A Distributed Policy Gradient Algorithm for Optimal SFC Deployment in NFV-Enabled STINs

Jing Wang
Department of Electrical Engineering
Illinois State University, Normal, IL 61790
jwang77@ilstu.edu

Khanh Pham
Air Force Research Laboratory
Kirtland AFB, NM, 87117
khanh.pham.1@spaceforce.mil

Xin Tian, Genshe Chen
Intelligent Fusion Technology, Inc
Germantown, MD 20874
xtian@intfusiontech.com
gchen@intfusiontech.com

***Abstract*—This paper investigates the dynamic mapping and scheduling of multiple service function chains (SFCs) within network function virtualization (NFV)-enabled, multi-layer satellite-terrestrial integrated networks (STINs). A new framework for dynamic deployment of SFCs is proposed. We first introduce a novel network traffic modeling approach that treats SFCs as dynamical systems, effectively capturing the time-varying characteristics of service demands and resource availability. The deployment of multiple SFCs across NFV server nodes is formulated as a constrained optimal resource allocation problem for these dynamical systems. To address the computational challenges and improve scalability in large-scale networks, we then propose a distributed policy gradient algorithm designed to solve the posed problem. We rigorously show that the proposed distributed algorithm is theoretically sound in determining the sub-optimal placements of virtual network functions (VNFs), while minimizing a global cost function that balances service provider expenditures with user quality of service (QoS). The proposed framework adds to the disposal of scalable and practical solutions for dynamic SFC orchestration in STINs.**

***Index Terms*—Network Function Virtualization (NFV), Service Function Chain (SFC), Network Traffic Model, Distributed Policy Gradient**

I. INTRODUCTION

The rapid evolution of 5G/6G networks and the growing demand for low-latency, high-reliability services have driven the integration of terrestrial and satellite networks into a unified and network function virtualization (NFV)-enabled Satellite-Terrestrial Integrated Network (STIN). Such networks promise global coverage, enhanced resilience, and improved quality of service (QoS) for diverse applications, including IoT, autonomous systems, and emergency communications. However, the inherent dynamics of STINs, such as time-varying topology, intermittent connectivity, and resource constraints, pose significant challenges for efficient service function chain (SFC) orchestration, where virtualized network functions (VNFs) must be dynamically deployed and interconnected to meet service requirements [1].

Satellite networks introduce unique challenges, such as frequent handovers, limited onboard processing, and heterogeneous resource distribution [2], [7]. Several studies have investigated the use of software-defined network (SDN)/NFV technologies in satellite networks. In [2], three scenarios for using SDN/NFV technologies in satellite networks were investigated in terms of virtualization within the satellite ground segment (satellite hubs, terminals, and networking equipments), of 4G/5G satellite backhauling services, and of satellite-terrestrial hybrid access services. In [7], a SDN/NFV based satellite networks framework was proposed, which consisted of five planes: management plane on ground for routing policy calculation, security and resource management, control plane on GEO layer for communicating between satellites and management plane, forwarding plane on MEO layer for searching flow tables to find the matching forwarding information, accessing plane on LEO, and service plane on satellites gateway for processing date from and to satellites. In [3], SDN/NFV-enabled satellite networks were studied by applying in different segments in a typical satcom network configuration including terrestrial gateways, satellites, and customer terminals.

Existing SFC mapping and scheduling approaches primarily focus on terrestrial networks, assuming relatively stable resource availability and predictable traffic patterns. However, these assumptions may not hold in STINs, where network conditions change rapidly due to satellite mobility, limited onboard resources, and fluctuating service demands. Moreover, most existing solutions rely on centralized optimization, which suffers from high computational complexity and poor scalability in large, dynamic environments. To address these challenges, we propose a new distributed framework that dynamically optimizes SFC deployment while balancing cost efficiency for service providers and QoS for end-users.

Traditional SFC mapping and scheduling solutions in terrestrial networks range from integer linear programming (ILP) to heuristic approaches [10]. However, these methods often assume static or semi-static network conditions, making them unsuitable for highly dynamic environments like STINs. Recent advances leverage machine Learning and reinforcement Learning for adaptive SFC deployment. For instance, a centralized reinforcement learning algorithm was proposed to solve the VNF scheduling problem with delay guarantees in [4]. In [9], a MDP model was introduced to capture dynamic network state transitions, and an adaptive, online, deep reinforcement learning approach was proposed to deploy SFCs. In [11], an adaptive interference-aware VNF deployment algorithms was proposed for 5G network slice. Cost-aware dynamic SFC mapping and scheduling was studied in [5]. A game-theoretical approach was proposed in [6] for multiple service function chains embedding. However, most those algorithms focus on centralized controllers, which may have issues with scalability in large-scale, distributed networks.

