

Deep Q-Networks 기반 Service Function Chaining 구성 연구

이도영, 유재형, 홍원기

포항공과대학교 컴퓨터공학과

{dylee90, jhyoo78, jwkhong}@postech.ac.kr

A study on Deep Q-Networks-based Service Function Chaining Composition

Doyoung Lee, Jae-Hyoung Yoo, James Won-Ki Hong

Department of Computer Science and Engineering, POSTECH

요약

5G 시대에서는 급변하는 네트워크 환경 속에서 다양한 요구사항을 갖는 서비스들을 수많은 기기에게 제공하는 것이 요구된다. 이를 위해 5G 네트워크는 소프트웨어 정의 네트워킹 (SDN, Software-Defined Networking)과 네트워크 기능 가상화 (NFV, Network Function Virtualization)를 기반 기술로 활용하여, 유연하고 민첩하게 네트워크를 구축하고 운용한다. SDN/NFV는 새로운 서비스를 신속하게 네트워크에 배치할 수 있는 장점이 있으며, 특히 일련의 네트워크 기능들로 구성된 서비스인 서비스 펄스 체이닝 (SFC, Service Function Chaining) 구현을 용이하게 한다. 하지만, 한편으로는 관리가 필요한 다양한 가상 자원들을 생성하기 때문에 네트워크 및 서비스 운용을 복잡하게 하는 원인이 되기도 한다. 이를 해결하기 위해 본 논문에서는 강화학습 (RL, Reinforcement Learning) 알고리즘 중 하나인 Deep Q-Networks (DQN)을 활용한 SFC 구성 방법을 제안한다. DQN 기반 SFC 구성 방법은 동적인 네트워크 환경에서 서비스의 성능 지표인 서비스 응답 시간을 만족하는 최적의 SFC를 효과적으로 구성한다.

I. 서론

5G 시대는 급변하는 네트워크 상황 속에서 다양한 요구사항을 갖는 서비스들을 수많은 기기에게 제공하는 것이 요구되고 있다. 이를 위해, 5G 네트워크에서는 소프트웨어 정의 네트워킹 (SDN, Software-Defined Networking)과 네트워크 기능 가상화 (NFV, Network Function Virtualization)를 기반 기술로 활용하여, 네트워크를 유연하게 구축하고 운용한다. 특히, NFV는 고가의 전용 하드웨어를 통해 제공되는 네트워크 기능을 소프트웨어 형태로 가상화하여, 일반 상용 서버에서 가상 네트워크 기능 (VNF, Virtual Network Function)으로 제공하는 기술이다. SDN과 NFV는 네트워크 운용 비용을 절감하고 새로운 서비스를 네트워크에 신속하게 배치할 수 있는 장점이 있지만, 한편으로는 관리가 필요한 많은 가상 자원들을 생성하기 때문에 네트워크 관리를 복잡하게 만든다.

복잡한 네트워크 관리 문제를 해결하기 위해 최근에는 기계학습을 네트워크 운용 및 관리에 적용하는 시도가 주목받고 있다. 특히, 지도학습 (Supervised learning)과 비지도학습 (Unsupervised learning)을 통해 트래픽 분류 및 분석, 침입 탐지 등 보안 문제에 활용한 사례가 다수 존재한다 [1]. 하지만, 5G 시대의 NFV 환경에서는 동적으로 서비스를 배치하고 운용하는 것이 필요하기 때문에, 보안 뿐 아니라 VNF 배치 (VNF Deployment), 서비스 펄스 체이닝 (SFC, Service Function Chaining), 오토 스케일링 (Auto-scaling) 등 전반적인 VNF 라이프사이클 (Life-cycle) 관리 기능에 대한 연구가 요구된다. 이들 중, SFC는 네트워크 서비스를 제공하는 방법으로, 일련의 네트워크 기능들을 트래픽에 적용하는 기술이다.

현재까지 SDN/NFV 환경에서 최적 SFC를 구성하는 다양한 방법들이 제안되었지만, 복잡한 5G 네트워크의 급변하는 상황에 대응하여 서비스 요구사항을 만족하는 SFC를 운용하기 위해서는 아직 많은 연구 이슈가

존재한다. 특히, 한정된 네트워크 자원을 고려하여 서비스 성능 요구사항을 만족하는 SFC를 효과적으로 생성할 수 있어야 한다. 따라서, 본 논문에서는 강화학습 (RL, Reinforcement Learning)을 활용하여 최적 SFC를 구성하는 방법을 제안한다. 제안하는 방법은 클라우드 컴퓨팅 환경의 제한된 가용 자원을 고려하며, 서비스 성능 목표인 서비스 시간을 만족하는 SFC를 효과적으로 생성하는 것을 목표로 한다.

