On the Orchestration and Provisioning of NFV-enabled Multicast Services

by

Omar Alhussein

A thesis
presented to the University of Waterloo
in fulfillment of the
requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2020

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Xianbin Wang Professor, Dept. of Electrical and Computer Engineering, Western University
Supervisor:	Weihua Zhuang Professor, Dept. of Electrical and Computer Engineering, University of Waterloo
Internal-External Member:	Jun Liu Professor, Dept. of Applied Mathematics, University of Waterloo
Internal Member:	Oleg Michailovich Professor, Dept. of Electrical and Computer Engineering, University of Waterloo
Internal Member:	Liang-Liang Xie Professor, Dept. of Electrical and Computer Engineering, University of Waterloo

Author's Declaration

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

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

Abstract

The paradigm of network function virtualization (NFV) with the support of software-defined networking has emerged as a prominent approach to foster innovation in the networking field and reduce the complexity involved in managing modern-day conventional networks. Before NFV, functions, which can manipulate the packet header and context of traffic flow, used to be implemented at fixed locations in the network substrate inside proprietary physical devices (called middlewares). With NFV, such functions are softwarized and virtualized. As such, they can be deployed in commodity servers as demanded. Hence, the provisioning of a network service becomes more agile and abstract, thereby giving rise to the next-generation service-customized networks which have the potential to meet new demands and use cases.

In this thesis, we focus on three complementary research problems essential to the orchestration and provisioning of NFV-enabled multicast network services. An NFV-enabled multicast service connects a source with a set of destinations. It specifies a set of NFs that should be executed at the chosen routes from the source to the destinations, with some resources and ordering relationships that should be satisfied in wired core networks.

In Problem I, we investigate a static joint traffic routing and virtual NF placement framework for accommodating multicast services over the network substrate. We develop optimal formulations and efficient heuristic algorithms that jointly handle the static embedding of one or multiple service requests over the network substrate with single-path and multipath routing. In Problem II, we study the online orchestration of NFV-enabled network services. We consider both unicast and multicast NFV-enabled services with mandatory and best-effort NF types. Mandatory NFs are strictly necessary for the correctness of a network service, whereas best-effort NFs are preferable yet not necessary. Correspondingly, we propose a primal-dual based online approximation algorithm that allocates both processing and transmission resources to maximize a profit function that is proportional to the throughput. The online algorithm resembles a joint admission mechanism and an online composition, routing, and NF placement framework. In the core network, traffic patterns exhibit time-varying characteristics that can be cumbersome to model. Therefore, in Problem III, we develop a dynamic provisioning approach to allocate processing and transmission resources based on the traffic pattern of the embedded network service using deep reinforcement learning (RL). Notably, we devise a model-assisted exploration procedure to improve the efficiency and consistency of the deep RL algorithm.

Acknowledgements

Words are not enough to convey my appreciation and gratitude for my remarkable supervisor, Prof. Weihua Zhuang. I am in ever-renewing awe of how Prof. Zhuang manages the research group, shuffle between many tasks, and yet maintain a just-right amount of macro and micro grasp and influence on the research directions and outcomes. Prof. Zhuang nurtured my writing, research, and critical thinking skills. More importantly, she gave me sufficient space to find my flow and research philosophy, and she trained me to be patient, agile, and resilient. I am whole-heartedly indebted to your guidance and support.

I wish to express my sincere gratitude to Prof. Xuemin Shen for myriad reasons. Our weekly group meetings with Prof. Shen are vital to keeping us all in check, not lose the big image, be enthusiastic, live as a family, work as a team, help each other, and maintain both horizontal and vertical grasp on the research in our field. And, thank you for the excellent research and presentation lessons, and for being friendly and approachable.

My sincere appreciation also goes to the internal thesis committee, Profs. Jun Liu, Oleg Michailovich, and Liang-Liang Xie, for devoting their valuable time and intellect to oversee my progress, and for the fruitful critiques and observations during the proposal. I am also very grateful to Prof. Xianbin Wang for gracefully agreeing to join us as an external examiner, and for setting aside his precious time and effort to examine the thesis and visit.

Also, my sincere gratitude goes to Xu Li, Jaya Rao, and Hang Zhang, from Huawei Technologies Canada, for their provoking insights, guidance, and support.

I wish to extend my deep gratitude and admiration to the members of the SDN research sub-group. It was an enriching, collaborative experience for everything I learned from each member and for the good times spent together. Thank you to Jiayin Chen, Thinh Phu Do, Junling Li, Kaige Qu, Yan Si, Weisen Shi, Peng Yang, Qiang Ye, Ning Zhang, and Shan Zhang. I would also like to thank the BBCR research group members, for it has been a rewarding experience and friendship with everyone.

I want to thank my remarkable MASc supervisors, Prof. Jie Liang and Prof. Sami Muhaidat, for being a crucial part of my academic journey. I am whole-heartedly indebted to your support, encouragement, and advice.

Going by Maslow's hierarchy of needs, this research would not have seen any light has it not been for feeling a sense of connection and love with my friends at the Waterloo region. Thank you, Tayyaba, for your friendship and for your spontaneous escapades, which return me my spirit. And my special regards and love to Ahmad Abdulrahman, Heba Alattas, Khalid Aldubaikhi, Mohammad Alhasani, Manaf Ben Yeha, Jamal Busnaina, Hazem Elgabra, Ilyas Farhat, Omar Farhat, Farid Farmani, Jinane Harmouche, A-long Jin, Jobanmeet Kaur, Hesham Moussa, Melad Olaimat, Hassan Omar, Mehdi Parvizi, Romain Ru, Amr Salah, Mohammad Shahab, and Mohammad Shorman.

Finally, I want to thank Ma'ali for her ever-growing love, support, and patience. And, my deepest gratitude and affection goes to my parents, brothers, and sister (Ghina), for their unparalleled love! One call with them has metaphysical effects.

Dedication

To my beloved parents,
Ahmad Alhussein and Intisar Bataineh

To my beloved sister,
Ghina

To my beloved fiancé,
Ma'ali

"Upward, yet not northward,"

–Edwin A. Abbott, Flatland: A Romance of Many Dimensions

Table of Contents

List of Figures

Acronyms

5G	5th Generation
DDPG	Deep Deterministic Policy Gradient
DQN	Deep Q-network
E2E	End-to-End
MILP	Mixed Integer Linear Program
MST	Minimum Spanning Tree
NF	Network Function
NFV	Network Function Virtualization
RL	Reinforcement Learning
SDN	Software-defined Networking

Nomenclature

$\mathbb{R}$	The achieved aggregate throughput obtained by hosting network services (Chapter 3)
$\mathcal{A}$	Action space for a Markov decision process
$\mathcal{D}^r$	Set of destination nodes for service r
$\mathcal{F}_i$	Set of NFV nodes that can host NF f_i, where $\mathcal{F}_i \in \mathcal{N}$
$\mathcal{L}$	Set of physical links in network substrate $\mathcal{G}$
$\mathcal{M}$	Set of NFV nodes in network substrate $\mathcal{G}$, where $\mathcal{M} \in \mathcal{N}$
$\mathcal{N}$	Set of nodes in network substrate $\mathcal{G}$
$\mathcal{N}(0, \sigma_n^2)$	Gaussian noise with zero mean and σ_n^2 variance
$\mathcal{P}(r)$	Set of all possible paths/trees for unicast/multicast service r
$\mathcal{R}$	Set of service requests
$\mathcal{G} = (\mathcal{N}, \mathcal{L})$	Network substrate $\mathcal{G}$, where $\mathcal{N}$ and $\mathcal{L}$ are the set of nodes and links
$\mathcal{G}_M = (\mathcal{N}_M, \mathcal{L}_M)$	A directed multilayer graph, where $\mathcal{N}_M \subseteq \mathcal{N} \times \mathcal{X}$ is the set that contains all nodes, in which a node $(n \in \mathcal{N})$ is present in a corresponding layer $(x \in \mathcal{X})$
$\mathcal{G}_v$	Virtual multicast topology
$\mathcal{L}_A$	Set of all intra-layer edges, $\mathcal{L}_A = \{((u,a),(v,b)) \in \mathcal{L}_M \mid a = b\}$, which represent the conventional connections between the network elements in each layer
$\mathcal{L}_I$	Set of all inter-layer edges, $\mathcal{L}_I = \{((u,a),(u,b)) \in \mathcal{L}_M \mid a \neq b\}$, which are used to encode the placement decisions in which a function is processed in transition from $a \in \mathcal{X}$ to $b \in \mathcal{X}$
$\mathcal{L}_M$	Set of all inter- and intra-layer edges in $\mathcal{G}_M$, where $\mathcal{L}_M \subseteq \mathcal{N}_M \times \mathcal{N}_M$
$\mathcal{S}$	State space for a Markov decision process
$\mathcal{V}^r$	Set of all NFs for service r

$\mathcal{V}_b^r$	Set of best-effort NFs in service r
$\mathcal{V}_m^r$	Set of mandatory NFs in service r
$\mathcal{X}$	Set of all layers in the multi-layer network, $\mathcal{G}_M$
Ω_m^n	Set of integers from m to n ($> m$), i.e., $\Omega_m^n \triangleq \{m, m+1, \dots, n\}$ with $m, n \in \mathbf{Z}_+$
ϵ	Probability of invoking the step-wise solution (in Chapter 5)
λ	Policy in a reinforcement learning setup
$\mu(\boldsymbol{s}\|\boldsymbol{\theta}^\mu)$	Parameterized actor policy μ with parameters $\boldsymbol{\theta}^\mu$
$\kappa_{li}^P(\tau)$	Continuous flow variable that represents the fraction of flow used in link l to direct traffic from f_i^P to f_{i+1}^P of path P
π^{jr}	Binary decision variable that indicates tree j of service r is activated
σ	An ongoing input of sequence of unicast and multicast service requests, $\sigma = (S^1, S^2, \dots)$
$\boldsymbol{\theta}^Q$	Parameters of deep neural network for action-value function Q
ρ^r	Profit value that correspondes to the processing resources that is accrued from accepting service r
ζ^r	Binary decision variable, where $\zeta^r = 1$ when service r is accepted
ϱ^r	Profit value that correspondes to the transmission resources that is accrued from accepting service r
η^r	Integer decision variable, with $\eta^r = \eta_b^r$ indicating that the set of best-effort NFs from r is included, and $\eta^r = \eta_m^r$ indicating otherwise.
τ	Discrete timestep in a reinforcement learning setup
υ	Rate of updating the primary deep neural network in the deep deterministic policy gradient algorithm
ϑ	Discount factor which represents the value of future rewards, $\vartheta \in (0, 1]$
ξ	Competitive ratio of the online algorithm (in Chapter 4), where $\xi = \max\{\phi, \varphi\}$
A	The value of the objective function of the primal as a result of the online algorithm (in Chapter 4)
$B_l(\tau)$	Residual transmission resources of link $l \in \mathcal{L}$ at time τ (in Chapter 5)
$B(l)$	Residual transmission resources of link $l \in \mathcal{L}$

$C(f(\tau))$	The processing requirement for NF f at time τ (in Chapter 5)
$C(f_i^r)$	The processing requirement of the i^{th} NF for service r
$C_n(\tau)$	Residual processing resources in node n in packet/s at time τ (in Chapter 5)
$C(n)$	Residual processing resources in node n in packet per second
D	The value of the objective function of the dual as a result of the online algorithm (in Chapter 4)
J^r	The maximum number of multicast trees to deliver multicast service r
K	An ongoing input of sequence of unicast and multicast service requests, $\sigma = (S^1, S^2, \dots)$
L	Maximum number of hops for a unicast service (or maximum number of links in a tree for a multicast service)
M	Big-M constant
$\tilde{R}_\tau$	The discounted accumulation of future rewards in a reinforcement learning setup
$Q^\lambda(s_\tau, a_\tau)$	Action-value function which is the expected future return given state s_τ and action a_τ while following policy λ
$Q(\boldsymbol{s}_\tau, \boldsymbol{a}_\tau \vert \boldsymbol{\theta}^Q)$	Parameterized action-value function with parameters $\boldsymbol{\theta}^Q$)
U^r	The size for service r, where $U^r = R^r(1 - g^r)$
$W_{i,i+1}^{t,k}$	The kth path between two embedded NFs (f_i, f_{i+1}) along the network substrate for destination t ($\in \mathcal{D}$)
Y^r	The area of the smallest convex polygon that spans all destinations of service r
Y	The area of the smallest convex polygon that spans all nodes in the network
d^r	Data rate requirement for service request r; superscript r is dropped in single-service formulations
d^{jr}	Fractional transmission rate of tree j for service request r. Superscript is dropped in single-service formulations
$d(\tau)$	Data rate requirement for a service request at time τ in packet/s (in Chapter 5)

$d^P(\tau)$	Fraction of flow assigned for path P at time τ (in Chapter 5)
f_i^r	Represents the i-th network function for service r
g^r	Distribution level of service request r
q^r	The distance from source to the center point of the set of destinations in service r
q	The largest distance between two arbitrary nodes in the substrate network
r_τ	Reward at timestep τ
s^r	Source node for service r
u_{nit}^r	Binary decision variable that indicates whether an instance of f_i is deployed on node n for destination t of service r
x_{li}^{jr}	Binary decision variable that indicates that link l ($\in \mathcal{L}$) is used for forwarding traffic for service r in multicast tree j from f_i^r to f_{i+1}^r
$x_{li}^P(\tau)$	Binary decision variable that indicates that link l ($\in \mathcal{L}$) is used for forwarding traffic for path P from f_i to f_{i+1} at time τ (in Chapter 5)
$\bar{x}(l)$	Primal variable that represents the cost of link l ($\mathcal{L}$)
$\tilde{x}(n)$	Primal variable that represent the cost of node n ($\mathcal{N}$)
y_{lit}^{jr}	Binary decision variable that indicates that link l is used to direct traffic for service r in multicast tree j from f_i^r to f_{i+1}^r for destination $t\,(\in \mathcal{D}^r)$
y_P^r	The fraction of flow allocated for service S^r along path P
z_{ni}^r	Binary decision variable that indicates that an instance of f_i is deployed on NFV node n for service r
$z_{ni}^P(\tau)$	Binary decision variable that indicates that an instance of f_i is deployed on NFV node n for path P
z^r	Primal variable that is associated with service r

Chapter 1

Introduction

1.1 Next-generation Networks

Data communication networks have endured a long-lasting legacy. Nonetheless, managing today's communication networks has become very tedious and complex. In addition, the demand for mobile and cellular communications is expected to grow at an unimaginable rate due to the increasing utilization of mobile broadband services, the emergence of ubiquitous Internet of Things devices, new bandwidth-hungry applications such as virtual and augmented reality, and vehicular communication technology. Consequently, the 5th Generation (5G) era is to contain new use cases (some of which are extreme), business models, and value creation, which is set to transform the status-quo socio-economic reality.

In recent years, communication networks have been experiencing a radical and fundamental change in the way they are designed and managed. This shift is mainly due to two paradigms, namely Software-defined Networking (SDN) and Network Function Virtualization (NFV). SDN and NFV are considered to be crucial technical approaches for next-generation networks, representing two driving innovation platforms in the upcoming 5G era and beyond.

In SDN, the control plane is decoupled from the data plane. The underlying data plane elements perform tasks according to the instructions given by the control plane, which provides the control plane with a network-wide view and total control of the network substrate. On the other hand, NFV refers to the decoupling of functions from proprietary

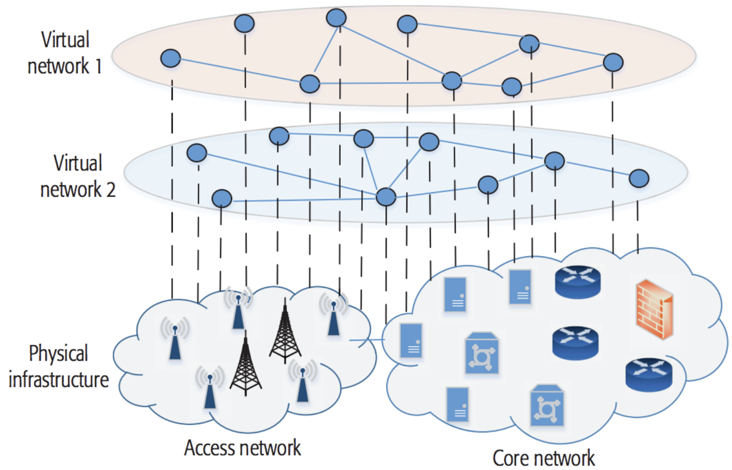

Figure 1.1: Service-oriented virtual networks [1].

physical devices to instantiate them as software-based virtual Network Functions (NFs) on off-the-shelf commodity servers using common virtualization infrastructures. Together, SDN and NFV provide a global view of the network substrate and a new degree of freedom to deploy virtual NFs instances in the data plane on demand, respectively.

Via NFV and SDN, an approach for establishing service-customized networks has a potential to greatly resolve the complexity in conventional networks and to meet new demands and use cases [1, 3–7]. A service-customized network provides traditional connectivity between a set of terminals, and allows for the deployment of abstract network applications in the data plane. An NFV-enabled network service can be represented by a logical topology, called NF chain, which specifies a set of virtual NFs that are orchestrated and deployed along the chosen routes from the source(s) to the destination(s), with some properties and dependencies satisfied. That is, a network service can manipulate the packet header and context of traffic flows, in addition to providing traditional connectivity to a class of end users. Some examples of virtual NFs include cache, transcoder, firewall, 5G evolved packet core, wireless access network optimizer and network address translator, which can be hosted and dynamically employed at NFV nodes. As shown in Fig. 1.1, several virtual networks are virtually overlayed onto the physical network substrate. Here, the NFV orchestrator is responsible for determining the routing policy for each network service, and the placement and instantiation of the NFs on the NFV nodes. In contrast, the SDN controller realizes the vision of the NFV orchestrator by populating the routing tables on the forwarding elements of the network substrate.

In this approach, the service provider requests a network service to satisfy certain expected demands and requirements as per the service level agreement. In turn, the responsibility of the service orchestrator or the network operator is to efficiently design such a virtual network, and place it on the physical network substrate. The network operator aims at reducing the provisioning cost, and maximizing the utilization of its network, while simultaneously meeting the binding service level agreement.

To further improve resource utilization, service providers are increasingly demanding service requests with multicast traffic to provide efficiency through the use of packet replication at network edges [8]. For the case that multiple destination nodes in the core network require the same information content, the source node transmits each packet only one time, and then packet replication occurs at edges close to the destinations. It has been shown that the multicast mode of communication can reduce the bandwidth consumption in the backbone network by over 50% in contrast to unicast mode [9].

The primary motivation of this thesis is to devise an orchestration and provisioning framework with three complementary components pertaining to the orchestration, the admission mechanism, and the dynamic provisioning of NFV-enabled multicast services.

1.2 Research Objectives

Fig. 1.2 illustrates an example of a multicast NF chain for a typical video streaming service that distributes content to several destinations. Note that a multicast replication point occurs before the Cache instances. Semantically, multicast replication points are not restricted and can occur anywhere in the NF chain. Therefore, a network service can have multiple NF chain topologies that are semantically equivalent. Thus, the orchestrator can deploy multiple NF instances in the data plane, whereas a routing policy forwards the traffic from the source to each destination while traversing the NF instances as per the network service. Here, an NF chain has both transmission and processing resources that need to be considered. Therefore, one research problem is to investigate the joint routing and NF placement of flexible multicast service. That is, we investigate how many NF instances should be placed, where to place the NF instances, and how to route the traffic

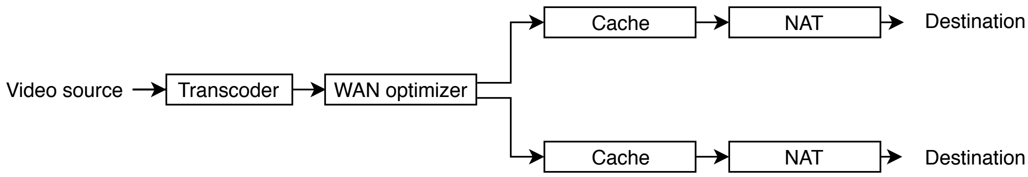

Figure 1.2: Illustraion of a multicast NF chain for a basic video streaming service.

from the source to destinations while traversing the NF instances. Considering the NF placement process along with traffic routing entails a new fundamental challenge, whereby a non-trivial tradeoff between the processing and transmission resources interplay. Such a tradeoff becomes more conspicuous when considering a flexible multicast service in which traffic is routed to more than one destination while not restricting the location of the multicast replication points.

Multiple network services can cohabit the network substrate which has limited processing and transmission resources. Therefore, the embedding of multiple service requests needs to be jointly considered. The order and method by which one service request is embedded onto the network substrate can affect the overall utilization of the network substrate and the efficiency of embedding future service requests. Therefore, a second research problem is how to treat multiple service requests both in batches and (more importantly) in an online setup.

The scale of demands in core networks is large relative to the routing granularity. Moreover, NFV-enabled services can exhibit time-varying traffic demand which need not be periodic nor tractable. Therefore, the third research problem is on the provisioning of a network service while considering the dynamics of the network service.

Next-generation networks are endowed with enhanced capabilities, thanks to SDN and NFV, albeit with new challenges. In this thesis, a general theme is to strive for flexible, versatile, and modular orchestration and provisioning methods to cater to the new era of next-generation networks. In the following, we identify three research questions related to aspects of the orchestration, admission, and dynamic provisioning of NFV-enabled services to present the respective research objectives.

1.2.1 Joint Routing and NF Placement for Multicast Services

If a multicast service requires connecting some terminal nodes without intermediate NFs, the optimal routing that reduces the link provisioning cost can be found by constructing a Steiner tree or one of its variants. A Steiner tree is a generalization of the Minimum Spanning Tree (MST) which finds the subset of weighted edges (and nodes) that connect all vertices in a graph with the minimum possible link provisioning cost. Constructing a Steiner tree is an NP-hard problem [10, 11]. However, polynomial time approximation algorithms exist to build a Steiner tree. With the emergence of SDN providing a global view of the physical network topology and network states, Steiner tree-based routing approaches become feasible [12–14]. However, such methods do not incorporate NFs in their formulation, and cannot be extended directly to the joint multicast routing and NF placement problem. Practically, there exist a massive number of NF placement configurations, each of which requires a multicast routing topology construction (e.g., one instance of such configurations is shown in Fig. 1.2).

The NF placement and multicast routing are correlated, which leads to technical challenges for orchestrating a single NFV-enabled multicast service. Selecting just enough NFV nodes for NF placement inevitably increases the link provisioning cost for building an appropriate multicast routing topology; Conversely, instantiating NF instances on more NFV nodes may yield a decreased link provisioning cost with traffic load balancing at the expense of an increased function provisioning cost. Therefore, how to balance the tradeoff between link and function provisioning costs is a challenging issue.

Another major challenge stems from the fact that multiple network services share the network substrate. As the network substrate has limited transmission and processing resources, the efficiencies of embedding multiple service requests interplay. Prioritizing a low-rate network service for embedding can fragment the network resources, thereby hindering other high-rate network services from being successfully (or efficiently) embedded.

Some recent studies address the orchestration of NFV-enabled multicast service to minimize the function and link provisioning costs [2, 15–19]. However, most existing works assume a design scenario where all NFs are hosted in one NFV node for each network service and multicast replication points can occur only after the deployment of NFs. More realistic and flexible design (e.g., one NFV node is only capable of hosting specific types of NFs, multipath routing is enabled between NFs) can make the existing solutions not feasible or not optimized. More details of relevant literature are discussed in Subsection 2.3.1.

In Problem I, the objective is to develop an optimization framework for the orchestration of multiple multicast service requests over the network substrate. We consider flexible multicast service requests, whereby multicast replication points need not be restricted in the NF chain. First, we study a joint multicast traffic routing and NF placement problem for a single service request to minimize the function and link provisioning costs, under the physical network resource constraints, flow conservation constraints, and NF placement rules. Second, we investigate the *static* embedding of multiple service requests over the physical network substrate, i.e., how to determine the optimal combination of multiple services for embedding and their joint routing and placement configurations, such that the aggregate throughput of the network substrate is maximized and the function and link provisioning costs are minimized.

1.2.2 Online Joint Routing and NF Placement for Unicast and Multicast Services

In the literature, including the research outcomes from Problem I, a considerable number of works are carried out for the static orchestration of NFV-enabled service requests [20–29]. Earlier research considers the orchestration of a single service request, where the focus is on minimizing the provisioning cost of a single service without taking other services into consideration [19, 26, 30, 31]. However, as mentioned in Section 1.2.1, the admission and embedding of one service request affects the service provisioning of other requests. In Problem I, we consider a static scenario where all service requests are known a priori, i.e., all service requests are assumed to arrive in one batch. In practice however, network services arrive in an online manner without knowledge of future requests [32, 33].

Due to the increased flexibility and agility brought-forth by NFV, future (over-the-top) service requests are envisaged to be hardly predictable [34]. Future service quality and data traffic patterns for new use cases are arguably not well understood, and advanced knowledge of future patterns can be difficult to obtain or predict. Moreover, such traffic patterns can vary dramatically over short periods due to the inherent agility of NFV-based networks.

Some relevant studies deal with the online handling of service requests without statistical assumptions [17,32,33,35–37]. The NFV-enabled frameworks are based on the seminal work by Awerbuch et al. [38], where some new aspects are due to the inclusion of NF instances and the need for an admission mechanism for service requests with multiple resource types. To our knowledge, in the existing NFV-enabled works, service requests have either one resource type or one NF instance. Also, online (routing and NF placement) algorithms can be classified as either *all-or-nothing* or *all-or-something*. In the all-or-nothing scenario, service requests need to be fully served in the network substrate. In the all-or-something scenario, services can be partially (fractionally) served, e.g., admitting a service request while reducing the required data rate [34,39]. Current works in the relevant NFV literature can be considered as all-or-nothing schemes. More details of relevant literature are discussed in Subsections 2.3.2 and 2.3.3.

Problem II deals with two resource types simultaneously, namely the processing and transmission resources. The two resource types are often conflicting in their utilization. Therefore, there is a need to design a generalized admission mechanism and an online joint composition, routing, and NF placement algorithm that takes the multiple resource types into account. We consider unicast and multicast service requests that can have multiple NF instances. Furthermore, we consider two NF types, namely best-effort and mandatory. Successful placement of a network service is contingent only on successfully placing the set of mandatory NFs. The functionality of a best-effort NF is not necessary for the correctness of a network service [40]. Therefore, the set of best-effort NFs can be removed from a service request when it is deemed "too prohibitive". In practice, best-effort NFs can improve either the performance, the quality of service, or the security of a network service, such as in the case of compression and intrusion detection. Consider for instance a video/image compression NF type for a voice over IP network service. Such an NF type enhances the quality of service by compressing the incoming data flow. However, when the available processing resources (or the available subscription) for the NF type in the network substrate are scarce, a network service can take a rather unnecessarily long (i.e., expensive) route, which would be too costly and can conversely degrade the overall quality of service. Therefore, such NF type can be declared as best-effort, whereby including it should be contingent on whether a certain profit is achieved.

In Problem II, the objective is to develop a robust admission mechanism and an online joint composition, routing, and NF placement framework (online algorithm, in short) that aims to maximize a profit function, which is proportional to the so-called amortized throughput, while considering unicast and multicast NFV-enabled services with best-effort and mandatory NF instances, subject to resource constraints on physical links and NFV nodes. The amortized throughput is defined as the weighted total transmission and processing resources reserved for all the accepted service requests.

1.2.3 Model-free Dynamic Provisioning of NFV-enabled Services

In both Problems I and II, we consider service requests with a fixed data rate requirement. However, in the core network, traffic patterns can exhibit time-varying patterns. Inspired by the service composition (or topology design) literature [30,41,42], first we recognize that in many network scenarios, semantically several NF instances can be deployed in parallel or a sequential manner. Considerl, for instance, a firewall-protected Cache dissemination service, where a source sends a signal to instruct the cache to disseminate some information to a destination (as shown in Fig. 1.3-(a)). The illustrated network service is comprised of a Firewall and a Cache. Here, Fig. 1.3-(b) is another valid NF chain topology in which the firewall is implemented in a distributive manner and the cache is parallelized. Deploying the firewall in a distributed manner can be needed when the resources for such an NF type are low on each NFV node. Moreover, splitting the cache can be more processing-efficient when the traffic demand increases or when the processing resources are particularly scarce close to destination 1. Therefore, in a time-varying environment, the topology of the network service can alter between different topologies depending on the dynamics of the traffic demand and the network substrate. For instance, the topology of the network service in Fig. 1.3 can alternate between Fig. 1.3-(a), Fig. 1.3-(b).

Here the problem can be regarded as a dynamic joint composition, routing, and NF placement with time-varying traffic demand. Deciding a logical NF chain can be ineffective if not considered jointly with the traffic routing and NF placement. The three aforementioned problems, namely the service composition, traffic routing, and NF placement, are joint problems with correlated and conflicting effects. Communication networks are becoming increasingly time-varying, dynamic, and more challenging to model. In practice, optimization-based methods produce rigid solutions with arguably small degrees of freedom. And, a deviation from an assumed model can lead to a significant degradation in the performance of the underlying solution (such as in the probabilistic routing literature [43]). To this end, the research community is collectively in the nascent stage of

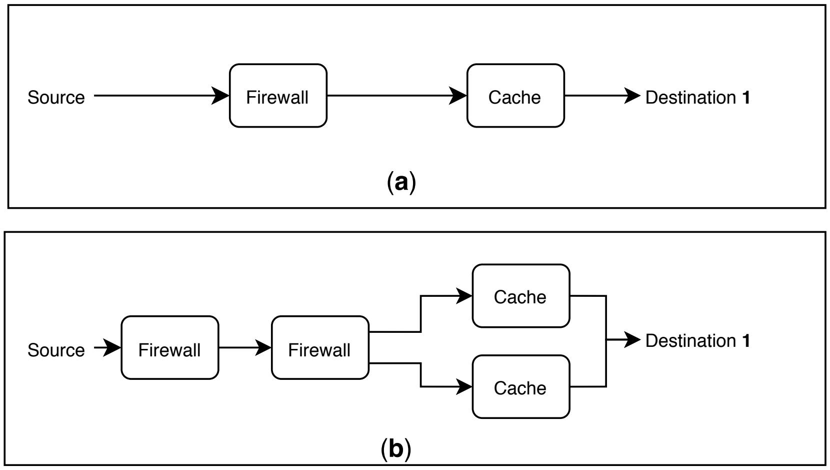

Figure 1.3: Two NF chain topologies that realize the same network service request. The network service is firewall-protected and disseminates traffic to several destinations from a web-based cache which can be replicated.

exploring model-free data-driven provisioning methods that are powered by contemporary machine learning [44–46].

From a design perspective, the majority of the works focus on NF (and link) migration to deal with the time-varying traffic patterns and to alleviate network bottlenecks. Instead of considering complete NF instance and link migrations, we aim to enforce the dynamic scaling of the network service by initializing (and tearing down) new NF instances along with already placed NF instances, which is desired for time-varying traffic characteristics. For instance, the topology of network service in Fig. 1.3 can alternate between Fig. 1.3-(a), Fig. 1.3-(b) and other topologies based on the characteristics of the network substrate and the time-varying traffic demand.

While some network service topologies can ameliorate the efficiency for some types of resources (e.g., function and link provisioning costs), it can exacerbate other types of resources (e.g., the routing and signaling overhead). Therefore, in addition to the transmission and processing resources, the routing overhead should be taken into consideration. Moreover, it is crucial to consider the NF setup which incurs a one-time cost. Due to the routing overhead and the cost of setting up a new NF instance, a myopic alteration of the routing and NF placement topology should be discouraged. Therefore, there is a need to learn from the traffic pattern of the service request to minimize the NF setup cost in the long run. The traffic pattern of a network service change in real-time; therefore, we assume a proactive framework that adjusts the topology and embedding solution by relying on the upcoming data rate, which can be predicted to a great accuracy by some recent works [47,48], thanks to the recent advents in deep learning and deep Reinforcement Learning (RL). Here, without focusing on traffic prediction, we assume that the upcoming data rate is given.

