TCP/IP

TCP/IP의 개요

1. TCP/IP 등장배경

1960년대 후반 미국방성(DOD)를 주축으로 하여 통신 기술의 연구 개발에 착수하였습니다. 이는 패킷 기반의 데이터 통신의 가능성을 실험하기 위한 것으로, 한 쌍의 단말간에 패킷을 송/수신하는 일대일 패킷 교환 형태였습니다. 이러한 형태는 여러 사용자가 회선을 사용할 경우 패킷의 출처와 목적지를 알아내기 어렵다는 단점을 안게 되었습니다.

이후 DOD와 4개의 대학과 연구기관(UCLA, UCSB, SRI, UTAH)을 연결하는 소규모 패킷 교환 네트워크(4개의 노드 연결)인 ARPANET(Advanced Research Projects Agency Network: 고급 연구 기간망)를 구축하게 되었습니다. 이후 ARPANET 실험의 성공으로 기존의 4개의 노드에서 50개 이상의 노드로 확대되어 패킷에 의한 데이터 통신의 실용성을 입증하게 되었습니다.

그 이후 ARPANET내의 IFIP(International Federation of Information Processing Work group 6.1) 연구 그룹에 의해 1975년 TCP/IP는 사양이 결정되어 현재의 모습으로까지 이르게 된 것입니다.

2. TCP/IP의 정의

TCP/IP(Trnasmission Control Protocol/ Internet Protocol: 전송제어/ 인터넷 프로토콜)은 보통 TCP/IP라고 하면 TCP와 IP를 각각의 프로토콜로 보는 경향이 있습니다. 하지만 TCP/IP는 각각의 프로토콜을 말하는 것이 아니라 TCP/IP 통신에 포함되는 많은 프로토콜을 의미합니다. OSI-7 Layer 측면에서 본다면 1,2 계층을 제외한 3계층 이상에서 지원되는 프로토콜군(아래그림 참조)이라고 보시면 됩니다.

[TCP/IP 프로토콜의 종류]

7	애플리케이션 계층	TELNET, FTP, SMTP, SNMP, NIS, NFS, DNS	
6	프리젠테이션 계층		
5	세션 계층	NETBIOS	
4	트랜스포트 계층	TCP	UDP
3	네트워크 계층	IP, ARP, RARP, ICMP	
2	데이터링크 계층	제외	
1	물리계층		

또한 TCP/IP는 인터넷 프로토콜의 대명사로 자리잡고 있습니다.

3. OSI-7 Layer 모델과 TCP/IP 프로토콜의 종류

3.1 네트워크 계층

① IP (Internet Protocol)

IP는 서로 다른 네트워크 간의 상호 통신을 위한 규약으로 교환망 간의 신호 방식을 말합니다. 통신망을 상호 접속할 경우 회선 특성 등의 물리적 조건이나 정보 전송 확립과 네트워크 어드레스와 호스트 어드레스 정의에 의해 통신양단 간의 주소를 지정하는 역할을 합니다.

② ARP(Address Resolution Protocol)

ARP는 TCP/IP 프로토콜의 하나로 OSI의 데이터링크층에 해당한다. IP어드레스를 하드웨어가 갖는 물리적 어드레스(MAC 어드레스)로 다이나믹하게 변환시킵니다. 네트웍상에 브로드캐스트 패킷을 보내 해당하는 머신으로부터 응답을 받아 IP 어드레스의 해당 MAC 어드레스를 얻습니다. 이의 역으로 어드레스 변환을 실현하는 것이 RARP(Reverse Address Resolution Protocol)이기도 합니다.

③ RARP(Reverse Address Resolution Protocol)

RARP는 데이터 링크 계층의 주소로부터 네트웍 계층의 주소를 얻어오는 프로토콜을 말합니다. 디스크를 가지고 있지 않은 호스트가 자신의 IP 주소를 서버로부터 얻어내기 위해서는 RARP라는 TCP/IP 인터넷

프로토콜을 사용해야 하며 일반적으로 자체의 디스크 기억장치가 없는 웍스테이션이나 지능형 터미널에 의해 이용됩니다.

④ ICMP(Internet Control Message Protocol)
ICMP는 인터넷에서 오류 메시지의 생성 및 검사 메시지와 IP에 관련된 정보를 제공하는 통신 규약을 말합니다. ICMP의 메시지에는 목적지 도착 불가능, 플로우 제어, 게이트웨이로부터의 경로 변경 요구, 에코 요구/응답, 타임아웃, 파라미터 에러 등이 있습니다.

3.2 전송계층

① TCP(Transmission Control Protocol: 전송 제어 프로토콜)
TCP는 네트워크를 통한 전송 제어 프로토콜로, 네트워크를 통한 자료 전송이 이루어질 때 데이터는 패킷(Packet)이라는 단위로 잘라져서 전송되는데 IP는 이때 데이터 패킷을 한 장소에서 다른 장소로 옮기는 역할을 하고, TCP는 데이터의 흐름을 관리하고, 데이터가 정확한지 확인하는 역할을 하게 됩니다.

② UDP(User Datagram Protocol: 사용자 데이터그램 프로토콜)
UDP는 TCP와는 달리 비연결 프로토콜로서 상대방이 보낸 응답을 확인하지 않으며, 송신한 데이터가 목적지에 도착되었는지를 확인하지 않은채 통신을 하게 됩니다. 때문에 방송적인 통신을 할 때에는 매우 유용합니다.

3.3 상위계층

① TELNET
TELNET은 네트워크를 통한 가상 단말 기능을 제공해 줍니다. 즉 TELNET에서의 가상 단말 기능은 TCP/IP 프로토콜상의 TELNET 프로토콜로 실현됩니다.

② FTP(File Transfer Protocol)
FTP는 TCP/IP 기반의 네트워크에서 사용되는 파일 전송 프로토콜로서 컴퓨터 간의 파일 송/수신을 지원해 줍니다.

③ SMTP(Simple Mail Transfer Protocol: 간이 메일 프로토콜)
SMTP는 멀티 벤더 환경하에서 제공하는 프로토콜로서 텍스트형 전자우편을 특정 사용자에게 보내는 프로토콜을 말합니다.

④ NNTP(Network News Transfer Protocol)
NNTP는 전자 뉴스의 전송을 제공해 주는 프로토콜로서 신뢰성 있는 스트림형의 데이터 전송 프로토콜(TCP)을 사용하여 네트웍에 있는 뉴스 데이터베이스를 가입자가 검색하고 읽기가 가능하도록 하는 프로토콜을 말합니다.

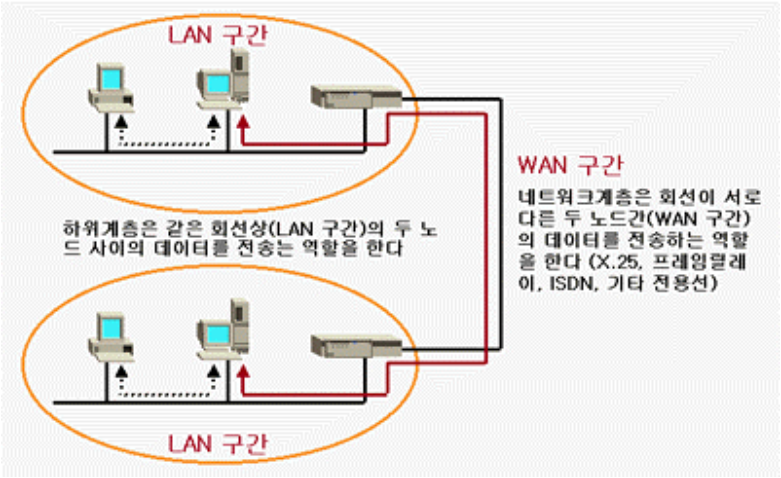
⑤ SNMP(Simple Network Management Protocol)
SNMP는 네트워크 관리 기능을 제공하는 프로토콜로서 사용자의 시스템에 장애 유무를 정기적으로 관리자에게 보내게 됩니다. 만약 사용자의 시스템에 장애가 발생될 경우 트랩(Trap)을 사용하여 관리자에게 통보하게 됩니다.

IP

1. IP의 역할

IP(Internet Protocol)는 네트워크 계층의 프로토콜의 가장 대표적인 것으로 그 주요역할은 하위계층의 서비스를 이용하여 두 노드간의 데이터전송 경로를 확립해 주는 것입니다. 여기서 하위계층은 같은 회선상(LAN 구간)의 두 노드 사이의 데이터를 전송는 역할을, 네트워크계층은 회선이 서로 다른 두 노드간(WAN 구간)의 데이터를 전송하는 역할을 하게 됩니다.

이를테면 서울에 있는 노드에서 부산에 있는 노드로 데이터를 전송하고자 한다면, 이 데이터는 서로 다른 여러 회선(X.25, 프레임릴레이, 기타 다른 전용선등)을 통하여 전송될 것입니다. 이때 네트워크계층에서 하는 역할은 자신과 상대방의 회선사이에 복수개의 회선이 접속되어 있다고 하더라도 데이터를 상대방에게 무사히 도착하도록 하는 것입니다. 이것을 다른 말로 단말 장치간 패킷 전송 서비스라고 합니다.



2. 네트워크 계층의 역할

① 주소지정

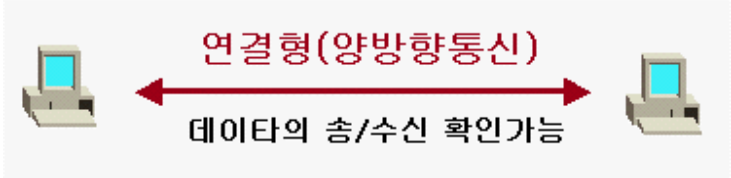
각 네트워크 상에 접속해 있는 노드의 주소를 지정해 주며, 이를 통해서 데이터를 전송할 목적지를 지정해 주는 역할을 하게 됩니다. TCP/IP의 IP Address라는 것이 바로 이 네트워크 계층에 해당되는데, 우리가 말하는 IP Address라는 말은 네트워크상에 접속해 있는 각각의 노드를 식별하는 역할을 하게 되는 겁니다.

② 경로 설정

네트워크 계층에서는 위에서 말한 바와 같이 결정된 노드의 주소를 가지고 패킷을 목적지로 전송하기 위해 최적의 경로를 설정해 주는 역할을 하게 됩니다. 이 것으로 주소 지정에서 결정된 목적지와 통신할 때 어떤 코스로 갈지를 결정해 주게 됩니다. 그리고 이렇게 결정된 경로를 따라 데이터 전송이 이루어 지게 됩니다.

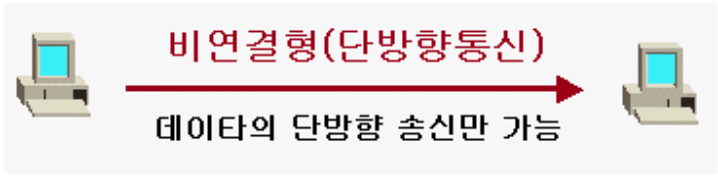
3. 네트워크 계층에서의 연결방식

① 연결형



연결형은 각 단말간의 데이터 송/수신시 데이터 통신을 위한 상호 연결 통로를 설정해 줍니다. 즉, 데이터 송신 전에 각 노드간의 통신 경로를 설정되므로 데이터는 반드시 그 경로를 통하여 상대방 노드로 확실히 전송하게 됩니다. 그러므로 확실한 데이터 전송을 요구할 때 쓰이게 됩니다.

② 비연결형



비연결형은 각 노드간의 통신 경로의 확립없이 데이터가 가진 제어 정보에 따라 데이터를 전송하게 됩니다. 따라서 데이터의 전송순서 제한이나, 상대방 노드로의 데이터 수신여부를 확인하지 않아도 되므로 연결형과는 데이터를 전송할 때 굳이 데이터 경로를 설정해 줄 필요가 없게 됩니다. 그러므로 짧은 메시지나 수신 여부의 확인이 필요없는 고속 데이터 통신의 경우 쓰이게 됩니다.

4. IP의 연결방식과 TCP의 역할

① IP의 연결방식 (비연결형)

IP가 비연결형인 이유는 물리적 계층이 가지는 신뢰성입니다. 즉, 물리 계층의 신뢰성이 높으므로 데이터의 송수신 확인 없이도 목적 노드로의 전송이 가능하기 때문입니다. 이 때문에 사용자는 목적지 노드가 실제로 어디에 있는지 신경 쓰지 않고 통신할 수 있게 되는 것입니다. 즉, 목적지 노드의 식별자인 IP 주소만 알고 있으면 기본적으로 그 이외의 것은 신경 쓰지 않고도 통신할 수 있게 되는 것입니다. 또한 데이터의 수신여부의 확인은 IP보다 상위 계층에서 확인하는 것이 신뢰성이 높기 때문에 굳이 네트워크 계층에서 확일 할 필요가 없게 됩니다.

IP의 비연결형 방식으로 가질 수 있는 가장 큰 특징으로는 고속 데이터 전송이 가능하다는 점입니다. 통신을 할 때마다 여러 가지 처리가 뒤따르는 연결형과는 달리 비연결형은 통신 처리가 단순해 지므로 그만큼 고속 전송이 가능해 지기 때문입니다.

② TCP의 역할

IP는 비연결형으로 데이터의 전송여부를 확실히 보장할 수 없습니다. 때문에 상대방과의 통신 신뢰성을 확보하기 위해 제 4계층인 전송계층의 TCP 프로토콜이 존재하는 것입니다. 네트워크 계층에서 사용되는 대부분의 프로토콜은 고속의 데이터 전송과 통신할 때 가능하면 패킷의 처리를 간소화하기 위해서 비연결형을 이용하고, 데이터의 도착여부는 상위계층인 제 4계층인 전송계층에서 처리하게 되는 것입니다.

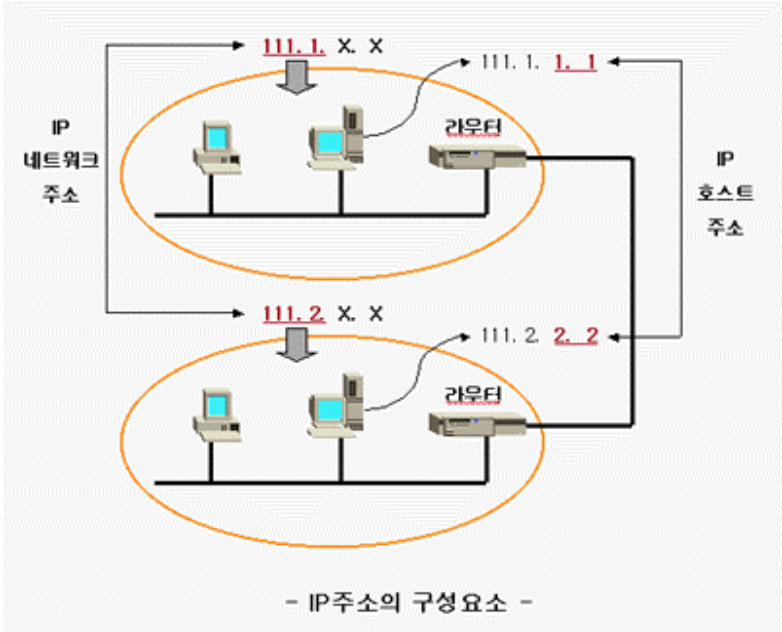
5. IP 주소의 구성

각 개인마다 고유의 주소를 가지고 있어야 우편을 받을 수 있듯이 네트워크 상에서 통신을 하기 위해서는 반드시 고유의 IP 주소가 필요하게 됩니다. IP 주소는 TCP/IP 통신의 기본이 되므로 여기서 자세히 알아보기로 하겠습니다.

IP주소는 32비트 체계(XXX. XXX. XXX. XXX)로 이루어져 있으며, 이는 네트워크의 규모나 접속된 단말기 대수에 따라 주소 체계를 선택할 수 있도록 세 종류로 분류되어 있습니다.

이 세 종류는 각각 "IP 네트워크 주소"와 "IP 호스트 주소"의 조합으로 구성되어 있습니다.

IP 네트워크 주소는 네트워크의 식별을 위한 것이고, IP 호스트 주소는 그 네트워크의 호스트(단말기)를 식별하기 위한 것입니다.



- IP주소의 구성요소 -

IP 주소는 실제로 32비트의 2진 데이터(0, 1)로 표현되어 있습니다. 하지만 실제로 설정할 때는 8비트씩 4개로 구분 짓고 마침표(.)를 넣어 10진수로 표현합니다.

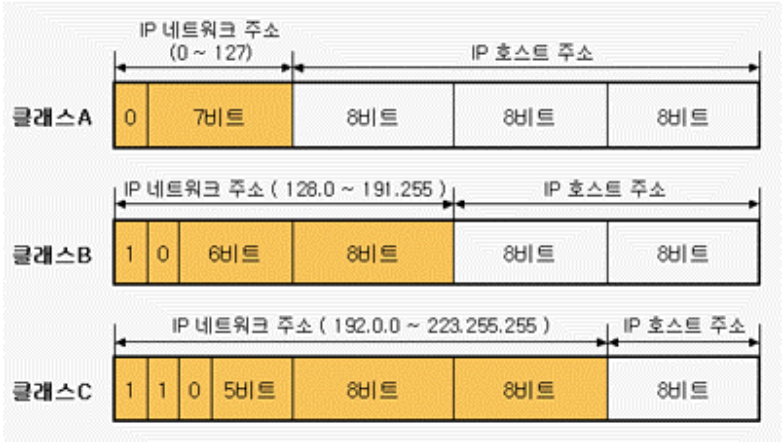
10011101000000010000000100000001 (IP주소)

-> 10011101. 000000010. 00000001. 00000001 (8비트씩 4개로)

-> 157. 2. 1. 1 (10진수로 표현)

6. IP클래스의 종류

IP클래스의 종류에는 클래스 A, 클래스 B, 클래스 C가 있습니다. 이는 각각 앞부분의 4비트로 식별할 수 있습니다. 또한 각각의 클래스에 따라 IP 네트워크 주소의 지정하는 범위가 서로 다르게 나타납니다. 또한 각 클래스의 IP 네트워크 주소 중 식별자(클래스 A: 0, 클래스 B: 10, 클래스 C: 110제외 나머지 비트)를 제외한 나머지 비트가 모두 "0" 또는 모두 "1"인 주소는 규약에 의한 예약된 주소로 사용할 수 없습니다.



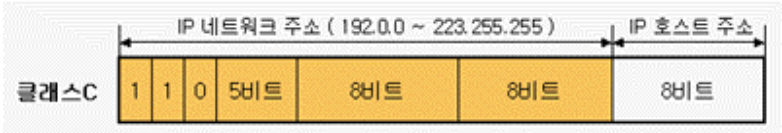
▶ 클래스 A

클래스 A는 앞의 1비트(식별용 비트)가 "0"으로 시작되는 경우입니다. IP 호스트 네트워크 주소에 8비트, IP 네트워크 주소에 24비트 주소가 지정되어 있습니다.
IP 네트워크 주소는 0 ~ 127까지이며, 식별용 비트를 뺀 나머지 비트이므로 2^ 7 = 128개이며, 이중 0과 127은 예약되어 있으므로 실제로 사용할 수 있는 IP 네트워크 주소의 수는 126개로 다음과 같습니다.
00000000. 00000000. 00000000. 00000000 (0. 0. 0. 0) 예약주소
00000001. 00000000. 00000000. 00000000 (1. 0. 0. 0) 사용가능
……
01111110. 00000000. 00000000. 00000000 (126. 0. 0. 0) 사용가능
01111111. 00000000. 00000000. 00000000 (127. 0. 0. 0) 예약주소
IP 호스트 주소는 IP네트워크 주소를 제외한 24비트 부분으로서 2^ 24 = 16777216개를 가질 수 있습니다. 이 중 모두 "0" 또는 "1"인 것은 규약에 의해 예약된 것으로 사용할 수 없습니다. 때문에 클래스 A의 IP 호스트 주소로 할당할 수 있는 것은 IP 네트워크 주소 1개당 16777214개가 됩니다.

▶ 클래스 B

클래스 B는 앞의 2비트(식별용 비트)가 "10"으로 시작되는 경우입니다. IP 호스트 네트워크 주소에 16비트, IP 네트워크 주소에 16비트 주소가 지정되어 있습니다.
IP 네트워크 주소는 128.0 ~ 191.255까지이며, 식별용 비트를 뺀 나머지 비트이므로 2^ 14 = 16284개가 되며, 이중 128.0과 192.255은 예약되어 있으므로 실제로 사용할 수 있는 IP 네트워크 주소의 수는 16282개로 다음과 같습니다.
10000000. 00000000. 00000000. 00000000 (128. 0. 0. 0) 예약주소
10000000. 00000001. 00000000. 00000000 (128. 1. 0. 0) 사용가능
……
10111110. 11111110. 00000000. 00000000 (191. 254. 0. 0) 사용가능
10111111. 11111111. 00000000. 00000000 (191. 255. 0. 0) 예약주소
IP 호스트 주소는 IP네트워크 주소를 제외한 16비트 부분으로서 2^ 16 = 65536개를 가질 수 있습니다. 이중 모두 "0" 또는 "1"인 것은 규약에 의해 예약된 것으로 사용할 수 없습니다. 때문에 클래스 A의 IP 호스트 주소로 할당할 수 있는 것은 IP 네트워크 주소 1개당 65534개가 됩니다.

▶ 클래스 C



클래스 C는 앞의 3비트(식별용 비트)가 "110"으로 시작되는 경우입니다. IP 호스트 네트워크 주소에 24비트, IP 네트워크 주소에 8비트 주소가 지정되어 있습니다.
IP 네트워크 주소는 192.0.0~ 223.255.255까지이며, 식별용 비트를 뺀 나머지 비트이므로 2^ 21 = 2097152개가 되며, 이중 192.0.0과 223.255.255은 예약되어 있으므로 실제로 사용할 수 있는 IP 네트워크 주소의 수는 2097150개로 다음과 같습니다.

11000000. 00000000. 00000000. 00000000 (192. 0. 0. 0) 예약주소
11000000. 00000000. 00000001. 00000000 (192. 0. 1. 0) 사용가능
……
11011111. 11111111. 11111110. 00000000 (223. 255. 254. 0) 사용가능
11011110. 11111111. 11111111. 00000000 (223. 255. 255. 0) 예약주소
IP 호스트 주소는 IP네트워크 주소를 제외한 8비트 부분으로서 2^ 8 = 256개를 가질 수 있습니다. 이중 모두 "0" 또는 "1"인 것은 규약에 의해 예약된 것으로 사용할 수 없습니다. 때문에 클래스 A의 IP 호스트 주소로 할당할 수 있는 것은 IP 네트워크 주소 1개당 254개가 됩니다.