In this paper, we further tackle the challenge of dynamically deploying multiple service function chains (SFCs) in large-scale, multi-layer satellite-terrestrial integrated networks (STINs). We propose a scalable and tractable framework for optimizing SFC deployment, focusing on two core objectives: (1) optimal placement of virtual network functions (VNFs) on physical NFV nodes, and (2) dynamic allocation of server resources.

The main contributions of this paper can be summarized in three main points: (1) Dynamic network traffic modeling: each SFC is modeled as a dynamical system to capture real-time traffic fluctuations and resource demands. (2) Joint cost optimization: we formulate a novel cost function that simultaneously minimizes operational expenses for NFV providers while satisfying quality of service (QoS) requirements. (3) Data-driven distributed policy gradient method: to avoid explicit use of model of network dynamics, we propose a fully data-driven policy gradient algorithm. Each NFV node is equipped with a parameterized policy network, updated via gradient descent. A distributed estimation mechanism is integrated to compute global cost-related terms using only local information exchange among NFV nodes. The proposed framework adds to the disposal of scalable and practical solutions for dynamic SFC orchestration in STINs. While this paper rigorously demonstrates the theoretical soundness of the proposed distributed algorithm within its scope, comprehensive simulations and experimental validation are reserved for future work.

II. SFC-BASED NETWORK TRAFFIC MODEL

Consider a large scale STIN in which NFV and SDN technologies are being employed to reduce the operation cost and to improve the QoS for customers. In the design of NFV modules for STIN, one of the main problems is how to determine the deployment of SFCs onto different types of server nodes in STIN while considering the challenges due to long communication latency and mobility of satellite nodes.

We consider a multi-layered architecture of STIN as shown in Figure 1. NFV nodes are distributed across multiple layers, namely the Fog layer, Cloud layer, and Satellite layer. Generally speaking, nodes in the satellite layer exhibit characteristics such as extensive coverage range, relatively constrained resources, and prolonged latency. Nodes in the cloud layer, situated within data centers, have high-end storage and computing resources with comparatively longer latency. Communication in this layer occurs through base stations and access points. Nodes in the fog layer are characterized by more limited storage and computing resources but offer low-latency communication. They can establish communication through routers, switches, and gateways.

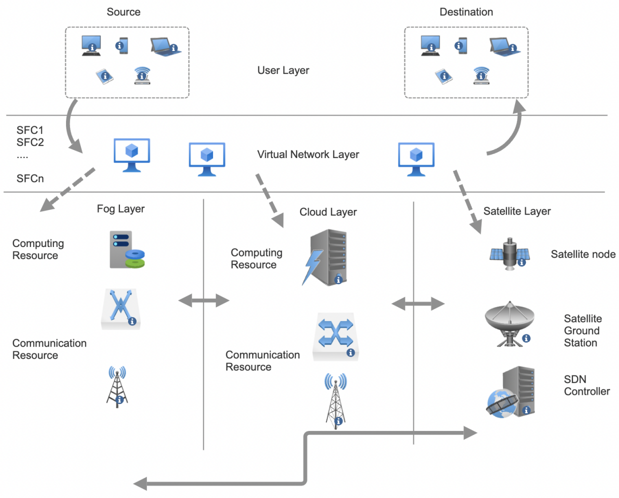

Fig. 1. A multi-layered architecture of STIN

We formulate the deployment problem of multiple SFCs under the framework of discrete-time dynamical systems. Particularly, we treat each SFC as a dynamical system, and adopt the optimal control strategy to solve the problem by considering the minimization of the overall cost functions in terms of deployment cost and throughput of all SFCs. The design objective is to reduce the cost of using server nodes in terms of resources and at the same time to improve the QoS by reducing the completion time of each service.

A. Notations and Symbols

We consider a discrete-time setting, where the deployment problem is solved dynamically by making decisions at discrete time steps $k = 0, 1, \cdots$. We assume that at the time instant k, there are a number of SFCs to be served, and each SFC consists of a sequence of VNFs.