In Problem III, the objective is to develop an efficient RL-based dynamic provisioning approach to allocate processing and transmission resources based on the traffic dynamics of the network service and the network substrate while taking into account the routing overhead and the NF setup cost. The intuition is to design a dynamic provisioning solution that can change based on experiences learned from the traffic patterns. We consider that multipath splitting and NF splitting can be invoked due to the variations in the traffic pattern in the network service; this is in contrast to invoking NF and link migrations which can be ineffective in a large-scale time-varying environment.

1.3 Research Contributions

In this section, we present the research contributions for each research problem.

1.3.1 Joint Routing and NF Placement for Multicast Services

In Chapter 3, we first develop a joint multicast traffic routing and NF placement framework for a single service request to minimize the function and link provisioning costs, under physical processing and transmission resource constraints, flow conservation constraints, and NF placement rules [19]. For practical applications, our formulated problem focuses on *flexible* multicast routing and NF placement, where we allow *one-to-many* and *many-to-one* NF mappings. That is, several NF instances can be hosted at one NFV node if permissible, and one type of NF can be replicated and deployed on different NFV nodes as NF *instances* to serve different sets of destinations. In doing so, we do not impose any constraints on the locations of the multicast replication points, i.e., the deployment of NF instances can occur both before and after the replication points in the multicast topology, thereby providing considerable flexibility in the deployment of network services. Furthermore, our formulated problem incorporates both single path and multipath traffic routing between the embedded NF instances. Since the formulated problem is NP-hard, we devise a low-complexity heuristic algorithm to obtain an efficient and flexible solution, based on a key-node preferred MST approach.

Second, we consider a general scenario of placing multiple service requests over the physical network [20]. Since multiple services compete with each other to be hosted on a substrate network with limited resources, we accept the services which maximize the aggregate throughput with the least provisioning cost. We formulate an Mixed Integer Linear Program (MILP) that jointly determines multicast topologies for multiple service requests, where we find the combination of network services that maximize the aggregate network throughput while minimizing the overall function and link provisioning costs. Moreover, we develop a simple heuristic algorithm that prioritizes the service requests, aiming at maximizing the aggregate throughput with the minimum overall provisioning cost.

1.3.2 Online Joint Routing and NF Placement for Unicast and Multicast Services

In Chapter 3, we develop an online algorithm that consists of two main components, namely (i) an admission mechanism that rejects or accepts a service request based on a profit function while taking best-effort NFs into consideration, and (ii) an online joint composition, routing, and NF placement algorithm that provides unicast-enabled and multicast-enabled routing and NF placement configurations for the admitted service requests [49]. The online algorithm is developed through a primal-dual analysis, which provides an approximately optimal result with provable competitive performance. A formal definition of the competitive ratio (performance) is stated as follows. For a profit-maximization problem, let $A_{\text{OPT}}(\sigma)$ be the profit of the (optimal) offline solution for a sequence of requests (σ). An online algorithm is c-competitive if the produced solution is feasible and its profit is at least ${}^{A_{\text{OPT}}(\sigma)}/_{c} - e$, where e is an additive term that is independent of the service requests [39]. The primal-dual approach exists for solving offline optimization problems. Buchbinder and Naor extended the framework for the treatment of online algorithms [39]. Problem II offers the following new contributions:

- We propose a primal-dual based online algorithm to allocate both processing and transmission resources for network services with multiple NF instances. In addition, we consider heterogeneous NFV-enabled services with mandatory and best-effort NFs. We provide a natural generalization to relevant works that focus on the provisioning of services without a processing requirement or with only one NF instance. The online algorithm can be regarded as an all-or-nothing/all-or-something algorithm in the sense that the requested data rate and the required processing resource for each NF instance should be fully satisfied, yet best-effort NF instances need not necessarily be included in the accepted service request, thereby providing the flexibility to recompose the logical topology of a service request before admission;

- The primal-dual analysis offers an alternative analysis and generalized treatment to approaches adopted in recent relevant works [17, 32, 33, 36, 37]. For instance, the competitive performance in the aforementioned works is shown to associate not only with an optimal integer solution but also with an optimal fractional solution;

- We propose a "one-step" algorithm for the routing and NF placement of unicast and multicast services for an unconstrained scenario. The algorithm relies mainly on the construction of an auxiliary network transformation that has a one-to-one mapping from the NF placement and routing problem to an equivalent routing problem.

1.3.3 Model-free Dynamic Provisioning of NFV-enabled Services

In Chapter 5, we develop an end-to-end deep RL-based provisioning mechanism to dynamically allocate network resources based on the traffic pattern of the network service and the network substrate. First, we propose a pre-processing method that provides several end-to-end routing and NF placement configurations. Different combinations of such configurations provide several embedded network service topologies. Second, we offer a deep RL algorithm that mainly relies on an actor-critic architecture with a DDPG learning algorithm. Due to considering multiple resource types with strongly adverse effects, and since some events in the state space are sparse, the use of the vanilla DDPG algorithm is observed to be inefficient. It does not always yield the desired behavior. Therefore, we propose a model-assisted exploration procedure that utilizes experiences learned from a perturbed optimal step-wise solution.

1.4 Scholarly Publications

- O. Alhussein, and W. Zhuang, "Dynamic provisioning of NFV-enabled services using deep reinforcement learning," *under preperation*.
- O. Alhussein, and W. Zhuang, "Robust online composition, routing and NF placement for NFV-enabled services," IEEE J. Sel. Areas Commun., 2020, in press. [Online]. Available: arXiv:2002.03464 [cs.NI]
- O. Alhussein, P. T. Do, Q. Ye, J. Li, W. Shi, W. Zhuang, X. Shen, X. Li, and J. Rao, "A virtual network customization framework for multicast services in NFV-enabled core networks," IEEE J. Sel. Areas Commun., 2020, in press. [Online]. Available: arXiv:2002.03301 [cs.NI]
- O. Alhussein, P. T. Do, J. Li, Q. Ye, W. Shi, W. Zhuang, X. Shen, X. Li, and J. Rao, "Joint VNF placement and multicast traffic routing in 5G core networks," in Proc. IEEE Globecom, pp. 1-6, 2018.

1.5 Thesis Outline

The rest of this thesis is organized as follows. Chapter 2 presents a background and literature survey. First, in Section 2.1, we provide a brief overview on SDN, NFV and the history that preceded it. Second, in Section 2.3, we provide an overview of related works for the three investigated research problems. Next, Chapters 3, 4, and 5 presents Problems I, II, and III, respectively.

In Chapter 3, Section 3.1 presents the system model under consideration for Problem I, which includes the representation of the network substrate, NFs, and multicast service requests. Section 3.2 addresses the joint NF placement and routing problems for multicast services with multipath routing for both single-service and multi-service cases. Section 3.3 presents MILP formulations for a single-service multipath scenario, and a generalized multi-service multi-path scenario, respectively. Section 3.4 proposes simple modular heuristic algorithms to address the complexity of the resultant MILP formulations. Simulation results are presented in Section 3.5.

In Chapter 4, Section 4.1 describes the system model under consideration for Problem II. Section 4.2 presents the problem description. Section 4.3 presents the problem formulation, which includes the design of a profit function, and the primal-dual based problem formulation for the offline routing and NF placement framework. In Section 4.4, we develop the primal-dual based admission mechanism, followed by an analysis of the competitive performance of the proposed admission mechanism. Section 4.5 presents the routing and NF placement algorithm for the proposed admission mechanism. Finally, Section 4.6 presents some discussions on the proposed framework, followed by simulation results, to investigate and corroborate the competitive performance of the proposed work.

In Chapter 5, Section 5.1 presents the system model under consideration for Problem III. Section 5.2 presents the problem statement and formulation. Section 5.3 presents the deep RL framework for the proposed problem, which includes the pre-processing stage and the model-assisted deep RL algorithm. In Section 5.4, we conduct numerical performance evaluation of the proposed deep RL scheme. Finally, concluding remarks are drawn in Chapter 6.

Chapter 2

Background and Literature Survey

In this chapter, we first provide a brief introduction to SDN and NFV. Then, we present a more involved literature survey on the research works relevant to the research questions investigated in this thesis.

2.1 Introduction to SDN and NFV

2.1.1 History Predating SDN and NFV

During the 1980s, with the rapid growth of networking, researchers stumbled upon numerous challenges and problems such as the difficulty of introducing new technologies and services, and that the separate protocol layers performed redundant operations which resulted in inadvertently poor performance. As a result, networking researchers turned an eye into a so-called clean-slate paradigm [50], which was catalyzed by the active networking initiative from the 1990s [51]. Active networking adopted a notion of providing network services via a programming interface. Here a programming interface would facilitate the construction of custom functionalities that is applied to a subset of packets passing through the node [52]. Active networking adopted two programming models, namely the capsule model and the programmable switch model. The former, which was more prevalent, envisioned that packets would carry both information and instructions (procedures) that specifies its handling, and the programmable switch would have to carry over the execution. The latter approach focused on the proposal of programmable switches. The programmable switches would perform custom functions that depend on some packet header information. Active networking is very similar to SDN in the sense that it is the first to propose clean-slate programmable networking. Nonetheless, it is worth mentioning that active networking focused on providing programmability in the data plane rather than the control plane. Although active networking was relatively involved in the research community, yet the implementation phase did not see the light mainly because of performance and security concerns [53].

With the inevitable increasing complexity of the Internet in the early 2000s, researchers and service providers started looking for more reliable traffic engineering methods that would deliver more services with more excellent reliability and performance. In 2003-2004, the Internet Engineering Task Force proposed the ForCES protocol [54]. The ForCES architecture defined an interface control channel to control various forwarding elements, such as forwarding, shaping, and classification. However, the ForCES architecture was not widely adopted. Another common approach was the routing control protocol (RCP) in 2005 [55]. Unlike active networking and the ForCES architecture, the RCP was easily deployable. Yet, it was limited to some routing protocols, and its versatility concerning the number of applications that can be supported is limited.

Since the beginning of the 2000s, researchers started advocating the notion of centralized and separate control and management plane. One of the influential works is the clean-slate four-dimensional architecture proposed in [56]. Here the authors suggested a refactoring of the network architecture into four planes, namely to decision, dissemination, discovery, and data. Many works and implementations followed the concepts that were articulated by the aforementioned four-dimensional architecture, e.g., Ethane [57], OpenFlow [58], and NOX network operating system [59].

2.1.2 Network (Function) Virtualization

Network virtualization allows diverse network topologies to cohabit on a shared physical network substrate. In terms of sharing the network substrate, various technologies closely relate to network virtualization, albeit with smaller capacities as compared to today's notion of network virtualization. Examples include virtual local area networking, virtual private networking, overlay networks, and active networking.

Recently NFV became one of the important key cornerstones towards the realization of next-generation networks. In October 2012, several operators composed a white paper [60] on the NFV initiative. The purpose of the first white paper was to introduce and promote the concept of NFV and illustrate its benefits and challenges. As mentioned, NFV refers to the decoupling of NFs from the proprietary physical devices and allowing them to run on commodity servers on-demand by exploiting existing virtualization technologies. In NFV, proprietary NFs are decomposed into virtual NFs. Several virtual NFs comprises a network service that can be virtually overlayed onto the physical network substrate. In the literature, NF chains are also called 'service chains,' and 'NF forwarding graphs.'

The NFV-enabled architecture reduces the capital and operational expenditures and power consumption by reducing the cost of equipment and relying on NFV nodes, allowing for a multi-version and multi-tenancy environment in which different applications and users can share a single platform, and scaling network services as demanded. In terms of service provisioning, NFV introduces the following new features, such as decoupling software from hardware, flexible network deployment, and dynamic scaling through instantiable NFs. Moreover, NFV adds new value creation and use cases by reducing the the barrier for the introduction of network services (especially for short-lived ones) and allowing for a continuous network adjustment and improvement.

2.2 Software Defined Networking

SDN is an architecture that decouples the control plane from the data plane. In doing so, the intelligence and control mechanism is centralized, where the control plane would have the upper hand on how to handle and manage traffic. The underlying data plane elements, such as routers, forward traffic according to the decisions and methodology instructed by the control plane. This approach allows the data plane to be abstracted for applications and network services, whereas the control plane would become completely programmable.

SDN can enable rapid innovation. Control mechanisms can be developed and changed without the need to change the hardware, thereby enabling separable and parallel innovation on both the data and control planes. Moreover, it allows a network-wide view from the perspective of the controller, which allows for better performance and development. Therefore, it introduces more flexibility concerning new services and applications.

The term SDN was first coined in 2009 by Kate Greene in [61]. He referred to this term to describe the OpenFlow platform [58]. Briefly, the OpenFlow platform is an application programming interface that facilitates for the control plane to exercise direct control over the data plane elements by installing OpenFlow tables, which includes a set of fine-grained or coarse-grained instructions. The concept of SDN has had inarguably outgrew the OpenFlow platform since then.

2.3 Literature Survey

The era of 5G and beyond is set to target new diverse services and business models beyond what is achievable by the current technology [62–64]. The main 5G service types are enhanced mobile broadband, ultra machine-type communication, and massive machine-type communication. The first type spans several human-centric use cases, which are characterized by low to high throughput requirements (e.g., virtual reality gaming) and dense deployment in some concentrated areas. The second and third types are machine-centric, where the former type contains use cases of stringent quality of service requirements, and the third type is characterized by a vast number of connected devices with a rather more relaxed End-to-End (E2E) delay requirement. Moreover, we are experiencing increasing demands for services of multicast nature such as video streaming, multi-player augmented reality games, and file distribution. There is a consensus that the ubiquitous service types in 5G and beyond cannot be realized by a one-size-fits-all architecture. A pluralistic service-customized (e.g., NFV-enabled) network architecture can be better equipped to enable the next-generation of networks.

In what follows, we discuss some literature works pertinent to the three research problems in this thesis.

2.3.1 Routing and Placement of NFV-enabled Multicast Services

SDN provides a global network view and centralized control over the substrate network topology with the associated network states. Some existing works focus on constructing efficient centralized routing topologies for multicast services without NFs [12–14,65], which mainly rely on constructing a Steiner tree (or one of its variants) to reduce the transmission resource consumption, network state consumption, E2E delay, or to improve the reliability. For instance, Zhao *et al.* propose an SDN-based video conferencing solution by constructing a minimum-delay Steiner tree with iterative enhancements [12]. To deal with the increased network state consumption due to multicasting, a Steiner tree that jointly minimizes the number of links and branch nodes (with network states) is established, assuming unicasting across links with no branches [65]. The state-of-the-art SDN-enabled multicast routing approaches cannot be adopted or modified directly to incorporate the NF placement.

The virtual NF placement and traffic routing problem has been extensively studied for the unicast service case [1, 26, 28, 31, 66–70]. In its most basic form, the orchestration of an NFV-enabled service poses two correlated and conflicting subproblems, i.e., how to place (or select) the NF instances, and how to route the traffic to traverse the NF instances. Placing a minimal feasible number of NF instances can lead to a large link provisioning cost; conversely, deploying more NF instances can reduce the link provisioning cost at the expense of an increased function provisioning cost. This tradeoff becomes more conspicuous when considering a multicast service in which traffic is routed to more than one destination. For the multicast service case, a simple approach is to apply the unicast-based NF placement and routing approaches to each source-destination pair independently, which leads to a waste of network resources with a large service provisioning cost.

There are relatively few works in the joint multicast routing and NF placement for multicast services [2, 15–19], where one needs to jointly build a multicast topology and place the NFs. Zhang *et al.* consider the joint routing and placement for NFV-enabled multicast requests, under the assumption that all NFs should be served by only one NFV node [15], which is extended for a multiple NFV node case under the assumption of an uncapacitated substrate network [2]. It is assumed that multicast replication points occur only after deployment of NFs in the constructed multicast topology, which reduces the degree of flexibility of the orchestration framework. Zeng *et al.* jointly consider placement, multicast routing, and spectrum assignment for a fibre optical network, where the objective is to minimize the function, link, and frequency spectrum provisioning costs [16]. The heuristic solution clusters each group of destinations that share one specific NF. Then, the solution is divided into two steps to allocate the NF in an NFV node for each cluster, and to find the MST that spans the source, the allocated NF, and the destinations. Similarly, it is assumed that, for each multicast service, the traffic flowing to each destination is processed by one NF.

Xu *et al.* consider that multiple NFV nodes can host all types of NFs [17]. For each source-destination pair, the multicast stream needs to pass through only one NFV node where all NFs are placed before arriving at each destination. Since the destinations are distributed in a large area, the algorithm allows activating multiple NFV nodes, where each NFV node is responsible for a subset of destinations. However, there is a possibility that some NFV nodes can only host specific types of NFs due to hardware-based or subscription-based restrictions [16, 67]. A recent work tackles the so-called service forest problem for the traffic routing and NF placement of multiple service requests [18]. For a generalized scenario where each source-destination pair requires multiple service requests, the last NF is placed on the physical substrate first, which divides the multicast topology into two parts. The first part is from the source to the last NF, while the second part from the last NF to all destinations. The first part is solved by the so-called k-stroll algorithm. In the second part, a minimum Steiner tree approximation connects the last NF to all destinations. It is assumed that multicast replication points at the topology always occur after the last NF, and an exhaustive search finds the best place to host the last NF among all candidate NFV nodes.

To address some research gaps, we aim at developing a *flexible* multicast routing and NF placement framework for both single-service and multi-service scenarios. We consider that multicast replication points can occur before and after the deployment of NF instances, thereby providing flexibility for topology customization, which is particularly crucial for geographically distributed NF chains (such as in NFV-enabled core networks). In addition, we incorporate multipath routing between the embedded NFs to improve the utilization of the network substrate's resources.

2.3.2 Competitive Online Routing (Predating NFV)

Prior to enabling NFV, in traditional circuit switching, call requests (respectively, service requests) resembled a routing request from the source to the destinations with a data rate requirement that need to be routed in a capacitated network substrate [38]. With the emergence of NFV, service requests subsumed call requests with the additional requirement that virtual NFs need to be instantiated on NFV nodes along the route.

Related works can be classified according to the parameter that measures the performance of the intended design, typically the throughput or the congestion. In throughput-maximization frameworks, we measure the transmission resources of all admitted service requests. In congestion-minimization, we measure the maximum link congestion, i.e., the maximum ratio of the allocated transmission resources on a link to its total transmission capacity. This work can be considered as a generalized case of the throughput-maximization framework.

In 1993, Aspnes et al. developed a competitive strategy for congestion-minimization that achieves a competitive ratio of $\mathcal{O}(\log n)$ for service requests of infinite holding time, where n is the total number of nodes in the substrate network [71]. Assuming service requests have finite holding time (which is revealed only upon the arrival for each service request), the authors extended their result to achieve an $\mathcal{O}(\log nT)$ competitive ratio, where T is the maximum holding time of all service requests. For the throughput-maximization model, Awerbuch et al. achieved a competitive ratio of $\mathcal{O}(\log n)$ [38].

2.3.3 Competitive Online Routing and NF Placement

One main challenge in devising a competitive online routing and NF placement algorithm is pertaining to the inclusion of processing resources along with transmission resources. To this end, Lukovszki et al. consider that all unicast service requests contain the same set of requested NFs and identical transmission rates, where service requests vary with regard to the source and destination [33]. They propose an $\mathcal{O}(\log K)$-competitive admission mechanism, where K is the number of NFs of a service request. Interestingly, the competitive ratio is logarithmic with the number of NFs, which is small in practice. In [36, 72], the authors consider both unicast and multicast requests without NFs, where routing a request utilizes transmission resources from the physical links and routing rules from the forwarding table of the traversed switches. Notably, the achieved competitive ratio can be shown to be $\mathcal{O}(\max\{\log 2L, \log 2E\})$, where L and E are the maximum number of physical links and switches for a service request, respectively. The competitive ratio for the two resources is balanced since $L = E - 1$.

Xu et al. consider a multicast request with one NF, and develop an $\mathcal{O}(\log L)$-competitive algorithm [17]. Ma et al. consider the dynamic admission of delay-aware requests for services in a distributed cloud with the objective of maximizing a designed profit function [32]. They first provide a heuristic algorithm for the delay-aware scenario, followed by an online algorithm with an $\mathcal{O}(\log L)$-competitive ratio for the special case where the end-to-end delay is negligible.

In Chapter 4, we propose a primal-dual based online algorithm that accommodates both unicast and multicast service requests with multiple NF instances that can be deployed at different NFV nodes. We consider heterogeneous services with best-effort and mandatory NFs. In doing so, we propose an all-or-nothing/all-or-something admission mechanism that re-composes the logical topology of the service request before admitting it (depending on whether or not best-effort NF instances can be included). Moreover, based on a primal-dual framework, we offer a new alternative, generalized description, and analysis to the aforementioned NFV-enabled literature.

2.3.4 Routing and NF Placement with Service Composition

Although several NF chain topologies express the same network policy semantically, the embedding process for each logical topology can yield different provisioning costs and long-run characteristics in a non-trivial manner. In Problem III, we aim to leverage service composition (with the routing and NF placement) to alter the NF chain topology per the time-varying traffic demands. Generally, there are two approaches for the joint composition, NF placement, and traffic routing. The first approach is to decouple the service composition (or topology design) from the NF placement and routing process [41, 66]. However, such an approach can be far from the globally optimal solution.

Mehraghdam et al. propose a language model that formalizes an NFV-enabled multicast service [66]. Such a language model is general as it allows for service composition. However, the addressed topological design aspect reduces to re-ordering the NFs from lowest to highest traffic inflation factors to minimize the required data rate of the NF chain. Mehraghdam et al. extend their work to account for the service composition, where a heuristic approach yields an extensive Pareto set of the possible combinations of different services for different optimization objectives [41].

Another approach is to design a heuristic algorithm such as a breadth-first search, where some composition operator is included to minimize the function and link provisioning costs [30, 42]. More specifically, Ye et al. jointly consider the topology design and embedding, where they formulate an integer linear program and develop a heuristic algorithm under the assumption that NFs in a service chain can be combined [30]. The heuristic algorithm resembles an iterative two-step search, which iteratively combines two

NFs, and then performs embedding onto the physical substrate. The algorithm eventually selects an NF chain topology that minimizes the placement and routing cost. Beck and Botero present a coordinated breadth-first algorithm that minimizes the function and link provisioning costs, where the algorithm attempts to find a path from the source to the destination while being able to replicate NFs if needed [42]. Li et al. develop a two-stage heuristic for the composition and embedding, where the location and functionality of the substrate nodes are taken into consideration [73].

More recently, Coa et al. consider a genetic framework that designs an overall NF forwarding graph for multiple traffic flows to minimize the function and link provisioning costs and improve the acceptance ratio of services [74]. Jalalitabar et al. design a heuristic algorithm for the NF placement and routing that takes the dependence relationship between the NFs into account [75]. The works above present model- and optimization-based frameworks in which long-run provisioning costs can be challenging to incorporate. Moreover, they consider a fixed data rate for a service request.

2.3.5 Deep Reinforcement Learning based Dynamic Provisioning

Due to the recent advances in deep learning and deep RL, the literature is enjoying a renewed interest in the application of (deep) RL on networking problems. More specifically, very recent works have been proposed for various traffic engineering problems that span routing, load balancing, NF placement/migration, NF scheduling, and Caching [76–81].

One of the early works aims to maximize a network utility for a general communication network with K end-to-end communication sessions [76], in which a traffic engineering-aware exploration while utilizing the DDPG algorithm with a prioritized experience replay technique is proposed. Pei et al. study the NF placement (or NF migration) problem in an NFV-enabled network under a time-varying traffic pattern [82]. Due to the large solution space, the authors divide the network substrate into smaller regions and use a double deep Q-network algorithm to select a small number of potential NFV nodes. Troia et al. consider an NFV-enabled metro-core optical network represented as a multi-layer network to maximize the number of successfully routed NF chains, while minimizing the reconfiguration penalty, blocking probability and power consumption. Gu et al. consider the VNF orchestration and flow scheduling for NFV-enabled network services [77], whereby a model-assisted DDPG is used to aid in the convergence speed.

In Chapter 5, we consider the joint dynamic composition, traffic routing, and NF placement for unicast network services under time-varying traffic patterns. To do so, we incorporate multipath splitting and consider that NFs can be embedded sequentially and in parallel to provide a malleable network service topology.

Chapter 3

Joint Routing and NF Placement for Multicast Services

3.1 System Model

3.1.1 Network Functions and Multicast NF Chains

With NFV, traditional applications and functionalities (which used to be implemented in the control plane or at the end-users) are now deployable in the data plane in NFV nodes.

Let the set of all multicast service requests be denoted by $\mathcal{R}$. A multicast service, $S^r \in \mathcal{R}$, is described by a multicast NF chain, represented by a weighted acyclic directed graph,

$$S^r = (s^r, \mathcal{D}^r, \mathcal{V}^r, d^r), \quad S^r \in \mathcal{R} \tag{3.1}$$

where s^r and $\mathcal{D}^r$ represent the source node and the set of destinations, $\mathcal{V}^r = \{f_1^r, f_2^r, \dots f_{|\mathcal{V}|}^r\}$ represents the set of functions that have to be traversed in an ascending order for every source-destination pair, and d^r is the data rate requirement in packet/s. Each NF f_i^r requires a processing rate $C(f_i^r)$, $i \in \{1, \dots, |\mathcal{V}^r|\}$, in packet/s. An NF instance belongs to one service request, and it cannot be shared among multiple NF chains [2, 15, 18, 66].

3.1.2 Network Substrate

Consider a network substrate, $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ and $\mathcal{L}$ are the set of nodes and links, respectively. The nodes can be either switches or NFV nodes (represented by set $\mathcal{M}$). Switches are capable of forwarding and replicating traffic, and NFV nodes are capable of hosting and operating NFs. We assume that each NFV node has a forwarding capability, and has available processing rate $C(n)$, $n \in \mathcal{M}$, in packet per second (packet/s) [83–85]. Moreover, an NFV node is capable of provisioning a number of NFs simultaneously as long as the available processing rate satisfies the NF processing requirements. Each physical link has limited transmission rate $B(l)$, $l \in \mathcal{L}$, in packet/s. Denote the set of NFV nodes that can host an NF f_i by $\mathcal{F}_i$, where $\mathcal{F}_i \subseteq \mathcal{N}$.

3.2 Problem Definition

In this chapter, we investigate the orchestration of multiple multicast services over the network substrate in two sequential problems. In the first problem, joint multipath-enabled multicast routing and NF placement is studied for a single service. Given the network substrate and the service description of a multicast NF chain, we intend to design an embedded multicast topology for the NF chain. For each source-destination pair in the embedded topology, all NF types should be traversed in a specified order.

Therefore, the first problem consists of two joint subproblems: (i) NF placement on the network substrate, and (ii) multicast traffic routing design from the source to the destinations, passing through a sequence of the required NFs. Note that multipath routing is enabled for the paths between embedded NFs to increase the flexibility of topology customization and improve the physical resource utilization especially for geographically-dispersed large-scale core networks. Our objective is to minimize the function and link provisioning costs in determining an optimal embedded multicast topology. However, the minimization of both costs are conflicting. Instantiating a large number of NF instances at more network locations achieves a balanced traffic load at the expense of an increased function provisioning cost, whereas fewer NF instantiations reduce the function provisioning cost at the expense of less load balancing and inefficient network resource exploitation. Therefore, it is required to balance the tradeoff to minimize the overall provisioning cost for the NF chain embedding. The first problem is defined as follows:

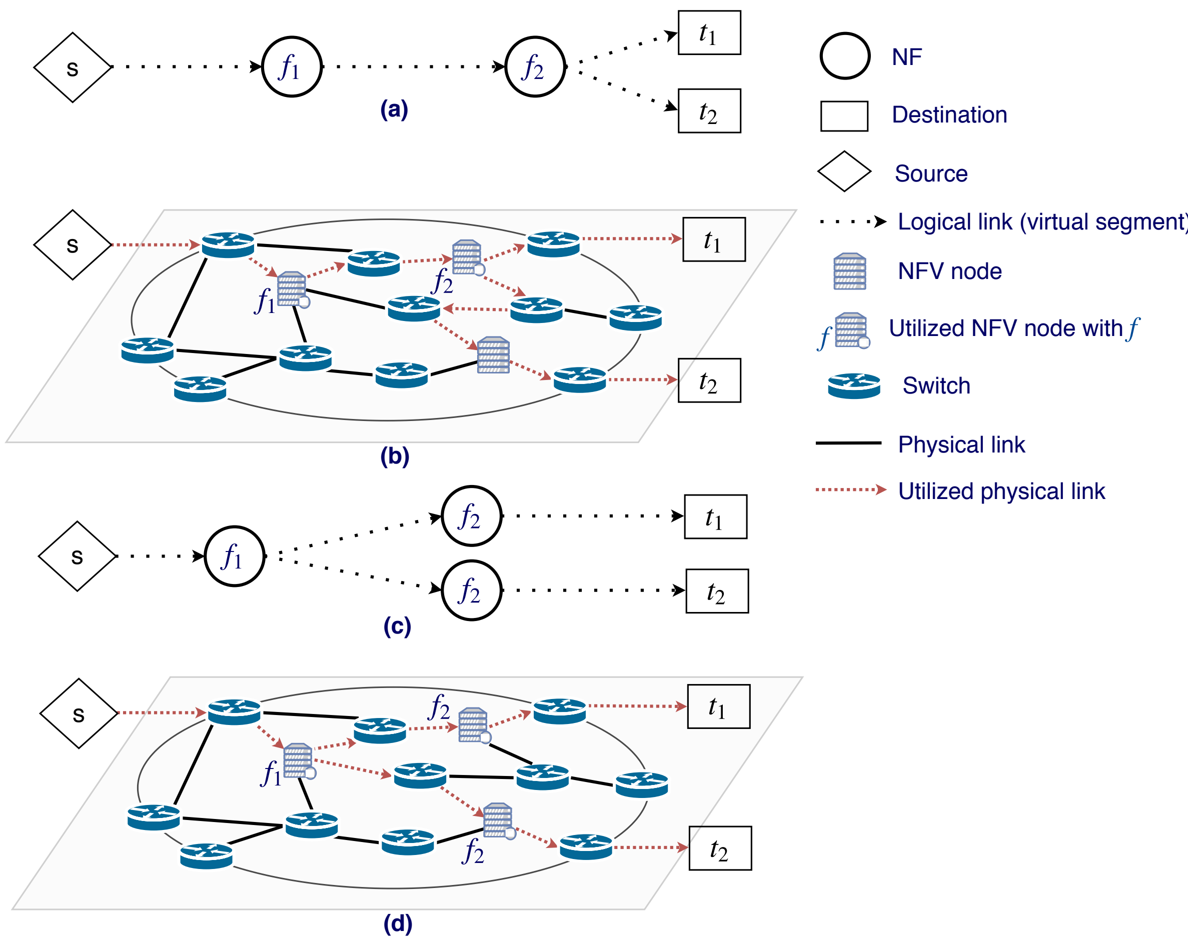

Figure 3.1: Comparison of flexible and non-flexible embedding for a multicast request. (a) Topology of NF chain request; (b) Embedding result of NF chain on network substrate with 11 links due to the non-flexible scheme; (c) Modified topology of NF chain request due to the flexible scheme; (d) Embedding result of modified NF chain on network substrate with 10 links due to the flexible scheme.

Problem 1. *To determine an optimal multipath-enabled multicast topology for an NF chain to minimize the function and link provisioning costs, with all the required NFs traversed in order and with the required processing and transmission resource constraints satisfied.*

In the second problem, we study a joint multicast routing and NF placement problem for a multi-service scenario. NFV allows multiple NF chains to run over a common network substrate. However, as the network resources are limited, multiple NF chains may not be accepted on the network substrate simultaneously. We need to decide which service requests should be embedded such that the aggregate throughput is maximized, while the function and link provisioning costs are minimized. We consider the static NF chain embedding for the multi-service scenario, where all service requests are available a priori[1].

