

SDN/NFV 환경에서 효율적인 서비스 체이닝을 위한

Q-learning기반 동적 서비스 경로 선택 기법

김선혁⁰, 홍충선*

경희대학교 컴퓨터공학과

{kshyuk0605, cshong}@khu.ac.kr

Dynamic Service Function Path Selection Based on Q-learning
for Efficient Service Function Chaining in SDN/NFV EnvironmentSeonhyeok Kim⁰, ChoongSeon Hong*

Department of Computer Science and Engineering, Kyung Hee University

요 약

서비스 체이닝(Service Function Chaining)은 Network Function Virtualization(NFV)에서 가상화된 서비스 기능들을 순차적으로 처리 및 전달하는 기술로 Software Defined Networking(SDN)과 결합하여 다양한 연구가 진행되고 있다. 본 논문에서는 SDN/NFV 환경에서 효율적인 서비스 체이닝을 위해 수명이 긴 트래픽 플로우를 식별하고 링크의 대역폭, 서비스 기능 수행 시간, 서비스 체이닝 지연 시간, 서비스 노드 사용량 등을 고려한다. 이러한 정보를 바탕으로 Q-learning과 결합하여 새로운 서비스 경로를 선택하는 동적 서비스 체이닝 기법에 대해 제안한다. 제안하는 방법을 통해 대역폭 오버헤드, 서비스 노드의 부하, 서비스 체이닝 수행시간을 감소시켰고 이에 따라 효율적인 서비스 체이닝 환경을 보장하였다.

1. 서 론

Network Function Virtualization(NFV)은 가상화 기술의 발전에 따라 네트워크상에 존재하는 다양한 네트워크 기능들을 소프트웨어 형식으로 제어, 관리가 가능하도록 가상화하여 제공하는 기술이다. 이를 통해 네트워크 서비스 기능이 데이터 전달 경로에 존재하는 것이 아니라, 데이터 트래픽이 자신이 필요로 하는 네트워크 서비스 기능이 존재하는 곳을 경유하는 형태로 제공된다. NFV 발전에 따라 특정 네트워크 서비스를 위한 서비스 체이닝(Service Function Chaining)이 등장하였고, 이는 서비스 기능들을 순차적으로 추상화하는 기술을 의미한다[1].

서비스 체이닝은 서비스 경로(Service Function Path)라고 하는 서비스 그래프를 만들기 위해 특정 네트워크 노드의 서비스 기능 인스턴스를 선택함으로써 형성된다. 즉, 데이터 패킷에 적용될 네트워크 기능으로 데이터 패킷을 라우팅 시켜주는 기술이다. 서비스 체이닝의 주요 기능으로는 트래픽 발생에 따른 트래픽의 서비스 체인을 분석하고 판별하는 서비스 분류자(Service Classifier)와 서비스 체인을 기반으로 유입되는 트래픽을 정해진 서비스 기능 순서대로 전달하는 서비스 기능 전달자(Service Function Forwarder)가 있다. 이러한 서비스 체이닝 기술

은 Software Defined Networking(SDN)과 결합하여 제어 및 관리를 유연하게 처리할 수 있다[2].

서비스 체이닝 기술은 Internet Engineering Task Force(IETF) Service Function Chaining(SFC) Working Group(WG)에서 주로 논의되고 있으며, 서비스 체이닝 기술에 대한 문제 정의, 유즈 케이스, 기능 및 구조 등에 대한 표준을 정의하고 있다. 그중 파일 전송, 비디오 스트리밍과 같은 수명이 긴 플로우에 대해 draft-ietf-sfc-long-lived-flow use-cases-03 문서[3]에서 정의하며, 수명이 긴 트래픽 선형 플로우의 오버헤드 문제를 해결하고자 한다.

