

SDN 환경에서 오픈플로우 스위치 애플리케이션을 활용한 전송 경로 설정 기법 연구

김도현^o, 홍충선^{*}
경희대학교 컴퓨터공학과
{ doma^o, cshong^{*} }@khu.ac.kr

Routing Method through OpenFlow Switch Application on Data Plane in Software-Defined Network Environment

Do Hyeon Kim^o, Choong Seon Hong^{*}
Department of Computer Science and Engineering, Kyung Hee University

요 약

Software-Defined Network에서 Controller는 Data Plane상의 OpenFlow Enabled Switch들을 관리하고 Host 간의 통신을 제어한다. 아울러 Controller 상에 다양한 관리정책을 뒷받침 할 수 있는 기능들을 소프트웨어 형태로 개발할 수 있기 때문에, SDN 환경에서 Controller의 역할은 그 비중이 상당히 크다고 할 수 있다. 하지만 반대로 많은 역할을 하는 Controller에 부하가 가중되고, 그로인해 맡은 역할을 다 할 수 없다면 Network 환경에서 치명적인 요소로 바뀔 가능성도 높다. 이에 본 논문에서는 Data Plane상의 OpenFlow 스위치들이 Application을 통하여 전송 경로를 설정하는 기법을 제안한다. 이는 OpenFlow Enabled Switch 상에서 Application이 각 Switch의 State를 파악하고 그에 대한 효율적인 Action을 수행함과 동시에, Network 관리 부분에서 많은 역할을 하는 Controller에서 전송 경로 설정 기능을 Data Plane에서 보조함으로써 Controller의 부하를 줄일 수 있다.

1. 서 론

Software Defined Network(SDN)는 다양한 Network 관리 정책을 소프트웨어의 형태로 적용시킬 수 있는 새로운 패러다임의 환경으로써 Internet of Things(IoT), Cloud, Data Center Network(DCN) 등 다양한 분야에서 활용되고 있다[1]. 기존 하드웨어 기반의 환경에서는 특정 기능, 예를 들어 방화벽이나 Deep Packet Inspection(DPI), Network Monitoring 기능 등을 네트워크에 적용시키려 할 때, 그 기능을 수행하는 하드웨어 장비를 구입하고 설치해야 하기 때문에, 비용적인 측면과 유지보수 측면에서 많은 부담이 된다. 하지만 SDN 환경에서는 적용되는 분야에서 목적에 맞는 기능들을 추가 장비가 아닌 소프트웨어로 개발하여 애플리케이션 형태로 적용할 수 있다. 이렇게 각 측면에서 유연성을 지닌 SDN을 통해 적용하고자 하는 환경에 대한 다양한 관리정책이 연구 개발되고 있다[2]. 하지만 반대로 소프트웨어로 구성된 모든 기능들이 Controller상에 적용되기 때문에 Large Scale Network에서 Controller의 부하 가중 문제는 반드시 고려되어야 할 사항이다. 이를 위해 분산 Controller 방식이 주로 제시 되어왔는데, [3]에서는 여러 개의 Sub Controller를 배치하여 Large Scale Network환경에서 Flow 요청에 대한 Main Controller의 부하를 분산시키는 기법을 제안하였다. 아울러 [4]에서는 유입되는 Packet에

대하여 Flow Table과 매칭하기 전에 Data Plane 상의 Bloom Filter와 Flow Cache 모듈을 기반으로 Packet을 분류하는 기법을 제안하였다. 이는 SDN의 기본 특성인 Packet_In 메시지 전송 단계에서 Unknown 트래픽과 악성 트래픽이 Controller로 전송되지 않도록 사전에 분류하여 차단하기 위한 방식으로 Data Plane을 적극 활용하였다. 본 논문에서는 Controller의 부하를 줄이고 원활한 Network환경을 제공하고자 Data Plane상의 Application을 통하여 전송경로를 설정하는 기법을 제안한다. 이는 Controller의 전송경로 설정 기능을 보조하는 역할을 하며, 강화학습을 기반으로 Switch의 상태에 따른 효율적인 Action을 선택하는 기능을 수행한다.