Problem 2. *To find an optimal combination of multicast NF chains that maximizes the aggregate throughput of the network substrate, while minimizing the respective function and link provisioning costs.*

We consider that NFs and virtual links have one-to-many and many-to-one mapping with physical NFV nodes and links, respectively, whereby each NF instance can serve a subset of destinations, and the deployment of NF instances can occur after packet replication in the multicast topology. Such practical consideration achieves higher flexibility and efficiency in the routing and placement processes for largely distributed networks, since destinations may be geographically far away. Restricting all destinations to share the same set of NF instances can be inefficient for transmission resource limited scenarios. When the destinations are geographically dispersed, an efficient solution is to duplicate and deploy the function instances close to each of the destinations for a reduced link provisioning cost due to flexible routing. We give an illustration in Fig. 3.1, where the logical topology of an NF chain request with two NFs and two destinations (i.e., $t_1, t_2 \in \mathcal{D}$) are embedded onto the network substrate. Assume that each physical link can be used only once. With a non-flexible scheme, NFs cannot be replicated and multicast replication points can exclusively occur after the deployment of the last NF. Hence, the embedding result of the multicast request (Fig. 3.1-(a)) on the physical substrate requires 11 links as shown in Fig. 3.1-(b). With a flexible scheme, the NF chain topology is modified, as shown in Fig. 3.1-(c), where the respective embedding result on the physical substrate requires 10 links as shown in Fig. 3.1-(d).

[1]In this chapter, we consider that all service requests are available a priori. An online treatment of the service requests is addressed in Chapter 4.

3.3 Problem Formulation

We first present the problem formulation for a single-service multipath scenario, followed by a generalized multi-service multipath scenario.

3.3.1 Single-service Multipath Scenario

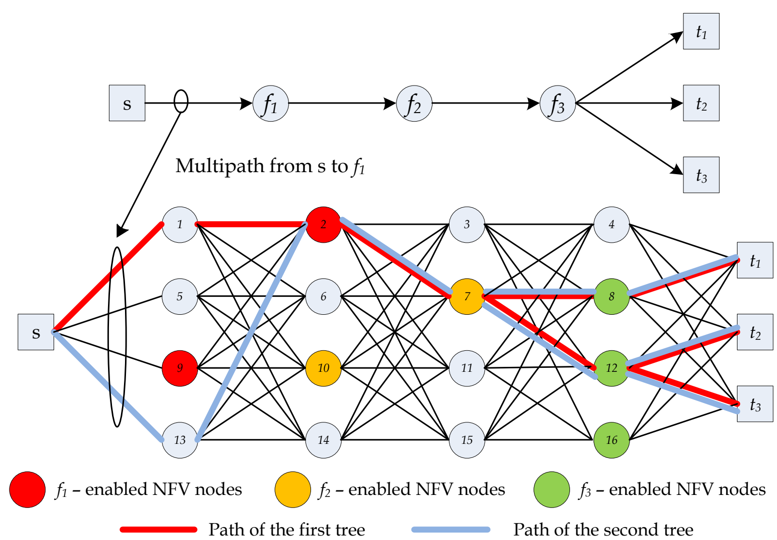

Figure 3.2: Two Steiner trees that share the same source, traversed functions, and destinations.

To establish a joint multipath-enabled multicast routing and NF placement framework for a single-service, let J^r denote the maximum number of multicast trees to deliver multicast service S^r ($\in \mathcal{R}$) from the source to the destinations[2]. Each tree emanates from the source and passes by the same set of traversed NFV nodes and destinations. Fig. 3.2 illustrates one multicast request with three destinations and three functions. We use two

[2]Note that the superscript r which is used in the single-service formulation (subsection 3.3.1) can be dropped since $|\mathcal{R}| = 1$. However, it is necessary as we develop the MILP formulation for the multi-service scenario in subsection 3.3.2.

trees (i.e., $J^r = 2$) to support up to two multipath routing paths. If f_1 is embedded on node 2 for both trees, we have multipath routes from source node s to node 2, and the two trees converge afterwards. When $J^r = 1$, the problem reduces to the single-path routing case. The maximum number of multicast trees (resembling the maximum possible number of multipath routes) J^r is an input to the problem formulation. The formulation allows the multicast trees to overlap with each other, and can be deactivated if needed. In what follows, we describe the details of the MILP formulation for the single-service scenario.

Let Ω_m^n denote the set of integers from m to n $(> m)$, i.e., $\Omega_m^n \triangleq \{m, m+1, \dots, n\}$ with $m, n \in \mathbf{Z}_+$. Define binary variable $x_{li}^{jr} \in \{0, 1\}$, where $x_{li}^{jr} = 1$ indicates that link l $(\in \mathcal{L})$ is used for forwarding traffic for service S^r in multicast tree j from f_i^r to f_{i+1}^r where $i \in \Omega_1^{|\mathcal{V}^r|-1}$; Binary variable $x_{l0}^{jr} = 1$ indicates that link l carries traffic from s^r to f_1^r, and $x_{l|\mathcal{V}|}^{jr} = 1$ indicates that link l carries traffic from $f_{|\mathcal{V}|}^r$ to any destination node $t\,(\in \mathcal{D}^r)$; Define $y_{lit}^{jr} \in \{0, 1\}$, where $y_{lit}^{jr} = 1$ indicates that link l is used to direct traffic for service S^r in multicast tree j from f_i^r to f_{i+1}^r for destination $t\,(\in \mathcal{D}^r)$; Binary variable $y_{l0t}^{jr} = 1$ indicates that link l is used to direct traffic for service S^r in tree j from s^r to f_1^r for destination t, and $y_{l|\mathcal{V}|t}^{jr} = 1$ indicates that link l directs traffic for service S^r in tree j from $f_{|\mathcal{V}|}^r$ to destination t.

With the definitions of $\boldsymbol{x} = \{x_{li}^{jr}\}$ and $\boldsymbol{y} = \{y_{lit}^{jr}\}$, we have

$$y_{lit}^{jr} \leq x_{li}^{jr}, \quad l \in \mathcal{L}, i \in \Omega_0^{|\mathcal{V}^r|}, j \in \Omega_1^{J^r}, t \in \mathcal{D}^r,\ S^r \in \mathcal{R}. \tag{3.2}$$

Furthermore, define binary variables $z_{ni}^r \in \{0, 1\}$, where $z_{ni}^r = 1$ indicates that an instance of f_i^r is deployed on NFV node n for service S^r where $i \in \Omega_1^{|\mathcal{V}|}$, and $u_{nit}^r \in \{0, 1\}$, where $u_{nit}^r = 1$ indicates that an instance of f_i^r is deployed on n for service S^r for destination t. Similarly, we have a relationship constraint between $\boldsymbol{z} = \{z_{ni}^r\}$ and $\boldsymbol{u} = \{u_{nit}^r\}$ as

$$u_{nit}^r \leq z_{ni}^r, \quad n \in \mathcal{N}, i \in \Omega_1^{|\mathcal{V}^r|},\ t \in \mathcal{D}^r,\ S^r \in \mathcal{R}. \tag{3.3}$$

For each service S^r $(\in \mathcal{R})$, we build J^r multicast trees to exploit the multipath property, where each tree can provide a fractional transmission rate d^{jr} of the overall required transmission rate d^r. Therefore, to meet the total required transmission rate d^r, we impose the constraint

$$\sum_{j=1}^{J^r} d^{jr} \geq d^r,\ S^r \in \mathcal{R}. \tag{3.4}$$

Next, we incorporate the routing and placement constraint in (3.5) to ensure that traffic flows pass from the source to multiple destinations through the NF chain. Let f_0^r and $f_{|\mathcal{V}|+1}^r$ denote dummy functions (without processing requirements) that are placed on the source node s^r and each destination node t ($\in \mathcal{D}^r$), respectively. In our model, some of the multicast trees can be deactivated if needed. Consequently, we have

$$\sum\nolimits_{(n,m)\in\mathcal{L}} y^{jr}_{(n,m)it} - \sum\nolimits_{(m,n)\in\mathcal{L}} y^{jr}_{(m,n)it} = \begin{cases} u^r_{nit} - u^r_{n(i+1)t}, & \text{tree } j \text{ is activated} \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

for $n \in \mathcal{N}, i \in \Omega_0^{|\mathcal{V}^r|}, t \in \mathcal{D}^r, S^r \in \mathcal{R}$, where

$$u^r_{s0t} = 1,\ u^r_{n0t} = 0,\ \forall t \in \mathcal{D}^r,\ n \in \mathcal{N}\backslash\{s^r\}$$

$$u_{t(|\mathcal{V}^r|+1)t} = 1,\ u_{n(|\mathcal{V}^r|+1)t} = 0,\ \forall t \in \mathcal{D}^r,\ n \in \mathcal{N}\backslash\mathcal{D}^r$$

$$u^r_{sit} = 0,\ u^r_{tit} = 0,\ \forall i \in \Omega_1^{|\mathcal{V}^r|},\ t \in \mathcal{D}^r.$$

Define binary variable $\pi^{jr} \in \{0,1\}$, where $\pi^{jr} = 1$ indicates that tree j of service S^r is activated. Consequently, all variables related to deactivated trees (i.e., x^{jr}_{li}, y^{jr}_{lit}, and d^{jr}) should be forced to zero. Therefore, we impose the constraint

$$x^{jr}_{li} \le \pi^{jr},\ y^{jr}_{lit} \le \pi^{jr}, d^{jr} \le \pi^{jr} d^r,\quad l \in \mathcal{L}, i \in \Omega_0^{|\mathcal{V}^r|}, j \in \Omega_1^{J^r}, t \in \mathcal{D}^r,\ S^r \in \mathcal{R}. \tag{3.6}$$

To guarantee that the routing and placement constraint is considered only when tree j of service S^r ($\in \mathcal{R}$) is activated, we modify (3.5) to

$$\begin{aligned} \sum_{(n,m)\in\mathcal{L}} y^{jr}_{(n,m)it} - \sum_{(m,n)\in\mathcal{L}} y^{jr}_{(m,n)it} =& \pi^{jr}\left(u^r_{n(i+1)t} - u^r_{nit}\right), \\ & n \in \mathcal{N}, i \in \Omega_0^{|\mathcal{V}^r|}, t \in \mathcal{D}^r, S^r \in \mathcal{R}. \end{aligned} \tag{3.7}$$

Since we have $y^{jr}_{lit} \le x^{jr}_{li}$ in (3.2), the constraint $y^{jr}_{lit} \le \pi^{jr}$ in (3.6) can be removed. Thus, we re-write (3.6) as

$$x^{jr}_{li} \le \pi^{jr},\ d^{jr} \le \pi^{jr} d^r, l \in \mathcal{L}, i \in \Omega_0^{|\mathcal{V}^r|}, j \in \Omega_1^{J^r}. \tag{3.8}$$

Constraint (3.8) means that we consider x^{jr}_{li} and d^{jr} when tree j is activated; otherwise, we simply set these variables to zero. Define indicator function $k_{ni} \in \{0,1\}$, such that $k_{ni} = 1$

if NFV node n can admit function f_i, i.e., $n \in \mathcal{F}_i$. We require that exactly one instance of function f_i^r is traversed for every source-destination pair, which can be expressed as

$$\textstyle\sum_{n\in\mathcal{M}} u_{nit}^r = 1, \; i \in \Omega_1^{|\mathcal{V}^r|}, t \in \mathcal{D}^r, S^r \in \mathcal{R}. \tag{3.9}$$

Moreover, function f_i^r is hosted at node n only when admittable and when the resources at node n are sufficient. We have

$$\sum_{S^r\in\mathcal{R}} \sum_{i=1}^{|\mathcal{V}^r|} z_{ni}^r C(f_i^r) \leq C(n), \; n \in \mathcal{M}, \; i \in \Omega_1^{|\mathcal{V}^r|} \tag{3.10a}$$

$$z_{ni}^r k_{ni} = 1, \; n \in \mathcal{M}, i \in \Omega_1^{|\mathcal{V}^r|}, S^r \in \mathcal{R} \tag{3.10b}$$

where $k_{ni} = 1$ indicates that node n can admit function f_i; otherwise, $k_{ni} = 0$.

Objectives — Following the relevant research on NFV-enabled service provisioning [16–18,66–68], we aim to minimize the function (processing) and link (transmission) provisioning costs over all J^r multicast trees for service S^r, in addition to balancing the substrate network resources in the long run as

$$\min_{\boldsymbol{d},\boldsymbol{x},\boldsymbol{z}} \alpha \sum_{l\in\mathcal{L}} \sum_{j=1}^{J^r} \sum_{i=0}^{|\mathcal{V}^r|} \left(\frac{d^{jr}}{B(l)} + 1 \right) x_{li}^{jr} + \beta \sum_{i=1}^{|\mathcal{V}^r|} \sum_{n\in\mathcal{M}} \frac{C(f_i^r)}{C(n)} z_{ni}^r. \tag{3.11}$$

In (3.11), we minimize the weighted sum of the cost of forwarding the traffic over the utilized physical links of the substrate network for all activated trees, and the cost of provisioning the NF instances in the NFV nodes. Parameters α and β are the weighting coefficients to reflect the importance level of minimizing the cost of traffic forwarding and minimizing the cost of NF provisioning respectively, where[3] $\alpha + \beta = 1$ and $\alpha, \beta > 0$. The terms $d^{jr} x_{li}^{jr}/B(l)$ and $C(f_i^r) z_{ni}^r / C(n)$ ensure load balancing over the physical links and NFV nodes [88]. Moreover, the term '+1' in (3.11) minimizes the number of links in building the multicast topology. Denote the product term $d^{jr} x_{li}^{jr}$ in the objective function (3.11) by γ_{li}^{jr} as

$$\gamma_{li}^{jr} = x_{li}^{jr} d^{jr}, \; l \in \mathcal{L}, i \in \Omega_0^{|\mathcal{V}^r|}, j \in \Omega_1^{J^r}, S^r \in \mathcal{R}. \tag{3.12}$$

[3]Beyond trial and error, how to exactly set the weights (α and β) is an involved question that is subject to many factors, such as the available network resources and future service requirements. For interested readers, the literature on the linear scalarization of multi-objective optimization problems can help, cf. [86,87].

The term, γ_{li}^{jr}, can be interpreted as the transmission rate over link l to deliver traffic from f_i^r to f_{i+1}^r in tree j. The aggregate rate from all services over link l is upper bounded by the available link transmission rate $B(l)$, i.e.,

$$\sum_{S^r \in \mathcal{R}} \sum_{j=1}^{J^r} \sum_{i=0}^{|\mathcal{V}^r|} \gamma_{li}^{jr} \leq B(l),\ l \in \mathcal{L}. \tag{3.13}$$

In summary, the optimization problem for the single service scenario is formulated as

$$\textbf{(P1)}: \min_{\boldsymbol{\mathcal{X}}} \alpha \sum_{l \in \mathcal{L}} \sum_{j=1}^{J^r} \sum_{i=0}^{|\mathcal{V}^r|} \left(\frac{\gamma_{li}^{jr}}{B(l)} + x_{li}^{jr} \right) + \beta \sum_{i=1}^{|\mathcal{V}^r|} \sum_{n \in \mathcal{M}} \frac{C(f_i^r)}{C(n)} z_{ni}^r \tag{3.14a}$$

$$\text{subject to} \quad (3.2)-(3.4), (3.7)-(3.10), (3.12), (3.13) \tag{3.14b}$$

$$\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{\pi} \in \{0, 1\},\ \boldsymbol{d} \succeq 0,\ \boldsymbol{\gamma} \succeq 0. \tag{3.14c}$$

where $\boldsymbol{\mathcal{X}} = \{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{\pi}, \boldsymbol{d}, \boldsymbol{\gamma}\}$. In (3.14), the objective function and all constraints are linear except constraints (3.7), (3.12), (3.10b). In the next step, we transform these non-linear constraints to equivalent linear constraints such that a standard MILP solver can handle them. To do so, for nonlinear constraint (3.7), the bilinear term $\pi^{jr} u_{nit}^r$ can be handled by introducing a new variable $w_{nit}^{jr} = \pi^{jr} u_{nit}^r$. Constraint (3.7) is changed to

$$\sum_{(m,n) \in \mathcal{L}} y_{(m,n)it}^{jr} - \sum_{(n,m) \in \mathcal{L}} y_{(n,m)it}^{jr} = w_{nit}^{jr} - w_{n(i+1)t}^{jr},$$
$$n \in \mathcal{N}, i \in \Omega_0^{|\mathcal{V}^r|}, t \in \mathcal{D}^r, j \in \Omega_1^{J^r}, S^r \in \mathcal{R}. \tag{3.15}$$

The corresponding relations among w_{nit}^{jr}, π^{jr}, and u_{nit}^r are given by

$$w_{nit}^{jr} \leq \pi^j,\ w_{nit}^{jr} \leq u_{nit}^r,\ w_{nit}^{jr} \geq \pi^{jr} + u_{nit}^r - 1,$$
$$n \in \mathcal{N}, i \in \Omega_0^{|\mathcal{V}^r|}, t \in \mathcal{D}^r, j \in \Omega_1^{J^r}, S^r \in \mathcal{R}. \tag{3.16}$$

For nonlinear constraint (3.10b), denote the product term $z_{ni}^r k_{ni}$ by g_{ni}^r. Consequently, (3.10b) can be expressed by

$$g_{ni}^r \leq z_{ni}^r,\ n \in \mathcal{M},\ i \in \Omega_1^{|\mathcal{V}^r|} \tag{3.17a}$$

$$g_{ni}^r \leq k_{ni},\ n \in \mathcal{M},\ i \in \Omega_1^{|\mathcal{V}^r|} \tag{3.17b}$$

$$g_{ni}^r \geq z_{ni}^r + k_{ni} - 1,\ n \in \mathcal{M},\ i \in \Omega_1^{|\mathcal{V}^r|}. \tag{3.17c}$$

For nonlinear constraint (3.12), we utilize the big-M method, and express it equivalently as

$$d^{jr} - M(1 - x_{li}^{jr}) \leq \gamma_{li}^{jr} \leq d^{jr},\ l \in \mathcal{L}, i \in \Omega_0^{|\mathcal{V}^r|},\ j \in \Omega_1^{J^r},\ S^r \in \mathcal{R} \tag{3.18a}$$

$$0 \leq \gamma_{li}^{jr} \leq M x_{li}^{jr},\ l \in \mathcal{L}, i \in \Omega_0^{|\mathcal{V}^r|},\ j \in \Omega_1^{J^r},\ S^r \in \mathcal{R} \tag{3.18b}$$

where M is a large positive number. Since d^{jr} is upper bounded by d^r, γ_{li}^{jr} given by (3.12) is bounded above by d^r. Thus, it suffices to set $M = d^r$.

As a result, the non-linear optimization problem for a single-service in (3.14) can be re-written in an MILP form as

$$\textbf{(P1}'\textbf{)}: \min_{\boldsymbol{\mathcal{X}}, \boldsymbol{w}} \sum_{l \in \mathcal{L}} \sum_{j=1}^{J^r} \sum_{i=0}^{|\mathcal{V}^r|} \alpha \left(\frac{\gamma_{li}^{jr}}{B(l)} + x_{li}^{jr} \right) + \beta \sum_{i=1}^{|\mathcal{V}^r|} \sum_{n \in \mathcal{M}} \frac{C(f_i^r)}{C(n)} z_{ni}^r \tag{3.19a}$$

$$\text{subject to } (3.2) - (3.4), (3.8) - (3.10a), (3.13), (3.14c) - (3.18) \tag{3.19b}$$

3.3.2 Multi-service Multipath Scenario

In this subsection, we consider the scenario of jointly handling multiple multicast service requests. We formulate an MILP that jointly constructs the multicast topology for multiple service requests, where the goal is to find the combination of service requests such that the aggregate throughput is maximized while the overall function and link provisioning costs are minimized. First, we need to maximize the achieved aggregate throughput ($\mathbb{R}$) obtained by hosting network services on the substrate network. The achieved aggregate throughput is given by

$$\mathbb{R} = \sum_{S^r \in \mathcal{R}} R^r \zeta^r, \tag{3.20}$$

where $\zeta^r \in \{0, 1\}$ is a binary decision variable with $\zeta^r = 1$ when service S^r is accepted, and R^r is the corresponding throughput, defined as

$$R^r = a_1 \sum_{i=1}^{|\mathcal{V}^r|} C(f_i^r) + a_2 \Big(|\mathcal{V}^r| + |\mathcal{D}^r| \Big) d^r,\ S^r \in \mathcal{R} \tag{3.21}$$

where the first and second terms denote the required amount of processing and transmission rates, respectively, in packet/s [83–85]. The two parameters, a_1 and a_2, are used to tune the priority of processing and transmission rates respectively, where $a_1+a_2=1$ and a_1, $a_2>0$.

Second, in addition to maximizing the achieved aggregate throughput, an efficient configuration to utilize the resources efficiently is needed. Specifically, there can be multiple solutions to maximize the aggregate throughput defined above. Among such solutions, we find the configuration with the least function and link provisioning cost to save resources for future services. The multi-service scenario is cast as a two-step MILP, the first step aims to find the maximum achievable aggregate throughput, followed by a formulation which finds the routing and NF placement for each admitted NF chain subject to the maximum achievable aggregate throughput.

In the multi-service scenario, some of the service requests can be rejected due to limited resources. Therefore, we first generalize some of the previous constraints as follows. Constraint (3.4) is generalized to

$$\sum_{j=1}^{J^r} d^{jr} \geq \zeta^r d^r,\ S^r \in \mathcal{R} \tag{3.22}$$

where the summation of the fractional transmission rate from all trees for service S^r ($\in \mathcal{R}$) is forced to zero when the service is rejected (i.e. when $\zeta^r = 0$). Moreover, an instance of f_i^r is deployed at only one NFV node if service S^r is accepted, i.e., (3.9) becomes

$$\sum_{n\in\mathcal{M}} u^r_{nit} = \zeta^r,\ i \in \Omega_1^{|\mathcal{V}^r|},\ t \in \mathcal{D}^r,\ S^r \in \mathcal{R}. \tag{3.23}$$

Similarly, when service S^r is rejected, all variables related to the service should be zero, i.e.,

$$\pi^{jr} \leq \zeta^r,\ z^r_{ni} \leq \zeta^r,\ n \in \mathcal{N}, i \in \Omega_1^{|\mathcal{V}^r|},\ j \in \Omega_1^{J^r},\ S^r \in \mathcal{R}. \tag{3.24}$$

Objectives — The objective of the first step is to find the maximum aggregate throughput $\mathbb{R}^*$. Then, we aim to minimize the function and link provisioning costs for all admitted services, subject to the maximum achievable aggregate throughput $\mathbb{R}^*$, in the second step.

Now we present the first step of maximizing the aggregate throughput as

$$\textbf{(P2)}: \max_{\boldsymbol{\mathcal{X}},\boldsymbol{\zeta},\boldsymbol{w}} \sum_{S^r\in\mathcal{R}} R^r \zeta^r \tag{3.25a}$$

$$\text{subject to} \quad (3.2), (3.3), (3.8), (3.10a), (3.17) \tag{3.25b}$$

$$(3.13), (3.15), (3.16), (3.18), (3.22), (3.23), (3.24) \tag{3.25c}$$

$$\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{\zeta}, \boldsymbol{\pi}, \boldsymbol{w} \in \{0, 1\}, \ \boldsymbol{d}, \boldsymbol{\gamma} \geq 0. \tag{3.25d}$$

After solving (3.25), we obtain a configuration that provide a maximal aggregate throughput, $\mathbb{R}^*$, for the given $|\mathcal{R}|$ services and substrate network. However, such configuration can be one among many that can yield $\mathbb{R}^*$. Among all possible configurations, we choose one such that the function and link provisioning costs are minimized.

Therefore, in the second step, we find the combination of admitted services and their multicast topologies such that the function and link provisioning costs are minimized, subject to the maximum achievable aggregate throughput $\mathbb{R}^*$, i.e.,

$$\textbf{(P3)}: \quad \min_{\boldsymbol{\mathcal{X}}, \boldsymbol{\zeta}, \boldsymbol{w}} \sum_{S^r \in \mathcal{R}} \sum_{l \in \mathcal{L}} \sum_{j \in \Omega_1^{J^r}} \sum_{i=0}^{|\mathcal{V}^r|} \alpha \left(\frac{\gamma_{li}^{jr}}{B(l)} + x_{li}^{jr} \right) + \sum_{S^r \in \mathcal{R}} \sum_{i=1}^{|\mathcal{V}^r|} \sum_{n \in \mathcal{N}} \beta \frac{C(f_i^r)}{C(n)} z_{ni}^r \tag{3.26a}$$

$$\text{subject to} \quad (3.2), (3.3), (3.8), (3.10a), (3.17) \tag{3.26b}$$

$$(3.13), (3.15), (3.16), (3.18), (3.22), (3.23), (3.24) \tag{3.26c}$$

$$\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{\zeta}, \boldsymbol{\pi}, \boldsymbol{w} \in \{0, 1\}, \ \boldsymbol{d}, \boldsymbol{\gamma} \geq 0 \tag{3.26d}$$

$$\sum_{S^r \in \mathcal{R}} R^r \zeta^r \geq \mathbb{R}^*. \tag{3.26e}$$

The problem in (3.26) is an MILP, and can be solved optimally by a standard MILP solver. After solving (3.26), we obtain optimal solutions such that the maximal aggregate throughput $\mathbb{R}^*$ is achieved with minimal function and link provisioning costs.

Remark. *The joint multicast routing and NF placement problems for both single-service and multi-service cases are NP-hard.*

Proof. We first show that our single-service problem (P1) can be reduced from the Steiner tree problem in polynomial time. Assume a service request with only one function (f) and multiple destinations ($\mathcal{D}$). We have a physical substrate ($\mathcal{G}$) such that the source (s) is a leaf vertex (in $\mathcal{G}$) connected to the only feasible NFV node for f. The optimal solution can be obtained in two steps. First, we build a Steiner tree from the NFV node to the destinations. Second, we place f on the NFV node, and connect the NFV node to the source (as this is the only feasible option). The first step is NP-hard, while the second step is performed in polynomial time. Thus, (P1) is NP-hard. It is then proved that the multi-service problem (P3) is NP-hard as it includes (P1) as a special case [89, 90]. □

3.4 Heuristic Algorithms

Even though (P1′), (P2), and (P3) in Section 3.3 can be solved optimally by an MILP solver, the computational time is high. A low-complexity heuristic algorithm is needed to efficiently find a solution. The proposed framework is modular in its design, and can be divided into two main steps. First, a mechanism is employed to prioritize service requests based on some heuristics that aim to maximize the aggregate throughput. Second, the prioritized service requests are embedded sequentially using the joint routing and NF placement algorithm for a single-service.

3.4.1 Joint Routing and NF Placement for Single-service Scenario

We design a single-service heuristic algorithm based on the following considerations: (i) Different types of NFs can run simultaneously on an NFV node; (ii) The traversed NF types and their order should be considered for each source-destination (S-D) pair; (iii) The objective is to minimize the provisioning cost of the multicast topology based on (3.11). According to the aforementioned principles, a two-step heuristic algorithm is devised as follows: We first minimize the link provisioning cost by constructing a key-node preferred MST-based multicast topology that connects the source with the destinations; Then, we greedily perform NF placement such that the number of NF instances is minimized. The pseudo-code of the joint routing and NF placement heuristic is shown in Algorithm 1, which is explained in more detail as follows.

First, we copy the substrate network $\mathcal{G}$ into $\mathcal{G}'$. Second, to prioritize the NFV node selection in building the key-node preferred MST, we calculate new link weights for $\mathcal{G}'$ as

$$\omega_{l'} = \alpha \left(\frac{d^r}{B(l')} + 1 \right) + \beta \frac{d^r}{C(m)}, \quad l' = (n,m) \in \mathcal{L}', \mathcal{L}' \subseteq \mathcal{G}',\ S^r \in \mathcal{R} \tag{3.27}$$

where $C(m)$ is set to a small value when m is a switch; otherwise, it represents the processing resource of NFV node m. Then, a key-NFV node is selected iteratively. We construct the metric closure of $\mathcal{G}'$ as $\mathcal{G}''$, where the metric closure is a complete weighted graph with the same node set $\mathcal{N}$ and with the new link weight over any pair of nodes given by the respective shortest path distance. From the metric closure, we find the MST which connects the source node, destination nodes, and the key-NFV node. An initial multicast routing topology ($\mathcal{G}_v$) can be constructed by replacing the edges in the MST with corresponding paths in $\mathcal{G}'$ wherever needed. We then greedily place the NFs from the source of the multicast topology to its destinations with the objective of minimizing the number of NF instances. The cost $\mathbb{C}(\mathcal{G}_v)$ of the new multicast topology (as in (3.11)) and the number $\mathbb{A}(\mathcal{G}_v)$ of successfully embedded NF instances are computed. The objective is to jointly maximize the number of successfully placed NFs and minimize the provisioning cost by iterating over all candidate key-NFV nodes. In every iteration, a new key-NFV node is selected. If $\mathbb{A}(\mathcal{G}_v)$ is increased, we update the selected multicast topology with the new key-NFV node; If $\mathbb{A}(\mathcal{G}_v)$ is unchanged and $\mathbb{C}(\mathcal{G}_v)$ is reduced, we also update the selected multicast topology. If a path cannot accommodate all required NFs (*i.e.*, $f_1, f_2, \ldots, f_{|\mathcal{V}|}$) after selecting a key-NFV node, we devise a corrective subroutine that places the missing NF instances on the closest NFV node from the multicast topology, and the corresponding physical links are rerouted as follows. Let P_t be the path from s to t in $\mathcal{G}_v$, $P_{t,f}$ be the union of the paths such that a missing function (f) is not hosted $\Big(\text{i.e.,}\{P_{t,f} = \cup_{t\in\mathcal{D}} P_t |\, f \,\text{is not hosted}\}\Big)$, and $P_{c,t,f}$ be longest common path before first branch in P. Correspondingly, for each missing NF, we link the nearest applicable NFV node to $P_{c,t,f}$, place the missing NF instance, and remove all unnecessary edges.

So far, a flexible multicast topology that connects the source with the destinations, with all NF instances traversed in order, is constructed. We first resume from the single-path scenario ($J = 1$) to check whether each path in G_v satisfies the link transmission rate requirement as per (3.13), and find an alternative path for each infeasible path. If a single-path solution is infeasible due to (3.13), the heuristic algorithm for the single-path case is extended to the multipath-enabled NF chain embedding problem (with $J > 1$). Enabling multipath routing provides several advantages. It is activated when the transmission rate requirement between two consecutively embedded NF instances cannot be satisfied (i.e., when $B(l) < d^r$, $l \in \mathcal{L}$); and multipath routing is to reduce the overall link provisioning cost compared with the single-path case. For each path between two embedded NF instances that does not satisfy the transmission rate requirement d^r, we increment the number of multipath routes gradually, and look for a feasible solution up to the predefined J. The algorithm declares that the problem instance is infeasible when the number of multipath routes exceeds J. For $J > 1$, we trigger a path splitting mechanism for each path between two embedded NF instances for each S-D pair as follows.

Let $W_{i,i+1}^{t,k}$ be the kth path between two embedded NFs (f_i,f_{i+1}) along the network substrate for destination t ($\in \mathcal{D}$), where the cardinality of all such possible paths is $K^t_{i,i+1}$. We first rank all candidate paths in a descending order based on the amount of residual transmission resource. Then, we sequentially choose the paths from the candidate paths, such that the summation of all chosen paths' residual transmission rate meets the required transmission rate d^r. Assuming the current number of trees is j, the transmission rates allocated on the kth path ($W^{t,k}_{i,i+1}$) is then calculated as

$$\mathbb{R}(W^{t,k}_{i,i+1}) = \frac{B^k_{\min} d^r}{\sum_{k=1}^{\min\{j, K^t_{i,i+1}\}} B^k_{\min}},\ t \in \mathcal{D}, i \in \Omega_0^{|\mathcal{V}|}, k \in \Omega_1^{\min\{j, K^t_{i,i+1}\}},\ S^r \in \mathcal{R} \tag{3.28}$$

where $B^k_{\min}$ is the amount of minimum residual transmission resources for path $W^{t,k}_{i,i+1}$, i.e., $B^k_{\min} = \min_{l \in W^{t,k}_{i,i+1}} B(l)$. Here, the multipath extension method essentially computes a link-disjoint multipath configuration from a single-path route. Therefore, the proposed multipath extension is necessarily prone to the so-called path diminuition problem, in which not all end-to-end multipath-enabled configurations can be devised from a single-path discovery [91, 92].

