

# Network Subnet

## 개요

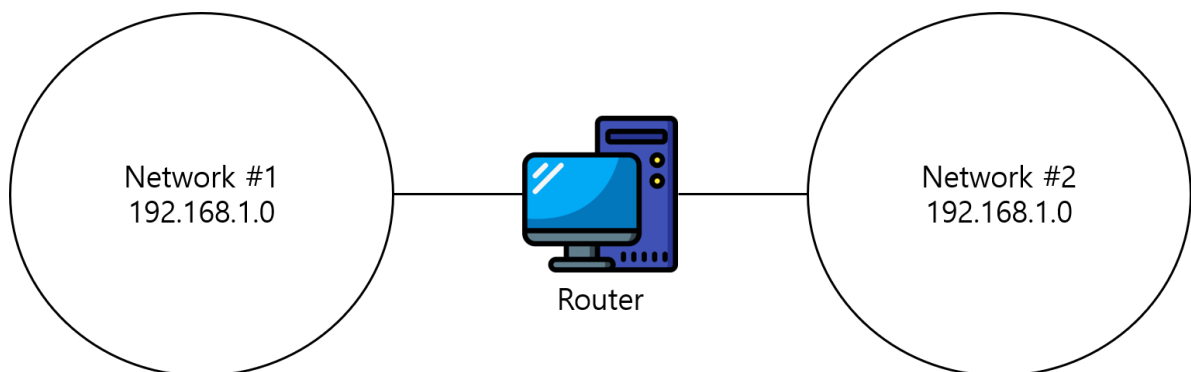
- **네트워크 서브넷 : Network Subnet**

- Subnet mask를 사용하여 여러 클래스(A,B,C)의 네트워크를 더 작은 단위로 나눈 네트워크
- 목적1) 네트워크를 더 작은 단위로 나누어 Broadcast 시 트래픽을 분산하기 위함

### Ex1) Network Address 1.0.0.0 / Host Address 1.0.0.1 ~ 1.255.255.254

- 해당 네트워크에서 할당가능한 호스트의 수 = 16,777,214개  $((256 * 256 * 256) - 2)$
- 극단적인 예이지만 이와 같은 네트워크에 수많은 호스트가 존재한다면 Broadcast(1.255.255.255)시 급격하게 트래픽이 증가되어 해당 네트워크가 과부하 될 것이다.
- 이러한 문제를 해결하기 위해 Subnet mask를 통해 네트워크를 더 작은 단위로 나누는 것이다.

