Online Virtual Network Function Scheduling Towards Deterministic Latency

Zhenran Kuai and Shaowei Wang
School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China
Email: DZ1923021@smail.nju.edu.cn, wangsw@nju.edu.cn

GLOBECOM 2023 - 2023 IEEE Global Communications Conference | 979-8-3503-1090-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/GLOBECOM54140.2023.10437982

***Abstract*—Network function virtualization aims to deploy virtual network functions (VNFs) on commercial off-the-shelf hardware, allowing for efficient implementation of network services by scheduling various VNFs. However, the requirements of deterministic networking poses higher demands on VNF scheduling to achieve guaranteed latency performance. In this paper, we study the online VNF scheduling problem with the objective of minimizing the system cost regarding to unfinished services while improving the determinism of latency. The formulated optimization task yields a mixed integer program problem and we propose an online VNF scheduling algorithm to tackle it, which transforms the end-to-end latency constraint into deadlines for each VNF and adopts an online primal-dual method to generate scheduling decisions. Numerical results show that the proposed scheduling algorithm outperforms two traditional rule-based scheduling methods in terms of latency and acceptance ratio.**

***Index Terms*—Deterministic networking, network function virtualization, online scheduling, service function chain.**

I. Introduction

Network function virtualization (NFV) aims to enhance network flexibility and scalability by virtualizing network functions and deploying them on commercial off-the-shelf hardware. The virtualized nature of network functions introduces new complexities concerning resource allocation, load balancing, and Quality of Service compliance [1]. Different from traditional hardware-based deployments, which rely on dedicated devices to implement network functions, NFV allows for dynamic and on-demand instantiation of virtual network functions (VNFs) on shared infrastructure. Multiple instances of VNFs can be interconnected in a series to form a service function chain (SFC), which facilitates the delivery of network services [2]. When different services share the same VNF instances, an effective VNF scheduling scheme is critical to ensure optimal resource utilization, minimize latency, and meet service level agreements.

The traditional scheduling approach of best-effort delivery may not suffice to meet the stringent requirements of time-sensitive applications. Such applications typically necessitate precise timing, low-latency communication, and deterministic performance to ensure operational efficiency and safety [3]. Best-effort delivery, in contrast, can result in a wide probability distribution with a long tail in terms of end-to-end delays [4]. Moreover, the rise of the Internet of Things and the proliferation of connected devices have led to an exponential surge in network traffic and a pressing need for reliable and predictable communication [5]. In this context, deterministic networking emerges as a crucial solution capable of guaranteeing smooth and efficient data exchange.

Existing resource allocation and scheduling schemes for VNFs often prioritize improving latency performance without providing latency guarantees. In [6], VNF mapping and traffic routing is considered for services with strict deadlines within an ultra-low latency network slice. A game-theoretic approach is developed to tackle this challenge by leveraging a decentralized scheduling process. In [7], an efficient heuristic framework is introduced to address the joint VNF placement and routing problem with delay constraints. This framework sequentially handles the two sub-problems, resulting in an improved system acceptance ratio. In [4], the authors investigate the deterministic SFC lifetime management problem in beyond 5G edge fabric and propose algorithms for SFC deployment and adjustment to ensure the deterministic latency. Even if deterministic latency is considered, the focus remains on a fixed set of SFCs to be deployed and scheduled.

However, service requests typically arise continuously, thereby changing current network state and disrupting previous scheduling [8]. Real-time scheduling schemes are necessary to address this issue. In [9], the authors study an online VNF mapping and scheduling problem and attempt to solve it using greedy strategies and tabu search. In [10], the migration and re-instantiates of VNFs are allowed to enable an online VNF remapping and rescheduling scheme, which improves the system acceptance ratio. It should be noted that real-time scheduling solutions can potentially compromise the determinism of previous and future scheduling processes [11]. Therefore, scheduling VNF in an online manner while maintaining a deterministic network presents a significant challenge.

In this paper, we investigate the online VNF scheduling problem to mitigate the long-tail effects of end-to-end latency in network services. The problem yields a mixed-integer programming task with the objective of minimizing the fractional weighted completion time while satisfying the latency constraints. We first examine the online VNF scheduling process for a single node to comprehend how to design a rule that determines the priority of each VNF. Then we propose

This work was supported in part by the National Natural Science Foundation of China under Grants 61931023 and U1936202.

