Traffic Engineering for Service-Oriented 5G Networks with SDN-NFV Integration

Kaige Qu, Weihua Zhuang, Qiang Ye, Xuemin (Sherman) Shen, Xu Li, and Jaya Rao

Abstract

Network slicing is a promising approach to provisioning service-level virtual network customization, based on the integration of SDN and NFV technologies. Although different network slices are logically independent, they are physically operated over a common infrastructure, resulting in challenges for QoS guarantee among slices in the presence of traffic dynamics. In this article, a TE framework is proposed for efficient resource management among slices, to avoid congestion and prevent the consequent QoS performance degradation. An NFV architecture integrated with two-level SDN controllers located in tenant and infrastructure domains can support the proposed TE framework. A case study is presented to evaluate the effectiveness of the proposed TE framework, in terms of QoS performance guarantee, improved resource utilization, and reduced reconfiguration overhead.

Introduction

The fifth generation (5G) communication networks are envisioned to support a broad range of new services. There are three typical 5G use case families: enhanced mobile broadband (eMBB), massive machine-type communication (mMTC), and ultra-reliable and low latency communication (uRLLC), whose disparate performance requirements are difficult to be satisfied by the legacy one-size-fits-all network architecture. Instead, network slicing is required on a per-service basis, to provide service-level performance guarantees. Multiple network slices with diverse performance requirements are embedded over a common physical infrastructure [1]. This requires a flexible and programmable network architecture, with abstraction on both the plane and layer dimensions [2]. Software-defined networking (SDN) brings the plane-dimension abstraction by decoupling the data and control planes. With a global network view and flow awareness brought by SDN, end-to-end (E2E) data delivery paths can be dynamically established and resources are explicitly allocated to different paths by an SDN controller. Network function virtualization (NFV) provides the layer-dimension abstraction, by abstracting physical resources to virtual resources with a virtualization layer and realizing service-level functionalities, referred to as virtual network functions (VNFs) [2, 3]. Traditionally, service providers rely on dedicated hardware middleboxes to realize network functions as in-path packet processing units required by a service. Compared with middleboxes, VNFs are more cost-efficient and flexible for deployment and management. Several frameworks have been proposed for SDN-NFV integration, to fully exploit their advantages and provide an integrated architecture with abstractions in both the plane and layer dimensions for customized service provisioning [2, 4].

With SDN-NFV integration, a tenant such as a service provider requests network services in the form of service function chains (SFCs). An SFC is composed of multiple VNFs in a predefined order, to fulfill a composite service with certain processing and transmission resource demands, according to service-level agreements (SLAs) negotiated with an infrastructure provider (InP). The resource demands are usually static and estimated from long-term traffic statistics and quality-of-service (QoS) requirements [5–7]. The InP customizes network services over the physical infrastructure, generating network slices tailored for each service. However, with traffic load fluctuations during the operation of network slices, the states of both processing and transmission resources alternate between overutilized and underutilized, causing temporal mismatch between traffic load and resource availability. Also, the imbalanced load distribution over both processing and transmission resources can result in local performance bottlenecks and inefficient resource usage at the same time, causing spatial mismatch between traffic load and resource availability. The temporal-spatial mismatch between traffic load and resource availability is detrimental to both service performance and resource utilization. Therefore, how to overcome traffic load fluctuations for consistent service-level QoS guarantee requires further investigation [8, 9].

In this article, we propose a traffic engineering (TE) framework for efficient resource management among slices, by taking into account traffic load dynamics. The goal of TE is to guarantee the E2E delay performance of each service, to achieve load balancing for long-term efficient resource utilization, and to reduce the reconfiguration overhead. The TE problem is formulated as a multi-objective optimization problem, and a heuristic algorithm is proposed to obtain a time-efficient solution. To support the proposed TE framework in service-oriented 5G networks, the extended functionalities of different blocks in a reference network architecture with SDN-NFV integration are discussed. A case study is presented for performance evaluation of the proposed TE framework.

Digital Object Identifier:
10.1109/MNET.001.1900508

Kaige Qu, Weihua Zhuang, and Xuemin (Sherman) Shen are with the University of Waterloo; Qiang Ye (corresponding author) is with Minnesota State University; Xu Li and Jaya Rao are with Huawei Technologies Canada Inc.