### Ex2) C클래스 주소에서 200개의 Host를 2개의 Network로 나눌 때



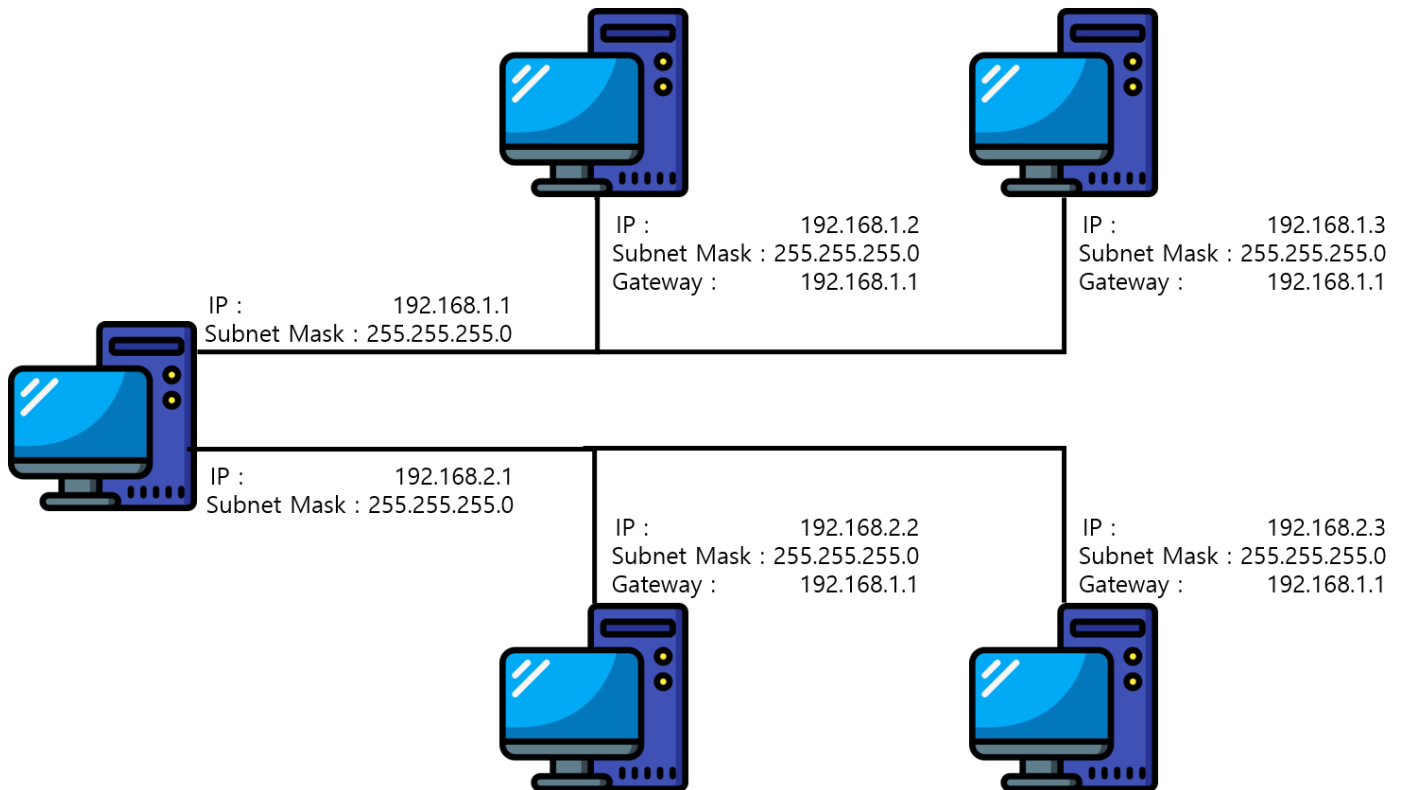
- Network #1) Network Address 192.168.1.0 / Host Address 192.168.1.1 ~ 192.168.1.99
- Network #2) Network Address 192.168.1.0 / Host Address 192.168.1.100 ~ 192.168.1.199
- 간단하게 200개의 Host를 위 두 Network와 같이 나누었을 시 Network #1의 192.168.1.1에서 Network #2의 192.168.1.100으로 패킷을 보낼 수 없는 현상이 발생한다.
- 왜냐하면 192.168.1.1은 192.168.1.100을 같은 Network(192.168.1.0) 영역에 속해 있다고 생각하여 자신의 Network #1에서만 목적지(192.168.1.100)를 찾게 되는 것이다.
- 이러한 문제를 해결하기 위해 Subnet mask를 통해 서로 다른 Network 주소를 생성하여 라우터가 패킷 포워딩을 할 수 있게 한다.

- **Subnet Mask의 역할**

- 간단하게 Subnet Mask는 Network Address와 Host Address를 구별해주는 역할을 한다.
- 예를 들어, 192.168.1.2라는 IP주소가 있을 때 Subnet Mask가 255.255.255.0 이라면 Network Address는 192.168.1이 되고 Host Address는 2가 되는 것이다.
- IP주소와 Subnet Mask 값이 서로 AND 연산하여 Network와 Host Address를 나눈다.
- 다시 말해, 특정 Network의 Subnet은 Subnet Mask를 통해서 나뉘어 지는 것이다.

## Subnet

- 이제 Subnet mask를 이용하여 두 개의 네트워크 영역으로 나눈 뒤 각 네트워크 영역 간의 통신이 어떻게 이루어지는지 설명할 것이다.



- 위 그림에서 두 Network 영역 192.168.1.0과 192.168.2.0이 존재하는 것을 확인할 수 있다.
- 만약, 192.168.1.2라는 Host가 192.168.2.2라는 Host로 패킷을 보낼 때 아래와 같은 프로세싱이 일어난다.

