제6장
인터넷 프로토콜 소개

이 장에서는 다음과 같은 CompTIA Network+ 시험 목표를 다룹니다.

- **1.1 개방형 시스템 상호 연결(OSI) 모델의 계층과 캡슐화 개념을 비교하고 대조하십시오.**
  - OSI 모델
    - 1단계 – 물리적
    - 레이어 2 – 데이터 링크
    - 레이어 3 – 네트워크
    - 4계층 – 전송
    - 레이어 5 – 세션
    - 6단계 – 프레젠테이션
    - 7단계 – 응용 프로그램
  - OSI 모델 맥락에서의 데이터 캡슐화 및 역캡슐화
    - 이더넷 헤더
    - 인터넷 프로토콜(IP) 헤더
    - 전송 제어 프로토콜(TCP)/사용자 데이터그램 프로토콜(UDP) 헤더
    - TCP 플래그
    - 유효 탑재량
    - 최대 전송 단위(MTU)
- **1.5 일반적인 포트와 프로토콜, 그 응용 분야 및 암호화된 대안에 대해 설명하십시오.**
  - 프로토콜 포트
    - 파일 전송 프로토콜(FTP) 20/21
    - 보안 셸(SSH) 22
    - 보안 파일 전송 프로토콜(SFTP) 22
    - 텔넷 23
    - 단순 메일 전송 프로토콜(SMTP) 25
    - 도메인 이름 시스템(DNS) 53
    - 동적 호스트 구성 프로토콜(DHCP) 67/68

- 단순 파일 전송 프로토콜(TFTP) 69
- 하이퍼텍스트 전송 프로토콜(HTTP) 80
- 우체국 프로토콜 v3(POP3) 110
- 네트워크 시간 프로토콜(NTP) 123
- 인터넷 메시지 액세스 프로토콜(IMAP) 143
- 단순 네트워크 관리 프로토콜(SNMP) 161/162
- 경량 디렉터리 액세스 프로토콜(LDAP) 389
- HTTPS(Hypertext Transfer Protocol Secure) [SSL(Secure Sockets Layer)] 443
- HTTPS [전송 계층 보안(TLS)] 443
- 서버 메시지 블록(SMB) 445
- 시스템 로그 514
- SMTP TLS 587
- 경량 디렉터리 액세스 프로토콜(SSL 기반)(LDAPS) 636
- SSL을 통한 IMAP 993
- SSL을 통한 POP3 995
- 구조화 질의 언어(SQL) 서버 1433
- SQLnet 1521
- MySQL 3306
- 원격 데스크톱 프로토콜(RDP) 3389
- 세션 시작 프로토콜(SIP) 5060/5061
- IP 프로토콜 유형
  - 인터넷 제어 메시지 프로토콜(ICMP)
  - TCP
  - UDP
  - 일반 라우팅 캡슐화(GRE)
  - 인터넷 프로토콜 보안(IPSec)
  - 인증 헤더(AH)/캡슐화 보안 페이로드(ESP)

TCP/IP(전송 제어 프로토콜/인터넷 프로토콜) 제품군 은 국방부(DoD)에서 데이터 무결성을 보장 및 보존하고 대규모 전쟁 발생 시 통신을 유지하기 위해 개발되었습니다.

따라서 TCP/IP 네트워크는 올바르게 설계하고 구현한다면 진정으로 견고하고 신뢰할 수 있으며 복원력이 뛰어난 네트워크 솔루션이 될 수 있습니다. 이 장에서는 TCP/IP 프로토콜에 대해 다루겠습니다.

먼저 미 국방부의 TCP/IP 버전에 대해 살펴보고, 2장 "개방형 시스템 상호 연결 사양"에서 논의한 OSI 참조 모델과 이 버전 및 해당 프로토콜을 비교하겠습니다.

국방부 모델의 각 계층에서 발견되는 다양한 프로토콜을 살펴본 후, 2장 에서 시작한 데이터 캡슐화 설명에 더 자세한 내용을 추가하여 이 장을 마무리하겠습니다 .

Todd Lammle의 CompTIA 동영상 및 연습 문제를 보려면 다음 링크를 참조하십시오 www.lammle.com .

TCP/IP를 소개합니다

TCP/IP는 인터넷과 인트라넷 운영에 있어 매우 중요한 기술이므로, 이를 자세히 이해하는 것이 필수적입니다. 먼저 TCP/IP의 배경과 탄생 배경을 살펴보고, 최초 설계자들이 설정한 주요 기술적 목표를 설명하겠습니다. 그 후, TCP/IP가 OSI(개방형 시스템 상호 연결) 모델이라는 이론적 모델과 어떻게 다른지 알아보겠습니다.

TCP/IP의 간략한 역사

최초의 RFC(Request for Comments)는 1969년 4월에 발표되었으며, 이는 오늘날의 인터넷과 그 프로토콜의 토대를 마련했습니다. 이러한 프로토콜들은 각각 수많은 RFC에 명시되어 있으며, 인터넷 엔지니어링 태스크 포스(IETF)에서 이를 관찰, 유지, 승인, 보관하고 있습니다.

TCP는 1974년에 처음 등장했습니다. 1978년에는 TCP와 IP라는 두 개의 독립적인 프로토콜로 분리되었고, 1980년에 RFC 문서로 최종 확정되었습니다. 그리고 1983년, TCP/IP는 네트워크 제어 프로토콜(NCP)을 대체하고 ARPAnet에 연결되는 모든 장치의 공식 데이터 전송 프로토콜로 승인되었습니다. ARPAnet은 미국 국방부 산하 고등연구계획국(ARPA)에서 1969년에 개발한 인터넷의 전신입니다.소련의 *스푸트니크* 발사 이후 , ARPA는 곧 DARPA로 개칭되었고, ARPAnet과 MILNET으로 분리되었습니다(역시 1983년). 두 조직 모두 1990년에 해체되었습니다.

하지만 여러분이 생각하는 것과는 달리, TCP/IP 개발 작업의 대부분은 캘리포니아 북부의 UC 버클리에서 이루어졌습니다. 그곳에서 한 무리의 과학자들이 버클리 버전의 유닉스를 개발하고 있었는데, 이것이 곧 BSD(Berkeley Software Distribution) 시리즈로 알려지게 되었습니다. TCP/IP는 매우 뛰어난 성능을 보여주었기 때문에 이후 BSD 유닉스 배포판에 포함되어 다른 대학과 기관에서도 배포 테이프를 구매하면 사용할 수 있게 되었습니다. 기본적으로 TCP/IP가 포함된 BSD 유닉스는 학계에서 공유 소프트웨어로 시작하여 오늘날 인터넷의 엄청난 성공과 기하급수적인 성장은 물론 소규모 사설 및 기업 인트라넷의 기반이 되었습니다.

늘 그렇듯, TCP/IP 애호가들의 소규모 그룹으로 시작했던 것이 점차 발전하면서 미국 정부는 새로 발표되는 표준을 검증하고 특정 기준을 충족하는지 확인하는 프로그램을 만들었습니다. 이는 TCP/IP의 무결성을 보호하고 개발자가 프로토콜을 지나치게 변경하거나 독점적인 기능을 추가하는 것을 방지하기 위한 것이었습니다. 바로 이러한 개방형 시스템 접근 방식, 즉 TCP/IP 프로토콜 제품군의 이러한 특성이 수많은 하드웨어 및 소프트웨어 플랫폼 간의 안정적인 연결을 제약 없이 보장함으로써 TCP/IP의 인기를 확고히 했습니다.

TCP/IP와 국방부 모델

국방부 모델은 기본적으로 OSI 모델의 축소판이며, 7개가 아닌 4개의 계층으로 구성되어 있습니다.

- 프로세스/애플리케이션 계층
- 호스트 간 계층
- 인터넷 계층
- 네트워크 액세스 계층

그림 6.1은 국방부 모델과 OSI 참조 모델을 비교한 것입니다. 보시다시피 두 모델은 개념적으로 유사하지만, 계층의 개수와 명칭이 다릅니다.

IP 스택의 다양한 프로토콜을 논의할 때, OSI 모델과 DoD 모델의 두 계층은 서로 대체 가능합니다. 다시 말해, 인터넷 계층과 네트워크 계층은 동일한 것을 나타내며, 호스트 간 통신 계층과 전송 계층도 마찬가지입니다. DoD 모델의 나머지 두 계층인 프로세스/애플리케이션 계층과 네트워크 액세스 계층은 OSI 모델의 여러 계층으로 구성됩니다.

국방부 모델의 프로세스/애플리케이션 계층 에서는 OSI의 상위 3개 계층(애플리케이션, 표현, 세션)에 해당하는 다양한 활동과 임무를 통합하기 위해 수많은 프로토콜이 작동합니다 . 이 장의 다음 부분에서는 이러한 프로토콜을 자세히 살펴보겠습니다. 프로세스/애플리케이션 계층은 노드 간 애플리케이션 통신을 위한 프로토콜을 정의하고 사용자 인터페이스 사양도 제어합니다.

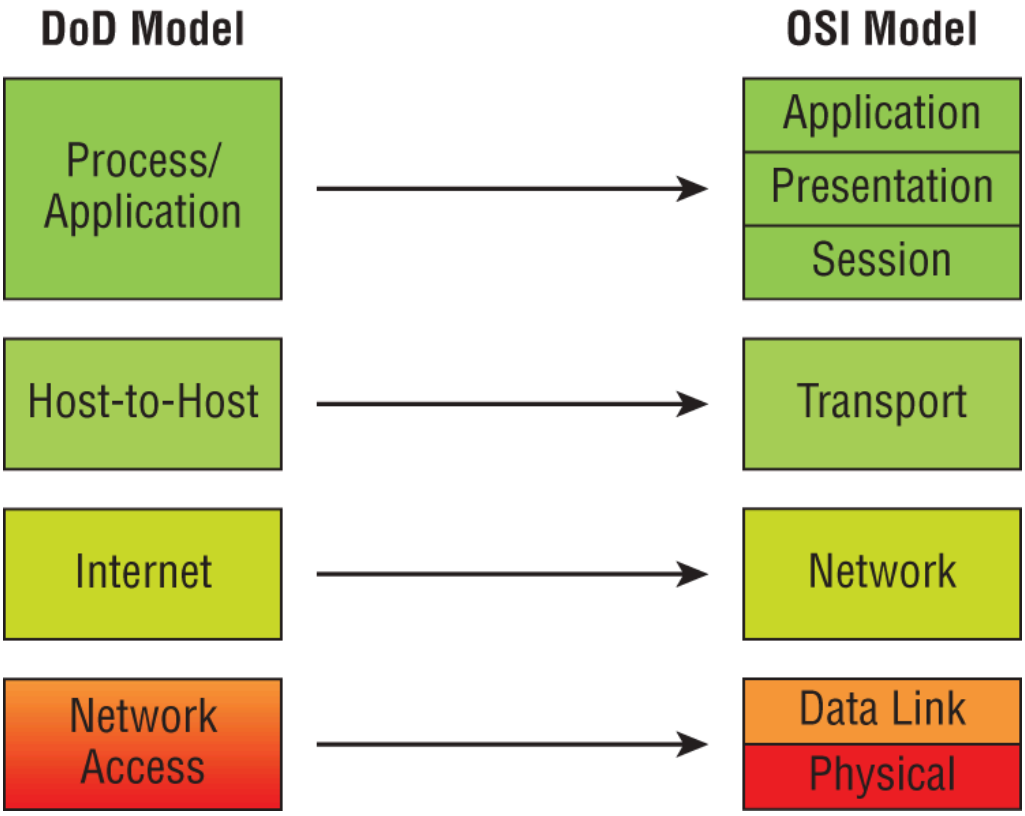

그림 6.1 국방부(DoD) 및 OSI 모델

호스트 -투-호스트 *계층은* OSI 모델의 전송 계층과 유사한 기능을 수행하며, 애플리케이션에 대한 전송 서비스 수준을 설정하는 프로토콜을 정의합니다. 이 계층은 신뢰할 수 있는 종단 간 통신을 구축하고 오류 없는 데이터 전송을 보장하는 등의 문제를 다룹니다. 또한 패킷 순서를 관리하고 데이터 무결성을 유지합니다.

인터넷 *계층은* OSI 모델의 네트워크 계층에 해당하며, 전체 네트워크를 통한 패킷의 논리적 전송과 관련된 프로토콜을 나타냅니다. 호스트에 IP 주소를 할당하여 논리적으로 주소를 지정하고, 여러 네트워크 간의 패킷 라우팅을 처리합니다.

DoD 모델의 최하단에 있는 *네트워크 액세스 계층은* 호스트와 네트워크 간의 데이터 교환을 모니터링합니다. OSI 모델의 데이터 링크 계층과 물리 계층에 해당하는 네트워크 액세스 계층은 하드웨어 주소 지정을 관리하고 데이터의 물리적 전송을 위한 프로토콜을 정의합니다.

국방부(DoD) 모델과 OSI 모델은 설계 및 개념이 유사하며, 유사한 계층에서 비슷한 기능을 수행합니다. 그림 6.2는 TCP/IP 프로토콜 제품군과 해당 프로토콜이 국방부 모델의 계층과 어떻게 관련되는지를 보여줍니다.

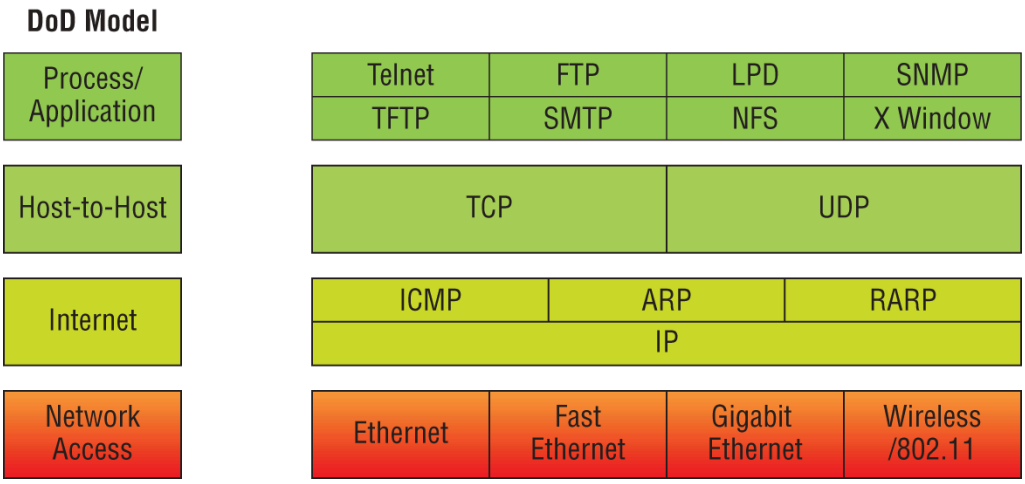

그림 6.2 TCP/IP 프로토콜 제품군

이제 프로세스/애플리케이션 계층 프로토콜부터 시작하여 다양한 프로토콜을 좀 더 자세히 살펴보겠습니다.

프로세스/애플리케이션 계층 프로토콜

다음 섹션에서는 IP 네트워크에서 일반적으로 사용되는 다양한 애플리케이션과 서비스에 대해 설명하고, 이 장에서 자세히 다루는 관련 포트 번호도 나열하겠습니다.

파일 전송 프로토콜(TCP 20, 21)

파일 전송 프로토콜(FTP) 은 IP 네트워크를 통해 파일을 전송할 수 있게 해주는 프로토콜이며, FTP를 사용하는 두 컴퓨터 간에 파일 전송이 가능합니다. 하지만 FTP는 단순히 프로토콜일 뿐만 아니라 프로그램이기도 합니다. 프로토콜로서 FTP는 애플리케이션에서 사용되며, 프로그램으로서 사용자는 FTP를 이용하여 수동으로 파일 관련 작업을 수행할 수 있습니다. FTP는 디렉터리와 파일 모두에 접근할 수 있도록 해주며, 파일을 다른 디렉터리로 이동하는 것과 같은 디렉터리 작업도 수행할 수 있습니다. 그림 6.3은 FTP의 예시를 보여줍니다.

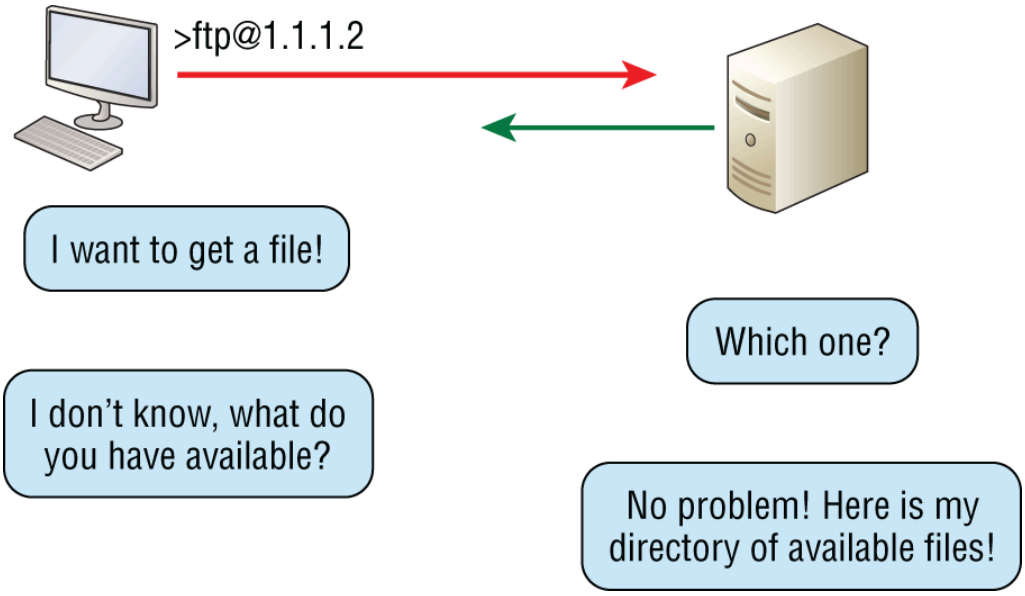

그림 6.3 파일 전송 프로토콜

FTP를 통해 호스트에 접속하는 것은 단지 첫 단계일 뿐입니다. 사용자는 시스템 관리자가 접근을 제한하기 위해 설정한 암호와 사용자 이름으로 보호되는 인증 로그인 절차를 거쳐야 합니다. *익명* 사용자 이름을 사용하면 이러한 인증 절차를 어느 정도 우회할 수 있지만 , 접근 가능한 범위는 제한될 것입니다.

