints and the cost model in service chain placement problem. Finally, we summary the service chain problem formulation in Section 3.4.
3.2 System model of network service chain placement in NFV

3.2.1 General assumptions

In this study, we consider the network system of one network provider, who equips a controller to receive demand workloads (network implementation requests) and makes a decision for implementing network service chains in each time slot [18,51]. We suppose that time is discrete and the network operator need to make a network service chain placement within one time period. This assumption means that there is no input parameters (such as amount of demand workloads, network topology configuration, physical node configurations) that will be changed within the considered time slot. The proposed network service chain placement algorithm must be converged before the end of time slot. Without loss of generality, this assumption is used in many works with the same area [13, 18, 23, 51].

3.2.2 System model for the network service chains placement problem

In this study, we consider the network service chain placement for a network service provider, who is owning \( M^{\text{max}} \) heterogeneous physical nodes (servers). At one time slot, the network provider carries out their network system with a subset \( \mathcal{M} \) of active nodes to handle a set \( \mathcal{C} \) of network service chains. The service chain \( c \) requires a list of vNFs with a specific order. Each vNF \( n \) in the set \( \mathcal{N} \) vNFs belongs to only one network service chain \( c \). When implementing a network service chain \( c \), the network provider has to guarantee that specific order of all vNFs in the chain \( c \). Furthermore, to implement vNFs for a network service chain, each service chain \( c \) requires a specific network traffic rate between vNF \( n \) and \( n' \) presented by \( a_{nn'}^c \).
3.3 Problem formulation of network service chain placement

3.3.1 Constraints of network service chain placement in multi-resource node

Basically, NFV is using on virtualization technique to deploy vNFs. Hence, to run its service function, each vNF requires a specific configuration including multiple computing resources, such as memory, CPU, storage. Here, we consider a set $\mathcal{P}$ of resource types. We denote $r_p^v$ as the computing resource requirement of resource type $p \in \mathcal{P}$ of vNF $n$. Moreover, we use $r_{m}^{p}$ and $r_{c}^{p}$ to denote the computing resource type $p$ of node $m$ and network service chain $c$, respectively. In common, $r_{c}$, $r_{n}$ and $r_{m}$ present a vector resource of network service chain $c$, vNF $n$, and node $m$, respectively.

To formulate the network service chain placement, we define $X_{cmn}$ as a indicator variable that implies whether vNF $n$ of network service chain $c$ is allocated on physical node $m$ ($X_{cmn} = 1$) or not ($X_{cmn} = 0$). Furthermore, a node cannot allocate resources over its capacity, we consider the constraint presented as follows:

$$\sum_{c \in \mathcal{C}} \sum_{n \in \mathcal{N}} r_{n}^{p} \cdot X_{cmn} \leq r_{m}^{p}, \forall m \in \mathcal{M}, \forall p \in \mathcal{P}. \tag{3.1}$$

This constraint can be consider for all computing resources that need to execute an vNF. The next constraint is used to ensure that given network service chain $c$, an vNF $n$ cannot be separated to multi-nodes.

$$\sum_{c \in \mathcal{C}} \sum_{m \in \mathcal{M}} X_{cmn} = 1, \forall n \in \mathcal{N}. \tag{3.2}$$

Finally, we capture the constraint of network traffic as follows. Here, we consider the aggregated traffic of any all embedded link $nn'$ on $mm'$. The total aggregated network traffic of embedded virtual connection $nn'$ has to be lower than the physical link $mm'$ capacity.

$$\sum_{c \in \mathcal{C}} \sum_{n,n' \in \mathcal{N}} a_{nmn'}^{c} X_{cmn}^{c} X_{cmn'}^{c} \leq B_{mm'}, \forall m, m' \in \mathcal{M}, m \neq m'. \tag{3.3}$$

where $B_{mm'}$ is the traffic rate capacity of $mm'$, which can be measured based on the cloud network topology (e.g., VL2 [25]).
Considered constraints (3.1), (3.2) and (3.3) are traditional constraints in the network service chain placement problem. In fact, there are some other constraints that are used to formulate the shared network functions (e.g., anti-virus) that can be implemented through multiple network service chains; or to formulate the network traffic rate after processing at a specific vNF (e.g., network traffic will be reduced after filtering by a firewall). In this study, we consider a common model of network service chain placement, which does not focus on specific cases of network service chain placement. Therefore, the capacity and the requirement can be considered as average values in the system. We next provide system cost models for network service chain placement problem as follows.

3.3.2 The network traffic cost in the network service chain placement

To deploy network service chains using vNF instances, such vNFs must be hosted on physical nodes by VMs [13, 30, 52]. These vNFs will be interconnected to run their network services. A physical node can host many vNFs as well as several virtual links of vNFs based on its physical resource capacity. A different vNFs assignment will lead to a different network traffic cost to serve vNFs. In this study, we consider the cost to handle a virtual link, which can be seen as the network delay to execute a network service.

From one physical node to another node, the physical link can be traversed in the same cluster or different cluster. Hence, we consider the network traffic cost model using the hop distance calculation and the aggregated bandwidth of virtual links. In this model, we do not consider the change of traffic rate after processing at one virtual network function. Given a network service chain implementation, each virtual link in this network service chain requires a specific bandwidth.

We define the network traffic cost when placing vNFs \(n\) and \(n'\) based on the hop distance \(D_{mn'}\) of those nodes that host vNFs \(n\) and \(n'\), respectively. The hop distance of the network topology in a cloud/datacenters, which is well-studied in [25, 37], can be given in this study. The
network cost model is formulated as follows

\[
G(X) = \sum_{m, m' \in M, m \neq m'} D_{mm'} \sum_{c \in C} \sum_{n, n' \in N, n \neq n'} a^c_{nm} X^c_{nm} X^c_{n'm'}.
\]  

(3.4)

In practice, an inefficient network service chain placement will suffer high overhead of inter-traffic among vNFs, where the network provider needs to allocate a lot of network bandwidth and handle a lot of virtual connection for virtual links. Traversing through several hops, the network delay in customer’s services also increase, which reduces the users’ quality of services. However, if the network provider only minimizes the used network bandwidth, it will reduce the resource utilization, which is contradictory on the objective of the consolidation resource policy \cite{18}. Observing the important of that point, we focus on optimizing both resource utilization and network traffic-aware in the network service chain placement problem.

### 3.3.3 The operational cost in the network service chain placement

According to \cite{31}, datacenters do not prefer to turn on all their servers (nodes) that can increase energy cost in the network system and reduce providers’ revenue. The energy consumption issue in datacenters today is still a big challenge in practice and research. Moving these idle servers to sleep mode that consumes the less power instead of running on the idle status is the target of consolidation policy in datacenters. Hence, reducing amount of active nodes is a significant challenge when the cloud network provider hosts network service chains on cloud/datacenters. The network provider can be a cloud datacenter or a broker who will be charged amount of money to operate the physical network. As the concept of cloud/datacenters, “pay-as-you-go”, there is no network provider, who would like to use more resources as they need in their implementations. Based on this objective, we formulate the cost model in the system that needs to serve all user demands underlying the power consumption of all active nodes. Some current works consider this cost as the amount of money to handle an active server as follows \( \alpha \sum_{m \in M} Q_m \), where \( Q_m \) is average amount of energy consumption of node \( m \), and \( \alpha \) can be seen as a monetary term.
Traditionally, $Q_m$ is calculated based on the CPU utilization of a physical node $m$. In this study, we consider $Q_m$ as the average power consumption, that can be measured by the history.