The following notations and symbols are introduced: τ time interval from k to $k+1$; N_v: the number of NFV nodes; N_f: the max number of VNFs can be processed on each NFV node; N_s: the number of SFCs to be served in a batch; $c_i(k)$ the available CPU/memory capacity of NFV node i; C_i the max CPU/memory capacity of NFV node i; $b_{ij}(k)$ the available bandwidth capacity for the link between node i and node j; B_{ij} the max bandwidth; $\mathcal{N}_i$: the neighboring node set for NFV node i; $SFC_i(k) = \{x_{i1}, \cdots, x_{im_i}\}$ service function chain i, which contains m_i VNF components; x_{ij} the jth VNF in SFC_i, $j = 1, \cdots, m_i$; C_{xij} capacity demand by x_{ij}; B_{xij} bandwidth demand by x_{ij}; $u_i = [u_{i1}, \cdots, u_{im_i}]$ decision variables; u_{ij} takes value from discrete set $[1, \cdots, N_v]$, $u_{ij_1} \neq u_{ij_2}$.

B. SFC-Based Dynamic Traffic Model

State and Decision Variables: For each SFC_i, assume that there are m_i VNFs. Define the state variable x_i to represent the deployment status of VNF_{ij} (the jth VNF of SFC i), and let u_i be the decision variable for SFC_i. That is,

$$x_i = \begin{bmatrix} x_{i1} \\ \vdots \\ x_{im_i} \end{bmatrix}, \; u_i = \begin{bmatrix} u_{i1} \\ \vdots \\ u_{im_i} \end{bmatrix}$$

$$x_{ij} = \begin{cases} 0, & VNF_{ij} \text{ is waiting to be served} \\ 1, & VNF_{ij} \text{ is deployed} \\ -1, & VNF_{ij} \text{ service completed} \end{cases}$$

$$u_{ij} = \begin{cases} n, & VNF_{ij} \text{ is deployed on server node } n, \\ & n \in \{1, \cdots, N_v\} \\ 0, & VNF_{ij} \text{ is not deployed} \\ -1, & VNF_{ij} \text{ service completed} \end{cases}$$

State Transition Dynamics (State Equation): From stage k to $k+1$, the state transition dynamics for SFC_i satisfies

$$\begin{aligned} x_{i1}(k+1) &= \text{SGN}(u_{i1}(k)), \\ x_{ij}(k+1) &= \text{SGN}(u_{ij}(k)(1 - x_{i(j-1)}(k))|x_{i(j-1)}(k)|) \end{aligned}$$

where $j = 2, \cdots, m_i$, and $\text{SGN}(\cdot)$ is defined as

$$\text{SGN}(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x = 0 \\ -1, & \text{if } x < 0 \end{cases}$$

State Transition Dynamics (Output Equation): For NFV server node n, define $y_n = [y_{n1}, y_{n2}]^T$ with y_{n1} being the available CPU/memory resource on node n, and y_{n2} being the available output bandwidth on node n. It follows

$$y_{n1}(k) = C_n - \sum_{j=1}^{N_s} \sum_{l=1}^{m_j} \phi(u_{jl}(k) - n) c_{jl} \tag{1}$$

$$y_{n2}(k) = B_n - \sum_{j=1}^{N_s} \sum_{l=1}^{m_j} \phi(u_{jl}(k) - n) b_{jl} \tag{2}$$

where C_n is the max CPU/memory capacity of node n, B_n is the max output bandwidth of node n, c_{jl} is the capacity demand by VNF_{jl}, b_{jl} is the bandwidth demand by VNF_{jl}, and function $\phi(\cdot)$ is defined by

$$\phi(x - y) = \begin{cases} 1, & \text{if } x = y, x \geq 0 \\ 0, & \text{if } x \neq y, x \geq 0 \\ -1, & \text{if } x < 0 \end{cases} \tag{3}$$

Constraints: Node capacity constraint $y_{n1}(k) \geq 0, \forall n, \forall k$. Node bandwidth constraint $y_{n2}(k) \geq 0, \forall n, \forall k$.