FTP는 사용자가 수동으로 프로그램을 실행하더라도 디렉터리 목록을 보고 조작하거나, 파일 내용을 입력하거나, 호스트 간에 파일을 복사하는 등의 기능만 제공합니다. 원격 파일을 프로그램으로 실행할 수는 없습니다. FTP의 가장 큰 문제는 텔넷처럼 모든 데이터가 평문으로 전송된다는 점입니다. FTP 전송의 보안을 강화해야 한다면 다음 섹션에서 설명하는 SFTP를 사용해야 합니다.

보안 셸(TCP 22)

보안 *셸(SSH)* 프로토콜은 표준 TCP/IP 연결을 통해 안전한 Telnet 세션을 설정하며, 다른 시스템에 로그인하거나 원격 시스템에서 프로그램을 실행하고 파일을 한 시스템에서 다른 시스템으로 이동하는 등의 작업에 사용됩니다. 그림 6.4는 SSH의 예시를 보여줍니다.

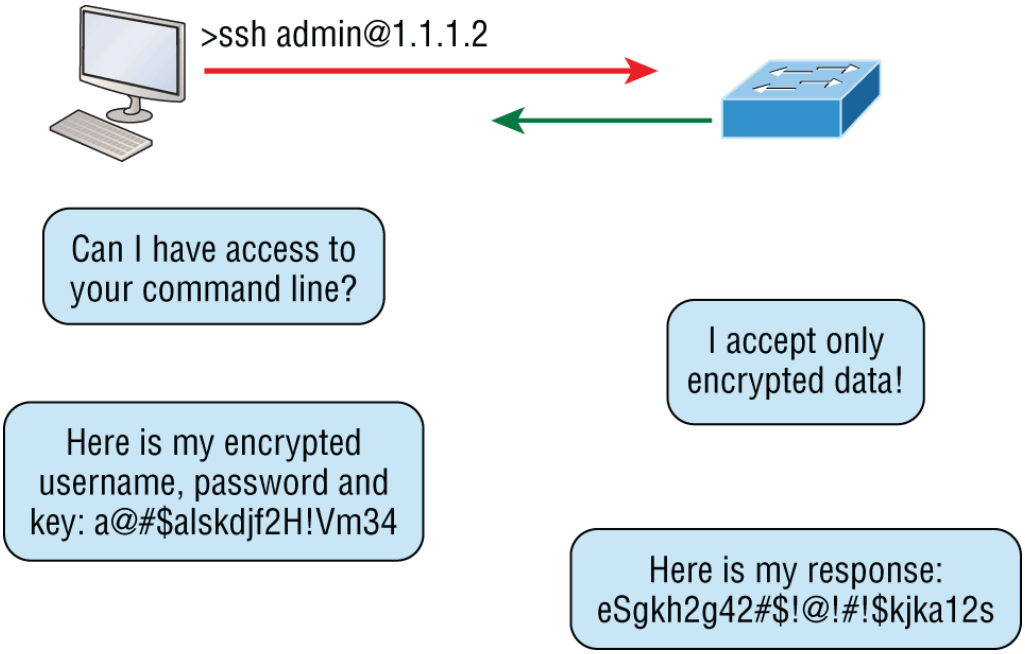

그림 6.4 SSH

`rsh` 게다가 이 모든 것을 안정적이고 강력한 암호화 연결을 유지하면서 수행합니다. 이것은 현재 텔넷을 비롯한 기존 프로토콜을 대체하는 차세대 프로토콜이라고 생각하시면 됩니다 `rlogin` .

보안 파일 전송 프로토콜(TCP 22)

SFTP(보안 파일 전송 프로토콜) 는 암호화된 연결을 통해 파일을 전송해야 할 때 사용됩니다. SFTP는 이전에 다룬 SSH 세션을 사용하여 연결을 암호화하며, SSH는 22번 포트를 사용하므로 SFTP도 22번 포트를 사용합니다.

보안 부분을 제외하면 FTP와 마찬가지로 인터넷과 같은 IP 네트워크 상에서 컴퓨터 간에 파일을 전송하는 데 사용됩니다.

텔넷(TCP 23)

텔넷은 프로토콜계의 카멜레온과 같습니다. 그 특기는 터미널 에뮬레이션입니다. 텔넷 클라이언트라고 불리는 원격 클라이언트 컴퓨터의 사용자가 텔넷 서버라고 불리는 다른 컴퓨터의 리소스에 접근할 수 있도록 해줍니다. 텔넷은 텔넷 서버를 속여 클라이언트 컴퓨터가 마치 로컬 네트워크에 직접 연결된 터미널인 것처럼 보이게 함으로써 이를 가능하게 합니다. 이 모습은 실제로는 소프트웨어 셸, 즉 선택한 원격 호스트와 상호 작용할 수 있는 가상 터미널입니다. 그림 6.5는 텔넷의 예시를 보여줍니다.

이러한 에뮬레이션된 터미널은 텍스트 모드 유형이며, 사용자가 옵션을 선택하고 복제된 서버의 애플리케이션에 액세스할 수 있는 메뉴를 표시하는 등의 정교한 절차를 실행할 수 있습니다. 사용자는 Telnet 클라이언트 소프트웨어를 실행한 다음 Telnet 서버에 로그인하여 Telnet 세션을 시작합니다.

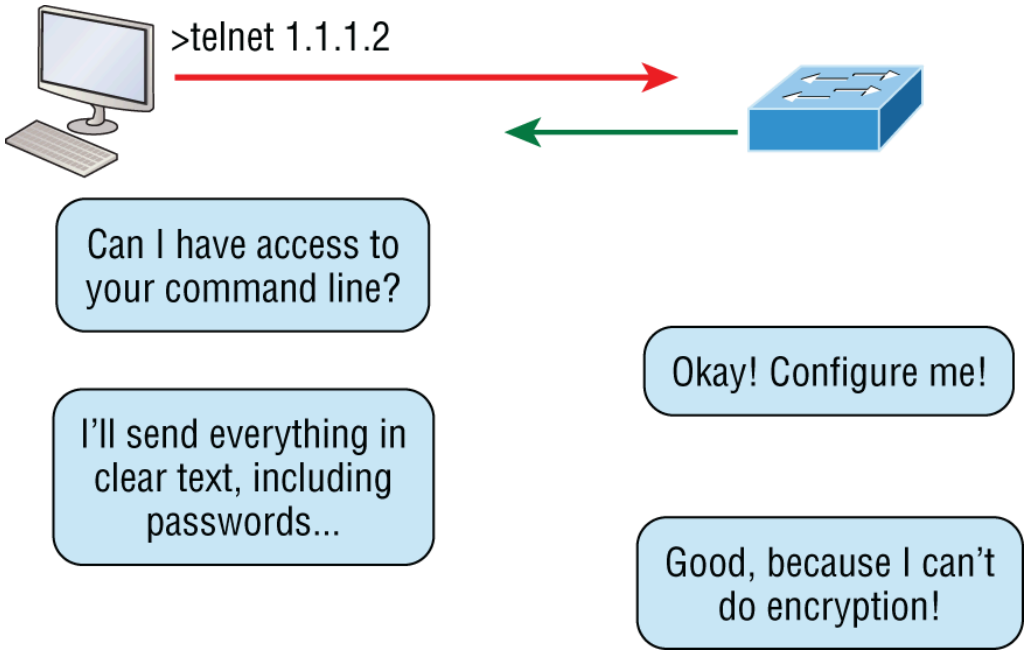

그림 6.5 텔넷

Telnet은 보안이나 암호화 기능을 제공하지 않으므로 원격 구성 세션 전반에 걸쳐 보안이 필요하거나 원하는 경우 Secure Shell(SSH)로 대체됩니다.

단순 메일 전송 프로토콜(TCP 25)

*우리가 흔히 사용하는 이메일 프로토콜인 SMTP(Simple Mail Transfer Protocol)*는 스풀 방식 또는 큐 방식을 사용하여 메일을 전달합니다. 메시지가 목적지로 전송되면 일반적으로 디스크와 같은 장치에 저장됩니다. 목적지 측 서버 소프트웨어는 큐를 주기적으로 확인하여 메시지가 있는지 점검합니다. 메시지가 감지되면 목적지로 전달합니다. SMTP는 메일을 보내는 데 사용되고, POP3는 메일을 받는 데 사용됩니다.

도메인 이름 서비스(TCP 및 UDP 53)

*도메인 이름 서비스(DNS)*는 호스트 이름, 특히 인터넷 이름(예: `/hostname/your_hostname`)을 www.lammle.com 해당 IP 주소로 변환합니다.

DNS를 사용하지 않고도 통신하려는 장치의 IP 주소를 입력할 수 있습니다. IP 주소는 네트워크와 인터넷 상의 호스트를 식별하는 데 사용됩니다. 하지만 DNS는 우리의 편의를 위해 설계되었습니다. 예를 들어, 웹 페이지를 다른 서비스 제공업체로 이전하고 싶다면 어떻게 될까요? IP 주소가 변경될 텐데, 아무도 새로운 IP 주소가 무엇인지 알 수 없을 것입니다. DNS를 사용하면 도메인 이름을 통해 IP 주소를 지정할 수 있습니다. 따라서 IP 주소를 원하는 만큼 자주 변경해도 아무도 알아채지 못합니다. 그림 6.6은 DNS의 예시를 보여줍니다.

DNS는 정규화된 도메인 이름(FQDN, 예: `www.example.com` www.lammle.com 또는 `www.example.com`)을 IP 주소 로 변환하는 데 사용됩니다 todd.lammle.com . FQDN 또는 DNS 네임스페이스는 도메인 식별자를 기반으로 시스템의 논리적 위치를 파악할 수 있는 계층 구조입니다.

'todd'라는 이름을 확인하려면 , 해당 이름의 FQDN을 직접 입력 todd.lammle.com 하거나 PC 또는 라우터와 같은 장치에서 접미사를 추가해야 합니다. 예를 들어, 시스코 라우터에서는 명령어를 사용하여 `ip domain-name` lammle.com 각 요청에 lammle.com 도메인을 추가할 수 있습니다. 이렇게 하지 않으면 DNS에서 이름을 확인하려면 FQDN을 직접 입력해야 합니다.

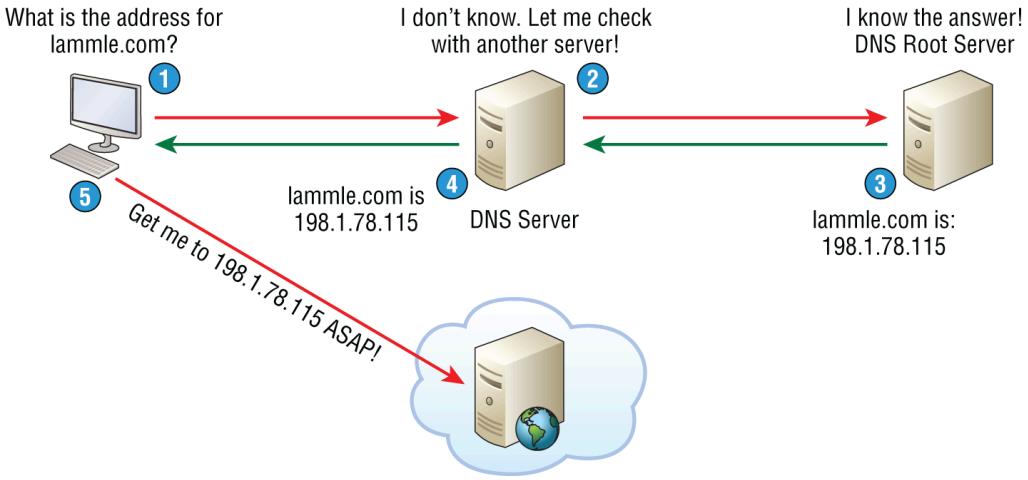

그림 6.6 도메인 이름 서비스

DNS에 대해 기억해야 할 중요한 점은 IP 주소로는 장치에 ping을 보낼 수 있지만 FQDN으로는 ping을 보낼 수 없는 경우 DNS 구성 오류가 있을 수 있다는 것입니다.

동적 호스트 구성 프로토콜/부트스트랩 프로토콜(UDP 67/68)

*DHCP(동적 호스트 구성 프로토콜)*는 서버에서 제공하는 정보를 사용하여 호스트에 IP 주소를 할당합니다. DHCP는 관리를 간소화하고 소규모 네트워크 환경부터 매우 큰 네트워크 환경까지 잘 작동합니다. 라우터를 포함한 다양한 하드웨어를 DHCP 서버로 사용할 수 있습니다.

DHCP는 부트스트랩 프로토콜(BootP)과 달리 호스트에 IP 주소를 할당하지만, 호스트의 하드웨어 주소는 BootP 테이블에 수동으로 입력해야 합니다. DHCP는 동적 BootP라고 생각할 수 있습니다. 하지만 BootP는 호스트가 부팅할 수 있는 운영 체

제를 전송하는 데에도 사용된다는 점을 기억해야 합니다. DHCP는 운영 체제를 전송할 수 없습니다.

아직 읽지 않으셨다면 5장 "네트워킹 장치" 의 DHCP 및 DNS 서버 관련 부분도 읽어보시기 바랍니다 . 두 내용 모두 시험 목표에서 중요한 부분을 차지합니다.

하지만 호스트가 DHCP 서버에 IP 주소를 요청할 때 DHCP 서버는 호스트에게 많은 정보를 제공할 수 있습니다. 다음은 DHCP 서버가 제공할 수 있는 정보의 일부 목록입니다.

- IP 주소
- 서브넷 마스크
- 도메인 이름
- 기본 게이트웨이(라우터)
- DNS
- Windows 인터넷 이름 지정 서비스(WINS) 정보

DHCP 서버는 이보다 더 많은 정보를 제공할 수 있지만, 목록에 있는 항목들은 가장 일반적인 것들입니다.

IP 주소를 받기 위해 DHCP Discover 메시지를 보내는 클라이언트는 레이어 2와 레이어 3 모두에 브로드캐스트를 보냅니다. 레이어 2 브로드캐스트는 16진수로 모두 *'F* '로 표현되며, 예를 들어 FF:FF:FF:FF:FF:FF와 같습니다. 레이어 3 브로드캐스트는 255.255.255.255이며, 이는 모든 네트워크와 모든 호스트를 의미합니다. DHCP는 비연결형 프로토콜로, 전송 계층(호스트 간 통신 계층이라고도 함)에서 UDP(사용자 데이터그램 프로토콜)를 사용합니다. UDP에 대해서는 나중에 자세히 설명하겠습니다.

제 말을 믿지 못하시겠다면, 제가 신뢰하는 분석 도구의 출력 예시를 보여드리겠습니다.

```
Ethernet II,Src:192.168.0.3(00:0b:db:99:d3:5e),Dst:Broadcast(ff:ff:ff:ff:ff
Internet Protocol,Src:0.0.0.0(0.0.0.0),Dst:255.255.255.255(255.255.255.255)
```

데이터 링크 계층과 네트워크 계층 모두 "도와주세요! 제 IP 주소를 모르겠어요!"라는 내용의 전체 구조 요청을 보내고 있습니다.

그림 6.7은 DHCP 연결을 사용한 클라이언트-서버 관계의 과정을 보여줍니다.

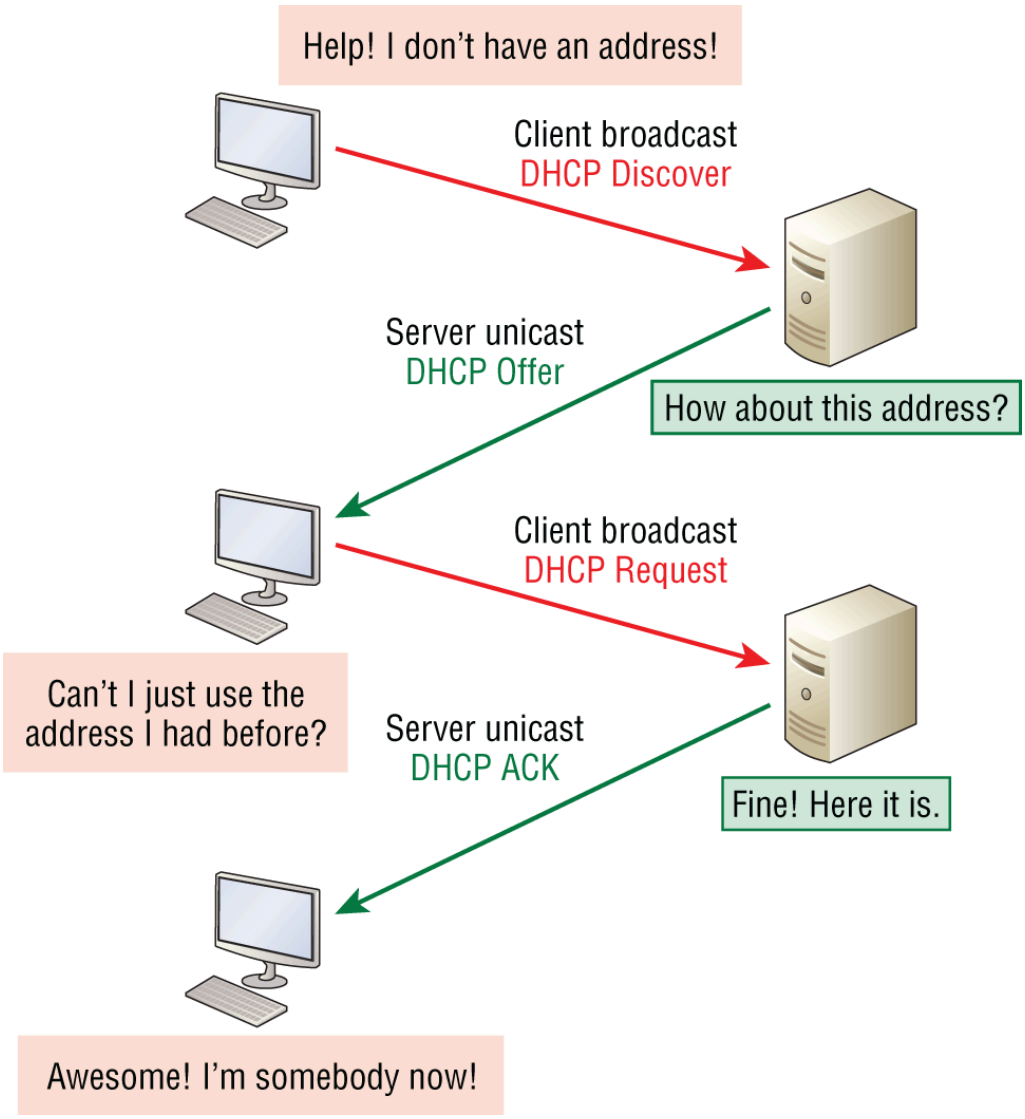

그림 6.7 DHCP 클라이언트 4단계 프로세스

다음은 클라이언트가 DHCP 서버로부터 IP 주소를 할당받기 위해 거치는 4단계 프로세스(때때로 DORA 프로세스라고도 함)입니다.