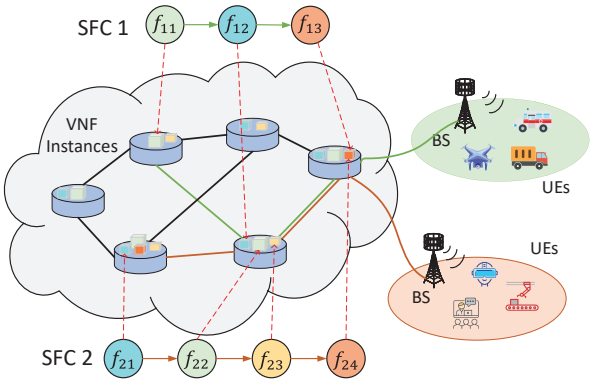

Fig. 1. Illustration of SFCs in a virtual network.

an online algorithm for the scheduling task. We employ a primal-dual method to assign VNFs to virtual nodes, and transform the latency constraint of an SFC into deadlines for each VNF. The assigned VNFs are then scheduled by each individual virtual node. Numerical results demonstrate that the proposed algorithm effectively mitigates both delays and delay variations of services.

The rest of the paper is organized as follows: In Section II, we give system model and formulate optimization problem. In Section III, the online VNF scheduling algorithm is presented. In Section IV, numerical results and discussions are presented. In Section V, we conclude this work.

II. System Model and Problem Formulation

A. System Model

Consider an edge network comprising multiple edge nodes and switches. Virtual machines and containers have been deployed on the edge nodes, on which VNF instances are deployed. All VNF instances form a virtual network represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here, $\mathcal{V}$ represents the set of virtual nodes, and $\mathcal{E}$ represents the set of virtual links. At cell sites, user devices send out service requests over time. As shown in Fig. 1, each service request is fulfilled by passing its traffic flow through the chain of VNFs represented by $f_{k1} \to \cdots \to f_{kn_k}$, where f_{kj} represents the j-th VNF in SFC k and n_k denotes the number of VNFs. We consider the scenario with virtual nodes deployed at the fully connected edge network, where transmission delay is negligible compared to processing delay. For each SFC $k \in \mathcal{K}$, the arrival time is denoted by r_k^s. Let $\mathcal{J}_k$ be the set of VNFs in k and $\mathcal{J}$ be the set of all VNFs.

The required VNFs in the continuous arriving service requests should be deployed in the form of virtual nodes in advance. Then, for VNF j in a given SFC, it should be assigned to the same type virtual nodes $\mathcal{V}(j)$. The processing sequences of VNFs owned by different SFCs on the same node should be scheduled to meet the specific QoS requirement of each service. Since the process of VNF placement is time-consuming and should be performed on a large time scale, we focus on the VNF assignment and scheduling for continuously arriving SFC requests, which together constitute online VNF scheduling process.

The context of deterministic network necessitates satisfying the end-to-end delay requirement for each SFC. Let c_k^s be the completion time of k. We denote the processing deadline of SFC k by d_k^s. The latency constraint is expressed by $c_k^s \le d_k^s + \epsilon(d_k^s - r_k^s)$, where ϵ implies the tolerance of the latency variation. Addressing the potential disruptions of schedules caused by continuous SFC requests and dynamic network environment is critical, since these disruptions may interrupt well-designed schedules and lead to significant latency variation. To maintain the latency and its variation within a deterministic range, we consider decomposing the latency constraint of each SFC into specific deadlines for all VNFs. The deadline of VNF j is denoted by d_j. The set of slots is denoted by $\mathcal{T}$, where $|\mathcal{T}| = \max_k d_k^s$.

B. Problem Formulation

We first provide the offline form of the VNF scheduling problem with identical arrival time. Suppose that all SFCs arrive at slot 0. We use x_{ijt} to denote the ratio of the maximum rate at which node i processes VNF j at slot t. The total processing rate of all VNFs assigned to a node cannot exceed its processing capacity, which is given by

$$\sum_{j \in \mathcal{J}} x_{ijt} \le 1, \quad \forall i \in \mathcal{V}, t \in \mathcal{T}. \tag{1}$$

Each VNF j has a predefined load q_j for processing, which is revealed to the virtual node i upon the arrival of the SFC. Let v_i be the maximum processing rate of each virtual node and q_{jt} be the remaining load at slot t. The amount of load that has been processed is given by

$$\sum_{i \in \mathcal{V}} \sum_{t=1}^{t'} x_{ijt} v_i = q_j - q_{jt'}, \quad \forall j \in \mathcal{J}, \forall t' \in \mathcal{T}. \tag{2}$$

