Layer 4.
Transport Layer (UDP)

Game Server Programming

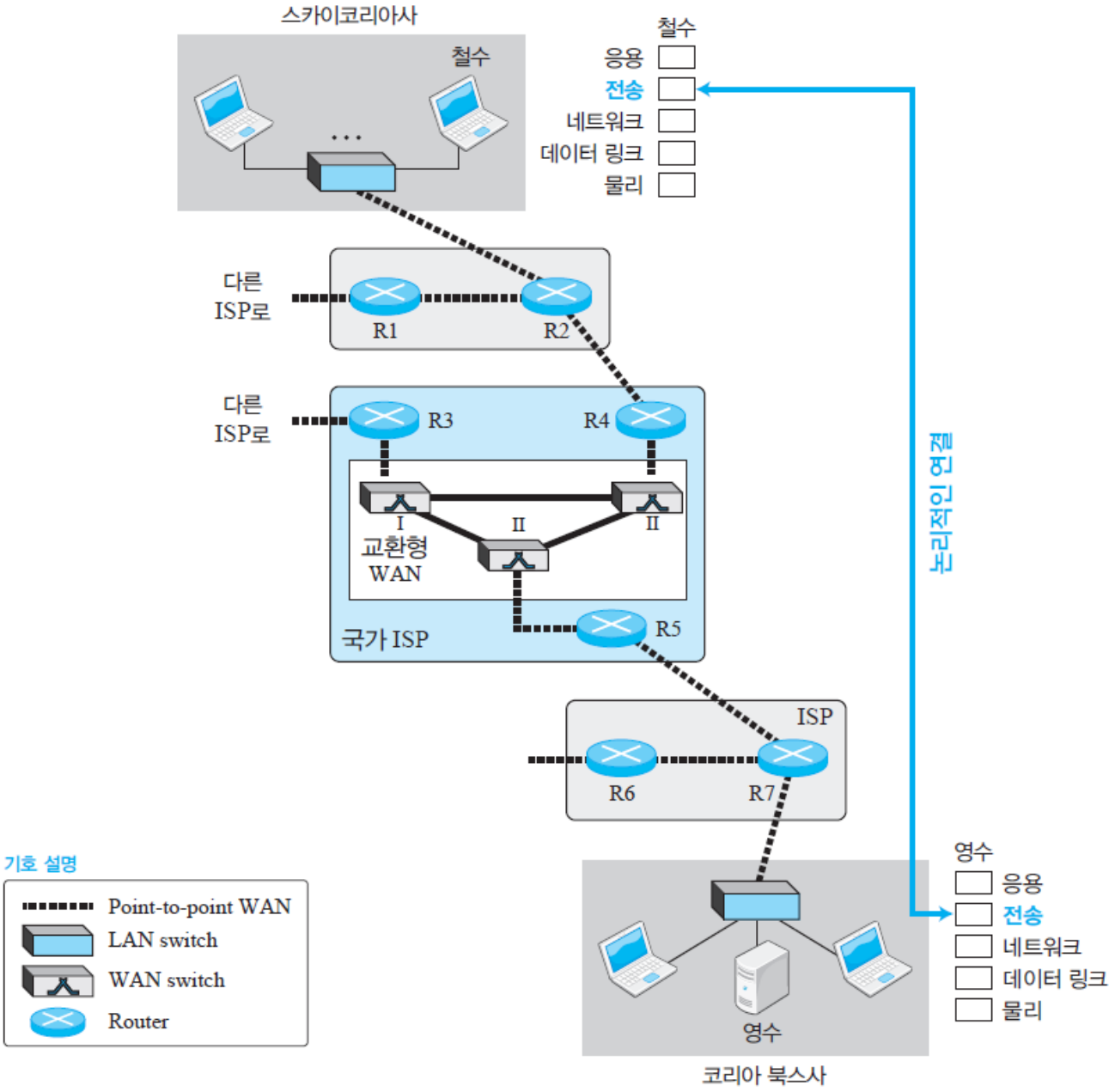
목 차

- 전송층 개요
- 개요
- 사용자 데이터그램
- UDP 서비스
- UDP 응용

개요

- 전송층
  - ▶ 네트워크층과 응용층 사이에 위치
  - ▶ 응용층에게 서비스를 제공할 의무가 있음
  - ▶ 네트워크로부터 서비스를 제공받음

■ 전송층의 논리적인 연결


스카이코리아사
철수
철수
응용
전송
네트워크
데이터 링크
물리
다른 ISP로
R1
R2
다른 ISP로
R3
R4
I
II
II
교환형 WAN
국가 ISP
R5
ISP
R6
R7
논리적인 연결
영수
응용
전송
네트워크
데이터 링크
물리
영수
코리아 북스사
기호 설명
Point-to-point WAN
LAN switch
WAN switch
Router

