

Q-Learning 기반 VNF 자원 인식 Service Function Chaining

이도영, 유재형, 홍원기

포항공과대학교 컴퓨터공학과

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

Q-Learning based VNF Resource-aware Service Function Chaining

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

Department of Computer Science and Engineering, POSTECH

요약

오늘 날 5G 네트워크 시대가 도래 하면서, 급변하는 서비스 요구사항을 만족시키기 위해 유연하고 민첩하게 네트워크를 구축하고 관리하는 것이 요구되고 있다. 소프트웨어 정의 네트워킹 (SDN, Software-Defined Networking)과 네트워크 기능 가상화 (NFV, Network Function Virtualization)는 네트워크를 소프트웨어 기반으로 전환하여 유연한 네트워크 관리를 가능케 하는 핵심 기술들이다. 특히, NFV는 네트워크 기능들을 소프트웨어 형태로 가상화하여 상용 서버에서 운영하며, 동적으로 네트워크 기능을 관리할 수 있는 장점이 있다. 하지만, 한편으로는 물리 네트워크 자원 뿐 아니라 수 많은 가상 네트워크 및 자원들로 인해 네트워크 관리를 복잡하게 만드는 원인이 된다. 이를 위해 최근에는 인공지능 기술을 도입하여 복잡한 NFV 환경 및 기술들을 관리하는 연구가 주목받고 있다. 특히, 서비스 평선 체이닝 (SFC, Service Function Chaining)은 필수 NFV 기술 중 하나로, 효율적인 SFC를 생성하는 것이 요구된다. 본 논문에서는 인공지능 기술 중 하나인 강화학습 (RL, Reinforcement Learning)을 활용하여 VNF 자원 사용량을 고려한 최적 SFC 경로를 찾는 방법을 제안한다.

I. 서론

네트워크 기능 가상화 (NFV, Network Function Virtualization)는 전용 하드웨어를 통해 제공되는 네트워크 기능을 소프트웨어 형태로 가상화하여 상용 서버에서 운영하는 기술이다. 5G 네트워크에서 NFV는 소프트웨어 정의 네트워킹 (SDN, Software-Defined Networking)과 함께 신속하고 유연하게 네트워크를 관리하여 서비스를 제공할 수 있는 핵심 기술로 활용되고 있다. NFV는 서비스 요구사항에 따라 동적으로 네트워크 기능을 VNF (Virtual Network Function) 형태로 배치할 수 있지만, 한편으로는 수 많은 가상 네트워크와 자원들을 관리하게 만들어, 네트워크 관리를 복잡하게 만드는 원인이 된다. 또한, 서비스 요구사항이 급변하고 다양한 기기들이 네트워크에 연결되는 5G 네트워크 시대에, 네트워크는 더욱 복잡해지고 있으며 이는 관리자에 의한 네트워크 운영 및 관리를 어렵게 만들고 있다.

복잡해지는 네트워크 관리 문제를 해결하기 위해 최근에는 인공지능 기술을 네트워크 관리에 접목하려는 연구가 주목받고 있다. 기존에는 주로 기계학습 알고리즘들을 침입 탐지, 비정상 트래픽 분류 등을 위해 활용하였지만, 오늘 날에는 다양한 인공지능 기술들을 VNF 배치 (VNF Deployment), 서비스 평선 체이닝 (SFC, Service Function Chaining), 오토 스케일링 (Auto-Scaling), 마이그레이션 (Migration) 등 전반적인 NFV 관리 기술에 적용하기 위한 연구가 진행되고 있다. 그 중에서도 SFC는 네트워크 트래픽에 일련의 네트워크 기능을 적용할 수 있도록 트래픽 경로를 제어해 VNF를 경유시키는 기술로써, QoS를 보장하는 최적의 SFC 경로를 선택하는 것이 요구된다.

SDN/NFV를 활용해 최적의 SFC 경로를 찾는 방법에 대한 많은 연구가 진행되어 왔지만, 동적으로 변화하는 네트워크 환경에 대응하는 SFC 생

성 방법에 대해서는 아직 많은 연구 이슈가 존재한다. 이를 위해 본 논문에서는 인공지능 기술 중 하나인 강화학습 (RL, Reinforcement Learning)을 활용하여 SFC를 구성하는 VNF들의 현재 자원 사용률과 VNF 설치 위치를 고려한 SFC 경로 선택 방법을 제안한다. 제안하는 방법은 OpenStack 기반 테스트베드에서 실제 VNF들을 설치한 후, 임의로 선택한 SFC 경로보다 강화학습 기법 중 하나인 Q-Learning을 사용해서 SFC 경로를 선택했을 때, 패킷 응답 시간 측면에서 더 효과적인 경로인 것을 검증하였다.

