

SDN 기반 네트워크에서의 Q-Learning 을 이용한 플로우 관리

문승일⁰, Rossi Kamal, 홍충선*, 이성원

경희대학교 컴퓨터공학과

{moons85, rossi, cshong and drsungwon}@khu.ac.kr

요 약

오늘날의 인터넷은 장비와 인프라의 폐쇄적인 구조로 인하여 사용자의 다양한 요구사항을 반영하기에 한계가 있다. 이러한 한계를 넘어 관리자로 하여금 보다 유연하게 네트워크를 제어할 수 있게 하기 위한 방안으로 OpenFlow 가 제안되었다. 이러한 OpenFlow 기반의 네트워크에서 보다 효율적인 플로우 관리를 위해 본 논문에서는 Q-Learning 을 활용한 opportunistic 한 플로우 관리 알고리즘인 OpenFlow-QL 를 제안한다. 또한 제안하는 OpenFlow-QL 를 OpenFlow 기반 네트워크에 적용하기 위해 Virtual Network Manager 를 제안한다.

1. 서론

인터넷의 발달로 인하여 인터넷을 통한 다양한 서비스의 지원이 가능하게 되면서 보다 다양한 형태의 서비스와 요구되고 있다. 하지만 다양한 요구사항을 반영하기에는 기존의 인터넷은 장비와 인프라의 구조가 폐쇄적이라는 태생적인 문제를 갖고 있기에 자유롭게 네트워크를 제어하기가 어렵다. 이러한 문제를 해결하기 위한 방안 중의 하나로 OpenFlow[1]가 제안 되었다. OpenFlow 는 SDN (Software Defined Network)[2]의 기술 중 하나로 Control Plane 과 Data Plane 이 분리되어 있고, OpenFlow 프로토콜[3]이라는 개방형 프로토콜을 제공함으로써 관리자로 하여금 프로그램 가능한 네트워크 환경을 제공한다.

본 논문에서는 OpenFlow 기반의 네트워크에서 효율적인 경로 플로우 관리를 위하여 Q-Learning[4] 개념을 적용한 opportunistic 경로 플로우 관리 알고리즘인 OpenFlow-QL(Q-Learning)을 제안하였다.

OpenFlow 기반 네트워크에서 OpenFlow-QL 를 사용하여 경로 플로우를 생성하는 과정은 다음과 같다. 먼저 관리자가 Path Manager 를 통해 새로운 경로 생성을 요청한다. 그 다음, Path Manager 는 VN(Virtual Network) Manager 에게 최적의 경로를 요청한다. 경로 생성을 요청 받은 VN Manager 는 경로 생성을 위해 NOX Controller 에게 Q-table 을 요청하게 되고, NOX Controller[5]는 각각의 OpenFlow 스위치에게서 업데이트 된 Q-table 을 전달받아 VN Manager 에게 전달한다. Q-table 을 전달받은 VN Manager 는 Path Flow Optimizer 를 사용하여 최적의

플로우를 생성하고 이를 Path Manager 에게 전달한다.

2. OpenFlow

OpenFlow 는 서로 다른 스위치와 라우터의 플로우 테이블을 프로그래밍 할 수 있는 오픈 프로토콜을 제공한다. 아래의 그림 1 은 OpenFlow 스위치의 개념도를 보여주고 있다. 그림에서 알 수 있듯이 OpenFlow 는 FlowTable, Secure Channel, OpenFlow Protocol 의 세 개의 파트로 나눌 수 있다. FlowTable 은 각 플로우 엔트리와 연관된 동작이 정의된 부분으로 이를 통해 스위치는 플로우에 대한 처리를 할 수 있으며, Secure Channel 은 컨트롤러와 스위치 사이에서 명령어나 패킷이 전달 될 수 있도록 도와주는 역할을 한다. 마지막으로 OpenFlow Protocol 은 공개되어있는 표준적인 방식을 제공 함으로써 스위치와 Allocator 가 서로 통신할 수 있도록 한다.