On the contrary, the network providers also do not prefer to frequently turn-on/off their servers, which is not also used in practice. Switching-on/off can cause the increasing *wear-and-tear cost*, a cost model is mentioned in [31,44,45], and [49]. Moreover, the network providers have to suffer the reconfiguration cost when turning-on a physical node. For example, the network system needs to update the available resources, reactive physical connections of those nodes. It also increases fault and latency in the network system. However, to measure clearly this cost, it is impossible. Thus, we define a linear wear-and-tear cost function as follows:

$$
\beta |M - M'|,
$$

where $\beta$ is an average monetary weight when turning off a physical node (i.e., $$/nodes), $M$ is the number of active nodes at current time slot corresponding to the subset $M$ of active nodes, and $M'$ is the amount of active nodes in previous time slot.

In the literature of datacenters, the wear-and-tear cost function is not well considered [17, 27, 32, 33] as well as in NFV architecture [43, 48]. The system models mainly focus on optimizing the number of active servers or physical nodes, while ignoring the wear-and-tear cost. Actually, the wear-and-tear cost is difficult to measure exactly in operation. To improve the revenue of network provider in network service chain placement, the network operation has to make a careful consideration on this.

Consequently, the operational cost in the system can be formulated as follows:

$$
E(\mathcal{M}) = \beta |M - M'| + \sum_{m \in \mathcal{M}} \alpha Q_m.
$$

### 3.3.4 The problem formulation for the network service chain placement

We formulate the problem of network service chain placement by combining the two cost models above. To make a summation between them, we use a weighted factor $\sigma \in [0, 1]$ to convert these
cost into the same dimension. The cost model is rewritten as follows:

\[ C(\mathcal{M}, X) = \sigma E(\mathcal{M}) + (1 - \sigma)G(X). \]  

(3.7)

The objective function includes two variables, such as the set \( \mathcal{M} \) of active nodes and the placement variable \( X \). The designed parameter \( \sigma \) is changeable that can be adjusted depending on any desired performance/cost tradeoff of the network provider.

Based on all constraints and cost functions, we introduce the joint operational and network traffic problem (OPNET) as follows:

\[
\begin{align*}
\min_{\mathcal{M}, X} & \quad C(\mathcal{M}, X), \\
\text{s.t.} & \quad \sum_{c \in C} \sum_{n \in \mathcal{N}} r_{np}^c X_{nm}^c \leq r_m^p, \\
& \quad \sum_{c \in C} \sum_{m \in \mathcal{M}} X_{nm}^c = 1, \\
& \quad \sum_{c \in C} \sum_{n,n' \in \mathcal{N}} a_{nm}^c X_{nm}^c X_{n'm'}^c \leq B_{mm'}, \\
& \quad \forall n,m \in \mathcal{M}, m \neq m', \\
& \quad |\mathcal{M}| \leq M_{\text{max}}, \\
& \quad X_{nm}^c = \{0, 1\}, \quad \forall c \in C, \forall n \in \mathcal{N}, \forall m \in \mathcal{M}. 
\end{align*}
\]  

(3.8)

The target of OPNET is to find a network service chain placement with minimum total system cost, which is represented by the summation of the operational cost and traffic cost in the objective function. Unfortunately, OPNET cannot be found the solution in the polynomial time since it is NP-hard. There are several heuristic solutions that can solve this problem; however, they do not guarantee the gap with the optimal solution, and the time complexity. Here, we advocate the Markov approximation-based approach as the solution that has many advantages to due with the set of combinatorial problems.
3.4 Summary

To deal with the problem of network service chain placement, we formulate OPNET, which includes multiple objectives, such as energy cost, wear-and-tear (configuration) cost and network traffic cost. We design a weight parameter that is used for trade-offs between different objectives. The designed network service chain placement problem, OPNET, is a combinatorial NP-hard problem that cannot be solved in polynomial time.
Chapter 4

Markov Approximation Based Approach for Network Service Chain Placement

4.1 Introduction

The robust framework, Markov approximation, is introduced by M Chen and \textit{et al} in \cite{20} for a set of combinatorial network optimization problem. By using log-sum-exp function, this framework can approximate the original problem to the close optimal solution. In this chapter, we present the Markov approximation framework and our proposed method for service chain placement.

The chapter is organized as follows. In section 4.2, we present the log-sum-exp approximation in service chain placement problem. The designed algorithm for service chain placement based on Markov approximation approach is presented in section 4.3. Finally, in section 4.4 we discuss about the drawback of Markov approximation framework in OPNET.
Figure 4.1: A simple network service chain placement with 3 vNFs and 3 physical nodes.

4.2 Log-sum-exp approximation method

We consider a configuration $f = \{M, X\}$ for OPNET, and $\mathcal{F}$ be the set of all configuration that satisfy all placement constraints in OPNET. Configuration $f$ implies a specific vNF placement scheme that maps on network service chain on a subset of active nodes $M$. Since OPNET is the combinatorial optimization problem, there are a lot of feasible solutions that can be the candidates for our target. If we consider each configuration $f$ has a probability of choice (i.e., weight), the best configuration will have the highest probability. Consequently, we let $C_f = C(M, X)$. Then,
we have \( \min_{f \in F} C_f \). The original problem can be given as follows

\[
\begin{align*}
\min_{p > 0} & \sum_{f \in F} p_f C_f, \\
\text{s.t.} & \sum_{f \in F} p_f = 1.
\end{align*}
\] (4.1a)

This problem also cannot be solved in the polynomial time. Hence, we approximate OPNET to the new tractable convex problem by using log-sum-exp function:

\[
\begin{align*}
\min_{p > 0} & \sum_{f \in F} p_f C_f + \frac{1}{\delta} \sum_{f \in F} p_f \log(p_f), \\
\text{s.t.} & \sum_{f \in F} p_f = 1,
\end{align*}
\] (4.2)

where the second term is entropy term and \( \delta \) can be set as a large positive constant.

By solving the Karush-Kuhn-Tucker (KKT) conditions [19] of (4.2), the optimal probability distributions \( p^* \) is presented as follows

\[
p^*_f(C_f) = \frac{\exp(-\delta C_f)}{\sum_{f' \in F} \exp(-\delta C_{f'})}, \quad \forall f \in F,
\] (4.3)

and the optimal objective value is

\[
-\frac{1}{\delta} \log \left[ \sum_{f \in F} \exp(-\delta C_f) \right] \approx \min_{f \in F} C_f.
\] (4.4)

This is the close-form solution that fits to the assumption, which we know exactly the feasible set \( F \). Unfortunately, in practice, it is difficult to calculate (4.3), since the controller cannot find all feasible solution in \( F \). In this framework, this optimal probability can be treat by designing an algorithm following the time-reversible Markov chain. As the stationary distribution of the time reversible Markov chain, we can reach to \( p^*_f(C_f) \).

### 4.3 Markov approximation-based algorithm for OPNET

#### 4.3.1 Design a Markov chain and transition rate

As we mentioned above, the network operator can easily find a feasible configuration; however, it is not a solution. Among those configurations, how the operator can reach the solution that he
wants. The designed algorithm based on Markov chain is to find the method that can reach the solution starting as any beginning configuration.

We consider that each feasible configuration $f$ is a state in the Markov chain and the stationary distribution $p^*_f(C_f)$ given in (4.3). As the properties of time-shared in the Markov chain, $p^*_f$ can be achieved when the Markov chain converges and the best configurations will be chosen most of the time [20] (as the highest probability). The proof in [20] shows that there always exists a stationary distribution $p_f(C_f)$ in the time-reversible Markov chain.

