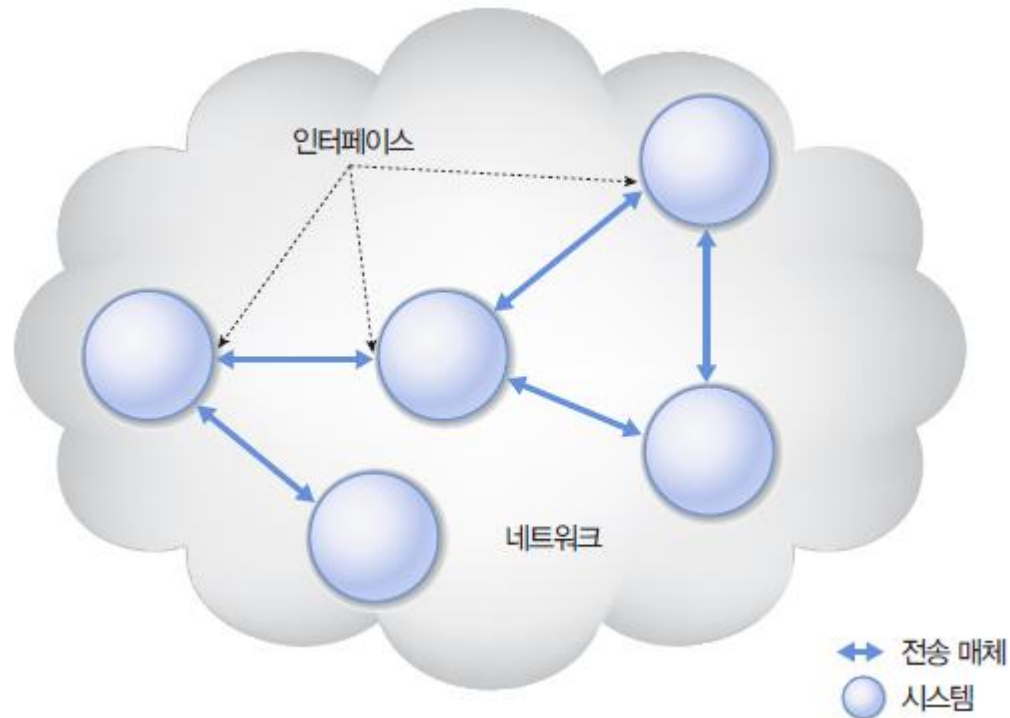


네트워크의 기초 용어와 기능 / 네트워크 보안 개요

01_네트워크 관련 기초 용어

❖ 네트워크 기초 용어

- 시스템, 인터페이스, 전송 매체, 프로토콜, 네트워크, 인터넷, 표준화
- 네트워크: 전송 매체로 서로 연결된 시스템의 모음



01_네트워크 관련 기초 용어

❖ 네트워크 기초 용어

■ 시스템

- 내부 규칙에 따라 능동적으로 동작하는 대상
- 예: 컴퓨터, 자동차, 커피 자판기, 마이크로 프로세서, 운영체제, 프로세스

■ 인터페이스

- 시스템과 전송 매체의 연결 지점에 대한 규격
- 예: RS-232C, USB

■ 전송매체

- 시스템끼리 데이터를 전달하기 위한 물리적인 전송 수단

■ 프로토콜

- 전송 매체를 통해 데이터를 교환할 때의 임의의 통신 규칙

01_네트워크 관련 기초 용어

■ 네트워크

- 프로토콜을 사용하여 데이터를 교환하는 시스템의 집합을 통칭

■ 인터넷

- 전세계의 네트워크가 유기적으로 연결되어 동작하는 통합 네트워크
- 공통 기능: IP(Internet Protocol)

■ 표준화

- 서로 다른 시스템이 상호 연동해 동작하기 위한 통일된 연동 형식

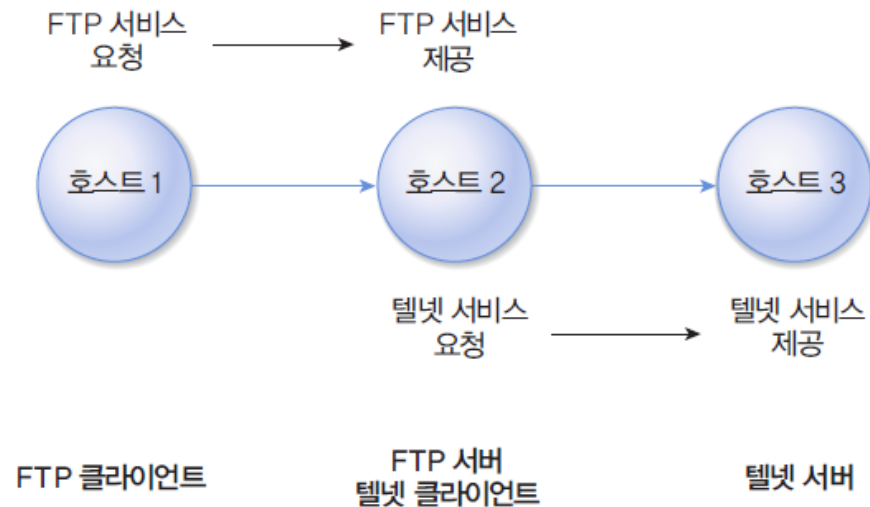
01_네트워크 관련 기초 용어

❖ 시스템 기초 용어

■ 시스템의 구분

- 노드: 인터넷에 연결된 시스템의 가장 일반적인 용어
- 호스트: 컴퓨팅 기능이 있는 시스템
- 클라이언트: 서비스를 요청하는 시스템
- 서버: 서비스를 제공하는 시스템

