IP, TCP, UDP

About..

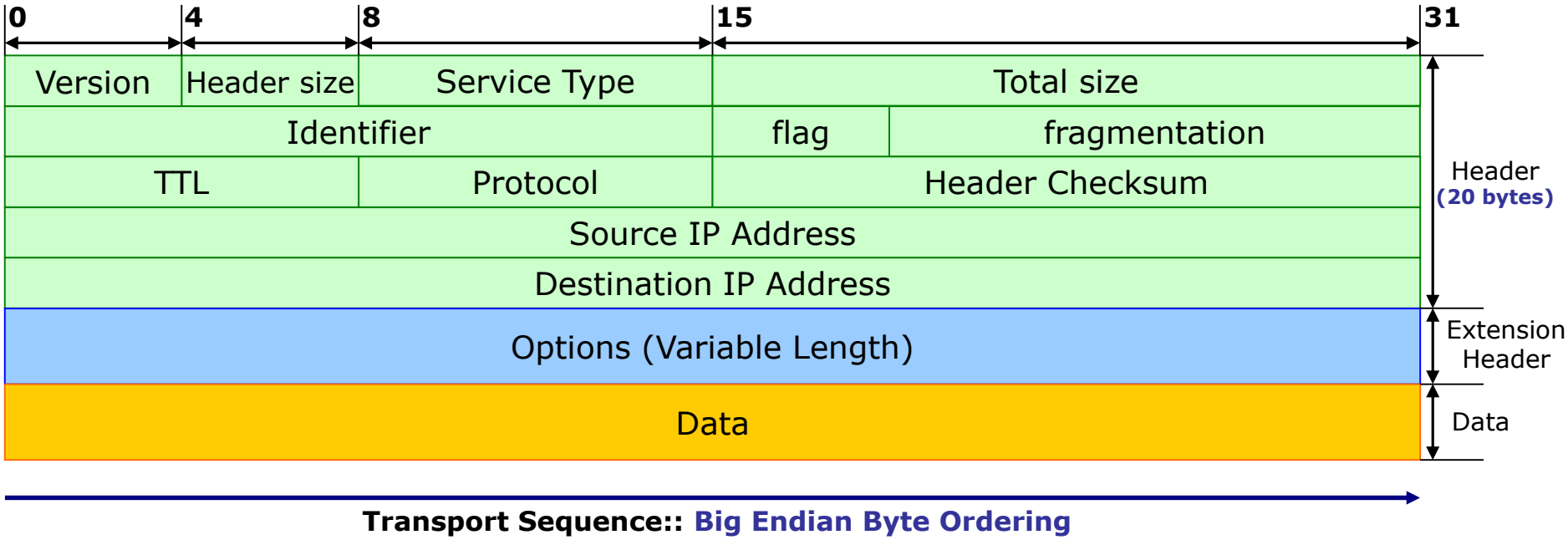
컴퓨터소프트웨어공학과
김 원 일

Internet Protocol – 1

- IP의 특징
  - 비신뢰성(**Unreliable**)
    - 가능한 범위 내에서 목적지까지 전달하는 최선형 서비스
    - **Best Effort Service**
  - 비접속형(**Connectionless**)
    - 연결(connection) 설정 없이 패킷을 전송
  - 주소 지정
    - 네트워크 내의 노드를 고유하게 지정하기 위한 수단으로 IP 주소를 사용
  - 경로 결정
    - 목적지 IP 주소를 기반으로 패킷 전달 경로를 판단

Internet Protocol – 2

- IP Packet 구성
  - 헤더 필드와 데이터 필드로 분리
  - 헤더 필드에는 패킷을 목적지까지 전송하기 위한 값 포함
  - Data는 상위 계층의 PDU가 포함됨

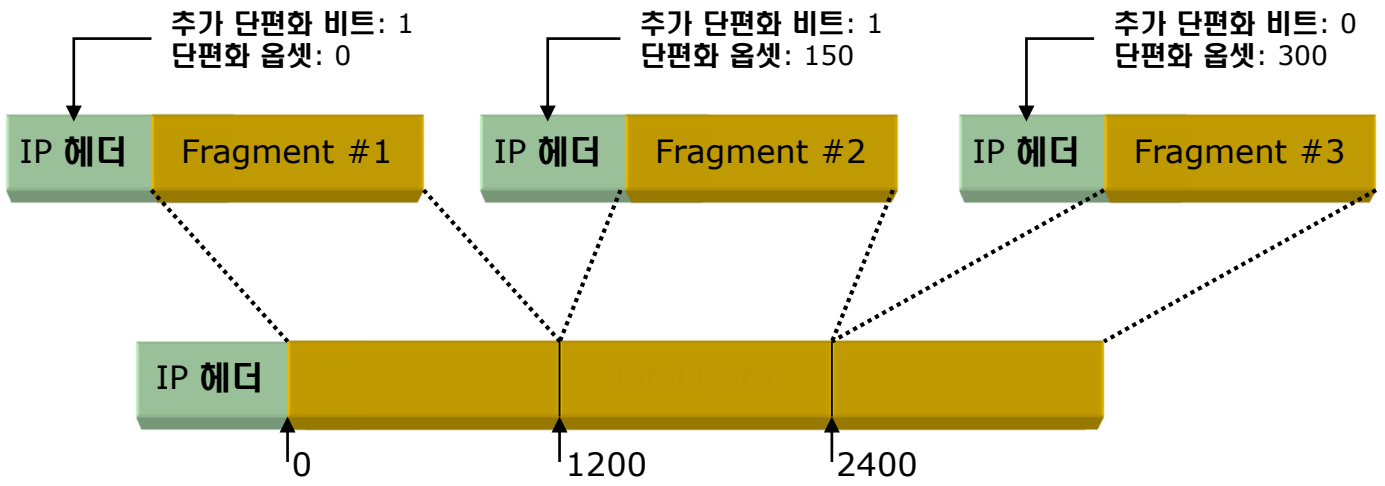


Internet Protocol – 3

- **버전**(Version)
  - IP **프로토콜의 버전을 의미**
- **헤더 길이**(Header Length)
  - **옵션 필드를 포함한 헤더의 총 길이**, 4**바이트로 단위로 산정**
- **서비스 타입**(Type-Of-Service)
  - **우선권**(Precedence)**필드**(3bit), TOS(Type-Of-Service) **필드**(4bit), **예약 필드**(1bit)
  - **우선권 필드는 패킷의 우선순위 정의**
  - TOS **필드는 최소 지연, 최대 처리량, 최대 신뢰성, 최소 비용을 나타내는 필드**
- **전체 길이**
  - **헤더와 데이터를 포함한** IP **패킷의 전체 길이를 바이트 단위**
- **식별자**(Identification)
  - **동일한 데이터로부터 분할된 패킷을 재조립할 수 있도록 분할된 패킷들은 같은 식별자를 가짐**

Internet Protocol – 4

- 플래그
  - 첫번째 비트 : 예약 (사용하지 않음)
  - 두번째 비트 : 단편화 금지, 분할해야 하는 경우 폐기
  - 세번째 비트 : 추가 단편화 비트
- 단편화 옵셋(Fragment Offset)
  - 패킷 재 조립 시 분할된 패킷 간의 순서에 대한 정보 포함
  - 분할된 패킷의 상대 위치를 8 바이트 단위로 기술

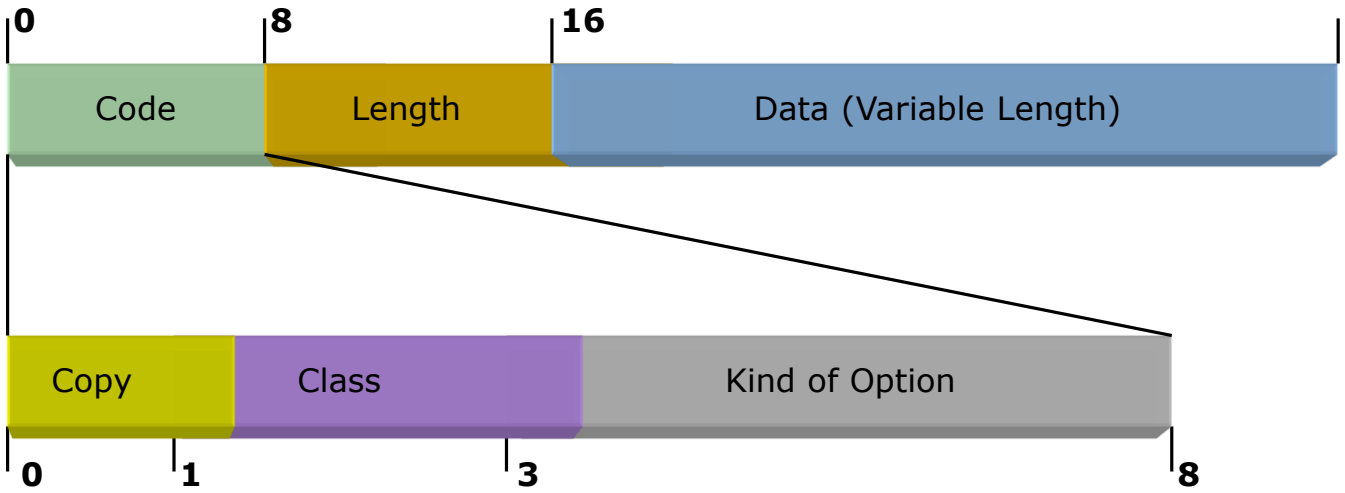


Internet Protocol – 5

- Time-To-Live(TTL)
  - TTL 필드는 패킷이 경유할 수 있는 최대 홉 수를 의미
  - 패킷이 라우터를 통과할 때마다 TTL 값은 1씩 감소
  - TTL 값이 0이 되면 패킷은 폐기
  - 송신측으로 ICMP 메시지가 전달
  - 무한 루프의 방지
  - tracert utility에서 사용
- 프로토콜
  - 관련된 상위 프로토콜을 나타내기 위해 사용
  - TCP: 6
  - UDP: 17
- 헤더 체크섬(Header Checksum)
  - IP 패킷 헤더의 오류 발생을 검사하기 위한 필드