II. 관련 연구

SFC가 네트워크 서비스를 제공하기 위한 핵심 기술로 자리 잡으면서, SDN/NFV 기반 네트워크에서 기계학습을 활용한 SFC 구성 방법에 대한 많은 연구가 진행되었다. SFC를 구성하는 방법 중 하나는 SFC 경로를 선택하는 것으로, SFC에 포함되는 VNF들이 네트워크에 이미 배치된 상황을 가정한다. [2]에서는 네트워크에 배치된 SF (Service Function)의 CPU 사용률과 대역폭 (Bandwidth)을 고려하여, Temporal Difference (TD) 알고리즘으로 최적 SFC 경로를 선택하는 방법을 제안하였다. 또한, [3]에서는 지도학습을 활용하여 VNF가 배치된 서버들을 연결하는 링크 상태 (Link state)를 학습해 SFC를 통한 패킷 처리량 (Throughput)을 예측한 후, 이를 최적 SFC 경로 선정에 활용하였다.

SFC를 구성하는 다른 방법으로는 SFC 생성 요청이 발생했을 때, SFC에 포함될 VNF들을 새롭게 배치하여 SFC를 구성하는 것이다. [4]에서는 지도학습으로 SFC의 성능 목표를 만족시킬 수 있는 각 VNF 종류의 인스턴스 수를 예측한 후, 해당 개수의 VNF 인스턴스들을 생성하여 SFC를 구성한다. 하지만, 이 방법은 SFC를 구성하는 VNF들의 위치를 고려하지 않았기 때문에, 이를 개선하기 위한 [5]가 제안되었다. [5]에서는 그래프 인공 신경망 (GNN, Graph Neural Network)을 활용하여, SFC를 구성할

VNF가 설치될 서버 위치를 결정한 후, 해당 서버들에 VNF 인스턴스들을 설치하여 SFC를 구성한다.

앞서 설명한 SFC 구성 방법들 중, SFC 경로를 선택하는 방법은 새로운 VNF 인스턴스를 생성하지 않기 때문에 VNF 배치 비용 측면에서 효과적인 SFC 구성이 가능하다. 하지만, 배치된 VNF 상태에 의존하여 SFC를 구성하기 때문에 유연한 SFC 구성에는 한계가 있다. 반면, VNF를 새롭게 배치하여 SFC를 구성하는 방법은 요구사항에 따른 최적 SFC 구성이 가능하지만, 구성 비용이 크다는 단점이 있다. 또한, 네트워크 자원은 한정되어 있기 때문에 수많은 기기들이 접속하는 5G 시대에서는 주어진 자원을 효과적으로 각 서비스들에게 할당하는 것이 요구된다. 즉, SFC 구성을 위해 이미 배치된 VNF를 공유하거나 [6], 새롭게 배치하는 방법을 같이 적용할 필요가 있다.

III. Deep Q-Networks 기반 SFC 구성 방법

본 논문에서 제안하는 DQN 기반 SFC 구성 방법은 클라우드 컴퓨팅 환경의 제한된 가용 자원을 고려하여 서비스 성능 목표를 만족하는 SFC를 구성한다. DQN은 심층 강화학습 방법 중 하나로, 인공 신경망 (Neural network)을 통해 수 많은 상태 (State)가 존재하는 최적화 문제를 설계할 수 있다. [그림 1]은 SFC 구성을 위한 DQN 모델을 나타낸다. DQN 모델은 SFC를 구성하는 VNF의 종류와 순서가 명시된 SFC 요청이 있을 때, VNF 인스턴스들을 선택 또는 생성하는 과정을 반복하여 SFC를 구성한다. 즉, 이미 배치된 VNF 인스턴스를 활용하여 성능 목표를 달성할 수 있다면, 해당 인스턴스로 SFC를 구성하기 때문에 VNF 생성 비용이 들지 않아 제한된 자원 환경 속에서 효율적으로 SFC를 생성한다. DQN 모델은 2개의 은닉층을 가지며, 활성화 함수 (Activation function)로 ReLU (Rectified Linear Unit)를 사용한다.