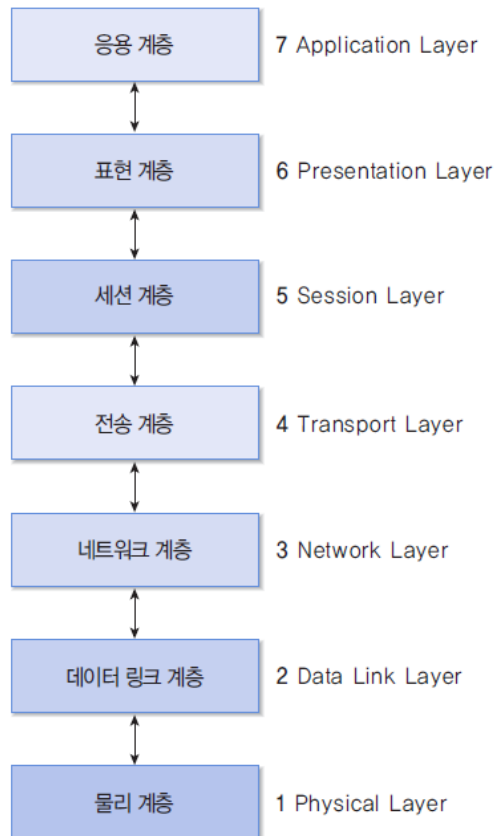
■ 클라이언트와 서버



02_네트워크의 기능

❖ 계층 모델

■ ISO의 OSI(Open System Interconnection) 7계층 모델



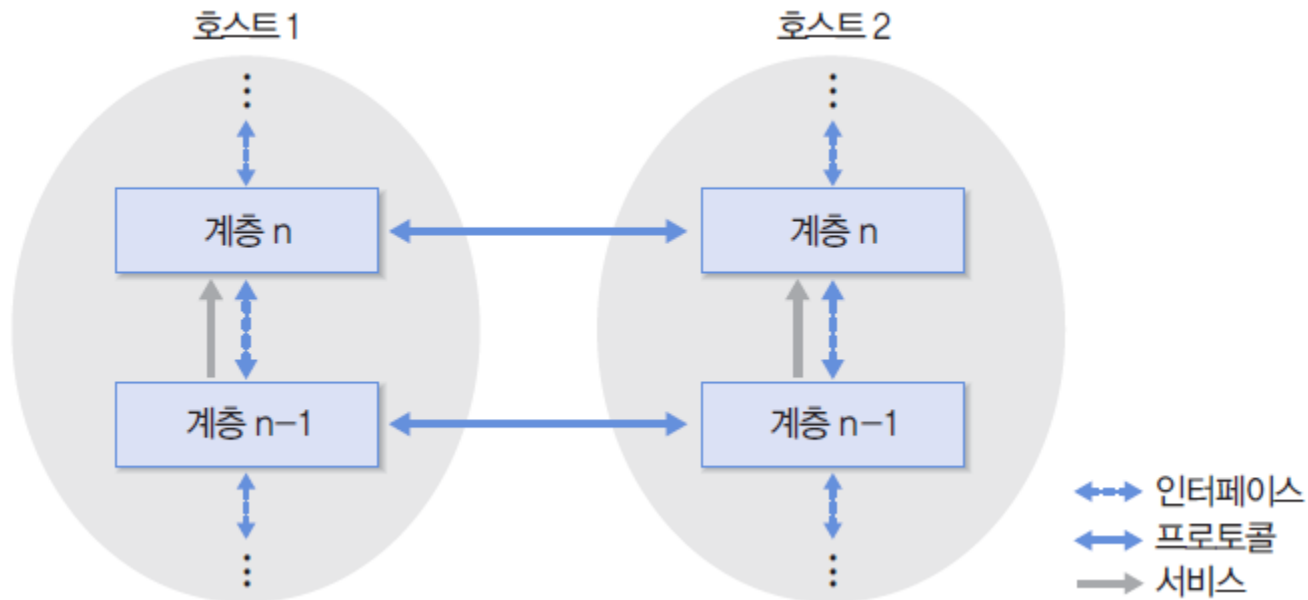
■ OSI 7계층 모델의 계층별 기능

- 물리 계층: 물리적으로 데이터를 전송하는 역할을 수행
- 데이터 링크 계층: 물리적 전송 오류를 해결 (오류 감지 / 재전송 기능)
- 네트워크 계층: 올바른 전송 경로를 선택 (혼잡 제어 포함)
- 전송 계층: 송수신 프로세스 사이의 연결 기능을 지원
- 세션 계층: 대화 개념을 지원하는 상위의 논리적 연결을 지원
- 표현 계층: 데이터의 표현 방법
- 압축: 전송되는 데이터의 양
- 암호화: 전송되는 데이터의 의미
- 응용 계층: 다양한 응용 환경을 지원

02_네트워크의 기능

■ 프로토콜과 인터페이스

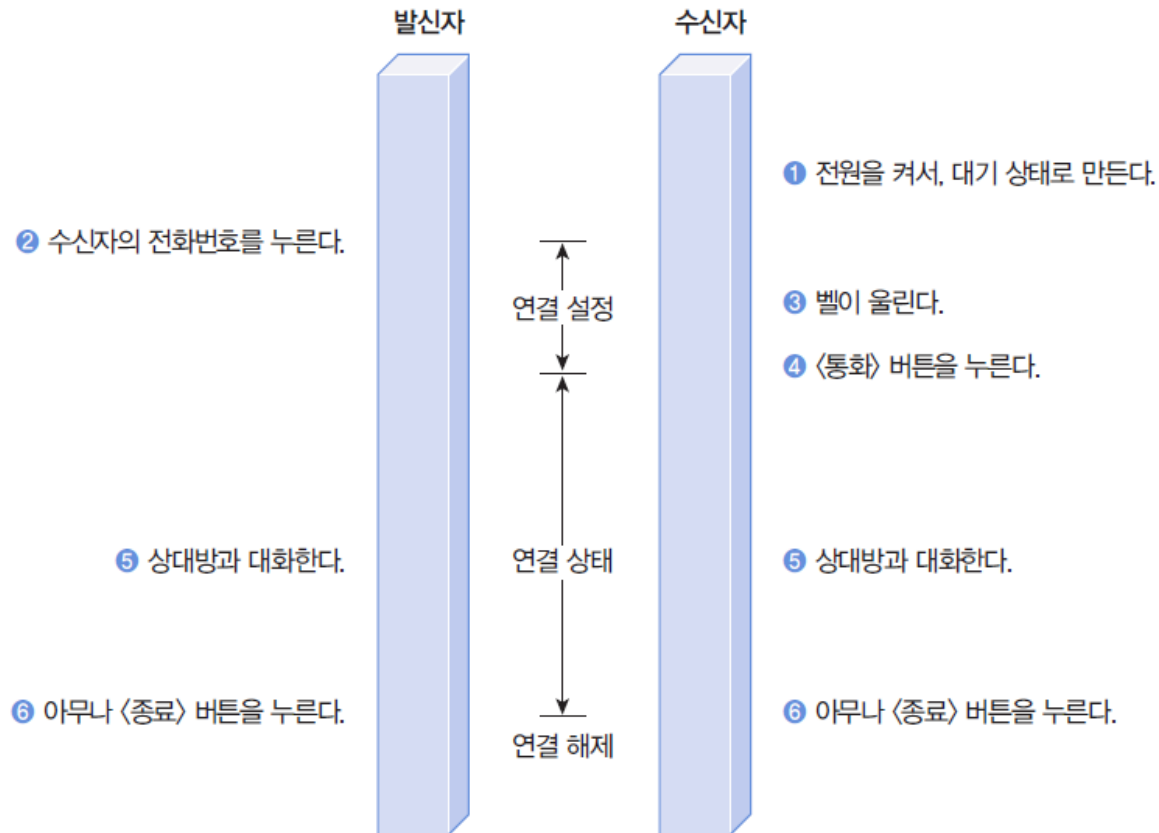
- 프로토콜: 서로 다른 호스트에 위치한 동일 계층끼리의 통신 규칙
- 인터페이스: 같은 호스트에 위치한 상하위 계층 사이의 규칙
- 서비스: 하위 계층이 상위 계층에 제공하는 인터페이스



02_네트워크의 기능

❖ 프로토콜

■ 프로토콜 예



02_네트워크의 기능

❖ 프로토콜에 대한 이해

- 본래 의미는 외교에서 의례 또는 의정서
- 톰 마릴이 '컴퓨터와 컴퓨터 사이에서 메시지를 전달하는 과정'이라 정의



03_네트워크의 역사 _ 유선통신의 시작

■ 네트워크의 정의

- 지역적으로 분산된 위치에서 컴퓨터 시스템 간에 데이터 통신을 하기 위한 하드웨어 및 소프트웨어의 집합

■ 모스와 전신기

- 1800년경 볼타가 최초로 전지를 발명
- 이후 전선을 통해 신호를 보내는 방법을 연구하기 시작했고, 모스가 처음 실질적인 성과를 냄.
- 1832년 알파벳 문자에 점과 대시를 사용하여 모스 부호를 발명
- 1843년 워싱턴에서 볼티모어까지 전신기 선을 가설
- 1844년 첫 번째 공식 메시지는 '하나임이 행하신 일이 어찌 그리 크뇨(민수기 23:23)'

