Service Function Chain Embedding for NFV-Enabled IoT Based on Deep Reinforcement Learning

Xiaoyuan Fu, F. Richard Yu, Jingyu Wang, Qi Qi, and Jianxin Liao

Recently, network Function Virtualization has attracted attention because of its prospect to achieve efficient resource management for IoT. In NFV-enabled IoT infrastructure, a SFC is composed of an ordered set of Virtual Network Functions that are connected based on the business logic of service providers. However, the inefficiency of SFC embedding process is one major problem due to the dynamic nature of IoT networks and the abundance of IoT terminals.

Abstract

It is challenging to efficiently manage different resources in the IoT. Recently, Network function virtualization has attracted attention because of its prospect to achieve efficient resource management for IoT. In NFV-enabled IoT infrastructure, a service function chain (SFC) is composed of an ordered set of virtual network functions (VNFs) that are connected based on the business logic of service providers. However, the inefficiency of the SFC embedding process is one major problem due to the dynamic nature of IoT networks and the abundance of IoT terminals. In this article, we decompose the complex VNFs into smaller VNF components (VNFCs) to make more effective decisions since VNF nodes and physical network devices are usually heterogeneous. In addition, a deep reinforcement learning (DRL)-based scheme with experience replay and target network is proposed as a solution that can efficiently handle complex and dynamic SFC embedding scenarios. Simulation results present the efficient performance of the proposed DRL-based dynamic SFC embedding scheme.

Introduction

With the devices of the Internet of Things (IoT) expected to reach an extent of 40 billion before long, IoT systems could be ultra-dense and multivariate [1, 2]. Consequently, efficient and dynamic resource management is critical for the performance of IoT systems. In recent years, network function virtualization (NFV) has been regarded as a prospective network framework to efficiently and dynamically manage the resources in IoT systems [3]. NFV transfers network functions from dedicated hardware devices to software-based network nodes. In the NFV framework, an ordered combination of virtual network function (VNF) instances and their logical connections comprise a service function chain (SFC) that can be embedded in physical networks. User traffic flows through multiple VNFs according to the established application policies.

Several SFC embedding solutions have been proposed in the literature. In [4], mixed integer linear programming is formulated to solve a combinatorial problem related to SFC embedding. The authors of [5] proposed a group mapping scheme with dependency perception to satisfy network service requirements. A heuristic-based algorithm was proposed in [6] to make VNFs' forwarding graph embedding more tractable.

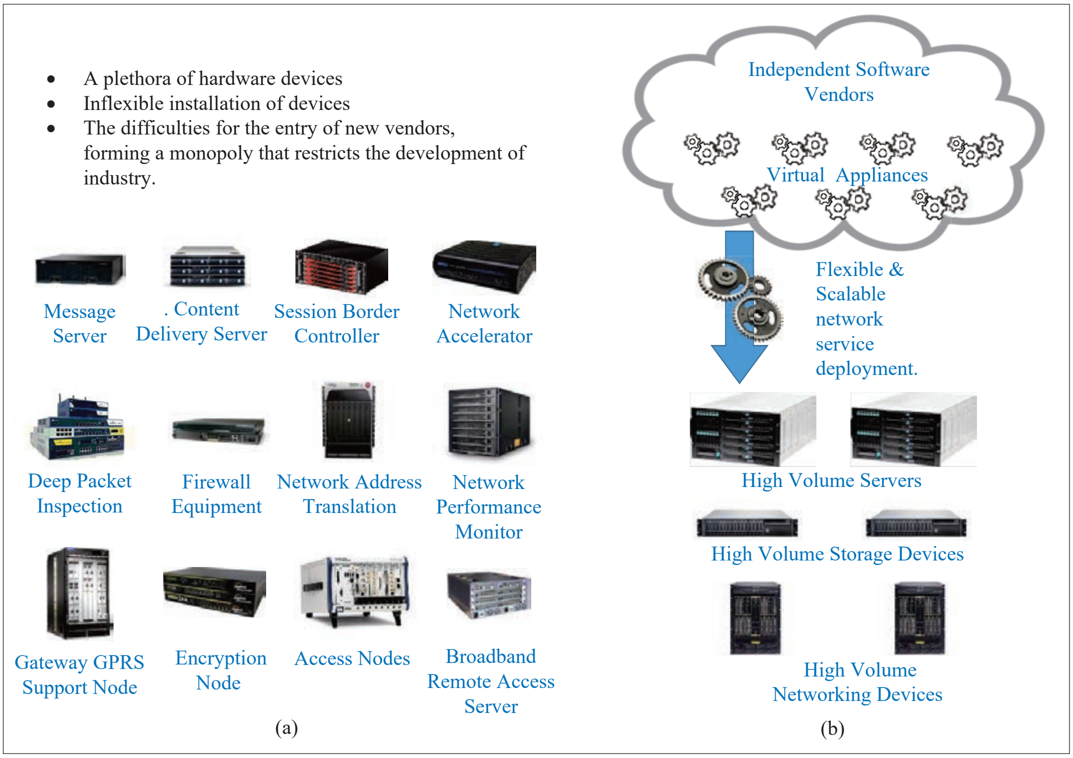
Although several excellent works have been done to address the SFC embedding problem in generic NFV frameworks, it is challenging to perform SFC embedding in NFV-enabled IoT due to the very large number of IoT terminals (users or devices). The enormous number of IoT terminals cause the dynamic nature of IoT networks and produce abundant network traffic. Communications among these heterogeneous entities in large-scale dynamic environments will produce massive high-speed and real-time data flows of services [3]. It is essential that NFV-enabled IoT infrastructures should evolve to become more intelligent to deal with SFC embedding when there are massive service requests that are extremely diverse in nature.

