SDN 환경에서 효율적인 패킷 전송을 위한 Multi-Armed Bandit 접근법 기반 분산 라우팅 기법 연구

김도현O, 홍충선*
경희대학교 컴퓨터공학과
{ domaO, cshong* }@khu.ac.kr

Multi-Armed Bandit Approach based Distributed Routing Method for Efficient Packet Forwarding in Software-Defined Network

Do Hyeon KimO, Choong Seon Hong*
Department of Computer Science and Engineering, Kyung Hee University

요 약

Software-Defined Network 기반의 네트워크 환경에서는 다양한 네트워크 관리 기능을 소프트웨어로 구현하여 SDN Controller상에 배치함으로써 Data Plane상의 모든 네트워크 노드들을 관리하고 Host간의 통신을 제어한다. 하지만 네트워크의 확장성을 고려하였을 때, 전송경로 설정에 대한 모든 기능들을 Controller상에서 관리하기에는 Data Plane 자원 활용의 비효율성 및 전체 네트워크 안정성 등의 문제가 발생할 수 있다. 이에 본 논문에서는 SDN 네트워크 환경에서 Data Plane의 자원을 적극 활용하여 효율적으로 패킷을 전송할 수 있는 기법을 제안한다. 이는 Controller와 각 OpenFlow 스위치상의 Sub Module 간 Interaction을 통해 전송경로를 설정하는 기법으로써 경로 설정에 대한 역할을 각 스위치로 분산하여 보다 효율적으로 전송 경로를 설정 할 수 있다.

1. 서 론

Software-Defined Network(SDN) 기술이 발전함에 따라 다양한 네트워크 관리 기능이 소프트웨어화 되어 전통적인 네트워크 환경에 비해 유연하고 비용 효율적인 환경으로 변화하였다. 전통적인 하드웨어 기반의 네트워크 환경에서는 네트워크 관리를 위한 특정 기능(예: Firewall, Deep Packet Inspection(DPI), Network Monitoring)을 적용시키려 할 때, 해당 기능을 위해 제작된 하드웨어 장비를 구입하고 설치해야하기 때문에, 비용적, 공간적인 측면과 설치 및 유지보수 측면에서 많은 부담이 된다. 하지만 SDN 기반의 네트워크 관리방식의 경우 Programmable Network 및 Network Function Virtualization(NFV) 등과 같은 특징을 통해 소프트웨어 형태의 기능들을 운용할 수 있으므로 Internet of Things (IoT)와 무선 네트워크 등 특정 네트워크 분야에서 SDN 기반 기술이 연구 개발 되고 있다[1][2]. 최근 SDN 관련 연구에서는 Controller의 역할을 Data Plane에서 수행하게 하는, 다시 말해 Data Plane의 자원을 활용하는 연구가 진행되고 있다. [3]에서는 OpenFlow 스위치 상에 App Table을 배치하여 유입되는 패킷이 App table상에 정해놓은 패킷과 부합할 경우, 해당 Application(예: Firewall, Load Balancer)이 동작하여 해당 패킷을 처리하는 기법을 제안하였다. 이는 네트워크 서비스를 활용함에 있어 Controller와 Switch간의 메시지 전달 과정을 줄이는 장점을 갖고 있다. 이처럼 SDN 환경에서 Data Plane을 활용하게 될 경우, 보다 유연한 서비스 제공이 가능하다. 이에 본 논문에서는 SDN 기반 네트워크 환경에서 Controller 뿐만 아니라 Data Plane 내 OpenFlow 스위치의 자원을 활용한 분산 방식의 라우팅 기법을 제안한다. Packet_IN 메시지를 통해 Controller에서 Source에서 Destination까지의 다중 경로에 대한 Flow Table을 각 스위치에게 전달하고, 각 OpenFlow 스위치 내에 위치한 Routing Sub Module이 Flow Entry들 중 가장 효율적인 Output 포트를 갖는 Entry를 선택하여 패킷을 전송하는 방식이다. 이는 패킷을 전송함에 있어 Controller와 스위치 간 메시지 전송을 일부 감소시킬 수 있으며 효율적인 경로로 패킷을 전송할 수 있다.

