Term Project_2

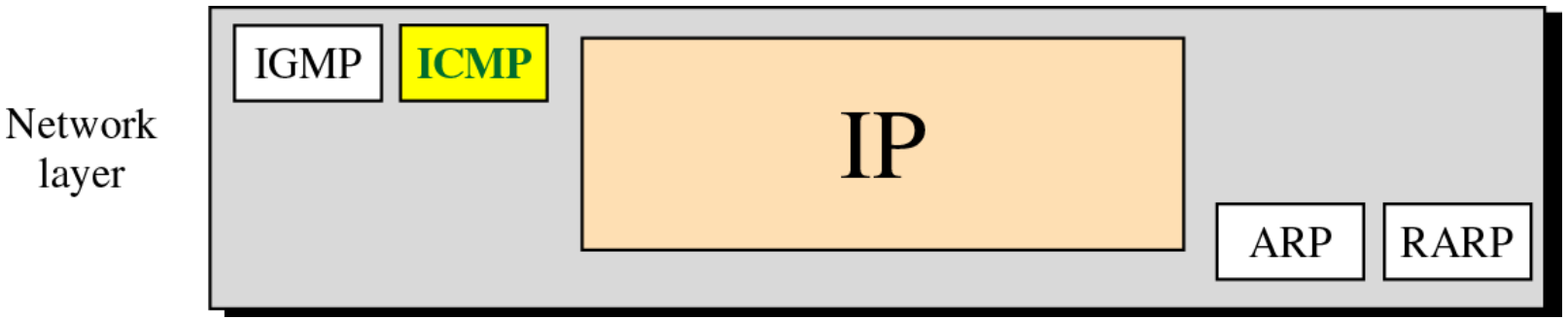
유명성

1. ICMP

1. ICMP

1.1 ICMP

- IP는 오류보고와 오류수정 기능이 없다.
- 호스트와 Management를 위한 Query 메커니즘이 없다.
- 위 단점 보완을 위해 ICMP가 설계되었다
- ICMP는 IP 데이터그램을 통해 전송된다.

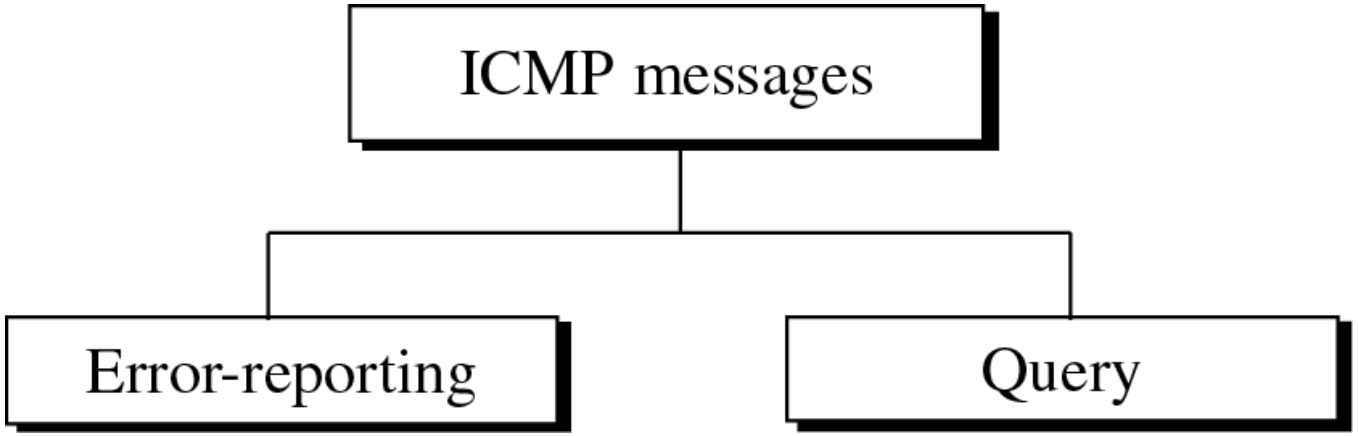


1. ICMP

1.1 ICMP

ICMP 메시지의 종류

- Error_reporting
  - IP패킷 처리 중 발생하는 문제를 Report
- Query
  - 관리자가 라우터나 다른 호스트로부터 특정 정보획득에 사용



