IP Sub-Networking Mini-Howto

Robert Hart, *hartr@interweft.com.au* <mailto:hartr@interweft.com.au> v1.0, 31 March 1997

번역: 정재영 *dannybox@nownuri.net* <mailto:dannybox@nownuri.net>

이 문서는 IP 네트워크의 서브넷을 사용하는 이유와 방법에 대해 설명한다. 즉 몇개의 네트워크에 연결된 상황에서 제대로 동작하는 A,B,C 클래스 네트워크 번호를 사용하는 것이다.

목 차

제 1 절 Copyright

제 2 절 Introduction

사용 가능한 IP 네트워크 번호가 점점 희귀해짐에 따라, 이 부족한 자원을 효율적 으로 이용하는 것이 중요해졌다. 이 문서는 하나의 IP 네트워크 번호를 여러개의 다른 네트워크에서 사용할 수 있도록 하나의 IP 네

트워크 번호를 어떻게 나누는 가에 대해 설명하고 있다. 이 문서는 주로 클래스 C IP 네트워크 번호에 대해 설명할 것이지만, 그 규칙은 A, B 클래스에도 동일하게 적용된다.

2.1 다른 유용한 정보들

IP 번호에 대한 자세하고 기본적인 정보를 알려주는 몇 가지 문서가 있다. 저자가 읽어보길 권하는 문서들은

- The Linux Network Administrators Guide <*sunsite.unc.edu/LDP/LDP/nag/nag.html* `<http://sunsite.unc.edu/LDP/LDP/nag/nag.html>`>.
- The Linux System Administration Guide <*www.db.erau.edu/SAG/* `<http://www.db.erau.edu/SAG/>`>.
- TCP/IP Network Administration by Craig Hunt, published by O'Reilly and Associates <*www.ora.com/catalog/tcp/noframes.html* `<http://www.ora.com/catalog/tcp/noframes.html>`>.

제 3 절 IP 번호의 해부

서브네트워킹의 즐거움을 만끽하기 전에 먼저 IP 번호에 대한 기본적인 지식을 가져야 한다.

3.1 IP 번호는 호스트가 아닌 인터페이스에 속한 것이다.

무엇보다도 먼저 잘못된 이해의 기본적인 원인을 분명히 해야 한다. IP 번호는 호스트에 할당된 것이 아니다. IP 번호는 호스트의 네트워크 인터페이스에 할당된 것이다. 많은(대부분이 아닐지라도) IP 네트워크상에 존재하는 컴퓨터가 하나의 네트워크 인터페이스를 소유하지만(그리고 그 결과 하나의 IP 번호를 가지지만), 이러한 경우만 있는 것은 아니다. 컴퓨터와 다른 디바이스들은 몇 개의(많지는 않더라도) 네트워크 인터페이스를 가진다. 그리고 이 각각의 네트워크 인터페이스는 각각의 IP 번호를 가진다. 그러므로 6개의 활성화된 인터페이스를 가진 디바이스는 6개의 IP 번호를 가진다. 그리고 각각의 IP 번호는 연결된 각각의 네트워크에 대한 인터페이스에 할당된다. 그 이유는 IP 네트워크를 볼 때 명확해진다.

그럼에도 불구하고, 대부분의 사람은 IP 번호를 말할 때 대개 호스트 주소를 언급한다. 여기서 기억해야 할 것은 이것은 호스트에 존재하는 특별한 인터페이스에 대한 IP 번호를 줄여서 지칭한다는 것이다. 인터넷에 존재하는 많은 디바이스 (다수가 아니더라도)는 하나의 인터페이스만을 가지므로 하나의 IP 번호를 가진다.

3.2 "Dotted Quads"로 표현된 IP 번호

현재의 IP 번호를 구현한 것에 따르면(IPv4), IP 번호는 4(8비트)개의 바이트로 구성되어 총 32비트를 사용하여 정보를 표시한다. 그 결과 꽤 넓은 영역을 표현할 수 있다.(10진수로 표현 되더라도). 그리고 사람들이 읽기 쉽도록 IP 번호는 대개 "dotted quad" 형식으로 표현된다. IP 번호

192.168.1.24 는 점(.)으로 구별된 4개의 숫자(10진수)의 예이다.

