

P a r t

01

네트워크의 이해

Chapter 1 | 네트워크의 기초

Chapter 2 | 네트워크 모델

Chapter 3 | 네트워크 기술

네트워크의 기초

* 학습목표

- 네트워크 관련 기초 용어를 익힌다.
- OSI 7계층 모델의 필요성을 이해하고 각 계층의 기능을 살펴본다.
 - 네트워크 프로토콜의 기초 의미와 역할을 살펴본다.
- 주소의 역할과 기호로 된 이름의 차이와 필요성을 이해한다.
 - 주소와 이름을 변환하는 DNS의 필요성을 이해한다.

01. 네트워크 관련 기초 용어

02. 네트워크의 기능

03. 네트워크 주소의 표현

요약

연습문제

● Preview

일상에서 인터넷이 차지하는 비중은 '일반인들의 인터넷 생활 보편화'라는 간단한 표현만으로도 그 역할을 충분히 설명할 수 있다. 인터넷은 초등학교 입학 전부터 사용하기 시작해 성인이 되어서도 직업에 상관없이 활용하는 범위가 늘면서, 인터넷은 정보 창고로써 지식 정보화 사회를 선도하고 있다.

자신만의 주관적 관점으로 세계를 바라보는 나만의 독립 공간인 인터넷은 현대인의 삶을 새로운 차원으로 변화시키고 있다. 단순히 정보 검색과 전달이라는 기술 체계를 넘어, 인터넷이라는 거대한 광장에 펼쳐진 대화와 공유의 장은 인류의 새로운 패러다임을 이끌고 있다.

인터넷을 올바르게 이해하고 활용하려면 세부 기술에 대한 접근보다는 인터넷 용어를 정확히 이해하고 이를 바탕으로 전체 시스템을 먼저 파악해야 한다. 특히 새로운 용어를 기존 단어의 의미로 막연히 짐작해, 잘못된 지식의 우물에 갇히는 오류를 범해서는 안 된다.

중국의 철학자 노자(老子)는 '도가도비상도(道可道非常道)'라는 유명한 말로 언어의 내재적 한계를 설명했다. 직역하면 '개념적)도를 (언어적)道라고 표현하면 이미 (개념적)道가 아니다'이다. 이 말은 의미가 복잡한 (개념적)도의 실체를 (언어적)道 한 단어로 정확히 표현할 수 없음을 지적한다. 인간의 사고 (Semantic: 의미)는 언어로 표현(Syntax: 문법)되는 그 순간 의미가 왜곡된다고 할 정도로 언어는 근원적인 한계를 지니고 있다. 결과적으로 단어의 의미에 대한 부분적인 이해 부족과 그에 따른 왜곡이 단어로 표현되는 전체 시스템을 이해하는데 방해 요인이 될 수 있다.

1 네트워크 관련 기초 용어

보통 인터넷이라 불리는 네트워크는 그 활용 속도만큼이나 기술이 빠르게 변한다. 더불어 새로운 개념과 이론도 계속 등장하는데, 등장 속도가 너무 빨라 전문가조차 모두 이해하기 어려워한다. 게다가 새로운 용어를 표현할 때는 ‘Transport Control Protocol’ 처럼 전체 단어로 정의하지 않고, ‘TCP’ 처럼 첫 자만 따는 약어 형식을 선호해 혼란을 더욱 부채질한다.

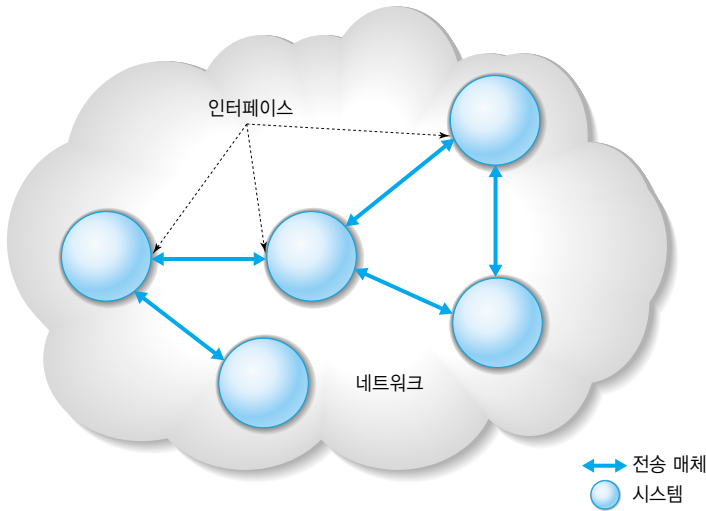
용어의 의미를 올바르게 이해하려면 단어가 지닌 사전적 의미에 선입견을 가지면 안 된다. 새로운 기술에 용어를 부여할 때는 보통 의미가 가장 유사한 기존 단어를 재활용한다. 그러다 보니 단어의 원래 의미를 토대로 새로운 용어를 이해하는 경향이 나타나는데, 이는 새로운 학문을 접하는 데 커다란 장애가 되기도 한다. 그러므로 이 부분을 경계하며, 이 책의 핵심 주제인 네트워크의 기초 용어부터 하나씩 알아보자.

1 네트워크 기초 용어

많은 사람들에게 친숙한 인터넷(Internet)은 연구소, 기업, 학교 등의 소규모 조직에서 사용한 작은 단위의 네트워크(Network)를 서로 연결하면서 발전하였다. 그 과정에서 자연스럽게 연결 방식이 표준화되고 전 세계로 확산되어 오늘날처럼 거대한 인터넷이 되었다.

네트워크를 이해하려면 시스템, 인터페이스, 전송 매체, 프로토콜, 네트워크, 인터넷 같은 용어를 먼저 알아야 한다. 네트워크(Network)는 전송 매체(Transmission Media)로 서로 연결해 데이터를 교환하는 시스템(System)의 모음이다. 시스템과 전송 매체의 연결 지점에 대한 규칙을 인터페이스(Interface)라 하고, 시스템이 데이터를 교환할 때 임의의 통신 규칙을 따르는데 이 규칙을 프로토콜(Protocol)이라 한다. 서로 다른 시스템이 상호 연동해 동작하려면 연동 형식의 통일이 필요하고 이 통일을 표준화(Standardization)라 한다.

[그림 1-1]은 시스템과 전송 매체로 네트워크를 구성한 예다. 시스템은 반드시 컴퓨터일 필요는 없지만 보통 컴퓨터 시스템으로 가정한다. 시스템은 물리적으로 공유하는 전송 매체로 서로 연결한다. 시스템이 전송 매체를 통해 데이터를 교환하려면 반드시 표준화된 프로토콜을 사용해야 한다. 흔히 사용하는 인터넷은 IP(Internet Protocol)라는 네트워크 프로토콜을 사용하는 네트워크 집합체다.



[그림 1-1] 네트워크의 구성

네트워크 구성요소의 개념을 좀 더 자세히 알아보자.

시스템

시스템(System)은 체제라고도 하는데, 내부 규칙에 따라 능동적으로 동작하는 대상이다. 자동차, 커피 자판기, 컴퓨터, 마이크로 프로세서, 하드 디스크 같은 물리적인 대상뿐만 아니라, 신호등으로 교통을 제어하는 운영 시스템, MS 윈도우즈 등의 운영체제, 프로그램의 실행 상태를 의미하는 프로세스 같은 논리적인 대상도 시스템이다.

시스템의 동작에 필요한 외부 입력이 있을 수 있으며, 내부 정보와 외부 입력의 조합에 따른 출력이 있을 수 있다. 작은 시스템이 여러 개 모여 더 큰 시스템을 구성할 수 있으므로 크기로는 시스템을 구분하지 않는다.

인터페이스

인터페이스(Interface)는 시스템과 시스템을 연결하기 위한 표준화된 접근 방법이다. 예를 들어, 컴퓨터 본체와 키보드를 연결하려면 키보드의 잭을 본체의 정해진 위치에 꽂아야 한다. 이를 위해서는 상호 간의 데이터 교환을 위한 RS-232C, USB 같은 논리적인 규격뿐만 아니라, 잭의 크기·모양 같은 물리적인 규격도 표준화되어야 한다. 인터페이스를 이용해 연결하는 시스템은 능력과 권한이 같은 경우도 있지만 서비스를 주고받는 상하 관계일 수도 있다.