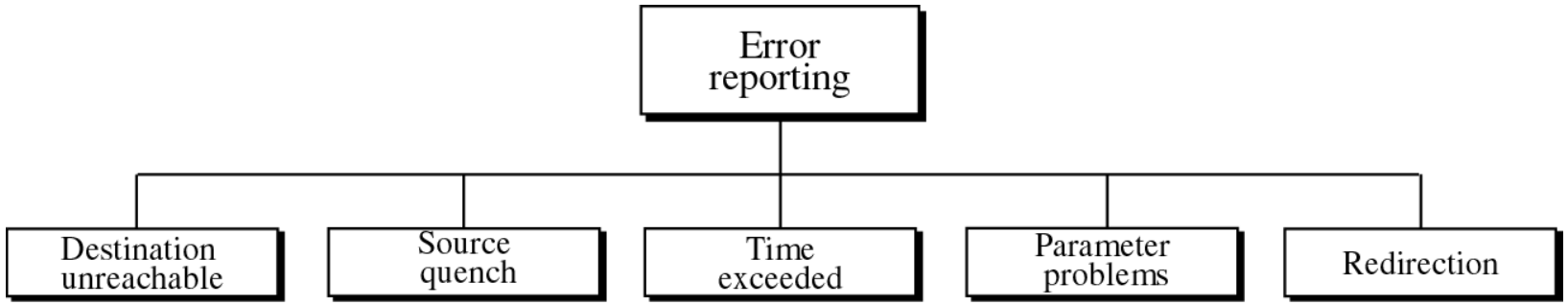
1. ICMP

1.1 ICMP

ICMP 메시지의 종류

Category	Type	Message
Error-reporting message	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query message	8 or 0	Echo request or reply
	13 or 14	Time stamp request or reply
	17 or 18	Address mask request or reply
	10 or 9	Router solicitation and advertisement

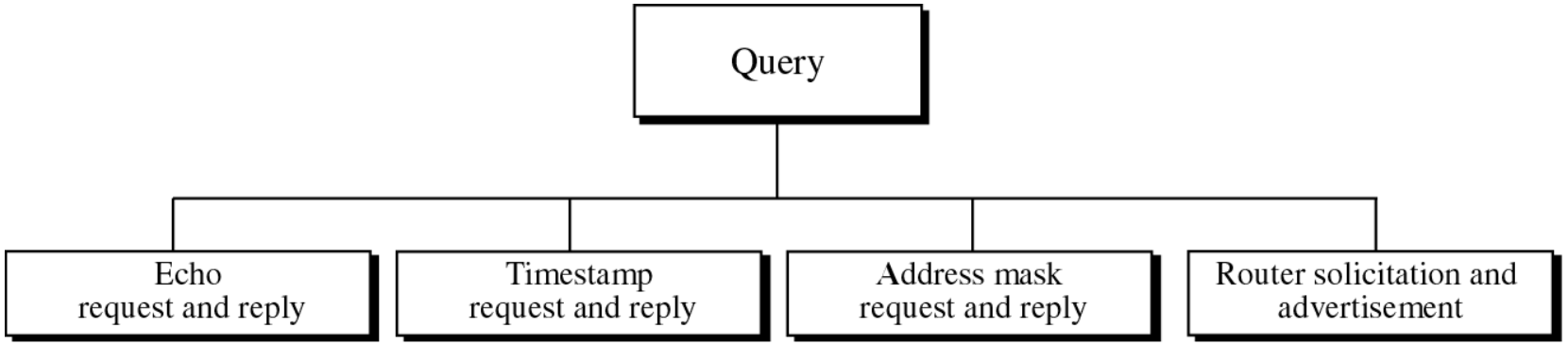
ICMP 오류 보고 종류



- 오류보고의 형식은 5가지로 구별되어 있다
- ICMP의 주임무는 오류 수정이 아니라 보고이다.
- 오류 메세지는 최초 발신자에게 보내진다.
  - 패킷의 정보에는 발신지와 목적지 밖에 없다.