The VNF should be fully processed by the assigned node and the latency constraints for each VNF should be satisfied:

$$\sum_{i \in \mathcal{V}} \sum_{t=1}^{d_j} x_{ijt} v_i = q_j, \quad \forall j \in \mathcal{J}. \tag{3}$$

One VNF can only be assigned to one virtual node:

$$\left(\sum_{t \in \mathcal{T}} x_{ijt} v_i - q_j \right) \sum_{t \in \mathcal{T}} x_{ijt} = 0, \quad \forall i \in \mathcal{V}, j \in \mathcal{J}. \tag{4}$$

One VNF can only be assigned to the VNF instances of the same type:

$$x_{ijt} = 0, \quad \forall i \notin \mathcal{V}(j), j \in \mathcal{J}, t \in \mathcal{T}. \tag{5}$$

In addition, the traffic flow should be processed by the predecessor VNF before being traversed through the successor. Denote the order of VNF in the SFC by $\mathrm{sq}(j)$. We have

$$\sum_{i' \in \mathcal{V}(j')} x_{i'j't} \left(\sum_{i \in \mathcal{V}(j)} \sum_{t=1}^{t'} x_{ijt} v_i - q_j \right) = 0,$$

$$\forall k \in \mathcal{K}, j, j' \in \mathcal{J}_k : \mathrm{sq}(j) < \mathrm{sq}(j'), t' \in \mathcal{T}. \quad (6)$$

The deadline of all VNFs along the SFC should be set in ascending order and cannot be greater than that of the SFC, which is given by

$$d_j \leq d_{j'}, \quad \forall k \in \mathcal{K}, j, j' \in \mathcal{J}_k : \mathrm{sq}(j) < \mathrm{sq}(j'), \quad (7)$$

$$d_j \leq d_k^s(1+\epsilon), \quad \forall k \in \mathcal{K}, j \in \mathcal{J}_k. \quad (8)$$

Each virtual node can only process the requirement of at most one VNF at a time slot:

$$x_{ijt} \in \{0,1\}, \quad \forall i \in \mathcal{V}, j \in \mathcal{J}, t \in \mathcal{T}. \quad (9)$$

$$d_j > 0, \quad \forall j \in \mathcal{J}. \quad (10)$$

Our goal is to dispatch the VNFs to the virtual nodes and process them on their assigned nodes so as to improve the determinism of service provisioning while satisfying latency constraints shown in (3). We use w_j to denote the importance of a VNF j. Incorporating the optimization of the total completion time, the system cost is the weighted impact of remaining processing time on the subsequent scheduling process, i.e., $\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{J}}\sum_{t=1}^{d_j}\frac{w_j p_{jt}}{p_j}\mathbb{I}(\sum_{\tau\in\mathcal{T}} x_{ij\tau} = \frac{q_j}{v_i})$, where $p_{jt} = \frac{q_{jt}}{v_i}$ and $p_j = \frac{q_j}{v_i}$. $\mathbb{I}(\sum_{\tau\in\mathcal{T}} x_{ij\tau} = \frac{q_j}{v_i})$ indicates whether VNF j is assigned to node i. The remaining processing load increases the system cost over time when unprocessed. We aim to minimize the system cost to improve the determinism of completing SFCs, i.e.,

$$\min_{x_{ijt}, d_j} \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{J}}\sum_{t=1}^{d_j}\frac{w_j q_{jt}}{q_j}\mathbb{I}(\sum_{\tau\in\mathcal{T}} x_{ij\tau} = \frac{q_j}{v_i}), \quad (11)$$

$$s.t.\ (1)-(10).$$

III. Primal-dual Method For VNF Scheduling

In this section, we first explore the VNF scheduling process on each virtual node. Then, we extend the process to multiple virtual nodes and propose an online primal-dual VNF scheduling algorithm.

