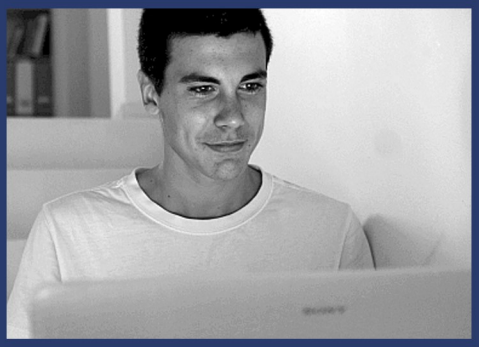
Hacker Highschool

SECURITY AWARENESS FOR TEENS



LESSON 3
BENEATH THE INTERNET

HACKING IS LEARNING
www.hackerhighschool.org

ISECOM

Warning - WORLD

Hacker Highschool Project(HHS)는 학습 도구로 만들어졌지만, 일부 내용의 경우 오용될 시 물리적 피해가 발생할 수 있습니다. 또한 특정 기술들로 인한 파급력과 영향력을 충분히 숙지하고 있지 못할 시 추가적인 문제가 발생할 수 있습니다. 학생들이 배운 내용들을 활용하는데 있어 주의가 필요하지만 적극적으로 배우고 연습할 수 있도록 지도해 주시기 바랍니다. 하지만 배운 내용들을 악의적으로 사용해 발생한 문제들에 대해 ISECOM은 책임을 지지 않습니다.

해당 교제에서 배울 내용들과 연습문제들은 공개되어 있으며 ISECOM에서 제시하는 다음의 조건들을 따를 시 누구나 사용할 수 있습니다:

HHS의 자료들은 초등학생, 중학생, 고등학생들에게 무료로 제공됩니다. 해당 자료들은 판매를 목적으로 생산될 수 없습니다. 대학, 직업학교, 단기과정, 방학특강 등에서 실시하는 어떤 학습과정에서도 허가 없이 학생들에게 해당 자료들로 인한 비용을 부과해서는 안 됩니다. 허가증을 받기 위해서는 HHS 홈페이지의 http://www.hackerhighschool.org/licensing.html를 방문하여 라이센스 섹션을 참조하시기 바랍니다.

HHS는 공개참여를 통한 오랜 노력과 여러 사람들의 지원으로 만들어졌습니다. 해당 프로젝트에 도움을 주고 싶으시다면 허가증 구매, 기부, 후원을 통해 HHS에 일조 하실 수 있습니다.

Warning – KOREA

HHS 는 ISECOM의 프로젝트입니다.
HHS 외 ISECOM의 모든 프로젝트의 한국 내에서의 관리 및 감독 등의 모든 권한은 ISECOM Korea 에 있습니다.
본 문서의 용도는 개인 학습용이며 무료 공개자료입니다.
해당 용도 외 사용 시 법적인 처벌을 받으실 수 있습니다.
해당 용도 외 문의는 ISECOM Korea 에 문의해 주시기 바랍니다.

목차

도움을 주신 분들

Marta Barceló, ISECOM
Pete Herzog, ISECOM
Chuck Truett, ISECOM
Bob Monroe, ISECOM
Kim Truett, ISECOM
Gary Axten, ISECOM
Marco Ivaldi, ISECOM
Simone Onofri, ISECOM
Greg Playle, ISECOM
Tom Thomas, ISECOM
Mario Platt
Ryan Oberto, Johannesburg South Africa
Vadim Chakryan, Ukraine
Peter Houppermans

ISECOM Korea

왕응석 EungSeok Wang
정진우 JinWoo Jung
박영후 YongHoo Park
최홍선 HongSeon Choe

Introduction and Objectives

인터넷이 발명되기 전 전자 통신의 개념은 마법과 같이 신비한 것으로 여겨졌다. 컴퓨터 제작사들은 전선을 통해 기기 간의 통신을 가능하게 할 독자적인 방법들을 각기 개별적으로 가지고 있었다. 하지만 어느 누구도 제조사가 다른 기기들 간의 통신이 가능할 것이라고는 생각하지 못했다.

과학자들과 학생들이 터미널을 이용해 메인프레임 컴퓨터(mainframe computer)에 접속하는 즐거움을 알게 됐을 때 세상은 크게 변하기 시작했다. 명성이 자자한 IBM 컴퓨터가 등장하고 컴퓨터를 가진 사람들은 자신들의 개인 컴퓨터를 사용해 메인프레임 컴퓨터에 접속하고 싶어 안달이 나있었다. 곧 모뎀과 전화선을 이용한 연결이 가능해졌고 터미널 에뮬레이터를 통한 작업 환경이 갖춰졌다. 네트워킹은 마법의 경지로 여겨졌고 초기 사용자들은 **구루(gurus)**로 추앙 받았다.

군사용으로 기획된 인터넷이 대중들에게 공개되었을 때 다시 한번 극적인 변화가 발생했다. 초기의 네트워킹은 로컬(local)영역에 한정되어 주로 사무실이나 캠퍼스 내에서만 가능했다. 어떻게 다른 시스템들 간의 연결이 가능했을까?

정답은 "wedge"라고 불리는 보편적인 주소체계 덕분에 기존의 네트워크에 접속할 수 있게 되었기 때문이다. 이 시스템은 **인터넷 프로토콜(Internet Protocol: IP)**라고 불린다. IP라는 것은 바다 건너에 있는 친구에게 소포를 보내는 것과 유사하다. 소포는 비행기, 기차, 차량을 통해 운반되지만 수신자는 항공 스케줄이나 가장 가까운 기차역에 대해 알 필요가 없다. 중요한 것은 소포는 어차피 수신자의 주소에 도착하게 된다는 것이다. **IP 주소(IP Address)**의 적용 방법은 이와 유사하다. 패킷은 전자신호, 광신호, 전파를 통해 이동하지만 이 역시 수신자가 알 필요가 없다. 중요한 것은 자신의 IP 주소와 연결해야 하는 시스템의 IP 주소 뿐이다.

이러한 개념을 복잡하게 하는 것은 실제 생활에서는 다수의 사람들이 하나의 주소에서 살 수 있다는 것이다. 이는 네트워킹의 세계에서도 일어날 수 있는 일이다. 하나의 서버가 FTP뿐 아니라 일반적인 HTTP와 보안이 강화된 HTTPS를 동시에 제공하는 경우가 있다. 축약어들의 끝부분에 있는 P라는 단어를 주목해 보자. 이는 연결 방법을 의미하는 **프로토콜(Protocol)**이 사용되고 있다는 것을 의미한다.

이번 레슨을 통해 프로토콜(protocol)과 포트(port)가 윈도우, 리눅스, OSX에서 어떻게 작동하는지 알게 될 것이다. 또한 시스템의 네트워크 능력을 알아내기 위해 몇 가지 도구들의 사용법을 배우게 될 것이다.

이번 레슨을 통해 다음과 관련된 기본 지식들을 습득하게 된다.

- 네트워크의 개념과 연결이 이루어지는 방법
- IP 주소(IP addresses)
- 포트와 프로토콜(ports and protocols)

네트워킹의 기본 개념

네트워킹 학습의 첫 단계는 **근거리 통신망(Local Area Network: LAN)**의 개념을 익히는 것이다. LAN은 물리적으로 연결되어 있는 컴퓨터들이 프린터나 드라이브 공간과 같은 자원들을 공유할 수 있게 해주고 **관리자(administrator)**계정을 통해 접근을 통제한다. 아래의 내용들은 일반적인 네트워크 장비와 네트워크 구성도(topology)에 대한 개념을 보여준다.

장비

해커로 계속 활동 한다면 다양한 네트워크 구성도를 마주하게 될 것이다. 다음의 부호들을 알고 있다면 큰 도움이 될 것이다.