2. 관련 연구

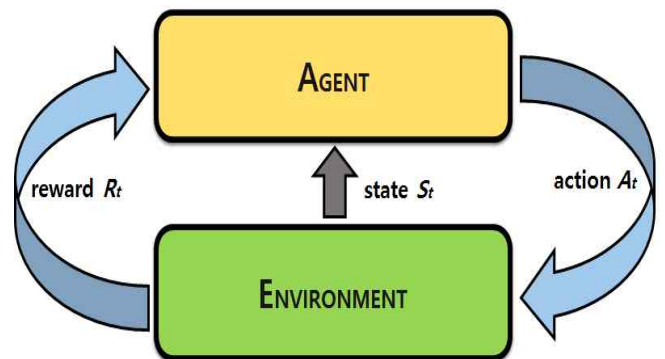


그림 1 Reinforcement Learning Process

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학 ICT연구센터육성 지원사업의 연구결과로 수행되었음 (IITP-2016-(H8501-16-1015)
*Dr. CS Hong is the corresponding author

강화학습은 기계학습의 범주에 속하는 방법 중 하나로, 어떠한 환경 안에서 Agent가 현재의 State를 관측하여 수행 가능한 행동들 중 Reward를 최대화하거나 Penalty를 최소화하는 행동 혹은 행동순서를 선택하여 학습해 나가는 방식이다. 즉, Markov Decision Process(MDP) 모델을 기반으로 Agent는 Environment와의 상호작용을 통해 점진적인 학습절차를 거치게 된다[5]. 경험을 통해 학습이 이루어지는 Reinforcement Learning은 특정 데이터 및 상황들을 예측하고 결정하기 힘든 환경에서 유용하게 쓰일 수 있다.

3. 제안 사항

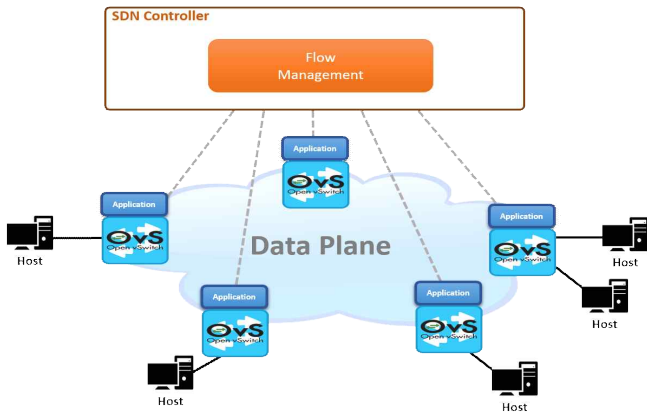


그림 2 Network Structure

그림 2는 제안하는 Network 환경의 전체적인 구조도이다. 기본적으로 SDN Controller는 유입되는 패킷을 전송하기 위해 정책에 부합하는 전송 경로상의 Switch에게 Flow Table을 배치한다. 하지만 제안하는 환경에서는 Src-Dst 간의 홑수 기반의 다중 경로에 해당하는 Flow Table을 배치하여 Data Plane상의 Application이 강화학습 과정을 통해 각 State에 따른 효율적인 전송포트를 찾게된다.

3.1 State & Reward Monitor

$$\text{each Port } S_{\text{Current}} = \begin{cases} 1 & \text{if } RX_{t+1} \neq RX_t \text{ or } TX_{t+1} \neq TX_t \\ 0 & \text{if } RX_{t+1} = RX_t \text{ and } TX_{t+1} = TX_t \end{cases} \quad (1)$$

$$\text{Definition of State} = \text{output port } S_{\text{Current}} \text{ Set on Switch} \quad (2)$$

$$\text{Number of State} = S_1, \dots, S_{2^n}, \text{ where } n \text{ is number of output port} \quad (3)$$