The controller can start with any configuration $f$ and select a new configuration $f'$ based on a transition probability. Following [20], we define $q_{(f \to f')}$ and $q_{(f' \to f)}$ as the probability (called transition rates) that is used to determine how much percentage to choose a configuration $f'$ from $f$ and vice versa, respectively, as shown in the example of Fig. 4.2. Then, based on the design Markov chain, the balance equations for all $f, f' \in \mathcal{F}$ is presented as follows:

$$p^*_f(C_f)q_{(f \to f')} = p^*_f(C'_f)q_{(f' \to f)},$$

$$\exp(-\delta C_f)q_{(f \to f')} = \exp(-\delta C'_f)q_{(f' \to f)}.$$  (4.5)

According to (4.5), we can derive the transition rate

$$q_{(f \to f')} = \exp(-\tau) \cdot \frac{1}{1 + \exp[-\delta(C_f - C_{f'})]},$$

$$q_{(f' \to f)} = \exp(-\tau) \cdot \frac{1}{1 + \exp[-\delta(C_{f'} - C_f)],}$$  (4.6)

where $\tau$ is a constant [20].

Based on the calculation in (4.6), we can see that the transition rate is calculated depending on the total cost $C_f$ and $C_{f'}$ as follows.

- If $C_f < C_{f'}$, then system stays at $f$ with $q_{(f' \to f)} \approx 1$.
- If $C_f > C_{f'}$, then the system selects $f'$ with $q_{(f \to f')} \approx 1$.
- If $C_f = C_{f'}$, then the system stays at $f$ or jumps to $f'$ with equal probability.
Based on this design, the system almost spends more time on the configuration that has lower total cost. Therefore, with the optimal solution, the system will be stayed on its with high probability.

Note that the state Markov chain here needs to guarantee (i) from any state, it can reach to any other states; and (ii) (4.5) has to satisfied. Hence, at any configuration \( f \), the network operator can find randomly a new configuration by changing only one placement of any vNF to create a new configuration, as shown in Fig. 4.1a. The designed algorithm based on Markov chain is presented in the next subsection.

### 4.3.2 Network service chain placement algorithm based on Markov approximation method

Now, we present the algorithm for OPNET based on time-reversible Markov chain. Again, we can easily find a configuration \( f \) but it is difficult to find the best solution. Markov approximation-based algorithm is a randomized algorithm, which starts randomly at any configuration then find the close optimal solution based on the chosen probability. The controller can start with a solution \( f \), and may move to another feasible solution \( f' \) according to the probability \( q(f \rightarrow f ') \). The convergence occurs when it reaches to the steady-state distribution \( p_f^*(C_f) \), it means that the system prefers to select that configuration almost the time. All steps of the algorithm can be sketched out as follows:

1. Initialize randomly one configuration \( f_0 \) with \( M_0 \) and \( X_0 \). Set \( f_0 \rightarrow f^*, X_0 \rightarrow X^*_0 \), and \( M_0 \rightarrow M^*_0 \).
2. Randomly select \( n \), then change vNF \( n \) to create \( f' \), with \( M' \) and \( X' \).
3. Probabilistically select \( f' \) according to (4.6) and the cost \( C_{f'} \). If the controller chooses \( f' \), then \( f' \rightarrow f^*, M' \rightarrow M^* \) and \( X' \rightarrow X^* \), respectively.
4. Return to Step 2 until the stopping criteria are met.

This algorithm iteratively repeats within two phases, exploring and exploiting phases. Exploring phase is implemented to find a random feasible configuration. Exploiting phase is to place
vNFS into physical nodes. At the stationary distribution, the state that has highest time-sharing, will be the chosen solution (convergence status).

4.4 Challenges in applying Markov approximation framework

As presented in the above algorithm, Markov approximation-based algorithm can find a close-optimal solution for OPNET, even though it faces a very slow convergence because of the exploration of feasible placements on the huge feasible set. The large number of combinations can be explained by both variables, the subset of active nodes variable, $M$, and the placement variable, $X$, which are tackled in our model.

The slow convergence can be partially solved by adjusting $\delta$. A parameters is seen as a trade-off factor used to control between exploiting the current solution and exploring solutions. If $\delta$ increases, a better solution is kept with a higher probability. By using a large value of $\delta$ the system is stay at a local optimum for a long time before successfully exploring other better solutions. In practice, using a large value of $\delta$ looks impossible due to the limitation of the computing capacity.

With a given value of $\delta$, the original problem OPNET is approximated by an entropy term $\frac{1}{\delta} \sum_{f \in \mathcal{F}} p_f \log(p_f)$. Hence, the optimality gap is $\frac{1}{\delta} \log(|\mathcal{F}|)$, which depends on $|\mathcal{F}|$. 
In the worst case with maximum $K$ active nodes that are used to host $N$ vNFs. The size of the state space in the Markov chain now is $N!K$. We can calculate the state space by $N!2^K$; then the gap is $\frac{1}{\delta} \log N!2^K$, which is $O(N \log N + K)/\delta$.

The network operator can mitigate the optimality gape with a large $\delta$ [20]. In fact, we cannot use a big value of $\delta$ due to the limitation of computing. Finding a new approach to reduce the state space of Markov chain is necessary. In this study, we combine Markov approximation method with the matching game for solving this issue.
Chapter 5

Joint Approximation and Matching Approach for Network Service Chain Placement Problem

5.1 Introduction

The proposed Markov approximation-based algorithm can find successfully a near-optimal solution of OPNET. This result is also proven in [20]. However, its convergence is very slow because of the huge configurations. To reduce the overhead of exploring in the huge feasible state of OPNET, we study a joint Markov approximation and matching game in service chain placement problem. By combining advantages of Markov approximation framework and matching game, we can enhance the performance of Markov approximation.

The chapter is organized as follows. In section 5.2, we present the a new designed algorithm for service chain placement problem. In section 5.3, we present the matching game for solving sub-problem in OPNET.
5.2 SAMA: A new designed algorithm for OPNET

Reducing the state space in Markov chain is the main issue to improve the slow convergence of Markov approximation. In this chapter, we next introduce a new approach find a solution following Markov chain model, named SAMA.

Presented in Chapter 3, There are two variables, $M$ and $X$, that are coupled in OPNET, which create a huge state space in the Markov chain. We propose an approach to improve the computation of the exploitation state. Given the set of chosen nodes, the exploitation state will find a vNF placement scheme. However, this is still an NP-hard problem, which exists many combinations. Fortunately, the exploitation state of vNF placement can be solved by a matching game with low computational cost.

In summary, all steps of SAMA are following the basic Markov approximation framework, where it repeatedly invokes two phases: exploring the set of nodes and exploiting to place vNFs into nodes. All looping steps will be stopped at the stationary distribution of Markov chain. The procedure of combination Markov approximation framework and matching theory is presented in details, as follows.

- **Initialized step.** The controller selects randomly a subset $M_0$ of active nodes that can deploy all vNFs. Set the current set as the optimal set $M^* \leftarrow M_0$.

- **Calculate the current configuration step.** Given the chosen subset $M^*$, so OPNET to get the minimum cost $C_f^*$.

- **Calculate the new configuration step.** This step is to generate a new configuration following the Markov chain model. To find the new state $f'$, randomly choose the new subset $M'$, satisfying the resource demands. In practice, the controller can base on the current active node set and turn-on/off one-by-one node randomly until satisfying the total resource demand.