1. DHCP 클라이언트는 DHCP 서버(포트 67)를 찾기 위해 DHCP Discover 메시지를 브로드캐스트합니다.
2. DHCP Discover 메시지를 수신한 DHCP 서버는 호스트에게 유니캐스트 DHCP Offer 메시지를 다시 보냅니다.
3. 클라이언트는 서버에 DHCP 요청 메시지를 전송하여 제공된 IP 주소와 기타 정보를 요청합니다.
4. 서버는 유니캐스트 DHCP 승인 메시지를 통해 통신을 완료합니다.

스위치나 허브로 여러 호스트가 연결되어 있고 DHCP 서버가 없는 경우 어떻게 해야 할까요? IP 정보를 수동으로 추가하는 방법(*고정 IP 주소 지정*)이 있지만, Windows 최신 운영 체제에서는 APIPA(자동 개인 IP 주소 지정)라는 기능을 사용할 수 있습니다. APIPA를 사용하면 DHCP 서버를 사용할 수 없을 때 클라이언트가

IP 주소와 서브넷 마스크(호스트 간 통신에 사용되는 기본 IP 정보로, 7장 "IP 주소 지정" 및 8장 "IP 서브넷팅, IP 문제 해결 및 NAT 소개"에서 자세히 다룹니다)를 자동으로 구성할 수 있습니다. APIPA에서 사용하는 IP 주소 범위는 169.254.0.1부터 169.254.255.254까지입니다. 또한 클라이언트는 기본 클래스 B 서브넷 마스크인 255.255.0.0으로 자체 구성됩니다. DHCP 서버가 있고 호스트가 이 IP 주소를 사용하고 있다면, 호스트의 DHCP 클라이언트가 작동하지 않거나 서버가 다운되었거나 네트워크 문제로 인해 연결할 수 없다는 의미입니다.

단순 파일 전송 프로토콜(UDP 69)

TFTP(Trivial File Transfer Protocol) 는 FTP의 기본 버전으로, 필요한 파일과 그 위치를 정확히 알고 있다면 사용하기 쉽고 속도도 빠르다는 장점이 있습니다. 하지만 FTP처럼 다양한 기능을 제공하지는 않습니다. TFTP는 디렉토리 탐색 기능을 지원하지 않으며, 파일 송수신 기능만 제공합니다. 그림 6.8은 TFTP의 예시를 보여줍니다.

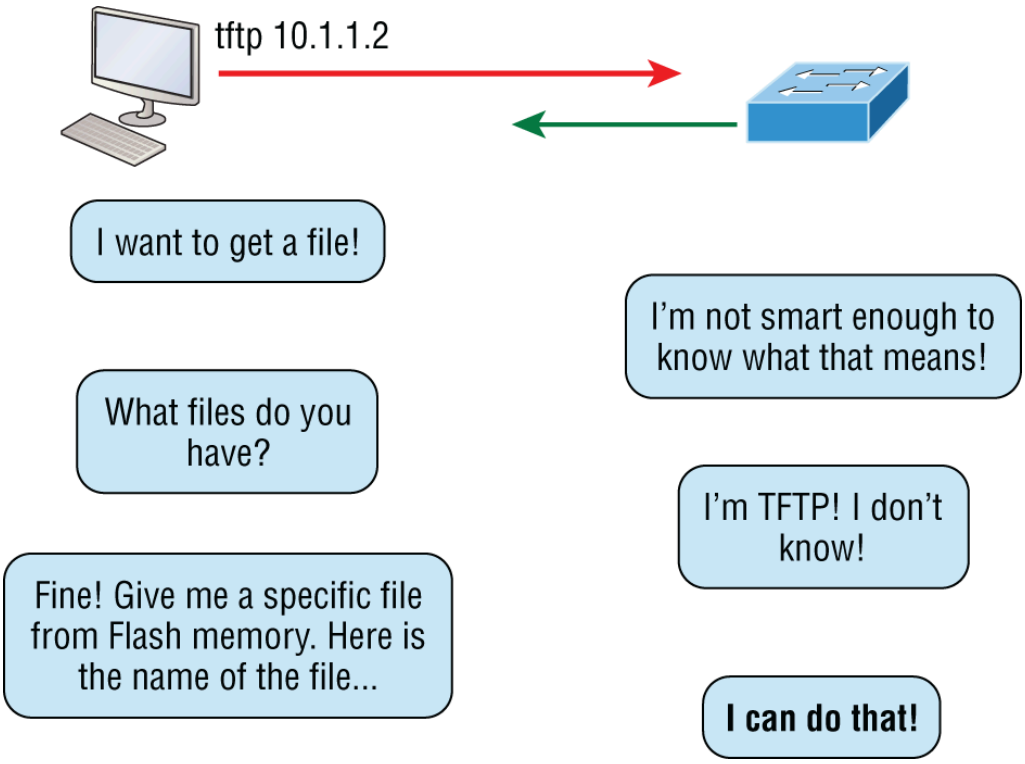

그림 6.8 사소한 FTP

이 작고 간결한 프로토콜은 데이터 전송량도 적어 FTP보다 훨씬 작은 데이터 블록을 전송하며, FTP처럼 인증 절차가 없어 보안에 취약합니다. 이러한 보안 위험 때문에 이 프로토콜을 지원하는 웹사이트는 거의 없습니다.

실제 시나리오

FTP는 언제 사용해야 할까요?

샌프란시스코 사무실 직원들이 50MB 파일을 즉시 이메일로 받아야 합니다. 어떻게 하시겠습니까? 대부분의 이메일 서버는 파일 크기 제한 때문에 이러한 파일을 거부할 것입니다. 서버에 크기 제한이 없더라도 이렇게 큰 파일을 전송하는 데는 시간이 오래 걸릴 것입니다. 이럴 때 FTP가 해결책이 될 수 있습니다! 대부분의 인터넷 서비스 제공업체(ISP)는 10MB보다 큰 파일의 이메일 전송을 허용하지 않으므로, 파일을 주고받아야 할 경우 FTP를 고려해 볼 만합니다.

대용량 파일을 주고받아야 할 때 FTP는 좋은 선택입니다. 용량이 작은 파일(10MB 미만)은 대역폭만 충분하다면(요즘 누가 대역폭이 부족하겠습니까?) 압축된 파일이라도 이메일로 보낼 수 있습니다. FTP를 사용하려면 파일을 공유할 수 있도록 인터넷에 FTP 서버를 구축해야 합니다.

게다가 FTP는 이메일보다 빠르기 때문에 대용량 파일을 주고받을 때 FTP를 사용하는 것이 유리합니다. 또한 FTP는 TCP를 사용하고 연결 지향 방식이기 때문에 세션이 끊어지더라도 중단된 부분부터 다시 시작할 수 있습니다. 이메일 클라이언트로는 불가능한 일이죠!

하이퍼텍스트 전송 프로토콜(TCP 80)

그래픽, 텍스트, 링크 등이 혼합된 멋진 웹사이트들은 모두 *하이퍼텍스트 전송 프로토콜(HTTP)* 덕분에 가능한 것입니다. 그림 6.9는 HTTP의 예시를 보여줍니다.

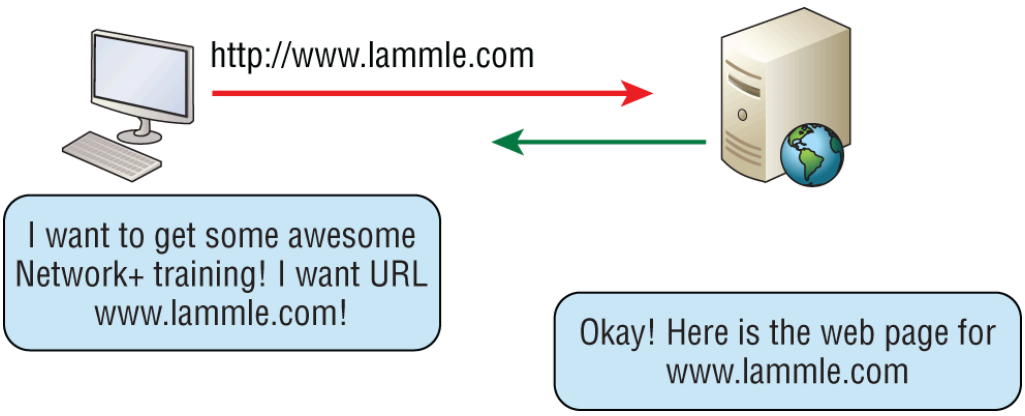

그림 6.9 HTTP

HTTP는 웹 브라우저와 웹 서버 간의 통신을 관리하는 데 사용되며, 링크를 클릭하면 해당 리소스가 실제로 어디에 있든 올바른 리소스를 엽니다. 이러한 유형의 데이터 전송에 일반적으로 사용되는 HTTPS에 대한 내용은 해당 섹션을 참조하십시오.

우체국 프로토콜 v3(TCP 110)

POP(Post Office Protocol) 은 수신 메일을 저장하는 기능을 제공하며, 최신 버전은 POP3라고 합니다(어디서 많이 들어보셨나요?). 기본적으로 이 프로토콜은 클라이언트 장치가 POP3 서버에 연결되면 해당 클라이언트로 전송된 메시지를 다운로드할 수 있도록 제공하는 방식으로 작동합니다. 메시지를 선택적으로 다운로드하는 기능은 없지만, 일단 다운로드가 완료되면 클라이언트와 서버 간의 상호 작용이 종료되고 사용자는 로컬에서 메시지를 삭제하거나 수정할 수 있습니다. 보안상의 이유로 POP3 대신 IMAP이라는 새로운 표준이 점점 더 많이 사용되고 있습니다.

네트워크 시간 프로토콜(UDP 123)

델라웨어 대학교의 데이비드 밀스 교수는 우리 컴퓨터의 시계를 하나의 표준 시간 소스(일반적으로 원자 시계)에 동기화하는 데 사용되는 이 편리한 프로토콜을 고안해 냈습니다. *네트워크 시간 프로토콜(NTP)*은 다른 동기화 유틸리티와 함께 작동하여 주어진 네트워크의 모든 컴퓨터가 시간을 일치하도록 합니다. 그림 6.10을 참조하십시오 .

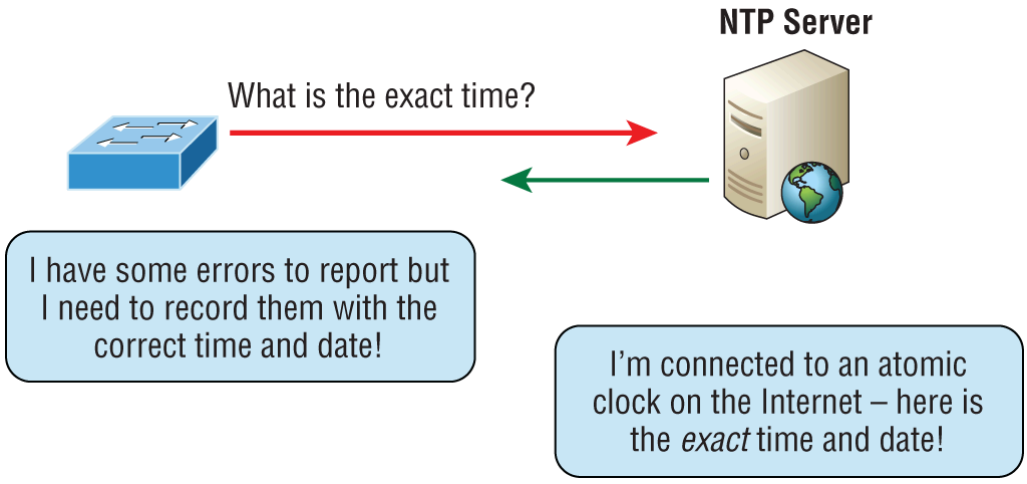

그림 6.10 네트워크 시간 프로토콜

언뜻 보기엔 간단해 보이지만, 오늘날 이루어지는 수많은 거래에 시간과 날짜가 기록되기 때문에 매우 중요합니다. 소중한 데이터베이스를 생각해 보세요. 서버가 연결된 기기들과 동기화되지 않으면, 단 몇 초 차이만으로도 서버에 심각한 문제가 발생할 수 있습니다(시스템 다운!). 예를 들어, 어떤 기기에서 새벽 1시 50분에 거래를 입력했는데 서버에는 그 거래가 새벽 1시 45분에 발생한 것으로 기록된다면, 이는 심각한 오류입니다. 즉, NTP는 마치 타임머신을 타고 과거로 돌아가는 것처럼 네트워크가 마비되는 것을 방지하는 역할을 합니다. 정말 중요한 기능이죠!

인터넷 메시지 액세스 프로토콜(TCP 143)

*인터넷 메시지 액세스 프로토콜(IMAP)*을 사용하면 메일을 다운로드하는 방식을 직접 제어할 수 있으므로, 필요한 보안을 강화할 수 있습니다. 메시지 헤더를 미리

보거나 메시지의 일부만 다운로드할 수 있어, 마치 미끼를 조금씩만 먹는 것처럼 안전하게 메일을 확인할 수 있습니다. 마치 메일 전체를 삼켰다가 안에 숨겨진 낚싯바늘에 걸려 숨이 막히는 위험을 피하는 것과 같습니다!

IMAP을 사용하면 이메일 서버에 메시지를 계층적으로 저장하고 문서 및 사용자 그룹과 연결할 수도 있습니다. IMAP은 제목, 헤더 또는 내용을 기반으로 메시지를 검색할 수 있는 검색 명령도 제공합니다. 예상할 수 있듯이 강력한 인증 기능을 갖추고 있으며, MIT에서 개발한 Kerberos 인증 체계를 지원합니다. 그리고 현재 버전은 IMAP4입니다.

단순 네트워크 관리 프로토콜(UDP 161/162)

*단순 네트워크 관리 프로토콜(SNMP)*은 중요한 네트워크 정보를 수집하고 처리합니다. SNMP는 관리 스테이션에서 네트워크 상의 장치들을 고정된 또는 임의 간격으로 폴링하여 데이터를 수집하고, 장치들이 특정 정보를 공개하도록 요구합니다. 모든 것이 정상일 때, SNMP는 *기준선(baseline*)이라는 보고서를 수신하는데, 이는 건전한 네트워크의 운영 특성을 나타내는 지표입니다. 이 프로토콜은 네트워크 감시자 역할을 하여 갑작스러운 상황 변화를 관리자에게 신속하게 알릴 수도 있습니다. 이러한 네트워크 감시자를 *에이전트 라고 하며, 이상이 발생하면 에이전트는 트랩* (trap) 이라는 경고를 관리 스테이션으로 보냅니다. 네트워크 관리 시스템(NMS)은 관리 정보 베이스(MIB)를 통해 에이전트들을 폴링합니다. MIB는 기본적으로 NMS가 장치 또는 네트워크의 상태에 대해 에이전트에게 질문할 수 있는 미리 정의된 질문들이 저장된 데이터베이스입니다. 그림 6.11은 NMS 스테이션의 작동 모습을 보여줍니다.

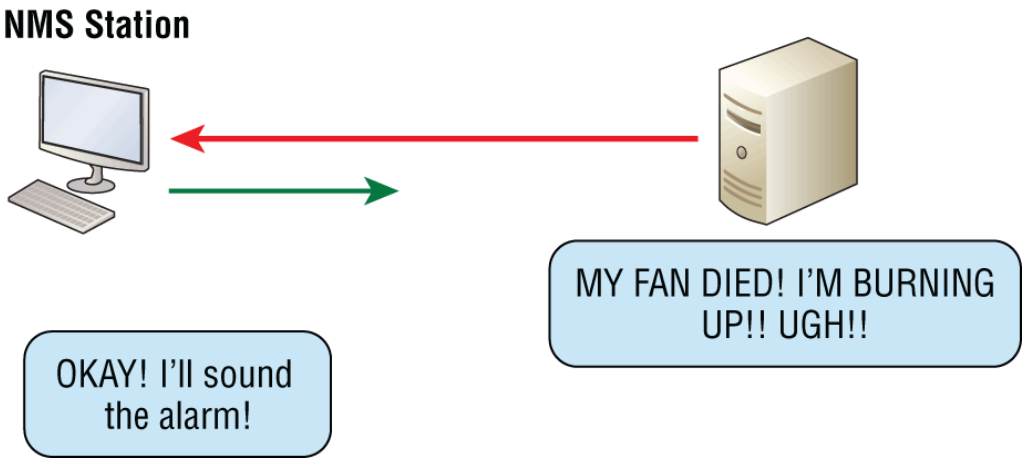

그림 6.11 네트워크 관리 스테이션

또한 SNMP는 네트워크 설정 과정과 전체 인터넷워크 관리를 간소화하는 데 도움이 될 수 있습니다.

SNMP 버전 1, 2 및 3

SNMP 버전 1과 2는 사실상 구식입니다. 네트워크에서 완전히 사라진 것은 아니지만, 버전 1은 매우 오래되었고, 말 그대로 시대에 뒤떨어졌습니다. SNMP 버전 2는 특히 성능 면에서 개선된 기능을 제공했습니다. 그중에서도 가장 큰 장점 중 하나는 호스트가 한 번에 많은 양의 데이터를 가져올 수 있게 해주는 GETBULK 기능입니다. 하지만 버전 2는 네트워크 업계에서 널리 사용되지는 않았습니다. 현재 표준으로 자리 잡은 SNMP 버전 3은 버전 1이 UDP만 사용했던 것과 달리 TCP와 UDP를 모두 사용합니다. 버전 3은 보안, 메시지 무결성, 인증 및 암호화 기능을 더욱 강화했습니다. 따라서 SNMP 버전 1과 2를 사용할 때는 패킷 스니퍼에 의해 데이터가 읽힐 수 있으므로 주의해야 합니다.

경량 디렉터리 액세스 프로토콜(TCP 389)

규모가 꽤 큰 네트워크의 시스템 관리자라면 누구나 장치와 사용자 같은 네트워크 리소스를 관리하는 디렉터리 시스템을 가지고 있을 가능성이 높습니다. 그런데 이러한 디렉터리에 어떻게 접근할까요? 바로 *LDAP(Lightweight Directory Access Protocol)* 를 통해서 입니다. LDAP는 Microsoft Active Directory와 같은 디렉터리 서비스 시스템에 접근하고 정보를 조회하는 데 사용되는 프로토콜입니다. 그리고 LDAP에는 보안 버전인 LDAPS라는 것이 있는데, 이는 636번 포트를 사용하며, 이에 대해서는 나중에 자세히 설명하겠습니다.

이 프로토콜은 디렉터리에 접근하는 방식을 표준화하며, 최초 및 두 번째 초안은 각각 RFC 1487 및 1777에 설명되어 있습니다. 초기 두 버전에는 몇 가지 오류가 있었기 때문에 이러한 문제를 해결하기 위해 세 번째 버전(오늘날 가장 일반적으로 사용되는 버전)이 만들어졌으며, 이는 RFC 3377에 설명되어 있습니다.

