**정보보안기사**

**5장 네트워크 보안**

**19절 네트워크 개요(최종수정일 7/28)**

**프로토콜:** 신뢰/효율/안전하게 정보를 주고받기 위해 정보의 송/수신자측 또는 네트워크 내에서 사전에 약속된 규약 또는 규범. 송신자가 수신자를 호출해 연결하는 과정, 통신회선에서의 접속방식, 전달하는 정보의 형태, 오류 제어, 송/수신자측 간의 동기방식 등에 대한 약속.

**프로토콜의 구성 요소: 구문(Syntax) – 데이터의 구조와 형식 표현순서, 의미(Semantics) – 비트의 영역별 의미, 타이밍(Timing) – 데이터를 언제 얼마나 빨리 전송할 것인지**

**물리적/논리적 프로토콜:** 물리적 프로토콜은 데이터 전송에 사용되는 전송 매체, 접속용 커넥터 및 전송 신호등 물리적 요소에 관한 규약.(RS-232C, V.21) 논리적 프로토콜은 데이터 표현 형식 단위인 프레임의 구성, 프레임 내의 각 항목의 의미와 기능, 데이터 전송 절차 등을 말한다. SNA, SDLC 등 IBM에서 만든 전용 프로토콜(동일 회사 제품끼리 이용)과 TCP/IP 등의 범용 프로토콜로 나뉜다.

**OSI 모델과 TCP/IP 프로토콜**

**OSI 모델:** ISO 표준은 **개방 시스템 상호연결**(OSI, Open System Interconnection) 모델이다. 기반 구조에 관계없이 모든 정보 처리 시스템의 통신을 가능하게 하는 프로토콜의 집합. OSI 모델은 하드웨어나 소프트웨어 기반에 관계없이 다른 시스템 간의 통신을 원활하게 하는데 목적을 둔다.

**물/데/네: 물리, 데이터 링크, 네트워크 계층 -> 물리적인 면(전기 규격, 물리적 연결, 물리 주소, 전송시간과 신뢰도 등) 처리**

**전송 계층: 위 아래 두 그룹을 연결하고 하위 계층(물/데/네)에서 전송한 내용을 상위 계층이 사용할 수 있는 형태가 되도록 보장**

**세/표/응: 세션, 표현, 응용 계층 -> 사용자 지원 계층(관련이 없는 소프트웨어 시스템 간의 상호 연동을 가능하게 함)**

|  |  |  |  |
| --- | --- | --- | --- |
| **계층** | **특징** | **데이터 종류** | **예** |
| **응용** | 각종 응용프로그램  네트워크 관리 | 메시지 | FTP, TFTP, SNMP, SMTP,  Telnet, HTTP, DNS, DHCP |
| **표현** | 네트워크 보안(암,복호화)  압축/압축해제, 포맷 변환 수행 | ASCII, Mpeg, jpg, MME |
| **세션** | 소켓 프로그램  동기화  세션 연결/관리/종료 | 전송모드 결정(반이중, 전이중), SQL, RPC |
| **전송** | 데이터 전송보장  흐름 제어  Quality Of Service(QOS) | 세그먼트 | TCP, UDP, SCTP |
| **네트워크** | 통신경로 설정, 중계기능 담당  라우팅  IPv4 & IPv6 | 패킷 | IP, ICMP, IGMP, ARP, RARP, NAT, RIP, BGP |
| **데이터 링크** | 오류제어, Frame화  매체제어(MAC)  에러검출, 에러정정, 흐름제어 | 프레임 | 이더넷, 토큰링, PPP, SLIP, 802.11(WLAN) |
| **물리** | 물리적 연결설정, 해제  전송방식, 전송매체 | 비트 스트림 | 기계적 전기적, 절차적 규칙 |

**TCP/IP 프로토콜 그룹**

현재 인터넷에서 사용하는 프로토콜 그룹. 상호작용하는 모듈로 이루어진 계층적 프로토콜이다. 각 모듈은 특정 기능을 제공한다. 4 or 5 계층 모델로 간주된다. 두 모델 모두 적용 가능하다.

응용, 표현, 세션 -> 애플리케이션, 전/네/데/물 로 나뉜다.

|  |  |
| --- | --- |
| 계층 | 설명 |
| 데이터링크 | 전송 매체에 프레임을 송수신하는 역할 |
| 네트워크 | IP -> 호스트 네트워크 주소를 관리, 패킷을 라우팅  ARP -> 네트워크 호스트들의 하드웨어 주소를 얻는데 사용  ICMP -> 패킷 전송에 관한 오류 처리를 담당 |
| 전송 | 종단간 통신 서비스 제공을 담당한다. TCP/UDP가 있다. TCP는 데이터의 확실한 전송이 필요한 경우 사용, UDP는 데이터의 정확한 전달 보장 x |
| 응용 | 응용프로그램이 네트워크에 접근 가능하도록 인터페이스 기능 수행 |

**통신단위**

물리 – 비트, 네트워크 – 패킷, 전송 – 세그먼트 || 데이터그램 || 패킷, 응용 – 메시지

**캡슐화와 역캡슐화**

발신지 – 캡슐화, 수신지 – 역캡슐화, 데이터 링크 – 캡슐화/역캡슐화 이뤄지지 않음

**주소 지정**

TCP/IP 프로토콜을 이용한 인터넷은 3개의 다른 주소를 이용한다. -> 물리(MAC), 논리(IP), 포트 주소

**논리 주소:** 물리 주소는 홉에서 홉으로 갈 때마다 변경되지만 논리 주소는 변경되지 않는다. 물리 네트워크와 독립적으로 전세계적인 통신 서비스를 위해 필요. 현재 인터넷에 연결된 호스트를 유일하게 식별할 수 있는 32비트 주소 체계를 사용한다. 인터넷상의 두 호스트는 동일한 IP 주소를 사용할 수 없다.