In this article, with current attention to deep reinforcement learning (DRL) [7], we present a DRL-based SFC embedding scheme in NFV-enabled IoT. DRL is able to input high-dimensional data and output an optimal policy for the learning agent. With more and more applications of machine learning in daily life, DRL methods have been successfully used to solve computing [8, 9], gaming [10], and resource management problems [11], among others. In this work, DRL is used to deal with the SFC embedding in NFV-enabled IoT, which has not been well studied in previous works. Our main contributions are summarized in the following:

- To solve the SFC embedding problem in NFV-enabled IoT systemically, we decompose the complex VNFs into a set of VNF components (VNFCs) and an internal connection graph to formulate the VNF forward graph (VNF-FG). The VNF-FG makes it easier to allocate substrate resources to network services.
- We present a dynamic SFC embedding scheme that improves the performance of IoT applications and services. Unlike most existing works, which assume static network conditions and IoT traffic [4, 5, 6], we consider the dynamic nature of network conditions and IoT traffic, which are more realistic in practical IoT systems [12].

Digital Object Identifier:
10.1109/MCOM.001.1900097

Xiaoyuan Fu, Jingyu Wang, Qi Qi, and Jianxin Liao are with Beijing University of Posts and Telecommunications; F. Richard Yu is with Carleton University. The corresponding authors are F. Richard Yu and Jingyu Wang.



- A plethora of hardware devices
- Inflexible installation of devices
- The difficulties for the entry of new vendors, forming a monopoly that restricts the development of industry.

Message Server
. Content Delivery Server
Session Border Controller
Network Accelerator
Deep Packet Inspection
Firewall Equipment
Network Address Translation
Network Performance Monitor
Gateway GPRS Support Node
Encryption Node
Access Nodes
Broadband Remote Access Server
(a)
Independent Software Vendors
Virtual Appliances
Flexible & Scalable network service deployment.
High Volume Servers
High Volume Storage Devices
High Volume Networking Devices
(b)

Figure 1. Network functions implementation with: a) conventional method; b) network function virtualization.

The NFV platform has dynamic scalability and it can realize fine-grained traffic control and efficient resource utilization. As a result, NFV technology has attracted more and more attention due to its increased agility of networks and significant reductions in expenditures.

- When the dynamic nature is considered, the IoT system becomes very complex, and it is difficult to use traditional optimization solutions to the SFC embedding problem. Therefore, the SFC embedding problem is formulated as a DRL problem, which can efficiently handle complex and dynamic scenarios.
- We illustrate the corresponding environment state, actions, and rewards of the formulated DRL model. In order to enhance the convergence performance of the proposed scheme, we use both experience reply [13] and target network [14], which are mechanisms in DRL to disrupt the correlation of learning experiences to enhance the algorithm performance.
- We use Google TensorFlow to realize the proposed DRL approach. The simulation results present the efficient performance of our proposed dynamic SFC embedding scheme.

This article is organized as follows. NFV-enabled IoT is described in the following section. Then we describe the system model and problem formulation. The proposed DRL-based approach is presented next. Following that, we discuss the simulation results. In the final section, we conclude the article.

System Description

In this section, we introduce NFV, followed by the SFC of NFV in IoT scenarios. Next, we introduce the SFC embedding process in the NFV-enabled IoT framework.