In the new configuration $f'$, $X'$ and $C_{f'}$ can be obtained by solving $P_1$ as follows.
Chapter 5. Joint Approximation and Matching Approach for Network Service Chain Placement Problem

\[ \textbf{P1:} \min_{X} C(\mathcal{M}', X) \]  
\[ \text{s.t.} \quad \sum_{c \in C} \sum_{n \in \mathcal{N}} r^p_n \cdot X^c_{nm} \leq r^m_p, \]  
\[ \sum_{c \in C} \sum_{m \in \mathcal{M}'} X^c_{nm} = 1, \]  
\[ \sum_{c \in C} \sum_{m \in \mathcal{M}'} a^c_{nm'} X^c_{nm} X^c_{n'm'} \leq B_{mm'}, \]  
\[ X^c_{nm} = \{0, 1\}, \]  
\[ \forall c \in C, \forall n \in \mathcal{N}, \forall m, m' \in \mathcal{M}', m \neq m', \forall p \in \mathcal{P}. \]  

We can see that \textbf{P1} is still an NP-hard problem, which belongs to the set of bin-packing problems that can be cast as a the generalized matching game [14]. This problem is formulated in details based on matching game theory as shown in Section 5.3.

- **Randomly select the new configuration \( f' \) step.** This step is to select a new state following the Markov chain. After solving \textbf{P1} to find the optimal value \( C_{f'} \), the controller probabilistically stays at \( f \) or jumps to \( f' \) based on the transition probability (4.6). Following the design of transition probability, if a configuration \( f \) is better than \( f' \), the system will spend much time on its.

- **Convergence.** The system will converge to the stationary distribution similar to the algorithm in Section 4.3.1.

The advantage in SAMA can be observed by the state space of Markov chain that is now reduced from \( \mathcal{N}!2^K \) to \( 2^K \) in the worst case; however, the controller needs to find a placement scheme for \textbf{P1}.

The state space \( \mathcal{F} \) now is reduced by removing the combination between \( \mathcal{M} \) and \( X \). However, the new challenge appears in here is how to solve \textbf{P1} in terms of a low computational cost.
Chapter 5. Joint Approximation and Matching Approach for Network Service Chain Placement Problem

Algorithm 1: SAMA: Service chain Allocation based on Markov chain Algorithm.

1. Initialization: Create randomly a subset $\mathcal{M}_0$ of active nodes that can host all network service chains. Find the placement scheme $X_0$ and $C_{f_0}$ by solving $P1$. Set the current configuration $f_0$ to the optimal configuration $f^* \leftarrow f_0$ and $C_{f^*} \leftarrow C_{f_0}$.

2. Randomly change one node to create a new set $\mathcal{M}'$. Find $X'$ and $C_{f'}$ by solving $P1$.

3. Calculate $q$ by (4.6) to determine moving to $f'$ or staying with $f$. 4. Return to Step 2 until meet the criteria condition.

In this study, we use matching game approach for $P1$. This approach does not guarantee an optimal solution; however, it can achieve a stable solution in a low computational cost.

5.3 Network service chain placement algorithm based on matching game theory

Given $\mathcal{M}$ and $\mathcal{C}$, we have to solve $P1$ to find an optimal vNF placement scheme. However, $P1$ belongs to the set of bin-packing problems [34], which are NP-hard. To find an efficient solution, in this thesis, we advocate the matching theory framework [14, 26].

A matching game deals with two sides of players, here, it is similar to map vNFs into physical nodes. In that game, each vNF has a specific resource requirement, while physical nodes has limit resource capacity (quota). In state-of-the-art, this matching game has been widely used in resource allocation in cloud/datacenter such as in [54] and [53]. Applying matching game theory in resource allocation of NFV seems the first study in the literature.

5.3.1 Definitions of the many-to-one matching game

To handle a matching game in resource allocation in NFV, we define some basic concepts as follows.
Chapter 5. Joint Approximation and Matching Approach for Network Service Chain Placement Problem

Definition 5.1. The outcome of a vNF placement in problem P1 is a matching $\mu$. Formally, a matching is a function $\mu : \mathcal{N} \cup \mathcal{M} \rightarrow 2^{\mathcal{N} \cup \mathcal{M}}$ satisfying:

- $\mu(m) \subseteq \mathcal{N}$ such that $|\mu(m)|^p \leq r^p_m$, $\forall m \in \mathcal{M}, \forall p \in \mathcal{P}$, where $|\mu(m)|^p$ is the amount of aggregation resources of type $p$ of all vNFs that are matched to $m$.

- $\mu(n) \subseteq \mathcal{M}$ such that $|\mu(n)|^p = r^p_n$, or $|\mu(n)|^p = 0$, $\forall n \in \mathcal{N}, m \in \mathcal{M}, \forall p \in \mathcal{P}$, where $|\mu(n)|^p$ is the amount of resources of type $p$ of node $m$ that is matched to $n$ ($|\mu(n)|^p = 0$ means that vNF $n$ is unassigned).

- $n \in \mu(m)$ if and only if $\mu(n) = m$, $\forall n \in \mathcal{N}, m \in \mathcal{M}$.

As implied in the definition, a matching happens between two sides of agents, where a node can match with multiple vNFs. The outcome of a stable matching does not exist a pair that would like to exchange their matching results. The definition states that a many-to-one matching in network service chain placement is to find a stable and efficient allocation, where each player is “happy” with that matching. The matching totally depends on agent’s preferences over the opposite set.

Preference list of a node.

The preference list of node can be built based on the computing resource utilization. Each node prefers to place an vNF that can improve its resource utilization. It implies that a node prefers hosting any vNF to be idle. In this matching game, suppose that each node $m \in \mathcal{M}$ has a complete, strict, transitive preference relation $P(m)$ over the vNF set $\mathcal{N}$. We denote $n \succ_m n'$ to imply that node $m$ prefers vNF $n$ to vNF $n'$, and if $m$ prefers to remain unmatched instead of being matched to vNF $n$, i.e., $\emptyset \succ_m n$, then $n$ is said to be unacceptable to $m$. 
Preference list of a network service chain.

Intuitively, vNFs in a network service chain prefer to be located in the same node to reduce the interconnections between them [51]. Hence, network service chain ranks nodes based on their capacities that can be seen as the probability to host a network service chain. All vNFs in a network service chain have the same preference list, called the network service chain’s preference list. We assume that network service chain \( c \in \mathcal{C} \) has a complete, strict, transitive and preference relation \( P(c) \) over the node set \( \mathcal{M} \). We denote \( m \succ c m' \) to indicate that a network service chain \( c \) prefers node \( m \) to node \( m' \) and if \( c \) prefers to remain unmatched instead of being matched to node \( m \), i.e., \( \emptyset \succ c m \), then \( m \) is said to be rejected by \( c \).

In this work, we propose a novel approach, named as SAMA that is applied in the algorithm of Markov approximation, which can reduce the state space significantly. In particular, we formulate a network service chain placement underlying two sides of a many-to-one matching game with the main target is to find a stable allocation that can match vNFs into physical nodes. The advantages of this matching approach for the service chain placement problem include 1) covering multiple issues in the model of network service chain placement, 2) a suitable solution, in terms of stability and optimality and 3) an efficient algorithmic implementations that requires low complexity and fast convergence in implementation.

The outcome of a matching game [24] is a stable solution, where there is no pair of agents that would like to exchange their matching instead of keeping the current matching to improve their benefit. In this network of NFV, it implies that in a stable matching, a node does not want to exchange their allocation with others to improve its resource utilization. In the side of vNFs, an vNF does not prefer to place on another node to improve its network cost.