A. Single Node Online Scheduling

Consider a virtual node that is responsible for scheduling the set of J independent VNFs assigned to it, denoted by $\mathcal{J}^S$. The arrival time of all VNFs is 0. The deadlines of all VNFs are sorted in descending order as $d_1 \leq d_2 \leq \cdots \leq d_J$. The virtual node should complete each VNF before its specified deadline and minimize $\sum_j \sum_{t=1}^{d_j}\frac{w_j q_{jt}}{q_j}$. For the convenience of study, we relax the problem to continuous time and replace q_{jt} with a continuous function $q_j(t)$. The cost function for each VNF can be rewritten as $\int_0^{d_J}\frac{q_j(t)}{q_j}w_j dt$. The objective is then given by

$$\sum_{j\in\mathcal{J}^S}\int_0^{d_J}\frac{q_j(t)}{q_j}w_j dt = \sum_{j\in\mathcal{J}^S}\int_0^{d_J}\frac{v w_j t}{q_j}x_j(t)dt, \quad (12)$$

where $x_j(t) = -\frac{dq_j(t)}{vdt}$ is the ratio of the maximum speed v at which VNF j is processed. The problem is given by

$$\mathrm{P1}: \min \sum_{j\in\mathcal{J}^S}\int_0^{d_J}\frac{v w_j t}{q_j}x_j(t)dt, \quad (13)$$

Algorithm 1 Single Node Online Scheduling (SNOS)

Input: A set of VNFs $\mathcal{J}$ and a virtual node.
Output: A VNF schedule.
while there exist uncompleted VNFs **do**
  Generate a schedule for all unfinished VNFs with the EDF rule.
  if no VNF reaches its deadline **then**
    Schedule the VNF with the highest density.
  else
    Find the first VNF that reaches its deadline.
    Schedule the VNF with the highest density from the VNFs whose deadline are less than or equal to the above VNF.
  end if
end while

$$s.t. \int_0^{d_j} x_j(t)dt \geq \frac{q_j}{v}, \forall j \in \mathcal{J}^S, \quad (14)$$

$$\sum_{j\in\mathcal{J}^S} x_j(t) \leq 1, \forall t \in [0, d_J], \quad (15)$$

$$x_j(t) \geq 0, \forall j \in \mathcal{J}^S, t \in [0, d_J]. \quad (16)$$

Constraint (14) ensures that the completion of a VNF should satisfy its latency constraint. Constraint (15) ensures that the total processing speed of all VNFs cannot exceed the processing capacity of the node.

If we replace the latency constraint in (14) with $\int_0^\infty x_j(t)dt = \frac{q_j}{v}$ and define $\frac{v w_j}{q_j}$ as the density of VNF j, we could know from the Theorem 1 in [12] that processing the VNF with the highest density is optimal. Building upon the highest density first (HDF) rule, we propose a single node online scheduling (SNOS) algorithm for P1. The key idea is to schedule the VNF with the HDF rule while trying to meet the delay requirements as much as possible.

Specifically, we first generate a schedule to figure out if scheduling with the HDF will result in any deadline violations for the currently unfinished VNFs. For this purpose, the earliest deadline first (EDF) rule is used, as it allows us to check if all the VNFs can be completed before their deadlines in subsequent scheduling processes. If none of the VNFs exceed their deadlines, we could schedule with the HDF rule. Otherwise, such violations should be avoided while minimizing the objective value. To achieve this, we identify the first VNF that violates its deadline in the generated schedule and designate it as the reference VNF. We schedule the VNF with the highest density from the set of VNFs with deadlines at least as early as the reference VNF. Upon the arrival of a VNF, the SNOS algorithm treats all unfinished VNFs as new VNFs with a remaining load and an arrival time of 0 to perform scheduling. The proposed SNOS procedure is summarized in Algorithm 1.

B. Extension to Multiple Nodes

Since each virtual node manages its scheduling process to meet the deadlines for the VNFs assigned to it by us-

Algorithm 2 Multiple Node Online Scheduling (MNOS)

Input: A set of SFCs $\mathcal{K}$, a virtual network and a deadline control parameter δ.
Output: A VNF schedule.
while there exist uncompleted SFCs **do**
  if there is a newly arrived SFC k **then**
    for VNF j in SFC k **do**
      Let $\{I_l : l \in \mathcal{J}(r_j)\}$ denote the old scheduling intervals before the arrival time r_j of VNF j for node i. Let $\beta'_{it} = \sum_{l\in\mathcal{J}(t)} \sum_{\tau=t}^{T} \frac{v_i w_l}{q_l} x_{il}(\tau)$ be the old β' on node i.
      Assign VNF j to machine $i' = \arg\min_{i\in\mathcal{V}}\{\min_{t>r_j}(\beta'_{it} + \frac{v_i w_j}{q_j}(t-r_j))\frac{q_j}{v_i}\}$.
      Generate a schedule with HDF until j ends.
    end for
    for VNF j in SFC k **do**
      Set a available processing time $\hat{p}_j = (p_j/\sum_{n\in\mathcal{J}_k} p_n)(d_k^s - r_k^s)$.
      Set the deadline for the VNF as $d_j = \hat{r}_j + \hat{p}_j + \delta(d_k^s - r_k^s)$, where $\hat{r}_j = \sum_{n=1}^{j-1} \hat{p}_n$.
    end for
  end if
  Schedule with the SNOS for unfinished VNFs;
end while