서브넷 마스크

1. 각 클래스별 IP 네트워크 기본값

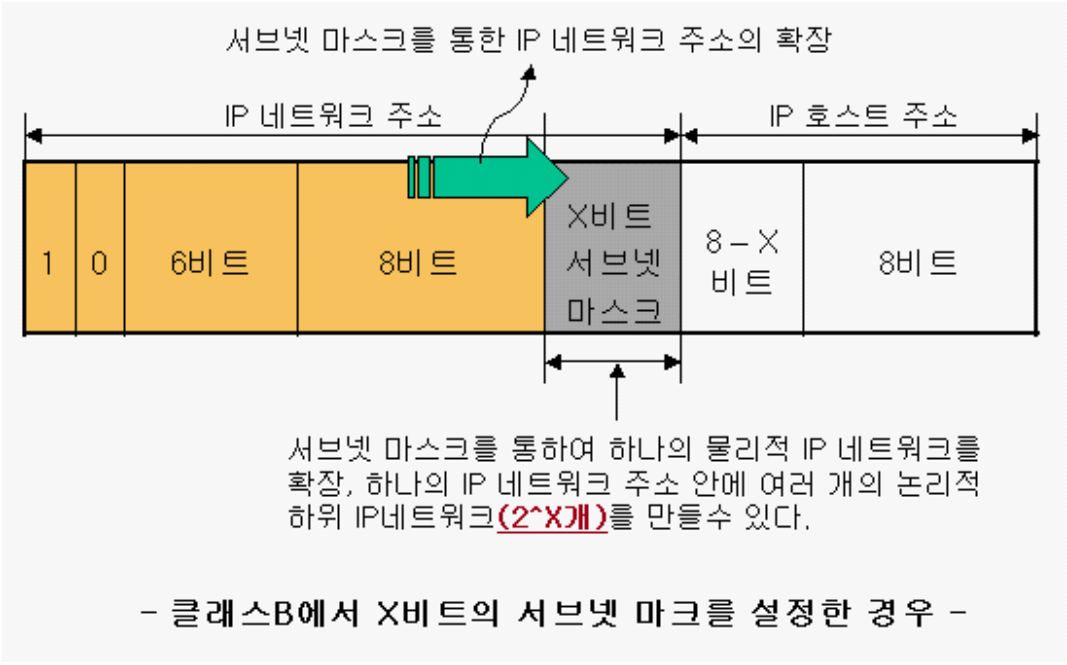
기본값은 해당 클래스별 IP 네트워크 주소의 범위를 나타내는 것으로 클래스 A는 8비트(식별 비트 "0"을 제외하면 7비트), 클래스 B는 16비트(식별 비트 "10"을 제외하면 14비트), 클래스 C는 24비트(식별비트 "110"을 제외하면 21비트)의 네트워크 주소부를 "1"로 나타내는 것이고 나머지 IP 호스트 주소는 "0"으로 나타내 주게 됩니다.

각 클래스별 IP 네트워크 기본값은 다음과 같습니다. (IP 네트워크 주소는 "1"로, IP 호스트 주소는 "0"으로 나타냄)

클래스 A 11111111. 00000000. 00000000. 00000000 (255. 0. 0. 0)
클래스 B 11111111. 11111111. 00000000. 00000000 (255. 255. 0. 0)
클래스 C 11111111. 11111111. 11111111. 00000000 (255. 255. 255. 0)

2. 서브넷 마스크

서브넷 마스크는 하나의 IP 네트워크 주소를 다시 여러 IP 서브 네트워크로 나누는 역할을 합니다. 즉 IP 네트워크 주소부를 나타내는 비트 수를 IP 호스트 주소 부분까지 확장하여 하나의 IP 네트워크 주소 속에서 다시 여러 개의 IP 네트워크 주소를 만들어 주는 것입니다.



그러면 클래스 B를 통한 서브넷 마스크의 예를 들어 보겠습니다.

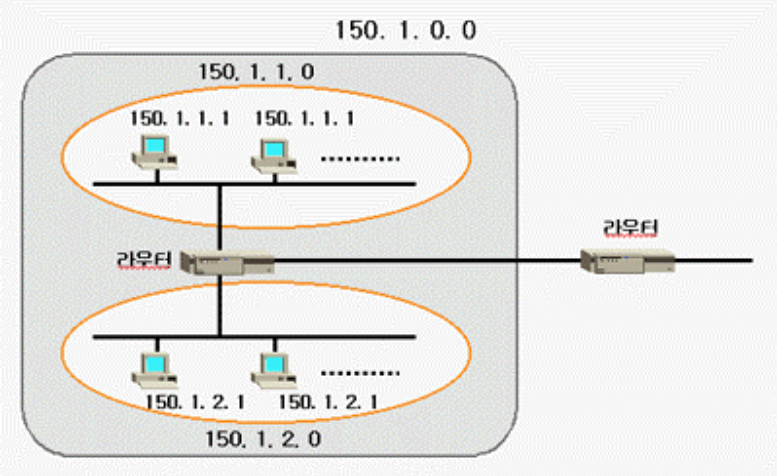
150. 1. 0. 0 (10010110. 00000001. 00000000. 00000000)의 클래스 B의 IP 네트워크 주소가 있다가 가정해 보겠습니다. 클래스 B의 IP 네트워크 주소는 16비트까지므로 클래스 B인 150.1.0.0의 기본값은 다음과 같습니다.

--> 11111111. 11111111. 00000000. 00000000 (255. 255. 0. 0)

이제 하나의 물리적인 IP 네트워크 150.1.0.0 (10010110. 00000001. 00000000. 00000000)안에 8비트 서브넷 마스크를 이용하여 256개(2^ 8 = 256)의 논리적인 IP 네트워크 주소를 만들어 보겠습니다.

우선 클래스 B인 150.1.0.0의 기본값에서 IP 네트워크 주소를 나타내는 앞부분의 16비트를 8비트의 서브넷 마스크를 통하여 확장(IP 네트워크 주소는 앞에서부터 24비트까지로 확장)하면 다음과 같이 됩니다.

--> 11111111. 11111111. 11111111 (8비트 서브넷 마스크). 00000000 (255. 255. 255. 0)
따라서 앞에서 16비트까지의 IP주소를 다시 8비트(2^ 8 = 256개) 서브 네트워크로 나누고, 나머지 8비트는 IP 호스트 주소가 됩니다.
즉 150.1.1.1과 150.1.2.1은 같은 IP 네트워크 주소 안에 존재하지만, 서브넷에 따라 분할되어 논리적으로 서로 다른 IP 네트워크가 된다. 이렇게 하면 클래스 B의 IP 네트워크 주소이면서 호스트 주소는 클래스 C처럼 할당할 수 있게 되는 것입니다.



3. 주의 사항

기본적으로 서브 네트워크 부분 중 비트 값이 모두 0 또는 모두 1일 부분은 예약되어 있으므로 사용할 수 없다는 것입니다. 따라서 서브넷 마스크로 분할된 서브네트워크 수에서 2를 뺀 수 만큼 실제 서브 네트워크로 정의할 수 있게 됩니다.

ARP

1. 방송 주소

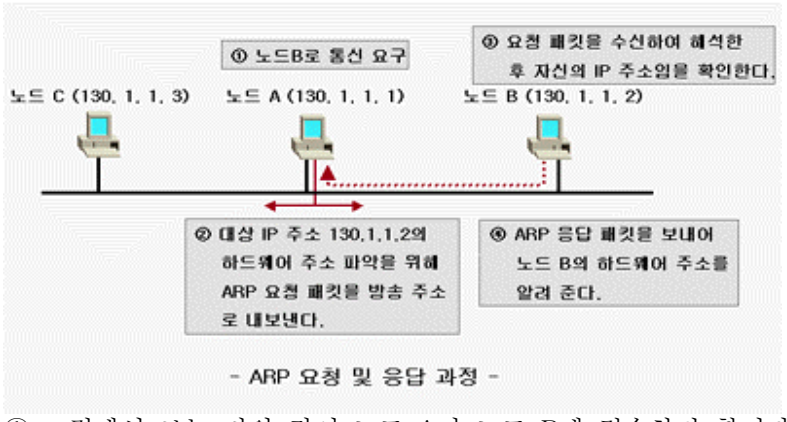
IP 호스트 주소를 할당할 때 0과 255는 노드의 고유 IP 주소로 정의할 수 없게 됩니다.
예를 들면 0은 그 주소를 알 수 없는 노드용으로서 130.1.0.0는 130.1이라는 IP 네트워크 주소를 자기고 있지만 호스트 주소는 알 수 없는 노드가 됩니다.
255는 방송 주소로 사용되며, 이 방송 주소는 네트워크 전체에 동시에 보내기 위한 목적지 IP 주소로서 일반적으로 "호스트 1의 방송 주소"라고 말합니다. 예를 들어 250대의 모든 노드에 정보를 보내야 할 경우 각 목적지 IP 주소에 250개의 패킷을 보내야만 할 것입니다. 하지만 방송 주소를 사용하여 패킷을 보내면 패킷 하나로 모든 노드에 정보를 전달할 수 있게 됩니다. 이처럼 네트워크의 모든 노드에 정보를 전송하고자 할 때 방송주소를 사용하게 되는 것입니다.

2. ARP

우리가 실생활에서 상대방의 이름만 알고서 그 사람을 찾기란 힘들 것입니다. 어느 한 사람을 찾기 위해서는 이름과, 그 사람의 주소나 전화번호를 알아야 비로서 그 사람과 연락이 가능하게 됩니다. 이렇듯 컴퓨터간의 통신에서도 목적지 IP 주소만 알아서는 통신이 이루워지는 것은 아닙니다. 각 노드와의 통신을 위해서는 IP 주소와 이에 대응하는 물리적인 주소 즉 하드웨어 주소(MAC 어드레스)를 알아야만 합니다. 일반적으로 컴퓨터는 자신의 하드웨어 주소는 인식하지만 목적지의 하드웨어 주소는 인식하지 못합니다. 이러한 이유 때문에 목적지 IP 주소를 단서로 하여 하드웨어 주소를 알아 내기 위한 방법으로 ARP(Address Resolution Protocol: 주소 결정 프로토콜)이라는 프로토콜이 쓰이게 되는 것입니다.

3. ARP를 이용한 통신 메카니즘

실제로 ARP를 이용하여 하드웨어 주소를 알아내기 위해서는 ARP 요청 패킷과 ARP 응답 패킷 두종류가 필요하게 됩니다.
그럼 다음의 예를 들어 설명하도록 하겠습니다.



① 그림에서 보는 바와 같이 노드 A가 노드 B에 접속하려 합니다.
② 노드 A는 노드 B의 MAC 주소(하드웨어 주소)를 알아내기 위해 노드 B의 IP 주소(130.1.1.2)의 정보가 들어있는 ARP 요청 패킷을 방송하게 됩니다.
③ 같은 세그먼트 안의 다른 노드는 이 방송된 ARP 요청 패킷을 수신하여 내용을 해석합니다.
④ 대상 IP 주소에 해당하는 노드 B가 MAC 주소(하드웨어 주소)를 알려주기 위해 ARP 응답 패킷을 노드 A로 반송하게 됩니다.

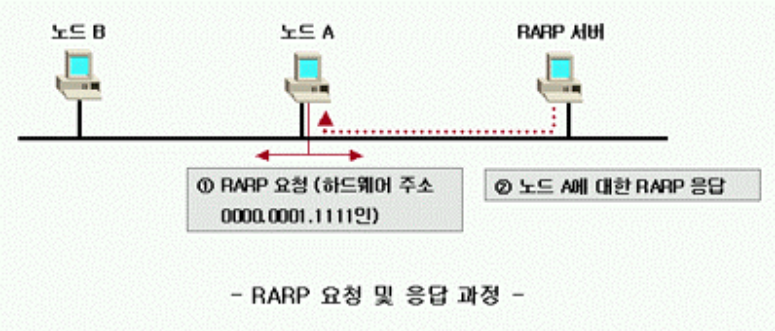
즉, ARP 요청 패킷이란 IP 주소에서 MAC 주소를 알기 위해 보내는 패킷이며, ARP 응답 패킷은 이 ARP 요청의 대상 IP 주소에 해당하는 노드가 자신의 MAC 주소를 알려주기 위해 반송하는 패킷을 말합니다. 이처럼 ARP 응답과 요청에 의해 IP 주소로부터 하드웨어 주소를 알 수 있으므로 통신이 가능해 지는

것입니다.

ARP 요청과 응답에 의해 얻어진 하드웨어 주소는 대개 ARP 패스라는 표에 등록되어 IP 주소와 하드웨어 주소에 관한 정보가 ARP 캐시에서 지워질 때까지 유지가 되므로, 한 번 처리된 IP 주소에 대한 ARP 요청 및 응답은 이 후 처리할 필요가 없이 바로 알 수 있게 됩니다.

4. RARP(Reverse ARP)

IP 주소로부터 하드웨어 주소를 알아 낼 때 사용하는 ARP와는 반대되는 개념으로 RARP는 하드웨어 주소로부터 IP 주소를 알아 낼 때 사용합니다. RARP를 사용하기 위해서는 RARP의 요구에 응답하기 위한 서버가 필요하게 됩니다.



- RARP 요청 및 응답 과정 -

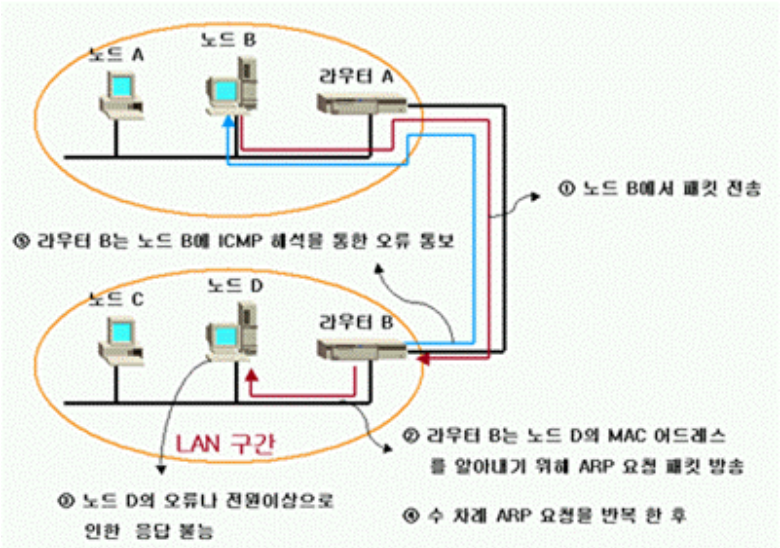
ICMP

(Internet Control Message Protocol)

1. ICMP의 개념

IP 프로토콜은 비연결형으로 패킷이 확실히 전송된다는 보장이 없기 때문에 라우터나 노드(호스트)등에서 오류가 생겨 목적지까지 도달하지 못하게 되는 경우가 생기게 됩니다. 이 때 송신측의 상태를 알려줄 필요가 있는데 이 때 필요한 것이 ICMP 프로토콜입니다. ICMP 프로토콜은 송신쪽의 상황과 목적지 노드의 상황을 진단하는 기능을 하게 됩니다.

다음은 ICMP에서의 역할을 간략하게 나타낸 그림입니다.



위의 그림은 노드 B가 노드 D로 패킷을 전송하는데 있어 노드 B의 장애로 인해 라우터 B가 노드 D를 발견하지 못하는 경우입니다. 이 때 라우터 B는 IP와 ICMP로 노드 B에게 알리게 됩니다. 노드 B는 ICMP의 정보를 해석하여 노드 D에 장애가 있는지 여부를 알 수 있게 됩니다.

2. ICMP 메시지의 종류

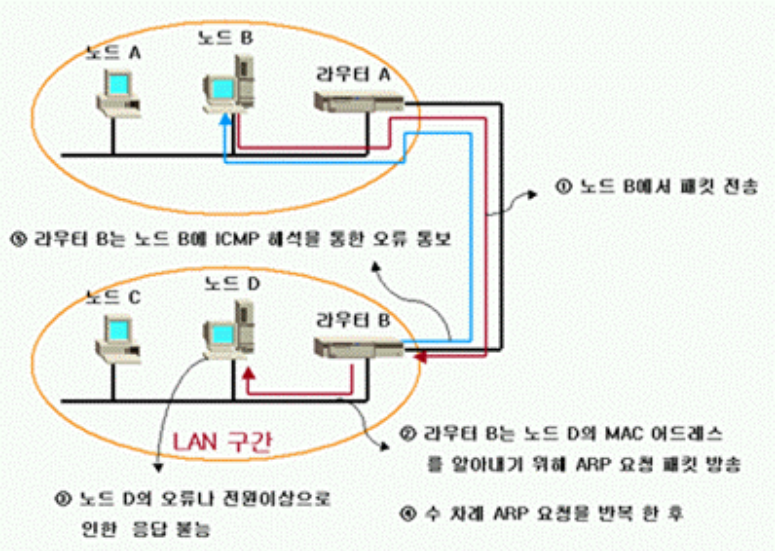
ICMP는 크게 오류 통지를 위한 오류 메시지와 진단용 문답 메시지 두 종류로 구분할 수 있습니다.

① ICMP Destination Unreachable (ICMP 목적지 도착 안함) 메시지

라우터가 특정 노드의 패킷을 목적지에 보내지 못할 경우, 송신 노드에 대해 "ICMP Destination Unreachable 메시지"를 보내게 됩니다. 이 메세지 안에는 목적지까지 전송되지 못한 이유를 나타내는 정보를 포함하게 됩니다. 목적지 노드의 IP 주소의 경로를 찾아내지 못한 라우터는 이를 다시 송신측 라우터로 이 메시지를 되돌려 보내게 되는 것입니다. 송신측 노드는 이 메시지를 해석해 보고 패킷이 목적지에 도착하지 못했음을 알 수 있게 됩니다.

② ICMP Redirect(ICMP 재지정) 메시지

라우터가 송신측 노드에 적합하지 않은 경로 설정이 되어 있을 경우 그 노드에 대한 최적화된 경로를 재지정해 주는 ICMP Redirect 메시지를 보내게 됩니다.



③ ICMPTime Exceeded(ICMP 시간 초과) 메시지
패킷이 네트워크 사이에서 무한정 돌아가지 않도록 하기 위해 각 라우터들이 패킷을 처리할 때마다 TTL(Time To Live: 활성화 지속 시간)을 감소시키다가 그 값이 "0"이 되면 패킷을 폐기하기 위해 송신측 라우터에 "ICMP Time Exceeded(ICMP 시간 초과) 메시지"를 되돌려 보냄으로서 패킷이 폐기된 사실을 알리게 됩니다.

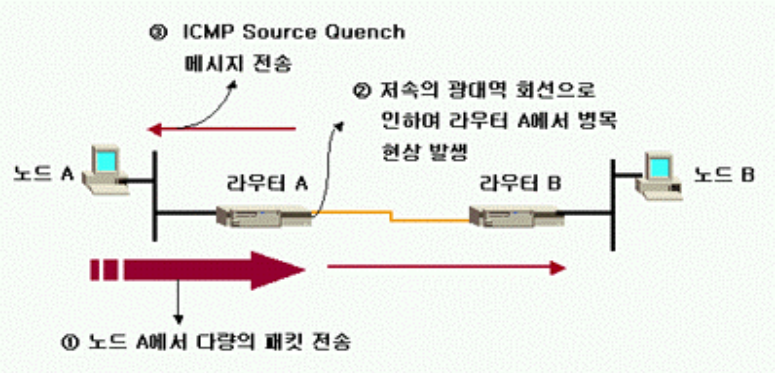
④ ICMP Information Request(ICMP 정보 요청) 메세지
자신의 IP 주소를 알아 내기 위해 "ICMP Information Request 메시지"를 보내고, 그에 대한 "ICMP Information Reply(ICMP 정보 응답) 메시지"를 받아서 자신의 IP 주소를 인식하기 위한 메시지를 말합니다. 지금은 RARP로 대처되고 있습니다.

⑤ ICMP Timestamp Request(ICMP 시간 소요 요청) 메시지
"ICMP Timestamp Request 메시지"는 송신쪽 노드가 상대방 노드의 현재 시간값을 알기 위해 송신하는 메시지를 말합니다. 이에 대해 대상 노드는 "ICMP Timestamp Reply(ICMP 시간 소요 응답) 메시지"로 답신을 하게 됩니다. 이 메시지에는 송신쪽이 데이터를 다시 보내는 시간과 대상 노드가 수신한 시간 및 응답하기 직전의 시간 정보가 들어가 있습니다. 송신쪽은 이 정보를 통해 네트워크 통과 시간을 계산할 수 있게 됩니다.

⑥ ICMP Address Mask(ICMP 주소 마스크) 메시지
자기 기계의서브넷 마스크를 알기 위해 "ICMP Address Mask 메시지"를 보내고, 그에 대해 "ICMP Address Mask Reply(ICMP 주소 마스크 응답) 메시지"에 의해 마스크 값을 알 수 있게 됩니다.

⑦ ICMP Source Quench(ICMP 소스 억제) 메시지
저속 광역 회선 등을 사용할 경우에는 IP 라우터의 WAN쪽에서 집중이 발생할 수 있습니다. 이 집중을 완화시키기 위해, 송신측 큐 값이 "0"으로 남아 송신 불능 상태가 되면 "ICMP Source Quench 메시지"를

송신측 노드측에 보내게 되고, 송신측은 이 메시지의 정보를 해석하여 회선의 어딘가가 혼잡하다는 것을 인식하고 송신 패킷의 양을 제어하게 됩니다.



⑧ ICMP Echo Request(ICMP 반향 요청) 메시지

송신측의 전송 패킷이 목적지 노드 또는 라우터에 도착했는지 여부를 확인하는데 사용합니다. 송신측 노드는 목적지 노드에 대해 "ICMP Echo Request(ICMP 반향 요청) 메시지"를 송신하고, 목적지 노드로부터 "ICMP Echo Reply(ICMP 반향 응답 메시지"가 회신되면 패킷이 무사히 전송된 것으로 인식하게 됩니다.

IP 헤더

1. IP 헤더의 정의

IP 헤더는 안에는 IP 통신을 하기 위한 모는 정보가 들어 있으며, IP 데이터에 붙어 상위층에서 받은 데이터를 IP 헤더에 포함시켜 하위층에 건네 주고, 하위층에서 받은 데이터의 IP를 해석하여 필요할 경우 상위층으로 건네주는 역할을 하게 됩니다.

2. IP 헤더의 구성

다음은 IP 헤더의 구조와 각각의 역할에 대해 살펴보도록 하겠습니다.

① VER(Version): 버전
IP 헤더의 버전 번호로서 4비트로 이루어져 있다.