하이퍼텍스트 전송 프로토콜 보안(TCP 443)

HTTPS(Hypertext Transfer Protocol Secure) 는 HTTP의 보안 버전으로, 웹 브라우저와 서버 간의 안전한 데이터 전송을 위한 다양한 보안 기능을 제공합니다. 브라우저에서 양식을 작성하고, 로그인하고, 인증하고, 온라인 예약이나 상품 구매 시 HTTP 메시지를 암호화하는 데 필요한 것이 바로 HTTPS입니다.

SSH(포트 22)와 HTTPS(포트 443)는 모두 인트라넷과 인터넷을 통해 패킷을 암호화하는 데 사용됩니다.

전송 계층 보안/보안 소켓 계층(TCP 995/465)

TLS(전송 계층 보안) 와 그 전신인 SSL(보안 소켓 계층)은 모두 웹 브라우징, 인스턴트 메시징, 인터넷 팩스 등과 같은 안전한 온라인 데이터 전송 활동을 가능하게 하는 데 매우 유용한 암호화 프로토콜입니다. 두 프로토콜은 매우 유사하여 이 책에서 차이점을 자세히 설명하는 것은 적절하지 않습니다. 둘 다 X.509 인증서와 비대칭 암호화를 사용하여 통신 대상 호스트에 인증하고 키를 교환합니다. 이 키는 호스트 간에 전송되는 데이터를 암호화하는 데 사용됩니다. 이를 통해 데이터/메시지의 기밀성, 무결성 및 인증이 보장됩니다.

Gmail을 사용하는 경우 TLS/SSL이 995번 포트와 465번 포트를 사용한다고 명시했지만, TLS/SSL은 특정 포트에 국한되지 않고 다양한 포트를 사용할 수 있습니다.

서버 메시지 블록(TCP 445)

SMB(Server Message Block) 는 Microsoft Windows 네트워크에서 호스트 간에 파일 및 프린터 액세스 공유 및 기타 통신에 사용됩니다. SMB는 현재 대부분 TCP 포트 445에서 실행되지만, UDP 포트 137 및 138, 그리고 NetBIOS를 사용하는 TCP 포트 137 및 139에서도 실행될 수 있습니다.

시스템 로그(UDP 514)

스위치나 라우터의 내부 버퍼에서 시스템 메시지를 읽는 것은 특정 시점에 네트워크에서 무슨 일이 일어나고 있는지 확인하는 가장 일반적이고 효율적인 방법입니다. 하지만 가장 좋은 방법은 *syslog 서버에 메시지를 기록하는 것입니다. syslog* 서버는 사용자가 입력한 메시지를 저장하고, 타임스탬프를 추가하거나 순서대로 정렬해 주는 기능까지 제공하며, 설정 및 구성도 간편합니다!

Syslog를 사용하면 메시지를 표시하고 정렬하고 검색할 수도 있어 문제 해결에 매우 유용한 도구입니다. 특히 키워드와 심각도 수준을 활용하여 검색할 수 있는 강력한 기능을 제공합니다. 또한 서버는 메시지의 심각도 수준에 따라 관리자에게 이메일을 보낼 수 있습니다.

네트워크 장치는 syslog 메시지를 생성하고 다양한 대상으로 전달하도록 구성할 수 있습니다. 다음 네 가지 예는 Cisco 장치에서 메시지를 수집하는 일반적인 방법입니다.

- 로깅 버퍼(기본적으로 활성화됨)
- 콘솔 라인(기본적으로 활성화됨)
- 터미널 라인(명령어 사용 `terminal monitor`)
- Syslog 서버

심각도 수준은 가장 심각한 수준부터 가장 덜 심각한 수준까지 표 6.1 에 설명되어 있습니다 . 정보(Information)는 기본값이며, 이 경우 모든 메시지가 버퍼와 콘솔로 전송됩니다.

표 6.1 심각도 수준

심각도 수준	설명
응급 상황 (심각도 0)	시스템을 사용할 수 없습니다.
경고 (심각도 1)	즉각적인 조치가 필요합니다.
심각도 2 (위급)	위독한 상태입니다.
오류 (심각도 3)	오류 조건.
경고 (심각도 4)	경고 상황입니다.
알림 (심각도 5)	흔하지만 중요한 상태입니다.
정보 (심각도 6)	일반 정보 메시지입니다.
디버깅 (심각도 7)	디버깅 메시지입니다.

SMTP TLS (TCP 587)

앞서 설명했듯이, 우리가 일상적으로 사용하는 이메일 전송 프로토콜인 SMTP(Simple Mail Transfer Protocol)는 TCP 포트 25를 사용하여 스풀링 또는 큐 방식을 통해 메일을 전송합니다. 하지만 이 프로토콜로 전송되는 이메일은 평문으로 전송되며, 일부 이메일 서버는 여전히 이를 허용할 수 있습니다.

SMTP TLS는 이메일을 전송할 때 암호화하며, 대부분의 이메일 서버는 현재 이메일 전송에 587번 포트를 사용하거나 사용할 수 있으며, 일부 서버는 이 포트 사용을 필수로 요구하기도 합니다. 이 포트와 TLS 암호화를 통해 이메일은 IETF에서 제시한 지침을 준수하여 안전하게 전송됩니다.

SSL을 통한 경량 디렉터리 액세스 프로토콜(TCP 636)

앞서 TCP 389 LDAP에 대해 논의했듯이, 이 트래픽은 보안되지 않은 상태로 전송됩니다. 이제 SSL을 사용하는 TCP 636 LDAP에 대해 알아보겠습니다. 이는 오늘날 네트워크에서 권장되는 LDAP 사용 방식입니다. 이 기능을 제대로 사용하려면 Microsoft 인증 기관(CA) 또는 다른 유형의 CA에서 발급한 적절한 인증서를 설치하기만 하면 됩니다.

SSL을 통한 IMAP(TCP 993)

인터넷 메시지 액세스 프로토콜(IMAP)은 사용자가 메일을 다운로드하는 방식을 제어할 수 있도록 해주기 때문에 보안을 강화해줍니다. 특히, SSL을 통한 IMAP은 IMAP 트래픽이 보안 소켓을 통해 보안 포트(일반적으로 TCP 포트 993)로 전송됨을 의미합니다.

SSL을 통한 POP3(TCP 995)

앞서 설명했듯이 POP(Post Office Protocol)는 수신 메일을 저장하는 기능을 제공하며, POP의 최신 버전은 POP3입니다. 이 이메일은 암호화되지 않은 일반 텍스트로 다운로드되었을 가능성이 높습니다. 오늘날 서버에서 이메일을 다운로드할 때는 POP3 over SSL 또는 IMAP over SSL을 사용하여 이메일을 암호화하는 것이 일반적입니다.

구조화 질의 언어(SQL) 서버(TCP 1433)

Microsoft SQL Server는 단순한 관계형 데이터베이스 엔진에서 다목적 엔터프라이즈급 데이터 플랫폼으로 발전해 왔습니다. TCP 포트 1433은 SQL Server의 기본 포트이며, 인터넷 주소 관리 기관(IANA)에서 지정한 공식 소켓 번호이기도 합니다. 클라이언트 시스템은 TCP 1433 포트를 사용하여 데이터베이스 엔진에 연결합니다.

SQLnet (TCP 1521)

SQLNet(SQL*Net 및 Net8이라고도 함)은 Oracle의 네트워킹 소프트웨어로, Oracle 데이터베이스를 사용하는 프로그램 간의 원격 데이터 액세스를 가능하게 합니다. 애플리케이션과 데이터베이스는 서로 다른 컴퓨터에서 공유되지만 마치 로컬에 있는 것처럼 통신을 유지합니다.

SQLnet은 모든 네트워크 프로토콜에 대한 범용 인터페이스를 제공하는 네트워크 기술인 Oracle의 투명 네트워크 기판(TNS)을 기반으로 합니다. 하지만 오늘날 TCP/IP가 존재하기 때문에 더 이상 필요하지 않다는 것은 쉽게 짐작할 수 있습니다.

SQL*Net은 클라이언트와 서버가 서로 통신하는 데 사용됩니다. Net8 계층이 인터프리터 역할을 하지 않으면 클라이언트 프로세스와 서버 프로세스는 상호 연결될 수 없습니다(이 섹션에서 세 가지 이름을 모두 사용했는데, 모두 동일한 것을 의미합니다).

MySQL (TCP 3306)

MySQL은 구조화 질의 언어(SQL)를 기반으로 하는 관계형 데이터베이스 관리 시스템입니다. 기업에서는 데이터 웨어하우징, 전자상거래, 로깅 애플리케이션 등 다양한 분야에서 사용됩니다. MySQL은 특히 클라우드 기반 데이터베이스 구축에 가장 널리 사용됩니다.

원격 데스크톱 프로토콜(TCP 3389)

원격 데스크톱 프로토콜(RDP) 은 마이크로소프트에서 개발한 독점 프로토콜입니다. 이를 통해 다른 컴퓨터에 연결하여 프로그램을 실행할 수 있습니다. RDP는 텔넷과 유사하게 작동하지만, 텔넷처럼 명령 프롬프트가 표시되는 대신 원격 컴퓨터의 그래픽 사용자 인터페이스(GUI)가 표시됩니다. 대부분의 Windows 버전용 클라이언트가 있으며, Mac에는 RDP 클라이언트가 사전 설치되어 있습니다.

마이크로소프트는 현재 공식 RDP 서버 소프트웨어를 원격 데스크톱 서비스(Remote Desktop Services)라고 부릅니다. 이전에는 터미널 서비스(Terminal Services)라고 불렸습니다. 마이크로소프트의 공식 클라이언트 소프트웨어는 현재 원격 데스크톱 연결(RDC)이라고 하며, 과거에는 터미널 서비스 클라이언트(Terminal Services Client)라고 불렸습니다.

RDP는 원격 클라이언트에게 매우 유용한 도구로, 사용자가 집에서 회사 컴퓨터에 연결하여 집 컴퓨터에 소프트웨어를 실행하거나 설치하지 않고도 이메일을 확인하거나 다른 응용 프로그램을 사용하여 작업을 수행할 수 있도록 해줍니다.

SIP(VoIP)(TCP 또는 UDP 5060/TCP 5061)

SIP(Session Initiation Protocol) 는 음성 및 화상 통화, 화상 회의, 멀티미디어 스트리밍, 인스턴트 메시징, 현재 상태 정보, 온라인 게임 등 다양한 멀티미디어 통신 세션을 구성하고 해제하는 데 사용되는 매우 인기 있는 신호 프로토콜입니다.

RTP(VoIP)(UDP 5004/TCP 5005)

실시간 전송 프로토콜(RTP)은 인터넷을 통해 오디오와 비디오를 전송하기 위한 패킷 형식 표준입니다. 원래 멀티캐스트 프로토콜로 설계되었지만, 현재는 유니캐스트 애플리케이션에도 사용됩니다. 스트리밍 미디어, 화상 회의, 푸시 투 토크 시스

템 등에서 널리 사용되며, 이러한 이유로 VoIP(Voice over IP) 업계에서 사실상의 표준으로 자리 잡았습니다.

MGCP(멀티미디어)(TCP 2427/2727)

MGCP(미디어 게이트웨이 제어 프로토콜) 는 멀티미디어 회의 중에 필요한 신호 처리 및 세션 관리를 위한 표준 프로토콜입니다.

이 프로토콜은 회선 교환 네트워크에 필요한 형식의 데이터를 패킷 교환 네트워크에 필요한 형식으로 변환하는 미디어 게이트웨이와 미디어 게이트웨이 컨트롤러 간의 통신 수단을 정의합니다.

MGCP는 여러 엔드포인트 간의 통화를 설정, 유지 및 종료하는 데 사용할 수 있습니다.

H.323 (비디오) (TCP 1720)

H.323 은 IP 네트워크 상에서 실시간 오디오, 비디오 및 데이터 정보의 전송 방식을 정의하는 비디오 표준 프로토콜입니다. 이 표준은 신호 처리, 멀티미디어 및 대역폭 제어 메커니즘을 제공합니다. H.323은 통신에 RTP 표준을 사용합니다.

인터넷 그룹 관리 프로토콜

인터넷 그룹 관리 프로토콜(IGMP) 은 IP 멀티캐스트 세션을 관리하는 데 사용되는 TCP/IP 프로토콜입니다. IGMP는 네트워크를 통해 고유한 IGMP 메시지를 전송하여 멀티캐스트 그룹 현황을 파악하고 어떤 호스트가 어떤 멀티캐스트 그룹에 속해 있는지 확인합니다. IP 네트워크의 호스트 컴퓨터는 IGMP 메시지를 사용하여 그룹에 가입하거나 탈퇴할 수 있습니다. IGMP 메시지는 그룹 멤버십 및 활성 멀티캐스트 스트림을 추적하는 데 매우 유용합니다. IGMP는 네트워크 계층에서 작동하며 포트 번호를 사용하지 않습니다.

NetBIOS(TCP 및 UDP 137~139)

네트워크 기본 입출력 시스템(BIOS)은 OSI 모델의 상위 계층에서만 작동하며, 서로 다른 컴퓨터들이 네트워크를 통해 통신할 수 있도록 인터페이스를 제공합니다.

NetBIOS는 1980년대 초 IBM LAN에서 작동하도록 처음 개발되었으며 독점 기술이었습니다. 마이크로소프트와 노벨은 각각 자사 호스트가 서버와 통신할 수 있도록 NetBIOS 구현체를 개발했지만, 마이크로소프트 버전이 사실상 표준이 되었습니다.

호스트 간 계층 프로토콜

호스트 간 계층의 주요 목적은 상위 계층 애플리케이션을 네트워크의 복잡성으로부터 보호하는 것입니다. 이 계층은 상위 계층에게 "데이터 스트림과 필요한 지침만 제공해 주시면, 정보를 전송할 준비를 시작하겠습니다."라고 말합니다.

다음 섹션에서는 이 계층의 두 가지 프로토콜에 대해 설명합니다.

- 전송 제어 프로토콜(TCP)
- 사용자 데이터그램 프로토콜(UDP)

또한, 주요 호스트 간 프로토콜 개념과 포트 번호에 대해서도 살펴보겠습니다.

전송 제어 프로토콜

*TCP(전송 제어 프로토콜)*는 애플리케이션에서 전송되는 대량의 정보를 여러 세그먼트로 나눕니다. 각 세그먼트에는 번호를 매기고 순서를 지정하여 수신측 TCP 프로세스가 애플리케이션이 의도한 순서대로 세그먼트를 다시 배열할 수 있도록 합니다. 이러한 세그먼트가 전송된 후, 송신측 TCP는 수신측 TCP 프로세스로부터 수신 확인 응답을 기다리고, 확인 응답을 받지 못한 세그먼트를 재전송합니다.

신뢰할 수 있는 전송 작업에서 데이터를 전송하려는 장치는 세션을 생성하여 원격 장치와 연결 지향 통신을 설정한다는 점을 기억하십시오. 전송 장치는 먼저 상대 시스템과 연결 지향 세션을 설정하는데, 이 세션을 *호출 설정* 또는 *3방향 핸드셰이크*라고 합니다 . 그런 다음 데이터가 전송되고, 전송이 완료되면 가상 회로를 종료하기 위해 호출 종료가 이루어집니다.

TCP는 전이중 연결 지향형 프로토콜로, 신뢰성과 정확성이 뛰어납니다. 하지만 이러한 모든 조건을 충족하고 오류 검사까지 구현하는 것은 결코 쉬운 일이 아닙니다. TCP는 매우 복잡하기 때문에 네트워크 오버헤드 측면에서 비용이 많이 드는 것은 당연합니다. 게다가 오늘날의 네트워크는 과거보다 훨씬 더 안정적이기 때문에 이러한 추가적인 안정성이 불필요한 경우가 많습니다. 대부분의 프로그래머는 프로그래밍 작업을 줄이기 위해 TCP를 사용하지만, 실시간 비디오 및 VoIP의 경우 오버헤드가 더 적은 *UDP(사용자 데이터그램 프로토콜)가* 더 나은 경우가 많습니다.

TCP 세그먼트 형식

상위 계층은 전송 계층의 프로토콜로 데이터 스트림을 그대로 보내기 때문에, 그림 6.12를 사용하여 TCP가 데이터 스트림을 어떻게 분할하고 인터넷 계층에 전달할 준비를 하는지 설명하겠습니다. 인터넷 계층은 데이터 스트림을 수신하면, 각 세그먼트를 패킷으로 변환하여 인터넷 네트워크를 통해 전송합니다. 수신 호스트의 호스트 간 통신(H2H) 계층 프로토콜은 이러한 세그먼트를 다시 구성하여 상위 계층 애플리케이션이나 프로토콜에 전달합니다.

TCP 헤더는 24바이트 길이이며, 옵션을 사용하면 최대 60바이트까지 가능합니다. 그림 6.12는 TCP 세그먼트 형식과 TCP 헤더 내의 다양한 필드를 보여줍니다.

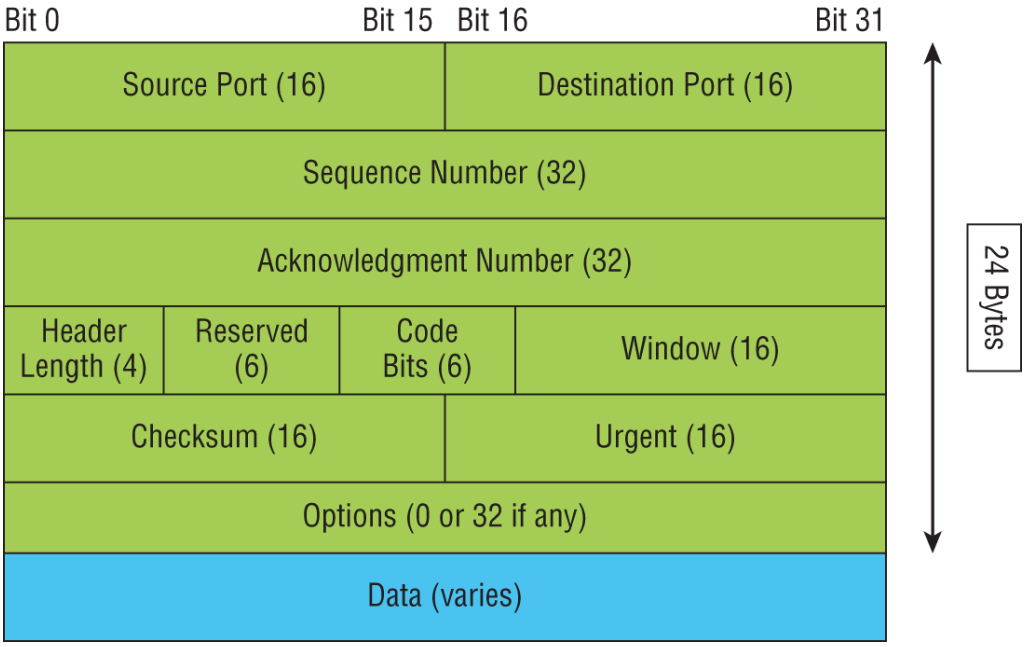

그림 6.12 TCP 세그먼트 형식

다시 말하지만, 탄탄한 교육적 기반을 구축하기 위해서는 TCP 부문의 각 분야가 무엇인지 이해하는 것이 좋습니다.

