Article

Wireless Virtual Network Embedding Algorithm Based on Deep Reinforcement Learning

Qi Gao, Na Lyu *, Jingcheng Miao and Wu Pan

School of Information and Navigation, Air Force Engineering University, Xi'an 710077, China;
fyzz98@163.com (Q.G.); zhongdcz@163.com (J.M.); pw1422947346@163.com (W.P.)
* Correspondence: lvnn2007@163.com

Abstract: Wireless network virtualization is widely used to solve the ossification problem of networks, such as 5G and the Internet of Things. The most crucial method of wireless network virtualization is virtual network embedding, which allows virtual networks to share the substrate network resources. However, in wireless networks, link interference is an inherent problem while mapping virtual networks because of the characteristics of wireless channels. To distribute resources efficiently and address the problem of interference, a dynamic embedding algorithm with deep reinforcement learning is proposed. During the training stage, we take resource use and interference from substrate networks as observations to train the agent, and then the agent generates a resource allocation strategy. Aiming at realizing load balance, we reshape the reward function considering the execution ratio and residual ratio of substrate network resources as well as the cost consumed by current virtual network request. Numerical tests show that our embedding approach increases the acceptance ratio and maintains better robustness. Moreover, the results also illustrate that our algorithm maintains a high acceptance ratio while producing less interference and lower cost.

Keywords: virtual network embedding; wireless network virtualization; resource allocation; deep reinforcement learning; interference

Citation: Gao, Q.; Lyu, N.; Miao, J.; Pan, W. Wireless Virtual Network Embedding Algorithm Based on Deep Reinforcement Learning. *Electronics* **2022**, *11*, 2243. https://doi.org/10.3390/electronics11142243

Academic Editor: Martin Reisslein

Received: 13 June 2022
Accepted: 15 July 2022
Published: 18 July 2022

1. Introduction

Wireless networks have received a lot of interest with the development of 5G networks [1,2] and vehicular networks [3]. They provide customers with highly qualified services with low latency through data transmission between base stations and users. However, as the number of wireless network users and the individual requirements increase, the traditional resources management scheme cannot satisfy the updated demand. To deal with the problem of resources allocation, Wireless Network Virtualization (WNV) has become a hot topic [4–7]. WNV overcomes the rigidity of the Internet [8] by enabling many virtual networks (VNs) to use the resources of a shared substrate network (SN), often known as physical network. It can relieve the update and maintenance pressure of internet by the core concept of sharing. As a result, it is critical to find out how to properly allocate substrate network resources with its finite nature.

In the process of wireless network virtualization, it is decoupled into Infrastructure Providers (InPs) and Service Providers (SPs) [9]. SPs rent resources from InPs to provide subscribers with tailored services, while InPs are in charge of deploying resources and managing the allocation policy of the substrate network. This specific technique of WNV is named virtual network embedding (VNE) [10]. The goal of VNE is to incorporate as many VNs as possible with limited resources of the substrate network. As a NP-hard problem [11], the price is hard to bear for finding the optimal resources allocation strategy with exact algorithms. The heuristic algorithm [12] used to be the mainstream approach to solve the issue with affordable computation complexity [13,14]. However, along with the emergence of Machine Learning (ML) techniques, such as graph neural networks

(GNN) [15,16] and deep reinforcement learning (DRL) [17–21], intelligent algorithms are rapidly used to satisfy the diverse requirements of next-generation wireless networks [22] and design communications system [23,24]. ML approaches can not only simulate wireless communication process with inference [25,26] but also assess resource distribution strategy from a deep perspective due to the nature of extracting information [27]. Especially for reinforcement learning algorithms, the parallel framework [16] could speed up the process of embedding and the capability to deal with continuous problems [19] corresponds to the system model of wireless virtual network embedding.

The characteristic of WNV is that resources are complementary and interchangeable [28], which implies that abundant resources can partially compensate for deficient resources. The replacement attribute provides more efficient resource management and promotes resource usage. During the stage of wireless virtual network embedding, the substrate node resource and link resource could adjust each other to satisfy the transmission rate requirement of VNRs. However, this work used to be performed manually in previous literature [28] and the adjustment strategy may produce quite different embedding results. Therefore, our previous work [29] handles the problem by designing a hierarchical scheme before executing the embedding process. However, the several levels cannot reflect the deep nature of complementary in wireless virtual network embedding.

For the above backgrounds, a dynamic wireless virtual network embedding algorithm based on deep reinforcement learning (DWVNE-DRL) is proposed. The primary goal of our algorithm is to maximize the number of embedded VNs. Another significant objective is to enhance the resource use of the substrate network. The significant contributions are given below.

- A novel dynamic embedding strategy is proposed, which applies the deep reinforcement learning strategy Deep Deterministic Policy Gradient (DDPG) [30]. To deal with the complementary attribute of resources, an innovative reinforcement learning environment for WNV is set with continuous state space. During the training stage, we take resource use and interference from substrate network as observations to train the agent, and then the agent generates a resource allocation strategy.
- A new learning technique is used to effectively manage resources. Different resource allocation schemes will be implemented based on the current resource usage state of the substrate network to achieve a higher resource use ratio and embed more VNs. We reshape the reward function considering the execution ratio and residual ratio of substrate network resources as well as the cost consumed by current virtual network request.
- A thorough simulation is run to assess the performance of our approach. The analysis of multiple factors is covered in comparison to traditional wireless virtual network embedding algorithms.

The remainder of this paper is structured as follows: Section 2 introduces the WVNE system model and evaluation metrics. The details of our approach, including the reinforcement learning environment configuration and embedding procedure, are shown in Section 3. Section 4 discusses the simulation results. The concluding part of this paper is Section 5.

2. System Model and Evaluation Metrics

In this section, we first introduce thewireless embedding system model. Then, the embedding constraints are briefly represented, which will be detailed in process of the algorithm. Finally, some evaluation metrics are described to analyze the performance of the algorithm.