Cost functions: The immediate cost of SFC_i is defined based on the operation cost of NFV nodes at time instant k as well as the time needed to complete the service. The operation cost of using NFV node n at time instant k is defined by

$$r_n(k) = \rho_{i1}(C_n - y_{n1}(k))^2 + \rho_{i2}(B_n - y_{n2}(k))^2 \tag{4}$$

where $\rho_{i1} > 0$ is the unit cost of using resource of NFV node n, and $\rho_{i2} > 0$ is the unit cost of bandwidth use for NFV node n. In terms of deployment and completion tasks for VNFs, the objective is to make $x_{ij}(k)$ become -1. The performance can be measured by

$$\sigma_i(k) = \sum_{j=1}^{m_i} (x_{ij}(k) + 1)^2 \tag{5}$$

End-to-end delay is measured by the total time used to process all SFC VNFs. While multiple VNFs can be deployed on the same node, we assume that at each time interval τ, there is only one VNF being processed. Assume that for VNF_{jl}, the time steps needed to process it is κ_{jl}. It follows that the projected processing time for completion of VNF_{ij} for SFC i can be defined by

$$\mathcal{T}_i(k) = \sum_{j=1}^{m_i} \left[\kappa_{ij} - \sum_{q=0}^{\kappa_{ij}-1} x_{ij}(k - q) \right] \tag{6}$$

The total cost for SFC_i starting at k with the decision making window size N_d

$$J_i(k) = \sum_{l=k}^{k+N_d} \sigma_i(l) + \sum_{l=k}^{\infty} \mathcal{T}_i(l)$$

The overall cost for all SFC

$$\mathbb{J}(k) = \sum_{i=1}^{N_s} J_i(k) + \sum_{l=k}^{k+N_d} \sum_{n=1}^{N_v} r_n(l)$$

For the ease of algorithm development, a modified cost function using penalty function method and softmax function to integrate constraints can be obtained as follows:

$$\begin{aligned} \mathbb{J}(k) = & \sum_{i=1}^{N_s} J_i(k) + \sum_{l=k}^{k+N_d} \sum_{n=1}^{N_v} r_n(l) \\ & + \rho_1 \sum_{l=k}^{k+N_d} \sum_{n=1}^{N_v} \log(1 + e^{-y_{n1}(l)}) \\ & + \rho_2 \sum_{l=k}^{k+N_d} \sum_{n=1}^{N_v} \log(1 + e^{-y_{n2}(l)}) \end{aligned} \tag{7}$$

where $\rho_1 > 0$ and $\rho_2 > 0$ are penalty coefficients to be chosen.

Remark 2.1: Note that the total number of NFV nodes $N_v = N_{vs} + N_{vc} + N_{vf}$, where N_{vs}: is the number of satellite nodes, N_{vc} is the number of cloud nodes, and N_{vf} is the number of fog nodes. It should be noted that while the proposed algorithm is developed for general NFV nodes, the following basic decision rules are also embedded in the use of the proposed algorithm. In other words, 1) If $\tau T_{jl} > \tau_{Ls}$, then VNF_{jl} can choose any nodes in three layers; 2) if $\tau_{Lc} < \tau T_{jl} < \tau_{Ls}$, then VNF_{jl} can choose any nodes in cloud nodes and fog nodes; 3) if $\tau T_{jl} < \tau_{Lc}$, then VNF_{jl} can choose any nodes in fog nodes, where τ_{Ls} is the communication latency on links and processing latency on satellite node, τ_{Lc} is the communication latency on links and processing latency on cloud node, τ_{Lf} is the communication latency on links and processing latency on fog node, $\tau_{Lf} < \tau_{Lc} < \tau_{Ls}$, and T_{jl} time steps to process VNF_{jl}.

III. A DISTRIBUTED POLICY GRADIENT ALGORITHM

In what follows, we propose a policy gradient method by directly learning the policy from optimizing the total cost without explicitly relying on network dynamics.

Each NFV node i employs a parametric probability density function (pdf) to define its policy, denoted by π_θ, where θ represents the set of learnable parameters. The function π_θ maps the current system state to a probability distribution over possible actions. For implementation, a policy network—illustrated in Figure 2—can be constructed using a deep neural network architecture. The policy network produces $\sum_{i=1}^{N_s} m_i(N_v+2)$ output values, corresponding to the action space across all N_s SFCs. For SFC i, the network generates $m_i(N_v+2)$ outputs, representing the probabilities of selecting each possible action for all VNFs. These probabilities are normalized to ensure they sum to 1, thereby forming a valid distribution over the action space for each SFC.