II. 관련 연구

최근까지 인공지능 기술을 SDN/NFV 기반 네트워크 환경의 효율적인 관리를 위해 활용하는 연구가 활발하게 진행되고 있다. 그 중, [1]에서는 대용량 데이터 전송 트래픽이 흐르는 SDN 환경에서 네트워크 혼잡 (Congestion)을 피하기 위해 Flow 경로 설정에 강화학습 Q-Learning 알고리즘을 활용하였다. 또한, [2]에서는 강화학습 중 Temporal Difference (TD) 알고리즘을 Q-value에 적용해 최적의 SFC 경로를 선택하는 방법을 제안하였다. 이 때, SFC를 구성하는 SF (Service Function)의 현재 CPU 사용량과 대역폭 (Bandwidth)을 고려해서 보상 (Reward) 값을 책정하였으며, 보상이 가장 높은 SF들을 선택하여 SFC를 구성하였다. 그 외 [3]에서는 Q-Learning을 사용하여, SFC를 구성하는 SF를 물리 노드 (Physical Node)에 분산 배치시키는 Load-balancing 방법을 제안하였다. 마지막으로 [4]에서는 지도학습 (Supervised Learning) 알고리즘인 RFR (Random Forest Regression)과 FNN (Feed-forward Neural Network)으로 SFC를 구성하는 링크 상태 (Link State)를 학습하여 SFC를 통한 Throughput을 예측하는 방법을 제안하였다.

위 연구들은 인공지능 기술을 SFC 생성에 활용할 수 있는 방향을 제시하였지만, 검증을 위한 실험 환경은 Mininet 또는 자체적으로 구현한 시뮬레이션을 통해 구축되었다는 한계가 존재한다. 한편, ETSI에서는 NFV 구조와 SFC 기능 표준화도 지속적으로 추진하고 있는데, [5]에서는 SFC 기능을 제공하는 ETSI 표준에 부합하는 프레임워크가 없음을 지적하고 있다. 따라서 강화학습을 활용한 향후 SFC 연구들은 최적의 SFC를 생성하는 문제 뿐 아니라 ETSI 표준 프레임워크에서 동작할 수 있는 형태로 구현하는 것도 고려할 필요가 있다.

III. Q-Learning 기반 SFC 경로 선택

강화학습은 인공지능 기술 중 하나로, 최대의 누적 보상을 주는 행동(Action)들을 수행할 수 있도록 시행착오를 거쳐 최적의 정책(Policy)을 찾는 학습 방법이다. 일반적으로 강화학습은 에이전트(Agent)와 환경(Environment)으로 구성되며, 에이전트는 정책에 따라 현재 상태(State)에서 특정 행동을 수행하게 된다. 이를 통해 다음 상태로 이동하게 되며, 동시에 보상을 얻게 된다. 강화학습의 목적은 각 상태에서 다양한 행동을 수행하며 보상을 최대화하는 정책을 찾는 것이기 때문에 상태와 행동, 그에 대한 보상의 정의가 필요하다.