- **소스 포트:** 이는 데이터를 전송하는 호스트의 애플리케이션 포트 번호입니다. 이에 대해서는 나중에 포트 번호 섹션에서 더 자세히 설명하겠습니다.
- **대상 포트:** 이는 대상 호스트에서 애플리케이션이 요청되는 포트 번호입니다.
- **시퀀스 번호는** TCP에서 데이터를 올바른 순서로 복원하거나 시퀀싱이라는 프로세스 중에 누락되거나 손상된 데이터를 재전송하는 데 사용되는 번호입니다.
- **확인 응답 번호는** 다음에 수신될 것으로 예상되는 TCP 옥텟 값입니다.
- **헤더 길이** TCP 헤더에 포함된 32비트 워드의 개수로, 데이터가 시작되는 위치를 나타냅니다. TCP 헤더(옵션 헤더 포함)는 모두 32비트의 정수 길이입니다.
- **예약됨.** 항상 0으로 설정됩니다.
- **코드 비트/TCP 플래그는** 세션을 설정하고 종료하는 데 사용되는 기능을 제어합니다.
- **윈도우:** 송신자가 허용할 수 있는 윈도우 크기(옥텟 단위).
- **체크섬은** TCP가 하위 계층을 신뢰하지 않고 모든 것을 검사하기 때문에 사용되는 순환 중복 검사(CRC)입니다. CRC는 헤더와 데이터 필드를 검사합니다.
- **긴급** A 필드는 코드 비트의 긴급 포인터가 설정된 경우에만 유효합니다. 설정된 경우, 이 값은 현재 시퀀스 번호에서 비긴급 데이터 세그먼트가 시작되는 위치의 오프셋(옥텟 단위)을 나타냅니다.
- **옵션은** 0일 수도 있고(옵션이 전혀 없어도 됨), 32비트의 배수일 수도 있습니다. 하지만 옵션 필드의 합계가 32비트의 배수가 되지 않는 경우, 데이터가 32비트 경계(워드)에서 시작하도록 0으로 채워야 합니다.

- **페이로드(데이터)는** 상위 계층 헤더를 포함하여 전송 계층에서 TCP 프로토콜로 전달됩니다.

네트워크 분석기에서 복사한 TCP 세그먼트를 살펴보겠습니다. 다음 출력에서 패킷이 목적지 호스트로 전달하는 페이로드(데이터) 영역을 굵게 표시했습니다.

```
TCP - Transport Control Protocol
Source Port: 5973
Destination Port: 23
Sequence Number: 1456389907
Ack Number: 1242056456
Offset: 5
Reserved: %000000
Code: %011000
Ack is valid
Push Request
Window: 61320
Checksum: 0x61a6
Urgent Pointer: 0
No TCP Options
TCP Data Area:
vL.5.+.5.+.5.+.5 76 4c 19 35 11 2b 19 35 11 2b 19 35 11
2b 19 35 +. 11 2b 19
Frame Check Sequence: 0x0d00000f
```

앞서 언급한 모든 내용이 해당 세그먼트에 포함되어 있다는 것을 눈치채셨나요? 헤더의 필드 수를 보면 TCP가 상당한 오버헤드를 발생시킨다는 것을 알 수 있습니다. 바로 이 때문에 애플리케이션 개발자들은 안정성보다 효율성을 우선시하여 오버헤드를 줄이고 UDP를 선택하는 경우가 있습니다. UDP는 전송 계층에서 TCP의 대안으로 정의되어 있기도 합니다.

사용자 데이터그램 프로토콜

UDP(사용자 데이터그램 프로토콜) 와 TCP를 비교하자면 , UDP는 흔히 ' *얇은 프로토콜*' 이라고 불리는 축소된 경제 모델이라고 할 수 있습니다 . 공원 벤치에 앉아 있는 마른 사람처럼, 얇은 프로토콜은 많은 공간을 차지하지 않습니다. 이 경우에는 네트워크 대역폭을 많이 사용하지 않는다는 뜻입니다.

UDP는 TCP처럼 모든 고급 기능을 제공하지는 않지만, 신뢰할 수 있는 전송이 필요하지 않은 정보를 전송하는 데 매우 효과적이며, 훨씬 적은 네트워크 자원을 사용합니다.

개발자가 TCP 대신 UDP를 선택하는 것이 현명한 경우가 분명히 있습니다. 프로세스/애플리케이션 계층에서 작동하는 감시 프로토콜인 SNMP를 기억하시나요? SNMP는 네트워크를 모니터링하고, 특히 대규모 네트워크에서 실행될 때 간헐적으로 메시지를 보내고 지속적으로 상태 업데이트와 경고를 전송합니다. 이러한 작은 메시지 하나하나에 대해 TCP 연결을 설정하고 유지하고 닫는 데 드는 오버헤드는 그렇지 않아도 효율적이고 원활한 네트워크를 순식간에 마비시킬 수 있습니다!

UDP를 TCP보다 선호하는 또 다른 경우는 프로세스/애플리케이션 계층에서 이미 안정성 문제가 해결된 경우입니다. DNS는 자체적으로 안정성 문제를 처리하므로 TCP를 사용하는 것은 비실용적이고 불필요합니다. 하지만 궁극적으로 UDP와 TCP 중 어떤 프로토콜을 사용할지는 데이터 전송 속도를 원하는 사용자가 아니라 애플리케이션 개발자가 결정해야 합니다.

UDP는 세그먼트의 순서를 *신경 쓰지 않으며*, 세그먼트가 목적지에 도착하는 순서에도 관심이 없습니다. 하지만 전송 후에는 세그먼트에 대한 후속 조치나 도착 여부 확인, 심지어는 수신 확인조차 하지 않습니다. 즉, 완전히 방치하는 방식입니다. 이러한 이유로 UDP는 *신뢰할 수 없는* 프로토콜로 불립니다. 이는 UDP가 비효율적이라는 의미가 아니라, 신뢰성 문제를 제대로 처리하지 못한다는 뜻입니다. UDP는 애플리케이션이 자체적인 신뢰성 확보 방법을 사용할 것이라고 가정하기 때문에, 자체적인 신뢰성 확보 방법을 사용하지 않습니다. 따라서 애플리케이션 개발자는 IP 스택을 실행할 때 신뢰성을 위한 TCP 또는 빠른 전송 속도를 위한 UDP 중에서 선택할 수 있습니다.

또한 UDP는 가상 회로를 생성하지 않으며, 정보를 전달하기 전에 목적지와 연결하지도 않습니다. 이러한 이유로 UDP는 비 *연결형* 프로토콜로도 간주됩니다.

그림 6.13은 TCP의 과도한 오버헤드에 비해 UDP의 오버헤드가 현저히 낮다는 것을 명확하게 보여줍니다. 그림을 자세히 살펴보세요. UDP는 윈도잉을 사용하지 않고 UDP 헤더에 확인 응답을 포함하지 않는다는 것을 알 수 있습니까?

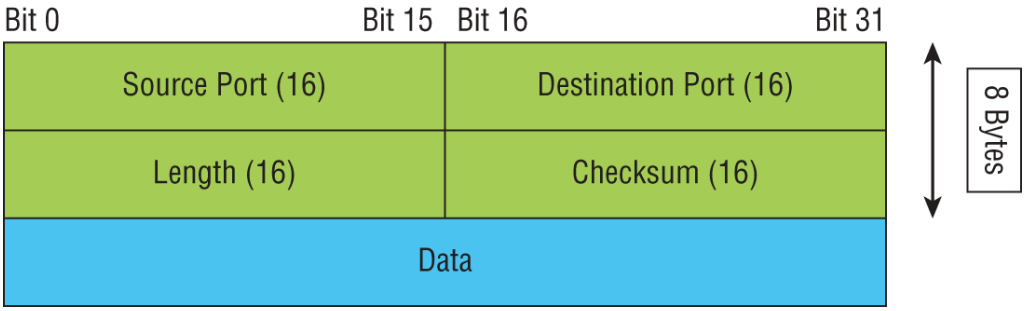

그림 6.13 UDP 세그먼트

UDP 헤더에 대한 자세한 내용은 CompTIA Network+ 시험 목표 범위를 벗어나므로, 저의 저서인 *CCNA: Cisco Certified Network Associate Study Guide(Sybex 2019)*를 참조하시기 바랍니다 .

호스트 간 프로토콜의 주요 개념

이제 연결 지향형 프로토콜(TCP)과 비연결형 프로토콜(UDP)이 실제로 어떻게 작동하는지 살펴보았으니, 두 프로토콜의 특징을 요약해 보겠습니다. 표 6.2는 이 두 프로토콜과 관련하여 기억해야 할 주요 개념들을 보여줍니다. 이 표를 암기해 두는 것이 좋습니다.

표 6.2 TCP와 UDP의 주요 특징

TCP	UDP
순차적	순서가 지정되지 않음
믿을 수 있는	신뢰할 수 없는
연결 지향적	연결 없음
가상 회로	낮은 간접비
감사의 말씀	감사 인사 없음
윈도우 흐름 제어	윈도우 함수나 흐름 제어 기능은 전혀 사용하지 않습니다.

전화 통신에 비유하면 TCP의 작동 방식을 이해하는 데 큰 도움이 될 수 있습니다. 대부분의 사람들은 전화 통화를 하기 전에 먼저 상대방과 연결을 설정해야 한다는 것을 알고 있습니다. TCP에서도 이와 유사한 가상 회로를 사용합니다. 통화 중에 중요한 정보를 전달할 때 "알고 계세요?" 또는 "이해하셨어요?"와 같은 말을 할 수 있습니다. 이러한 표현은 TCP 연결 확인과 매우 유사합니다. 즉, 상대방의 확인을 얻기 위한 것입니다. 특히 휴대전화 통화에서는 "아직 계신가요?"와 같은 질문을 하기도 합니다. 통화를 마칠 때도 "안녕히 가세요"와 같은 인사말을 건네며 통화를 마무리합니다. TCP도 이와 같은 기능을 수행합니다.

또 다른 관점에서 보면, UDP를 사용하는 것은 엽서를 보내는 것과 같습니다. 엽서를 보낼 때 상대방과 먼저 연락할 필요가 없습니다. 메시지를 작성하고 주소를 적은 다음 우편으로 보내면 됩니다. 이는 UDP의 비연결형 통신 방식과 유사합니다. 엽서에 적힌 메시지가 생사를 가르는 중요한 문제가 아니므로 수신 확인이 필요하지 않은 것처럼, UDP 역시 수신 확인 절차를 거치지 않습니다.

포트 번호

TCP와 UDP는 상위 계층과의 통신을 위해 포트 번호를 사용해야 합니다. 포트 번호는 로컬 호스트에서 시작되거나 수신되는 여러 동시 통신을 추적하는 데 사용되기 때문입니다. 발신 포트 번호는 발신 호스트에서 동적으로 할당되며 일반적으로 1024 이상의 값을 가집니다. 1023 이하의 포트는 RFC 3232에 정의되어 있으며, 이 문서에서는 이를 *잘 알려진 포트 번호(well-known port numbers)* 라고 합니다 .

잘 알려진 포트 번호를 사용하는 애플리케이션을 사용하지 않는 가상 회로에는 특정 범위에서 임의로 포트 번호가 할당됩니다. 이러한 포트 번호는 TCP 세그먼트에서 소스 및 대상 애플리케이션 또는 프로세스를 식별합니다.

그림 6.14는 TCP와 UDP 모두 포트 번호를 사용하는 방식을 보여줍니다.

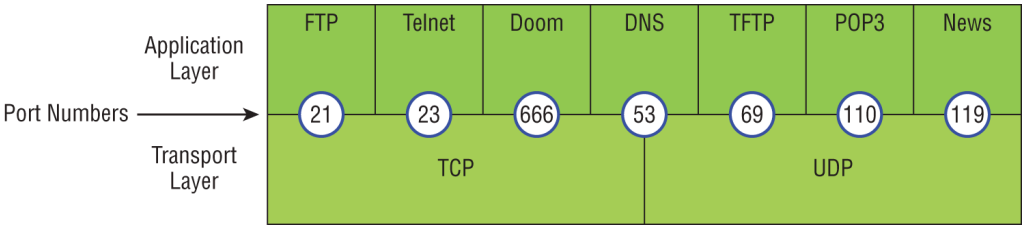

그림 6.14 TCP 및 UDP 포트 번호

1024 미만의 숫자는 잘 알려진 포트 번호로 간주되며 RFC 3232에 정의되어 있다는 점을 기억해야 합니다. 1024 이상의 숫자는 상위 계층에서 다른 호스트와의 세션을 설정하는 데 사용되며 TCP에서는 TCP 세그먼트의 출발지 및 목적지 식별자로 사용됩니다.

표 6.3은 TCP/IP 제품군에서 일반적으로 사용되는 애플리케이션, 잘 알려진 포트 번호, 그리고 각 애플리케이션 또는 프로세스에서 사용되는 전송 계층 프로토콜 목록을 제공합니다. CompTIA Network+ 시험을 위해서는 이 표를 반드시 학습하고 암기해야 합니다.

표 6.3 TCP 및 UDP를 사용하는 주요 프로토콜

TCP	UDP
텔넷 23	SNMPv1/2 161
SMTP 25	TFTP 69
HTTP 80	DNS 53
FTP 20, 21	BOOTPS/DHCP 67,68
SFTP 22	NTP 123
DNS 53	
HTTPS 443	
SSH 22	
SMB 445	
POP3 110	
IMAP4 143	
RDP 3389	
SNMPv3 161	

DNS는 TCP와 UDP 프로토콜을 모두 사용한다는 점에 유의하세요. 어떤 프로토콜을 사용할지는 수행하려는 작업에 따라 달라집니다. 두 프로토콜을 모두 사용할 수

있는 애플리케이션은 DNS뿐만이 아니지만, 학습 과정에서 반드시 기억해 두어야 할 중요한 예입니다.

인터넷 계층 프로토콜

미 국방부 모델에서 인터넷 계층이 존재하는 주요 이유는 두 가지입니다. 하나는 라우팅이고, 다른 하나는 상위 계층에 단일 네트워크 인터페이스를 제공하는 것입니다.

상위 또는 하위 계층 프로토콜 중 어느 것도 라우팅과 관련된 기능을 가지고 있지 않습니다. 이 복잡하고 중요한 작업은 전적으로 인터넷 계층의 몫입니다. 인터넷 계층의 두 번째 역할은 상위 계층 프로토콜에 단일 네트워크 인터페이스를 제공하는 것입니다. 이 계층이 없다면 애플리케이션 프로그래머는 각기 다른 네트워크 액세스 프로토콜에 맞춰 모든 애플리케이션에 소위 ' *훅*'을 작성해야 할 것입니다. 이는 매우 번거로울 뿐만 아니라, 이더넷용, 토큰링용 등 각 애플리케이션의 여러 버전을 만들어야 하는 결과를 초래할 것입니다. 이러한 문제를 방지하기 위해 IP는 단일 인터페이스를 제공합니다.상위 계층 프로토콜을 위한 네트워크 인터페이스입니다. 이것이 완료되면, IP와 다양한 네트워크 액세스 프로토콜들이 서로 호환되어 작동하는 것이 중요합니다.

모든 네트워크 경로가 로마로 통하는 것은 아닙니다. 모든 경로는 IP로 통합니다. 그리고 이 계층의 다른 모든 프로토콜뿐만 아니라 상위 계층의 모든 프로토콜도 IP를 사용합니다. 이 점을 절대 잊지 마십시오. DoD 모델을 통한 모든 경로는 IP를 거칩니다. 다음 섹션에서는 인터넷 계층의 프로토콜에 대해 설명합니다.

- 인터넷 프로토콜(IP)
- 인터넷 제어 메시지 프로토콜(ICMP)
- 주소 확인 프로토콜(ARP)
- 역방향 주소 확인 프로토콜(RARP)
- GRE/IPSec

인터넷 프로토콜

인터넷 프로토콜(IP) 은 기본적으로 인터넷 계층입니다. 여기에 있는 다른 프로토콜들은 단지 IP를 지원하기 위해 존재합니다. IP는 전체적인 그림을 보고 있으며, 상호 연결된 모든 네트워크를 인식한다는 점에서 "모든 것을 볼 수 있다"고 할 수 있습니다. 이는 네트워크상의 모든 장치가 IP 주소라고 하는 소프트웨어 또는 논리적 주소를 가지고 있기 때문에 가능합니다. IP 주소에 대해서는 다음 장에서 더 자세히 다루겠습니다.

IP는 각 패킷의 목적지 주소를 확인합니다. 그런 다음 라우팅 테이블을 사용하여 패킷이 다음에 어디로 전송될지 결정하고 최적의 경로를 선택합니다. 국방부 모델의

최하단에 있는 네트워크 액세스 계층 프로토콜은 IP처럼 전체 네트워크를 포괄하는 광범위한 범위를 가지고 있지 않으며, 물리적 링크(로컬 네트워크)만 처리합니다.

네트워크에서 장치를 식별하려면 다음 두 가지 질문에 답해야 합니다. 해당 장치가 어떤 네트워크에 속해 있는가? 그리고 해당 네트워크에서 장치의 ID는 무엇인가? 첫 번째 질문에 대한 답은 *소프트웨어 주소*, 즉 *논리적 주소* (정확한 도로명 주소)입니다. 두 번째 질문에 대한 답은 *하드웨어 주소* (정확한 사서함 주소)입니다. 네트워크의 모든 호스트는 IP 주소라는 논리적 ID를 가지고 있습니다. 이 IP 주소는 소프트웨어 주소, 즉 논리적 주소이며, 암호화된 중요한 정보를 포함하고 있어 복잡한 라우팅 작업을 크게 단순화합니다. (IP에 대한 자세한 내용은 RFC 791을 참조하십시오.)

IP는 호스트 간 계층에서 세그먼트를 수신하고 필요한 경우 이를 패킷으로 분할합니다. 그런 다음 수신 측에서 패킷을 다시 세그먼트로 재조립합니다. 각 패킷에는 송신자와 수신자의 IP 주소가 할당됩니다. 패킷을 수신하는 각 라우터(레이어 3 장치)는 패킷의 목적지 IP 주소를 기반으로 라우팅 결정을 내립니다.

