네트워크보안 과제8

방화벽

202246109
김기현
2025년 6월 4일

사용 소프트웨어

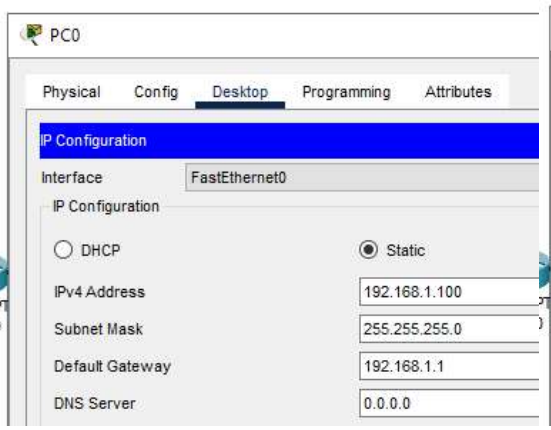
- Cisco Packet Tracer (네트워크 시뮬레이터)

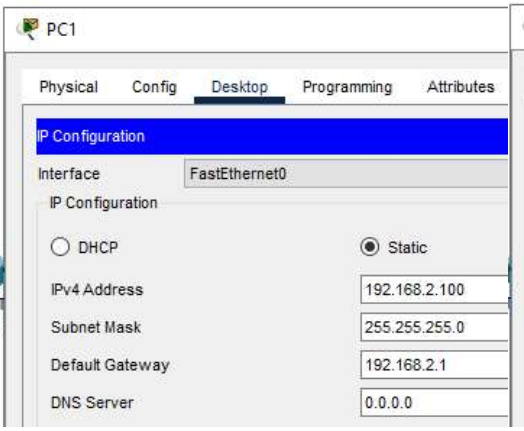
표준 ACL(Access Control List)은 네트워크 장비(주로 라우터)에서 데이터 패킷의 흐름을 제어하는 가장 기본적인 접근 제어 방식입니다. 이번 과제에서는 packettracer를 활용해서 다음과 같이 수행하겠습니다.

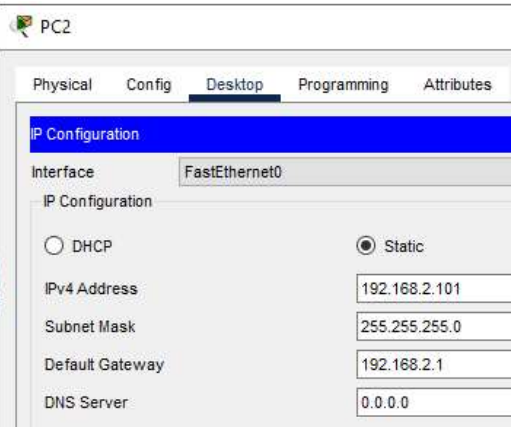
- 네트워크 구성
  - 네트워크 1(192.168.1.x)는 PC 1대와 스위치 1대로 구성
  - 네트워크 2(172.168.1.x)는 PC 2대와 스위치 1대로 구성
  - 두 네트워크를 Router0, Router1로 연결
- ACL 설정

- Router0에서 PC2의 ping을 거부하도록 설정

1. PC 설정하기

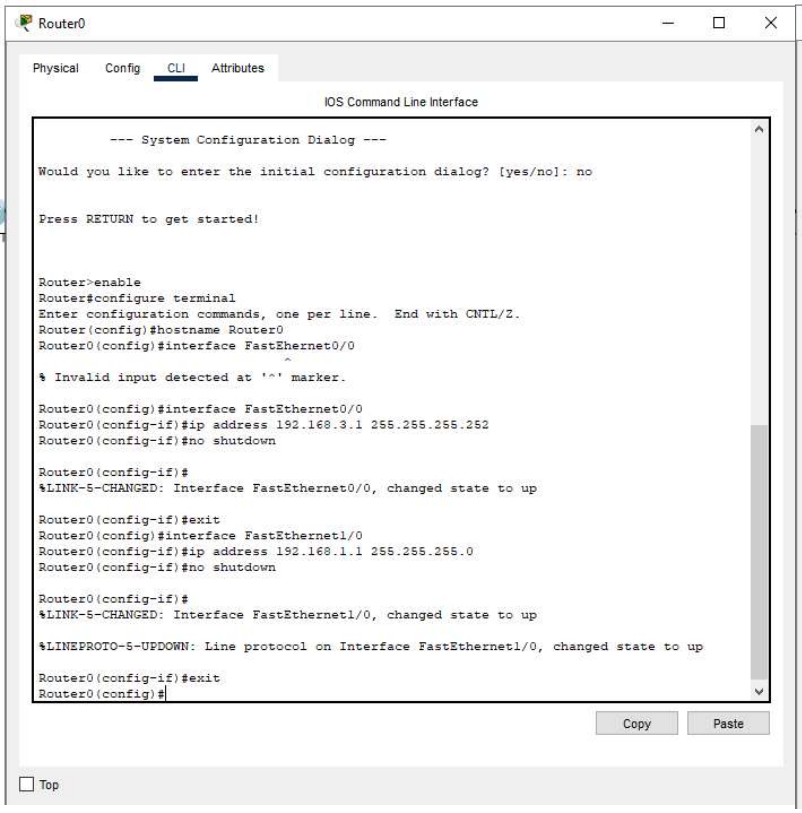


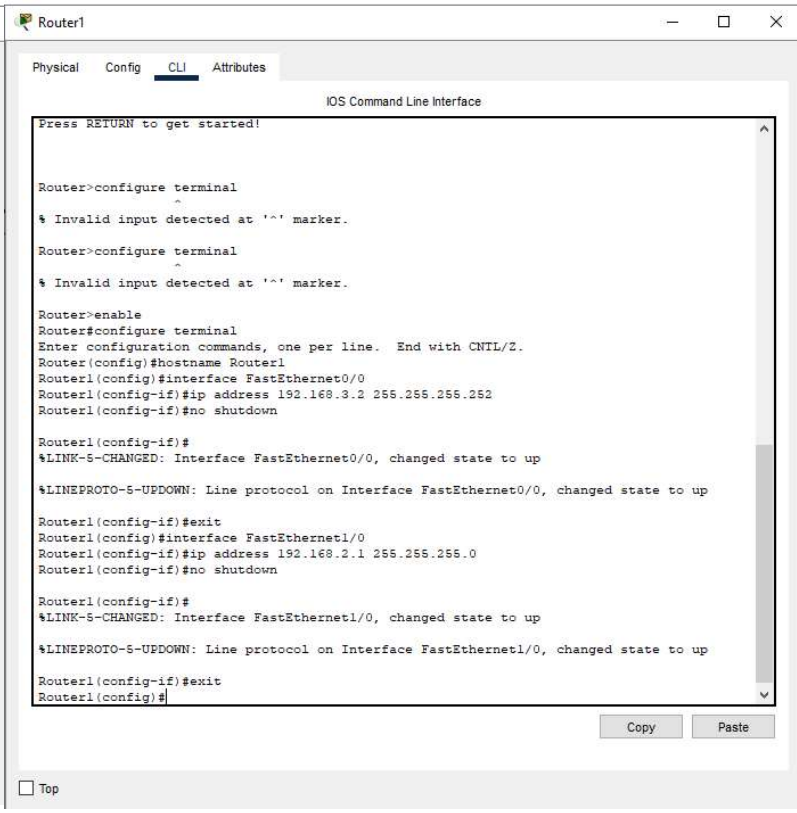




장치	IP주소	서브넷 마스크	게이트웨이
PC0	192.168.1.100	255.255.255.0	192.168.1.1
PC1	192.168.2.100	255.255.255.0	192.168.2.1
PC2	192.168.2.101	255.255.255.0	192.168.2.1

2. 라우터 설정하기





3. 라우팅 설정하기

```
Router0(config)#ip route 192.168.2.0 255.255.255.0 192.168.3.2
Router0(config)#
```

```
Router1(config)#ip route 192.168.1.0 255.255.255.0 192.168.3.1
Router1(config)#
```

네트워크 구성을 마치고, ACL 설정하기 전에 PC 간 연결을 ping 으로 테스트 해보겠습니다

- 연결 테스트 -

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.2.100

Pinging 192.168.2.100 with 32 bytes of data:

Request timed out.
Request timed out.
Reply from 192.168.2.100: bytes=32 time=10ms TTL=126
Reply from 192.168.2.100: bytes=32 time=10ms TTL=126

Ping statistics for 192.168.2.100:
    Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
Approximate round trip times in milli-seconds:
    Minimum = 10ms, Maximum = 10ms, Average = 10ms

C:\>
```

PC0 -> PC1

```
C:\>ping 192.168.2.101

Pinging 192.168.2.101 with 32 bytes of data:

Request timed out.
Reply from 192.168.2.101: bytes=32 time=10ms TTL=126
Reply from 192.168.2.101: bytes=32 time=10ms TTL=126
Reply from 192.168.2.101: bytes=32 time=10ms TTL=126

Ping statistics for 192.168.2.101:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 10ms, Maximum = 10ms, Average = 10ms