Network Function Virtualization

In traditional networks, the network functions use dedicated hardware, which brings lots of inconvenience. One inconvenience is that the location of the network topology and the service provider is fixed, so modifying network services means modifying the network topology and configurations. Therefore, there are enormous investment and operational costs for deploying and upgrading physical infrastructures. NFV is a novel network function realization design that promises to solve the problem of continual growth of devices for providing different network services. NFV takes advantage of virtualizing network functions that are implemented in specific hardware equipment and transferring the network functions to software-based devices. Figure 1 presents the implementation of network functions with the traditional network method and NFV.

NFV is considered a promising framework to accomplish network services. It provides the possibility of optimizing network resource efficiency and reducing network costs. It realizes flexible deployment of network functions and rapid delivery of network services. The NFV platform has dynamic scalability, and it can realize fine-grained traffic control and efficient resource utilization. As a result, NFV technology has attracted more and more attention due to its increased agility of networks and significant reductions in expenditures.

Service Function Chain of Network Function Virtualization in IoT Scenarios

In the NFV platform, an ordered chain of VNFs comprises a network service. The chain of VNFs is formed and deployed with a number of VNFs, the functionally and logically related order of VNFs in the chain, and the embedding of the chain in the substrate network. The chain of VNFs for realizing network services is named SFC in the NFV platform. In IoT scenarios, SFCs can be flexibly used to compose a sequence of heterogeneous VNF

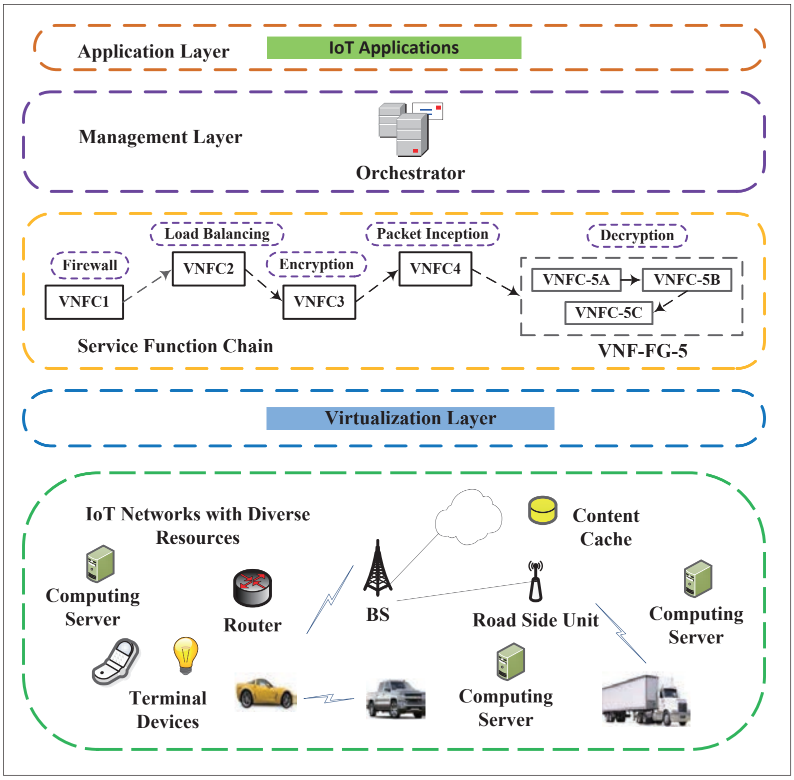


Figure 2. Service function chain embedding in the NFV-enabled IoT framework.

nodes to deal with the expected large number of data flows. The SFCs in the NFV platform will filter and compress the massive data flows from multiform IoT terminals. Then the processed data flows are sent to the cloud, which takes advantage of existing computing and bandwidth resources, and enhances the availability and reliability of network services. Therefore, SFCs in the NFV framework are able to implement network services both efficiently and economically for applications in IoT scenarios.

To improve the utilization of substrate resources and the SFC request reception rate, we can split complex VNFs into several sub-VNFs. In this article, we consider that complex VNFs can be represented by a set of VNFCs and their internal connection links. The set of VNFCs and the internal connection links of a VNF form a VNF-FG. The VNF-FG is the refined representation of a complex VNF, which makes the SFC embedding process easy to deploy under the proposed DRL approach. Each VNFC corresponds to a virtual processing node, and each internal connection link corresponds to a set of physical links, requiring certain link resources (e.g., bandwidth and quality of service [QoS]). The purpose of DRL-based SEC embedding in this work is to deploy VNFCs in the newly formed SFC topology to the substrate networks to satisfy the QoS. It is also necessary to mention that we do not consider how to split VNF and virtual links.

Service Function Chain Embedding in the NFV-Enabled IoT Framework

In the NFV-enabled IoT framework, different types of IoT services are accomplished by different kinds of SFCs. The allocation of virtual machines and other resources for the SFCs is accomplished by the management orchestration domain of the NFV architecture. Each request of SFC has two layers. One is the logical business layer, and the other is the execution embedding layer. The latter is to deploy the SFC request to the physical facilities through the process of SFC embedding.