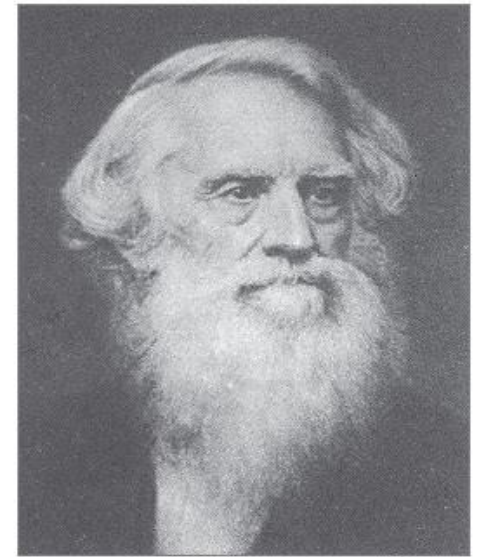


그림 1-1 사무엘 모스

03_네트워크의 역사 _ 유선통신의 시작

■ 벨의 전화기

- 1876년 알렉산더 그레이엄 벨이 전화기를 개발
- 1876년 3월 미국 특허청에 전화를 특허로 등록
- 1877년 1월 30일 상자 모양의 첫 전화기가 등장(일대일 통신만 가능)
- 1878년 1월 28일 코네티컷의 뉴헤이븐에서 처음으로 교환기가 설치되어 사용자의 전화가 중앙의 교환수를 거쳐 연결됨.
- 우리나라는 1896년 궁내부 전화가 전화기의 효시
- 1902년에는 서울과 인천 사이에 전화가 개설

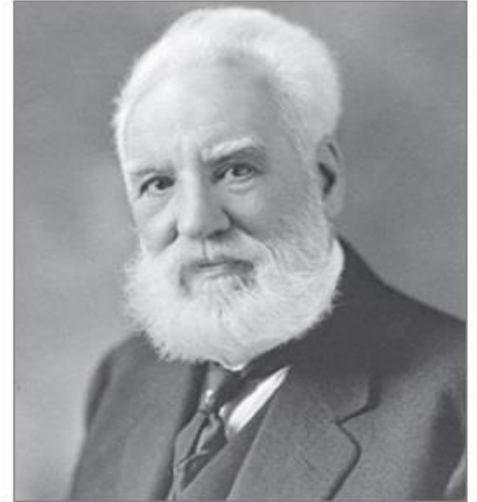


그림 1-3 알렉산더 그레이엄 벨

03_네트워크의 역사 _ 무선통신의 시작

■ 전자기파의 발견

- 1864년 제임스 클러크 맥스웰이 전자기파가 대기중으로 전파된다고 처음 예측
- 전자기장의 기초 방정식인 맥스웰방정식(전자기방정식)을 도출하여 전자기파의 존재를 증명
- 전자기파의 전파 속도가 광속도와 같고, 횡파라는 사실을 밝힘으로써 빛의 전자기파설에 대한 기초를 세움.



그림 1-4 제임스 클러크 맥스웰

03_네트워크의 역사 _ 무선통신의 시작

■ 전파 이용의 시작

- 1888년 하인리히 루돌프 헤르츠가 실험을 통해 라디오파를 주고받음으로써 전파의 존재가 실제로 입증됨.
- 굴리엘모 마르코니는 전자기파를 실제 통신에 이용할 수 있는 형태로 만들
- 마르코니는 무선 전신을 발전시킨 공로로 1909년 노벨 물리학상을 수상
- 1922년 영국방송공사(BBC)가 세계 최초로 음성 뉴스를 무선 방송으로 전송



그림 1-5 하인리히 루돌프 헤르츠



그림 1-6 굴리엘모 마르코니

03_네트워크의 역사 _ 컴퓨터통신의 시작

■ 모뎀의 개발과 네트워크의 시작

- 벨 텔레폰 연구소의 조지 스티비츠가 전화 교환 회로를 '산술 기기'로 발전시킨 '모델-K' 기기를 개발했고, 후에 CNC로 발전
- 스티비치는 1940년 뉴욕의 CNC와 전화선으로 연결해 데이터를 입력하는 과정을 시연했는데, 이는 네트워크 컴퓨팅의 효시로 기록됨.
- 1958년 벨 연구소에서 최초의 상업용 모뎀인 데이터폰을 개발
- 미국 국방부는 군사 작전 수행을 위한 고성능 컴퓨터를 개발하려는 목적으로 1965년 세계 최초의 컴퓨터 네트워크 개발에 착수
- 이때 개발을 시작한 네트워크는 1969년 오늘날 인터넷의 모태가 된 사상 최초의 대단위 컴퓨터 네트워크인 ARPANET의 탄생으로 이어짐.

03_네트워크의 역사 _ 컴퓨터통신의 시작

■ 장거리 컴퓨터 통신과 인터넷의 시작

- 1965년 ARPA는 MIT 링컨 연구소의 TX-2와 캘리포니아 산타모니카 SDC의 Q-32 컴퓨터와 전화선으로 직접 통신하는 장거리 데이터 통신을 최초로 시도
- 프로토콜 : 컴퓨터와 컴퓨터 사이에서 메시지를 전달하는 과정(톰 마릴)
- 1967년 ARPA는 ACM에서 각 호스트를 IMP라는 특정 컴퓨터에 연결하고, IMP를 서로 연결하는 ARPANET을 제안(현재의 라우터와 개념이 유사)
- 1969년 네 개의 노드(UCLA, USCB, SRI, UU)를 네트워크로 구성하고 NCP라는 프로토콜을 호스트 간 통신에 사용
- 1971년 레이 톰린슨이 전자 메일 프로그램을 발명

03_네트워크의 역사 _ 컴퓨터통신의 시작

■ 장거리 컴퓨터 통신과 인터넷의 시작

- 1972년 빈트 서프와 로버트 칸이 게이트웨이를 개발
- 1973년 빈트 서프는 로버트 칸과 함께 TCP/IP 프로토콜과 인터넷 구조를 설계

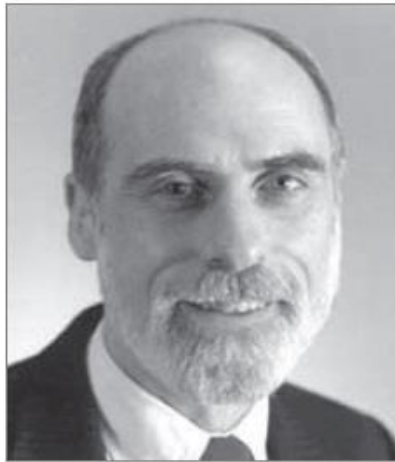


그림 1-7 빈트 서프



그림 1-8 로버트 칸

- 호스트 컴퓨터와 터미널로 구성된 네트워크는 IBM의 SNA 망이 최초
- 1974년 제록스가 이더넷을 개발(오늘날의 클라이언트-서버 구조로 전환)

03_네트워크의 역사 _ 컴퓨터통신의 시작

■ 장거리 컴퓨터 통신과 인터넷의 시작

- 1979년 유즈넷 탄생
- 1981년 유닉스 운영체제에 TCP/IP가 포함되어 배포되었고, TCP/IP가 ARPANET의 공식 프로토콜이 됨.
- 1983년 군사용 MILNET과 군사용이 아닌 ARPANET으로 분리
- 1983년 존 포스텔이 도메인이름 시스템 개발
- 1984년 DNS가 구성되어 네트워크가 폭발적으로 확장됨.
- 1990년 ARPANET이 해체되고 NSFNET이 만들어짐.