C:\>
```

PC0 -> PC2

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.1.100

Pinging 192.168.1.100 with 32 bytes of data:

Reply from 192.168.1.100: bytes=32 time=20ms TTL=126
Reply from 192.168.1.100: bytes=32 time=22ms TTL=126
Reply from 192.168.1.100: bytes=32 time=21ms TTL=126
Reply from 192.168.1.100: bytes=32 time=21ms TTL=126

Ping statistics for 192.168.1.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 20ms, Maximum = 22ms, Average = 21ms

C:\>
```

PC1 -> PC0

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.1.100

Pinging 192.168.1.100 with 32 bytes of data:

Reply from 192.168.1.100: bytes=32 time=10ms TTL=126
Reply from 192.168.1.100: bytes=32 time=10ms TTL=126
Reply from 192.168.1.100: bytes=32 time=10ms TTL=126
Reply from 192.168.1.100: bytes=32 time=10ms TTL=126

Ping statistics for 192.168.1.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 10ms, Maximum = 10ms, Average = 10ms

C:\>
```

PC2 -> PC0

테스트 결과 모든 연결이 정상적으로 이루어졌음을 확인했습니다.

4. ACL 설정하기

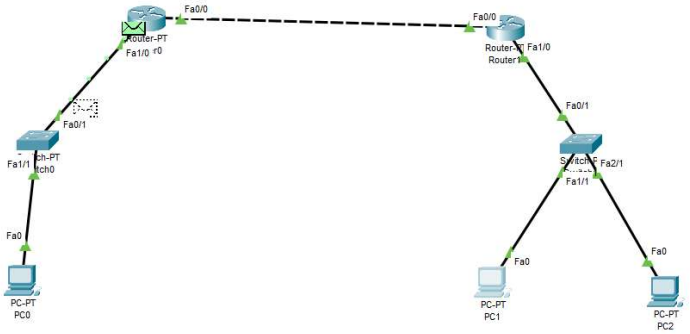
Router0에서 PC2(192.168.2.101)의 ping을 거부하도록 설정하겠습니다.

```
Router0(config)#access-list 1 deny 192.168.2.101 0.0.0.0
Router0(config)#access-list 1 permit any
```

설정한 표준 ACL를 인터페이스에 적용하기 위해 Router0에서 나가는 방향의 첫 번째 포트인 FastEthernet1/0을 out으로 설정하겠습니다.

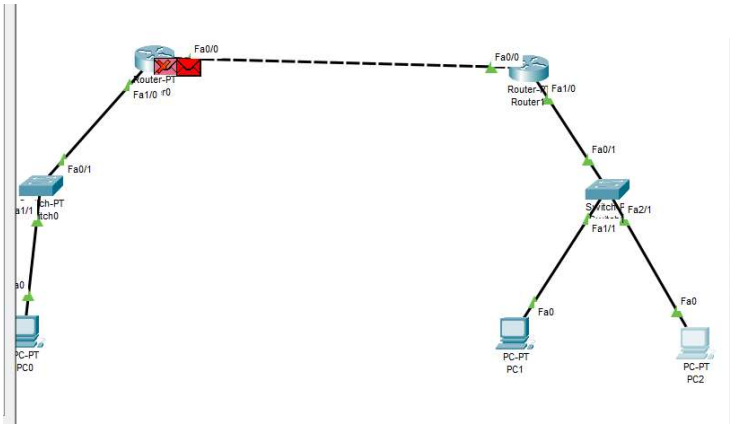
```
Router0(config)#interface FastEthernet1/0
Router0(config-if)#ip access-group 1 out
```

ACL 설정 후 ping 을 날려 결과를 확인해보겠습니다.




```
C:\>ping 192.168.1.100

Pinging 192.168.1.100 with 32 bytes of data:

Reply from 192.168.1.100: bytes=32 time=22ms TTL=126
Reply from 192.168.1.100: bytes=32 time=24ms TTL=126
Reply from 192.168.1.100: bytes=32 time=18ms TTL=126
Reply from 192.168.1.100: bytes=32 time=24ms TTL=126

Ping statistics for 192.168.1.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 18ms, Maximum = 24ms, Average = 22ms
```

PC1 -> PC0과의 통신은 정상적으로 이루어지지만



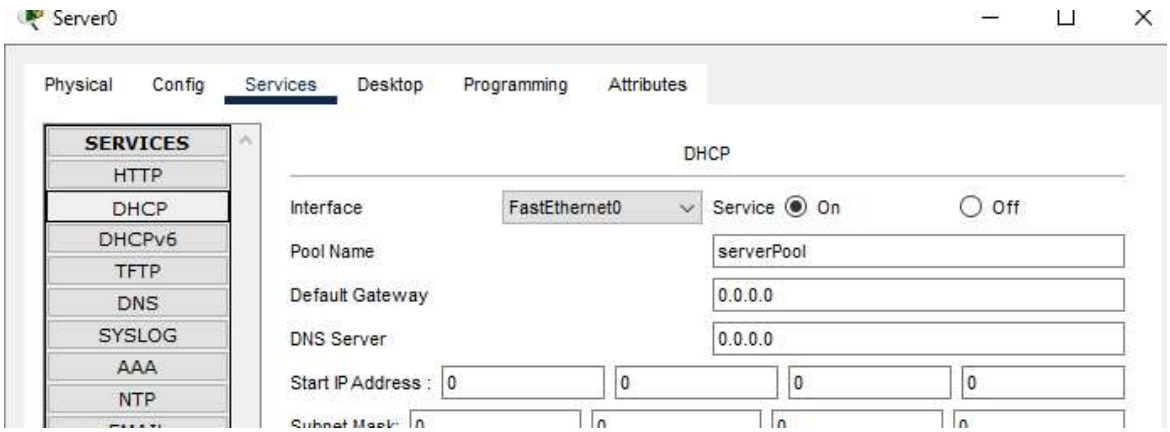

```
C:\>ping 192.168.1.100

Pinging 192.168.1.100 with 32 bytes of data:

Reply from 192.168.3.1: Destination host unreachable.
Reply from 192.168.3.1: Destination host unreachable.
Reply from 192.168.3.1: Destination host unreachable.
Reply from 192.168.3.1: Destination host unreachable.

