

Mininet 환경 구성 및 기본 사용법

2024년 8월

경북대학교 사물인터넷표준연구실

라 연 (ry0218@knu.ac.kr)

요 약

Mininet(미니넷)은 가상 네트워크 환경을 쉽게 구축하고 실험할 수 있는 오픈소스 도구로, 주로 SDN(소프트웨어 정의 네트워킹) 및 네트워크 연구에 활용된다. 실제 네트워크 장비 없이도 쉽게 가상으로 구현하여 다양한 네트워크 환경을 실험할 수 있다. Mininet 설치와 환경 설정 구축 그리고 간단한 예제를 통해 사용법을 숙지하는 것을 목표로 한다.

목 차

1. 서론	2
2. MININET.....	2
2.1 MININET 설치 및 환경 설정	2
2.2 MININET 간단한 사용법	4
2.2.1 가상 네트워크 환경 구성.....	4
2.2.2 Log 파일 분석하는 법	5
3. 결론	6
참고 문헌.....	6

1. 서론

본 문서에서는 가상 네트워크 환경을 쉽게 구성할 수 있는 도구인 Mininet의 설치 및 환경 설정 방법과 예제를 통한 간단한 사용법을 설명할 것이며, Linux 운영체제 환경에서 구현한다.

2. Mininet

2.1 Mininet 설치 및 환경 설정

```
○ → repo cd mininet_exp_env
```

(1) 먼저 리눅스 환경에서 [mininet_exp_env.tar] 파일의 압축을 풀고 터미널에서 압축을 푼 폴더 경로에 접속한다.

```
○ → mininet_exp_env git:(main) X sudo apt install mininet
```

(2) 해당 폴더에서 `sudo apt install mininet` 명령어를 통해 Mininet을 설치한다.

```
● → mininet_exp_env git:(main) X mn --version  
2.3.0
```

미니넷이 잘 설치됐는지 `mn -version` 명령어를 통해 확인할 수 있다.

(3) 다음으로는 Mininet에 필요한 필수 유틸리티를 설치해야 한다.

```
○ → mininet_exp_env git:(main) X git clone https://github.com/mininet/mininet.git
```

git clone 후

```
○ → mininet_exp_env git:(main) X bash mininet/util/install.sh -fw
```

install.sh 파일을 실행시켜 주면 된다.

(4) Mininet을 실행시킬 때 runner.py 파일 실행 과 mininet 파이썬 모듈 설치를 위해 python3 및 pip를 설치해줘야 한다. 그리고 로그 분석에서 필요한 iperf3도 추가로 설치해 준다.

```
○ → mininet_exp_env git:(main) X sudo apt install python3-pip iperf3
```

*** 여기서 'iperf3'를 설치해 주는 이유는 기본적으로 [ping 명령어]를 통해 응답시간과 패킷 손실률과 같은 네트워크 연결 상태를 확인할 수 있지만, [iperf3 명령어]를 통해 대역폭, 지연 시간, 지터(jitter) 등 더 상세한 성능 데이터를 확인할 수 있기 때문이다.

(5) runner.py 파이썬 파일을 통해 가상 환경을 구성하고 실행시키기 때문에 python의 Mininet 패키지를 설치해야 한다.

```
○ → mininet_exp_env git:(main) X sudo su  
root@raspberrypi:/home/admin/repo/mininet_exp_env# pip install mininet
```

*** [sudo su 명령어]는 su(switch user) 즉 현재 사용자로부터 루트 사용자 권한을 얻고, 루트 사용자로서 명령어를 실행할 수 있게 해준다.

(6) 다음으로는 Mininet을 실행시키기 전 필수로 해줘야 할 IP Forwarding 활성화이다.

```
● → mininet_exp_env git:(main) X sudo sysctl -w net.ipv4.ip_forward=1  
[sudo] password for admin:  
net.ipv4.ip_forward = 1
```