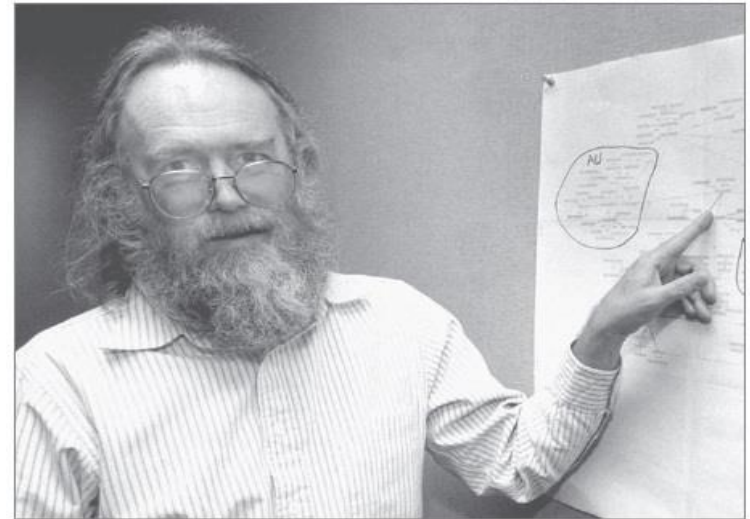


그림 1-9 존 포스텔

03_네트워크의 역사 _ 컴퓨터통신의 시작

■ 장거리 컴퓨터 통신과 인터넷의 시작

- 1989년 3월 버너스-리가 웹 개념을 제안
- 1990년 동료 로버트 카이유와 개정된 개념을 소개
 - 서로 다른 컴퓨터끼리 정보를 공유하고 서로 링크하여 찾기 쉬운 하이퍼텍스트 형태의 서비스를 도입하자는 것

03_네트워크의 역사 _ 컴퓨터통신의 시작

■ 국내 인터넷의 역사

- 1982년 서울대학교와 KIET(전자통신연구소의 전신)가 TCP/IP로 SDN 시작
- 1983년 미국으로 UUCP 다이얼 업(Dial-Up) 연결
- 1984년 유럽으로 X.25를 이용한 UUCP 연결
- 1987년 교육 연구 전산망 추진 위원회 구성
- 1988년 연구 전산망 기본 계획 확정, 교육망을 BITNET과 연결
- 1990년 HANA/SDN이 56Kbps로 인터넷에 연결
- 1991년 연구 전산망이 56Kbps로 인터넷에 연결
- 1993년 HANA/SDN이 56Kbps에서 256Kbps로 확충
- 1994년 한국통신, 데이콤에서 인터넷 상용 서비스 시작
- 1995년 INET, 나우콤에서 인터넷 상용 서비스 시작

03_네트워크의 역사 _ 컴퓨터통신의 시작

■ 국내 인터넷의 역사

- 1995년 초고속 정보통신망 구축 사업 시작
- 1996년 7천 대 이상의 호스트 컴퓨터가 연결됨
- 1997년 한국인터넷협회 설립
- 1998년 초고속 정보통신망 구축 사업 1단계 완료
- 2000년 초고속망 구축 기술로 각종 서비스가 이루어짐(하나로, 두루넷, 드림라인, 신비로 등)
- 2001년 초고속 광전송망 구축(155Mbps~40Gbps)
- 2002년 서울대, 한국전자통신연구원ETRI 등이 참여한 IPv6 활성화를 위한 프로젝트 시작
- 2004년 한국 인터넷 이용자 수 3천만 명 돌파
- 2005년 한국 IPv6 주소 보유율 세계 3위로 평가
- 2012년 한국 인터넷 속도 세계 1위로 평가

04_네트워크 주소의 표현

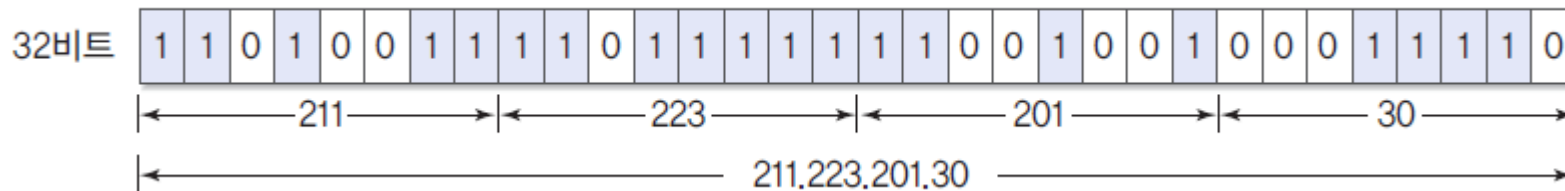
❖ 주소와 이름

■ IP 주소

- IPv4 프로토콜에서 사용
- 32 비트 크기의 주소 체계
- IPv6에서는 128 비트 주소 체계로 확장
- 예: 211.223.201.30

IP주소란?

-> TCP/IP 프로토콜을 사용하는 모든 네트워크 장비를 구분해주는 주소



- ex) 192.168.0.1
1100 0000.1010 1000.0000
0000.0000 0001



1. 다음 10진수를 2진수, 8진수, 16진수로 바꿔보아라.

$$892_{(10)} \Rightarrow$$

2. 2진수 $1000101.1011_{(2)}$ 를 10진수로 변환하여라.

$$1000101.1011_{(2)} \Rightarrow$$

3. 16진수 $A91_{(16)}$ 를 10진수, 8진수, 2진수로 변환하여라.

$$A91_{(16)} \Rightarrow$$

10진수-2진수 변환

- 정수부분과 소수부분으로 나누어 변환
- 정수부분은 2로 나누고, 소수부분은 2를 곱한다.
- 10진수 75.6875를 2진수로 변환

2		75	나머지	→	2진수
2		37	...	1	→ 1
2		18	...	1	→ 11
2		9	...	0	→ 011
2		4	...	1	→ 1011
2		2	...	0	→ 01011
2		1	...	0	→ 001011
		0	...	1	→ 1001011
					몫

2진수	←	정수	소수
		0.	6875
		X	2
0.1	←	1.	3750
		X	2
0.10	←	0.	7500
		X	2
0.101	←	1.	5000
		X	2
0.1011	←	1.	0

곱셈결과 정수를 적는다.