4 숫자 각각은 8비트 바이트를 10진수로 표현한 것이고, 각각의 4개 숫자는 0부터 255까지 표현할 수 있다.(여기서 기억해야 할 것은 0도 하나의 숫자이므로 총 256개라는 것이다)

또한 호스트 IP 번호의 일부는 호스트가 위치한 네트워크를 표시하고, 남은 숫자 는 호스트(네트워크 인터페이스) 그 자체를 표시한다. 정확하게 몇 개의 비트 가 네트워크 ID에 사용되고, 네트워크에 있는 호스트를 구별하는데 몇 개의 비트 가 사용되는가 하는 것은 네트워크의 '클래스'에 결정된다.

3.3 IP 번호의 클래스들

IP 번호에는 3개의 클래스가 있다.(역자 주: 실제로는 멀티캐스트를 위한 클래스 D가 있으므로 4개이다. 클래스 D는 224.0.0.0-240.0.0.0의 주소를 가진다)

- 클래스 A IP 번호는 네트워크를 구별하는데 가장 왼쪽의 8개 비트(dotted quad의 가장 왼쪽)를 사용하고, 나머지 24비트(나머지 세 개 dotted quad)를 사용하여 네트워크의 인터페이스를 구별한다. 클래스 A 주소의 가장 왼쪽 비트는 항상 0이다 - 즉 첫 번째 dotted quad는 십진수로 0에서 127의 값을 가진다. 그러므로 최대 128개의 클래스 A 네트워크 번호가 가능하고, 각각 33,554,430개의 사용 가능한 인터페이스를 가진다. 그러나 네트워크 0.0.0.0(기본 라우트(default route)) 와 127.0.0.0 (루프 백 네트워크)는 특별한 의미를 가지기 때문에 네트워크를 구별하는데는 사용되지 않는다. 그러므로 126개의 클래스 A 네트워크 번호만을 사용할 수 있다.
- 클래스 B IP 번호는 네트워크를 구별하는데 왼쪽 16개의 비트(왼쪽의 2개 dotted quad)를 사용하고 나머지 16비트를(나머지 2개의 dotted quad) 사용하여 호스트 인터페이스를 구별한다. 클래스 B 주소는 가장 왼쪽 바이트의 2개의 왼쪽 비트를 항상 10으로 가진다.(역자 주: 2진수로 표현하면 10xxxxxx.xxxxxxxx .xxxxxxxx.xxxxxxxx). 그러므로 나머지 14비트를 사용하여 32767개의 클래스 B 네트워크를 사용할 수 있다. 그러므로 클래스 B 네트워크는 dotted quad의 첫 번째 바이트에 대해 128에서 191까지의 영역을 가지고, 각각의 네트워크는 최대 32,766개의 이용 가능한 인터페이스를 가진다.
- 클래스 C IP 번호는 처음의 24비트를 사용하여 네트워크를 구별하고, 나머지 8비트로 호스트 인터페이스를 구별한다. 클래스 C 주소는 왼쪽 3개 비트가 항상 110이거나 dotted quad로 192에서 255의 영역을 가진다. 그러므로 사용 가능한 클래스 C 네트워크 번호는 4,194,303개이고 각각은 254개의 인터페이스를 가진다. (첫번째 바이트가 223보다 큰 네트워크는 이미 예약이 되어 있어 사용할 수 없다) 요약하면

네트워크 클래스	사용 가능한 첫 번째 바이트의 범위(십진수)
A	1 에서 126
B	128 에서 191
C	192 에서 254

’unconnected’ 네트워크를 위해 미리 예약된 특별한 주소가 있다. 즉 IP를 사용하는 네트워크이지만 인터넷에 연결되어 있지 않는 것들이다. 이 주소들은

- 한개의 클래스 A 네트워크 10.0.0.0
- 16개의 클래스 B 네트워크 172.16.0.0 - 172.31.0.0
- 256개의 클래스 C 네트워크 192.168.0.0 - 192.168.255.0 이 문서에서는 ’실제’ 네트워크나 호스트와의 혼동을 피하기위해 이 순서들을 사용할 것이다.