3.4.2 Multi-service Scenario

To achieve high throughput and to efficiently utilize the network resources, our key strategy is to selectively prioritize the network services contributing significantly to the aggregate throughput with least provisioning cost. Here, we consider a static algorithm, i.e., service requests are available *a priori*. Three principles serving as criteria to prioritize each service for embedding are identified. The first principle is to rank the given network services based on the aggregate throughput, which is defined in (3.21). A network service with higher achieved throughput has higher priority to be embedded in the substrate network, since such service contributes more to the achieved aggregate throughput than a lower ranked service. However, ranking a service based on the throughput alone does not take into account the impact of the provisioning cost. It is impossible to obtain the exact provisioning cost of embedding a service request prior to the routing and placement process, as the problem itself is NP-hard. However, the relative positions of source and destinations can provide hints on the cost necessitated to host such service. Given a network service with the destinations far from each other (i.e., highly distributive), the provisioning cost is large since more physical links and multicast replication points are expected to connect

Algorithm 1: Heuristic algorithm for the joint routing and NF placement

Procedure constrainedJRP $(\mathcal{G}, S^r)$;

Input : $\mathcal{G}'(\mathcal{N}', \mathcal{L}')$, $S^r = (s^r, \mathcal{D}^r, f_1^r, f_2^r, \ldots, f_{|\mathcal{V}|}^r, \overline{d^r})$

Output: $\mathcal{G}_v$

$\mathbb{C}_r \leftarrow \infty$; $\mathbb{A}_r \leftarrow 0$;

for $n \in \mathcal{M}$ **do**

  $\mathcal{G}'' \leftarrow$ MetricClosure$(\mathcal{G}', \{n, s, \mathcal{D}\})$; $\mathcal{G}_v^{temp}(\mathcal{N}_v, \mathcal{L}_v) \leftarrow$ KruskalMST$(\mathcal{G}'')$;

  for *path from s to each $t \in \mathcal{D}$* **do** place NFs from $\mathcal{V}$ on available NFV nodes sequentially subject to (3.10);

  if $\mathbb{A}(\mathcal{G}_v^t) = \mathbb{A}_r \,\&\, \mathbb{C}(\mathcal{G}_v^t) < \mathbb{C}_r$ **then** $\mathcal{G}_v \leftarrow \mathcal{G}_v^t$; $\mathbb{C}_r \leftarrow \mathbb{C}(\mathcal{G}_v^t)$;

  else if $\mathbb{A}(\mathcal{G}_v^t) > \mathbb{A}_{ref}$ **then** $\mathcal{G}_v \leftarrow \mathcal{G}_v^t$; $\mathbb{A}_{ref} \leftarrow \mathbb{A}(\mathcal{G}_v^t)$; $\mathbb{C}_{ref} \leftarrow \mathbb{C}(\mathcal{G}_v^t)$;

end

for *each missing NF (f) from $\mathcal{G}_v$* **do** link nearest NFV node that can host f to $P_{c,t,f}$, and remove unnecessary edges;

for *path from f_i to f_{i+1} for each t in $\mathcal{G}_v$* **do**

  success $\leftarrow$ false;

  for $j = 1 : J$ **do**

    Find $temp = \min\{K_{i,i+1}^t, j\}$ paths from $\mathcal{G}$;

    if $\sum_{k=1}^{temp} \min_{l \in W_{i,i+1}^{t,k}} B(l) \geq \overline{d^r}$ **then** allocate transmission resource for each kth path $(W_{i,i+1}^{t,k})$ using (3.28); success $\leftarrow$ true; break;

  end

  if *success = false* **then** break;

end

if *success = false* **then** $\mathcal{G}_v \leftarrow$ none;

else return $\mathcal{G}_v$;

the destinations. Moreover, the distance from the source to destinations is proportional to network service's provisioning cost as a long routing path with a relatively large number of NF instances is needed to establish the multicast topology.

Combining both effects of the distances between destinations and the distance from source to destinations, we define a *distribution level*, denoted by g^r, as the product of two components. The first component is Y^r/Y, where Y is the area of the smallest convex polygon that spans all nodes in the network, and Y^r is the area of the smallest convex polygon that spans all destinations of service S^r. The ratio Y^r/Y provides an estimate of how dense a set of destinations of one service is in a given area of the network. Note that existing algorithms to determine the convex hull of a set of points and to calculate the area of an arbitrary shape are available [93]. The second component is q^r/q, where q^r is the distance from source to the center point of the set of destinations in service S^r, and q is the largest distance between two arbitrary nodes in the substrate network. The center point of the set of destinations in one service plays a role as a representer for all destinations in that service. The ratio q^r/q can measure how far is the source from the destinations. The distribution level metric g^r is thus expressed as

$$g^r = \frac{Y^r q^r}{Y q}, \quad S^r \in \mathcal{R}. \tag{3.29}$$

A larger value of g^r implies a higher distribution level, where the source is positioned farther away from its destinations and the destinations are more distribution in the whole network coverage area. A largely distributive network service consumes more network resources, resulting in a high provisioning cost. Therefore, a service with a lower value of g^r has a higher priority to be embedded in order to preserve the substrate network resources. Although the parameters in (3.29) can be calculated with regard to the hop-count (e.g., via computing the shortest path) which is a more representative metric than the Euclidean distance, it is exhaustive to do so.

Next, we introduce a third metric named *size* (U^r) to incorporate both (3.21) and (3.29) as follows,

$$U^r = R^r(1 - g^r), \quad S^r \in \mathcal{R} \tag{3.30}$$

where the goal is to prioritize a service with higher throughput subject to a correction factor of $1 - g^r$ for how distributive (costly) it is.

To summarize, we calculate the throughput, the distribution level, and the size for each service request using (3.21), (3.29), and (3.30). Then, service requests are sorted according to their sizes in a descending order, and embedded using Algorithm 1.

3.4.3 Complexity Analysis

First, the heuristic algorithm for the single-service scenario (Algorithm 1) iterates over $|\mathcal{M}|$ NFV nodes to find a key-NFV node. For each potential key-NFV node, a multicast topology is constructed and compared with the previous iteration (in Lines 3-8). Denote the set of destinations, source node, and the key NFV node by $\mathcal{T}$ (i.e., $\mathcal{T} = \{n, s, \mathcal{D}\}$). For each potential key-NFV node, we construct the metric closure on $\mathcal{T}$, which is computed by considering the all-pairs shortest-path algorithm on $\mathcal{T}$. Thus, the worst-case running time is $\mathcal{O}(|\mathcal{N}||\mathcal{T}|^2)$. Then, we find an MST on the constructed metric closure. The MST is transformed to the Steiner tree by replacing each edge with the shortest path, and removing unnecessary edges (Line 4). The worst-case time complexity of the MST-based Steiner tree is dominated by the metric closure. The construction of the Steiner tree is followed by an NF placement process that requires $\mathcal{O}\{|\mathcal{D}||\mathcal{M}|\}$ time in the worst-case since a path from the source to each destination passes by at most $|\mathcal{M}|$ NFV nodes (Line 5). To extend to the multipath scenario, in Lines 10-17, for each path between two embedded NF instances, we find up to J paths and split the traffic according to (3.28), which requires $\mathcal{O}(J\,|\mathcal{D}|\,|\mathcal{V}|\,|\mathcal{N}|\log|\mathcal{N}|)$ time in the worst-case. For the multi-service scenario, we measure the size of each request, followed by a sorting algorithm based on the size in (3.30), which requires $\mathcal{O}(|\mathcal{R}|\log|\mathcal{R}|)$ time. Consequently, each service request is embedded sequentially. Hence, the overall worst-case time complexity of the multi-service multi-path scenario is $\mathcal{O}\Big(|\mathcal{R}|\log|\mathcal{R}| + |\mathcal{N}||\mathcal{D}|(|\mathcal{D}| + J\,|\mathcal{V}|\log|\mathcal{N}|)\Big)$.

3.5 Simulation Results

In this section, simulation results are presented to evaluate the optimal and heuristic solutions to the joint multicast routing and NF placement problems for the single- and multi-service scenarios, considering both single-path and multipath routing cases. The simulated substrate network is a mesh-topology based network [94], with $|\mathcal{N}| = 100$ and $|\mathcal{L}| = 684$, as shown in Fig. 3.3. We randomly choose 25 vertices as NFV nodes in the mesh network, and the transmission rate of physical link l and the processing rate of NFV

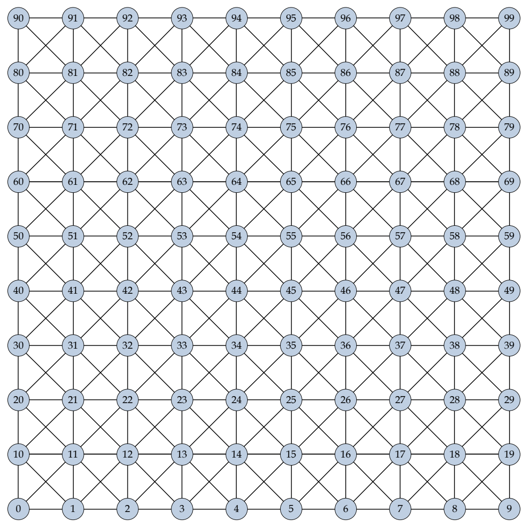

Figure 3.3: Mesh topology with $|\mathcal{N}| = 100$ and $|\mathcal{L}| = 684$.

node n are uniformly distributed between 0.5 and 2 Million packet/s (Mpacket/s), i.e., $B(l), C(n) \sim \mathcal{U}(0.5, 2)$ Mpacket/s. To solve the formulated MILP problems, we use the Gurobi solver with the branch and bound mechanism, where the weighting coefficients are set as $\alpha = 0.6$, $\beta = 0.4$. The processing rate requirement of the NFs are set to be linearly proportional to the incoming data rate, i.e., $C(f^r) = d^r$ [95].

First, we conduct a comparison between the optimal solution of the single-service single-path MILP formulation and the solution of the heuristic. We generate random service requests where the numbers of NFs and destinations are varied from 3 to 14 and 2 to 11, respectively. The data rate of the generated service requests are set to $d^r = 0.2$ Mpacket/s. The total provisioning cost obtained from both optimal and heuristic solutions are shown in Fig. 3.4, as the number of destinations $|\mathcal{D}|$ or NFs $|\mathcal{V}|$ increases. It can be seen that the total provisioning cost increases with $|\mathcal{D}|$ or $|\mathcal{V}|$. As $|\mathcal{D}|$ increases, the costs obtained from both optimal and heuristic solutions grows with nearly a constant gap. Adding destinations incurs a higher cost than adding NF instances, since additional physical links and NF instances are required for each destination.

Next, we compare the proposed heuristic algorithm with the heuristic algorithm named HA-TAA[4] in [2]. For a fair comparison, we consider only the single-path scenario. To add

[4]The heuristic algorithm (HA-TAA) in [2] assumes uncapacitated network substrate. We modify the

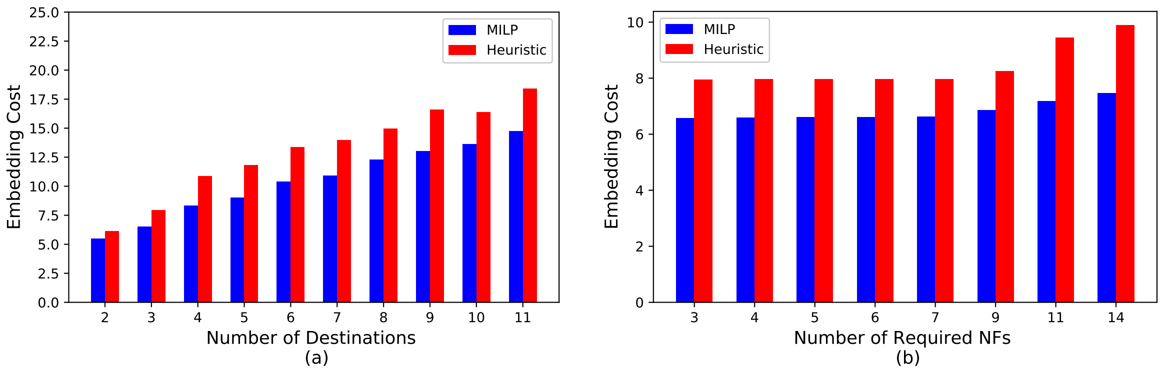

Figure 3.4: Embedding cost with respect to (a) the number of destinations and (b) the number of required NFs.

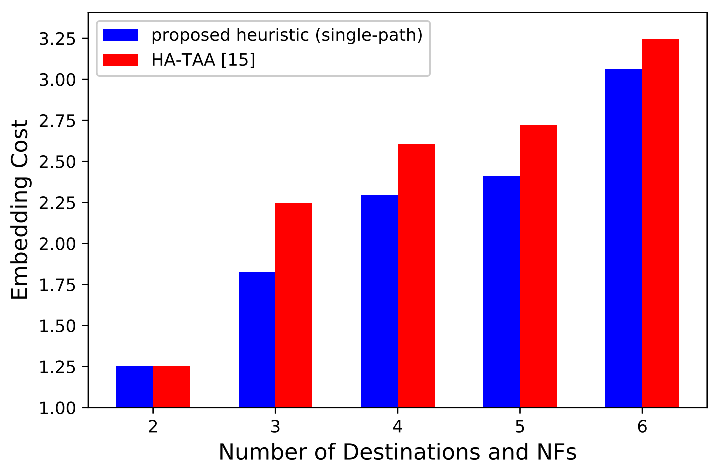

Figure 3.5: Comparison of embedding cost between the proposed heuristic algorithm and HA-TAA heuristic algorithm in [2].

heterogeneity to NFV nodes, among the 25 NFV nodes and up to 6 types of NFs, each NF type can be hosted in an NFV node with the chance of 80%. We randomly generate 10 multicast requests that have equal number of NFs and destinations (i.e., $|\mathcal{V}| = |\mathcal{D}|$), and each multicast request is embedded on 30 network substrate instances to reduce the effect of randomness. As shown in Fig. 3.5, the proposed algorithm consistently outperforms HA-TAA when $|\mathcal{V}| = |\mathcal{D}| > 2$. However, the average running time for the modified HA-TAA algorithm was around 0.115 seconds, whereas our proposed heuristic algorithm took about 2.110 seconds due to the higher time complexity of the involved algorithm. In [2], the HA-TAA algorithm first finds the shortest path from the source to the NFV nodes that can host the required NFs sequentially, and the shortest path from the last NFV node to the respective closest destination. Second, it connects the destinations through an MST. Since the placement of each NF is only dependent on the previous NF, the HA-TAA algorithm may take a long path to place NFs before connecting the last NFV node to the closest destination. In our algorithm, we first focus on finding a Steiner tree from the source to the destinations through a key-NFV node, thereby optimizing the link provisioning cost. Then, we modify some of the links to host all required NF instances (if needed). Here, NFs can be duplicated to serve different sets of destinations, thereby providing more flexibility and reduced overall provisioning cost. When $|\mathcal{V}| \leq 2$, the HA-TAA is specifically more efficient as the NF placement is related to the locations of both the source and the closest destination.

Fig. 3.6 shows the advantage of multipath routing over single-path routing using the proposed optimal formulation. We depict the maximum supported data rate d^r, with which the NF embedding is feasible. Compared to the single-path routing case, multipath routing ($J = 2$) always supports higher or equal data rates. With an increase of the number of required NFs, the maximum supported data rate decreases, and converges to the single-path scenario; the processing cost becomes more significant and the number of candidate NFV nodes and paths decrease.

Next, we demonstrate the effectiveness of the proposed heuristic admission mechanism for the multi-service scenario discussed in Subsection 3.4.2. First, we divide the mesh topology into four access network regions and one core network region as indicated in Fig. (3.7). Three scenarios with different available processing and transmission rates on

algorithm to the capacitated scenario, change the weights to match our objective function, and assume that an NFV node can host multiple NF instances subject to processing resources. The objective function excludes the minimization of number of hops for fairness.

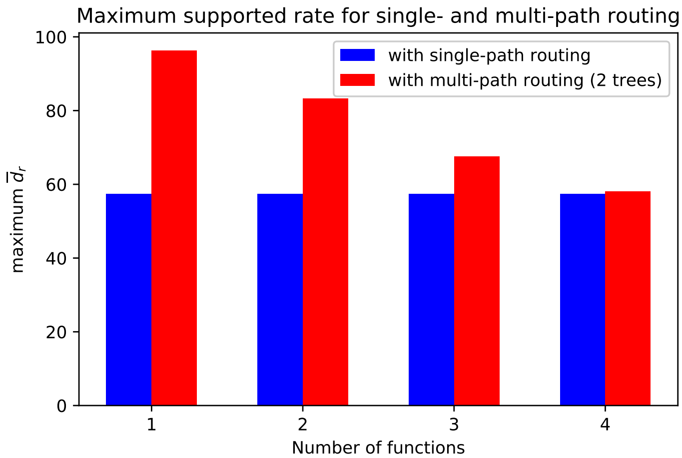

Figure 3.6: Maximum supported data rate (d^r) for both single-path and multipath routing scenarios using the proposed optimal formulation.

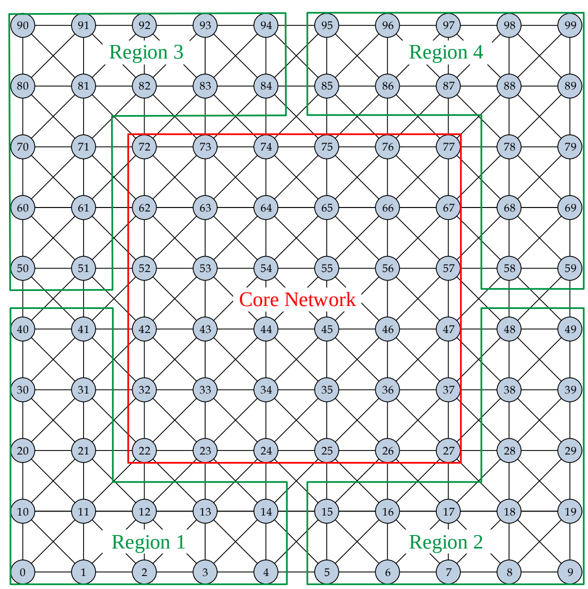

Figure 3.7: Mesh topology with $|\mathcal{N}| = 100$, $|\mathcal{L}| = 684$, and 4 access regions and 1 core network region.

Table 3.1: Processing and transmission rates

Scenarios	Processing rate	Link transmission	NFV nodes
Scenario 1	$\mathcal{U}(3,8)$ Mpacket/s	$\mathcal{U}(3,8)$ Mpacket/s	47
Scenario 2	$\mathcal{U}(4,9)$ Mpacket/s	$\mathcal{U}(4,9)$ Mpacket/s	50
Scenario 3	$\mathcal{U}(5,10)$ Mpacket/s	$\mathcal{U}(5,10)$ Mpacket/s	53

the NFV nodes and physical links are considered in the scale of Mpacket/s, as listed in Table 3.1. Each service randomly originates from one access network region, traverses the core network region, and terminates in one of the other three access network regions. For each network scenario, to simulate network congestion, 35 multicast service requests are randomly generated and submitted for embedding, where the data rate d^r of service S^r is randomly distributed between $[1.5, 3.5]$ Mpacket/s, and the number of NFs and destinations are randomly generated as $|\mathcal{V}^r| = \{3, 4\}$ and $|\mathcal{D}^r| = \{3, 4, 5\}$. For each scenario, the service generation and embedding are repeated 5 times to reduce the effect of randomness. As a benchmark, we use a second strategy that randomly selects the service requests for embedding.

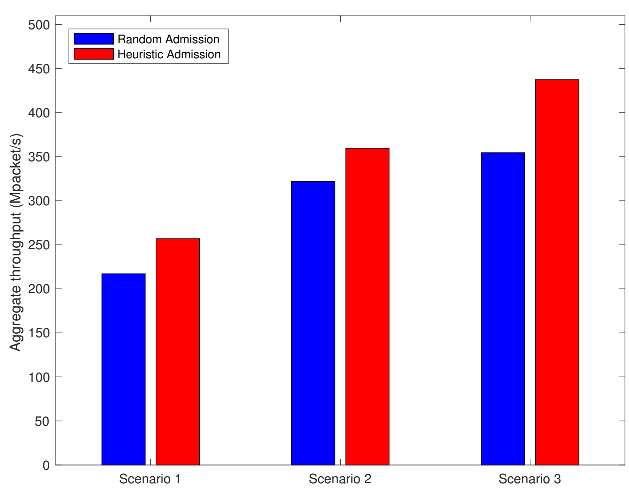

Figure 3.8: Aggregate throughput comparison of the random admission and the proposed heuristic admission.

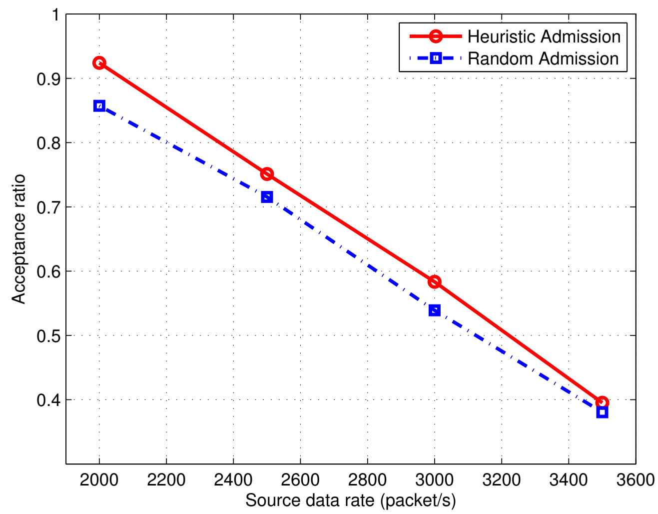

Figure 3.9: Acceptance ratio comparison of the random admission and the proposed heuristic admission.

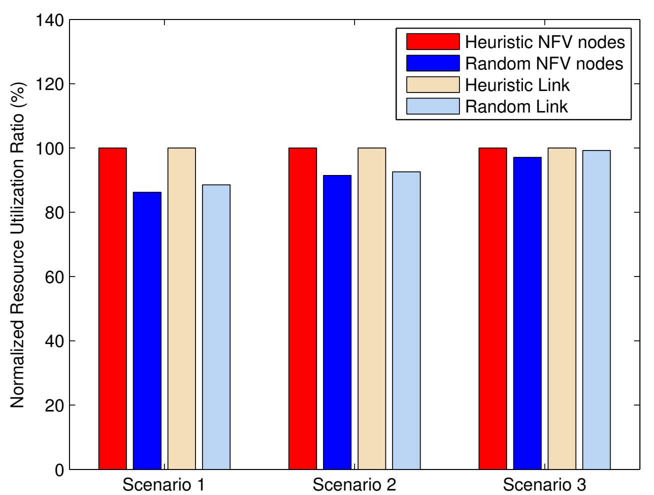

Figure 3.10: Resource utilization ratio comparison of the random admission and the proposed heuristic admission.

Fig. 3.8 shows the aggregate throughput $\mathbb{R}$ as in (3.20) achieved by the random and heuristic admission solutions under three scenarios specified in Table 3.1, which increases as the available processing and transmission rates increase. As shown, the aggregate throughput of the proposed heuristic solution exceeds the aggregate throughput of the random admission by 15.63% on average over all the scenarios. This is because the size metric used in the heuristic solution ensures that the services with a larger throughput are embedded with a higher priority. Fig. 3.9 shows the acceptance ratio of the total 35 service requests under different average data rates. The acceptance ratio of the heuristic solution exceeds the random solution over all source data rate levels by 4% on average.

Fig. 3.10 compares the normalized resource utilization between the random and heuristic admission solutions. The utilization ratio is calculated as the amount of resources consumed by all embedded services over the total available resources of the substrate network. We normalize the measured utilization ratio by the utilization ratio of the proposed heuristic admission to highlight the enhancement brought by the heuristic solution. In scenarios 1 and 2, the heuristic solution increases the utilization ratio by approximately 12.62% and 7.97% over that of the random admission solution, respectively. In scenario 3, the difference in the utilization ratio between the two solutions decreases, especially the link transmission usage. Recall that the heuristic scheme aims to maximize the aggregate throughput as defined in (3.20). Therefore, although in scenario 3 the utilization ratio achieved by the heuristic scheme is close to that of the random scheme, the aggregate throughput achieved by the former is larger as shown in Fig. 3.8.

Chapter 4

Online Joint Routing and NF Placement for Unicast and Multicast Services

4.1 System Model

In this section, we present the system model under consideration for Problem II, which is similar to that in Chapter 4, albeit with some minor differences concerning the NFs and the handling of service requests.

4.1.1 Network Functions and Online Service Requests

From the perspective of quality of service, we consider two types of NFs, namely mandatory and best-effort. As discussed in Subsection 1.2.2, a successful placement of a network service is contingent on successfully placing only the set of mandatory NFs, whereas best-effort NFs are not necessary for the correctness of a network service.

We consider an ongoing input sequence of unicast and multicast service requests $\sigma = (S^1, S^2, \dots)$ that arrive in an online fashion. The rth service request is expressed as

$$S^r = (s^r, \mathcal{D}^r, \mathcal{V}^r, d^r), \qquad S^r \in \sigma \tag{4.1}$$

where the source and destination nodes are s^r and $\mathcal{D}^r$, respectively; parameter d^r denotes the required transmission rate in packet per second (packet/s); $\mathcal{V}^r = \{f_1^r, f_2^r, \ldots, f_{|\mathcal{V}^r|}^r\}$ represents the set of NFs that need to be traversed in an ascending order for the source-destination pair. For simplicity, for each service request, each NF requires an equal amount of processing resources of $C(f^r)$ in packet/s. The online algorithm to be developed thereafter can be generalized for NFs with arbitrary processing requirements. The sets of mandatory and best-effort NFs are denoted by $\mathcal{V}_m^r$ and $\mathcal{V}_b^r$, respectively.

4.1.2 Network Substrate

We are given a capacitated network substrate $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ and $\mathcal{L}$ are the sets of nodes and links, respectively. Each physical link l ($\in \mathcal{L}$) has a residual transmission resource, $B(l)$, in packet/s. Each node n ($\in \mathcal{N}$) has a residual processing resource, $C(n)$, in packet/s. Nodes can be either (i) switches that are capable of forwarding traffic only (with $C(n) = 0$), or NFV nodes (e.g., commodity servers) that are capable of both forwarding traffic and operating a set of NF instances. An NFV node is capable of provisioning a number of NF instances simultaneously as long as the available processing resources satisfy the deployed NF processing requirements.

4.2 Problem Description

We are given a sequence of service requests σ that is revealed over time, i.e., the service requests arrive one by one without knowledge of future arrivals. We need to define a profit function, whose main goal is to maximize the amortized processing and transmission throughput. Recall the amortized throughput is the weighted total transmission and processing resources reserved for all the accepted service requests. However, the profit function (and the online algorithm) should capture both the mode of communication (i.e., unicast and multicast) and the heterogeneity of the NF types (i.e., best-effort and mandatory). That is, maximizing the amortized throughput alone would unfairly favor unicast to multicast services due to the larger number of destinations in the latter. Yet, the multicast mode of communication is more efficient as it is shown to reduce the bandwidth consumption in backbone networks by over 50% in contrast to the unicast mode [9]. Moreover, although best-effort NFs are optional, their use should be incentivized.

In the following, we first define a profit function that accurately captures the system model and operational design requirements. Then, we develop a path-based formulation for an offline profit-maximization problem. The offline formulation is omniscient, where it has complete a priori knowledge of the entire sequence of service requests. Moreover, it yields the optimal combination of service requests and their routing and NF placement configurations, such that the profit function is maximized. Given the offline formulation, through a primal-dual analysis, we develop an all-or-nothing/all-or-something online algorithm to deal with each service request in a dynamic manner, while providing competitive guarantees against the optimal offline adversary.

4.3 Problem Formulation

4.3.1 The Objective (Profit) Function

We consider two profit functions, ϱ^r and ρ^r, that correspond to the transmission and processing resource types, respectively. We know that the number of physical links required for a service in multicast mode is always less than or equal to the overall number of links needed for the equivalent services in unicast mode. Therefore, for the rth service, an upper bound on the ratio of the number of links in a multicast topology to the number of links in an equivalent unicast service is given by $|\mathcal{D}^r|$. Notably, it has been experimentally shown that the respective ratio is $|\mathcal{D}^r|^k$, where $k = 0.8$ for many real and generated network topologies [96]. Therefore, for the transmission resources, to provide a non-discriminatory treatment between multicast and unicast services, we define ϱ^r to be proportional to both (i) the required data rate of the service request and (ii) the kth power of the number of included destinations,

$$\varrho^r = d^r|\mathcal{D}^r|^k, \qquad S^r \in \sigma \tag{4.2}$$

where $k = 0.8$.

For the processing resources, let the amount of the incentive (and disincentive) for including (and excluding) the set of best-effort NFs for the rth service request be given by η_b^r (and η_m^r), respectively, where $\eta_b^r \geq \eta_m^r \geq 1$. We let ρ^r be proportional to (i) the processing throughput accrued from placing the NFs, and (ii) the incentive from including (or excluding) the set of best-effort NFs,

$$\rho^r = \eta^r C(f^r), \qquad S^r \in \sigma \tag{4.3}$$

where $\eta^r \in \{\eta_m^r, \eta_b^r\}$ is a decision variable, with $\eta^r = \eta_b^r$ indicating that the set of best-effort NFs from the rth service is included, and $\eta^r = \eta_m^r$ indicating otherwise. One method is to set η^r to the number of included NFs in the rth service (e.g., $\eta_b^r = |\mathcal{V}^r|$ when all best-effort NFs are accepted, and $\eta_m^r = |\mathcal{V}_m^r|$ otherwise). In contrast to existing relevant literature, ρ^r varies with the logical topology of the composed service request. Solutions that exclude the set of best-effort NFs can have lower provisioning costs since they require less number of NF instances, and therefore can be accepted if otherwise not feasible by the admission mechanism. Since we have two resource types, the overall profit from accepting the rth service request is given by $\alpha\varrho^r + \beta\rho^r$, where α and β are two coefficients to indicate the relative importance (or scalarization) of each profit function, with $\alpha, \beta \geq 1$.

4.3.2 Primal-Dual Schema

First, we develop a path-based formulation for the offline multi-resource profit-maximization problem. There are two possible routing and NF placement models for the offline formulation, namely unsplittable (or fixed) and splittable. In the unsplittable model, a service request is restricted to an integral solution, where only one path is used for a unicast service (or only one tree is used for a multicast service).

Formally, let all the possible paths/trees for unicast/multicast service request S^r be given by set $\mathcal{P}(r)$. Let P ($\in \mathcal{P}(r)$) be a path/tree on the network substrate that is selected to host service request S^r. Here, P comprises the physical links and NFV nodes that host the virtual edges and NF instances, respectively. Hereafter, the term "path" is used liberally; throughout this chapter, P can be regarded as a *tree* when provisioning a multicast service. Define y_P^r as the fraction of flow allocated for service S^r along path P ($\in \mathcal{P}(r)$). In the unsplittable model, $y_p^r \in \{0, 1\}$ and $\sum_{p \in \mathcal{P}(r)} y_p^r = 1$. In the splittable model, a service can be routed on several paths, where multipath routing is enabled and NF instances can be split, i.e., $y_p^r \in [0, 1]$ and $\sum_{p \in \mathcal{P}(r)} y_p^r = 1$. The optimal splittable model provides a larger profit compared to the unsplittable variant. This is because the linear relaxation provides an upper bound to the unsplittable (combinatorial) problem. Even stronger performance (in terms of maximizing the profit) can be achieved when the splittable model is relaxed to allow for the sum of fractional allocations to be at most 1, i.e., $\sum_{p\in\mathcal{P}(r)} y_p^r \leq 1$.

In data networks (such as in fifth-generation networks and the Internet), the scale of demands is large relative to the granularity at which it can be managed/routed [97], especially in software-defined networks. Therefore, it is desired to design and measure the (competitive) performance of a designed apparatus against a splittable offline model (i.e., with multipath routing and NF splitting) [98]. Therefore, although the online algorithm provides an unsplittable solution, its performance will be measured against the splittable offline model.

The path-based offline profit-maximization formulation for the fractional splittable model is expressed in (4.4).