소수부분이 0이 될 때까지 계산한다.

$$75.6875_{(10)} = 1001011.1011_{(2)}$$

04_네트워크 주소의 표현

- IP 주소는 네트워크주소와 호스트주소로 이루어져있다.
- 네트워크 주소는 한 인터페이스의 제공된 주소이고 호스트주소는 해당 인터페이스에서 객체를 구분하는 주소

ex) 인터페이스1의 주소 192.168.0.1
192.168.0.1 ~ 192.168.0.255

인터페이스2의 주소 192.168.200.1
192.168.200.1 ~ 192.168.200.255



04_네트워크 주소의 표현

❖ IP Class 개념

- IP주소를 보고 네트워크와 호스트주소를 구분할 때 사용
- IP Address는 5가지의 class로 나뉘집니다. (네트워크의 호스트 수로 구분)

· 5개의 Class로 이루어진 IP주소

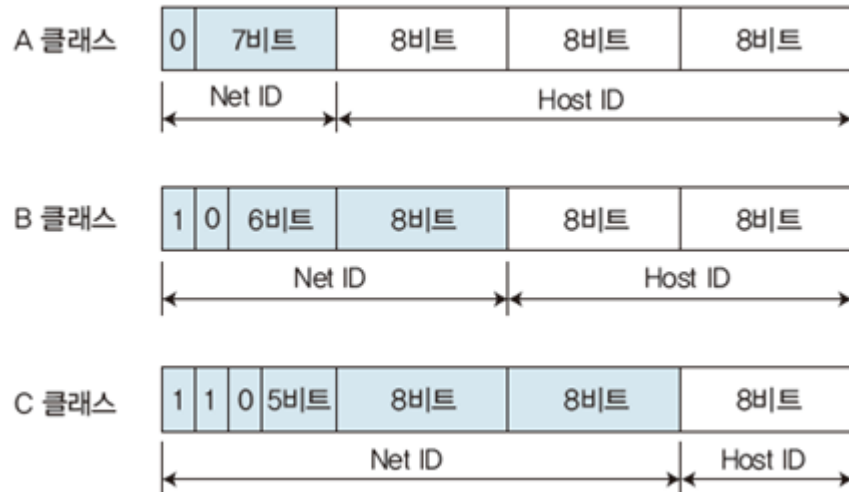
A클래스 1-126

C클래스 192-223

E클래스 240-255 연구 용

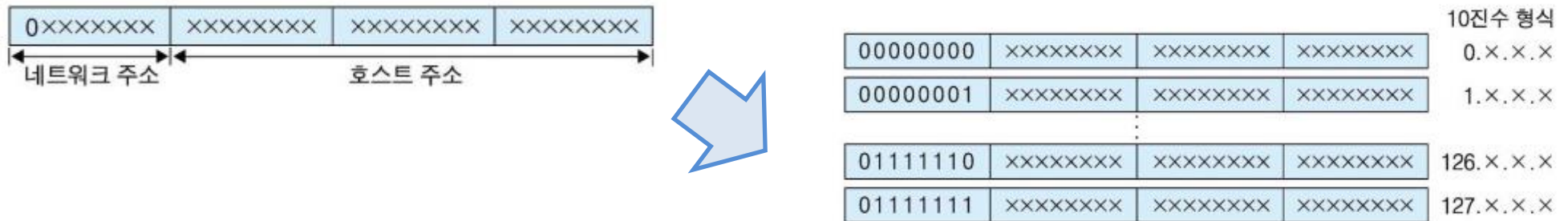
B클래스 128-191

D클래스 224-239 멀티캐스트 용



04_네트워크 주소의 표현

클래스 A는 처음 8개의 비트가 네트워크 주소고, 24개의 비트가 호스트 주소인데 첫 번째 비트가 '0'이다.



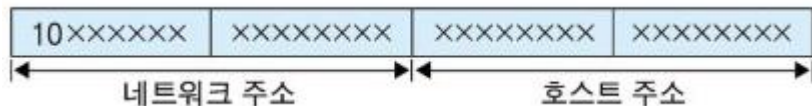
이런 클래스 A로 나타낼 수 있는 네트워크 주소는 다음과 같이 0.x.x.x~127.x.x.x가 되는데, 이 중 127.x.x.x는 다른 용도로 예약되어 있으므로 사용할 수 없다. (loopback address, 루프백 주소)

				10진수 형식
0xxxxxxx	00000000	00000000	00000000	x.0.0.0
0xxxxxxx	00000000	00000000	00000001	x.0.0.1
...				...
0xxxxxxx	00000000	00000000	11111111	x.0.0.255
0xxxxxxx	00000000	00000001	00000000	x.0.1.0
...				...
0xxxxxxx	11111111	11111111	11111110	x.255.255.254
0xxxxxxx	11111111	11111111	11111111	x.255.255.255

각 네트워크마다 나타낼 수 있는 호스트 주소는 x.0.0.0~x.255.255.255가 된다.

04_네트워크 주소의 표현

클래스 B는 처음 16개의 비트가 네트워크 주소고, 16개의 비트가 호스트 주소인데 처음 두 비트가 '10'이다.



클래스 B로 나타낼 수 있는 네트워크 주소는 128.0.x.x~191.255.x.x가 된다.

				10진수 형식
10000000	00000000	xxxxxx	xxxxxx	128.0.x.x
10000000	00000001	xxxxxx	xxxxxx	128.1.x.x
⋮				
10000000	11111111	xxxxxx	xxxxxx	128.255.x.x
10000001	00000000	xxxxxx	xxxxxx	129.0.x.x
⋮				
10111111	11111110	xxxxxx	xxxxxx	191.254.x.x
10111111	11111111	xxxxxx	xxxxxx	191.255.x.x

04_네트워크 주소의 표현

클래스 C는 처음 24개의 비트가 네트워크 주소고, 8개의 비트가 호스트 주소인데 처음 세 비트가 '110'이다



클래스 C로 나타낼 수 있는 네트워크 주소는 192.0.0.x~223.255.255.x가 된다.

				10진수 형식
11000000	00000000	00000000	xxxxxxxx	192.0.0.x
⋮				
11011111	11111111	11111111	xxxxxxxx	223.255.255.x

그리고 각 네트워크마다 나타낼 수 있는 호스트 주소는 x.x.x.0~x.x.x.255가 된다.

04_네트워크 주소의 표현

- A클래스 0000 0000 . 0000 0000 . 0000 0000 . 0000 0000
0 111 1111 . 1111 1111 . 1111 1111 . 1111 1111
- B클래스 1000 0000 . 0000 0000 . 0000 0000 . 0000 0000
10 11 1111 . 1111 1111 . 1111 1111 . 1111 1111
- C클래스 1100 0000 . 0000 0000 . 0000 0000 . 0000 0000
110 1 1111 . 1111 1111 . 1111 1111 . 1111 1111
- D클래스 1110 0000 . 0000 0000 . 0000 0000 . 0000 0000
1110 1111 . 1111 1111 . 1111 1111 . 1111 1111
- E클래스 1111 0000 . 0000 0000 . 0000 0000 . 0000 0000
1111 1111 . 1111 1111 . 1111 1111 . 1111 1111

#.TEST

다음 IP주소를 보고 해당 Class와 네트워크 주소와 호스트 주소를 구분하세요.

1) 192. 12. 100. 2

2) 132. 12. 11. 4

3) 10. 3. 4. 3

4) 203. 10. 1. 1

5) 261. 12. 4. 1

6) 130. 11. 4. 1

#.TEST (풀이)

1. 192.12.100.2	클래스: C Class	네트워크주소:192.12.100.0	호스트주소: 2
2. 132.12.11.4	클래스: B Class	네트워크주소:132.12.0.0	호스트주소: 11.4
3. 10.3.4.3	클래스: A Class	네트워크주소:10.0.0.0	호스트주소: 3.4.3
4. 203.10.1.1	클래스: C Class	네트워크주소:203.10.1	호스트주소: 1
5. 261.12.4.1	클래스: X	네트워크주소:X	호스트주소: X
6. 130.11.4.1	클래스: B Class	네트워크주소:130.11.0.0	호스트주소: 4.1