그림 6.15는 IPv4 헤더를 보여줍니다. 이 그림을 통해 상위 계층에서 원격 네트워크로 사용자 데이터가 전송될 때마다 IP가 거쳐야 하는 과정을 이해할 수 있습니다.

IP 헤더는 다음 필드들로 구성됩니다.

- **버전** IP 버전 번호.
- **헤더** 길이(HLEN)는 32비트 워드 단위입니다.
- **우선순위 및 서비스 유형** 서비스 유형은 데이터그램을 어떻게 처리해야 하는지를 나타냅니다. 처음 3비트는 우선순위 비트이며, 현재는 차별화 서비스 비트라고 부릅니다.
- **전체 길이:** 헤더와 데이터를 포함한 패킷의 길이입니다.

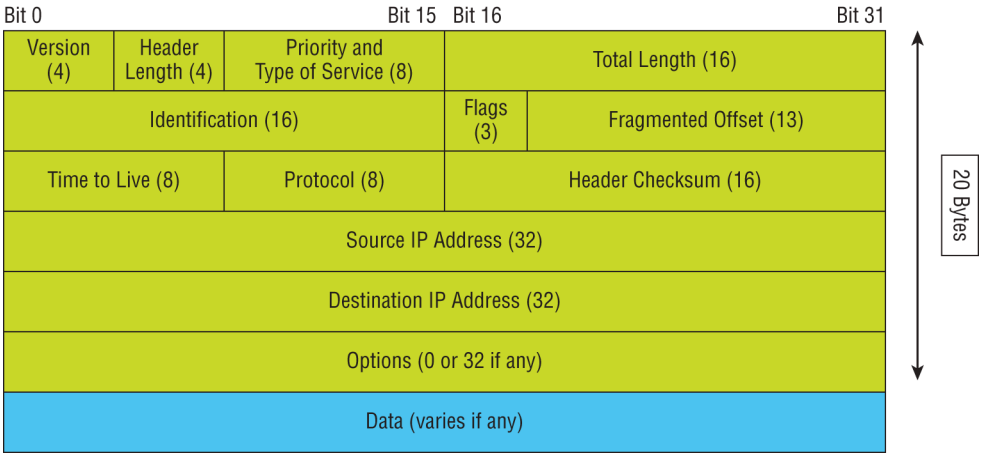

그림 6.15 IPv4 헤더

- **식별 번호는** 서로 다른 데이터그램에서 생성된 조각난 패킷을 구분하는 데 사용되는 고유한 IP 패킷 값입니다.
- **플래그** 이 필드는 패킷 분할이 발생해야 하는지 여부를 지정합니다.
- **단편화 오프셋은** 패킷 크기가 프레임에 담기에는 너무 큰 경우 패킷을 단편화하고 재조립하는 기능을 제공합니다. 또한 인터넷에서 서로 다른 최대 전송 단위(MTU, 패킷 크기를 정의함)를 사용할 수 있도록 합니다.
- **TTL(Time To** Live)은 패킷이 생성될 때 설정되는 시간입니다. 만약 패킷이 TTL이 만료되기 전에 목적지에 도달하지 못하면, 패킷은 사라집니다. 이는 IP 패킷이 목적지를 찾아 네트워크를 계속해서 맴도는 것을 방지합니다.
- **프로토콜** 포트는 상위 계층 프로토콜의 포트입니다. 예를 들어 TCP는 6번 포트, UDP는 17번 포트입니다. ARP, ICMP와 같은 네트워크 계층 프로토콜도 지원하며, 일부 분석 도구에서는 '유형' 필드로 표시되기도 합니다. 이 필드에 대해서는 잠시 후에 자세히 설명하겠습니다.
- **헤더 체크섬** 순환 중복 검사(CRC)는 헤더에만 적용됩니다.
- **발신국 IP 주소는** 송신 스테이션의 32비트 IP 주소입니다.
- **목적지 IP 주소:** 이 패킷이 전송될 스테이션의 32비트 IP 주소입니다.
- 네트워크 테스트, 디버깅, 보안 등에 사용되는 **옵션입니다 .**
- **IP 옵션 필드 다음에 오는 데이터는** 상위 계층 데이터가 됩니다.

다음은 네트워크 분석기로 포착한 IP 패킷의 스냅샷입니다. 앞서 설명한 모든 헤더 정보가 여기에 나타나는 것을 확인할 수 있습니다.

```
IP Header - Internet Protocol Datagram
Version: 4
Header Length: 5
Precedence: 0
Type of Service: %000
Unused: %00
Total Length: 187
Identifier: 22486
Fragmentation Flags: %010 Do Not Fragment
Fragment Offset: 0
```

```
Time To Live: 60
IP Type: 0x06 TCP
Header Checksum: 0xd031
Source IP Address: 10.7.1.30
Dest. IP Address: 10.7.1.10
No Internet Datagram Options
```

Type 필드는 일반적으로 Protocol 필드이지만, 이 분석기는 이를 IP Type 필드로 인식합니다. 이는 중요한 점입니다. 헤더에 다음 계층에 대한 프로토콜 정보가 없으면 IP는 패킷에 담긴 데이터를 어떻게 처리해야 할지 알 수 없습니다. 앞의 예시는 IP에게 해당 세그먼트를 TCP에 전달하도록 명확하게 지시하고 있습니다.

그림 6.16은 네트워크 계층이 상위 계층 프로토콜에 패킷을 전달해야 할 때 전송 계층의 프로토콜을 어떻게 인식하는지를 보여줍니다.

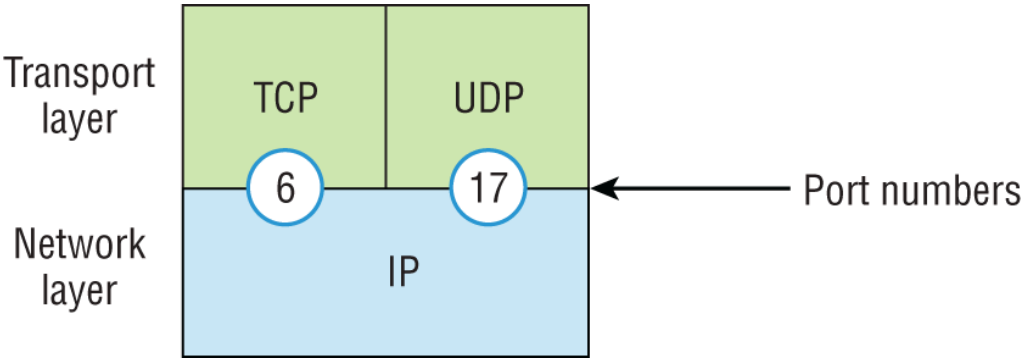

그림 6.16 IP 헤더의 프로토콜 필드

이 예시에서 Protocol 필드는 IP에게 데이터를 TCP 포트 6 또는 UDP 포트 17로 보내도록 지시합니다. 하지만 데이터가 상위 계층 서비스나 애플리케이션으로 향하는 데이터 스트림의 일부인 경우에만 UDP 또는 TCP를 사용하게 됩니다. 인터넷 제어 메시지 프로토콜(ICMP), 주소 확인 프로토콜(ARP) 또는 다른 네트워크 계층 프로토콜을 사용할 수도 있습니다.

표 6.4 는 프로토콜 필드에 지정할 수 있는 몇 가지 인기 있는 프로토콜 목록입니다.

표 6.4 IP 헤더의 Protocol 필드에서 발견될 수 있는 프로토콜

규약	프로토콜 번호
ICMP	1
IP in IP (터널링)	4
TCP	6
UDP	17
EIGRP	88
OSPF	89
IPv6	41
GRE	47
레이어 2 터널(L2TP)	115

프로토콜 필드 번호의 전체 목록은 다음에서 확인할 수 있습니다 https://www.iana.org/assignments/protocol-numbers.

인터넷 제어 메시지 프로토콜

*인터넷 제어 메시지 프로토콜(ICMP)*은 네트워크 계층에서 작동하며 IP에서 다양한 서비스에 사용됩니다. ICMP는 IP의 관리 프로토콜이자 메시징 서비스 제공자입니다. ICMP 메시지는 IP 패킷으로 전송됩니다.

ICMP 패킷은 다음과 같은 특징을 가지고 있습니다.

- 이들은 호스트에게 네트워크 문제에 대한 정보를 제공할 수 있습니다.
- 이러한 정보는 IP 데이터그램 내에 캡슐화되어 있습니다.

다음은 ICMP와 관련된 몇 가지 일반적인 이벤트 및 메시지와 ICMP를 사용하는 가장 인기 있는 두 가지 프로그램입니다.

- **목적지에 도달할 수 없음** 라우터가 IP 데이터그램을 더 이상 보낼 수 없는 경우, ICMP를 사용하여 발신자에게 상황을 알리는 메시지를 보냅니다. 예를 들어, 그림 6.17을 보면 실험 B 라우터의 이더넷 인터페이스가 다운된 것을 알 수 있습니다.

  e0 on Lab B is down. Host A is trying to communicate to Host B. What happens?

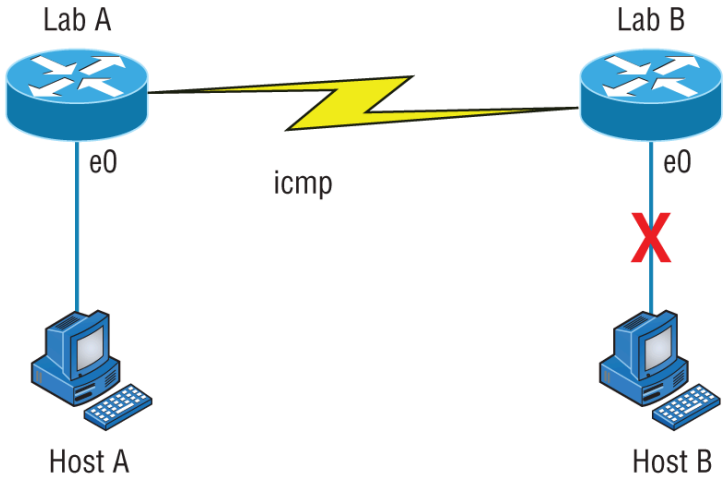

  그림 6.17은 원격 라우터에서 송신 호스트로 ICMP 오류 메시지를 전송하는 모습을 보여줍니다.

  호스트 A가 호스트 B로 향하는 패킷을 전송하면, 랩 B의 라우터는 전송 장치(이 예에서는 호스트 A)로 ICMP 목적지 도달 불가 메시지를 다시 보냅니다.
- **버퍼 가득 참:** 라우터의 수신 데이터그램을 위한 메모리 버퍼가 가득 차면, 혼잡이 해소될 때까지 ICMP를 사용하여 이 메시지를 전송합니다.
- **홉(Hops)이란** *각* IP 데이터그램이 거쳐야 하는 특정 개수의 라우터를 의미합니다 . 데이터그램이 목적지에 도달하기 전에 지정된 홉 수에 도달하면, 마지막으로 데이터그램을 수신한 라우터가 해당 데이터그램을 삭제합니다. 그런 다음, 삭제 라우터는 ICMP를 사용하여 발신 라우터에 데이터그램의 소멸을 알리는 부고 메시지를 보냅니다.
- **Ping** 은 ICMP 에코 요청 및 응답 메시지를 사용하여 인터넷 네트워크 상의 기기 간 물리적 및 논리적 연결 상태를 확인합니다.
- **트레이서** 루트는 IP 패킷의 시간 제한(Time to Live)을 이용하여 패킷이 인터넷 네트워크를 통과할 때 거치는 경로를 파악합니다.

Ping과 Traceroute(Trace라고도 하며 Microsoft Windows에서 사용됨 `tracert`)는 모두 인터넷 네트워크의 주소 구성을 확인할 수 있도록 해줍니다.

주소 확인 프로토콜

*주소 확인 프로토콜(ARP)*은 알려진 IP 주소로부터 호스트의 하드웨어 주소를 찾습니다. 작동 방식은 다음과 같습니다. IP는 전송할 데이터그램이 있을 때, 이더넷이나 토큰링과 같은 네트워크 액세스 프로토콜에 목적지의 하드웨어 주소를 알려야 합니다. (상위 계층 프로토콜로부터 목적지의 IP 주소는 이미 전달받은 상태입니다.) IP가 ARP 캐시에서 목적지 호스트의 하드웨어 주소를 찾지 못하면, ARP를 사용하여 해당 정보를 찾습니다.

IP의 탐정 역할을 하는 ARP는 지정된 IP 주소를 가진 컴퓨터에 하드웨어 주소를 회신해 달라는 브로드캐스트를 전송하여 로컬 네트워크를 탐색합니다. 즉, ARP는 소프트웨어(IP) 주소를 하드웨어 주소, 예를 들어 대상 컴퓨터의 이더넷 주소로 변환합니다. 그림 6.18은 로컬 네트워크에서 ARP 브로드캐스트가 어떻게 보이는지 보여줍니다.

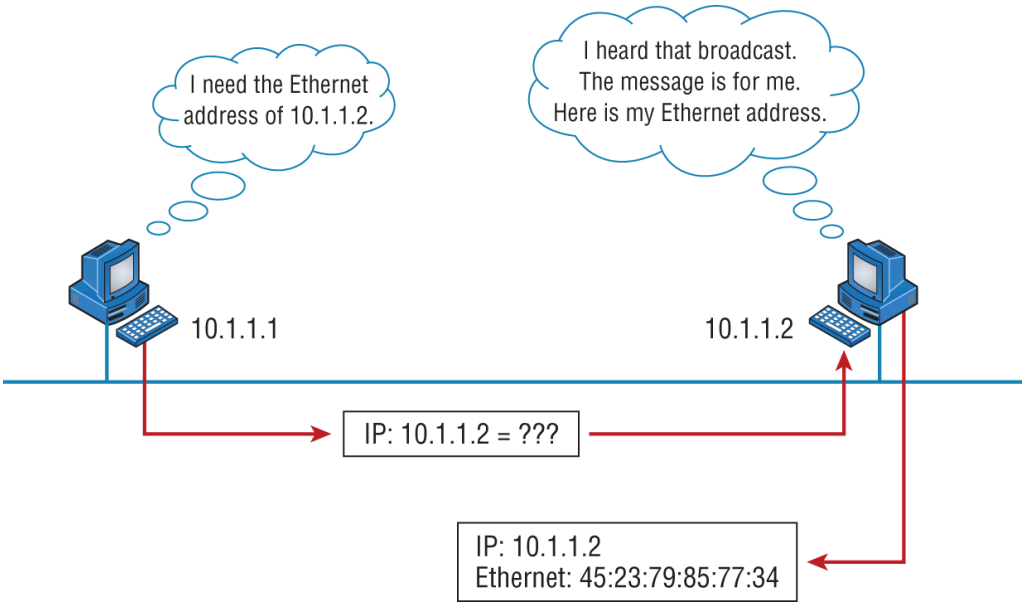

그림 6.18 로컬 ARP 방송

ARP는 IP 주소를 이더넷(MAC) 주소로 변환합니다.

다음 추적 결과는 ARP 브로드캐스트를 보여줍니다. 대상 하드웨어 주소가 알 수 없으며 ARP 헤더의 모든 문자가 0으로 되어 있는 것을 확인할 수 있습니다. 이더넷 헤더에서 16진수로 모두 *F* (2진수로는 모두 1)로 표현되는 대상 주소는 하드웨어 주소 브로드캐스트에 사용되며, 이는 로컬 링크의 모든 장치가 ARP 요청을 수신하도록 하기 위한 것입니다.

```
 Flags:          0x00
 Status:         0x00
 Packet Length: 64
 Timestamp:      09:17:29.574000 12/06/21
Ethernet Header
 Destination:    FF:FF:FF:FF:FF:FF Ethernet Broadcast
 Source:         00:A0:24:48:60:A5
 Protocol Type: 0x0806 IP ARP
ARP - Address Resolution Protocol
 Hardware:                 1 Ethernet (10Mb)

 Protocol:                 0x0800 IP
 Hardware Address Length: 6
 Protocol Address Length: 4
 Operation:                1 ARP Request
 Sender Hardware Address: 00:A0:24:48:60:A5
 Sender Internet Address: 172.16.10.3
 Target Hardware Address: 00:00:00:00:00:00 (ignored)
 Target Internet Address: 172.16.10.10
Extra bytes (Padding):
 ……. 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
  0A 0A 0A 0A 0A
Frame Check Sequence: 0x00000000
```

역방향 주소 확인 프로토콜(RARP)

디스크가 없는 IP 기기는 처음에는 자신의 IP 주소를 알 수 없습니다. 하지만 MAC 주소는 알고 있습니다. *역방향 주소 확인 프로토콜(RARP)*은 디스크가 없는 기기의 IP 주소를 알아내기 위해 MAC 주소와 해당 MAC 주소에 할당된 IP 주소를 요청하는 패킷을 전송합니다. *RARP 서버* 라고 불리는 지정된 기기가 응답을 보내면 기기 식별 문제가 해결됩니다. RARP는 기기의 MAC 주소에 대한 정보를 활용하여 IP 주소를 알아내고 기기의 신원을 완성합니다.

그림 6.19는 디스크가 없는 워크스테이션이 RARP 브로드캐스트를 통해 자신의 IP 주소를 요청하는 모습을 보여줍니다.

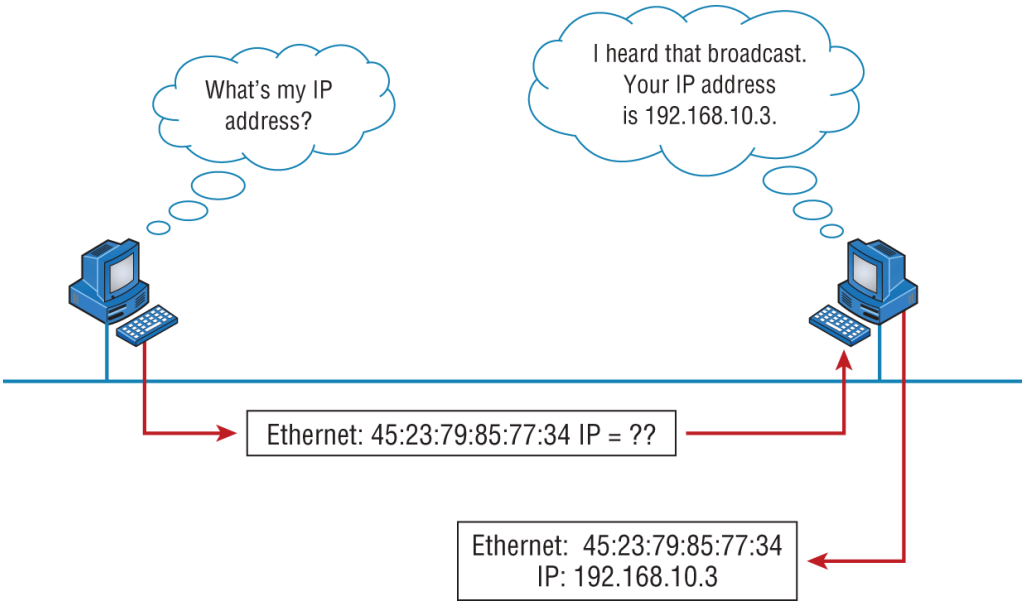

그림 6.19 RARP 방송 예시

일반 라우팅 캡슐화(GRE)

GRE(Generic Routing Encapsulation) 는 IP 터널 내부에 다양한 프로토콜을 캡슐화할 수 있는 터널링 프로토콜입니다. 예를 들어 EIGRP, OSPF와 같은 라우팅 프로토콜과 IPv6 라우팅 프로토콜이 있습니다. 그림 6.20은 GRE 헤더의 각 구성 요소를 보여줍니다.