② IHL(Internet Header Length): 헤더 길이
IP 헤더의 크기를 나타내며 4비트 크기로 이루어져 있다.

③ TOS(Type of Service): 서비스 유형
송신중인 IP의 서비스 품질을 나타내며 8비트로 이루어져 있다.
TOS의 서비스 유형은 다음과 같다.

비트번호	서비스유형
0, 1, 2	우선순위
3	낮은 지연 시간 요청
4	높은 처리 능력 요청
5	높은 신뢰성 요청
6, 7	사용하지 않음

④ TL(Total Length): 패킷 길이
IP 헤더와 IP 데이터를 포함함 패킷 전체의 옥텍 길이로서 16비트로 이루어져 있다.

⑤ ID(Identification): 식별자
데이터를 상위계층에 넘겨 줄 때 참고가 되는 정보로서 16비트로 이루어져 있다. 상위 계층은 이 정보를 기준으로 분할된 데이터를 다시 구성한다.

⑥ FL(Flag): 프래그
패킷 분할에 관한 제어를 나타내며 3비트로 이루어져 있다.
각 비트는 다음과 같은 의미를 지니다.

비트 번호	의미
0	사용하지 않음
1	분할 여부(0 : 분할가능, 1 : 분할 불가)
2	분할 패킷인 경우 마지막 패킷인지 여부 판단 (0 : 마지막 분할 패킷, 1 : 중간 분할 패킷)

⑦ FO(Fragment Offset): 분할 간격
원본 데이터 중 분할된 정보의 위치를 나타내며, 13비트로 이루어져 있다.

⑧ TTL(Time To Live): 활성화 지속 시간
패킷이 네트워크 상에서 존재하는 지속 시간(단위: 초)을 나타내며 8비트로 이루어져 있다. 이 값이 "0"이 되면 패킷은 폐기된다.

⑨ PROT(Protocol): 프로토콜
상위 계층의 프로토콜의 종류를 나타내며 8비트로 이루어져 있다.
예) 1 (ICMP), 6 (TCP), 17 (UDP)

⑩ HC(Header Checksum): 헤더 책섬
IP 헤더 책섬을 나타내며 16비트로 이루어져 있다.

⑪ SA(Source IP Address): 송신측 IP 주소
송신측 IP 주소를 나타내며 32비트로 이루어져 있다.

⑫ DA(Destination IP Address): 목적지 IP 주소
목적지 IP 주소를 나타내며 32비트로 이루어져 있다.

⑬ OPTION(Option): 옵션
보통은 사용 않하며, 시험이나 디버그 할 때 사용되며, 유동적인 길이를 가지고 있다.

⑭ PAD(Padding): 패딩
옵션을 선택하여 헤더가 32비트 정수 배가 되지 않는 경우 패딩(자리 채우기)을 통해 정수 배로 만든다.

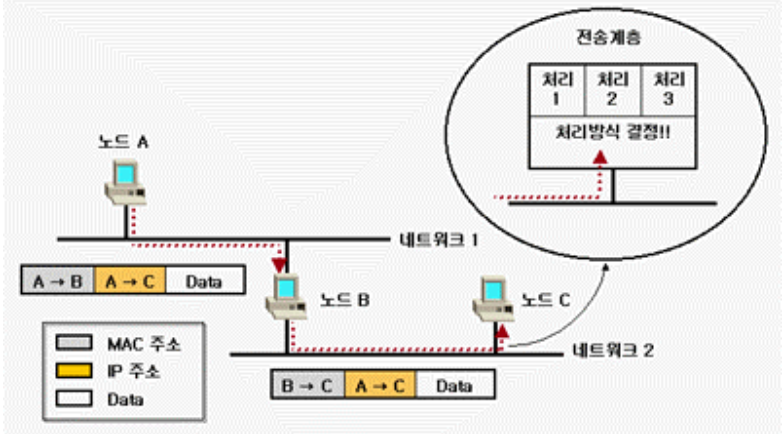
⑮ DATA(Data): 데이터
데이터가 들어가는 부분을 나타낸다.

TCP와 UDP

1. 4계층(전송계층)의 역할

3계층(네트워크 계층)에서는 데이터의 목적지 주소를 지정한 것에 대한 경로를 설정하고 4계층(전송계층)에서는 3계층에서 설정된 경로로 전송되는 패킷에 대한 데이터 전송, 에러복구 및 흐름제어, 네트워크 어드레싱 등의 처리를 담당하게 됩니다.

다음 예를 통해 전송계층의 역할을 자세히 살펴 보도록 하겠습니다.



위와 같이 노드A에서 노드B를 경유하여 노드C로 패킷이 때를 나타냅니다.

① 노드 A → 노드 B

노드 A에서 노드 C로 보낸 패킷을 전송할 때 노드 A의 패킷 안에 우선 노드 C로 향하는 MAC 주소를 붙여 보내게 됩니다.

② 노드 B → 노드 C

노드 B는 자신의 MAC 주소임을 확인하고 패킷을 받아 IP주소를 해석하여 노드 C로 향하는 패킷임을 확인하게 됩니다. 그리고 노드 B에서 노드 C로 향하는 이더넷 헤더를 붙여 네트워크 2로 전송하게 됩니다.

③ 노드 C

노드 C에서는 도착한 패킷에 대한 처리 방식을 결정하게 됩니다. 예를 든다면 주소와 이름이 적혀져 있는 편지를 생각해 볼 수 있습니다. 편지가 해당 주소로 도착한 경우, 가족의 구성원은 편지의 내용이나 이름을 확인하고 그 편지를 어떻게 처리할지 판단하게 됩니다. 이와 마찬가지로 4계층(전송계층)은 수신된 패킷에 대한 처리를 결정하고 정의하는 역할을 하게 되는 것입니다.

2. TCP와 UDP의 개요

TCP/IP 통신에서 전송 계층의 기능을 하는 대표적인 프로토콜은 "TCP"와 "UDP"입니다. 앞에서 설명하였듯이 목적지에 도착된 패킷에 대한 처리와 정확한 전문을 넘겨주기 위한 처리라는 점에서 이 두 가지 프로토콜 모두 같다고 할 수 있습니다.

① UDP(User Datagram Protocol: 사용자 데이터그램 프로토콜)

UDP는 단지 배달된 패킷의 처리와 수취 확인의 기능만을 가지고 있습니다, 그 외의 처리부분은 상위 계층의 프로토콜에 의존하게 되는 비연결형 프로토콜입니다. UDP는 비연결형 프로토콜로서 패킷의 수속이나 결정을 생략함으로서 많은 처리로 인한 부담을 줄일 수 있게 됩니다.

② TCP(Transmission Control Protocol: 전송 제어 프로토콜)

TCP는 UDP 프로토콜에 비해, 결정이나 수속이 많은 프로토콜로 목적지 확인 이외에 패킷의 흐름이나 전송 순서 확인 등 많은 기능을 지원하여 패킷 처리의 신뢰성이 높습니다.

3. TCP와 UDP의 비교

▶ UDP

① 단일 네트워크의 폐쇄된 환경에서 전송의 신뢰성보다 고속 처리가 필요할 경우
② 프로토콜 부분에 크게 신경쓰지 않아도 애플리케이션 사이의 신뢰성을 유지할 수 있는 경우
③ 프로그램 크기를 작게 억제해야 하는 경우에 UDP 프로토콜을 선택한다.

▶ TCP

① 다중 네트워크를 경유하는 경우
② 전송 속도보다는 다소 늦더라도 높은 신뢰성이 필요하고 자체적으로 신뢰성을 유지하기 어려운 경우

다음은 UDP와 TCP를 간략히 비교해 본 표입니다.

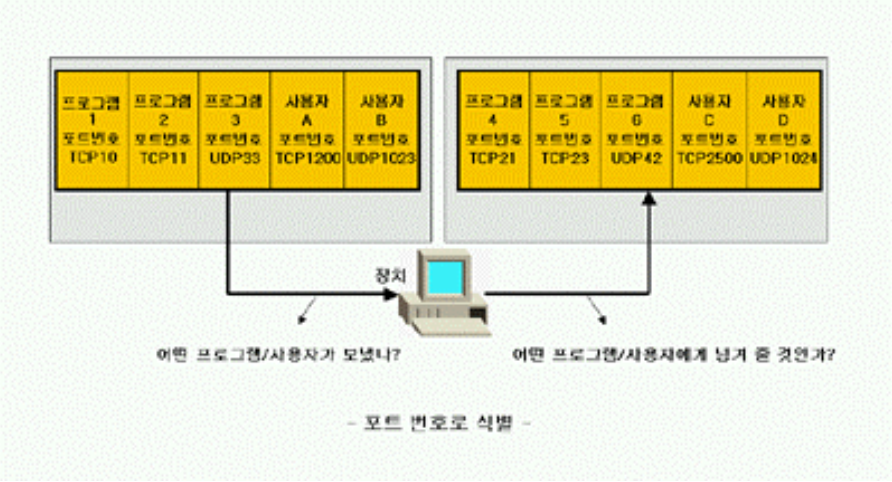
	UDP	TCP
신뢰성	낮음	높음
전송속도	높음	낮음

4. TCP/UDP에서의 포트 번호

네트워크상의 여러 장치들과 프로그램들은 한사람의 소유가 아닌 여러 사람들이 함께하는 공유된 것들입니다. 이처럼 네트워크상에서 두 대 이상의 장치로 통신을 하게 될 때 데이터를 어느 장치까지 운반해야 할 것인지는 IP 주소에 의해 결정이 됩니다.

이들 장치에서 패킷의 송수신을 처리할 때 한 사람만이 네트워크상의 장치를 독점한다면 상관없겠지만 여러 사용자가 하나의 장치를 공유할 경우 그 패킷의 처리를 장치에 공유된 어떤 사용자 또는 어떤 프로그램에 맡겨야 하는지를 결정해야만 합니다.

이처럼 프로그램의 처리와 사용자 지정에 관한 식별을 하기 위하여 포트 번호가 필요하게 됩니다. 이때 각각의 장치들은 누가 또는 어느 프로그램이 보낸 것인지를 식별하기 위해 송신쪽 포트 번호(Source Port Number)와 누가 또는 어느 프로그램이 수신해야 하는지를 식별하기 위한 목적지 포트 번호(Destination Port Number)가 규정되어 있습니다.



4. 포트 번호의 결정

네트워크 상에서 각각의 장치간에 통신을 하기 위하여 포트 번호를 결정해야 하는데 그 방법에는 공식 기관에 의해 지정된 공식 지정 포트 번호와 각각의 장치에 의해 동적으로 포트 번호가 할당되는 방법이 있습니다.

① 공식 지정 포트 번호

일반적으로 애플리케이션(TELNET, FTP등)을 사용하여 통신할 경우, 공식 기관이 각 애플리케이션에 대하여 지정한 포트 번호를 사용하는 방법입니다. 이렇게 공식화된 포트 번호를 유명 포트 번호(Well-Known Port Number) 또는 유명 포트 할당(Well-Known Port Assignment)라고 합니다.

아래는 지정된 포트 번호를 나타낸 표입니다.

포트 번호	키워드	내 용
0	-	예약
1	TCP MUX	TCP 멀티플렉서(다중화기)
5	RJE	원격 작업 항목
7	ECHO	반향
9	DISCARD	삭제
11	SYSTAT	활성 사용자
13	DAYTIME	주간
15	NETSTAT	네트워크 상태 프로그램
17	QOTD	해당일 인용문
19	CHARGEN	문자 발생기
20	FTP-DATA	파일 전송 프로토콜(데이터)
21	FTP	파일 전송 프로토콜
23	TELNET	단말기 연결
25	SMTP	간이 메일 운반 프로토콜
37	TIME	시간
42	NAME	호스트 이름 서버
43	WHOIS	누구
53	NAMESERVER	도메인 이름 서버
77	RJE	개인 RJE 서비스
79	FINGER	Finger
93	DCP	장치 제어 프로토콜
95	SUPDUP	SUPDUP 프로토콜
101	HOSTNAMES	NIC 호스트 이름 서버
102	ISO-TSAP	ISO-TSAP
103	X400	X.400 메일 서비스
104	X400-SND	X.400 메일 보내기
111	SUNRPC	SUN 원격 절차 호출
113	AUTH	인증 서비스
117	UUCP-PATH	UUCP 경로 서비스
119	NNTP	USENET 뉴스 전송 프로토콜
129	PWDGEN	암호 발생기 프로토콜
139	NETBIOS-SSN	NETBIOS 세션 서비스
160-223	-	예약

TCP 포트의 번호

포트 번호	키워드	내 용
0	-	예약
7	ECHO	반향
9	DISCARD	삭제
11	SYSTAT	활성 사용자
13	DAYTIME	주간
15	NETSTAT	Up 또는 NETSTAT는 누구?
17	QOTD	해당일 인용문
19	CHARGEN	문자 발생기
37	TIME	시간
42	NAME	호스트 이름 서버
43	WHOIS	누구
53	NAMESERVE	도메인 이름 서버
67	BOOTPS	부트스트랩 프로토콜 서버
68	BOOTPC	부트스트랩 프로토콜 클라이언트
69	TFTP	평범한 파일전송
111	SUNRPC	Sun Microsystems RPC
123	NTP	네트워크 시간 프로토콜
161	SNMP	SNMP 넷 모니터
162	SNMP-TRAP	SNMP 트랩
512	BIFF	UNIX 콤샛 통신
513	WHO	Unix Rwho 디먼
514	SYSLOG	시스템 로그
525	TIMED	시간 디먼

UDP의 포트 번호

② 동적 할당 방법

특정 장치에서 통신 처리를 필요로 할 때 장치 중 패킷간의 충돌을 방지하기 위해 포트 번호를 생성하여 사용하는 방법을 말합니다.

컴퓨터 상에서 계산이나 통신을 할 때 OS가 제공하는 작업 환경을 프로세스라고 합니다. 보통 여러 사용자가 동시에 이용하는 장치는 OS에 예약된 프로세스가 아닌 것은 필요에 따라(사용자 요청에 따라) 프로스세를 생성하가나 삭제하게 됩니다.

일반 사용자들은 이 OS 예약 프로세스의 제어를 받으면서 동적으로 생성된 프로세스 중에서 시작되어 송수신 처리를 하므로 이를 동적인 할당이라 말하는 것입니다.

그런데 이 포트 번호는 각각의 프로토콜마다 규정되어 있으므로 서로 다른 프로세서에서 동시에 같은 포트 번호가 동적으로 할당되어 사용될 때 문제가 발생합니다. 예를 든다면 같은 장치의 동일한 포트 번호를 TCP와 UDP가 동시에 사용할 경우라 할 수 있습니다. 이와 같이 경우에는 포트 번호를 할당받아 한 프로세스만이 하나의 포트를 사용할 수 있으며 다른 프로세스는 다른 포트 번호를 사용하게 됩니다.

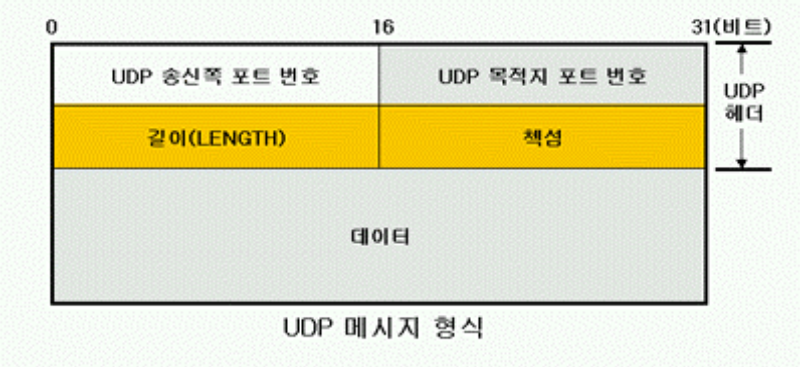
③ 프로토콜마다 포트 번호가 규정되어 있는 이유

프로토콜마다 포트 번호가 규정되어 있는 이유는 각각의 포트 번호를 읽기 전에 전송 계층의 프로토콜이 무엇인지를 먼저 판별하기 때문입니다. 그 후 포트 번호의 처리는 각각의 프로토콜(TCP 또는 UDP)에 의존하므로 같은 포트 번호라 해도 처리가 독립적으로 이루어지기 때문입니다.

UDP

(User Datagram Protocol)

이젠 UDP에 대해 설명해 보도록 하겠습니다.
다음은 UDP 패킷에 대한 구조를 나타내고 있습니다. 데이터를 제외한 부분이 바로 UDP 헤더에 해당하며, 이는 송신측 포트 번호, 목적지 포트 번호, LENGTH(길이) 및 체섬으로 구성되어 있습니다.



UDP 메시지 형식

① 송신측 포트 번호와 목적지 포트 번호
이 필드에는 목적지와 송신측 각각에 대한 포트 번호가 설정되어 있으며 16비트의 영역으로서 16비트의 정수를 기억하게 됩니다. 또한 송신측 포트 번호는 지정되지 않을 수도 있는 선택값입니다. 이 때 포트번호는 지정되지 않았다는 뜻으로 ""0""을 사용하게 됩니다. 이것은 답신이 필요 없는 통신을 하게 될 때 이용하게 됩니다.

② LENGTH(길이)
LENGTH 필드는 UDP 헤더의 길이와 사용자 데이터 길이를 더한 옥텟(바이트) 단위로 설정되어 있습니다. 그러므로 적어도 8이 설정되며, 8을 설정하면 UDP 헤더만 있고 사용자 데이터는 없다는 뜻입니다. 그것은 UDP 헤더의 길이만이 8옥텟을 사용하기 때문입니다.

③ 체섬과 UDP 허위 헤더
체섬은 선택사항으로 계산하지 않아도 통신이 가능합니다. 이 것은 고속통신을 할 경우 프로토콜의 오버헤드를 최소한으로 억제하기 위한 것입니다. 하지만 U에 통신의 신뢰성을 높이기 위한 유일한 방법이므로 대부분 UDP의 체섬이 사용되고 있습니다.
UDP에서는 체섬을 계산할 때 헤더와 사용자 데이터 앞에 직접 통신과는 무관한 UDP 허위 헤더를 추가하고, 끝 부분에는 ""0""만으로 구성되는 바이트(옥텟)로 자리를 채워 16비트의 배수가 되도록 만들어 줍니다. 이 때 UDP 헤더의 체섬 필드에는 ""0""으로 채워지게 됩니다.
이 때 사용되는 UDP 허위 헤더에는 아래의 그림과 같이 목적지의 IP 주소, 송신쪽의 IP 주소, IP 헤더에 사용되는 UDP의 프로토콜 유형 번호(17), UDP 헤더와 사용자 데이터만을 보내주게 됩니다. 수신쪽은 UDP 헤더와 사용자 데이터를 보낸 후, IP 헤더로부터 송신쪽과 목적지 IP 주소를 추출하고, UDP 허위 헤더를 작성한 후, CPR섬을 다시 계산하여 확인하게 됩니다.

UDP 하위 헤더

여러 장치와 통신할 때 수신쪽과 송신쪽 모두 양쪽의 포트 번호만으로 통신 목적지를 규정하는 경우, 유명 포트 번호와 같은 번호를 사용하지 않고 통신할 때에는 아무래도 동적으로 할당된 포트 번호에 의존하게 된다. 동적으로 할당된 이 포트 번호는 그 번호를 할당하는 장치마다 따로 관리되므로, 이는 여러 장치와 통신할 경우에 우연히 같은 번호가 존재할 수 있기 때문입니다. 그래서 IP 주소까지 포함한 형태로 양쪽 통신의 최종 프로세스를 인식해야 합니다.

동적으로 할당된 포트 번호는 그 번호를 할당하는 쪽 장치의 고유 번호로 되어 있습니다. 그러므로 여러 장치와 동시에 통신할 경우에는 IP 주소를 상대방 장치에 맞는 방법으로 사용하고 그 장치의 프로세스에 맞는 포트 번호를 사용하여 오류가 발생되지 않도록 합니다.

TCP

1. TCP 개요

이젠 TCP에 대해 알아보도록 하겠습니다. 여기서 말하는 TCP란 Transmission Control Protocol(전송제어 프로토콜)로서 패킷 전송에 대한 흐름 및 제어를 하는 4계층의 프로토콜을 말합니다.

2. TCP의 기능

TCP는 패킷의 송수신 과정에서 패킷의 손상, 유실, 중복 및 도착 지연 등의 문제를 검출하고 수정하여 통신의 신뢰성을 높여주게 됩니다. 또한 TCP는 체섬, 응답확인, 재전송, 연결 관리, 창 제어 등을 통해 신뢰성을 보장해 줍니다.

2.1 연결 관리(가상 회로)

가상 회로란 각종 장치나 회선 또는 네트워크 중에서 통신을 하는 두 애플리케이션이 정보를 전달하기 위해 독점으로 사용할 수 있는 가상의 통로를 말합니다. 즉 가상 회로에서 통신을 수행하는 애플리케이션에 파이프와 같은 가상의 통로를 만들고 통신을 수행하는 애플리케이션은 파이프 출입구에 대해 송수신을 하기만 하므로 자동으로 통신을 할 수 있게 되는 것입니다.

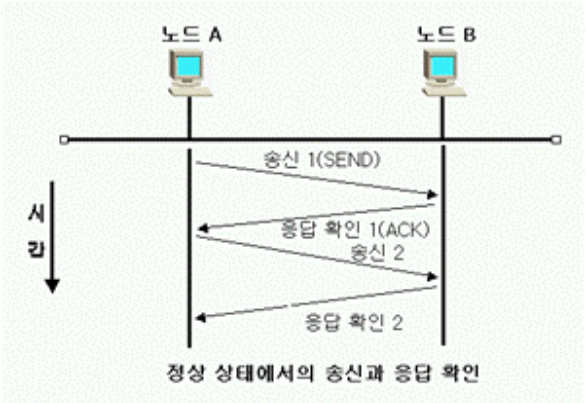
이 가상 회로를 실현하기 위해서는 연결의 확립과 종료 그리고 유지라는 작업이 필요하게 되는데 TCP는 이러한 연결을 관리하게 됩니다. 다시 말하면 TCP 헤더에 들어 있는 제어용 비트와 필드를 이용하여 접속 요청 및 확인, 끊기 요청 및 확인, 응답 확인 등의 처리를 수행하게 됩니다.

2.2 응답 확인(Acknowledgement)

응답 확인이란 송신된 데이터에 대하여 수신측이 도착한 사실을 송신측에 알려주는 것을 말합니다. 즉, 송신측은 송신을 한 후에 이 응답 확인을 기다리고, 응답 확인을 수신한 후, 그 다음 데이터를 보내게 됩니다.