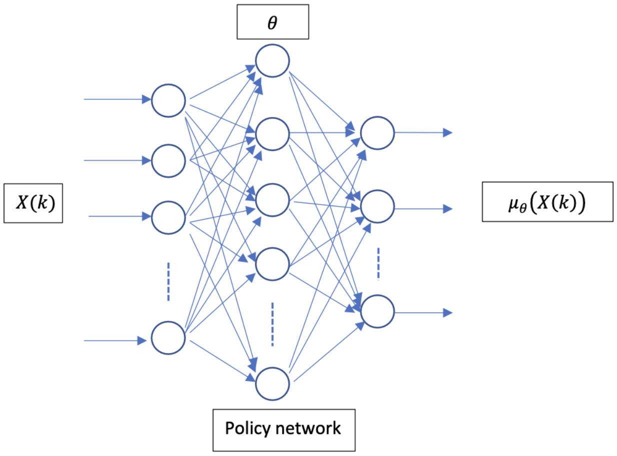

Fig. 2. A policy network producing the pdf π_θ

To this end, leveraging equation (7), the objective reduces to solving the following optimization problem:

$$\min_{\theta} \mathbb{E}_{\tau_\theta \sim \pi_\theta}[\mathbb{J}(k)] \tag{8}$$

where τ_θ is a trajectory obtained from sampling the policy.

The probability of the trajectory τ_θ can be defined as

$$P(\tau_\theta|\theta) = \prod_{l=k}^{k+N_d-1} p(x(l+1)|x(l),u(l))p(u(l)|\theta), \tag{9}$$

where x and u denote the stacked state and input vectors, respectively, encompassing all service function chains (SFCs), and for the deterministic system $p(x(l+1)|x(l),u(l))=1$ and $p(u(l)|\theta)$ is defined by

$$p(u(l)|\theta) \;=\; \prod_{i=1}^{N_s} p(u_i(l)|\theta), \tag{10}$$

It then follows

$$\begin{aligned}
\nabla_\theta \mathbb{J}(k) &= \nabla_\theta \int_{\tau_\theta} P(\tau_\theta|\theta)\mathbb{J}(k) \\
&= \int_{\tau_\theta} \nabla_\theta P(\tau_\theta|\theta)\mathbb{J}(k) \\
&= \int_{\tau_\theta} P(\tau_\theta|\theta)\nabla_\theta \log P(\tau_\theta|\theta)\mathbb{J}(k) \\
&= \mathbb{E}\left[\mathbb{J}(k) \sum_{l=k}^{k+N_d-1} \nabla_\theta \log p(u(l)|\theta(k))\right].
\end{aligned} \tag{11}$$

To this end, the policy parameter θ can be iteratively updated using a gradient descent approach. That is,

$$\theta(k+1) = \theta(k) - \alpha \nabla_\theta \mathbb{J}(k) \tag{12}$$

where $\alpha > 0$ is a learning rate.

Remark 3.1: In general, a centralized computing server can be used to implement the parameter update algorithm in (12), and then transmit the the deployment policy for SFCs to NFV nodes. However, to enhance robustness and scalability, we propose a distributed solution in which each NFV node locally computes the deployment policy for SFCs.

Let θ_i denote the policy parameters maintained by NFV node i. These parameters can be updated locally using a gradient descent algorithm as follows:

$$\theta_i(k+1) = \theta_i(k) - \alpha \nabla_{\theta_i} \mathbb{J}(k) \tag{13}$$

It should be noted that in the parameter updating algorithm (13), the global information $\mathbb{J}(k)$ is needed. To further reduce the communication load, in what follows, we present a distributed estimation algorithm for $\mathbb{J}(k)$ through local information exchange among NFV nodes. We use the following adjacency matrix to describe the information exchange topology among server nodes.

$$\mathcal{A}(k) = \left[a_{ij}(k)\right] \in \Re^{N_v \times N_v} \tag{14}$$

where $a_{ii}=0, a_{ij}(k)>0$ denotes that node i can receive the transmitted data from node j; otherwise, $a_{ij}(k)=0$. The neighboring set of agent i is defined as $\mathcal{N}_i \overset{\triangle}{=} \{j|a_{ij}>0\}$. Let $d_i = \sum_{j\neq i} a_{ij}$. Define the degree matrix $\mathcal{D} = \mathrm{diag}\{d_i\}$, and we have the corresponding Laplacian matrix

$$\mathcal{L} = \mathcal{D} - \mathcal{A} \tag{15}$$

Assumption 3.1: The information exchange topology defined in (14) is time-invariant and strongly connected.