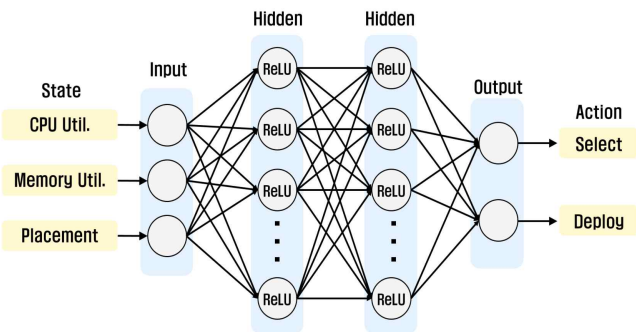


그림 1. Deep Q-networks (DQN) 기반 SFC 구성 모델

DQN 모델의 입력 계층에서는 SFC를 구성할 VNF 인스턴스들의 평균 CPU 사용률 (CPU Util.), 메모리 사용률 (Memory Util.), 배치 상태 (Placement)로 구성된 현재 상태를 입력값으로 받는다. CPU와 메모리는 VNF 인스턴스에서 패킷 처리에 사용되는 자원들이며, 배치 위치는 SFC에서 처리되는 패킷들의 경로로 인한 전파 지연 시간 (Propagation delay)에 영향을 미친다. VNF 인스턴스들의 배치 상태는 이전 순서에 선택된 VNF 인스턴스로부터 다음 순서의 VNF 인스턴스들 사이의 평균 거리를 Hop으로 계산한 값이다. 따라서 DQN 모델의 상태는 선택 대상이 되는 인스턴스들의 패킷 처리 성능과 위치를 고려하여 SFC를 구성한다.

제안하는 모델의 출력 계층에서는 SFC를 구성할 VNF를 기존에 존재하는 VNF 인스턴스들 중 선택할지 (Select), 또는 새로운 VNF 인스턴스를 생성할지 (Deploy)에 대한 행동 값 (Action value)을 출력한다. SFC 구성을 위

한 VNF 인스턴스를 선택하는 행동을 결정했을 경우, CPU와 메모리 사용률이 작고, 이전 순서에 선택된 VNF 인스턴스와 가까운 인스턴스를 선택한다. 반면, VNF 인스턴스를 생성하는 행동을 결정했을 때는 이전 순서에 선택된 VNF 인스턴스와 가까운 서버들 중, 가용 자원이 충분한 위치를 선택하여 새로운 인스턴스를 생성한다. SFC를 구성할 때, 기존에 존재하는 VNF 인스턴스들 중 하나를 선택하는 것과 새로운 VNF 인스턴스를 생성하는 것은 SFC 성능과 구성 비용 측면에서 장단점이 존재하기 때문에 이를 고려한 행동 결정이 필요하다. 따라서 본 논문에서는 수식 (1)과 같은 보상 모델을 정의하였다.

$$Reward = -\ln(1 + res\ Time) \times Cost_{deploy} \quad (1)$$

DQN 모델은 SFC 구성을 위한 최적 정책 (Policy)을 학습하기 위해 행동을 결정할 때마다 그에 대한 보상 (Reward)을 얻는다. 제안하는 방법의 목표는 적은 자원과 비용으로 서비스 성능 요구사항을 만족하는 SFC를 구성하는 것이다. 네트워크 서비스에서 성능을 측정하기 위한 주요 지표 중 하나는 응답 시간이기 때문에 수식 (1)에서는 응답 시간 (resTime)을 고려한다. 응답 시간은 사용자가 생성한 패킷이 VNF 인스턴스에 의해 처리된 후, 그에 대한 응답을 사용자가 받기까지 소요된 시간이다. 제안하는 방법에서 DQN 모델의 행동은 다음 순서의 VNF 인스턴스를 선택하는 것이기 때문에, 보상 계산을 위한 응답 시간은 이전 순서에 선택된 VNF 인스턴스로부터 새롭게 선택 또는 생성된 VNF 인스턴스로 Ping 메시지를 전달하여 millisecond 단위로 측정한다. 또한, 인스턴스 생성에 요구되는 비용은 가중치 ($Cost_{deploy}$)를 통해 측정된 서비스 성능, 즉 응답 시간을 보정한다. 예를 들어, VNF 인스턴스를 선택했을 때와 생성했을 때 동일한 응답 시간이 측정되었을 경우, 인스턴스를 생성했을 때 받는 보상은 가중치 ($Cost_{deploy}$)로 1.2를 곱하여 감소시킨다. 이와 같은 보상 모델을 통하여 SFC 구성을 위한 서비스 성능과 구성 비용을 함께 고려할 수 있다. 마지막으로, 본 논문의 DQN 모델 학습 과정 중, 특정 상황에서 수행할 행동을 선택할 때는 ϵ -greedy 알고리즘을 활용한다.