Figure 2 presents the deployment of a network service {*Firewall* → *Load Balancing* → *Encryption* → *Packet Inspection* → *Decryption*} in the NFV-enabled IoT framework. We use this end-to-end SFC that contains different kinds of VNFs, including a complex VNF, as an instance to explain the embedding process in this work. The SFC embedding process is concerned with four entities: orchestrator, network service chain, virtualization layer, and IoT networks. The orchestrator is responsible for the orchestration between the network service chain and the virtualized physical layer to actualize network services. Physical resources are virtualized through the virtualization layer to better deploy the services. In SFC embedding, the orchestrator realizes an optimization algorithm that makes embedding decisions. To support the embedding process, the physical network layer generally contains computing, memory, and connection facilities that support processing, memory, and interconnection between VNFCs.

Problem Formulation

In this part, we describe our SFC embedding model with DRL. We describe the SFC embedding model first, and then present how to model it as a DRL task.

System Model

We consider the allocation of chained VNFs of different services to the NFV infrastructure, which can be described as VNF-FG embedding. The problem is to get an efficient strategy to allocate the VNFCs in the NFV-enabled IoT framework, considering different requested network services. We assume that the embedding order of VNFCs is done in a round-robin manner between multiple SFCs, and the embedding order of VNFCs within an SFC depends on the processing logic of the service. In the NFV framework, the requests of network services arrive at the physical network layer in discrete time steps, and VNFCs of a service chain will be allocated to virtual underlying nodes with appropriate resources. We consider an underlying network with multiple resources. In particular, the resource request model for VNFCs is described according to the data arrival rate and flow change ratio of the network service chain, both of which vary over time. In addition, our model assumes immutable embedding results after the end of the SFC embedding in the NFV-enabled IoT environment.

Therefore, the response of an SFC embedding process in the NFV-enabled IoT environment at time $t + 1$ to the action taken at time t depends on the VNFCs embedding that has happened earlier. We consider that the SFC embedding problem has the finite Markov property, which is defined with the transition probabilities from one state to another. A problem with this property is known as a Markov decision process (MDP), which can naturally be formulated as a DRL problem.

The transition probabilities are very hard to get directly in practical networks due to the dynamics of the SFC embedding environment in IoT scenarios. However, this dynamic property of an NFV-enabled IoT environment can be exploited by the DRL agent through learning from the transition of states. The DRL agent gets an expected long-term reward given the learned experience, current state, and action through all the learning steps. In this model, the transition probabilities and the expected value of the next reward completely represent the significant features for the dynamics in the SFCs embedding model.

Therefore, the SFC embedding in complex IoT scenarios can be formulated as a DRL problem. In order to get the best embedding strategy, it is essential to illustrate the state space, the action, and the reward in our DRL model, which are described in the following.

System State and Action

We represent the state space of the SFCs embedding environment by the states of current available resources in the physical layer and the available bandwidth of underlying links. Generally, the quantities of available resources will be dynamic and vary with the dynamics of complex IoT systems. To make things easy to deal with, we assume that the required resources of different VNFCs can from a normal distribution. We sample these resources at different time points to represent the resource consuming state of the substrate nodes. In this DRL model built for the SFC embedding problem, we use the current quantities of different types of resources to form a vector to represent the current state. The central management agent faces abundance and dynamics of cluster states, including the resource states of all substrate nodes.

As mentioned above, the agent has to decide which physical nodes are assigned to VNFCs of a requested service chain. We consider there are N physical nodes in the substrate platform, and we can easily get the underlying topological structure of them, including the available resources information and the links between them. We make the DRL agent execute one action at each time step, $action \in \{1, 2, ..., N\}$. The action space is determined by the number of physical nodes. In the SFC embedding process of this article, an agent allocates one VNFC at each time step, and the embedding process ends after the execution of the final VNFC of the requested service chain. In each valid decision, one VNFC is embedded in a physical node; then the current state is transferred to the next. Once the current allocation of a VNFC is completed, time proceeds, and the next VNFC of the service chain will be allocated.

Reward Function

In our dynamic SFC embedding scheme based on DRL, the total delay of the SFC is used as the reward metric, including processing delay of VNFCs on physical nodes and the sum of their transmission delays on physical links in the NFV infrastructure. Considering the simplicity of the model, we only consider the transmission delay of VNFCs in our learning algorithm. The processing delay is the time taken to process an SFC on the network nodes. We adopt the definition of processing delay model based on a common processor sharing algorithm. It is worth mentioning that other objectives, including energy saving, load balancing, fault tolerance, and so on, could also be used for the reward determination in the DRL task.