2. 관련 연구

◎ Multi Aramed Bandit Problem

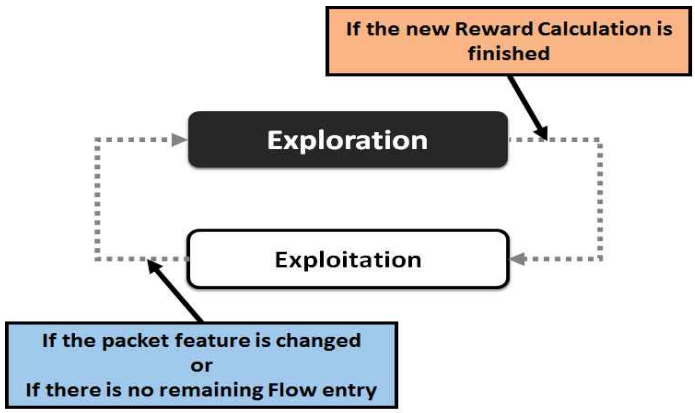

그림 1 Exploration와 Exploitation간의 Trade-Off

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음 (IITP-2017-2013-0-00717)
*Dr. CS Hong is the corresponding author

Multi Aramed Bandit Problem(MABP)은 단위 시간 동안 여러 개의 레버가 있는 슬롯머신에서 누적된 보상 값이 가장 큰 레버를 선택하는 의사결정 문제이다. 보상 값이 가장 큰 레버를 선택하기 위해서는 탐색(Exploration)과 이용(Exploitation) 간의 Trade-off를 효율적으로 조절해야한다. Exploration을 하지 않을 경우 부족한 정보를 통해 Exploitation을 하게 되며, 반대로 Exploration을 많이 하게 될 경우, 이미 의사 결정을 위한 정보를 가졌음에도 불구하고 더 정보를 얻기 위한 비효율적인 Exploration을 하게 된다. 따라서 둘 간의 Trade-off를 얼마나 효율적으로 조절하느냐가 MABP의 핵심이다[4].

본 논문에서는 각 스위치가 전송하려는 패킷에 대하여 그림 1과 같이 Exploration과 Exploitation 간의 Trade-Off를 각각의 조건을 두어 수행하게 함으로써 다양한 Output 포트를 가질 때, 어떠한 Output 포트로 전송하는 것이 가장 효율적일지에 대한 문제를 MABP를 적용하여 해결하였다.

3. 제안 사항

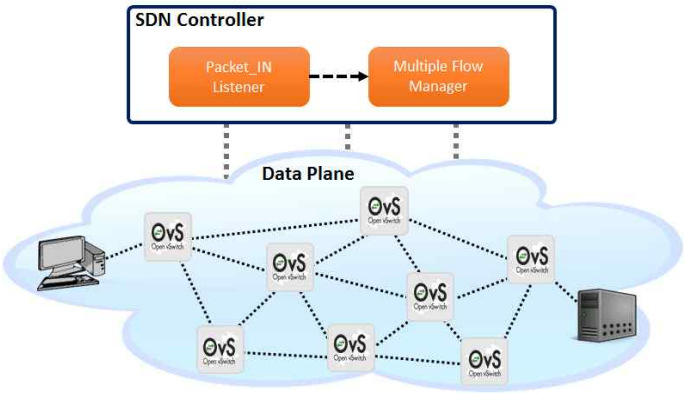

그림 2 Overall Network Structure

그림 2는 전체적인 네트워크 구조를 나타낸 것으로 네트워크 호스트에서 OpenFlow 스위치로 패킷이 유입되었을 때, 해당 패킷은 Packet_IN 메시지를 통해 Controller로 전달된다. 이때, Controller는 Source MAC/IP와 Destination MAC/IP를 확인 후 Hop수 기반의 Dikstra 알고리즘을 통해 최대 3개의 전송 경로에 대한 Flow Entry를 계산한 후 각 노드로 배치한다.

표 1 다중 경로 Flow Table

Flow Entry	Match Field	Action
Flow 1	priority=3, in_port=1, dst_mac/IP, src_mac/IP	output:2
Flow 2	priority=3, in_port=2, dst_mac/IP, src_mac/IP	output:1
Flow 3	priority=2, in_port=1, dst_mac/IP, src_mac/IP	output:3
Flow 4	priority=2, in_port=3, dst_mac/IP, src_mac/IP	output:1
Flow 5	priority=1, in_port=1, dst_mac/IP, src_mac/IP	output:4
Flow 6	priority=1, in_port=4, dst_mac/IP, src_mac/IP	output:1

표 1은 컨트롤러에서 스위치로 다중 경로에 대한 Flow Table을 배치한 결과이다. 각 경로는 서로 다른 우선순위를 갖게 되며 이를 통해 각 노드에 공존할 수 있다. 스위치에서는 위와 같은 Flow Table을 기반으로 각 Entry에 대한 Reward 값을 계산하게 된다.