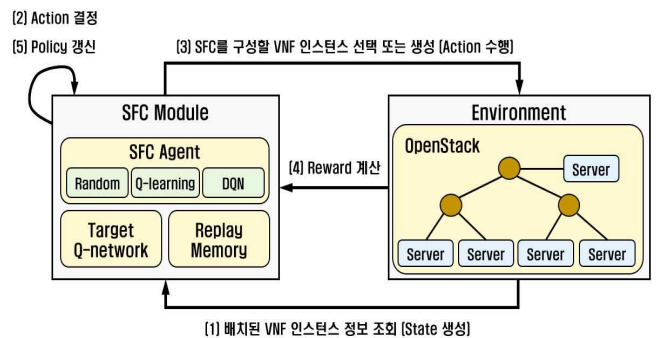


그림 2. SFC 구성 모듈

제안하는 DQN 기반 SFC 구성 방법은 향후 서술할 OpenStack 기반 실험환경에서 동작하는 모듈로 구현되었다. [그림 2]는 SFC 모듈이 동작하는 과정을 보인다. SFC 모듈은 SFC 구성 요청 메시지를 받았을 때, 해당 SFC를 구성하기 위해 환경 (Environment)에 배치된 VNF 인스턴스 정보를 조회한다. 그리고, SFC 구성을 위한 행동을 결정하여 VNF 인스턴스를 SFC에 포함시키고 행동에 대한 보상을 받는다. SFC 구성이 완료될 때까지 VNF 인스턴스 선택 또는 생성을 반복하며, 주기적으로 SFC 구성 정책을 갱신한다. 또한, SFC 모듈이 DQN 알고리즘을 사용할 때, 안정적인 학습을 위해 [7]에서 제안한 Replay memory와 Target Q-network를

활용한다. Replay memory와 Target Q-network는 강화학습 알고리즘의 학습 과정에서 순차적인 행동과 그에 대한 보상 값으로 정책을 갱신할 경우 각 행동들의 상관 관계 (Correlation)로 인해 최적 정책을 찾지 못하는 문제를 해결한다. 구현된 SFC 모듈은 GitHub을 통해 공개되어있다 [8].

IV. 성능 평가

DQN 기반 SFC 구성 방법의 성능 평가를 위해 [그림 3]과 같이 OpenStack 기반의 실험환경을 구축하였다. 실험환경은 OpenStack 환경을 관리하는 컨트롤러 노드 (Controller Node)와 VNF 인스턴스가 배치되고 SFC가 구성되는 9대의 컴퓨터 노드 (Compute Node)로 구성된다. 또한, 본 논문에서 제안하는 DQN 기반 SFC 구성 알고리즘은 AI 노드에서 실행되며, 알고리즘에 요구되는 모니터링 데이터들은 모니터링 노드 (Monitoring Node)를 통해 수집된다.