Internet Protocol – 6

- IP 패킷 헤더 옵션
  - 경로 추적, 네트워크 상황 파악 등 특정 목적을 위해 사용
  - 하나 이상의 옵션을 동시 사용 가능
  - 코드 필드
    - 복사(copy) 필드 : 단편화시 단편화된 모든 패킷에 헤더 옵션 필드의 복사 여부를 결정
    - 클래스 필드 : 값이 0인 경우는 패킷 제어를 위해 사용, 2인 경우에는 디버깅이나 측정을 하기 위해 사용



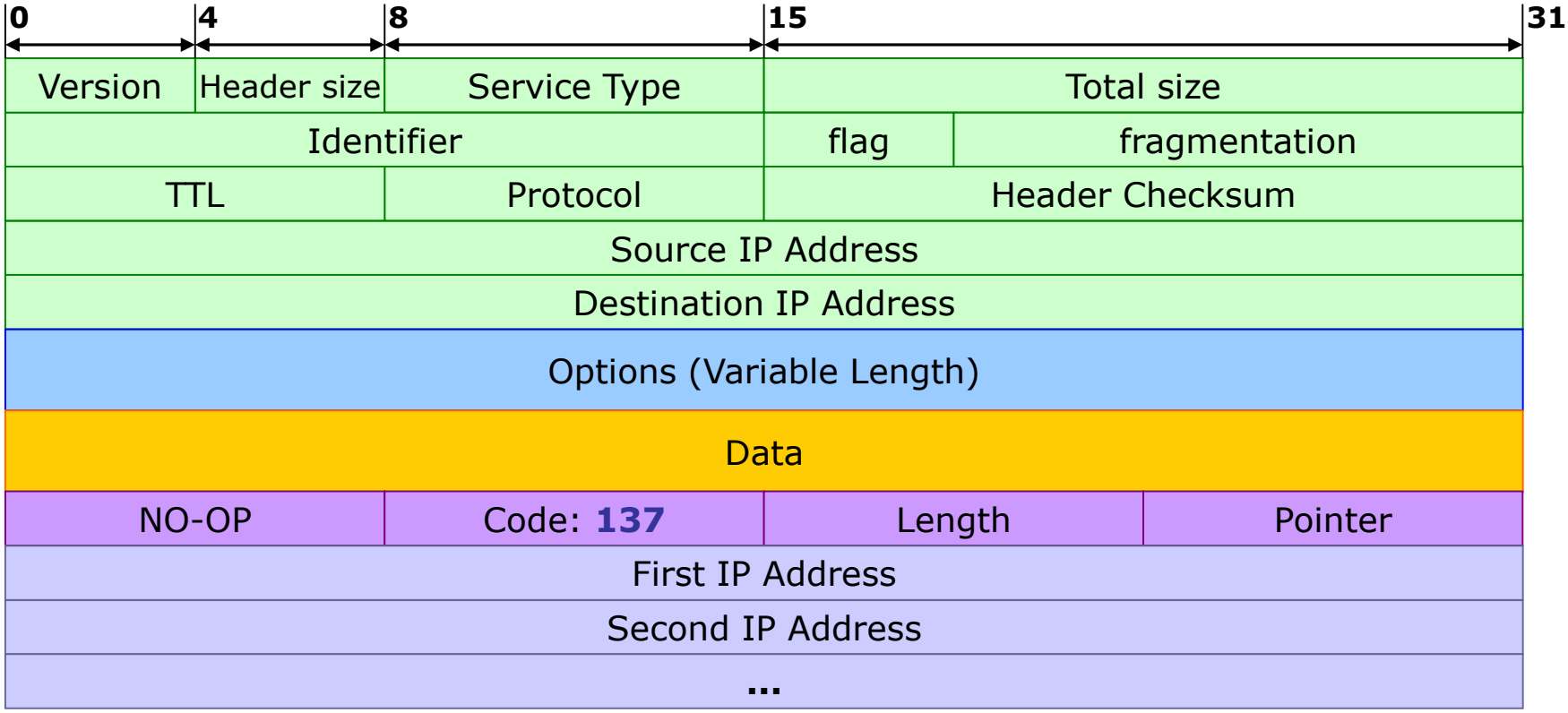
Internet Protocol – 7

- **옵션 종료**(End of Option)
  - **옵션 필드의 끝을 나타내는 한 바이트 옵션**
  - **옵션 종료는 마지막 옵션에서만 사용 가능**
  - **코드 값은 0**

- **무동작**(No Operation)
  - **옵션 간의 경계로 사용되는 한 바이트 옵션**
  - **다른 옵션을 32 비트 경계에 맞추기 위해 사용**
  - **코드 값은 1**

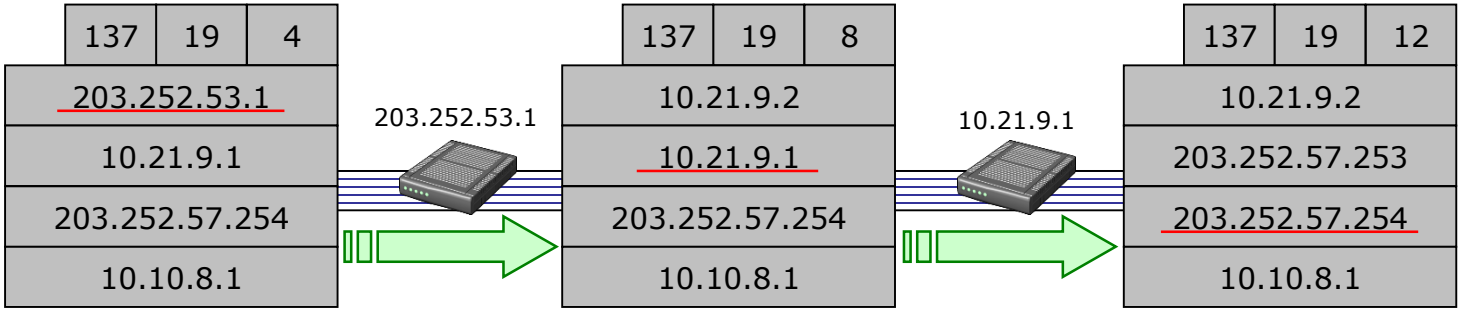
Internet Protocol – 8

- **엄격한 소스 루트**(Strict Source Route)
  - **전달 경로를 송신 측에서 미리 지정하고 그 경로를 따라서만 패킷 전송**



Internet Protocol – 9

- **엄격한 소스 루트**(Strict Source Route)
  - **포인터 값은 홉을 지날 때마다 단계적으로 증가하여 다음 경유할 주소를 가리킴**
  - **기술된 경로대로 패킷을 전송할 수 없다면 패킷을 폐기하고 ICMP 메시지를 송신측으로 전송**



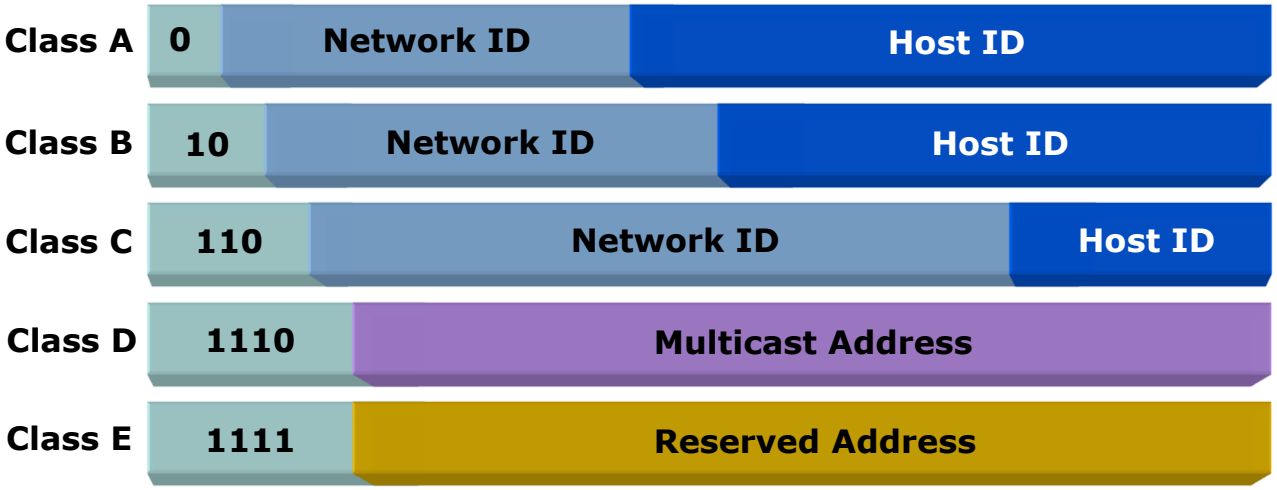
Internet Protocol Addressing – 1

- IP 주소 체제
  - IP 주소
    - 모든 장비들이 갖는 고유한 논리적 네트워크 식별자
    - 32bit로 구성
    - 각 IP 대역에서 기본적으로 3개의 IP는 사용 불가능
      - 0: Network 자체를 가리킴
      - 254: Gateway (Router)를 가리킴
      - 255: Broadcasting 주소
  - 클래스별 분류
    - IP 주소는 네트워크를 구분
      - 네트워크 식별자(Network id)
      - 호스트를 구분하기 위한 호스트 식별자(Host id)로 구성
    - 네트워크, 호스트의 주소 개수에 따라 5개의 클래스로 구분