**포트 주소:** 인터넷의 최종 목적은 한 프로세스가 다른 프로세스와 통신할 수 있도록 하는 것이다. A가 TELNET을 이용해 C와 통신하는 동시에 FTP를 이용해 B와 통신하려면 서로 다른 프로세스를 식별하기 위한 포트가 필요하다.

**다중화와 역다중화:** TCP/IP는 몇몇 계층에서 여러 프로토콜을 사용하기 때문에 발신자측은 다중화, 수신자측은 역다중화한다.

20절. TCP/IP

**물리 계층**

**데이터와 신호:** 응용, 전송, 네트워크, 데이터 링크의 통신은 논리적, 물리 계층 통신은 물리적이다. 호스트-대-라우터, 라우터-대-라우터, 라우터-대-호스트, 교환기 등이 물리적 통신이다.

**데이터링크 계층**

**노드와 링크:** 데이터링크 계층의 통신은 노드-대-노드. 데이터 유닛은 하나의 지점으로부터 LAN과 WAN 등 많은 네트워크를 통해 전달됨. LAN과 WAN은 라우터를 통해 전달. 이것을 노드와 링크로 나타내는 게 데이터링크 계층이다.

**서비스 순서**

**1. 프레임 짜기(Framing):** 각 노드에서 다음 노드에게 데이터그램을 전송하기 전에 네트워크계층에서 전달받은 데이터그램(패킷)을 프레임으로 캡슐화한다.

**2. 흐름 제어(Flow Control):** 서로 다른 데이터링크 계층 프로토콜은 서로 다른 흐름 제어 전략을 활용한다. 전송 계층에서 흐름 제어는 매우 중요.

**3. 오류 제어(Error Control):** 전기 신호는 오류에 취약하기 때문에 프레임 또한 오류에 취약. 이에 따라 첫 번째로 오류 검출이 필요하다. 오류 검출 후에 송신자 노드에서 오류를 수정하거나 폐기하고, 재전송을 송신 노드에게 요청해야만 한다.

**4. 혼잡 제어(Congestion Control):** 프레임 혼잡으로 프레임 손실이 발생해도 어떤 광역 네트워크 이외의 대부분의 데이터링크 계층 프로토콜은 직접적으로 혼잡을 완화하기 위해 혼잡 제어를 사용하지 않는다. 일반적으로 혼잡 제어는 종단-대-종단 통신이기 때문에 네트워크 또는 전송 계층 문제로 여겨진다.

**프레이밍 – 흐름 제어 – 오류 제어 – 혼잡 제어(일반적으로는 네트워크나 전송 계층의 일)**

**네트워크 계층**

**라우팅:** 라우팅은 route의 현재 진행형으로 경로배정을 의미한다. 패킷이 송신지에서 목적지까지 갈 수 있도록 경로를 배정하는 것.

**포워딩:** 포워딩은 라우팅 프로토콜을 실행할 때, 라우터상 하나의 인터페이스에 패킷이 도착했을 때 라우터가 취하는 행동이다. 라우터는 일반적으로 포워딩 테이블 혹은 라우팅 테이블이라는 표에 의거해 행동한다.

**IPv4:** 8 \* 4 =32bit. 전체 주소 공간은 A, B, C, D, E 5개의 클래스로 구분된다(Classful addressing).

**클래스A:** 첫 비트 0, 첫 비트의 나머지 7비트가 네트워크 주소. 하위 세 바이트는 호스트 주소

**클래스B:** 처음 두 비트 10, 이후 두 번째 바이트까지 네트워크 주소. 마지막 두 바이트는 호스트 주소

**클래스C:** 처음 세 비트 110, 이후 세 번째 바이트까지 네트워크 주소. 네트워크마다 254개 호스트 수용 가능, 작은 규모의 네트워크에서 사용

**클래스D:** 처음 네 비트 1110. 이후 네트워크 주소와 호스트 주소 구분 없음. 전체 주소가 멀티캐스트용. 멀티미디어 데이터, 리얼타임 비디오 등을 보낼 때 사용

**클래스E:** 처음 네 비트 1111. 추후 사용(시험용)을 위해 예약된 주소

**주소 고갈:** 클래스 기반 주소는 주소 고갈로 인해 더이상 사용하지 않는다.

**서브네팅과 슈퍼네팅:** 주소 고갈을 완화하기 위해 등장. 네트워크부는 부족한데 호스트부는 낭비되는 문제 발생. 서브네팅 도입으로 해결. 서브네팅의 핵심은 할당된 서브넷마스크(ex.255.255.255.0)를 원래의 IP주소와 AND 연산하는 것. 네트워크ID를 서브 네트워크ID로 쪼개어 더 많은 네트워크 주소를 할당하고 호스트 부에 원래보다 적은 양의 주소를 할당한다.