$$rxRate(t) = rxBytes(t) - rxBytes(t-1) \quad (1)$$

$$txRate(t) = txBytes(t) - txBytes(t-1) \quad (2)$$

$$AvailableBW(t) = [MaxBW - (txRate(t) + rxRate(t)) * 8] \quad (3)$$

$$TDelay(t) = \frac{(PLength * 8)}{AvailableBW(t)} \quad (4)$$

$$Reward(t) = \frac{1}{TDelay(t)} \times Weight \quad (5)$$

수식 (1), (2), (3)으로 부터 얻을 수 있는 Output의 가용대역폭은 최대 대역폭에서 현재 사용되고 있는 Tx, Rx Rate를 더한 값(현재 대역폭 사용량)을 제외한 나머지로 정의된다. Tx, Rx Rate 값은 패킷이 유입된 시간에서의 Tx, Rx Bytes와 1초 이전의 Tx, Rx Bytes 값을 통해 구할 수 있다. Reward 값은 시간 t에서 유입되는 Packet의 Length에 대한 Output 포트의 전송 지연 평가 값의 역수로써 수식 (4), (5)에 의해 계산되며, 전송지연 값이 가장 작은 Output 포트가 패킷 전송의 최우선순위를 갖는 포트로 선택된다. 또한 Input 포트에 대하여 하나의 Output 포트를 갖는 Flow Entry는 반드시 선택되어 대기상태에 들어간다. 이는 Sub Module이 Exploitation을 수행하여 가장 높은 보상 값을 갖는 포트를 통해 패킷을 전송하는 과정이다.

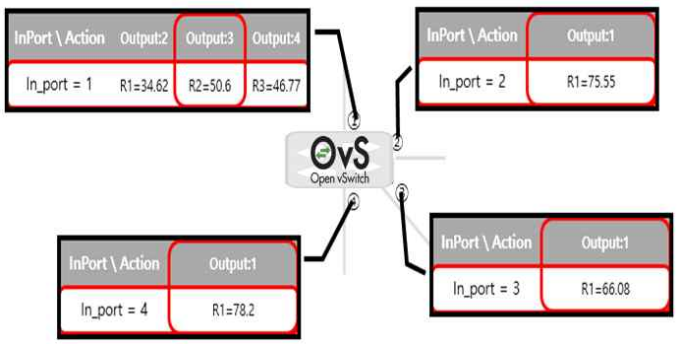

그림 3 Input 포트 별 Reward Table

그림 3은 Sub Module에 의해 Exploitation이 완료된 상태의 OpenFlow 스위치를 나타낸 모습이다. 선택된 Flow Entry를 제외한 나머지 Entry의 경우, Idle_Timeout 설정에 의해 15초간 사용되지 않으면 Flow Table에서 삭제된다.

알고리즘 Path Decision on Sub Module

```
1:  FT ← Flow table from Controller
2:  FE ← Flow entry in FT
3:  FEC ← Flow entry chosen by Module for forwarding
4:  FEE ← Existing flow entry for forwarding
5:  while true
6:    if FE for incoming packet in FT then
7:      forward packet through FEE
8:    else
9:      calculate reward of output port
10:     if input port has multiple output port then
```

```
11:            choose the output port which has best reward
12:        else
13:            choose the output port
14:        end if
15:        update FEc in FT with higher priority than other FE
16:        forward packet through FEc
17:    end if
```

유입되는 패킷의 특징들, 예를 들어 Packet Length, Source IP, Destination IP 등이 변하게 되면 Packet_IN 메시지에 의해 새로운 다중 경로에 대한 Flow Table이 배치됨으로써 Exploration을 수행하게 된다. 또한 Idle_Timeout에 의해 Flow Entry가 삭제되었을 때, 패킷의 특징이 같다 하더라도 Table 매치 Miss가 발생해 Packet_IN 메시지가 전송되므로 다시 Exploration이 수행되게 된다.

4. 성능 평가

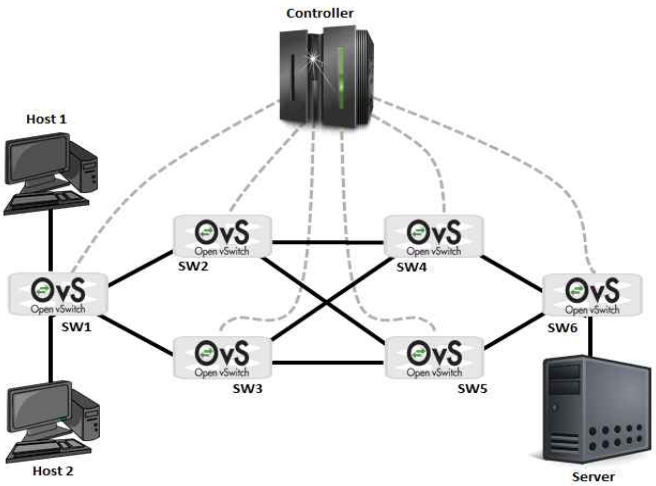

그림 4 성능평가를 위한 네트워크 토폴로지

그림 4는 성능평가에 활용된 네트워크 토폴로지로써, OpenVSwitch 여섯 노드와 2대의 Host, 1대의 서버로 구성되어 있으며 Floodlight 컨트롤러가 해당 네트워크를 관리한다.