Internet Protocol Addressing – 2

• 클래스별 분류

클래스	설명
Class A	첫번째 비트가 "0"을 가짐. 2^24 – 2 = 16,777,214 개의 호스트를 수용할 수 있기 때문에 큰 규모의 호스트를 갖는 기관에서 사용한다.
Class B	처음 두비트의 값이 "10"을 가짐. 2^16 - 2개의 호스트를 수용할 수 있다.
Class C	처음 세 비트의 값이 "110"인 주소를 클래스 C 주소라고 한다. 세 번째 바이트까지가 네트워크 식별자이고 마지막 한 바이트는 호스트 식별자이다. 클래스 C 주소는 네트워크마다 254개까지 호스트를 수용할 수 있기 때문에 작은 규모의 네트워크에서 사용된다.
Class D	처음 네 비트의 값이 "1110"인 주소를 클래스 D 주소라고 한다. 클래스 D 주소는 네트워크 식별자와 호스트 식별자의 구분이 없고 전체 주소가 멀티캐스트용으로 사용된다.
Class E	처음 네 비트의 값이 "1111"인 주소를 클래스 E 주소라고 하며 추후 사용을 위해 예약된 주소이다.



Internet Protocol Addressing – 3

–전송 방법에 따른 분류

전송방식	설명
유니캐스트 (Unicast)	하나의 송신자가 하나의 수신자에게 패킷을 보내는 방식의 주소이다.
멀티캐스트 (Multicast)	하나의 송신자가 다수의 수신자에게 패킷을 보내는 경우로 일대다 방식의 패킷 전송 주소이다. 멀티캐스트 전송을 수행하기 위해서는 네트워크 장치가 멀티캐스트를 지원해야 하며, 원하는 그룹에 가입되어 있어야 한다.
브로드캐스트 (Broadcast)	송신자가 네트워크의 모든 호스트에게 패킷을 보내는 방식으로 브로드캐스트 주소에서는 호스트 식별자 필드를 모두 1로 설정한다.

–특별한 IP 주소

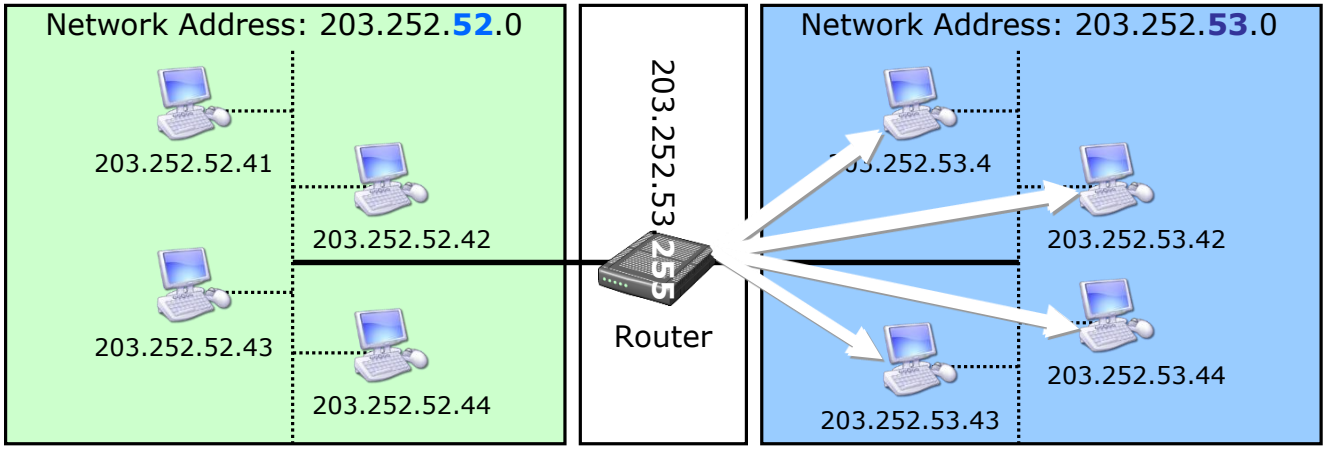
네트워크 식별자	호스트 식별자	목적	송신	수신	용도
네트워크	모두 0	네트워크 주소	불가	불가	Network
네트워크	모두 1	직접적 브로드캐스트	불가	가능	Router
모두 1	모두 1	제한적 브로드캐스트	불가	가능	Host
모두 0	모두 0	네트워크의 한 호스트	가능	불가	DHCP
127	관계 없음	루프백 주소	불가	가능	Test

Internet Protocol Addressing – 4

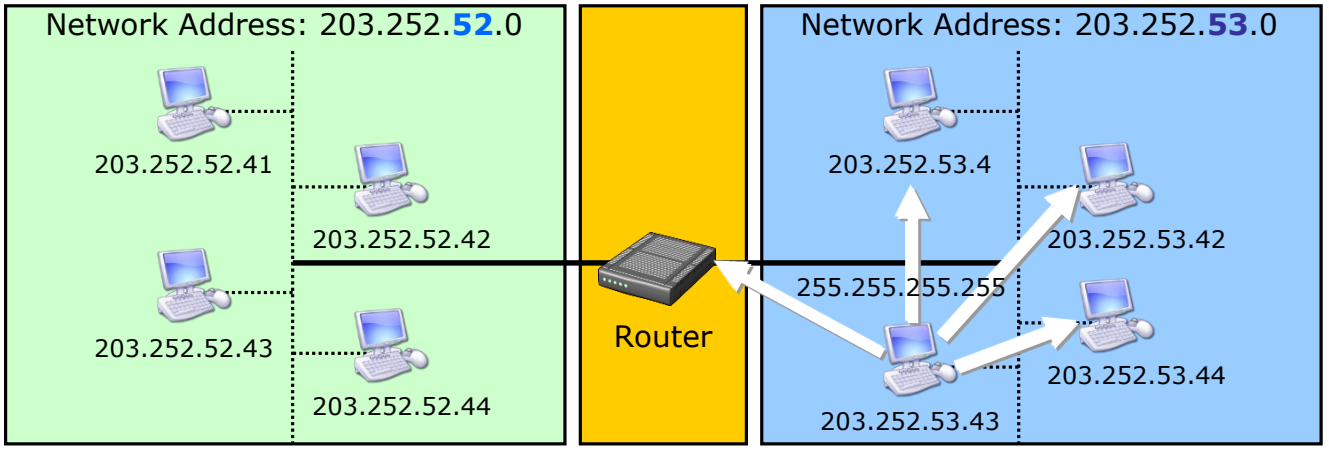
- **특별한 IP 주소**
  - **네트워크 주소**
    - **네트워크 자체**를 의미
    - 라우팅 프로토콜에서 **네트워크를 지칭**할 때 사용
    - 패킷의 송신지나 수신지 주소로써 사용 불가
  - **직접적 브로드캐스트**
    - **라우터**가 네트워크의 모든 호스트로 패킷을 보낼 때 사용
    - 수신 주소로만 사용
  - **제한적 브로드캐스트**
    - 네트워크에 있는 모든 호스트로 패킷을 보낼 때 사용
    - 수신 주소로만 사용

Internet Protocol Addressing – 5

- 직접적 브로드캐스트: 라우터가 직접 전체 호스트에 전송

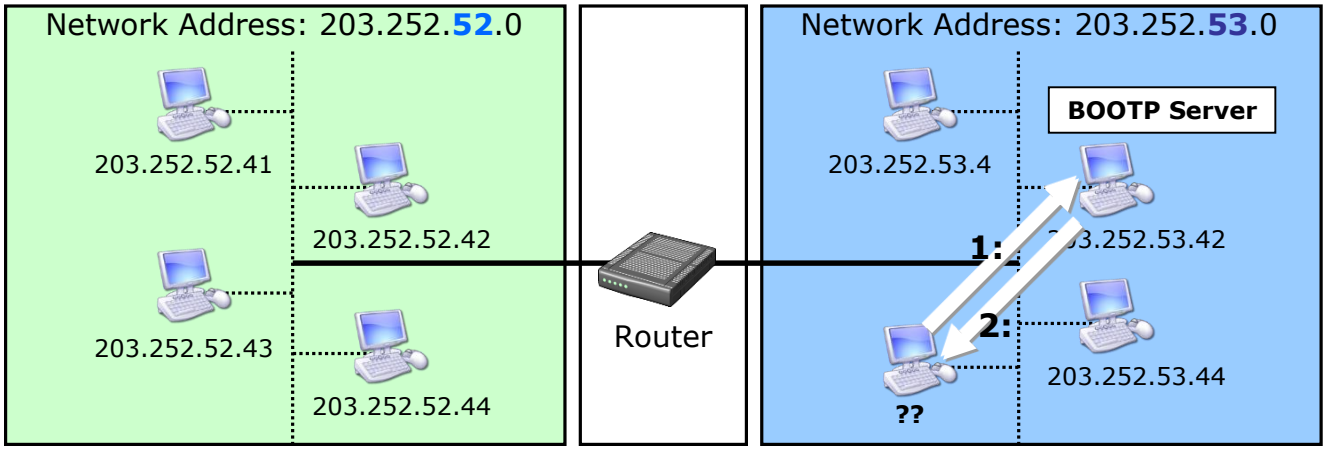


- 제한적 브로드캐스트: 라우터 밖으로 전송되지 않음



Internet Protocol Addressing – 6

- 네트워크의 한 호스트
  - 임의의 호스트를 지칭할 때 사용
  - 송신 주소로만 사용
    - 1: "0.0.0.0"으로 BOOTP Server에 자신의 IP 요청
    - 2: 서버가 IP 정보를 전송