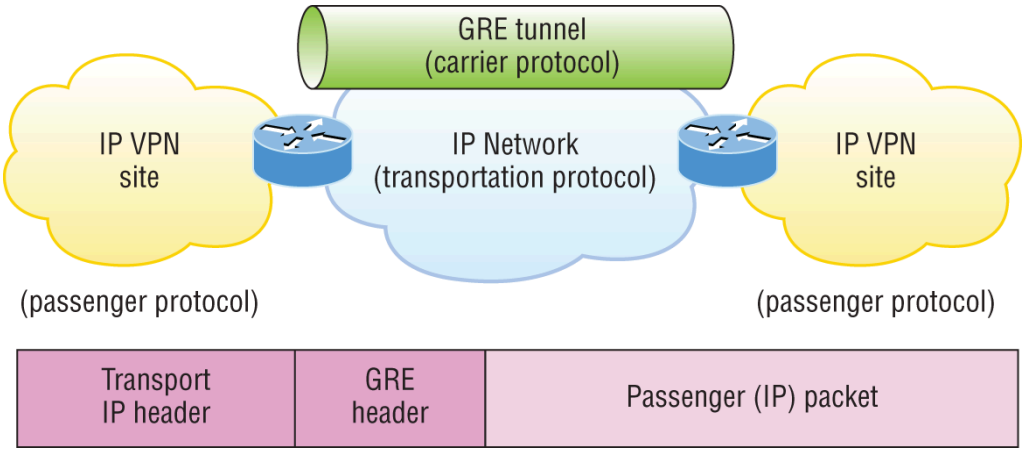

그림 6.20 일반 라우팅 캡슐화(GRE) 터널 구조

GRE 터널 인터페이스는 다음 각각에 대한 헤더를 지원합니다.

- 승객 프로토콜 또는 IP나 IPv6와 같은 캡슐화된 프로토콜은 GRE에 의해 캡슐화되는 프로토콜입니다.
- GRE 캡슐화 프로토콜
- 전송 전달 프로토콜(일반적으로 IP)

GRE 터널은 다음과 같은 특징을 가지고 있습니다.

- GRE는 GRE 헤더에 프로토콜 유형 필드를 사용하므로 모든 레이어 3 프로토콜을 터널을 통해 사용할 수 있습니다.
- GRE는 상태를 저장하지 않으며 흐름 제어를 하지 않습니다.
- GRE는 보안을 제공하지 않습니다.
- GRE는 터널링된 패킷에 대해 최소 24바이트의 추가 오버헤드를 발생시킵니다.

인터넷 프로토콜 보안(IPSec)

앞서 언급했듯이 GRE 자체는 보안을 제공하지 않습니다. 즉, 페이로드의 기밀성이나 암호화 기능을 제공하지 않습니다. 공용 네트워크에서 패킷을 도청하면 내용이 평문으로 노출되며, IPsec은 IP 네트워크를 통해 데이터를 안전하게 터널링하는 방법을 제공하지만 한계가 있습니다.

IPSec은 IP 브로드캐스트나 IP 멀티캐스트를 지원하지 않기 때문에 라우팅 프로토콜처럼 이를 필요로 하는 프로토콜을 사용할 수 없습니다. 또한 IPSec은 멀티프로토콜 트래픽도 지원하지 않습니다. GRE는 IP 브로드캐스트나 IP 멀티캐스트와 같은 다른 프로토콜은 물론 비IP 프로토콜까지 "전송"할 수 있는 프로토콜입니다. 따라서 IPSec과 함께 GRE 터널을 사용하면 여러 프로토콜을 동시에 사용할 수 있습니다.이 기능을 사용하면 라우팅 프로토콜, IP 멀티캐스트 및 멀티프로토콜 트래픽을 네트워크 전체에서 실행할 수 있습니다.

일반적인 허브 앤 스포크 토폴로지(예: 본사-지점)에서는 본사와 지사 간에 GRE over IPSec과 같은 정적 터널을 구현할 수 있습니다. 네트워크에 새로운 스포크를 추가하려면 허브 라우터에서 해당 스포크를 구성하기만 하면 됩니다. 스포크 간의 트래픽은 허브를 거쳐야 하며, 허브에서 하나의 터널을 빠져나와 다른 터널로 들어가야 합니다. 정적 터널은 소규모 네트워크에는 적합한 솔루션일 수 있지만, 스포크 수가 점점 많아질수록 실제로는 허용할 수 없는 문제가 됩니다.

인증 헤더(AH)/캡슐화 보안 페이로드(ESP)

IPSec에서 사용하는 두 가지 주요 보안 프로토콜은 *인증 헤더(AH)* 와 *캡슐화 보안 페이로드(ESP)* 입니다 .

인증 헤더(AH)

AH 프로토콜은 패킷 인증을 위해 단방향 해시를 사용하여 패킷의 데이터와 IP 헤더에 대한 인증을 제공합니다. 작동 방식은 다음과 같습니다. 송신자가 단방향 해시를 생성하면 수신자도 동일한 단방향 해시를 생성합니다. 패킷에 어떤 변경 사항이든 있으면 인증되지 않고 폐기됩니다. 따라서 IPsec은 AH를 통해 인증을 보장합니다. AH는 패킷 전체를 검사하지만 암호화 서비스는 제공하지 않습니다.

이는 패킷 데이터의 무결성 검사만 제공하는 ESP와는 다릅니다.

캡슐화된 보안 페이로드(ESP)

ESP는 나스닥 지수가 슈퍼볼처럼 언제 어떻게 오르락내리락할지 알려주지는 않지만, 기밀성, 데이터 출처 인증, 비연결형 무결성, 재전송 방지 서비스, 그리고 트래픽 흐름 분석을 무력화하여 제한적인 트래픽 흐름 기밀성을 제공합니다. 이는 거의 그에 못지않은 효과입니다! 어쨌든 ESP는 다섯 가지 구성 요소로 이루어져 있습니다.

- **기밀성(암호화)** 은 송신 장치가 도청을 방지하기 위해 전송 전에 패킷을 암호화할 수 있도록 합니다. 기밀성은 대칭 암호화 알고리즘을 사용하여 제공됩니다. 기밀성은 다른 모든 서비스와 별도로 선택할 수 있지만, 선택한 기밀성 수준은 VPN의 양쪽 끝점에서 동일해야 합니다. IPSec에서 사용하도록 정의된 암호화 알고리즘은 다음과 같습니다.
  - 무결성 및 진위성 보호를 위한 HMAC-SHA1/SHA2
  - 기밀 유지를 위한 TripleDES-CBC
  - AES-CBC 및 기밀 유지를 위한 AES-CBC
  - AES-GCM과 ChaCha20-Poly1305를 결합하여 기밀성과 인증을 효율적으로 제공합니다.
- **데이터 무결성은** 수신자가 수신된 데이터가 전송 과정에서 어떠한 방식으로도 변경되지 않았는지 확인할 수 있도록 합니다. IPSec은 체크섬을 사용하여 데이터를 간단하게 검사합니다.
- **인증** 은 연결이 올바른 파트너와 이루어졌는지 확인하는 과정입니다. 수신자는 정보의 출처를 보장하고 인증함으로써 패킷의 출처를 인증할 수 있습니다.
- **재전송 방지 서비스는** 수신자에 따라 결정되므로, 수신자가 시퀀스 번호를 확인하는 경우에만 서비스가 유효합니다. 재전송 공격이란 해커가 인증된 패킷의 복사본을 만들어 원래 목적지로 전송하는 공격을 말합니다. 복제된 인증된 IP 패킷이 목적지에 도달하면 서비스 중단 및 전반적인 시스템 장애를 초래할 수 있습니다. 시퀀스 번호 필드는 *이러한* 유형의 공격을 차단하도록 설계되었습니다.
- **트래픽 흐름** 기밀성을 확보하려면 최소한 터널 모드를 선택해야 합니다. 이 기능은 트래픽이 집중되는 보안 게이트웨이에 구현할 때 가장 효과적입니다. 이러한 환경은 네트워크 보안을 침해하려는 악의적인 공격자에게 실제 출발지-도착지 패턴을 숨길 수 있기 때문입니다.

데이터 캡슐화

2장 에서 데이터 캡슐화에 대해 논의를 시작했지만 , 당시에는 가상 회로에서 포트가 어떻게 작동하는지에 대한 확실한 이해가 필요했기 때문에 개괄적인 내용만 다룰 수 있었습니다. 하지만 지난 5개 장에 걸쳐 기초적인 내용을 충분히 숙지하셨다면, 이제 캡슐화의 세부적인 내용으로 넘어갈 준비가 되셨습니다.

호스트가 네트워크를 통해 다른 장치로 데이터를 전송할 때, 데이터는 *캡슐화 과정*을 거칩니다 . 즉, OSI 모델의 각 계층에서 프로토콜 정보로 감싸집니다. 각 계층은 수신 장치의 해당 계층과만 통신합니다.

통신 및 정보 교환을 위해 각 계층은 *프로토콜 데이터 단위(PDU)*를 사용합니다 . PDU는 모델의 각 계층에서 데이터에 첨부된 제어 정보를 담고 있습니다. 일반적으로 데이터 필드 앞의 헤더에 첨부되지만, 데이터 필드의 끝부분(트레일러)에 위치할 수도 있습니다.

각 PDU는 OSI 모델의 각 계층에서 데이터를 캡슐화하여 데이터에 연결되며, 각 PDU는 헤더에 제공된 정보에 따라 고유한 이름을 갖습니다. 이 PDU 정보는 수신 장치의 피어 계층에서만 읽습니다. 읽은 후에는 헤더가 제거되고 데이터가 다음 계층으로 전달됩니다.

그림 6.21은 PDU와 각 계층에 제어 정보를 첨부하는 방식을 보여줍니다. 이 그림은 상위 계층의 사용자 데이터가 네트워크를 통해 전송될 수 있도록 어떻게 변환되는지를 보여줍니다.

데이터 스트림은 전송 계층으로 전달되고, 전송 계층은 동기 패킷을 전송하여 수신 장치와의 가상 회로를 설정합니다. 다음으로, 데이터 스트림은 더 작은 조각으로 분할되고, 전송 계층 헤더(PDU)가 생성되어 데이터 필드의 헤더에 추가됩니다. 이제 이 데이터 조각을 *세그먼트*라고 합니다 . 각 세그먼트는 순서대로 배열되어 수신 측에서 데이터 스트림을 전송된 순서대로 정확하게 다시 조합할 수 있도록 합니다.

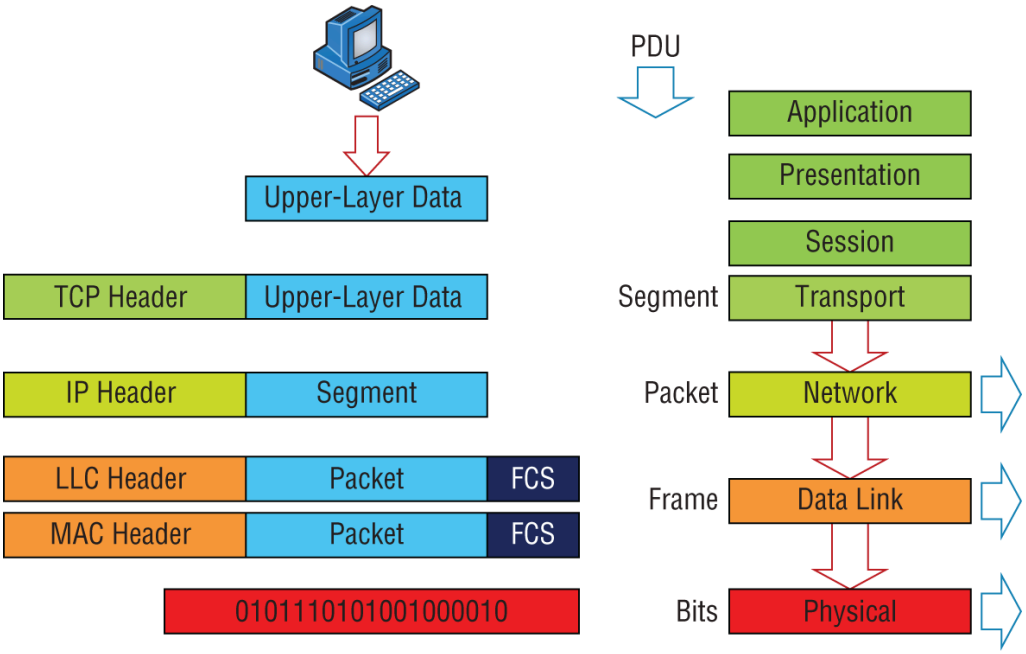

그림 6.21 데이터 캡슐화

각 세그먼트는 네트워크 계층으로 전달되어 네트워크 주소 지정 및 인터넷워크를 통한 라우팅을 거칩니다. 논리적 주소 지정(예: IP)은 각 세그먼트를 올바른 네트워크로 전달하는 데 사용됩니다. 네트워크 계층 프로토콜은 전송 계층에서 전달된 세

그먼트에 제어 헤더를 추가하며, 이렇게 만들어진 것을 *패킷* 또는 *데이터그램* 이라고 합니다 . 전송 계층과 네트워크 계층은 수신 호스트에서 데이터 스트림을 재구성하기 위해 협력하지만, PDU를 로컬 네트워크 세그먼트에 배치하는 것은 이들의 역할이 아닙니다. 라우터나 호스트에 정보를 전달하는 유일한 방법은 바로 이 로컬 네트워크 세그먼트를 통해서입니다.

데이터 링크 계층은 네트워크 계층에서 패킷을 받아 네트워크 매체(케이블 또는 무선)에 전송하는 역할을 합니다. 데이터 링크 계층은 각 패킷을 프레임으로 캡슐화하며 , 프레임 헤더에는 출발지와 목적지 호스트의 하드웨어 주소가 포함됩니다. 목적지 장치가 원격 네트워크에 있는 경우, 프레임은 라우터로 전송되어 인터넷워크를 통해 라우팅됩니다. 목적지 네트워크에 도달하면 새로운 프레임을 사용하여 패킷을 목적지 호스트로 전달합니다.

이 프레임을 네트워크에 전송하려면 먼저 디지털 신호로 변환해야 합니다. 프레임은 실제로 1과 0의 논리적 그룹이므로 물리 계층은 이러한 숫자를 디지털 신호로 인코딩하는 역할을 담당하며, 이 디지털 신호는 동일한 로컬 네트워크 상의 장치에서 읽힙니다. 수신 장치는 디지털 신호를 동기화하고 디지털 신호에서 1과 0을 추출(디코딩)합니다. 이 시점에서 장치는 프레임을 구성하고 순환 중복 검사(CRC)를 실행한 다음 프레임의 프레임 검사 시퀀스(FCS) 필드에 있는 응답과 자신의 응답을 비교합니다. 응답이 일치하면 패킷을 프레임에서 분리하고 프레임의 나머지 부분은 버립니다. 이 과정을 *캡슐화 해제* 라고 합니다 . 패킷은 네트워크 계층으로 전달되어 주소를 확인합니다. 주소가 일치하면 해당 세그먼트를 패킷에서 분리하고 패킷의 나머지 부분은 버립니다. 세그먼트는 전송 계층에서 처리되어 데이터 스트림을 재구성하고 송신국에 각 조각을 수신했음을 확인합니다. 그런 다음 해당 데이터 스트림을 상위 계층 애플리케이션에 기꺼이 전달합니다.

요약하자면, 송신 장치에서 데이터 캡슐화 방식은 다음과 같이 작동합니다.

1. 사용자 정보는 네트워크 전송을 위한 데이터로 변환됩니다.
2. 데이터는 세그먼트로 변환되고, 송신 호스트와 수신 호스트 간에 안정적인 연결이 설정됩니다.
3. 세그먼트는 패킷 또는 데이터그램으로 변환되고, 각 패킷이 인터넷워크를 통해 라우팅될 수 있도록 헤더에 논리적 주소가 추가됩니다.
4. 패킷 또는 데이터그램은 로컬 네트워크에서 전송하기 위해 프레임으로 변환됩니다. 하드웨어(이더넷) 주소는 로컬 네트워크 세그먼트에서 호스트를 고유하게 식별하는 데 사용됩니다.
5. 프레임은 비트로 변환되고, 디지털 인코딩 및 클럭킹 방식이 사용됩니다.

레이어 주소 지정을 사용하여 이를 더 자세히 설명하기 위해 그림 6.22를 사용하겠습니다 .

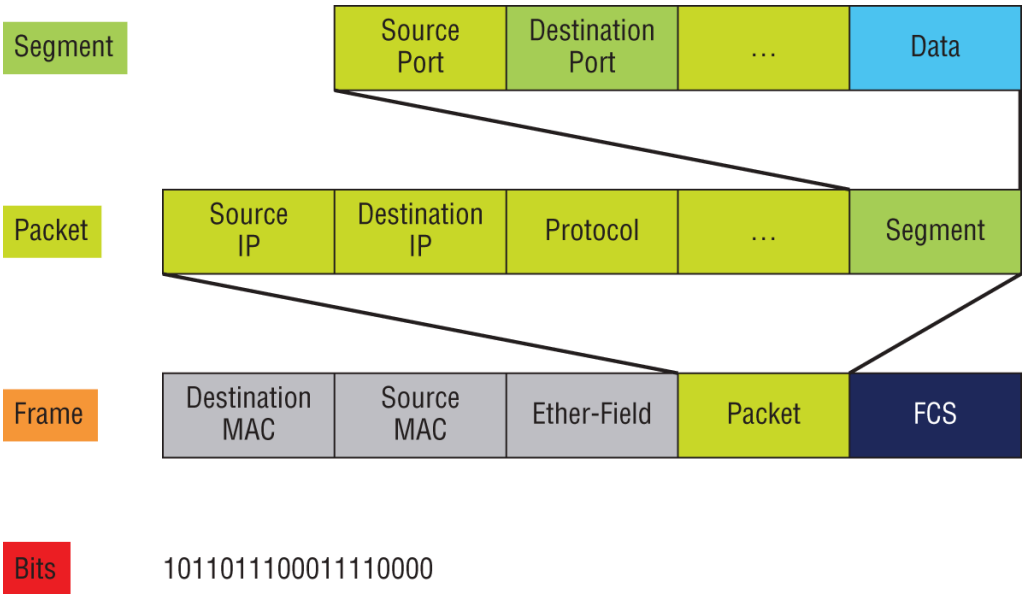

그림 6.22 PDU 및 계층 주소 지정

데이터 스트림은 상위 계층에서 전송 계층으로 전달된다는 점을 기억하세요. 기술자로서 우리는 데이터 스트림의 출처가 어디인지에는 관심이 없습니다. 그건 프로그래머의 문제이기 때문입니다. 우리의 임무는 데이터 스트림을 안정적으로 재구성하여 수신 장치의 상위 계층에 전달하는 것입니다.