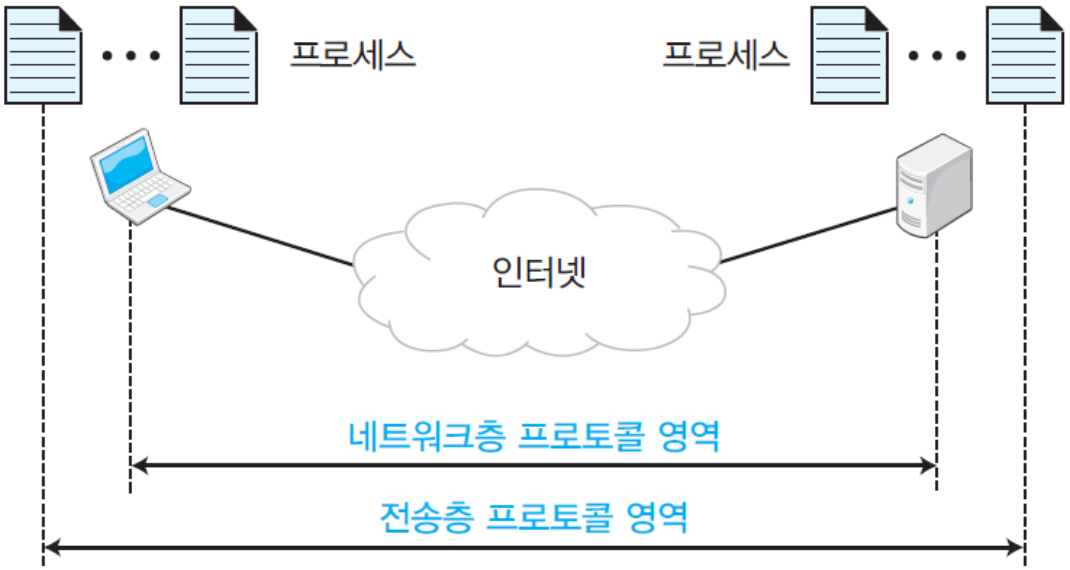
전송층 서비스

■ 프로세스-대-프로세스 통신 제공

▶ 프로세스는 전송층 서비스를 사용하는 응용층 개체(실행 중인 프로그램)

▶ 메시지를 적절한 프로세스에 전달 책임

네트워크층–대–전송층 영역



전송층 서비스

- 주소 체계: 포트 번호
  - ▶ 프로세스-대-프로세스 통신 방법: 클라이언트/서버
  - ▶ 클라이언트와 서버 프로세스는 같은 이름을 가짐
    - daytime client process/daytime server
  - ▶ 원격 컴퓨터는 여러 개의 서버 프로그램 실행
  - ▶ 로컬 컴퓨터로 여러 개의 클라이언트 프로그램 수행

* daytime service: 서버에서 현재 날짜와 시간을 클라이언트에 전달해 주는 간단한 프로토콜. RFC 867에서 정의되었으며, 주로 테스트나 디버깅 목적으로 사용.

전송층 서비스

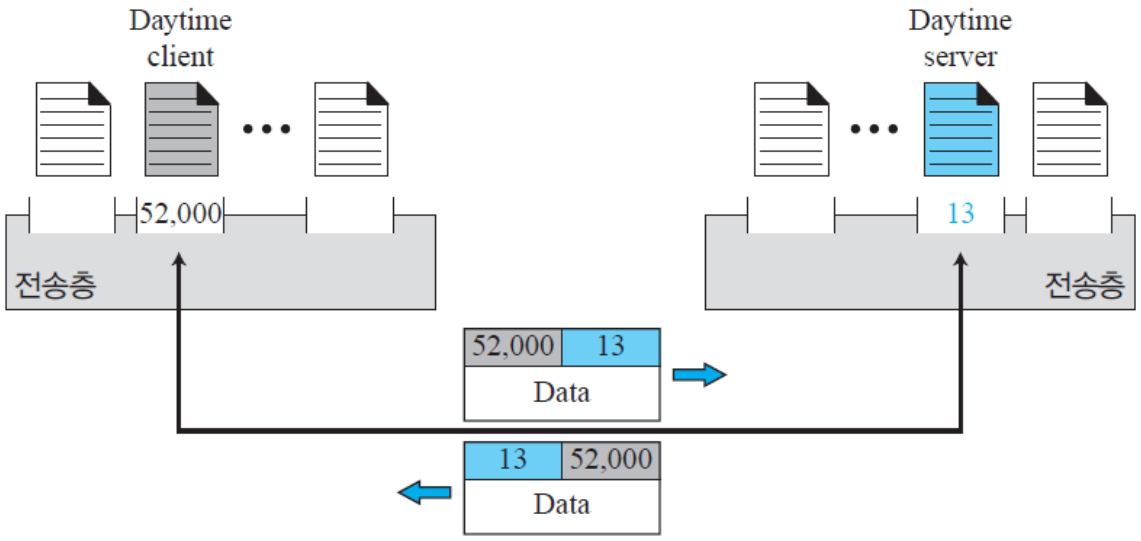
▶ 프로세스 통신을 위해 필요한 사항

- 로컬 호스트(local host)
- 로컬 프로세스(local process)
- 원격 호스트(remote host)
- 원격 프로세스(remote process)