Dual – profit-maximization problem

$$\max \alpha \sum_{S^r\in\sigma} \sum_{P\in\mathcal{P}(r)} \varrho^r y_P^r + \beta \sum_{S^r\in\sigma} \sum_{P\in\mathcal{P}(r)} \rho^r y_P^r \tag{4.4a}$$

subject to :

$$\forall S^r \in \sigma : \sum_{P\in\mathcal{P}(r)} y_P^r \leq 1 \tag{4.4b}$$

$$\forall l \in \mathcal{L} : \sum_{S^r\in\sigma} \sum_{P\in\mathcal{P}(r)|l\in P} d^r y_P^r \leq B(l) \tag{4.4c}$$

$$\forall n \in \mathcal{N} : \sum_{S^r\in\sigma} \sum_{P\in\mathcal{P}(r)|n\in P} C(f^r) y_P^r \leq C(n) \tag{4.4d}$$

$$\forall S^r \in \sigma, P \in \mathcal{P}(r) : \; y_P^r \geq 0. \tag{4.4e}$$

If all service requests in σ are known a priori, solving the formulation in (4.4) yields the optimal splittable all-or-something/all-or-something packing configuration for all the accepted services from σ. In (4.4a), we maximize the overall profit function accrued by the accepted service requests.

The first set of constraints in (4.4b) requires that the sum of the fractional allocations for each service request along all possible paths is bounded above by unity. Constraints (4.4c) and (4.4d) represent the transmission and processing resource constraints on the physical links and the NFV nodes, respectively.

In the context of the online version of the problem, service requests in σ are revealed over time (in discrete steps). The idea is to develop an online solution that maintains a feasible set whenever a new service request arrives *in a controlled manner* to guarantee certain competitive performance. This is achieved by first deriving the primal of (4.4). Second, we need to ensure that the online algorithm produces solutions such that the objective function of the primal and dual are bounded, which will be explained in Subsection 4.4.1.

Next, we present the corresponding primal formulation in (4.5). Given (4.4b), we assign variable z^r for each request S^r, where $z^r \in [0, \max\{\varrho^r, \rho^r\}]$. Given (4.4c) and (4.4d), we assign variables $\bar{x}(l)$ and $\tilde{x}(n)$ for each physical link l ($\in \mathcal{L}$) and NFV node n ($\in \mathcal{N}$), respectively, where $\bar{x}(l) \in [0, |\mathcal{D}|^k_{\max}]$, $\tilde{x}(n) \in [0, \frac{\eta_{\max,}}{\eta_{\min}}]$, $\eta_{\max} = \max_{S^r \in \sigma} \eta^r$, $\eta_{\min} = \min_{S^r \in \sigma} \eta^r$, and $|\mathcal{D}|_{\max} = \max_{S^r \in \sigma} |\mathcal{D}^r|$. Through the tableau method, the primal formulation is expressed in (4.5).

Primal

$$\min \sum_{l \in \mathcal{L}} B(l)\, \bar{x}(l) + \sum_{n \in \mathcal{N}} C(n)\, \tilde{x}(n) + \sum_{S^r \in \sigma} z^r \tag{4.5a}$$

$$\text{subject to :}$$

$$\forall S^r \in \sigma, P \in \mathcal{P}(r) : \sum_{l \in P \cap \mathcal{L}} d^r \bar{x}(l) + z^r \geq \alpha \varrho^r \tag{4.5b}$$

$$\forall S^r \in \sigma, P \in \mathcal{P}(r) : \sum_{n \in P \cap \mathcal{N}} \tilde{x}(n) + z^r \geq \beta \rho^r \tag{4.5c}$$

$$\forall S^r \in \sigma, l \in \mathcal{L}, n \in \mathcal{N} : z^r, \bar{x}(l), \tilde{x}(n) \geq 0. \tag{4.5d}$$

In what follows, we develop an admission mechanism that is based on the primal-dual formulation in (4.4) and (4.5).

4.4 Primal-Dual based Admission Mechanism

4.4.1 The Approach

In this subsection, we provide a systematic approach to deriving the operational cost model of the physical links and NFV nodes as well as the admission mechanism. Let A and D

be the value of the objective function of the primal and the dual solutions as a result of the online algorithm, respectively. Using the weak duality concept, for the aforementioned primal-dual formulation, we know that $D \leq A$. Therefore, in order to have a provable competitive ratio, we need to bound the objective functions of the primal and dual such that

$$A \leq 2\xi D, \tag{4.6}$$

while maintaining the constraints of the primal and dual formulations satisfied, where 2ξ will be the competitive ratio. However, service requests arrive in an online fashion. Therefore, instead, it is sufficient for the online algorithm to bound the change between the primal and dual objectives whenever a new service request arrives such that [39]

$$\frac{\partial A}{\partial y_P^r} \leq 2\xi \frac{\partial D}{\partial y_P^r}, \quad S^r \in \sigma. \tag{4.7}$$

Due to the multi-resource form of the objective functions in (4.4a) and (4.5a), we can re-express inequality (4.7) as

$$\sum_{l \in \mathcal{L}} B(l) \frac{\partial \bar{x}(l)}{\partial y_P^r} + \sum_{n \in \mathcal{N}} C(n) \frac{\partial \tilde{x}(n)}{\partial y_P^r} + \frac{\partial z^r}{\partial y_P^r} \leq 2\varphi\alpha\varrho^r + 2\phi\beta\rho^r,\ S^r \in \sigma \tag{4.8}$$

where $\xi = \max\{\varphi, \phi\}$, and φ and ϕ are some other constants. The right-hand side of inequality (4.8) has two profit functions, each of which corresponds to a resource type. Therefore, to satisfy inequality (4.8), it is sufficient to find some functions, $\bar{x}(l)$, $\tilde{x}(n)$ and z^r, such that

$$\sum_{l \in \mathcal{L}} B(l) \frac{\partial \bar{x}(l)}{\partial y_P^r} \leq 2\varphi\alpha\varrho^r, \quad S^r \in \sigma \tag{4.9a}$$

$$\sum_{n \in \mathcal{N}} C(n) \frac{\partial \tilde{x}(n)}{\partial y_P^r} \leq 2\phi\beta\rho^r, \quad S^r \in \sigma \tag{4.9b}$$

$$\frac{\partial z^r}{\partial y_P^r} \leq 0, \quad S^r \in \sigma \tag{4.9c}$$

while maintaining the constraints of the primal and dual formulations satisfied.

Starting with the transmission resource type, we need to satisfy (4.9a) while maintaining feasibility in constraints (4.4c) and (4.5b). To do so, let the solution of the first partial derivative in (4.9a) follow the following form,

$$\sum_{l\in\mathcal{L}} B(l)\frac{\partial \bar{x}(l)}{\partial y_P^r} = \varphi \sum_{l\in\mathcal{L}} \Big(d^r \bar{x}(l) + \frac{\alpha\varrho^r}{L}\Big), \quad S^r \in \sigma \tag{4.10}$$

where L is the maximum number of hops in a path for a unicast service (or maximum number of physical links in a tree for a multicast service), i.e., $\sum_{l\in\mathcal{L}} 1 \leq L$. After substituting (4.10) in (4.9a), we impose a requirement that $\sum_{l\in\mathcal{L}} d^r \bar{x}(l) \leq \alpha\varrho^r$ for (4.9a) to hold. Having satisfied (4.9a), we need to derive cost function $\bar{x}(l)$ by solving the differential equation in (4.10). Rearranging the terms in (4.10), we have a differential equation of the following form,

$$\sum_{l\in\mathcal{L}} \frac{\partial \bar{x}(l)}{\partial y_P^r} + \sum_{l\in\mathcal{L}} \frac{-\varphi d^r}{B(l)} \bar{x}(l) = \sum_{l\in\mathcal{L}} \frac{\varphi\alpha\varrho^r}{B(l)L}. \tag{4.11}$$

Define the integrating factor

$$\begin{aligned} I &= \exp\Big(\sum_{S^r\in\sigma|l\in P\in\mathcal{P}(r)} \int \frac{-\varphi d^r}{B(l)} \partial y_P^r\Big) \\ &= C\exp\Big(\frac{-\varphi}{B(l)} \sum_{S^r\in\sigma|l\in P\in\mathcal{P}(r)} d^r y_P^r\Big) \end{aligned} \tag{4.12}$$

where C is an arbitrary constant, and multiply both sides of (4.11) by I, we get

$$I\Big(\frac{\partial \bar{x}(l)}{\partial y_P^r} + \frac{-\varphi d^r}{B(l)} \bar{x}(l)\Big) = \frac{\varphi\alpha\varrho^r}{B(l)L} I. \tag{4.13}$$

Through the following identity by the product rule,

$$\frac{\partial}{\partial y_P^r}(Ix) = I\frac{\partial x}{\partial y_P^r} + \frac{-\varphi d^r}{B(l)} Ix, \tag{4.14}$$

we can express $\bar{x}(l)$ as

$$\begin{aligned} \bar{x}(l) &= I^{-1}\frac{\varphi d^r}{B(l)L}\int I\partial y_P^r, \\ &= \frac{1}{L}\big(-Ce^{\frac{\varphi}{B(l)}\sum_{S^r\in\sigma|l\in P\in\mathcal{P}(r)} d^r y_P^r} - 1\big), \quad l\in\mathcal{L}. \end{aligned}$$

Initially, before the arrival of any service request, we require that $\bar{x}(l) = 0$, which occurs when $\frac{1}{B(l)}\sum_{S^r\in\sigma|l\in P\in\mathcal{P}(r)} d^r y_P^r = 0$. Thus, we set $C = -1$. We also require that $\bar{x}(l) \geq$

$\alpha|\mathcal{D}^r|^k$ when physical link l ($\in \mathcal{L}$) is saturated, i.e. when $\frac{1}{B(l)} \sum_{S^r \in \sigma | l \in P \in \mathcal{P}(r)} d^r y_P^r = 1$. Hence, we need $\varphi \geq \ln(\alpha L |\mathcal{D}|^k_{\max} + 1)$. Therefore, $\bar{x}(l)$ can be expressed as

$$\bar{x}(l) = \frac{1}{L}(e^{\varphi \frac{\sum_{S^r \in \sigma | l \in P \in \mathcal{P}(r)} d^r y_P^r}{B(l)}} - 1), \quad l \in \mathcal{L}. \tag{4.15}$$

Note that constraint (4.4c) is maintained feasible over the whole range of $\bar{x}(l)$. In the admission mechanism, we need to ensure that there is a sufficient protection to the resources before accepting a service to avoid the scenario of accidentally violating the resources (which occurs when $\bar{x}(l) \geq \alpha|\mathcal{D}^r|^k$). It turns out that, due to Lemma 1 (to be derived later), if the required data rate is bounded above by $d^r \leq \frac{\min_{l \in \mathcal{L}} B(l)}{\varphi}$, we need $\varphi \geq \ln(2\alpha L|\mathcal{D}|^k_{\max} + 2)$. Note that the edge costs include variables from future requests (i.e., y_P^r, $\forall S^r \in \sigma$). However, since this is an online algorithm, future variables can be initialized to zero until the respective service requests are parsed through the admission mechanism. We can express the edge costs in a multiplicative recursive manner as

$$\bar{x}^r(l) = \bar{x}^{r-1}(l) e^{\varphi \frac{d^r}{B(l)}} + \frac{1}{L}(e^{\varphi \frac{d^r}{B(l)}} - 1), \quad l \in \mathcal{L},\ S^r \in \sigma \tag{4.16}$$

where $\bar{x}^r(\cdot)$ is the edge cost after embedding the rth service request, and $\bar{x}^0(\cdot)$ is set to zero. Now, we need to ensure that constraint (4.5b) is maintained feasible. Since we set $\sum_{l \in \mathcal{L}} d^r \bar{x}(l) \leq \alpha \varrho^r$, we require that $z^r \geq \alpha \varrho^r - \sum_{l \in \mathcal{L}} d^r \bar{x}(l)$ for (4.5b) to be maintained feasible, and for (4.9c) to hold.

By following a similar procedure, for the processing resource type, we need to satisfy (4.9b) while maintaining feasibility in constraints (4.4d) and (4.5c). To do so, let the solution of the partial derivative in (4.9b) follow the following form,

$$\sum_{n \in \mathcal{N}} C(n) \frac{\partial \tilde{x}(n)}{\partial y_P^r} = \phi \sum_{n \in \mathcal{N}} \Big(C(f^r)\tilde{x}(n) + \frac{\beta \rho^r}{K} \Big), \quad S^r \in \sigma \tag{4.17}$$

where K is the maximum number of NF instances in a service request (including both best-effort and mandatory NF instances), i.e., $\sum_{n \in \mathcal{N}} 1 \leq K$. Similarly, we can satisfy (4.9b) by imposing a condition that $\sum_{n \in \mathcal{N}} C(f^r)\tilde{x}(n) \leq \beta \rho^r$. Now, we need to solve differential equation (4.17). A similar procedure yields a cost function that is expressed as

$$\tilde{x}(n) = \frac{1}{K}(-Ce^{\phi \frac{\sum_{S^r \in \sigma | n \in P \in \mathcal{P}(r)} C(f^r) y_P^r}{C(n)}} - 1), \quad n \in \mathcal{N}.$$

Initially, before the arrival of any service request, we require that $\tilde{x}(n) = 0$, which occurs when $\frac{1}{C(n)} \sum_{S^r \in \sigma} C(f^r) y_P^r = 0$. Therefore, we have $C = -1$. We also require that $\tilde{x}(n) \geq \beta\eta^r$ when NFV node n ($\in \mathcal{N}$) is saturated, i.e. when $\frac{1}{C(n)} \sum_{S^r \in \sigma} C(f^r) y_P^r = 1$. Therefore, we need $\phi \geq \ln(\beta K \frac{\eta_{\max}}{\eta_{\min}} + 1)$. Hence, $\tilde{x}(n)$ can be expressed as

$$\tilde{x}(n) = \frac{1}{K}\left(e^{\phi \frac{\sum_{S^r \in \sigma | n \in P \in \mathcal{P}(r)} C(f^r) y_P^r}{C(n)}} - 1\right), \quad n \in \mathcal{N}. \tag{4.18}$$

Note that constraint (4.4d) is maintained feasible over the whole range of $\tilde{x}(n)$. Similar to the transmission resource, due to Lemma 2 (to be given in Subsection 4.4.3), if the required processing rate is bounded above by $C(f^r) \leq \frac{\min_{n \in \mathcal{N}} C(n)}{\phi}$, we need $\phi \geq \ln(2\beta K \frac{\eta_{\max}}{\eta_{\min}} + 2)$ to ensure a sufficient protection against violating the processing resources. Since we set $\sum_{n \in \mathcal{N}} C(f^r) \tilde{x}(n) \leq \beta\rho^r$, we require that $z^r \geq \max\{\alpha\varrho^r - \sum_{l \in \mathcal{L}} d^r \bar{x}(l), \beta\rho^r - \sum_{n \in \mathcal{N}} C(f^r) \tilde{x}(n)\}$ for (4.5b) and (4.5c) to be maintained feasible, and for (4.9c) to hold. The cost function can be updated multiplicatively each time after embedding a service request as follows,

$$\tilde{x}^r(n) = \tilde{x}^{r-1}(n) e^{\phi \frac{C(f^r)}{C(n)}} + \frac{1}{K}\left(e^{\phi \frac{C(f^r)}{C(n)}} - 1\right), \quad n \in \mathcal{N}, S^r \in \sigma$$

where $\tilde{x}^r(n)$ is the cost of the NFV node n ($\in \mathcal{N}$) after embedding the rth service request, and $\tilde{x}^0(n)$ is set to zero. Now, we are ready to state the all-or-nothing/all-or-something admission mechanism.

4.4.2 Admission Mechanism

The procedure commences with the arrival of an rth service request, which resembles an augmentation of a new decision variable (y_P^r) to the dual formulation. Correspondingly, in the primal formulation, the arrival is equivalent to the augmentation of two new constraints, namely (4.5b) and (4.5c). The rth service request is accepted if there exists a path, P[1], such that the following two conditions hold:

$$\sum_{l \in P \cap \mathcal{L}} d^r \bar{x}^{r-1}(l) \leq \alpha\varrho^r \tag{4.19}$$

[1]How to find path P for the routing and NF placement problem is addressed in Subsection 4.5.

and

$$\sum_{n \in P \cap \mathcal{N}} C(f^r)\tilde{x}^{r-1}(n) \leq \beta\rho^r. \tag{4.20}$$

If the two conditions are satisfied, accept the request and route it on P, and set $y_P^r = 1$. To maintain feasibility in (4.5b) and (4.5c), set

$$z^r = \max\Big(\alpha\varrho^r - \sum_{l \in P \cap \mathcal{L}} d^r \bar{x}(l), \beta\rho^r - \sum_{l \in P \cap \mathcal{N}} C(f^r)\tilde{x}(n)\Big). \tag{4.21}$$

Finally, update the costs of the edge variables $\bar{x}(l)$ and NFV nodes $\tilde{x}(n)$ in a multiplicative manner as follows,

$$\bar{x}^r(l) = \bar{x}^{r-1}(l)e^{\varphi\frac{d^r}{B(l)}} + \frac{1}{L}(e^{\varphi\frac{d^r}{B(l)}} - 1), \quad l \in P \cap \mathcal{L} \tag{4.22}$$

$$\tilde{x}^r(n) = \tilde{x}^{r-1}(n)e^{\phi\frac{C(f^r)}{C(n)}} + \frac{1}{K}(e^{\phi\frac{C(f^r)}{C(n)}} - 1), \quad n \in P \cap \mathcal{N} \tag{4.23}$$

where $\varphi = \ln(2\alpha L|\mathcal{D}|^k_{max} + 2)$ and $\phi = \ln(2\beta K\frac{\eta_{\max}}{\eta_{\min}} + 2)$.

Treatment of best-effort NFs – We first find a routing and NF placement configuration that includes the set of best-effort NFs since it provides a larger (incentivized) profit with $\rho^r = C(f^r)\eta_b^r$. If such configuration is rejected by the admission mechanism, we find another configuration that excludes the set of best-effort NFs, and check against the admission mechanism with the nominal profit function $\rho^r = C(f^r)\eta_m^r$. If both configurations (with and without the set of best-effort NFs) do not satisfy (4.19) and (4.20), the service request is rejected. Fig. 4.1 provides a representation of the all-or-nothing/all-or-something admission mechanism. Algorithm 2 summarizes the online admission framework. In the algorithm's pseudocode, an assignment is denoted by "←".

4.4.3 Performance Analysis

In what follows, we analyze the performance of the admission mechanism. We show that the proposed mechanism does not violate the transmission and processing resources of physical links and NFV nodes. Then, we prove the competitive ratio for the all-or-nothing/all-or-something admission mechanism.

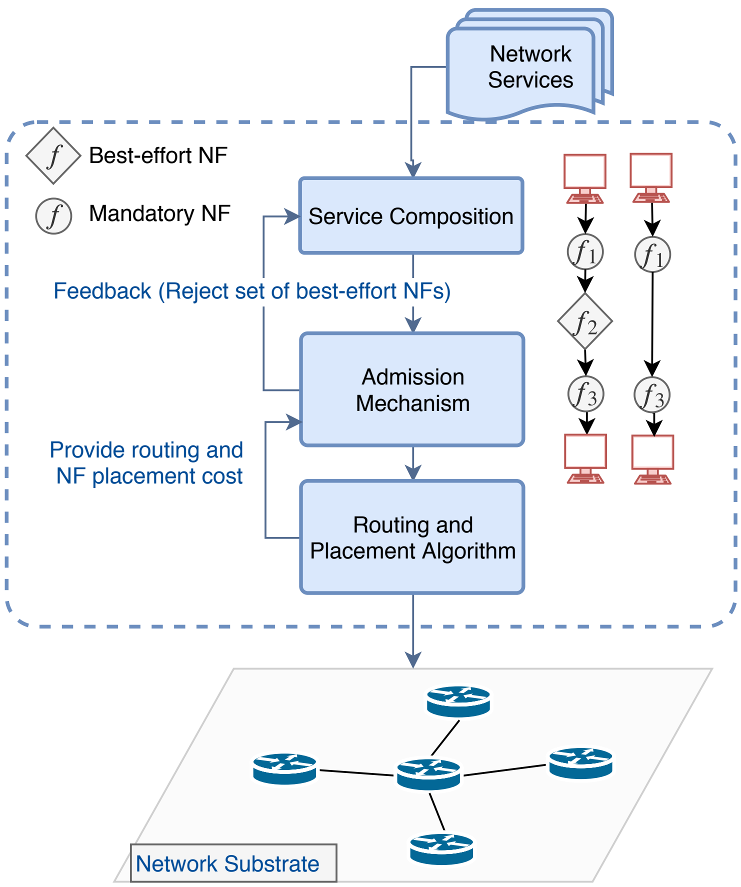

Figure 4.1: The joint all-or-nothing/all-or-something admission mechanism and online joint composition, routing, and NF placement framework.

Algorithm 2: Admission control and online joint composition, routing, and NF placement framework

Procedure OnlineJRP($\mathcal{G}, S^r$);
Input : $\mathcal{G}(\mathcal{N}, \mathcal{L})$, S^r
Output: Embed or reject S^r
$\bar{x}(l) \leftarrow 0$ (*only once*); $\tilde{x}(n) \leftarrow 0$ (*only once*);
services $\leftarrow$ (S^r, $S^r - \mathcal{V}_b^r$);
for $i = 1 : 2$ **do**
    $P \leftarrow$ unconstrainedJRP($\mathcal{G}$, services[i]);   ▷ JRP is in Algorithm 3 in Section 4.5.
    if $\sum_{l \in P \cap \mathcal{L}} d^r \bar{x}(l) \leq \alpha \varrho^r$ *and* $\sum_{n \in P \cap \mathcal{N}} C(f^r) \tilde{x}(n) \leq \beta \rho^r$ **then**
        Accept request ($y_P^r \leftarrow 1$);
        $z^r \leftarrow \max\left(\alpha \varrho^r - \sum_{l \in P \cap \mathcal{L}} d^r \bar{x}(l), \beta \rho^r - \sum_{n \in P \cap \mathcal{N}} C(f^r) \tilde{x}(n)\right)$;
        $\bar{x}(l) \leftarrow \bar{x}(l) e^{\varphi \frac{d^r}{B(l)}} + \frac{1}{L}(e^{\varphi \frac{d^r}{B(l)}} - 1), \quad \forall l \in P \cap \mathcal{L}$;
        $\tilde{x}(n) \leftarrow \tilde{x}(n) e^{\phi \frac{C(f^r)}{C(n)}} + \frac{1}{K}(e^{\phi \frac{C(f^r)}{C(n)}} - 1), \quad \forall n \in P \cap \mathcal{N}$;
        return;
    end
end
return (Reject request);

Lemma 1. *The transmission resource constraint on the physical links cannot be violated, i.e.,* $\sum_{S^r \in \sigma | l \in P \in \mathcal{P}(r)} d^r \leq B(l)$, $l \in \mathcal{L}$, *if* $\varphi \geq \ln(2\alpha L |\mathcal{D}|^k_{\max} + 2)$ *and* $d^r \leq \frac{\min_{l \in \mathcal{L}} B(l)}{\varphi}$, $\forall S^r \in \sigma$.

Proof. Assume that the transmission resources on physical link l ($\in \mathcal{L}$) become exceeded when the rth service request arrives. Then, we have $B(l) - \sum_{j=1}^{r-1} d^j < d^r$. Therefore, after admitting the $(r-1)$th service, the value of $\bar{x}^{r-1}(l)$ can be expressed as

$$\begin{aligned} \bar{x}^{r-1}(l) &= \frac{1}{L}(e^{\varphi \frac{\sum_{j=1}^{r-1} d^j}{B(l)}} - 1) \\ &= \frac{1}{L}(e^{\varphi (1 - \frac{B(l) - \sum_{j=1}^{r-1} d^j}{B(l)})} - 1) \\ &> \frac{1}{L}(e^{\varphi (1 - \frac{d^r}{B(l)})} - 1). \end{aligned} \tag{4.24}$$

Assuming that the required data rate of a service request is "small enough", i.e. $d^r \leq \frac{\min_{l \in \mathcal{L}} B(l)}{\varphi}$, $\forall S^r \in \sigma$. Then, inequality (4.24) becomes

$$\begin{aligned} \bar{x}^{r-1}(l) &\geq \frac{1}{L}(e^{\varphi (1 - \frac{1}{\varphi})} - 1) \\ &= \frac{1}{L}(\frac{e^{\varphi}}{e} - 1). \end{aligned} \tag{4.25}$$

Therefore, for the rth service request to be rejected, we need $d^r \bar{x}^{r-1}(l) \geq \alpha \varrho^r$, $\forall S^r \in \sigma$, which translates to $\bar{x}^{r-1}(l) \geq \alpha |\mathcal{D}|^k_{\max}$. Therefore, we need $\frac{1}{L}(\frac{e^{\varphi}}{e} - 1) \geq \alpha |\mathcal{D}|^k_{\max}$, which entails that $\varphi \geq \ln(2\alpha L |\mathcal{D}|^k_{\max} + 2)$. That is, φ is set such that the admission mechanism rejects any service request that would violate the transmission resources of a physical link. $\square$

Lemma 2. *The processing resource constraint on the NFV nodes cannot be violated, i.e.* $\sum_{S^r \in \sigma | n \in P \in \mathcal{P}(r)} C(f^r) \leq C(n)$, $n \in \mathcal{N}$, *if* $\phi \geq \ln(2\beta K \frac{\eta_{\max}}{\eta_{\min}} + 2)$ *and* $C(f^r) \leq \frac{\min_{n \in \mathcal{N}} C(n)}{\phi}$, $\forall S^r \in \sigma$.

Proof. Assuming that the processing resources on the NFV node n ($\in \mathcal{N}$) become exceeded when request S^r is accepted, we have $C(n) - \sum_{j=1}^{r-1} \sum_{f \in \mathcal{V}^j} C(f^j) < C(n)$. Therefore, the value of $\tilde{x}^{r-1}(n)$ can be expressed as

$$\tilde{x}^{r-1}(n) = \frac{1}{K}(e^{\phi \frac{\sum_{j=1}^{r-1} \sum_{f \in \mathcal{V}^j} C(f^j)}{C(n)}} - 1)$$

$$
\begin{aligned}
&= \frac{1}{K}(e^{\phi(1-\frac{C(n)-\sum_{j=1}^{r-1}\sum_{f\in\mathcal{V}^j} C(f^j)}{C(n)})} - 1) \\
&> \frac{1}{K}(e^{\phi(1-\frac{\sum_{f\in\mathcal{V}^r} C(f^r)}{C(n)})} - 1).
\end{aligned} \tag{4.26}
$$

Under the assumption that the processing requirements of a service request is "small enough", i.e., $C(f^r) \leq \frac{\min_{n\in\mathcal{N}} C(n)}{\phi}$, $S^r \in \sigma$, inequality (4.26) becomes

$$
\begin{aligned}
\tilde{x}^{r-1}(l) &\geq \frac{1}{K}(e^{\phi(1-\frac{1}{\phi})} - 1) \\
&= \frac{1}{K}(\frac{e^{\phi}}{e} - 1).
\end{aligned} \tag{4.27}
$$

Therefore, for the rth service to be rejected, we need $\sum_{n\in\mathcal{N}} C(f^r)\tilde{x}(n) \geq \beta\rho^r, \forall S^r \in \sigma$, which translates to $\frac{1}{K}(e^{\phi}/e - 1) \geq \beta\frac{\eta_{\max}}{\eta_{\min}}$, i.e., $\phi \geq \ln(2\beta K\frac{\eta_{\max}}{\eta_{\min}} + 2)$. That is, ϕ is set such that the admission mechanism rejects any request that would violate the processing resources of an NFV node. $\square$

Theorem 1. *The competitive ratio of the admission mechanism is $\mathcal{O}\Big(\max(\varphi, \phi)\Big)$, where $d^r \leq \frac{\min_{l\in\mathcal{L}} B(l)}{\varphi}$, $C(f^r) \leq \frac{\min_{n\in\mathcal{N}} C(n)}{\phi}$, $\varphi = \ln(2\alpha L|\mathcal{D}|^k_{\max} + 2)$, and $\phi = \ln(2\beta K\frac{\eta_{\max}}{\eta_{\min}} + 2)$.*

Proof. Let ΔA and ΔD be the change in the primal and dual cost in each iteration, respectively. Starting with $A = D = 0$, when the rth service request is accepted, the objective function of the dual formulation is increased by $\Delta D = \alpha\varrho^r + \beta\rho^r$. The objective function of the primal is increased by

$$
\Delta A = \sum_{l\in\mathcal{L}} B(l)\Big(\bar{x}^r(l) - \bar{x}^{r-1}(l)\Big) + \sum_{n\in\mathcal{N}} C(n)\Big(\tilde{x}^r(n) - \tilde{x}^{r-1}(n)\Big) + z^r. \tag{4.28}
$$

Substituting (4.22) and (4.23) in (4.28), we obtain

$$
\begin{aligned}
\Delta A = &\sum_{l\in\mathcal{L}}(e^{\varphi\frac{d^r}{B(l)}} - 1)(\bar{x}^{r-1}(l) + {}^1\!/\!{}_L)B(l) \\
&+ \sum_{n\in\mathcal{N}}(e^{\phi\frac{C(f^r)}{C(n)}} - 1)(\tilde{x}^{r-1}(n) + {}^1\!/\!{}_K)C(n) + z^r.
\end{aligned} \tag{4.29}
$$

Using inequality $e^x - 1 \leq x$ for $0 \leq x \leq 1$, we get

$$
\Delta A \leq \varphi\sum_{l\in\mathcal{L}} d^r(\bar{x}^{r-1}(l) + {}^1\!/\!{}_L) + \phi\sum_{n\in\mathcal{N}} C(f^r)(\tilde{x}^{r-1}(n) + {}^1\!/\!{}_K) + z^r. \tag{4.30}
$$

From the admission mechanism, substituting z^r from (4.21) in (4.30), we obtain

$$\begin{aligned}\Delta A =&\varphi \sum_{l\in\mathcal{L}} d^r(\bar{x}^{r-1}(l) + {}^1\!/\!_L) + \phi \sum_{n\in\mathcal{N}} C(f^r)(\tilde{x}^{r-1}(n) + {}^1\!/\!_K)\\ &+\max\Big(\alpha\varrho^r - \sum_{l\in P\cap\mathcal{L}} d^r\bar{x}(l), \beta\rho^r - \sum_{n\in P\cap\mathcal{N}} C(f^r)\tilde{x}(n)\Big) \qquad (4.31)\\ &\leq\varphi\sum_{l\in\mathcal{L}} d^r(\bar{x}^{r-1}(l) + \frac{\alpha\varrho^r}{L}) + \phi\sum_{n\in\mathcal{N}} C(f^r)(\tilde{x}^{r-1}(n) + \frac{\beta\rho^r}{K})\\ &+\alpha\varrho^r + \beta\rho^r - \sum_{l\in P\cap\mathcal{L}} d^r\bar{x}(l) - \sum_{n\in P\cap\mathcal{N}} C(f^r)\tilde{x}(n) \qquad (4.32)\\ &=(\varphi-1)\sum_{l\in\mathcal{L}} d^r\bar{x}^{r-1}(l) + \alpha\varrho^r + \beta\rho^r + (\phi-1)\\ &\times\sum_{n\in\mathcal{N}} C(f^r)\tilde{x}^{r-1}(n) + \alpha\varphi\varrho^r\sum_{l\in\mathcal{L}}\frac{d^r}{L} + \beta\phi\rho^r\frac{1}{K}\sum_{n\in\mathcal{N}} C(f^r) \qquad (4.33)\\ &\leq\alpha\varrho^r + \beta\rho^r + (\varphi-1)\sum_{l\in\mathcal{L}} d^r\bar{x}^{r-1}(l)\\ &+(\phi-1)\sum_{n\in\mathcal{N}} C(f^r)\tilde{x}^{r-1}(n) + \alpha\varrho^r\varphi + \beta\rho^r\phi. \qquad (4.34)\end{aligned}$$

Since the rth service request is accepted, i.e., $\sum_{l\in\mathcal{L}} d^r\bar{x}^{r-1}(l) \leq \alpha\varrho^r$ and $\sum_{n\in\mathcal{N}} C(f^r)\tilde{x}^{r-1}(n) \leq \beta\rho^r$, inequality (4.34) becomes

$$\begin{aligned}\Delta A \leq& 2\alpha\varrho^r\varphi + 2\beta\rho^r\phi \qquad (4.35)\\ \leq& 2(\alpha\varrho^r + \beta\rho^r)\max\{\varphi,\phi\}. \qquad (4.36)\end{aligned}$$

It is shown in Lemmata 1 and 2 that the online algorithm ensures that the transmission and processing resource constraints are always satisfied, where variables $\bar{x}(l)$, $\tilde{x}(n)$, and z^r are designed such that a feasible primal solution is maintained. Therefore, using weak duality (i.e., $\Delta D \leq \Delta A$) and from (4.36), a competitive performance of $\mathcal{O}\Big(\max\{\varphi,\phi\}\Big)$ is concluded.