0890-8044/20/$25.00 © 2020 IEEE

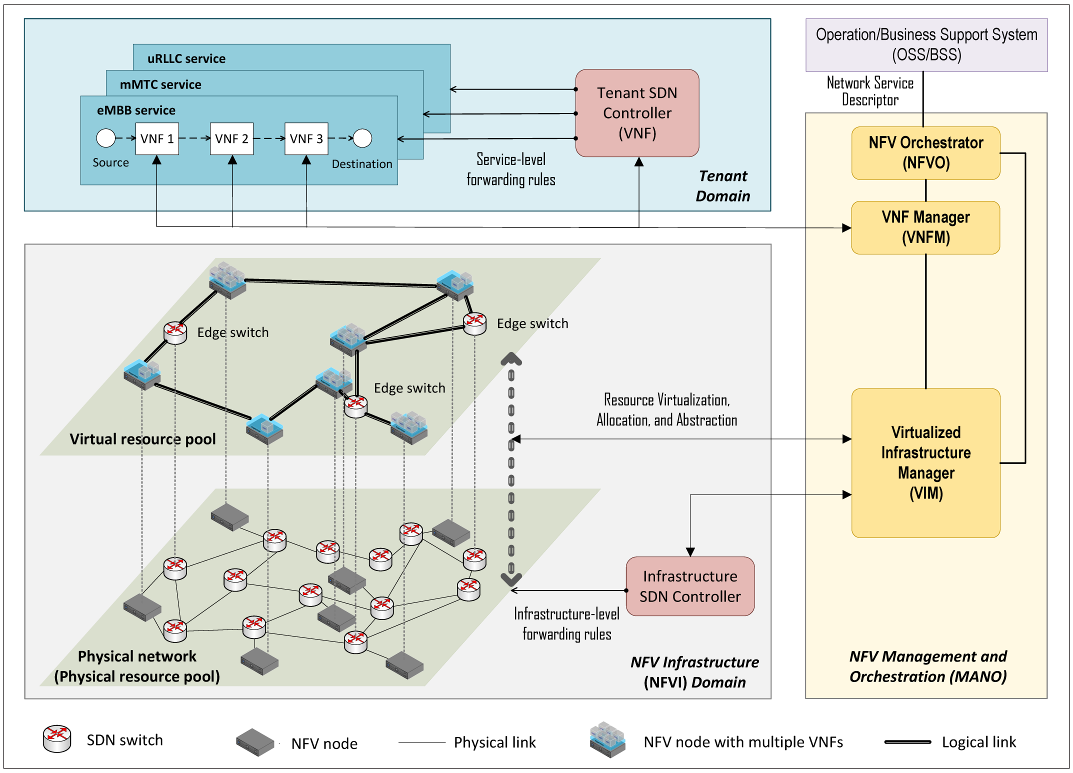

uRLLC service
mMTC service
eMBB service
VNF 1
VNF 2
VNF 3
Source
Destination
Tenant SDN Controller (VNF)
Service-level forwarding rules
Tenant Domain
Operation/Business Support System (OSS/BSS)
Network Service Descriptor
NFV Orchestrator (NFVO)
VNF Manager (VNFM)
Edge switch
Edge switch
Edge switch
Virtual resource pool
Resource Virtualization, Allocation, and Abstraction
Virtualized Infrastructure Manager (VIM)
Infrastructure SDN Controller
Infrastructure-level forwarding rules
Physical network (Physical resource pool)
NFV Infrastructure (NFVI) Domain
NFV Management and Orchestration (MANO)
SDN switch
NFV node
Physical link
NFV node with multiple VNFs
Logical link

FIGURE 1. Network slicing and traffic engineering supported by an extended NFV MANO architecture with SDN integration.

System Model

A time-slotted system is considered, in which the timeline is divided into time intervals with equal length. TE is performed for a time interval with predicted QoS violations. We describe the system model in three parts, as illustrated in Fig. 1.

NFV Infrastructure Domain

Physical Network: A physical network consists of SDN switches and NFV nodes interconnected by physical links. Switches forward traffic from incoming physical links to outgoing physical links. Some switches act as edge switches for service access. NFV nodes, such as commodity servers and data centers (DCs), have both forwarding and processing capabilities. The physical network contains a physical resource pool, including transmission resources on physical links and processing resources at NFV nodes. A path in the physical network, that is, a physical path, is composed of a series of physical links and SDN switches between two NFV nodes or between one NFV node and one edge switch. The maximum transmission rate supported by a physical path is allocated by a central orchestrator.

Virtual Resource Pool: We call a logical abstraction of all physical paths with pre-allocated transmission resources between two different NFV nodes or between one NFV node and one edge switch as a logical link. The maximum transmission rate supported by a logical link is the aggregate maximum transmission rate over all its underlying physical paths. Transmission resources on logical links are seen as virtual resources, since the mapping between logical links and physical paths are transparent to service flows traversing the logical links. With the consideration that processing resources on NFV nodes can be distributed among several VNFs through virtualization, we introduce the concept of virtual resource pool. A virtual resource pool with a certain topology is represented as a directed graph, in which vertices include all NFV nodes and edge switches, and edges include all logical links. The virtual resource pool is abstracted from the physical resource pool, which ignores composition details of the physical paths associated with the logical links. It makes both SDN switches and physical links fully transparent to service flows on the logical links. A path in the virtual resource pool, that is, a virtual path, is composed of a series of logical links and NFV nodes between two edge switches. It is possible that the virtual resource pool is not a fully connected graph, that is, not every two NFV nodes or edge switches are directly connected by a logical link. Assume that there are sufficient transmission resources available in the physical network. We can scale up/down resources on existing logical links, remove an existing logical link, and find physical paths with sufficient resources for extra logical links. In this way, the topology of the virtual resource pool can be updated, which enables flexible logical link provisioning among the fixed NFV nodes.

Infrastructure SDN Controller: With SDN, packet forwarding rules are configured in SDN switches by an infrastructure SDN controller to route traffic flows over a physical path. For logical link provisioning, the infrastructure SDN controller is responsible for configuring forwarding rules on physical paths associated with each logical link, and enforcing a traffic splitting ratio among corresponding physical paths for each logical link. When a topology update for the virtual resource pool is required, the infrastructure SDN controller is responsible for (re-)configuring forwarding rules on physical paths for the scaled and extra logical links, and removing those associated with the removed logical links.

Tenant Domain

Services: A service request is represented as an SFC with specified QoS requirements, including average E2E delay requirement and maximal tolerable downtime in one service interruption. There are two levels of connectivity in an SFC, namely, service-level and infrastructure-level. The service-level connectivity requires that VNFs be chained in a predefined order between the source and destination nodes (fixed at edge switches), to facilitate the E2E service delivery. The service-level connectivity is achieved by mapping an SFC to a virtual path between the source and destination nodes. For two neighboring VNFs in an SFC, packets processed by the upstream VNF are transmitted to the downstream VNF, generating traffic between consecutive VNFs, that is, inter-VNF subflows. The infrastructure-level connectivity requires that each subflow be routed over at least one physical path, if its upstream and downstream VNFs are not co-located. The infrastructure-level connectivity is achieved by mapping each subflow to a logical link which is provisioned via the infrastructure SDN controller.