▶ 프로세스 통신에서 포트 번호의 역할

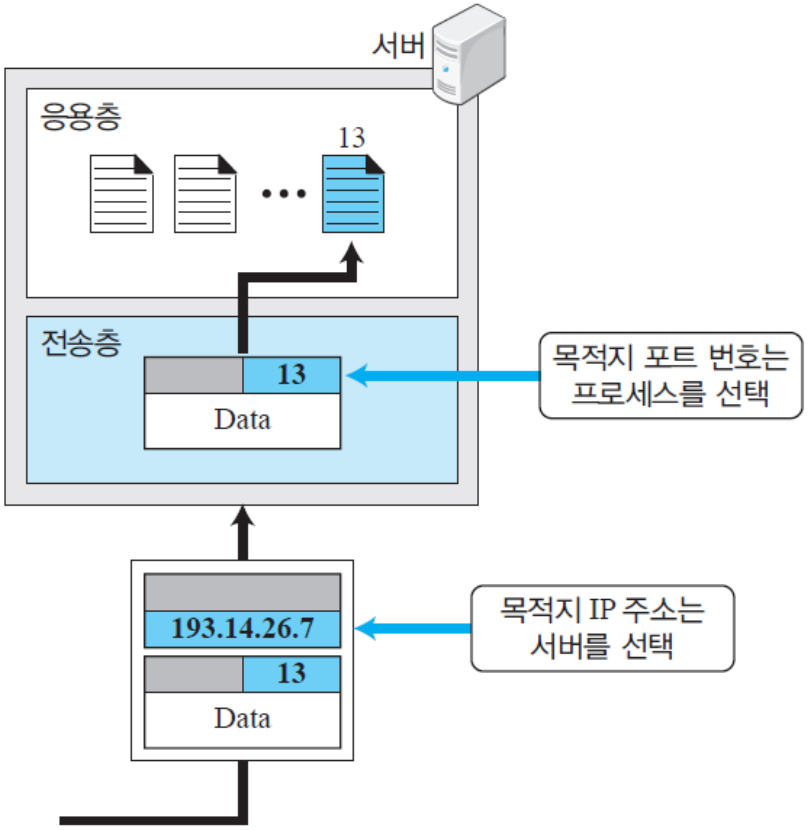
- 로컬 호스트와 원격 호스트: IP 주소
- 프로세스: 포트 번호
- 포트 번호 범위: 0 ~ 65,535 사이 정수
- 잘 알려진 포트 번호(well-known port number)
- 임시 포트 번호(ephemeral port number)
  - 클라이언트 프로그램이 서버와 통신하기 위해 운영체제가 일시적으로 자동 할당하는 포트 번호

포트 번호



▶ IP 주소 대 포트 번호

IP 주소–대–포트 번호

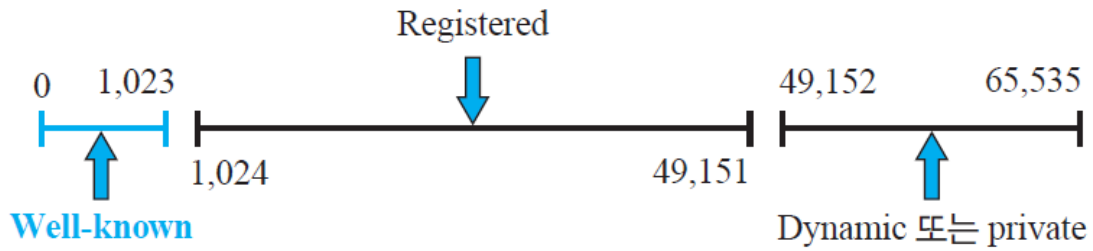


전송층 서비스

► ICANN 범위

- Internet Corporation for Assigned Names and Numbers
  - ✓ IANA (Internet Assigned Numbers Authority)
- 잘 알려진 포트: 0 ~ 1,023
  - ✓ Ex> http:80, ssh:22
- 등록된 포트(registered port): 1,024 ~ 49,151
  - ✓ Ex> MySQL:3306, RDP:3389
- 동적 포트(dynamic port): 49,152 ~ 65,535

ICANN 범위



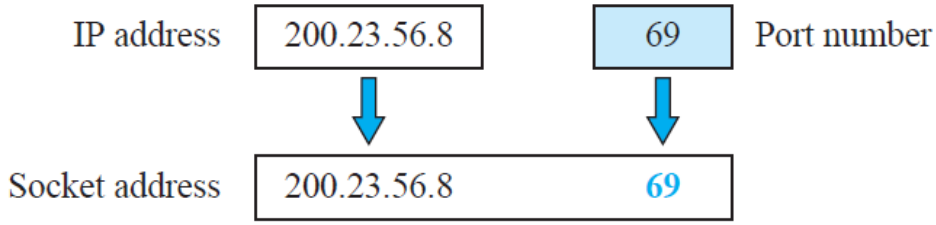
*ICANN: DNS 관리, IP주소할당, 프로토콜 파라미터 할당 등을 조정하는 비영리 국제 기구

전송층 서비스

▶ 소켓 주소(Socket address, end point)