활성화가 잘 되었는 지 확인하기 위해선 [sudo sysctl net.ipv4.ip_forward 명령어]를 입력해 주면 되고 만약 0으로 설정되어 있다면 IP Forwarding을 1로 활성화 해 줘야 한다.

```
● → mininet_exp_env git:(main) X sudo sysctl net.ipv4.ip_forward  
net.ipv4.ip_forward = 1
```

*** 시스템을 재시작(reboot) 하거나 시스템을 새로 다시 켜다면 이 설정을 꼭 해줘야 정상적으로 Mininet이 실행된다.

2.2 Mininet 간단한 사용법

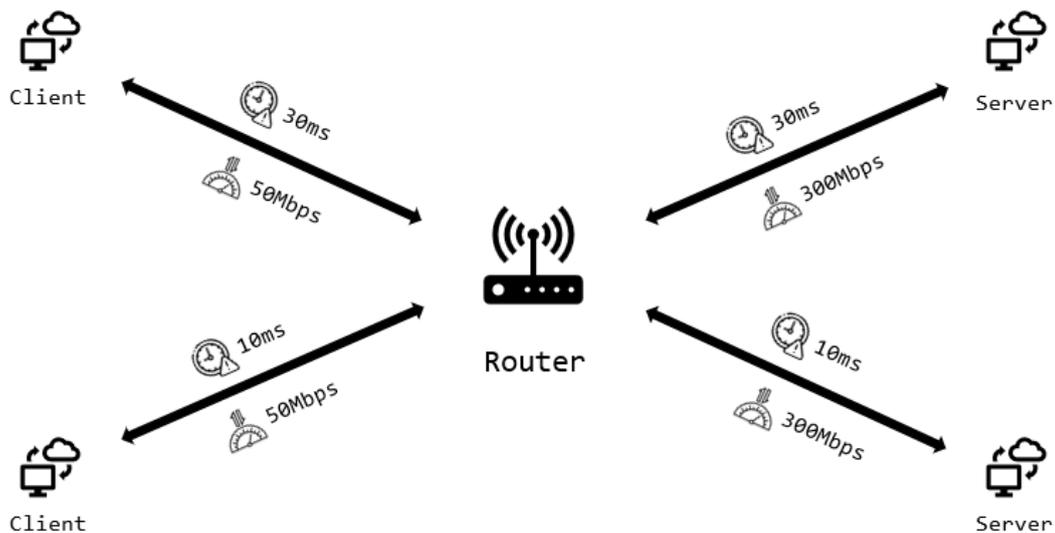
2.2.1 가상 네트워크 환경 구성

가상 네트워크 환경을 구성하기 위해선 config/topo 폴더 안에 topo-15_1000_100_0 파일의 값들을 적절하게 설정해주면 된다.

```
config > topo > ≡ topo-15_1000_100_0
1  clients:8
2  servers:5
3  leftSubnet:192.168.
4  rightSubnet:10.1.
5  path_c2r_0:30,1000,50,0
6  path_c2r_1:10,1000,50,0
7  path_r2s_0:30,1000,300,0
8  path_r2s_1:10,1000,300,0
9  topoType:MultiInterfaceMultiClients
```

1~8 line의 설정을 통해 Mininet 환경 구성을 쉽게 할 수 있다.

- 1) 먼저 실험을 진행시킬 Client와 Server 수를 설정한다.
- 2) 그리고 Client와 Server의 Subnet을 설정한다. 여기서 left는 Client, right는 Server이다.
- 3) 5~8 line은 Client와 Router, Router와 Server 사이의 경로에 대한 설정 값이다.
(c2r은 Client to Router, r2s는 Router to Server 를 의미하고 path는 1개 이상 설정 가능하다.)
차례로 Delay, Queue Size, Bandwidth, Loss이다.
결과로 Client, Router, Server 사이의 지연시간, 대역폭 등과 같은 실험 값을 확인할 수 있다.