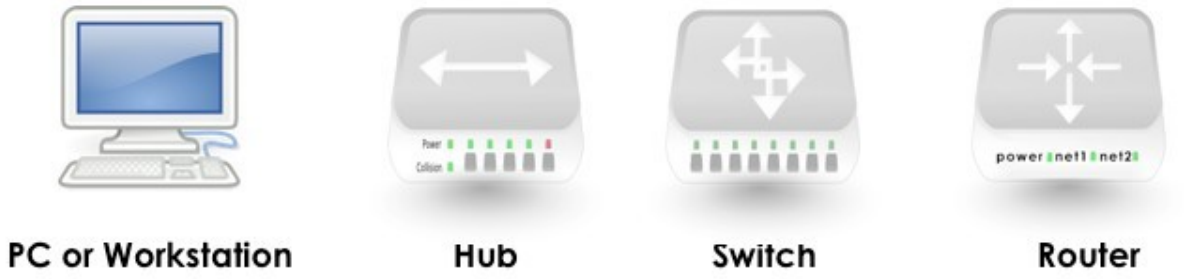


Figure 3.1 : 공통 네트워크 기호

허브(Hub)는 예전에 많이 사용하던 유선 전화망과 유사하다. 같은 연결망을 공유하고 있는 사람들은 다른 사람의 통신을 들을 수 있다. 이로 인해 랜(LAN)의 트래픽 소모량이 증가한다.

스위치(Switch)는 좀 더 발전된 장비로 트래픽을 구분하여 오직 송신자와 수신자만이 통신을 유지할 수 있도록 해준다. 하지만 허브와 마찬가지로 랜(LAN)에서만 사용 가능하다.

라우터(Router)는 여러 개의 랜(LAN) 사이에 존재한다. 라우터는 IP주소를 사용해서 다른 네트워크와 인터넷 간의 접속을 가능하게 만든다. 라우터는 자신에게 온 패킷의 도착지를 확인하고 해당 도착지가 있는 “다른” 네트워크로 보내주는 역할을 한다.

네트워크 구성도

네트워크 구성도(Topology)는 네트워크 내의 연결망이 어떻게 구성되어 있는지 보여준다. 네트워크 구성도를 어떻게 구성하느냐에 따라 장단점이 뚜렷하게 나타나며 구성 방식은 사용 기술, 기술적 한계, 성능, 보안 요구사항, 규모, 조직의 유형 등에 따라 달라진다.

랜(LAN)의 네트워크 구성방식은 다음과 같다.

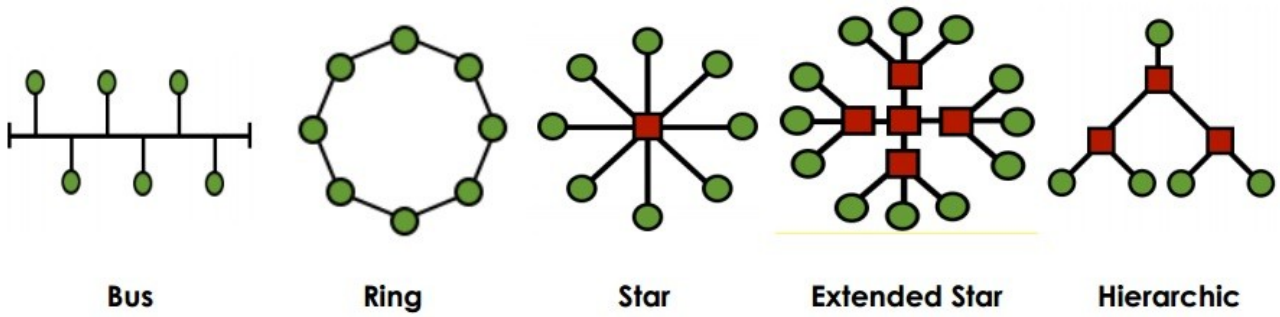


Figure 3.2: 네트워크 구성도

버스(Bus)형은 모든 컴퓨터들이 단일 케이블에 연결되어 있는 형태이다. 각각의 컴퓨터들은 전송 회로를 이용해 다른 컴퓨터와 직접 통신할 수 있다. 하지만 전송 회로의 일부분이 파괴된다면 모든 이용자의 네트워크가 단절된다.

링(Ring)형은 첫 번째 컴퓨터에서 부터 마지막 컴퓨터까지 원형으로 이어져 있다. 각각의 컴퓨터들은 인접한 두 개의 컴퓨터들과 직접 통신할 수 있다.

하지만 버스 네트워크 구성도는 현재 잘 사용되고 있지 않다. 링형 구성도의 경우에는 대규모 도시 간에 이중 역순환 링의 형태로 사용되기도 한다. 이중 역순환 링 형태는 서로 반대 방향으로 통신할 수 있고 신뢰성과 오류 복구능력이 향상된 네트워크 구성도이다.

스타(Star)형에서는 컴퓨터들 서로 직접 연결되어 있지 않고 허브나 스위치를 통해 컴퓨터들 간의 정보를 교환한다.

여러 개의 허브나 스위치들이 서로 연결되어 있는 경우에는 **확장형 스타(Extended Star)형**으로 불린다.

스타형과 확장형 스타형의 중심부은 **피어(peers)**라고 불리며 동등한 권한과 역할을 가지고 있으며 이런 네트워크 구성도가 현재 가장 많이 사용되고 있다.

스타형과 확장형 스타형의 네트워크들을 서로 연결시키기 위해서는 중앙에서 두 네트워크간의 트래픽을 통제하고 제한하는 중심 부분(central point)을 활용하여 **계층적(hierarchical) 형태**로 네트워크를 구성해야 한다. 이런 형태는 큰 규모의 기업에서 주로 사용한다.

Game On: Leaving the Back Door Open

살이 타버릴 듯 더운 여름날 에어컨이 빵빵하게 틀어져 있는 경찰서에 있을 수 있다는 사실에 감사하며 제이스는 자신에게 맡겨진 일을 하고 있는 중이다. 지금 제이스는 경찰서의 소규모 네트워크 구축을 돕고 있다. 덕분에 여름의 더위를 피하고 쿠키를 제공 받으며 백도어를 설치할 기회를 얻게 되었다. 몇 십년동안 제자리를 고수해온 작업용 책상 아래를 기어 다니던 제이스는 드디어 와이파이 엑세스 포인트(WiFi Access Point)를 숨길만한 지저분하고 어수선한 공간을 찾아냈다. 제이스는 그곳에 엑세스 포인트를 연결한 후 주변의 쓰레기들을 이용해 감추고 이더넷 케이블 더미를 끌어와 그 위에 올려놓았다.
누군가 갑자기 책상을 치는 소리에 놀란 제이스는 머리를 책상에 들이 받았다.

“아! 내 머리! 일하는 거 안보이세요? 그만 할까요?”
형사는 목소리를 가다듬은 다음에 말했다.

“그런건 아니고, 뭔가 복잡하게 하고 있는 것 같은데 어떻게 되어 가는지 알고 싶어서.”
제이스는 짜증난 10대 소녀를 연기하며 말했다.

“그 정도 수준의 헛소리를 잘도 하시는군요. 그것보다 제 쿠키는 언제 주실꺼죠, 킥암 형사님?”
“제이스, 부탁이니까 행크라고 불러줘. 네가 그렇게 부를 때 마다 내가 나이든 사람처럼 여겨진다고.”
행크 형사가 상처받았다는 듯이 말하고 있지만 쿠키를 주기 싫어서 자꾸 딴소리를 한다고 제이스는 생각했다.

“행크 형사님, 이런 말 해서 죄송하지만 형사님은 충분히 나이 드셨어요.”
“거참, 가슴이 아프구만. 나 아직 안 늙었어. 아직 한창이라고.”
행크 형사는 광이 나는 자신의 구두와 책상 아래로 사라진 제이스의 낡은 운동화를 바라보며 맞받아쳤다. 곧이어 거미줄과 먼지로 뒤덮힌 제이스의 얼굴이 책상 아래에서 튀어나왔다. 제이스는 아직도 케이블 더미를 손에 들고 있었다. 행크는 제이스를 일으켜주고 얼굴과 어깨에 묻어있는 거미줄들을 손으로 털어주었다.

“도와주세요, 경찰이 사람 팬다.”
제이스는 장난치듯 말했다.

“이런 범죄자 녀석, 네가 가진 끔찍한 계획을 다 불어라.”
근육질의 형사가 하는 말이었지만 제이스에게는 애원하는 목소리로 들렸다.
이건 좋은 징조로 여겨졌다.

"정말로 네트워킹에 대해서 알고 싶으세요?"
행크 형사는 열성적으로 고개를 끄덕였다. 정말 단순한 사람이라고 제이스는 생각했다.