Tenant SDN Controller: The tenant SDN controller configures service-level forwarding rules at edge switches and NFV nodes to guide packets belonging to a flow traversing an SFC (i.e., an SFC flow) through a virtual path, thus enabling the service-level connectivity. In the presence of traffic variations, an SFC flow can be rerouted to an alternative virtual path via the tenant SDN controller, according to TE decisions made by a central orchestrator.

SDN-NFV Integration

An NFV management and orchestration (MANO) architecture can efficiently manage the life cycle of network functions, services, and their constituent resources in a common NFV infrastructure (NFVI) [1]. The architecture is extended with SDN integration to realize service function chaining and provide TE architectural support [1]. In the following, we discuss the main functional blocks in the architecture and their interactions with the tenant and infrastructure SDN controllers.

Virtualized Infrastructure Manager (VIM): Is responsible for managing resources in the NFVI. Specifically, the VIM deals with resource virtualization and allocation, and maintains the mapping between the virtual resource pool and physical resource pool. The VIM is also in charge of logical link provisioning via an infrastructure SDN controller;

VNF Manager (VNFM): Is in charge of the life cycle management of VNFs, including instantiation, configuration, and scaling. In addition to VNFs serving as network service components, the tenant SDN controller is regarded as a VNF;

NFV Orchestrator (NFVO): Is responsible for central orchestration, and contains a resource orchestrator (RO) and a network service orchestrator (NSO). The RO is responsible for orchestrating NFVI resources. For TE, the RO contains an engine to make dynamic TE decisions. Specifically, it determines the rerouted virtual paths for SFC flows, including both the VNF to NFV node remapping and the consequent subflow to logical link remapping. It also determines the logical link to physical path remapping, to facilitate logical link provisioning. The NSO is responsible for the life cycle management of network services, including service instantiation and dynamic network service capacity scaling. For TE, it triggers TE requests to the RO when potential QoS violations are predicted, due to traffic load fluctuations.

Traffic Engineering for SFCs within Virtual Resource Pool

Overview

Traffic engineering (TE) has been extensively investigated in traditional networks, to find paths for traffic transmission from source to destination within link capacity. However, traditional TE methods cannot be directly applied in service-oriented 5G networks due to the following reasons. First, an SFC flow requires two-dimensional (processing and transmission) resource provisioning, and the potential mismatch between the two-dimensional resources should be addressed. Second, the transfer of VNF states should be considered, since simply rerouting in-progress flows on a state-dependent VNF (i.e., a VNF in which states are stored and updated locally together with packet processing) to an alternative NFV node introduces state inconsistency and processing inaccuracy [10]. Here, we consider TE for SFCs in a virtual resource pool, and focus on the service-level connectivity. How to map a logical link to physical paths is not the focus of this work. We rely on the NFVO to perform such tasks, using typical TE methods such as solving a multi-commodity flow problem. A TE decision within the virtual resource pool determines the remapping between VNFs and NFV nodes. Since the logical link to which a subflow is mapped is uniquely determined by the locations of the corresponding upstream and downstream VNFs, the subflow to logical link remapping is a byproduct of a TE decision. However, to maintain the infrastructure-level connectivity of SFCs, the potential topology update requirements for the virtual resource pool is considered as an overhead for TE.

Elastic VNF Provisioning

Traffic Rate, Packet Processing Rate, and Processing Resources: With traffic load fluctuations, the processing resource demand of each VNF should be determined to guarantee at least an average E2E processing delay for an SFC flow. Consider that the E2E processing delay requirement is initially decomposed into per-hop delay requirements on each VNF, and that TE is performed in a time scale much larger than packet inter-arrival time of a traffic flow. Packet arrivals of an SFC during a sufficiently large time interval are modeled as a Poisson process, with different rates (in packet/s) across different time intervals. Assume that VNF packet processing time is exponentially distributed. Then, packet processing at a VNF is modeled as an M/M/1 queue,[1] with different arrival and service rates across different time intervals. Hence, the per-interval packet processing (service) rate demand of a VNF can be

[1] The M/M/1 queue model is used for simplicity of the presentation.

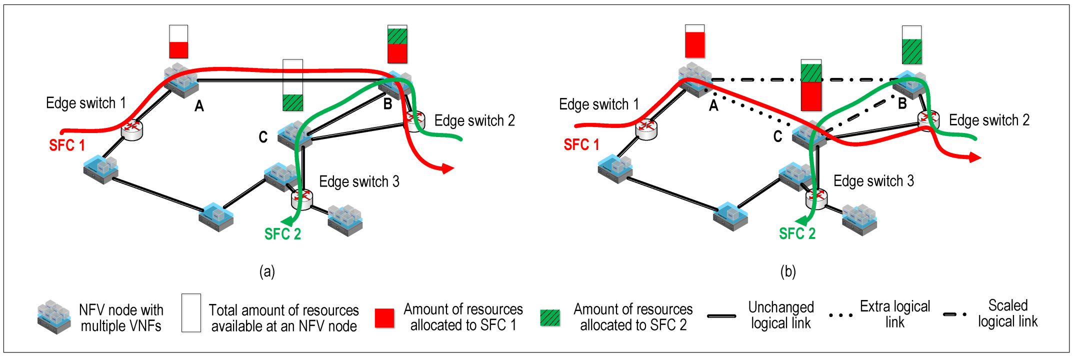


FIGURE 2. An illustration for traffic engineering in a virtual resource pool with topology update: a) before traffic engineering; b) after traffic engineering.

