ICMP

인터터넷의 동작은 라우터들에 의해 면밀히 감시된다. 예기치 못한 어떤 일이 발생할 때, 그 이벤트는 인터넷 제어 메시지 프로토콜(Internet Control Message Protocol, ICMP)에 의해 통보된다. 그리고 이것은 또한 인터넷을 검사하는데도 사용된다. 약 12개 정도의 ICMP 메시지 타입이 정의되어 있다. 각각의 ICMP 메시지 타입은 IP 패킷에 캡슐화되어 있다.

ICMP는 에러메시지나 제어메시지를 전송하는 프로토콜로 TCP/IP에 접속된 컴퓨터나 네트워크 기기들 간에 서로의 상태를 확인하기 위해 사용된다. 우리가 무심코 컴퓨터의 상태를 진단하기 위해 사용하는 ping 또한 ICMP 를 이용한다.

Cont'd

Message Type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Inavalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo	Ask a machine if it a alive
Echo Reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

recv() & send()

1. read & write 함수와 달리 데이터 입 출력 방법에 있어서 옵션 부여.

```
#include <sys/types.h>
#include <sys/socket.h>

int recv(int sock, void * buf, int len, unsigned int flags);
int send(int sock, const void * buf, int len, unsigned int flags);
```

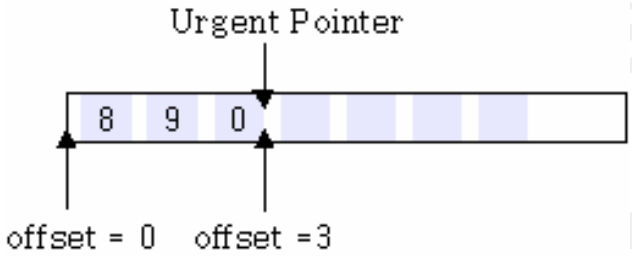
2. 옵션의 종류와 의미.

[표 13-1]

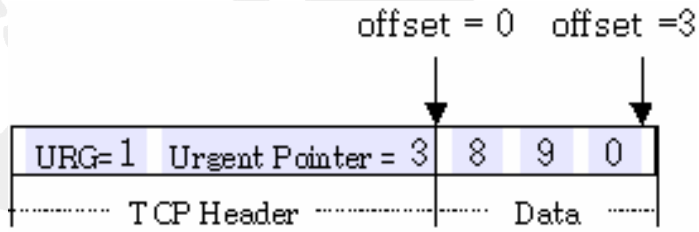
flags	Description	recv	send
MSG_DONTROUTE	데이터 전송 시 라우팅 테이블 참조 하지 않음.		●
MSG_DONTWAIT	넌-블록킹(non-blocking) I/O	●	●
MSG_OOB	데이터 전송 시, 긴급 데이터(Out-of-band data) 전송	●	●
MSG_PEEK	버퍼에 데이터 유 무 확인	●	

긴급 데이터 전송 시 생성되는 패킷

1. 긴급 데이터 전송을 위한 패킷의 형성



[그림 13-4 : 출력 버퍼]



[그림 13-4 : TCP 패킷]

2. 긴급 데이터 전송의 특징.
   - 버퍼 상황에 관계 없이 데이터 전송
   - Nagle 알고리즘 무시

긴급 데이터 전송 예제[MSG_OOB]

[oob_send.c]

```
write(sock, "123", 3);
send(sock, "4", 1, MSG_OOB); //긴급 메시지 송신
write(sock, "567", 3);
send(sock, "890", 3, MSG_OOB); //긴급메시지 송신, urgent point –offset:3, “0”송신
```

[oob.recv.c]

```
29: struct sigaction act;  // 긴급 메시지 처리위해 구조체 선언

37: act.sa_handler=urg_handler;  //긴급메시지 수신 핸들 대입
    sigemptyset(&act.sa_mask);
    act.sa_flags=0;

57:  fcntl(send_sock, F_SETOWN, getpid()); //소켓의 핸들 소유를 현재 프로세스 대입
    state=sigaction(SIGURG, &act, 0);
    if(state != 0)
        error_handling("sigaction() error ");

62: while( (str_len=recv(send_sock, buf, sizeof(buf), 0)) != 0) { // 일반 메시지

72: void urg_handler(int signo) //긴급 메시지 루틴선언
//urgent point는 긴급 수신 메시지 마지막 바이트 다음 위치 설정됨
76: str_len=recv(send_sock, buf, sizeof(buf)-1, MSG_OOB); // 긴급 메시지 처리
```

긴급 데이터 전송 예제[MSG_OOB]

root@localhost.localdomain: /booksource/advanIO

```
[root@localhost advanIO]# gcc oob_send.c -o send
[root@localhost advanIO]# ./send 127.0.0.1 9190
[root@localhost advanIO]#


[영어][완성][두벌식]
```

root@localhost.localdomain: /booksource/advanIO

```
[root@localhost advanIO]# gcc oob_recv.c -o recv
[root@localhost advanIO]# ./recv 9190
123
긴급 메시지 전송 : 4
긴급 메시지 전송 : 0
56789
[root@localhost advanIO]#


[영어][완성][두벌식]
```