ing the SNOS algorithm, we can extend the algorithm to multiple nodes by incorporating the processes of determining the deadlines for all VNFs in an SFC and assigning VNFs to virtual nodes. However, setting a deadline for a VNF is dependent on its processing time, which is only determined after the assignment. Hence, we first delve into the single node scheduling process to obtain some insights for VNF assignment. Let $\boldsymbol{q}(t) = (q_1(t), \cdots, q_k(t))^T$. We define the optimal value function of the scheduling process as

$$V^o(\boldsymbol{q}(t), t) = \min\{\int_t^{d_J} \sum_{j\in\mathcal{J}^S} \frac{v w_j \tau}{q_j} x_j(\tau) d\tau\}. \tag{17}$$

If the latency constraint of P1 is replaced with a constraint that ensures fully processing, expressed as $\int_0^{d_J} x_j(t)dt = \frac{q_j}{v}$, the dual problem is given by

$$\text{P2}: \max \sum_{j\in\mathcal{J}^S} \alpha_j - \int_0^{d_J} \beta_t dt, \tag{18}$$

$$s.t.\ \frac{v}{q_j}\alpha_j \le \beta_t + \frac{v w_j}{q_j} t, \forall j \in \mathcal{J}^S, \tag{19}$$

$$\alpha_j, \beta_t \ge 0, \forall j \in \mathcal{J}^S, t \in [0, d_J]. \tag{20}$$

Let $I_j = [t_j^{b_1}, t_j^{e_1}) \cup \cdots \cup [t_j^{b_{n_j}}, t_j^{e_{n_j}})$ be the intervals where $x_j(t)$ equals 1. By using complementary slackness, the constraint in (19) is tight for optimal dual variables whenever a VNF is scheduled [13]:

$$\frac{v}{q_j}\alpha_j = \beta_t + \frac{v w_j}{q_j} t, \forall j \in \mathcal{J}^S, t \in I_j, \tag{21}$$

Based on the aforementioned conclusions, it is derived in [12] that the optimal dual variables of P1 are related to the optimal value function, which is given by

$$\frac{v}{q_j}\alpha_j = \frac{\partial}{\partial q_j} V^o(\boldsymbol{q}(t), t), \forall j \in \mathcal{J}^S, \forall t \in I_j, \tag{22}$$

$$\beta_t = \frac{\partial}{\partial t} V^o(\boldsymbol{q}(t), t), \forall j \in \mathcal{J}^S, \forall t \in I_j, \tag{23}$$

$$\beta_t \le \sum_{j\in\mathcal{J}^S} \sum_{a=1}^{n_j} \frac{v w_j}{q_j}(t_j^{e_a} - t_j^{b_a}), \forall t \in [0, d_J]. \tag{24}$$

Thus determining the dual variables may help determine $x_j(t)$.

As for the online VNF scheduling for multiple nodes, it can be formulated without latency constraints to gather some instructions of determining the dual variables. We use $\mathcal{J}(t)$ to denote the set of VNFs that are available for processing at t. The online scheduling problem for $\mathcal{J}(0)$ is expressed by

$$\text{P3}: \min \sum_{i\in\mathcal{V}} \sum_{j\in\mathcal{J}(0)} \int_0^{d_M} \frac{v_i w_j t}{q_j} x_{ij}(t) dt,$$

$$s.t. \sum_{i\in\mathcal{V}} \int_0^{d_M} x_{ij}(t)dt \ge \frac{q_j}{v}, \forall j \in \mathcal{J}(0),$$

$$\sum_{j\in\mathcal{J}(0)} x_{ij}(t) \le 1, \forall i \in \mathcal{V}, t \in [0, d_M],$$

$$x_{ij}(t) \ge 0, \forall i \in \mathcal{V}, j \in \mathcal{J}(0), t \in [0, d_M].$$

d_M denotes the maximum deadline of all VNFs. We regard the online VNF scheduling process as repeatedly solving P3 for currently available VNFs. The dual of P3 is given by

$$\text{P4}: \max \sum_{j\in\mathcal{J}(0)} \alpha_j - \sum_{i\in\mathcal{V}} \int_0^{d_M} \beta_t dt,$$

$$s.t.\ \frac{v_i}{q_j}\alpha_j \le \beta_{it} + \frac{v_i w_j}{q_j} t, \forall i \in \mathcal{V}, j \in \mathcal{J}(0), t \in [0, d_M],$$

$$\alpha_j, \beta_{it} \ge 0, \forall i \in \mathcal{V}, j \in \mathcal{J}(0), t \in [0, d_M].$$