**클래스 없는 주소 지정:** 클래스 기반 주소 지정에서는 서브네팅을 적용해도 고갈 문제를 해결할수 없었다. 그런데 왜 바로 IPv6로 넘어가지 않았나? -> 더 큰 주소 공간을 확보하려면 IP주소 길이의 증가로 인해 IP패킷 형식까지 변경해야 했기 때문… 단기적 관점의 해결책으로 클래스 없는 IPv4 주소지정 방식이 사용되었다. -> 클래스 권한 제거

**CIDR(Classless Inter-Domain Routing):** ISP가 클라이언트가 요구하는 양만큼만 주소를 잘라서 공급할 수 있는 장점이 있다. 따라서 라우팅 테이블의 비대화를 막고 인터넷을 여러 개의 Addressing Domain으로 나누어 라우팅 정보량을 줄여준다. 한 도메인 내에서는 도메인 내의 모든 라우팅 정보가 공유된다.

CIDR표기법(슬래시 표기법) -> byte.byte.byte.byte/prefix length -> 클래스가 없기 때문에 슬래시 뒤에 네트워크 ID의 길이를 적어준다.

**VLSM(Variable Length Subnet Mask, 가변 길이 서브넷 마스킹):** VLSM은 서브네팅을 여러 번 반복해 네트워크를 크기가 다른 여러 세브넷으로 구분한다.

문제. C class IP주소를 가진 회사에서 100개의 주소를 필요로 하는 부서와 25개의 주소를 필요로 하는 4개의 부서를 서브네팅 하는 경우의 적용(192.168.120.0)

**특수 주소: 다섯 가지 ->** this-host 주소, 제한된 브로드캐스트 주소, 루프백 주소, 사설 주소, 멀티캐스트 주소

**디스-호스트 주소: 0.0.0.0/32** -> 호스트가 IP데이터그램을 보내려 하지만 근원지 주소인 자신의 주소를 모를 때 사용

**제한된 브로드캐스트 주소: 255.255.255.255/32** -> 호스트나 라우터가 네트워크상의 모든 장치로 데이터그램을 보낼 때 사용. 네트워크상의 라우터가 이런 패킷을 차단하기 때문에 네트워크 외부로 패킷을 보낼 수는 없다.

**루프백 주소: 127.0.0.0/8** -> 이 블록 내의 패킷은 호스트를 벗어나지 않고 호스트에 남는다. 소프트웨어 테스트용. 127.0.0.1이 가장 많이 쓰인다.

**사설 주소: RFC 1918 ->** RFC1918이 정의한 사설 주소에서만 쓸 수 있는, 라우팅이 불가능한 특수 주소 집합. 공중 인터넷에 존재하지 않는 주소들. 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16이 사설 주소로 지정되어 있다.

**사설 클래스A: 24bit block, 10.0.0.0 – 10.255.255.255**

**사설 클래스B: 20bit block, 172.16.0.0 – 172.31.255.255**

**사설 클래스 C: 16bit block, 192.168.0.0 – 192.168.255.255**

**네트워크 계층 프로토콜**

**IP(인터넷 프로토콜):** 비신뢰/비연결형 데이터그램 프로토콜. 최선형 전송 서비스(Best-Effort Delivery Service)최선형 전송이란 IPv4패킷이 훼손, 손실, 틀린 순서로 도착, 지연도착, 네트워크 혼잡 등을 발생을 시킬 수 있음을 의미한다. 신뢰성이 중요하다면 IPv4는 TCP처럼 신뢰성을 보장하는 전송 계층 프로토콜과 함께 사용되어야 한다. TCP/UDP, ICMP, IGMP 데이터는 모두 IP 데이터그램을 사용해 전송된다.

**IPv4 데이터그램 암기사항**

**데이터그램이란? -> IP가 사용하는 패킷으로 TCP/UDP, ICMP, IGMP 등이 데이터그램을 통해 전송된다. IP 프로토콜은 비신뢰성, 비연결성(Connectionless), 주소 지정, 경로 설정 방식의 프로토콜이며 데이터그램은 가변 길이의 패킷으로 헤더(주소/제어필드)와 페이로드(데이터)로 이루어져 있다. 헤더는 20~60바이트 길이이며 라우팅과 전송에 필수적인 정보를 가지고 있다. TCP/IP에서는 일반적으로 헤더를 4바이트로 표현한다. 버전 -> IPv4, 헤더 길이 -> 옵션을 포함한 헤더길이, TOS(Type Of Service) -> 3비트의 우선권(Precedence) 필드와 4비트의 TOS 필드 그리고 1비트의 예약 필드, Total Length -> 전체 길이를 바이트로 표현, TTL(Time To Live) -> 패킷이 경유할 최대 홉수, 식별자(Identification) -> 호스트가 보낸 각 데이터그램을 식별, 플래그 -> 세 개의 비트로 단편화 정보 표시 이하 그림**