Ping statistics for 192.168.1.100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
```

PC2 -> PC0과의 통신에서는 Router0에서 거절 하는 모습을 확인할 수 있습니다.

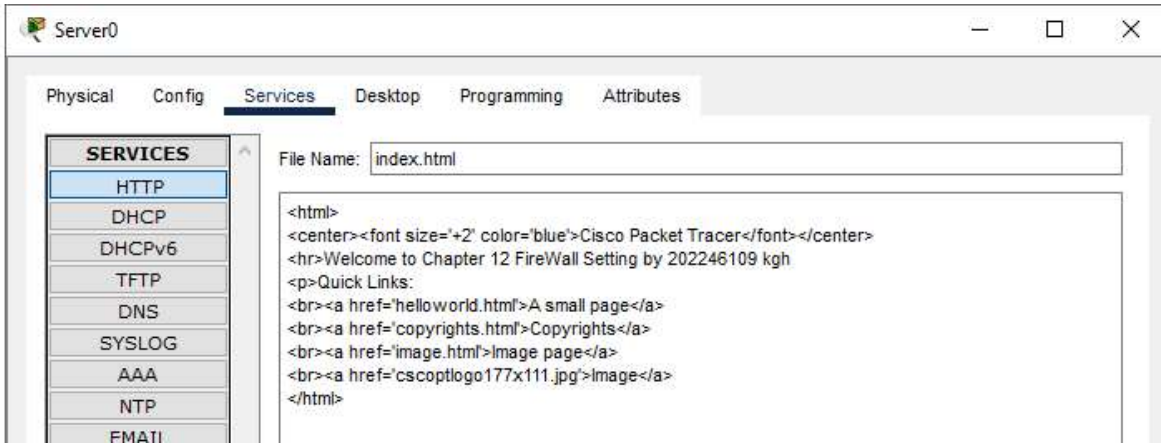
방화벽을 이용해 서로 다른 protocol을 허용하고 거부하는 방법을 실습해보겠습니다.

- 네트워크 구성
  - 스위치 1대에 PC 3대(PC0, PC1, PC2)와 서버 1대를 고속 이더넷으로 연결
  - 서버는 DHCP 서버, 웹 서버, 방화벽 기능을 모두 수행함

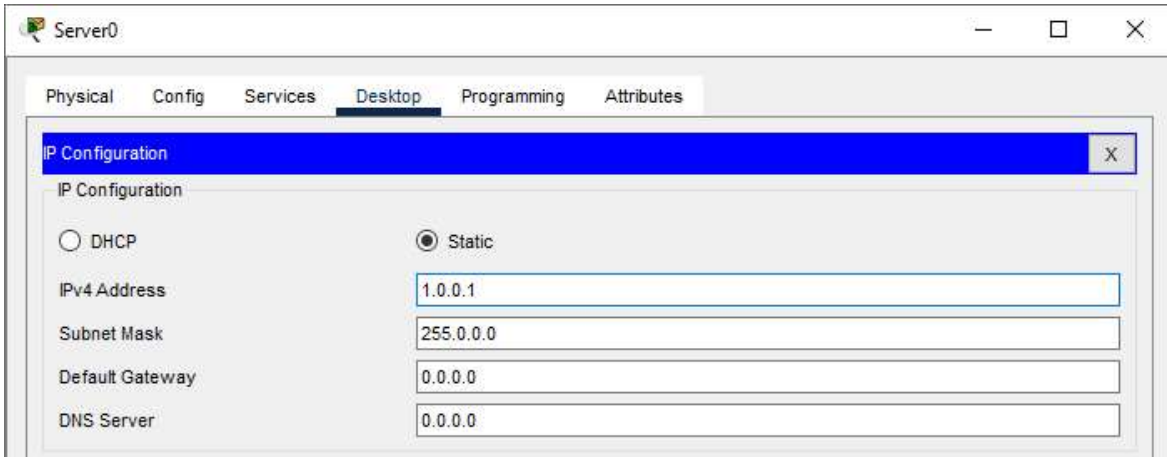
1. 서버 설정하기



DHCP 설정

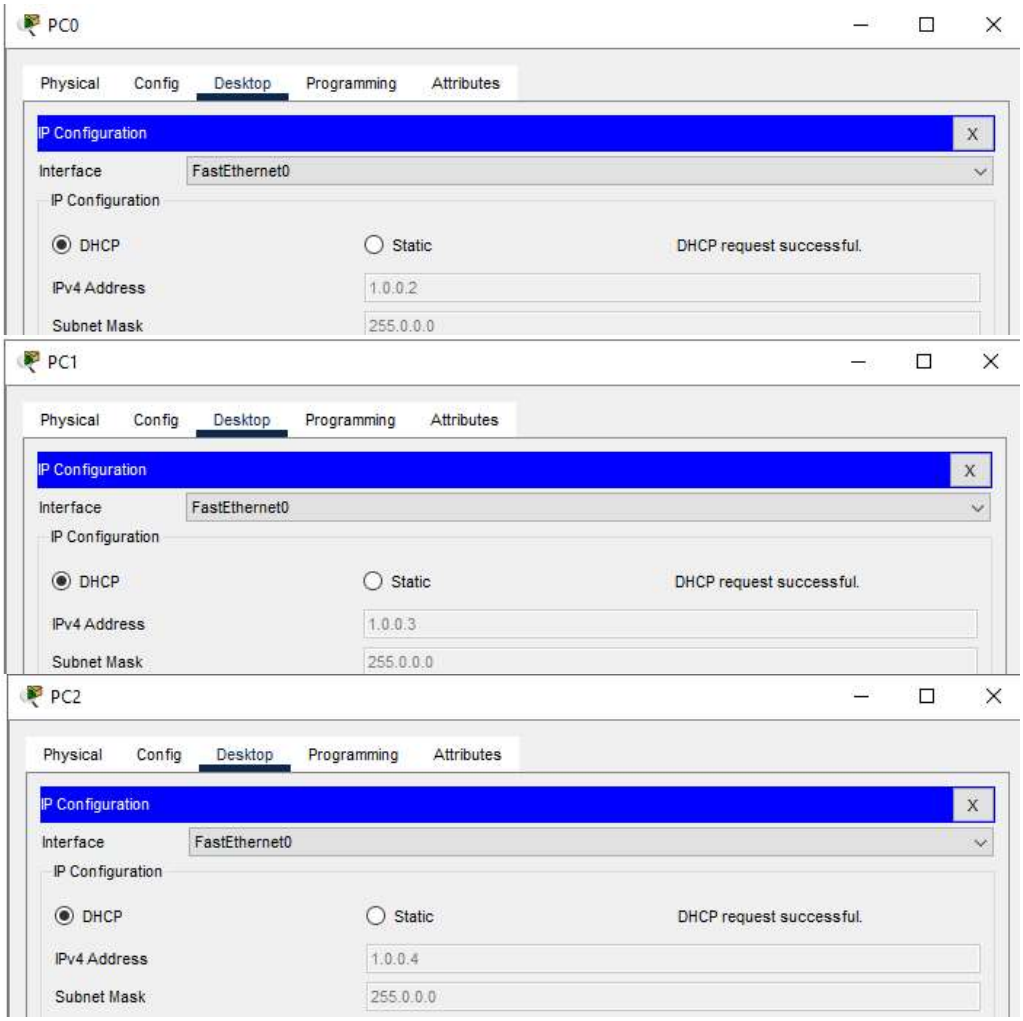


HTTP 설정



IP 설정

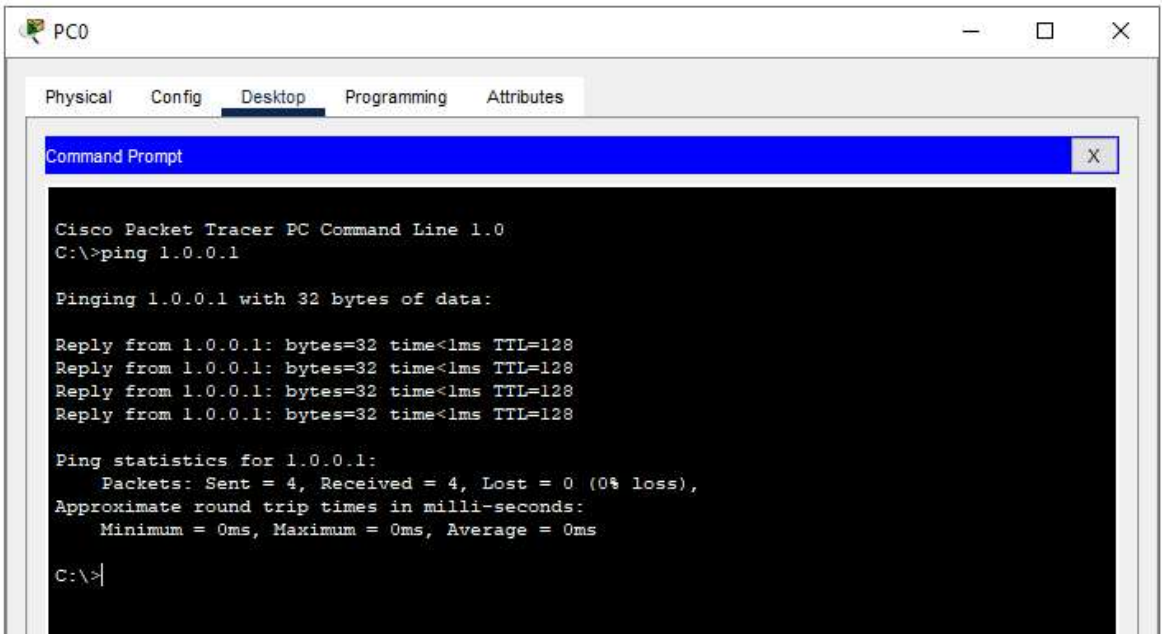
2. PC 설정하기



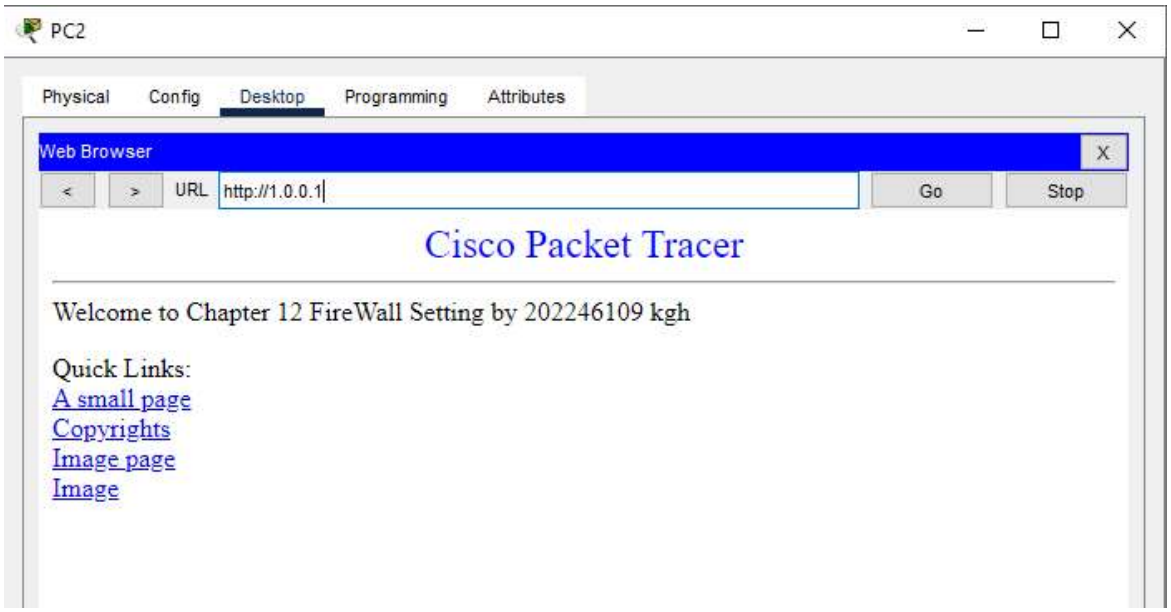
DHCP로 IP 자동 할당

- 연결 테스트 -

네트워크 구성을 마치고, 방화벽을 설정하기 전에 연결 테스트를 해보겠습니다.



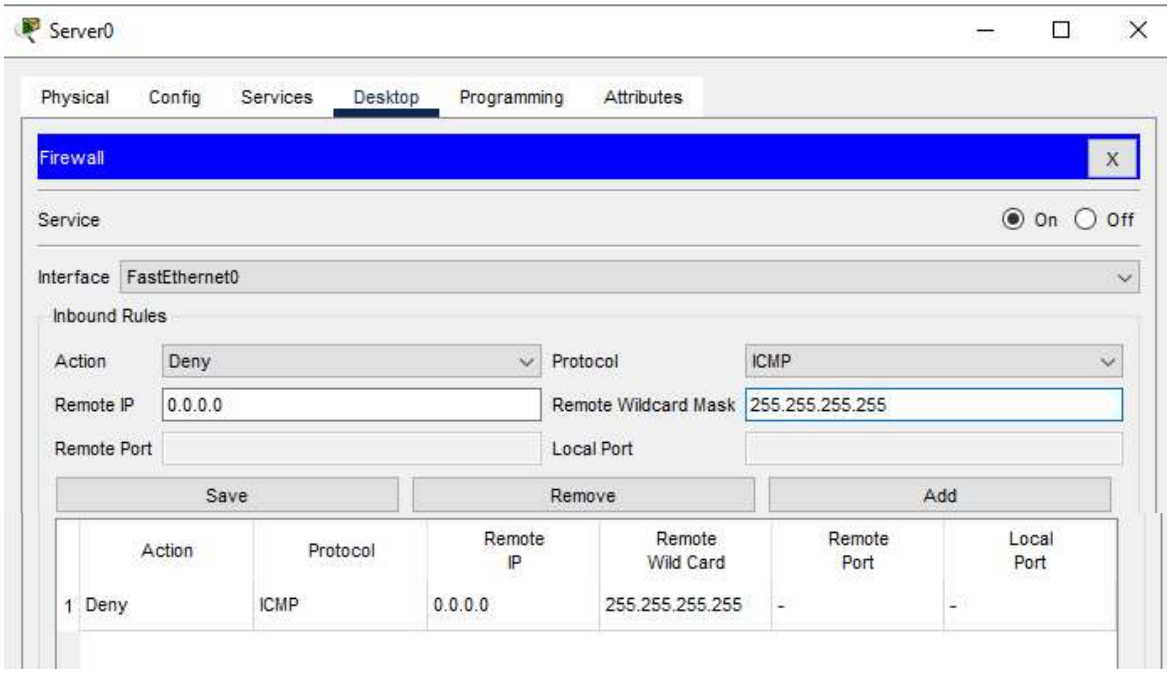
ICMP PC0 -> Server(1.0.0.1)



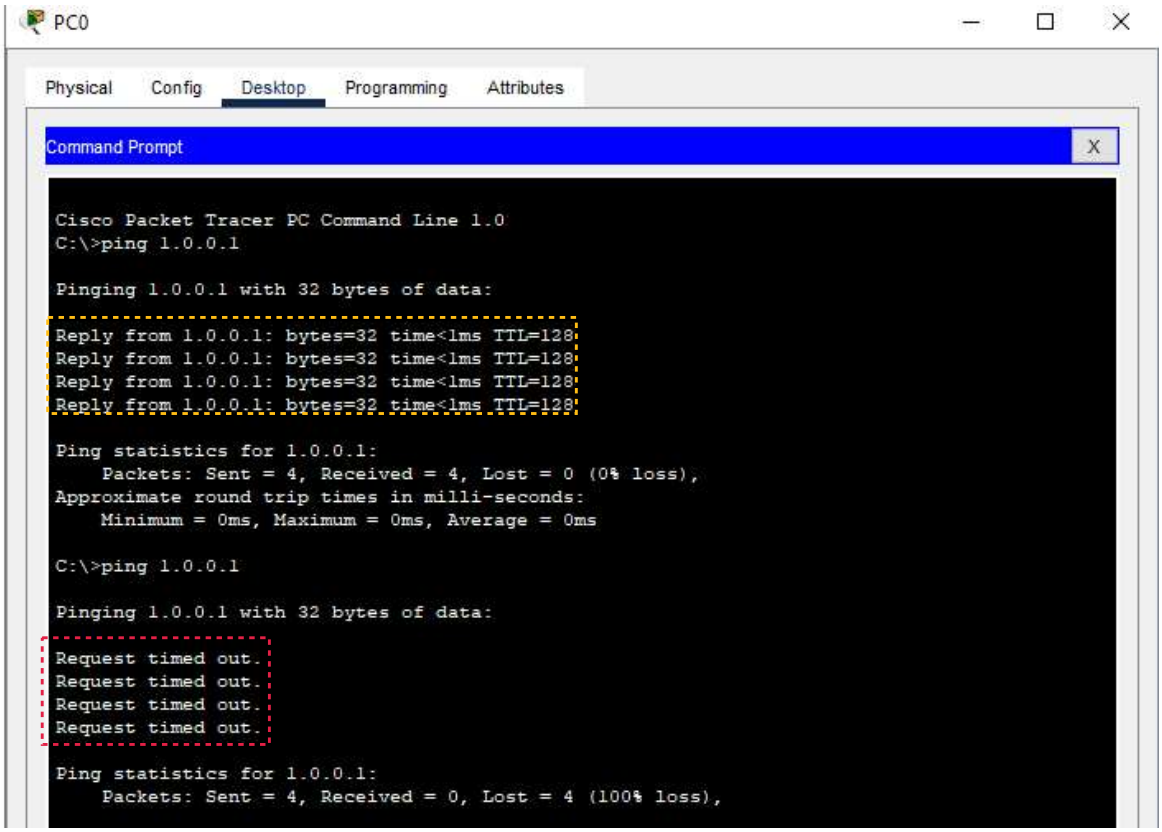
HTTP PC2 -> Server(1.0.0.1)

모든 통신이 정상적으로 수행됨을 확인 했습니다.

3. ICMP 프로토콜 차단 설정하기