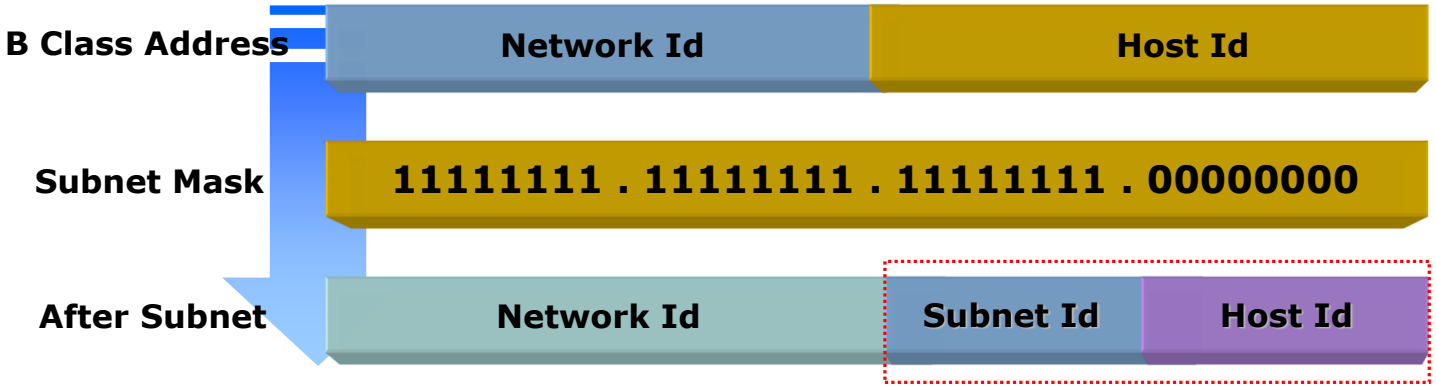
- 루프백 주소
  - 주로 소프트웨어 테스트를 위해 사용
  - 물리 계층까지 전달되지 않고 다시 상위 계층으로 전달

Internet Protocol Addressing – 7

- IP 주소 관리 방식
  - IP 주소 낭비와 부족 현상 심화
    - 클래스별 분류 방식 이외의 IP 주소 관리 방법에 대한 필요성이 제기
  - 새로운 방법 도입
    - 낭비되는 IP를 줄이기 위해 **Subnetting** 개념이 도입
    - 부족한 IP를 효율적으로 사용하기 위해 C 클래스 주소 몇 개를 합쳐 하나의 네트워크 주소로 관리할 수 있는 **Supernetting**이 제안
    - 클래스를 구분하지 않는 주소 방식인 **CIDR** (Classless Inter-Domain Routing)이 제안

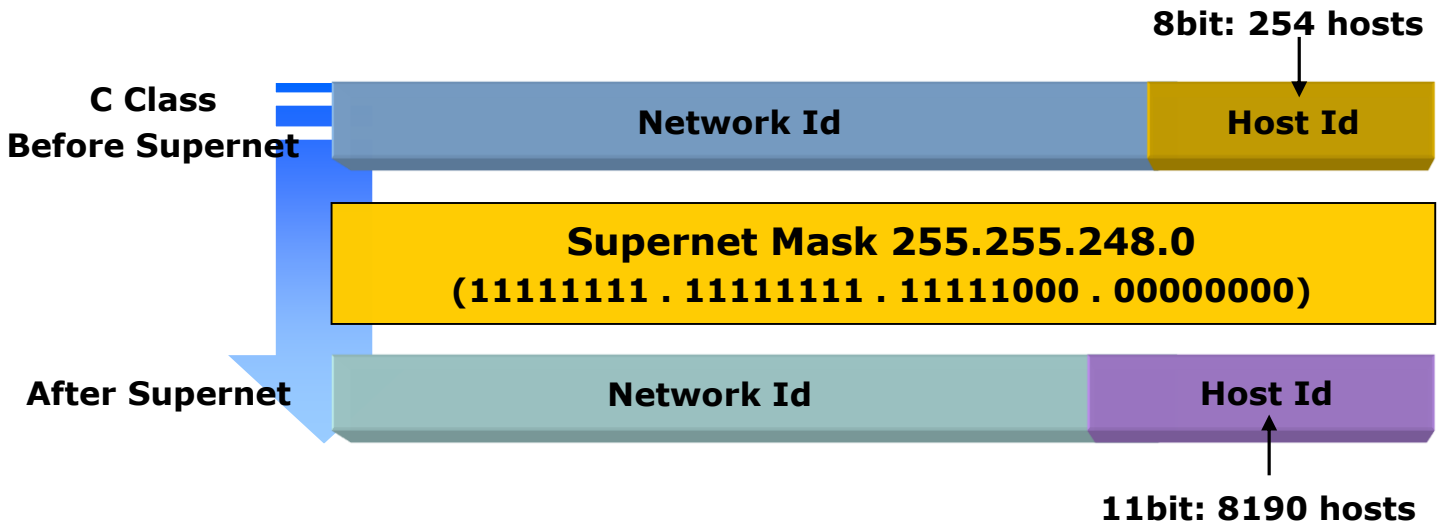
Internet Protocol Addressing – 8

- 서브네팅: **XOR**을 이용
  - 하나의 큰 네트워크를 몇 개의 작은 논리적인 네트워크로 분할하여 사용하는 방식
  - 서브네팅에서는 호스트 식별자를 다시 서브넷 식별자와 호스트 식별자로 세분화
  - Subnet Masking : IP 주소로부터 서브넷 주소만을 식별할 수 있는 방법



Internet Protocol Addressing – 9

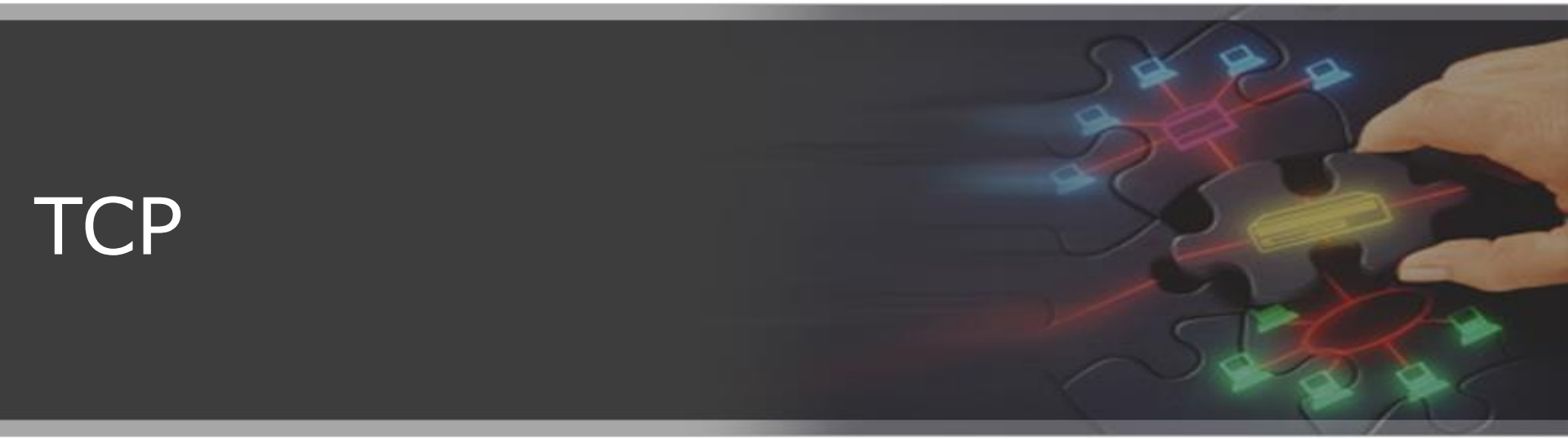
- 슈퍼네팅
  - C 클래스에서 사용할 수 있는 호스트의 수는 254개로 제한
  - 여러 개의 C 클래스 주소를 묶어 하나의 네트워크로 구성하는 슈퍼네팅을 사용
  - 네트워크 식별자 중 일부를 호스트 식별자로 사용하는 방법
  - 통합하는 C 클래스 주소 개수는 2의 지수 승이어야 하며 순차적인 주소를 가져야 함



Internet Protocol Addressing – 10

- **CIDR**
  - CIDR은 A, B, C 클래스별로 IP 주소를 구분하지 않고 네트워크 식별자 범위를 자유롭게 지정할 수 있도록 하여 IP 주소 운영의 융통성을 제공
  - 네트워크 주소를 자유롭게 설정할 수 있기 때문에 IP 주소의 낭비를 방지하고 효과적으로 네트워크를 구성
  - CIDR을 이용하면 도메인간의 라우팅에 사용되는 IP 주소를 매우 효과적으로 관리 가능
- IP **알리어스**(IP Alias)
  - **하나의** NIC(Network Interface Card)**에 여러 개의** IP **주소를 할당하는 방법**
  - **한 대의 서버에 여러 개의** IP **주소를 할당해야 할 경우 하나의 이더넷 카드에 여러 개의** IP **주소를 할당하여 운영할 수 있음**

TCP



About..