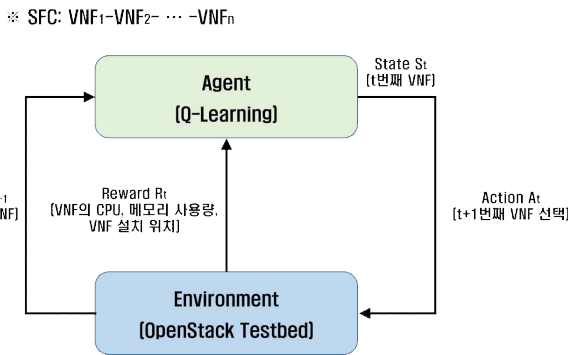


그림 1 강화학습 기반 SFC 경로 선택 모델

본 논문에서 제안하는 강화학습 기반 SFC 경로 선택 방법은 [그림 1]처럼 미로 찾기와 유사한 방법으로 단순화하여 문제를 정의한다. 미로 찾기 문제에서는 현재 미로 내 위치를 상태로 정의하며, 어느 방향으로 이동할 것인지 행동으로 정의한다. 그리고 시행착오를 거쳐 목적지에 도달하게 되면 미로를 탈출했다는 보상을 준다. 제안하는 강화학습 기반 SFC 경로 선택 문제도 이와 유사하게 SFC를 구성하는 각 VNF를 패킷이 통과해야 하는 상태로 정의하고, 다수의 VNF 중 어떤 것을 다음 VNF로 선택할지를 행동으로 정의한다. 그리고 미로에서 탈출구에 도달하는 것처럼 SFC를 구성하는 VNF들을 순서대로 올바르게 선택했을 때 보상을 준다. 다만, SFC 경로 선택에서는 단순히 VNF들을 순서대로 선택한 것만으로 보상을 주는 것이 아니라, 최적의 성능을 제공하는 VNF를 선택해 SFC 경로를 구성 했는지 여부도 가중되어 보상으로 고려되어야 한다. 본 논문에서는 SFC를 구성할 때, 각 VNF들의 현재 CPU 사용량 및 메모리 사용량, 그리고 각 VNF가 설치된 물리 노드의 위치를 고려하여 보상으로 책정한다. CPU와 메모리는 패킷 처리 성능에 영향을 미치는 자원들로, 각 자원들이 충분치 않으면 패킷 처리가 지연(Delay)되거나 이로 인해 패킷 손실(Packet loss)이 발생할 수 있다 [6, 7]. 또한, VNF가 설치된 물리 노드 위치는 SFC 경로를 지나는 패킷 전달 시간과 간접적으로 관련이 있는 요소이다. SFC를 구성하는 VNF들이 같은 물리 노드에 위치해 있을 경우, 패킷들은 다른 물리 노드로 전달 될 필요 없이 동일한 물리 노드 내에서만

이동하기 때문에 VNF 간 패킷 전달 시간이 감소한다.

오늘 날 존재하는 여러 가지 강화학습 알고리즘 중, 본 논문에서는 Q-Learning을 최적의 SFC 경로를 선택하는 데 활용한다. Q-Learning은 마르코프 결정 과정(MDP, Markov Decision Process)의 최적 정책을 찾기 위해 활용할 수 있는 강화학습 방법이다. 이 때, 현재 상태에서 특정 행동을 수행할 때 얻을 수 있는 보상을 예측하는 지표인 Q값을 반복적으로 학습하여 특정 상태에서 어떤 행동을 수행하는 것이 좋을지 나타내는 정책으로 사용한다. 제안하는 방법에서 Q-Learning의 Q값 갱신 과정은 수식 (1)과 같이 표현할 수 있다. SFC 경로 선택 문제에서 S_t 는 패킷이 지나가는 SFC의 VNF를 의미하며, A_t 는 다음 VNF로 이동하는 행동을 의미한다. R_t 은 특정 상태에서 선택한 행동으로 인해 받을 수 있는 보상, 즉 선택된 다음 VNF의 자원 사용량과 물리 노드 위치에 따라 책정된 보상 값을 의미한다. 이 외에 η 은 학습률(Learning rate), γ 은 할인율(Discount factor)을 의미한다. 결과적으로, 제안하는 방법은 수식 (1)에 따라서 자원 사용량 및 물리 노드 위치를 고려해 최적의 SFC를 구성할 수 있는 VNF들을 선택해 경로를 생성하는 것이다.

$$Q(S_t, A_t) = \eta(R + \gamma \max(Q(S_{t+1}, A))) \quad (1)$$

또한, Q-Learning 수행 중, 특정 상황에서 행동을 선택할 때, 어떤 비율로 탐사(Exploitation)와 탐험(Exploration)을 결정할지 고려하는 것이 필요하다. 일반적으로 탐사는 특정 경우를 우선시하여 정책을 찾는 방식이고, 탐험은 모든 경우를 고려하여 최적의 정책을 찾기 위한 방식이다. 즉, Q-Learning에서 Q값이 가장 높은 행동을 선택하는 것이 탐사이며, 임의로 다른 행동을 선택하는 게 탐험이다. 본 논문에서는 학습 초기에 탐험 비중을 더 중시해 다양한 VNF를 선택하여 최적 SFC 경로를 찾는 데, 이를 위해 ϵ -greedy 알고리즘을 활용한다. ϵ -greedy 알고리즘은 ϵ 값을 기준으로 탐험과 탐사 중 어떤 것을 수행할지 결정하는 것이며, ϵ 값이 클수록 탐험을 선택할 확률이 커진다. 제안하는 방법에서는 ϵ 값을 매 학습 때마다 일정 비율을 감소시켜서, 탐험을 할 확률을 점차 낮춘다.