In this work, we consider that the physical path of VNFCs is achieved by the shortest path first protocol, and user traffic through VNFCs does not get rerouted because of changes in load. Once the VNFCs are placed, a connection stays on the same path until it expires or is removed. Generally, the goal of an agent is to get the maximum long-term reward, which aims to maximize the cumulative payback for a long time. We use the reward of each time step to direct the agent toward the solution to our goal, minimizing the average VNFC processing delay of SFCs. The reward at each time step is in inverse proportion to the processing delay of the VNFC. Therefore, the variation trend of the long-term reward is the same with the inverse change of service processing time.

Deep Reinforcement Learning Approach

In this section, we present deep Q-learning (DQL), and introduce the two key mechanisms in DRL to disrupt the correlation of learning experiences.

Q-learning is an off-policy temporal difference algorithm in RL. Before DQL, RL is shown to be unstable, especially with neural networks as the approximation of action value function. DQL makes many significant contributions to stabilize the learning process of DRL. It is necessary to mention the replay of learning experience and target network. In addition, DQL has emerged as a model-free DRL approach, which can train agile neural networks with the same learning approach and different network structures to make it compliant to diverse application areas.

To put it simply, DQL has a memory bank for pre-learned experiences. Q-learning is a kind of offline learning method, which can learn from current experience, past experience, and even the experience of others. Therefore, the learning experiences may be highly time-dependent. The learning agent always needs to solve a possibly long-span time dependence of learning experiences. In some cases, the effect of an action in an environment could materialize only when the learning agent has experienced enough state transitions, which is called temporal assignment of DQL.

Experience replay is to learn from a random selection of previous experiences in each update step of a deep-Q network. Its main contribution is that it overcomes the problem of correlation (correlated data) and non-stationary distribution of the empirical data. The correlation of continuous samples makes the variance of parameters update larger, which can be reduced by this mechanism. Random selection disrupts the correlation between experiences and makes neural network update more efficient. In addition, it can improve the utilization rate of the experience data because a sample can be used multiple times.

The target network is also a mechanism for disrupting the correlation of experiences. If we use a target network, we use two neural networks with the same structure but different parameters

In our dynamic SFC embedding scheme based on DRL, the total delay of the SFC is used as the reward metric, including processing delay of VNFCs on physical nodes and the sum of their transmission delays on physical links in the NFV infrastructure. Considering the simplicity of the model, we only consider the transmission delay of VNFCs in our learning algorithm.

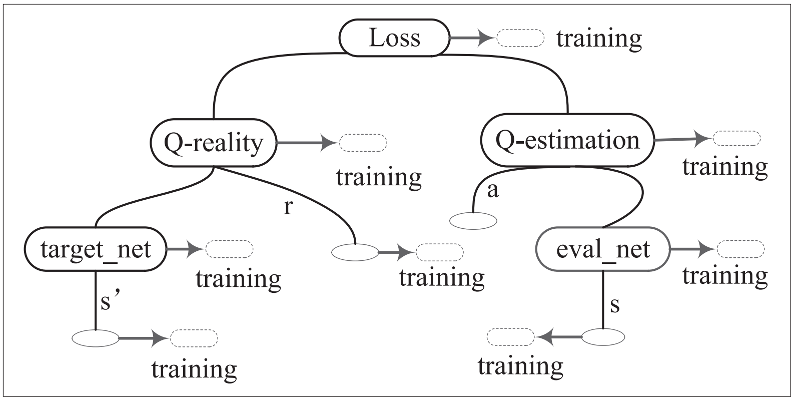


Figure 3. The visualized training graph of a target network.

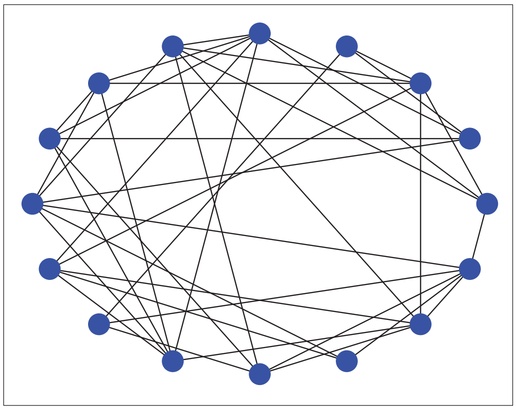


Figure 4. A random network used in our simulation with 16 nodes (N = 16) and connectivity probability of 0.3 (p = 0.3).

in deep Q-networks. Figure 3 shows the training process of the fixed Q-target mechanism. The neural network that predicts Q-estimation is called eval_net and it will have the latest parameters. The neural network that predicts Q-reality is called target_net, and the parameters of this network are from a period of time ago. These two neural networks are designed to fix the parameters of one neural network (i.e., target_net. This set of parameters is fixed for a while when the eval_net is replaced by new parameters. The eval_net is constantly being promoted. The utilization of the target network surpasses natural Q-Learning as a result of increasing difference between the update time of Q-reality network and the update time of Q-estimation network. This mechanism effectively reduces the oscillations of training process.