컴퓨터소프트웨어공학과
김 원 일

TCP (Transmission Control Protocol)

- TCP 특징
  - 연결형(**Connection-Oriented**)
    - IP 계층 위에 가상의 회선을 설정
    - 종단간 데이터 송수신 서비스 제공
  - 신뢰성(**Reliability**)
    - 확인을 통한 신뢰성 있는 통신서비스 제공
  - 흐름제어(**Flow Control**)
    - 처리할 수 있는 범위 내의 데이터를 보내도록 제어
  - 혼잡제어(**Congestion Control**)
    - 혼잡 현상을 방지하거나 제어하는 기능
  - 스트림 통신
    - 데이터를 바이트 단위로 나눠서 전송

TCP Port and Socket – 1

- TCP 포트
  - 프로세스를 구분하기 위한 방법
  - 16비트로 표현 (0~65,535)
  - 잘 알려진 포트와 임시 포트로 구분
    - well-known port
    - ephemeral port
  - 잘 알려진 포트
    - 응용프로그램에 따라 정해진 포트
    - 보통 서비스를 제공하는 서버에서 사용
    - 범위 : 1~1,023
  - 임시 포트
    - 1023~65,535 값 중 운영체제로부터 할당 받음
    - 서비스 사용 중에만 유효

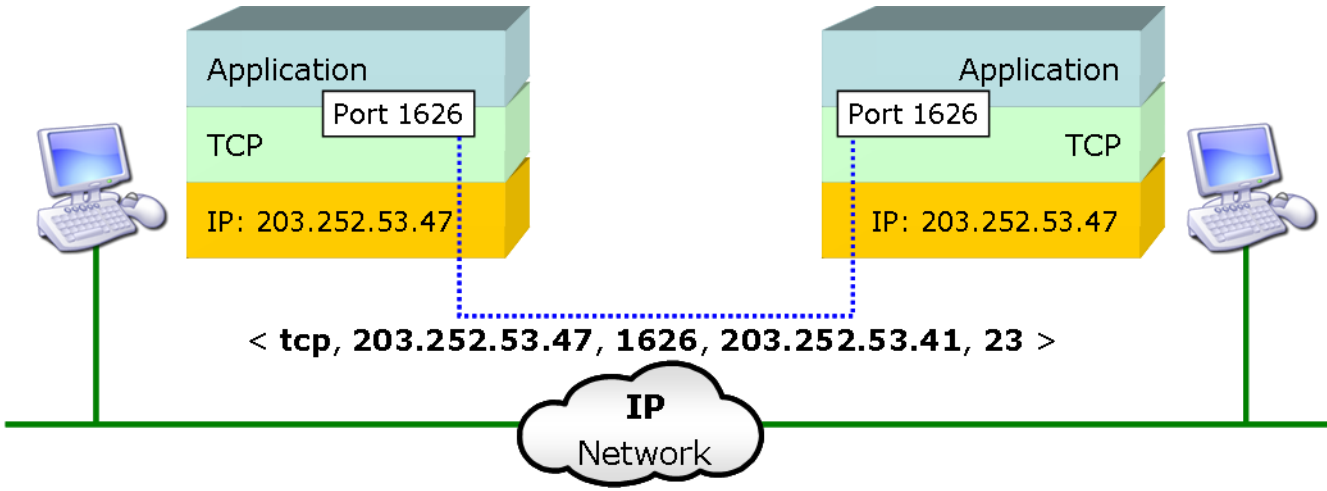
TCP Port and Socket – 2

- Well-Known Port Number

프로토콜	포트번호	설 명
Echo	**7**	**수신한 데이터를 송신자에게 되돌려준다.**
Discard	**9**	**수신한 데이터를 모두 폐기한다.**
Users	**11**	**활동중인 사용자를 나타낸다.**
DayTime	**13**	**날짜와 시간을 알려준다.**
FTP, Data	**20**	**파일 전송 프로토콜**(File Transfer Protocol), **데이터전송**
FTP, Control	**21**	**파일 전송 프로토콜, 제어 전송**
TELNET	**23**	**터미널 네트워크**(TErnimaL NETwork)
SMTP	**25**	**간단한 메일 전송 프로토콜**(Simple Mail Transfer Protocol)
DNS	**53**	**도메인 네임 서버**(Domain Name Server)
BOOTP	**67**	**부트스트랩 프로토콜**(BOOTstrap Protocol)
Finger	**79**	**사용자의 정보 제공**
HTTP	**80**	**하이퍼텍스트 전송 프로토콜**(Hypertext Transfer Protocol)
RPC	**111**	**원격 함수 호출**(Remote Procedure Call)

TCP Port and Socket – 3

- TCP 소켓
  - TCP/IP 통신 인터페이스중의 하나
  - 소켓 주소
    - <프로토콜, 로컬 IP 주소, 로컬 프로세스 번호>
  - 두 프로세스의 소켓 식별
    - <프로토콜, 로컬 IP 주소, 로컬 포트 번호, 원격지 IP 주소, 원격지 포트 번호>
  - 두 프로세스간에 전 이중 바이트 스트림 연결 제공



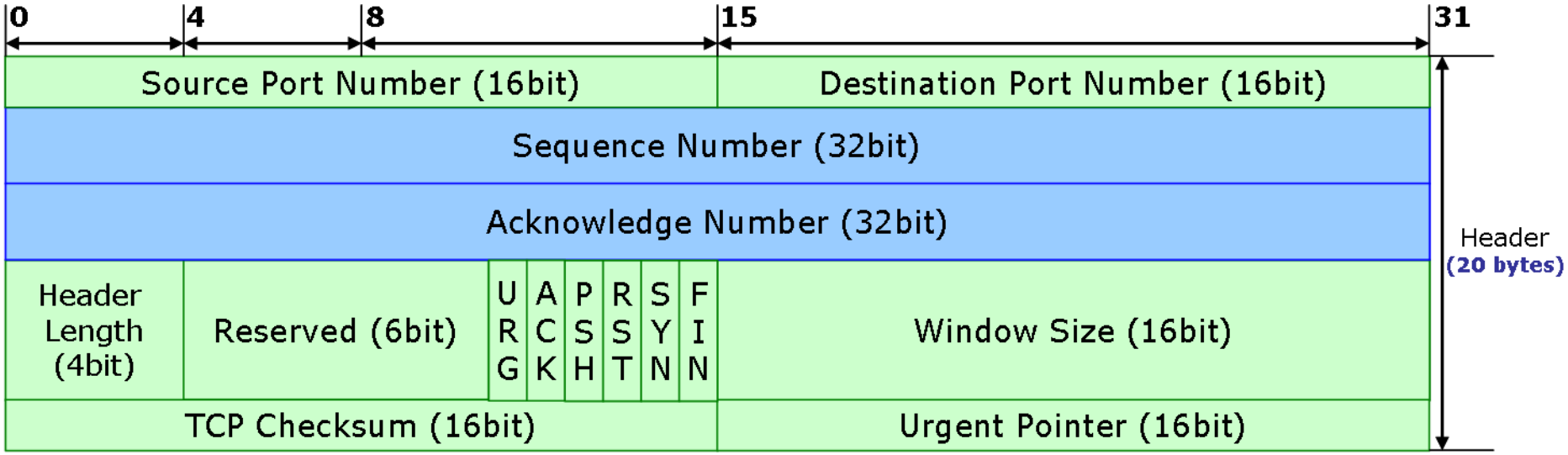
TCP Port and Socket – 3

- **소켓 시스템 함수**

함수 명	설 명
socket	소켓을 생성하는 프로세스 소켓을 생성함에 있어 패밀리, 종류(type), 프로토콜을 먼저 지정 생성된 소켓은 소켓 지시자(socket descriptor)라고 불리는 유일한 정수 값을 통해 구별
bind	bind 함수는 socket 함수로부터 생성된 소켓 지시자에 주소를 지정
connect	connect 함수는 원격지에 연결을 하기 위해 사용한다. connect 함수는 보통 클라이언트가 서버에 접속하기 위해 소켓에 서버의 주소를 지정하여 사용한다.
listen	listen 함수는 TCP 서버에 의해 사용 동 열림(Passive Open)을 생성 socket 함수가 생성되어 지역 주소가 바인딩 된 상태에서 listen 함수가 호출 생성된 소켓을 통하여 연결을 수용할 수 있음을 나타냄
read	열린 연결을 통하여 데이터를 수신
write	프로세스가 다른 프로세스로 데이터를 전송할 때 사용
close	소켓을 종료하고 TCP 연결을 마침

TCP Message – 1

- TCP 헤더
  - TCP 헤더 형식
    - 소스 포트번호
      - 세그먼트를 전송하는 응용 프로그램의 포트번호
    - 목적지 포트번호
      - 세그먼트를 수신하는 응용 프로그램의 포트번호
    - 순서번호
      - 데이터 스트림의 내부적인 순서를 구분
      - 32비트의 부호 없는 번호, 2^32-1 값 초과시 다시 0부터 시작



