# A Sequential VNF Deployment Mechanism for Privacy-Preserving Multi-Domain SFC Deployment

Arif Indra Irawan*†, Yukinobu Fukushima*, Hideyoshi Miura‡, Tokumi Yokohira‡
*Faculty of Environmental, Life, Natural Science and Technology, Okayama University, Japan
†School of Electrical Engineering, Telkom University, Indonesia
‡Faculty of Interdisciplinary Science and Engineering in Health Systems, Okayama University, Japan
Email: p8e79mc7@s.okayama-u.ac.jp, fukusima@okayama-u.ac.jp, h-miura@okayama-u.ac.jp, yokohira@okayama-u.ac.jp

***Abstract*—Service function chaining (SFC), based on the network function virtualization (NFV) technology, is a promising candidate for realizing agile and elastic network service provisioning. This paper tackles a SFC deployment across multiple domains where only abstracted intra-domain resource information is available. For the multi-domain SFC deployment, a privacy preserving deployment mechanism called PPDM has been proposed. To conceal privacy of domains, PPDM abstracts intradomain resource information to a binary matrix, and adopts deep reinforcement learning (DRL) for flexible SFC deployment. However, because PPDM deploys all Virtual Network Functions (VNFs) included in a single SFC in a batch manner, substrate nodes need to redundantly reserve the resources required by the deployable VNFs until the construction of the SFC is completed. As a result, the deployment of the SFC may fail because of the lack of the resources on substrate nodes. In this paper, to improve acceptance ratio of SFC requests, we propose a privacy-preserving sequential deployment mechanism (PPSDM) for SFC deployment across multiple domains. The key idea behind PPSDM is to deploy VNFs in a single SFC one by one in a sequential manner to avoid redundant reservation of resources. Comprehensive simulations demonstrate that PPSDM achieves up to 22% higher acceptance ratio of SFC requests compared to the baseline approaches such as PPDM.**

***Index Terms*—Service Function Chain (SFC), Virtual Network Embedding (VNE), Virtual Network Function (VNF), Privacy Protection, Deep Q-network (DQN)**

## I. Introduction

The increasing diversity of applications have exposed the limitation of traditional network service provisioning, which is often rigid and static. These limitations hinder the ability of network providers to support dynamic and heterogeneous service demands, especially in evolving environments such as real-time communication services [1]. To address these challenges, Network Function Virtualization (NFV) [2] decouples network functions from dedicated hardware, allowing them to run as software on general-purpose infrastructure. This abstraction enables faster, more flexible, and cost-effective service provisioning. Building on NFV, Service Function Chaining (SFC) [3] allows traffic to traverse a specific sequence of Virtualized Network Functions (VNFs), enabling customizable service paths based on application types, network condition, or policy requirements.

While SFC provides agile and elastic service provisioning, its practical realization depends heavily on network virtualization technologies. These technologies decouple networks from physical infrastructure allowing Infrastructure Providers (InPs) to operate as Virtual Network Operators (VNOs) or Service Providers (SPs) without owning the substrate network [2]. In single-domain SFC deployment scenarios, SPs typically have a global view of the substrate network, enabling them to optimize objectives such as acceptance ratio of SFC requests [3], [4]. However, this assumption no longer holds in modern, large-scale service provisioning where SFCs often span across multiple administrative domains. In such multi-domain SFC deployment scenarios, each domain operates under its own administrative policies and typically does not share its internal topology or resource status due to security, privacy, and business concerns [5]. To manage these complexities, a hierarchical SFC orchestration model is commonly used, consisting of a centralized Multi-Domain Controller (MDC), multiple Intra-Domain Controllers (IDC), and open interfaces that expose limited domain information to the MDC [6].