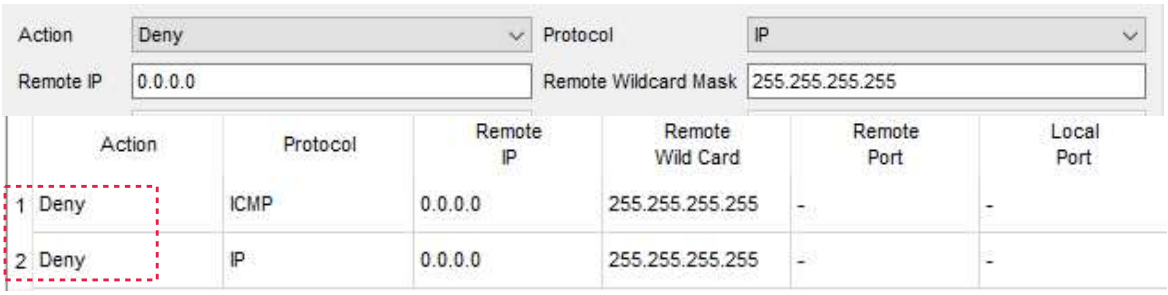
Remote IP 0.0.0.0/32로부터의 ICMP 패킷을 차단하여 모든 외부에서의 Ping 요청을 차단했습니다.



ICMP PC0 -> Server(1.0.0.1)

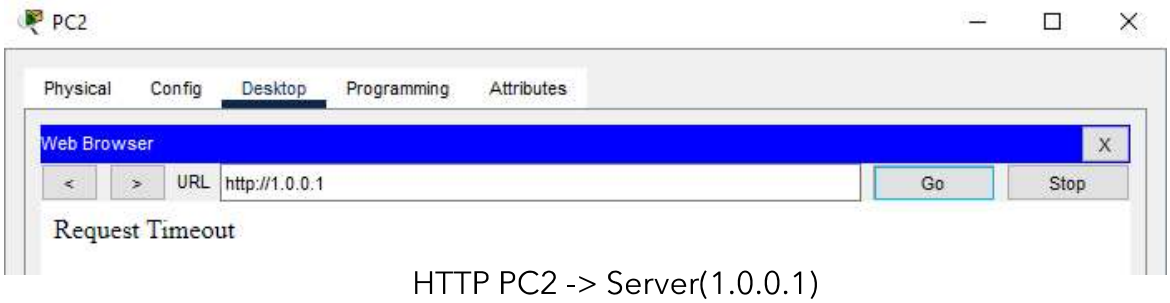
기존과 달리 Request timed out 메시지가 나타난다.

4. IP 프로토콜 차단 설정하기



	Action	Protocol	Remote IP	Remote Wild Card	Remote Port	Local Port
1	Deny	ICMP	0.0.0.0	255.255.255.255	-	-
2	Deny	IP	0.0.0.0	255.255.255.255	-	-

모든 외부 IP로부터 들어오는 모든 IP 프로토콜을 차단하도록 설정했습니다.



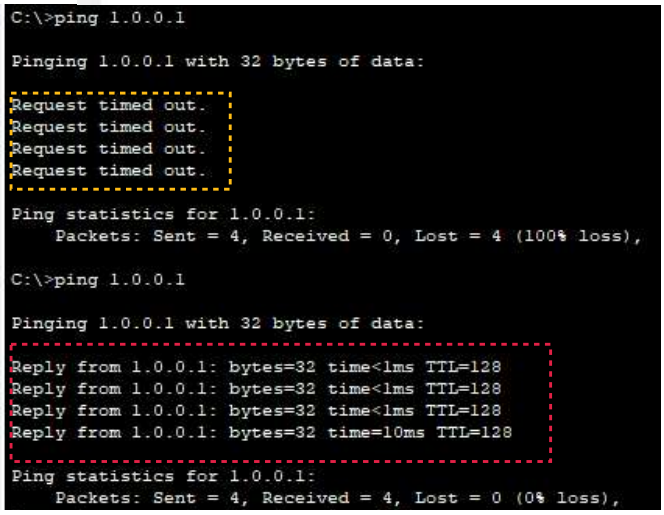
HTTP PC2 -> Server(1.0.0.1)

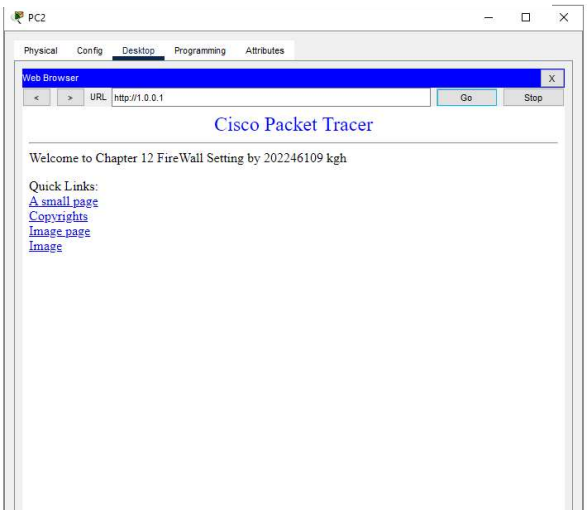
기존과 다르게 Request Timeout 메시지가 나타난다.

5. ICMP/IP 프로토콜 허용 설정하기

방화벽 규칙에서 Action 을 'Allow'로 수정 후 다시 한 번 시도해보겠습니다.

	Action	Protocol	Remote IP	Remote Wild Card	Remote Port	Local Port
1	Allow	ICMP	0.0.0.0	255.255.255.255	-	-
2	Allow	IP	0.0.0.0	255.255.255.255	-	-



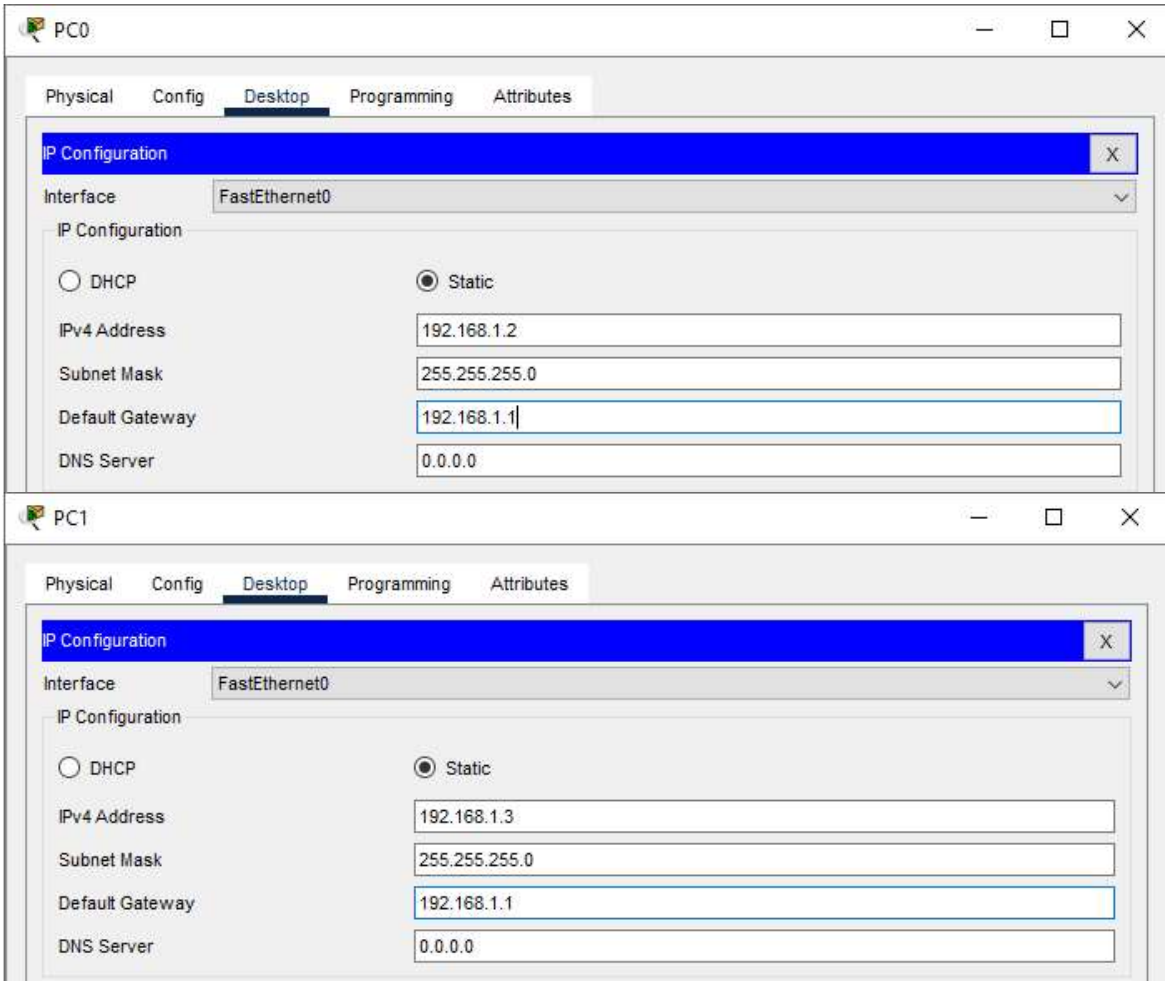