**IPv4 데이터그램**

**텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명**

**ARP(Address Resolution Protocol):** 자신의 IP, MAC, 수신자의 IP는 알고 있지만 수신자 측의 MAC은 모르는 상황에서 물리 계층 브로드캐스트를 통해 모든 호스트에게 패킷을 전송한다. -> ARP 메시지를 수신한 호스트 또는 라우터는 수신 IP 주소와 자신의 IP 주소를 대조한 뒤 자신의 MAC을 요구하는 경우라면 ARP 응답 메시지를 송신자에게 전송한다. -> 각 시스템에는 ARP Cache가 있고 Cache에 이 정보를 보관한다. 정보는 일정 시간 경과 후 삭제한다.

**MAC(Media Access Control):** NIC의 ROM칩에 생산자가 프로그램한 고유 물리주소. 데이터 링크 계층이 이 MAC주소를 이해하고 작업하며 네트워크 계층은 IP 주소를 이해하고 작업한다.

**ARP Cache Table:** ARP를 통해 알아낸 MAC 정보는 OS에 따라 다르지만 통산 1~2분 정도 캐시에 저장한다. arp -a 명령을 통해 ARP Cache table을 살펴보면 dynamic과 staticd으로 타입이 나뉜다. Static은 관리자에 의해 정적으로 생성된 것으로 임의로 삭제하거나 시스템 종료 전까지 지속되고 dynamic은 ARP에 의해 동적으로 설정되어 일정 시간 동안만 유지된다.

**ARP 메시지 종류:**

1. ARP 요청 메시지 -> 특정 IP주소에 대한 물리적 주소를 요구한다. 호스트는 수신자측 물리 주소를 모르기 때문에 브로드캐스트로 전송한다.
2. ARP 응답 메시지 -> 유니캐스트로 물리 주소를 알려준다. ARP 요청 메시지를 수신한 호스트 또는 라우터는 ARP 응답 메시지를 전송한다.(호스트가 라우터를 넘어 다른 네트워크에 있으면 해당 호스트를 대신해 응답 메시지를 전송한다.)

**RARP(Reverse Address Resolution Protocol):** ARP의 반전. MAC은 알고 IP를 모를 때 사용한다. RARP는 저장장치가 없는 호스트에서 주로 사용한다. 또 RARP 응답은 일반적으로 RARP 서버에서 생성한다. 호스트가 자신의 MAC 정보를 담은 RARP 요청 메시지를 만들어 브로드캐스트 한다. -> RARP 서버는 요청자의 IP 주소를 담은 RARP 응답 메시지를 만들어 요청자의 MAC 주소로 유니캐스팅한다.

**GARP(Gratuitous ARP):** 별도의 프로토콜이 아니다. Sender IP와 Target IP가 동일한 ARP 요청을 말한다.

장비가 ARP 요청 브로드캐스트를 통해 다른 장비에게 네트워크에 있는 자신의 존재를 알릴 때 사용한다. 이 패킷을 수신한 장비는 자신의 ARP Cache에 해당 정보가 있다면 이를 갱신한다. 자신의 MAC을 동일 네트워크 상의 다른 장비에게 알려 ARP Cache를 갱신하기 위해 사용한다.

**GARP 쓰임**

1. **IP충돌 감지:** 자신과 동일한 IP가 설정된 호스트가 있다면 해당 호스트로부터 ARP 응답이 오기 때문에 충돌 여부를 확인 가능. 호스트 IP를 변경하거나 재부팅 시에 GARP 패킷이 생성된다.
2. **상대방 ARP Cache 정보 갱신:** 수신 측에서 자신의 ARP Cache에 Sender IP 정보를 갱신한다. 이 부분이 취약점이 될 수 있다. 상대방을 인증하지 않고 Cache정보를 갱신해 악의적 목적의 공격자에 의해 MAC 정보가 위조, 변조될 수 있다.

**ICMPv4(Internet Control Message Protocol version 4):** 호스트가 라우터 상태나 다른 호스트, 라우터의 정보를 획득하기 위해 사용. IP의 동반 프로토콜이다.

**ICMPv4 메시지:** 오류보고와 질의 메시지. 오류보고는 라우터(목적지)나 호스트가 IP 패킷을 처리하는 도중에 탐지하는 문제를 보고한다. 첫 번째 필드인 Type이 메시지 유형을 나타낸다. 코드 필드는 특정 메시지 유형의 이유를 지정한다. 마지막 공통 필드는 검사 합 필드.