2.1. System Model

The wireless embedding system consists of a substrate network and multiple virtual networks. Usually, a substrate network is defined as a weighted undirected graph $G^S(N^S, L^S, A_N^S, A_L^S)$, where N^S represents a set of substrate nodes and L^S represents a collection of substrate links. A_N^S and A_L^S denote node attributes (e.g., power and location) and link attributes (e.g., bandwidth), respectively. Similarly, a virtual network is defined as a weighted undirected graph $G^V(N^V, L^V, R_N^V, R_L^V)$, where N^V and L^V are the set of virtual nodes and links, respectively. R_N^V and R_L^V represent the virtual nodes and links resource requirement for the substrate network. In contrast to wired VNE, a virtual network requires resource as transmission rate $r(l^V)$ in wireless VNE process rather than specific substrate node and link resources, such as computation capability, storage memory, and link bandwidth. Based on the Shannon theorem, the relation between virtual network resource requirement and substrate network resource can be formulated by

$$b\left(l^V\right)\mathrm{lb}\left(1+\frac{p(n^V)g(l^V)}{\sigma^2+\sum_{l^S\in L^S\setminus\left(l^V\to l^S\right)} I(l^S)}\right)\geq r\left(l^V\right) \quad (1)$$

where $b(l^V)$ and $p(n^V)$ denote the bandwidth and power resources allocated by the substrate network, respectively. σ^2 is the white Gaussian noise, and $I(l^S)$ is the adjacent channel interference suffered from other substrate links except the current link embedded to the virtual link l^V. $g(l^V)$ is the channel gain, which can be formulated as

$$g\left(l^V\right)=\mathrm{dis}^k\left(\mathrm{loc}\left(n_1^S\right),\mathrm{loc}\left({n_2}^S\right)\right) \quad (2)$$

where dis() represents the Euclidean distance between substrate nodes n_1^S and n_2^S, and loc() is the location coordinates of substrate nodes. k is the gain coefficient based on the system model.

Therefore, when a virtual network arrives, it will be accepted if the substrate network can afford the corresponding power and bandwidth resources based on the transmission rate required by the virtual network, and it will be rejected if there are not enough resources available. The specific network model is shown in Figure 1 with the virtual network above the dotted line, where the number represents the link transmission rate requirement $r(l^V)$. The substrate network is shown below the dotted line, where the number above the substrate node represents the power resource $p(n^S)$, and the number in the rectangle represents the bandwidth resource $b(l^S)$.

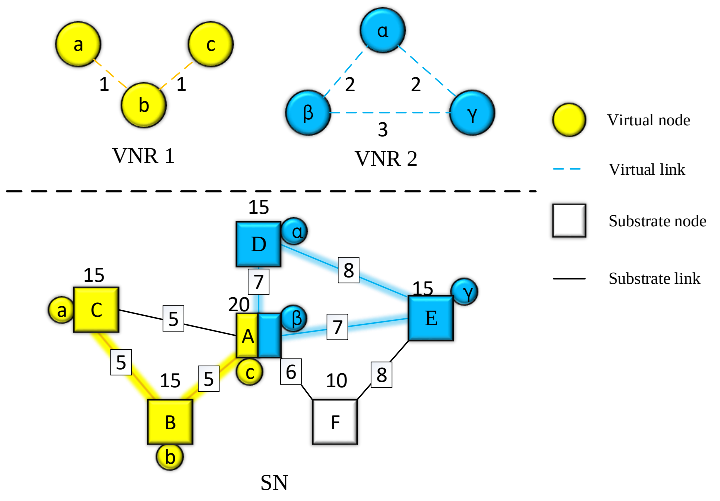

Figure 1. Embedding process.

2.2. Constraints

During the wireless VNE process, different virtual nodes in the same virtual network cannot be embedded on the same substrate node, which is shown as

$$x_{ij} = \begin{cases} 1, & n_i \to n_j \\ 0, & \text{else} \end{cases}$$
$$n_i \in N^V, n_j \in N^S, \sum_{n^S} x_{ij} \le 1 \tag{3}$$

where x_{ij} is a binary number and $x_{ij} = 1$ if the virtual node n_i is embedded to the substrate node n_j.

According to Formula (1), the resources distributed from the substrate network must satisfy the required transmission rate. Otherwise, the current VNR is rejected. Therefore, once a couple of substrate nodes and link are embedded, the maximum rate it can provide is fixed. The virtual nodes and links can only gain resources less than the maximum power $p_{\max}(n_j)$ and bandwidth $b_{\max}(l^S)$ that the substrate network possesses. The restriction is denoted as follows

$$n_i \to n_j : p(n_i) \le p_{\max}(n_j) \tag{4}$$

$$l^V \to l^S : l^V \in L^V, l^S \in L^S, b\left(l^V\right) \le b_{\max}\left(l^S\right) \tag{5}$$

Besides, the selected substrate node must satisfy the constraint of location and cannot exceed the embedding range. It can be expressed as

$$\text{dis}(\text{loc}(n_i), \text{loc}(n_j)) \le \varphi(n_i) \tag{6}$$

where dis() represents the Euclidean distance between virtual node n_i and substrate node n_j. The distance must within the coverage radius $\varphi(n_i)$ of virtual node n_i.

2.3. Evaluation Metrics

The evaluation metrics are introduced in this subsection. Three main metrics are formulated to quantify the performance of the VNE algorithms. Concerning the VNR acceptance ratio, it is the specific value between the number of successfully embedded VNRs and the number of total VNRs arrived in the given period. A VNE algorithm is highly estimated with a high VNR acceptance ratio for adopting more VNRs and providing more services to users. It can be defined as

$$acc = \frac{Num_{acc}(vnr)}{Num_{\text{tot}}(vnr)} \tag{7}$$

With respect to the use of power resource and bandwidth resource, it evaluates the resource allocation strategy of the algorithm. A high use ratio represents the better resource allocation strategy for distributing resources rationally and effectively. It is formulated as

$$\eta = \frac{A^S_{use}}{A^S_{all}}, A^S = p\left(n^S\right), \text{ or } b\left(l^S\right) \tag{8}$$

The cost reflects the resources that the substrate network consumes to embed VNRs. In previous work, there is another metric, revenue, which represents the resources that VNRs acquire from SN. In this work, we focus on the resource consumption of wireless network after considering interference in the embedding process. It is formulated as (9), where $p(n^V)$ and $b(l^V)$ are the resources distributed to virtual node and virtual link. $\beta(l^S)$ is the number of substrate links covered by virtual link l^V.

$$\text{cost}\left(G^V\right) = \sum_{n^T \to n^S} p\left(n^V\right) + \sum_{l^I \to l^S} \sum_{l^S} \beta\left(l^S\right) b\left(l^V\right) \tag{9}$$

3. Embedding Algorithm

In this section, the proposed DWVNE-DRL algorithm is detailed. At first, the reinforcement learning environment is presented where we set state space and action space based on the wireless network embedding model and innovate the reward function. Furthermore, the process of the proposed algorithm is implemented, including the node embedding process and link embedding process. In the section, we analyze the time complexity of our algorithm.

