Optimal Placement of Virtual Network Functions in a Distributed Network: A Deep Reinforcement Learning Based Approach

Bijoy Ahmed Saiem
Department of CSE, BUET
1905052@ugrad.cse.buet.ac.bd

Md. Raihan Sobhan
Department of CSE, BUET
1905095@ugrad.cse.buet.ac.bd

Dr. Rezwana Reaz
Department of CSE, BUET
rezwana@teacher.cse.buet.ac.bd

***Abstract*—Network Function Virtualization (NFV) enables scalable and fault-tolerant network functions by replacing dedicated hardware with virtualized instances. However, managing the state of these functions during elastic scaling and failure recovery remains challenging due to the overhead of state migration and replication. In this work, we propose DeftRL, a Deep Reinforcement Learning-based state management approach to optimize the placement of stateful Virtual Network Functions (VNFs) in a distributed network. This work is built on a prior work, called DEFT, which ensures consistency and fault tolerance in state management but selects backup nodes randomly. In contrast, DeftRL intelligently places backup VNFs based on network conditions, leveraging a policy neural network to make optimal placement decisions. It employs a two-phase commit protocol for efficient state synchronization. Experimental evaluations demonstrate that DeftRL significantly improves latency, throughput, and overload mitigation compared to DEFT, offering an adaptive and resilient approach to NFV deployment.**

***Index Terms*—Network Function Virtualization, Virtual Network Functions, Deep Reinforcement Learning**

I. Introduction

NFV has revolutionized modern networking by replacing dedicated hardware appliances with virtual network functions based on software running on commodity servers [1]. This paradigm shift improves scalability, elasticity, and cost-efficiency while enabling dynamic network management. However, managing the state of stateful VNFs remains a critical challenge, especially when handling dynamic scaling, failure recovery, and resource allocation. Ensuring efficient state synchronization and migration without excessive overhead is essential to maintain network performance and reliability.

Existing state management approaches have explored various strategies. OpenNF [1] facilitates state sharing through a central controller but suffers from a single point of failure. StatelessNF [2] and S6 [3] avoid state migration by storing states remotely, yet introduce inefficiencies due to frequent remote state access. Fault-tolerant solutions such as Pico Replication [4] and FTMB [5] rely on state replication but do not address elastic scaling and load balancing. In addition, RL-based optimization [6], [7] have been explored for the placement of VNFs, but they do not deal with implementing elastically scalable and fault-tolerant VNFs.

Motivation

Existing NFV state management techniques exhibit trade-offs between scalability and fault tolerance. While some systems prioritize state consistency, they introduce significant overhead, others minimize state migration at the cost of increased latency. RL provides a promising avenue for optimizing state replication and failure recovery, yet prior RL-driven approaches do not fully incorporate dynamic state management and fault tolerance. Addressing these gaps is crucial for enhancing NFV infrastructure resilience and efficiency.

Problem Definition

Initially, we are given a network with a set of nodes, each having resource constraints such as CPU capacity, memory availability, channel bandwidth, etc. Our goal is to determine the optimal placement of a set of primary and backup VNFs across these nodes to minimize processing and state management overhead, prevent single points of failure, and balance resource utilization while dynamically adapting to changing network conditions.

Our Contribution

DeftRl extends DEFT [8] by integrating DRL techniques to optimize VNF placement. The key contributions are:

1) We use PPO to dynamically select backup nodes based on network conditions, improving fault tolerance.
2) Our system leverages DRL for adaptive resource allocation, ensuring optimal load balancing and resilience.
3) DeftRL significantly improves latency, throughput, and overload mitigation compared to DEFT.

II. Methodology

In our study, we optimized VNF placement in a network using PPO, to minimize state migration overhead and enhance fault tolerance. The system dynamically selects backup nodes for VNFs based on network conditions, ensuring efficient resource utilization and load balancing. Key components are:
Network Topology: Incoming traffic is routed through a Stamper module, which assigns unique identifiers to packets. Packets are duplicated at the switch and sent to both primary and backup VNFs, with the DRL model dynamically selecting backup nodes to minimize synchronization overhead.

Deep Reinforcement Learning (DRL) Model: The DRL agent uses PPO to determine optimal placements for primary and backup VNFs, ensuring fault tolerance and minimizing costs. The model considers CPU capacity, memory availability, bandwidth, and latency factors to make placement decisions. The cost function includes processing, transmission, latency, and load balancing costs, while the reward function incentivizes optimal placement and penalizes inefficiencies.

Failure Detection and Recovery: A Failure Detection Unit monitors primary and backup VNFs. Upon detecting a failure, the FDU triggers the backup to take over, and the DRL model selects a new backup node. The system ensures strict consistency between primary and backup VNFs, even during failover, by maintaining state synchronization and packet order.

State Management: Each VNF maintains input and output buffers to ensure packets are processed in order, with global state updates managed through a consensus protocol(Raft) across the cluster of NFs. The transaction coordinator handles atomic commits between primary and backup VNFs, ensuring state consistency after processing each batch of packets.

Scaling and Load Balancing: When a VNF becomes overloaded, the DRL model redistributes flows to other NFs, ensuring optimal resource utilization and minimal disruption. State migration is handled via a two-phase commit protocol, ensuring no packet loss and consistency during scaling.