Building on the hierarchical SFC orchestration model, Cai *et al.* [7] proposes a privacy-preserving deployment mechanism (PPDM) for SFC deployment across multiple domains. To conceal privacy information such as topology and resource information within a domain, PPDM abstracts resource information of substrate nodes within

This work was supported by JSPS KAKENHI Grant Number JP23K11065

a domain into a binary matrix called SIRM (Service Intention Response Matrix) where ($i$, $j$) element means whether $i$th VNF can be deployed to $j$th substrate node or not. In addition, to enable flexible SFC deployment, PPDM adopts deep reinforcement learning (DRL) to VNF deployment. However, because PPDM deploys all VNFs included in a single SFC in a batch manner, substrate nodes need to redundantly reserve the resources required by the deployable VNFs until the construction of the SFC is completed despite whether those VNFs are actually deployed or not (The situation is called "virtual occupation"). As a result, the deployment of the SFC may fail because of the lack of the resources on substrate nodes.

In this paper, to improve acceptance ratio of SFC requests, we propose a privacy-preserving sequential deployment mechanism (PPSDM) for SFC deployment across multiple domains. The key idea behind PPSDM is to deploy VNFs in a single SFC one by one in a sequential manner. In PPSDM, every time the location (i.e., substrate node) of the current VNF is determined, MDC sends the result of VNF deployment to each IDC. Then, each IDC can immediately release the virtually occupied resource required by the VNF on the non-selected substrate nodes while it assigns the virtually occupied resource to the VNF on the selected substrate node. We expect this sequential deployment mechanism avoids virtual occupation of resources on substrate nodes and consequently improves acceptance ratio of SFC requests.

The remainder of this paper is organized as follows. Section II introduces related work on multi-domain SFC deployment. Section III explains the models for substrate network and SFC, and the multi-domain SFC deployment problem tackled in this paper. Section IV explains the limitation of PPDM, and proposes a privacy-preserving sequential deployment mechanism. Section V shows the simulation results. Finally, Section VI concludes this paper.

## II. Related Work

This section reviews the most relevant approaches in multi-domain SFC deployment, emphasizing their trade-offs in scalability, privacy, and performance. *Zhang et al.* [8] applied Particle Swarm Optimization (PSO) to the SFC deployment problem, which improves search efficiency but requires detailed topology and resource information to be shared across domain posing significant risks to privacy and scalability. Yu *et al.* [9] formulated the SFC deployment problem using Mixed-Integer Linear Programming (MILP), achieving high deployment accuracy; however the method incurs high computational complexity and suffers from poor scalability, making it impractical for large-scale or real-time SFC deployment.

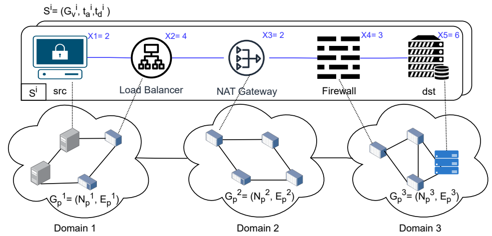



Fig. 1: Substrate network model and SFC model

To address the limitations of centralized orchestration, Chen et al. [10] proposed a distributed federated service chaining (DFSC) framework. Their model enables each domain to make independent deployment decisions based solely on inter-domain path and border node information, thus preserving administrative boundaries and reducing the need for global topology exposure. Nonetheless, this distributed approach requires non-trivial coordination among orchestrators and may suffer from performance degradation due to fragmented decision-making and constrained inter-domain visibility.

Lin *et al.* [11] introduced a column generation-based technique that effectively balances deployment cost and privacy preservation. Despite these benefits, the approach suffers from high computational latency as network size grows, limiting its use in dynamic or large-scale scenarios. To address these concerns, Cai *et al.* [7] proposed the PPDM, which combines a SIRM with DQN to make informed deployment decisions without exposing internal domain details. Although PPDM enhances the acceptance ratio and respects privacy constraints, it relies heavily on accurate node resource prediction from IDC and involves virtual occupation of resources that potentially leading to resource inefficiencies. In light of these limitations, We propose a sequential VNF deployment strategy to avoid virtual occupation of resources and improve resource efficiency, while maintaining domain privacy.