수식 (1), (2)와 (3)은 OpenVSwitch의 State를 정의한 것으로, 각 Port에서 전송 또는 수신되는 패킷의 유무를 확인하여 RL Application이 State를 요청할 때 정보를 전송한다. 또한 각 State에서 Action을 통한 패킷 전송 시, 그에 따른 해당 포트의 Transmission Delay 값을 계산함으로써 Reward 값을 측정하고 이를 RL Application으로 전송한다.

$$\text{Reward} = \frac{\text{Available Bandwidth}}{\text{Packet Length} * 8} = \frac{1}{\text{Transmission Delay}} \quad (4)$$

수식 (4)은 Reward 값을 정의한 수식으로써 State에 따른

Action을 수행했을 때 측정되는 Transmission Delay값을 기반으로 계산된다. Transmission Delay값이 클 경우, Reward 값은 줄어들기 때문에, State에서의 Action을 선택할 때, Transmission Delay값이 작은, 즉 Reward 값이 큰 Action을 선택하게 된다.

3.2 RL Application

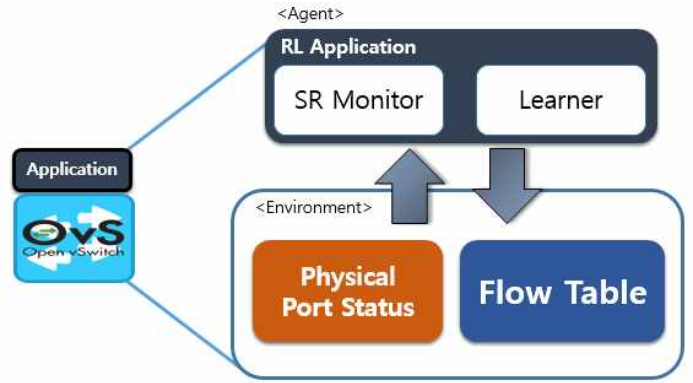


그림 3 Structure of Proposal

그림 3은 Data Plane상의 OpenVSwitch와 Application의 구조로써, 그림 1과 유사한 구조를 갖고 있다. Python으로 구현된 Reinforcement Learning Application은 Agent로써 OpenVSwitch의 Port를 모니터링하여 Environment의 State를 파악한다. 또한 Action의 경우 Flow Entry상의 정보를 토대로 각 Output Port의 Reward 값을 얻음으로써 전송포트를 결정하게 된다.

Algorithm Path Decision on OpenFlow Switch

- 1: Initialize $Q(s, a)$ arbitrarily
- 2: if Packet inflow is detected
- 3: Choose a from s Randomly
- 4: loop per Packet
- 5: Take action a
- 6: Observe reward r and Choose a' from s
- 7: Update $R(s, a)$ in Table
- 8: $a \leftarrow a'$
- 9: until Reward table complete
- 10: Choose a which has high Reward value

기존 강화학습 알고리즘의 경우, 에피소드가 끝나는 시점에서 Learning한 후 그때의 Q Table을 Policy로 사용할

다. 하지만 제안하는 방식에서는 포트의 사용유무를 확인함과 동시에 해당 포트에 대한 Reward 값을 계산함으로써 보다 효율적인 전송포트를 찾는 형태로 알고리즘을 적용하였다.

4. 성능 평가

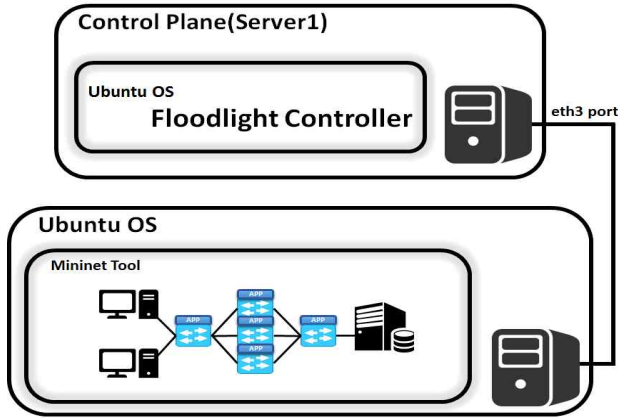


그림 4 Evaluation Environment

표 1 성능평가 시나리오

End to End Device	Test Tool	Time Duration
Host 1 - Server	Iperf	0 - 30s
Host 2 - Server	Iperf	15 - 40s