Another important concept in the matching game is a blocking pair, which is defined as follows.

**Definition 5.2.** Blocking par: A matching \( \mu \) is blocked by a pair of agents \((n, m)\) if there exists a pair \((n, m)\) with \( n \notin \mu(m) \) and \( m \notin \mu(n) \) such that \( n \succ_m \mu(m) \) and \( m \succ_n \mu(n) \). Such a pair is
called a blocking pair in general.

To satisfy all requirements from users, all network service chains have to be deployed into nodes. Hence, in a stable solution, there is no unsaturated network service chain, which is defined as follows.

**Definition 5.3.** Saturated network service chain: A network service chain $c \in C$, is saturated if all vNFs in SC $c$ are assigned. Similarly, a node is saturated if all its capacity is utilized. If node $m$ has available resources, then it will accept any vNF $n$ that $r^p_m \geq r^p_n, \forall p \in P$.

Consequently, we define a stable matching definition for network service chain placement as follows.

**Definition 5.4.** A stable matching: A stable matching is observed stable if (i) there is no blocking pair and (ii) all vNFs are embedded to nodes.

### 5.3.2 Network service chain and physical nodes’s preference lists

**Node’s preference list.**

To build the node’s preference list, we apply the Best-Fit algorithm [34] with the main target is to find vNFs that can improve node’s resource utilization. The algorithm finds one-by-one vNF that can shed the smallest space left when being placed into node $m$. The smaller the space left by the vNF, the higher the ranking it has in the preference list. All steps of building the preference list of a node is presented in Algorithm 2.

**Network service chain’s preference list.**

As mentioned above, each network service chain $c$ prefers to find a node that can place the most number of vNFs in the network service chain. Hence, we use the norm-based metric to deal with multiple resources of vNFs and nodes, which is calculated as follows
Algorithm 2: Creating a node’s preference list

**Input:** All vNFs in the requirement $\mathcal{N}$.

**Output:** Preference list $P(m)$

1. Find a vNF $n$ that sheds the smallest space when placing on $m$;
2. Add $n$ to $P(m)$;
3. Go to step 1 until all vNFs are added;

Algorithm 3: Creating a network service chain’s preference list

**Input:** The set of the chosen active nodes $\mathcal{M}$;

**Output:** $P(c)$.

1. Calculate $L_2$ for all node $m \in \mathcal{M}$;
2. Rank $m \in \mathcal{M}$ depending on increasing of $L_2$;

$$L_2 = \sum_{p \in \mathcal{P}} \omega_p (r^p_c - r^p_m)^2, \forall c \in \mathcal{C}, \forall m \in \mathcal{M}.$$ (5.6)

The $L_2$ factor can be seen as the resource deviation between node $m$ and network service chain $c$. A network service chain $c$ prefers to match with a node that has the lowest deviation. Based on $L_2$, a network service chain can rank for all available nodes.

5.3.3 Matching game algorithm for vNF placement problem

In this part, we propose an matching game-based algorithm to deal with vNF placement problem. The steps is similar to the basic many-to-one matching algorithm. To ensure the resource constraint before matching, all nodes that do not have enough resources to host any vNF will not considered in matching. We consider with multiple resource types in matching game, hence, we name it MDM.

All the steps can be sketched out as follows. First, all vNFs in each network service chain propose to available nodes following their shared preference lists. However, to guarantee the
order of the network service chain, each vNF proposes to nodes following the order in the network service chain. When a precedent vNF is accepted, the next vNF can be start for its process. A node then accepts/rejects vNFs depending on its quota and preference list. If its quota is lower than vNF requirement, or vNF is not in its list, the proposed vNF will be rejected by that node. All unassigned vNFs iteratively propose to the next nodes based on their remaining preference lists. When a node rejects the accepted vNFs to accept new vNFs, it will rejects all the other lower ranking vNFs, accepted before.

We present the pseudo code in Algorithm 4. Algorithm 4 starts with the unsaturated network service chain that exists an unassigned vNFs \( n \) (line 3). The algorithm start with the side of vNFs, where the unassigned vNF \( n \) selects the first node \( m \) in its preference list to propose (line 4) and wait for response. Node \( m \) will accept vNF \( n \) (lines 6-8) if it has enough resource. Otherwise, node \( m \) rejects \( n \) and announces to \( n \) by a rejected message. Node \( m \) can also reject all matched vNFs \( n' \) in previous steps such that \( n \succeq_m n' \) (lines 10-16) and vNF \( n \) also removes \( m \) out of its preference list \( P(n) \). The proposal will be start again for the next node.

Next, we prove that our algorithm converges and always obtains the stable matching in the matching outcome.

Theorem 1. The MDM algorithm eventually converges at finite steps.

Proof. We now prove the outcome by contradiction. We assume that the system has available resource to embed all submitted SCs. This assumption is possible in the concept of unlimited resource pool in a cloud [12]. Suppose that MDM matching produces infinity steps. It means that there is an unassigned vNF \( n \) in the result and all nodes rejected vNF \( n \). Hence, the preference list of vNF \( n \) does not exist any node to propose. It also implies that all node rejects \( n \) because of lack of available resource. It contrasts to the assumption above.

Theorem 2. The MDM algorithm obtains the stable matching in the matching outcome.

The design of MDM is to saturate and locate vNFs on their best nodes possibly according to their preference lists. Steps of MDM algorithm basically follows the deferred acceptance algorithm in [24] that is already proven the existing of stable matching in the result.
Algorithm 4: MDM: Multi-dimension matching algorithm for vNF placement

**Input:** The set $\mathcal{N}$ of vNFs, and the set $\mathcal{M}$ of nodes.

**Output:** Place all vNFs in $\mathcal{N}$ to nodes in $\mathcal{M}$.

while $\exists c \in C$, who is not saturated do

while $\exists n \in \mathcal{N}$ is unassigned do

$m \leftarrow$ Get the highest rank from $P(n)$;

if $r^p_m \geq r^p_n$, $\forall p \in \mathcal{P}$ then

Allocate $n$ to $m$;

$r^p_m = r^p_m - r^p_n$, $\forall p \in \mathcal{P}$;

end

else

Find all vNF $n'$ that can satisfy $n \succ_m n'$;

Reject all $n'$ and set $n'$ as unassigned, and update the resources of node $m$:

$r^p_m = r^p_m + r^p_n'$, $\forall p \in \mathcal{P}$;

Remove $n'$ out of $P(m)$;

Remove $m$ out of $P(n')$;

end

end

end
5.4 Discussion about the complexity analysis

Using the NFV architecture, the statistical states collected from resource nodes and network devices can be stored and processed in the controller.

As shown in [24, 53, 54], matching algorithm achieves a very low complexity in each procedure. The complexity of the matching approach only depends on the algorithm that we use to create preference lists for both sides. In this work, we use well-known heuristic algorithms. Such as For network service chain’s preference lists, the algorithm requires $O(M \log_2 M)$ complexity [34]. For a node’s preference list, the algorithm has the complexity is $O(MN \log_2 N)$. Therefore, the complexity of MDM can be measured totally in the worse case as follows $O(CM(C \log_2 M + N \log_2 N))$.

5.5 Distributed matching game approach for vNF placement problem

In this section we introduce the distributed approach to place vNF at each node, where a node can be controlled by a network controller (SDN) that can handle the vNF placement for itself.

The distributed process includes the node procedure and the vNF procedure. These procedures are presented in Algorithms 6, and Algorithm 5.