"좋아요, 제가 한 일은 각각의 장비, 컴퓨터, 허브, 잭, 스위치, 라우터, 방화벽들이 어디로 가야 되는지를 알려주는 지도를 만든 거에요. 이 지도를 네트워크 구성도라고 부르죠. 이런 종류의 프로젝트를 시작하기 위해서는 지도가 필수죠. 이 네트워크 구성도 덕분에 모든 노드들이 문제없이 통신할 수 있죠. 구성도의 노드들은 버스(Bus)라고 불리는 전송 회로들을 이용해 통신하기 때문에 연결된 노드들 중 하나가 고장 난다면 다른 노드들도 영향을 받게 되요."
행크는 고개를 끄덕였고 제이스는 계속해서 말했다.

"네트워킹을 이 경찰서라고 생각해 보세요. 누군가 용의자를 데리고 온다면 한 명이 독점하는 것이 아니라 모든 경찰관들이 이 용의자를 두들겨 팰 수 있도록 공평하게 순번을 나눠야 하죠. 만약 얻어맞은 사람이, 그러니까 용의자가 다른 방으로 옮겨간다면 아직 순번을 기다리고 있는 경찰관이 가서 두들길 수 있도록 현재 용의자의 위치를 알아야 하죠."
"맙소사 제이스, 우리처럼 평화를 사랑하는 사람들을 계속 폭력범으로 몰아간다면 네가 우선 맞아야 될 것 같다."
행크 형사는 자신의 총이 메여있는 벨트를 슬쩍 들어 올리며 말했다. 제이스는 콧방귀를 뀌고 이야기를 계속했다.

"여기서 용의자는 데이터 패킷이고 경찰관들은 네트워크 장비라고 여기면 돼요. 스위치, 라우터, 방화벽, 다른 서버 등과 같은 모든 장비들은 자신들이 처리해야 하는 데이터 패킷이 어디에 있는지 반드시 알아야 하죠. 경찰들이 곤봉으로 용의자들을 처리하듯 말이에요."
행크 형사는 눈동자를 굴리며 자신의 곤봉을 찾았지만 들고 있지 않다는 것을 깨달았다. 킥킥대던 제이스는 케이블 다발을 방패처럼 들어 올리며 말했다.

"지금 들고 있는 있는 케이블들을 언제든지 사용할 수 있어요. 커피잔을 그대로 내려놓는다면 유혈사태는 일어나지 않을 거에요."
계속 웃고 있던 제이스는 균형을 잃었고 그대로 행크 형사의 품에 떨어졌다. '우와 무슨 몸이 돌덩이 같네'라고 생각하던 제이스는 행크 형사의 손이 어깨에 닿자 화들짝 놀라며 일어났다.
너무 빨리 일어나서 얼굴이 살짝 붉어진 채 제이스는 이어서 말했다.

"자, 네트워크 장비에는 두 종류가 있어요. 똑똑한 장비와 경찰처럼 둔한 장비가 있죠."
이때 4명의 경찰관들이 다가왔고 안타깝게도 "경찰처럼 둔한 장비가 있죠" 라는 말을 듣고 말았다. 표정이 변한 제이스는 소극적인 어투로 말을 이었다.

"똑똑한 장비는 자신들이 무엇을 했는지 전부 다 기억할 수 있어요. 자신들이 한 여러 가지 활동들의 로그를 간직하죠."

"그래서 둔한 장비들은 뭘 하지? 경찰처럼 둔한 장비들 말이야."
경찰 서장이 한 질문이었다.

Game Over

TCP/IP(DoD) 모델

TCP/IP는 1970년대에 미국 **국방부(DoD: Department of Defense)**와 미국 **방위고등연구계획국(DARPA: Defense Advanced Research Project Agency)**에 의해 개발되었다. TCP/IP는 누구든지 사용 가능할 수 있는 공개용으로 만들어 졌다. TCP/IP는 이용자들이 서로간의 컴퓨터를 연결하여 정보를 교환할 수 있게 해주었고, 결국 인터넷의 표준이 되었다.

TCP/IP 모델은 주로 **DoD 모델**로 불려진다. 우선 DoD 모델에 대해서 알아보자.

계층구조

DoD모델은 4개의 간단한 독립 계층으로 나눠지며 각 계층들을 통해 컴퓨터 간의 통신과정을 단계별로 처리한다. 정보를 처리하는 계층구조(layer)는 다음과 같다:

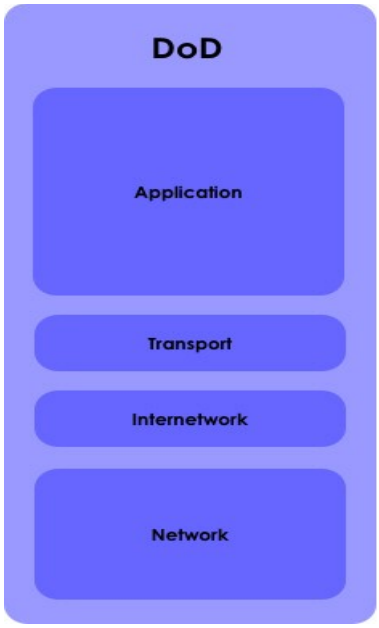


Figure 3.3: DoD 모델

응용프로그램 계층

응용프로그램(Application) 계층은 많은 사람들에게 익숙한 계층이다. 이 계층에서는 파이어폭스, 오페라, 이메일, 소셜네트워크 사이트, 인터넷 메신져, 채팅 프로그램과 같은 다양한 응용프로그램들이 실행된다. 실제로, 상당히 많은 응용프로그램들이 인터넷을 이용한다. 예를 들어 일부 오피스용 응용프로그램들은 인터넷을 통해 온라인 클립아트를 다운 받을 수 있다. 응용프로그램 계층은 다른 계층들이 전송해야 하는 페이로드(Payload)를 만들어낸다. 해당 계층을 배송 시스템에 비유하면 이해하기가 수월하다. 응용프로그램에서 만들어진 배송품은 처리방법이 적혀있는 포장지에 포장되어 우편실로 옮겨진다. 이 우편실은 전송계층에 해당한다.

전송계층

전송(Transport) 계층은 네트워크의 연결을 담당하며 **세션(Session)**을 제공한다. 전송계층의 가장 중요한 프로토콜은 **TCP(Transmission Control Protocol)**이다. TCP는 애플리케이션에서 온 수화물에 지시사항을 덧씌워 어떤 수화물인지, 어떻게 도착했는지, 손상 되었는지에 관한 정보를 표시한다.

어머니에게 이메일을 보내는 경우를 생각해 보자. 메일의 용량이 크든 작던 간에 인터넷을 통해 통째로 전송하는 것은 불가능하다. 그렇기 때문에 TCP는 메일을 **세그먼트(Segment)**조각으로 나눈 후 번호를 매기고 오류 점검 코드를 가장 마지막에 붙여서 목적지에 전송한다. 조각들이 전송 중에 손실되거나 오류가 난다면 목적지의 TCP는 재전송을 요청한다. 모든 조각들이 전송되면 목적지의 TCP는 조각들을 순서에 맡게 조립하여 완성된 메일을 어머니에게 전달한다.

전송계층에는 TCP말고도 **UDP**가 작동한다. UDP는 세션을 만들어 내지 않고 단순히 연속적인 **데이터그램(Datagram)**을 전송한다. UDP의 데이터그램은 세그먼트와 유사하지만 목적지에 도착했을 때 TCP처럼 오류를 검사하지는 않는다.

전송계층에서 TCP를 사용하던 UDP를 사용하던 모든 트래픽들은 특정 **포트번호**에 할당된다.

인터넷 계층

인터넷(Internetwork) 계층은 **패킷(Packet)**이 어디에서 시작되고 어디에서 끝나게 되는지에 관한 정보인 출발지와 목적지의 주소정보를 추가한다. 인터넷 계층이 하는 일은 올바른 주소로 소포를 배달하는 배송업체의 업무와 유사하다. 이 계층에서는 패킷이 제대로 전달 되었는지와 손상된 부분이 있는지는 신경쓰지 않는다. 그건 전송계층에서 해야 할 일이다. 인터넷 계층에서 가장 중요한 프로토콜은 **IP(Internet Protocol)**이다. 이 계층은 IP주소를 사용해서 패킷이 가장 적절한 루트를 통해 올바른 장소에 도착할 수 있도록 한다.

