Laborator 5 - ICMP

Pachet cu care lucram:

ETH HEADER	**IP HEADER**	**ICMP HEADER**	**DATE**

Header IP (din linux/ip.h)

```
struct iphdr {
#if defined(__LITTLE_ENDIAN_BITFIELD)
      __u8  ihl:4,
            version:4;
#elif defined (__BIG_ENDIAN_BITFIELD)
      __u8  version:4,
            ihl:4;
#else
#error      "Please fix <asm/byteorder.h>"
#endif
      __u8  tos;
      __u16 tot_len;
      __u16 id;
      __u16 frag_off;
      __u8  ttl;
      __u8  protocol;
      __u16 check;
      __u32 saddr;
      __u32 daddr;
      /*The options start here. */
};
```

.version = 4
.ihl = 5
.tos = 0
.tot_len = **htons**(dimensiunea totala a pachetului MINUS dimensiunea header-ului ETH)
.id = **htons**(getpid()) // **htons(25)**
.frag_off = **htons(0)**
.ttl = 64
.protocol = IPPROTO_ICMP (8)
get_interface_ip(sockfd, IFNAME, &ip_hdr->saddr);
.daddr = ip_addr.s_addr
.check = 0
.check = checksum(ip_hdr, sizeof(struct iphdr))

Header ICMP (din netinet/ip_icmp.h)

```
struct icmphdr
{
  u_int8_t type;                /* message type */
  u_int8_t code;                /* type sub-code */
  u_int16_t checksum;
  union
  {
    struct
    {
      u_int16_t        id;
      u_int16_t        sequence;
    } echo;                         /* echo datagram */
    u_int32_t        gateway;        /* gateway address */
    struct
    {
      u_int16_t        __unused;
      u_int16_t        mtu;
    } frag;                         /* path mtu discovery */
  } un;
};
```

.type = ICMP_ECHO
.code = 0
.un.echo.id = **htons**(getpid()) // **htons(25)**
.un.echo.sequence = **htons**(i + 1)
.checksum = 0
.checksum = checksum(...)

BIG ENDIAN -> MSB first
LITTLE ENDIAN -> LSB first

0x 00 11 (17 in decimal)

00 11 (Big Endian)
11 00 -> 0x1100 -> 4000+

Network order = Big Endian
Host order

Cand trimit: transform in NETWORK ORDER (htons - 16 biti, htonl - 32 biti)
Cand primesc: transform in HOST ORDER (ntohs - 16 biti, ntohl - 32 biti)

S -> Hop1 -> Hop2 -> Hop3 -> D

.type = ICMP_TIME_EXCEEDED

For (i = 0; i < count; i++)
Send

While (.type == TIME_EXCEEDED)
send