UDP와 검사합(checksum)

한기대 박승철교수

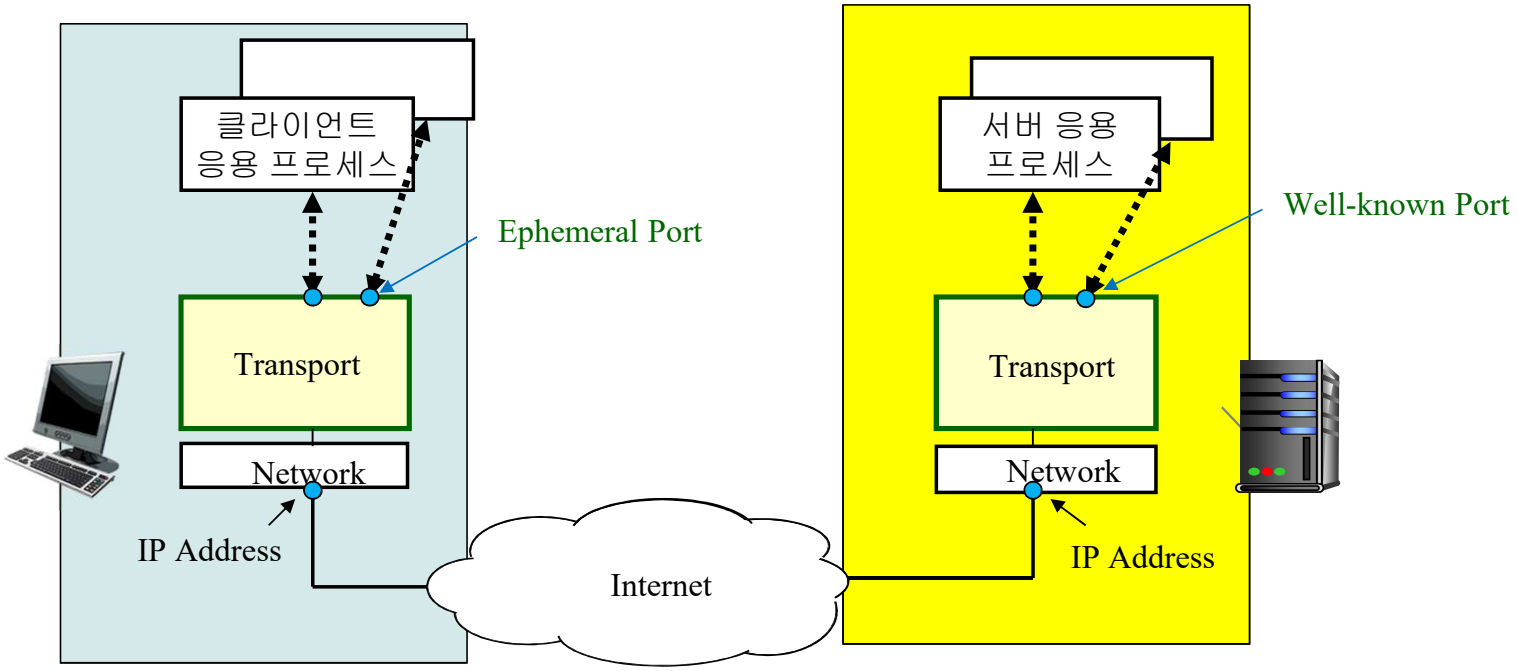
강의 내용

- UDP 서비스
- UDP 데이터그램 구조
- 검사합(checksum)