3.1. Reinforcement Learning Environment

In a wireless network, there exists a difference in the process of VNE compared with a wired network. Users intend to request a transmission rate similar to the download and upload rate in wireless network, different from specific CPU compacity and bandwidth requirements. Inspired by the Shannon theory, multiple combinations of power and bandwidth resources are available when a concrete transmission rate is given. Therefore, it not only considers how to allocate substrate network resources to accept more VNRs, but also focuses on how to balance the relationship between power and bandwidth resources for the current arrived VNR.

There are four important parameters in reinforcement learning, which are state space, action space, rewards and strategy. The state space of an agent is what the agent can see in the environment, and the agent then picks the next action from the action space. From the perspective of the agent, action selection is dictated by the strategy of receiving rewards from the environment, which indicates whether the present strategy is good or poor. After the strategy adjusted, the agent follows the updated strategy to obtain new action. During the iteration, the strategy is promoted to generate better actions.

In the literature [16,31], virtual networks acquire specific node and link resources, and the strategies based on the RL algorithm aim at selecting substrate nodes and links with abundant resources to embed VNRs. However, it is different in wireless network, where the requirement is transmission rate and the substrate network resources allocation strategy is indeterminate. To find a better resources allocation strategy from the global view, a dynamic VNE model is proposed based on deep reinforcement learning algorithm.

3.1.1. State Space

From the wireless network environment, an agent can obtain lots of parameters while not all of them are vital to the agent, so it is necessary to simplify observation values. In our wireless VNE model, the goal of the agent is to find the appropriate power and bandwidth satisfying the transmission rate raised from the arrived VNR. Because of the continuous property, we select several parameters for agent to learn. There are two classes of state values, the substrate network and virtual network. In terms of the substrate network, the reminder and use ratio of physical resources are necessary for the agent to make decisions. The two state values are different from each other while inputing to the agent. Furthermore, the link interference, the most unique feature in wireless network, is incorporated on the basis of Formula (1). For the virtual network, the required transmission rate is considered for determining the scale of resources to be allocated. Certainly, there are also other important attributes such as network topology that indicates the location in space, but they are less useful for the agent to learn a better resources allocation strategy from the environment than the selected state values. It is important that in our proposed model only the few nodes and link connecting them are focused on, which means they are waiting to be embedded currently. It would be a large computation task for the agent if the whole network parameters were considered. Therefore, for the sake of simplification, the state space is formulated as

$$\text{state\_space} = \left(\text{Uti\_ratio}^{T}, \ \text{Rem\_res}^{T}, \ \text{Infe}, \text{Rate} \right) \tag{10}$$

where Uti_ratio ^{T} represents the use ratio of physical resources, including two selected substrate nodes and one link. Rem_res ^{T} is the spare physical resources of selected nodes and link. Infe and Rate are link interference of substrate network and transmission rate of current VNR, respectively. The dimension of state space is 8 from Formula (10).

3.1.2. Action Space

Unlike other RL algorithms, the action in our proposed model is not a decision that selects a substrate node but instead seeks the moderate balance between power and bandwidth as transmission rate is given. Due to the stochastic total and unpredictable use ratio of resources in different substrate nodes and links, especially the continuous characteristic of resources, it is hard to take the concrete value of resources as action considering various nodes and links. Therefore, the occupation ratio of resources to be allocated is qualified as action for the agent to adopt. The internal relevance between state space and action is expected to be recognized by the agent so that resources are allocated effectively and rationally. Based on Formula (1), both power and bandwidth contribute to the transmission rate. It would become hard to converge for the agent if the two decisions were made together, because the number of successful allocation schemes is countless. Therefore, only the power allocation ratio is selected as the action, and bandwidth resource to be occupied is calculated with the Shannon formula. Action space is defined as

$$\text{action\_space} = \text{p\_ratio} \tag{11}$$

where p_ratio is the allocation ratio of power resource. The agent aims at providing the current pair of node and link distribution strategy. Furthermore, for simulation, the specific value is set to (0, 1) with step of 0.001.

3.1.3. Shape of Rewards

The learning capacity of the agent is up to how the reward is being shaped. The agent is told how good the current strategy is by assigning various rewards. In the proposed wireless VNE model, the rewards function is shaped as follows.

As mentioned in last subsection, the action space is defined as the occupation ratio for the next-step resources' allocation. In our proposed wireless VNE model, one of the most important goals is to balance the power and bandwidth resources to be allocated. Therefore, the agent is expected to distribute more power and less bandwidth when the embedded node has adequate power resources but insufficient bandwidth resources, and vice versa. For instance, if the agent made a decision following a bad strategy, it would happen to the substrate network such that there were sufficient power resources with little or no bandwidth resources and then it would not be able to accept subsequent arrived VNRs. Furthermore, this part of the reward is defined as

$$r_o = \exp\left(-\left|s_p \times a_p - s_{bw} \times a_{bw}\right|\right) \tag{12}$$

where s_p is the use ratio of the current substrate node, and s_{bw} is the use ratio of the substrate link bandwidth resource between selected nodes. a_p is the ratio of power to be allocated, so the same as a_{bw} is the ratio of bandwidth to be distributed. The content of the absolute value $s_p \times a_p - s_{bw} \times a_{bw}$ indicates the resources executed condition. A good action should be adopted based on the current resources' use, so it will be rewarded a relatively large value that allocate power resources more when bandwidth resources of selected link is lack.

Owing to the limited resources of wireless network, the strategy with low cost will be regarded as a good strategy. The function of cost, as Formula (13), contributes to a part of reward by adding unit cost to reflect expenses of current action. The cost can be adjusted as follows

$$r_c = \alpha \times p\left(n^V\right) + \beta \times b\left(l^V\right), \alpha = \frac{1}{p_L(n^S)}, \beta = \frac{1}{b_L(l^S)} \tag{13}$$

where α and β are unit cost of power and bandwidth resource, respectively. $p_L(n^S)$ represents the left power resource of substrate node n^S, and $b_L(l^S)$ indicates the left bandwidth resource.

Due to the state space including a couple of substrate nodes and one link, some nodes may act as relay nodes and resources could be occupied more. In the final formation of the reward, the two connections between node and link are expressed as

$$reward = \frac{(r_{o1} + r_{o2})}{r_c} \tag{14}$$

where r_{o1} and r_{o2} are partial reward of resource use ratio on physical node n_1^S and n_2^S with their link, respectively. The correlation between *reward* and r_o is positive while r_c is negative to final reward.