* **오류보고 메시지:** ICMP의 주된 임무 중 하나는 IP 데이터그램의 프로세싱 동안 발생하는 오류를 보고하는 것. 오류 수정은 직접 할 수 없고 상위 계층 프로토콜에 맡긴다. 데이터그램으로 알 수 있는 경로 정보는 근원지와 목적지 IP밖에 없기 때문에 오류 메시지는 언제나 최초의 근원지에 보내진다.
  + **ICMP 메시지 중요한 점**

1. **ICMP 오류 메시지는 메시지를 운반하는 데이터그램의 응답으로 생성되지 않는다.**
2. **ICMP 오류 메시지는 처음 단편이 아닌 단편 데이터그램을 위해 생성되지 않는다.**
3. **ICMP 오류 메시지는 멀티캐스트 주소를 가진 데이터그램을 위해 생성되지 않는다.**
4. **ICMP 오류 메시지는 127.0.0.0 또는 0.0.0.0과 같은 특수 주소의 데이터그램을 위해 생성되지 않는다.**

* **목적지 도달 불가(Destination Unreachable, Type 3):** 가장 널리 사용되는 오류 메시지. 데이터그램이 왜 목적지에 도달하지 못했는지에 대한 오류 메시지를 정의하기 위해 0~15까지의 다른 코드를 사용한다.
* **근원지 억제(Source Quench, Type 4):** 송신자에게 네트워크 충돌이 발생해 데이터그램이 폐기되었음을 알린다. 이 메시지를 받은 근원지는 데이터그램 송신을 천천히(또는 억제) 수행한다. ICMP가 이런 메시지를 통해 IP프로토콜에는 원래 없는 혼잡제어 메커니즘을 추가하는 것이다.
* **재지정 메시지(Redirection, Type 5):** 발신자가 잘못된 라우터를 통해 메세지를 전송할 때 사용. 라우터는 알아서 적절한 라우터로 메시지를 보내도록 재지정되지만 발신자에게 향후 디폴트 라우터로 변경하라고 알린다. 따라서 메시지 안에 디폴트 라우터 IP가 포함된다.
* **시간 경과 메시지(Time Exceeded, Type 11):** 타임아웃이 발생해 IP패킷이 폐기되었음을 알린다. 코드를 통해 타임아웃 사유를 알 수 있다.

1. **Code 0(TTL exceeded In Transit): IP패킷이 최종 목적지에 도달 전에 TTL 값이 0이 됨**
2. **Code 1(Fragment reassembly time exceeded): IP 패킷 재조합 과정에서 타임아웃이 발생해 해당 IP 데이터그램이 모두 폐기됨. IP 데이터그램의 단편이 전송과정에서 손실될 경우 재조합 실패로 발생**

* **매개변수 문제(Parameter Problem, Type 12):** 헤더에 문제가 있거나(코드 0) 어떤 옵션이 없거나 옵션의 의미를 알 수 없는 경우(코드 1)에 전송될 수 있다.

**대표적인 ICMP 메시지**

|  |  |
| --- | --- |
| **ICMP** | **메시지 기능** |
| Echo Request | 원하는 호스트와 IP 연결 됐는지 확인 |
| Echo Reply | ICMP Echo Request 응답 메시지 |
| Redirect | 데이터를 보낸 호스트에게 목적 IP주소에 좀 더 적합한 경로가 있음을 알리기 위해 라우터가 보내는 메시지 |
| Source Quench | 데이터를 보낸 호스트에게 IP 데이터그램이 라우터의 집중 현상에 의해 손실되고 있음을 알리기 위해 라우터가 보내는 메시지. 데이터를 보내는 호스트는 전송률을 낮추게 된다. Source Quench는 ICMP에서 선택적 메시지이며 대부분 구현되지 않는다. |
| Destination Unreachable | 라우터나 목적 호스트에 의해 보내지며 데이터그램이 전달되지 못했다는 것을 데이터를 보낸 호스트에게 알린다. |

**질의 메시지:** 인터넷에서 호스트나 라우터가 활성화되었는지를 알아보거나 두 장치 사이에 IP 데이터그램이 단방향 시간인지 왕복 시간인지 찾는다. 에코요청(type 8)과 에코응답(type 0) 메시지 쌍은 다른 호스트나 라우터가 활성화됐는지 여부를 테스트하기 위해 호스트나 라우터가 사용한다. 핑과 트레이스루트(traceroute) 같은 디버깅 도구에서 에코 요청과 에코 응답으로 메시지 쌍을 확인할 수 있다.

**IPv6:** IPv4의 한계(32비트, 43억개)로 등장, 32 비트 주소체계를 4배 확장한 128비트 주소 길이를 사용한다. 약 3.4 \* 10^38(2^128) 개 주소 생성 가능. 보안 문제, 라우팅 효율성 문제 해결, QoS(Quality of Service) 보장, 무선 인터넷 지원과 같은 다양한 기능을 제공한다. 이동전화, 가전제품 등에 도입 추진 중.