## III. Multi-domain SFC deployment problem

### A. Models for Substrate Network and SFC

Fig. 1 depicts a substrate network consisting of multiple domains and SFCs deployed on the substrate network.

The substrate network comprises $K$ domains and is modeled as an undirected graph $G_p = (N_p, E_p)$ where $N_p$ denotes the set of substrate nodes and $E_p$ denotes the set of substrate links. A subset of the links $E_{\text{inter}} \subset E_p$ represents inter-domain links. Each domain $k \in \{1, \ldots, K\}$ is represented by a subgraph $G_p^k = (N_p^k, E_p^k)$. Each substrate node $n_j^k \in N_p^k$ has an available resource capacity denoted as $X^a_{n_j^k}$ and a maximum capacity $X^m_{n_j^k}$. Each intra- or inter-domain link is associated with a propagation delay.

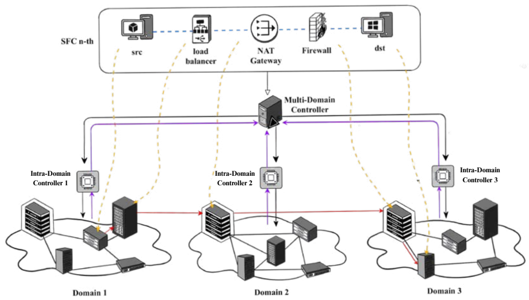


Fig. 2: Hierarchical architecture for multi-domain SFC deployment

The set of requested SFCs are modeled as a set $\mathcal{S}$ where $l$th requested SFC $s^l \in \mathcal{S}$ is represented as a tuple $s^l = \langle G_v^l, t_a^l, t_d^l \rangle$. Here $G_v^l = (N_v^l, E_v^l)$ denotes the topology of SFC $s^l$, with $N_v^l$ representing the set of VNFs in SFC $s^l$ and $E_v^l$ the set of virtual links in SFC $s^l$. $t_a^l$ and $t_d^l$ are the request arrival time and the departure time of SFC $s^l$, respectively. Each VNF $v_i^l \in N_v^l$ requires a computational resource $x_{v_i^l}$ (e.g., CPU). As illustrated in Fig. 1, a requested SFC $s^l$ may span across multiple domains, and each VNF must be deployed to a suitable substrate node while respecting resource constraints.

## B. Problem Formulation

In the multi-domain SFC deployment problem tackled in this paper, we deploy VNFs to substrate nodes and virtual links to substrate paths so that the objective metrics are optimized while satisfying the constraints.

As the objective metrics, we adopt 1) acceptance ratio of SFC requests and 2) end-to-end delay of SFCs. The acceptance ratio of SFC requests is the ratio of the number of successfully deployed SFCs to the total number of requested SFCs. The end-to-end delay of a SFC is the total latency experienced by the SFC, and is calculated as the sum of processing delays of all the VNFs and propagation delays of all the virtual links in the SFC.

Solutions for the SFC deployment problem must satisfy the following three constraints to ensure valid and conflict-free deployment of VNFs and virtual links. Let $\phi_{n_j^k}^{v_i^l} \in \{0, 1\}$ be a binary variable indicating whether VNF $v_i^l$ is deployed to substrate node $n_j^k$, and let $\varphi_p^{v_i^l, v_j^l} \in \{0, 1\}$ be a binary variable indicating whether the virtual link between VNFs $v_i^l$ and $v_j^l$ is deployed to substrate path $p$.

a. *Node Capacity Constraint*— For any substrate node, the sum of node resources requested by the VNFs deployed to the substrate node must not exceed its maximum capacity.

$$\sum_{s_l \in \mathcal{S}} \sum_{v_i^l \in N_v^l} x_{v_i^l} \cdot \phi_{n_j^k}^{v_i^l} \leq X_{n_j^k}^m, \quad \forall n_j^k \in N_p \tag{1}$$