### 1. 송신측 IP의 Network Address 구하기 (Subnet Mask와 IP를 AND 연산)

: IP = 192.168.1.2<sub>(10)</sub> = 11000000.10101000.00000001.00000010<sub>(2)</sub>  
: Subnet Mask = 255.255.255.0<sub>(10)</sub> = 11111111.11111111.11111111.00000000<sub>(2)</sub>  
: Network Addr = 192.168.1.0<sub>(10)</sub> = 11000000.10101000.00000001.00000000<sub>(2)</sub>

### 2. 목적지 IP의 Network Address 구하기 (Subnet Mask와 IP를 AND 연산)

: IP = 192.168.2.2<sub>(10)</sub> = 11000000.10101000.00000010.00000010<sub>(2)</sub>  
: Subnet Mask = 255.255.255.0<sub>(10)</sub> = 11111111.11111111.11111111.00000000<sub>(2)</sub>  
: Network Addr = 192.168.2.0<sub>(10)</sub> = 11000000.10101000.00000010.00000000<sub>(2)</sub>

### 3. 송신측, 목적지측 간의 Network Address가 다른 것을 확인하였으므로 송신측은 자신의 Network 영역에 목적지측 Host가 존재하지 않는다고 판단

### 4. 송신측은 최종 목적지로 패킷을 전달하기 위해 Routing Table을 조회하여 Gateway로 패킷을 전달

### 5. Gateway와 인접한 노드에도 Host가 존재하지 않는다면 Routing Table을 조회하여 다른 Network로 포워딩, 존재한다면 목적지 Host로 패킷 포워딩

## Routing Table

```
C:\Users\#INLAB>route print
```

### 인터페이스 목록

```
19...98 83 89 1f db 72 .....Realtek PCIe GbE Family Controller
18...ca ff 28 27 fa 11 .....Microsoft Wi-Fi Direct Virtual Adapter
3...da ff 28 27 fa 11 .....Microsoft Wi-Fi Direct Virtual Adapter #2
6...c8 ff 28 27 fa 11 .....Qualcomm Atheros QCA61x4 Wireless Network Adapter
8...c8 ff 28 27 fa 12 .....Bluetooth Device (Personal Area Network)
1.....Software Loopback Interface 1
```

### IPv4 경로 테이블

#### 활성 경로:

네트워크 대상	네트워크 마스크	게이트웨이	인터페이스	메트릭
0.0.0.0	0.0.0.0	192.168.1.1	192.168.1.44	50
127.0.0.0	255.0.0.0	연결됨	127.0.0.1	331
127.0.0.1	255.255.255.255	연결됨	127.0.0.1	331
127.255.255.255	255.255.255.255	연결됨	127.0.0.1	331
192.168.1.0	255.255.255.0	연결됨	192.168.1.44	306
192.168.1.44	255.255.255.255	연결됨	192.168.1.44	306
192.168.1.255	255.255.255.255	연결됨	192.168.1.44	306
224.0.0.0	240.0.0.0	연결됨	127.0.0.1	331
224.0.0.0	240.0.0.0	연결됨	192.168.1.44	306
255.255.255.255	255.255.255.255	연결됨	127.0.0.1	331
255.255.255.255	255.255.255.255	연결됨	192.168.1.44	306

- 위 사진은 Window Dos창에서 Routing Table을 조회한 것이다. (명령어 : route print)
- 위 사진에서 확인할 수 있듯이 해당 컴퓨터는 192.168.1.0 이라는 Network에 소속되어 있는 것을 확인할 수 있다.
- 또한, 192.168.1.0에 소속된 목적지로 패킷을 보내는 것이 아니라면 192.168.1.1이라는 게이트웨이로 패킷이 전달되는 것을 확인할 수 있다.

## 참고문헌

- <https://blog.naver.com/danhan09/150025465055>
- <https://velog.io/@hidaehyunlee/%EC%84%9C%EB%B8%8C%EB%84%B7%ED%8C%85subnetting%EC%9C%BC%EB%A1%9C-%EB%84%A4%ED%81%AC%EC%9B%8C%ED%81%AC%EB%A5%BC-%ED%9A%A8%EC%9C%A8%EC%A0%81%EC%9C%BC%EB%A1%9C-%EA%B4%80%EB%A6%AC%ED%95%98%EC%9E%90>
- <https://m.blog.naver.com/PostView.nhn?blogId=csubcad&logNo=120168732795&proxyReferer=https:%2F%2Fwww.google.com%2F>
- <https://limkydev.tistory.com/168>