응답확인에 대한 다음의 예를 들어보겠습니다.

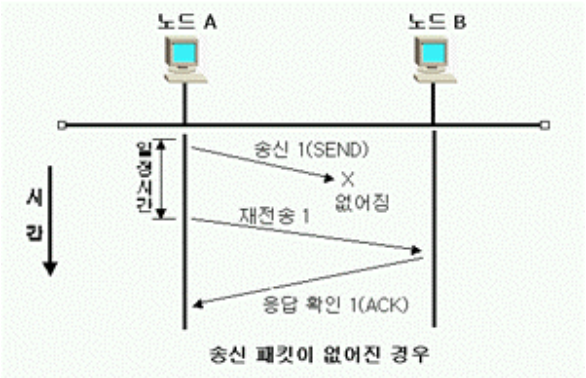
① 정상 상태에서의 송신과 응답



정상 상태에서의 송신과 응답 확인

정상 상태일 경우에는 노드 A가 B로 송신하고, 그에 대해 노드 B가 노드 A로 응답확인을 보내게 됩니다.

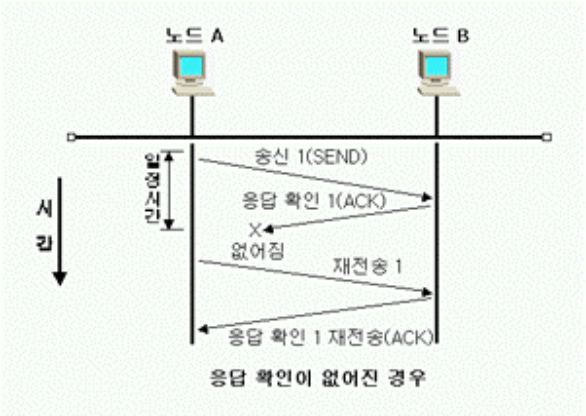
② 송신 패킷이 없어진 경우



송신 패킷이 없어진 경우

노드 A가 송신한 패킷에 대해 노드 B에서 일정시간 내에 응답이 없으면 노드 A는 패킷이 없어진 것으로 판단하고 재전송을 하게 됩니다.

③ 응답 확인이 없어진 경우

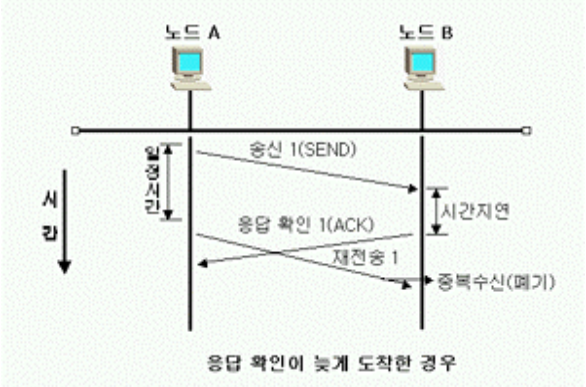


응답 확인이 없어진 경우

노드 B의 응답 확인1이 노드 A에 도착하지 않은 경우에도 노드 A는 재전송을 하게 됩니다. 그러나 노드 B는 이미 송신 1을 수신했으므로 재전송 1을 폐기하고 다시 한번 응답 확인 1을 보내게 됩니다.

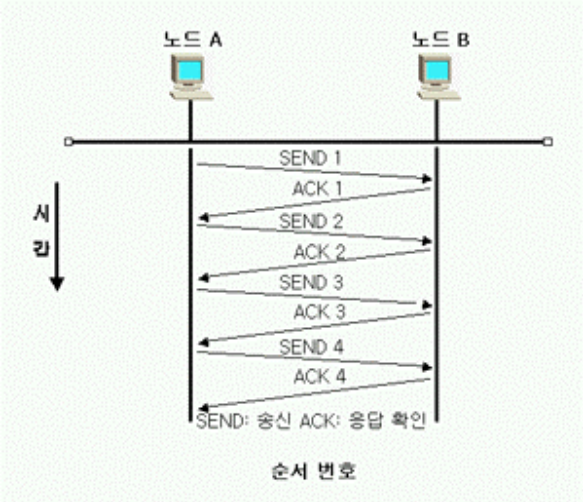
2.3 순서 번호에 의한 관리

이번에는 응답 확인이 늦게 도착하는 경우를 살펴보겠습니다.



응답 확인이 늦게 도착한 경우

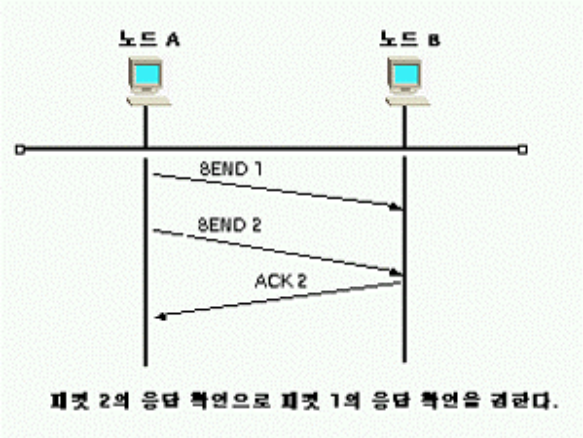
송신측은 패킷을 보낸 후 응답 확인 패킷이 도착하지 않으면 재전송 처리를 하게 되는데 이 때 수신측의 어떠한 이유로 응답이 지연이 되어 응답 확인 패킷이 뒤늦게 도착할 때에도 일정 시간 동안 기다리던 송신측은 응답 확인이 없으므로 재전송을 하게 됩니다. 이때 수신측은 상위 애플리케이션의 혼란을 방지하고 연결의 정확성을 유지하기 위해 중복으로 재 전송된 패킷을 식별하고 폐기합니다. 여기서 수신된 패킷에 대한 중복 여부를 판단하기 위한 방법으로 등장한 것이 순서 번호라는 것입니다.



처음 연결을 확립할 때에는 양쪽의 순서 번호를 동기 시켜야 하지만, 그 후에는 송신하는 데이터의 양에 따라 순서 번호를 더하게 됩니다. TCP 헤더에는 송신할 때 필요한 번호와 다음 번에 자신이 수신할 때 동기 시키는 번호를 넣어서 송신할 내용을 규정하게 됩니다. 이 번호를 이용하면 정확한 응답 확인 패킷을 되돌려주거나 응답 확인의 패킷이 올바른지 여부와 중복된 패킷을 판단할 수 있게 됩니다.

2.4 창 제어

창 제어란 쉽게 말하면 여러 개의 수신용 버퍼를 가지고 패킷을 한꺼번에 처리하는 것을 말합니다. 예를 들면 3개의 패킷을 동시에 처리한다고 할 때 송신측은 각 패킷에 대한 응답 확인을 기다리지 않고 3개의 패킷을 차례로 송신하게 되면 수신측은 여러 개의 버퍼로 이 패킷들을 차례로 처리한 후 순차적으로 응답 확인을 되돌려 줍니다.



그림과 같이 각각의 패킷에 대하여 응답 확인을 기다릴 때보다 전송 시간이 줄어들게 됩니다. 또한 어떤 사정에 의해 패킷 1의 응답 확인을 보내기 전에 패킷 2가 도착하게 된다면, 패킷 1의 응답 확인을 생략하고 앞에서 말한 순서 번호의 기법에 따라 패킷 2의 응답 확인으로 패킷 1의 응답 확인도 겸하게 됩니다. 이와 같이 동시에 처리할 수 있는 버퍼의 집단을 "창"이라고 합니다. 즉 동시에 전송 가능한 버퍼의 숫자를 말합니다. 만약에 창 크기가 "5"로 규정되어 있다면 5개까지의 데이터를 동시에 처리할 수 있음을 의미하게 됩니다.

창 크기로 규정되는 단위는 각각의 프로토콜마다 다르지만 TCP에서는 옥텟 단위로 처리하게 됩니다. 또한 창의 크기에 대한 표준은 없고 통시하는 각 장치의 버퍼 여유 분에 따라 달리 지게 됩니다.

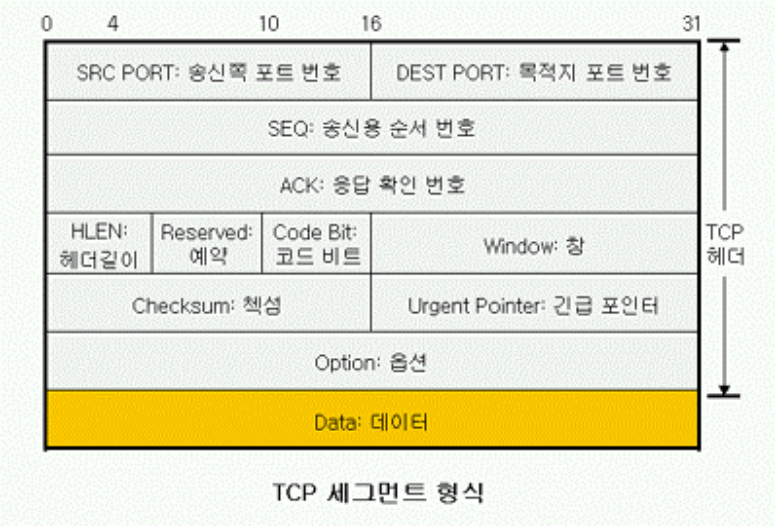
2.5 흐름 제어

일반적으로 패킷을 보내는 송신측은 상관이 없지만 자신의 의도와는 상관없이 패킷을 받게 되는 수신측은 패킷 처리에 시간을 뺏겨 수신을 할 수 없게 될 수도 있습니다. 이와 같은 경우 수신측의 장치가 수신할 수 있는 한계를 미리 송신측에 알려 주게 되는데 이러한 처리를 흐름제어라고 하게 됩니다.

TCP의 경우, TCP 헤더에 사용할 수 있는 창 크기를 나타내는 필드가 있어, 가변 길이의 창 크기를 설정할 수가 있게 됩니다. 보통은 이 필드를 크게 하여 최대한의 처리 능력으로 통신을 하게 됩니다. 하지만 어떠한 이유로 수신측이 처리하기 어려우면 이 TCP 헤더의 창 크기를 작게 하여 제어할 수 있게 됩니다.

3. TCP 세그먼트 형식

단말과 단말 사이에 데이터를 전송할 때 쓰이는 TCP의 기본 단위를 세그먼트라고 하고 이때 TCP의 통신 패킷(데이터) 형식을 TCP 세그먼트 형식이라고 합니다. UDP에서는 UDP 메시지 형식이라고 하지요. UDP와 마찬가지로 TCP 패킷의 구조 또한 데이터와 데이터를 제외한 나머지 부분인 헤더로 구성되어 있습니다.



TCP 세그먼트 형식

① 송신측 포트 번호(Source Port)
16비트의 길이로서 송신측 포트 번호를 나타냅니다.

② 목적지 포트 번호(Destination Port)
16비트의 길이로서 목적지 포트 번호를 나타냅니다.

③ 송신용 순서 번호(Sequence Number)
32비트의 길이로서 송신측이 송신하는 데이터의 단위를 나타내는 순서 번호를 나타냅니다.

④ 응답 확인 번호(Acknowledgement Number)
32비트의 길이로서 송신측 순서 번호를 가지고 있어 송신되는 패킷에 대한 재전송의 요구이나 흐름 제어 등에 사용됩니다. 예를 들어 송신측이 송신한 데이터의 순서번호와 응답 확인에서 돌아오는 응답 확인 번호가 같으면 정상적으로 통신할 수 있는 것입니다. 만약에 송신한 데이터의 순서 번호보다 낮은 응답 번호가 돌아온다면 송신측은 이를 재전송 요구를 해석하고 이 응답 확인 번호에 기재된 번호로 재전송 처리를 하게 됩니다. 또한 창 필드에 나타난 수치를 보고, 수신측이 다음 수신시 응답 확인 번호 중 앞부분부터 얼마만큼의 데이터를 받을 수 있는지를 판단하게 됩니다.

⑤ 헤더 길이(Header Length)
4바이트의 단위로서 TCP 헤더의 길이를 나타냅니다. 특별한 경우를 제외하고는 보통 ""5""로 지정되어

있습니다. 4바이트가 한 단위이므로 ""5""가 지정되어 있으면 20바이트(옥텟)까지가 TCP 헤더이고 그 나머지 부분이 데이터에 해당되는 것입니다.

⑥ 예약(Reserve)
6바이트의 길이로서 추후 확장하기 위해 준비해둔 필드를 나타냅니다. 보통은 사용되지 않으므로 ""0""으로 해 두어야 합니다.

⑦ 코드 비트(Code Bit)
6바이트 길이로서 왼쪽부터 각 비트에는 UGR, ACK, PSH, RST, SYN, FIN의 값이 규정되어 있습니다. 각 비트에 ""1""이 지정되면 다음과 같은 의미를 가지게 됩니다. 이들을 제어 플래그(Control Flag) 또는 제어 비트라고도 합니다.
▶ UGR(Urgent Flag: 긴급 플레그)
이 비트에 ""1""이 지정되면 긴급 처리를 요구하는 데이터가 들어 있다는 뜻입니다.
▶ ACK(Acknowledgement Flag: 응답 확인 플래그)
이 비트에 ""1""이 지정되면 응답 확인 번호를 사용한다는 뜻이고, ""0""이 지정되면 응답 확인을 하지 않는다는 뜻입니다.
▶ PSH(Push Flag: 푸시 플레그)
이 비트에 ""1""이 지정되면 TCP가 받은 데이터를 바로 윗층의 애플리케이션에 보내지고 ""0""이 지정되면 받은 데이터를 상위 어느 층의 애플리케이션에 보낼 것인지를 TCP의 판단에 의존하게 됩니다.
▶ RST(Reset Flag: 초기화 플래그)
재전송을 해도 통신이 회복되지 않는 경우, 즉 가상 회로를 유지할 수 없는 경우, 일방적으로 TCP의 가상 회로를 끊는 것을 초기화라고 하는데, 이 비트에 ""1""이 지정되면 어떤 원인에 의해 통신 장애가 발생하여 제어 할 수 없는 것으로 판단하고 TCP의 가상 회로를 강제로 끊게 됩니다.
▶ SYN(Synchronize Flag: 동기화 플래그)
이 비트에 ""1""이 지정되면 순서 번호를 TCP 헤더에 나타난 송신용 순서 번호로 초기화하여 가상 회로를 확립하게 됩니다.
▶ FIN(Fin Flag: 종료 플레그)
이 비트에 ""1""이 지정되면 송신측이 보낸 데이터가 종료되었다는 것을 나타냅니다. 그러나 수신은 계속해서 가능하게 됩니다. 예를 들어 정상 종료를 하게 될 경우 종료를 요청하는 쪽에서 먼저 FIN을 지정하고 TCP 헤더에 실어 종료 요청을 보내게 됩니다. 이 요청을 받은 쪽에서는 종료 처리를 하고 필요한 데이터를 보낸 후, 송신할 데이터가 없어졌을 때, FIN을 지정한 TCP 헤더를 되돌려 주어 모든 처리를 종료하게 되는 것입니다.

⑧ 창(Window)
16비트의 길이로서 같은 TCP 헤더에 포함된 응답 확인 번호로 나타난 위치로부터 어느 정도의 데이터를 수신할 수 있는지를 알리기 위한 것으로 여기에 나타난 데이터 양을 초과하여 통신하는 것을 허용하지 않게 됩니다.

⑨ 체섬(Checksum)
TCP의 체섬도 UDP에 사용되는 체섬과 거의 같습니다. UDP와 마찬가지로 허위 헤더를 사용하여 계산을 하게 됩니다. 체섬을 계산할 때, TCP 헤더와 사용자 데이터 앞에 통신과 직접 관계가 없는 허위 헤더를 추가하고, 끝에 ""0""만으로 구성된 바이트(옥텟)을 패팅(자리 채우기)하여 전체가 16비트의 배수가 되게 합니다. 이 때 TCP 헤더의 체섬 필드는 ""0""으로 패딩하게 됩니다.
이 때 이용되는 허위 헤더에는 아래 그림과 같이 목적지의 IP 주소와 송신측의 IP 주소, IP 헤더로 사용되는 TCP의 프로토콜 유형 번호(6), TCP 헤더 및 사용자 데이터의 길이를 더한 값이 들어 있게 됩니다.

TCP의 체섬을 계산할 때 이용하는 허위 헤더

송신측은 이 TCP 허위 헤더를 작성하고 체섬을 계산한 후, 결과를 TCP 헤더에 넣어 TCP 헤더와 사용자 데이터만을 보내게 됩니다.
수신측은 TCP 헤더와 사용자 데이터를 수신한 후 IP 헤더로부터 송신쪽과 목적지 IP 주소를 추출하고 허위 헤더를 작성한 후 체섬을 다시 계산하여 확인하게 됩니다.
이 체섬의 계산에 기술되어 있는 수치와 계산 결과가 다르지 않으면, 그 TCP 헤더와 데이터는 통신을 해치지 않고 도착한다는 것을 보증할 수 있게 되는 것입니다.

⑩ 긴급 포인터(Urgent Pointer)
16비트 길이로서 앞에서 설명한 코드 비트(Code Bit)의 URG에 ""1""이 지정되어 있을 때 한하여 쓰여지게 됩니다. 여기에 나타난 수치는 긴급 처리를 해야하는 데이터가 들어 있는 데이터 필드를 가르키는 포인터로 사용됩니다. 즉, TCP 헤더 뒤에 추가된 데이터의 앞부분(송신용 순서 번호를 나타내는 장소)으로부터 이 긴급 포인터에 나타난 수치만큼의 바이트(옥텟)가 데이터로 처리가 되는 것입니다.

⑪ 옵션(Option)
옵션은 통신의 세부 사항을 조정하기 위해 사용됩니다. 일반적으로는 거의 사용되지 않으며, 옵션이므로 지정하지 않아도 통신을 할 수 있습니다. 이 옵션의 지정 여부는 사용자가 선택하며, 필드 크기는 임의로 되어 있지만, 옵션 필드는 전체가 32비트의 배수가 되도록 지정해야만 합니다.

라우팅 프로토콜

1. 라우팅 프로토콜의 개요

현재 구축되어 있는 네트워크는 LAN이나 WAN 등의 복잡한 구성으로 되어 있습니다. 이 때 복잡한 네트워크 구성을 통해 자신이 원하는 상대방으로 데이터를 전송하기 위해 ""라우팅(경로 설정)""이라는 메커니즘이 필요하게 되는데, 네트워크 상에서 이 ""라우팅(경로 설정)""을 지원하는 것이 바로 라우터입니다.

이 라우터는 네트워크 계층의 프로토콜이나 전송 계층의 프로토콜의 일부를 해석해서 전송하게 됩니다. 이것은 네트워크 계층에 규정되어 있는 주소(IP 주소 등)를 판단하게 됩니다. 이 때문에 사용중인 프로토콜 이외의 패킷은 폐기하게 되며, 불필요한 방송 및 다중 방송 패킷 등 일괄 통신 패킷에 대해서도 인접 세그먼트에 전송해야 할지 여부를 판단하게 됩니다. 바로 이 라우터를 사용함으로써 능률적인 데이터 전송이 가능하게 되는 것입니다. 또한 라우터는 네트워크 계층의 주소를 관리함으로써 다른 세그먼트에 전송해야 할지 여부를 판단하게 됩니다. 즉, 라우터는 다른 네트워크로 향하는 통신을 할 때에만 패킷을 다른 네트워크에 보내게 되며, 자신의 네트워크 내에는 영향을 미치지 않게 되는 것입니다. 이러한 이유 때문에 라우터를 ""게이트웨이(하나의 네트워크로 통하는 출입구)""라고도 합니다.

라우터를 사용할 때의 제약은 라우터에서 결정된 프로토콜 외에는 라우터를 통과할 수 없다는 것입니다. 또한 네트워크 계층의 주소가 정확하지 않으면 통과할 수 없으며, 라우터 본체는 어느 경로를 통해 전송해야 하는지 인식해야만 하는데 이와 같은 정보를 ""라우팅(경로 설정) 정보""라고 합니다.

라우팅 설정 방법에는 라우터 자체에서 자동으로 인식하는 동적 경로 설정 방법과 관리자가 정의하는 정적 경로 설정의 두 가지 방법이 있습니다.

2. 라우팅 프로토콜

사용하는 프로토콜에 따라 라우팅(경로설정) 정보는 서로 다르게 취급됩니다.

라우팅 정보를 여러 라우터 사이에 자동으로 주고받을 경우에는 그 프로토콜이 일치해야 합니다. 고성능 라우터에서는 여러 경로 설정 프로토콜을 동시에 사용하여 각각의 프로토콜을 일치시키는 기능도 가지고 있으며, 거기에 더해 보안 관리 기능까지 지원하고 있습니다. 또한 특정 주소에 한하여 라우터의 통과시킬지 여부를 설정할 수도 있습니다

라우터 및 일부 컴퓨터는 어느 IP 네트워크가 어디에 존재하는지, 또는 데이터를 이 IP 네트워크에 중계하는 경우 어느 기계에 전송해야 하는지에 관한 정보를 관리하는 라우팅 테이블(경로 설정표)를 가지고 있습니다. 이 데이터를 참고하여 경로 설정에 의한 통신을 수행하게 되는 것입니다.

따라서 각 라우터는 자신이 직접 접속되어 있는 지역 IP 네트워크의 정보를 라우터와 교환하여 적절한 라우팅 테이블을 작성하게 되는데, 이와 같은 IP 네트워크에 관한 정보를 교환하기 위한 프로토콜을 ""라우팅 프로토콜""이라고 합니다.

3. IP 주소와 경로 설정

네트워크를 사용하여 통신할 경우, 목적지의 IP 주소가 어느 IP 네트워에 속하는지는 IP 주소를 보고 판단할 수 있게 됩니다. 그러나 패킷을 그 IP 주소가 실제로 속하는 IP 네트워크에 보내려면 전송 목적지라는 정보가 필요하게 되는데 이러한 정보를 가지고 있는 것이 바로 라우팅 테이블이며 이를 형성하기 위해 정보를 교환할 때 이용되는 것이 바로 라우팅 프로토콜입니다.

통신을 할 때에는 반드시 이 라우팅 테이블을 참조하여 그것에 해당되는 IP 네트워크의 정보를 추출하고 어느 중계 장치에 패킷을 넘겨줄지를 판단하게 됩니다. 따라서 라우팅 테이블에는 올바른 경로 설정 정보가 들어 있어야 하며, 만약 잘못된 경로 설정 정보가 들어 있으면 통신이 정상적으로 동작하게 되어 문제가 발생될 수 있게 됩니다.