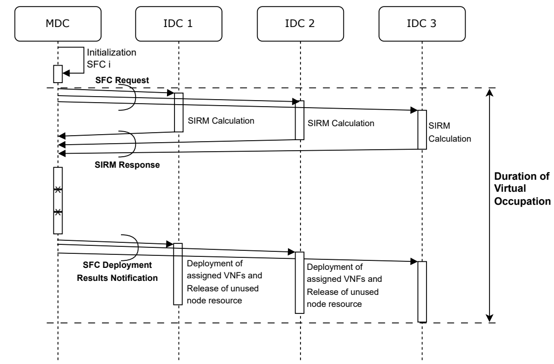


Fig. 3: Timing diagram of the SFC deployment procedure by PPDM

b. *Node Mapping Constraint*— Each VNF must be deployed to at most one substrate node to avoid duplication.

$$\sum_{n_j^k \in N_p} \phi_{n_j^k}^{v_i^l} \leq 1, \quad \forall v_i^l \in N_v^l, \forall s_l \in \mathcal{S} \tag{2}$$

c. *Link Mapping Constraint*— Each virtual link must be deployed to at most one substrate path that connects both ends of the virtual link.

$$\sum_{p \in P_{n(v_i^l), n(v_j^l)}} \varphi_p^{v_i^l, v_j^l} \leq 1, \quad \forall (v_i^l, v_j^l) \in E_v^l, \forall s_l \in \mathcal{S} \tag{3}$$

Here $P_{i,j}$ is the set of substrate paths between substrate nodes $i$ and $j$, and $n(v_i^l)$ is the substrate node to which VNF $v_i^l$ is deployed.

# IV. Proposed Method

## A. Limitiations of PPDM

Privacy-Preserving Deployment Mechanism (PPDM) [7] was proposed for multi-domain SFC deployment. PPDM employs a hierarchical architecture (Fig. 2) consisting of two main layers: a centralized Multi-Domain Controller (MDC) and multiple Intra-Domain Controllers (IDCs). Each IDC operates independently within a specific administrative domain and maintains full knowledge of its local substrate network. The MDC, typically operated by SP, coordinates the end-to-end SFC deployment across domains. To preserve privacy, detailed substrate-level information is not shared with the MDC; instead, abstracted resource information called Service Intention Response Matrix (SIRM) is shared with MDC. SIRM is a binary matrix where ($i$, $j$) element ($\xi_j^i$) means whether $i$th VNF in the current SFC can be deployed to $j$th substrate node or not.

Figure 3 depicts the timing diagram of the SFC deployment procedure by PPDM. When the MDC receives an SFC request, the MDC sends the SFC request to each

IDC. Each IDC then calculates the elements ($\xi_j^i$s) of SIRM for the substrate nodes within the domain. If $i$th VNF in the current SFC can be deployed to $j$th substrate node in the domain, $\xi_j^i$ is set to one, otherwise zero. Each IDC returns the elements of SIRM as for the substrate nodes within the domain to the MDC. Finally, the MDC gives the received SIRM to the DQN agent as input, and obtains the substrate nodes to accommodate the VNFs in the current SFC. As for the virtual links in the current SFC, MDC deploys them to the shortest substrate paths.

Because PPDM deploys all the VNFs in the current SFC in a batch manner, each IDC has to judge whether each substrate node in its domain can accommodate each of the VNFs in the current SFC before knowing the result of the SFC deployment. Therefore, when an IDC judges the substrate node can accommodate the VNF, the substrate node has to reserve the node resource requested by the VNF until the SFC deployment is completed (The situation is called "virtual occupation"). Due to the virtual occupation, the available resource capacity of substrate node $n_j^k$ when judging whether substrate node $n_j^k$ can accommodate $i$th virtual node $v_i^l$ in the current SFC is calculated as follows (The second term in the RHS corresponds to the virtually occupied resource).