- 각 종단점에서 연결을 만들기 위해 필요한 주소

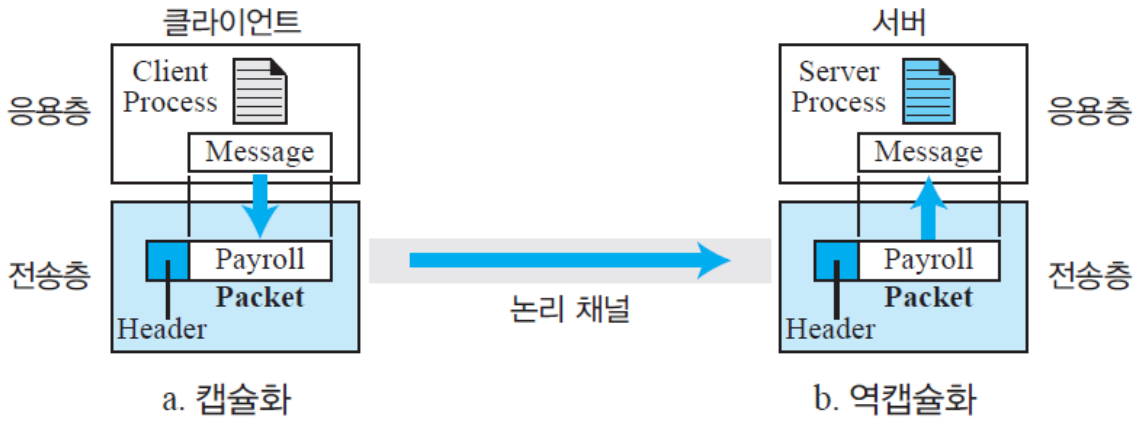
소켓 주소



전송층 서비스

■ 캡슐화와 역 캡슐화

캡슐화와 역캡슐화



전송층 서비스

■ 비연결형과 연결형 서비스

► 비연결형 서비스

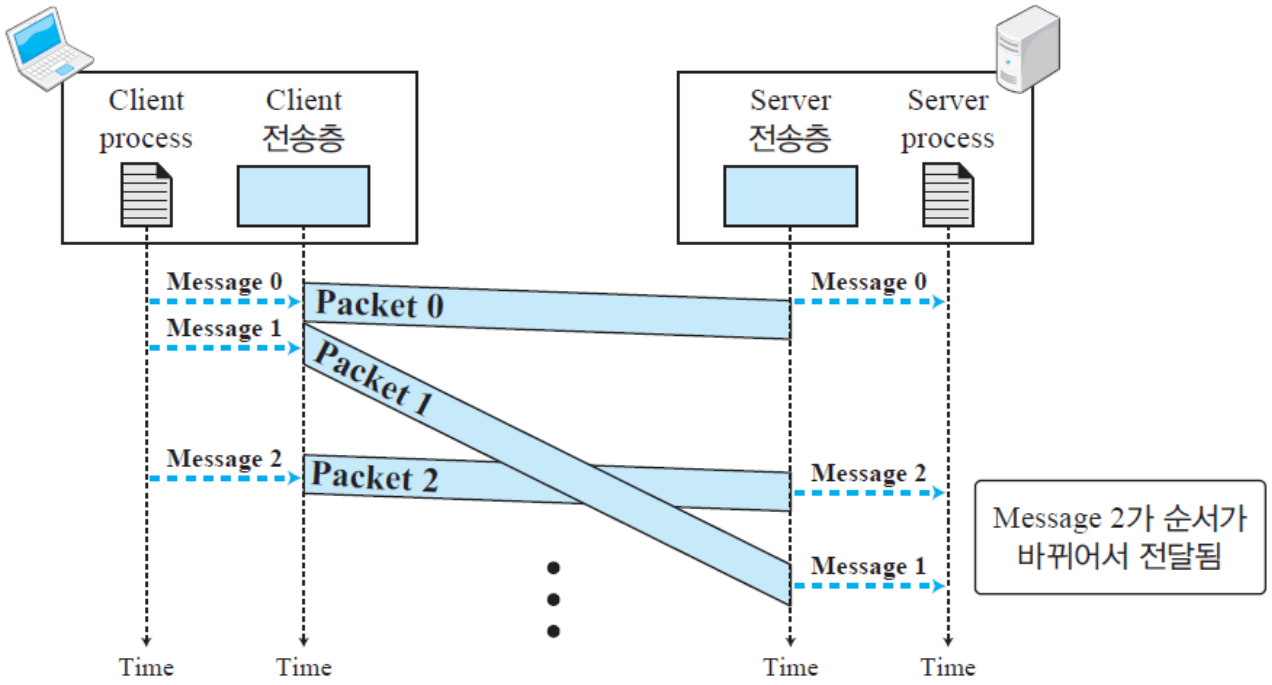
- 메시지를 전송 가능한 작은 단편으로 나누어 전송
- 각각의 조각을 독립적인 단위로 간주
- 순서에 어긋나서 도착하는 경우 발생
- UDP

► 연결형 서비스

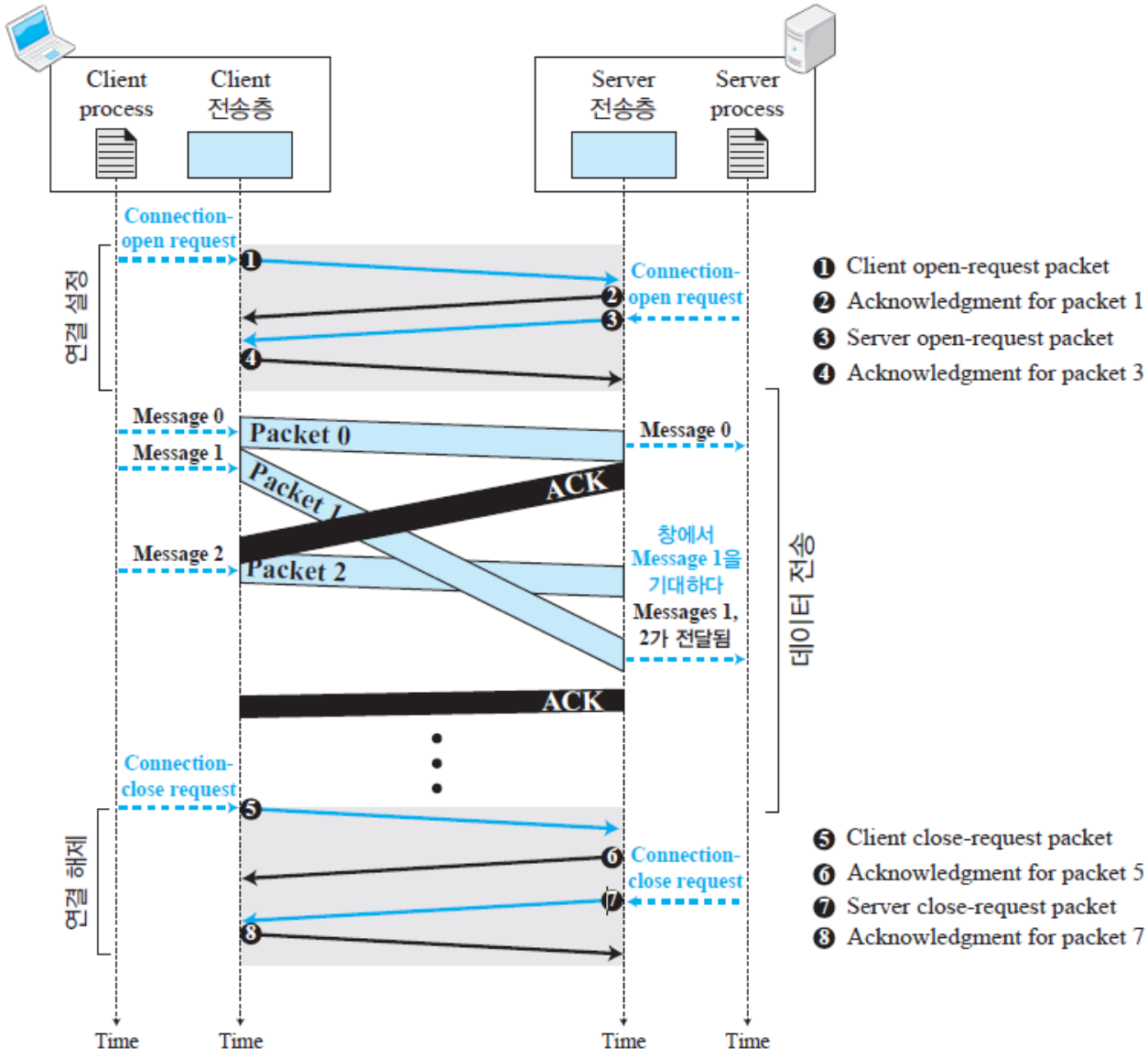
- 먼저 서버와 클라이언트간 연결 설정
- 데이터 교환 완료 후 연결 해제
- TCP