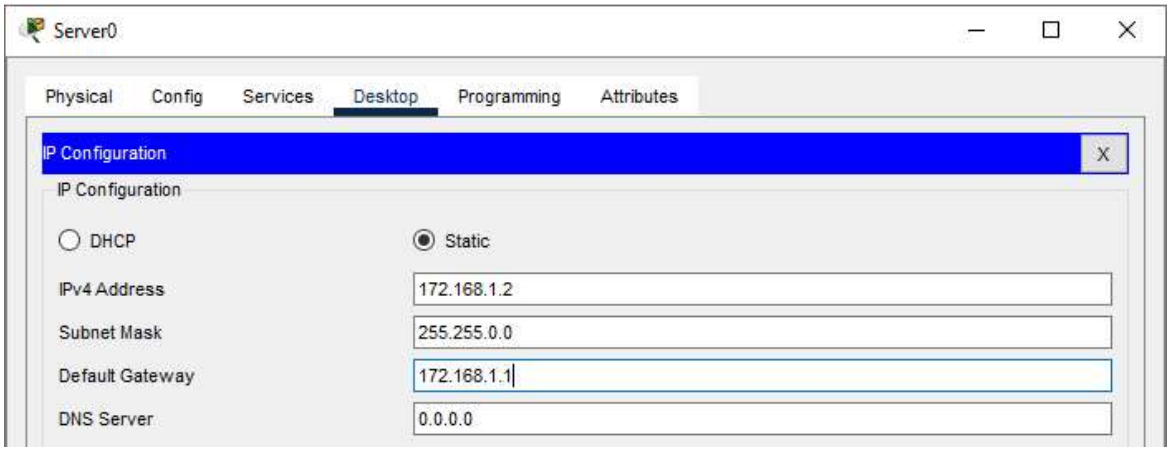
기존과 달리 통신이 정상적으로 수행되는걸 확인 했습니다.

NAT(Network Address Translation)는 내부 네트워크에서 사설 주소를 사용하다가 외부로 접근할 때 라우팅이 가능한 외부공인 주소를 할당하는 기술입니다.
Normal NAT을 설정해 내부 네트워크에 있는 클라이언트에서 외부 네트워크에 있는 서버에 접속해 보겠습니다.

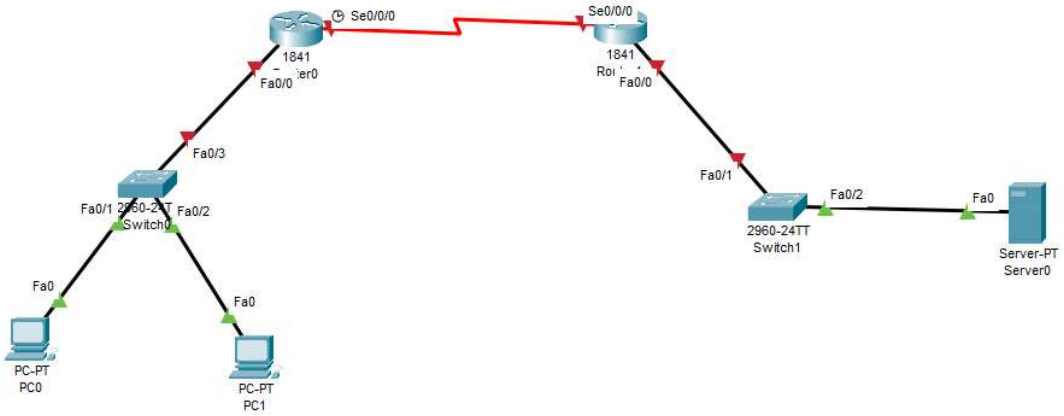
- 네트워크 구성
  - 내부 네트워크(192.168.1.x)는 스위치 1대, 라우터 1대, PC 1대로 구성
  - 외부 네트워크(172.168.1.x)는 스위치 1대, 서버 1대로 구성
  - 내부 네트워크와 외부 네트워크는 라우터를 통해 연결

1. PC/Server 설정하기



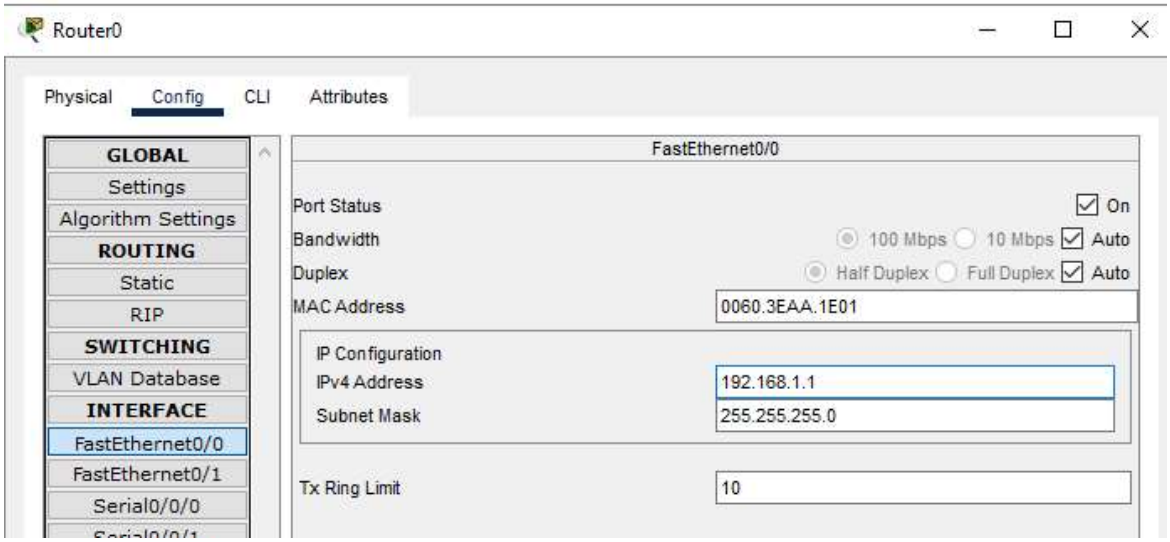


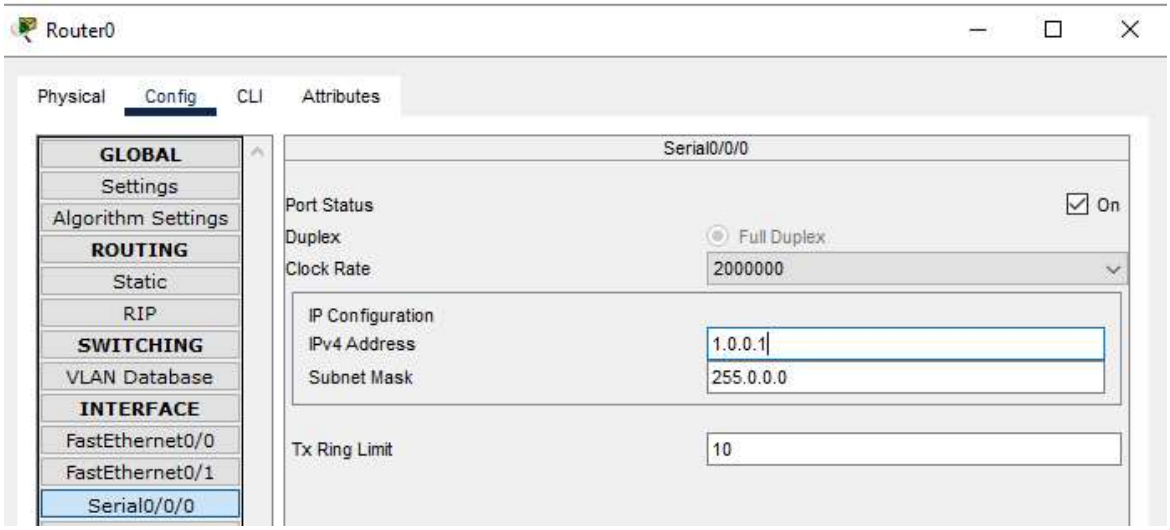
2. 시리얼 모듈 설치하기

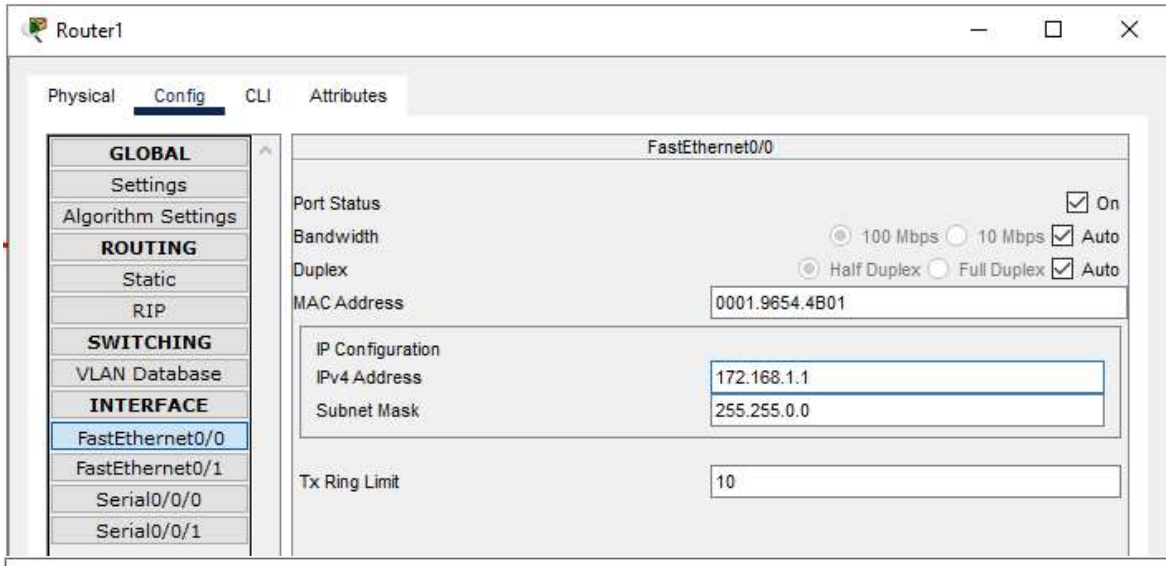


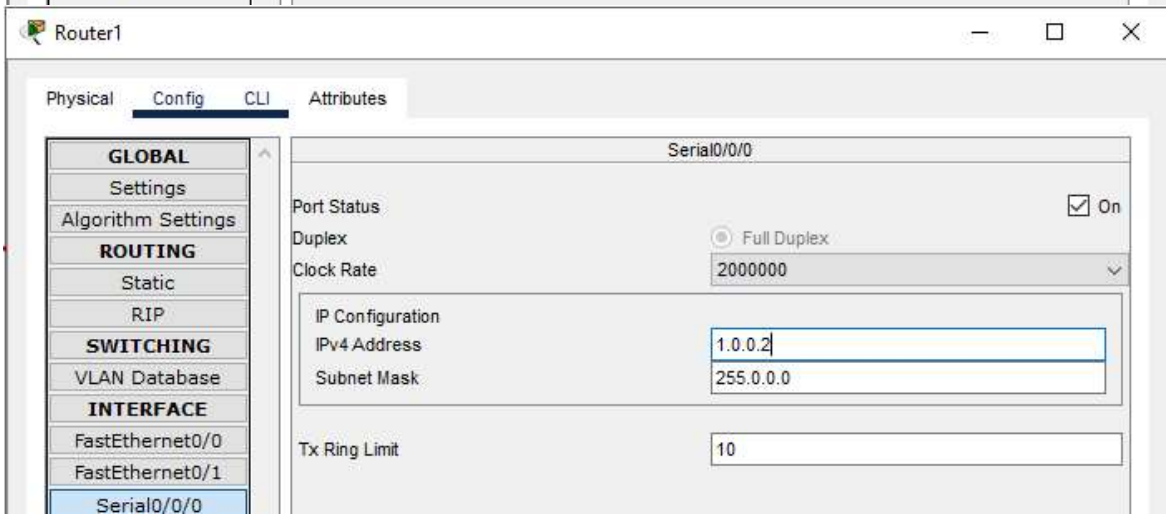
시리얼 모듈을 활성화 후 라우터간 연결을 마쳤습니다.

3. 라우팅 설정하기









장치	IP주소	서브넷 마스크	인터페이스
Router0	192.168.1.1	255.255.255.0	Fa0/0
	1.0.0.1	255.0.0.0	Se0/0/0
Router1	172.168.1.1	255.255.255.0	Fa0/0
	1.0.0.2	255.0.0.0	Se0/0/0

4. NAT 설정하기