네트워크 계층

네트워크(Network Access) 계층은 인터넷을 사용하기 위해 이용자가 사용하는 물리적인 네트워크를 말한다. 만약 전화선을 사용한다면 단순한 형태인 PPP 연결을 사용하고 있는 것이다. 만약 DSL을 사용하고 있다면 ATM이나 **매트로 이더넷(Metro Ethernet)**을 사용하고 있는 것이다. 그리고 케이블 인터넷을 사용하고 있다면 DOCSIS 물리 네트워크를 사용하고 있는 것이다. TCP/IP를 통해 모든 네트워크는 연결되기 때문에 어떤 종류의 네트워크를 사용하고 있는지는 중요하지 않다. 네트워크 계층은 이더넷 케이블과 NIC(Network Interface Card)나 무선카드와 엑세스 포인트로 구성되어 있고 가장 낮은 단계의 네트워크 구성요소들과 비트 단위의 데이터들을 다룬다.

Feed Your Head: See "The OSI Model"

네트워킹 모델의 대안인 OSI 7계층을 이해하기 위해 이번 레슨의 마지막 장을 살펴보자.

프로토콜

이제 여러분들의 컴퓨터는 인터넷에 연결되었다. 그 과정은 아주 간단하지만 한번 생각해 봐야할 의문점이 있다. 인터넷에서 영화를 다운받는 중에도 여러분들은 인터넷 검색과 같은 다른 기능들을 얼마든지 사용할 수 있다. 이때 발생하는 트래픽들은 왜 섞이지 않는 것일까? 어떻게 네트워크는 트래픽들을 분리시키는 것일까?

그 답은 **프로토콜(Protocols)**에 있다. 프로토콜은 여러 종류의 트래픽들이 사용하는 고유한 언어라고 생각하면 된다. 웹 트래픽은 자신들만의 프로토콜을 사용하고, 파일을 전송하거나 이메일을 보낼 때 사용되는 프로토콜 역시 종류가 다르다. 디지털 세계의 규칙이 그렇듯이 포트들도 고유한 이름을 사용하는게 아니라 IP주소와 **포트번호**로 자신들을 구분 짓는다.

응용프로그램 계층 프로토콜

FTP(File Transfer Protocol)는 두 컴퓨터간의 파일 전송에 사용된다. FTP는 한 포트에서 데이터를 전송하고 다른 포트를 통해서는 제어 신호를 전송한다(전송결과 확인 등). 가장 많이 사용되는 포트번호는 20번과 21번이다(TCP).

HTTP(Hyper Text Transfer Protocol)는 웹페이지에서 사용되고 TCP의 80번 포트를 통해 연결된다. **HTTPS**는 보안이 추가된 것으로 TCP의 443번 포트를 통해 연결되며 네트워크 트래픽을 암호화하는 기능을 가지고 있다.

SMTP(Simple Mail Transfer Protocol)은 메일을 보내는데 사용되고 TCP의 25번 포트를 통해 연결된다.

DNS(Domain Name Service)는 ISECOM.org와 같은 도매인명을 216.92.116.13이라는 특정 IP 주소로 연결시켜주는 역할을 한다. DNS는 UDP의 53번 포트를 사용한다.

전송 계층 프로토콜

TCP와 **UDP**는 전송계층의 중요 프로토콜로 데이터를 전송하기 위해 사용된다.

TCP(Transfer Control Protocol)는 네트워크로 연결된 두 호스트들 간의 논리적 연결(**세션 session**)을 확립한다. 논리적 연결은 3단계 연결과정(Three-Way Handshake)을 통해 이루어진다.

1. A컴퓨터가 B컴퓨터와의 연결을 원할 경우 A는 B에게 SYN 패킷을 전송한다. **SYN**은 타임스탬프(Timestamp)와 트래픽을 교환하기 위해 클록(Clock) 동기화를 요청하는 패킷이다.

2. SYN 패킷을 받은 B는 연결 승인을 알리기 위해 **SYN/ACK** 패킷을 전송한다.

3. SYN/ACK 패킷을 받은 A는 최종 승인을 위해 **ACK** 패킷을 마지막으로 보내고 B가 이 패킷을 받으면 연결이 완료된다.

이러한 연결과정은 TCP에서만 필요하며 **UDP(User Datagram Protocol)**에서는 이러한 연결과정이 발생하지 않는다. UDP에서는 일방적인 데이터 전달이 이루어지며 상대방이 제대로 전달 받았는지는 중요하지 않다. 덕분에 UDP는 빠른 전송속도를 유지할 수 있다는 장점을 가지고 있으며 음성전달, 영화 감상, 게임플레이처럼 전송되던 프레임이 손실되더라도 크게 문제가 되지 않는 서비스에서 주로 사용된다.

인터넷 계층 프로토콜

IP(Internet Protocol)는 다른 종류의 네트워크를 사용하는 컴퓨터들을 연결 시켜주는 역할을 한다. 우편 수송차량과 같이, IP는 자신에게 온 패킷을 올바른 목적지에 전달한다.

ICMP

ICMP(Internet Control Message Protocol)는 네트워크 장비와 네트워크 관리자가 네트워크의 오류를 감지하고 처리하기 위해 사용하는 프로토콜이다. ICMP의 기능으로 사용되는 **ping**(Packet Internet Groper)이나 유사한 명령어들은 네트워크 상태를 점검하고 오류를 보고한다. 대량의 ping으로 인해 호스트 컴퓨터나 네트워크가 마비될 수 있기 때문에 대부분의 시스템들은 ICMP가 초당 한번만 반응 할 수 있도록 제한한다.

포트와 프로토콜을 요약하면 다음 그림과 같다.

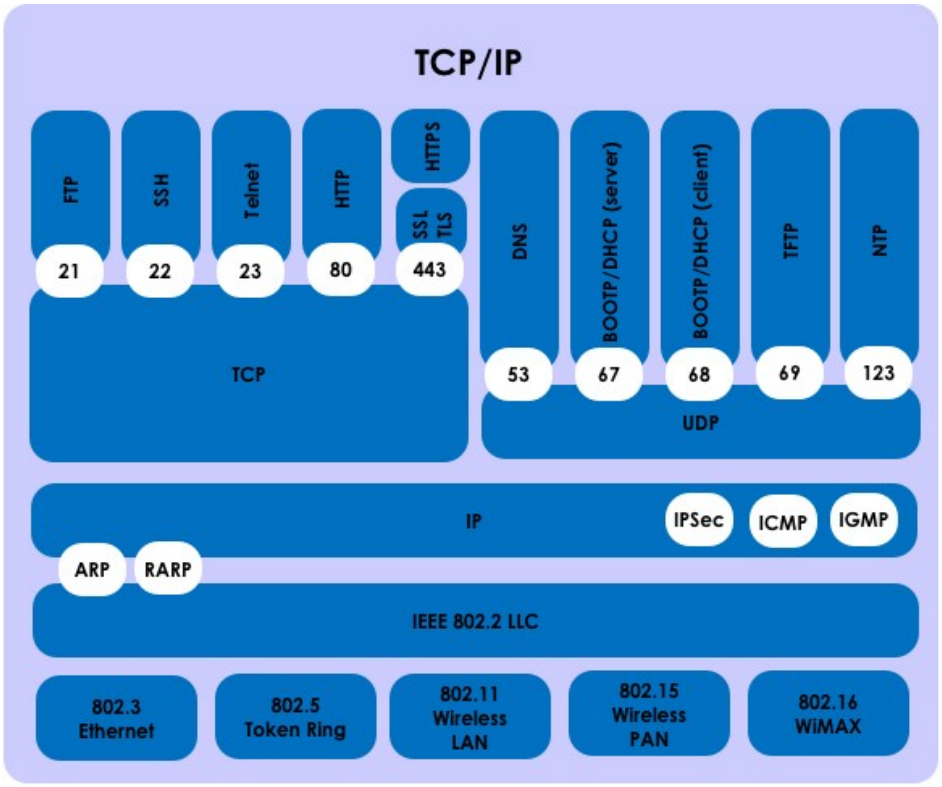


Figure 3.4 : TCP/IP 구성도

IPv4 주소체계

ISECOM.org와 같은 도메인명은 사람들이 기억하기에는 유용하지만 네트워크는 도메인명을 인식할 수 없다. 네트워크는 숫자로 된 IP 주소만을 인식할 수 있기 때문에 누군가 인터넷 주소창에 ISECOM.org라는 도메인명을 입력하면 컴퓨터는 **DNS(Domain Name Service)**에 빠르게 접속하여 도메인명에 해당하는 IP 주소를 찾아낸다.