3.4 네트워크 번호, 인터페이스 주소와 브로드캐스트 주소들

IP 번호는 3가지 의미를 가질 수 있다.

- IP 네트워크의 주소(공통된 전달 매개체를 사용하는 IP 디바이스의 그룹 - 같은 이더넷 세그먼트에 존재하는 모든 것들). 네트워크 번호는 항상 주소 영역의 인터페이스(호스트) 비트를 0로 설정한다. (네트워크가 서브넷으로 나누어지지 않는한)
- IP 네트워크에 대한 브로드캐스트 주소(주소는 IP 네트워크의 모든 디바이스에 동시에 ’이야기’하기 위해 사용된다.). 네트워크에 대한 브로드캐스트 주소는 인터페이스(호스트) 비트를 1로 설정한다.(역시 네트워크가 서브넷으로 나누어 지지 않는 한)

- 인터페이스의 주소(호스트에 있는 이더넷 카드나 PPP 인터페이스, 라우터 , 프린터 서버 등등). 이 주소들은 모두 0 이거나 1인 경우를 제외한 임의의 값들을 가질 수 있다. 만일 호스트 비트가 모두 0이면 그 주소는 네트워크 주소가 되고, 호스트 비트가 모두 1이면 그 주소는 브로드캐스트 주소가 된다.

요약하면

```
 클래스 A 네트워크에 대해
 (1바이트의 네트워크 주소 영역과 3바이트의 호스트 주소 영역)
   10.0.0.0 은 주소의 모든 호스트 비트가 0이므로 클래스 A 네트워크 번호이다.
   10.0.1.0 은 이 네트워크의 한 호스트의 주소이다.
   10.255.255.255 는 주소의 모든 호스트 비트가 1이므로 이 네트워크의
   브로드캐스트 주소이다.

클래스 B 네트워크에 대해
 (2바이트의 네트워크 주소 영역과 2바이트의 호스트 주소 영역)
   172.17.0.0 은 클래스 B 네트워크 번호이다.
   172.17.0.1 은 이 네트워크의 한 호스트의 주소이다.
   172.17.255.255 는 네트워크 브로드캐스트 주소이다.

클래스 C 네트워크에 대해
 (3바이트의 네트워크 주소 영역과 1바이트의 호스트 주소 영역)
   192.168.3.0 은 클래스 C 네트워크 번호이다.
   192.168.3.42 는 이 네트워크의 한 호스트의 번호이다.
   192.168.3.255 는 네트워크 브로드캐스트 주소이다.
```

현재 호스트에 할당이 가능한 대부분의 IP 네트워크 번호는 클래스 C 주소이다.

3.5 네트워크 마스크(network mask)

네트워크 마스크는 '서브넷 마스크'로 불려야 더 정확하다. 그러나 대개 네트워크 마스크라 부른다.

여기서 우리가 가장 관심을 갖고 있는 내용인 IP 주소가 IP 네트워크 영역에서 어떻게 지역적으로 해석되는지는 네트마스크에 의존하며, 이것이 서브네트워킹의 형태를 결정한다. 기본적인 네트워크 마스크는 모든 네트워크 비트가 '1'이고 모든 호스트 비트가 '0'이다. 이것들은 세가지 클래스의 네트워크에 대한 기본적인 네트워크 마스크이다:-

- 클래스 A 네트워크 마스크: 255.0.0.0
- 클래스 B 네트워크 마스크: 255.255.0.0
- 클래스 C 네트워크 마스크: 255.255.255.0 네트워크 마스크에 대해 기억해야 할 두가지 중요한 사항이 있다:-
- 네트워크 마스크는 오직 지역 IP 번호의 지역적인 해석에만 영향을 미친다. (여기서 '지역'이란 특별한 네트워크 부분을 의미한다)
- 네트워크 마스크는 IP 번호가 아니다 - 이것은 지역 IP 번호가 어떻게 지역적 으로 해석되는지 결정할 때 사용된다.

제 4 절 서브넷이란 무엇인가?

서브넷이란 하나의 IP 네트워크 주소를 지역적으로 나누어 이 하나의 네트워크 IP 주소가 실제로 여러개의 서로 연결된 지역 네트워크로 사용할 수 있도록 하는 방법이다.