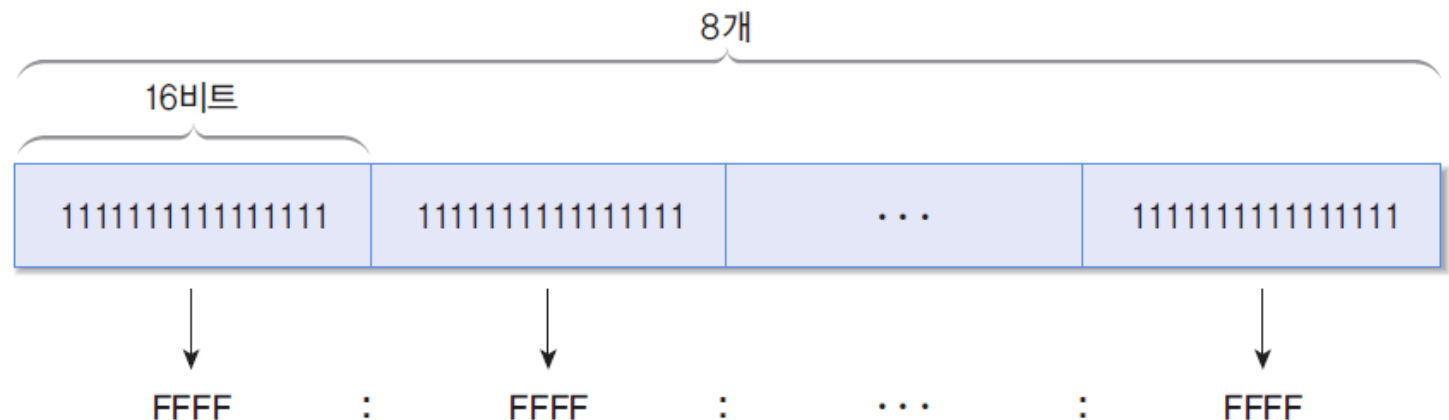
04_네트워크 주소의 표현

전 세계인구 약 74억 명 가운데 이슬람 인구는 대략 16억 명.

2^{32} 4,294,967,296

IPv6에서는 128 비트 주소 체계로 확장 → 16비트의 숫자 8개를 콜론(:)으로 구분

2^{128} 4,294,967,296 × 4,294,967,296 × 4,294,967,296 × 4,294,967,296



05_서브넷 마스크와 서브네팡팅

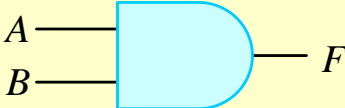
❖ 서브넡 마스크에서 중요한 2가지

- 2진수의 이해
- 논리적 AND에 대한 이해

#. 논리연산

❖ AND 연산

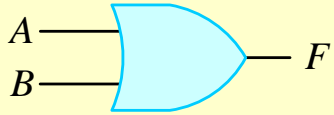
- 입력이 모두 1(on)인 경우에만 출력은 1(on)이 되고, 입력 중에 0(off)인 것이 하나라도 있을 경우에는 출력은 0(off)이 된다.

진리표	동작파형	논리기호															
<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	0	1	0	0	1	1	1	<p>입력 A 0 0 1 1 0 0 1 1</p> <p>입력 B 0 1 0 1 0 1 0 1</p> <p>출력 F 0 0 0 1 0 0 0 1</p>	<p></p> <p>논리식</p> <p>$F = AB = A \cdot B$</p>
A	B	F															
0	0	0															
0	1	0															
1	0	0															
1	1	1															

#. 논리연산

❖ OR 연산

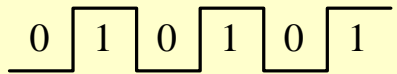
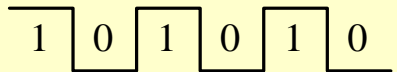
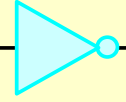
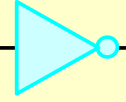
- 입력이 모두 0인 경우에만 출력은 0이 되고, 입력 중에 1이 하나라도 있으면, 출력은 1이 된다.

진리표			동작파형		논리기호															
<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>			A	B	F	0	0	0	0	1	1	1	0	1	1	1	1	<div>입력 A 0 0 1 1 0 0 1 1</div> <div>입력 B 0 1 0 1 0 1 0 1</div> <div>출력 F 0 1 1 1 0 1 1 1</div>		<div></div>
A	B	F																		
0	0	0																		
0	1	1																		
1	0	1																		
1	1	1																		
논리식																				
$F = A + B$																				

#. 논리연산

❖ NOT 연산 (논리부정)

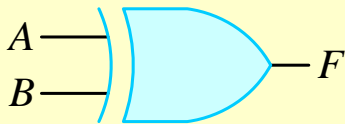
- 입력이 0인 경우에는 출력이 1이 되고, 입력이 1인 경우에는 출력이 0이 된다.

진리표	동작파형	논리기호						
<table><tr><th>A</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	F	0	1	1	0	<div>입력 A </div> <div>출력 F </div>	<div></div> <div>A —  — F</div>
A	F							
0	1							
1	0							
		논리식						
		$F = \overline{A} = A'$						

#. 논리연산

❖ XOR 연산 (Exclusive-OR)

- 입력 중 홀수 개의 1이 입력된 경우에 출력은 1이 되고 그렇지 않은 경우에는 출력은 0이 된다.
- 입력값이 같으면 0, 다르면 1이 출력된다.

진리표	동작파형	논리식															
<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	0	<p>입력 A 0 0 1 1 0 0 1 1</p> <p>입력 B 0 1 0 1 0 1 0 1</p> <p>출력 F 0 1 1 0 0 1 1 0</p>	$F = A \oplus B = \bar{A}B + A\bar{B}$
A	B	F															
0	0	0															
0	1	1															
1	0	1															
1	1	0															
		논리기호															
																	

[illegible]

05_서브넷 마스크와 서브네팡팅

❖ 서브넡 마스크 (Subnet Mask)

- IP 주소 체계의 Network ID와 Host ID를 서브넡 마스크를 통해 변경하여서 '네트워크 영역을 분리 또는 합체'시키는 개념
 - 네트워크를 나누기 위해 / 네트워크 주소 부분을 구별하기 위해 사용
 - 네트워크 주소 부분의 모든 비트는 1로 설정
 - 호스트 주소 부분의 모든 비트는 0으로 설정

05_서브넷 마스크와 서브네팡팅

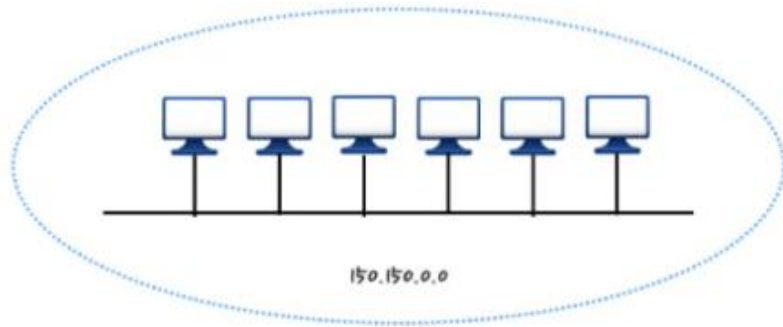
❖ 서브넡 마스크 (Subnet Mask)

- IP 주소 체계의 Network ID와 Host ID를 서브넡 마스크를 통해 변경하여서 '네트워크 영역을 분리 또는 합체'시키는 개념
 - 네트워크를 나누기 위해 / 네트워크 주소 부분을 구별하기 위해 사용
 - 네트워크 주소 부분의 모든 비트는 1로 설정
 - 호스트 주소 부분의 모든 비트는 0으로 설정