전송 매체

시스템끼리 정해진 인터페이스를 연동해 데이터를 전달하려면 물리적인 전송 수단인 전송 매체(Transmission Media)가 반드시 있어야 한다. 전송 매체는 사람의 눈으로 볼 수 있는 동축 케이블을 포함하여 소리를 전파하는 공기, 무선 신호 등 다양하다. 인터페이스는 시스템 간의 연동을 위한 논리적인 규격이고, 인터페이스에서 정해진 규칙은 매체를 통해 전송됨으로써 구현되고 동작된다고 볼 수 있다.

프로토콜

상호 연동되는 시스템이 전송 매체를 통해 데이터를 교환할 때는 특정 규칙을 따르는데, 이 규칙을 프로토콜(Protocol)이라 한다. 일반적으로 프로토콜은 주종(主從) 관계가 아닌 동등한 위치에 있는 시스템 사이의 규칙이라는 측면이 강조되어 인터페이스와 구분된다. 인터페이스는 두 시스템이 연동하기 위한 특정한 접속 지점(Access Point)을 의미하는 경우가 많지만, 프로토콜은 주고받는 정보의 형식과 그 과정에서 발생하는 일련의 절차적 순서에 무게를 둔다.

네트워크

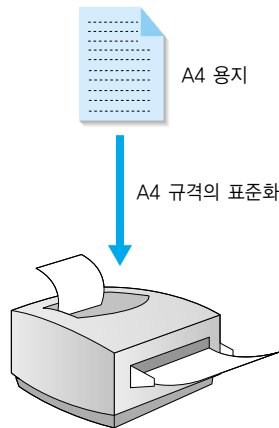
통신용 매체를 공유하는 여러 시스템이 프로토콜을 사용하여 데이터를 주고받을 때 이들을 하나의 단위로 통칭하여 네트워크(Network)라 부른다. 일반적인 컴퓨터 네트워크에서는 물리적인 통신 매체로 연결된 컴퓨터 간에 동일한 프로토콜을 이용해 데이터를 주고받는다. 소규모 네트워크가 모여 더 큰 네트워크를 구성할 수 있는데, 네트워크끼리는 라우터(Router)라는 중개 장비를 사용해 연결한다.

인터넷

인터넷(Internet)은 전 세계의 네트워크가 유기적으로 연결되어 동작하는 통합 네트워크다. 인터넷으로 연결된 시스템과 인터페이스, 전송 매체, 프로토콜 등은 종류가 매우 다양하지만 데이터 전달 기능에 한해서는 공통적으로 IP(Internet Protocol) 프로토콜을 사용한다. 인터넷이라는 용어는 IP의 첫 번째 단어인 Internet에서 유래되었다.

표준화

서로 다른 시스템이 상호 연동해 동작하려면 표준화(Standardization)라는 연동 형식의 통일이 필요하다. 예를 들어, 프린트 용지를 생각해보자. 일반적으로 프린터와 프린트 용지를 만드는 회사는 다르다. 하지만 사전에 A4 규격이라는 통일된 틀을 만들어두었기 때문에 여러 회사 제품의 프린트 용지를 자유롭게 사용할 수 있는 것이다.



[그림 1-2] A4 규격의 표준화

현대 산업 사회가 눈부시게 성장한 이면에는 증기 기관의 개발을 시초로 한 에너지 동력원의 발전이 있었다. 그러나 이와 다른 관점에서 더 근원적으로 살펴보면 표준화 원리를 바탕으로 한 '레고 조합'의 개념이 산업 전반에 존재함을 이해해야 한다.



여기서 잠깐

IP 프로토콜

IP는 'Internet Protocol'의 약자이지만, 고유 명사화되어 있으므로 IP 프로토콜이라고 부를 수 있다. TCP, UDP 등 대부분의 네트워크 프로토콜도 마찬가지다.

2- 시스템 기초 용어

[그림 1-1]과 같이 네트워크는 외형적으로 시스템과 전송 매체로 구성된다. 데이터 통신을 위한 전송 매체는 전송 대역, 전송 속도, 전송 오류율 같은 물리적인 특성이 주 관심사므로 논리적인 기능은 비교적 단순하다. 그러므로 여기서는 시스템만 좀 더 살펴본다. **시스템은 전송 매체를 이용해 논리적으로 다양한 연동 형태로 구성할 수 있어 개념의 확장이 폭넓고 복잡하다.**

시스템의 구분

네트워크를 구성하는 시스템이 반드시 컴퓨터처럼 복잡한 기능을 수행해야 하는 것은 아니지만, **일반적으로 컴퓨터 시스템**으로 가정한다. 네트워크 시스템은 수행 기능에 따라 다음과 같이 다양하게 부를 수 있다.

■ 노드

노드(Node)는 인터넷에 연결된 시스템을 가장 일반화한 용어다. **데이터를 주고받을 수 있는 모든 시스템**을 통칭한다.

■ 호스트

호스트(Host)는 일반적으로 **컴퓨팅 기능이 있는 시스템**이다. 일반 사용자가 응용 프로그램을 실행할 수 있어 사용자가 네트워크에 접속하는 창구 역할을 한다.

■ 클라이언트

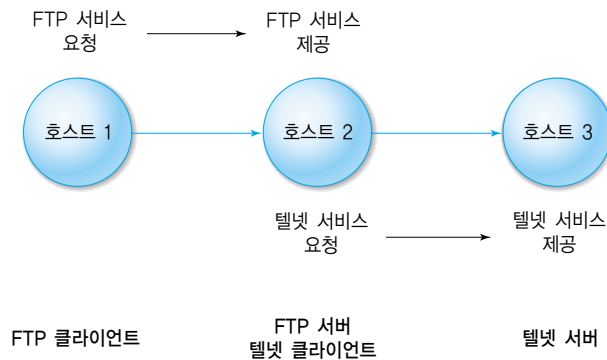
호스트를 세분할 때 호스트 사이에 제공되는 서비스를 기준으로 클라이언트와 서버로 나눈다. **클라이언트(Client)**는 **서비스를 요청하는 시스템**이다. 호스트는 다양한 서비스를 서로 주고받기 때문에 임의의 호스트가 클라이언트나 서버로 고정되지는 않는다. 이용하는 서비스의 종류에 따라서 클라이언트가 될 수도 있고, 서버가 될 수도 있다. 그러므로 특정 서비스를 기준으로 클라이언트와 서버라는 상대적 용어로 구분한다.

■ 서버

서버(Server)는 서비스를 주고받는 호스트들의 관계에서 **특정 서비스를 제공하는 시스템**이다. 일반적으로 서버는 클라이언트보다 먼저 실행 상태가 되어 클라이언트의 요청에 대기해야 한다. 그리고 영원히 종료하지 않으면서 **클라이언트의 요청이 있을 때마다 서비스를 반복해서 제공**해야 한다.

클라이언트와 서버

[그림 1-3]은 임의의 서비스를 기준으로 클라이언트와 서버의 상대적인 관계를 설명한다. 그림에서 호스트 2는 파일 **송수신 기능을 제공하는 FTP**(File Transfer Protocol) 서비스를 제공하며, 호스트 3은 **원격 단말기 기능인 텔넷(Telnet)** 서비스를 제공한다.



[그림 1-3] 클라이언트와 서버

FTP 서비스를 먼저 살펴보면, 호스트 1은 호스트 2에 FTP 서비스를 요청한다. 따라서 FTP 서비스를 기준으로 하면 호스트 1이 클라이언트가 되고, 호스트 2는 서버가 된다. 반면, 텔넷 서비스는 호스트 2가 호스트 3에 요청한다. 텔넷 서비스를 기준으로 하면 호스트 2가 클라이언트고, 호스트 3은 서버다. 따라서 호스트 2는 사용하는 서비스의 종류에 따라 클라이언트가 되기도 하고, 서버가 되기도 한다. 결론적으로 클라이언트와 서버라는 용어는 **서비스 이용의 상대적 위치에 따라 결정된다**.

클라이언트와 서버의 명칭을 **호스트 시스템에 부여해 사용**할 수도 있다. 특히 다양한 서비스 기능을 제공하는 대형 시스템을 서버로 설정해 다수의 클라이언트가 서버에 접속해 서비스를 이용하도록 할 수 있다. 그러나 기능적인 관점에서는 [그림 1-3]처럼 호스트에서 실행되는 응용 서비스별로 구분하는 것이 더 정확하다. **인터넷에서 네트워크 서비스의 기능은 대부분 응용 프로그램으로 구현되므로 보통 클라이언트 프로세스 서버 프로세스라는 호칭이 더 자연스러울 수 있다.**

2 네트워크의 기능

네트워크는 다수의 시스템을 전송 매체로 연결한 것이다. 소규모 네트워크를 전송 매체로 연결해 더 큰 네트워크로 확장할 수 있다.