Therefore, we make use of DQL for our SFC embedding problem formulated in this article in complex and dynamic IoT scenarios.

Simulation Results and Discussions

In this section, we briefly introduce the simulation settings and the used software. Then we discuss the results.

Simulation Settings

For simulating practical networks with diverse characteristics, the SFC embedding needs to be conducted on different types of network topologies. It is necessary to mention that the actual features of network topologies are not completely understood now as the substrate networks can be complex and diverse. Therefore, we choose three different topologies to simulate the performance of our proposed scheme, including the random network, the BA scale-free network, and the small-world network. Simulation results for these network topologies have the same changing trend with the increasing of the number of edges in the substrate network. Thus, considering the applicability and generality of our SFC embedding algorithm, we use a typical substrate network topology, random network, to present our simulation results.

Generally, a random network can denote many physical networks, for example, Internet service provider (ISP) networks. It can be considered as the simulation topology because of its universality. The generation of a random network is in relation to the connectivity probability between two network nodes. The connectivity probability for a random network depends on the distance between a couple of network nodes.

Figure 4 is the simulation topology of a random network when there are 16 nodes. We consider the time-varying feature of current workloads and resource quantities. All of the resource demands of service requests and available resources of IoT nodes are simulated by linear functions of Gaussian function curves. The variable of them is determined by the random samples of these linear functions. Moreover, we use a deep neural network (DNN) in the proposed approach, which includes three fully connected layers. Each layer has 64 neural nodes.

With the above settings, we implement the proposed DRL-based dynamic SFC embedding scheme in NFV-enabled IoT scenarios. We use TensorFlow [15] to implement deep learning and NetworkX to simulate the substrate networks of the NFV-enabled IoT infrastructure in our simulations. In NetworkX, a random network with N nodes and connectivity probability p can be generated by using the *random_graphs.erdos_renyi_graph*(N, p) method.

Results and Discussions

In the simulations, we assume that there are two SFCs in the NFV-enabled platform, and they are both composed of four different VNFCs. First, the performance of the proposed dynamic scheme is tested by our simulations, and the results are compared to different scenarios, respectively, in the substrate topology mentioned above. For the performance comparison of dynamic and static scenarios, we choose the following static resource allocation schemes to compare to the dynamic one:

- The static resource allocation scheme
- The proposed scheme with static computing resource allocation
- The proposed scheme with static bandwidth allocation

Figure 5 shows reward values in the scenarios that are mentioned above, when the connectivity probability of the random network changes from p = 0.1 to p = 0.5.

From Fig. 5, we can observe that the characteristics of network topology have effects on the quality of service. We can find that the rewards of all the allocation schemes increase with the increase of connectivity probability p before p = 0.3. This is due to the increase in connectivity probability, resulting in an increase in the number of edges in the substrate network. The increased edges bring more bandwidth resources so that the agent can get a better choice for the best reward. After the peak of the reward value when $p = 0.3$, there is a decreasing trend that is also due to the increase of edges. It increases the possibility of sharing resources between VNFCs, resulting in more delays and fewer rewards.

The dynamic resource allocation scheme can always get the best reward over the other three cases. This is because the resource demands of user services in an IoT-dedicated network can always be dynamic. It is very likely that the resource requirements of the services are varying with time. They may have higher resource requirements at some times and low resource requirements at other times. The dynamic scheme can take both the high demand case and the low demand case into consideration to make the resource management more efficient and reduce the total delay of the services.

However, the static resource allocation scheme possibly stays in the high-resource-consuming case in peak times and has to request more resources from the substrate platform. This will make the strategy decision of SFC embedding management node tend to waste more resources during the whole service, and result in high delay of network services. In addition, we make simulations in the static CPU allocation scenario and the static bandwidth allocation scenario for the proposed scheme. In these cases, we have dynamic allocation of other resources that will change with time according to the dynamic environment. We can find from the figure that the rewards of these scenarios in each step are both lower than the dynamic scenario. These simulation results clearly show the favorable performance of SFC embedding based on the DQL approach in dynamic IoT scenarios.