calculated based on Little's Law. Given processing resources, the maximum packet processing rate supported by a VNF depends on both the service (e.g., packet size, service type, function type) and platform (e.g., packet I/O and virtualization technology at an NFV node). We use a processing density (in cycle/packet) to represent the processing resource demand (in cycle/s) of a VNF at an NFV node, corresponding to one packet/s in processing rate. With the queueing and processing density models, the relationships among the traffic rate, processing rate, processing resources, and the VNF processing delay requirement can be established. Then, the time-varying processing resource demand of a VNF at an NFV node can be determined, to satisfy the delay requirement with a time-varying traffic rate.

Processing Resource Sharing: Consider multiple-to-one mapping between VNFs and NFV nodes. To achieve processing resource sharing among multiple VNFs from different SFCs at an NFV node, a CPU polling scheme is employed. Each VNF gets a portion of CPU processing resources, which is linear with the allocated CPU time share in a polling period. Based on the generalized processor sharing (GPS) discipline, the VNFs are guaranteed minimum processing resources (in cycle/s). The percentage of the total allocated CPU time in a CPU polling period is defined as the NFV node loading factor.

Joint VNF Migration and Vertical Scaling: NFV enables elastic scaling of processing resources allocated to VNFs in a cost-effective manner, which facilitates agile service provisioning and management. Dynamic VNF operations, including horizontal scaling, vertical scaling, and migration, are widely employed to provide elastic VNF provisioning [11]. With horizontal scaling, the number of instances for a VNF is scaled in/out, with a constant amount of processing resources for each instance. With vertical scaling, the amount of processing resources for a VNF instance is scaled up/down. With VNF migration, a VNF instance migrates to an alternative NFV node, without changing the amount of processing resources. To avoid overloading and achieve load balancing, we consider joint VNF migration and vertical scaling, under the assumption that a VNF is instantiated once, which means that a VNF can migrate to an alternative NFV node with sufficient resources to satisfy its processing resource scaling demand. Figure 2 illustrates joint VNF migration and vertical scaling for two SFCs, with each SFC composed of two VNFs. We use rectangle height to represent the amount of processing resources. Specifically, one VNF of SFC 1 migrates from NFV node B to NFV node C, and all VNFs are vertically scaled.

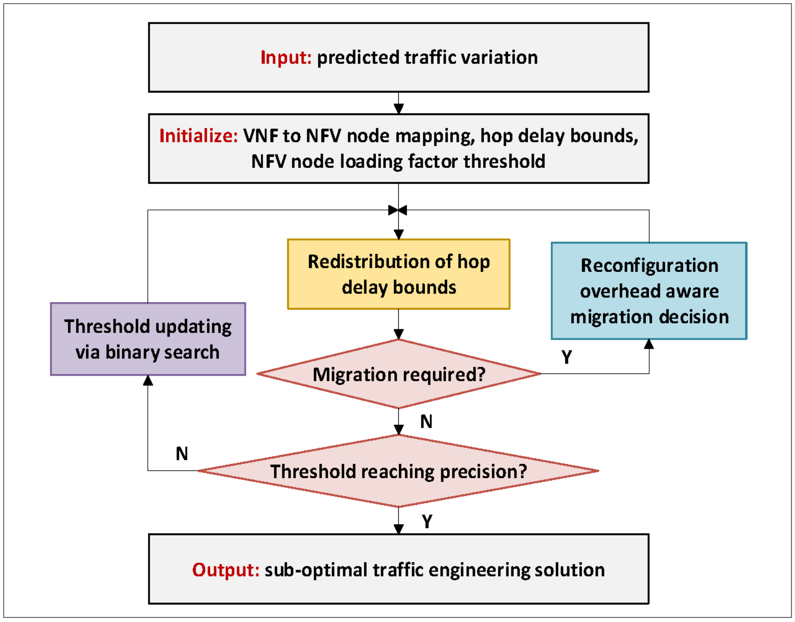
Redistribution of VNF Processing Delay Requirements: Another dimensionality of elasticity comes from redistribution of VNF processing delay requirements, since the E2E processing delay requirement of an SFC is satisfied as long as the aggregation of all VNF processing delays in an SFC does not exceed a specified upper bound.

Flexible Logical Link Provisioning

Assume that there are sufficient transmission resources and no queueing on logical links. For a subflow, if its upstream or downstream VNF migrates to an alternative NFV node, it should be re-mapped to an alternative logical link accordingly, and resources on the original logical link should be released. However, it is possible that the virtual resource pool is not fully connected and extra logical links are required. As shown in Fig. 2, a subflow of SFC 1 is released from logical link A → B, and re-mapped to an extra logical link A → C. Accordingly, the transmission rate over logical link A → B can be scaled down. In this way, logical links are flexibly provisioned, which addresses the potential mismatch between processing resources at fixed NFV nodes and transmission resources on existing logical links. The extra logical links for flow rerouting incur signaling overhead between the infrastructure SDN controller and SDN switches, due to forwarding rule reconfiguration along the underlying physical paths.