The conclusions in (22)-(24) are utilized to design an effective assignment policy. Suppose we are addressing the assignment and scheduling of the first VNF j of an SFC at slot r_j and the existing VNFs with their remaining load are scheduled over the disjoint time intervals $\{I_l, l \in \mathcal{J}(r_j)\}$. The r.h.s of (24) is the upper bound of β, which is used to obtain a feasible estimate β'_{it} instead of the unknown value function. Since time is slotted, β'_{it} for virtual node i is calculated as

$$\beta'_{it} = \sum_{l\in\mathcal{J}(t)} \sum_{\tau=t}^{T} \frac{v_i w_l}{q_l} x_{il}(\tau), \tag{25}$$

where T is large enough to cover the scheduled intervals. Furthermore, based on (21), it can be inferred that we should make the constraints as tight as possible to maintain dual feasibility while achieving higher dual objective. Therefore, α_j should be feasible with respect to β'_{it} for any virtual node i. We set α_j to be equal to the minimum value among all the constraints that need to be satisfied, given by

$\min_{i,t>r_j}\{(\beta'_{it}+\frac{v_i w_j}{q_j}(t-r_j))\frac{q_j}{v_i}\}$. The virtual node to be assigned is chosen as

$$i' = \arg\min_{i\in\mathcal{V}}\{\min_{t>r_j}(\beta'_{it}+\frac{v_i w_j}{q_j}(t-r_j))\frac{q_j}{v_i}\}. \quad (26)$$

Once a VNF have been assigned, the HDF is optimal for scheduling in that node without latency constraints. Here, we apply the HDF to the existing VNFs on the nodes to generate a preliminary schedule until the VNF is completed, which helps update β'_{it} and assign the successor VNF in the SFC.

After completing the assignment of an SFC and determining the processing time for each VNF, it is possible to set a processing time margin for each VNF based on the latency constraint of the SFC. Specifically, the available processing time, $d_k^s - r_k^s$, is allocated proportionally to each VNF based on its individual processing time. The overall tolerance time of the SFC is also allocated to each VNF by using a deadline control parameter δ. The processing time is $p_j = \frac{q_j}{v_{i'}}$. The available processing time for a VNF is calculated as

$$\hat{p}_j = \frac{p_j}{\sum_{n\in\mathcal{J}_k} p_n}(d_k^s - r_k^s). \quad (27)$$

The preset arrival time $\hat{r}_j$ is calculated as $\hat{r}_j = \sum_{n=1}^{j-1}\hat{p}_n$ and the deadline of a VNF j is set as $d_j = \hat{r}_j + \hat{p}_j + \delta(d_k^s - r_k^s)$. Then the scheduling process on each virtual node is handled with the SNOS algorithm. The proposed MNOS procedure is summarized in Algorithm 2.

IV. Numerical Results

A. Settings

Consider a virtual network of 20 nodes and 10 types of VNF instances are deployed. Each type of VNF is implemented with two instances. The process rate of each VNF instance is uniformly chosen form $\{1, 2, 3\}$ Gbps. As for SFCs, the arrival of requests is modeled by a Poisson process with intensity $\lambda \in [0.05, 0.20]$. The length of SFC is an integer uniformly distributed over $[5, 15]$. The time slot length is 1 ms. The traffic size of each VNF is uniformly distributed over $[1, 10]$ Mbits. The maximum processing time of a VNF is calculated by processing it using the node with the minimum rate. Let η be the total maximum processing time of all VNFs belonging to an SFC. The deadline of the SFC is uniformly distributed between $[r_k^s+1.2\eta, r_k^s+1.8\eta]$. These parameters are generated independently for each SFC. The tolerance of accepting an SFC for the system is set to $\epsilon = 0.15$, which means an SFC is accepted by the system if it is completed within $r_k^s+\epsilon(d_k^s-r_k^s)$. The deadline control parameter is set to $\delta = 0$. Each online scheduling process lasts for 10000 slots.