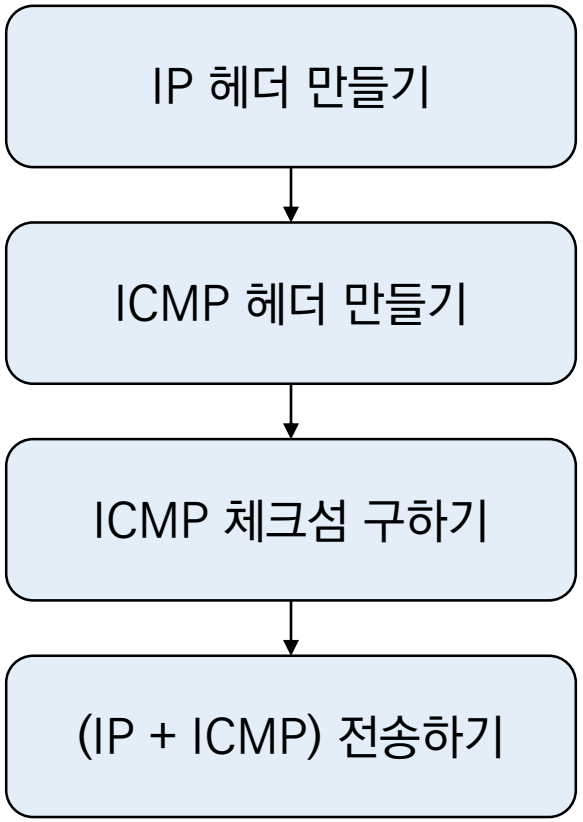
ICMP 질의(Query) 종류

- 네트워크 문제의 진단에 사용
- 4가지 질의 메시지가 있음
  - 에코 요청과 응답
  - 타임스탬프 요청과 응답
  - 주소마스크 요청과 응답
  - 라우터 요청과 광고



1. ICMP

1.2 ICMP 보내기



1. ICMP

1.2 ICMP 보내기

```
socket.socket(${family}, socket.SOCK_RAW, ${protocol})
```

AF_INET

IPPROTO_RAW

IPPROTO_ICMP

IPPROTO_UDP

IPPROTO_TCP

- IP 헤더의 ToL, Checksum은 0으로 채울 경우 커널 IP 스택이 채워준다.
- ICMP 헤더의 Checksum은 직접 계산해야 한다.
- AF_INET, SOCK_RAW, IPPROTO_RAW 소켓은 송신만 가능하다.

1. ICMP

1.3 ICMP 체크섬 계산

ICMP Checksum 계산

1. 헤더의 checksum 필드를 0x0000으로 채운다.
2. 헤더를 2byte 단위로 끊어서 더한다. 만약 홀수라면 0x00을 더한다.
3. 더한 값이 4byte 이상이라면 올림수를 값에 다시 더한다.
4. 3에서 계산한 값에 1의 보수를 취한다.