Parallel VNF State Transfer

Typical examples of state-dependent VNFs include network address translators that store mappings between ports and hosts, and intrusion detection systems that keep track of pattern matchings for accurate attack detection. Some frameworks such as OpenNF are proposed to solve the state inconsistency problem, by not only moving packets of the rerouted SFC flow but also transferring the associate VNF states [10]. Packet processing is halted during state transfer, causing a service downtime. For a VNF state transfer, the product of state transfer time and transmission rate is equal to the state size [12]. For a remapped SFC with multiple state transfers, we use parallel state transfer in which all state transfers take place simultaneously in the data plane, thus reducing


Input: predicted traffic variation
Initialize: VNF to NFV node mapping, hop delay bounds, NFV node loading factor threshold
Redistribution of hop delay bounds
Reconfiguration overhead aware migration decision
Threshold updating via binary search
Migration required?
Y
N
Threshold reaching precision?
N
Y
Output: sub-optimal traffic engineering solution

AFIGURE 3. A flowchart of the proposed heuristic algorithm.

the transferring latency at the cost of transmission resource overhead [13]. The service downtime with parallel state transfer is the maximum state transfer time along the E2E path, instead of the total time for sequential state transfer. Under the assumption that a TE time interval is sufficiently long, the service downtime is much shorter than the stable service operation time for any service.

A state transfer belonging to one service can share a logical link with subflows from other services. For example, as shown in Fig. 2, a state transfer from SFC 1 and a subflow from SFC 2 both happen on logical link B → C. Generally, since services experience different downtimes, transmission resources allocated to subflows can be opportunistically used by state transfers before the subflows resume packet forwarding. However, for the example in Fig. 2, transmission resources allocated to the subflow cannot be used by the state transfer, since SFC 2 experiences no service downtime. Thus, the transmission rate over logical link B → C should be scaled up. In the worst case, all state transfers happen on logical links where no subflows are mapped or no transmission resources allocated to subflows can be opportunistically used. In this case, dedicated logical links should be established for state transfers, or the transmission resource capacity of existing logical links should be scaled up to support state transfers. Therefore, we consider the total amount of transmission resources required by state transfers as a TE overhead.

Problem Definition and Heuristic Solution

Problem Definition

We consider multiple SFCs in a processing resource limited network. Given predicted traffic rate variations, a delay-aware TE problem within the virtual resource pool is to 1) find the remapping between VNFs and NFV nodes, and 2) vertically scale the amount of processing resources allocated to VNFs, to satisfy the average E2E (processing) delay requirements without violating the maximal tolerable service downtime.

Objective: As discussed above, the reconfiguration overhead in TE consists of two parts: the signaling overhead for configuring extra logical links required for flow rerouting, and the transmission resource overhead incurred by state transfers. Assume that the total signaling overhead has a linear relation with the number of extra logical links. Besides, under the assumption that all VNF states have the same size and all services have the same downtime limits, the amount of transmission resources required by the parallel state transfers linearly varies with the total number of state transfers, that is, the number of VNF migrations. Hence, the minimization of reconfiguration overhead in TE leads to limited modification to the original VNF to NFV node mapping. In this case, the loads on different NFV nodes can be imbalanced, which can result in more migrations in the subsequent time intervals. A balanced load distribution makes the network more tolerant of future demand changes, which is beneficial for achieving long-term efficient resource utilization [14]. Therefore, we formulate the TE problem as an optimization problem, by jointly considering the reconfiguration overhead and load balancing. The VNF to NFV node remapping is denoted by a set of binary decision variables, to represent whether a VNF is mapped to an NFV node or not. The VNF processing resource allocation is denoted by a set of continuous decision variables, to represent the amount of resources allocated to a VNF at an NFV node. The average processing delay for the VNFs at the NFV nodes are represented by a set of continuous decision variables. The objective function to be minimized is a weighted sum of the number of extra logical links, the number of VNF migrations, and the maximum NFV node loading factor. Both the numbers of extra logical links and VNF migrations are dependent on the VNF to NFV node remapping decision variables. The maximum NFV node loading factor is dependent on the VNF processing resource allocation decision variables.

Constraints: The multiple-to-one mapping between VNFs and NFV nodes is enforced by a constraint on the binary VNF to NFV node remapping decision variables. The loading factors of all NFV nodes should not be beyond an upper limit η_U, for example, 0.95. There is also a relationship constraint on the VNF to NFV node remapping decision variables and the VNF processing resource allocation decision variables, since the amount of resources allocated to a VNF at an NFV node should be zero if the VNF is not mapped to the NFV node. The average processing delay for a VNF at an NFV node is expressed based on the M/M/1 queueing model. Then, the average E2E (processing) delay of an SFC can be represented by a summation of all average VNF processing delays in the SFC, which should not exceed the average E2E delay requirement.

Heuristic Algorithm

The formulated TE problem is an NP-hard mixed integer non-convex optimization problem. For time tractability, a heuristic algorithm is proposed to obtain a sub-optimal solution, with a flowchart given in Fig. 3. Consider that the E2E (processing) delay requirement for each SFC is initially decomposed into a set of per-hop processing delay requirements at the VNFs (i.e., hop delay bounds). We first calculate NFV node loading factors with predicted traffic variations and the initial hop delay bounds, based on the M/M/1 delay model and processing density model, for the initial VNF to NFV node mapping. The calculated NFV node loading factors can be even larger than 1. By comparing the calculated NFV node loading factors with threshold η_{th} (initial value set as η_U), a set of overloaded NFV nodes is identified as potential bottlenecks.