TCP Message – 2

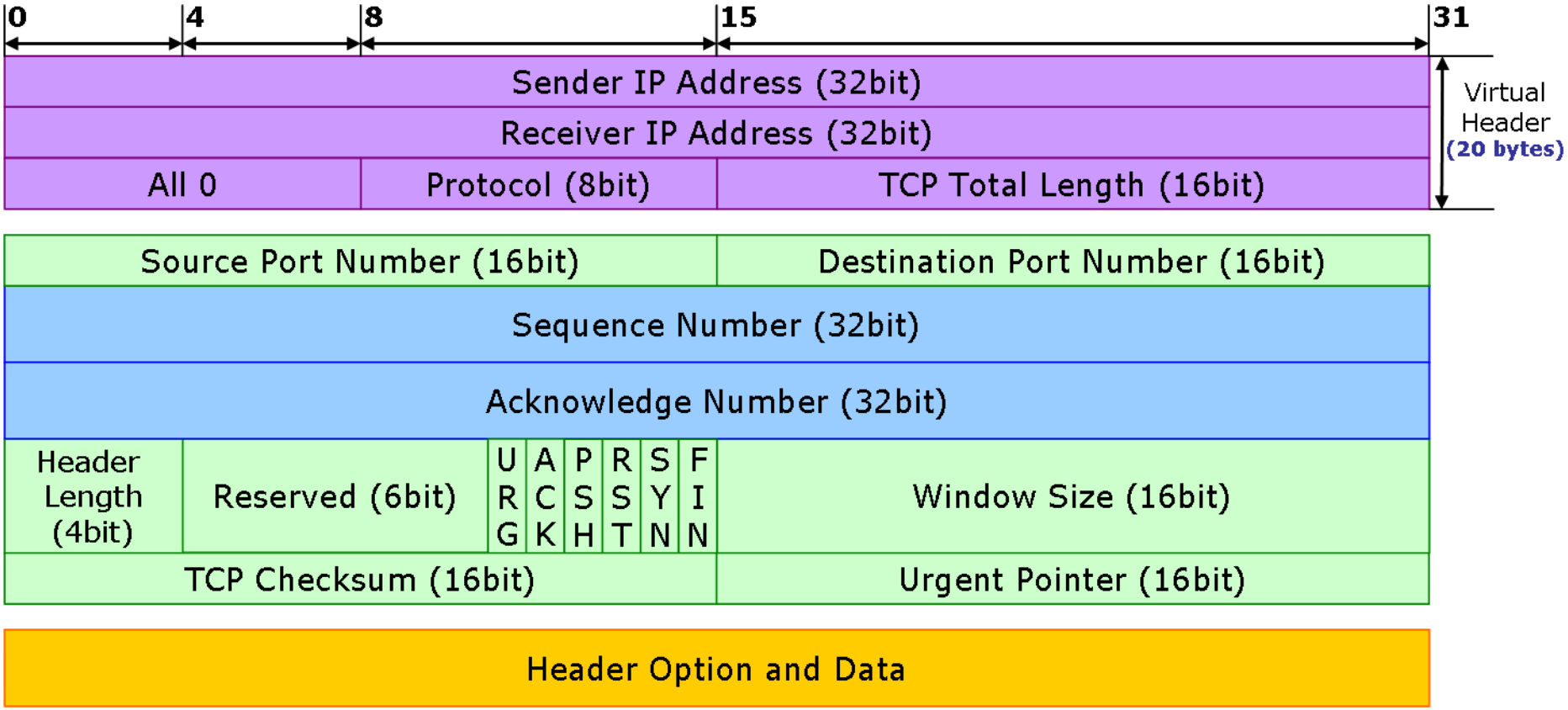
- **확인응답번호**
  - 앞으로 받고자 하는 바이트의 순서번호
- **헤더 길이**
  - 4바이트 단위로 최대 길이는 60바이트
- **플래그 비트**

플래그	의 미
URG	**긴급 포인터**(urgent pointer)**가 있음을 표시**
ACK	**확인응답번호**(acknowledgment number)**가 유효함**
PSH	**데이터를 가능한 빨리 응용계층으로 보내야 함**
RST	**연결을 재설정**
SYN	**연결을 초기화하기 위해 순서번호**(sequence number)**를 동기화**
FIN	**송신측이 데이터 전송을 종료함**

- **윈도우 크기**
  - 흐름 제어를 위해 사용
  - 수신측은 수신 가능 데이터 양을 바이트 단위로 송신측에 통지
  - 16비트로 최대 크기는 65,535바이트
  - 옵션 필드에 scale factor를 이용하여 더 크게 지정 가능

TCP Message – 3

- **체크섬**(Checksum)
  - TCP **헤더와 데이터에 수행**
  - **수신측에서 에러를 검출**



TCP Message – 4

- **긴급포인터**
  - **URG 플래그가 설정되어 있을 때만 유효**
  - **사용 목적은 주로 응용 프로그램에 달려 있음**
  - **송신측**
    - **먼저 전해져야 하는 데이터를** URG **플래그를 설정**
    - **긴급포인터로 긴급 데이터의 마지막을 지정하여 전송**
  - **수신측**
    - **긴급 포인터의 데이터까지 "긴급 모드(urgent mode) "로 응용 프로그램에 알림**
  - **긴급포인터 사용의 예**
    - FTP **전송 중에 송신자가** <Ctrl+C>**를 눌러 전송을 중지시켰을 경우**

TCP Message – 5

- TCP **헤더 옵션**
  - **옵션의 끝**
    - **코드** : 0x00
    - **마지막 옵션임을 나타냄**
  - **무작동**
    - **코드** : 0x01
    - **한 바이트로 옵션들 사이에 패드 필드로 사용**
  - **최대 세그먼트 크기** (MSS : Maximum Segment Size)
    - **코드** : 0x02
    - 16**비트로 최대 세그먼트 크기는** 65,535**바이트**
    - **연결 설정 시 양단간에 합의에 의해 결정**

TCP Message – 6

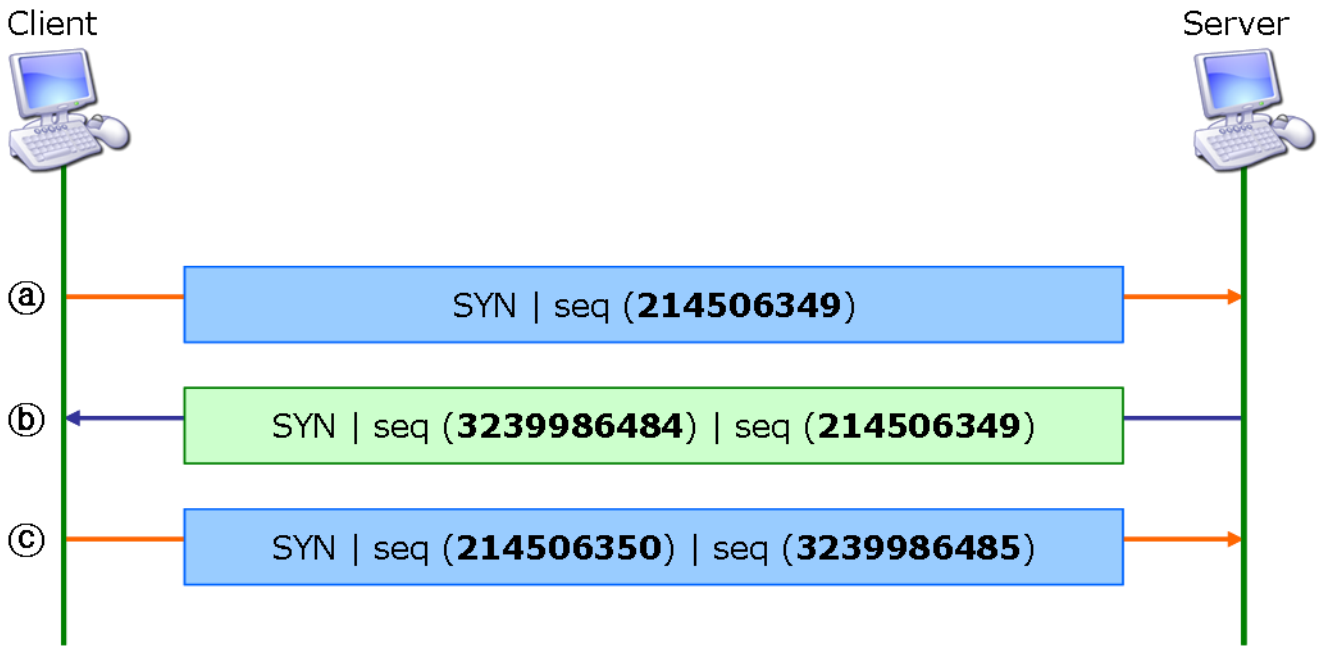
- 윈도우 스케일 팩터
  - TCP 헤더의 윈도우 크기는 16비트(65,535바이트)로 제한
    → 고속 네트워크에서 처리량을 최대화 하기 위해서는 부족
  - 스케일 팩터를 두어 윈도우 크기를 2배씩 증가 시킬 수 있음
  - 새로운 윈도우 크기 = 헤더에 정의된 윈도우 크기 * 2^윈도우 스케일 팩터
- 타임스탬프
  - 타임스탬프 옵션은 코드와 길이 1바이트 , 타임스탬프 값(timestamp value)와 타임스탬프 에코 응답(timestamp echo reply)가 각각 4바이트로 총 10바이트의 크기
  - 송신측: 전송 시의 시간을 타임스탬프 값에 채움
  - 수신측: 확인응답을 전송 시 타임스탬프 에코 응답 필드에 원래의 타임스탬프 값을 넣어 전송
  - RTT(Round-Trip Time) : 확인 응답 메시지를 수신한 송신측은 현재의 시간과 타임스탬프 에코 응답에 기록된 시간의 차로써 구함