1. ICMP

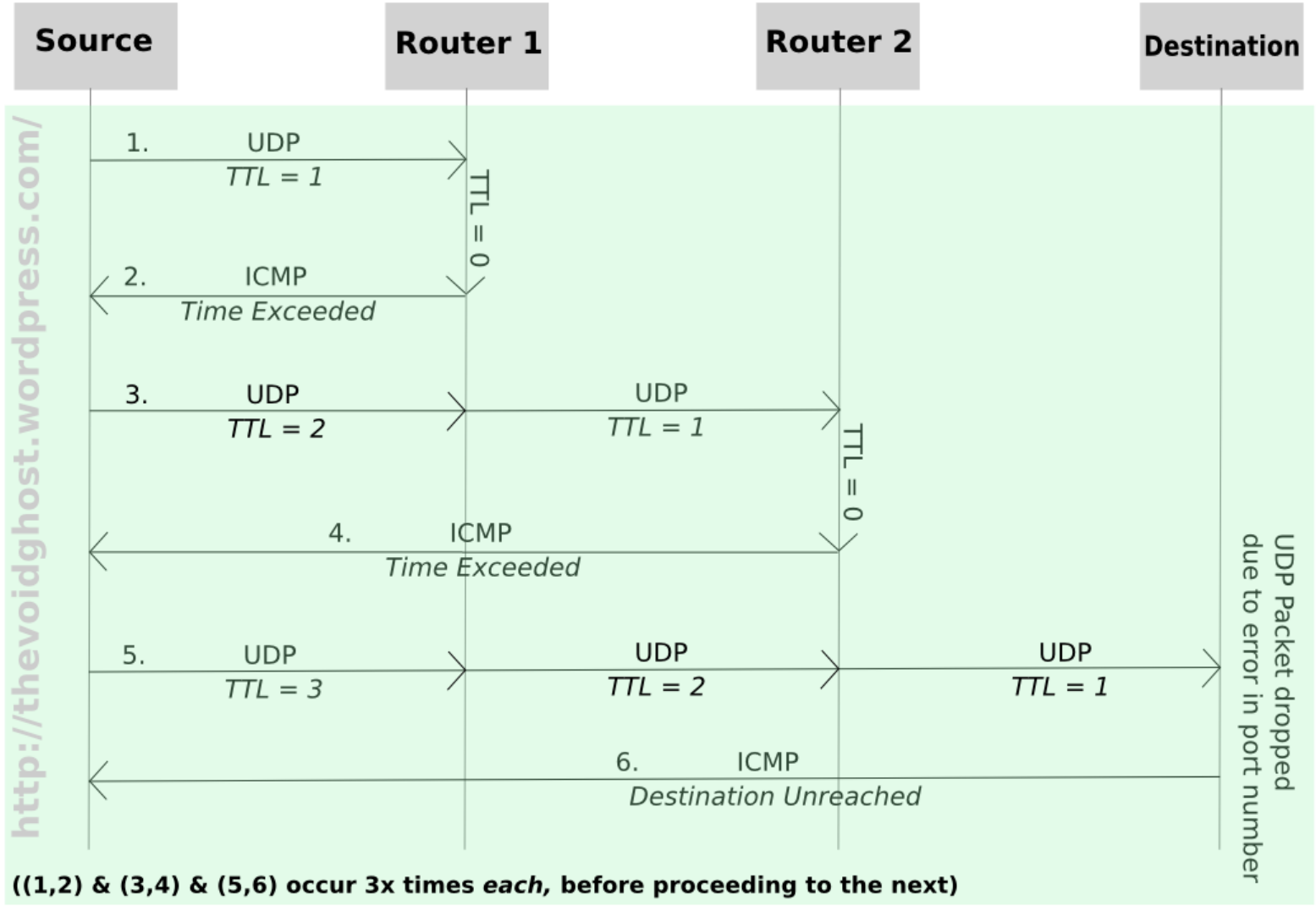
1.3 ICMP 체크섬 계산

```
17          @staticmethod
18          def make_checksum(header):
19              size = len(header)
20              if (size % 2) == 1:
21                  header += b'₩x00'
22                  size += 1
23              size = size // 2
24              header = struct.unpack('!' + str(size) + 'H', header)
25              sum = reduce(lambda x, y: x+y, header)
26              chksum = (sum >> 16) + (sum & 0xffff)
27              chksum += chksum >> 16
28              chksum = (chksum ^ 0xffff)
29
30              return chksum
```

2. Traceroute

2. Traceroute

2.1 To-Do



2. Traceroute

2.1 To-Do

ICMP를 이용한 Traceroute

1. IP 헤더 만들기(TTL=n, dst=host. id=random)
2. ICMP 헤더 만들기(Type 8, Code 0, id=random, seq=n, data)
3. 패킷 전송 (IP + ICMP)
4. 응답 수신(timeout)
5. Time Exceeded(Type 11, Code 0)이고, 포함된 IP헤더가 일치하면 중간경로
6. Echo Reply(Type 0, Code 0)이고 ICMP id, ICMP data가 일치하면 종료