Since no existing work has studied the problem while considering deterministic latency, we introduce two priority rule-based scheduling schemes. These schemes are composed of commonly used assignment and scheduling rules.

- Shortest processing time first and earliest deadline first (SPF-EDF): Assign the VNF to the virtual node that has the shortest total processing time for the already assigned VNFs and schedule the VNF of the SFC with the earliest deadline.
- Shortest processing time first and shortest slack first (SPF-SSF): Assign the VNF using the SPF rule and schedule the VNF with the shortest slack time. The slack time is calculated as $d_k^s - r_j - p_j$.

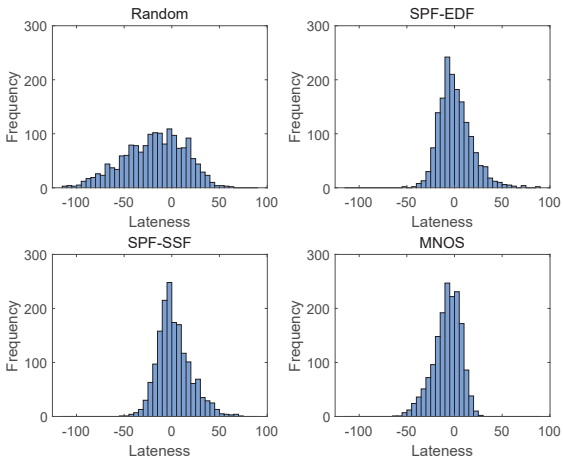

Fig. 2. Distribution of lateness in the online scheduling process of the MNOS algorithm.

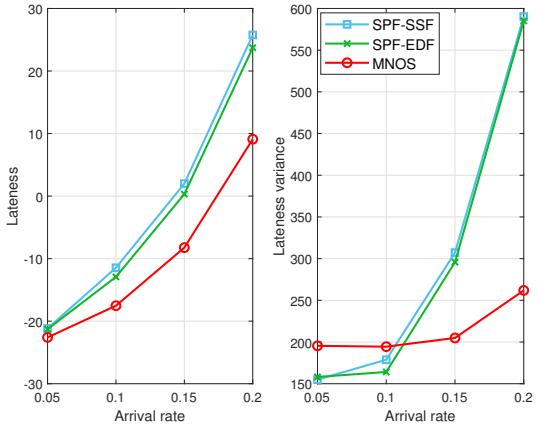

Fig. 3. Average lateness and lateness variance as a function of the arrival rate.

B. Online VNF Scheduling Performance

We define the lateness of an SFC as its completion time subtracted by the deadline, whose distribution implies the satisfaction of latency constraints and delay variation. The SFC failing to meet the delay constraints may be rejected by the system in practical scenarios, but we focus on the lateness and acceptance ratio to comprehensively demonstrate the performance. Fig. 2 shows the lateness distribution in an online VNF scheduling process of 10000 slots, in which the arrival rate of services is 0.15. It can be observed that both the SPF-EDF and the SPF-SSF, compared to the random algorithm, exhibit a higher concentration of lateness less than

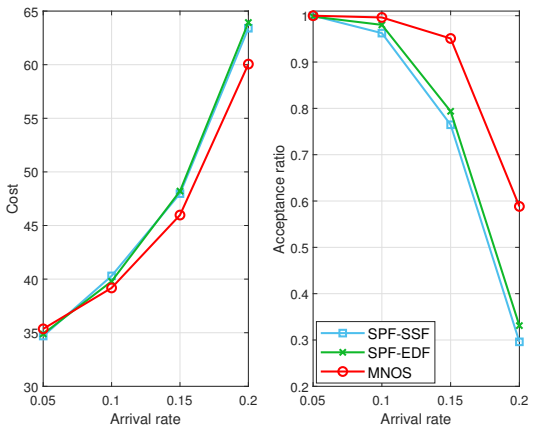

Fig. 4. Average cost and acceptance ratio as a function of the arrival rate.

0. This suggests that these algorithms effectively meet latency requirements. However, these algorithms also display a long-tail effect on latency, indicating their weak ability to guarantee strict latency satisfaction. In contrast, the proposed MNOS algorithm mitigates this issue by prioritizing the execution of VNFs with different deadlines through the characteristics of the EDF rule, thereby maintaining latency satisfaction as much as possible.

