NFVdeep: Adaptive Online Service Function Chain Deployment with Deep Reinforcement Learning

Asmaa Elhadad

Summer term 2020

The development of network function virtualization (NFV) has introduced a new method,which is assuming that network services are service function chains (SFCs) that are consisted of multiple virtual network functions (VNFs)(Software instead of hardware).this system is called NFVdeep.However many problems face the system,the most important is that the system receives many requests with different quality of service (QoS) requirements. we can solve that by using adaptive online SFC deployment with the help of MDP model and PG policy

1 Introduction

service requests are represented as service function chains (SFCs) that are consisted of multiple virtual network functions (VNFs). using the VNFs can increase the flexibility and efficiency of the system. The system continuously receives requests of different Quality of service requirement and trac conditions,which causes a real time network variations. we can solve this problem by using Marcov decision process model to capture dynamic network state transition, in other words capturing the state of the network which continuously changes between SFCs and each sfc contains number of vnfs and so on. in order to solve the real time network state variations problem we agreed on using adaptive online SFC deployment in figure 1 we explain two different ways of SFC deployment.

2 Architecture of NFVdeep

3 principals

in order to understand the principal of adaptive online SFC deployment we have to understand some concepts first.

3.1 Markov decision process Model

In the NFVdeep we deal with or face real time network variations which is caused by continuously arriving requests. We use the concept of time slots where we start processing the arriving requests according to the time slot they arrived in or even parallel in a pipline. if a request arrives alone in a time slot then it will be processed immediately. At each time slot the NFV system will 1- scan all servers in the network, 2- remove all timeout requests, 3- receive new requests arriving in the time slot, 4-take SFC deployment decisions and finally updating the network state. THE REWARD FUNCTION: we use the reward as an expression to note the NFV and customer‘s profits as we are most concerned about these 2 things. reward is the total accepted requests minus the total cost of occupied servers to deploy the arriving requests

3.2 types of policies

there are two types of policies 1-Value based approach (ex:DQN) 2-Policy based approach. in most applications the requests are predefined and known

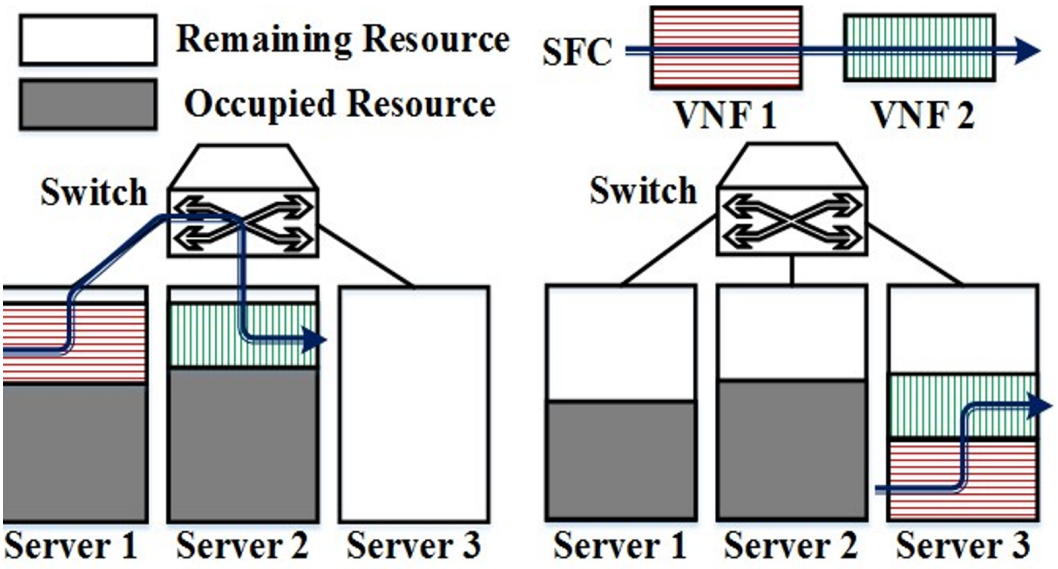

Figure 1: fig 2 NFV architecture[1] [**\cite {NVF_deep:_handb_schol}**]

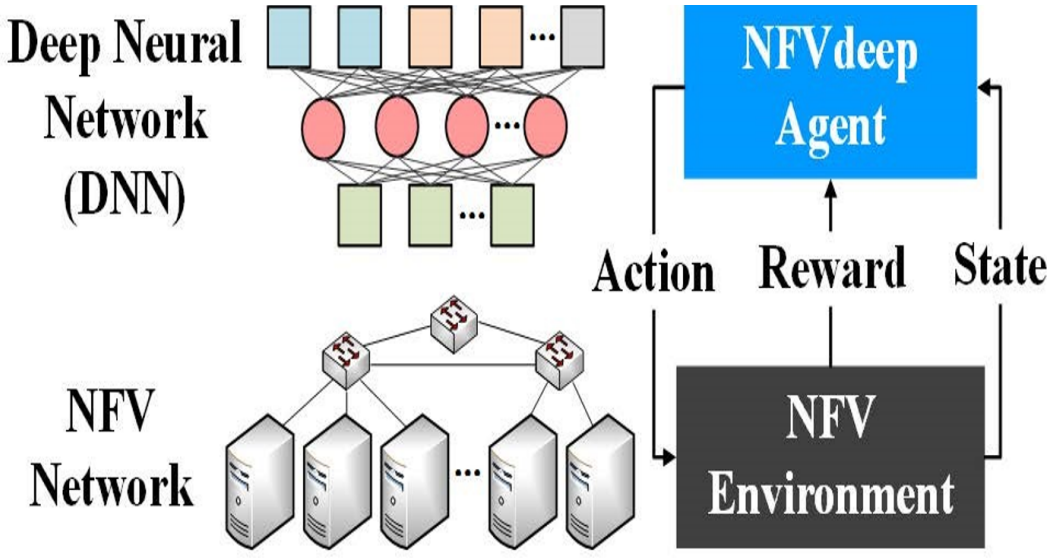

Figure 2: caption 1 [**jon:2019**]