2. Traceroute

2.1 To-Do

UDP를 이용한 Traceroute

1. IP 헤더 만들기(TTL=n, dst=host, id=random)
2. UDP 헤더 만들기(src port, dst port=random, data)
   - tip: dst port는 매우 큰 값으로 해야 한다.(well-known port X)
3. 패킷 전송 (IP + UDP)
4. 응답 수신(timeout)
5. Time Exceeded(Type 11, Code 0)이고 포함된 IP 헤더가 일치하면 중간경로
6. Destination unreachable(Type 3, Code 3)이고, IP의 id, UDP의 dst port가 일치하면 종료

2. Traceroute

2.2 ICMP query messages

Echo request [edit]

The *echo request* ("ping") is an ICMP/ICMP6 message.

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Type = 8(IPv4, ICMP) 128(IPv6,ICMP6)								Code = 0								Checksum															
Identifier																Sequence Number															
Payload																															

Echo reply [edit]

The *echo reply* is an ICMP message generated in response to an echo request; it is mandatory for all hosts, and must include the exact payload received in the request.

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Type = 0(IPv4,ICMP) 129(IPv6,ICMP6)								Code = 0								Checksum															
Identifier																Sequence Number															
Payload																															

2. Traceroute

2.3 ICMP error messages

Time exceeded message[5]:5

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Type = 11								Code								Header checksum															
unused																															
IP header and first 8 bytes of original datagram's data																															

Destination unreachable message[5]:3

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Type = 3								Code								Header checksum															
unused																Next-hop MTU															
IP header and first 8 bytes of original datagram's data																															

2. Traceroute

2.3 F.Y.I

TCP를 이용한 Traceroute

1. IP 헤더 만들기(TTL=n, dst=host, id=random)
2. TCP 헤더 만들기(src port, dst port=80, option[mss, sack, windows scale, nop])
   - tip: dst port 해당 호스트에서 사용하고 있는 포트여야 한다.ex) 80, 443, 22, 25…
3. 패킷 전송 (IP + UDP)
4. 응답 수신(timeout)
5. Time Exceeded(Type 11, Code 0)이고 포함된 IP 헤더가 일치하면 중간경로
6. Destination unreachable(Type 3, Code 3)이고, IP의 id, UDP의 dst port가 일치하면 종료

Term Project

- Traceroute 작성
  - send 소켓 : socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_RAW)
  - struct 모듈을 사용해 직접 IP, ICMP, UDP의 내용 작성
  - host : 목적지 ip 주소 혹은 도메인 네임, size : 패킷의 사이즈(IP헤더부터)
  - -t : RECV TIMEOUT, -c : MAX_HOPS
  - -I : ICMP, -U : UDP, -p : UDP 포트번호(기본 53)
  - 스니핑할 때 자신이 보낸 UDP, ICMP인지 확인하는 로직 작성
- 팀 대표가 barcel@naver.com으로 제출 (6.11까지)
  - Title : [컴퓨터네트워크][학번][이름][과제_N]
  - Content : github repo url
    팀명 : 길동이네
    팀원 : 홍길동(학번), 고길동(학번)

2. Traceroute

2.4 Traceroute

```
root@ubuntu:/home/famous/TA# python3 traceroute.py google.com 100
traceroute to google.com (172.217.31.238), 30 hops max, 100 byte packets
1    0.3 ms   0.35 ms   0.32 ms [_gateway, 192.168.200.2]
2    *   *   *
3    *   *   *
4    *   *   *
5    *   *   *
6    *   *   *
7    *   *   *
8    *   *   *
9    *   *   *
10   *   *   *
11   *   *   *
12   *   *   *
13   *   *   *
14   *   *   *
15   *   *   *
16   *   *   *
17   2.48 ms   2.76 ms   1.45 ms [hkg07s28-in-f14.1e100.net, 172.217.31.238]
root@ubuntu:/home/famous/TA#
```

2. Traceroute

2.4 Traceroute