TCP Connection – 1

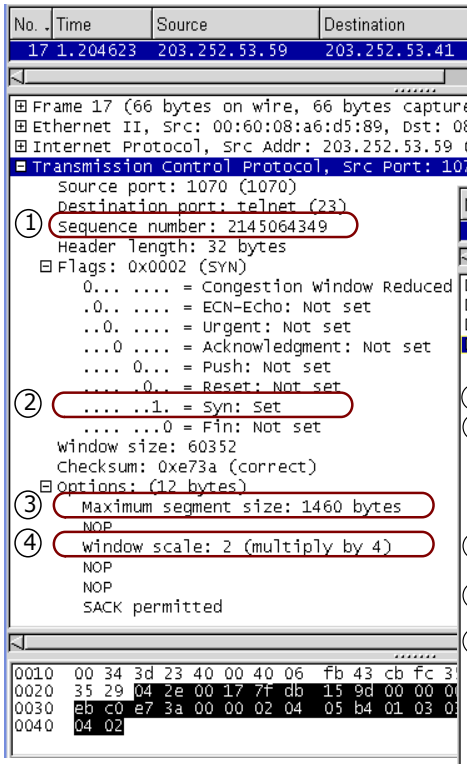
- TCP **연결 설정**
  - **능동적 열림**(Active open)
    - **클라이언트 서버가 열고 있는 포트로 응용을 요청**
    - **보통 처음에** SYN**을 보내는 쪽**
  - **수동적 열림**(Passive Open)
    - **보통 서버는 네트워크 응용을 수행하기 위해 정해진 포트를 열고 클라이언트의 요청을 기다림**

TCP Connection – 2

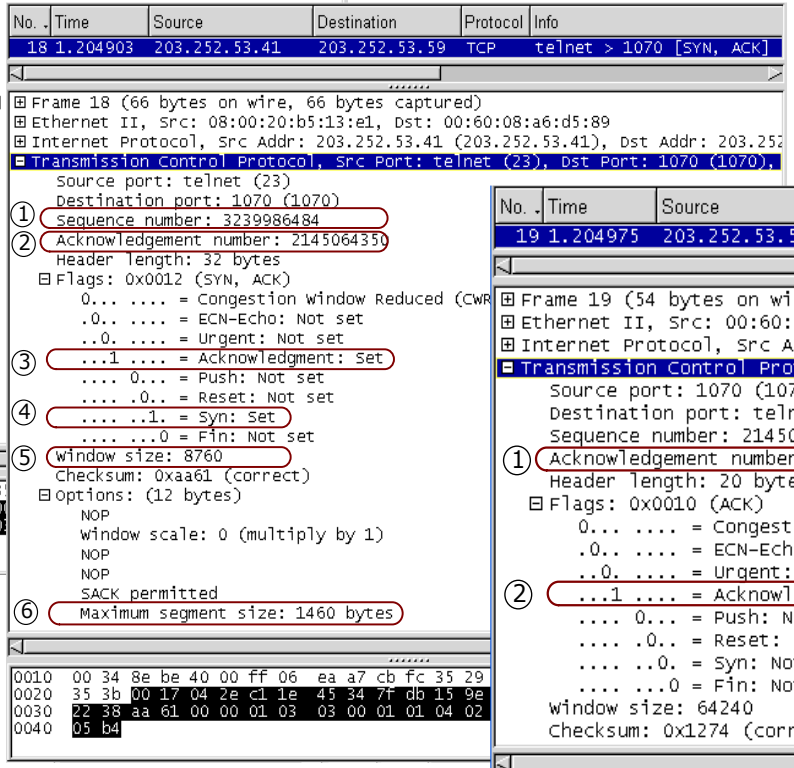
- **Three-way handshake**
  - ⓐ **클라이언트는** TCP **세그먼트에** SYN **비트를 설정**, ISN (Initial Sequence Number) **를 보냄**
  - ⓑ **서버측에서는 이에 대한 수락으로** SYN**과 클라이언트의 순서번호에** 1**을 더한** ISN**값을 갖는** ACK**를 보냄**
  - ⓒ **클라이언트는** ⓑ**를 수신한 후 서버에게** ACK**를 보냄으로써 양방향 연결이 성립**