□

Now, we discuss the method to find path P for a service request. Recall the admission conditions in (4.19) and (4.20). For each service request, the admission mechanism requires checking *all* possible paths for the performance guarantees to hold. If any of such paths satisfies the admission mechanism, the service request should be accepted. Notably, in general, it is not necessary to route the service on the minimum-cost path, but rather a secondary routing objective can be invoked. However, there exists an exponential number of possible paths for a service request. It is more convenient (and sufficient) to check against the minimum-cost path only. If it was rejected, all other paths would be rejected. Next, we propose an algorithm to find the minimum-cost routing and NF placement solution for a unicast and multicast service request.

4.5 Routing and NF Placement Approximation Algorithm

Due to Lemmata 1 and 2, the proposed admission mechanism guarantees that a violation in the processing and transmission resources is always avoided if the exponential cost functions are used for the physical links and NFV nodes with $\varphi \geq \ln(2\alpha L|\mathcal{D}|^k_{\max} + 2)$ and $\phi \geq \ln(2\beta K \frac{\eta_{\max}}{\eta_{\min}} + 2)$. Therefore, it is sufficient to design a routing and NF placement algorithm for the unconstrained (or uncapacitated) scenario. Here, we propose a *one-step* algorithm for the routing and NF placement of unicast and multicast services for the unconstrained scenario. The algorithm relies mainly on the construction of an auxiliary multilayer network transformation that has a one-to-one mapping from the NF placement and routing problem to an equivalent routing problem. This facilitates the use of existing (approximation) algorithms, such as the Dijkstra shortest path for the unicast scenario and MST-based Steiner tree for the multicast scenario.

4.5.1 Auxiliary Network Transformation and Routing and NF Placement Algorithm

To jointly consider the provisioning costs of both NF (processing) and virtual links (transmission), for each service request, we construct an auxiliary multilayer graph from the network substrate, in which the constructed edges represent either (i) the hosting of a virtual link or (ii) the processing of some NF type. For the sake of exposition, since the joint routing and NF placement algorithm treats each service request separately, we drop superscript r (which alludes to the rth service request) in this subsection.

Algorithm 3: Joint routing and NF placement algorithm for a single service request for the unconstrained scenario

<u>Procedure</u> unconstrainedJRP($\mathcal{G}, S^r$);
Input : $\mathcal{G}_M$, $S^r = S = (s, \mathcal{D}, f_1, f_2, \ldots, f_{|\mathcal{V}|}, d)$
Output: P
$\{\mathcal{G}^k(N^k, L^k)\}_{k=0}^{|\mathcal{V}|+1} \leftarrow \mathcal{G}(N, L)$;
$s^0 \leftarrow s$ (source node at layer 0 as source s);
$\mathcal{D}^{|\mathcal{V}|} \leftarrow \mathcal{D}$;
$\mathcal{L}_I \leftarrow \{\}$;
for $k = 0 : (|\mathcal{V}| - 1)$ **do**
  if $n^k \in \mathcal{F}_k$ **then**
    Add $l \leftarrow (n^k, n^{k+1})$ to $\mathcal{L}_I$;
    $C_l(r) = C(n^k)$;
  end
end
For multicast services, find an MST-based Steiner tree from s^0 to $\mathcal{D}^{|\mathcal{V}|}$, while utilizing the cost functions in (4.22) and (4.23), and save on P;
For unicast services, find a Dijkstra shortest path from s^0 to $t^{|\mathcal{V}|}$, while utilizing the cost functions in (4.22) and (4.23), and save on P;
return P;

The auxiliary multilayer graph is modeled as a directed graph $\mathcal{G}_M = (\mathcal{N}_M, \mathcal{L}_M)$, where $\mathcal{N}_M \subseteq \mathcal{N} \times \mathcal{X}$ is the set that contains all nodes, in which node n ($\in \mathcal{N}$) is present in a corresponding layer a ($\in \mathcal{X}$); denote such a node by n^α. Correspondingly, $\mathcal{L}_M \subseteq \mathcal{N}_M \times \mathcal{N}_M$ is the set of all inter- and intra-layer edges. Intra-layer edges, $\mathcal{L}_A = \{(u^a, v^b) \in \mathcal{L}_M | a = b \in \mathcal{X}\}$, represent the routing connections between the network elements (nodes) in each layer. Inter-layer edges, $\mathcal{L}_I = \{(u^a, u^b) \in \mathcal{L}_M | a \neq b \in \mathcal{X}\}$, are used to encode the placement decisions in which a traversal of an edge from one layer a ($\in \mathcal{X}$) to another layer b ($\in \mathcal{X}$) maps to the processing of the ath NF instance in a service request. Hence, for service request S^r, the number of layers is equivalent to the number of NFs plus one (i.e., $|\mathcal{X}| = |\mathcal{V}| + 1$).

Upon the arrival of the rth service request, we construct a multilayer graph $\mathcal{G}_M$ as follows:

1. Create $|\mathcal{X}|$ $(= |\mathcal{V}|+1)$ copies of the network substrate $\mathcal{G}$. Each copy $(\mathcal{G}^i(\mathcal{N}^i, \mathcal{L}^i))$ represents one layer, where $i = \{0, \dots, |\mathcal{V}|\}$. The transmission and processing resources for each NFV node $(n \in \mathcal{N})$ and physical link $(l \in \mathcal{L})$ are equal in each copy;
2. Assign the source node to its corresponding node at layer 0 $(= s^0,\ s^0 \in \mathcal{N}^0)$;
3. Assign the destination nodes to their corresponding nodes at the last layer $(= t^{|\mathcal{V}|},\ t^{|\mathcal{V}|} \in \mathcal{N}^{|\mathcal{V}|})$;
4. For the first $|\mathcal{V}|$ layers, construct inter-layer edges from layer i to layer $i+1$ $(l \leftarrow (n^i, n^{i+1}))$ for each NFV node that can host f_i (i.e., if $n^i \in \mathcal{F}_i$). The processing resources of each inter-layer edge, $l \leftarrow (n^i, n^{i+1})$, is that of the corresponding NFV node n^i.

We provide an illustrative example of the construction of the auxiliary graph transformation in Figs. 4.2 and 4.3. Fig. 4.2 illustrates a unicast service request of two NFs (f_1 and f_2), where the source is n_1 and the destination is n_4. We also have a network substrate of 4 NFV nodes that can host either NF type or both. Fig. 4.3 illustrates the construction of the auxiliary graph transformation. Since we have two NFs, the transformation has three layers. NFV nodes n_1 and n_3 can host f_1. Therefore, we construct the inter-layer edges, $n_1^1 \to n_1^2$ and $n_3^1 \to n_3^2$. Similarly, n_1 and n_2 can host f_2. Therefore, we construct the inter-layer edges, $n_1^2 \to n_1^3$ and $n_2^2 \to n_2^3$.

With the new auxiliary graph transformation, a path traversal (while considering only the edge costs) from node s^0 to node $t^{|\mathcal{V}|}$ represents a routing and NF placement solution in the original network substrate graph. Algorithm 3 summarizes the online joint routing and NF placement framework, which comprises the construction of the auxiliary graph transformation with the minimum-cost routing algorithm for a unicast or multicast service. For multicast services, the MST-based Steiner tree algorithm is a 2-approximation algorithm. Therefore, the competitive-ratio of the admission mechanism for the multicast services is $\mathcal{O}(2\max\{\varphi, \phi\}) = \mathcal{O}(\max\{\varphi, \phi\})$.

Theorem 2. *For unicast services, the overall time complexity of the online routing and NF placement framework is $\mathcal{O}(h \log h)$, where $h = |\mathcal{N}|(K+1)$. For multicast services, the overall time complexity is $\mathcal{O}\Big(|\mathcal{D}|^2 h\Big)$.*

Proof. For an efficient runtime, the full construction of the auxiliary network transformation can be performed once in the beginning. With the arrival of a new service request,

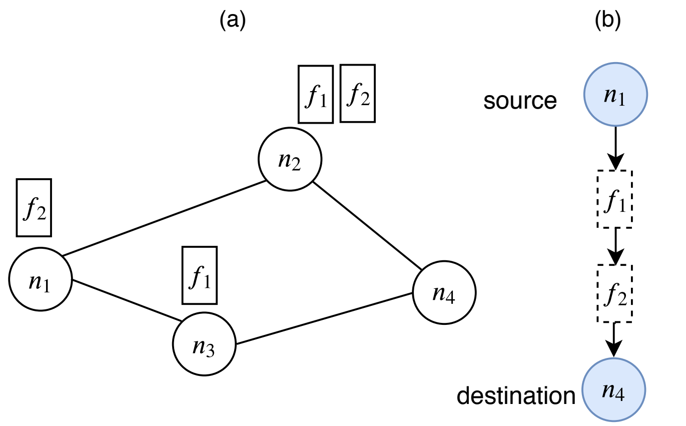

Figure 4.2: A problem input: (a) A network substrate along with the permissible NFs on each network element, and (b) the logical topology of a service request.

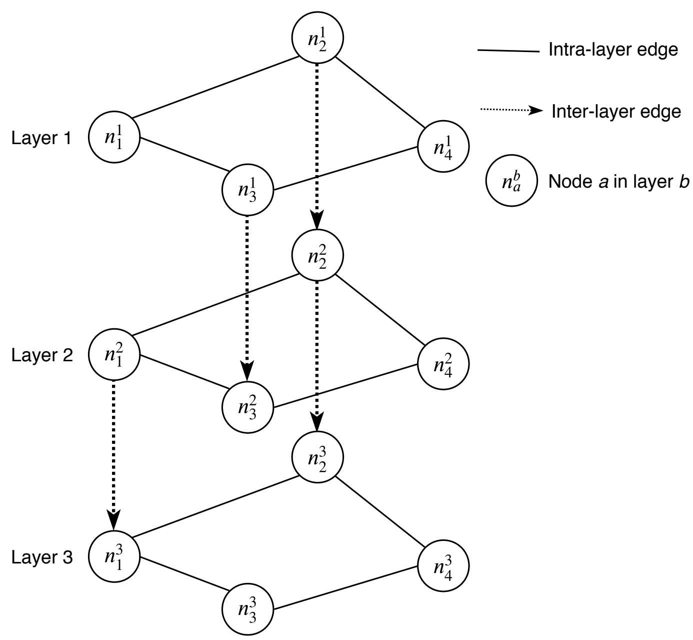

Figure 4.3: The auxiliary network transformation for the problem input in Fig. 4.2.

depending on the requested NFs, the relevant layers can be connected, while temporarily deactivating irrelevant layers (by disconnecting them). The auxiliary network transformation is constructed by creating $K+1$ copies of the original network substrate and connecting them. Therefore, the auxiliary transformation has $h = |\mathcal{N}|(K+1)$ nodes. The major component of the runtime is due to running the Dijkstra shortest path for unicast services and the MST-based Steiner tree for multicast services over the auxiliary transformation, which correspond to $\mathcal{O}(h \log h)$ and $\mathcal{O}\Big(|\mathcal{D}|^2 h\Big)$, respectively. $\square$

4.6 Discussions and Simulation Results

4.6.1 Discussions

On the obtained competitive ratio – As seen from the derivations in Subsection 4.4.1, to bound the performance of the primal and the dual, exponential cost functions are used for the physical links and NFV nodes. In doing so, if we can guarantee that the residual resources for the physical links and NFV nodes are not violated, the admission mechanism yields a competitive ratio of $\mathcal{O}\Big(\max\{\varphi, \phi\}\Big) = O\Big(\max\{\ln \alpha L |\mathcal{D}|^k_{\max}, \ln \beta K \frac{\eta_{\max}}{\eta_{\min}}\}\Big)$. However, the routing and NF placement problem for the constrained scenario is NP-hard. Therefore, to protect against a possible violation in the resources without relying on a constrained routing and NF placement algorithm, Lemmata 1 and 2 require that $\varphi = \ln(2\alpha L |\mathcal{D}|^k_{\max} + 2)$ and $\phi = \ln(2\beta K \frac{\eta_{\max}}{\eta_{\min}} + 2)$ at least, for which the competitive ratio is increased to $\mathcal{O}\Big(\max\{\ln 2\alpha L |\mathcal{D}|^k_{\max}, \ln 2\beta K \frac{\eta_{\max}}{\eta_{\min}}\}\Big)$. A consequential drawback is that the utilization of physical links and NFV nodes will not exceed $1 - \frac{1}{\varphi}$ and $1 - \frac{1}{\phi}$, respectively. Therefore, when φ or ϕ are relatively small, a considerable amount of processing and transmission resources will be wasted. Hence, the online algorithm is expected to not perform well for networks of a small size relative to the intended competitive performance.

On the design of profit function – The proposed framework applies for both unicast and multicast services. Moreover, it includes both best-effort and mandatory NF types. The profit functions allows a variation that depends on the maximum number of included destinations and the maximum incentive for including the set of best-effort NFs. To this effect, we can observe that the optimality is penalized due to the large variation of the profit functions, where maximizing the amortized throughput only (i.e., with $\varrho^r = d^r$ and

$\rho^r = C(f^r)$) improves the competitive ratio by a logarithmic factor in $|\mathcal{D}|^k_{\max}$ and $\frac{\eta_{\max}}{\eta_{\min}}$, respectively.

Recall that the derived competitive ratio is $\mathcal{O}\Big(\max\{\ln 2\alpha L|\mathcal{D}|^k_{\max}, \ln 2\beta K\frac{\eta_{\max}}{\eta_{\min}}\}\Big)$. In practice, $L|\mathcal{D}|^k_{\max}$ is larger than the maximum number of NFs (K). Therefore, the use of an incentive for including the best-effort NFs can help to scale up the second term without necessarily degrading the competitive performance, which offers an appropriate generalization.

4.6.2 Numerical Analysis

In this subsection, we analyze three online algorithms. The first algorithm is the proposed approximation algorithm with $\varphi = \ln(2\alpha L|\mathcal{D}|^k_{\max} + 2)$ and $\phi = \ln(2\beta K\frac{\eta_{\max}}{\eta_{\min}} + 2)$ (as in Algorithm 1). As shown, a resource violation is always avoided, and the competitive performance is guaranteed. The second online algorithm is similar to the first one but with $\varphi = \ln(\alpha L|\mathcal{D}|^k_{\max}+1)$ and $\phi = \ln(\beta K\frac{\eta_{\max}}{\eta_{\min}}+1)$. Here, resources are not necessarily protected from future violations. As a heuristic algorithm, if the routing and NF placement solution violates any processing or transmission constraint, it is removed. The third online algorithm is a greedy algorithm that attempts to accept all services as long as there are sufficient resources. The greedy algorithm resembles the heuristic algorithm but without invoking the admission conditions in (4.19) and (4.20). That is, it is the output of Algorithm 2 (without Algorithm 1), and with an extra step of checking if the service request violates any processing or transmission constraint.

In the experiments, we analyze the performance of the three algorithms on linear, random, and real network substrate topologies. Throughout the experiments, we set the scalarization coefficients (α, β) to unity (since the processing and transmission resources are appropriately scaled). Each NFV node can host 2/3 of the possible NFs in random. The transmission and processing resources are randomly distributed between 1000 and 5000 packet/s. The required data rate for service requests is uniformly distributed between 1 and 20 packet/s. The processing rate requirement of NF instances are linearly proportional to the incoming data rate $C(f^r) = d^r$ [30]. In all trials, we terminate an algorithm when it no longer can accept any request, i.e., when the network substrate instance reaches the maximum possible utilization.

In the first experiment, we generate a directed network substrate with a linear topology

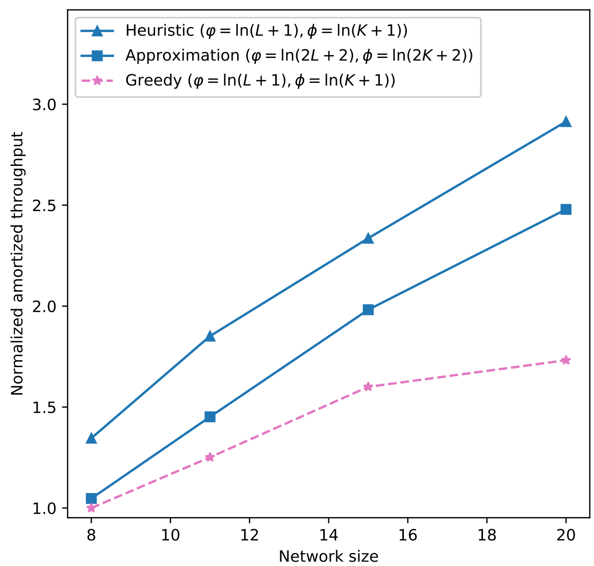

Figure 4.4: Normalized aggregate throughput ($\varrho^r + \rho^r$) for the three algorithms for a linear topology, with $L = K = 4$.

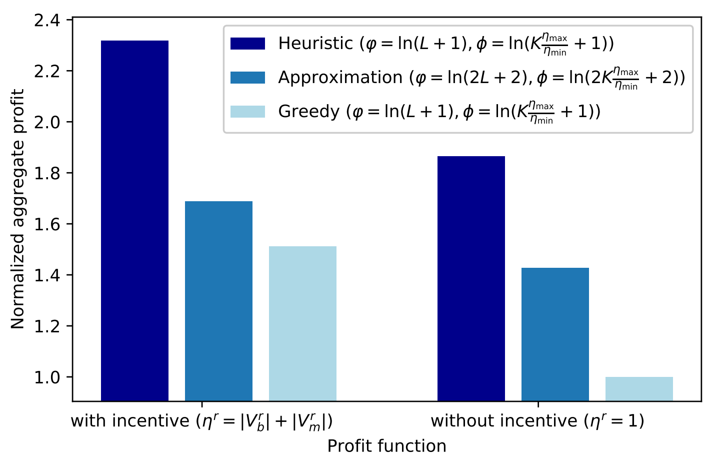

Figure 4.5: Normalized aggregate profit ($\varrho^r + \rho^r$) for three algorithms for a linear topology with and without incentivizing the use of best-effort NFs, with $L = 4$ and $K = 3$.

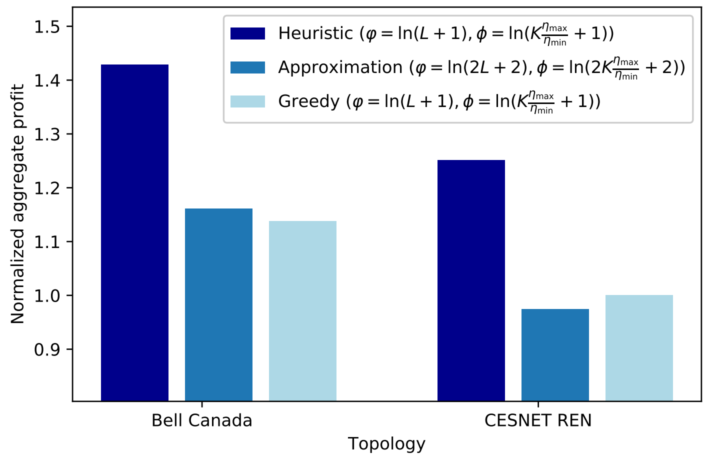

Figure 4.6: Normalized aggregate profit $(\varrho^r + \rho^r)$ for the three algorithms for two real topologies, namely Bell Canada and CESNET REN, with $K = 5$ and $L = 13$ and 6, respectively.

for unicast service requests. Each request has a random pair of source and destination nodes, and an overall number of required NFs ($|\mathcal{V}^r|$) of 3, with the number of best-effort NFs ($|\mathcal{V}_b^r|$) uniformly distributed between 0 and 3. The incentive for including the best-effort NFs (η^r) is set to unity. Fig. 4.4 shows the normalized aggregate profit for the three algorithms as the size of the linear network substrate ($|\mathcal{N}|$) grows, with $L = K = 4$. The aggregate profit increases almost linearly for all the algorithms. The heuristic algorithm (with $\varphi = \ln(L+1)$ and $\phi = \ln(K+1)$) outperforms the approximation algorithm (with $\varphi = \ln(2L+2)$ and $\phi = \ln(2K+2)$) by a constant gap of approximately 30%. Interestingly, this is equal to the wasted utilization by the latter algorithm, which is given by $1 - \frac{1}{\varphi}$ and $1 - \frac{1}{\phi}$ for the transmission and processing resources, respectively. The heuristic algorithm outperforms the greedy algorithm by almost 40%. When the network size is small, with $|\mathcal{N}| = 8$, the performance of the approximation and greedy algorithms is very close. This is expected since φ and ϕ are not small compared to the size of the network. Moreover, the approximation algorithm wastes a potential utilization of 30%.

In the second experiment, we aim to observe the effect of the incentive for including the set of best-effort NFs (η^r) on the competitive performance. The experiment is performed over a linear topology with 20 nodes. We generate unicast service requests, each with an

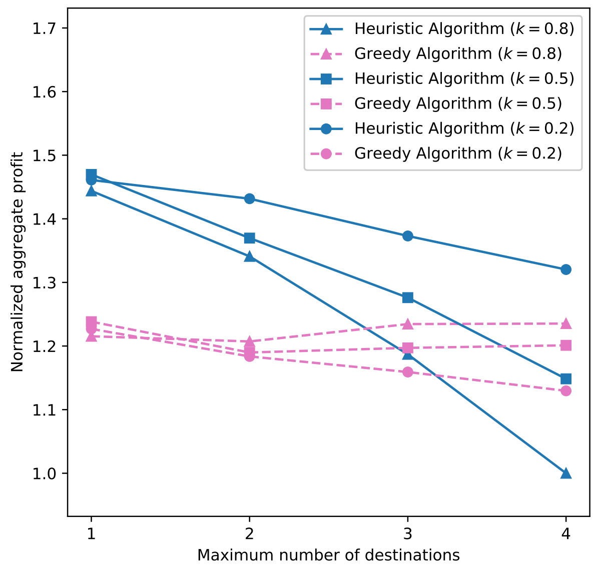

Figure 4.7: Normalized aggregate profit ($\varrho^r + \rho^r$) for the heuristic and greedy algorithms over random topology with $|\mathcal{N}| = 25$, $K = 4$, $\eta^r = 1$, and L is set to the maximum hop distance between any pair of nodes.

overall number of required NFs ($|\mathcal{V}^r|$) of 2, where either one or none of the NFs is set as best-effort in random. Here, we set η^r to the number of included NFs, i.e., $\eta_b = 2$ and $\eta_m = 1$. Fig. 4.5 shows the normalized aggregate profit for the three algorithms. As expected, when an incentive is used for including the best-effort NFs, the aggregate profit is scaled up for all the algorithms, which implies service requests with best-effort NFs are encouraged to maximize the aggregate profit. However, this comes at the expense of an increased competitive ratio compared to the greedy algorithm. With an incentivized profit function, the percent increase of the approximation algorithm to the greedy algorithm is 11%, whereas the percent increase without using an incentive is 39%.

Next, we evaluate the three algorithms on two real topologies from the Topology Zoo dataset [99]. The first topology, namely Bell Canada, is a commercial topology with 48 nodes and 64 links. The second topology, namely CESNET, is a research and education network (REN) with 52 nodes and 63 links. We generate unicast service requests with

5 required NFs ($|\mathcal{V}^r|$). For each service request, the number of best-effort NFs ($|\mathcal{V}_b|$) is uniformly distributed between 1 and $|\mathcal{V}^r|$. In this experiment, the design goal is to penalize requests which would take unnecessarily long routes due to deploying NF instances that are far-away from the shortest path between the source and destination. Therefore, for the two topologies, we set L to the maximum shortest path between any pair of nodes, which corresponds to 13 and 6, respectively for each topology. Moreover, we set $K = 5$ and $\eta^r = 1$. Fig. 4.6 shows the normalized aggregate profit for the two topologies. The approximation and greedy algorithms have close performance, whereas the heuristic algorithm yields a 25% and 23% improvement for the two topologies.

The next experiment is to test the performance of the online algorithms on both multicast and unicast service requests over a random topology with 25 nodes ($|\mathcal{N}| = 25$). The random topology is generated using the Barabási–Albert preferential attachment model, which provides scale-free network topologies [100]. For each service request, the number of destinations varies randomly between 1 and 4, and the number of required NFs in each service request is uniformly random between 1 and 3. Recall that, to provide a non-discriminatory treatment between unicast and multicast service requests, $\varrho^r \propto |\mathcal{D}|^k$, where k is recommended to be 0.8. Fig. 4.7 shows the normalized aggregate profit for the online heuristic and greedy algorithms as $|\mathcal{D}|_{\max}$ grows for different values of k. The performance of the greedy algorithm remains almost constant as $|\mathcal{D}|^k_{\max}$ increases. However, the heuristic algorithm shows a downtrend as $|\mathcal{D}|^k_{\max}$ increases, especially for large k (e.g, $k = 0.8$). The experiment demonstrates that the competitive ratio of the online heuristic algorithm is increased due to the allowed variation in the profit function (as $|\mathcal{D}|^k_{\max}$ increases).

Chapter 5

Model-free Dynamic Provisioning of NFV-enabled Services

5.1 System Model

In this section, we present the system model for Problem III. Mainly, we generalize the previous system models to consider the time-varying traffic pattern of the service request and network substrate.

5.1.1 Network Functions and the Network Service

Depending on the operation semantics (which can be specified a priori), here we consider that some NF types can be deployed in a parallel or sequential manner in geo-distributed NFV nodes. We consider a unicast service with time-varying traffic demand, expressed as

$$S = (s, t, \mathcal{V}, d(\tau)), \qquad \tau \in (0, T] \tag{5.1}$$

where the source and destination nodes are s and t, respectively; $\mathcal{V} = \{f_1, f_2, \ldots, f_{|\mathcal{V}|}\}$ represents the set of NFs that need to be traversed in an ascending order for the source-destination pair; parameter $d(\tau)$, $\tau \in (0, T]$, denotes the required transmission rate at time τ in packet/s. Each NF requires an amount of processing resources of $C(f(\tau))$, $\tau \in (0, T]$, in packet/s.

5.1.2 Network Substrate

We are given a capacitated network substrate, $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ and $\mathcal{L}$ are the sets of nodes and links, respectively. Each physical link l ($\in \mathcal{L}$) has a residual transmission resource at time τ, $B_l(\tau)$, $\tau \in (0, T]$, in packet/s. Each node n ($\in \mathcal{N}$) has a residual processing resource, $C_n(\tau)$, $\tau \in (0, T]$, in packet/s. Here, $B_l(\tau)$ and $C_n(\tau)$ represent the residual resources while taking into account the background traffic and the embedded NF chain at time τ. Nodes can be either (i) switches that are capable of forwarding traffic only (with $C_n(\tau) = 0$), or NFV nodes (e.g., commodity servers) that are capable of both forwarding traffic and operating a set of NF instances. An NFV node is capable of provisioning a number of NF instances simultaneously as long as the available processing resources satisfy the deployed NF processing requirements.

5.2 Problem Statement and Formulation

5.2.1 Problem Statement

Given network substrate $\mathcal{G}$ and network service S with time-varying data rate requirement, we need to develop a dynamic joint composition, routing and NF placement framework to minimize the function, link, and routing provisioning costs. We consider a system that operates in a time-slotted fashion over a potentially large time span, where certain dynamic decisions are taken at each timestep. Due to the bursty and time-varying nature of the data rate, altering the routing and NF placement configuration at each timestep should be discouraged due to the cost of setting up new NF instances.

5.2.2 Problem Formulation

An embedded NFV-enabled network service can be modeled as a composition of several paths. Each path emanates from the source to the destination, and it traverses all the required NF instances. Let all the possible paths for a single unicast service S be given by $\mathcal{P}$. Let P ($\in \mathcal{P}$) be a path on the network substrate that is activated to (partially) host the service request.

Let $x_{li}^P(\tau) \in \{0,1\}$ be a decision variable at timestep of constant duration τ, where $x_{l0}^P(\tau) = 1$ indicates that link l is used to direct traffic from s to the P-th instance of the first NF (f_1^P) for path P ($\in \mathcal{P}$), and $x_{li}^P(\tau) = 1$ indicates that link l is used to direct traffic from f_i^P to f_{i+1}^P. Correspondingly, define $\kappa_{li}^P(\tau) \in [0,1]$ as a continuous flow variable that represents the fraction of flow used in link l to direct traffic from f_i^P to f_{i+1}^P such that

$$\begin{aligned}\kappa_{li}^P(\tau) =& d^P(\tau) x_{li}^P(\tau), \\ & \tau \in (0,T],\, P \in \mathcal{P},\, l \in \mathcal{L},\, i \in \Omega_0^{|\mathcal{V}|},\end{aligned} \tag{5.2}$$

where $d^P(\tau) \in [0,1]$ is a continuous decision variable that represents the fraction of flow assigned for path P, and Ω_m^n denotes the set of integers from m to n ($> m$), i.e., $\Omega_m^n \triangleq \{m, m+1, \ldots, n\}$. To meet the total required transmission rate, we impose constraint

$$\sum_{P \in \mathcal{P}} d^P(\tau) = 1, \quad \tau \in (0,T]. \tag{5.3}$$

Define binary decision variable $z_{ni}^P(\tau) \in \{0,1\}$, where $z_{ni}^P(\tau) = 1$ indicates that node n hosts f_i^P with an assigned fractional resources of $C(f_i^P(\tau)) \in [0,1]$. For each NF, to conserve the distributive processing resource, we impose

$$\sum_{P \in \mathcal{P}} C(f_i^P(\tau)) = C(f_i), \quad \tau \in (0,T],\, i \in \Omega_1^{|\mathcal{V}|}. \tag{5.4}$$

The overall data rate from all activated paths should not exceed the link transmission rate, $B_l(\tau)$, i.e.,

$$\sum_{P \in \mathcal{P}} \sum_{i=0}^{|\mathcal{V}|} \kappa_{li}^P(\tau) \le B_l(\tau), \quad l \in \mathcal{L},\, \tau \in (0,T]. \tag{5.5}$$

Moreover, the overall rate of the NF instances that are placed on an NFV node should not exceed the node processing rate $C_n(\tau)$, i.e.,

$$\sum_{i=1}^{|\mathcal{V}|} \sum_{P \in \mathcal{P}} C(f_i^P(\tau)) \le C_n(\tau), \quad n \in \mathcal{N},\, \tau \in (0,T]. \tag{5.6}$$

Objective function: Different logical service topologies and embedding configurations can incur different routing and NF setup costs. To route traffic that belongs to a service request, a switch needs to add one routing entry in the forwarding table that specifies the header and the next port (or physical link). Therefore, for the routing overhead, we consider that each physical link hosting the service request incurs an additional cost of η_1 [101]. Therefore, the routing cost is expressed as

$$c_1(\tau) = \sum_{i=0}^{|\mathcal{V}|} \sum_{l \in \mathcal{L}} \eta_1 x_{li}(\tau),\ \tau \in (0, T] \tag{5.7}$$

where $x_{li}(\tau) \in \{0, 1\}$ is a decision variable at time τ, where $x_{li}(\tau) = 1$ indicates that $x_{li}(\tau)^P = 1$ for some path $P\,(\in \mathcal{P})$, and

$$x_{li}^P(\tau) \leq x_{li}(\tau), \quad \tau \in (0, T],\ P \in \mathcal{P}. \tag{5.8}$$

The link and function provisioning costs can be expressed as

$$c_2(\tau) = \sum_{i=0}^{|\mathcal{V}|} \sum_{l \in \mathcal{L}} \sum_{P \in \mathcal{P}} d(\tau)\, \frac{\kappa_{li}^P(\tau)}{B_l(\tau)},\ \tau \in (0, T] \tag{5.9}$$

and

$$c_3(\tau) = \sum_{i=1}^{|\mathcal{V}|} \sum_{n \in \mathcal{N}} \sum_{P \in \mathcal{P}} d(\tau) \frac{C(f_i^P(\tau))}{C_n(\tau)} z_{ni}^P(\tau), \quad \tau \in (0, T] \tag{5.10}$$

respectively. We also consider the NF instance setup cost, which is incurred only when an NF instance is initialized at an NFV node. Let $z_{ni}(\tau) \in \{0, 1\}$ be a decision variable, where $z_{ni}(\tau) = 1$ indicates that function f_i is initialized on node n. Then, the NF setup cost can be expressed as

$$c_4(\tau) = \sum_{n \in \mathcal{N}} \sum_{i=1}^{|\mathcal{V}|} \eta_2 z_{ni}(\tau),\ \tau \in (0, T] \tag{5.11}$$

where η_2 is the NF setup cost which can include the installation cost and the respective signalling overhead from the controller to the node, and

$$\sum_{P \in \mathcal{P}} z_{ni}^P(\tau) - \sum_{P \in \mathcal{P}} z_{ni}^P(\tau - 1) \leq z_{ni}(\tau),\ \tau \in [0, T]. \tag{5.12}$$

In summary, the optimization problem for the dynamic joint composition, routing and NF placement problem with time-varying data rate can be expressed as

$$\text{minimize} \qquad \sum_{\tau=0}^{T} \sum_{k=1}^{4} \omega_k c_k(\tau) \tag{5.13a}$$

subject to:

$$\forall \tau \in (0,T],\, l \in \mathcal{L} : \sum_{i=0}^{|\mathcal{V}|} \sum_{P \in \mathcal{P}} \kappa_{li}^{P}(\tau) \leq B_l(\tau) \tag{5.13b}$$

$$\forall \tau \in (0,T],\, n \in \mathcal{N} : \sum_{i=1}^{|\mathcal{V}|} \sum_{P \in \mathcal{P}} C(f_i^P(\tau)) \leq C_n(\tau) \tag{5.13c}$$

$$(5.2), (5.3), (5.4), (5.8), (5.12) \tag{5.13d}$$

The problem in (5.13) is an online constrained problem since the data rate, $d(\tau)$, is revealed in an online manner, where the objective is to minimize the long-run provisioning cost of the time-varying service request. Knowledge of the traffic pattern of the service request and the background traffic is particularly crucial for the NF setup cost ($c_4(\tau)$).