$$X_{n_j^k}^a = X_{n_j^k}^m - \sum_{v_m^l \in N_d} \xi_{n_j^k}^{v_m^l} \cdot x_{v_m^l} - \sum_{v_q^o \in N_c} \phi_{n_j^k}^{v_q^o} \cdot x_{v_q^o} \quad (4)$$

where $N_d$ is the set of virtual nodes in the current SFC whose deployability to substrate node $n_j^k$ have already been judged and $N_c$ is the set of virtual nodes whose SFCs have already been deployed and operated on the substrate network. Although PPDM can mitigate virtual occupancy by limiting the ratio of the substrate nodes that return $\xi_j^i$=1 to top $K\%$ in terms of available resources, virtual occupancy still occurs on the top $K\%$ substrate nodes. As a result of virtual occupation, much node resources are virtually occupied at many substrate nodes, and consequently failures of SFC deployment are more likely to happen because of the lack of node resources.

### B. Sequential VNF Deployment Policy

To cope with virtual occupation in PPDM and improve acceptance ratio of SFC requests, we propose a privacy-preserving sequential deployment mechanism (PPSDM) for SFC deployment across multiple domains. The key idea behind PPSDM is to deploy VNFs in the current SFC one by one in a sequential manner to avoid virtual occupation.

Figure 4 depicts the timing diagram of the SFC deployment procedure by PPSDM. In PPSDM, every time the location (i.e., substrate node) of the current VNF is determined, MDC sends the result of VNF deployment to each IDC. Then, each IDC can immediately release

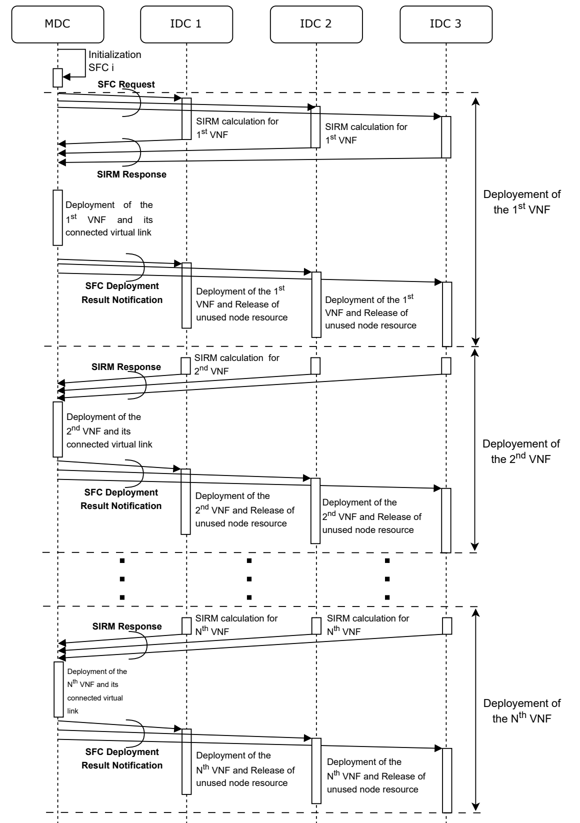



Fig. 4: Timing diagram of the SFC deployment procedure by PPSDM

the virtually occupied resource required by the VNF on the non-selected substrate nodes while it assigns the virtually occupied resource to the VNF on the selected substrate node. With this sequential approach, the available resource capacity of substrate node $n_j^k$ when judging whether substrate node $n_j^k$ can accommodate $i$th virtual node $v_i^l$ in the current SFC is calculated as follows.

$$X_{n_j^k}^a = X_{n_j^k}^m - \sum_{v_m^l \in N_d} \phi_{n_j^k}^{v_m^l} \cdot \xi_{n_j^k}^{v_m^l} \cdot x_{v_m^l} - \sum_{v_q^o \in N_c} \phi_{n_j^k}^{v_q^o} \cdot x_{v_q^o} \quad (5)$$

Please note that the second term in the RHS is reduced from the virtually occupied resource to the actually occupied one.