**IPv6 특징:**

* **확장된 주소 공간:** NAT(Network Address Translation) 불필요, 유니캐스트, 애니캐스트, 멀티캐스트 주소 형태 지원, 100억명이 1000개 단말기 사용 가능 -> 2^128
* **새로운 헤더 포맷:** IPv4에서 사용하지 않던 헤더를 지우고 확장 헤더를 도입해 선택적 사용을 가능하게 함. 패킷을 중계하는 라우터의 부하를 줄인다. 또 헤더를 고정 길이로 변경해 시스템이 헤더 길이를 예측해 빠른 처리가 가능하게 만들었다.
* **패킷 단편화(fragmentation) 필드 삭제:** 패킷을 보내기 전에 전송에 적합한 패킷크기를 알아내는 MTU(Maximum Transfer Unit) 탐색 기능 추가
* **체크섬 필드 삭제:** 데이터링크 계층에서 체크섬 계산이 행해지므로 삭제

**IPv6는 향상된 서비스를 지원한다:** IPv6 헤더에 플로우 레이블 필드를 사용해 응용프로그램에 의해 생성된 트래픽을 실시간/비실시간으로 구분할 수 있다.

**보안기능:** IPv4는 설계 시 보안 규정이 없어 별도의 보안 프로토콜(IPSec)을 필요로 한다.(Add-on)

IPv6는 프로토콜 내부에 보안 기능을 탑재할 수 있다. 인증 절차, 데이터 무결성 보호, 메시지 발신지 확인 기능을 제공한다. 또 확장 헤더를 통해 네트워크 계층의 종단간 암호화를 제공, 패킷 변조를 방지한다.

**IPv6 주소 체계:** IPv4와 달리 브로드캐스트가 없고 애니캐스트가 있는 게 특징. 유니캐스트는 단일 인터페이스(컴퓨터, 라우터)를 정의하고 특정 컴퓨터에게만 전달한다. 애니캐스트 주소는 단일 주소를 공유하는 컴퓨터 집합으로 정의된다. 애니캐스트 주소로 결정된 패킷은 가장 가까이 있는 애니캐스트 그룹의 구성원(가장 짧은 경로를 가지는)에게만 전송된다. 멀티캐스트 주소는 컴퓨터 그룹을 정의한다. 애니캐스트와 차이는 애니캐스트는 패킷의 복사본이 그룹 중 하나의 컴퓨터에만 전송되는데 반해 멀티캐스트는 그룹의 각 컴퓨터가 복사본을 수신한다는 것이다. 브로드캐스트는 멀티캐스트의 특수한 경우로 처리한다.

**IPv4에서 IPv6로 변환: 이중 스택**(동시 사용), **터널링** – 종단간 두 컴퓨터가 IPv6로 소통할 때 Ipv6 패킷이 IPv4를 사용하는 지역을 지나가면 IPv4로 캡슐화 한 뒤 빠져나올 때 역캡슐화 하는 것, **헤더 변환** – IPv6가 주류가 되고 IPv4는 소수가 된 시점에서 IPv6를 사용하는 컴퓨터가 IPv4를 이용하는 컴퓨터와 소통하기 위해 헤더를 변환시켜 통신한다.

**전송 계층**

**UDP(User Datagram Protocol):** 비연결형/비신뢰성 프로토콜. 래퍼 프로토콜 -> 단순히 애플리케이션 데이터를 메시지 포맷으로 포장해 IP에게 전달만 하기 때문. 호스트 간 통신 대신에 프로세스 간 통신을 제공하는 것 외에 어떠한 서비스도 추가하지 않는다. 최소의 오버헤드를 가진 간단한 프로토콜. UDP 패킷은 사용자 데이터그램(User Datagram)이라고도 불리며 고정된 크기의 8바이트 헤더를 가진다. 8바이트의 헤더는 2바이트 짜리 필드 4개로 만들어진다.

**헤더구조 -> 송신 포트 주소(2바이트), 도착지 포트 주소(2바이트), 총 길이(2바이트), 체크섬(2바이트)**

**UDP 서비스:**

* **프로세스-대-프로세스 통신:** IP주소와 포트 번호의 결합인 소켓 주소를 통해 프로세스-대-프로세스 통신을 한다.
* **비연결형 서비스:** 각 사용자 데이터그램은 독립적이다. 동일한 근원 프로세서에서 동일한 목적지 프로그램으로 들어가도 서로 관계없다.
* **흐름 제어:** UDP는 단순 프로토콜로 흐름 제어와 윈도우 메커니즘이 없다. 따라서 들어오는 메시지로 수신자 측에서 오버플로우가 발생할 수 있다. 흐름 제어의 결여는 UDP를 이용하는 프로세스가 스스로 제공해야 한다.
* **오류 제어:** 검사합을 제외하고 UDP에는 오류 제어가 없다. 송신자가 메시지가 손실이 되었는지 또는 중복이 되었는지 알 수 없다.
* **혼잡 제어:** 혼잡 제어도 없다. UDP는 전송 패킷이 너무 작고 산발적이어서 혼잡을 발생시키지도 않는다. 네트워크 혼잡에 관한 설명은 UDP가 요즘 음성과 영상의 실시간 스트리밍에 사용되는 시점에 맞을 수도 맞지 않을 수도 있다.
* **다중화와 역다중화:** 호스트에서 하나의 UDP가 있지만 UDP 서비스를 사용하기 원하는 다수의 프로세스가 있을 수 있다. 이 문제를 해결하기 위해 UDP는 다중화 및 역다중화 한다.