```
Router>en
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int fa0/0
Router(config-if)#ip nat inside
Router(config-if)#ex
Router(config)#int serial 0/0/0
Router(config-if)#ip nat outside
Router(config-if)#ex
Router(config)#ip nat inside source satic 192.168.1.2 1.0.0.1
                                  ^
% Invalid input detected at '^' marker.

Router(config)#ip nat inside source static 192.168.1.2 1.0.0.1
Router(config)#ip route 0.0.0.0 0.0.0.0 serial 0/0/0
```

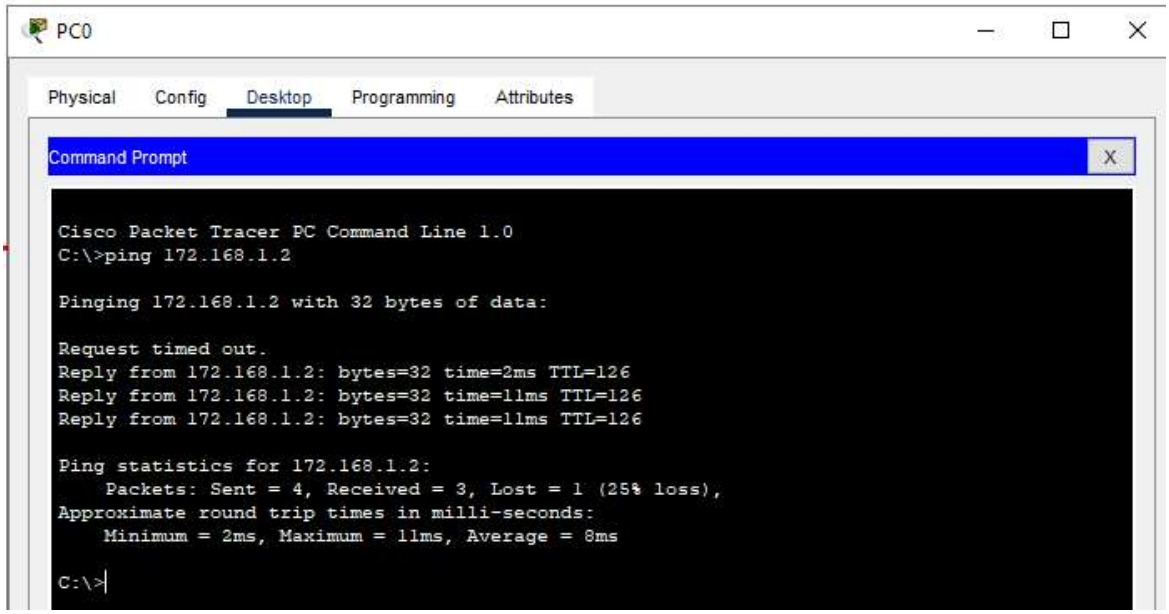
fa0/0은 내부 네트워크로 serial 0/0/0 는 외부 네트워크로 설정했습니다.

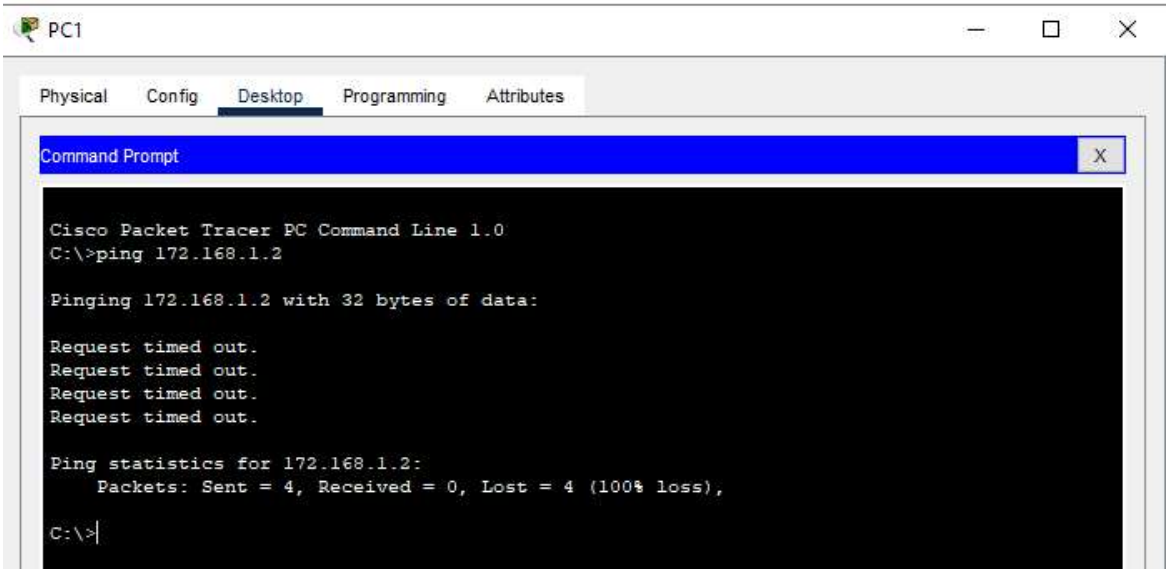
내부 네트워크의 클라이언트 PC0의 출발지 주소를 외부 공인 주소와 매칭했습니다.

5. NAT 설정 확인하기