Redistribution of Hop Delay Bounds: Even if potential bottlenecks are identified, migration may not be necessary. For a given threshold, η_{th}, how an E2E delay requirement is decomposed into hop delay bounds affects the number of overloaded NFV nodes. By making hop delay bounds less stringent on overloaded NFV nodes and more stringent on underloaded ones, it is possible to reduce the number of overloaded NFV nodes. The basic idea is as follows: if an SFC traverses both overloaded and underloaded NFV nodes, loading factors of the underloaded ones are increased to η_{th}, by reducing corresponding hop delay bounds, and loading factors of the overloaded ones are decreased, by increasing corresponding hop delay bounds. This strategy is referred to as delay scaling, which is performed iteratively until there is no SFC traversing both overloaded and underloaded NFV nodes. The iterative delay scaling procedure with given threshold, η_{th}, is referred to as the redistribution of hop delay bounds.

Reconfiguration Overhead Reduction: A redistribution of hop delay bounds with the initial threshold ($\eta_{th} = \eta_U$) is performed after the initialization step. If the number of overloaded NFV nodes is reduced to zero, no migration is required. Otherwise, migration is necessary to overcome traffic overloading. Migration decisions are made sequentially, each followed by a redistribution of hop delay bounds with the initial threshold, until no more migration is required. With alternate migration decision and redistribution of hop delay bounds, reconfiguration overhead is greedily reduced in two ways. One is the potential reduction of overloaded NFV nodes. The other is the consideration of reconfiguration overhead in migration decisions. A migration decision includes three steps, that is, identification of a bottleneck NFV node, selection of an SFC to migrate, and selection of a target NFV node. First, the most heavily loaded NFV node is identified as the bottleneck. Next, an SFC to migrate from the bottleneck NFV node and a target NFV node to accommodate the migrated SFC are jointly selected to minimize the reconfiguration overhead, that is, 1 plus the number of extra logical links for flow rerouting. If there are multiple choices, an SFC with the largest resource demand is migrated to the closest target NFV node.

Load Balancing: After the sequential migration decision procedure, all NFV node loading factors are less than or equal to the initial threshold η_U. The gap between the smallest and largest NFV node loading factors can be large, which is undesired in terms of load balancing. Actually, if one SFC traverses two NFV nodes with different loading factors, its hop delay bound can be relaxed at the NFV node with the larger loading factor and be shrunk on the other NFV node, to reduce the gap between the two loading factors. Based on this idea, an iterative procedure with alternate threshold updating and redistribution of hop delay bounds is performed to gradually reduce the gap and to balance the NFV node loading factors. The threshold, η_{th}, is first reduced stepwise from the initial value η_U, with an initial step size, until some overloaded NFV nodes are detected, after redistribution of hop delay bounds with the updated threshold. The emergence of overloaded NFV nodes indicates that the latest step of threshold reduction is too aggressive. Then, a binary search between the latest two threshold values is performed, until no overloaded NFV nodes are detected and a sufficient precision is reached.

A Case Study

A case study is presented to evaluate the performance of the proposed TE heuristic algorithm, within a 64-node mesh virtual resource pool in which logical links exist only between neighboring nodes. We consider homogeneous processing densities and a maximum NFV node processing rate of 1000 packet/s. There are three SFCs, each with four VNFs, initially mapped to the virtual resource pool. SFC 3 shares two NFV nodes with SFC 1 and one NFV node with SFC 2. The average E2E delay requirement and the maximal tolerable service downtime for each SFC are 20ms and 5ms, respectively. The VNF state size is a constant, equal to 10 bytes. All SFCs have an initial traffic rate of 200 packet/s during the previous time interval (k – 1). Denote the predicted traffic load during current time interval k for SFC s as $\lambda^{(s)}(k)$. We have $\lambda^{(2)}(k)$ = 200 packet/s, and vary both $\lambda^{(1)}(k)$ and $\lambda^{(3)}(k)$ from 200 packet/s to 780 packet/s. The initial step size and the precision for threshold updating are set to 0.1 and 0.0001, respectively.

Load Balancing and Reconfiguration Overhead Trade-off

Figure 4 shows the TE performance with the increase of $\lambda^{(1)}(k)$ and $\lambda^{(3)}(k)$. Performance metrics include the maximum NFV node loading factor, $\eta(k)$, the number of migrations, $N(k)$, and the number of extra logical links, $S(k)$, for flow rerouting. We see a zigzag trend for $\eta(k)$ and step-wise increasing trends for both $N(k)$ and $S(k)$. In Fig. 5, the relationships among the three performance metrics with the increase of $\lambda^{(3)}(k)$ are illustrated, for $\lambda^{(1)}(k)$ = 280, 480, 700 packet/s, respectively. We observe that $\eta(k)$ drops sharply when $N(k)$ or $S(k)$ is increased by 1. When $N(k)$ and $S(k)$ are stable, $\eta(k)$ shows either a linear increasing trend or a flat trend. The linear increasing trend indicates the dominance by $\lambda^{(3)}(k)$, while the flat trend indicates the dominance by $\lambda^{(1)}(k)$. The relationships among $\eta(k)$, $N(k)$, and $S(k)$ demonstrate the trade-off between load balancing and reconfiguration overhead.

