

# TCP/IP 완벽 가이드

- II-8부 전송 계층 프로토콜 개요와 UDP -

임연주 ([yeonjoo@pel.smuc.ac.kr](mailto:yeonjoo@pel.smuc.ac.kr))

상명대학교 프로토콜공학연구실

# 목 차

---

- TCP와 UDP 개요와 비교
- TCP와 UDP 주소지정: 포트와 소켓
- TCP/IP 사용자 데이터그램 프로토콜(UDP)

# TCP와 UDP 개요와 비교

---

- 개요
  - IP 계층의 제약
    - 비연결형이고, 신뢰성이 없으며, 승인을 하지 않음
  - TCP/IP 모델에서 많이 쓰이는 전송 계층 프로토콜은 두 가지가 있음
    - TCP (Transmission Control Protocol)
      - 애플리케이션이 네트워크 계층 문제를 걱정하지 않고 데이터를 안정적으로 송신을 보장
    - UDP (User Datagram Protocol)
      - 단순하고, 빠르며, 애플리케이션이 네트워크 계층에 접근할 수 있도록 하는 인터페이스만 제공할 뿐 다른 것은 거의 하지 않음

# TCP와 UDP 개요와 비교

## • TCP와 UDP 비교 표

특성/설명	TCP	UDP
연결 수립	연결형	비연결형
데이터 입력 인터페이스	스트림 기반	메시지(데이터그램) 기반
신뢰성과 승인	모든 데이터에 승인과 신뢰성 제공	신뢰성과 승인 미제공
재전송	모든 데이터를 관리해 자동적으로 재전송	탐지 후 필요할 경우에만 수동적으로 재전송
데이터 흐름 관리	슬라이딩 윈도우를 이용해 제공	없음
부하	낮지만 UDP보단 높음	매우 낮음
전송 속도	빠르지만 UDP보단 느림	매우 빠름
적합한 데이터 양	소형에서 초대형	소형에서 중형
프로토콜을 사용하는 앱의 유형	신뢰할 수 있는 방법으로 전송해야 하는 대부분의 앱	데이터의 완전성보다 전달 속도가 중요하거나, 소량의 데이터를 전송하는 경우, 멀티캐스트/브로드캐스트를 사용하는 앱
관련 유명 앱과 프로토콜	FTP, HTTP, DNS, BGP	멀티미디어 앱, DNS, BOOTP, DHCP, RIP

# TCP와 UDP 개요와 비교

---

- TCP와 UDP 애플리케이션
  - TCP 애플리케이션
    - 신뢰성과 여러 서비스를 필요로 하는 대부분의 애플리케이션이 사용하고 있음, 그에 따르는 성능 부하가 존재
  - UDP 애플리케이션
    - 애플리케이션 자체에서 UDP의 부족한 기능을 보충
    - 매우 적은 양의 데이터를 전송하거나 데이터 일부가 손실 돼도 중요치 않는 애플리케이션이 사용하고 있음
- TCP와 UDP는 매우 다른 방식으로 동작
  - 개발자에게 유연성 제공