표 2 성능평가 시나리오

Evaluation Scenario		
End to End	Test Tool	Time Duration
Host1-Server	Iperf Tool	0-30s
Host2-Server	Iperf Tool	15s-45s

표 2는 성능평가 시나리오로써 Iperf Tool을 활용하여 각 호스트의 패킷 전송에 따른 대역폭 사용량을 확인하기 위한 시나리오이며 각 호스트가 전송하는 시간 동안의 평균값을 통해 대역폭 사용량을 확인하였다.

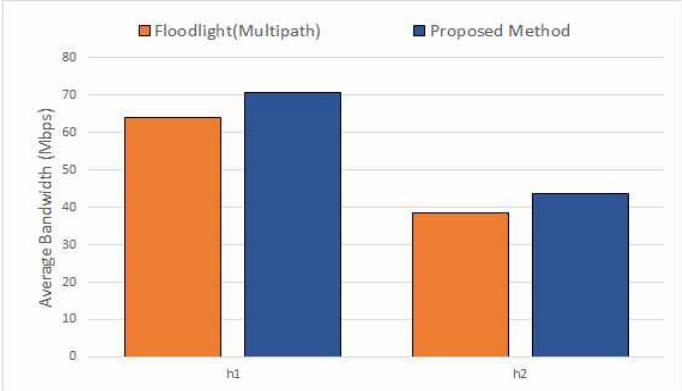

그림 5 패킷 전송에 따른 평균 대역폭 사용량

그림 5는 성능평가의 결과로써 패킷 전송에 따른 평균 대역폭 사용량을 나타낸 그래프이다. Multipath 전송을 지원하는 Floodlight Controller[6]의 경우 63.9Mbps와 38.5Mbps였으며 제안하는 기법의 경우 70.7Mbps와 43.7Mbps로 Multipath Floodlight Controller 보다 약 10.6%, 13.5% 증가하였다. 이는 제안하는 기법에서 Host 2가 패킷을 전송할 때, 사용 중인 경로를 제외한 나머지 경로, 즉 각 전송포트의 Reward값에 의한 경로를 통해 패킷을 전송하였기 때문에 전체적인 평균 대역폭 사용량이 Multipath를 활용할 때보다 높게 측정되었다.

5. 결론 및 향후 연구

본 논문에서는 SDN에서 Data Plane을 활용한 분산 라우팅 기법을 제안하였다. 기존 Controller 기반의 라우팅의 경우 라우팅 메트릭에 있어 많은 요소를 고려함으로써 Data Plane으로부터 다양한 정보를 얻어야 하고 네트워크 scale이 커질수록 Controller의 부하로 이어진다. 이에 Data Plane을 활용하여 전송 경로를 설정한다면 네트워크 관리에 있어 보다 유연한 라우팅 기능을 제공할 수 있다. 향후 연구로는 SDN 기반의 무선 메쉬 네트워크 환경에서 본 제안사항이 적용된 라우팅 Mechanism을 연구할 계획이며 AODV, OSLR, HWMP 등 기존 메쉬 네트워크에 활용되는 라우팅 프로토콜과 성능을 비교하고자 한다.

6. 참고 문헌

[1] Danda B. Rawat, Swetha R. Reddy, "Software Defined Networking Architecture, Security and Energy Efficiency: A Survey," IEEE Communications Survey & Tutorials, Vol 19, No. 1, Firstquater 2017

[2] Israat Tanzeena Hqaue, Nael Abu-Ghazaleh., "Wireless Software Defined Networking: A Survey and Taxonomy," IEEE Communications Surveys & Tutorials, Vol 18, No. 4, Fourthquarter 2016

[3] Hesham Mekky, Fang Hao, Sarit Mukherjee, Zhi-Li Zhang, T.V. Lakshman, "Application-aware data plane processing in SDN," 2014 third workshop on Hot topics in software defined networking, Chicaho, USA, Aug 22, 2014

[4] Mohammed Labraoui, Michael Mathias Boc, Anne Fladenmuller, "Software Defined Networking-assisted routing in wireless mesh networks," 2016 International Conference on Wireless Communications and Mobile Computing(IWCMC 2016), Cyprus, Sep.5 - Sep.9, 2016

[5] Wikipedia, "Multi-armed bandit," Online Available https://en.wikipedia.org/wiki/Multi-armed_bandit

[6] Github, "MultipathRouting," Online Available http://github.com/hwchiu/MultipathRouting