전송층 서비스

비연결형 서비스



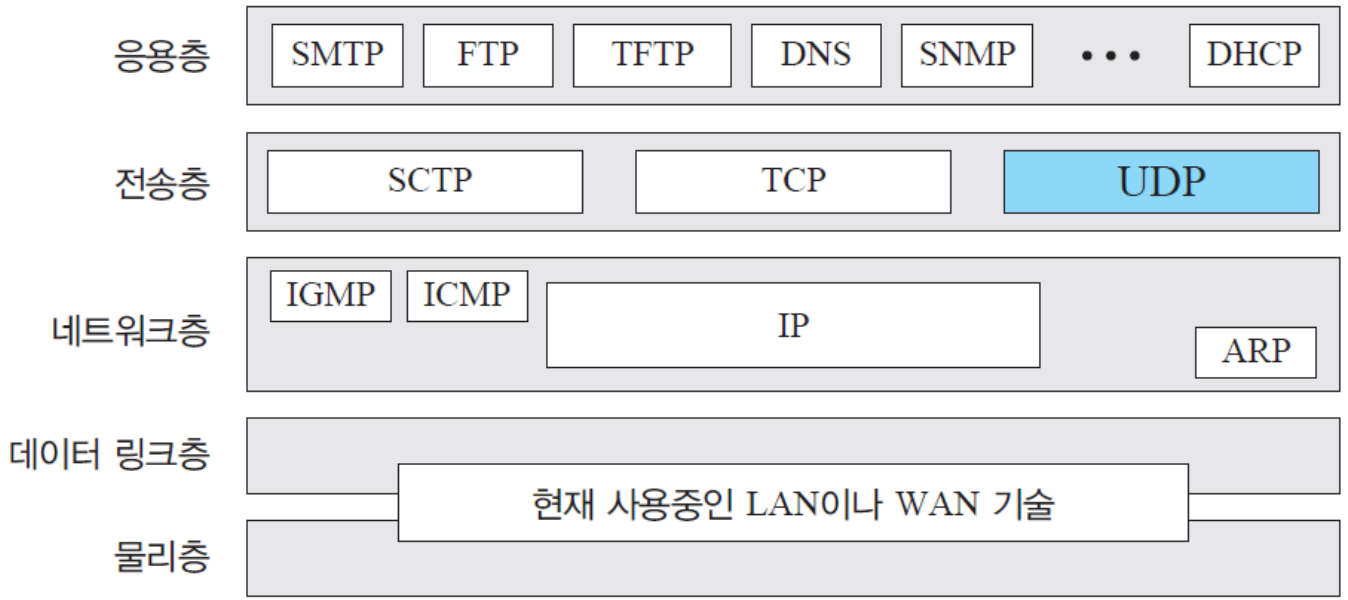
연결형 서비스


Client process
Client 전송층
Server 전송층
Server process
Connection-open request
연결 설정
Connection-open request
Message 0
Message 1
Packet 0
Packet 1
ACK
Message 2
Packet 2
Message 0
창에서 Message 1을 기대하다
Messages 1, 2가 전달됨
ACK
데이터 전송
Connection-close request
연결 해제
Connection-close request
Time
Time
Time
Time
❶ Client open-request packet
❷ Acknowledgment for packet 1
❸ Server open-request packet
❹ Acknowledgment for packet 3
❺ Client close-request packet
❻ Acknowledgment for packet 5
❼ Server close-request packet
❽ Acknowledgment for packet 7