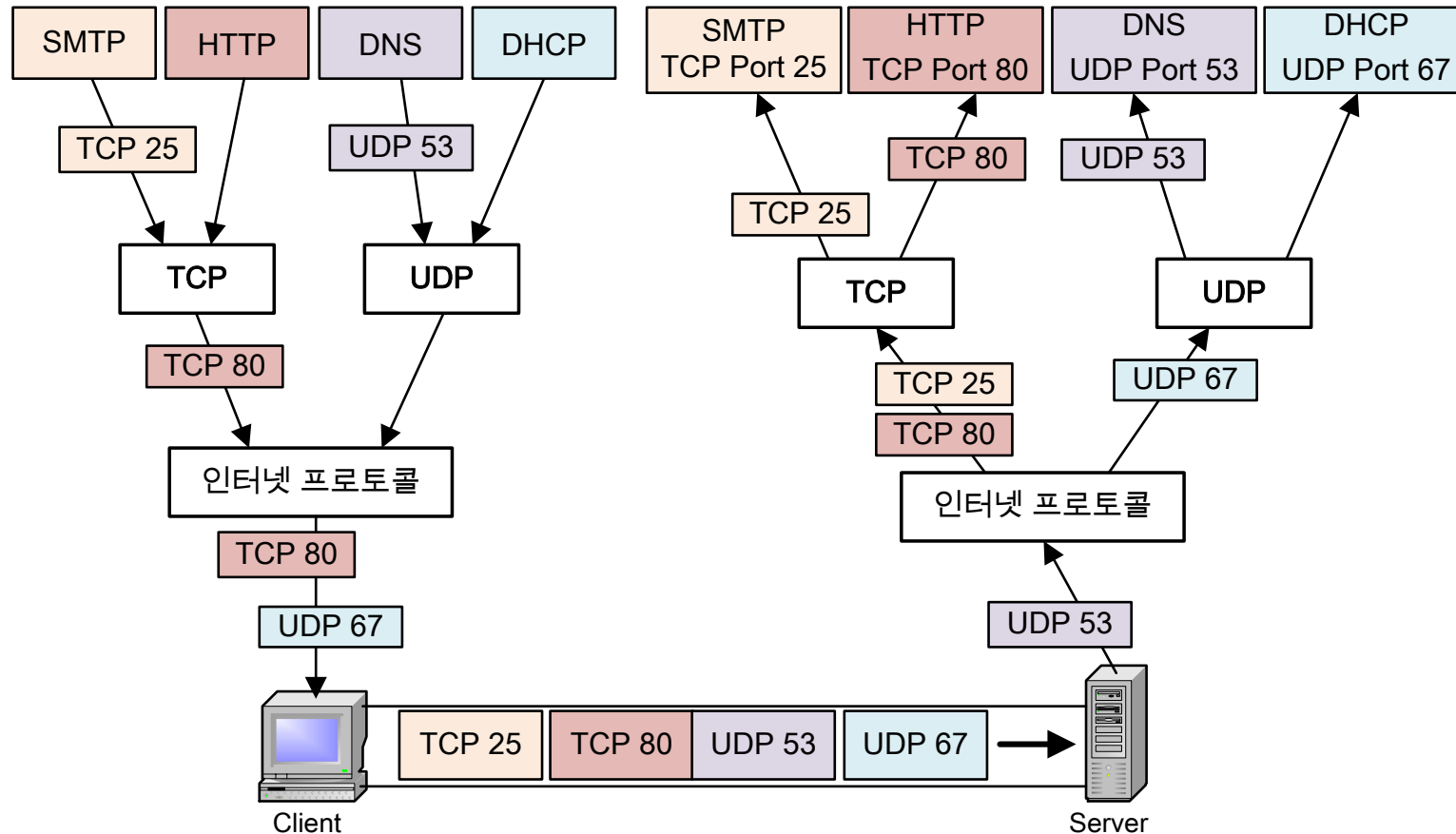
# TCP와 UDP 주소지정: 포트와 소켓

---

- 프로세스 수준 다중화, 역다중화: 소켓
  - 하나의 IP를 가지고 여러 애플리케이션과 통신하는 경우가 많음
    - 프로세스의 동시성 제공
- 여러 애플리케이션에서 온 데이터를 다중화, 역다중화
  - 다중화
    - 발신지 전송계층에서 서로 다른 데이터를 모아 다중화하여 네트워크로 전달
  - 역다중화
    - 수신지 전송계층에서 서로 다른 데이터를 소켓을 가지고 알맞은 프로세스에 전달

# TCP와 UDP 주소지정: 포트와 소켓

- TCP/UDP 포트를 이용한 통신 과정 그림



- 다중화/역다중화는 IP의 프로토콜 필드와 TCP/UDP의 출발지 포트/목적지 포트 필드를 통해 구현

# TCP와 UDP 주소지정: 포트와 소켓

---

- TCP/IP 포트: TCP와 UDP 주소지정
  - IP 주소 내에서 좀더 구체적인 소프트웨어 프로세스들을 식별을 위해 포트번호 필드를 추가
    - 16비트 필드로 0~65535까지 값을 가짐
    - 각각의 목적을 위해 구분됨 (예약 영역, 임시 영역 등)
- TCP와 UDP의 포트번호는 같은 범위를 사용하지만 독립적
  - IP 패킷의 프로토콜 필드에 TCP/UDP를 명시