In PPSDM, after receiving the elements ($\xi_j^i$s) of SIRM for the current VNF from IDCs, the MDC gives the received SIRM to the DQN agent as input, and obtains the substrate node to accommodate the current VNF. In addition, we also deploy the virtual links, the locations (i.e., substrate nodes) of whose both ends are determined this time, to the shortest substrate paths. For the DQN agent, we define state, action and reward as follows. The state is defined as the elements ($\xi_j^i$s) of SIRM for the

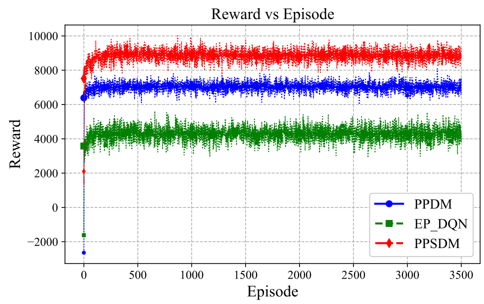



Fig. 5: Cumulative reward as a function of episodes

current VNF to reflect the resource availability of the substrate nodes. The action is defined as selecting the substrate node to accommodate the current VNF. As well as PPDM, the reward is defined as follows.

$$r(t) = r(\delta|s,a) + \alpha \cdot [\rho_{PN}(t)] + \beta \cdot [O_{end}^{Ds}(t)] \cdot r(\rho|s,a) \tag{6}$$

where (1) $r(\delta|s,a)$, (2) $\rho_{PN}(t)$, (3) $O_{end}^{Ds}(t)$ and (4) $r(\rho|s,a)$ are the rewards for selecting a substrate node (1) with $\xi_j^i = 1$, (2) with higher resource utilization, (3) with shorter end-to-end delay, and (4) for avoiding repeating the same choice, respectively.

## V. Experiments and Results

### A. Simulation Setup

We evaluate the performance of PPSDM using the AARNET topology from the Internet Topology Zoo [12] as the substrate network model, which represents a real-world network comprising 19 node from Australian universities. This topology is divided into three administrative domains: Domain 1 includes nodes 0–6, Domain 2 includes nodes 7–13, and Domain 3 include nodes 14–18. Each substrate node is initialized with a resource capacity randomly generated using uniform distribution from the range of 7 to 19 units.

TABLE I: Link Delay Settings [7]

| Domain | Domain 1 | Domain 2 | Domain 3 |
|---|---|---|---|
| **Domain 1** | 5–10 ms | 11–20 ms | 21–30 ms |
| **Domain 2** | 11–20 ms | 5–10 ms | 11–20 ms |
| **Domain 3** | 21–30 ms | 11–20 ms | 5–10 ms |

To reflect realistic latency conditions in the substrate node, link delays are configured as shown in Table I. Diagonal entries indicate intra-domain delays, while off-diagonal values represent inter-domain delays. Inter-domain delays are intentionally assigned higher values to simulate inter-domain communication overhead, consistent with the configuration used in [7]. SFC requests are generated using a Markov Modulated Poisson Process (MMPP), which models bursty and time-varying traffic patterns. The generator alternates between two state with mean inter-arrival times of 12 and 8 ms, transitioning between states every 10 ms with a 5% switching probability. Each SFC consists of 3 to 6 sequentially connected VNFs. Each VNF is assigned a resource demand between 2 and 8 units and a processing delay between 5 and 10 ms.