TCP Connection – 3

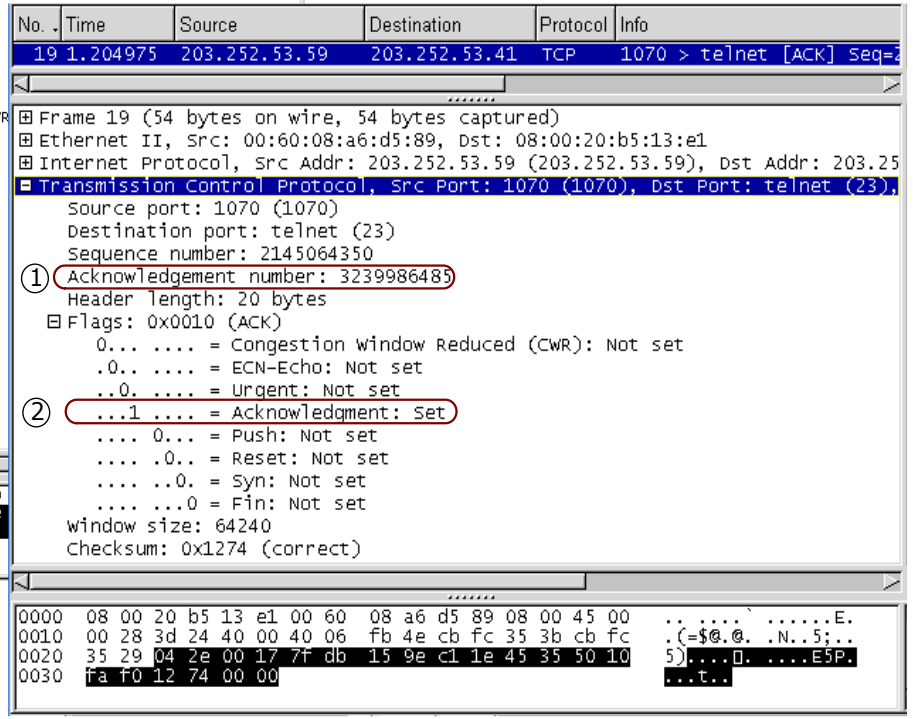


<Request>



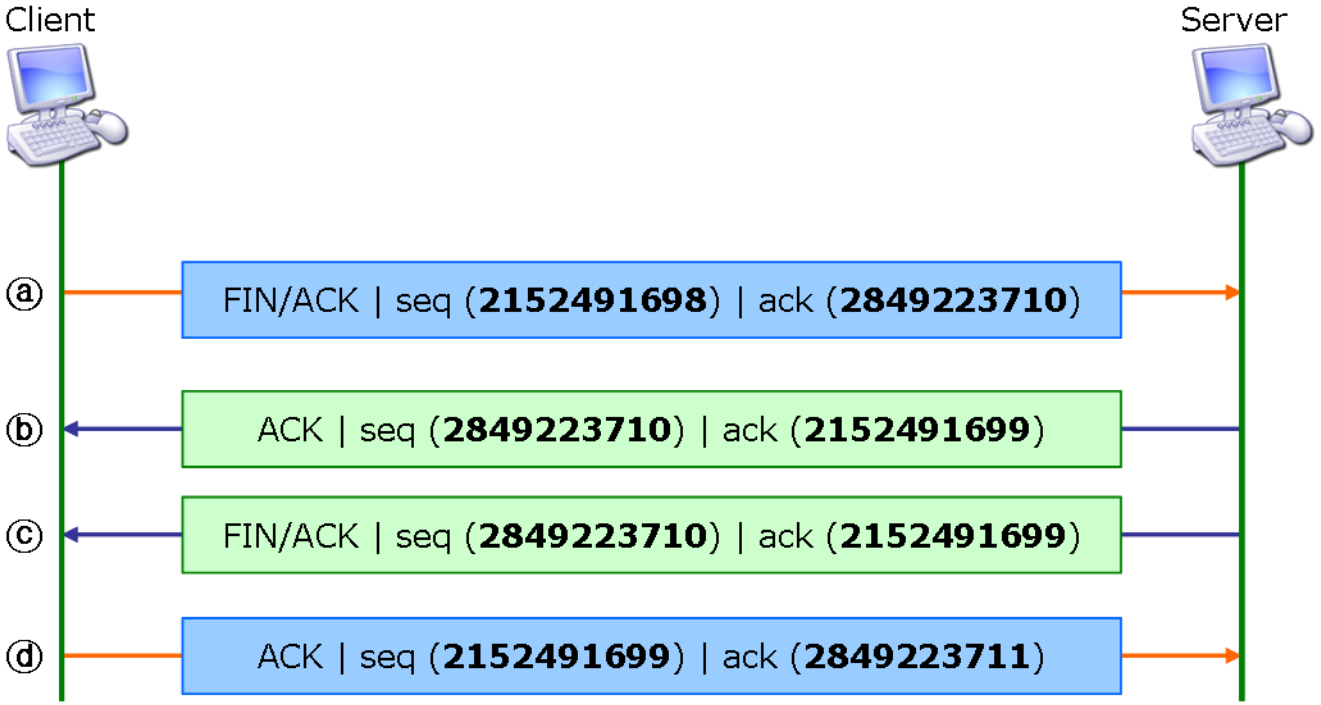
<Response>

<Connection>



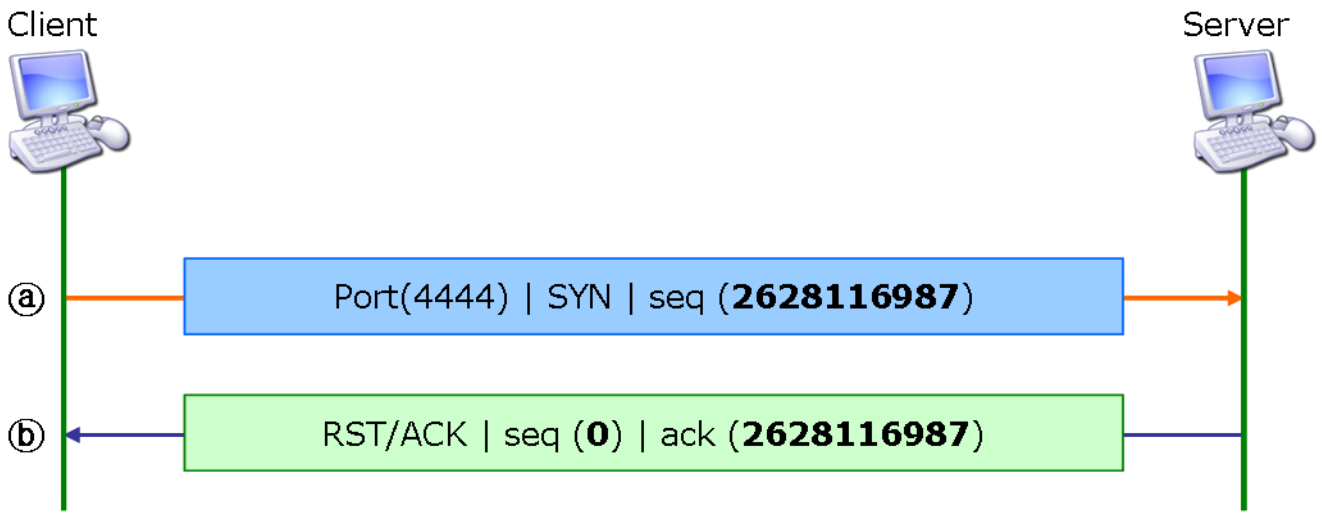
TCP Connection – 4

- **Four-way handshake**
  - ⓐ : 연결을 종료하려는 송신측은 FIN 비트를 설정하여 전송
  - ⓑ : 수신측에서 확인응답 메시지 전송으로 연결 종료.
  - ⓒ : 수신측은 FIN 비트를 설정하여 송신측으로 전송
  - ⓓ : 송신측이 이 세그먼트에 확인 응답함으로써 수신측에서 송신측으로의 연결 또한 완전히 종료



TCP Connection – 5

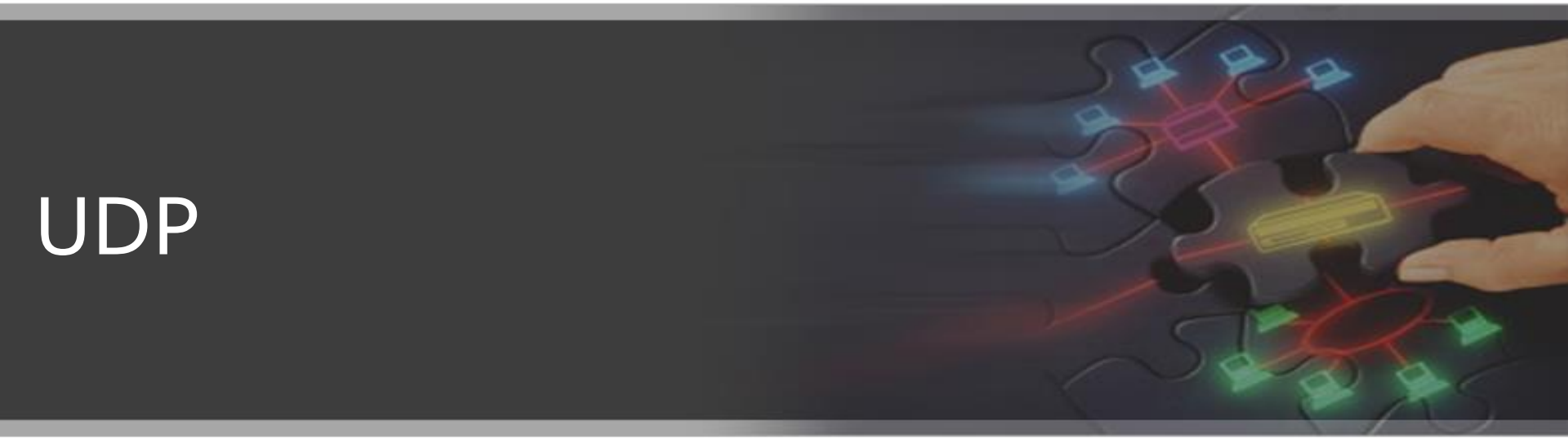
- TCP **연결 리셋** (Reset)
  - **존재하지 않는 포트에 연결 요청 시**
    - SYN **비트가 설정되어 있는 세그먼트를 수신하였으나 목적지 포트가 존재하지 않는 경우**
    - FTP **접속을 포트번호** 4444**로 할 때 수행되는 결과**
      - ⓐ : **송신측이 수신측에 존재하지 않는 포트로 연결을 요청**
      - ⓑ : **수신측은** RST **비트 설정, 순서 번호를** 0**으로 설정한 세그먼트를 송신측으로 보내어 연결이 종료되었음을 알림**



TCP Connection – 6

- **비정상적인 종료를 요청 시**
  - **비정상적으로 종료를 해야 하는 경우 FIN 대신 RST 비트를 이용 연결 중단**
  - **클라이언트과 서버의 텔넷 연결 상태에서 클라이언트측의 텔넷 프로세스를 [ALT+E]하여 강제 종료**

UDP



About..

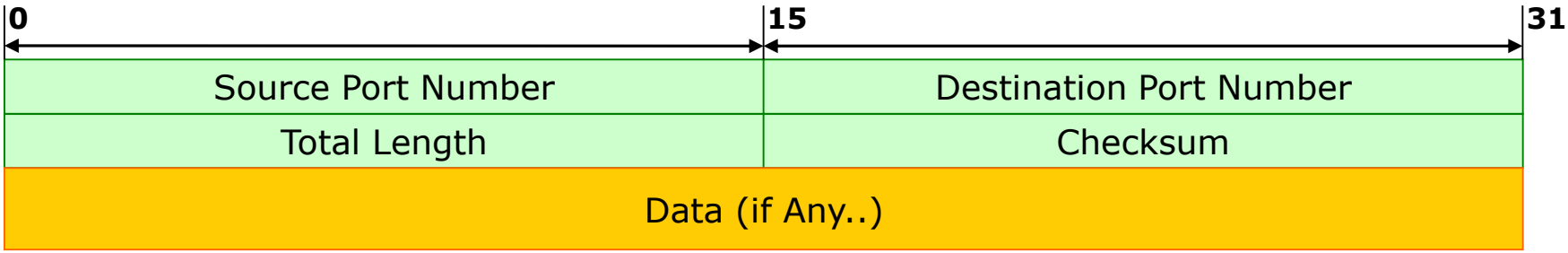
컴퓨터소프트웨어공학과
김 원 일

User Datagram Protocol – 1

- **비 연결형**(Connectionless)
  - VS. TCP는 **데이터를 전송하기 전에 연결을 설정**
- **비 상태정보**(Non-state)
  - VS. TCP **종단 시스템에서는 각 연결 상태정보를 유지**
- **비정규적인 송신률**(Unregulated Send Rate)
  - **일부 패킷 손실이 발생하더라도 지속적인 최소 전송률을 요구하는 실시간 영상서비스에 적합**
- **최선형 서비스**(Best Effort service)
  - **수신확인 및 재전송 기능** 無

User Datagram Protocol – 2

- **경량의 오버헤드**(Small overhead)
  - TCP Segment Header: 20**바이트**
  - UDP Header : **8바이트**
  - **송신 포트번호**
    - **송신측 프로세스에서 사용되는 포트번호**
  - **수신 포트번호**
    - **수신측 프로세스에서 사용되는 포트번호**
  - **전체 길이**
    - **헤더 길이 + 사용자 데이터 그램의 전체 길이**
  - **체크섬**
    - **에러 검출을 위해 사용**



User Datagram Protocol – 3

- **비연결형 서비스**
  - UDP를 통해 전송되는 모든 데이터그램들은 **독립적** 취급
  - 512**바이트** 이하의 데이터그램을 전송하는 것이 효과적
- **흐름/에러 제어**
  - TCP에서 사용하는 윈도우 메커니즘이 없기 때문에 흐름제어 수행 못함
  - 수신 측에서는 체크섬을 이용, 데이터그램의 오류 검사
    - If 에러 발생, **discard**
    - 송신 측은 메시지가 전송 중에 폐기된 사실을 인지 못함