IP 주소는 우편 주소체계와 유사하다. 우편물을 받기 위해서는 반드시 주소를 가지고 있어야 한다. **IPv4 주소체계**는 32비트로 구성되어 4구간으로 분리되어있다. 각 구간은 마침표로 구별되며 **8비트(Octet)**씩 분배되어 있다. 32비트로 구성된 IPv4 주소체계를 통해 분배될 수 있는 주소의 숫자는 2^{32}(4,294,967,296)에 달한다. IP 주소의 각 구간을 통해 네트워크와 네트워크상의 컴퓨터 정보를 알아낼 수 있다. 이는 우편 주소체계를 통해 시, 군, 구, 동 등의 정보를 알 수 있는 것과 유사하다.

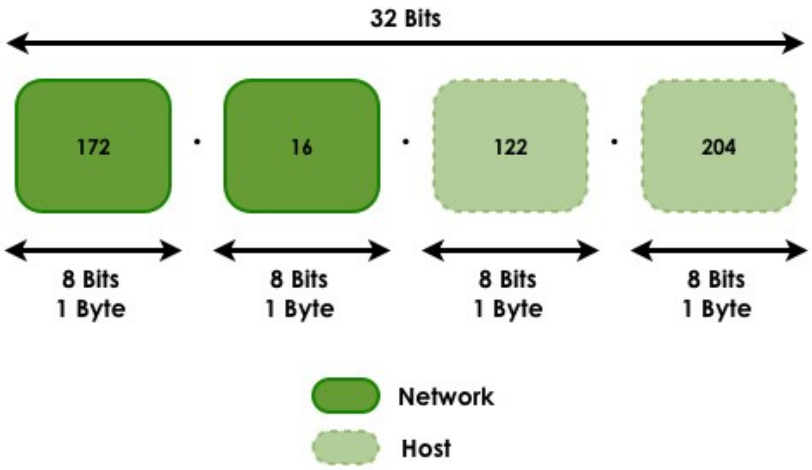


Figure 3.5 : 네트워크 숫자와 호스트 ID

다시 한번 배송 시스템에 비유하자면 IP는 우체국에 배송 물품을 배달하는 트럭의 역할을 한다. TCP는 지금 옮기고 있는 배송 물품이 총 몇 개의 패킷으로 되어 있고 몇 번째 패킷인지 설명하는 외부 포장지 역할을 한다. IP주소는 배송 물품이 최종적으로 전달되는 집(컴퓨터)이 어디에 있는지 알려준다.

IP 주소는 **공인 IP 주소(Public IP Address)**와 **사설 IP 주소(Private IP Address)**로 구분된다. 사설 IP 주소는 라우터를 통해 연결되지 않는 사설 네트워크(Private Network) 내에서 사용되며 라우터에 탐색되지 않는다.

사설 네트워크 내에서 연결되어 있는 컴퓨터들은 동일한 사설 IP 주소를 사용할 수 없다. 하지만 A와 B라는 사설 네트워크가 서로 연결되어있지 않다면 A에 연결되어 있는 컴퓨터와 B에 연결되어 있는 컴퓨터는 동일한 사설 IP주소를 사용할 수 있다(RFC1918을 확인해 보자).

```
10.0.0.0 ~ 10.255.255.255          (Class A)
172.16.0.0 ~ 172.31.255.255        (Class B)
192.168.0.0 ~ 192.168.255.255      (Class C)
```

클래스

IP 주소의 어떤 부분은 네트워크를 식별하는데 사용되고 어떤 부분은 개별 컴퓨터를 식별하는데 사용되는 것에 기초한 클래스들(Classes)로 나누어 진다.

각 부분에 할당된 크기에 따라, 더 많은 장치들이 네트워크 내에 허용되거나 많은 네트워크가 허용될 것이다. 기존의 클래스들은 다음과 같다:

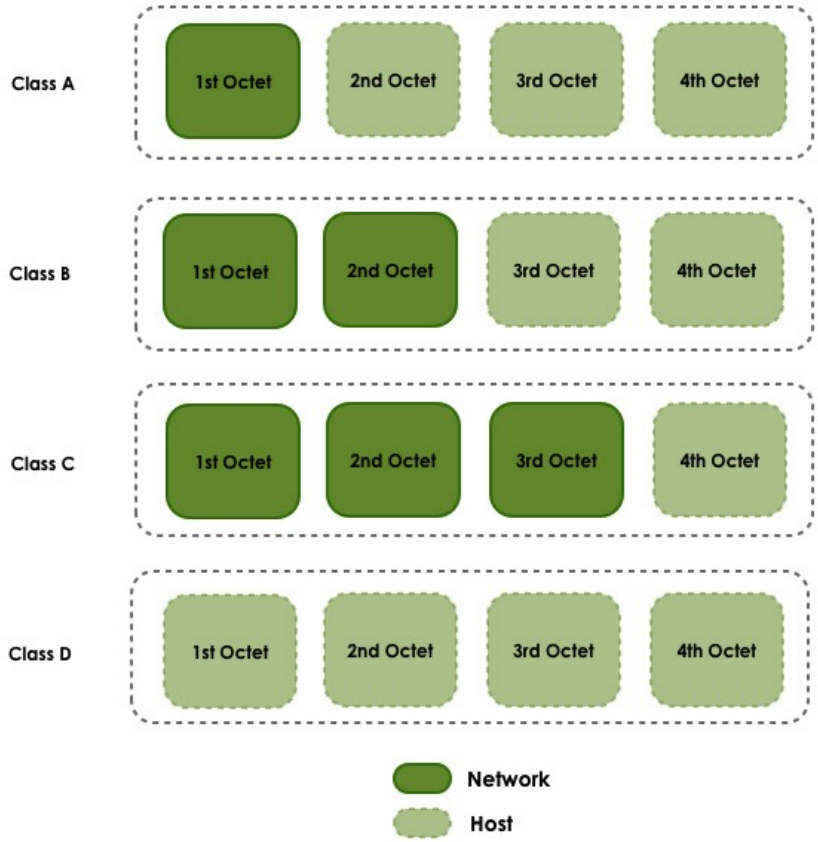


Figure 3.6 : IP 클래스 구분

Class A : 첫 번째 옥텟의 비트는 항상 '0'으로 시작된다. 할당된 구간은 0.0.0.0(관례상 사용되지는 않는다)부터 126.255.255.255까지이다. 127.xxx.xxx.xxx는 루프백(Loopback)이나 로컬 호스트(Local Host) 서비스를 위해 사용된다.

Class B : 첫 번째 옥텟의 비트는 '10'으로 시작하고 할당된 구간은 128.0.0.0부터 191.255.255.255까지이다.

Class C : 첫 번째 옥텟의 비트는 '110'으로 시작하고 할당된 구간은 192.0.0.0부터 223.255.255.255까지이다.

Class D : 첫 번째 옥텟의 비트는 '1110'으로 시작하고 할당된 구간은 224.0.0.0부터 239.255.255.255까지이다.

클래스에 할당된 주소 외에는 연구개발이나 추가 할당을 위에 비축되어 있다.

넷마스크(Netmask)는 아이피 주소 클래스의 구분을 명확하게 해준다. IP 주소를 2진법으로 변환 시켰을 때 '1'은 네트워크 부분을 나타내고 '0'은 호스트를 의미한다. 표준 넷마스크 클래스는 다음과 같다.

```
255.0.0.0         (Class A)
255.255.0.0       (Class B)
255.255.255.0     (Class C)
```

옥텟 하나를 사용하면 클래스 A이고, 2개의 옥텟은 클래스 B, 3개의 옥텟은 클래스 C에 해당한다. 표준 넷마스크 클래스를 사용하는게 편하지만 의무사항은 아니다.