# TCP와 UDP 주소지정: 포트와 소켓

---

- TCP/UDP 포트번호 범위
  - IANA (Internet Assigned Numbers Authority)에서 관리
- 유명 포트번호 (0~1023)
  - 가장 범용적인 애플리케이션을 위해 번호 예약
  - RFC 절차를 통해 표준화 되거나 될 가능성이 있는 프로토콜에게만 할당
- 등록 포트번호 (1024~49151)
  - 애플리케이션 개발자가 자유롭게 포트번호를 요청할 수 있음
  - IANA에서 승인 할 경우 포트 번호를 등록 후 할당

# TCP와 UDP 주소지정: 포트와 소켓

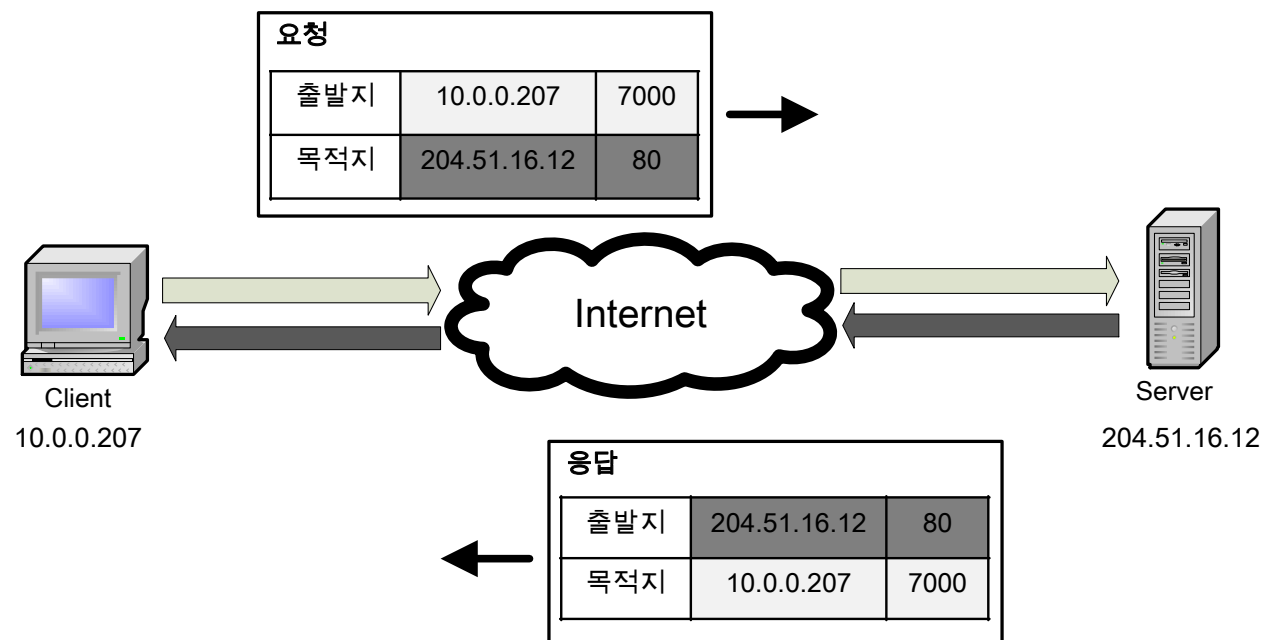
---

- TCP/UDP 포트번호 범위
  - 사설/동적 포트번호 (49152~65535)
    - IANA는 관리하지 않음, 누구나 등록 없이 사용가능
    - 특정 기관에서만 사용하는 사설 프로토콜에 적합
- TCP/IP 클라이언트(임시) 포트
  - 서버 소프트웨어: 유명 포트번호(예약됨)를 사용
    - 어떤 클라이언트든 서버의 포트번호를 알아야 함
  - 클라이언트 프로세스: 어떤 포트 번호든지 사용할 수 있음
    - 주로, 임시(Ephemeral) 번호를 사용
      - 서버/클라이언트 역할을 다 할 수 있는 장비 경우, 구분하기 어려움