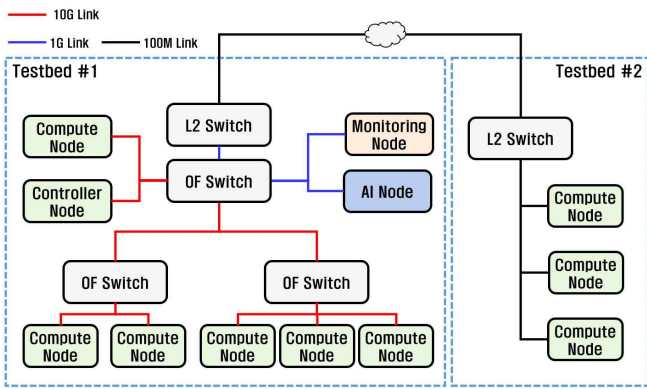


그림 3. SFC 구성 성능 평가 실험환경

성능 평가 시나리오는 [그림 4]와 같다. 성능 평가 시나리오는 미리 구축된 실험환경에서 5개의 VNF로 구성된 SFC를 생성하고, 클라이언트 가상 머신 (VM, Virtual Machine)에서 SFC를 통과하는 패킷을 생성한다. 이를 위해, 패킷의 목적지가 되는 Web VM 2개를 특정 컴퓨터 노드에 생성한 후, 임의의 컴퓨터 노드에 클라이언트 VM을 생성한다. 성능 평가는 클라이언트 VM이 생성한 패킷에 대한 서비스 시간 (Service time)을 측정하여 비교한다. 서비스 시간은 사용자로부터 생성된 패킷이 SFC를 구성하는 모든 VNF 인스턴스에서 처리되어 목적지에 도달한 후, 사용자가 다시 응답 패킷을 받기까지 소요된 시간을 의미한다. 따라서 서비스 시간은 SFC를 구성하는 VNF 인스턴스의 패킷 처리 성능과 각 VNF 인스턴스 사이에서 패킷이 전달되는 시간을 포함한다.

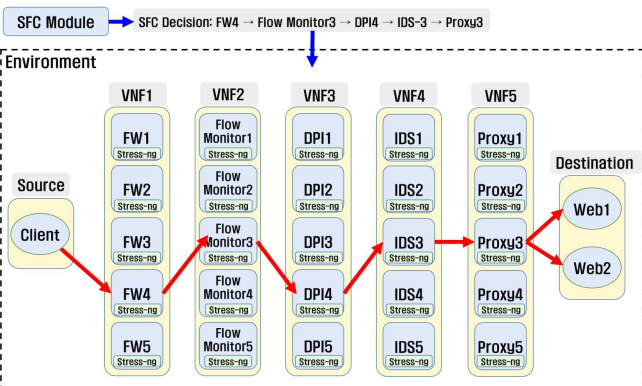


그림 4. SFC 구성 성능 평가 시나리오

SFC는 Firewall, Flow monitor, Deep Packet Inspection (DPI), Intrusion Detection System (IDS), Proxy 기능을 가진 VM들로 구성된다. 각 VNF 기능은 오픈소스 소프트웨어인 Iptables [9], ntopng [10], nDPI [11], Suricata [12], HAProxy [13]를 설치해 구현하였다. VNF 종류마다 5개의 VM들이 실험환경 내 임의의 위치에 미리 배치되어 있으며, 각 VM들은 Stress-ng [14]를 사용하여 동적으로 변하는 임의의 자원 사용률을 가지도록 설정하였다.

표 1. Q-Learning 및 DQN 초기 변수 정보

변수 종류	값
[DQN/Q-learning] η (학습률)	0.01
[DQN/Q-learning] γ (할인율)	0.98
[DQN/Q-learning] ϵ (탐험 확률)	0.10
[DQN] Sampling 크기 (Mini-batch)	16

성능 평가 실험에서는 3개의 서로 다른 SFC 구성 방법을 사용하여 SFC를 생성한다. 첫 번째 방법은 기존에 존재하는 VM들, 즉 VNF 인스턴스들을 임의로 선택하여 SFC를 생성한다 (Random 방법). 두 번째 방법은 본 저자들의 선행 연구 [15]에서 제안한 Q-learning 기반의 SFC 구성 방법 (Q-learning 방법)으로, 미리 배치된 VNF 인스턴스들의 자원 사용량과 배치 위치를 Q-learning으로 학습하여 SFC를 구성한다. 세 번째 방법은 본 논문에서 제안하는 DQN 모델을 기반으로 VNF 인스턴스들을 선택 또는 생성하여 SFC를 구성한다 (DQN 방법). 이 때, Q-learning과 DQN은 학습을 위한 매개변수의 할당이 필요하다. 따라서, 성능 평가 실험에서 Q-learning과 DQN 방법은 (표 1)의 값들을 사용한다.