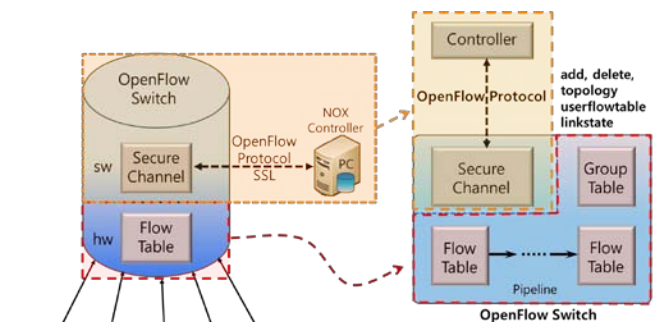


그림 1. OpwnFlow 스위치

3. 시스템 구조

본 논문에서 제안하는 OpenFlow 기반의 네트워크 관리를 위한 시스템의 구조는 그림 2 에서 알 수

이 논문은 2012 년도 정부(교육과학기술부)의 재원으로 한국연구재단-차세대정보컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(2012-0006421).

*Dr. CS Hong is corresponding Author

있듯이 Path Manager, VN Manager, NOX Controller 로 구성되어 있다.

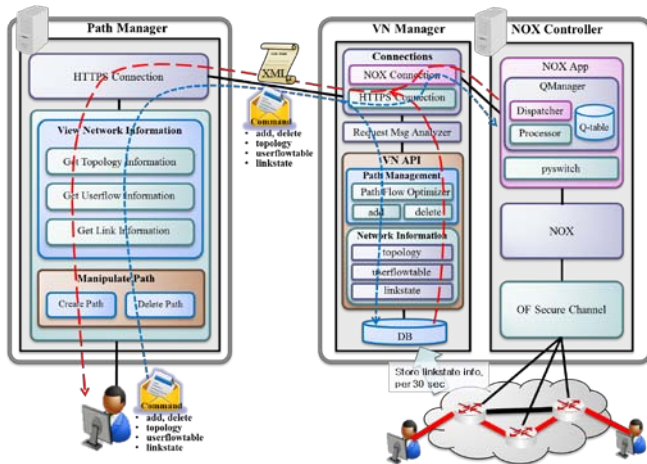


그림 2. 전체 시스템 구조

Path Manager 의 하위 모듈로는 View Network Information, Manipulate Path 가 있고 HTTPS Connection 을 통해 VN Manager 와 통신을 한다.

VN Manager 는 크게 Connections, Request Msg Analyzer, VN API 로 나눌 수 있다. Connections 는 Path Manager 및 NOX Controller 와의 통신을 담당하고, Request Msg Analyzer 는 Path Manager 의 Request Message 를 분석하여 그에 해당하는 작업을 호출한다. VN API 는 OpenFlow 기반의 네트워크를 관리하기 위한 기능들을 제공한다.

NOX Controller 는 네트워크 제어를 위한 Application 이 동작하는 NOX App, 네트워크 운영체제인 NOX, NOX 와 OpenFlow 스위치간의 통신을 담당하는 OF Secure Channel 로 구성되어 있다.

관리자가 Path Manager 를 통해 서비스를 요청한 경우, Path Manager 는 Request Message 를 생성하여 VN Manager 에게 전달한다. Request Message 를 전달 받은 VN Manager 는 Request Msg Analyzer 를 통해 Request Message 를 분석하고 VN API 를 통해 그에 맞는 API 를 호출한다. 관리자의 요청이 수행이 되고 해당 결과값을 XML 로 변환한 VN Manager 는 이를 다시 Path Manager 에게 전달하여 관리자에게 서비스 요청에 대한 결과를 알려준다.

4. Q-Learning 기반 Opportunistic 플로우 관리 알고리즘

OpenFlow-QL 은 알고리즘 1 에서와 같이 NOX Controller 에 의해 동작하는 중앙집중 방식의 알고리즘으로 다음과 같은 단계를 거친다.

첫 번째 단계는 Initialization 단계로 상태 천이 확률과 보상을 초기화 한다. 상태 천이 확률은 QoS-Status 의 변화를 제어한다. 보상은 QoS 를 고려하는 플로우를 얻기 위해 지불하는 비용과 관련이 있다.

두 번째 단계는 Processing 이다. NOX Controller 는 이웃한 노드들에게 향하는 플로우를 고려한 OpenFlow 스위치의 QoS 에 기반하는 보상을 계산한다. 각 OpenFlow 스위치 i 의 보상은 j QoS Parameter 의 가중치 계수 합에 의한 값이다.

세 번째 단계는 Activation 이다. NOX Controller 는 각 노드 사이의 최적의 플로우를 선택하게 된다. 이것은 Path Manger 가 출발지와 목적지 사이의 플로우를 생성을 요청 했을 때 NOX Controller 는 즉각 Q-table 를 통하여 최적의 플로우를 선택할 수 있는 것을 의미한다.