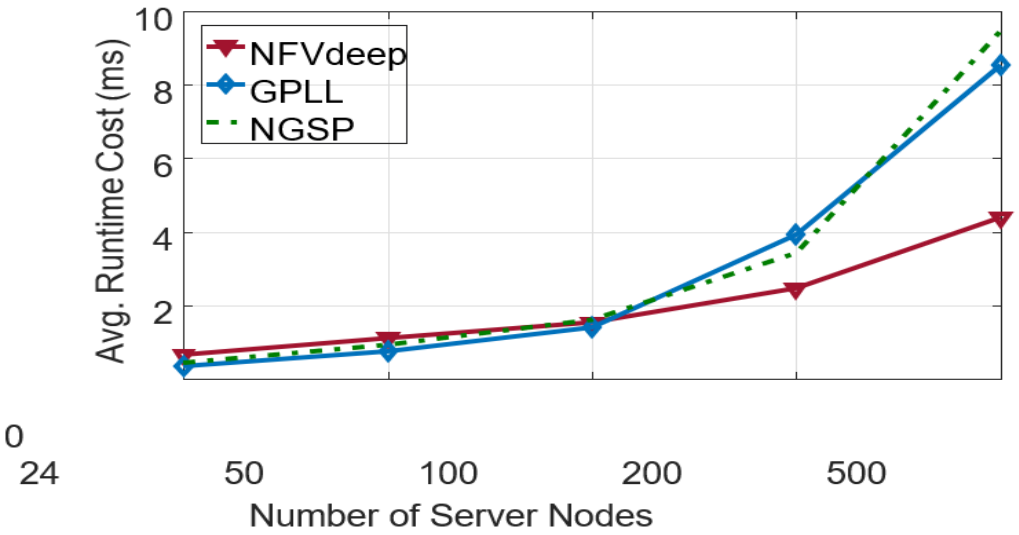

Figure 3: caption 1 [**jon:2019**]

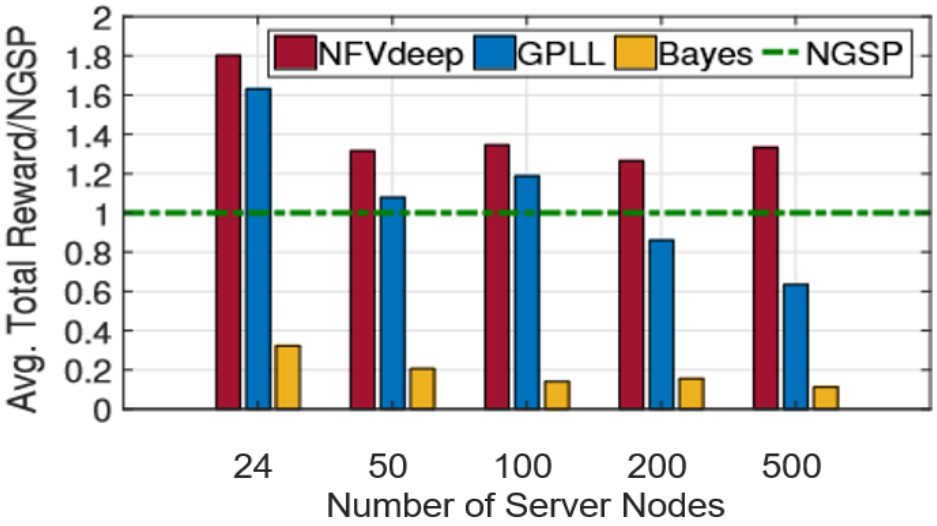

Figure 4: caption 1 [**jon:2019**]

before beginning the process so we can use value based approach. the VNF chaining,even one VNF placement makes the action space larger also,in SFC deployment we depend on real time requests so it is hard to calculate all the value functions to get the best strategy with the value based approach.

4 Adaptive, Online Approach for SFC Deployment

as defined before there are two kinds of time slots 1-inter time slot: when there is no request arriving so the reward will stay as it is and the network state also stays as it is. 2-intra time slot: in this case the network is receiving several requests in the time slot. Assume two requests are received at the same time slot [1] .

5 Comparison with other systems

in this section we show some graphs to compare between NFVdeep and other systems in terms of performance , cost and the number of servers used

6 Conclusion

involving VNFs in the NFV has improved the network functions in terms of efficiency , operational costs and convergence which then improves the performance. the main idea of NFVdeep is representing the requests as SFC.each SFC consists of number of VNFs. the NFVdeep is connected to an agent which is a deep neural network (DNN).when NFVdeep receives a reqyest the DNN calculats the new reward after taking the decision of accepting the request or rejecting it in case of timed out requests and depending on the time slot avoiding the network state transition problem.

References

[1] X. et al. *NFVdeep: Adaptive Online Service Function Chain Deployment with Deep Reinforcement Learning*. IWQoS, 2019.