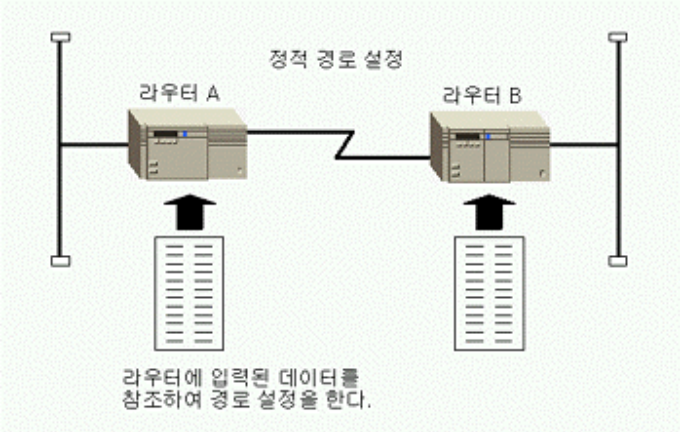
4. 경로 설정에 관한 2가지 방법

IP 네트워크의 경로 설정에 관한 방법에는 정적 경로 설정방법과 동적 경로 설정 방법 2가지가 있습니다.

① 정적 경로 설정 방법

정적 경로 설정 방법은 라우터 자체 내에서 경로 설정 정보를 고정적으로 설정하는 방법을 말합니다.

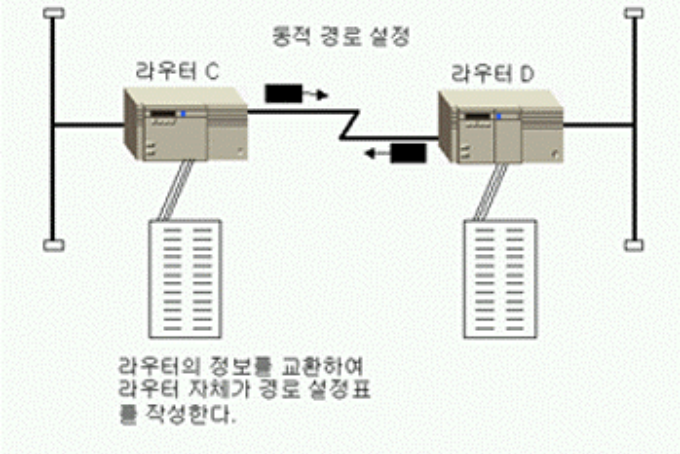
정적 경로 설정 방법은 네트워크 전반에 경로 설정 정보를 관리자가 직접 입력해 주어야만 합니다. 이를테면 10개의 IP 네트워크가 있다면 10개 관한 경로 설정을 관리자가 직접 설정해 주어야만 하는 것이죠. 또한 여기에 하나의 네트워크가 하나 더 추가 된다면, 추가된 라우터에 전체 경로 설정 정보를 입력해 주어야만 하며, 이미 설치되어 있는 10대의 라우터에도 새로 입력된 IP 네트워크 정보를 입력해 주어야만 합니다. 이 때문에 관리자의 부담이 커지게 되는 단점을 가지게 됩니다.



② 동적 경로 설정 방법

동적 경로 설정은 라우터 사이에서 라우팅(경로 설정) 프로토콜을 작동시켜 경로 설정 정보를 주고 받아, 경로 설정 정보를 자동으로 설정해 주는 방법을 말합니다.

동적 경로 설정 방법은 각각의 라우터가 접속되어 있는 IP 네트워크의 정보와 작동시킬 라우팅(경로 설정) 프로토콜에 관한 설정만 입력하면 됩니다. 즉, 정적 경로 설정 방법과는 달리 또 다른 라우터가 추가 된다고 해도 추가된 라우터에만 필요한 최소한의 정보만 입력해 주면 나머지 라우터들은 라우팅 정보를 교환하여 자동으로 전체 네트워크의 정보를 재 설정해 주게 되는 것입니다.



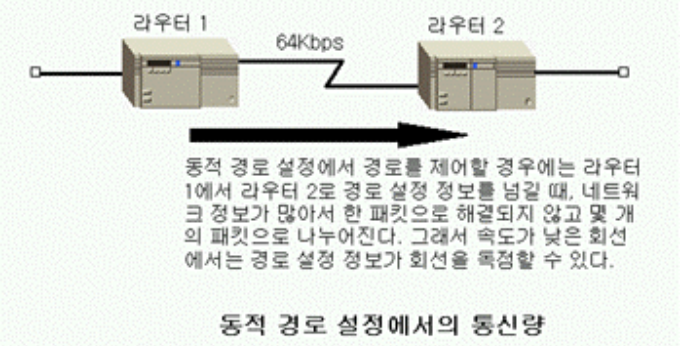
③ 정적 경로 설정 방법과 동적 경로 설정 방법의 장단점

이상에 살펴 본 바와 같이 정적 경로 설정 방법과 동적 경로 설정 방법에 관한 장/단점을 살펴 보면 다음과 같습니다.

	장 점	단 점
정적 경로 설정	•경로 정보가 변하지 않음 •통신량이 발생하지 않음	•설정 관리가 어려움 •동적 경로 변경이 불가능
동적 경로 설정	•설정이 쉬움 •동적 경로 변경이 가능함	•경로 정보가 유동적임 •통신량이 발생

위와 같이 본다면 당연히 동적 경로 설정 방식이 정적 경로 설정 방식에 비해 우수한 점이 많을 것이라 생각이 들것입니다. 하지만 네트워크의 전반적인 환경에 따라 이 두가지 방법 중 하나를 선택해서 사용하게 됩니다.

예를 들어 정적 경로 설정 방법은 각 라우터마다 모든 정보를 입력해야 하는 단점이 있는 반면에 네트워크 상에 라우팅 프로토콜을 작동하지 않아도 되기 때문에 라우터 사이에 경로 설정에 관한 데이터 교환이 없어도 되므로 네트워크 상에 트래픽이 발생하지 않는다는 장점을 가지고 있습니다. 이에 비해 동적 경로 설정 방법은 각각의 라우터마다 라우팅(경로 설정) 알고리즘이 적용되므로 경로 설정에 관한 데이터 교환으로 인하여 네트워크 상에 트래픽이 발생하게 됩니다. 이 것은 소규모 네트워크에서는 큰 문제가 안되지만, 대규모 네트워크에서는 경로 설정에 관한 정보 교환 트래픽으로 인하여 네트워크 상에 무리가 생길 수 있는 문제를 안게 됩니다. 또한 LAN과는 비교되지 않을 만큼 작은 용량을 가지는 WAN쪽의 회선을 경로 설정 관한 정보를 가진 트래픽으로 인하여 일반 통신 데이터의 전송에 방해가 될 가능성도 높아지게 됩니다. 따라서 통신량 측면에서 생각해 본다면 정적 경로 설정 방법이 더 났습니다.



동적 경로 설정에서의 통신량

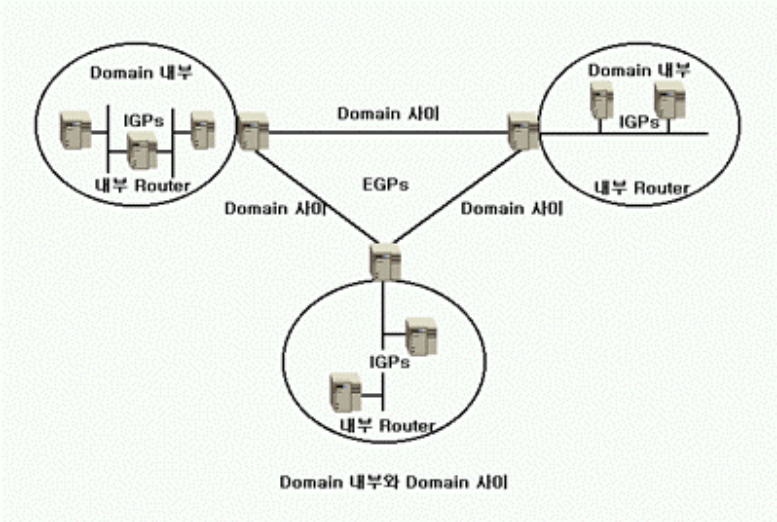
하지만 최근에는 라우팅 프로토콜의 발전과 더불어 관리가 쉬운 동적 경로 설정 방법을 이용하는 경우가 훨씬 더 많아지고 있습니다. 이를 테면 목적지까지 여러 경로를 확보해 놓고, 이용하는 경로에 장애가 발생하면 다음 경로로 변경하는 이중 시스템을 구축하므로서 시스템의 안전성을 높이고 있습니다. 이와 같은 경우 동적인 경로 설정이 불가능한 정적 경로 설정 방식보다는 동적 경로 설정 방식이 사용되게 되는 것입니다.

LGP와 EGP

1. 영역 구분으로 본 Routing Protocol 2가지

Routing Protocol의 동적 경로 설정과 정적 경로 설정 이외에 통신 영역으로 구분 짓는 방법이 있습니다. 그 두 가지 방법이 바로 IGP(Interior Gateway Protocol: 게이트웨이 역내 프로토콜)과 EGP(Exterior Gateway Protocol: 게이트웨이 역외 프로토콜)입니다.

이 방식으로 네트워크의 발전에 따라 전체 네트워크 정보의 증가에 따른 일괄적인 관리가 불가능하게 되어, 네트워크의 효율적인 관리를 위해 그 경로를 계층화시키고 네트워크 경로를 두 가지로 분류하여 경로 설정 대한 제어를 하게 되었습니다. 그 방법이 바로 게이트웨이 영역 내 경로를 설정 프로토콜인 IGP와 게이트웨이 영역 밖의 경로 설정 프로토콜인 EGP입니다.



① IGP(Interior Gateway Protocol: 게이트웨이 역내 프로토콜)

라우터로 상호 접속되어 있는 여러 개의 네트워크 집합을 "도메인(Domain)" 혹은 "자율 시스템(AS)"라고 합니다. 같은 Domain 내에 존재하는 라우터는 Domain 내부 라우터가 되고 Domain 외부에 존재하는 라우터는 Domain 외부 라우터가 되는 것입니다. 여기서 Domain 내부 경로 설정에 관한 프로토콜이 IGP이며 Domain간 경로 설정에 관한 프로토콜이 EGP입니다.

IGP는 단일 시스템의 네트워크에서 경로 제어 정보 등을 전달하는 데 사용되는 프로토콜로서 RIP(Routing Information Protocol: 경로 설정 정보 프로토콜), HELLO, IGRP(Interior Gateway Routing Protocol: 게이트웨이 역내 경로 설정 프로토콜) 등이 이에 해당 됩니다.

② EGP(Exterior Gateway Protocol: 게이트웨이 역외 프로토콜)

EGP는 시스템 사이에 경로 설정 정보 등을 교환하기 위해 사용하는 프로토콜로서 EGP(Exteior Gateway Protocol: 게이트웨이 역외 프로토콜)와 BGP(Boarder Gateway Protocol: 종속 게이트웨이 프로토콜) 등이 이에 해당 됩니다.

경로 설정 알고리즘과 경로설정 프로토콜

1. 경로 설정 알고리즘

경로 설정 알고리즘은 크게 거리 벡터(Distance-Vector)와 링크 상태(Link-States) 등 크게 두 가지로 나누어집니다. 이들 두 알고리즘 중 어느 한 방법에 따라 기존의 동적 경로 설정 프로토콜이 만들어지게 됩니다.

	동적 경로 설정	
	거리 벡터	링크상태
IGP(도메인 내부)	RIP HELLO IGRP	OSPF
EGP(도메인 외부)	-	EGP BGP

경로 설정 프로토콜의 분류

① 거리 벡터

거리 벡터 알고리즘은 각 라우터가 인접해 있는 라우터와 경로 설정 정보를 교환하여 네트워크의 구성이나 장치 배치에 관한 모든 정보, 즉 네트워크 토폴로지 관한 정보를 교환하는 구조를 말합니다.

실제로 네트워크상에서는 라우터는 인접해 있는 다른 라우터에게 새로운 정보를 받을 때마다 알려주게 됩니다. 이와 같이 반복되면 최종에는 모든 라우터가 전체의 IP 네트워크 정보를 갖게 되는 것이죠.

이 거리 벡터 알고리즘은 비교적 간단해서 비교적 쉽게 설정할 수 있습니다. 그러나 네트워크 규모가 점점 커지게 되면 네트워크 정보에 관한 프로세스 처리나 통신량이 많아져서 네트워크에 부담이 더해지게 됩니다.

거리벡터 알고리즘은 기본적으로 인접 라우터가 어떤 정보를 가지고 있는지 고려하지 않기 때문에 네트워크 토폴로지 변화에 대한 경로 설정 정보의 재편성 지연과 같은 사태에 대해 진단하지 못하는 단점을 안고 있습니다. 하나의 라우터는 네트워크 자체의 토폴로지를 알지 못하지만 인접 라우터로부터 정보를 받아 인접해 있는 각각의 네트워크 토폴로지를 판단하여 이어주는 역할을 하게 됩니다. 이때 라우터 사이에 교환되는 정보는 기본적으로 거리 정보 뿐이며, 따라서 단순하고 다루시 쉬운 반면에 장애 등의 원인을 알아내기 어렵다는 단점이 있습니다.

② 링크 상태

링크 상태 알고리즘에서는 라우터가 먼저 각각의 독립된 네트워크의 영역을 보는 방법을 결정하여, 그 정보를 다른 모든 라우터들에게 전달하는 구조를 말합니다.

각 라우터가 각 지역 환경을 방송으로 전달하므로, 모든 라우터는 네트워크 토폴로지의 완전한 정보를 알릴 수 있으며, 각 라우터는 다른 라우터가 어디에 존재하고 어디에 접속되어 있는지를 알 수 있게 됩니다.

링크 상태는 토폴로지가 변해도 한 번의 처리로 새 정보를 얻을 수 있습니다. 이 때문에 거리 벡터와 같이 시간 지연이 발생 문제가 없습니다. 그러나 링크 상태 알고리즘은 아주 복잡한 구조로 간단한 경로 설정 방법을 실현하므로 주소 배분의 설계나 장치 설정 등이 어려운 단점을 가지고 있습니다.

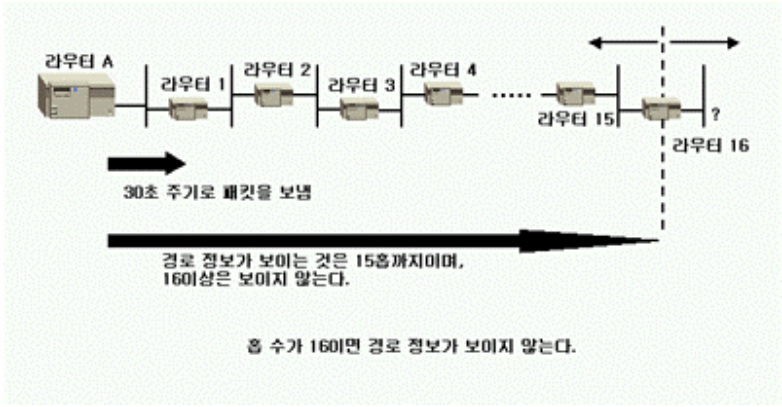
2. 주요 경로 설정 프로토콜

경로 설정 프로토콜에는 다양한 종류가 있지만 여기서는 주요 경로 설정 프로토콜에 대해 알아보도록 하겠습니다.

① RIP(경로 설정 정보 프로토콜)

RIP은 현재 가장 널리 사용되고 있는 경로 설정 프로토콜로서 경로 설정 정보의 송신 간격은 30초이며 거리 벡터형 프로토콜에 속합니다. 현재의 경로 제어에 대한 정보를 주기적으로 네트워크에 방송하는 역할을 합니다.

RIP의 경로 설정 기준은 HOP(정보가 통과한 라우터(네트워크)의 개수) 수로서 통과한 네트워크의 수를 규명하며, hop수가 "16"이면 경로가 보이지 않게 됩니다.



② HELLO

HELLO 프로토콜은 NSFnet(National Science Foundation Backbone Network)을 형성하고 있는 노드들(라우터들) 사이에서 사용되는 프로토콜로서 경로 설정 정보의 송신 간격은 15초 주기이며, 거리 벡터형 경로 설정 프로토콜에 속합니다.

입니다. HELLO 프로토콜을 사용할 때 경로를 선택하는 기준은 HOP 수가 아니라 네트워크 지연을 바탕으로 한 경로 제어 거리이며, 지연시간(거리)이 30,000밀리초에 이르면 경로 정보가 보이지 않게 됩니다.

③ IGRP(게이트웨이 역내 경로 설정 프로토콜)

IGRP는 Cisco사의 경로 설정 프로토콜로서 Cisco 라우터들 사이에서 사용되며, 경로 설정 정보의 송신 간격은 기본 값이 90주기이며, 거리벡터형 프로토콜에 속합니다.

IGRP는 RIP과 마찬가지로 인접한 Cisco 라우터간에만 경로 설정 정보를 교환합니다. 이 정보 중에는 나머지 네트워크의 요약 정보도 포함되어 있습니다.

IGRP를 사용할 때에는 척도(거리) 정보를 경로 선택 기준으로 이용하며, 이 척도 정보는 토폴로지상의 지연 시간(Delay), 매체의 대역폭(Bandwidth), 채널 점유도(Loading), 버스의 신뢰성(Reliability) 로 되어 있습니다.

토폴로지상의 지연 시간이란 부하가 걸려 있지 않은 네트워크를 가정할 경우, 버스를 통해 목적지까지 걸리는 시간을 말합니다. 물론 네트워크에 부하가 걸려 있으면 거기에 지연 시간을 더하게 됩니다. IGRP에서는 이것을 채널 점유도 값으로 추측하게 됩니다. 따라서 채널 점유도는 그 대역폭이 현재 어느 정도 사용되고 있는지를 나타내며 부하에 따라 갱신됩니다. 신뢰성은 한 시점에서의 오류율을 나타내는 것으로 패킷이 손상되지 않고 목적지에 도착한 비율을 의미합니다

이들 정보를 바탕으로 버스에 대한 하나의 척도를 계산하고 이 척도를 최적 경로 결정의 기준으로 삼게 됩니다. 따라서 IGRP는 다양한 척도 요소에 의한 영향을 고려하여 최적의 경로를 결정하게 됩니다.

④ EGP(게이트웨이 역외 프로토콜)

경로 제어 정보를 교환하는 두 IP 라우터가 서로 다른 두 AS(자율 시스템 혹은 Domain)에 속할 때 이를

역외 환경이라 부릅니다.
이와 같이 역외 환경에서 IP 라우터가 도착 가능성 정보를 다른 AS에 알리기 위해 사용하는 프로토콜이 바로 EGP로 NSFnet(National Science Foundation Backbone Network: 미 국립 기초 과학망)이나 DDN(국방 데이터망)과 같은 대규모 백본 네트워크에서 사용되고 있습니다. 각 AS는 자체 AS에 도착할 가능성을 알리기 위해 이 EGP를 사용하게 됩니다.
EGP에는 다음과 같은 세 가지 주요 특징이 있습니다.
▶ 첫째, 다른 AS에 속한 라우터에 경로에 관한 제어 정보를 교환하여 그 라우터와 인접 관계를 맺게 됩니다. 이것을 "EGP 환경(EGP-Neighbor)을 얻는다"라고 합니다. 이렇게 해서 EGP에서의 통신 중개자 및 통신 상대를 결정하게 됩니다.
▶ 둘째, 라우터가 이 EGP 환경에 대해 지속적으로 응답할지 여부를 확인하게 됩니다.
▶ 셋째, 경로 설정 정보를 넘겨줌으로써 네트워크의 정보를 정기적으로 교환하게 됩니다.

⑤ BGP(광역 게이트워에 프로토콜)
BGP는 EGP의 경로 설정 프로토콜에 포함된 EGP의 개정판프로토콜로서 링크 상태형 프로토콜에 속합니다.
EGP가 네트워크의 도착 가능성을 알리기 위한 역할만을 가지고 있는데 비해, BGP는 가중치라는 개념을 이용하여 우선 순위를 추가해 주게 됩니다. 따라서 EGP와 달리 BGP 내에서의 조작을 임으로 할 수 있는 프로토콜로서 EGP의 개정판이라고 할 수 없는 융통성있는 경로 설정 제어를 미리 정의할 수 있게 되어 있습니다.

⑥ OSPF(최단 경로 먼저 열기)
OSPF는 최근에 주목받고 있는 경로 설정 프로토콜이니다.
OSPF는 IGPDP 속하는 경로 설정 프로토콜로서 IP 서브넷이나 TOS 등의 정보를 지원합니다, OSPF는 같은 AS(도메인) 내에서 사용되며, RIP의 ?은 문제점을 해결하기 위해 고안된 링크 상태형 프로토콜입니다.
OSPF의 다음과 같은 주요 특징이 있습니다.
▶ 첫째, 서브넷 마스크를 지원 - 이 기능에 의해 기존 경로 설정 프로토콜로는 불가능했는 가변 길이 서브넷에서의 네트워크 구성이 가능해졌습니다.
▶ 둘째, 통신량을 줄이기 위해 영역이라는 개념 도입 - 영역 체계를 만들어 불필요한 경로 설정 프로토콜이나 교환을 감소시키게 됩니다.
▶ AS 번호, 영역 및 IP 네트워크라는 체계에 의해 경로 설정의 효율적인 제어

⑦ RIP2
RIP2란 RIP Version 2로서 RFC1387번으로 표준화되었습니다. 이것은 RIP1의 다양한 경험을 바탕으로 개정된 것으로, 기본적인 방식은 RIP1과 비슷하지만 다음과 같은 기능이 추가되었습니다.
▶ 다중 방송 사용
RIP1에서는 경로 설정 정보를 교환할 때 방송 패킷을 이용하지만 RIP2에서는 다중 방송을 이용합니다. 이에 따라 관계없는 네트워크나 호스트에 끼치는 영향을 줄이고 방송에 의한 통신량도 감소시킬 수 있다.
▶ 서브넷 마스크 대응
OSPF와 같이 경로 설정 정보 중 서브넷 마스크 정보도 포함시킬 수 있다.
▶ 경로 설정 도메인
OSPF의 영역과 같이 한 네트워크에서 이론적으로 독립된 여러 RIP를 사용할 수 있다.
▶ 외부 경로 태그
EGP나 BGP 등에서 얻은 경로 설정 정보를 RIP를 통해 AS에 알릴 때 사용한다.
▶ 인증키
OSPF처럼 암호를 이용하여 자신이 인식할 수 있는 암포를 가진 패킷만 수용한다. 다른 암호의 RIP 패킷은 인식하지 못한다.