The proposed method is benchmarked against baseline methods: PPDM and EP-DQN using top $K = 30\%$ candidate substrate nodes. Each model undergo training for 3,500 episodes, with each episode comprising 300 randomly generated SFC requests. SFC requests are characterized by an inter-arrival of 6 and a burst inter-arrival of 4. During training, the reward function utilizes weighted parameters $\alpha = 0.8$, $\beta = 0.2$ along with a learning rate of 0.005.The reward term $r(\delta|s,a)$ is defined within the range $[-10, 10]$, while the term $r(\rho|s,a)$ ranges from $-1$ to 1. Following training, the resulting DQN model is employed to evaluate performance under various inter-arrival times of SFC requests.

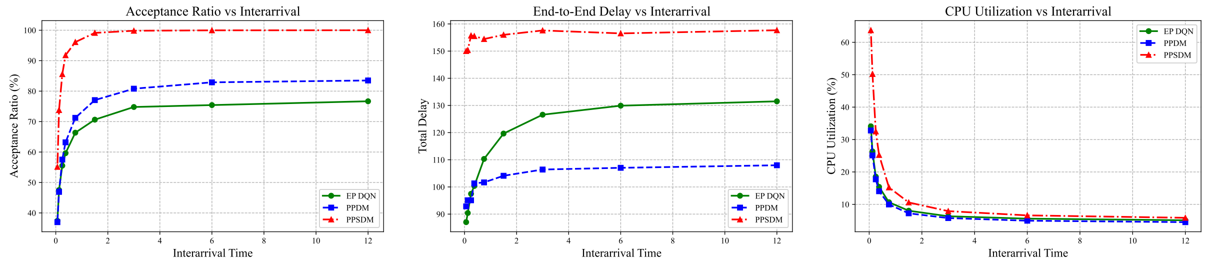
### B. Training Results

Figure 5 shows the cumulative reward of PPSDM, PPDM and EP_DQN during the training phase. All the methods demonstrate rapid convergence, reaching a stable performance level within the first 200 episodes. This fast stabilization indicates that the agents effectively captures the substrate network environment and successfully learn a robust deployment strategy. PPSDM consistently achieves higher cumulative rewards compared to PPDM and EP_DQN, validating the effectiveness of the sequential deployment strategy.

### C. Performance Evaluation under Various Inter-Arrival Times

To evaluate the performance of PPSDM, the trained model for each benchmark is tested under bursty and time-varying SFC requests, similar to the training setup. In this evaluation, the bursty SFC request is configured to be 50% more intense than normal SFC request. The testing scenarios simulate a range of traffic intensities, starting from high traffic (inter-arrival times of 0.0625, 0.125, 0.25, 0.375, and 0.75) to lower traffic conditions (inter-arrival times of 1.5, 3, 6, and 12). Performance is assessed based on three key metrics: acceptance ratio of SFC request, end-to-end delay, and node resource utilization.

As shown in Fig. 6a, PPSDM consistently outperforms both EP-DQN and PPDM across all traffic intensities, particularly under high-load conditions. Specifically, PPSDM achieves average 22% higher acceptance ratios in high-traffic scenarios. This improvement is attributed to the method's sequential deployment mechanism, which provides a clearer and more accurate state representation. In contrast, both EP-DQN and PPDM




(a) Acceptance Ratio (b) Average Delay (c) Resource Utilization

Fig. 6: Performance comparison under varying inter-arrival times.

tend to generate SIRMs with predominantly zero values due to excessive virtual resource occupation.

Consistent with the findings in [7], PPDM achieves an average of 4% higher acceptance ratio than EP-DQN when trained under high-traffic conditions.

With respect to end-to-end delay, as shown in Fig. 6b. PPDM and EP-DQN outperform PPSDM across all traffic testing condition. On average, the delay difference between PPSDM and EP-DQN exceeds 44 ms, and more than 53 ms compared to PPDM. This increased latency in PPSDM is primarily due to the no procedure for limiting $K$ top candidate substrate node; while this design choice enhances acceptance ratio, it comes at the cost of longer path selections, thereby increasing overall delay.