Given the available information $\mathbb{J}_i(k)$ and $\mathbb{J}_j(k), j \in \mathcal{N}_i$ for NFV node i, $i=1,\cdots,N_v$, where

$$\begin{aligned}
\mathbb{J}_i(k) &= \sum_{l=k}^{k+N_d} r_i(l) + \rho_1 \sum_{l=k}^{k+N_d} \log(1+e^{-y_{i1}(l)}) \\
&\quad + \rho_2 \sum_{l=k}^{k+N_d} \log(1+e^{-y_{i2}(l)}) + \frac{\sum_{l=1}^{N_s} J_l(k)}{N_v},
\end{aligned} \tag{16}$$

we design a distributed estimation algorithm for the estimate of $\mathbb{J}(k)$ by NFV node i. Let $\hat{\mathbb{J}}_i(k)$ be the estimate $\mathbb{J}(k)$ by NFV node i. The proposed distributed estimation algorithm is given by

$$\begin{aligned}\hat{\mathbb{J}}_i(k+1) &= \hat{\mathbb{J}}_i(k) + \frac{1}{1+d_i}\sum_{j\in\mathcal{N}_i} a_{ij}(\hat{\mathbb{J}}_j(k) - \hat{\mathbb{J}}_i(k)) \\ &\quad + \frac{1}{w_{1i}}(\mathbb{J}_i(k+1) - \mathbb{J}_i(k))\end{aligned} \tag{17}$$

where $w_1 = [w_{11}, \cdots, w_{1N_v}]^T$ is the normalized left eigenvector of matrix $\mathcal{F} = I_{N_v} - (I_{N_v} + \mathcal{D})^{-1}\mathcal{L}$ corresponding to eigenvalue $\lambda_1 = 1$, and initial condition satisfies $\hat{\mathbb{J}}_i(0) = \frac{1}{w_{1i}}\mathbb{J}_i(0)$. Define $\hat{\mathbb{J}}(k) = [\hat{\mathbb{J}}_1(k), \cdots, \hat{\mathbb{J}}_{N_v}(k)]$ and $\tilde{\mathbb{J}}(k) = \hat{\mathbb{J}}(k) - \mathbf{1}\sum_{i=1}^{N_v}\mathbb{J}_i(k)$. It follows from (17) that

$$\tilde{\mathbb{J}}(k) = \mathcal{F} \otimes I_{N_v}\tilde{\mathbb{J}}(k-1) + \Delta\mathbb{J}(k) \tag{18}$$

where $\Delta\mathbb{J}(k)$ is a drift term induced by $\frac{1}{w_{1i}}(\mathbb{J}_i(k+1) - \mathbb{J}_i(k))$. The solution to (18) is

$$\tilde{\mathbb{J}}(k) = (\mathcal{F} \otimes I_{N_v})^k\tilde{\mathbb{J}}(0) + \sum_{m=0}^{k-1}(\mathcal{F} \otimes I_{N_v})^{k-m-1}\Delta\mathbb{J}(m+1)$$

Thus, following the strong connectivity assumption on communication topology $\mathcal{A}$ in Assumption 3.1, it can be established that [8]

$$\lim_{k\to\infty}\|\tilde{\mathbb{J}}(k)\| \le \epsilon$$

for some small constant ϵ. To this end, with the use of the estimate $\hat{\mathbb{J}}_i(k)$ in (17), the gradient in (11) becomes

$$\nabla_{\theta_i}\hat{\mathbb{J}}(k) = \mathbb{E}\left[\hat{\mathbb{J}}_i(k)\sum_{l=k}^{k+N_d-1}\nabla_{\theta_i}\log p(u(l)|\theta_i(k))\right], \tag{19}$$

and the algorithm in (13) becomes

$$\theta_i(k+1) = \theta_i(k) - \alpha\nabla_{\theta_i}\hat{\mathbb{J}}(k) \tag{20}$$

The proposed distributed policy gradient algorithm for the optimal deployment of multiple service function chains (SFCs) is outlined in Algorithm 1.