본 논문에서는 파일 전송, 비디오 스트리밍과 같은 수명이 긴 트래픽 플로우에 대해 다루며 서비스 기능 수행 시간, 서비스 체이닝 지연 시간, 각 링크의 대역폭, 서비스 노드 사용량 등을 고려한다. 이러한 정보를 바탕으로 Q-learning 기반의 서비스 경로를 재설정하고 오버헤드 문제 및 서비스 노드의 부하를 감소시켜, 서비스 체이닝 시간을 줄이는 동적 서비스 체이닝 기법을 제안한다.

2. 관련연구

2.1. Q-learning

환경(environment)을 탐색하는 에이전트(agent)가 현재의 상태(state)를 인식하여 특정 액션(action)을 취하고 에이전트(agent)는 환경(environment)으로부터 보상(reward)을 얻게 된다. 이를 통해 에이전트(agent)의 보상(reward)

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (B0190-16-2013, 유무선 통합 네트워크에서 접속 방식에 독립적인 차세대 네트워킹 기술 개발).

*Dr. CS Hong is the corresponding author

을 최대화하는 방법을 찾는 과정을 강화 학습(Reinforcement Learning)이라 한다.

Q-learning은 대표적인 강화 학습 알고리즘 중 하나로 Markov decision process에 기반을 두며, 특정 상태(state) S 에서 최적의 액션(action) A 를 찾기 위해 사용된다[4]. 결국 Q-learning 알고리즘의 학습을 통해 보상(reward) R 을 기반으로 출발지와 도착지 사이에서 최적의 경로를 구할 수 있다. 식 (1)은 Q-learning 알고리즘의 수학적 모델이다.

$$Q(state, action) = R(state, action) + \gamma \cdot \text{Max}[Q(next\ state, all\ actions)] \quad (1)$$

3. 제안사항

3.1 문제정의

본 장에서는 서비스 체이닝 환경에서 파일 전송, 비디오 스트리밍 등 수명이 긴 트래픽 플로우 식별 및 링크의 대역폭, 서비스 체인 지연 시간, 서비스 노드 부하 등을 고려한 Q-learning기반 동적 서비스 체이닝 알고리즘을 제안한다. 제안하는 방법은 트래픽 발생 시 Layer2/3/4 패킷 헤더 필드와 전송시간을 고려하여 수명이 긴 트래픽 플로우를 식별한다(e.g. FTP, TFTP => 파일전송, RTP, HTTP => 비디오 스트리밍, etc.)[3]. 그리고 전체 토폴로지를 바탕으로 서비스 기능 수행 시간, 서비스 노드 사용량, 서비스 체이닝 지연 시간, 각 링크의 대역폭을 측정하여 네트워크의 오버헤드를 최소화하고 QoS를 보장한다. 표 1은 제안하는 방법에서 사용하는 변수와 의미를 정의한 것이다[5].

표 1 변수 및 의미 정의

변수	의미
$G=(V_i, E_j)$	토폴로지
V_i	서비스 노드 i
E_j	서비스 노드 및 서비스 기능 인스턴스들을 연결하는 j개의 링크들의 집합
SP_x	서비스 x가 수행되어질 때 거쳐야 하는 서비스 경로 $SP_x = SF_0 \rightarrow SF_1 \rightarrow \dots \rightarrow SF_n$
SF_n	서비스 기능 인스턴스 n
PT_{SF_n}	서비스 기능 인스턴스 i가 처리되는 시간
LT_x	서비스 x가 수행될 때 서비스 체이닝 지연 시간 $LT_x = \frac{1}{tBW_j - PAR_{SF_n}}$
PAR_{SF_n}	서비스 기능 인스턴스 n의 패킷 도착 비율
tBW_j	링크 j의 전체 대역폭
cBW_j	링크 j의 현재 대역폭
$cCPU_k$	서비스 노드 k의 현재 CPU 사용량
$Last\ cCPU_k$	서비스 노드 k의 이전 CPU 사용량

3.2 Q-learning기반 동적 서비스 체이닝

알고리즘 1은 서비스 가능한 모든 서비스 체이닝 경로를

탐색하는 알고리즘의 수도코드이다. 네트워크 경로 내 모든 서비스 기능 인스턴스를 탐색하여 저장하고, 서비스 기능 인스턴스의 출발지와 목적지의 경로를 탐색한다. 목적지가 다음 서비스 기능 인스턴스와 일치한다면 새로운 서비스 체이닝 경로를 추가한다. 이러한 과정을 반복하여 서비스 가능한 모든 서비스 체이닝 경로를 탐색한다.