Regarding resource utilization shown in Fig. 6c, the proposed PPSDM achieves the highest node-level utilization—more than twice that of the benchmark methods—driven by its superior acceptance ratio. Both PPDM and EP-DQN show resource utilization trends that closely follow their acceptance ratios, indicating a strong correlation between acceptance success and substrate node utilization. The figure also highlights that, despite their lower acceptance ratio under high-traffic conditions, both PPDM and EP-DQN exhibit relatively low substrate node utilization. This inefficiency can be attributed to persistent virtual occupancy, which blocks resources without successful embedding, as discussed previously.

## VI. Conclusion

This paper presented a novel privacy-preserving multi-domain Service Function Chain (SFC) deployment approach using a sequential Virtual Network Function (VNF) deployment strategy optimized by Deep Q-Network (DQN) reinforcement learning. Comprehensive simulations demonstrate that the proposed method significantly outperforms baseline approaches such as EP-DQN and PPDM. Notably, it achieves up to 22% higher acceptance ratios under high-traffic scenarios.

Our future work includes the improvement of the proposed method to lower end-to-end delay of SFCs.

## Acknowledgment

This work was supported by JSPS KAKENHI Grant Number JP23K11065.

## References

[1] J. Xu, X. Cao, Q. Duan, and S. Li, "Service function chain deployment using deep q learning and tidal mechanism," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 5401–5416, 2024.

[2] Z. Yan, J. Ge, Y. Wu, L. Li, and T. Li, "Automatic virtual network embedding: A deep reinforcement learning approach with graph convolutional networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1040–1057, 2020.

[3] H.-K. Lim, I. Ullah, Y.-H. Han, and S.-Y. Kim, "Reinforcement learning-based virtual network embedding: A comprehensive survey," *ICT Express*, vol. 9, no. 5, pp. 983–994, Oct. 2023.

[4] T. Wang, L. Shen, Q. Fan, T. Xu, T. Liu, and H. Xiong, "Joint admission control and resource allocation of virtual network embedding via hierarchical deep reinforcement learning," *IEEE Transactions on Services Computing*, vol. 17, no. 3, pp. 1001–1015, 2024.

[5] D. Dietrich, A. Rizk, and P. Papadimitriou, "Multi-provider virtual network embedding with limited information disclosure," *IEEE Transactions on Network and Service Management*, vol. 12, no. 2, pp. 188–201, 2015.

[6] N. Toumi, M. Bagaa, and A. Ksentini, "Hierarchical multi-agent deep reinforcement learning for sfc placement on multiple domains," in *2021 IEEE 46th Conference on Local Computer Networks (LCN)*, 2021, pp. 299–304.

[7] J. Cai, Z. Zhou, Z. Huang, W. Dai, and F. R. Yu, "Privacy-preserving deployment mechanism for service function chains across multiple domains," *IEEE Transactions on Network and Service Management*, vol. 21, no. 1, pp. 1241–1256, 2024.

[8] P. Zhang, C. Wang, Z. Qin, and H. Cao, "A multidomain virtual network embedding algorithm based on multiobjective optimization for internet of drones architecture in industry 4.0," *Softw. Pract. Exp.*, vol. 52, no. 3, pp. 710–728, Mar. 2022.

[9] H. Yu, T. Wen, H. Di, V. Anand, and L. Li, "Cost efficient virtual network mapping across multiple domains with joint intra-domain and inter- domain mapping," *Optical Switching and Networking*, vol. 14, pp. 233–240, 2014.

[10] C. Chen, L. Nagel, L. Cui, and F. P. Tso, "Distributed federated service chaining: A scalable and cost-aware approach for multi-domain networks," *Computer Networks*, vol. 212, p. 109044, 2022.

[11] R. Lin, S. Yu, S. Luo, X. Zhang, J. Wang, and M. Zukerman, "Column generation based service function chaining embedding in multi-domain networks," *IEEE Transactions on Cloud Computing*, vol. 11, no. 1, pp. 185–199, 2023.

[12] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, 2011.