알고리즘 1. Q-Learning 기반 SFC 경로 선택

초기화 (Initialization) 과정:

- Q-value \leftarrow 모든 Q-value 0으로 초기화
- R-value \leftarrow 각 State를 선택할 때 받는 보상
 - ※ CPU 및 Memory utilization,
 - VNF 설치 Node 위치로 보상 책정
- $(\eta, \gamma, \epsilon) \leftarrow$ 변수 초기화 ※ (학습률, 할인율, ϵ -greedy 값)

Q-Learning 과정:

- iteration_num \leftarrow n
- Episode \leftarrow 1
- while (Episode \leq iteration_num) ※ n번 학습 반복
 - $\epsilon \leftarrow$ ϵ -greedy 값 일정 비율 감소
 - while (SFC path is not decided) ※ SFC 경로 선택
 - $A_t \leftarrow a$ ※ 현재 상태 S_t 에서 수행할 행동 결정
 - $S_{t+1} \leftarrow (S_t, A_t)$ ※ S_{t+1} 로 이동하는 다음 VNF 선택
 - Q-value \leftarrow 새로운 Q-value 갱신 ※ 수식 (1)
- Episode \leftarrow Episode + 1

SFC 생성 과정:

- SFC_path \leftarrow S_0 부터 Q-Value가 큰 행동을 선택, VNF 결정

본 논문에서 제안하는 Q-Learning 기반 SFC 경로 선택 알고리즘의 전체적인 수행과정은 알고리즘 1과 같다. 학습에 앞서, 특정 상황에서 어떤 행동을 선택할지 결정하는 정책인 Q값은 0으로 초기화하며, 현재 실행 중인 각 VNF들의 자원 정보 (CPU 및 메모리 사용량)와 배치된 물리 노드 위치를 환경 (Environment)로부터 받아온 후 보상 값으로 책정하여 저장한다. 또한, 학습 과정에서 사용되는 각 변수의 초기 값도 설정한다. 다음 순서로는 iteration_num로 정의된 횟수만큼 SFC 경로를 찾아가는 과정에서 Q-Learning을 반복하며 Q값을 갱신하는 것이다. 학습 과정에서 초기 ϵ -greedy 값은 큰 값을 할당하여 학습이 반복될 때마다 일정 비율로 감소시킨다. 이는 학습 초기에는 Q값에 근거한 행동 선택보다 임의의 행동을 선택하는 탐험 위주로 다양한 VNF를 선택하며 이에 대한 보상을 얻기 위해서이다. 이후, 학습이 일정 수준 반복되었을 때에는 ϵ 값이 초기 값보다 작아졌기 때문에 탐험보다는 Q값에 근거하여 최적의 행동을 선택하게 된다. 모든 학습을 마친 후에는 계산된 Q값들을 활용하여, 초기 상태 (S_0), 즉, SFC를 경유해야 하는 패킷이 트래픽 분류기 (Traffic Classifier)에 도착했을 때부터 어떤 VNF들로 흘러야 하는지 SFC 경로를 결정한다. 최종 결정된 SFC 경로는 실제 네트워크 환경에 SFC로 구성하여 Q-Learning 기반 SFC 경로 선택 및 생성까지 마무리된다.

IV. 성능 평가

Q-Learning 기반 SFC 경로 선택 방법의 성능 검증을 위해 [그림 2]와 같이 OpenStack 기반으로 테스트베드를 구축하였다. 테스트베드는 OpenStack 환경을 관리하는 컨트롤러 노드 (Controller Node)와 VNF이 설치될 수 있는 4개의 컴퓨트 노드 (Compute Node)로 구성된다. 테스트베드 내에 VNF를 설치할 때에는 임의의 컴퓨터 노드를 선택하여 VNF가 설치된 VM 인스턴스를 생성하며, 각 VM 인스턴스에는 하나의 VNF만 동작한다. 본 성능 평가에서는 모든 VM 인스턴스들을 미리 테스트베드에 생성해 놓은 상태에서 SFC를 구성하는 VM 인스턴스들을 선택해 SFC를 구성한다. 또한, 테스트베드 내 위치한 모니터링 노드 (Monitoring Node)는 실시간으로 테스트베드 내 생성된 VNF의 자원 사용량 정보를 모니터링하여 시계열 데이터베이스에 저장한다.