그림 6.22 에 대한 논의를 더 진행하기 전에 포트 번호를 복습하고 제대로 이해하고 있는지 확인해 보겠습니다. 그림 6.23 에서 볼 수 있듯이 전송 계층은 포트 번호를 사용하여 가상 회로와 상위 계층 프로세스를 모두 정의합니다 .

전송 계층은 데이터 스트림을 받아 세그먼트로 나누고 가상 회로를 생성하여 안정적인 세션을 설정합니다. 그런 다음 각 세그먼트에 순서를 부여(번호를 매김)하고 확인 응답과 흐름 제어를 사용합니다. TCP를 사용하는 경우 가상 회로는 소스 포트 번호로 정의됩니다. 호스트는 1024부터 임의로 포트 번호를 할당합니다(0부터 1023까지는 잘 알려진 포트 번호로 예약되어 있습니다). 대상 포트 번호는 수신 호스트에서 데이터 스트림이 안정적으로 재구성될 때 해당 데이터 스트림을 전달할 상위 계층 프로세스(애플리케이션)를 정의합니다.

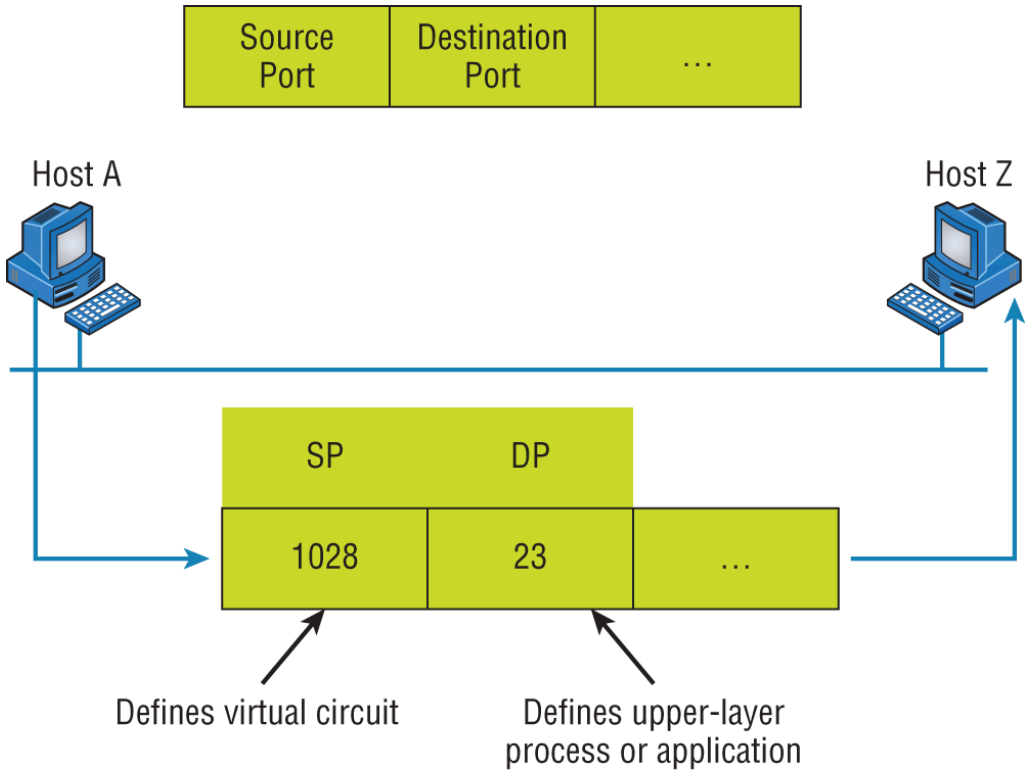

그림 6.23 전송 계층의 포트 번호

이제 포트 번호와 전송 계층에서의 사용 방식을 이해했으니 그림 6.22 로 돌아가 보겠습니다 . 전송 계층 헤더 정보가 데이터에 추가되면 해당 데이터는 세그먼트가 되어 목적지 IP 주소와 함께 네트워크 계층으로 전달됩니다. (목적지 IP 주소는 상위 계층에서 데이터 스트림과 함께 전송 계층으로 전달되었으며, 상위 계층에서 이름 확인 방법(아마도 DNS)을 통해 발견되었습니다.)

네트워크 계층은 각 세그먼트 앞에 헤더와 논리적 주소(IP 주소)를 추가합니다. 헤더가 추가되면 PDU를 *패킷* 이라고 합니다 . 패킷에는 세그먼트가 어디에서 왔는지(UDP 또는 TCP)를 설명하는 프로토콜 필드가 있어, 수신 호스트에 도달했을 때 전송 계층에서 해당 세그먼트를 올바른 프로토콜로 전달할 수 있습니다.

네트워크 계층은 패킷이 로컬 네트워크에서 어디로 전송되어야 하는지를 결정하는 목적지 하드웨어 주소를 찾는 역할을 합니다. 이를 위해 ARP(Adaptive Request Proposal)를 사용합니다. 네트워크 계층의 IP는 목적지 IP 주소를 확인하고 이를 자신의 출발지 IP 주소 및 서브넷 마스크와 비교합니다. 로컬 네트워크 요청인 경우, ARP 요청을 통해 로컬 호스트의 하드웨어 주소를 요청합니다. 패킷이 원격 호스트로 향하는 경우, IP는 구성 정보에서 기본 게이트웨이의 IP 주소를 찾은 다음, 기본 게이트웨이(라우터)의 하드웨어 주소를 ARP를 통해 요청합니다.

패킷은 로컬 호스트 또는 기본 게이트웨이의 목적지 하드웨어 주소와 함께 데이터 링크 계층으로 전달됩니다. 데이터 링크 계층은 패킷 앞에 헤더를 추가하고, 이렇게 해서 데이터는 프레임이 됩니다 . (헤더와 트레일러가 패킷에 추가되어 마치 책의 양 끝이나 액자처럼 보이기 때문에 이를 프레임이라고 부릅니다.) 그림 6.22 에 나와 있습니다 . 프레임은 Ether-Type 필드를 사용하여 네트워크 계층에서 어떤 프로

토콜을 통해 패킷이 전송되었는지 설명합니다. 이제 프레임에 대해 CRC 검사가 수행되고, CRC 결과는 프레임 트레일러에 있는 FCS 필드에 저장됩니다.

이제 프레임은 비트 단위로 물리 계층으로 전달될 준비가 되었습니다. 물리 계층은 비트 타이밍 규칙을 사용하여 데이터를 디지털 신호로 인코딩합니다. 네트워크 세그먼트의 모든 장치는 클록과 동기화하고 디지털 신호에서 1과 0을 추출하여 프레임을 구성합니다. 프레임이 재구성되면 CRC 검사를 실행하여 프레임의 유효성을 확인합니다. 모든 것이 정상적이면 호스트는 목적지 주소를 확인하여 프레임이 자신에게 온 것인지 확인합니다.

이 모든 내용이 너무 복잡하고 머리가 아프시더라도 걱정하지 마세요. 책을 읽어 나가다 보면 훨씬 더 명확해질 겁니다. 정말이에요! 곧 9장 "IP 라우팅 소개"에서 데이터가 어떻게 캡슐화되고 인터넷 네트워크를 통해 라우팅되는지 아주 자세하고 이해하기 쉬운 단계별 설명으로 다룰 예정입니다.

요약

프로토콜, 정말 어디에나 프로토콜이 있습니다. 프로토콜이 존재하는 이유는 다양하고, 우리에게 제공하는 역할도 정말 많습니다! 게다가 때로는 서로 연동해서 작동하기도 하죠. 너무 많은 정보처럼 느껴질 수도 있지만, 걱정 마세요. 다양한 계층과 그 기능을 이해하게 되면, 이 계층 구조가 얼마나 견고하고 탄탄한 네트워킹 기반인지 금방 알게 될 겁니다.

마찬가지로, TCP/IP의 전체적인 그림을 이해하게 되면 이러한 프로토콜들이 존재하고 필요한 이유도 훨씬 쉽게 이해할 수 있게 될 것입니다. 이 프로토콜들은 마치 한 팀처럼 계층별로 협력하여 TCP/IP 네트워크를 훌륭한 도구로 만들어 줍니다.

시험 필수 사항

- **프로세스/애플리케이션 계층 프로토콜을 기억하세요** . Telnet은 원격 호스트에 로그인하여 프로그램을 실행할 수 있게 해주는 터미널 에뮬레이션 프로그램입니다. 파일 전송 프로토콜(FTP)은 파일을 전송할 수 있게 해주는 연결 지향 서비스입니다. Trivial FTP(TFTP)는 연결 없이 파일을 전송하는 프로그램입니다. Simple Mail Transfer Protocol(SMTP)은 sendmail 프로그램입니다.
- **OSI 모델의 맥락에서 데이터 캡슐화 및 역캡슐화를 이해합니다** . 여기에는 이더넷 헤더, 인터넷 프로토콜(IP) 헤더, 전송 제어 프로토콜(TCP)/사용자 데이터그램 프로토콜(UDP) 헤더, TCP 플래그, 페이로드 및 최대 전송 단위(MTU)가 포함됩니다.
- **일반적인 포트와 프로토콜, 그 활용법, 그리고 암호화 방식에 대해 설명할 수 있어야 합니다** . FTP, SSH, SFTP 등 학습 목표에 나열된 모든 프로토콜 포트를 숙지해야 합니다.

- **프로토콜 유형을 식별하고 정의할 수 있어야 합니다** . 여기에는 인터넷 제어 메시지 프로토콜(ICMP), TCP, UDP, 일반 라우팅 캡슐화(GRE), 인터넷 프로토콜 보안(IPSec), 인증 헤더(AH)/캡슐화 보안 페이로드(ESP)가 포함됩니다.
- **호스트 간 계층 프로토콜을 기억하세요** . 전송 제어 프로토콜(TCP)은 연결 지향 프로토콜로, 확인 응답과 흐름 제어를 사용하여 안정적인 네트워크 서비스를 제공합니다. 사용자 데이터그램 프로토콜(UDP)은 연결 없는 프로토콜로, 오버헤드가 낮지만 신뢰성이 떨어지는 것으로 간주됩니다.
- **인터넷 계층 프로토콜을 기억하세요** . 인터넷 프로토콜(IP)은 인터넷 네트워크를 통해 논리적 네트워크 주소 지정 및 라우팅을 제공하는 비연결형 프로토콜입니다. 주소 확인 프로토콜(ARP)은 알려진 IP 주소에서 하드웨어 주소를 찾습니다. 인터넷 제어 메시지 프로토콜(ICMP)은 진단 및 목적지 도달 불가 메시지를 제공합니다.

필기 실험

다음 질문에 답하십시오. 답은 부록 A 에서 찾을 수 있습니다 .

1. ARP 대상 MAC 주소는 어떤 형식으로 표시될까요?
2. TCP 포트 20과 21을 모두 사용하는 프로토콜의 이름을 쓰시오.
3. DNS 서버는 어떤 전송 계층 프로토콜을 사용합니까?
4. IP 주소를 직접 사용하여 패킷을 구성함으로써 소스 호스트에 오류를 동적으로 보고하는 프로토콜은 무엇입니까?
5. 핑 테스트에는 성공한 서버가 FTP, HTTP 등과 같은 특정 TCP/IP 서비스를 제공하지 않는 이유는 무엇일까요?
6. RDP에 사용되는 잘 알려진 포트 번호는 무엇입니까?
7. MGCP 프로토콜은 어떤 포트를 사용합니까?
8. 윈도우 운영 체제의 명령 실행에 `ping` 핵심 적인 역할을 하는 프로토콜은 무엇입니까 ? `tracert`
9. TFTP 클라이언트는 네트워크를 통해 파일을 전송할 때 어떤 대상 전송 계층 프로토콜과 포트 번호를 사용합니까?
10. SMTP, POP3, RDP 및 IMAP4 서버는 어떤 잘 알려진 포트 번호를 사용합니까?

복습 문제

복습 문제에 대한 답은 부록 B 에서 찾을 수 있습니다 .

1. OSI 모델은 7개의 계층으로 이루어져 있고, 미국 국방부(DoD) 모델은 4개의 계층으로 이루어져 있습니다. SMTP는 두 모델 모두에서 어느 계층에서 작동합니까?

   A. 회로망

   B. 수송

C. 세션
D. 애플리케이션
E. 인터넷

2. HTTPS를 사용하여 안전한 통신을 해야 합니다. 기본적으로 사용되는 포트 번호는 무엇입니까?
A. 69
B. 23
C. 21
D. 443

3. IP 주소, 서브넷 마스크, 기본 게이트웨이 및 DNS 정보를 포함한 IP 구성을 자동화하는 메커니즘을 구현하려고 합니다. 이를 위해 어떤 프로토콜을 사용하시겠습니까?
A. SMTP
B. SNMP
C. DHCP
D. ARP

4. 로컬 장치의 하드웨어 주소를 찾는 데 사용되는 프로토콜은 무엇입니까?
A. RARP
B. ARP
C. IP
D. ICMP
E. 부트P

5. 보안되지 않은 네트워크를 통해 Unix/Linux 서버에 로그인해야 합니다. 다음 프로토콜 중 어떤 것을 사용하면 이 서버를 원격으로 안전하게 관리할 수 있습니까?
A. 텔넷
B. SSH
C. SFTP
D. HTTP

6. IP 주소로는 ping이 되지만 호스트 이름(FQDN)으로는 ping이 되지 않는다면, 관련된 서버 프로세스의 포트 번호는 다음 중 어느 것입니까?
A. 21
B. 23
C. 53
D. 69
E. 80

7. 다음 중 DHCP Discover 메시지를 설명하는 것은 무엇입니까? (두 개를 선택하십시오.)
A. 레이어 2 브로드캐스트로 FF:FF:FF:FF:FF:FF를 사용합니다.
B. 이 시스템은 전송 계층 프로토콜로 UDP를 사용합니다.
C. 이 시스템은 전송 계층 프로토콜로 TCP를 사용합니다.

D. 레이어 2 목적지 주소를 사용하지 않습니다.

8. 텔넷 연결에 사용되는 레이어 4 프로토콜은 무엇이며, 기본 포트 번호는 무엇입니까?
   A. IP, 6
   B. TCP, 21
   C. UDP, 23
   D. ICMP, 21
   E. TCP, 23
9. ICMP 패킷에 관한 다음 설명 중 맞는 것은 무엇입니까? (두 가지를 선택하십시오.)
   A. 그들은 TCP 세그먼트의 수신을 확인합니다.
   B. 그들은 데이터그램 전송을 보장합니다.
   C. 이들은 호스트에게 네트워크 문제에 대한 정보를 제공할 수 있습니다.
   D. 이러한 정보는 IP 데이터그램 내에 캡슐화되어 있습니다.
   E. 이것들은 UDP 데이터그램 내에 캡슐화되어 있습니다.
10. 다음 서비스 중 TCP를 사용하는 서비스는 무엇입니까? (네 가지를 선택하십시오.)
    A. DHCP
    B. SMTP
    C. SNMP
    D. FTP
    E. HTTP
    F. TFTP
11. 다음 서비스 중 UDP를 사용하는 서비스는 무엇입니까? (세 가지를 선택하십시오.)
    A. DHCP
    B. SMTP
    C. SNMP
    D. FTP
    E. HTTP
    F. TFTP
12. OSI 모델의 응용 계층에서 사용되는 TCP/IP 프로토콜은 다음 중 무엇입니까? (세 가지를 선택하십시오.)
    A. IP
    B. TCP
    C. 텔넷
    D. FTP
    E. TFTP
13. 다음 중 이메일 서버들이 서로 메시지를 교환하는 데 사용하는 프로토콜은 무엇입니까?
    A. POP3

B. IMAP
C. SMTP
D. HTTP

14. 사무실 데스크톱 컴퓨터에만 설치된 애플리케이션을 실행하려면 인터넷 연결이 필요합니다. 어떤 프로토콜이 회사 컴퓨터에 GUI 인터페이스를 제공할까요?
A. 텔넷
B. FTP
C. RDP
D. IMAP
E. SMTP

15. 다음 프로토콜 중 TCP와 UDP를 모두 사용할 수 있고, 네트워크 장치에 대한 인증 및 보안 폴링을 허용하며, 네트워크 장치에 대한 자동 알림 및 보고를 제공하는 프로토콜은 무엇입니까?
A. DNS
B. SNMP
C. SMTP
D. TCP

16. 두 호스트 간에 파일을 전송해야 합니다. 어떤 프로토콜을 사용할 수 있습니까?
A. SNMP
B. 찢다
C. NTP
D. FTP

17. IP 스택에서 OSI 모델의 전송 계층에 해당하는 계층은 무엇입니까?
A. 애플리케이션
B. 호스트 대 호스트
C. 인터넷
D. 네트워크 액세스

18. 네트워크의 모든 장치에서 시간이 일관되게 유지되도록 해야 합니다. 네트워크에서 어떤 프로토콜을 실행해야 합니까?
A. FTP
B. SCP
C. NTP
D. RTP

19. 다음 중 서버가 동일한 호스트에서 동시에 발생하는 여러 요청을 구분할 수 있도록 하는 것은 무엇입니까?
A. 그들은 서로 다른 포트 번호를 사용합니다.
B. NAT 서버는 후속 요청에 대해 IP 주소를 변경합니다.
C. 서버는 동일한 호스트에서 동시에 여러 세션을 허용할 수 없습니다. 이전 세션이 종료되어야 다음 세션이 시작될 수 있습니다.
D. 각각의 MAC 주소는 고유합니다.

20. 다음 중 TCP와 UDP를 모두 사용하는 것은 무엇입니까?

A. FTP

B. SMTP

C. 텔넷

D. DNS