컴퓨터 네트워크는 (외형상) 호스트 시스템과 전송 매체로 구분된다. 하지만 이런 물리적인 형태와는 별도로, 그들이 수행하는 내부 기능을 기초로 다양한 구조로 구분할 수 있다. 특히, 호스트가 제공하는 서비스를 사용하는 방식에 따라서 계층 모델로 나눠 설명할 수 있다.

1 계층 모델

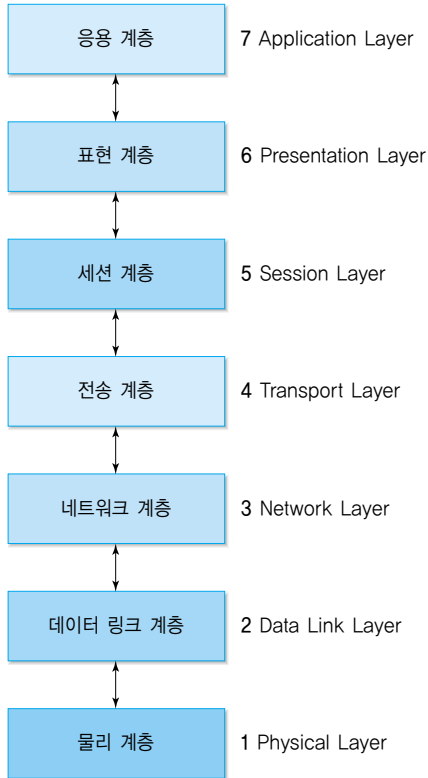
특징이 다른 여러 호스트를 서로 연결해 통신하려면 연결 방식을 표준화해야 한다. 국제 표준화 단체인 ISO(International Standard Organization)에서는 OSI(Open System Interconnection) 7계층 모델을 제안하였다. 또한 네트워크에 연결된 시스템이 갖추어야 할 기능을 상세히 정의하였다.

OSI 7계층 모델 소개

OSI 7계층 모델(OSI 7 Layer Model)에 따르면, 네트워크에 연결된 호스트는 [그림 1-4]와 같이 7개 계층으로 모듈화된 통신 기능을 갖추어야 한다. 일반 사용자는 OSI 7계층 맨 위에 있는 응용 계층을 통해 데이터의 송수신을 요청하며, 이 요청은 하위 계층으로 순차적으로 전달되어 맨 아래에 있는 물리 계층을 통해 상대 호스트에 전송된다. 그리고 요청이 각 계층으로 하달되는 과정에서 계층별로 담당하는 기능을 수행해 데이터를 안전하게 전달한다.

데이터를 수신한 호스트에서는 송신 호스트와 반대 과정으로 처리한다. 즉, 물리 계층으로 들어온 데이터는 순차적인 상향 전달 과정을 거쳐 응용 계층으로 올라간다. 그리고 처리가 완료된 결과를 회신할 때는 반대 과정을 순차적으로 밟아서 송신 호스트로 되돌아간다.

따라서 데이터를 송수신하는 최종 주체는 송수신 호스트 양쪽에 위치한 응용 계층이며, 하부의 계층은 데이터 전송에 필요한 기능만 나누어 처리한다.



[그림 1-4] OSI 7계층 모델

OSI 7계층 모델의 계층별 기능

OSI 7계층 모델의 각 계층은 고유 기능을 수행하며, 하위 계층이 바로 위 계층에 서비스를 제공하는 형식으로 동작한다. 여기서는 각 계층에서 제공하는 기능만 간단히 살펴보고 자세한 내용은 2장에서 설명한다.

■ 물리 계층

네트워크에서 호스트가 데이터를 전송하려면 반드시 전송 매체로 연결되어 있어야 한다.

물리 계층(Physical Layer)은 호스트를 전송 매체와 연결하기 위한 인터페이스 규칙과 전송 매체의 특성을 다룬다.

■ 데이터 링크 계층

물리 계층으로 데이터를 전송하는 과정에서는 잡음(Noise) 같은 여러 외부 요인에 의해 물리적 오류가 발생할 수 있다. 데이터 링크 계층(Data Link Layer)은 물리적 전송 오류를 감지(Sense)하는 기능을 제공해 송수신 호스트가 오류를 인지할 수 있게 해준다. 발생 가능한 물리적 오류의 종류에는 데이터가 도착하지 못하는 데이터 분실과 내용이 깨져서 도착하는 데이터 변형이 있다. 일반적으로 컴퓨터 네트워크에서의 오류 제어(Error Control)는 송신자가 원 데이터를 재전송(Retransmission)하는 방법으로 처리한다.

■ 네트워크 계층

송신 호스트가 전송한 데이터가 수신 호스트까지 도착하려면 여러 중개 시스템을 거친다. 이 과정에서 데이터가 올바른 경로를 선택할 수 있도록 지원하는 계층이 네트워크 계층(Network Layer)이다. 중개 시스템의 기능은 일반적으로 라우터(Router) 장비가 수행한다. 네트워크 부하가 증가하면 특정 지역에 혼잡(Congestion)이 발생할 수 있는데, 이것도 데이터의 전송 경로와 관계가 있으므로 네트워크 계층이 제어한다.

■ 전송 계층

컴퓨터 네트워크에서 데이터를 교환하는 최종 주체는 호스트 시스템이 아니고, 호스트에서 실행되는 프로세스다. 전송 계층(Transport Layer)은 송신 프로세스와 수신 프로세스 간의 연결(Connection) 기능을 제공하기 때문에 프로세스 사이의 안전한 데이터 전송을 지원한다. 계층 4까지의 기능은 운영체제에서 시스템 콜(System Call) 형태로 상위 계층에 제공하며, 계층 5~7의 기능은 사용자 프로그램으로 작성된다.

■ 세션 계층

세션 계층(Session Layer)은 전송 계층의 연결과 유사한 세션 연결을 지원하지만 이보다 더 상위의 논리적 연결이다. 즉, 응용 환경에서의 사용자 간의 대화(Dialog) 개념의 연결로 사용되기 때문에 전송 계층의 연결과는 구분된다.

■ 표현 계층

표현 계층(Presentation Layer)은 전송되는 데이터의 의미(Semantic)를 잃지 않도록 올바르게 표현하는 방법(Syntax)을 다룬다. 정보를 교환하는 시스템이 표준화된 방법으로 데이터를 인식할 수 있도록 해주는 역할을 한다. 표현 계층의 주요 기능은 압축과 암호화다. 동영상과 같은 대용량의 멀티미디어 데이터를 압축(Compression)해 전송 데이터의

양을 줄일 수 있다. 암호화는 외부의 침입자로부터 데이터를 안전하게 보호하는 기능인데, 전자상거래가 증가하면서 중요성이 커지고 있다.

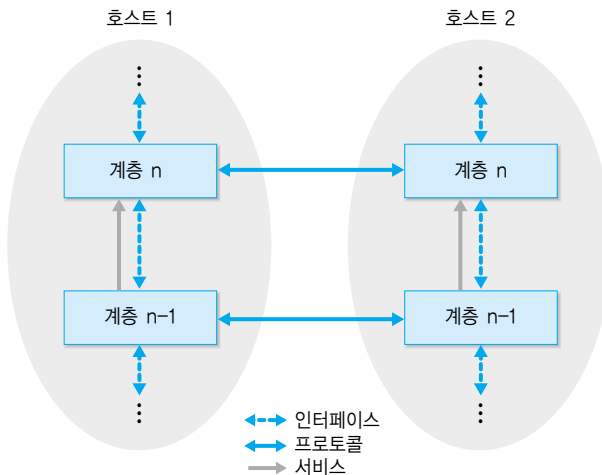
■ 응용 계층

응용 계층(Application Layer)은 사용자의 다양한 네트워크 응용 환경을 지원한다. 기능은 한 분야에 한정되지 않고 매우 광범위하다.

프로토콜과 인터페이스

네트워크 사용자가 통신한다는 의미는 데이터를 주고받을 수 있음을 의미한다. 최종 사용자가 데이터를 보내고 받으려면 양쪽 호스트에서 실행되는 OSI 7계층의 모듈이 유기적으로 연동되어야 한다. 호스트끼리 통신하는 과정에서는 각 계층의 모듈이 상대 호스트의 동일 계층과 개별적으로 논리적 통신을 수행해야 한다. 예를 들어, 한 호스트의 계층 n 모듈은 상대 호스트의 계층 n 모듈과 통신한다. 각 계층은 정해진 방식으로 통신하는데, 이 과정에서 필요한 규칙을 프로토콜(Protocol)이라 한다.