클래스	비트 표기	서브넡마스크
Class A	11111111 00000000 00000000 00000000	255.0.0.0 /8
Class B	11111111 11111111 00000000 00000000	255.255.0.0 /16
Class C	11111111 11111111 11111111 00000000	255.255.255.0 /24

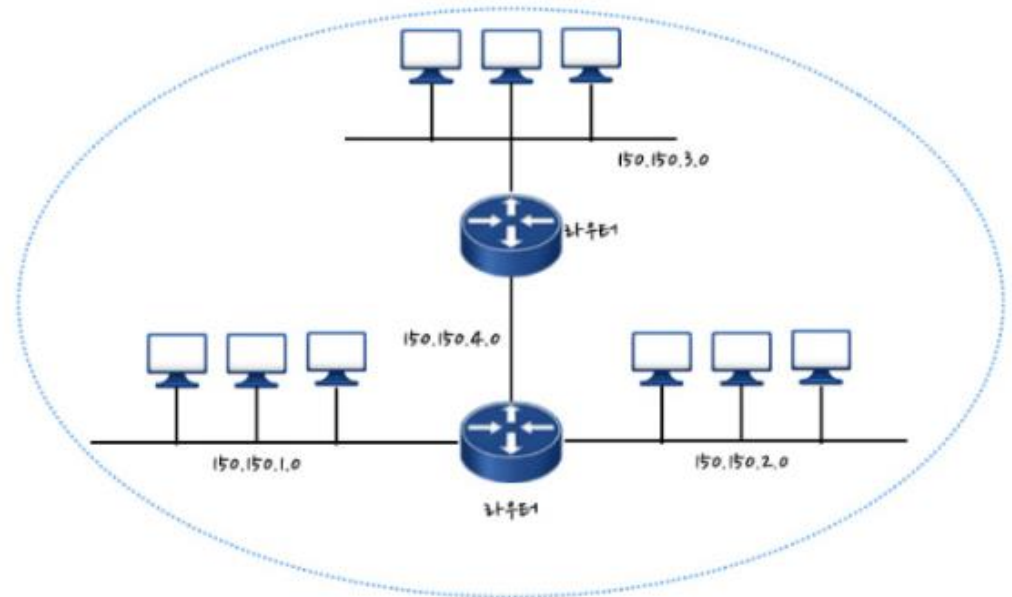
05_서브넷 마스크와 서브네팡팅

서브넡 마스크는 주어진 ip 주소를 넡트웍 환경에 맞게 나누어 주기 위해서 씌워주는 이진수의 조합



서브넡 마스크를 하지 않는 넡트웍

- 넡트웍 150.150.0.0(클래스 B, 호스트 수 약 65000개)
- 브로드캐스트 도메인이 너무 커짐
- 실제 상황에서 통신 불가능



서브넡 마스크를 한 넡트웍

- 클래스 B 넡트웍 150.150.0.0을 서브넡해서 사용함
- 서브넡 마스크는 255, 255, 255, 0
- 나눠진 서브넡 간의 통신은 라우터를 거쳐야만 가능

05_서브넷 마스크와 서브네팅

❖ 서브넷 마스크 (Subnet Mask)의 예시

- Class C인 192.168.0.0. 주소의 네트워크 주소와 서브넷 마스크를 표기

네트워크 주소	192.168.0.0
서브넷 마스크	255.255.255.0 → 192.168.0.0/24

☞ 같은 네트워크에 존재하는 모든 호스트는 동일한 네트워크 주소를 사용해야 하기 때문에 동일한 서브넷 마스크를 사용해야 한다.

B Class	192.168.23.21	192.168.0.1 ~ 192.168.255.254
C Class	192.168.23.21	192.168.23.1 ~ 192.168.23.254

05_서브넷 마스크와 서브네팡팅

❖ 서브넡 마스크 (Subnet Mask)에서의 넡트웍 주소 결정방법

- AND 연산을 이용

IP 주소	192.168.123.132	11000000. 10101000. 01111011. 10000100
서브넡마스크	255.255.255.0	11111111. 11111111. 11111111. 00000000
넡트웍 주소	192.168.123.0	11000000. 10101000. 01111011. 00000000

05_서브넷 마스크와 서브네팡팅

❖ 서브넡 마스크 (Subnet Mask)의 규칙

- 2진수로 썼을때 '1' 이 연속적으로 나와야 사용가능
(1의 사이에는 어떠한 경우에도 0이 오면 안된다. 즉 1이 연속적으로 나온 후에 0이 나오는게 규칙)

255.255.255.10	11111111. 11111111. 11111111. 00001010
255.255.255.15	11111111. 11111111. 11111111. 00001111
255.255.255.252	11111111. 11111111. 11111111. 11111100

- 서브넡 마스크로 만들어진 네트워크인 서브넡은 라우터를 통해서만 통신이 가능
 - B Class 150.100.0.0에서 150.100.100.1과 150.100.200.1
C Class로 서브넡 마스크를 씌운다면, 150.100.100.0과 150.100.200.0은 다른네트워크 (서브넡팅)

#.TEST

210.100.1.0 인 네트워크가 주어져 있다. 이 네트워크를 서브넷 당 호스트 수를 최대 30개, 서브넷 수는 4개 이상 만들어 운영하고자 한다. 그러면 서브넷 마스크를 어떻게 구성해야 하는가?

클래스 B 주소를 가지고 서브넷 마스크 255.255.255.240으로 서브넷을 만들었을 때 서브넷 수와 호스트수는 각각 얼마인가?

네트워크 주소가 201.34.12.64인 망이 있고 서브넷 마스크가 255.255.255.192이다. 서브넷에 연결되는 호스트들에 할당할 수 있는 주소의 개수는?

#.TEST-풀이

210.100.1.0 Class C

Class C 디폴트 마스크 = 255.255.255.0 subnetid와 hostid로 쓸 수 있는 비트수 = 8

네트워크당 호스트 주소로 30개까지 할당하기 위해선 hostid는 5비트 ($2^5 - 2 = 30$)이어야 한다.

따라서 $8 - 5 = 3$ 비트를 subnetid로 할당할 수 있고 (서브넷 수는 $2^3 - 2 = 6$ 개까지 사용가능)

서브넷 마스크 = 255.255.255.(11100000) = 255.255.255.224

클래스 B 디폴트 마스크 = 255.255.0.0 = 255.255.(00000000).(00000000)

문제의 subnet mask = 255.255.255.240 = 255.255.(11111111).(11110000)

subnetid 비트 수 = 12

서브넷 수 = $2^{12} - 2 = 4096 - 2 = 4094$

hostid 비트 수 = 4

호스트 수 = $2^4 - 2 = 16 - 2 = 14$

네트워크 주소 = 201.34.12.64 Class C

Class C 디폴트 마스크 = 255.255.255.0 = 255.255.255.(00000000)

문제의 서브넷마스크 = 255.255.255.192 = 255.255.255.(11000000)

subnetid 비트수 = 2

hostid 비트 수 = 6

호스트 수 = $64(2^6) - 2 = 62$ 개 (hostid의 모든 비트가 0인 것과 1인 것은 제외)

05_서브넷 마스크와 서브네팡팅

❖ 서브네팡팅 (Subneting)

- 배정받은 하나의 네트워크 주소를 다시 여러 개의 작은 네트워크로 나누어 사용 하는 것
- 할당받은 네트워크를 낭비되는 것을 방지하고자 여러 개의 네트워크로 나누는 기술

❖ 서브네팡팅 (Subneting)이 필요한 이유

- IP 주소를 효율적으로 사용하여 낭비를 막기 위해서
- 네트워크 분리하여 보안성 강화를 위해서

만약 하나의 부서에서 필요한 IP주소의 개수는 총 6개인데, C Class 주소 대역대 하나를 할당해 준다면...