The node procedure. The detail procedure is shown in Algorithm 6 with steps as follows.

1. Step 1: Wait for proposed messages from vNFs.

2. Step 2: Accept vNF $n$ that sends the proposed message if it has enough resources and $n$ is in the preference list. If $m$ prefers $n$ to the current accepted vNFs, these accepted vNFs may be rejected to accept $n$. Otherwise, it sends the “reject” message to $n$ and deletes $n$ and all vNFs $n', n \succ_m n'$ from its preference list.

We use $|list|^p$ to denote the total amount of resource type $p$ of all accepted vNFs of node $m$. 
Algorithm 5: Distributed matching procedure of vNF $n$

**Input**: $\mathcal{M}$.

**Output**: A node candidate.

$\text{end\_status} \leftarrow \text{false};$

\begin{algorithm}
\begin{algorithmic}
\State \While{\text{end\_status} and $P(n) \neq \emptyset$} do
\State $m \leftarrow$ Pick the first node in $P(n)$
\State Send a proposed message to node $m$;
\State $msg \leftarrow$ Receive a message from a node;
\If{$msg=$ “reject”} \State Delete nodes out of $P(n)$, who send rejected messages; \EndIf
\If{$msg=$ “stop”} \State $\text{end\_status} \leftarrow \text{true};$ \EndIf
\EndWhile
\end{algorithmic}
\end{algorithm}

3. **Step 3**: Back to Step 1 until receiving a stop message or the preference list be empty.

**The vNF procedure**. Every vNF $n$ proposes nodes based on its preference list $P(n)$.

1. **Step 1**: Send the proposal to the first choice according to its preference list.

2. **Step 2**: Wait for the response. If the response message is “accept”, it does nothing. If the reply is “reject”, it removes the sender of that message from its preference list.

3. **Step 3**: Back to Step 1 until receiving a stop message or the preference list be empty.

Each node can execute this procedure in asynchronous way. When a node accepts or rejects any vNFs, it will send message to the controller. In this procedure, the stop message can be controlled by the controller to handle the whole agents.
Algorithm 6: Distributed matching procedure of a node $m$

Input: $\mathcal{N}$.

Output: Match vNFs in $\mathcal{N}$ to a given node $m$.

Set $\text{end\_status} \leftarrow \text{false}$ and $\text{list} \leftarrow \emptyset$;

while $\neg \text{end\_status}$ and $P(n) \neq \emptyset$ do

$\text{msg} \leftarrow$ Get messages from vNFs;

if $\text{msg} =$ “propose” then

$n \leftarrow$ Get the proposed vNF index;

if $r^p_n \leq r^p_m$, $\forall p \in \mathcal{P}$ and $n \in P(m)$ then

Send an accepted message to $n$;

Add vNF $n$ to $\text{list}$ and update resources of $m$: $r^p_m = r^p_m - r^p_n$, $\forall p \in \mathcal{P}$;

end

else if $r^p_n \leq r^p_m - |\text{list}|^p$ and $n \in P(m)$ then

foreach $n'$ with $n \succeq_m n'$ do

Send a rejected message to $n'$;

Update resources of $m$: $r^p_m = r^p_m + r^p_{n'}$, $\forall p \in \mathcal{P}$;

end

if $r^p_n \leq r^p_m$, $\forall p \in \mathcal{P}$ then

Sends an accepted message to $n$;

Add vNF $n$ to $\text{list}$ and update resources of $m$: $r^p_m = r^p_m - r^p_n$, $\forall p \in \mathcal{P}$;

end

end

else

Send a rejected message to vNF $n$;

end

if $\text{msg} =$ “stop” then

$\text{end\_status} \leftarrow \text{true}$;

end

end
5.6 Numerical and simulation results

To validate the efficacy of our proposed methods, in this section we provide the simulation settings and numerical results based on many pictures of comparisons.

5.6.1 Settings for simulation

As mentioned in the system model, we work with network service chain placement in one time slot. However, the diversity of network requirement in practice is high. It needs both evaluation in terms of one time slot and long term consideration.

We create increasing the amount of vNFs and network service chain by selecting a trace workload [22] within 10 time slots. Moreover, the setting of electricity price is collected from [11], as shown in Fig. 5.2b.

About the instances we use in this simulation, all instances are set according to 4 representative instances of middleboxes [48] including WAN optimizer instances, firewalls, IPSEC, and intrusion
Chapter 5. Joint Approximation and Matching Approach for Network Service Chain Placement Problem

![Graphs](image.png)

(a) Trace of workload.  
(b) Electricity price.

Figure 5.2: Setting of workload and electricity price.

detection systems (IDS).

For the monetary weight settings, \( \alpha, \beta \) are set to 0.1. We conduct the level of operational cost and traffic cost in varying effect of \( \sigma \) as shown in Fig. 5.3. It shows that when \( \sigma < 0.173 \), the network traffic cost is higher than operational cost, and vice versa. In that case, the network traffic cost is emphasized while the operational cost is deduced.

For the power of active node, \( Q_i \) is set 250 Watts, which is considered as the average power consumption of an active server in datacenters. Since we do not focus on how much the CPU of that server is used, we measure the power of active node as uniform. We equally set the priority of all resources, where \( \omega_p = 1, \forall p \in P \). We also simulate our work with the configurations of vNFs based on the standard, large and extra instance in [42].

Finally, based on the FNSS tool [8], a network topology is created with the values of each element in \( D \) belongs to [1, 5] as shown in Fig. 5.1.

5.6.2 Results

To measure the performance of our approaches, we conduct the performance of the proposed mechanisms compared to three baselines and prior approaches as follows:

- **Optimal**: We quantify the gap between SAMA and the optimal solution, which is solved by the JuMP solver [3]. Note that because of the inherent complexity in the network service
chain placement problem, the JuMP solver may take a lot of time to find the solution in the large-scale problem.

- Baseline 1: We name the consolidation approach which only focuses on the computing resources utilization as the baseline 1.

- Baseline 2: This baseline is named based on the approach that only relies on the network traffic policy.

- Markov-JuMP: Instead of using Matching approach for finding the solution of OPNET, we solve the subproblem $P_1$ by the solver, which is named Markov-JuMP.

- Anchor: This is the matching-based approach proposed in [54].

**Convergence evaluation.** We evaluate the convergence of Algorithm 4 and Algorithm 1, which is shown in Fig. 5.4a and Fig. 5.4b, respectively. We create four settings, including type 1: N=200, M=120, type 2: N=400, M=220, type 3: N=600, M=350, type 4: N=1000, M=520. The matching algorithm quickly converges to the stable allocation in all settings. With the large
Chapter 5. Joint Approximation and Matching Approach for Network Service Chain Placement Problem

(a) Evaluation of the fast convergence of the matching approach.
(b) Evaluation of the convergence of Markov approximation and SAMA.

Figure 5.4: Evaluation of the convergence in our proposed approaches.

Figure 5.5: Evaluation of the execution time.
Figure 5.6: Evaluation of bandwidth allocation.

Figure 5.7: Evaluation of computing resource utilization.
Chapter 5. Joint Approximation and Matching Approach for Network Service Chain Placement Problem

(a) Evaluation of the operational cost

(b) Evaluation of the network traffic cost.

(c) Evaluation of the total cost.

Figure 5.8: Evaluation of the system cost within 10 time slots.
Chapter 5. Joint Approximation and Matching Approach for Network Service Chain Placement Problem

(a) In case of single vNF per network service chain.

(b) In case of multiple vNFs per network service chain.