호스트의 계층 사이에는 인터페이스(Interface)라는 규칙이 존재하고, 하위 계층이 상위 계층에 제공하는 인터페이스를 특별히 서비스(Service)라고 부른다. [그림 1-5]는 서로 다른 호스트에 위치한 계층 모델에서 두 호스트의 통신을 지원하기 위한 모듈 간의 관계를 프로토콜, 인터페이스, 서비스의 관점에서 설명한다.

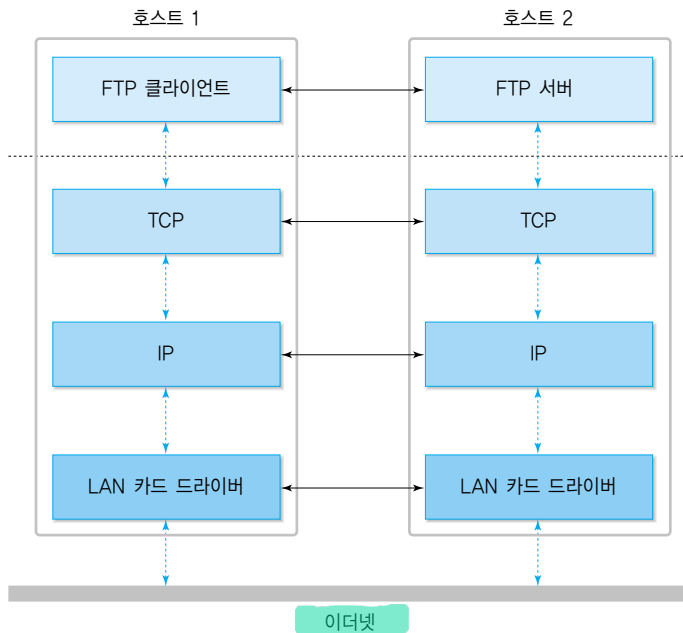


[그림 1-5] 인터페이스와 프로토콜

호스트 1과 호스트 2의 계층 n 프로토콜이 서로 통신하려면 계층 n-1 프로토콜의 도움이 필요하다. 즉, 호스트 1의 계층 n이 호스트 2의 계층 n에 데이터를 전송하는 과정은 계층 n-1을 통해 이뤄진다. 먼저, 호스트 1의 계층 n-1에 전송 데이터를 주어 호스트 2에 전송하도록 부탁한다. 그러면 호스트 1의 계층 n-1은 다시 하위 계층의 도움을 받아 호스트의 2의 계층 n-1에 데이터를 보낸다. 마지막으로 호스트 2의 계층 n-1이 받은 데이터를 계층 n에 올려 줌으로써 데이터 전송을 완료한다. 이 원리는 7개 계층에 모두 적용되며, 상대 호스트에 데이터를 실제로 전송하는 것은 맨 아래의 물리 계층이다. 물리 계층 위에 위치한 계층 프로토콜은 정해진 기능을 수행하기 위한 논리적인 통신을 하는 것이다.

인터넷 계층 구조

[그림 1-6]은 FTP 프로그램을 이용하는 경우를 예로 들어 계층 구조를 설명한다. 인터넷에서는 IP(Internet Protocol)가 네트워크 계층의 기능을 수행하며, TCP(Transmission Control Protocol)와 UDP(User Datagram Protocol)가 전송 계층의 기능을 수행한다. 전송 계층 이하의 기능은 호스트의 운영체제에서 구현되며 FTP, 텔넷, 전자 메일 같은 응용 프로그램은 사용자 프로그램 환경에서 계층 5~7이 함께 구현된다.



[그림 1-6] FTP의 계층 구조

[그림 1-6]과 같이 양쪽 호스트에는 동일한 기능을 하는 프로토콜 스택(Stack)이 각각 존재한다. 호스트 사이에는 중개 기능을 수행하는 라우터가 존재한다. 인터넷에서는 IP 프로토콜이 중개 기능을 하므로 라우터에는 계층 3까지의 기능이 구현되어 있다.

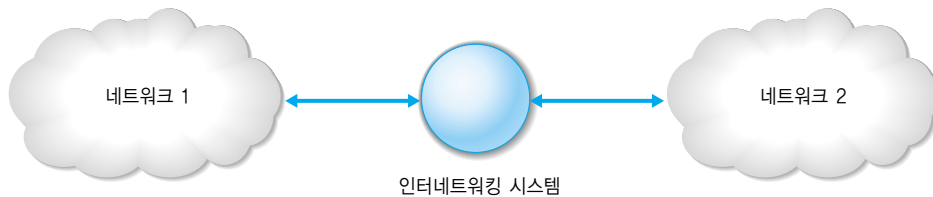
FTP 클라이언트가 FTP 서버에 데이터를 전송하려면 먼저 하위의 TCP에 데이터를 보내야 한다. TCP로 보내진 데이터는 IP 프로토콜과 LAN 카드를 거쳐서 FTP 서버가 위치하는 호스트에 전달된다. 서버 호스트에 도착한 데이터는 송신 순서의 반대인 LAN 카드, IP, TCP를 거쳐서 FTP 서버 프로그램에 도착한다.

2 인터네트워킹

네트워크와 네트워크의 연결을 인터네트워킹(Internetworking)이라 한다. 연결하는 네트워크 수가 증가할수록 복잡도가 커진다. 인터넷은 IP 프로토콜을 지원하는 전 세계의 모든 네트워크가 연결된 시스템을 의미하며, 라우터라는 중개 장비를 사용해 네트워크를 연결한다.

네트워크의 연결

[그림 1-7]처럼 서로 독립적으로 운영되는 두 네트워크가 연동되어 정보를 교환하려면, 이를 적절히 연결하여 데이터를 중개할 수 있는 인터네트워킹 시스템이 필요하다. 여기서 네트워크가 연동한다는 의미는 물리적인 연결뿐만 아니라, 데이터 중개에 필요한 상위 통신 프로토콜을 모두 지원함을 뜻한다.



[그림 1-7] 인터네트워킹



여기서 잠깐

프로토콜 스택

계층 구조로 이루어진 통신 프로토콜의 집합이다.

인터넷워킹 시스템에 의해 연결된 두 네트워크는 물리적으로 같은 종류일 필요는 없다. 상위 계층 프로토콜이 지원하는 논리적 기능이 달라도 된다. 하지만 인터넷워킹 시스템은 양쪽 네트워크에 대한 서로 다른 물리적·기능적 인터페이스를 모두 지원해야 한다. 즉, [그림 1-7]에서 인터넷워킹 시스템의 왼쪽은 네트워크 1과 연동할 수 있고 오른쪽은 네트워크 2와 연동할 수 있어야 한다. 또한 이 과정에서 데이터 표현 방식을 포함해 프로토콜이 다르면 필요한 변환 작업을 수행해야 한다. 이렇게 인터넷워킹 시스템은 둘 이상의 네트워크를 유기적으로 연동할 수 있다.

게이트웨이

인터넷워킹 기능을 수행하는 시스템을 일반적으로 게이트웨이(Gateway)라 부른다. 게이트웨이는 기능에 따라 종류가 다양하지만 리피터, 브리지, 라우터 등이 가장 일반적인 구분이다.

■ 리피터

리피터(Repeater)는 물리 계층의 기능을 지원한다. 물리적 신호는 전송 거리가 멀면 감쇄되기 때문에 중간에 이를 보완해 주어야 한다. 리피터는 한쪽에서 입력된 신호를 물리적으로 단순히 증폭하여 다른 쪽으로 중계하는 역할을 한다.

■ 브리지

브리지(Bridge)는 리피터 기능에 데이터 링크 계층의 기능이 추가된 게이트웨이이다.