5.3 Deep Reinforcement Learning Framework

5.3.1 Reinforcement Learning Background

We consider a standard RL setting consisting of an agent interacting with an environment in discrete timesteps, referred to as learning steps. A fully-observable environment is assumed. At each learning step τ, the agent observes a set of states, produces a set of actions to affect the states, and consequently receives a reward. Different from other learning paradigms (e.g., supervised learning), RL addresses a *sequential* decision making problem in a holistic manner, whereby an agent needs to find a desired behaviour (or policy) that maps the set of states to actions to maximize both immediate and *future* rewards. The environment can be stochastic. Formally, it is modeled as a Markov decision process with state space $\mathcal{S}$, action space $\mathcal{A}$, initial state distribution $\mathbb{P}(s_1)$, transition probabilities $\mathbb{P}(s_{\tau+1}|s_\tau, a_\tau)$, and reward function r_τ ($: \mathcal{S} \times \mathcal{A} \to \mathbb{R}$). Let the (discounted) accumulation of future rewards be the return ($\tilde{R}_\tau$), defined as

$$\tilde{R}_\tau = \sum_{i=\tau}^{\infty} \vartheta^{i-\tau} r_i \tag{5.14}$$

where $\vartheta \in (0,1]$ is a discount factor that represents the present value of future rewards. The larger ϑ, the more farsighted the agent is, i.e., the more valued future rewards are.

The agent aims to find a policy mapping that maximizes $\mathbb{E}_{s_\tau}[R_\tau|s_\tau]$, where $\mathbb{E}[\cdot]$ is the mathematical expectation. At learning step τ, the so-called action-value function (or Q-function) resembles the expected return after taking action a_τ while following policy λ, defined as $Q^\lambda(s_\tau, a_\tau) = \mathbb{E}[R_\tau|s_\tau, a_\tau]$. If policy λ is deterministic (i.e., $\lambda : \mathcal{S} \to \mathcal{A}$), the recursive Bellman equation can be used to learn the action-value function as follows,

$$Q^\lambda(s_\tau, a_\tau) = \mathbb{E}[r_\tau + \vartheta Q^\mu(s_{\tau+1}, \pi(s_{\tau+1}))]. \tag{5.15}$$

5.3.2 Pre-processing Stage

The described problem requires an end-to-end solution, whereby the design of service topology, placement of NFs, and routing configuration on the network substrate should be considered. Therefore, if approached naively, this problem can lead to an explosion in the number of states (and actions) for network substrates of a moderate size. A large number of empirical studies on the properties of real network substrates reveal that the average path length for a source-destination pair is very small. This is due to the observation that real networks tend to have a degree distribution with a power-law tail, which is known as the small-world phenomena [102, 103]. In such scale-free networks, the average path length is asymptotically logarithmic in the number of nodes of the network substrate (i.e., $= \mathcal{O}(\log |\mathcal{N}|)$). Therefore, in practice, the average number of edge-disjointed paths is small. Based on the aforementioned observations and to have a compact learning representation, we model an embedded service topology as a composition of several NF placement and routing configurations from the source to the destination. To generate the routing and NF placement configurations, we utilize the multilayer network substrate transformation from Subsection 4.5.1. Algorithm 4 summarizes the construction of the auxiliary transformation and the generation of routing and NF placement configurations.

We provide an illustrative example of constructing the auxiliary graph transformation and how service composition is incorporated in Figs. 5.1 and 5.2. Fig. 5.1 illustrates a service request of two NFs (f_1 and f_2), where the source is n_1 and the destination node is n_4. We also have a network substrate of 4 NFV nodes that can host either NF type or both. Fig. 5.2 illustrates the construction of the auxiliary graph transformation. Since we have two NFs, the auxiliary transformation has three layers. NFV nodes n_2 and n_3 can host f_1. Therefore, We construct the inter-layer edges, $n_2^1 \to n_2^2$ and $n_3^1 \to n_3^2$. Similarly, n_1 and n_2 can host f_2. Thus, we construct the inter-layer edges, $n_1^2 \to n_1^3$ and $n_2^2 \to n_2^3$.

Algorithm 4: Construction of multilayer network transformation and generation of several NF placement and routing configurations.

Procedure multipleJRP($\mathcal{G}, S$);
Input : $\mathcal{G}$, $S = (s, \mathcal{D}, f_1, f_2, \dots, f_{|\mathcal{V}|}, d)$
Output: $\mathcal{P}$
▷ Construction of multilayer transformation $\mathcal{G}_M$
$\{\mathcal{G}^k(N^k, L^k)\}_{k=0}^{|\mathcal{V}|+1} \leftarrow \mathcal{G}(N, L)$;
$s^0 \leftarrow s$ (source node at layer 0 as source s);
$\mathcal{D}^{|\mathcal{V}|} \leftarrow \mathcal{D}$;
$\mathcal{L}_I \leftarrow \{\}$;
for $k = 0 : (|\mathcal{V}| - 1)$ **do**
  for $n^k \in \mathcal{F}_k$ **do**
    Add $l \leftarrow (n^k, n^{k+1})$ to $\mathcal{L}_I$;
    $C(l) = C(n^k)$;
  end
end
▷ Populating $\mathcal{P}$ with virtual segment-disjoint configurations
$\mathcal{P} \leftarrow \{\}$;
for $i = 1 : \mathcal{V}$ **do**
  $\mathcal{G}_M^{\text{tmp}} \leftarrow \mathcal{G}_M$;
  while $\exists$ *ShortestPath(*$\mathcal{G}_M^{\text{tmp}}$*;* s^0*,* $t^{|\mathcal{V}+1|}$*)* **do**
    $\mathcal{P} \xleftarrow{+}$ ShortestPath($\mathcal{G}_M^{\text{tmp}}$; s^0, $t^{|\mathcal{V}+1|}$);
    Remove path for virtual segment f_i–f_{i+1} from $\mathcal{G}_M^{\text{tmp}}$;
  end
end
Remove replicated path (or tree) configurations;
return $\mathcal{P}$;

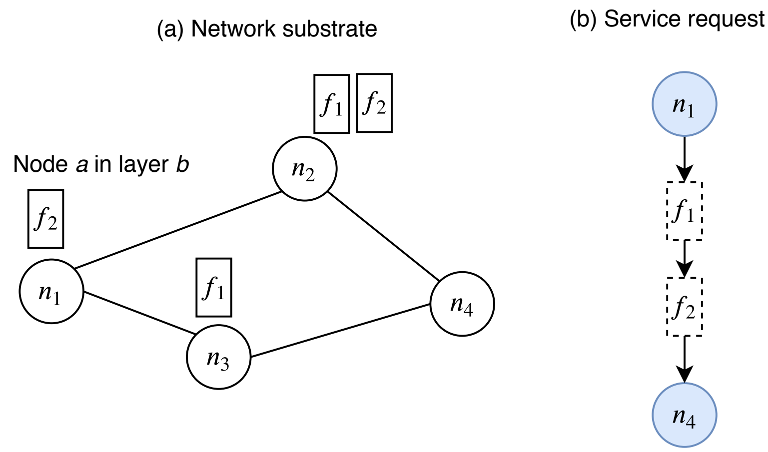

Figure 5.1: A problem input: (a) A network substrate along with the permissible NFs on each network element, and (b) the logical topology of a multicast service request.

With the auxiliary graph transformation, a path traversal (while considering only the edge costs) from the source to the destination represents a routing and NF placement solution for the service in the original network substrate graph. Here, we have an illustration of two different path traversals in Figs. 5.2-(a) and 5.2-(b). Let a path in the network substrate from f_i to f_{i+1} be called a virtual segment. Using the multilayer transformation, we find all the virtual segment-disjoint paths as shown in lines 13-22 in Algorithm 4. The intuition is that combining several paths yields a variety of service topologies and embedding configurations. The desired behavior is to have a dynamic service topology that changes depending on the traffic pattern of the service request in accordance with the objective function in (5.13a). That is, the service topology should grow in size and shrink (i.e., activate and tear down temporary NF instances) depending on the requested data rate and the background traffic in the network substrate while taking the model-free traffic patterns into account. For example, activating both paths in Figs. 5.2-(a), 5.2-(b) results in the service topology in Fig. 5.2-(c).

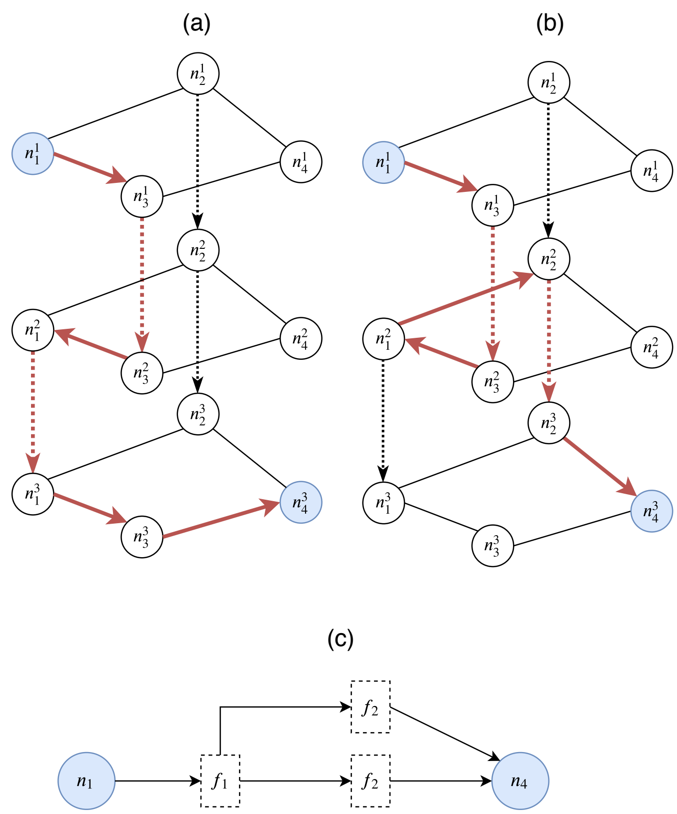

Figure 5.2: The auxiliary network transformation for the problem input in Fig. 5.1 with a different path traversals from source to destinations in (a) and (b). Activating both trees results in the logical topology shown in (c).

5.3.3 States, Actions, and Reward Function

Critical to the success of any RL framework is the design of an accurate yet compact state space and action space that capture the relevant features in the environment and accurate reward structure of the problem. In what follows, we describe the state space ($\mathcal{S}$), action space ($\mathcal{A}$), and the reward function.

Action space: Given the output of Algorithm 4, the action space is to choose which paths to activate and what the assigned proportion for each activated path from $\mathcal{P}$, i.e., $\boldsymbol{a}_\tau = \{d^P(\tau)\}_{P\in\mathcal{P}}$, where $\sum_{P\in\mathcal{P}} d^P(\tau) = 1$ and $\boldsymbol{a}_\tau \in \mathcal{A}$.

State Space: At the beginning of timestep τ, the agent should proactively adjust the service topology and the embedding solution of the service by relying on the available states. Therefore, the state space is comprised of:

1. the data rate of the previous timestep ($d(\tau-1)$);
2. the predicted data rate for the current timestep ($\hat{d}(\tau)$);
3. the utilization ratio of the network elements of all concerned paths in the previous timestep, i.e.,
$$\sum_{P\in\mathcal{P}} \frac{d^{P|l\in P}(\tau-1)}{B_l(\tau-1)}, \forall l \in \mathcal{L} \tag{5.16}$$
and
$$\sum_{P\in\mathcal{P}} \sum_{i=1}^{\mathcal{V}} \frac{C(f_i^{P|n\in P}(\tau-1))}{C_n(\tau-1)}, \forall n \in \mathcal{N}; \tag{5.17}$$
4. the path allocation of the previous timestep ($\boldsymbol{a}_{\tau-1}$).

Reward Function: The reward is to first maximize the additive inverse of the cost function in (5.13a). Second, to incorporate the constraints in (5.13b) and (5.13c), we penalize the reward function when a resource violation occurs in the physical links or NFV nodes. That is,
$$r_\tau = -(\sum_{k=1}^{4} \omega_k c_k(\tau) + \sum_{l\in\mathcal{L}} \psi'_l + \sum_{n\in\mathcal{N}} \psi'_n), \quad \tau \in [0, T] \tag{5.18}$$
where $\psi'_l \in \{0, \psi_l\}$ and $\psi'_n \in \{0, \psi_n\}$ are penalty factors, such that $\psi'_l = \psi_l$ and $\psi'_n = \psi_n$ are associated with violating constraints (5.13b) and (5.13c), respectively.

The reward structure exhibits strongly conflicting objectives, which can hinder an efficient exploration in the learning algorithm. Moreover, the rewards have varying degrees of sparsity. For example, the reward due to the routing cost (5.7) is only observed when a path activates/deactivates, which occur when $d^P(\tau) \in \{0,1\}$, $P \in \mathcal{P}$. A more sparse event is when a new NF instance is initialized at an NFV node. Therefore, we have a reward structure that is sparse and more sensitive to certain changes in the action space. One potential approach to tackle the challenges is to perform reward engineering, i.e., to smooth and shape the reward function. In our case, shaping the reward can further hinder the exploration. Moreover, a shaped reward does not accurately reflect the intended behavior, e.g., a smooth penalty for the resource violation can push the utilization ratio of the physical links and NFV nodes to certain inadvertent levels. Therefore, to tackle the aformentioned challenges, we propose a model-assisted deep RL algorithm as discussed in the following.

5.3.4 Deep RL Algorithm

For RL problems with a large state space, the use of non-linear function approximators is needed. Prior to the work of Mnih et al. [104], the use of deep neural networks (or generally non-linear functions) as parameterized function approximators for learning the action-value function was avoided due to the instability in the learning process. Mnih et al. proposed the Deep Q-network (DQN), which can learn the action-value function with large state space and small discrete action space. This success was achieved with two techniques, namely through (i) the use of an off-policy replay buffer to break the correlations between the sequential samples in the learning process, and (ii) the use of an earlier delayed replica of the primary Q-network (called the target network) to stabilize the learning process [104].

Here, our proposed problem is online control with a continuous action space. In continuous action spaces, we need to find a policy that optimizes the action space at every timestep through an iterative optimization process. Discrete learning algorithms, such as Q-learning and DQN, cannot be applied directly. The DQN is suitable for problems with a high-dimensional state space, yet it can only handle a discrete and low-dimensional action space. In our context, a discretization of the action space is not practical as the action space can grow exponentially for problems of moderate size due to the sensitivity of the reward structure. Recently, Lillicrap et al. adapt the DQN to the continuous action domain through the use of an off-policy actor-critic architecture with a DDPG learning algorithm [105].

Consider an approximate action-value function parameterized by $\boldsymbol{\theta}^Q$ $(Q(\boldsymbol{s}_\tau, \boldsymbol{a}_\tau|\boldsymbol{\theta}^Q))$. Such function can be optimized by minimizing the loss function,

$$L(\boldsymbol{\theta}^Q) = \mathbb{E}[(Q(\boldsymbol{s}_\tau, \boldsymbol{a}_\tau|\boldsymbol{\theta}^Q) - y_\tau)^2], \tag{5.19}$$

where $y_\tau = r_\tau + \vartheta Q(\boldsymbol{s}_{\tau+1}, \boldsymbol{a}_{\tau+1}|\boldsymbol{\theta}^Q)$.

The DDPG algorithm maintains a parameterized actor policy $(\mu(\boldsymbol{s}|\boldsymbol{\theta}^\mu))$ and a parameterized critic $(Q(\boldsymbol{s}_\tau, \boldsymbol{a}_\tau|\boldsymbol{\theta}^Q))$ in a primary network. The actor policy maps the states to a continuous action space. The parameterized critic function is learned using the Bellman equation as in (5.19). The actor policy is updated by optimizing the expected return from the start through the chain rule [105],

$$\begin{aligned}\nabla_{\boldsymbol{\theta}^\mu} H &\approx \mathbb{E}\Big[\nabla_{\boldsymbol{\theta}^\mu} Q(\boldsymbol{s}, \boldsymbol{a})|\boldsymbol{\theta}^Q)|_{s=s_\tau, a=\mu(s_\tau|\boldsymbol{\theta}^\mu)}\Big] \\ &= \mathbb{E}\Big[\nabla_{\boldsymbol{a}} Q(\boldsymbol{s}, \boldsymbol{a})|\boldsymbol{\theta}^Q))|_{s=s_\tau, a=\mu(s_\tau)} \nabla_{\boldsymbol{\theta}^\mu} \mu(\boldsymbol{s}|\boldsymbol{\theta}^\mu)|_{s=s_\tau}\Big].\end{aligned} \tag{5.20}$$

For stability, the DDPG algorithm maintains a target network which is a replica of the primary network with actor policy $\mu'(\boldsymbol{s}|\boldsymbol{\theta}^{\mu'})$ and critic $Q'(\boldsymbol{s}_\tau, \boldsymbol{a}_\tau|\boldsymbol{\theta}^{Q'})$. The weights of the target network are set to slowly track the primary network's weights such that $\theta^{Q'} = \upsilon\theta^Q + (1-\upsilon)\theta^{Q'}$, and $\theta^{\mu'} = \upsilon\theta^\mu + (1-\upsilon)\theta^{\mu'}$, where $\upsilon \ll 1$. Parameter υ represents a tradeoff between the stability and the rate of the learning process. A very small υ greatly stabilizes the learning process at the expense of slowing the learning process.

Model-assisted exploration – The exploration procedure can be treated independently from the learning algorithm because the DDPG is an off-policy algorithm. In the vanilla DDPG algorithm [105], a perturbed exploration policy, $\tilde{\mu}(\boldsymbol{s}_\tau)$, is constructed by adding a temporally correlated noise process directly to the action space, i.e., $\tilde{\mu}(\boldsymbol{s}_\tau) = \mu(\boldsymbol{s}_\tau) + \boldsymbol{n}$, where $\boldsymbol{n} \sim OU(0, \sigma^2)$ is the Ornstein-Uhlenbeck process. In each episode, the exploration process will produce a different action for the same state since the (correlated) noise is independent of the current state, $\boldsymbol{s}_\tau$. Such behavior is detrimental to our problem as some events are very sensitive to the changes in the action space such as in (5.7) and (5.11). A properly structured, state-dependent exploration methodology is to inject some noise directly to the parameters of the actor's deep neural network at the beginning of an episode, such that $\tilde{\theta}^{\tilde{\mu}} = \theta^\mu + n$, where $n \sim \mathcal{N}(0, \sigma_n^2)$ and $\tilde{\theta}^{\tilde{\mu}}$ is the perturbed actor policy

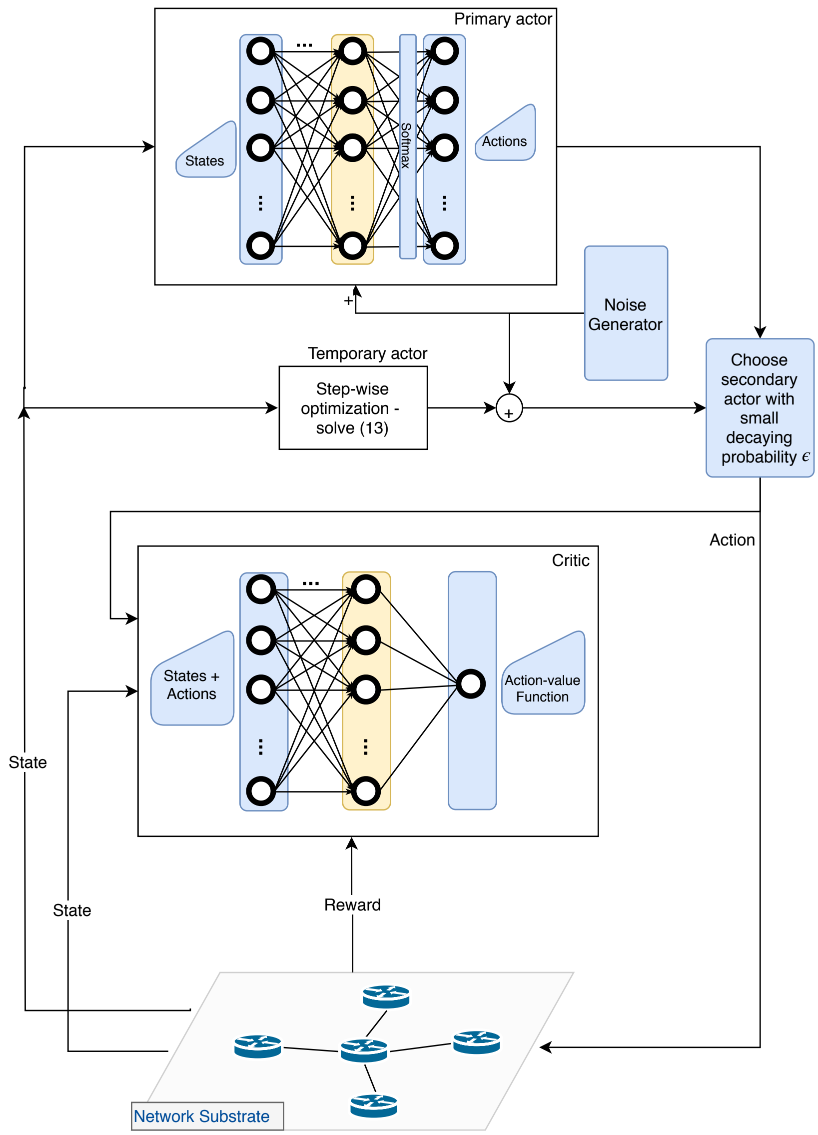

Figure 5.3: The model-assisted DDPG framework.

Algorithm 5: Model-assisted DDPG-based learning algorithm

Generate set of routing and NF placement configurations for S^r;
Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor network $\mu(s|\theta^\mu)$;
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$;
Initialize replay buffer R;
for e $= 1 : M$ **do**
    Receive initial observation states $\boldsymbol{s}_1$;
    Create a perturbed actor network such that $\tilde{\theta}^{\tilde{\mu}} = \theta^\mu + \mathcal{N}(0, \sigma_n^2)$;
    for t $= 1 : T$ **do**
        if $\mathcal{U}(0,1) \leq \epsilon$ **then**
            Select actions $\boldsymbol{a}_t$ by solving (5.13);
            Add noise to action $(\boldsymbol{a}_t = \boldsymbol{a}_t + \mathcal{N}(0, \sigma_n^e))$;
        end
        else
            Select actions $\boldsymbol{a}_t$ from perturbed actor network $\tilde{\mu}$;
        end
        Execute actions $\boldsymbol{a}_t$, and observe reward r_t and new states $\boldsymbol{s}_t$;
        Store transitions $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1})$ in R;
        Sample a random mini-batch of N transitions $(\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}_{i+1})$ from R;
        Set $y_i = r_i + \gamma Q'(\boldsymbol{s}_{i+1}, \mu'(\boldsymbol{s}_{i+1}|\theta^{\mu'})|\theta^{Q'})$;
        Update critic by minimizing loss function using (5.19);
        Update the actor policy using the sampled policy gradient using (5.20);
        Update the target networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$

$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

    end
    Decrease ϵ if bigger than zero ($\epsilon \leftarrow \epsilon^\Delta$ if $\epsilon > 0$);
end

[106]. Parameter σ_n can be controlled by measuring its induced variance on the action space [106]. Moreover, to aid the exploration process, we guide the learning algorithm with some domain-knowledge as follows. With a small decaying probability (ϵ), choose a perturbed action by invoking a step-wise version of the defined optimization problem in (5.13), which can be solved by Gurobi solver.

Algorithm 5 summarizes the model-assisted DDPG-based learning algorithm. First, using the proposed pre-processing stage, we generate a set of routing and NF placement configurations for the service request (in line 1). Then, we randomly initialize the primary and secondary actor and critic deep neural networks, and we initialized the replay buffer (lines 2-4). At the beginning of each episode, we create a perturbed actor network by injecting Gaussian noise to the parameters of the respective deep neural network (in line 7). Then, for each timestep, we select an action vector from either the perturbed actor policy or the perturbed step-wise solution with probability $1 - \epsilon$ and ϵ, respectively (in lines 9-15). The selected action vector is executed in the network substrate, for which a reward is observed, and new states are produced (in line 16). Then, we store the previous transition tuple (or experience), namely $(\boldsymbol{s}_t, \boldsymbol{a}_t, r_t, \boldsymbol{s}_{t+1})$, in the replay buffer (in line 18). Here, then we sample several transitions at random from the replay buffer to break the correlations caused by sequential experiences. Then, we update the critic network by computing the loss function using 5.19 (in line 20). Then, we update the actor network using 5.20 (in line 21). Finally, we slowly update the target network parameters to track the primary network (in line 22). After the end of each episode, we decrease step-wise exploration factor ϵ (in line 24).

Fig. 5.3 provides a pictorial representation of the main components in the model-assisted DDPG framework. Here, we have two deep neural networks that correspond to the actor and critic, respectively. The actor network takes the states as input and produces an action vector. For the actor network, we apply a Softmax layer at the output layer to produce an action vector that sums up to unity. The critic network takes the states and actions as input and produces one output that corresponds to the action-value function ($Q(\boldsymbol{s}_\tau, \boldsymbol{a}_\tau)$). We also have a temporary actor that produces the step-wise solution and is selected to act with small decaying probability ϵ.

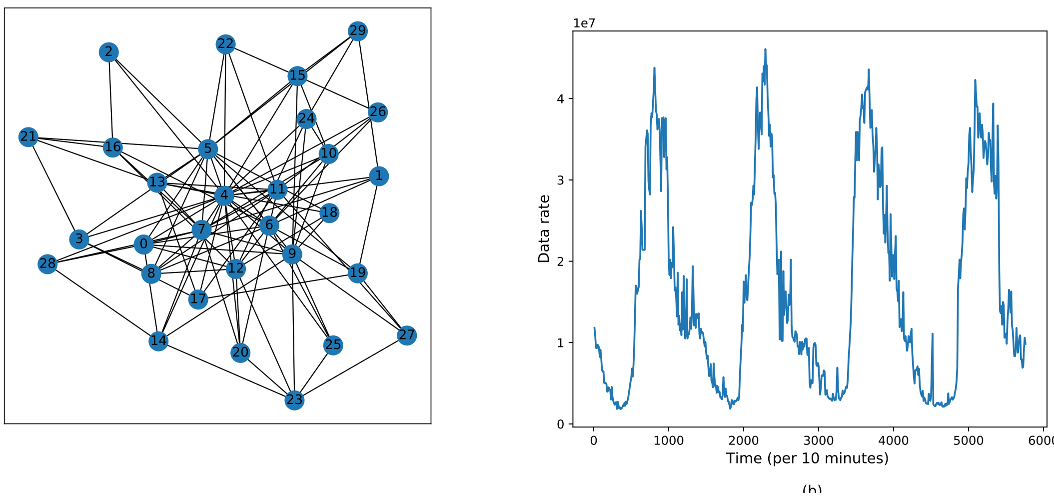

Figure 5.4: (a) A random topology with 30 nodes that is generated using the Barbási–Albert preferential attachment model; (b) HTTP-type traffic trace.

5.4 Performance Evaluation

In this section, we numerically evaluate and analyze the performance of the proposed model-assisted deep RL algorithm. To evaluate the efficiency of the proposed approach, we use two benchmarks, namely step-wise optimization and vanilla DDPG algorithm. In the step-wise optimization, we run an instantaneous version of (5.13) at each decision epoch to maximize the instantaneous reward. The vanilla DDPG algorithm does not utilize the domain-based exploration method that is described in Subsection 5.3.4. That is, for the vanilla DDPG algorithm, we omit lines 9-12 from Algorithm 5.

Both learning algorithms employ 2 hidden feed-forward neural network layers with 32 nodes for both the actor and the critic. Also, we use a leaky rectifier (ReLU) as activation functions in the hidden layers. The learning rates of the actor and critic are 10^{-4} and 10^{-3}, respectively. The discount factor (ϑ) and the learning rate (υ) are set to 0.99 and 0.001, respectively. The memory and batch sizes are set to 10^6 and 64, respectively, and the variance of the action noise (σ_n^2) is set to 0.1. For the model-assisted DDPG algorithm, we set the probability of invoking the model-based solution to $\epsilon = 0.08$ with a decaying factor of $\Delta = 1.001$.

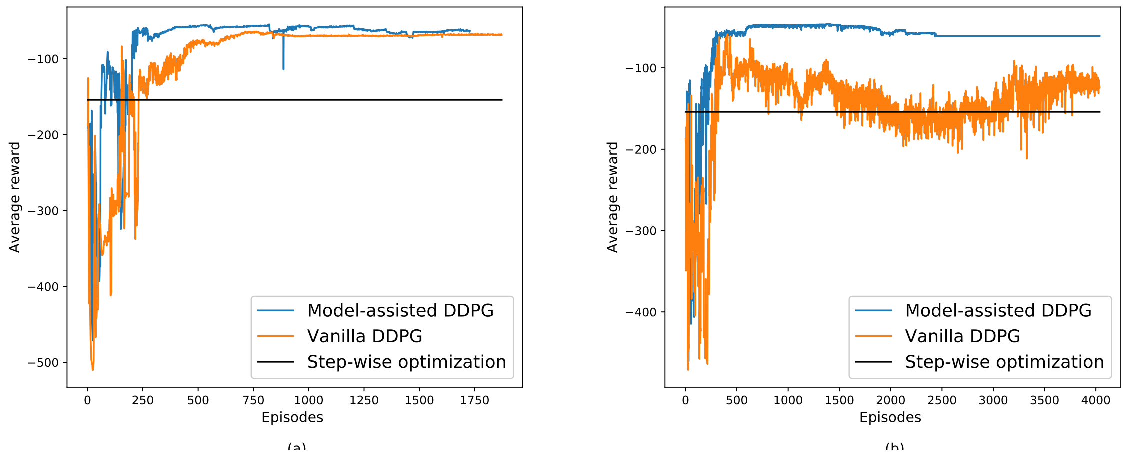

Figure 5.5: Average reward per episode of the proposed model-assisted DDPG algorithm compared to the step-wise optimization and the vanilla DDPG algorithm while varying the randomization seeds in (a) and (b) over a random topology with 30 nodes.