UDP 서비스

❖ 포트번호 기반의 다중화 서비스

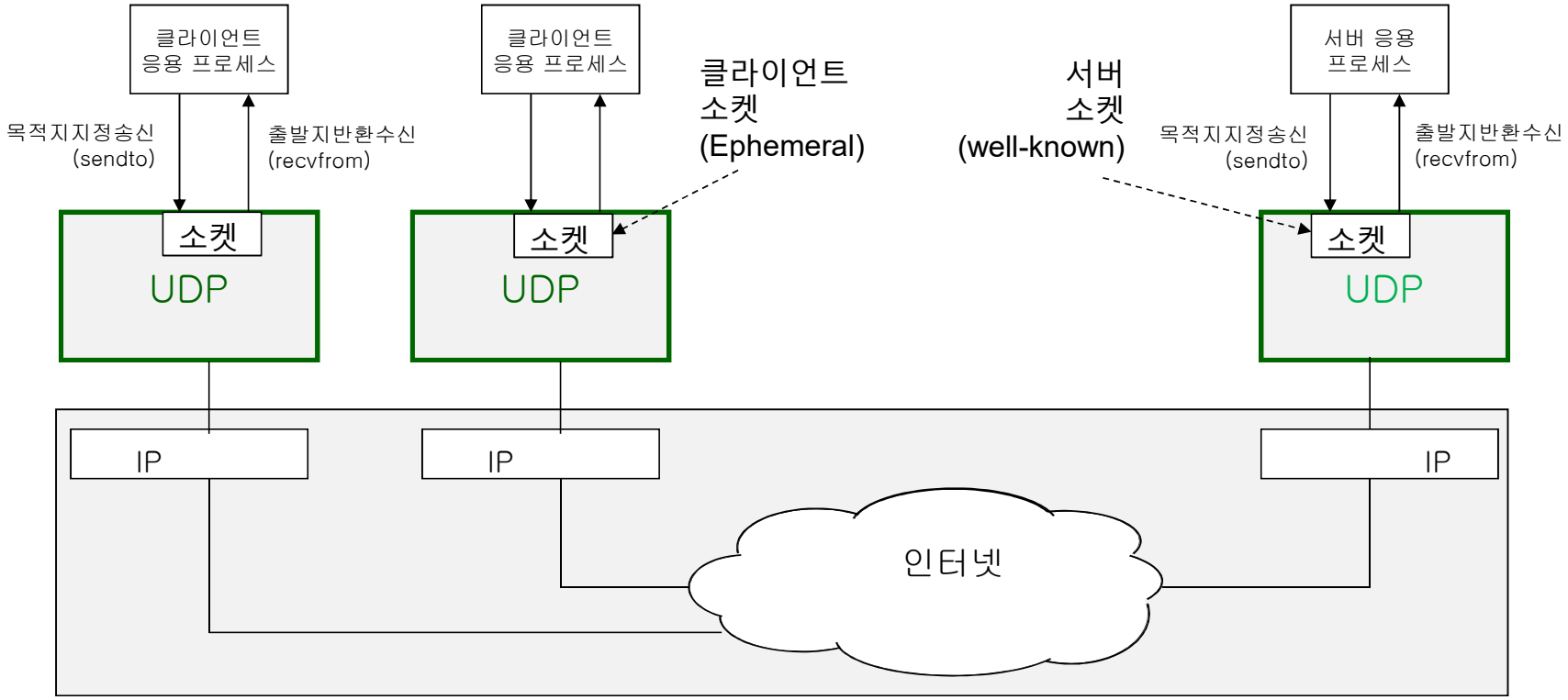
- 서버 : Well-known Port
- 클라이언트 : Ephemeral Port



UDP 서비스

❖ 비연결형(connectionless) 전송 서비스

- 통신 소켓 간의 연결 설정 없음 : 연결 설정 지연시간 회피



UDP 서비스

❖ 데이터그램 실시간 전송 서비스

- 응용 프로세스로부터 데이터가 송신 소켓에 전달되면,
- 송신 UDP는 해당 데이터를 포함하는 UDP 데이터그램(Datagram) 생성
- 각 데이터그램은 IP를 통해 독립적으로 목적지 UDP 소켓에 전송

❖ 응용

- 인터넷 전화 등 실시간 응용에 적합
- DNS(Domain Name System) 등 작은 데이터, 빠른 응답 지연시간이 요구되는 응용에 적합

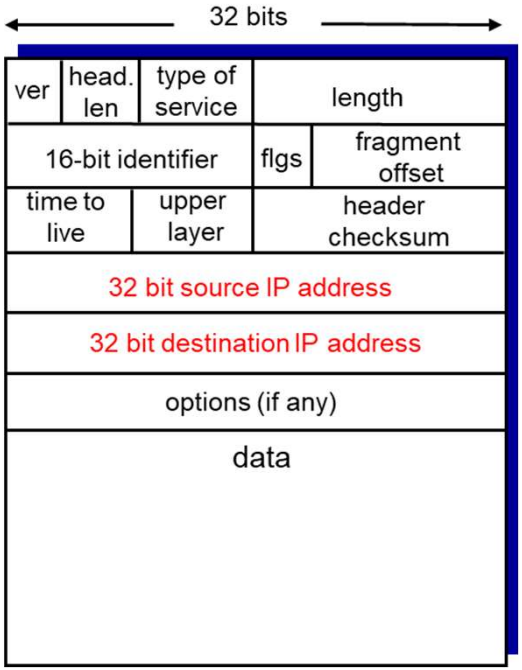
UDP 서비스

- **1:N, N:1 데이터그램 통신 서비스**
  - 목적지 IP 주소에 멀티캐스트 주소를 사용하여 다수의 목적지 소캣으로 데이터그램 전송 가능
  - 여러 개의 소켓으로부터 데이터그램 수신 가능
  - 응용 프로세스는 송신할 때마다 UDP에게 수신 소켓주소 지정
  - UDP가 응용 프로세스에게 데이터를 전달할 때마다 송신 소켓주소 표시
- **응용**
  - 인터넷 TV 등 멀티캐스트 응용 효과적 지원