```
Router#sh ip nat tr
Pro  Inside global     Inside local       Outside local      Outside global
---  1.0.0.1           192.168.1.2        ---                ---
```

Router0 에서 외부 공인 IP와 내부 비공인 IP가 매칭된 것을 확인했습니다





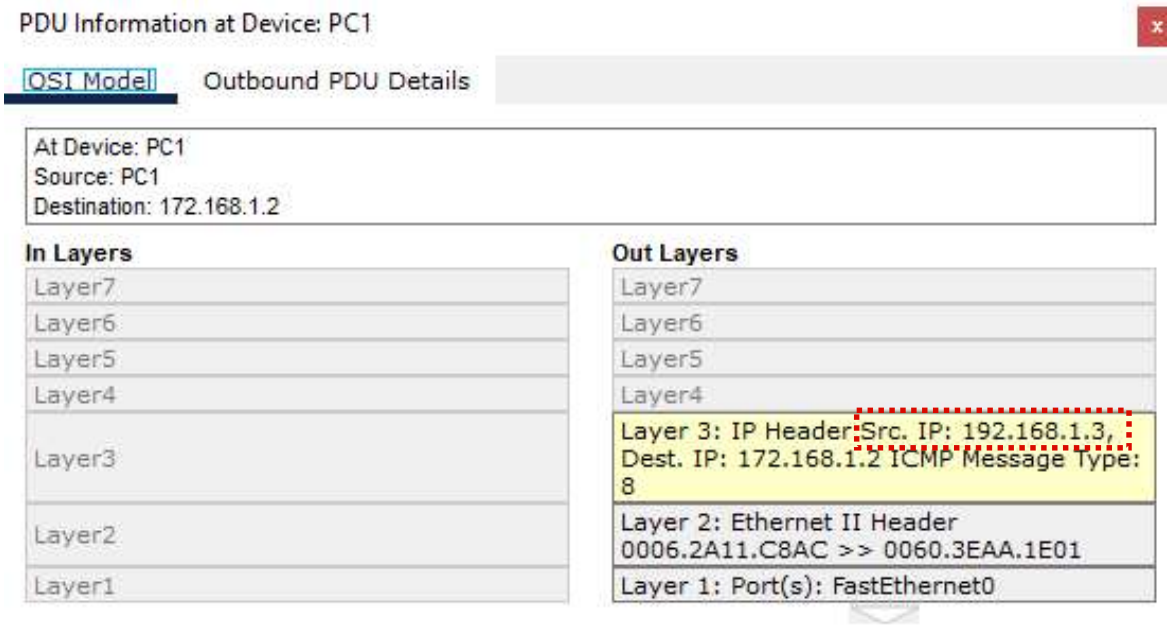
NAT 설정이 된 PC0에서 외부 네트워크에 존재하는 Server0 으로 통신은 가능하지만 아직 NAT 설정을 하지 않은 PC1 에서의 Server0 으로의 통신은 불가능 합니다.

Router0 에서 내부 네트워크의 PC1 출발지 주소를 외부 공인 주소와 매칭시키겠습니다.

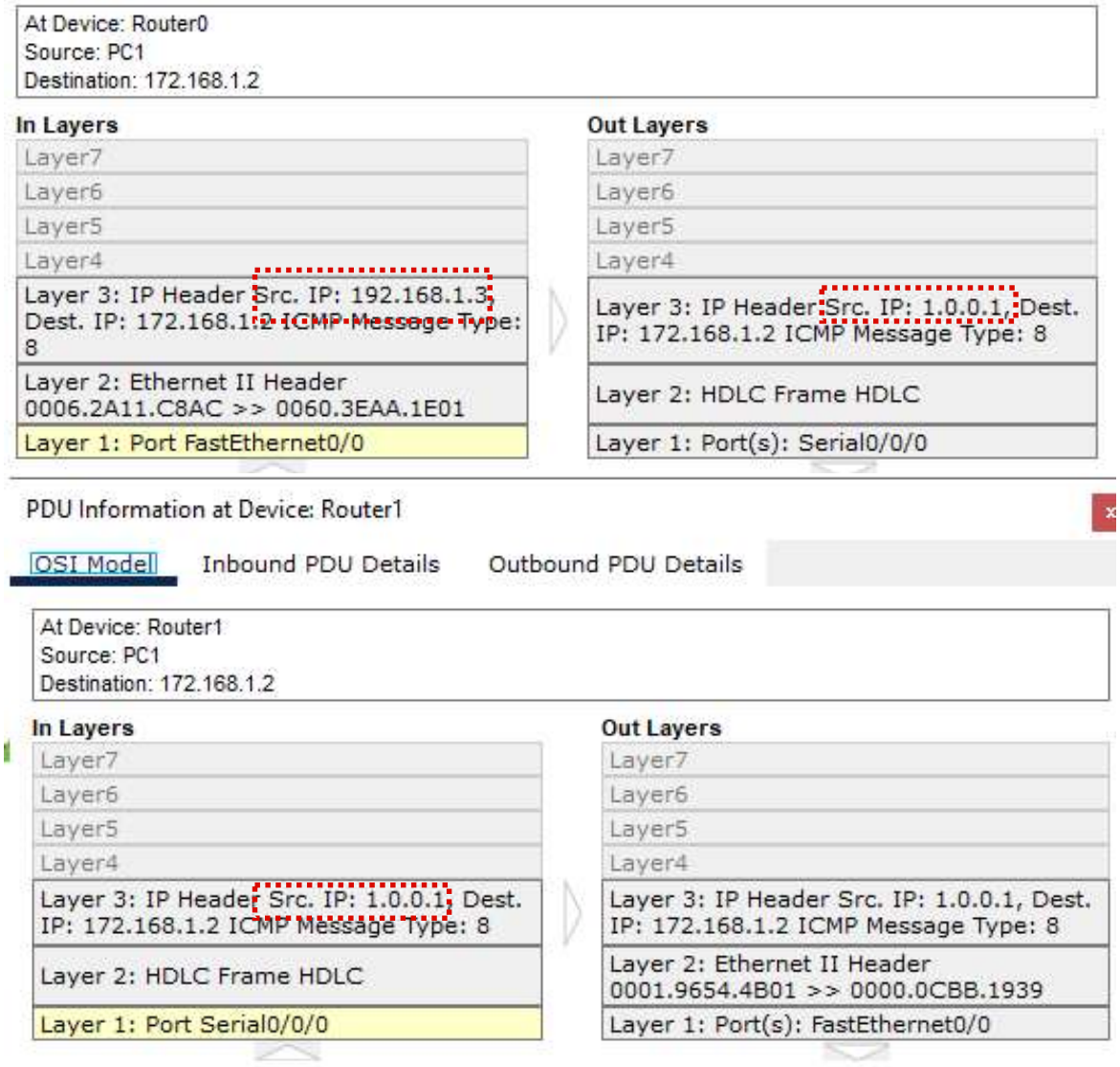
```
Router#sh ip nat tr
Pro  Inside global     Inside local       Outside local      Outside global
---  1.0.0.1           192.168.1.3        ---                ---