Delay and Impact of Traffic Burstiness

We carry out packet-level simulations using network simulator OMNeT++ to evaluate the E2E delay after TE. To verify the effectiveness of our TE model, not only Poisson packet arrivals but also MMPP packet arrivals are simulated. We use a two-state MMPP model with the same transition

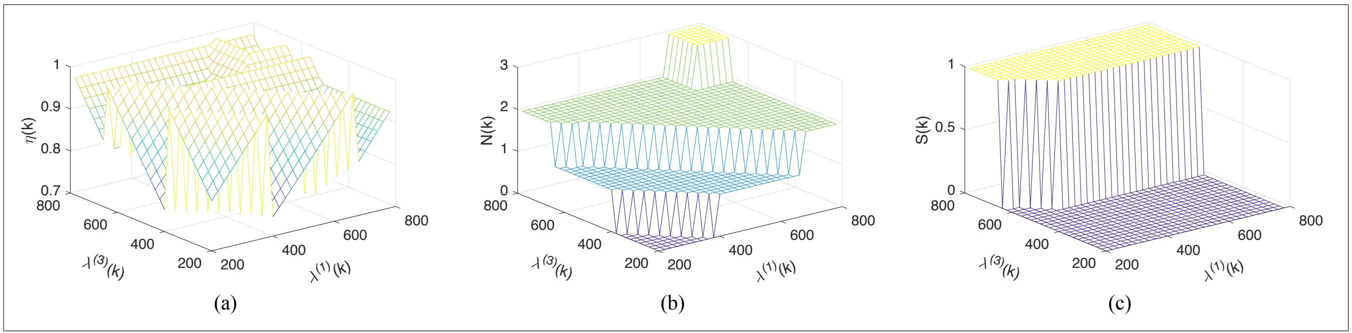


FIGURE 4. Performance metrics of the proposed TE framework with respect to $\lambda^{(1)}(k)$ and $\lambda^{(3)}(k)$: a) $\eta(k)$; b) $N(k)$; c) $S(k)$.

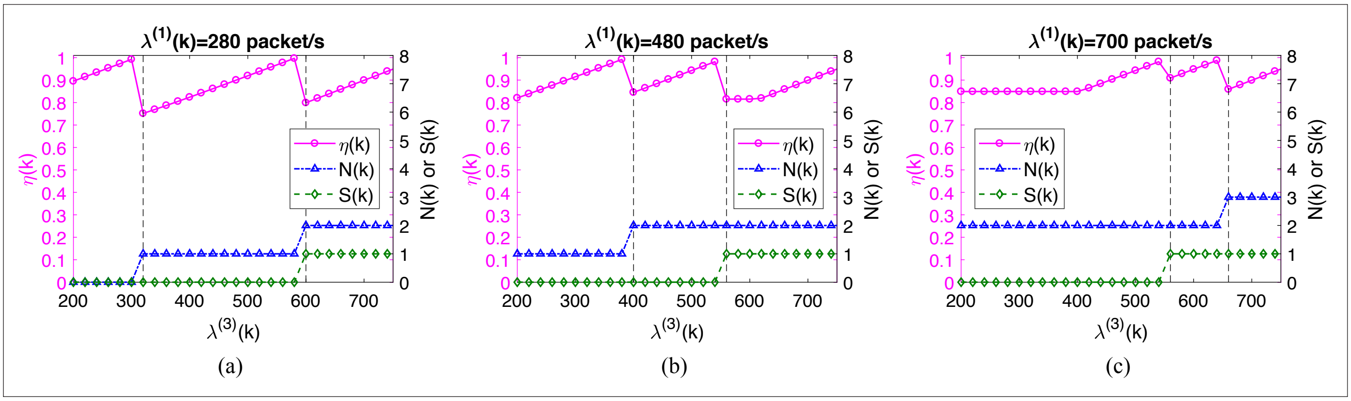


FIGURE 5. Performance trade-off with the increase of $\lambda^{(3)}(k)$ for: (a) $\lambda^{(1)}(k)$ = 280 packet/s; b) $\lambda^{(1)}(k)$ = 480 packet/s; c) $\lambda^{(1)}(k)$ = 780 packet/s.

Traffic arrival	Poisson					MMPP							
Rate (packet/s)	300	400	500	600	700	R_1	R_2	R_1	R_2	R_1	R_2	R_1	R_2
						350	250	450	150	500	100	550	50
Average delay after TE (ms)	19.3	19.7	19.5	19.8	19.7	20.6		22.7		23.3		25.7	

TABLE 1. Average E2E delay performance.

rate between the two states and an average traffic rate of $(R_1 + R_2)/2$ packet/s, with R_1 and R_2 being the individual average packet arrival rate s at the two states. A larger gap between R1 and R2 indicates a higher level of traffic burstiness. The average delay of Poisson traffic arrivals with average rates of 300, 400, 500, 600, 700 packet/s and MMPP traffic arrivals with an average rate of $(R_1 + R_2)/2$ = 300 packet/s after TE are given in Table 1. We observe that the E2E delay requirement (20ms) is satisfied for all Poisson traffic arrivals. However, with the increase of traffic burstiness for the MMPP traffic arrivals, the E2E delay performance degrades. How to incorporate traffic burstiness in the TE framework remains our future work.

Conclusion

In this article, we present a traffic engineering framework within an NFV/SDN architecture in service-oriented 5G networks, to achieve consistent QoS provisioning for multiple service-level network slices in the presence of traffic variations. We consider services in the form of SFCs in which VNFs are chained to fulfill a composite service delivery. The TE problem is formulated as a multi-objective mixed integer optimization problem, and a time-efficient heuristic algorithm is presented. In the case study, the performance of the proposed TE heuristic algorithm is evaluated, demonstrating a trade-off between load balancing and reconfiguration overhead. A packet-level simulation is performed to show the delay performance of different traffic arrivals with the proposed TE model.

References

[1] J. Ordonez-Lucena *et al.*, "Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, May 2017, pp. 80–87.