이 모든 것이 의미하는 것은 호스트를 식별하기 위해서는 IP 주소와 넷마스크가 필요하다는 것이다.

IP : 172.16.1.20
Mask : 255.255.255.0

루프백 주소

127.0.0.1부터 127.255.255.254까지의 IP 주소는 **루프백(Lopback)**이나 로컬호스트 주소로 사용하기 위해 남겨져 있고 이 주소들은 자신의 컴퓨터로 연결된다. 모든 컴퓨터는 127.0.0.1을 루프백 주소로 가지고 있다.

루프백 주소는 다른 장비를 식별하기 위해서는 사용될 수 없다. 이처럼 특정 주소는 사용할 수 없도록 규정되어 있고 **네트워크 주소(network address)**와 **브로드캐스트 주소(broadcast address)**도 여기에 해당된다.

네트워크 주소

네트워크 주소는 IP 주소의 네트워크 부분을 나타내며 **호스트 비트를 '0'으로 표시해서 구분한다**. 네트워크 주소는 해당 주소의 네트워크 전체를 의미하기 때문에 호스트에게 주어질 수 없다.

IP : 172.16.1.**0**
Mask : 255.255.255.0

브로드캐스트 주소

브로드캐스트(Broadcast) 주소는 기본적으로 네트워크 주소에 해당되며 **호스트 비트를 모두 '1'로 표시(10진법으로 255)한다**. 브로드캐스트 주소를 통해 해당 네트워크의 모든 호스트에게 메시지를 전달할 수 있다.

IP : 172.16.1.**255**
Mask : 255.255.255.0

포트

TCP와 UDP는 애플리케이션 간의 정보를 교환하기 위해 **포트(Port)**를 사용한다. 포트는 우편주소에서 가장 마지막에 적는 세부주소와 같은 역할을 한다. 보낸 편지가 특정 아파트에 정확하게 도달해도 몇 동 몇 호인지 알 수 없다면 수령인에게 전달될 수 없다.

아파트의 동과 호수의 기능을 하는 것이 바로 포트이다. 패킷이 올바른 IP 주소에 도착하더라도 적절한 포트를 알지 못한다면 어떤 애플리케이션에 패킷을 전달해야 할지 알 수 없다. 포트 번호는 16비트로 되어있고 10진법으로 바꾸면 0 ~ 65535(2의16승의 개수)에 해당한다.

또 다른 비유로 포트를 설명할 수 있다. 컴퓨터가 우체국이라면 애플리케이션은 우체국의 우편함에 해당된다. 우편함은 번호를 통해 구분되고 중복되는 번호는 없다. 즉 포트는 우편함의 번호에 해당한다.

포트 덕분에 한 IP 주소에서 여러 종류의 네트워크 정보를 받아들일 수 있고 올바른 애플리케이션에 전달할 수 있다. 네트워크 서비스는 포트 번호를 통해 자신에게 접속한 이용자가 어떤 종류의 정보를 원하고 어떤 프로토콜을 통해 정보를 전송해야 하는지 알 수 있고 여러 명의 사용자들과 동시에 통신할 수 있다.

예를 들어 한 컴퓨터의 IP 주소가 62.80.122.203이고 80번 포트의 웹 서버에서 관리하는 www.osstmm.org에 접속하고 싶다면 62.80.122.203:80이라는 **소켓 주소(Socket Address)**가 필요하다.

62.80.122.203:80

주로 사용되는 포트를 표준화하기 위해 IANA는 0부터 1024까지의 포트번호를 **많이 사용되는 서비스들(Privileged** or **Well-known Services)**에 지정해 놓았다. 1025부터 65535까지는 동적포트(Dynamic or Particular Ports)에 할당되어 있다.

인터넷 할당 번호 관리기관(IANA:Internet Assigned Numbers Authority)에 의하면 가장 많이 사용되는 포트번호는 다음과 같다.

포트가 하는 일		
번호	**키워드**	**설명**
5	rje	원격 작업 입력(Remote Job Entry:RJE)
0		예약됨
1-4		할당되지 않음
7	echo	Echo 프로토콜
9	discard	접속 테스팅을 위한 Null 서비스
11	systat	연결된 포트를 열거하는 시스템 상태 서비스
13	daytime	요청하는 호스트에게 날짜와 시간 정보를 보낸다
15	netstat	네트워크 상태(netstat)
17	qotd	오늘의 한마디를 연결된 호스트에게 보낸다
19	chargen	자막 생성 서비스 (Character Generator Service) 끊임없이 문자 스트림을 보낸다
20	ftp-data	파일 전송 프로토콜 (데이터 포트)
21	ftp	파일 전송 프로토콜 (제어 포트)
22	ssh	SSH 원격 접속 프로토콜
23	telnet	텔넷 서비스
25	smtp	단순 메일 전송 프로토콜 (Simple Mail Transfer Protocol)
37	time	시간 프로토콜
39	rlp	자원 위치 프로토콜 (Resource Location Protocol)
42	nameserver	인터넷 이름 서비스 (Host Name Server)
43	nicname	WHOIS 디렉토리 서비스
53	domain	DNS(Domain Name Server)
67	bootps	부트스트랩 프로토콜 서버; 동적 호스트 설정 프로토콜 서비스에 의해서도 사용됨
68	bootpc	부트스트랩 프로토콜 클라이언트; 동적 호스트 제어 프로토콜(DHCP) 서비스에 의해서도 사용됨

포트가 하는 일		
번호	키워드	설명
69	tftp	간단한 파일 전송 프로토콜
70	gother	Gopher 인터넷 문서 검색
75		개인 다이얼 아웃 서비스
77		개인 RJE 서비스
79	finger	사용자 연락 정보를 알아보는 핑거 (Finger) 서비스
80	www-http	WWW(World Wide Web) HTTP(HyperText Transfer Protocol) 프로토콜
95	supdup	Telnet 프로토콜 확장
101	hostname	SRI-NIC 기계의 호스트명 서비스
102	iso-tsap	ISO 개발 환경 (ISODE) 네트워크 응용 프로그램
110	pop3	POP3 - 전자우편
113	auth	인증과 식별 프로토콜
117	uucp-path	UUCP (Unix-to-Unix Copy Protocol) 경로 서비스
119	nntp	네트워크 뉴스 전송 프로토콜
123	ntp	네트워크 시간 프로토콜
137	netbios-ns	NETBIOS 이름 서비스
138	netbios-dgm	NETBIOS 데이터그램 서비스
139	netbios-ssn	NETBIOS 세션 서비스
140-159		할당되지 않음
160-223		예약됨

캡슐화

만약 이메일을 통해 메시지를 보낸다면 해당 메시지는 여러 단계의 전송 계층을 통과하게 된다. 우선 응용프로그램 계층에서 만들어진 데이터는 전송 계층으로 보내진다.

전송 계층은 해당 데이터를 전달받고 세그먼트 조각으로 쪼갠 후 포트 번호, 세그먼트 번호, 세션 정보가 들어있는 헤더를 추가한다.

세그먼트를 전달 받은 인터넷 계층은 출발지와 목적지의 IP 주소와 메타정보를 간직한 또 다른 헤더를 추가한다.

다음 계층에서 다시 한번 헤더가 추가되며 해당 계층의 네트워크는 주로 이더넷을 통해 연결된다. 이러한 과정들을 통틀어 **캡슐화(Encapsulation)**라고 한다.

첫 번째 계층에서 만들어진 데이터는 계층을 하나씩 통과할 때마다 중복해서 캡슐화가 진행되며 상대방의 마지막 계층에 도착해서야 실제적인 데이터 전송이 이뤄진다. 이러한 과정은 다음과 같다.