C:\>ping 172.168.1.2

Pinging 172.168.1.2 with 32 bytes of data:

Reply from 172.168.1.2: bytes=32 time=17ms TTL=126
Reply from 172.168.1.2: bytes=32 time=12ms TTL=126
Reply from 172.168.1.2: bytes=32 time=15ms TTL=126
Reply from 172.168.1.2: bytes=32 time=9ms TTL=126

Ping statistics for 172.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 9ms, Maximum = 17ms, Average = 13ms

C:\>tracert 172.168.1.2

Tracing route to 172.168.1.2 over a maximum of 30 hops:

  1    0 ms      0 ms      0 ms      192.168.1.1
  2    0 ms      3 ms      0 ms      1.0.0.2
  3    0 ms      6 ms      1 ms      172.168.1.2
```

PC1에서 라우터를 거쳐 외부 네트워크의 Server0으로의 통신이 수행됨을 확인했습니다.

좀 더 자세히 살펴보자면



PC1 에서 출발한 패킷의 Src IP는 PC1의 IP(192.168.1.3)이지만



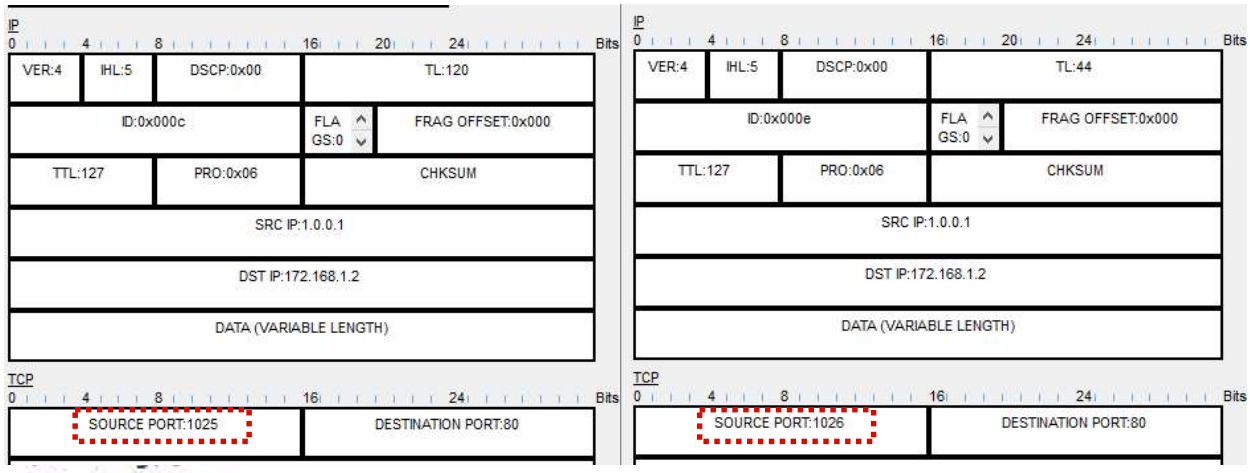
Router0 을 거쳐 Router1로 전송된 패킷의 Src IP는 Router0의 Serial IP(1.0.0.1)로 변환 되어있는걸 확인 할 수 있습니다.

PAT(Port Address Translation)는 여러 내부 IP를 출발지 포트 번호를 다르게 설정해 각 요청을 구분합니다.

1. PAT 설정

```
Router(config)#ip nat inside source list 1 interface Serial0/0/0 overload
Router(config)#access-list 1 permit 192.168.1.0 0.0.0.255
```

2. NAT 설정 확인하기




```
Router#sh ip nat tr
Pro  Inside global     Inside local       Outside local      Outside global
icmp 1.0.0.1:18        192.168.1.3:18     172.168.1.2:18     172.168.1.2:18
icmp 1.0.0.1:19        192.168.1.3:19     172.168.1.2:19     172.168.1.2:19
icmp 1.0.0.1:20        192.168.1.3:20     172.168.1.2:20     172.168.1.2:20
icmp 1.0.0.1:21        192.168.1.3:21     172.168.1.2:21     172.168.1.2:21
icmp 1.0.0.1:9         192.168.1.2:9      172.168.1.2:9      172.168.1.2:9
---  1.0.0.1           192.168.1.3        ---                ---
tcp 1.0.0.1:1025       192.168.1.2:1025   172.168.1.2:80     172.168.1.2:80
tcp 1.0.0.1:1026       192.168.1.2:1026   172.168.1.2:80     172.168.1.2:80
tcp 1.0.0.1:1027       192.168.1.2:1027   172.168.1.2:80     172.168.1.2:80
```

PC0 에서 외부 서버로 두 번의 HTTP요청을 보내고 TCP 레이어의 Source Port 항목을 비교한 결과

두 요청 모두 Src IP(1.0.0.1)는 동일하지만 포트 번호는 각각 다르게 설정된걸 확인할 수 있었습니다.