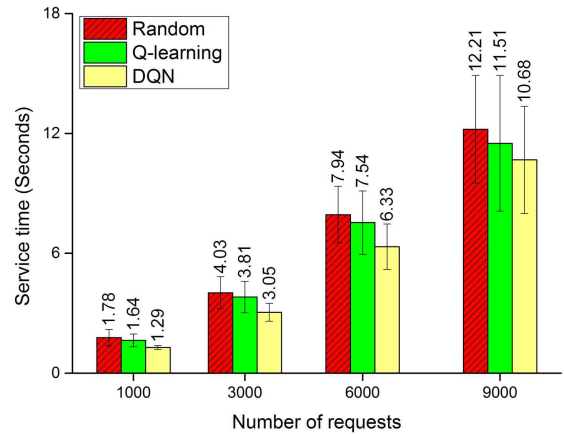


그림 5. SFC를 통한 서비스 시간 측정 결과

모든 설정을 마친 후, Q-learning과 DQN은 3000번 이상 행동을 수행하여 학습한 모델을 통해 성능 평가를 수행하였다. 또한, SFC 구성 방법들을 통해 각 SFC를 구성할 때마다, 클라이언트 VM에서 1,000개, 3,000개, 6,000개, 그리고 9,000개의 HTTP 요청 메시지를 발생시킨 후, 이에 대한 모든 응답을 받기까지 소요된 서비스 시간을 측정하였다. [그림 5]는 각 SFC 구성 방법들을 통해 SFC 구성을 100번 반복하여 계산한 평균 서비스 시간을 보인다. 전체적으로 HTTP 요청 메시지 개수가 늘어날수록 SFC에서 측정되는 서비스 시간도 증가하였으나, DQN 방법으로 SFC를 구성했을 때 가장 짧은 평균 서비스 시간이 측정되었다. 또한, 응답 시간의 표준편차도 DQN으로 SFC를 구성했을 때 가장 작았으며, 이를 통해 본 논문에서 제안하는 방법은 짧고 안정적인 서비스 시간을 제공하는

SFC를 구성할 수 있음을 확인하였다.

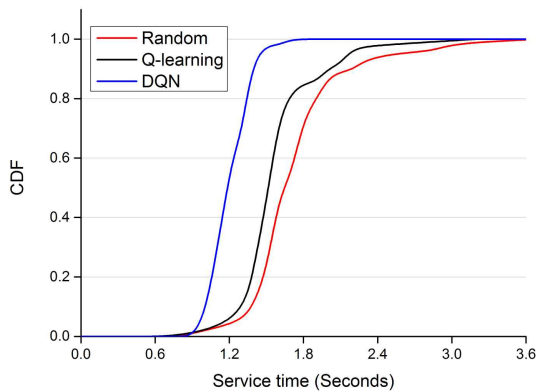


그림 6. SFC 서비스 시간 누적 분포 함수 결과

[그림 6]은 서로 다른 SFC 구성 방법을 100번씩 반복해 생성한 각 SFC에서 HTTP 요청 메시지 1,000개를 발생시켜 측정된 서비스 시간을 누적 분포 함수 (CDF, Cumulative Distribution Function)로 나타낸 결과이다. 이를 통해, DQN 기반 SFC 구성 방법은 95% 이상으로 SFC에서 1.4초 이내의 서비스 시간을 제공하는 것을 확인하였다. 하지만, Random 방법에서는 20%의 SFC와 Q-learning에서는 43%의 SFC만이 1.4초 내의 서비스 시간을 제공하였다. 또한, 제안하는 DQN 기반 SFC 구성 방법에서 현재 배치된 VNF 인스턴스들의 자원 사용률과 배치 상태로 인해 짧은 서비스 시간 보장이 힘들다고 판단할 경우, VNF 인스턴스 1~2개를 새롭게 배치하여 SFC를 구성하였다. 따라서, 제안하는 방식은 최소한의 VNF 인스턴스 추가 비용으로 짧은 서비스 시간을 갖는 SFC를 효과적으로 구성할 수 있다는 것을 확인하였다.