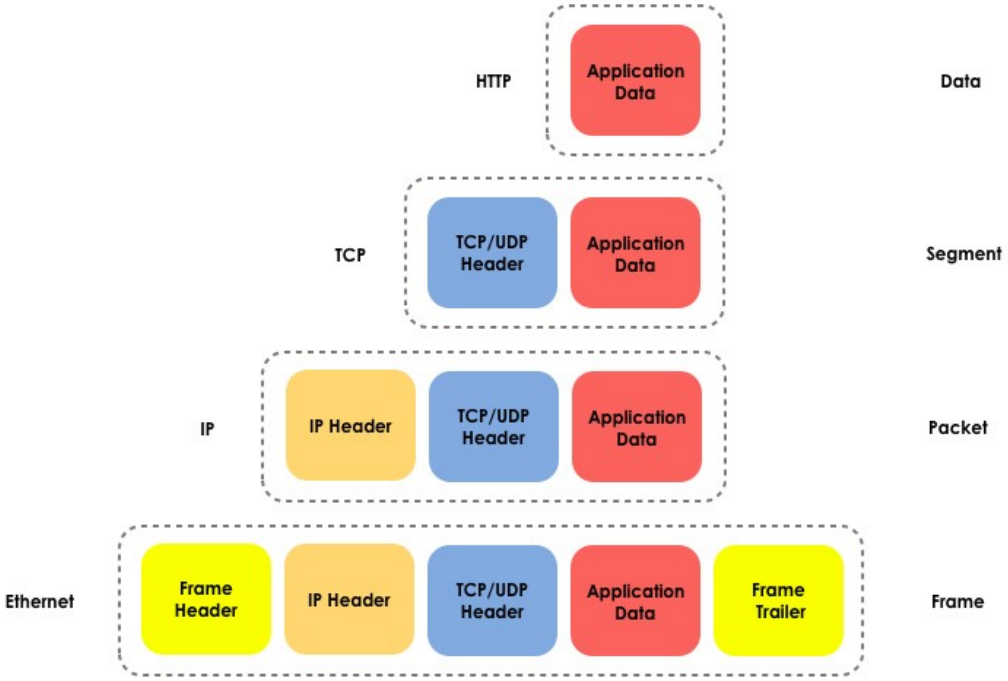


Figure 3.7 : 캡슐화

캡슐화된 정보가 목적지에 도착하면 캡슐화가 해제돼야 한다(Deencapsulation). 순차적으로 캡슐화되며 각 계층을 거슬러 내려온 정보는 다른 네트워크의 계층을 한 단계씩 올라가며 캡슐화를 하나씩 해제해 나간다.

이러한 주소지정체계(Addressing scheme)의 마지막 정보에는 컴퓨터의 NIC(Network Interface Controller)가 가진 고유주소가 들어있다. 이러한 고유주소는 **MAC(Media Access Controller)**주소라고 불린다. MAC 주소는 두 자리의 **16진수(hexadecimal)** 6쌍으로 구성되어 있고 각 쌍은 마침표나 하이픈으로 분리된다. MAC 주소는 네트워크 카드의 물리주소에 해당하며 마음대로 바꿀 수 없다(바꿀 수 있는 방법이 있지만 그걸 알아내는 것은 여러분의 몫이다). MAC 주소는 다음과 같이 표현된다.

00-15-00-06-E6-BF

Exercises

3.1 챕터 1과 2에서 배운 명령어를 사용하여 자신의 IP 주소, 넷마스크, DNS 호스트명, MAC 주소를 확인하고 결과를 파트너와 비교해보자. 비슷한 점과 차이점이 보이는가? 사용하고 있는 IP 주소체계가 공용 네트워크와 사설 네트워크 중 무엇을 의미하는가?

3.2 Netstat

netstat 명령어는 자신의 네트워크 상태를 보여준다. 네트워크의 접속 위치, 접속 시간 등을 알 수 있다. CLI를 열고 다음 명령어를 입력해 보자.

```
netstat
```

CLI 화면에서 접속된 네트워크의 리스트를 확인할 수 있다. 외부주소를 숫자 형태로 바꾸기 위해 다음 명령어를 입력해 보자.

```
netstat –n
```

다음 명령어는 연결되어 있는 네트워크와 활성화된 네트워크를 모두 보여준다.

```
netstat –an
```

다음 명령어는 netstat의 다른 옵션들을 보여준다.

```
netstat –h
```

netstat명령어의 결과물에서 로컬 주소와 외부 주소를 확인하고 현재 사용하고 있는 포트가 무엇인지 알아보자.

```
Proto Recv-Q Send-Q    Local Address      Foreign Address    (state)
tcp4       0      0    192.168.2.136.1043  66.220.149.94.443 ESTABLISHED
```

포트번호는 IP 주소 다음에 나와 있고 마침표로 구분된다. 로컬주소와 외부주소에서 사용하는 포트번호가 왜 다른가?

인터넷 브라우저나 브라우저의 탭을 여러 개 킨 후에 netstat 명령어를 다시 사용해보자.

여러 개의 탭을 사용하고 있다면 브라우저는 각 탭에 해당하는 정보를 어떻게 구별하는가?
인터넷 브라우저를 사용하고 있을 때는 왜 접속 대기 중인(Listening) 포트가 나타나지 않는가?

어떤 프로토콜이 사용되고 있는가?

하나 이상의 프로토콜이 사용되고 있을 때는 어떤 일이 발생하는가?

3.3 첫 번째 서버

이번 실습을 위해서는 **netcat(nc)** 프로그램을 사용해야 하니 자신의 OS에 적합한 netcat 프로그램을 다운받자.

1. CLI에서 다음 명령어를 입력한다.

```
nc –h
```

netcat에서 사용할 수 있는 옵션들을 보여준다.
간단한 서버를 만들기 위해 다음 명령어를 입력한다.

```
nc –l –p 1234 (리눅스와 윈도우용)
nc –l 1234 (OSX용)
```

1234포트에 접속 대기 중인 서버를 만들었다.

2. 두 번째 CLI를 키고 다음 명령어를 입력한다.

```
netstat –a
```

1234포트에서 접속 대기 중인 네트워크 서비스를 확인할 수 있다.

서버와 통신하기 위해서는 클라이언트가 필요하다. 두 번째 CLI에 다음 명령어를 입력한다.

```
nc localhost 1234
```

그러면 1234포트에서 접속 대기 중이던 서버에 접속한다. 이제 두 개의 CLI 중 어떤 곳에 글을 쓰던 다른 CLI에서도 볼 수 있다.

이 결과가 암시하는 것이 무엇인지 생각해 보자. 만약 이 기능을 악용해 다른 사람의 컴퓨터에 피해를 입히려면 어떤 방법이 있는가?

netcat은 어떤 트래픽이던 부담 없이 보낼 수 있는데 보안대책은 무엇일까?

3. 첫 번째 CLI에서 컨트롤 + C를 눌러 서버를 종료해 보자.

4. 이제 ‘test'라는 제목을 가진 텍스트 파일을 만들고 “Welcome to my server!"라는 내용을 입력해 보자.
텍스트 파일을 만들었다면 다음 명령어를 첫 번째 CLI에 입력하고 결과를 알아보자.

```
nc -l -p 1234 < test
```

다른 CLI에 다음 명령어를 사용하여 서버에 접속해 보자.

```
nc localhost 1234
```

클라이언트가 서버에 접속하면 텍스트 파일의 결과물을 보게 될 것이다.

해당 서버에 접속하기 위해서 어떤 프로토콜이 사용되는가?

netcat을 이용해 프로토콜을 바꿀 수 있는가? 가능하다면 방법은 무엇인가?

Feed Your Head: The OSI Model

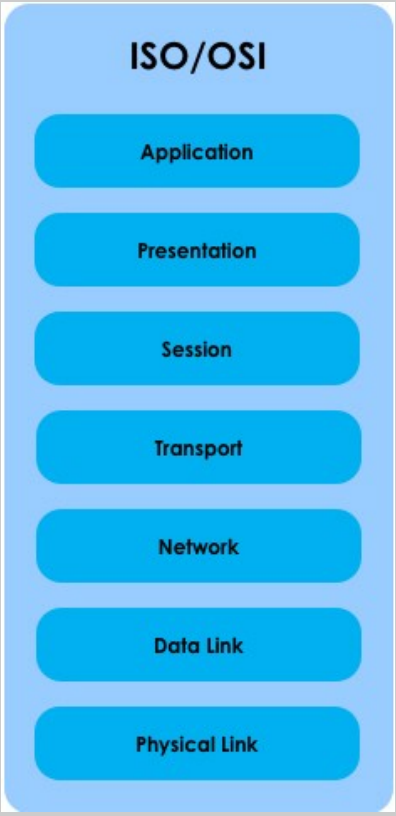


Figure 3.8 : ISO/OSI 모델

OSI 모델은 TCP/IP 모델이 나오고 10년 정도가 지난 1980년대에 **ISO(International Standards Organization)**에 의해서 개발되었다. **OSI(Open Systems Interconnection)**는 다양한 조직에서 만들어지는 네트워크 구성방식들을 표준화하기 위한 시도들 중 하나였다.