UDP 개요

■ TCP/IP 프로토콜 모음에서 UDP 위치

TCP/IP 프로토콜 모음에서 UDP의 위치



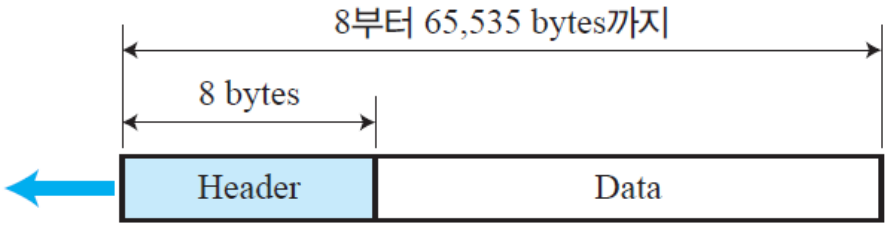
UDP 개요

- UDP 전송 프로토콜의 임무
  - ▶ 프로세스-대-프로세스 통신 생성: 포트번호 이용
  - ▶ 최소한의 오류 제어 메커니즘 수행
  - ▶ 프로세스로부터 데이터 단위를 받아 신뢰성 없는 전달 제공
  - ▶ 비연결형, 신뢰성이 없는 전송 프로토콜
  - ▶ 최소한의 오버헤드만 사용하는 간단한 프로토콜

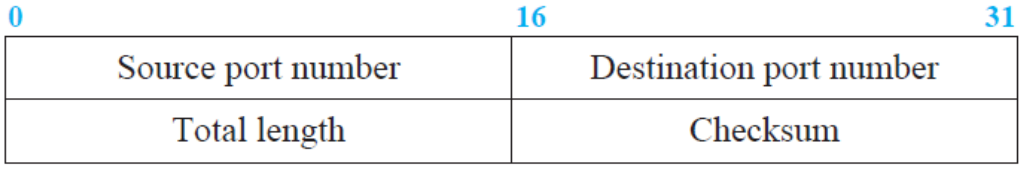
■ 사용자 데이터그램 형식

- ▶ 8바이트의 고정 크기 헤더
- ▶ 발신지 포트 번호(source port number)
  - 발신지 호스트에서 수행되는 프로세스가 사용하는 포트 번호
- ▶ 목적지 포트 번호(destination port number)
  - 목적지 호스트에서 수행되는 프로세스가 사용하는 포트 번호
- ▶ 길이(length): 헤더와 데이터를 합한 전체 길이
- ▶ 검사합(checksum): 오류 탐지에 사용

사용자 데이터그램 형식



a. UDP 사용자 데이터그램



b. 헤더 형식

사용자 데이터그램

▶ 예제 1

- 다음은 16진수의 형식으로 UDP 헤더를 덤프한 것이다.

```
CB84000D001C001C
```

a. 발신지 포트 번호는 얼마인가?

b. 목적지 포트 번호는 얼마인가?

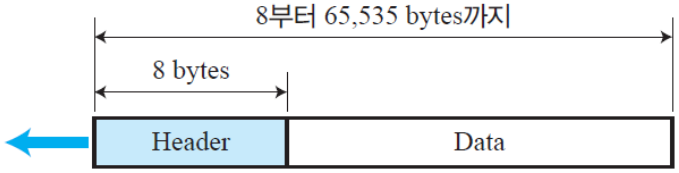
c. 사용자 데이터그램의 총 길이는 얼마인가?

d. 데이터의 길이는 얼마인가?

e. 데이터의 전송 방향이 클라이언트에서 서버쪽으로 가는 것인가? 아니면 반대의 방향인가?

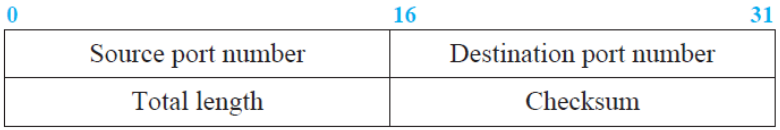
f. 클라이언트 프로세스는 무엇인가?

사용자 데이터그램



a. UDP 사용자 데이터그램

CB84000D001C001C



b. 헤더 형식

a. 발신지 포트 번호는 처음 네 자리 16진수($CB84_{16}$)이며, 10진수로 표현하면 52,100이다.

b. 목적지 포트 번호는 두 번째 네 자리 16진수($000D_{16}$)이며, 10진수로 표현하면 13이다.

c. 세 번째 네 자리 16진수($001C_{16}$)은 UDP 패킷의 전체 길이를 나타내며, 28바이트이다.

d. 데이터의 길이는 패킷의 전체 길이에서 헤더의 길이를 뺀 것으로, 28 – 8 = 20바이트이다.

e. 목적지 포트 번호가 13(잘 알려진 포트)이므로, 클라이언트로부터 서버로 패킷이 전송되었다.

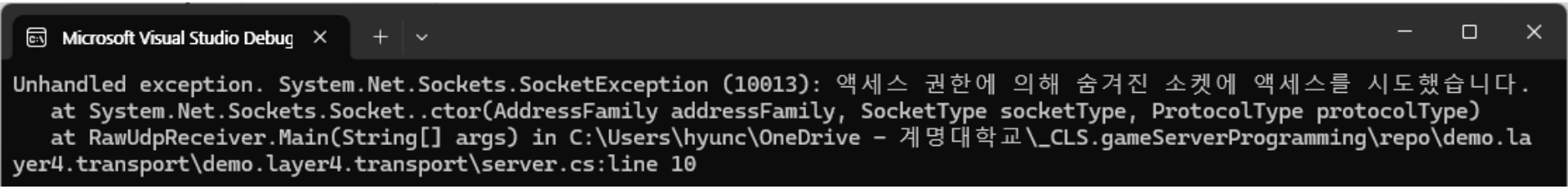
f. 클라이언트 프로세스는 Daytime이다.