**UDP의 일반적인 응용:** FTP와 같이 대량 데이터를 전송하는 프로세스에는 사용하지 않는다. UDP는 내부 흐름 및 오류 제어 기법을 가진 프로세스에 적당하다. TFTP는 흐름 및 오류 제어를 포함하므로 UDP를 사용할 수 있다. 또한 UDP는 멀티캐스팅을 위한 전송 프로토콜이다. UDP는 SNMP와 같은 관리 프로세스에 사용된다. RIP와 같은 경로 갱신 프로토콜을 위해서도 사용된다. UDP는 실시간 응용에 사용된다.

**TCP(Transmission Control Protocol)**

**TCP 서비스의 특징: 연결형, 신뢰성, 양방향, 승인, 스트림 중심, 흐름 관리**

1. **프로세스-대-프로세스 통신:** UDP처럼 포트 번호를 사용해 프로세스 간 통신 제공
2. **스트림 전송 서비스:** 바이트의 흐름으로 데이터를 송/수신한다. 가상의 튜브에 의해 2개의 프로세스가 연결되는 것처럼 보이는 환경을 만든다.
3. **송신 및 수신 버퍼:** 송/수신 서비스가 똑같은 속도로 전송/처리하는 게 아니기 때문에 TCP 버퍼가 필요하다. 송신 버퍼와 수신 버퍼 두 가지가 있다. 두 버퍼는 흐름 및 오류 제어에 사용된다.
4. **전이중 통신(Full-duplex service):** 데이터는 동시에 양방향으로 전달될 수 있다. 이를 위해 TCP는 송신/수신 버퍼를 가지며 세그먼트는 양방향으로 이동한다.
5. **다중화와 역다중화:** UDP와 마찬가지. 차이는 TCP는 연결-지향 프로토콜이기 때문에 프로세스 쌍을 위해 연결을 설정해야 한다.
6. **연결**-**지향 서비스:** 물리적 연결이 아니라 논리적 연결을 말한다. 논리적 연결 통로를 통해 데이터를 송/수신해 전송 순서를 보장한다. 이를 순서 제어(Sequence Control)라고 한다.
7. **신뢰성 있는 서비스**: 데이터의 안전하고 확실한 도착을 확인하기 위한 확인응답 기법 사용

**세그먼트**

**TCP에서 패킷을 세그먼트라고 한다.**

**세그먼트는 20~60바이트의 헤더와 응용 프로그램으로 구성된다. 헤더는 선택 항목이 없으면 20바이트, 있으면 60바이트까지**

**세그먼트 암기사항:**

1. **근원지 포트주소:** 송신 프로세스의 포트 번호(16비트)
2. **목적지 포트주소:** 수신 프로세스의 포트 번호(16비트)
3. **순서번호:** 32비트, 세그먼트에 포함된 첫 번째 데이터 바이트에 할당된 번호
4. **확인응답 번호:** 32비트, 송신자가 수신자로부터 받기를 기대하는 바이트 번호를 정의한다. 수신자가 상대방으로부터 바이트 번호 x를 성공적으로 수신하면 x + 1이 확인응답 번호가 된다. 확인응답과 데이터는 함께 피기백이 될 수 있다.
5. **헤더 길이:** 4비트, TCP 헤더를 4바이트 단위의 개수로 나타낸 것
6. **제어:** 1비트 \* 6, 6개의 제어(플래그) 비트 정의
7. **윈도우 크기:** 16비트, 수신측에서 송신측에 보내는 Receiver Window Size로 수신 버퍼의 여유 공간 크기를 의미한다. 송신측에서 상대의 여유 공간 크기를 파악하고 흐름제어를 수행할 수 있다.
8. **검사합(checksum):** 16비트, UDP와 동일한 절차를 따른다. UDP검사합은 옵션이지만 TCP 검사합은 필수이다.
9. **긴급 지시자(Urgent Pointer):** 16비트로 긴급 플래그 값이 설정되었을 때만 유효

![텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명]()

|  |  |  |
| --- | --- | --- |
| **플래그** | **의미** | **설명** |
| **URG** | **긴급** | ‘1’ 설정 시 우선순위, 긴급 필드값 사용 |
| **ACK** | **승인** | ‘1’ 설정 시 세그먼트가 승인 포함 |
| **PSH** | **밀어넣기** | 받는 즉시 애플리케이션에 송신 |
| **RST** | **리셋** | 송신 장비 문제, 연결 초기화 |
| **SYN** | **동기화** | 순서 번호 동기화, 연결 수립 |
| **FIN** | **종료** | 송신 장비가 연결 종료 요청 |

**TCP연결:** 연결-지향 프로토콜이기 때문에 근원지와 목적지 사이에 가상 경로를 설정하고 송/수신. 연결 설정, 데이터 전송, 연결 종료의 3단계를 거친다.(Three-way Handshaking)

1. 클라이언트가 SYN 플래그를 세팅한 세그먼트를 서버에 전송한다. 세그먼트는 순서번호와 동기화를 위한 것이다. SYN세그먼트는 데이터를 운반할 수는 없지만 순서번호 하나를 소비한다.
2. 서버는 2개의 플래그 비트가 설정된 두 번째 세그먼트 SYN + ACK 세그먼트를 전송한다. 마찬가지로 데이터를 운반할 수 없으며 순서번호 하나를 소비한다.
3. 클라이언트가 세 번째 세그먼트를 전송한다. ACK 세그먼트이다. ACK 플래그와 확인응답 번호 필드를 사용해 두 번째 세그먼트를 수신한 것에 대한 확인차 보내는 것이다. 데이터를 운반하지 않으면 ACK세그먼트는 순서번호를 소비하지 않는다.