IV. CONCLUSION

In this paper, we presented a novel framework for the online deployment of multiple service function chains (SFCs) in dynamic STINs. By modeling each SFC as a discrete-time dynamical system, we formulated the SFC deployment problem as a constrained optimal control problem. We developed a distributed scalable policy gradient algorithm that enables adaptive optimization. The proposed solution features numerically efficient implementations, with computation complexity scaling linearly with the number of SFCs, making them particularly suitable for large-scale STIN deployments. While the theoretical soundness of the proposed distributed algorithm was rigorously demonstrated within its scope, comprehensive simulations and experimental validation will be done in the future work.

Algorithm 1 Distributed Policy Gradient Algorithm (for NFV node i)

```
Input: Learning rate $\alpha$, and network topology parameter $w_1$.
1. Initialization $\theta_i(0), \hat{J}_i(0)$
2. For node $i$, measure initial state $x_i(0)$ and neighbors $x_j(0)$
for time step $k = 0, 1, 2, \cdots$ do
    3. Sample the trajectory
    for $l = 0, \cdots, N_d - 1$ do
        3.1. Sample a control action $u(k+l)$ based on the output of policy network.
        3.2. Apply $u(k+l)$, and observe $x_i(k+l+1)$,$x_j(k+l+1)$, NFV node capacity and bandwidth, and the cost $r_i(k+l+1)$
    end for
    4. Calculate $\mathbb{J}_i(k)$ and $\hat{\mathbb{J}}_i(k)$ using (17)
    5. Update $\theta_i(k+1)$ using (20)
end for
```

ACKNOWLEDGEMENT

This material is partially based upon work supported by the United States Air Force under contract No. FA9453-23-P-A048. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and don't necessarily reflect the views of the AFRL, United States Air Force, or the U.S. Government.

REFERENCES

[1] M. Bonfim, K. L. Dias, and S. F. L. Fernandes, "Integrated nfv sdn architectures: a systematic literature review," *ACM computing surveys*, vol. 51, pp. 1–39, 2019.

[2] R. Ferrus, H. Koumaras, and et.al., "Sdn/nfv-enabled satellite communications networks: opportunities, scenarios and challenges," *Physical communication*, vol. 18, pp. 95–112, 2016.

[3] G. Gardikis, V. koumaras, and H. Koumaras, "Towards sdn/nfv-enabled satellite networks," in *Telecommunication systems*, Dec 2017.

[4] J. Li, W. Shi, N. Zhang, and X. Shen, "Delay-aware vnf scheduling: a reinforcement learning approach with variable action set," *IEEE Trans. on Cognitive communications and networking*, vol. 7, pp. 304–318, 2021.

[5] J. Li, W. Shi, H. Wu, S. Zhang, and X. Shen, "Cost-aware dynamic sfc mapping and scheduling in sdn/nfv-enabled space-air-ground integrated networks for internet of vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5824 – 5838, 2022.

[6] J. Li, W. Shi, Q. Ye, N. Zhang, W. Zhuang, and X. Shen, "Multiservice function chain embedding with delay guarantee: a game-theoretical approach," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11 219 –11 232, 2021.

[7] T. Li, H. Zhou, H. Luo, Q. Xu, and Y. Ye, "Using sdn and nfv to implement satellite communication networks," in *2016 Int. Conf. on Networking and Network Applications*.

[8] J. Wang, *Emergent Behavior Detection and Task Coordination for Multiagent Systems: A distributed estimation and control approach.* Switzerland AG: Springer Nature, 2022.

[9] Y. Xiao, Q. Zhang, F. Liu, J. Wang, M. Zhao, Z. Zhang, and J. Zhang, "Nfvdeep: adaptive online service function chain deployment with deep reinforcement learning," in *2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS)*, Phoenix, AZ, June 24-25 2019.

[10] B. Yi, X. Wang, K. Li, S. Das, and M. Huang, "A comprehensive survey of network function virtualization," *Computer Networks*, vol. 133, pp. 212–262, 2018.

[11] Q. Zhang, F. Liu, and C. Zeng, "Online adaptive interference-aware vnf deployment and migration for 5g network slice," *IEEE/ACM Transactions on Networking*, vol. 29, no. 5, pp. 2115 – 2128, 2021.