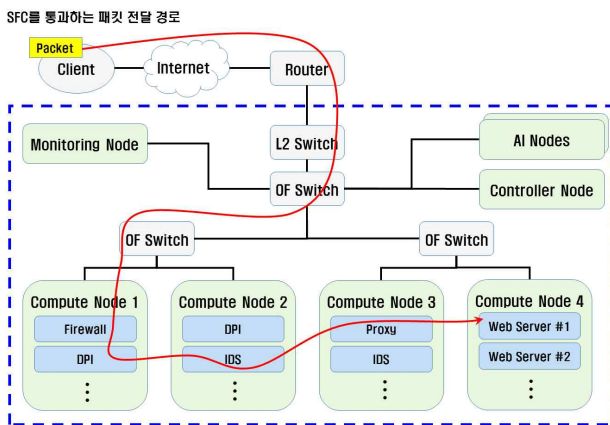


그림 2 강화학습 기반 SFC 성능 평가 테스트베드

성능 측정 시나리오는 [그림 2]의 클라이언트 (Client)에서 다수의 HTTP 요청 메시지 (HTTP Request Message)를 발생시켜 테스트베드 내 SFC 경로를 경유해 웹 서버로 전달 한 후, 요청 메시지에 대한 모든 응답 메시지 (Response Message)를 받는 데 소요되는 시간을 측정하였다. 이 때, HTTP 요청 메시지 생성은 Apache 웹 서버 상태를 측정하는 성능 측정 도구인 AB (Apache HTTP Server Benchmarking tool)을 사

용하였다. 웹 서버로 HTTP 요청 메시지가 전달되기 전에 경유하는 SFC는 4개의 VNF (Firewall, DPI, IDS, Proxy)로 구성된다. 방화벽 기능을 하는 Firewall VNF는 Iptables를 사용하였으며, DPI는 nDPI [8], IDS는 Suricata [9], 그리고 Proxy 기능을 제공하는 VNF는 HAProxy [10]를 설치하였다. 마지막으로, HTTP 요청 메시지에 대한 응답을 위해 두 개의 Apache 웹서버를 설치하고, SFC의 마지막 VNF인 HAProxy가 Round-robin으로 각 웹 서버로 패킷을 전달하도록 설정하였다. 또한, 선행연구 [4]와 동일하게 각 VNF 별로 3개의 VM 인스턴스를 생성하여 총 12개의 VM 인스턴스가 테스트베드 내에 존재한다. 이 때, VNF 자원 사용량에 의해 패킷 처리 성능이 쉽게 영향 받을 수 있도록 VNF가 동작하는 각 VM 인스턴스에 작은 양의 자원 (vCPU 1개, 메모리 512MB, 디스크 20GB)을 동일하게 할당하였다. 모든 VM 인스턴스 생성을 마친 후에 VNF들을 선택해 생성할 수 있는 SFC 경로 개수는 [그림 3]과 같이 81 (3^4)개이다.