여기서 중요한 단어는 '지역적' 라는 것이다. 머신 바깥의 세계와 서브넷으로 나뉜 IP 네트워크에 포함되는 실제 네트워크는 어떤 사항이 바뀌더라도 여전히 하나의 IP 네트워크이다. 기억해야 할 것은 하나의 IP 네트워크는 오직 하나의 네트워크에서만 사용될 수 있다는 것이다.

제 5 절 왜 서브넷을 사용하나?

서브네트워킹을 사용하는 이유를 알려면 초창기 IP 규약을 제정했던 시절로 돌아가야 한다. 이때는 클래스 A 네트워크 번호를 가지면서 몇 백만개의 호스트를 가질 수 있었던 몇 개의 사이트만이 존재했다.

만일 거대한 사이트에 있는 모든 IP 컴퓨터들이 서로 같은 네트워크에 연결되길 원한다면 엄청난 트래픽과 관리상의 문제점이 발생할 것은 명백했다: 네트워크를 관리하는 것은 거대한 짐승을 기르는 것과 같은 악몽일 것이고, 네트워크는 아마도 (거의 확실히) 자신의 트래픽 때문에 붕괴될 것이었다. 만일 서브네트워킹을 사용하면: 클래스 A 네트워크 번호는 몇개의 네트워크로 구분될 것이다. 물론 나뉘어진 각각의 네트워크는 쉽게 관리할 수 있을 것이다.

이것은 작고 관리하기 쉬운(상대적으로) 네트워크가 만들어지는 것을 의미한다. - 분명히 서로 다른 네트워크 기술을 이용하면서. 그러나 같은 물리적인 네트워크에서는 이더넷, 토큰 링, FDDI, ATM등을 섞어서 사용할 수 없다. 서브네트워킹을 사용하는 다른 이유는:-

- 실제 사이트 구성은 물리적인 인프라스트럭쳐들이 어떻게 연결되었는 가에 따라 제한(케이블 길이 등)을 가질 수 있다. 서브네트워킹은 하나의 IP 번호를 사용하여 IP 환경내에서 이것들을 가능하도록 한다. 실제로 이런 일들은 영구적으로 연결된 고객들에게 지역 네트워크의 정적 IP 번호(static IP number)를 주고 싶어하는 ISP(Internet Service Provider)들에 의해 매우 흔히 이루어지고 있다.
- 네트워크 트래픽은 심각한 속도 저하를 일으킬 정도로 충분히 많다. 네트워크를 서브넷으로 나눔으로써 한 네트워크 지역에 한정된 트래픽을 그 지역에만 한정되도록 할 수 있다. 이 경우 전체적인 트래픽을 줄여 실제 네트워크 대역폭을 늘이지 않고도 네트워크 연결속도를 증가시킬 수 있다.
- 보안상의 요구로 인해 다른 클래스에 있는 사용자들은 같은 네트워크를 공유할 수 없다 - 네트워크를 잘 아는 사람은 항상 다른 네트워크의 트패픽을 가로챌 수 있다. 서브네트워킹은 마케팅 부서가 R & D 부서의 네트워크 트래픽을 훔쳐보지 못하게 한다.(혹은 학생들이 관리자 네트워크를 훔쳐보지 못하도록 하거나)!
- 서로 호환되지 않는 네트워킹 기술을 사용하는 장비를 이용하면서, 이들 사이에 상호 연결이 필요한다.

제 6 절 IP 네트워크 번호를 서브넷으로 나누는 방법

여러분의 IP 네트워크 번호를 서브넷으로 나눌 필요가 있다고 결정했다면 이제 어떻게 해야할까? 아래의 내용은 해야 할 일들을 자세히 설명하고 있다.

- 실제적인 연결을 설정한다.(네트워크 선과 라우터와 같은 네트워크 연결들)
- 서브넷에 연결될 디바이스의 갯수로부터 서브넷이 얼마나 커야 할지를 결정한다. 다시 말하면 각각의 구역에 대해 얼마나 많은 IP 번호가 필요한 지 결정한다.
- . 적절한 네트워크 마스크와 네트워크 주소를 계산한다.
- 네트워크의 각각의 인터페이스에 고유의 IP 주소와 적절한 네트워크 마스크를 할당한다.
- . 라우터에서 라우트를 설정하고 네트워크 디바이스에서 적절한 게이트웨이와 라우트 그리고/혹은 기본 라우트를 설정한다.