3.1.4. Learning Strategy

Wireless VNE is a continuous decision problem since the values of power and bandwidth resources vary continuously for different strategy. Compared with the Q-learning strategy, the policy gradient method aims at obtaining better action directly with adjusting the parameters of strategy and is more suitable for our model. It is hard that create a q-table to update q-values and find a better action.

By applying traditional policy gradient algorithm, an agent with strategy π will undergo a continuous status sequence $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots\ldots, s_t, a_t, r_t)$, in which the strategy π is evaluated by $J(\theta)$ (shown in Formula (15)). π_θ represents the strategy π defined by the parameter θ. $J(\theta)$ is formulated as the same as expectation and can be simplified as Formula (16). To measure the quality of an action a in the state s , the action-value function $Q(s,a)$ is introduced. There is no best strategy in reinforcement learning algorithm because of the unsupervised property and fluctuate environment states, but a better solution is selected based on the update of rewards when obtaining the gradient of $J(\theta)$ in policy gradient algorithm.

$$J(\theta) = E[r_0 + r_1 + \ldots + r_t \mid \pi_\theta] \tag{15}$$

$$J(\theta) = E_{\tau - \pi_\theta}\left[\sum_t \tau\right] = \frac{1}{N}\sum_i \sum_t r(s_{i,t}, \pi(s_{i,t} \mid \theta)) \tag{16}$$

In our proposed algorithm, the Deep Deterministic Policy Gradient strategy [30], developed from policy gradient, is adopted to optimize the resources allocation strategy. To update θ in $J(\theta)$ more precisely, it introduces a neuro network to acquire strategy gradient. Based on Actor-Critic mode, DDPG applies two major networks in which exists four subnetworks. One of the major networks, called main network, is in charge of principal training task and updating strategy parameter θ in the normal iteration process. Another major network is target network, which shares the same network topology and parameters with main network but delays updating parameter θ. The delay offers the agent a global view to generate relatively better actions. The learning strategy is shown in Figure 2.

In each major network, there are two subnetworks, the actor network and critic network. According to the observation values, actor network outputs the current-better actions, and then the observation values and corresponding action values will be delivered to the critic network. The critic network evaluates the produced actions and adjusts the network parameters following the principle that rises in the maximum direction along the gradient of $J(\theta)$.

Take the main network for example. $J(\theta)$ turns into $J(\theta^\mu)$ in actor network μ as Formula (17), where γ is the decay factor decreasing the impact of long-term actions on the present. Learning from the actor network μ , the critic network c calculates the action-value function $Q(s, a|\theta^c)$ on the basis of the critic network parameter θ^c. The gradient of $J(\theta^\mu)$ in the main network is formulated as Formula (18), where $\pi(s|\theta^{\mu})$ is the strategy based on the input observation value s and the actor network parameter θ^{μ} .

$$J(\theta^{\mu}) = E_{\theta^{\mu}}\left[r_1 + \gamma r_2 + \gamma^2 r_3 + \dots\right] \tag{17}$$

$$\nabla J(\theta^{\mu}) = E_s\left[\frac{\partial Q(s, a|\theta^c)}{\partial a}\frac{\partial \pi(s|\theta^{\mu})}{\partial \theta^{\mu}}\right] \tag{18}$$

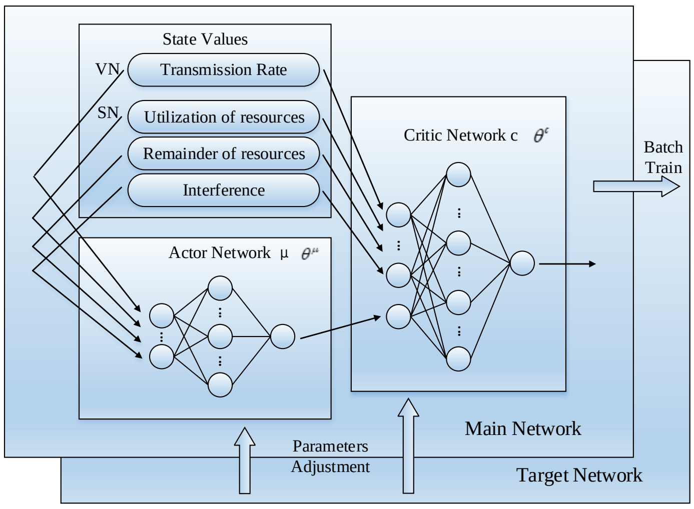

Figure 2. The learning strategy in agent.

As for the target network, it delays updating until a batch of state and action values input. The calculation of $\nabla J(\theta^{\mu})$ is similar to the main network, but modified slightly for the batch processing property. It is detailed as follows:

$$\nabla J(\theta^{\mu}) \approx \frac{1}{N}\sum_i \nabla_a Q(s, a \mid \theta^c)\Big|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^{\mu}}\mu(s \mid \theta^{\mu})\Big|_{s_i} \tag{19}$$

where N is the size of the calculated batch and s_i is the ith observation value. The actor network is built with two full connection layers. The size of first layers is (8, 30) and the activate function is set to ReLU. The output layer only yields one action so the activate function is Sigmoid. In terms of critic network, it has a similar structure to an actor network. The first layer inputs nine values including eight state values and one action values, and then outputs 60 values to next layer. The final output is also one value, named after the q-value, which is used to evaluate whether the current action is good. The structure and parameter of target network is the same as main network but update parameters slower.

3.2. Process of the Algorithm

Typically, there are two stages in the process of virtual network embedding, called node embedding and link embedding. The details of process are illustrated in this subsection.

3.2.1. Node Embedding Process

In the process of wired virtual network mapping, the requirements of virtual nodes for resources are relatively fixed, such as computing resources, memory, etc. In the wireless virtual network mapping, because the request for the virtual network is the transmission rate, the physical network considers the power resources of the physical nodes and the bandwidth resources of the connected physical links when allocating resources. Current research on wireless virtual network mapping uses an extended node resource approach, where the node power resources and all link bandwidth resources connected to the node are combined and ranked, in order to find nodes that meet both power and bandwidth requirements and improve the mapping success rate. This ordering takes into account all the link resources connected to the node, but does not take into account the actual mapping capacity of the link. To measure the average capacity, the node degree is introduced into a novel ranking method as

$$\operatorname{rank}\left(n^S\right) = p\left(n^S\right) + \frac{\sum_{n^S} b\left(l^{n^S}\right)}{\operatorname{Deg}(n^S)} \tag{20}$$

where $\operatorname{Deg}\left(n^S\right)$ is the degree of node n^S and $b\left(l^{n^S}\right)$ is the bandwidth resources of links connected to node n^S.