Figure 5.9: Comparison between SAMA and matching approach.
Table 5.1: Network traffic requirements.

<table>
<thead>
<tr>
<th>Service Chain</th>
<th>Chained vNFs</th>
<th>Traffic rate requirement (a)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Service</td>
<td>NAT-FW-WOC-IDPS</td>
<td>100 kbps</td>
</tr>
<tr>
<td>VoIP</td>
<td>NAT-FW-TM-FW-NAT</td>
<td>64 kbps</td>
</tr>
<tr>
<td>Online Gaming</td>
<td>NAT-FW-VOC-WOC-IDPS</td>
<td>50 kbps</td>
</tr>
</tbody>
</table>


vNF instances, one network service needs more nodes to host all its vNFs, hence, the rejected vNFs will be increased. It also make the number of iteration increasing. As shown in Fig. 5.4a, the convergence of MDM does not depend on $N$ and $M$. Based on this result, we can claim that matching approach can find a stable vNF allocation very fast which does not depending on the configuration setting of vNFs.

For SAMA, we conduct the convergence of Algorithm 1 in different values of $\delta$. $\delta$ is set largest up $10^5$, which leads to over the capacity of CPU in computation. The effect of $\delta$ is shown in Fig. 5.4b. When $\delta$ is larger, we can observe that SAMA can reach more closely to the optimal value, which the optimality gap is reduced. Considering the standard Markov approximation approach, it requires more iterations, which is more than 1000 to achieve the convergence.

**Execution time.** We evaluate the execution time of SAMA and compare to others in terms of increasing the number of vNFs. Our proposed approach has the longest execution time compared to others. However, the gap between SAMA and heuristic approaches is small as shown in Fig. 5.5. Matching approach can obtain the stable placement with short execution time compare to SAMA and heuristic algorithms, Baseline 1 and Baseline 2.

**Bandwidth allocation.** In different placement approaches, we evaluate the efficiency of network bandwidth allocation and compare SAMA to other approaches. In this evaluation, we limit
Chapter 5. Joint Approximation and Matching Approach for Network Service Chain Placement Problem

(a) The operational cost.

(b) The traffic cost in long-term evaluation.

(c) The incurred total cost in long-term evaluation.

(d) The number of active nodes in long-term evaluation.

Figure 5.10: Evaluation the performance in the long-run simulation.
Chapter 5. Joint Approximation and Matching Approach for Network Service Chain Placement Problem

<table>
<thead>
<tr>
<th>Instance Type</th>
<th>Memory</th>
<th>CPU</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firewall (small)</td>
<td>4 GB</td>
<td>2 vCPU</td>
<td>100 Mbps</td>
</tr>
<tr>
<td>Firewall (standard)</td>
<td>4 GB</td>
<td>8 vCPU</td>
<td>200 Mbps</td>
</tr>
<tr>
<td>Firewall (large)</td>
<td>4 GB</td>
<td>8 vCPU</td>
<td>300 Mbps</td>
</tr>
<tr>
<td>IDS</td>
<td>4 GB</td>
<td>6.5 vCPU</td>
<td>100 Mbps</td>
</tr>
<tr>
<td>IPSec (standard)</td>
<td>4 GB</td>
<td>4 vCPU</td>
<td>368 Mbps</td>
</tr>
<tr>
<td>IPSec (large)</td>
<td>4 GB</td>
<td>8 vCPU</td>
<td>580 Mbps</td>
</tr>
<tr>
<td>WAN-opt (standard)</td>
<td>2 GB</td>
<td>2 vCPU</td>
<td>20 Mbps</td>
</tr>
<tr>
<td>WAN-opt (large)</td>
<td>2 GB</td>
<td>4 vCPU</td>
<td>40 Mbps</td>
</tr>
</tbody>
</table>

Table 5.2: vNF instances.

the total bandwidth that can allocation to all physical links is 4500 Mbps and the number of nodes is limit with 50 active nodes. As shown in Fig. 5.6, SAMA can accept to allocation more network service chains compare to others until it reaches to limit of network bandwidth capacity. Baseline 2 needs to allocate network bandwidth that is the lowest compared to others; however, it reaches to the limitation of computing resource nodes. Meanwhile, Baseline 1 needs a lot of network bandwidth in its allocation scheme that results in the low number of accepted service chain.

**Resource utilization.** Fig. 5.7 shows the highest resource utilization of Baseline 1, while it is the worst placement scheme in Fig. 5.6. In contrary, Baseline 2 now achieves the lowest utilization in its placement scheme. Matching approach in both evaluation cases is acceptable that does not outperform to others. The resource utilization of SAMA is only lower than Baseline 1; however, accounting in two cases, SAMA shows the best choice to place the network services, where it can balance the resource utilization, network traffic allocation and the amount of accepted network service chains.
Chapter 5. Joint Approximation and Matching Approach for Network Service Chain Placement Problem

**System cost.** The system cost is measured within 10 time slots and compared SAMA with Baseline 1 and Baseline 2, as shown in Fig. 5.8. The resource utilization of Baseline 1 is expected as shown in Fig. 5.8a; however, it spends a lot of wear-and-tear cost as shown in time slots 7 to 9 and the network traffic as shown in Fig. 5.8b. For Baseline 2, this policy cannot achieve high resource utilization compared to others, but the network traffic cost is expected, as shown in Fig. 5.8b. The system cost of SAMA is better than others in all considered time slots as shown in Fig. 5.8c when controlling both operational cost and network cost. In particular, when we increase the workload, Baseline 1 and Baseline 2 disclose the drawback in their targets, which show the cost that does not improve. Within considered 10 time slots, SAMA reduces 19.1% and 9.28% of the total cost compared with the Baseline 1 and Baseline 2, respectively. Furthermore, we conduct the optimality gap with the solution of solver. Even though SAMA cannot get the close optimal solution, it shows the small game compared to Optimal as shown in Fig. 5.8d. Especially, to find the solution with JuMP solver for 100 node and 200 vNFs, it takes 8 hours, meanwhile SAMA spends 22 minutes to reach the close optimal solution.

**Compare the performance of SAMA and the Anchor framework.** We make two scenarios to analyze Anchor framework and SAMA. The first case is a single vNF instance in a network service chain. This setting prefers to show the importance of the inter-connection of vNFs. The result shows the similar total cost of SAMA and Anchor during 10 time slots, as shown in Fig. 5.9a. This result can be explained by the network traffic cost of all virtual links that are not changed during 10 time slots. Thus, OPNET mainly optimizes the operational cost. It implies the dynamic control in our approach, which the network provider can adjust depending on his target. For the second case, we measure SAMA and Anchor with multiple vNF instances in a network service chain. The matching Anchor does not optimize the network overhead, which makes the system cost increasing compared to SAMA, as shown in Fig. 5.9b.

**The long-run evaluation.** Our system allocates resources per each time slot, hence, we would like to measure the performance of the proposed algorithm over long-run duration.
In this case of evaluation, we also conduct with two kinds of network service chains to reflect the efficiency of proposed methods. For single vNF instance per network service chain, the result looks similar for both approaches, as shown in Fig. 5.10. For the second case, the difference is clarified in Fig. 5.10d. Baseline 1 and Anchor can use less number of active nodes to reduce operational cost compared to others. However, neglecting the wear-and-tear cost makes their operational costs higher than SAMA for long term, as illustrated in Fig. 5.10a. The efficiency of our method can be shown based on reducing costs by about 8.7% in operational cost compared to Baseline 1. SAMA also consumes similarly network traffic cost as that of Baseline 2. Hence, we can claim that SAMA can outperform in terms of the system cost compared to others.