그림 4는 제안한 기법의 성능을 평가하기 위한 네트워크 구성도로, Floodlight Controller와 Mininet을 활용하였다. 아울러 표 1은 성능평가를 위한 시나리오로써, Floodlight Controller가 설치된 Server 1과 Data Plane이 구성된 Server 2 사이의 Flow Table 요청에 따른 Server 1 eth3번 포트의 Packet 송수신량을 Wireshark 툴을 통해 확인함으로써 결과를 도출하였다.

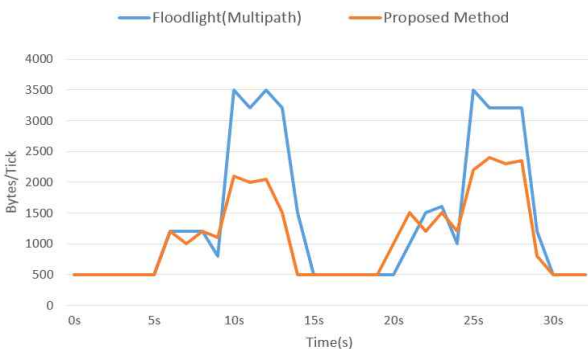


그림 5 Flow Table 업데이트에 따른 Packet 송수신량

그림 5는 표1의 시나리오를 기반으로 한 경로설정 과정에서 Server1의 eth3포트의 Packet Rate에 대하여 Multipath를 지원하는 Controller기반 설정방식과 Data Plane기반 설정방식을 비교한 그래프이다. Host1과 Host2가 Server로 패킷을 보낼 때, Multipath를 지원하는 Controller는 Packet_IN

메시지와 Packet_OUT 메시지를 주고받으며 각 링크의 상태를 고려한 경로를 설정하기 때문에 해당 구간에서 Packet Rate가 3500bytes 정도의 패킷이 송수신됨을 볼 수 있다. 하지만 제안하는 기법의 경우, Data Plane상의 Application을 통해 경로설정이 이루어지기 때문에 해당 구간에서 Packet Rate는 약 2500bytes 정도로 비교적 낮은 양이 송수신되었다.

5. 결론 및 향후 연구

본 논문에서는 SDN에서 Data Plane Application을 통한 경로설정 기법을 제안하였다. 이는 기존 Controller에 의존했던 경로설정을 Data Plane에서 보조함으로써 Flow Table 요청에 의해 Controller로 집중되는 트래픽과 부하를 일정 부분 감소시킬 수 있었다. 향후 노드의 개수에 따른 부하율을 토대로 추가적인 성능평가를 진행할 계획이며 Application을 경량화 하여 SDN 기반 무선 네트워크 환경 및 IoT환경에도 적용할 수 있도록 연구를 진행할 계획이다.

6. 참고 문헌

- [1] Nikos Bizanis, Fernando A. Kuipers., "SDN and Virtualization Solutions for the Internet of Things: A Survey," IEEE Access, Vol. 4 pp. 5591-5606, Sep 2016.
- [2] Jiaqiang Liu et al., "Software-defined internet of things for smart urban sensing," IEEE Communications Magazine, Vol. 53, No. 9, pp. 55-63, Sep 2015.
- [3] Yannan Hu et al., "BalanceFlow: Controller Load Balancing for OpenFlow Network," 2012 IEEE International Conference on Cloud Computing and Intelligent Systems(CCIS), Oct.30 - Nov.1, pp. 780-785, 2012
- [4] Luke McHale et al., "Stochastic Pre-Classification for SDN Data Plane Matching," 2014 IEEE International Conference on Network Protocols(ICNP), Oct.21-24, North Carolina, USA, 2014
- [5] Wikipedia, "Reinforcement Learning," Online Available: https://en.wikipedia.org/wiki/Reinforcement_Learning