Based on the ranking method, the substrate node with largest rank value will be chosen when a new VNR arrives. The greedy algorithm also decreases the searching space and time to embed VNR more effectively. The node embedding process is shown in Algorithm 1.

Algorithm 1 Node embedding.

```
1: while VNR arrives do
2:     Descend order substrate nodes based on Formula (20)
3:     Select nodes that satisfy Formula (6)
4:     Descend order virtual links based on the required transmission rate
5:     Select substrate nodes with high rank values for the top links until all virtual nodes
       are allocated
6: end while
```

3.2.2. Link Embedding Process

In wireless VNE model, link interference is an unavoidable factor for bandwidth resources distribution. The actual interference mainly comes from electromagnetic wave, limited spectrum resources, natural noise and aggressor source. For the sake of simplification, only link interfering and natural noise are considered in our proposed model. The link interference is formulated as follows:

$$d_{\mathrm{I}}\left(l^S\right) = \gamma \frac{d_l + 1}{b(l^S)} \tag{21}$$

where d_l denotes the number of other substrate links that might influence current link l^S and γ represents the weight how the link interfering occupies in whole interferences.

Due to the internal connection between power resources and bandwidth resources, the link embedding process is closely related to the node embedding process. The shortest path algorithm is adopted to find the optimal path with the least interference. In a wireless environment, the interference becomes higher if there are many signals on transmission in channel. To provide users with high-quality services, the substrate paths with less interference will be embedded in the link embedding process after substrate nodes are selected. Once the physical nodes and links are selected, the agent will generate action on the basis of the current strategy to allocate resources. The link embedding process is shown in Algorithm 2.

Algorithm 2 Link embedding.

```
1:  for link in virtual links do
2:      Calculate link interference of substrate links based on Formula (21)
3:      Find available path between substrate nodes from Algorithm 1
4:      Input action from DDPG and conduct resources distribution strategy
5:      if satisfy resources constraints do
6:          embedded = True
7:      else
8:          embedded = False
9:      end if
10: end for
```

For one node embedding process, the time complexity mainly lies in the node resources ranking which is $O(|N^S|)$. As for link embedding in one couple of nodes, the procedure of calculating interference and searching available paths contributes to the complexity, which can be defined as $O(|L^S| + |N^S|\text{lb}|N^S|)$. Therefore, the time complexity for one complete embedding process is $O(|N^S| + |N^V||L^S| + |N^V||N^S|\text{lb}|N^S|)$.

4. Performance Evaluation and Analysis

In this section, we first describe the environment and network settings, and then illustrate the training process of RL model. Compared with the state-of-the-art VNE algorithms, we analyze the different performances based on Section 2.3.

4.1. Evaluation Settings

In the range of 20 km $\times$ 20 km, the topology of the substrate network is generated using the Waxman random graph [32] set at $\alpha = 0.5$ and $\beta = 0.2$. The specific settings are shown in Table 1. For each VNR, the probability of links connecting adjacent nodes is set to be 0.5 and the lifetime is 500 time units. VNRs arrives as a Poisson process with an average rate of 4 per 100 time units. The gain coefficient k is set to 4 and environment Gauss white noise is 10^{-8}. The testing phase lasts 5000 time units for each simulation. The whole project is implemented on a tensorflow with an Intel R Core(TM) i7-7700 CPU @2.8 GHz and 8.00 GRAM Machine.

Table 1. Parameter settings.

Parameter	Value
Nodes number of SN	50
Power capacity	[50, 100], uniform distribution
Bandwidth capacity	[25, 50], uniform distribution
Nodes number of each VNR	[3, 5], uniform distribution
Transmission rate of each virtual link	[3, 8], uniform distribution
Learning rate of actor network	0.00025
Learning rate of critic network	0.0025

4.2. Main Evaluation Tests

To validate the performance of our algorithm, it is compared with two WVNE algorithms shown in Table 2. The two algorithms are both focused on the complementary feature of resources and intend to realize load balancing. They find the optimal resource allocation schemes by minimizing cost but are slightly different from each other in specific process of calculation.

Table 2. Compared algorithm.

Algorithm	Description
WVNE-JBP [28]	Greedy strategy for node embedding and shortest path with path splitted for link embedding
WVNE-JHR [29]	Hierarchical rank strategy and adjusted objective function for different level VNRs

Figure 3 shows the acceptance ratio of different algorithms. It can be found that the acceptance ratio decays as the simulation proceeds for all algorithms. According to the limited resources, the substrate network cannot embed all arrived VNRs. Furthermore, it needs more resources to embed as the wireless environment is gradually full of transmission signals that increase link interference. At the last of simulations, the acceptance ratio of our algorithm is 80% relative to the others while the WVNE-JHR and WVNE-JBP end at 66% and 45%, respectively. The reason for this is that our algorithm selects resource distribution strategy by observing the environment states, which include current resource states of substrate network and environment interference. However, another two algorithms do not take interference into consideration when calculate the optimal resource values to give virtual networks.

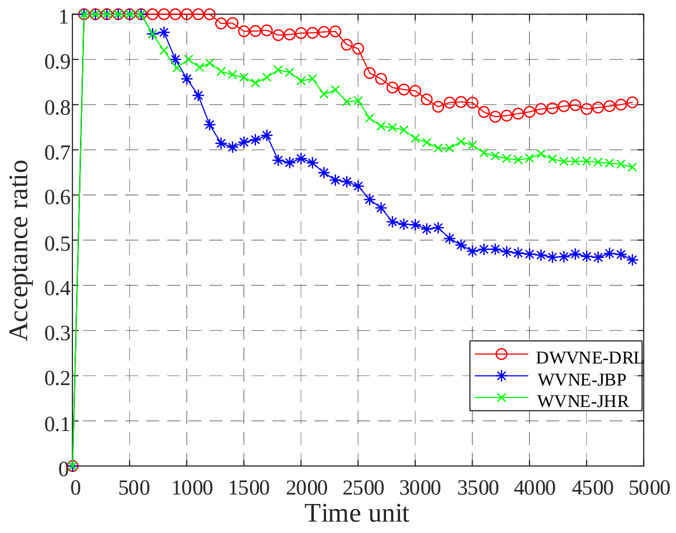

Figure 3. Comparison of acceptance ratio.