**SYN Flooding 공격:** 쓰리 웨이 핸드셰이크 과정에서 공격자가 무수한 가짜 IP와 포트를 가진 SYN 세그먼트를 날린다. -> 서버는 이를 클라이언트로 인식하고 SYN-ACK 패킷을 보내고 응답을 대기한다. 이 과정에서 서버의 리소스가 낭비되고 실제 클라이언트들이 서버와 연결을 설정하지 못하게 된다. -> 서버는 자원 부족으로 정상적인 응답을 처리하지 못하게 되고 마비되거나 느려진다. -> 모든 요청에 대해 서비스를 거부하는 DoS 공격에 속한다. -> 현재는 SYN Cookie 기법을 사용해 SYN Flooding을 감지하고 방어한다.

**데이터 전송:** 연결이 형성되면 양방향으로 데이터를 전송한다. 클라/서버가 모두 데이터와 확인 응답을 보낼 수 있다. 클라이언트가 보낸 데이터 세그먼트에는 서버 TCP가 데이터가 수신되자마자 서버 프로세스에게 데이터를 전달하는 것을 알도록 PSH(Push) 플래그가 설정되어 있다. 그러나 서버 -> 클라이언트로 전달된 세그먼트는 밀어넣기 플래그를 설정하지 않는다.

**연결 종료:** 시작은 클라이언트(능동 오픈)이지만 종료는 양쪽 모두 가능하다. 현재 대부분의 경우는 세 방향 핸드셰이킹과 절반-종료 옵션을 가진 네 방향 핸드셰이킹 두 방법을 허용한다.

* **세 방향 핸드셰이킹의 종료:** 클라이언트 TCP가 클라이언트 프로세서로부터 종료 명령을 수신한 후 FIN 플래그가 설정된 첫 FIN 세그먼트를 전송한다. -> 서버는 FIN 플래그 수신 후 상황처리를 알리고 FIN 수신을 확인하며, 동시에 다른 방향으로 연결 종료(FIN + ACK)를 위해 두 번째 세그먼트를 송신한다. -> 클라이언트가 FIN + ACK을 받고 마지막 세그먼트인 ACK 세그먼트를 전송한다.
* **네 방향 핸드셰이킹의 종료(절반-폐쇄):** TCP에서 데이터를 받고 있는 도중에 어느 한쪽이 데이터 전송을 중단하는 것을 절반 폐쇄라고 한다. 보통 클라이언트가 시작한다.
* 클라이언트가 FIN 세그먼트를 전송해 절반-폐쇄한다.
* 서버는 ACK 세그먼트를 송신해 절반-폐쇄를 받아들인다.
* 클라이언트는 서버로 데이터 전송을 멈추지만 서버는 여전히 데이터를 송신할 수 있다.
* 서버가 처리된 모든 데이터를 보냈을 때 ACK에 의해 확인 응답되는 FIN 세그먼트가 전송된다.
* 절반-폐쇄 이후에 데이터는 서버에서 클라이언트로 이동 가능하고 확인 응답은 클라이언트에서 서버로 이동할 수 있다. 하지만 클라이언트는 더 이상 서버에 데이터를 전송할 수 없다.
* **연결 재설정:** 한 종단의 TCP가 연결 요구를 거부하거나 현재 연결을 중지, 혹은 유휴 연결을 종료할 수 있다. 이런 재설정은 RST 플래그를 가지고 수행한다.

**응용 계층**

**주요 응용 계층 프로토콜**

**HTTP:** 웹페이지를 가져오기 위해 어떻게 클라이언트-서버 프로그램을 작성할지 정의한다. 클라이언트는 임시 포트를, 서버는 80포트를 사용해 클라이언트가 요청하고 서버는 응답한다.

**FTP:** 파일 전송 프로토콜. TCP/IP표준이다. 종단간 두 시스템은 서로 다른 파일 이름, 방법으로 데이터를 표현할 수 있다. 또 두 시스템은 서로 다른 디렉터리 구조를 가질 수도 있다. FTP는 이러한 모든 기능을 제공한다.

**TELNET: TER**mina**L NET**work의 약자. 최초의 원격 로그인 프로토콜. 로그인 이름과 암호가 필요하지만 평문으로 전송하기에 해킹에 취약하다. 해커는 로그인 이름과 암호를 도청해 얻을 수 있다. 이런 보안 문제 때문에 SSH(Secure Shell)을 사용한다.

**DNS:** 중앙 디렉터리 시스템이 모든 매핑을 보유하지 않고 매핑 정보를 작게 나누어 전 세계의 서로 다른 컴퓨터에 저장한다. 각 호스트는 매핑이 필요할 경우 DNS를 통해 해당 정보를 가진 가장 가까운 컴퓨터와 통신한다.