- ☞ 할당된 IP의 총 개수는 256개로 낭비되는 IP가 많아 진다. 따라서, 남은 IP를 필요한 곳에 사용할 수 있도록 네트워크를 분할하는 작업이 필요

05_서브넷 마스크와 서브네팡팅

❖ 서브네팡팅 (Subneting) 예시

IP가 150.150.100.1 일때, B Class 마스크	
150.150.100.1	10010110. 10010110. 01100100. 00000001
255.255.0.0	11111111. 11111111. 00000000. 00000000
150.150.0.0	10010110. 10010110. 00000000. 00000000

IP가 150.150.100.1 일때, C Class 마스크	
150.150.100.1	10010110. 10010110. 01100100. 00000001
255.255.255.0	11111111. 11111111. 11111111. 00000000
150.150.100.0	11111111. 11111111. 01100100. 00000000

같은 IP를 B Class와 C Class로 비교해 보았더니 네트워크 부분과 호스트 부분이 차이가 발생

05_서브넷 마스크와 서브네팡팅

❖ 서브네팡팅 (Subnetting) 순서

- 필요한 IP 개수를 구한다.
- 서브네팡 마스크를 구한다.
- 서브네팡당 서브네팡 마스크에 따라 네트워크를 나눈다.

ex) 192.168.1.0 /24 서브네팡팅을 하려고 한다면? (host=30)

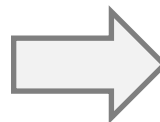
☞ 서브네팡팅 조건 : 호스트 30대, $2^5 - 2 = 30$

$2^x - 2 = y$ (x는 호스트 비트수, -2는 Network ID 와 Broadcast 주소, Y는 필요한 IP 개수)

192.168.1.0	11000000. 10101000. 00000001. 00000000
255.255.255.244	11111111. 11111111. 11111111. 11100000
192.168.1.0	11000000. 10101000. 00000001. 00000000

192.168.1.000 00000 = 192.168.1.0/27 (0~31)
192.168.1.001 00000 = 192.168.1.32/27 (32~63)
192.168.1.010 00000 = 192.168.1.64/27 (64~95)
192.168.1.011 00000 = 192.168.1.96/27 (96~127)
192.168.1.100 00000 = 192.168.1.128/27 (128~159)
192.168.1.101 00000 = 192.168.1.160/27 (160~191)
192.168.1.110 00000 = 192.168.1.192/27 (192~223)
192.168.1.111 00000 = 192.168.1.224/27 (224~255)

Network ID 와
Broadcast는 사용하지 않는다.



192.168.1.0/27 > 192.168.1.1 ~ 192.168.1.30
192.168.1.32/27 > 192.168.1.33 ~ 192.168.1.62
192.168.1.64/27 > 192.168.1.65 ~ 192.168.1.94
192.168.1.96/27 > 192.168.1.97 ~ 192.168.1.126
192.168.1.128/27 > 192.168.1.129 ~ 192.168.1.158
192.168.1.160/27 > 192.168.1.161 ~ 192.168.1.190
192.168.1.192/27 > 192.168.1.193 ~ 192.168.1.222
192.168.1.224/27 > 192.168.1.225 ~ 192.168.1.254

#.TEST

189.101.7.0/24 네트워크를 각 네트워크 당 50개의 Host가 사용할 수 있도록 Subnetting하시오.

- 1) 어떤 Subnet Mask가 효율적인가?
- 2) Subnet의 개수?
- 3) Host의 개수?
- 4) 마지막 Subnet의 Network-ID는?
- 5) 첫번째 Subnet의 Broadcast 주소는?
- 6) 두번째 Subnet의 사용 가능한 IP 범위는?

#.TEST (풀이)

189.101.7.0/24 네트워크를 각 네트워크 당 50개의 Host가 사용할 수 있도록 Subnetting하시오.

1) 어떤 Subnet Mask가 효율적인가?

2^6 189.101.7.0/26 → 255.255.255.192

2) Subnet의 개수?

$2^2 = 4$

189.101.7.0 ~

189.101.7.64 ~

189.101.7.128 ~

3) Host의 개수?

$2^6 - 2 = 62$

189.101.7.192 ~

4) 마지막 Subnet의 Network-ID는?

5) 첫번째 Subnet의 Broadcast 주소는?

189.101.7.0 ~ 189.101.7.63

189.101.7.64 ~

189.101.7.128 ~

189.101.7.192 ~

6) 두번째 Subnet의 사용 가능한 IP 범위는?

189.101.7.0 ~ 189.101.7.63

189.101.7.64 ~ 189.101.7.127
189.101.7.65 ~ 189.101.7.126

189.101.7.128 ~

189.101.7.192 ~

#.TEST

201.222.5.0(255.255.255.0)에서 서브넷당 필요한 IP수는 5개 이상이고, 서브넷은 20개 이상 나오게 서브네팅을 하세요.

■ 정보 보안의 3요소와 추가 요소

- 기밀성(Confidentiality)
- 무결성(Integrity)
- 가용성(Availability)
- 서버 인증(Server Authentication)
- 클라이언트 인증(Client Authentication)

06_네트워크 보안 _ 네트워크 보안의 요소

■ 기밀성

- 허락되지 않은 사용자 또는 객체가 정보의 내용을 알 수 없도록 하는 것
- 프라이버시 보호와도 밀접한 관계가 있음.
- 네트워크 보안 측면에서 기밀성은 '시스템 간 안전한 데이터 전송'과 관련이 있음.
- 스니핑(Sniffing)은 기밀성을 해치는 가장 일반적인 공격 형태
- 통신의 암호화가 가장 일반적인 보안 대책

06_네트워크 보안 _ 네트워크 보안의 요소

■ 무결성

- 허락되지 않은 사용자 또는 객체가 정보를 함부로 수정할 수 없도록 하는 것
- 네트워크에서의 무결성은 '클라이언트와 서버 간의 데이터가 변조되지 않고 전송되는가'와 관련이 있음.
- 중간에 유효한 다른 연결을 빼앗는 세션 하이재킹, 두 시스템 간의 데이터를 중간에 변조하는 MITM 공격은 무결성을 해치는 대표적인 공격
- 통신의 암호화가 가장 일반적인 보안 대책(PKI와 밀접한 관련이 있음)

■ 가용성

- 허락된 사용자 또는 객체가 정보에 접근하려고 할 때 방해받지 않도록 하는 것
- DoS가 가용성을 해치는 대표적인 공격

06_네트워크 보안 _ 네트워크 보안의 요소

■ 서버 인증

- '클라이언트가 올바른 서버로 접속하는가' 를 의미
- 서버 인증으로 생기는 문제는 무척 다양하며, 일반적으로 DNS 스푸핑이나 서버 파밍 등이 있음.
- SSL(HTTPS)을 통해 서버 인증을 하지만, 경고를 보여주고 사용자에게 선택을 하게 하는 것이 일반적이며 강제적인 서버 인증은 흔치 않음.

■ 클라이언트 인증

- '올바른 클라이언트가 접속을 시도하는가' 를 의미
- 웹 사이트에 접근할 때 사용하는 아이디와 비밀번호가 대표적인 클라이언트 인증
- 클라이언트 인증과 관련된 해킹은 스푸핑, 세션 하이재킹, 피싱 등이 있음.