Figures 4 and 5 display the resource use ratio of different algorithms. In Figure 4, the power use of our algorithm is far less than the other two algorithms while in Figure 5 all of the algorithms maintain similar bandwidth use. Combined with acceptance ratio, it is concluded that the proposed algorithm consumes fewer resources but acquires a higher acceptance ratio. The inharmonious situation results from the fact that our algorithm tends to provide resources with low interference. The more power substrate nodes occupy, the higher interference occurs in link embedding process. Furthermore, it contributes to cover more bandwidth to satisfy the transmission rate requirement. In node embedding process, our algorithm allocates relatively little power causing less interference. Therefore, in terms of link embedding process, more bandwidth resources are needed to compensate power for overcoming link interferences. Another two algorithms find the optimal solution through minimizing cost, only focusing on current remainder of resources, so the high use of power leads to much interference and more bandwidth requirement. However, the high consumption does not achieve the expected results but actually impedes them.

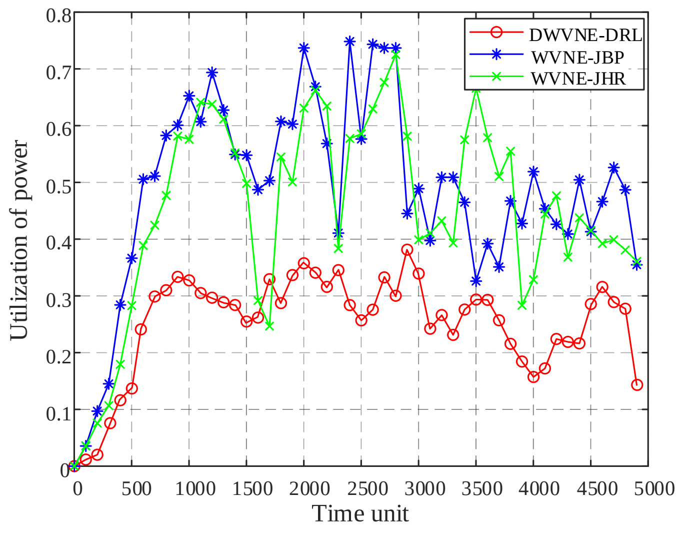

Figure 4. Comparison of power use.

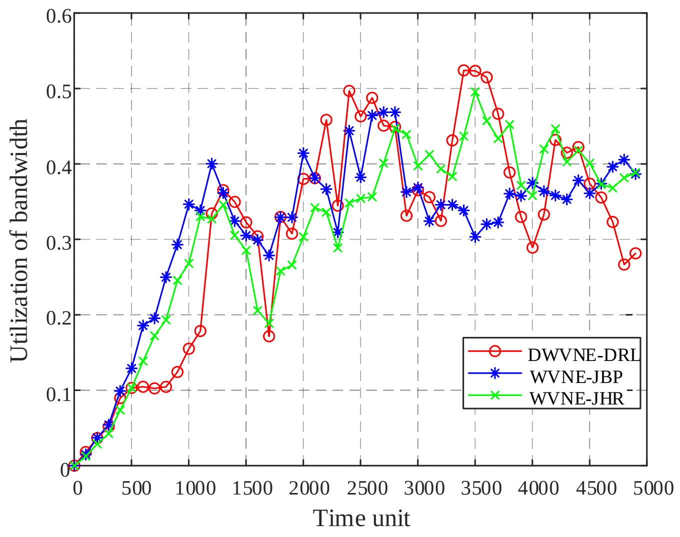

Figure 5. Comparison of bandwidth use.

The cost of different algorithms is depicted in Figure 6. From the above analysis, it is known that our algorithm uses fewer resources to embed VNRs so that the cost consumption is also relatively lower than the other algorithms. Due to three algorithms sharing the same simulation network, the cost differs not as much as in other evaluation metrics, such as use. The relationship between the requirements of the VNRs and the supply from substrate network is nearly the same.

Table 3 shows the average embedding time while successfully embedding VNRs. Derived from Table 3, we could learn that the proposed algorithm has a low time complexity compared with other algorithms. The reason for this is that we simplify the calculation of objective function and the process of finding the optimal resource distribution scheme is made by the neuro network which is trained before. Therefore, the proposed algorithm can embed VNRs in a limited time.

Table 3. Embedding execution time.

Algorithm	**Time Consumption**
DWVNE-DRL	3.27 s
WVNE-JBP	3.64 s
WVNE-JHR	3.73 s

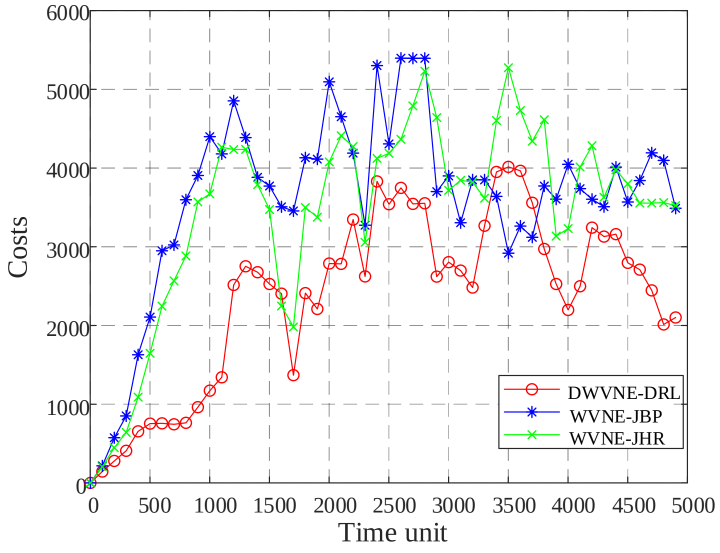

Figure 6. Comparison of costs.

4.3. Arrival Rate Tests

To assess the accommodated performance of algorithms in a busy time, we set different arrival rate of VNRs from four arrivals per 100 time units to 20 arrivals per 100 time units with step by 2. A comparison of the acceptance ratios is illustrated in Figure 7. As the arrival rate raises, the acceptance ratio of all algorithms declines and the falling range is quite large for about 60%. The contributing cause is that the substrate network cannot accommodate so much and frequent VNRs for its finite resources. Nevertheless, our algorithm also has a better performance than others.