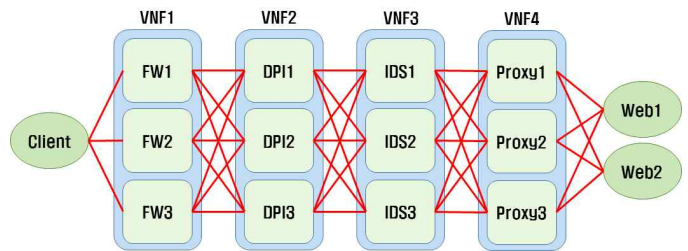


그림 3 테스트베드 내 선택 가능한 SFC 경로

VNF 자원 상태를 고려한 Q-Learning 기반의 SFC 경로 선택의 성능 평가를 위해, 동일한 성능 측정 시나리오에서 임의로 SFC를 구성한 경우의 성능과 비교하였다. 이 때, 성능 측정 과정에서 Stress-ng [11] 도구를 사용하여 임의의 VNF에서 CPU 및 메모리를 사용하도록 설정하였다. 임의로 SFC 경로를 선택하는 경우에는 각 VNF에서 소모하고 있는 자원을 고려하지 않고, 단순히 SFC를 구성하는 VNF들의 순서만을 고려하여 경로를 선택한 후 SFC를 생성한다. 반면, Q-Learning 기반의 SFC 생성은 현재 측정된 VNF의 자원 사용량 (CPU, 메모리), 그리고 다음 순서의 VNF와 동일한 물리 노드에 배치되어 있는 지 여부를 보상으로 고려하여 학습을 수행한다. 학습은 테스트베드 내 설치된 AI 노드에서 수행되며, 보상을 책정하기 위해 필요한 VNF 자원 사용량 및 물리 노드 위치 정보는 모니터링 노드로부터 받아온다. 마지막으로, Q-Learning 학습에 사용한 초기 변수 값은 [표 1]과 같다.

표 1. Q-Learning 초기 변수 정보

변수 종류	값
η (학습률)	0.1
γ (할인율)	0.6 ~ 0.9
ϵ (탐험 확률)	0.99
iteration_num (반복 횟수)	1000 ~ 3000

학습률 η 의 초기 값은 0.1로 고정하여 사용하였으며, 미래의 보상을 얼마나 가치를 두고 환산할지 결정하는 시간할인율 γ 값은 0.6에서 0.9 사이의 값으로 변경하면서 성능 평가를 진행하였다. 또한, ϵ -greedy 알고리즘의 초기 ϵ 값은 0.99로 설정하여, 초반 학습 과정에서는 대부분의 경우 Q값에 따른 행동 선택보다는 임의의 행동을 선택하도록 유도하였다. ϵ 값은 학습을 반복할수록 일정 비율로 줄어들게 된다. 마지막으로, 학습 반복

횟수는 1000회부터 3000회까지 임의로 수행하였다. 초기 ϵ 값을 높게 설정하여, 학습 반복 횟수가 작을 경우에는 탐험만으로 Q값을 갱신하게 되기 때문에 학습 횟수는 적절히 탐험과 탐사를 수행하며 Q값을 학습할 수 있도록 정하는 것이 필요하다. 본 성능 평가에서 1000회 이상으로 학습을 진행해본 결과, 4개의 VNF로 구성된 SFC 경로를 찾는 데에는 1000회 이상 학습으로 최적의 SFC 경로를 선택할 수 있음을 확인하였다.

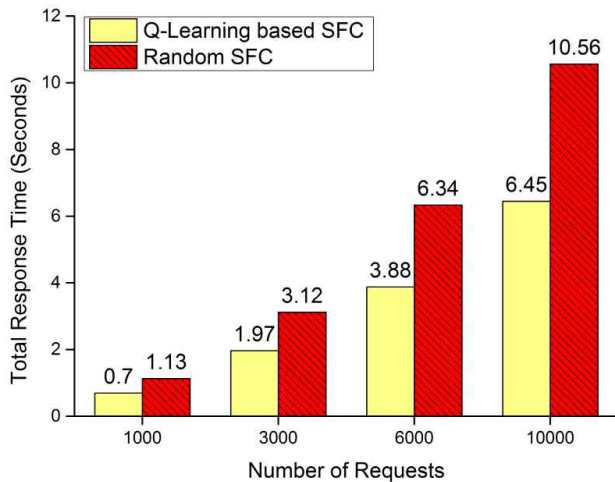


그림 4 SFC를 통한 HTTP 요청 메시지의 응답 시간

모든 설정을 마치고 나서, 서로 다른 방법으로 생성된 두 개의 SFC로 HTTP 요청 메시지를 1,000개, 3,000개, 6,000개, 그리고 10,000개를 발생시킨 후, 모든 응답 메시지가 클라이언트로 도착하는 시간 (Total Response Time)을 측정된 결과는 [그림 4]와 같다. 전반적으로, 임의로 각 VNF를 선택해 SFC를 구성하는 경우에는 Q-Learning 기반으로 생성한 SFC에 비해 응답 메시지를 모두 받을 때까지 더 긴 시간이 필요하였다. 또한, HTTP 요청 메시지 개수가 증가할수록 두 SFC에서 모든 응답 메시지를 받는 데 필요한 시간도 증가하지만, 임의로 생성한 SFC의 경우에 증가 폭이 더 큰 것을 확인할 수 있었다. 결과적으로, Q-Learning으로 생성한 SFC는 임의로 생성한 SFC에 비해 같은 HTTP 요청 메시지 개수 일 때, 응답을 받는 데 소요되는 시간 대비 61% 정도의 시간이 소요된다. 이를 통해, 제안하는 Q-Learning 기반 SFC 경로 선택 방법은 각 VNF들의 자원 사용량과 설치된 물리 노드 위치를 보상으로 활용하여 효과적으로 SFC 경로를 선택할 수 있음을 확인하였다.