■ 라우터

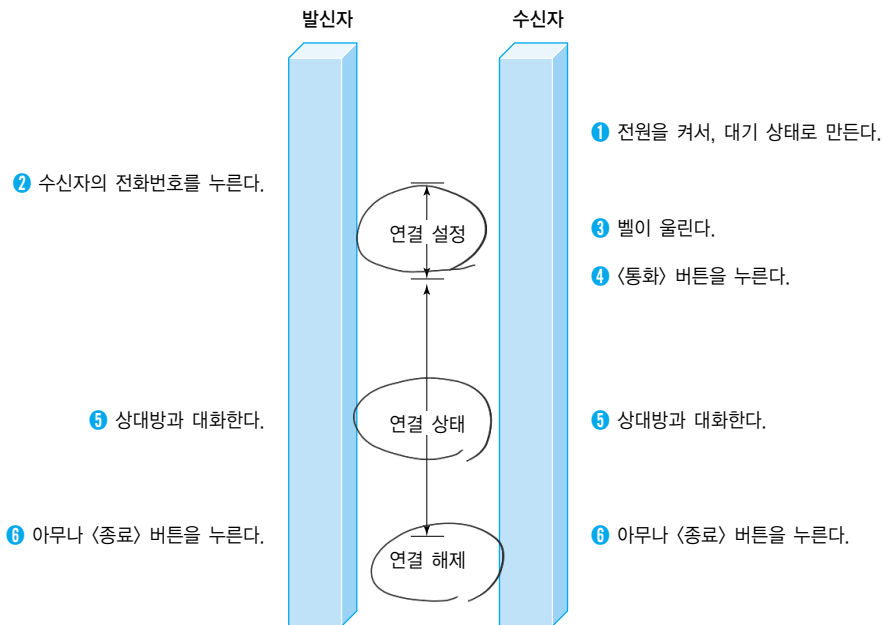
라우터(Router)는 물리 계층, 데이터 링크 계층, 네트워크 계층의 기능을 지원한다. 네트워크 계층은 경로 선택 기능을 제공해야 하므로 임의의 네트워크에서 들어온 데이터를 어느 네트워크로 전달할지 판단할 수 있어야 한다. 이를 지원하기 위하여 라우터는 자신과 연결된 네트워크와 호스트 정보를 유지하고 관리함으로써, 어떤 경로를 이용할 수 있고, 어떤 경로를 이용해야 빠르게 전송할 수 있는지를 판단한다. 이런 판단을 위한 정보는 일반적으로 라우팅 테이블(Routing Table)에 보관한다.

3- 프로토콜

프로토콜(Protocol)은 통신 시스템이 데이터를 교환하기 위해 사용하는 통신 규칙이다. OSI 7계층 모델에서는 각 계층에서 수행되는 프로토콜이 서로 독립적이라고 간주한다. 따라서 계층 1에는 계층 1끼리 통신할 수 있는 프로토콜이 존재하고, 계층 2에는 계층 2끼리 통신할 수 있는 프로토콜이 존재한다.

프로토콜 예

[그림 1-8]은 휴대폰으로 상대방과 통화하는 순서 규칙을 설명하는데, 이 규칙을 통칭하여 프로토콜이라고 정의할 수 있다. 시간은 위에서 아래 방향으로 흐르고, 왼쪽의 발신자가 수신자에게 통화를 시도하고 있다. 연결을 설정할 때는 항상 발신자의 요청에 의하여 시작하며 수신자는 미리 연결 대기 상태여야 한다. 연결의 설정은 상호 간의 합의하에 이루어지지만, 종료는 한쪽의 일방적인 종결 선언으로도 가능하다.



[그림 1-8] 전화 연결을 위한 규칙

먼저 ① 수신자의 휴대폰 전원이 켜진 대기 상태여야 한다. ② 발신자가 수신자의 전화번호를 누르면(또는 번호를 입력한 후 〈통화〉 버튼을 누르면) ③ 수신자 휴대폰의 벨이 울린다.

④ 수신자가 <통화> 버튼을 눌러 전화를 받으면 연결이 설정된다(수신자가 벨 소리를 무시하고 전화를 받지 않으면 연결이 되지 않으므로 연결 설정은 반드시 양자 합의 하에서만 이루어진다).

일단 연결이 설정되면 연결이 해제되기 전까지 ⑤ 데이터를 주고받을 수 있다. ⑥ 연결 해제는 외형상 한쪽의 일방적인 종료에 의해 이루어지지만 실제로는 대화 과정에서 쌍방의 합의로 이루어진다. 이런 상황은 컴퓨터 네트워크에도 비슷하게 적용된다.

데이터 단위

네트워크 프로토콜을 사용해 데이터를 교환할 때는 먼저 데이터를 특정 형태로 규격화하는 작업이 필요하다. 예를 들어, 우편물을 보낼 때 정해진 규격 봉투를 사용하는 것과 같다. 규격 봉투에 보내는 사람, 받는 사람, 우표 등의 규칙에 따라 정보를 기입하는 것처럼, 네트워크에서도 데이터를 프로토콜에 맞춰 묶어준다.

OSI 7계층 모델의 각 계층에서 규격화된 데이터는 다음과 같이 고유 명칭이 있다. 하지만 보통 계층에 상관없이 PDU(Protocol Data Unit)라고 부른다.

- **APDU(Application Protocol Data Unit)** : 응용 계층에서 사용하는 데이터의 단위
- **PPDU(Presentation Protocol Data Unit)** : 표현 계층에서 사용하는 데이터의 단위
- **SPDU(Session Protocol Data Unit)** : 세션 계층에서 사용하는 데이터의 단위
- **TPDU(Transport Protocol Data Unit)** : 전송 계층에서 사용하는 데이터의 단위. 인터넷에서 사용하는 전송 계층 프로토콜인 TCP에서는 세그먼트(Segment)라고 부르고, UDP에서는 데이터그램(Datagram)이라고 부른다. 데이터그램은 일반적인 의미로도 자주 사용한다.
- **NPDU(Network Protocol Data Unit)** : 네트워크 계층에서 사용하는 데이터의 단위. 보통 패킷(Packet)이라고 부른다. 패킷은 원래 네트워크 계층의 IP 프로토콜에서 유래되었지만 요즘은 일반 용어로 많이 사용한다.
- **DPDU(Data Link Protocol Data Unit)** : 데이터 링크 계층에서 사용하는 데이터의 단위. 보통 프레임(Frame)이라고 부른다.



네트워크 주소의 표현

시스템을 설계할 때는 기능이나 목적과 함께 구분자(Identifier)를 부여하는 방법을 가장 먼저 고려해야 한다. 디지털화된 컴퓨터에서는 구분자를 숫자로 된 주소로 표현할 수밖에 없다. 그러나 이런 주소 표현 방식은 일반 사용자에게 불편하므로 보통 외우기 쉬운 기호 형식의 이름을 추가로 부여한다. 주소와 이름은 일대일 관계가 이루어지며, 이들을 연결하는 기능이 필요하다. 대상을 유일하게 구별하는 구분자는 일반적으로 다음 네 가지 특징이 있다.

- **유일성** : 구분자의 가장 중요한 역할은 대상 시스템을 서로 구분하여 지칭하는 것이다. 따라서 서로 다른 시스템이 같은 구분자를 갖지 않는 유일성을 보장해야 한다.
- **확장성** : 시스템은 활용의 보편화가 진행되면서 자연스럽게 확장 과정을 거친다. 따라서 사용하는 구분자의 양도 증가한다. 시스템의 최대 수용 규모를 예측하여 구분자의 최대 한계를 올바르게 설정하지 않으면, 표현할 수 있는 공간의 크기가 제한되어 시스템의 확장성도 제한된다.
- **편리성** : 일반적으로 시스템의 설계 과정에서 부여되는 구분자는 시스템의 내부 처리 구조를 효율적으로 운용할 수 있도록 해준다. 컴퓨터 시스템은 내부적으로 숫자 기반으로 처리되기 때문에 구분자의 체계도 숫자 위주다. 또 배치, 검색 등을 원활하게 수행하기 위해 보통 일반인이 의미를 이해할 수 없는 형식이다. 이처럼 시스템 내부 동작에 종속된 구분자 주소 체계는 사용자가 의미를 파악하기 어렵기 때문에 기호로 된 이름을 추가로 부여해야 한다. 따라서 하나의 시스템이 숫자로 된 주소와 문자로 된 이름을 모두 가지므로 이를 매핑(Mapping)하는 기능이 필요하다.
- **정보의 함축** : 구분자는 응용 환경에 필요한 다양한 정보를 포함하는 경우가 많다. 예를 들어, 주민번호에는 생년월일, 성별 구분 등의 의미가 부여되어 있다. 집주소도 광역시부터 시작해 지역을 소규모로 분할하는 구조를 가져 위치를 쉽게 가늠할 수 있다. 이처럼 구분자는 응용 환경에 적절히 대응할 수 있는 정보를 포함해야 한다.