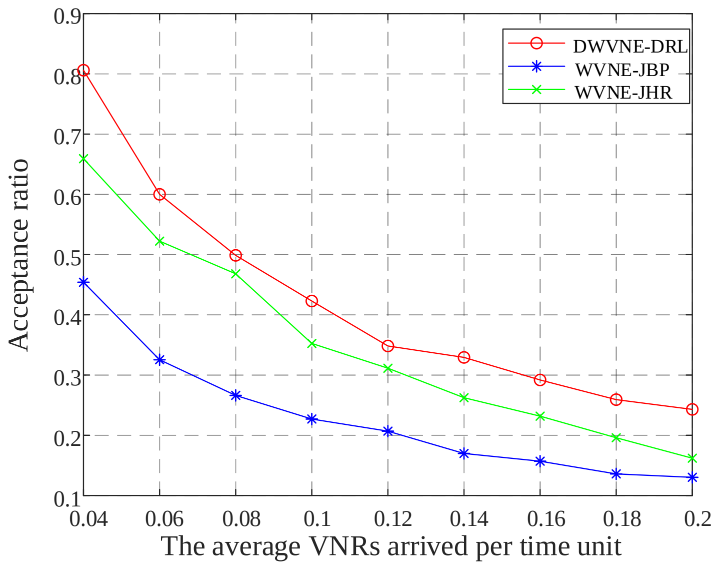

Figure 7. Comparison of acceptance ratio on different arrival rate.

Figures 8 and 9 demonstrate the resource use on different arrival rates. On the whole, the use rises as the number of VNRs increases; however, it fluctuates in specific arrival rate. In terms of power use, WVNE-JBP and WVNE-JHR always keep a higher value than the proposed algorithm. However, for bandwidth use, they share a similar trend and fluctuation. The reason for this is that the resource allocation principle does not change and the relationship between power usage and interference keeps the same. After adding interference into consideration, our algorithm achieves a higher acceptance ratio while economizing power and producing less interference.

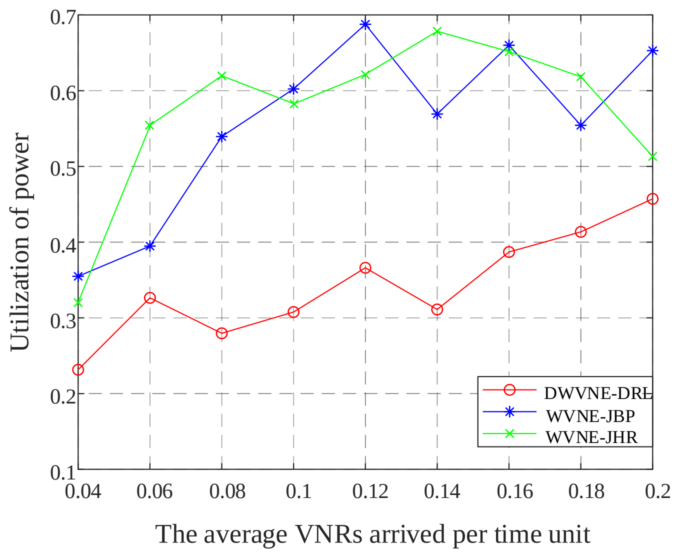

Figure 8. Comparison of power use on different arrival rate.

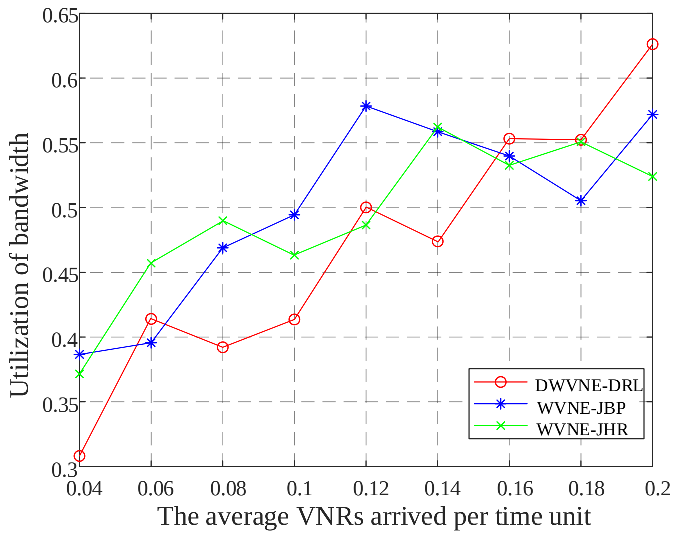

Figure 9. Comparison of bandwidth use on different arrival rate.

5. Conclusions

A wireless virtual network embedding algorithm based on DDPG is proposed in this paper to cope with load balance and create good action. Unlike previous algorithms, our method focuses on solving a continuous problem and assigning precise resource values. To make resource distribution more reasonable, we develop a new reward function. According to the simulation results, the suggested method surpasses traditional algorithms by 14% in terms of acceptance ratio. While keeping a low cost consumption, our algorithm uses fewer power resources, contributing to lower interference, but embeds more VNRs successfully. As for robustness, with the arrival rate increasing, the proposed DWVNE-DRL always outperforms the compared algorithms though the performance goes down gradually. Additional aspects will be added to our model in the future, such as: (1) substrate node mobility in a wireless network; (2) unexpected stoppage in previously embedded VNRs. Other characteristics of the wireless network will also be considered when training the agent.