- 시스템을 테스트해보고 문제점이 발견되면 고친다. 그리고 한숨 돌린다. ^^ 앞으로 설명할 예제를 위해 하나의 클래스 C 네트워크 번호 192.168.1.0을 서브넷으로 나누는 것으로 가정한다.

이 IP 네트워크는 최대 254개의 인터페이스(호스트)를 가질 수 있고, 반드시 네트워크 자신을 표현할 네트워크 번호(192.168.1.0) 그리고 브로드캐스트 주소 (192.168.1.255)를 가진다.

6.1 실제적인 연결 설정하기

여러분은 여러분이 관리해야할 네트워크의 실제 구조와 잘 부합되도록 모든 디바이스들이 상호연결 되도록 하기위해 정확한 케이블 인프라스트럭쳐 (infrastructure)를 설치해야 할 것이다.

또한 여러분은 다양한 부분(라우터, 미디어 변환기 등등)을 상호 연결할수 있는 매커니즘이 필요할 지도 모른다. 분명히 이것에 대한 자세한 설명은 불가능하다. 도움이 필요하다면 이러한 서비스를 제공하는 주위의 네트워크 디자인/설치 전문가를 만나보는 것이 좋다. 또한 유즈넷 뉴스 그룹을 보면 무료로 제공되는 많은 조언들을 얻을 수 있다. (*comp.os.linux.networking* `<http://comp.os.linux.networking>`과 같은 뉴스 그룹)

6.2 서브넷 크기 정하기

여러분이 만든 서브넷의 갯수와 '잃어버린' IP 번호의 개수 사이에는 타협이 필요하다. 모든 각 IP 네트워크에서는 2개의 주소를 인터페이스(호스트) 주소로 사용할 수 없게 된다 - 네트워크 IP 번호 그 자체와 브로드캐스트 주소. 여러분이 서브넷을 구성하면 각각의 서브넷은 네트워크 자신을 표시할 고유한 IP 네트워크 번호와 브로드캐스트 번호를 필요로 하며, 이 번호들은 서브넷의 IP 네트워크 에 의해 지정된 범위내에서 유효한 것이어야 한다.

그러므로 IP 네트워크를 두개의 서브넷으로 분리하는 것은 2개의 네트워크 주소와 두개의 브로드캐스트가 필요하다는 것이므로, 이는 곳 '사용불가능한' 인터페이스 (호스트) 주소가 증가한다는 것을 의미한다. 만일 4개의 서브 네트워크를 만든다면 이 숫자는 8로 증가하게 된다.

사실 가장 작은 서브넷도 4가지 IP 번호로 구성되어 있다.

- 두개의 사용가능한 IP 인터페이스 번호- 한개는 그 네트워크의 라우터 인터 페이스를 위한 것이고 나머지 한개는 그 네트워크의 싱글 호스트를 위한 것
- 한개의 네트워크 번호
- 한개의 브로드캐스트 주소

왜 이렇게 사용하지 못하는 IP 숫자를 늘여가면서 작은 네트워크를 만드는 가 하는 것은 또 다른 문제이다. 네트워크에 오직 하나의 호스트가 있다면 어떠한 네트워크 통신도 다른 네트워크로 나가야 한다. 그러나 앞으로 보게될 예제는 서브넷에 적용되는 수익 체감의 법칙을 잘 보여줄 것이다.

이론적으로 IP 네트워크 번호는 똑같은 크기를(서브넷으로 나눈 후 서브 네트워크들을 결합할 수도 있다.) 갖는 2^n개의 서브넷으로 나눌 수 있다.(이때 n은 여러분의 IP 네트워크 번호의 호스트 비트의 갯수보다 하나 적은 숫자이다)

그러므로 자신의 네트워크를 디자인하는데 현실적이어야 한다 - 여러분은 관리, 실제적인 장비, 보안상의 제한등을 고려하여 가능하면 로컬 네트워크의 숫자를 줄이는 방향으로 디자인을 해야한다.