UDP 데이터그램 구조

❖ 간단한 데이터그램 포맷

- 처리 시간 단축



검사합(checksum)

❖송신자

1) 송신하는 메시지를 정해진 길이의 데이터 단위로 나눔
2) 모든 데이터 단위를 1의 보수(1's complement) 연산으로 더하여 합(sum)을 구함
3) 합의 1의 보수를 검사합(checksum)으로 생성하고, 검사합을 추가하여 메시지 전송

검사합(checksum)

❖ 수신자

1) 수신된 메시지를 정해진 길이의 데이터 단위로 나눔(검사합 포함)
2) 모든 데이터 단위를 1의 보수(1's complement) 연산으로 더하여 합(sum)을 구함
3) 합이 0이면 성공적인 수신, 아니면 오류 발생

검사합(checksum)

❖ 1의 보수 연산

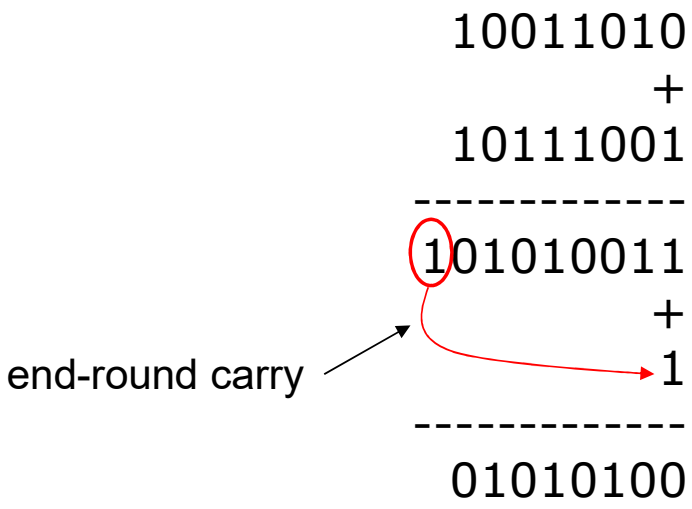
- 이진수 1의 보수 : 각 비트 값 0과 1을 서로 바꾼 값(원래 비트 값과 더하여 1이 되는 값)
- +0 : 모든 비트가 0, -0 : 모든 비트가 1

❖ m비트 이진수의 1의 보수 덧셈

- m비트 결과 값 생성
- 최종 비트 올림(carry)이 생길 경우 캐리 값을 m비트 결과에 더하여 최종 m비트 결과값 생성

검사합(checksum)

❖ 1의 보수 덧셈(8비트 이진수)



검사합(checksum)

❖ 8비트 검사합 계산

- 데이터 : 00111010 10011000 00111010

```
  00111010
         +
  10011000
------------
  11010010  ←  중간합
         +
  00111010
------------
  00001100
         +
         1  ←  carry
------------
  00001101  ⟶  검사합 : 11110010
  1의 보수
```

검사합(checksum)

❖ 오류 확인(데이터 : 00111010 10011000 00111010)

- 수신 데이터 : 00111010 10011000 00111010 11110010 ← 검사합

```
  00111010                    00001101
         +                           +
  10011000                    11110010
------------                ------------
  11010010 ← 중간합          11111111 ← -0
         +
  00111010
------------
  00001100
         +
         1 ← carry
------------
  00001101 ← 중간합
```

검사합(checksum)

- 장점
  - 높은 오류 검출 능력
  - 단순함
  - 소프트웨어 구현에 적합
- 단점
  - 각 데이터 단위에서 발생하는 오류의 합이 0이 되는 오류 검출 불가
  - 체섬을 변경하지 않는 오류 검출 불가

검사합(checksum)

❖ 오류 검출 불가(데이터 : **00111010 10011000 00111010**)

- 수신 데이터 : 00111000 10011010 00111010 11110010

```
  00111000                    00001101
         +                           +
  10011010                    11110010
 ----------                  ----------
  11010010 ← 중간합           11111111 ← -0
         +
  00111010
 ----------
  00001100
         +
         1 ← carry
 ----------
  00001101 ← 중간합
```

검사합(checksum)

- 오류 검출 불가(데이터 : **00111010 10011000 00111010**)
  - 수신 데이터 : 00111000 10011001 00111011 11110010

```
  00111000                    00001101
         +                           +
  10011001                    11110010
------------                ------------
  11010001 ← 중간합          11111111 ← -0
         +
  00111011
------------
  00001100
         +
         1 ← carry
------------
  00001101 ← 중간합 ──→ 검사합 : 11110010(변경 없음)
```