Then we make some simulations with different learning rates to study the convergence performance of our DRL-based algorithm. Figure 6 shows rewards of the proposed DRL-based SFC embedding scheme with different learning rates, 0.1, 0.05, 0.01, and 0.001. As we can see, the learning rate affects the value of learning rewards during the training steps of our algorithm. The reason is that the learning rate represents the learning step length of realizing the convergence of the reward function. A large learning rate might miss the global optimum of the learning process. A small learning rate may lead to slow learning speed so that more steps are necessary to achieve the global optimum. According to our simulation results, learning rate 0.05 has the best performance. Its learning speed is acceptable, and it can quickly lead to the convergence of the reward function. The computational efficiency of the proposed scheme can also be seen in this figure.

Conclusion and Future Work

In this article, a DRL-based SFC embedding scheme with the recent advances in DRL is proposed as a solution to deal with dynamic and complex IoT scenarios. To formulate the SFC embedding problem as a DRL model, we decomposed some complex VNFs into a set of VNFCs and an internal connection graph to formulate the VNF-FGs in the NFV-enabled IoT framework, which can lead to more effective decisions. Simulation results show the efficient performance of the proposed dynamic SFC embedding scheme. In the future, we will consider implementing the proposed DRL-based algorithms in realistic NFV-enabled IoT platforms.

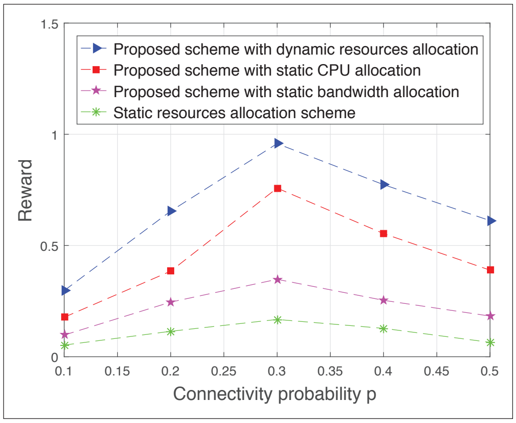


Figure 5. The rewards of different schemes in random network topologies with different node connectivity probabilities.

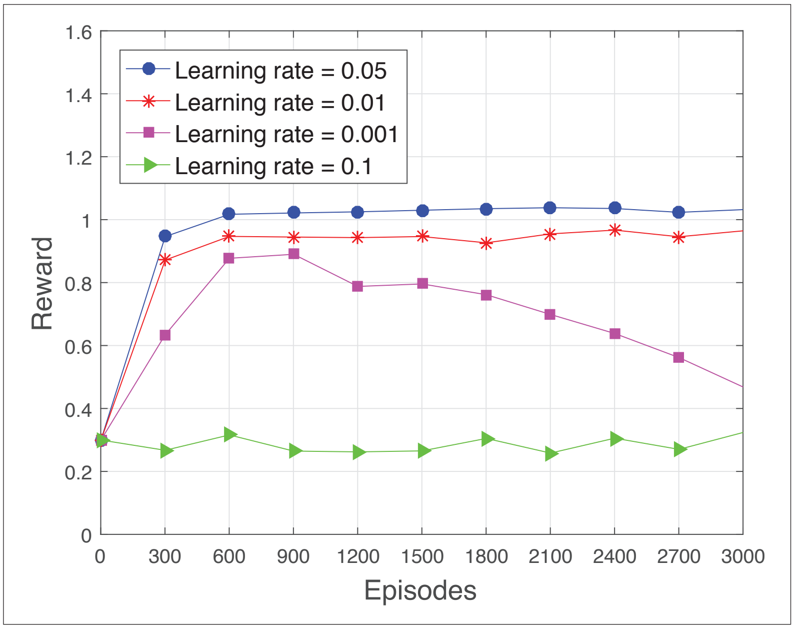


Figure 6. The training rewards of the proposed DRL-based SFC embedding scheme with different learning rates.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 61771068 and Grant 61671079, the Beijing Municipal Natural Science Foundation under Grant 4182041, the Fundamental Research Funds for the Central Universities under Grant 2018RC20, and the Ph.D. Student Short-Term Exchange Program of Beijing University of Posts and Telecommunications.

References

[1] M. Liu *et al.*, "Performance Optimization for Blockchain-Enabled Industrial Internet of Things (IIoT) Systems: A Deep Reinforcement Learning Approach," *IEEE Trans. Industrial Informatics*, vol. 15, no. 6, June 2019, pp. 3559–70.

[2] J. Chen *et al.*, "A Parallel Random Forest Algorithm for Big Data in a Spark Cloud Computing Environment," *IEEE Trans. Parallel and Distributed Systems*, vol. 28, no. 4, Apr. 2017, pp. 919–33.