마지막 단계는 Balancing 이다. NOX Controller 는 현재 보상과 앞으로 예상되는 보상을 모니터링 하여 Q-table 을 업데이트 한다.

알고리즘 1 OpenFlow-QL

1. **Initialization** Initialize $p(s|s,a)$, $R_{QoS}(s,a)$
2. **while** NOX-Controller receives Optimal-Flow selection request from Path Manager **do**
3. **Processing** NOX-Controller calculates QoS-based reward value $(R_{QoS})_i$ for each node i as follows
 $(R_{QoS})_i = sE \sum_j A_{dr} R_{bw}$, where
 $A_{dr} = AVR$ data rate (QoS parameter),
 $R_{bw} = 1/\text{remaining bandwidth}$ (QoS parameter),
 $sE = \text{state(END)}$, $MAX_Value(\text{defiend})$
4. **Activation**
Choose path selection action a
[Q table maintains actions for $(R_{QoS})_i$ entry of each i]
5. **Balancing**
Upgrade $Q(s,a)$ for all nodes as follows,
 $Q(s,a) = r_{QoS}(s,a) + Q(s,a)$
 $s \leftarrow s'$
6. **end while**

5. 구현사항

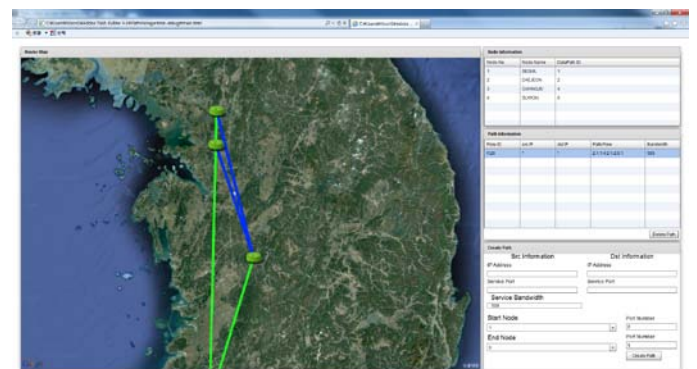


그림 3. Path Manager

본 논문에서는 OpenFlow-QL 를 적용하기 위하여 프레임워크를 구현하였다. 그림 3 은 앞서 설명한 시스템 구조로 OpenFlow 기반 네트워크에서 플로우를 관리하는 프로그램인 Path Manager 의 메인 화면이다. Flex 기반의 본 프로그램은 HTTPS 프로토콜을 통해 서비스를 VN Manager 에게 요청하고 그에 대한 결과로 전달 받은 XML 데이터를 이용하여 OpenFlow 기반의 네트워크를 관리하기 위한 기능을 시각적으로 제공할 수 있게 하였다.