Author Contributions: Writing original draft, Q.G.; Experiments and simulation, Q.G. and J.M.; Writing review and editing, N.L. and W.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, H.; Liu, Y.B.; Zhao, H.T.; Zhu, H.B.; Sun, Y.F. Wireless virtual embedding algorithm considering inter-cell interference in 5G ultra-dense Network. In Proceedings of the 2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN), Chongqing, China, 12–15 June 2019; pp. 62–67.
2. Feng, J.Y.; Zhang, Q.X.; Dong, G.Z.; Cao, P.F.; Feng, Z.Y. An approach to 5G wireless network virtualization: Architecture and trial environment. In Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA, 19–22 March 2017.
3. Liang, L.; Ye, H.; Li, G.Y. Toward intelligent vehicular networks: A machine learning framework. *IEEE Internet Things J.* **2019**, *6*, 124–135. [CrossRef]
4. Liang, C.C.; Yu, F.R. Wireless network virtualization: A survey, some research issues and challenges. *IEEE Commun. Surv. Tutor.* **2014**, *17*, 358–380. [CrossRef]
5. Liang, C.C.; Yu, F.R. Wireless virtualization for next generation mobile cellular networks. *IEEE Wirel. Commun.* **2015**, *22*, 61–69. [CrossRef]
6. Afifi, H.; Karl, H. Reinforcement learning for virtual network embedding in wireless sensor networks. In Proceedings of the 2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Thessaloniki, Greece, 12–14 October 2020; pp. 123–128.
7. Khan, I.; Belqasmi, F.; Glitho, R.; Crespi, N.; Morrow, M.; Polakos, P. Wireless sensor network virtualization: A survey. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 553–576. [CrossRef]
8. Anderson, T.; Peterson, L.; Shenker, S.; Turner, J. Overcoming the internet impasse through virtualization. *Computer* **2005**, *38*, 34–41. [CrossRef]
9. Belbekkouche, A.; Hasan, M.M.; Karmouch, A. Resource discovery and allocation in network virtualization. *IEEE Commun. Surv. Tutor.* **2012**, *14*, 1114–1128. [CrossRef]
10. Fischer, A.; Botero, J.F.; Beck, M.T.; de Meer, H.; Hesselbach, X. Virtual network embedding: A survey. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1888–1906. [CrossRef]
11. Chowdhury, N.M.M.K.; Boutaba, R. Network virtualization: State of the art and research challenges. *IEEE Commun. Mag.* **2009**, *47*, 20–26. [CrossRef]
12. Cao, H.T.; Hu, H.; Qu, Z.C.; Yang, L.X. Heuristic solutions of virtual network embedding: A survey. *China Commun.* **2018**, *15*, 186–219. [CrossRef]
13. Cao, H.T.; Zhu, Y.X.; Zheng, G.; Yang, L.X. A novel optimal mapping algorithm with less computational complexity for virtual network embedding. *IEEE Trans. Netw. Serv. Manag.* **2018**, *15*, 356–371. [CrossRef]
14. Cao, H.T.; Wu, S.C.; Hu, Y.; Mann, R.S.; Liu, Y.; Yang, L.X.; Zhu, H.B. An efficient energy cost and mapping revenue strategy for interdomain NFV-enabled networks. *IEEE Internet Things J.* **2020**, *7*, 5723–5736. [CrossRef]
15. Habibi, F.; Dolati, M.; Khonsari, A.; Ghaderi, M. Accelerating virtual network embedding with graph neural networks. In Proceedings of the 2020 16th International Conference on Network and Service Management (CNSM), Izmir, Turkey, 2–6 November 2020; pp. 1–9.
16. Yan, Z.X.; Ge, J.G.; Wu, Y.L.; Li, L.X.; Li, T. Automatic virtual network embedding: A deep reinforcement learning approach with graph convolutional networks. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 1040–1057. [CrossRef]
17. Rkhami, A.; Hadjadj-Aoul, Y.; Outtagarts, A. Learn to improve: A novel deep reinforcement learning approach for beyond 5G network slicing. In Proceedings of the 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 Jan 2021; pp. 1–6.
18. Yao, H.P.; Zhang, B.; Zhang, P.Y.; Wu, S.; Jiang, C.X.; Guo, S. RDAM: A reinforcement learning based dynamic attribute matrix representation for virtual network embedding. *IEEE Trans. Emerg. Top. Comput.* **2021**, *9*, 901–914. [CrossRef]
19. Yao, H.P.; Ma, S.H.; Wang, J.J.; Zhang, P.Y.; Jiang, C.X.; Guo, S. A continuous-decision virtual network embedding scheme relying on reinforcement learning. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 864–875. [CrossRef]
20. Lu, M.L.; Gu, Y.; Xie, D.L. A dynamic and collaborative multi-layer virtual network embedding algorithm in SDN based on reinforcement learning. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 2305–2317. [CrossRef]
21. Zeng, J.J.; Ju, R.S.; Qin, L.; Hu, Y.; Yin, Q.J.; Hu, C. Navigation in unknown dynamic environments based on deep reinforcement learning. *Sensors* **2019**, *19*, 3837. [CrossRef]
22. Jiang, C.X.; Zhang, H.J.; Ren, Y.; Han, Z.; Chen K.C.; Hanzo, L. Machine learning paradigms for next-generation wireless networks. *IEEE Wirel. Commun.* **2017**, *24*, 98–105. [CrossRef]
23. Ibnkahla, M. Applications of neural networks to digital communications—A survey. *Elsevier Signal Process.* **2000**, *80*, 1185–1215. [CrossRef]

24. O'Shea, T.; Hoydis, J. An introduction to deep learning for the physical layer. *IEEE Trans. Cogn. Commun. Netw.* **2017**, *3*, 563–575. [CrossRef]
25. Moldoveanu, M.; Zaidi, A. In-Network Learning: Distributed Training and Inference in Networks. Available online: https://arxiv.org/pdf/2107.03433.pdf (accessed on 17 September 2021).
26. Moldoveanu, M.; Zaidi, A. On in-network learning: A comparative study with Federated and Split Learning. In Proceedings of the 2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Lucca, Italy, 27–30 September 2021; pp. 221–225.
27. Aguerri, I.E.; Zaidi, A. Distributed variational representation learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 120–138. [CrossRef]
28. Cao, B.; Xia, S.C.; He, F.; Li, Y. Research of embedding algorithm for wireless network virtualization. *J. Commun.* **2017**, *38*, 35–43.
29. Gao, Q.; Lyu, N.; Miao, J.C. Wireless Virtual Network Embedding Algorithm Based on Load Balance. Available online: http://kns.cnki.net/kcms/detail/51.1307.TP.20220418.1803.011.html (accessed on 20 April 2022).
30. Gu, S.X.; Lillicrap, T.; Sutskever, I.; Levine, S. Continuous deep Q-learning with model-based acceleration. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016.
31. Zhang, S.D.; Wang, C.; Zhang, J.S.; Duan, Y.X.; You, X.H.; Zhang, P.Y. Network resource allocation strategy based on deep reinforcement learning. *IEEE Open J. Comput. Soc.* **2020**, *1*, 86–94. [CrossRef]
32. Waxman, B.M. Routing of multipoint connections. *IEEE J. Sel. Areas Commun.* **1988**, *6*, 1617–1622. [CrossRef]