OSPF의 개요

1. OSPF의 영역

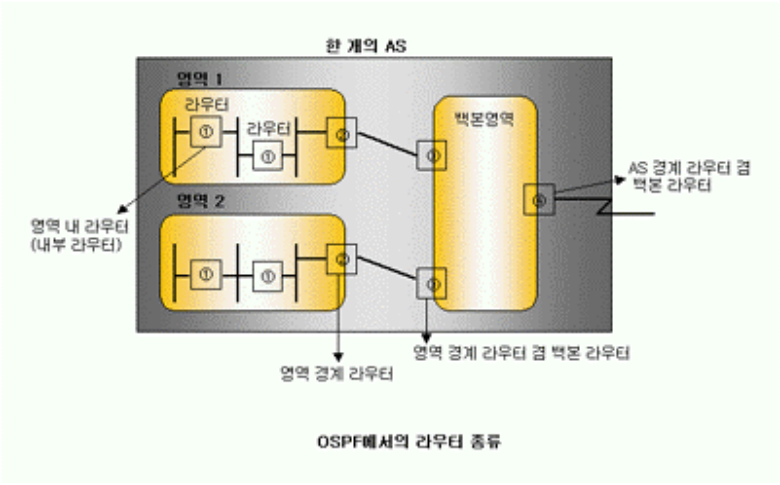
OSPF에서는 AS(자율시스템 또는 도메인) 내에서 네트워크들이나 혹은 호스트들을 묶어서 그룹화할 수 있는데, 이 그룹 내에 들어 있는 네트워크와 접속해 있는 라우터를 포함하고 있는 부분을 우리는 영역이라고 부르게 됩니다. 각각의 AS 내에는 이러한 영역이 여러개 존재할 수 있게 됩니다.

각각의 영역은 저마다 토폴로지의 데이터 베이스를 가지고 있으며, 다른 토폴로지는 보이지 않게 됩니다. 이 때문에 영역 내의 라우터는 다른 토폴로지와는 상관없이 영역내의 통신량을 억제할 수 있게 되는 것입니다.

2. OSPF에서의 백본 영역

OSPF에서는 반드시 AS 내에 백본 영역을 가지게 됩니다.

백본 영역은 영역밖에 있는 네트워크와 영역외의 네트워크에 접속되어 있는 라우터 그리고 각 영역에 속한 라우터로 구성되어 있습니다.



OSPF에서의 라우터 종류

3. OSPF에서의 라우터 종류

① 내부라우터

하나의 영역 내에 있는 네트워크와 직접 접속되어 있는 라우터를 말합니다. 백본 영역에 접속되어 있는 라우터도 이에 해당됩니다.

② 영역 경계 라우터

여러 영역에 같이 접속되어 있는 라우터를 말합니다.

③ 백본 라우터

백본 영역 내의 네트워크와 접속되어 있는 라우터를 말합니다.

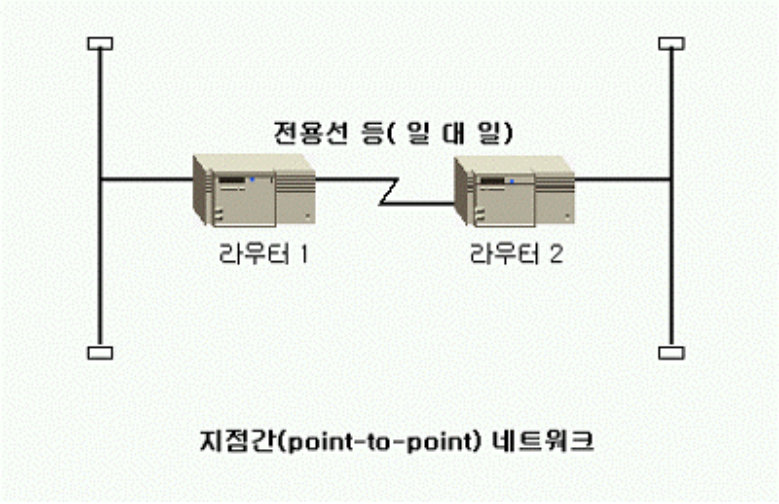
④ AS 경계 라우터

다른 AS에 속한 라우터와 경로 설정 정보를 교환하는 라우터를 말합니다. 각각의 AS 경계 라우터를 향한 경로는 AS 내의 모든 라우터들에게도 알려지게 되는데, 이 때문에 AS 내에서 AS 밖으로 경로 설정이 가능하게 되는 것입니다.

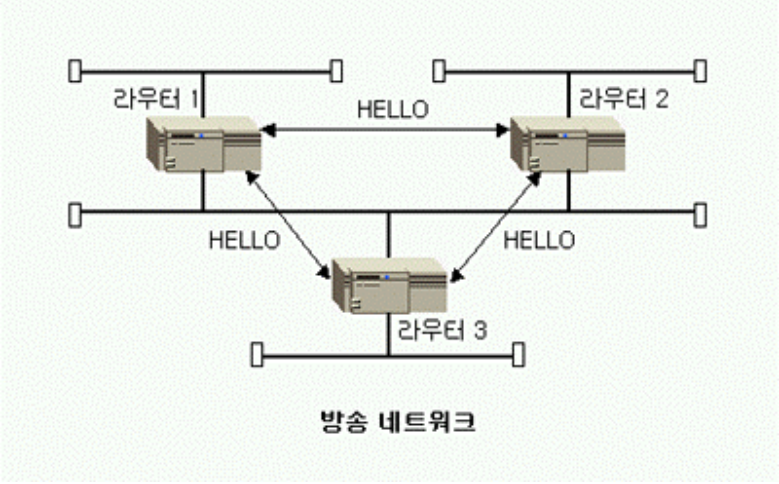
4. OSPF에서의 네트워크 종류

OSPF는 영역이라는 개념으로 네트워크 그룹을 나누게 됩니다. 영역 내의 토폴로지 변화 정보는 그 영역 내에서 흡수하여 이에 따른 통신량 등을 제어하게 됩니다. 또한 네트워크의 접속 형태에 따라서도 경로 설정 정보의 분배 방법이 달라지게 됩니다.

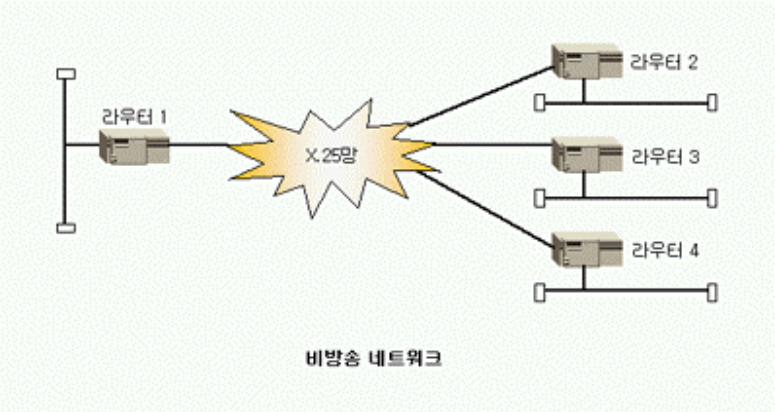
① Point-to-Point(지점간) 네트워크
2대의 라우터를 전용선 등으로 연결한 네트워크의 형태를 말합니. 통신하는 상대가 하나이므로 목적지를 고정시킨 채 정보를 전송하게 됩니다.



② 방송 네트워크
2대 이상의 라우터를 연결한 네트워크의 형태로 접속되어 있는 모든 네트워크에 경로 설정 정보를 방송할 수 있습니다.
접속되어 있는 라우터들은 HELLO 프로토콜을 사용하여 네트워크에 동적으로 인식되며, 라우터가 하나로 한정되어 있지 않기 때문에 정보를 한꺼번에 배포할 수 있게 됩니다.



③ 비방송 네트워크
2대 이상의 라우터로 연결되어 형태이지만 방송 네트워크와 달리 접속되어 있는 각각의 네트워크에 경로 설정 정보를 방송하는 기능을 가지고 있지 않아 경로 설정에 관한 별도의 설정이 필요한 네트워크 형태를 말합니다. 즉, 방송하는 기능이 없으므로 인접한 라우터에 대한 경로 설정을 별도로 지정해 주어야만 합니다.



비방송 네트워크

5. OSPF에서의 IP 서브넷화

OSPF는 경로 설정 정보에 서브넷 마스크 정보도 가지고 있게 됩니다. 따라서 각각의 경로 정보마다 서브넷의 범위가 나타나게 됩니다.

OSPF의 영역 개념은 IP 서브넷화한 네트워크를 바탕으로 모델링 된 것입니다.

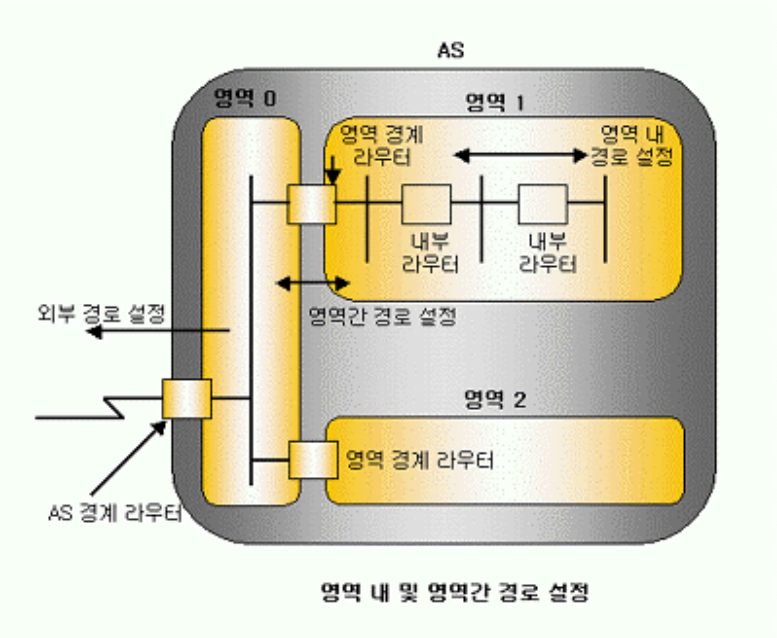
6. OSPF에서의 영역 내 경로 설정

송신지와 목적지가 같은 영역 내에 있을 때 그 경로 설정을 영역 내 경로 설정이라고 부릅니다.

라우터는 HELLO 패킷을 송수신하여 수신된 패킷을 통해 상대방 라우터의 정보를 가지 되며, 토폴로지의 데이터 베이스는 인접 관계를 가진 라우터들 사이에 일관성을 유지시키게 됩니다.

하나의 네트워크에 여러 대의 라우터가 존재하는 다중 접근 네트워크의 경우에서는 지명 라우터가 어느 라우터와 인접 관계를 맺을 것인지를 결정하게 됩니다.

이러한 인접 관계에 의해 경로 설정 패킷 전송에 관한 제어를 하게 됩니다. 경로 설정 프로토콜 패킷은 상대방 라우터하고만 송수신 됩니다.



영역 내 및 영역간 경로 설정

7. OSPF에서의 영역간 경로 설정

패킷의 송신지와 목적지가 서로 다른 영역에 있는 경우의 경로 설정을 영역간 경로 설정이라고 합니다.

영역 경계 라우터는 영역간 경로 설정에 대해서 영역 내부와 영역 외부와의 경로 설정 정보를 수신하게

됩니다.

8. OSPF에서의 외부 경로 설정

다른 AS에 관한 정보를 가진 AS 경계 라우터는 이 외부 경로 설정 정보를 AS 내의 어떤 라우터에도 전송할 수 있습니다. 단 AS 외부에 대한 경로 설정 정보를 가지고 있지 않는 영역인 스터브 영역만은 제외됩니다.

9. OSPF에서의 Stub(스터브) 영역

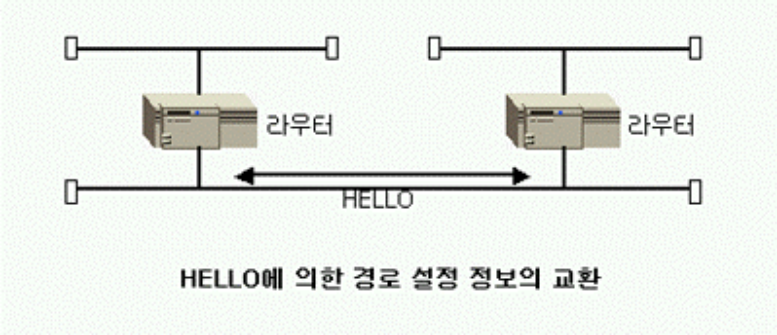
AS 외부에 대한 경로 설정 정보를 가지고 있지 않는 영역을 말합니다.
스터브 영역에서 AS 외부 목적지로 향하는 경로 설정은 각 영역마다 기본값으로 설정되어야만 합니다. 이에 따라 스터브 영역 내부의 경로가 유지되는 토폴로지 데이터 베이스의 크기와 용량을 줄일 수 있으며, AS 내부하고만 정보를 주고받을 수 있도록 설정할 수도 있습니다.

10. OSPF에서의 통신용 라우터와 접속용 라우터

OSPF에서는 공통의 네트워크에 접속되어 있는 통신용 라우터와 인접 관계를 만들어 경로 설정에 관한 정보를 쉽게 주고받을 수 있게 합니다.
다중 접근 네트워크에서는 OSPF의 HELLO 프로토콜을 사용하여 통신용 라우터를 동적인 방법으로 식별할 수 있게 됩니다.

11. OSPF에서의 HELLO 프로토콜

HELLO 프로토콜은 통신 상대들의 관계를 구축하고 그것을 유지하기 위해 사용되며, 또한 통신 상대들과 양방향 통신을 하게 됩니다. HELLO 패킷은 모든 라우터에서 정기적으로 송출되는데. 양방향 통신이므로 라우터는 통신 상대의 HELLO 패킷에 자신의 정가 동록되어 있는 것을 확인할 수 있게 됩니다.
다중 접근 네트워크에서는 HELLO 프로토콜이 네트워크의 지명 라우터를 선택하게 됩니다.



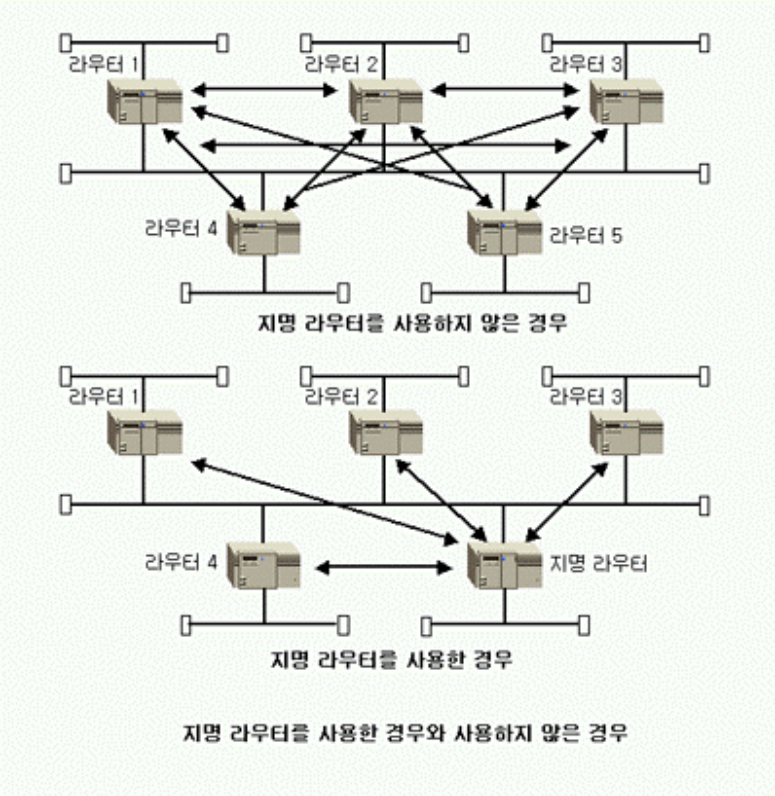
HELLO에 의한 경로 설정 정보의 교환

12. OSPF에서의 지명 라우터

다중 접근 네트워크에는 각각의 지명 라우터가 있게 되는데 각각의 지명 라우터는 다음과 같은 기능을 가지게 됩니다.

① 지명 라우터는 자신의 지명 라우터로 되어 있는 네트워크에 토폴로지 정보를 보내게 됩 니다. 이 통지는 지명 라우터를 포함하여 현재 네트워크에 접속되어 있는 라우터를 열거 하게 됩니다.

② 지명 라우터는 그 네트워크의 다른 모든 라우터와 인접 관계를 맺게 됩니다. 링크 상태 에 관한 데이터 베이스는 인접 관계를 맺은 지명 라우터와 일관성을 가져야하므로, 지명 라우터는 이 동기화 프로세스의 중심적인 역알을 하게 됩니다.



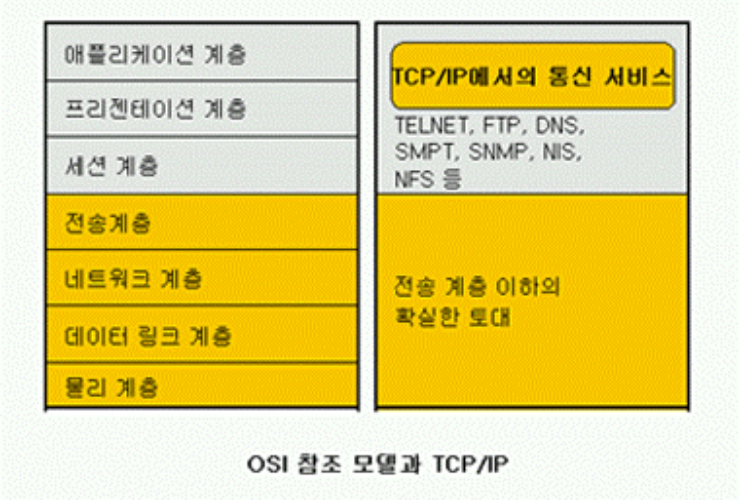
지명 라우터를 사용한 경우와 사용하지 않은 경우

TCP/IP의 통신서비스

1. 통신 서비스의 개요

지금까지의 살펴보았던 IP 프로토콜이나 TCP, UDP 프로토콜, 그리고 경로 설정 프로토콜들은 실제로 통신을 하는데 있어 가장 기초가 되는 부분입니다. 모든 TCP/IP 통신 서비스는 이를 기초로 하여 성립이 됩니다.

앞으로 살펴볼 통신 서비스란 OSI 참조 모델의 상위 계층으로 5, 6, 7계층에 해당되는 부분으로 우리가 컴퓨터를 다룰 때 가장 흔하게 접하게 되는 부분의 서비스입니다. 간단히 말하면 통신 서비스란 우리가 TCP/IP 통신에서 이용할 수 있는 즉, 일반적인 애플리케이션 서비스를 의미합니다.



OSI 참조 모델과 TCP/IP

2. 통신 서비스의 종류

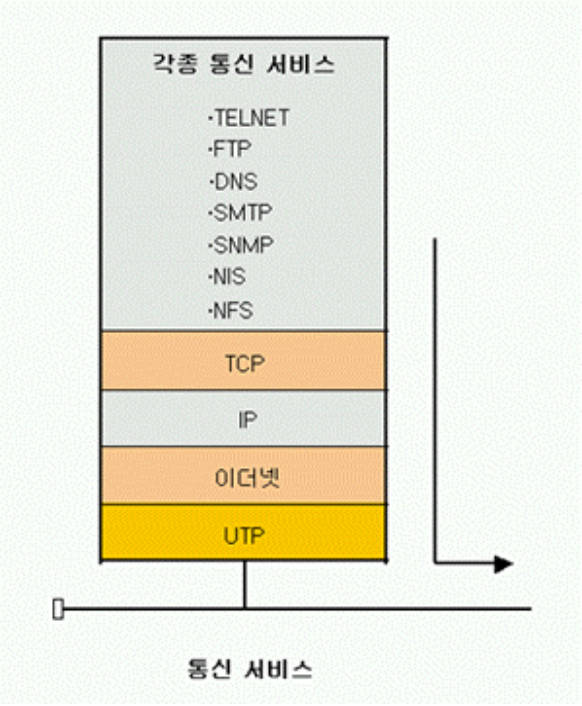
그러면 통신 서비스의 종류에는 어떠한 것들이 있을까요?

OSI 참조 모델은 확실한 계층화 구조와 아주 개방적이기 때문에, 상위 계층 부분은 사용자 스스로 독자적인 응용 소프트웨어를 쉽게 개발할 수 있어서 그 수를 짐작하기는 어렵습니다.

앞으로 소개할 통신 서비스는 RFC에 의해 공표 되고 여러 업체에서 제공되고 있는 대표적인 것들을 살펴보도록 하겠습니다. 또한 TCP/IP 통신을 지원하는 장치간에 서로 통신 할 수 있는 것과 연계시켜 설명하도록 하겠습니다.

TCP/IP 통신 서비스와 프로토콜에는 다음과 같은 것들이 있습니다.

- ▶ TELNET
- ▶ FTP(파일전송 프로토콜)
- ▶ DNS(도메인 네임 시스템)
- ▶ SMTP(간이 메일 전송 프로토콜)
- ▶ SNMP(간이 네트워크 관리 프로토콜)
- ▶ NIS(네트워크 정보 시스템)
- ▶ NFS(네트워크 파일 시스템)

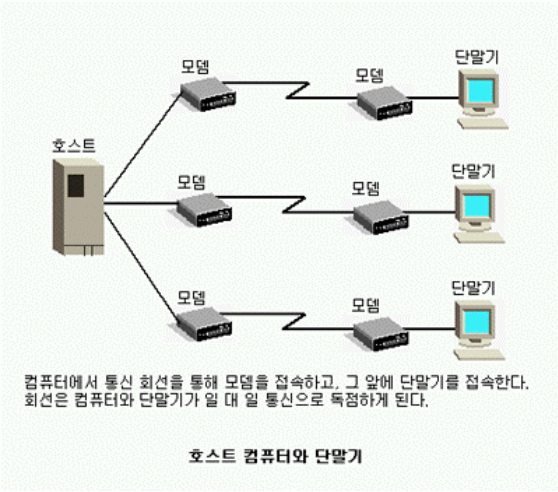


통신 서비스

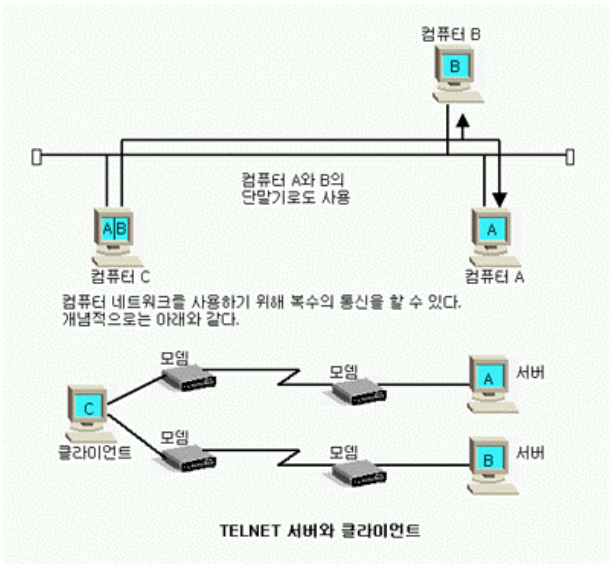
TELNET

1. TELNET의 개요

TCP/IP 통신에서 가장 대표적인 통신서비스가 TELNET입니다.
TELNET은 종전의 호스트 컴퓨터와 단말기 통신을 확장한 형태로써, 자제 처리 능력을 지닌 호스트(또는 서버)와 이 호스트에 연결되는 단말기(또는 클라이언트) 그리고 호스트와 단말기를 연결하는 단말용 통신 회선으로 이루어집니다. 이 호스트와 단말기 사이에 통신을 가능하게 해주는 프로토콜이 바로 TELNET입니다. 즉, 처리 능력을 가진 호스트에 통신회선을 이용하여 각각의 단말기로 접속한 후 원격제어가 가능한 형태를 말합니다.