V. 결론 및 향후 연구

본 논문에서는 심층 강화학습 알고리즘인 DQN을 활용하여 클라우드 컴퓨팅 환경에서의 최적 SFC 구성 방법을 제안하였다. 제안하는 방법은 SFC 구성 요청사항을 바탕으로 SFC의 성능 목표인 서비스 시간을 만족 시킴과 동시에, 구성 비용을 최소화하는 SFC를 구성한다. 성능 평가 결과, DQN 기반 SFC 구성 방법은 현재 배치된 VNF 인스턴스들의 자원 사용률과 위치를 고려하여 짧은 서비스 시간을 보장하는 SFC를 구성할 수 있었다. 하지만, 제안하는 방법에서 DQN은 VNF 인스턴스와 관련된 데이터만 사용해 학습을 수행하기 때문에, 네트워크 링크 상태, 혼잡 상황 등 네트워크 환경에 대응한 SFC 구성에는 한계가 존재한다.

따라서, 향후 연구로는 동적으로 변하는 네트워크 환경을 고려한 SFC 구성 모델을 설계할 예정이다. 또한, 본 논문의 DQN 모델에서는 간단한 인공 신경망으로 구성되었지만, 향후 개선될 모델에서는 순환 신경망 (RNN, Recurrent Neural Network)을 활용하여, VNF 인스턴스들의 패킷 처리 성능의 추세를 SFC 구성에 고려할 것이다. 마지막으로 다양한 시나리오에서 SFC 구성 방법에 대한 성능 평가를 수행할 계획이다.

ACKNOWLEDGMENT

이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(2018-0-00749, 인공지능 기반 가상 네트워크 관리기술 개발)과 대학ICT연구센터육성지원사업의 연구결과로 수행되었음 (IITP-2021-2017-0-01633).

참고 문헌

- [1] Boutaba, R., Salahuddin, M. A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., & Caicedo, O. M. (2018). A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications*, 9(1), 1-99.
- [2] Ku, Hye-Jin, J. H. Jung, and Gu-In Kwon. "A study on reinforcement learning based SFC path selection in SDN/NFV." *International Journal of Applied Engineering Research* 12, no. 12 (2017): 3439-3443.
- [3] Jeong, Seyeon, Heegon Kim, Jae-Hyoung Yoo, and James Won-Ki Hong. "Machine Learning based Link State Aware Service Function Chaining." In *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 1-4. IEEE, 2019.
- [4] Lange, S., Kim, H. G., Jeong, S. Y., Choi, H., Yoo, J. H., & Hong, J. W. K. (2019, October). Predicting vnf deployment decisions under dynamically changing network conditions. In *2019 15th International Conference on Network and Service Management (CNSM)* (pp. 1-9). IEEE.
- [5] Kim, H. G., Park, S., Heo, D., Lange, S., Choi, H., Yoo, J. H., & Hong, J. W. K. (2020, November). Graph Neural Network-based Virtual Network Function Deployment Prediction. In *2020 16th International Conference on Network and Service Management (CNSM)* (pp. 1-7). IEEE.
- [6] Mohamad, A., & Hassanein, H. S. (2020, November). PSVShare: A Priority-based SFC placement with VNF Sharing. In *2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)* (pp. 25-30). IEEE.
- [7] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540), 529-533.
- [8] DPNM, Network Intelligent Project. [Online]. Available: <https://github.com/dpnm-ni/ni-sfc-path-module-public>
- [9] Purdy, Gregor N. *Linux iptables Pocket Reference: Firewalls, NAT & Accounting*. "O'Reilly Media, Inc.", 2004.
- [10] Deri, Luca, Maurizio Martinelli, and Alfredo Cardigliano. "Realtime high-speed network traffic monitoring using ntopng." *28th large installation system administration conference (LISA14)*. 2014.
- [11] Deri, Luca, et al. "ndpi: Open-source high-speed deep packet inspection." *2014 International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2014.
- [12] Open Information Security Foundation, "Suricata: Open Source IDS/IPS/NSM engine", [Online]. Available: <https://suricata-ids.org/>
- [13] HAProxy, "HAProxy: The Reliable, High Performance TCP/HTTP Load Balancer", [Online]. Available: <http://www.haproxy.org/>
- [14] King, Colin Ian. "Stress-ng." URL: <http://kernel.ubuntu.com/git/cking/stressng.git/> (visited on 28/03/2018) (2017)
- [15] Doyoung Lee, Jae-Hyoung Yoo, James Won-Ki Hong, "Q-learning Based Service Function Chaining Using VNF Resource-aware Reward Model", *21st Asia-Pacific Network Operations and Management Symposium (APNOMS 2020)*, Daegu, Korea, Sep. 23-25, 2020.