topo파일 설정을 통해 구성된 가상환경 예시

2.2.2 Log 파일 분석하는 법

(1) 실험에 적용할 값들을 설정한 뒤 가상 네트워크를 구성하기 위해서 runner.py 파이썬 파일을 실행한다.

```
o → mininet_exp_env git:(main) X sudo python runner.py -t config/topo/topo-15_1000_100_0 -x config/xp/ping_exp-simple-no
```

-t 옵션 뒤에는 앞서 설정한 topo파일을, -x 옵션 뒤에는 네트워크 상태를 확인하기 위해 ping(config/xp/ping_exp-simple-no)이나 iperf(config/xp/iperf_exp-simple-no) 파일을 배치한다.

(2) 프로그램 실행이 완료되면, ping.log, client-n-log.txt 등 여러가지 log가 남게 된다.

먼저 ping.log는 Client-Server 간의 ping에 대한 log가 남는다.

```
≡ ping.log
```

```
1 PING 10.1.0.2 (10.1.0.2) from 192.168.0.2 : 56(84) bytes of data.
2 64 bytes from 10.1.0.2: icmp_seq=1 ttl=63 time=256 ms
3 64 bytes from 10.1.0.2: icmp_seq=2 ttl=63 time=121 ms
4 64 bytes from 10.1.0.2: icmp_seq=3 ttl=63 time=121 ms
5 64 bytes from 10.1.0.2: icmp_seq=4 ttl=63 time=121 ms
6 64 bytes from 10.1.0.2: icmp_seq=5 ttl=63 time=121 ms
7
8 --- 10.1.0.2 ping statistics ---
9 5 packets transmitted, 5 received, 0% packet loss, time 4005ms
```

여기서 앞서 설정해준 left, right Subnet 192.168. / 10.1. 뒤의 3번째, 4번째 자리는 각각 path의 index, Client나 Server의 index를 의미한다. 예를 들어 그림에서의 상황은 첫번째 Client와 첫번째 Server 사이에서 두개의 노드 모두 첫번째 경로를 이용해 통신한 경우이다. 만약 두번째 path를 거치고, 두번째 노드였다면 3번째, 4번째 자리가 192.168.1.3 / 10.1.1.3 이런 식으로 구성된다. (path 0 / path 1... , node 첫번째부터 index는 2, 3, 4, 5...)

추가로 iperf 명령어 결과 값은 밑의 그림과 같이 노드 간의 경로에서의 Bandwidth를 확인할 수 있다.

```
≡ client-0-log.txt
```

```
1 -----
2 Server listening on TCP port 5001
3 TCP window size: 85.3 KByte (default)
4 -----
5 [ 1] local 192.168.0.2 port 5001 connected with 192.168.0.3 port 47604 (icwnd/mss/irrt=14/1448/127130)
6 [ ID] Interval      Transfer      Bandwidth
7 [ 1] 0.0000-12.3359 sec 56.0 MBytes 38.1 Mbits/sec
8 [ 2] local 192.168.0.2 port 5001 connected with 10.1.0.3 port 46178 (icwnd/mss/irrt=14/1448/122154)
9 [ ID] Interval      Transfer      Bandwidth
10 [ 2] 0.0000-12.3477 sec 67.1 MBytes 45.6 Mbits/sec
```

3. 결론

지금까지 가상 네트워크 환경을 쉽게 구축하고 실험할 수 있는 오픈소스 도구인 Mininet(미니넷)에 대한 간단한 소개와 설치 방법 및 환경 구성, 마지막으로 간단한 사용법에 대해 알아보았다. 네트워크의 전반적인 통신 흐름에 대해 쉽게 이해할 수 있고 실험할 수 있어 네트워크를 공부하고 연구하는 데에 많은 도움이 될 것으로 생각된다.

참고 문헌

- [1] Mininet, <https://mininet.org/>
- [2] Mininet 설치 및 실행하기, <https://>
- [3] ping과 iperf 명령어 사용, <https://stackoverflow.com/questions/63907768/use-iperf-and-ping-at-the-same-time-mininet>