6.3 서브넷 마스크와 네트워크 번호 계산하기

IP 네트워크를 서브넷으로 나누는 마술을 발휘하는 것은 네트워크 마스크 이다. 서브넷으로 나누어지지 않은 IP 네트워크 번호에 대한 네트워크 마스크는 단순히 네트워크의 번호의 모든 '네트워크 비트'를 '1'로 설정하고 모든 호스트 비트를 '0'으로 설정한 것이다.

그러므로 IP 번호의 세가지 클래스에 대한 표준 네트워크 마스크는 다음과 같다.

- 클래스 A (8개의 네트워크 비트) : 255.0.0.0
- 클래스 B (16개의 네트워크 비트): 255.255.0.0
- 클래스 C (24개의 네트워크 비트): 255.255.255.0

서브넷이 동작하는 방법은 하나 혹은 그 이상의 사용가능한 호스트 비트를 빌려와서 인터페이스가 빌려온 비트를 네트워크의 일부로 해석하도록 하는 것이다. 그러므로 네트워크 번호를 두개의 서브넷으로 나누기 위해서는 첫번째 호스트의 네트워크 마스크 비트에서 적절한 비트를 '1'로 지정하여 한개의 호스트 비트를 빌려온다.

클래스 C 주소에 대해 네트워크 마스크는 11111111.11111111.11111111.10000000 또는 255.255.255.128가 된다. 우리가 살펴보고 있는 클래스 C 네트워크 번호인 192.168.1.0에 대해 여러분들은 몇개의 서브넷 옵션을 지정할 수 있다.

```
No of      No of
subnets    Hosts/net    netmask
2            126        255.255.255.128 (11111111.11111111.11111111.10000000)
4             62        255.255.255.192 (11111111.11111111.11111111.11000000)
8             30        255.255.255.224 (11111111.11111111.11111111.11100000)
16            14        255.255.255.240 (11111111.11111111.11111111.11110000)
32             6        255.255.255.248 (11111111.11111111.11111111.11111000)
64             2        255.255.255.252 (11111111.11111111.11111111.11111100)
```

이론적으로, 위에서처럼 서브넷을 만들때 반드시 네트워크 마스크 비트를 가장 중요한 호스트 비트에서 가장 덜 중요한 호스트 비트순으로 더할 필요는 없다. 그러나 만일 여러분이 이 방법을 사용하지 않는다면 만들어진 IP 번호는 매우 이상한 순서가 될 것이다. 이 경우 2진수에 익숙치 않는 우리들 로써는 IP 번호가 어떤 서브넷에 속해있는지 알기가 매우 어려워진다. (반면 컴퓨터는 여러분이 어떤 방법을 사용했는지 상관없이 잘 이해할 것이다) 적절한 네트마스크를 결정했다면 이제 여러분은 네트워크 주소와 브로드캐스트 주소와 각각의 네트워크의 IP 번호의 범위를 알아야 한다. 다시 한번 말하지만 이 예제에서는 클래스 C IP 네트워크 번호만을 고려하고 최종 호스트 번호만을 보여주도록 하겠다.

```
Netmask         Subnets Network B'cast  MinIP   MaxIP   Hosts  Total Hosts
---------------------------------------------------------------------------
    128             2        0     127       1     126    126
                           128     255     129     254    126      252
  192             4        0      63       1      62     62
                            64     127      65     126     62
                           128     191     129     190     62
                           192     255     193     254     62      248
  224             8        0      31       1      30     30
                            32      63      33      62     30
                            64      95      65      94     30
                            96     127      97     126     30
                           128     159     129     158     30
                           160     191     161     190     30
                           192     223     193     222     30
                           224     255     225     254     30      240
```

위에서 보이는 것처멈 이들 숫자에는 명백한 순서가 존재하기 때문에 확인하기가 매우 쉽다. 서브네트워킹의 아래쪽은 서브넷의 숫자가 증가함에 따라 사용가능한 호스트 주소의 전체 숫자가 감소한다는 사실을 보여준다. 이 정보로부터 이제 여러분들은 호스트와 네트워크 IP 번호 그리고 네트워크마스크 를 할당할 수 있게 되었다.