Fig. 3 shows the lateness and the lateness variance as the arrival rate increases, where each simulation last for 10000 slots. It can be seen that the MNOS algorithm achieves the lowest lateness and lateness variance when $\lambda \geq 0.15$. This indicates that the MNOS method effectively reduces both delay and delay variation under high service load. However, when the arrival rate is lower, the proposed MNOS algorithm exhibits higher variance. This is due to sufficient time being available to meet the delay requirements, causing the scheduling process prioritize minimizing the cost function with the HDF rule, resulting in higher variance. Moreover, as the arrival rate increases from 0.10 to 0.15, the lateness variance of the SPF-SSF and the SPF-EDF increase by 71.8% and 80.2%, respectively, while that of the MNOS only increases by 5.4%. Therefore, the MNOS is more robust than the others as the load increases.

Fig. 4 shows the cost and the acceptance ratio as the arrival rate increases. The MNOS algorithm achieves a lower cost and higher service acceptance ratio when $\lambda \geq 0.10$. This can be attributed to the SNOS scheduling process on each node, which minimizes objective values when there is no need to consider the delay constraints. In the online VNF scheduling process, a larger the amount of remaining load leads to a higher cost, which consumes more computing resources in subsequent scheduling processes. Optimizing the fractional completion time actually increases the available scope for scheduling subsequent SFCs and enhances the overall acceptance ratio. Consequently, the system becomes capable of handling scenarios with high arrival rates.

V. Conclusion

In this paper, we have investigated the online VNF scheduling problem in a deterministic network with the objective of minimizing fractional weighted completion time while providing guaranteed latency performance. We propose the MNOS algorithm, which utilizes the value function and dual variables to generate solutions for VNF assignment. In addition, the algorithm determines deadline for each VNF and uses the characteristics of the EDF rule to generate scheduling decisions, which enhances the overall determinism of the SFC completion by increasing the determinism of each VNF. The numerical results demonstrate that the proposed algorithm achieves lower delay and delay variation compared to the SPF-EDF and the SPF-SSF schemes. This effective mitigation of long-tail effects in latency distribution leads to improved system acceptance ratio.

References

[1] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 90–97, 2015.

[2] H. Hantouti, N. Benamar, and T. Taleb, "Service function chaining in 5G & beyond networks: Challenges and open research issues," *IEEE Netw.*, vol. 34, no. 4, pp. 320–327, 2020.

[3] N. Finn, P. Thubert, B. Varga, and J. Farkas, "Deterministic networking architecture," *RFC 8655*, 2019.

[4] H. Yu, T. Taleb, and J. Zhang, "Deterministic latency/jitter-aware service function chaining over beyond 5G edge fabric," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 3, pp. 2148–2162, 2022.

[5] Z. Tao and S. Wang, "Improved downlink rates for fdd massive mimo systems through bayesian neural networks-based channel prediction," *IEEE Trans. Wirel. Commun.*, vol. 21, no. 3, pp. 2122–2134, 2022.

[6] H. A. Alameddine, M. H. K. Tushar, and C. Assi, "Scheduling of low latency services in softwarized networks," *IEEE Trans. Cloud Comput.*, vol. 9, no. 3, pp. 1220–1235, 2021.

[7] A. Varasteh, B. Madiwalar, A. Van Bemten, W. Kellerer, and C. Mas-Machuca, "Holu: Power-aware and delay-constrained VNF placement and chaining," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 2, pp. 1524–1539, 2021.

[8] Z. Kuai, T. Wang, and S. Wang, "Fair virtual network function mapping and scheduling using proximal policy optimization," *IEEE Trans. Commun.*, vol. 70, no. 11, pp. 7434–7445, 2022.

[9] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and S. Davy, "Design and evaluation of algorithms for mapping and scheduling of virtual network functions," in *Proc. IEEE NetSoft'15*, London, UK, Apr. 2015.

[10] J. Li, W. Shi, P. Yang, and X. Shen, "On dynamic mapping and scheduling of service function chains in SDN/NFV-enabled networks," in *Proc. IEEE GLOBECOM'19*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.

[11] T. Wang and S. Wang, "Online convex optimization for efficient and robust inter-slice radio resource management," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 6050–6062, 2021.

[12] S. R. Etesami, "An optimal control framework for online job scheduling with general cost functions," *Oper. Res.*, vol. 70, no. 5, pp. 2674–2701, 2022.

[13] N. R. Devanur and Z. Huang, "Primal dual gives almost optimal energy-efficient online algorithms," in *Proc. ACM-SIAM SODA'14*, Portland, OR, USA, Jan. 2014, pp. 1123–1140.