호스트 컴퓨터와 단말기



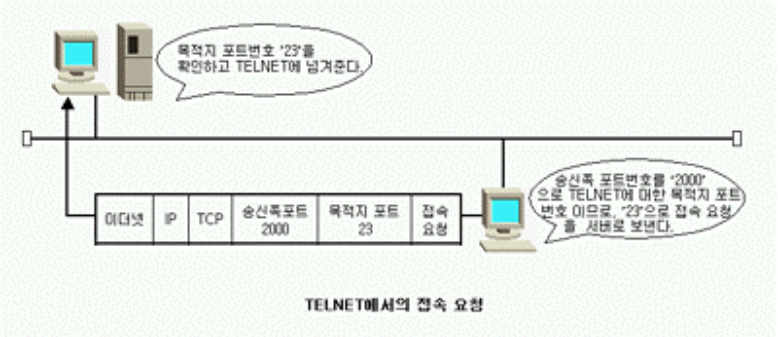
TELNET 서버와 클라이언트

2. Login과 Logout

TELNET을 이용하여 네트워크로 연결된 모든 컴퓨터에 들어갈 수 있게 되는데 이렇게 상대 컴퓨터에 들어가게 되는 것을 'Login'하고 연결을 끊고 나오는 것을 'Logout'이라고 합니다.
예를 들어 컴퓨터 A에서 임의의 컴퓨터 B에 있는 파일을 보기 위해서는 두 가지 방법이 있을 겁니다. 한가지는 컴퓨터 B에 직접 가서 파일을 수정하는 경우입니다, 이때 문제점은 컴퓨터 B가 원거리에 있을 경우 사용의 어려움이 있다는 것입니다. 이에 반해 다른 한가지는 컴퓨터 B가 네트워크에 연결되어 있어 TELNET으로 연결하는 경우입니다. 이 경우 TELNET으로 네트워크에 연결되어 있는 컴퓨터 B에 Login하여 파일을 수정할 수 있기 때문에 물리적 거리의 제약이 없어지게 됩니다. 이렇게 물리적으로 접속된 컴퓨터를 네트워크를 통해 논리적 단말기로 접속하는 형태를 '가상 단말기'라고 합니다. 여기서 TELNET은 TCP/IP 통신 환경에서 가상 단말기 기능을 실현하는 프로토콜을 말하게 됩니다. 이때, 가상의 단말기와 상대 컴퓨터와 연결에 있어 기본 전제가 신뢰성의 확보입니다. 때문에 TELNET 통신에서는 TCP 프로토콜을 사용하게 됩니다.

3. 메커니즘

① TELNETD(TELNETDaemon)
TELNET은 접속을 요청하는 클라이언트와 접속을 허용하는 서버의 형태로 가지고 있습니다. 이때 접속되는 쪽인 서버는 클라이언트의 접속에 대해 항상 응답할 준비가 되어 있어야 하는데, 이때 서버는

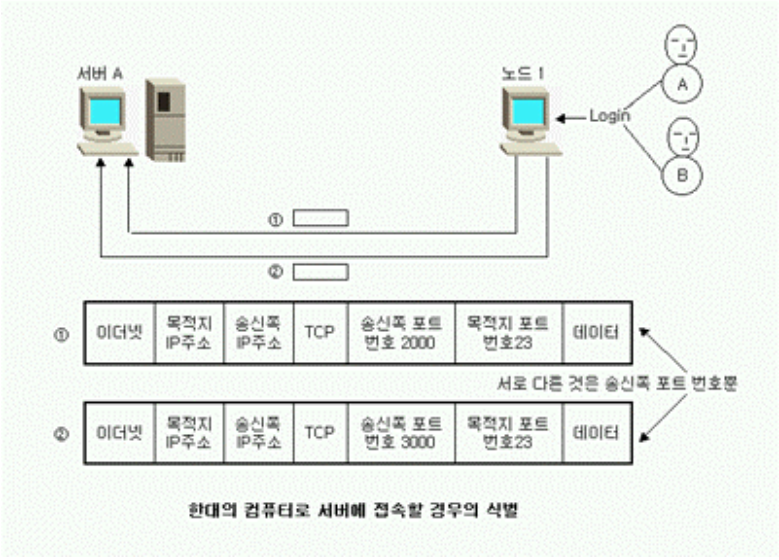


TELNET에서의 접속 요청

TELNETD(TELNETDaemon)이라는 프로세서를 항상 작동시켜 두어 클라이언트가 보내는 접속 요청에 대해 응답하게 합니다. 또한 클라이언트에서 보내는 TELNET 접속 요청은 반드시 TCP 헤더의 목적지 포트 번호가 '23'으로 되어 있습니다. 따라서 서버에서는 TCP 헤더의 목적지 포트 번호가 '23'이면 이것이 TELNET 접속 요청이라는 것으로 판단하게 되는 것입니다.

② TELNET으로 접속된 클라이언트의 식별 방법

서버는 TCP 헤더의 송신쪽 포트 번호를 보고 TELNET에 접속하는 모든 클라이언트를 식별하게 됩니다. 송신쪽 포트 번호를 보는 이유는 TELNET에 접속되는 송신쪽 TCP 헤더의 목적지 포트 번호가 '23'이므로, 여러 요청에 대하여 구별이 불가능해지기 때문입니다.



한대의 컴퓨터로 서버에 접속할 경우의 식별

만약에 하나의 컴퓨터로 사용자 A와 B가 TELNET으로 서버에 Login 할 경우, 송신쪽 IP 주소는 같게 되므로 판단할 수 없게 됩니다. 그래서 이 때 송신쪽 포트 번호로 구별하게 되는 것입니다. 이 때 송신쪽 포트 번호 값은 대게 1024번 이상으로 할당되는데, 그 이유는 TCP 포트 번호 1번부터 1024까지는 통신 서비스의 종류에 따라 이미 정해져 있기 때문에 1024번 이상의 번호를 동적으로 할당하고, 그 값을 TCP 헤더의 송신쪽 포트 번호에 넣어 서버에 넘겨주게 되는 것입니다. 이것을 수신한 서버는 TCP 헤더의 송신쪽 포트 번호를 보고, 어느 IP 주소와 포트 번호를 사용하고 있는지를 판단하고, 클라이언트로 응답을 보낼 때에는 그 송신쪽 포트 번호의 값을 목적지 포트 번호로, 반대로 '23'번이라는 포트 번호를 송신쪽 포트 번호에 넣어 응답하게 됩니다. 이렇게 해서 어떤 클라이언트에서 보낸 TELNET 접속 요청에 대해서도 클라이언트를 중복시키지 않고 통신할 수 있게 되는 것입니다.

4. TELNET의 서비스 종류

TELNET의 기본 서비스로는 크게 두 가지로 살펴 볼 수 있습니다.

① NVT(Network Virtual Terminal: 네트워크 가상 단말기서비스)

이것은 어떤 종류의 컴퓨터와 조합해도 서비스를 제공할 수 있도록 NVT라는 표준 단말기 유형을 정의해 놓은 것입니다. 예를 들어 10종류의 서로 다른 단말기가 존재한다고 하면, 이들을 모두 조합하여 문자 코드나 화면을 제어 변환하는 유틸리티가 필요하며, 이 것으로 관리해만 합니다. 그러나 NVT를 정해 두고 각 단말기 자체와 NVT 변환 유틸리티를 하나만 준비하면, 사용중인 단말기의 종류가 변한다 해도 NVT는 변하지 않으므로 같은 유틸리티를 사용할 수 있게 됩니다. 이렇게 해서 현재 존재하는 수백 종류의 단말기들을 작동시킬 수 있게 되는 것입니다.

② OPTION

클라이언트와 서버가 'Option'을 교섭하는 서비스를 말합니다. NVT는 많은 단말기가 공통으로 가지고 있는 기능의 특성을 정의한 것이지만 이 기능만으로는 부족한 경우도 있게 됩니다. 예를 들면, NVT에는 연결을 통해 건네준 데이터의 문자 코드가 2진 모드인지 또는 ASCII 모드인지가 정의되어 있지 않습니다. 그러므로 이 옵션 서비스를 이용하여 NVT로 부족한 부분을 클라이언트와 서버 사이에 교섭하여 보완하게 되는 것입니다.

FTP

(파일 전송 프로토콜)

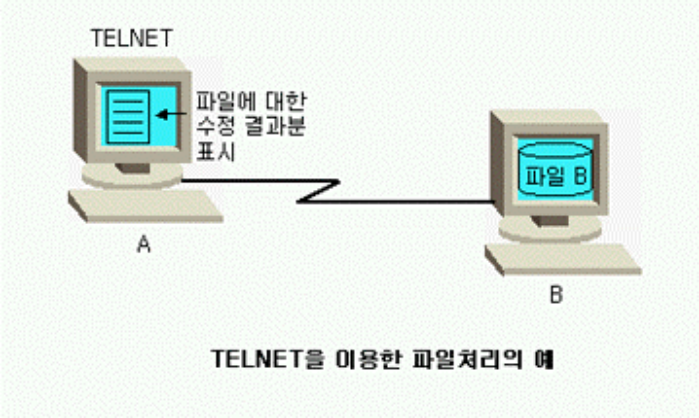
1. 개요

FTP는 서로 다른 컴퓨터들 사이에 파일을 주고받을 수 있게 해주는 파일 전송 프로토콜로서 네트워크에 연결되어 있는 하나의 컴퓨터에서 다른 컴퓨터 파일에 있는 파일을 가져오거나 그 반대 방향으로 보낼 수 있는 TCP/IP 통신 중에서 널리 사용되는 통신 서비스입니다.

TELNET과 마찬가지로 상대방 컴퓨터에 "로그인"하여야 합니다.

2. FTP 사용의 예

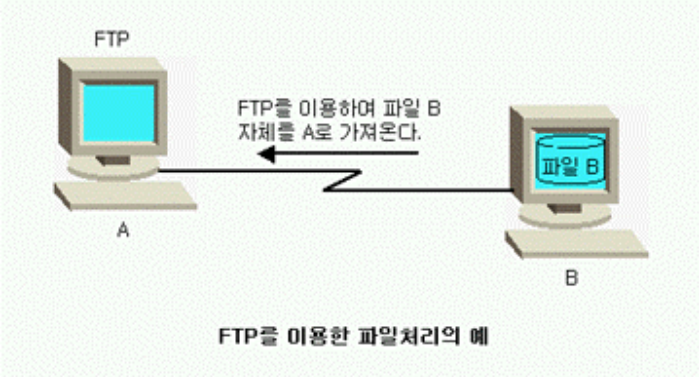
그럼 FTP와 TELNET을 이용한 파일 처리의 차이점을 알아보도록 하겠습니다.



TELNET을 이용한 파일처리의 예

우선 네트워크 상에 있는 컴퓨터 A에서 컴퓨터 B에 있는 파일을 볼 필요가 생겼을 때 TELNET을 사용하여 컴퓨터 B로 로그인하여 필요한 파일을 본 후, 로그아웃 하면 됩니다.

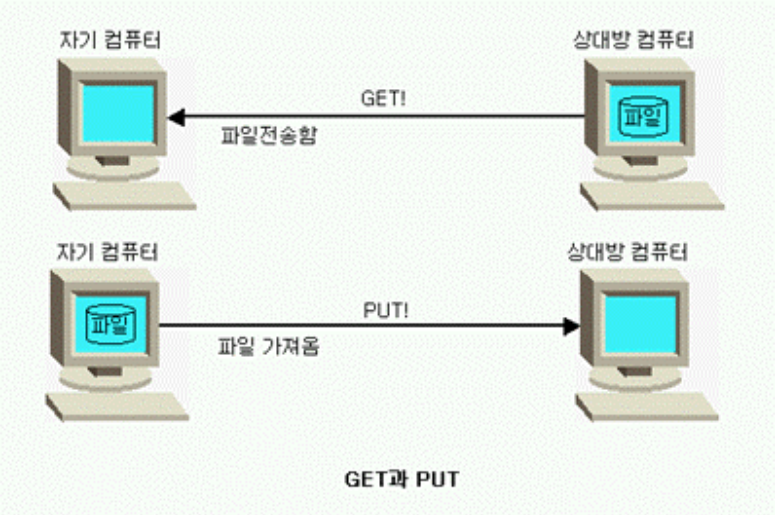
이 경우 컴퓨터 A의 사용자는 TELNET을 이용하여 컴퓨터 B에 로그인하고 컴퓨터 B의 하드디스크 안에 있는 파일을 참조하게 됩니다. 이 때 파일 자체는 컴퓨터 A에 전송되지 않고 컴퓨터 B에서 수행한 결과만을 컴퓨터 A에 보내게 됩니다. 즉 TELNET으로는 컴퓨터 A가 컴퓨터 B에 있는 파일을 참조할 할 뿐, 컴퓨터 A로 파일을 데이터에 대한 연산 처리는 불가능하다는 것입니다. 그래서 네트워크 상에 있는 컴퓨터 B의 파일을 컴퓨터 A로 가져와 데이터를 처리할 때 필요한 것이 바로 이 FTP인 것입니다.



FTP를 이용한 파일처리의 예

FTP를 사용하여 컴퓨터 B에 로그인하고, 필요한 파일을 자신의 컴퓨터 A로 가져오면 됩니다.

FTP에서는 이와 같이 상대방 컴퓨터에서 자신의 컴퓨터로 파일을 가져오는 것을 "GET"이라고 하고, 상대방 컴퓨터로 파일을 보내는 것을 "PUT"이라고 합니다.

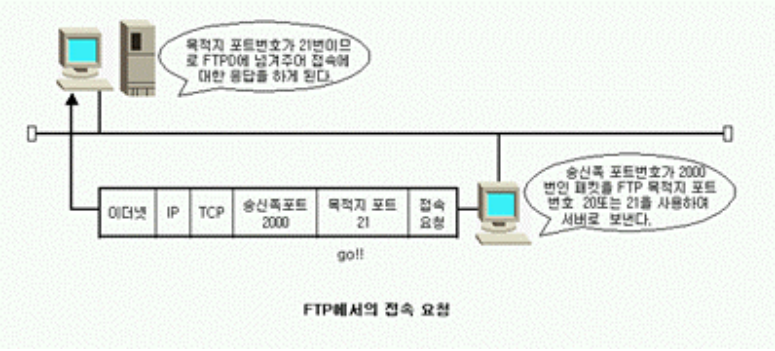


GET과 PUT

3. FTP 메커니즘

FTP에 의한 접속은 TELNET과 마찬가지로, 제어의 중추가 되는 서버와 서버의 처리에 의존해 작동되는 클라이언트로 이루어집니다.

FTP 또한 TELNET과 마찬가지로 접속되는 쪽인 서버는 FTPD(FTPDaemon) 프로세스를 항상 작동시켜 두어 접속하는 클라이언트에서 보내는 접속 요청에 대하여 언제든지 응답할 준비를 하게 됩니다.



FTP에서의 접속 요청

클라이언트에서 보내는 FTP 접속 요청은 반드시 TCP 헤더의 목적지 포트 번호가 20번과 21번으로 설정되어야 있어야만 합니다. 즉, 서버는 TCP 헤더의 목적지 포트 번호의 값이 20번과 21번임을 확인하고 이것이 FTP 요청인 것임을 인식하게 됩니다. 이 때 서버는 반드시 TCP 헤더의 송신측 포트 번호를 보고, 그 포트 번호와 한 쌍이 되도록 연결하게 되는데, 송신측 포트 번호를 보게 되는 이유는 모든 FTP 접속 요청이 TCP 헤더의 목적지 포트 번호를 20번과 21번으로 한 상태에서 이루어지므로, 여러 요청이 있을 경우 구별할 수 없기 때문입니다.

예를 들어 같은 컴퓨터를 여러 사용자가 FTP로 접속하는 경우 송신측 IP 주소와 FTP 접속을 위한 TCP 헤더의 목적지 포트 번호도 20번과 21번으로 같게 되어 이 요청이 어디서 온 클라이언트의 접속 요청인지 구별할 수 없게 됩니다. 따라서 클라이언트는 FTP로 접속 요청을 할 때, 목적지 포트 번호는 20번과 21번으로 하지만 송신측 포트 번호는 임의의 값으로 정하게 됩니다. 이 때 송신측 포트 번호의 값은 TELNET과 마찬가지로 보통 1024번 이상으로 할당됩니다. 클라이언트는 서버에 대해 FTP 접속 요청을 할 때, 현재 비어 있는 1024번 이상의 번호를 동적으로 할당하고, 그 값을 TCP 헤더의 송신측 포트 번호에 넣어서 서버에 보내게 됩니다. 이것을 수신한 서버는 TCP 헤더의 송신측 포트 번호를 보고 어느 IP 주소가 어느 포트 번호를 사용하는지를 인식하게 되는 것입니다. 서버에서 클라이언트로 응답을 보낼 때에는 그 송신측 포트 번호의 값을 목적지 포트 번호에 넣은 상태로 답신을 보내게 됩니다. 이에 따라 여러 클라이언트에서 보낸 FTP 접속 요청을 식별하게 되는 것입니다.

4. 포트 번의 처리

TCP의 포트 번호의 개념은 기본적으로 어느 서비스에서나 동일합니다.

TELNET에서의 포트 번호 취급도 FTP에서 포트 번호를 취급하는 것과 기본적으로 같습니다. 단지 다른 점은 포트 번호의 수치일 뿐입니다. 통신 서비스에서 이 포트 번호의 수치는 중요합니다. 다시 말해, TCP/IP 통신에서 이 포트 번호 값만 있으면 어느 서비스를 수행하고 있는지를 알 수 있게 됩니다.

이렇게 접속이 확립된 후, 클라이언트에서 보낸 수신 요청이나 송신 요청에 따라 파일을 블록 데이터로 전송하고, 전송이 끝나게 되면 클라이언트가 다음 지시를 보낼 까지 기다리게 됩니다. 이 때 종료 요청을 보내면 접속을 끊고 FTP 애플리케이션을 종료시킬 수 있게 됩니다.

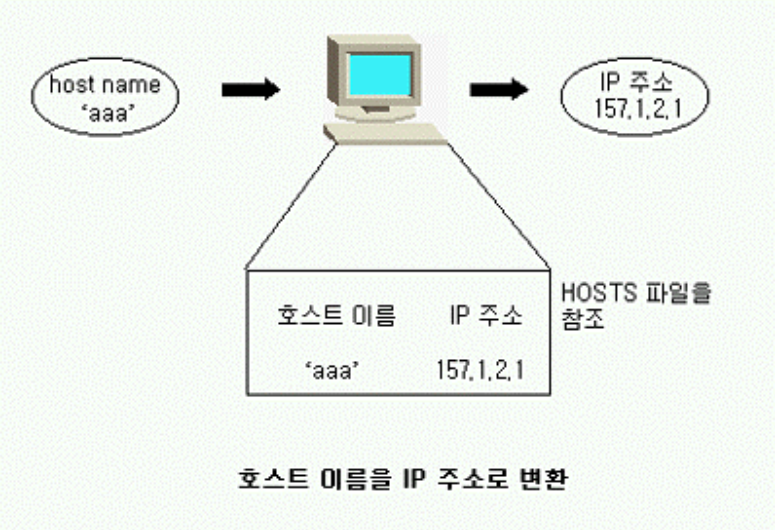
DNS

(Domain Name System)

1. 개요

네트워크를 통해 컴퓨터 상호간에 통신을 하기 위해서는 네트워크 상에 접속되어 있는 각각의 컴퓨터를 인식하기 위한, 컴퓨터 고유 IP 주소를 설정하고, 그 IP 주소를 바탕으로 통신을 해야만 합니다.

그러나 실제로 통신을 수행하게 될 경우에 일일이 목적지의 IP주소를 기억하여 사용하기에는 사용자 측면이나 관리측면으로 봤을 때 여러 가지 어려움이 따르게 되며 작업 능률 또한 떨어지게 됩니다. 그래서 일반적으로 각각의 컴퓨터의 IP에 대응하는 컴퓨터 이름을 지정하여 IP 주소를 입력할 때와 똑같이 상대방 컴퓨터와 통신 있게 합니다. 단지 실제 통신은 IP 주소를 기본으로 이루어지며, 이름을 입력하는 것은 어디까지나 조작상의 편의를 위한 것일 뿐입니다. 이렇게 각각의 컴퓨터에 붙여지는 이름을 "Host Name"이라고 불리게 됩니다.



호스트 이름을 IP 주소로 변환

보통의 워크스테이션에서는 이와 같은 IP 주소와 호스트 이름을 대응시키는 HOSTS 파일이라는 데이터베이스 파일을 보관하게 됩니다. 이를 바탕으로 사용자 컴퓨터의 이름이 입력이 되면 HOSTS의 데이터베이스 파일을 참조하여 컴퓨터 IP 주소로 변환시켜 주게 되는 것입니다.

하지만 네트워크에 접속되는 컴퓨터 수가 증가함에 따라 데이터 베이스 파일이 커지고, 네트워크의 규모 또한 확장함에 따라 컴퓨터 IP 주소 등록이나 변경 등의 일괄 관리에도 문제점이 생기게 되었습니다. 그래서 이러한 문제점들을 해결하기 위한 수단으로 DNS가 고안 된 것입니다.