1 주소와 이름

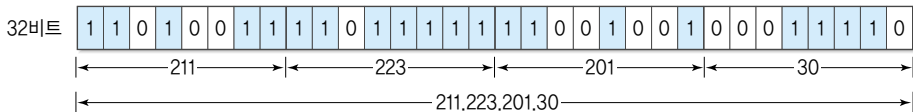
시스템을 지칭하는 구분자는 내부적으로 처리·관리되는 숫자 기반의 주소(Address)와 사용자의 이해와 편리성을 도모하는 기호로 된 이름(Name)을 함께 제공해야 한다. 일반 사용자는 내부 주소를 몰라도 이름만으로 시스템에 접근할 수 있어야 하며, 이름과 주소를 연결하는 방법은 내부적으로 처리되어야 한다.

네트워크 규모가 작아 관리하는 시스템이 작으면 간단한 형식의 주소와 이름을 사용할 수 있다. 따라서 주소와 이름을 관리하는 시스템도 그리 복잡하지 않다. 그러나 관리 대상이 많아지면 주소와 이름 공간도 커지고, 이를 관리하는 시스템의 기능도 복잡해진다.

네트워크에는 여러 종류의 주소와 이름이 존재한다. 각 계층의 기능을 담당하는 프로토콜마다 주소를 별도로 관리하기 때문이다. 예를 들어, IP 프로토콜은 호스트를 구분하기 위하여 IP 주소를 사용하며, 데이터 링크 계층에서는 LAN 카드별로 MAC 주소를 따로 부여한다. 전송 계층을 수행하는 TCP에서는 하나의 호스트에서 동시에 수행되는 프로세스마다 별도의 포트(Port) 주소를 할당하고 관리한다.

IP 주소

IP 주소(IP Address)는 네트워크 계층의 기능을 수행하는 IP 프로토콜이 호스트를 구분하려고 사용하는 주소 체계다. 임의의 호스트를 인터넷에 연결하려면 반드시 IP 주소를 할당받아야 한다. IP 주소는 32비트의 이진 숫자로 구성되는데, 보통 8비트씩 네 부분으로 나누어 십진수로 표현한다. [그림 1-9]처럼 32비트의 이진수 11010011 1101111 11001001 00011110은 일반 사용자가 외우기 쉽지 않기 때문에 이를 네 개의 10진수로 변환한 후 각각을 점(.)으로 구분한 211.223.201.30으로 표현한다.

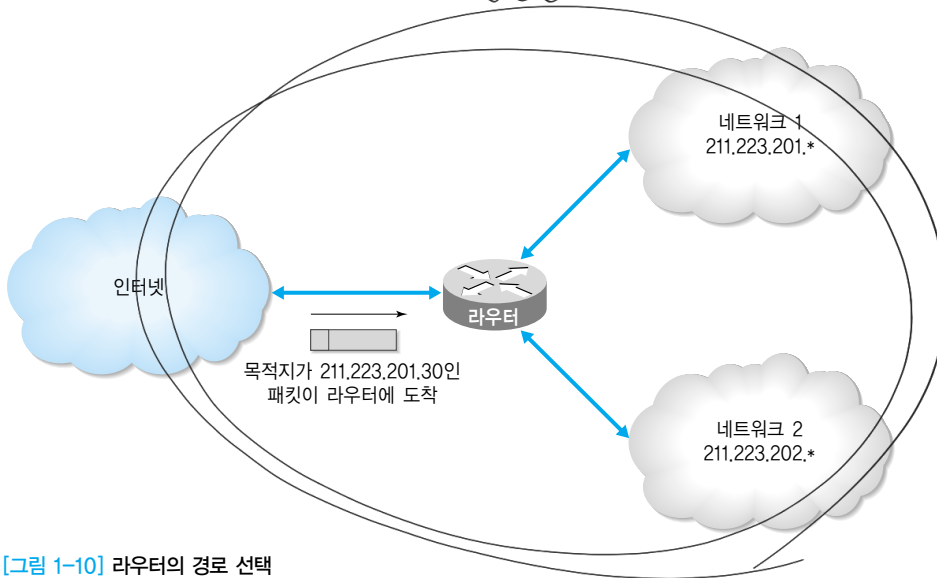


[그림 1-9] IP 주소의 표현

IP 주소는 유일성을 보장하기 위해 국제 표준화 기구에서 전체 주소를 관리하고 할당해 중복 주소의 사용을 원칙적으로 차단한다. IP 프로토콜이 처음 개발될 당시에는 현재처럼 폭넓게 활용되리라 예측하지 못했다. 따라서 IP 주소로 표현할 수 있는 최대 주소 공간의 크기를

32비트로 제한함으로써 확장성에 많은 문제점이 야기되고 있다. 이를 해결하려고 새로운 프로토콜 IPv6(Internet Protocol Version 6)에서는 주소 표현 공간을 128비트로 확장했다. 그리고 현재의 IP 프로토콜은 IPv6과 구분하기 위해 IPv4로 표현한다.

IP 주소는 임의로 할당되는 것이 아니라 어떤 규칙에 따라 인접한 숫자를 그룹으로 묶어 관리한다. 따라서 경로 선택 시 IP 주소를 중요한 기준으로 활용할 수 있다. [그림 1-10]에서 네트워크 1은 IP 주소가 211.223.201로 시작하는 호스트가, 네트워크 2는 211.223.202로 시작하는 호스트가 구성하고 있다. 왼쪽 인터넷에 있는 호스트에서 보낸 패킷이中间的 라우터에 도착했는데, 이 패킷의 목적지 주소가 211.223.201.30라면 당연히 네트워크 1로 중개 해주어야 한다. 이처럼 인터넷에서 IP 주소는 패킷의 경로를 찾는 데 중요한 역할을 한다.



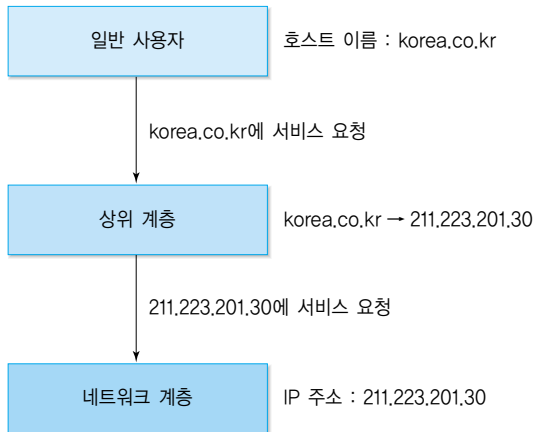
[그림 1-10] 라우터의 경로 선택

호스트 이름

인터넷에서 특정 호스트와 연결하려면 반드시 호스트의 IP 주소를 알아야 한다. 그런데 숫자로 된 IP 주소는 기억하기 힘들어 보통 의미 파악이 쉬운 문자로 된 호스트 이름을 사용한다.

[그림 1-11]은 일반 사용자가 문자로 된 호스트 이름을 사용하였을 때 IP 주소로 변환되는 과정을 보여준다. 맨 밑에 있는 네트워크 계층의 IP 프로토콜에서는 호스트를 구분하려고 IP 주소만 사용한다. 그에 비해 일반 사용자는 IP 주소 대신 문자로 된 호스트 이름을 사용하기 때문에 중간 계층에서 이를 변환하는 기능을 수행해야 한다. 일반적으로 FTP, 텔넷 같은

네트워크 응용 프로그램은 실행시 사용자에게 호스트 이름을 명령어 인수로 입력받는다. 따라서 가장 먼저 수행할 작업은 **DNS(Domain Name System)**라는 이름과 주소 변환 기능을 이용해 IP 주소를 얻는 것이다.



[그림 1-11] 호스트 이름과 IP 주소의 변환

DNS에서는 호스트 이름을 국가 도메인, 단체 종류, 단체 이름, 호스트라는 네 계층 구조로 나누고, 이들을 점(.)으로 구분해 표기한다. 예를 들어, zebra.korea.co.kr과 같은 호스트 이름은 대한민국(kr)에 있는 일반회사(co) 중에서 korea라는 이름의 회사에 소속된 zebra라는 호스트다.

〈호스트〉.〈단체 이름〉.〈단체 종류〉.〈국가 도메인〉

국가 도메인은 호스트가 위치한 국가의 이름을 두 글자의 약자로 표시한다. [표 1-1]처럼 한 국은 kr, 일본은 jp로 나라마다 고유한 국가 도메인이 존재한다. 단체 종류는 기관의 성격에 따라 부여하고 사용하는 형태는 [표 1-2]와 같다. co(company)는 회사, ac(academy)는 교육 기관, go(government)는 정부 소속 기관이다.