[3] I. Farris *et al.*, "A Survey on Emerging SDN and NFV Security Mechanisms for IoT Systems," *IEEE Commun. Surveys & Tutorials*, vol. 21, no. 1, First Quarter 2019, pp. 812–37.

[4] M. A. T. Nejad *et al.*, "vSPACE: VNF Simultaneous Placement, Admission Control and Embedding," *IEEE JSAC*, vol. 36, no. 3, Mar. 2018, pp. 542–57.

[5] M. Jalalitabar *et al.*, "Embedding Dependence-Aware Service Function Chains," *IEEE/OSA J. Optical Commun. and Networking*, vol. 10, no. 8, Aug. 2018, pp. 64–74.

[6] L. Wang *et al.*, "Joint Optimization of Service Function Chaining and Resource Allocation in Network Function Virtualization," *IEEE Access*, vol. 4, 2016, pp. 8084–94.

[7] K. Arulkumaran *et al.*, "Deep Reinforcement Learning: A Brief Survey," *IEEE Signal Processing Mag.*, vol. 34, no. 6, Nov. 2017, pp. 26–38.

[8] Y. He *et al.*, "Software-Defined Networks with Mobile Edge Computing and Caching for Smart Cities: A Big Data Deep Reinforcement Learning Approach," *IEEE Commun. Mag.*, vol. 55, no. 12, Dec. 2017, pp. 31–37.

[9] H. Chang *et al.*, "Distributive Dynamic Spectrum Access Through Deep Reinforcement Learning: A Reservoir Computing-Based Approach," *IEEE Internet of Things J.*, vol. 6, no. 2, Apr. 2019, pp. 1938–48.

[10] D. Silver *et al.*, "Mastering the Game of Go Without Human Knowledge," *Nature*, vol. 550, no. 7676, 2017, pp. 354–59.

[11] C. Qiu *et al.*, "Deep Q-Learning Aided Networking, Caching, and Computing Resources Allocation in Software-Defined Satellite-Terrestrial Networks," *IEEE Trans. Vehic. Tech.*, vol. 68, no. 6, June 2019, pp. 5871–83.

[12] X. Sun and N. Ansari, "Dynamic Resource Caching in the IoT Application Layer for Smart Cities," *IEEE Internet of Things J.*, vol. 5, no. 2, Apr. 2018, pp. 606–13.

[13] D. Zhao *et al.*, "Experience Replay for Optimal Control of Nonzero-Sum Game Systems with Unknown Dynamics," *IEEE Trans. Cybernetics*, vol. 46, no. 3, Mar. 2016, pp. 854–65.

[14] V. Mnih *et al.*, "Human-Level Control Through Deep Reinforcement Learning," *Nature*, vol. 518, no. 7540, 2015, pp. 529–33.

[15] M. Abadi *et al.*, "Tensorflow: A System for Largescale Machine Learning," *Proc. OSDI*, vol. 16, 2016, pp. 265–283.

Biographies

Xiaoyuan Fu received her B.S. degree from Harbin Engineering University, China, in 2014. She is currently pursuing a Ph.D. degree in the State Key Laboratory of Networking and Switching Technology at Beijing University of Posts and Telecommunications (BUPT). Her current research interests include big data, resource management, and machine learning.

F. Richard Yu [F] is a professor at Carleton University, Canada. His research interests include machine learning, connected vehicles, security, and wireless. He serves on the Editorial Boards of several journals, including Co-Editor-in-Chief of *Ad Hoc Sensor Networks*, and Lead Series Editor for *IEEE Transactions on Vehicular Technology*, *IEEE Transactions on Green Communications Networks*, and *IEEE Communications Surveys & Tutorials*. He is a Fellow of the IET. He is a Distinguished Lecturer and the Vice President (Membership) of the IEEE Vehicular Technology Society.

Jingyu Wang obtained his Ph.D. degree from BUPT in 2008. He is currently a professor in the State Key Laboratory of Networking and Switching Technology at BUPT. His research interests span broad aspects of SDN, big data, overlay networks, and traffic engineering.

Qi Qi obtained her Ph.D. degree from BUPT in 2010. Now, she is an associate professor in the State Key Laboratory of Networking and Switching Technology at BUPT. Her research interests include edge computing, IoT, ubiquitous services, and deep learning.

Jianxin Liao obtained his Ph.D. degree from the University of Electronics Science and Technology of China in 1996. He is currently a full professor in the State Key laboratory of Networking and Switching Technology at BUPT. His main research interests include cloud computing, mobile intelligent networks, and multimedia communications.