제 7 절 라우팅

만일 여러분이 두개의 네트워크 인터페이스를 가진 리눅스 PC를 두개(혹은 그 이상) 의 서브넷을 라우트하기 위해 사용한다면 커널에서 IP 포워딩(IP Forwarding)이 가능하도록 해야한다.

```
cat /proc/ksyms | grep ip_forward
```

이런 내용을 보게 될 것이다.

```
00141364 ip_forward_Rf71ac834
```

만일 이런 메시지가 나타나지 않는다면 여러분의 커널은 IP 포워딩이 되도록 설정 되지 않은 것이므로 커널을 재컴파일 한 후 새로운 커널을 설치해야 한다. 이 예제를 위해 클래스 C IP 네트워크 번호 192.168.1.0을 4개의 서브넷으로 나누는 것으로 가정한다.(각각은 62개의 사용가능한 인터페이스/호스트 IP 번호를 갖게 된다). 그러나 이들 서브넷중 두개는 좀 더 큰 하나의 네트워크 를 구성하도록 결합될 것이므로 결과적으로 세개의 실제적인 네트워크가 존재하게 된다. 즉

네트워크	브로드캐스트	네트마스크	호스트 갯수
192.168.1.0	192.168.1.63	255.255.255.192	62
192.168.1.64	192.168.1.127	255.255.255.192	62
192.168.1.128	192.168.1.255	255.255.255.128	124 (see note)

참고: 마지막 네트워크가 124개의 사용가능한 네트워크 주소만을 가지는 이유는(네트워크 마스크로부터 예상할 수 있는 126개가 아닌) 이것이 두개의 서브넷에 대한 '수퍼 넷'이기 때문이다. 다른 두개의 네트워크에 있는 호스트들은 192.168.1.192를 '존재하지 않는' 서브넷의 네트워크 주소로 해석할 것이다. 이와 유사하게 192.168.1.191 역시 '존재하지 않는' 서브넷의 브로드캐스트 주소로 해석할 것이다.

그러므로 만일 여러분이 192.168.1.191나 192를 네트워크의 호스트 주소로 사용한다면 이 더 작은 두개의 네트워크에 존재하는 머신들은 이들과 통신할 수 없을 것이다. 이것은 서브넷의 중요한 한가지 사실을 설명해 준다. 즉 사용가능한 주소는 그 주소 공간의 가장 작은 서브넷에 의해 결정된다는 것이다.

7.1 라우팅 테이블

이제 리눅스가 동작하는 컴퓨터가 이 네트워크의 라우터로 동작한다고 가정 해보자. 이 컴퓨터는 로컬 랜에 대해 3개의 네트워크 인터페이스를 가지고 있고 아마도 네번째 인터페이스를 통해 인터넷에 연결되어 있을 것이다. (이 인터페이스가 '기본 라우트'일 것이다.) 이제 리눅스 컴퓨터가 각각의 서브넷에서 가장 낮은 IP 주소를 인터페이스에 사용한다고 가정한다. 이제 네트워크 인터페이스는 다음과 같이 설정되었을 것이다.

인터페이스	IP 주소	넷마스크
eth0	192.168.1.1	255.255.255.192
eth1	192.168.1.65	255.255.255.192
eth2	192.168.1.129	255.255.255.128

이것에 대한 라우팅은 다음과 같을 것이다.

목적지 주소	게이트웨이	Genmask	인터페이스
192.168.1.0	0.0.0.0	255.255.255.192	eth0
192.168.1.64	0.0.0.0	255.255.255.192	eth1
192.168.1.128	0.0.0.0	255.255.255.128	eth2

각각의 서브넷에 대해 호스트는 자신의 IP 번호와 네트워크 마스크로 설정 되어 있을 것이다.(특별한 네트워크에 적합하도록) 각각의 호스트는 네트워크 의 인터페이스에 리눅스 PC의 IP 주소를 지정함으로써, 리눅스 컴퓨터를 자신의 게이트웨이/라우터로 선언한다. Robert Hart Melbourne, Australia March 1997.