알고리즘 1. 서비스 체이닝 경로 탐색 알고리즘

```

1: newSP[src][dst] ← G=(V_i, E_j)
2: s ← source, d ← destination
3: while ( V_i ≠ ∅ )
4:   Find possible all SF_i ⇒ allSF_i
5:   for all s ∈ allSF_i
6:     for all d ∈ allSF_i
7:       if ( d == Next SF_i )
8:         s⇒src, d⇒dst
9:         update newSP
    
```

알고리즘 2는 모든 서비스 체이닝 경로를 바탕으로 서비스 기능 수행 시간, 서비스 노드 사용량, 서비스 체이닝 지연 시간, 각 링크의 대역폭을 고려하여 Q-learning과 결합한 동적 서비스 체이닝 알고리즘이다. 다음 도착지가 서비스 기능이 수행될 수 있는 서비스 노드이거나 도착지인 경우 서비스 기능 수행 시간, 서비스 체이닝 지연 시간, 링크의 대역폭, 서비스 노드 사용량을 측정한다. 해당 조건에 따라 보상(R) 값을 감소시키거나 증가시킨다[6]. 이에 따라 변경된 R-Matrix를 기반으로 Q-learning 알고리즘에 적용하고 Q-Matrix 값을 구함으로써, 링크의 대역폭과 서비스 노드의 오버헤드 및 QoS를 고려한 서비스 경로를 구할 수 있다.

알고리즘 2. Q-learning기반 동적 서비스 체이닝 알고리즘

```

1: Q-Matrix ← 0
2: R-Matrix ← Reward(newSP)
3: while
4:   while ( Iterated n times )
5:     Select a random state S
6:     Find possible actions and Select a random action A
7:     S ⇒ Vsrc, A ⇒ Vdst
8:     if ( Vdst == Next SN || Goal state )
9:       if ( PT_{SF_n} > Last PT_{SF_n} && LT_x > Last LT_x )
10:        if ( cBW_j > tBW_j × 0.8 && cCPU_k > Last cCPU_k )
11:          R[Vsrc][Vdst] value is R[Vsrc][Vdst] * 0.5
12:        else
13:          R[Vsrc][Vdst] value is R[Vsrc][Vdst] * 0.9
14:        else
15:          R[Vsrc][Vdst] value is R[Vsrc][Vdst] * 2
16:      else
    
```

- 17: $R[Vsrc][Vdst]$ value is $R[Vsrc][Vdst]$
- 18: Update R-Matrix
- 19: Compute Q-learning Algorithm
- 20: Update Q-Matrix
- 21: Update SP

3.3 SDN기반 서비스 체이닝 구조

그림 2는 본 논문에서 제안하는 방법을 테스트하기 위해 설계한 SDN기반 서비스 체이닝 구조이다. ONOS SDN 컨트롤러와 OpenStack을 networking-onos를 활용하여 연동하였으며, networking-sfc를 이용하여 서비스 체이닝 환경을 구축하였다. 그리고 3가지 모듈을 적용하였는데, Monitoring 모듈은 전체 네트워크 토폴로지 상태를 모니터링하며, 서비스 노드의 사용량 및 링크의 대역폭 등을 측정한다. Classifier 모듈을 통해서는 인입되는 트래픽을 분석하여 수명이 긴 트래픽 플로우를 식별한다. 마지막으로 Flowrule 모듈에서는 수집된 정보와 앞서 정의한 알고리즘을 바탕으로 새로운 서비스 체이닝 경로를 설정하여 링크의 오버헤드 및 서비스 노드 부하 문제를 해결한다.