프로토콜 UDP를 이용하는 잘 알려진 포트

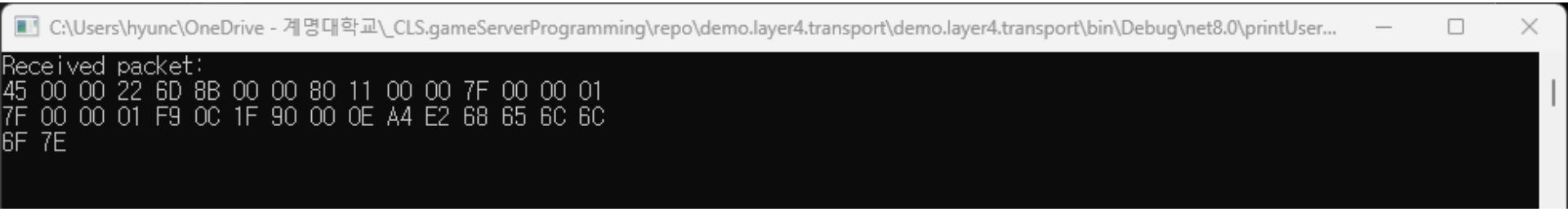
포트	프로토콜	설명
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
53	Domain	Domain Name Service (DNS)
67	Bootps	Server port to download bootstrap information
68	Bootpc	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

Demo. UDP 패킷 내용 확인

- UDP 헤더는 일반적으로 IP 스택에 의해 자동으로 처리되기 때문에, 고수준의 네트워크 프로그래밍에서 사용하는 UdpClient와 같은 클래스는 프로그래머에게 UDP 헤더를 직접 노출하지 않음
- 따라서 고수준의 소켓 프로그래밍에서는 UDP 헤더와 IP 헤더를 볼 수 없음
- 하지만 로우 레벨 네트워크 프로그래밍에서는 UDP 및 IP 헤더를 포함한 전체 패킷을 확인가능
  - ▶ Raw Socket (로우 소켓): IP 레벨에서 모든 패킷 제어 가능
  - ▶ C#에서 로우 소켓을 사용하려면, Socket 클래스를 사용해 SocketType.Raw로 소켓 생성
  - ▶ 운영체제의 권한이 필요할 수 있음 (관리자 권한)


Microsoft Visual Studio Debug
Unhandled exception. System.Net.Sockets.SocketException (10013): 액세스 권한에 의해 숨겨진 소켓에 액세스를 시도했습니다.
at System.Net.Sockets.Socket..ctor(AddressFamily addressFamily, SocketType socketType, ProtocolType protocolType)
at RawUdpReceiver.Main(String[] args) in C:\Users\hyunc\OneDrive - 계명대학교_CLS.gameServerProgramming\repo\demo.layer4.transport\demo.layer4.transport\server.cs:line 10


Microsoft Visual Studio Debug
Enter the message to send: hello~
Sent message: hello~


C:\Users\hyunc\OneDrive - 계명대학교_CLS.gameServerProgramming\repo\demo.layer4.transport\demo.layer4.transport\bin\Debug\net8.0\printUser...
Received packet:
45 00 00 22 6D 8B 00 00 80 11 00 00 7F 00 00 01
7F 00 00 01 F9 0C 1F 90 00 0E A4 E2 68 65 6C 6C
6F 7E

IP 헤더 필드

```
Received packet:
45 00 00 22 6D 8B 00 00 80 11 00 00 7F 00 00 01
7F 00 00 01 F9 0C 1F 90 00 0E A4 E2 68 65 6C 6C
6F 7E
```

- (45):4: IPv45 , 헤더 길이 (5 * 4 = 20바이트)
- 서비스 유형 (00): 일반적인
- 총 길이 (00 22): 34바이트 (IP 헤더 + UDP 헤더 + 데이터)
- 식별자 (6D 8B): 패킷 식별자 (전송된 패킷의 고유 식별 값)
- 플래그 및 오프셋 (00 00): 분할되지 않은 패킷 (플래그 0)
- TTL (Time to Live) (80): 패킷이 네트워크에서 허용되는 최대 홉 수 (128)
- 프로토콜 (11): UDP (0x11 = 17, 이는 UDP를 나타냄)
- 헤더 체크섬 (00 00): 체크섬 값 (일부 시스템에서는 0으로 설정)
- 출발지 IP (7F 00 00 01): 127.0.0.1
- (로컬호스트)목적지 IP (7F 00 00 01): 127.0.0.1 (로컬호스트)

UDP 헤더 필드

```
Received packet:
45 00 00 22 6D 8B 00 00 80 11 00 00 7F 00 00 01
7F 00 00 01 F9 0C 1F 90 00 0E A4 E2 68 65 6C 6C
6F 7E
```

- 출발지 포트 (F9 0C):16진수 = 63756번
- 포트목적지 포트 (1F 90):16진수 = 8080번
- 포트UDP 길이 (00 0E):16진수 000E = 14바이트 (UDP 헤더 + 데이터의 길이)
- 체크섬 (A4 E2):체크섬 값