**Wear-and-tear overhead.** At the end of simulation part, we evaluate the wear-and-tear cost for our proposed approaches compared to others as shown in Fig. 5.11. Reducing a lot of active nodes in Baseline 1 and Anchor, they have to pay more wear-and-tear cost in the comparison. Our approach does not turn off servers frequently that can improve this cost by about 27% compared to Baseline 1 during 10 time slots.
5.7 Applications

NFV has been a popular standard even though it is still infant. There are several applications of NFV that need to implement resource allocation algorithms, such as virtualization of mobile base stations, platforms as a service (PaaS), content delivery network (CDN), fixed access and home terminal-client environment. It shows the promising and signification ability on general purpose for network in future in which many major network equipment vendors have announced support for NFV.

Even though NFV shows the potential of automation and flexibility in implementation, the resources and functions in NFV are still needed enhance. There are several open source projects for NFV development, including OPNFV [5], OPEN SOURCE MANO [2], etc. One of them, OpenStack Tacker [6] shows a great promise in realizing the MANO framework, an official OpenStack project building a Generic VNF Manager (VNFM) and a NFV Orchestrator (NFVO) to manage Network Services and VNFs on an NFV infrastructure platform.

Through the web browser interface, we can deploy vNFs and modify the algorithm in Deployment module. Instead of using API of OpenStack, we modify and implement the proposed approach, SAMA, for the service chain placement. In the module vNF manager of OpenStack Tacker, we deploy the centralized algorithm, SAMA to control the placement decision. This module can also control the life cycle management operations of vNFs, which allocates the resources to run VMs. These functions are provided though Tacker API, for example, GET, GET/vnfds, POST/vnfds, DELETE/vnfds, PUT/vnfds.
Chapter 6

Conclusion and Future Directions

6.1 Conclusion

In this thesis, we have studied the joint operational and network traffic cost in network service chain placement. The contribution of our work for network service chain placement problem in NFV are briefly presented as follows:

- It enables a framework for network service chain placement to reduce the operational cost as well as the network traffic cost incurred in the system. The proposed method, SAMA, outperforms existing approaches in terms of total incurred cost and the resource utilization.

- It enables a tractable implementation framework for network service chain placement in both centralized and distributed implementation.

- A joint sampling and matching game approach can reduce the computation cost in finding the close-optimal solution in network service chain placement.

The main challenge of this research is to satisfy implementation requirements in network service chain placement in terms of multiple objectives. In particular, the network relationship between vNFs is represented by the specific order depending on network service deployment.
Each network service chain contains an ordered list of vNFs that are “stitched” together in the network implementation and operation. The order of vNFs in a service chain makes the barrier to apply well-known methods in virtual machine placement (e.g., bin-packing algorithms), where virtual machines are placed into the physical server independently.

Moreover, service chain placement problem is NP-hard, so that it requires an efficient solution to reduce the complexity of the placement algorithm. Currently, there is no joint consideration of multiple objectives in the service chain placement, which increases the complexity in computation. Thus, in this research, we analyse the service chain placement problem with different approaches. We also compare the complexity and the performance that convince the efficiency of our proposed method. The main challenge of this research is to satisfy the network relationship between vNFs. In particular, the network relationship between vNFs is represented by the specific order depending on network service deployment. Each network service chain contains an ordered list of vNFs that are “stitched” together in the network implementation and operation. The order of vNFs in a service chain makes the barrier to apply well-known methods in virtual machine placement (e.g., bin-packing algorithms), where virtual machines are placed into the physical server independently.

Moreover, service chain placement problem is NP-hard, so that it requires an efficient solution to reduce the complexity of the placement algorithm. Currently, there is no joint consideration of multiple objectives in the service chain placement, which increases the complexity in computation. Thus, in this research, we analyse the service chain placement problem with different approaches. We also compare the complexity and the performance that convince the efficiency of our proposed method.

It is clear that NFV brings flexible and agile implementations to network provider; however, it also rises the complexity in management and orchestration, especially, in case of resource allocation. Thus, NFV is requiring efficient solutions for this function, which is considered in this thesis. In chapter 3, we sketch out the system model of the network service chain placement and point out the drawback of current approaches. We formulate a joint multiple objectives for network service
Chapter 6. Conclusion and Future Directions

chain placement problem as a NP-hard combinatorial optimization problem. Unfortunately, it is impossible to find a solution in large scale.

Accounting multiple objectives in OPNET, it is significant to find a tractable solution that can reduce the computation cost in exploring the huge feasible placement schemes network service chain placement. We advocate the Markov approximation framework, a robust solution for a set of combinatorial network optimization problem, as shown in Chapter 4. However, the solution by using directly Markov approximation framework faces to long convergence. To tackle this issue, we propose a novel framework that combines Markov approximation framework and matching game theory, SAMA, as shown in 5. The performance study of SAMA shows the enhancement in convergence compared to MA. Furthermore, the total cost and the resource utilization also can improve significantly by 19.1% compared to consolidation method.

6.2 Future Directions

Although the proposed method can outperform current works in terms of reducing total cost, it still needs to improve the performance. In this model, we do not focus on the shared network functions in NFV that exists in the real implementation. For example, anti-virus functions can be deployed to multiple users without any reconfiguration. Considering a network provider, the shared functions can reduce significant resources used, however, this issue raises the high complexity of network service chain placement.

Furthermore, in this work, we ignore on the dynamic flow issue of network service placement problem, where the transmission rate between vNFs can be changed dynamically depending the deployed vNFs. For example, the transmission data rate is reduced after traversing firewall functions, or is changed after encoding via network encoding functions. In our model, we consider a static network data rate requirement for placement as usual model in resource allocation of NFV. To improve the performance in network service chain placement, this issue should be carefully considered. An approach with dynamic resource rate flows can reduce the redundant network
bandwidth allocation or improve resource utilization on the network system. However, it will increase the complexity in the system, which requires efficient solution to address.
Chapter 7

List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSS</td>
<td>Business Support System</td>
</tr>
<tr>
<td>CPE</td>
<td>Customer Premises Equipment</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol</td>
</tr>
<tr>
<td>EMS</td>
<td>Element Management System</td>
</tr>
<tr>
<td>EPC</td>
<td>Evolved Packet Core</td>
</tr>
<tr>
<td>ETSI</td>
<td>European Telecommunications Standards Institute</td>
</tr>
<tr>
<td>NAT</td>
<td>Network Address Translation (NAT)</td>
</tr>
<tr>
<td>NFV</td>
<td>Network function virtualization</td>
</tr>
<tr>
<td>NFVI</td>
<td>NFV infrastructure</td>
</tr>
<tr>
<td>MA</td>
<td>Markov approximation</td>
</tr>
<tr>
<td>MANO</td>
<td>Management and Orchestration</td>
</tr>
<tr>
<td>MWIS</td>
<td>The maximum weighted independent set</td>
</tr>
<tr>
<td>OPEX</td>
<td>Operating expenses</td>
</tr>
</tbody>
</table>
Chapter 8

List of Publications

8.1 Journals

International Journals


Chapter 8. List of Publications

13 December 2016.


8.2 Conference Papers

International Conferences


Chapter 8. List of Publications


Chapter 8. List of Publications


8.3 Domestic Conferences


[DC6] Chuan Pham, Choong Seon Hong, “Using Queueing Model to Analyze The Live Migra-
Chapter 8. List of Publications


Bibliography