# TCP와 UDP 주소지정: 포트와 소켓

- 클라이언트/서버 애플리케이션 포트
  - 임시 포트번호 할당
    - 프로세스가 필요로 할 때 마다 예약된 번호 풀에서 랜덤 방식으로 번호를 할당
      - 충돌을 최소화하기 위해 시간 간격을 극대화해서 풀을 순환
  - 운영체제마다 임시 포트번호에 사용하는 번호가 다를 수 있음

## • 동작 방식 그림



# TCP와 UDP 주소지정: 포트와 소켓

---

- 프로세스와 연결 식별: 소켓
  - 애플리케이션 프로세스와 통신하기 위해서 네트워크 인터페이스 주소와 포트번호를 알아야 함
  - 두 가지가 조합된 주소를 소켓이라 칭함
    - <IP 주소>:<포트 번호>로 표현
  - 전체적인 장비간의 연결은 소켓 쌍으로 표현 할 수 있음
    - e.g., (41.199.222.3:80, 177.41.72.6:30220)

# TCP와 UDP 주소지정: 포트와 소켓

## • TCP/IP 애플리케이션과 유명 포트번호 표

포트 #	TCP/UDP	키워드	프로토콜 약어	애플리케이션 또는 프로토콜
7	TCP + UDP	echo	-	에코 프로토콜
9	TCP + UDP	discard	-	디스카드 프로토콜
11	TCP + UDP	systat	-	활성 사용자 프로토콜
13	TCP + UDP	daytime	-	데이타임 프로토콜
17	TCP + UDP	qotd	QOTD	오늘의 문장 프로토콜
19	TCP + UDP	chargen	-	문자 생성기 프로토콜
20	TCP	ftp-data	FTP(데이터)	파일 전송프로토콜(기본 데이터 포트)
21	TCP	ftp	FRP(컨트롤)	파일 전송 프로토콜(제어/명령)
23	TCP	telnet	-	텔넷 프로토콜
25	TCP	smtp	SMTP	단순 메일 전송 프로토콜
37	TCP + UDP	time	-	타임 프로토콜
43	TCP	nickname	-	후이즈 프로토콜(Nickname)
53	TCP + UDP	domain	DNS	도메인 네임 서버
67	UDP	bootps	BOOTP/DHCP	부트스트랩 프로토콜/동적 호스트 설정 프로토콜(서버)
68	UDP	bootpc	BOOTP/DHCP	부트스트랩 프로토콜/동적 호스트 설정 프로토콜(클라이언트)
69	UDP	tftp	TFTP	간이 파일 전송 프로토콜
70	TCP	gopher	-	고퍼 프로토콜

# TCP와 UDP 주소지정: 포트와 소켓

## • TCP/IP 애플리케이션과 유명 포트번호 표

포트 #	TCP/UDP	키워드	프로토콜 약어	애플리케이션 또는 프로토콜
79	TCP	finger	-	핑거 사용자 정보 프로토콜
80	TCP	http	HTTP	하이퍼텍스트 전송 프로토콜(월드와이드웹)
110	TCP	pop3	POP	포스트 오피스 프로토콜(버전 3)
119	TCP	nntp	NNTP	유즈넷 뉴스 전송 프로토콜
123	TCP	ntp	NTP	네트워크 타임 프로토콜
137	TCP + UDP	netbios-ns	-	NetBIOS(네임 서비스)
138	UDP	netbiosdgm	-	NetBIOS(데이터그램 서비스)
139	TCP	netbios-ssn	-	NetBIOS(세션 서비스)
143	TCP	imap	IMAP	인터넷 메시지 접근 프로토콜
161	UDP	snmp	SNMP	단순 네트워크 관리 프로토콜
162	UDP	snmptrap	SNMP	단순 네트워크 관리 프로토콜(트랩)
179	TCP	bgp	BGP	경계 경로 프로토콜
194	TCP	irc	IRC	인터넷 릴레이 채팅
443	TCP	https	HTTP over SSL	SSL 위에서 동작하는 HTTP
500	UDP	isakmp	ike	IPsec 인터넷 키 교환
520	UDP	router	RIP	라우팅 정보 프로토콜(RIP-1과 RIP-2)
521	UDP	ripng	RIPng	라우팅 정보 프로토콜-차세대 버전