OSI의 계층화된 모델은 간결하게 구성되어 있다. 각각의 계층에는 유사한 기능들이 모여 있고 하위 계층이 상위 계층에 서비스를 제공하는 식으로 구성되어 있다(이 부분이 핵심이다).

계층화된 모델의 장점으로는 각 계층들이 자신들만의 고유한 통신방법을 고수할 수 있고 한 계층에 새로운 기능이 추가되어도 다른 계층에 피해를 주지 않는다는 것이다. 이러한 장점 덕분에 2000년대에 인터넷 붐이 일었을 때 새로운 응용프로그램과 서비스들이 언제든지 추가될 수 있었다.

OSI 모델이 가지는 두 가지 규칙(유사한 기능들의 그룹화와 하위 계층의 서비스 제공) 외에도 한 가지 엄격한 규칙이 존재한다. 두 컴퓨터가 통신하는 경우에는 각 컴퓨터에서 동일한 계층을 사용하여 통신해야 한다. 예를 들어 www.google.com에 접속한다면 본인 컴퓨터의 7 계층과 구글 웹 서버의 7계층 사이에 직접적인 연결이 생성되어야 한다. 이처럼 다른 계층이 사용될 때도 상대편과 동일한 계층을 사용해서 통신해야 한다.

이제 OSI 모델의 7 계층이 가지는 기능들을 알아보자.

응용프로그램 계층 Application Layer	응용프로그램과 응용프로그램 사용을 위한 유저 인터페이스 간의 직접적인 연결을 보장한다. 인터넷 익스플로워나 파이어폭스와 같은 인터넷 브라우저가 여기에 해당한다.
프레젠테이션 계층 Presentation Layer	호스트 간에 교환되는 데이터가 상실되지 않고 안전하게 전달되는 기능을 한다. 암호화를 사용하는 서비스에서 데이터를 암호화할 때 프레젠테이션 계층을 사용한다.
세션 계층 Session Layer	호스트 간의 대화개념의 연결(Dialogue)을 가능하게 한다. 기본적으로 두 호스트간의 연결개시, 연결관리, 연결종료 과정을 시행한다.
전송 계층 Transport Layer	호스트 간의 안전한 데이터 전송을 보장하고 상위 계층에 신뢰성 있는 데이터 전송 서비스를 제공한다. 즉 전송 계층에서는 작은 부분으로 쪼개져서 전달되는 데이터들을 다시 조립한다. 만약 패킷이 분실되거나 도착하지 않는다면 전송 계층에서 재전송을 요구하고 올바른 순서대로 다시 조립해야 한다.
네트워크 계층 Network Layer	호스트 간의 연결을 위해 주소를 지정하는 역할을 한다. 네트워크에서 자신이 필요한 고유 IP 주소를 찾고 어떤 길을 통해서 가야하는지 결정한 후 데이터를 목적지에 전달한다. 전달되는 데이터는 정해진 길에 있는 라우터들을 거쳐서 목적지에 도착한다.
데이터링크 계층 Data Link layer	물리 계층에서 발생하는 오류를 해결하고 다른 접속 매체를 관리한다. 기본적으로 데이터링크 계층은 물리적 수단을 통해 데이터를 전송할 수 있도록 준비하는 역할을 한다. 여기서 물리적 수단은 무선전파, 광섬유 케이블, 구리선 등을 의미한다.

물리 계층 Physical layer	데이터 전송을 위한 장비들의 물리적 특성과 선택된 장비를 통해서 정보를 전달하기 위해서는 무엇이 필요한지를 다룬다. 와이파이 연결의 경우에는 무선 신호가, 광섬유 연결은 광신호가, 구리선 연결은 전기신호가 해당된다.

위의 7계층들은 호스트 간의 신뢰성 있는 연결을 보장하기 위해 필요한 기능들로 구성되어 있다.

다음 그림을 통해 각 모델들을 비교해 볼 수 있다.

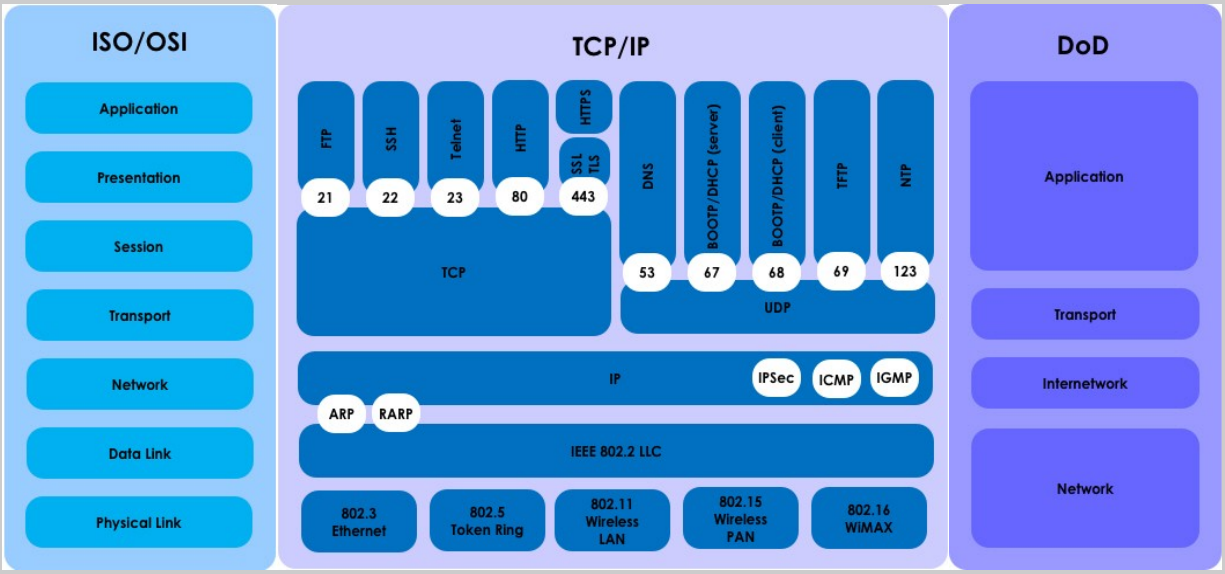


Figure 3.9 : 네트워킹 모델간 비교

오늘날의 십대들은 SNS, 인터넷 등을 통해 전 세계와 연결되어있다.
그러나 그들은 사기, 신원 위조, 개인정보 유출 등의
인터넷을 통한 공격방식과 이에 대한 대처방법을 잘 알지 못한다.
해커하이스쿨은 이러한 학생들을 위한 지침서가 될 것이다.

해커하이스쿨 프로젝트는 중·고등학생들에게 보안과 개인정보에 대한 인식을 제고하고 향상시킬 수 있는 교육자료이다

해커하이스쿨은 올바른 해커양성을 목적으로하며 이론과 실전파트로 구성되어있다. 해커는 지적능력, 창의성, 논리성을 모두 갖춰야한다. 우리는 학생들에게 일반적인 사이버보안 인식 혹은 IT 기술의 교육뿐만 아니라 어떻게 해커의 자질을 기르고 향상시킬 수 있는지를 가르칠 필요가 있다. 이 프로그램은 보안과 개인정보 인식교육 자료를 무상으로 포함하고 있으며 승인받은 중교등학교 선생님들을 위한 백앤드 서비스도 지원중이다. 다양한 언어로 지원되고 있으며 이 수업은 안전한 인터넷 사용, 웹 개인정보, 인터넷검색, 바이러스나 악성코드를 피하는법 윤리와 법 등을 포함한다.

HHS 프로그램은 ISECOM, 비영리단체, 보안 인식과 전문적인 보안 개발과 승인에 초점을 맞춘 오픈소스리서치 그룹에 의해 개발 되었다.

Hacker Highschool
SECURITY AWARENESS FOR TEENS

ISECOM