III. Results

Figure 1 compares the latency and throughput performance of DEFT and DeftRL. The results demonstrate that DeftRL, leveraging Deep Reinforcement Learning, significantly reduces latency and improves throughput compared to DEFT, highlighting the effectiveness of intelligent backup node selection and state management. Figure 2 illustrates the latency

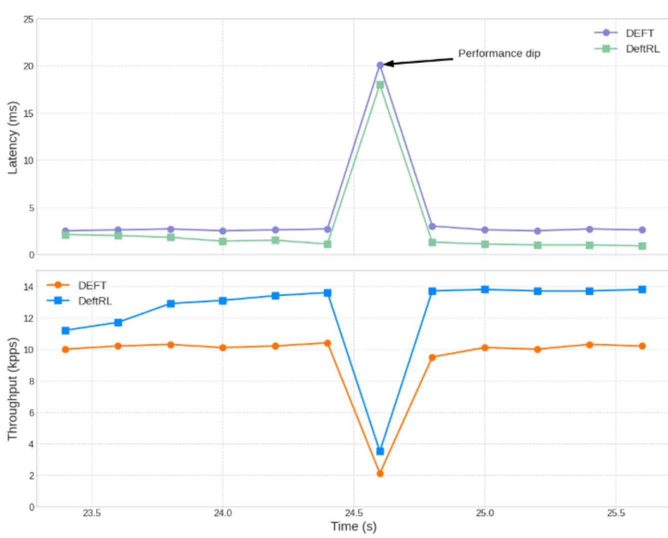

Fig. 1. Latency and Throughput Comparison: DEFT vs DeftRL

performance of various Deep Reinforcement Learning (DRL) algorithms against DEFT. It shows that DeftRL, utilizing Proximal Policy Optimization (PPO), achieves lower latency than other DRL algorithms and DEFT, emphasizing its superior efficiency in VNF placement.

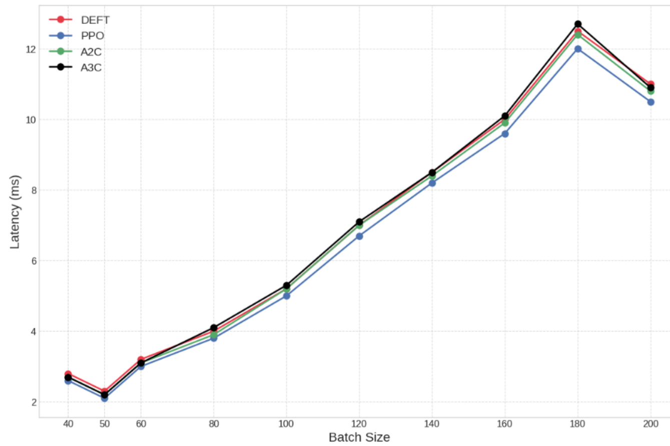

Fig. 2. Latency Comparison of DRL Algorithms with DEFT

IV. Conclusion

Our DRL-based approach, leveraging Proximal Policy Optimization, optimizes stateful VNF placement in NFV environments, significantly improving resilience, reducing state migration overhead, and enhancing performance metrics like latency and throughput. Experimental results demonstrate its superiority over traditional methods, offering intelligent, adaptive backup node selection and efficient resource allocation. Future work will focus on scalability and adaptability to dynamic, large-scale, and heterogeneous network environments.

References

[1] Gember-Jacobson, A., Viswanathan, R., Prakash, C., Grandl, R., Khalid, J., Das, S., & Akella, A. (2014). OpenNF: Enabling innovation in network function control. ACM SIGCOMM Computer Communication Review, 44(4), 163-174.

[2] Kablan, M., Caldwell, B., Han, R., Jamjoom, H., Keller, E. (2015, August). Stateless network functions. In Proceedings of the 2015 ACM SIGCOMM workshop on hot topics in middleboxes and network function virtualization (pp. 49-54).

[3] Woo, S., Sherry, J., Han, S., Moon, S., Ratnasamy, S., Shenker, S. (2018). Elastic scaling of stateful network functions. In 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18) (pp. 299-312).

[4] Rajagopalan, S., Williams, D., Jamjoom, H. (2013, October). Pico replication: A high availability framework for middleboxes. In Proceedings of the 4th annual Symposium on Cloud Computing (pp. 1-15).

[5] Sherry, J., Gao, P. X., Basu, S., Panda, A., Krishnamurthy, A., Maciocco, C., ... Shenker, S. (2015, August). Rollback-recovery for middleboxes. In Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication (pp. 227-240).

[6] Kibalya, G., Serrat, J., Gorricho, J. L., Bujjingo, D. G., Sserugunda, J., Zhang, P. (2021, May). A reinforcement learning approach for placement of stateful virtualized network functions. In 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM) (pp. 672-676). IEEE.

[7] He, N., Yang, S., Li, F., Trajanovski, S., Zhu, L., Wang, Y., Fu, X. (2023). Leveraging deep reinforcement learning with attention mechanism for virtual network function placement and routing. IEEE Transactions on Parallel and Distributed Systems, 34(4), 1186-1201.

[8] Shahriyar, M. M., Saha, G., Bhattacharjee, B., Reaz, R. (2023, October). Deft: distributed, elastic, and fault-tolerant state management of network functions. In 2023 19th International Conference on Network and Service Management (CNSM) (pp. 1-7). IEEE.