[2] Q. Duan, N. Ansari, and M. Toy, "Software-Defined Network Virtualization: An Architectural Framework for Integrating SDN and NFV for Service Provisioning in Future Networks," *IEEE Netw.*, vol. 30, no. 5, Sept. 2016, pp. 10–16.

[3] C. Lorenz *et al.*, "An SDN/NFV-Enabled Enterprise Network Architecture Offering Fine-Grained Security Policy Enforcement," *IEEE Commun. Mag.*, vol. 55, no. 3, Mar. 2017, pp. 217–23.

[4] ETSI NFV ISG, "NFV-EVE005: SDN Usage in NFV Architectural Framework," Oct. 2015.

[5] O. Alhussein *et al.*, "Joint VNF Placement and Multicast Traffic Routing in 5G Core Networks," *Proc. IEEE GLOBECOM*, Dec. 2018, pp. 1–6.

[6] Q. Ye *et al.*, "End-to-End Delay Modeling for Embedded VNF Chains in 5G Core Networks," *IEEE Internet of Things J.*, vol. 6, no. 1, Feb. 2019, pp. 692–704.

[7] Q. Ye *et al.*, "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, June 2018, pp. 65–74.

[8] K. Qu *et al.*, "Delay-Aware Flow Migration for Embedded Services in 5G Core Networks," *Proc. IEEE ICC*, May 2019, pp. 1–6.

[9] Y. Jia *et al.*, "Online Scaling of NFV Service Chains Across Geo-Distributed 8 Datacenters," *IEEE/ACM Trans. Netw.*, vol. 26, no. 2, Apr. 2018, pp. 699–710.

[10] A. Gember-Jacobson *et al.*, "OpenNF: Enabling Innovation in Network Function Control," *Proc. ACM SIGCOMM*, Aug. 2014, pp. 163–74.

[11] M. Ghaznavi *et al.*, "Elastic Virtual Network Function Placement," *Proc. IEEE CloudNet*, Oct. 2015, pp. 255–60.

[12] J. Xia *et al.*, "Reasonably Migrating Virtual Machine in NFV-Featured Networks," *Proc. IEEE Conf. Computer and Information Technology*, Dec. 2016, pp. 361–66.

[13] B. Zhang *et al.*, "Co-Scaler: Cooperative Scaling of Software-Defined NFV Service Function Chain," *Proc. IEEE Conf. Network Function Virtualization and Software Defined Networks*, Nov. 2016, pp. 33–38.

[14] L. Guo, J. Pang, and A. Walid, "Dynamic Service Function Chaining in SDN-Enabled Networks with Middleboxes," *Proc. IEEE ICNP*, Nov. 2016, pp. 1–10.

Biographies

Kaige Qu [S'19] received her B.S. degree in communication engineering from Shandong University, Jinan, China, in 2013. She received her M.S. degrees in integrated circuits engineering and electrical engineering from Tsinghua University, Beijing, China, and KU Leuven, Leuven, Belgium, respectively, in 2016. She is currently pursuing her Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. Her research interests include resource allocation in SDN/NFV-enabled networks, 5G and beyond, and machine learning for future networking.

Weihua Zhuang [M'93, SM'01, F'08] has been with the Department of Electrical and Computer Engineering, University of Waterloo, Canada, since 1993, where she is a professor and a Tier I Canada Research Chair in wireless communication networks. She was a recipient of the 2017 Technical Recognition Award from the IEEE Communications Society Ad Hoc and Sensor Networks Technical Committee, and several best paper awards from IEEE conferences. She is a Fellow of the Royal Society of Canada, Canadian Academy of Engineering, and Engineering Institute of Canada. She is an Elected BoG Member and VP–Publications of the IEEE Vehicular Technology Society.

Qiang Ye [S'16, M'17] has been an assistant professor with the Department of Electrical and Computer Engineering and Technology, Minnesota State University, Mankato, MN, USA, since September 2019. He received his Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2016. He had been with the Department of Electrical and Computer Engineering, University of Waterloo, as a post-doctoral fellow and then a research associate from December 2016 to September 2019. His current research interests include 5G networks, SDN/NFV, network slicing, AI and machine learning for future networking.

Xuemin (Sherman) Shen [M'97, SM'02, F'09] is currently a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on network resource management, wireless network security, social networks, 5G and beyond, and vehicular ad hoc networks. He is a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Chinese Academy of Engineering Foreign Fellow. He received the R.A. Fessenden Award in 2019 from IEEE, Canada, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, the Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society.

Xu Li is a senior principal researcher at Huawei Technologies Canada. He received a Ph.D. (2008) degree from Carleton University, an M.Sc. (2005) degree from the University of Ottawa, and a B.Sc. (1998) degree from Jilin University, China, all in computer science. Prior to joining Huawei, he worked as a research scientist (with tenure) at Inria, France. His current research interests are focused on 5G and beyond. He contributed extensively to the development of 3GPP 5G standards through 90+ standard proposals. He has published 100+ refereed scientific papers and is holding 40+ issued U.S. patents.

Jaya Rao received his B.S. and M.S. degrees in electrical engineering from the University of Buffalo, New York, in 2001 and 2004, respectively, and his Ph.D. degree from the University of Calgary, Canada, in 2014. He is currently a senior research engineer at Huawei Technologies Canada, Ottawa. Since joining Huawei in 2014, he has worked on research and design of CIoT, URLLC and V2X based solutions in 5G New Radio. He has contributed for Huawei at 3GPP RAN WG2, RAN WG3, and SA2 meetings on topics related to URLLC, network slicing, mobility management, and session management.