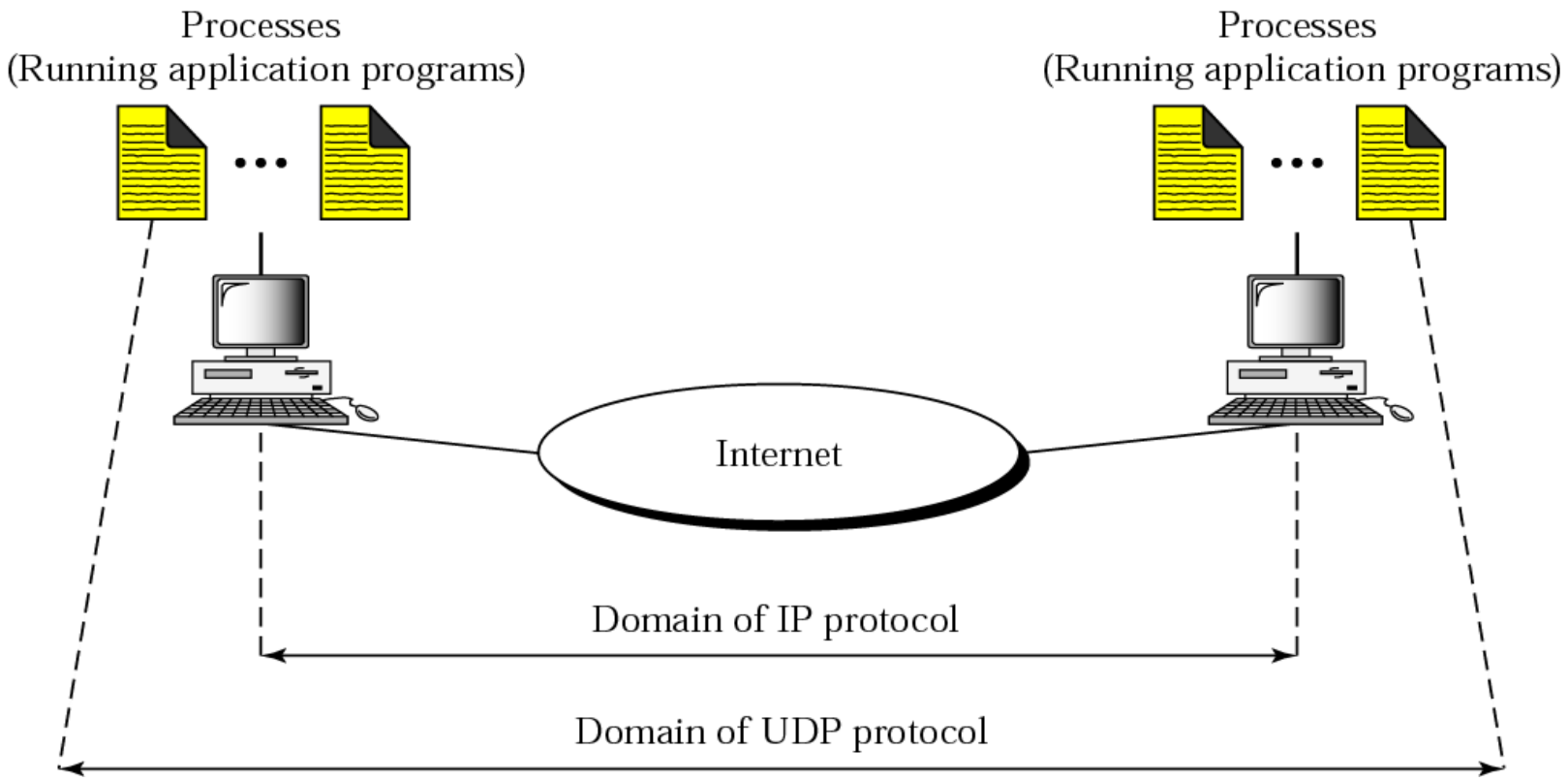
- Data 필드
  - ► 68 65 6C 6C 6F 7E → "hello~"

UDP 서비스

- UDP 서비스
  - 프로세스 대 프로세스 통신: IP 주소와 포트번호로 구성된 소켓 이용
  - 비연결형 서비스: 각 사용자 데이터그램은 독립적
  - 흐름 제어: 기능이 없고 윈도우 메커니즘도 없음
  - 오류 제어: 검사합을 제외한 기능 없음
  - 혼잡 제어: 기능 없음

UDP 서비스

■ 프로세스-대-프로세스 통신



UDP 서비스

- UDP와 일반 단순 프로토콜과 비교
  - 비연결형 단순 프로토콜의 한 예
  - 오류 감지를 위해서 부가적인 검사합을 패킷에 포함한다는 것만 차이.
  - 수신 UDP는 만일 패킷이 훼손되면 폐기.
  - 어떠한 피드백도 송신 측으로 전송되지 않음

UDP 응용

- UDP 특징
  - 비연결형 서비스: 각 UDP 패킷은 서로 독립적
  - 클라이언트 응용이 서버에게 짧은 요청을 전송하고 짧은 응답을 수신하는 경우에 장점
  - 응용에서 지연이 주요 이슈인 경우는 비연결형 서비스가 바람직
  - 오류 제어는 제공하지 않음
  - 혼잡 제어는 제공하지 않음

UDP 응용

- 인터넷을 통하여 매우 큰 문서 파일을 다운받고 있다고 가정
- 신뢰성 서비스를 제공하는 전송층을 사용해야 한다.
- 즉, 파일을 열었을 때 파일의 일부분이 손실되거나 훼손되는 것을 원하지 않는다.
- 파일의 일부분의 전달 사이에 발생하는 지연은 중요한 문제가 아니다.
- 단지 파일을 보기 전에 전체 파일이 구성되기만을 기다린다.
- 위경우에는 UDP는 적합한 전송 계층이 아니다.

EoF