[표 1-1] 국가 도메인

국가 도메인	해당 국가명
kr	한국
jp	일본
us	미국

[표 1-2] 단체 종류

단체 종류	기관 성격
co	회사
ac	교육 기관
go	정부 기관

단체 이름은 단체를 상징하는 이름을 사용한다. 예를 들어, 회사는 회사명을, 학교는 학교 이름을 사용한다. 마지막으로 호스트는 소속 단체의 네트워크 관리자가 내부 규칙에 따라 부여한 이름을 사용한다.

2 주소 정보의 관리

일반 사용자가 호스트를 지칭할 때 사용하는 호스트 이름을 **도메인 이름(Domain Name)**이라고 하며, 인터넷에서는 이를 IP 주소로 변환하는 작업이 필요하다. 초기 인터넷에서는 아주 간단한 방법으로 호스트 이름과 IP 주소가 변환되었으나, 지금은 DNS라는 분산 데이터베이스 기능을 사용해 체계적으로 수행한다.

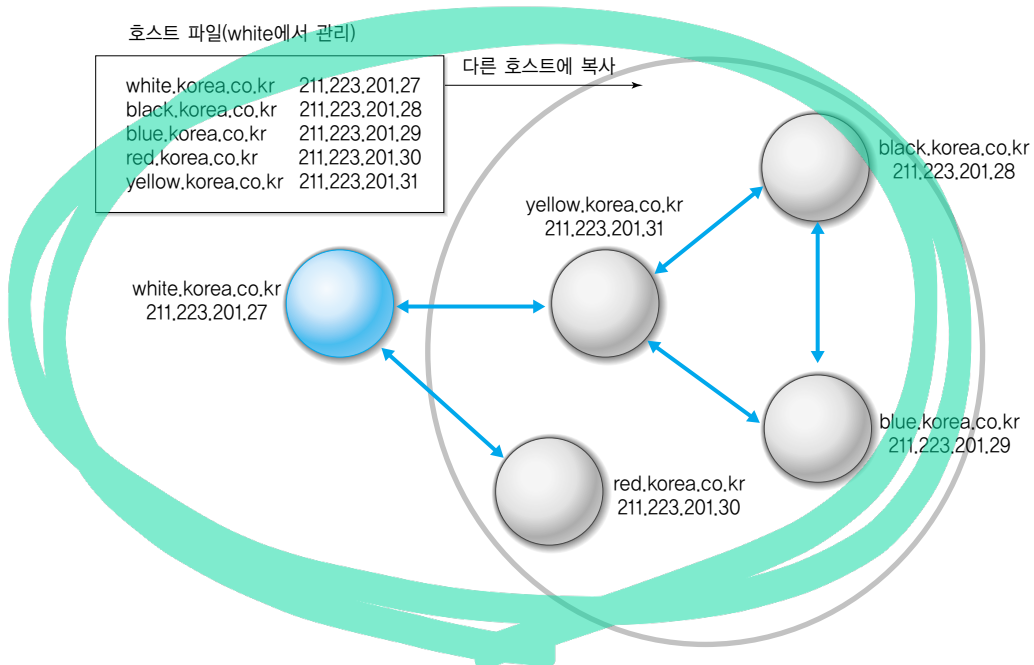
호스트 파일

호스트 이름과 IP 주소를 변환하는 가장 간단한 방법은 특정 파일(예 UNIX 시스템의 /etc/hosts)에 호스트 이름과 IP 주소의 조합을 보관하는 것이다. 네트워크 응용 프로그램에서는 사용자가 입력한 호스트 이름을 검색해 대응하는 IP 주소 정보를 쉽게 얻을 수 있다. 호스트 파일은 다음과 같이 한 행에 호스트 하나의 정보가 기록되며, 일반 텍스트 문서로 보관된다. 예를 들어 호스트 이름이 white.korea.co.kr인 시스템의 IP 주소는 211.223.201.27이다.



white.korea.co.kr	211.223.201.27
black.korea.co.kr	211.223.201.28
blue.korea.co.kr	211.223.201.29
red.korea.co.kr	211.223.201.30
yellow.korea.co.kr	211.223.201.31

네트워크 관리자는 관리 대상이 되는 모든 컴퓨터의 호스트 정보를 주기적으로 갱신하고, 이 정보를 네트워크에 있는 모든 컴퓨터가 복사하도록 함으로써 정보의 일관성을 유지해야 한다. [그림 1-12]에서 네트워크 관리자가 white.korea.co.kr에서 호스트 정보를 갱신하는 작업을 한다고 가정하면 갱신한 정보를 다른 네 개의 호스트가 복사할 수 있게 해주어야 한다. 호스트 파일을 갱신하고 복사하는 작업은 보통 시스템 관리자가 수동으로 한다.



[그림 1-12] 호스트 파일의 복사

호스트가 추가되거나 삭제되면 먼저 네트워크 관리자의 호스트에서 갱신 작업이 이루어진다. 그런데 호스트 파일 갱신은 생각보다 자주 발생하지 않기 때문에 호스트 파일을 복사하는 작업도 흔치 않다. 이는 시스템 관리자가 잦은 변경을 원하지 않아 급하지 않은 갱신은 부분적으로 늦추기 때문이기도 하다.

DNS

호스트 파일에 의한 주소와 이름 정보의 관리는 간단하지만 대부분 수동으로 이루어진다는 단점이 있다. 따라서 호스트 수가 증가할수록 네트워크 관리자가 호스트 파일을 갱신하고 복사하는 작업에 시간과 노력이 더 많이 든다. 특히 인터넷처럼 전 세계의 컴퓨터가 연결된 네트워크 환경에서는 호스트 파일에 의한 주소와 이름의 변환 작업이 사실상 불가능하다고 볼 수 있다.

DNS(Domain Name System)는 주소와 이름 정보를 자동으로 유지하고 관리하여 이러한 문제를 해결하는 분산 데이터베이스 시스템이다. 호스트 주소와 이름 정보는 네임 서버

(Name Server)라는 특정 호스트가 관리하고, 주소 변환 작업이 필요한 클라이언트는 네임 서버에 요청해 IP 주소를 얻는다.

네트워크가 커지면 네임 서버에 보관되는 정보의 양도 많아진다. DNS는 하나의 집중화된 네임 서버가 전체 호스트의 정보를 관리하지 않고, 여러 네임 서버에서 분산하여 관리하도록 설계되었다. 계층 구조로 연결된 네임 서버는 자신이 관리하는 영역에 위치한 호스트 정보만 관리하며, 정보를 상호 교환하는 협력 관계를 통해 전체 호스트 정보를 일관성 있게 유지한다.

기타 주소

네트워크에서 사용하는 주소는 이를 사용하는 응용 프로그램에 따라 다양하다. OSI 7계층 모델의 각 계층에서도 목적에 따라 여러 형태로 존재한다. 인터넷에서 일반 사용자가 접할 수 있는 주소는 다음과 같다.

■ MAC 주소

MAC 주소는 계층 2의 MAC(Medium Access Protocol)에서 사용하며 LAN 카드에 내장되어 있다. 물리 계층을 통해 데이터를 전송할 때는 MAC 주소를 이용해 호스트를 구분한다. 따라서 네트워크 계층에서 데이터 링크 계층으로 데이터를 전송할 때는 먼저 IP 주소를 MAC 주소로 변환해야 한다.

■ IP 주소

IP 주소는 네트워크 계층을 수행하는 IP 프로토콜에서 사용한다. IP 패킷을 전달할 경로를 결정하는 라우팅의 기준이 된다.

■ 포트 주소

포트 주소(Port Address)는 주소는 전송 계층에서 사용하며, 호스트에서 실행되는 프로세스를 구분해준다. TCP와 UDP가 독립적으로 포트 주소를 관리하며, 포트 번호 또는 소켓 주소라는 용어를 사용하기도 한다.

■ 메일 주소

메일 주소는 메일 시스템에서 사용자를 구분하려고 사용한다. hong@korea.co.kr처럼 사용자 이름과 호스트 이름을 @ 문자로 구분해 표기한다.