그림 1. SDN기반 서비스 체이닝 구조

4. 성능평가

본 장에서는 제안하는 Q-learning 기반 동적 서비스 체이닝 기법의 효율성을 검증한다. 이를 위해 5개의 서비스 노드와 서비스 노드 당 3개의 서비스 기능 인스턴스를 생성하고 10개의 서비스 체인을 구성하여 테스트를 진행하였다. 그리고 호스트와 서버를 두어 파일을 전송 받을 수 있도록 하였고, 파일 전송 시 플로우의 수를 증가시키며 링크의 대역폭, 서비스 노드의 사용량, 서비스 체이닝 시간을 측정하였다. 측정결과 플로우 증가에 따라 대역폭 및 서비스 노드의 사용량 증가하였지만, 제안하는 기법을 적용하며 대역폭 및 서비스 노드의 사용량이 감소하였고, 전체적인 서비스 체이닝 시간이 감소하였다. 그림 2는 플로우 수 증가에 따른 기존 방법과 제안하는 방법의 100MB당 서비스 체이닝 시간을 비교한 것으로 제안하는 방법이 기존 방법보다 효과적임을 확인할 수 있다.



그림 2. 플로우 수 증가에 따른 서비스 체이닝 시간

5. 결론 및 향후 연구

본 논문에서는 SDN/NFV 기반 서비스 체이닝 환경에서 수명이 긴 트래픽 플로우로 인한 대역폭 오버헤드 문제 및 서비스 노드 부하에 따른 문제점 등에 대해 살펴보았다. 이를 해결하기 위해 서비스 기능 수행 시간, 서비스 체이닝 지연 시간, 링크의 대역폭 및 서비스 노드 사용량 등을 고려하여 Q-learning 기반 동적 서비스 체이닝 기법을 제안하였다. 성능평가를 통해 제안하는 방법이 기존 방법보다 대역폭 오버헤드 및 서비스 노드의 부하를 감소시킬 수 있음을 확인할 수 있었고, 결과적으로 서비스 체이닝 수행 시간이 감소하는 결과를 얻을 수 있었다.

향후 연구에서는 더욱 효율적인 서비스 체이닝을 위해 서비스 기능 인스턴스를 특정 기능만을 수행하는 간소화된 서비스 기능 인스턴스로 구성하여 서비스 기능 인스턴스의 처리 속도를 증대할 수 있는 연구를 진행할 것이다.

참고 문헌

- [1] S. Sahaaf, W. Tavernier, M. Rost, S. Schmid, D. Colle, M. Pickavet, P. Demeester, "Network service chaining with optimized network function embedding supporting service decompositions," *Journal of Computer Networks*, Vol. 93, pp. 492-505, January 2016.
- [2] Yong Li, Feng Zheng, Min Chen and Depeng Jin, "A unified control and optimization framework for dynamical service chaining in software-defined NFV system," *IEEE Wireless Communications*, Vol. 22, pp. 15-23, December 2015.
- [3] R.Krishnan et al., "SFC Long-lived Flow Use Cases," draft-ietf-sfc-long-lived-flow-use-cases-03, February 2015.
- [4] F. Farahnakian, M. Ebrahimi, M. Daneshalab, P. Liljeberg and J. Plosila, "Q-learning based Congestion-aware Routing Algorithm for On-Chip Network," *Networked Embedded Systems for Enterprise Applications(NESEA)*, pp. 1-7, December 2011.
- [5] Minh-Tuan Thai, Ying-Dar Lin, Yuan-Cheng Lai, "A Joint Network and Server Load Balancing Algorithm for Chaining Virtualized Network Functions," *IEEE International Conference on Communications*, pp. 1-6, May 2016.
- [6] Seonhyeok Kim, Jaehyeok Son, Ashis Talukder and Choong Seon Hong, "Congestion Prevention Mechanism Based on Q-learning for Efficient Routing," *IEEE International Conference on Information Networking (ICOIN)*, pp. 124-128, January 2016.