# 사용자 데이터그램 프로토콜(UDP)

---

- 개요

- 등장 배경

- 전송 계층에서 신뢰성, 승인, 흐름제어 기능을 필요로 하지 않는 애플리케이션을 위해 개발

- TCP가 아닌 UDP를 사용해야 하는 이유

- 안정성보다 효율이 중요한 경우
    - e.g., 멀티 스트리밍 애플리케이션
  - 길이가 짧은 데이터를 전송하는 경우
    - 애플리케이션 계층에서 잠재적 문제 해결 가능한 경우
    - e.g., 연결 요청/응답 메시지 전송
  - 멀티캐스팅과 브로드캐스팅을 사용하는 경우

# 사용자 데이터그램 프로토콜(UDP)

---

- UDP의 장점

- 비교적 단순한 헤더를 가지고 있어 데이터 전송하는데 처리가 빠름
- 연결 수립 시간, 흐름 제어, 오류 제어로 인한 지연시간을 줄일 수 있음

- UDP의 단점

- 목적지에 도달함을 보장하지 못함
- 송신한 순서대로 수신하기 어려움
- 체크섬 이외에 흐름제어, 오류제어를 위한 기능 없음
  - 신뢰성 없음
  - 손실 메시지를 수동으로 탐지하여 재전송해야 함



# 사용자 데이터그램 프로토콜(UDP)

- UDP 메시지 포맷

- 가상 헤더 개념

- 실제 전송에 포함되지 않는 가상헤더를 포함해 체크섬 계산
- 수신 장비는 가상 헤더를 생성해 체크섬 값 비교
  - 메시지가 다른 장비로 수신되는 경우 탐지

- 가상 헤더 포맷 그림

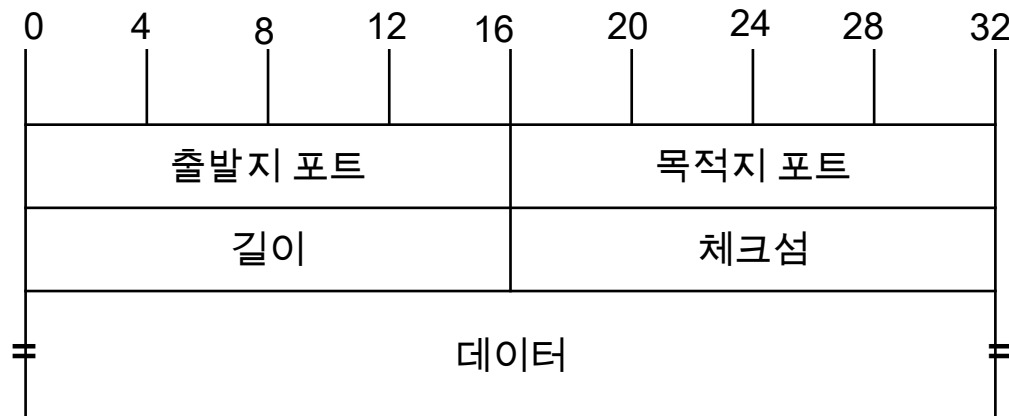


# 사용자 데이터그램 프로토콜(UDP)

- UDP 메시지 포맷

- 짧은 헤더를 가짐 (8바이트)
- 단순한 기능만 제공

- UDP 메시지 포맷 그림



---

감사합니다!