- 1 네트워크는 시스템, 인터페이스, 전송 매체, 프로토콜, 네트워크, 인터넷 같은 용어로 표현하는 구성 요소를 갖는다.
- 2 네트워크는 시스템과 공유 전송 매체의 집합으로 구성되며, 표준화된 프로토콜을 사용해 서로 데이터를 교환한다.
- 3 국제 표준화 단체인 ISO에서는 네트워크를 위한 OSI 7계층 모델을 제시하였다. OSI 7계층은 물리 계층, 데이터 링크 계층, 네트워크 계층, 전송 계층, 세션 계층, 표현 계층, 응용 계층으로 구성되어 있다.
- 4 인터넷은 IP 프로토콜을 사용해 네트워크 계층의 기능을 수행하고, 전송 계층에서는 TCP와 UDP를 사용해 데이터를 전송한다.
- 5 네트워크와 네트워크를 연결하려면 인터넷워킹 기능을 수행하는 게이트웨이가 필요하다. 게이트웨이에는 리피터, 브리지, 라우터 등이 있다.
- 6 OSI 7계층 모델의 각 계층에서 사용하는 데이터에는 별도의 명칭이 존재한다. 특별히 네트워크 계층에서는 패킷, 데이터 링크 계층에서는 프레임이라는 용어를 사용한다.
- 7 호스트를 구분하는 구분자에는 유일성, 확장성, 편리성, 함축성이라는 특징이 있다.
- 8 인터넷에 연결된 네트워크 호스트에는 IP 주소와 호스트 이름이라는 두 개의 구분자가 있고, DNS를 이용해 이들을 맵핑한다. DNS는 입력된 호스트 이름을 IP 주소로 변환하는 분산 데이터베이스 시스템이다.
- 9 호스트 주소에는 호스트 이름, IP 주소, MAC 주소 등이 있다. 호스트에서 실행되는 프로그램은 포트 주소를 사용해 구분한다.



★ 연습문제

Chapter 01

1 다음 용어를 설명하시오.

- 시스템
- 인터페이스
- 전송 매체
- 프로토콜
- 네트워크
- 인터넷

2 다음 용어의 차이점을 설명하시오.

- 노드
- 호스트
- 클라이언트
- 서버

3 OSI 7계층 모델의 각 계층 기능을 설명하시오.

4 계층 구조 모델에서 프로토콜과 인터페이스의 차이를 설명하시오.

5 인터넷 환경에서 계층 구조 프로토콜을 구현하는 모델을 운영체제와 사용자 프로그램 환경으로 구분해 설명하시오.

6 네트워크 연동을 위한 인터넷워킹 기능을 수행하는 장비의 종류를 나열하고, 각각을 설명하시오.

7 호스트를 구분하는 구분자의 특징을 나열하시오.

8 32비트의 IP 주소를 4자리의 10진수로 축약하여 표현하는 방법을 설명하시오.

9 호스트 이름의 필요성과 구조를 설명하시오.

10 호스트 이름을 IP 주소로 변환하는 DNS의 필요성을 설명하시오.

11 네트워크에서 사용하는 MAC 주소, IP 주소, 호스트 주소, 포트 주소의 고유 특징을 설명하시오.

어려운 것과 익숙하지 않은 것

글. 제임스 _소프트웨어 이야기(jamestic.egloos.com)

현 / 장 / 의
#01 목 / 소 / 리

대학교에서 수업을 받을 때, 네트워크(Network), 프로토콜(Protocol) 같은 것은 참으로 어려운 주제 중 하나였죠. 책을 볼 때는 이해한 것 같은데, 조금 지나서 떠올려 보면 머리 속에 아무 것도 남지 않는, 그런 구름 같은 과목이었습니다. 이론과 실체가 따로 노는 과목이었다고나 할까요. 그래서인지 대학 생활 내내 이런 생각이 굳어지고 있었습니다.

‘프로토콜은 어려운 거야’

대학을 졸업하고, 직장에 첫발을 내디딜 때였습니다. 공교롭게도 저에게 주어진 첫 일은 다름아닌 ‘프로토콜’이었습니다. 이름만 들어보았던 OSI 프로토콜 포팅(Porting)과 CMIP 에이전트를 구축 하라는 것이었습니다. 게다가 주변에 관련 업무를 도와줄 분이 없었기 때문에, 스스로 홀로써야 하는 상황이었습니다. 처음에는 그 상황이 이해되지 않았는데 나중에 알고 보니, 제가 X 윈도우즈 쪽을 하고 싶다는 이야기를, X.25 프로토콜로 헛갈리신 실장님이 계셨더군요. 어쨌든 실장님의 오해 덕분에 프로토콜을 시작하게 되었습니다.

주어진 업무를 해결하기 위해 CCITT 규격을 열심히 읽고, OSI 프로토콜 매뉴얼을 정독하고, 소스 코드도 보면서 1년이라는 시간을 보냈습니다. 운이 좋았는지, 다행스럽게도 첫 번째 일을 잘 끝낼 수 있었습니다. 프로토콜에 대한 두려움 때문에 처음에는 긴장했는데, 조금 공부하다 보니 각 프로토콜이 세부적으로는 달라도 근본 원리는 다 비슷하다는 생각이 들었습니다. 그리고 대학 때, 가진 생각이 조금씩 바뀌기 시작했습니다.

‘생각보다 프로토콜은 어렵지 않은 것 같아’

그 이후로도 프로토콜을 포팅하거나 구현하였으며, 가끔 프로토콜 자체를 설계하기도 하였습니다. 실장님의 오해로 시작한 프로토콜이 이제는 제가 하고 있는 일의 상당 부분이 되어버렸죠. 지금은 ‘프로토콜’ 하면 이런 생각이 떠오릅니다.

‘프로토콜은 쉬운 거야’

물론 프로토콜이 간단하지는 않습니다. 프로토콜에 따라서 규격만 몇 백 페이지 되는 복잡한 것도 있죠. 하지만 이해할 수 없을 정도로 어려운 것은 아닙니다. 컴퓨터 사이언스 자체가 사람이 쉽게

이해할 수 있도록 만든 것이기 때문입니다. 기본 원리와 프로토콜 설계 의도를 이해하면, 상상했던 것보다 어렵지 않은 것 같아요.

그럼 대학 시절에는 프로토콜을 왜 어렵다고 생각했을까요? 아마 익숙하지 않았기 때문인 것 같습니다. 프로토콜을 공부하려면 컴퓨터가 여러 대 있어야 하고, 네트워크 인터페이스(Network Interface)도 많아야 하며, 엔디안 컴퓨터(Endian Computer)도 필요합니다. 저만의 생각일지 모르지만, 학생 때는 이런 교육 환경을 쉽게 접할 수 없었을 것입니다.

프로토콜은 익숙하지는 않지만, 결코 어려운 것은 아닌 것 같습니다. 누구나 도전하여 노력한다면 친숙해 질 수 있습니다. 제가 학생 때 가졌던 막연한 두려움 같은 것은 떨쳐 버리고, 즐겁게 공부 하길 바랍니다.

마지막으로, 제가 프로토콜을 공부하면서 도움이 되었던 두 가지 팁을 알려드리지요.

실제 체험하기

프로토콜은 개념적인 공부가 되기 쉽습니다. 이를 보완하기 위해서는 실체를 보는 것이 큰 도움이 됩니다. 방법은 아주 간단합니다. 주로 로우 프로토콜(Lower Protocol)에 해당하는 것으로, Ethereal이나 Tcpdump를 사용하여 실제 전송되는 PDU 혹은 패킷(Packet)의 내용을 직접 확인해 봅니다. 한 옥텟(Octet)씩 규격 내용과 어떻게 일치하는지 꼼꼼히 살펴보면, 머리 속에서 이해한 내용과 실제의 차이점을 쉽게 이해할 수 있을 겁니다.

Completeness

프로토콜은 모든 상황에서 동작하도록 설계되어 있습니다. 따라서 모든 상황에서의 동작을 이해해야 비로소 완벽하게 소화를 하는 거죠. 마치 퍼즐을 맞추는 느낌인데, 마지막 퍼즐까지 맞추려면 상당한 인내가 필요합니다. 그래서 많은 분이 도중에서 멈추고 말죠. 처음에는 별 차이가 없어 보이지만, 시간이 흐르면 상당히 큰 차이를 만들어 냅니다. 그래서 처음부터 연습해보라고 권해드립니다. 방법은 쉽습니다. 작은, 아주 작은 프로토콜을 골라서, 생각할 수 있는 모든 경우에 대해 동작을 이해하는 것입니다. 더 이상 질문이 없을 때, complete하다고 이야기할 수 있는 거고요. 비로소 프로토콜에 대해 준비가 되었다고 생각합니다.

이 글이 프로토콜을 시작하는데 도움이 되기를 바라며, 즐겁고 행복한 네트워크 전문가의 시작을 기대해 봅니다.

고맙습니다.