DNS는 IP 주소와 호스트 이름을 대응시키기 위한 시스템이며, 이에 대한 설정을 각각의 조직 내에서 변경하도록 한 분산형 관리 시스템입니다. 즉, 조직 내의 변경이나 추가 장소를 자체적으로 관리하여 일괄 관리의 필요성을 없애기 위한 시스템으로 네트워크의 규모가 아무리 확대된다고 해도 효율적으로 대응할 수 있게 됩니다.

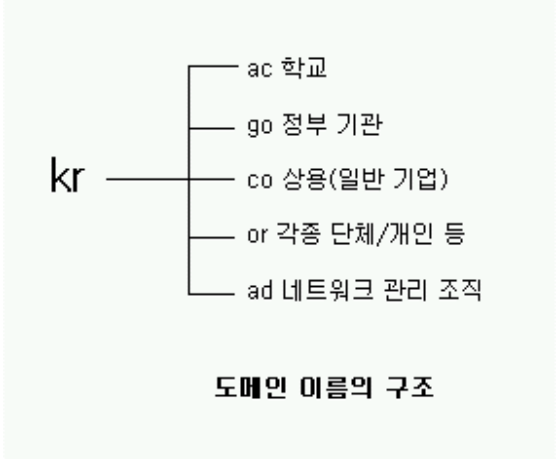
2. DNS 메커니즘

① DNS의 구조

DNS의 구조는 "Tree 구조"의 계층형 관리 구조로 되어 있습니다.

도메인의 계층 구조를 구성하는 도메인 이름은 반드시 그 상위 계층의 관리자가 등록하게 함으로써 일관성을 유지시키게 됩니다. 예를 들어 NetworkDesk는 "kr(한국 도메인 이름)" 중 "co(상용)"에 해당되며, 회사이름이 NetworkDesk이므로 도메인 이름은 다음과 같이 됩니다.

NetworkDesk.co.kr
만약 교육 관련 기관이라면 NetworkDesk.ac.kr이 되고, 정부 기관이라면 NetworkDesk.go.kr이 됩니다. 즉, "kr"이라는 루트 도메인 이름 아래 다음과 같이 분산이 되게 것입니다.



도메인 이름의 구조

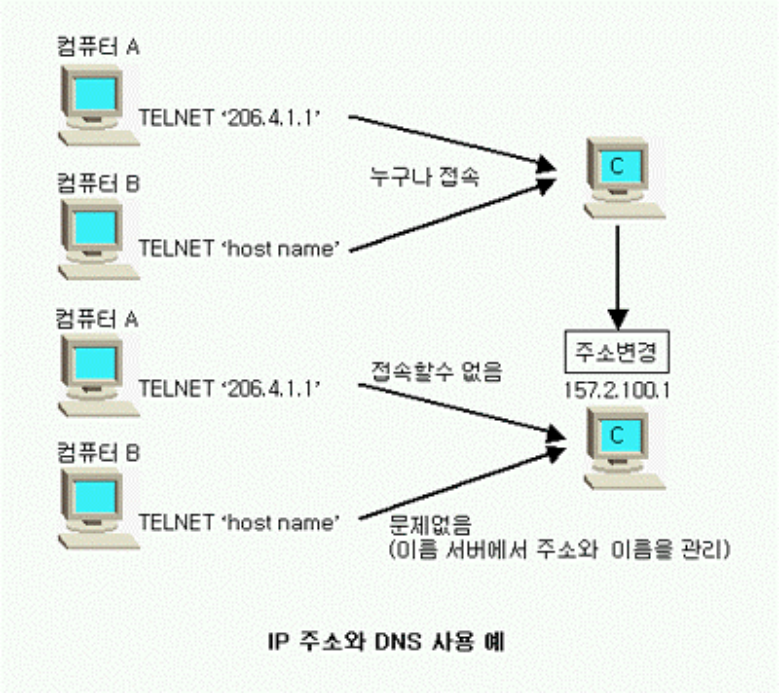
여기에서 도메인은 첫 번째 단계는 국가명을 두 번째 단계는 기관명을 세 번째 단계는 조직의 이름을 나타내게 됩니다. 여기서 각각의 단계마다 마침표(.)를 찍어 구분을 하게 됩니다. 예를 든다면 NetworkDesk.co.kr인 경우 첫 번째 단계가 "kr"이고, 두 번째 단계는 "co"이며 마지막으로 세 번째 조직에 해당되는 부분이 바로 "NetworkDesk"인 셈입니다.
그리고 이 NetworkDesk.co.kr 내에는 다양한 호스트 이름을 가진 컴퓨터가 존재하게 되는데 예를 든다면, aaa라는 호스트 이름을 가진 컴퓨터가 존재한다면 aaa.NetworkDesk.co.kr이라는 호스트 이름을 갖게 되는 것입니다. 그리고 새로 사내 컴퓨터를 추가하거나 IP 주소를 변경할 경우, 상위 조직과는 상관없이 자체적으로 변경만 하면 됩니다.
DNS는 이러한 도메인 이름을 기본으로 하여 3가지 요소로 구성이 되는데 하나는 지금까지 설명한 도메인 이름의 구조이며, 나머지 둘은 "Name Server"와 "Reserver"를 말합니다.

② 네임서버
네임서버란 도메인 이름을 관리하는 소프트웨어를 말합니다. 다시 말해 도메인의 어떤 부분의 호스트 이름과 IP 주소의 정보 또는 다른 이름 서버의 위치 정보를 가지고 있다면, 어느 호스트 이름이 어떤 IP 주소인지를 알 수 있으며, 자기가 관리하지 않는 도메인 이름일 경우, 거기에 적힌 이름 서버를 문의하게 됩니다.

③ 리졸버
리졸버란 클라이언트에서 컴퓨터의 이름을 물을 경우, 이름 서버를 조회하기 위한 통신을 실행하는 소프트웨어를 말합니다. 리졸버는 적어도 하나의 이름 서버에 접근할 수 있어야 하며, 이 이름 서버를 이용하여 묻는 내용에 대한 참조 정보를 받게 됩니다.

3. DNS의 잇점

DSN는 TELNET이나 FTP와 같은 통신 서비스와는 달리 추상적이므로 이해하기가 다소 어려울 수 있습니다. 하지만 TCP/IP 통신에서는 이 호스트 이름으로 접속하는 것이 보통이며, 호스트 이름과 IP 주소의 관리를 위하여 DNS가 효과적으로 쓰이게 됩니다.



IP 주소와 DNS 사용 예

예를 들어 항상 IP 주소를 입력하여 컴퓨터 C에 연결하는 사용자 A와 DNS 이름을 사용하여 컴퓨터 C에 연결하는 사용자 B가 있다고 가정하자. 만약에 컴퓨터 C의 관리자가 어떠한 사정에 따라 컴퓨터 C의 IP 주소를 변경할 경우, 사용자 B는 DNS를 사용하므로, 컴퓨터 C의 IP 주소가 바뀌면 DNS의 데이터 베이스도 자동으로 함께 바뀌므로 컴퓨터 C의 IP 주소 변경에 상관없이 컴퓨터 C와 통신할 수 있습니다.

반면에 IP 주소를 이용하여 컴퓨터 C에 연결하던 사용자 A는 컴퓨터 C의 IP 주소가 바뀜과 동시에 통신을 할 수 없게 됩니다. 결국 컴퓨터 C의 관리자에게 바뀐 IP 주소를 알아내야만 통신이 가능하게 되는 것입니다.

이러한 점을 볼 때 DNS를 사용함으로서 네트워크 관리의 효율성을 증대시킬 수 있으며, 전자 우편 등의 통신 서비스와의 연계를 고려할 때 그 중요성은 더욱 커지게 됩니다.

SMTP

(Simple Mail Transfer Protocol)

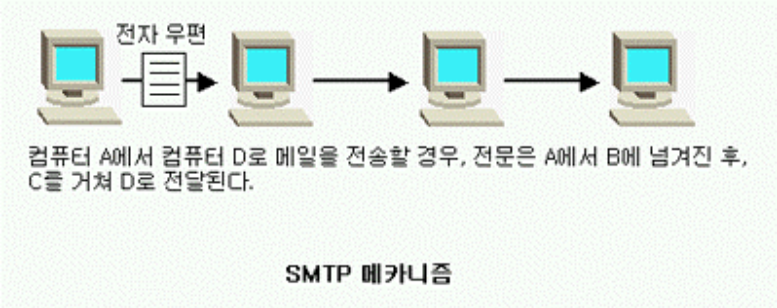
1. 개요

SMTP(간이 메일 전송 프로토콜)은 전자 우편을 지원하는 TCP/IP 통신 중 우리에게 가장 친숙한 서비스라고 할 수 있습니다. 기존에 우편이라던가 전화로 상대방과 통신하기 위해 먼 곳까지 메시지를 보내게 될 때, 도착지까지 걸리는 시간과 비용, 전화로 상대방 통화할 때 상대방이 자리에 없는 경우 등의 불편함이 뒤따랐지만, 네트워크를 통한 전자 우편의 등장으로 자신의 컴퓨터에서 상대방 컴퓨터로 쉽고도 거의 실시간으로 메시지를 전달할 수 있게 되었습니다. 연구 활동을 하는 사람들에게 전세계의 연구자와 의견 교환을 할 수 있는 전자 우편인 TCP/IP는 여러 통신 중에서도 가장 중요한 수단이 되었습니다. 특히 인터넷에 접속한 장치들 사이로 메일을 주고받는 일은 전세계 어디서나 몇 분 안에 처리되어 상대방 컴퓨터로 전송이 가능하게 되었습니다.

이 전자 우편 서비스를 멀티 벤더 환경에서 제공하고 문자형 전자 우편을 임의 사용자에게 보내 주는 프로토콜이 바로 SMTP입니다. SMTP는 메일을 능률적으로 상대방에게 확실히 도착시키기 위해 TCP 연결에서 사용이 됩니다.

2. SMTP 메커니즘

SMTP의 통신 메커니즘은 다음과 같습니다.



우선 컴퓨터 A에서 컴퓨터 C로 메일을 전송한다고 가정합니다. 컴퓨터 A의 사용자가 mail 명령으로 메일의 내용과 도착지를 지시합니다. 그리고 컴퓨터 A의 메일 서버인 컴퓨터 B에 넘겨집니다. 컴퓨터 A에서 메시지를 받은 컴퓨터 B는 컴퓨터 D의 메일 서버인 컴퓨터 C에 컴퓨터 A가 보낸 메일의 목적지 메일 주소가 있는지를 확인하고 컴퓨터 C에 메일을 넘기게 됩니다. 컴퓨터 D는 컴퓨터 C에서 메일을 수신하게 됩니다. 이렇게 해서 메일의 전송이 성공적으로 끝나게 되는 것입니다. 만일 컴퓨터 C에 메일의 목적지 메일 주소가 없다면 컴퓨터 B는 다른 메일 서버에서 찾거나, 그래도 찾지 못할 경우에는 컴퓨터 A에게 메시지를 다시 돌려주게 됩니다.

3. 전자 우편 주소의 형식

전자우편을 사용할 때에는 DNS에서 설명한 도메인 이름을 사용합니다. 도메인 이름은 흔히 전자 우편 주소라고 부릅니다. 일반 우편에서의 주소와 이름에 해당되는 것이지요.

전자 우편에서는 발신자와 수신자를 식별하기 위한 전자 우편 주소가 있습니다. 그럼 전자 우편 주소의 구체적인 예를 들어보도록 하겠습니다.

bbb@aaa.networkDesk.co.kr

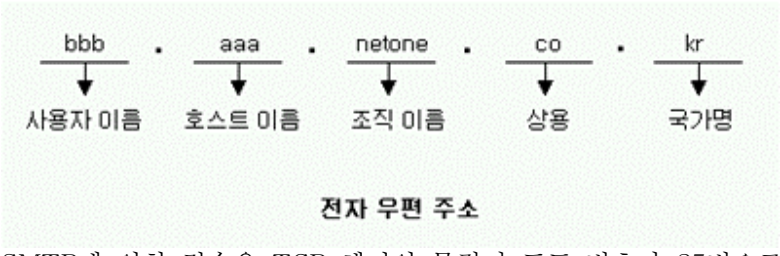
여기서 bbb는 사용자의 계정 이름 즉, 컴퓨터에 로그인할 때 사용하는 이름을 말하며, aaa는 그 컴퓨터의 호스트 이름을 말합니다. 이와 같은 전자 우편 주소를 사용하여 우편을 교환하므로 DNS에 의한 계층화 구조의 관리가 중요하게 됩니다.

DNS에는 각 호스트 이름에 대한 IP 주소와 별도로 메일의 전송 목적지 호스트 이름을 등록할 수 있습니다. 이것을 MX(Mail eXchanger: 메일 교환기) 레코드라고 합니다.

예를 들어 NetworkDesk.co.kr을 MX 레코드로 등록한다면 aaa.NetworkDesk.co.kr 또는 ccc.NetworkDesk.co.kr는 모두 NetworkDesk.co.kr의 메일이므로 여기에 등록된 호스트로 전송이 이루어집니다. 그리고 NetworkDesk에 aaa가 전송되고 마지막으로 aaa의 컴퓨터가 bbb라는 사람에게 메일을 건네주게 되는 것입니다. 이는 DNS에 의해 관리되는 도메인 이름을 그대로 전자 우편에서도 사용하는 것입니다.

네트워크를 구축하고 전 세계의 사람들과 전자 우편을 교환하거나 사내에서 전자 우편 시스템을 구축하려면 DNS의 체계를 확실히 구축해야 합니다.

전자 우편 주소의 구조를 살펴보며 다음과 같습니다.



전자 우편 주소

SMTP에 의한 접속은 TCP 헤더의 목적지 포트 번호가 25번으로 이루어집니다. 즉, 서버에서 TCP 헤더의 목적지 포트 번호의 값이 25번이면 그것이 SMTP 접속 요청이라는 것을 인식하게 것입니다.

SNMP

(Simple Network Management Protocol)

1. 개요

네트워크 기술과 규모가 급속히 진행됨에 따라 네트워크 관리작업에 관한 연구가 또 하나의 중요한 과제가 되어 왔습니다. 그 중에서 TCP/IP의 폭발적인 보급과 함께, 관리가 쉽고 네트워크 관리 프로그램을 장치에 설치하기 쉬운 SNMP(간이 네트워크 관리 프로토콜)가 널리 보급되기 시작하였으며, 지금은 TCP/IP 이외에 다른 프로토콜군이나 접속형태까지 포함한 형태로 발전하게 되었습니다.

2. 메커니즘

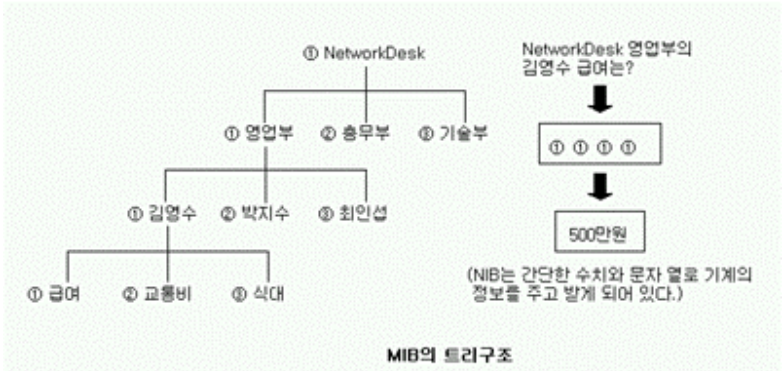
SNMP는 관리자가 시작하는 네트워크 감시 단말기(SNMP 관리자)와 SNMP가 설치되어 있는 장치(SNMP 관리자) 사이의 프로토콜을 규정하게 됩니다. 여기서 관리자 사이에서 통신할 때 주고받는 내용에 관하여 규정 짓는 것을, MIB(Message Information Base: 메시지 정보 기반)라고 합니다.

MIB는 크게 표준 MIB(MIB, MIB-II, FDDI_MIB 등)와 각 기관 고유의 정보를 업체가 독자적으로 전개한 확장 MIB로 나눌 수 있으며, 이 두 가지 모두 ISO 규정의 ASN.1(Abstract Syntax Notation 1: 추상적 문법 표기법, MIB 전용 컴퓨터 언어)에 정의된 문법으로 표현됩니다.

이 ASN.1의 규정에 따른 MIB 정보를 대행자나 관리자가 인식하게 하려면, 이기종 간의 일치성을 확보해야 합니다.

서버에 설치되어 있는 MIB는 ASN.1의 규약에 따라 기술되므로, SNMP 클라이턴트에서 그 MIB를 컴파일하면 SNMP를 이용한 통신이 가능하게 됩니다.

MIB에 따라 정의된 내용은 모두 트리 구조로 되어 있습니다. 통신은 그 트리구조에 알맞는 번호를 나열하고 거기에 수치와 문자열을 추구하여 구성되며, 정의된 항목에 대응하는 수치와 문자열을 주고받는 단순한 형식으로 통신하는 점이 가장 큰 특징입니다.



MIB의 트리구조

기본적인 동작은 정보 요청(Get-Request), 앞서 요구한 다음 정보 요청(Get-Next-Request), 정보 요청 응답(Get-Response), 설정 요청(Set-Request), 이벤트 통지(Trap)로 구성됩니다.

트랩(Trap)은 대개 올가미와 같은 장치를 의미하는데, SNMP의 트랩도 비슷한 의미를 가지게 됩니다. 어떤 원인으로 네트워크 장치의 작동 상황이 변화한 경우, MIB에서 그러한 상황의 변화를 관리자에게 알려야 할지 여부를 트랩으로 정의해 두고, 정의된 SNMP 메시지는 반드시 관리자에게 통지되어야 합니다.

이 트랩을 이용하여, 관리자는 상황의 변화를 계속 확인하는 일로부터 해방될 수 있게 됩니다. 현재 실현할 수 있는 내용이 유동적이고 사용자 인터페이스가 초보자가 이해하기에 어렵다는 단점이 있는 반면에 구성도를 아이콘화한 맵을 이용하여, SNMP를 장애 감시용 단말기로 이용하는 등 이미 많은 사용자들이 이용하고 있습니다.

NFS와 NIS

(Network File System) (National Information System)

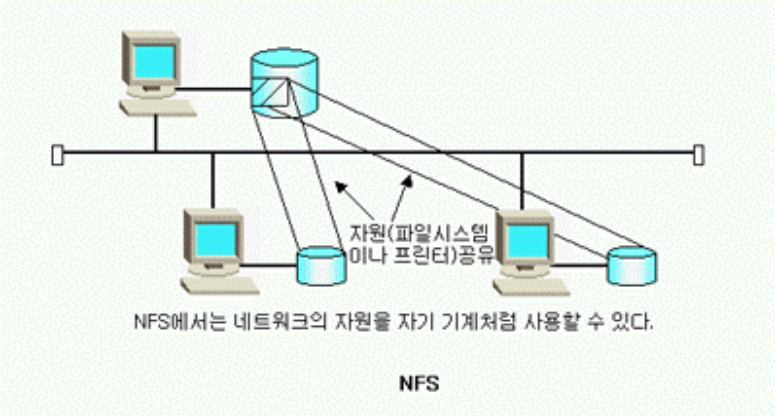
1. 개요

NFS(네트워크 파일 시스템)과 NIS(네트워크 정보 시스템)은 둘다 Sun Micro-Systems사에서 독자적으로 개발한 프로토콜로서 UNIX 워크스테이션의 자원을 효과적으로 이용하기 위해 만든 애플리케이션입니다. 이 들은 현재 널리 보급되어 사실상 표준으로 자리잡고 있습니다.

2. NFS(Network File System)

NFS는 네트워크 상에 있는 모든 컴퓨터의 하드디스크를 다른 컴퓨터에서도 똑같이 조작할 수 있도록 지원해 주는 애플리케이션으로서 네트워크의 파일 시스템을 쉽게 공유할 수 있게 해줍니다.

소프트웨어 개발이나 공동 개발 등, 여러 사람들이 동시에 장치를 이용하여 작업하려 할 때 어떤 사람이 파일을 갱신하려 하면 다른 사용자에게도 그 정보고 곧 바로 반영이 되어, 여러 사용자가 같은 조작으로 같은 파일 시스템을 공유할 수 있는 아주 유용한 애플리케이션입니다.

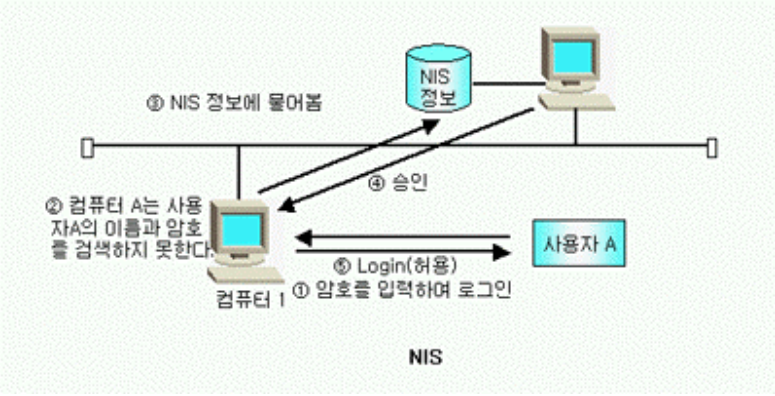


이와 같이 여러 사용자가 네트워크의 자원(파일 시스템이나 프린터)을 자기 장치처럼 조작하고 접근할 수 있도록 하는 구조가 NFS입니다. 또한 PC에서 UNIX 워크스테이션의 NFS를 이용하는 구조를 PC-NFS라 합니다.

3. NIS(National Information System)

NIS는 여러 UNIX 워크스테이션에서 사용자 관리나 호스트 이름 등의 관리 정보를 공유하기 위한 애플리케이션 입니다.

NIS에서는 각 장치가 공유할 수 있는 사용자 이름과 암호의 조합 및 호스트 이름 등의 정보를 일괄 관리하고 배포할 수 있는데 예를 든다면, 특정 사용자가 어떤 컴퓨터에 로그인할 때 사용자 이름과 암호를 그 장치가 가진 고유의 정보를 검색하지 않고 NIS로 일괄 관리되는 정보(데이터 베이스)를 확인하고 승인을 얻어 로그인 하는 것입니다.



TELNET 등에서 접속 목적지 호스트 이름을 입력하면, 장치는 대개 자체에 저장된 테이블로부터 입력된 그 이름의 IP 주소로 변환하여 처리하게 되는데, 이것을 장치 자체의 표 대신 NIS의 관리 정보로부터 변환한 후 처리할 수 있게 됩니다. 이에 따라 관리 저장 장소의 정보만 변경하면, 각 장치마다 필요했던 것과 같은 정의를 모든 장치에 반영시킬 수 있는 장점을 가지게 됩니다.