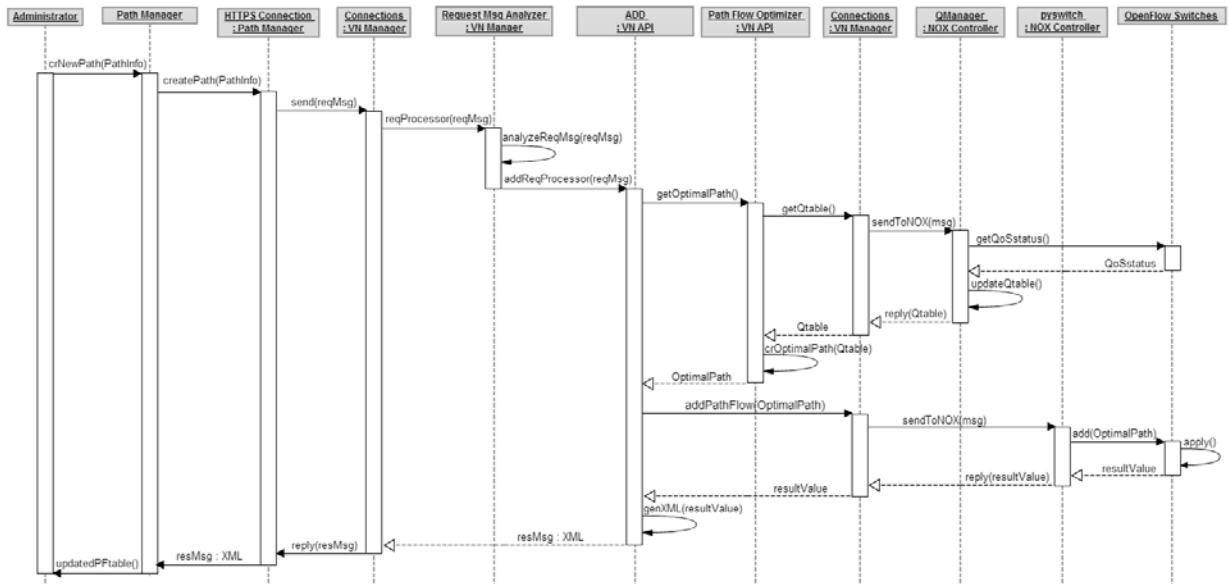


그림 4. OpenFlow-QL 를 이용한 플로우 생성과정

표 1. VN API 의 서비스 유형

API type	설명
topology	전체 OpenFlow 스위치의 id, 포트, 링크 정보 등을 제공한다.
add	가상의 플로우를 등록 한다.
delete	가상의 플로우를 제거 한다.
userflowtable	가상 망에 등록된 사용자 등록 플로우에 대한 정보를 제공한다.
linkstate	요구하는 링크에 대한 상태를 확인할 수 있다.

이러한 Path Manager 는 VN Manager 의 VN API 를 이용하여 구현되었으며, 표 1 에서와 같이 topology, add, delete, userflowtable, linkstate 등의 API type 을 제공한다.

예를 들어 그림 4 와 같이 플로우 생성 과정에 관한 동작을 설명하자면, 관리자가 경로 생성 프로그램을 통해 플로우 생성에 필요한 기본 정보를 입력하고 플로우 생성을 요청한다. 플로우 생성이 요청되면 Path Manager 는 입력 받은 데이터를 사용하여 Request Message 를 생성하고 이를 VN Manager 에게 전달한다. Request Message 를 전달 받은 VN Manager 는 Request Msg Analyzer 를 사용하여 Request Message 를 분석한 후 그에 맞는 API 를 호출한다. 플로우 생성 요청의 경우 Path Flow Optimizer 를 통해 최적의 플로우를 생성하게 된다. Path Flow Optimizer 는 플로우 생성을 위해 업데이트된 Q-table 값을 NOX Controller 에게 요청하고 전달 받은 Q-table 값을 사용하여 최적의 플로우를 생성한다. 이렇게 생성된 플로우는 다시 NOX Controller 에게 전달되고 NOX Controller 는 이를 OpenFlow 기반의 네트워크에 적용한다.

6. 결론 및 향후 계획

본 논문에서는 QoS 를 위한 정책을 기반으로 하는 Q-Learning 알고리즘과 이를 적용하기 위한 OpenFlow 기반의 네트워크를 관리할 수 있는 프레임워크인 VN Manager 와 Path Manager 를 제안 하였다. Q-Learning 의 학습과정을 통해 QoS 를 고려하는 최적의 플로우를 생성할 수 있게 되었고, 보다 네트워크 자원을 효율적으로 사용할 수 있게 되었다. 앞으로 보다 효율적인 네트워크 관리를 위해 QoS 뿐만 아니라, 보안, 안전성 등과 같은 다양한 변수들을 고려한 다양한 정책들을 구현하여 이를 OpenFlow-QL 에 적용시킬 계획이다. 또한 유선 환경의 네트워크뿐만 아니라 무선 환경을 고려하여 무선 자원 등을 효율적으로 할당하는 시스템에 대한 연구를 할 계획이다.

7. 참고 문헌

- [1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 69-74, Mar. 2008.
- [2] ONF, "Software-defined networking:the new norm for networks," ONF(Open Network Foundation) White Paper, 2012.
- [3] H. Jin, D. Pan, J. Liu, and N. Pissinou, "Openflow based flow level bandwidth provisioning for cicq switches," in INFOCOM, 2011 Proceedings IEEE, april 2011, pp. 476-480.
- [4] C. J. C. H. Watkins and P. Dayan, "Q-learning," Machine Learning, vol. 8, pp. 279-292, 1992.
- [5] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "Nox: towards an operating system for networks," SIGCOMM Comput. Commun. Rev., vol. 38, no. 3, pp. 105-110, Jul. 2008.