For the traffic pattern of the network services, we utilize an open-source traffic trace that is collected and maintained by the Widely Integrated Distributed Environment research group [107]. The traffic trace is collected from a 96-hour traffic on four consecutive days in 2019. We extract the HTTP traffic (port 443) from the raw data packet trace. Based on the timestamp of each packet arrival, we sample the number of packets every 10 minutes as shown in Fig. 5.4-(b). Here, the traffic varies from 5 to 65 Giga packet/s.

For the network substrate, we use the Barbási–Albert preferential attachment model to generate scale-free random networks [103] (see Fig. 5.4-(a)). The transmission and processing resources are randomly distributed between 25 and 55 Giga packet/s. We consider that all nodes can be NFV nodes, where each NFV node can host 2/3 of the possible NF types at random. The transmission and processing resources are randomly distributed between 25 and 55 Giga packet/s.

First, we conduct a case study on the performance of the proposed algorithm and the two benchmarks over a random network substrate with 30 nodes ($|\mathcal{N}| = 30$). We generate a service request with one NF from node 23 to node 15. Fig. 5.5 shows the average reward per episode for the considered problem setup with two different randomization seeds in Figs. 5.5-(a) and 5.5-(a), respectively. As shown, the proposed model-assisted DDPG

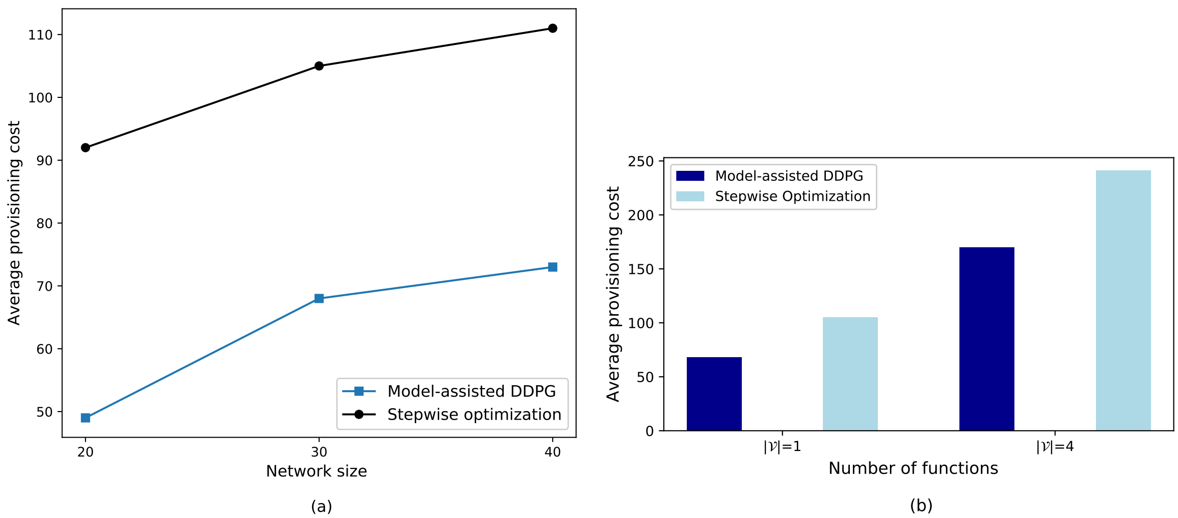

Figure 5.6: Performance of the proposed algorithm and the stepwise optimization over a random network substrate as the network size and the number of NFs vary in (a) and (b), respectively.

algorithm converges to a solution that outperforms the step-wise optimization, implying that the temporal traffic patterns are learned and harnessed to minimize the NF setup cost as the traffic varies. More specifically, it is observed that the proposed algorithm learns to activate a new path (or install a new NF instance) only due to the long-run time-varying characteristics rather than any short-term increase in the traffic demand.

Comparing the learning algorithms, we observe that the model-assisted algorithm is more robust and is faster to converge and stabilize. In Fig. 5.5-(a), the model-assisted algorithm stabilized after episode 400, whereas the vanilla algorithm stabilized after episode 750. In Fig. 5.5-(b), where only the randomization seed is changed, the performance of the vanilla DDPG kept fluctuating and did not converge until episode 4000. This experiment demonstrates how brittle is the Vanilla DDPG algorithm, as it has been observed to be very sensitive to hyperparameters. The brittleness and hyper-sensitivity of the Vanilla DDPG algorithm have been reported in other studies [108, 109].

Next, we analyze the performance of the trained agent, i.e., the proposed model-assisted algorithm after convergence. Similarly, we use the Barbási–Albert method to generate random network substrates. We repeat the experiment over different instantiations of the network substrate to measure the average performance. We generate a service request from node 23 to node 15 with varying number of NFs. Due to the extreme brittleness of the Vanilla DDPG algorithm, hereafter we only compare the trained model-assisted agent with the step-wise optimization.

Fig. 5.6-(a) shows the average provisioning cost of the trained agent and the step-wise optimization as the network size varies while the number of NFs is set to 1. We can observe an increasing trend in the provisioning cost for both algorithms due to the increase in the number of path lengths for the service request. We also observe an approximately constant performance gap between the model-assisted algorithm and the step-wise benchmark since the number of NFs does not vary for the same service request.

Fig. 5.6-(b) shows the average provisioning cost of the trained agent and the step-wise optimization as the number of NFs varies. Here, the gap between the model-assisted algorithm and the step-wise optimization widens as the number of NFs increases. This is because the event of setting up and tearing down NF instances becomes more frequent for the step-wise optimization as the number of NFs increases, thereby the inefficiency of the step-wise benchmark becomes more pronounced.

Chapter 6

Conclusions and Future Works

In this thesis, we investigate an orchestration and provisioning for NFV-enabled network services. Throughout the thesis, we emphasize on practical and flexible design considerations to harness the capabilities of NFV and SDN and cater towards pervasive next-generation networks. For example, we consider that multicast packet replication can occur before the last NF in a service to cater towards geo-distributed services (in Problem I), and that NFs can be mandatory and best-effort for increased flexibility in the admission mechanism (in Problem II), and that NFs can be split in a sequential or parallel manner to adapt to the time-varying traffic demand (in Problem III).

In more detail, in Chapter 3, we study a joint traffic routing and NF placement framework for multicast services over a substrate network under an SDN-enabled NFV architecture. Within the framework, we first investigate a joint multipath-enabled multicast routing and NF placement for a single-service scenario. Then, we extend the investigation for a multi-service scenario. For the single-service scenario, we formulate an optimization problem to minimize function and link provisioning cost, under the physical resource constraints and flow conservation constraints. Our problem formulation is flexible as it allows one-to-many and many-to-one NF mapping, and incorporates multipath routing by constructing multiple trees to deliver the multicast service. The formulated problem is an MILP, and thus can be solved to obtain optimal solutions as a benchmark. For the multi-service case, we present an optimization framework that jointly deals with multiple service requests. We aim to find an optimal combination of service requests and their joint routing and NF placement configurations, such that the aggregate throughput of the core network is maximized, while the function and link provisioning costs are minimized. To reduce the computational complexity in solving the problems in both scenarios, heuristic approaches are proposed to find accurate solutions close to that of the optimal solutions.

In Chapter 4, we propose a joint admission mechanism and an online composition, routing, and NF placement algorithm for unicast and multicast NFV-enabled services. We consider services with multiple mandatory and best-effort NF instances, which is shown to offer a natural generalization to previous works. Through a primal-dual based analysis, it is shown that a provable competitive performance can be achieved, which can be tuned depending on the allowed variability of the profit function and the desired optimality. The online framework herein does not assume any statistical models on the arrival pattern of the service requests, nor does it have any probabilistic assumptions on the sources, destinations, and NFs. Therefore, the provided analysis provides a fundamental understanding of the nature of the profit-maximization problem for NFV-enabled services with multiple resource types.

In Chapter 5, we develop a deep RL based dynamic provisioning mechanism for NFV-enabled services. In Problem III, we take into consideration the transmission and processing resources, the NF setup cost, and the routing overhead. The deep RL algorithm relies on an actor-critic architecture with the DDPG algorithm. However, incorporating such considerations renders the Vanilla DDPG algorithm incapable of consistently achieving the desired behavior due to its notorious sensitivity to hyperparameters. Therefore, to aid with the exploration process (which is a significant issue in RL problems) and to speed up the convergence of the learning, we propose to leverage and integrate domain-based knowledge obtained from a step-wise formulation with the deep RL algorithm.

Next, we highlight some avenues for future research. With regard to the routing and NF placement problem, future research is needed to incorporate the E2E delay requirement to achieve quality of service satisfaction. This is a challenging issue since directly expressing the E2E delay as a function of the decision variables of an optimization problem in a closed-form is cumbersome. However, given an embedded NF chain, one can model (or measure) the E2E delay, upon which an (iterative) algorithm can be developed to re-adjust the embedding solution of violated NF chains. For instance, Ye et al. propose an analytical E2E packet delay modeling framework, based on queueing network modeling, for multiple embedded NF chains while taking into account the computing and transmission resource sharing [110]. By incorporating the aforementioned E2E delay model, a delay-aware NF chain embedding algorithm can be developed, which is left as open research.

For the online routing and NF placement problem, we presented a primal-dual framework that incorporates the processing and transmission resource types with multiple mandatory and best-effort NFs. The analyzed worst-case competitive performance therein can be improved by including more contextual assumptions (e.g., adding probabilistic/stochastic assumptions). In doing so, the performance of the online algorithm is expected to be improved, while retaining the robustness of the competitive analysis (at least in a probabilistic/stochastic sense). Moreover, the problem of online multicast services, for which destinations arrive in an online manner, is worth investigating.

As previously mentioned, the research community is in the early stages of incorporating contemporary machine learning to networking and traffic engineering problems. Applying ready off-the-shelf deep learning and deep RL algorithms do not apply directly in many scenarios, thereby the need to adapt and modify RL algorithms to new domains. Traffic engineering solutions need to be robust and consistent. Therefore, the integration of robust and consistent conventional algorithms with RL can be useful. Moreover, in many networking problems, one is faced with highly conflicting objectives, rare events, and a multitude of constraints, for which carefully designed learning algorithms are needed. Scalability also is a notable challenge in current deep learning algorithms, for which (partially) distributed and multi-agent learning approaches can help.

Finally, we observe that efforts of the industry and the academia are shifting towards developing the next generation of wireless (access) networks. To that end, the research herein and the above ideas should be re-investigated for wireless access networks. For instance, one can investigate the orchestration and provisioning of (multicast) NFV-enabled services in wireless access networks, for which various new fundamental challenges arise due to, for instance, the inherent broadcast nature of the wireless medium, the ubiquity and heterogeneity of devices, and dynamicity of NFV nodes and users.

References

[1] N. Zhang, P. Yang, S. Zhang, D. Chen, W. Zhuang, B. Liang, and X. Shen, "Software defined networking enabled wireless network virtualization: Challenges and solutions," *IEEE Netw.*, vol. 31, no. 5, pp. 42–49, 2017.

[2] S. Q. Zhang, A. Tizghadam, B. Park, H. Bannazadeh, and A. Leon-Garcia, "Joint NFV placement and routing for multicast service on SDN," in *Proc. IEEE NOMS*, 2016, pp. 333–341.

[3] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Commun. Surv. Tuts.*, vol. 20, no. 3, pp. 2429–2453, 2018.

[4] H. Huang, P. Li, S. Guo, and W. Zhuang, "Software-defined wireless mesh networks: architecture and traffic orchestration," *IEEE Netw.*, vol. 29, no. 4, pp. 24–30, 2015.

[5] S. Zhang, W. Quan, J. Li, W. Shi, P. Yang, and X. Shen, "Air-ground integrated vehicular network slicing with content pushing and caching," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2114–2127, 2018.

[6] H. Peng, Le Liang, X. Shen, and G. Y. Li, "Vehicular communications: A network layer perspective," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1064–1078, 2019.

[7] W. Shi, H. Zhou, J. Li, W. Xu, N. Zhang, and X. Shen, "Drone assisted vehicular networks: Architecture, challenges and opportunities," *IEEE Netw.*, vol. 32, no. 3, pp. 130–137, 2018.

[8] B. Yi, X. Wang, M. Huang, and A. Dong, "A multi-stage solution for NFV-enabled multicast over the hybrid infrastructure," *IEEE Commun. Lett.*, vol. 21, no. 9, pp. 2061–2064, 2017.

[9] R. Malli, X. Zhang, and C. Qiao, "Benefits of multicasting in all-optical networks," *Proc. SPIE*, vol. 3531, pp. 209–220, 1998.

[10] V. V. Vazirani, *Approximation algorithms*, 2nd ed. Berlin, Germany: Springer-Verlag, 2004.

[11] E. N. Gilbert and H. O. Pollak, "Steiner minimal trees," *J. Appl. Math.*, vol. 16, no. 1, pp. 1–29, 1968.

[12] M. Zhao, B. Jia, M. Wu, H. Yu, and Y. Xu, "Software defined network-enabled multicast for multi-party video conferencing systems," in *Proc. IEEE ICC*, 2014, pp. 1729–1735.

[13] C. Qiao, H. Yu, J. Fan, W. Zhong, X. Cao, X. Gao, Y. Zhao, and Z. Ye, "Virtual network mapping for multicast services With max-min fairness of reliability," *J. Opt. Commun. Netw.*, vol. 7, no. 9, pp. 942–951, 2015.

[14] J. Blendin, J. Ruckert, T. Volk, and D. Hausheer, "Adaptive software defined multicast," in *Proc. IEEE NetSoft*, 2015, pp. 1–9.

[15] S. Q. Zhang, Q. Zhang, H. Bannazadeh, and A. Leon-Garcia, "Routing algorithms for network function virtualization enabled multicast topology on SDN," *IEEE Trans. Netw. Serv. Manag.*, vol. 12, no. 4, pp. 580–594, 2015.

[16] M. Zeng, W. Fang, and Z. Zhu, "Orchestrating tree-type VNF forwarding graphs in inter-DC elastic optical networks," *J. Lightw. Technol.*, vol. 34, no. 14, pp. 3330–3341, 2016.

[17] Z. Xu, W. Liang, M. Huang, M. Jia, S. Guo, and A. Galis, "Approximation and online algorithms for NFV-enabled multicasting in SDNs," in *Proc. IEEE ICDCS*, 2017, pp. 625–634.

[18] J.-J. Kuo, S.-H. Shen, M.-H. Yang, D.-N. Yang, M.-J. Tsai, and W.-T. Chen, "Service overlay forest embedding for software-defined cloud networks," in *Proc. IEEE ICDCS*, 2017, pp. 720–730.

[19] O. Alhussein, P. T. Do, J. Li, Q. Ye, W. Shi, W. Zhuang, X. Shen, X. Li, and J. Rao, "Joint VNF placement and multicast traffic routing in 5G core networks," in *Proc. IEEE Globecom*, 2018, pp. 1–6.

[20] O. Alhussein, P. T. Do, J. Li, W. Shi, W. Zhuang, and X. Shen, "Customization of virtual network topology for multicast services," *IEEE J. Sel. Areas Commun.*, 2020, in press. doi: 10.1109/JSAC.2020.2986591

[21] M. Liu, G. Feng, J. Zhou, and S. Qin, "Joint two-tier network function parallelization on multicore platform," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 3, pp. 990–1004, 2019.

[22] H. A. Alameddine, C. Assi, M. H. Kamal Tushar, and J. Y. Yu, "Low-latency service schedule orchestration in NFV-based networks," in *Proc. IEEE NetSoft*, 2019, pp. 378–386.

[23] K. Qu, W. Zhuang, Q. Ye, X. S. Shen, X. Li, and J. Rao, "Delay-aware flow migration for embedded services in 5G core networks," in *Proc. IEEE ICC*, 2019, pp. 1–6.

[24] H. Huang, S. Guo, J. Wu, and J. Li, "Service chaining for hybrid network function," *IEEE Trans. Cloud Comput.*, pp. 1–1, 2017.

[25] J. Chen, Q. Ye, W. Quan, S. Yan, P. T. Do, W. Zhuang, X. S. Shen, X. Li, and J. Rao, "SDATP: An SDN-based adaptive transmission protocol for time-critical services," *IEEE Netw.*, pp. 1–9, 2019.

[26] M. Ghaznavi, N. Shahriar, S. Kamali, R. Ahmed, and R. Boutaba, "Distributed service function chaining," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2479–2489, 2017.

[27] S. Palkar, C. Lan, S. Han, K. Jang, A. Panda, S. Ratnasamy, L. Rizzo, and S. Shenker, "E2: A framework for NFV applications," in *Proc. ACM SOSP*, 2015, pp. 121–136.

[28] Y. Xie, Z. Liu, S. Wang, and Y. Wang, "Service function chaining resource allocation: A survey," *CoRR*, vol. abs/1608.00095, 2016.

[29] J. Gil Herrera and J. F. Botero, "Resource allocation in NFV: a comprehensive survey," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 518–532, 2016.

[30] Z. Ye, X. Cao, J. Wang, H. Yu, and C. Qiao, “Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization,” *IEEE Netw.*, vol. 30, no. 3, pp. 81–87, 2016.

[31] J. Li, W. Shi, Q. Ye, W. Zhuang, X. Shen, and X. Li, “Online joint VNF chain composition and embedding for 5G networks,” in *Proc. IEEE Globecom*, 2018, pp. 1–6.

[32] Y. Ma, W. Liang, Z. Xu, and S. Guo, “Profit maximization for admitting requests with network function services in distributed clouds,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 5, pp. 1143–1157, 2019.

[33] T. Lukovszki and S. Schmid, “Online admission control and embedding of service chains,” in *Proc. SIROCCO*, 2015, pp. 104–118.

[34] G. Even, M. Medina, G. Schaffrath, and S. Schmid, “Competitive and deterministic embeddings of virtual networks,” *Theoretical Comput. Sci.*, vol. 496, pp. 184–194, 2013.

[35] N. R. Devanur, K. Jain, B. Sivan, and C. A. Wilkens, “Near optimal online algorithms and fast approximation algorithms for resource allocation problems,” *J. ACM*, vol. 66, no. 1, pp. 1–41, 2019.

[36] M. Huang, W. Liang, Z. Xu, W. Xu, S. Guo, and Y. Xu, “Online unicasting and multicasting in software-defined networks,” *Comput. Netw.*, vol. 132, pp. 26–39, 2018.

[37] Z. Xu, W. Liang, A. Galis, Y. Ma, Q. Xia, and W. Xu, “Throughput optimization for admitting NFV-enabled requests in cloud networks,” *Comput. Netw.*, vol. 143, pp. 15–29, 2018.

[38] B. Awerbuch, Y. Azar, and S. Plotkin, “Throughput-competitive on-line routing,” in *Proc. IEEE SFCS*, 1993, pp. 32–40.

[39] N. Buchbinder and J. S. Naor, “The design of competitive online algorithms via a primal-dual approach,” *Foundations and Trends in Theoretical Comput. Sci.*, vol. 3, no. 2–3, pp. 93–263, 2009.

[40] S. Y. Choi, “Resource configuration and network design in extensible networks,” Doctorial Dessertation, Washington University in St. Louis, 2003.

[41] S. Mehraghdam and H. Karl, “Placement of services with flexible structures specified by a YANG data model,” in *Proc. IEEE NetSoft*, 2016, pp. 184–192.

[42] M. T. Beck and J. F. Botero, “Coordinated allocation of service function chains,” in *Proc. IEEE Globecom*, 2015, pp. 1–6.

[43] P. Marbach, O. Mihatsch, and J. N. Tsitsiklis, “Call admission control and routing in integrated services networks using neuro-dynamic programming,” *IEEE J. Sel. Areas Commun.*, vol. 18, no. 2, pp. 197–208, Feb 2000.

[44] Z. Xu, J. Tang, J. Meng, W. Zhang, Y. Wang, C. H. Liu, and D. Yang, “Experience-driven networking: A deep reinforcement learning based approach,” in *Proc. IEEE INFOCOM*, Apr. 2018, pp. 1871–1879.

[45] Y. V. Kiran, T. Venkatesh, and C. S. Ram Murthy, “A reinforcement learning framework for path selection and wavelength selection in optical burst switched networks,” *IEEE J. Sel. Areas Commun.*, vol. 25, no. 9, pp. 18–26, 2007.

[46] X. Huang, T. Yuan, G. Qiao, and Y. Ren, “Deep reinforcement learning for multimedia traffic control in software defined networking,” *IEEE Netw.*, vol. 32, no. 6, pp. 35–41, 2018.

[47] Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, “State-of-the-art deep learning: Evolving machine intelligence toward tomorrow’s intelligent network traffic control systems,” *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2432–2455, 2017.

[48] W. Huang, G. Song, H. Hong, and K. Xie, “Deep architecture for traffic flow prediction: Deep belief networks with multitask learning,” *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, 2014.

[49] O. Alhussein and W. Zhuang, “Robust online composition, routing and NF placement for NFV-enabled services,” *IEEE J. Sel. Areas Commun.*, 2020, in press. doi: 10.1109/JSAC.2020.2986612

[50] K. Calvert, “Reflections on network architecture: an active networking perspective,” *J. ACM SIGCOMM Comput. Commun. Review*, 2006.

[51] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, and G. J. Minden, "A survey of active network research," *IEEE Commun. Magazine*, vol. 35, no. 1, pp. 80–86, 1997.

[52] N. Feamster, J. Rexford, and E. Zegura, "The road to SDN: An intellectual history of programmable networks," *J. ACM SIGCOMM Comput. Commun. Review*, 2014.

[53] B. Nunes and M. Mendonca, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Commun. Surveys Tuts.*, 2014.

[54] L. Yang, R. Dantu, T. Anderson, and R. Gopal, "Forwarding and control element separation (ForCES) framework," *RFC 3746*, 2004. [Online]. Available: https://tools.ietf.org/pdf/rfc3746.pdf

[55] M. Caesar, D. Caldwell, and N. Feamster, "Design and implementation of a routing control platform," *Proc. Symp. NSDI*, vol. 2, 2005.

[56] A. Greenberg and G. Hjalmtysson, "A clean slate 4D approach to network control and management," *J. ACM SIGCOMM Comput. Commun. Review*, 2005.

[57] M. Casado, M. Freedman, and J. Pettit, "Ethane: taking control of the enterprise," *J. ACM SIGCOMM Comput. Commun. Review*, 2007.

[58] N. McKeown and T. Anderson, "OpenFlow: Enabling innovation in campus networks," *J. ACM SIGCOMM Comput. Commun. Review*, 2008.

[59] N. Gude, T. Koponen, J. Pettit, and B. Pfaff, "NOX: towards an operating system for networks," *J. ACM SIGCOMM Comput. Commun. Review*, 2008.

[60] M. Chiosi, C. Don, W. Peter, and R. Andy, "White paper: Network functions virtualisation: an introduction, benefits, enablers, challenges & call for action," 2012. [Online]. Available: https://portal.etsi.org/NFV/NFV_White_Paper.pdf

[61] K. Greene, "TR10: Software-defined networking - MIT technology review," 2009. [Online]. Available: http://www2.technologyreview.com/news/412194/tr10-software-defined-networking/

[62] NGMN Alliance, "5G white paper," 2015. [Online]. Available: https://www.ngmn.org/wp-content/uploads/NGMN_5G_White_Paper_V1_0.pdf

[63] G. Wang, S. Zhou, S. Zhang, Z. Niu, and S. X., “SFC-based service provisioning for reconfigurable space-air-ground integrated networks,” *IEEE J. Sel. Areas Commun.*, 2020, in press.

[64] M. Li, C. N., Y. Wang, J. Gao, L. Zhao, and X. Shen, “Energy-efficient UAV-assisted mobile edge computing: Resource allocation and trajectory optimization,” *IEEE Trans. Veh. Technol.*, 2020, in press.

[65] L. Huang, H. Hung, C. Lin, and D. Yang, “Scalable steiner tree for multicast communications in software-defined networking,” *CoRR*, vol. abs/1404.3454, 2014.

[66] S. Mehraghdam, M. Keller, and H. Karl, “Specifying and placing chains of virtual network functions,” in *Proc. IEEE CloudNet*, 2014, pp. 7–13.

[67] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, “On orchestrating virtual network functions,” in *Proc. IEEE CNSM*, 2015, pp. 50–56.

[68] M. C. Luizelli, L. R. Bays, L. S. Buriol, M. P. Barcellos, and L. P. Gaspary, “Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions,” in *Proc. IFIP/IEEE IM*, 2015, pp. 98–106.

[69] S. Q. Zhang, Q. Zhang, A. Tizghadam, B. Park, H. Bannazadeh, R. Boutaba, and A. Leon-Garcia, “Sector: TCAM space aware routing on SDN,” in *Int. Teletraffic Congr.*, 2016, pp. 216–224.

[70] Q. Ye, J. Li, K. Qu, W. Zhuang, X. Shen, and X. Li, “End-to-end quality of service in 5G networks: Examining the effectiveness of a network slicing framework,” *IEEE Veh. Technol. Mag.*, vol. 13, no. 2, pp. 65–74, 2018.

[71] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts, “On-line load balancing with applications to machine scheduling and virtual circuit routing,” in *Proc. ACM STOC*, 1993, pp. 623–631.

[72] M. Huang, W. Liang, Z. Xu, W. Xu, S. Guo, and Y. Xu, “Dynamic routing for network throughput maximization in software-defined networks,” in *Proc. IEEE INFOCOM*, 2016, pp. 1–9.

[73] J. Li, W. Shi, Q. Ye, W. Zhuang, X. Shen, and X. Li, "Online joint VNF chain composition and embedding for 5G networks," in *Proc. IEEE Globecom*, 2018, pp. 1–6.

[74] J. Cao, Y. Zhang, W. An, X. Chen, J. Sun, and Y. Han, "VNF-FG design and VNF placement for 5G mobile networks," *Science China Information Sciences*, vol. 60, no. 4, p. 040302, 2017.

[75] M. Jalalitabar, E. Guler, D. Zheng, G. Luo, L. Tian, and X. Cao, "Embedding dependence-aware service function chains," *IEEE/OSA J. Optical Commun. Netw.*, vol. 10, no. 8, pp. 64–74, 2018.

[76] Z. Xu, J. Tang, J. Meng, W. Zhang, Y. Wang, C. H. Liu, and D. Yang, "Experience-driven networking: A deep reinforcement learning based approach," in *Proc. IEEE INFOCOM*, 2018, pp. 1871–1879.

[77] L. Gu, D. Zeng, W. Li, S. Guo, A. Y. Zomaya, and H. Jin, "Intelligent VNF orchestration and flow scheduling via model-assisted deep reinforcement learning," *IEEE J. Selected Areas Commun.*, pp. 1–1, 2019.

[78] S. Troia, R. Alvizu, and G. Maier, "Reinforcement learning for service function chain reconfiguration in NFV-SDN metro-core optical networks," *IEEE Access*, vol. 7, pp. 167 944–167 957, 2019.

[79] X. Fu, F. R. Yu, J. Wang, Q. Qi, and J. Liao, "Service function chain embedding for NFV-enabled IoT based on deep reinforcement learning," *IEEE Commun. Magazine*, vol. 57, no. 11, pp. 102–108, 2019.

[80] S. Wang, Y. Guo, N. Zhang, P. Yang, A. Zhou, and X. S. Shen, "Delay-aware microservice coordination in mobile edge computing: A reinforcement learning approach," *IEEE Trans. on Mobile Comput.*, pp. 1–1, 2019.

[81] L. Hou, L. Lei, K. Zheng, and X. Wang, "A Q-learning-based proactive caching strategy for non-safety related services in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4512–4520, 2019.

[82] J. Pei, P. Hong, M. Pan, J. Liu, and J. Zhou, "Optimal VNF placement via deep reinforcement learning in SDN/NFV-enabled networks," *IEEE J. Sel. Areas Commun.*, pp. 1–1, 2019.

[83] W. Wang, B. Liang, and B. Li, “Multi-resource generalized processor sharing for packet processing,” in *Proc. IEEE/ACM IWQoS*, 2013, pp. 1–10.

[84] A. Ghodsi, V. Sekar, M. Zaharia, and I. Stoica, “Multi-resource fair queueing for packet processing,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 4, pp. 1–12, 2012.

[85] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica, “Dominant resource fairness: Fair allocation of multiple resource types,” in *Proc. USENIX NSDI*, 2011, pp. 323–336.

[86] R. T. Marler and J. S. Arora, “The weighted sum method for multi-objective optimization: New insights,” *Structural and Multidisciplinary Optimization*, vol. 41, no. 6, pp. 853–862, 2010.

[87] M. A. Gennert and A. L. Yuille, “Determining the optimal weights in multiple objective function optimization,” in *Proc. IEEE Computer Vision*, 1988, pp. 87–89.

[88] M. Chowdhury, M. R. Rahman, and R. Boutaba, “ViNEYard: Virtual network embedding algorithms with coordinated node and link mapping,” *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 206–219, 2012.

[89] M. R. Garey and D. S. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*, 1st ed. San Francisco: Freeman, 1979.

[90] S. Chopra, “Comparison of formulations and a heuristic for packing Steiner trees in a graph,” *Ann. Oper. Res.*, vol. 50, no. 1, pp. 143–171, 1994.

[91] A. M. Abbas and B. N. Jain, “Mitigating path diminution in disjoint multipath routing for mobile Ad Hoc networks,” *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 1, no. 3, pp. 137–146, 2006.

[92] ——, “Path diminution in node-disjoint multipath routing for mobile Ad Hoc networks is unavoidable with single route discovery,” *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 5, no. 1, pp. 7–21, 2010.

[93] R. L. Graham, “An efficient algorithm for determining the convex hull of a finite planar set,” *Inf. Process. Lett.*, vol. 1, pp. 132–133, 1972.

[94] N. Zhang, Y. Liu, H. Farmanbar, T. Chang, M. Hong, and Z. Luo, "Network slicing for service-oriented networks under resource constraints," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2512–2521, 2017.

[95] M. T. Beck and J. F. Botero, "Coordinated allocation of service function chains," in *Proc. IEEE Globecom*, 2015, pp. 1–6.

[96] J. C.-I. Chuang and M. A. Sirbu, "Pricing multicast communication: A cost-based approach," *Telecommun. Syst.*, vol. 17, no. 3, pp. 281–297, 2001.

[97] C. Chekuri, S. Khanna, and F. B. Shepherd, "The all-or-nothing multicommodity flow problem," in *Proc. ACM STOC*, 2004, pp. 156–165.

[98] N. Buchbinder and J. Naor, "Improved bounds for online routing and packing via a primal-dual approach," in *Proc. IEEE FOCS*, 2006, pp. 293–304.

[99] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet topology zoo," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 9, pp. 1765–1775, 2011.

[100] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[101] S. Q. Zhang, Q. Zhang, A. Tizghadam, B. Park, H. Bannazadeh, R. Boutaba, and A. Leon-Garcia, "Sector: TCAM space aware routing on SDN," in *Proc. ITC*, vol. 01, 2016, pp. 216–224.

[102] S. H. Strogatz, "Exploring complex networks," *Nature*, vol. 410, no. 6825, pp. 268–276, 2001.

[103] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Rev. Mod. Phys.*, vol. 74, pp. 47–97, 2002.

[104] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *CoRR*, vol. abs/1312.5602, 2013.

[105] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *CoRR*, vol. abs/1509.02971, 2015.

[106] M. Plappert, R. Houthooft, P. Dhariwal, S. Sidor, R. Y. Chen, X. Chen, T. Asfour, P. Abbeel, and M. Andrychowicz, "Parameter space noise for exploration," *CoRR*, vol. abs/1706.01905, 2017.

[107] M. W. Group, "Packet traces from wide backbone," 2019. [Online]. Available: http://mawi.wide.ad.jp/mawi/

[108] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," *CoRR*, vol. abs/1709.06560, 2017.

[109] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *Proc. ICML*, 2016, pp. 1329–1338.

[110] Q. Ye, W. Zhuang, X. Li, and J. Rao, "End-to-end delay modeling for embedded VNF chains in 5G core networks," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 692–704, 2019.