V. 결론 및 향후 연구

본 논문에서는 강화학습 알고리즘 중 하나인 Q-Learning을 활용해 SFC 경로를 찾는 방법을 제안하였다. 제안하는 방법은 SFC를 구성하는 VNF의 현재 CPU 사용량과 메모리 사용량, 배치된 물리 노드 위치를 고려하여 보상을 책정하고, 이를 기반으로 짧은 HTTP 응답 시간을 제공하는 SFC 경로를 선택한다. 제안하는 Q-Learning 기반 SFC 경로 선택 방법은 임의로 SFC 경로를 선택하는 것보다는 응답 시간 측면에서 좋은 경로를 선택할 수 있도록 돕지만, 현재 자원 상태와 VNF 설치 위치 같은 제한된 정보만을 보상으로 고려한 경로 선택이기 때문에 향후 네트워크 혼잡 등 동적으로 네트워크 상태가 변할 때 대응하기 어렵다는 문제가 있다. 또한, 미로찾기와 유사하게 단순한 경로 선택 문제로 모델링하였기 때문에 SFC 성능에 영향을 미치는 다양한 경우를 상태로 정의하지 못했다는 한계가 존재한다. 따라서 향후 연구로는 단순한 Q-Learning이 아닌, 신경망을 도

입한 DQN (Deep Q Network) 등의 기법을 활용하여 NFV 환경에서 고려할 수 있는 다양한 경우들을 상태로 정의한 후, SFC 생성 연구에 활용할 예정이다. 또한, 최적의 SFC 경로를 찾기 위해 요구되는 학습 시간, 다른 최적화 기법과의 비교 등 다양한 시나리오에서의 성능 검증을 진행할 계획이다.

ACKNOWLEDGMENT

이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (2018-0-00749, 인공지능 기반 가상 네트워크 관리기술 개발)

참고 문헌

- [1] Kim, Seonhyeok, Jaehyeok Son, Ashis Talukder, and Choong Seon Hong. "Congestion prevention mechanism based on Q-learning for efficient routing in SDN." In 2016 International Conference on Information Networking (ICOIN), pp. 124-128. IEEE, 2016.
- [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] Kim, Sang Il, and Hwa Sung Kim. "A research on dynamic service function chaining based on reinforcement learning using resource usage." In 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), pp. 582-586. IEEE, 2017.
- [4] 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.
- [5] Mechtri, Marouen, Chaima Ghribi, Oussama Soualah, and Djamel Zeglache. "NFV orchestration framework addressing SFC challenges." IEEE Communications Magazine 55, no. 6 (2017): 16-23.
- [6] Gallenmüller, S., Emmerich, P., Wohlfart, F., Raumer, D., & Carle, G. (2015, May). Comparison of frameworks for high-performance packet IO. In 2015 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS) (pp. 29-38). IEEE.
- [7] Dobrescu, Mihai, Norbert Egi, Katerina Argyraki, Byung-Gon Chun, Kevin Fall, Gianluca Iannaccone, Allan Knies, Maziar Manesh, and Sylvia Ratnasamy. "RouteBricks: exploiting parallelism to scale software routers." In Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles, pp. 15-28. 2009.
- [8] Deri, Luca, Maurizio Martinelli, Tomasz Bujlow, and Alfredo Cardigliano. "ndpi: Open-source high-speed deep packet inspection." In 2014 International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 617-622. IEEE, 2014.
- [9] Open Information Security Foundation, "Suricata: Open Source IDS/IPS/NSM engine", [Online]. Available: <https://suricata-ids.org/>
- [10] HAProxy, "HAProxy: The Reliable, High Performance TCP/HTTP Load Balancer", [Online]. Available: <http://www.haproxy.org/>
- [11] King, Colin Ian. "Stress-ng." URL: <http://kernel.ubuntu.com/git/cking/stressng.git/> (visited on 28/03/2018) (2017)