```
/* Copyright (C) 1991, 92, 93, 95, 96, 97, 99 Free Software Foundation, Inc.
   This file is part of the GNU C Library.

   The GNU C Library is free software; you can redistribute it and/or
   modify it under the terms of the GNU Lesser General Public
   License as published by the Free Software Foundation; either
   version 2.1 of the License, or (at your option) any later version.

   The GNU C Library is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
   Lesser General Public License for more details.

   You should have received a copy of the GNU Lesser General Public
   License along with the GNU C Library; if not, write to the Free
   Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
   02111-1307 USA.  */

#ifndef __NETINET_IP_ICMP_H
#define __NETINET_IP_ICMP_H    1

#include <sys/cdefs.h>
#include <sys/types.h>

__BEGIN_DECLS

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;
};

#define ICMP_ECHOREPLY          0       /* Echo Reply                   */
#define ICMP_DEST_UNREACH       3       /* Destination Unreachable      */
#define ICMP_SOURCE_QUENCH      4       /* Source Quench                */
#define ICMP_REDIRECT           5       /* Redirect (change route)      */
#define ICMP_ECHO               8       /* Echo Request                 */
#define ICMP_TIME_EXCEEDED      11      /* Time Exceeded                */
#define ICMP_PARAMETERPROB      12      /* Parameter Problem            */
#define ICMP_TIMESTAMP          13      /* Timestamp Request            */
#define ICMP_TIMESTAMPREPLY     14      /* Timestamp Reply              */
#define ICMP_INFO_REQUEST       15      /* Information Request          */
#define ICMP_INFO_REPLY         16      /* Information Reply            */
#define ICMP_ADDRESS            17      /* Address Mask Request         */
#define ICMP_ADDRESSREPLY       18      /* Address Mask Reply           */
#define NR_ICMP_TYPES           18


/* Codes for UNREACH. */
```

```
#define ICMP_NET_UNREACH        0       /* Network Unreachable          */
#define ICMP_HOST_UNREACH       1       /* Host Unreachable             */
#define ICMP_PROT_UNREACH       2       /* Protocol Unreachable         */
#define ICMP_PORT_UNREACH       3       /* Port Unreachable             */
#define ICMP_FRAG_NEEDED        4       /* Fragmentation Needed/DF set  */
#define ICMP_SR_FAILED          5       /* Source Route failed          */
#define ICMP_NET_UNKNOWN        6
#define ICMP_HOST_UNKNOWN       7
#define ICMP_HOST_ISOLATED      8
#define ICMP_NET_ANO            9
#define ICMP_HOST_ANO           10
#define ICMP_NET_UNR_TOS        11
#define ICMP_HOST_UNR_TOS       12
#define ICMP_PKT_FILTERED       13      /* Packet filtered */
#define ICMP_PREC_VIOLATION     14      /* Precedence violation */
#define ICMP_PREC_CUTOFF        15      /* Precedence cut off */
#define NR_ICMP_UNREACH         15      /* instead of hardcoding immediate value */

/* Codes for REDIRECT. */
#define ICMP_REDIR_NET          0       /* Redirect Net                 */
#define ICMP_REDIR_HOST         1       /* Redirect Host                */
#define ICMP_REDIR_NETTOS       2       /* Redirect Net for TOS         */
#define ICMP_REDIR_HOSTTOS      3       /* Redirect Host for TOS        */

/* Codes for TIME_EXCEEDED. */
#define ICMP_EXC_TTL            0       /* TTL count exceeded           */
#define ICMP_EXC_FRAGTIME       1       /* Fragment Reass time exceeded */


#ifdef __USE_BSD
/*
 * Copyright (c) 1982, 1986, 1993
 *      The Regents of the University of California.  All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 4. Neither the name of the University nor the names of its contributors
 *    may be used to endorse or promote products derived from this software
 *    without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 *      @(#)ip_icmp.h   8.1 (Berkeley) 6/10/93
 */

#include <netinet/in.h>
#include <netinet/ip.h>

/*
```

```
   * Internal of an ICMP Router Advertisement
   */
  struct icmp_ra_addr
  {
    u_int32_t ira_addr;
    u_int32_t ira_preference;
  };

  struct icmp
  {
    u_int8_t  icmp_type;  /* type of message, see below */
    u_int8_t  icmp_code;  /* type sub code */
    u_int16_t icmp_cksum; /* ones complement checksum of struct */
    union
    {
      u_char ih_pptr;             /* ICMP_PARAMPROB */
      struct in_addr ih_gwaddr;   /* gateway address */
      struct ih_idseq             /* echo datagram */
      {
        u_int16_t icd_id;
        u_int16_t icd_seq;
      } ih_idseq;
      u_int32_t ih_void;

      /* ICMP_UNREACH_NEEDFRAG -- Path MTU Discovery (RFC1191) */
      struct ih_pmtu
      {
        u_int16_t ipm_void;
        u_int16_t ipm_nextmtu;
      } ih_pmtu;

      struct ih_rtradv
      {
        u_int8_t irt_num_addrs;
        u_int8_t irt_wpa;
        u_int16_t irt_lifetime;
      } ih_rtradv;
    } icmp_hun;
  #define icmp_pptr       icmp_hun.ih_pptr
  #define icmp_gwaddr     icmp_hun.ih_gwaddr
  #define icmp_id         icmp_hun.ih_idseq.icd_id
  #define icmp_seq        icmp_hun.ih_idseq.icd_seq
  #define icmp_void       icmp_hun.ih_void
  #define icmp_pmvoid     icmp_hun.ih_pmtu.ipm_void
  #define icmp_nextmtu    icmp_hun.ih_pmtu.ipm_nextmtu
  #define icmp_num_addrs  icmp_hun.ih_rtradv.irt_num_addrs
  #define icmp_wpa        icmp_hun.ih_rtradv.irt_wpa
  #define icmp_lifetime   icmp_hun.ih_rtradv.irt_lifetime
    union
    {
      struct
      {
        u_int32_t its_otime;
        u_int32_t its_rtime;
        u_int32_t its_ttime;
      } id_ts;
      struct
      {
        struct ip idi_ip;
        /* options and then 64 bits of data */
      } id_ip;
      struct icmp_ra_addr id_radv;
      u_int32_t   id_mask;
      u_int8_t    id_data[1];
    } icmp_dun;
```

```
#define icmp_otime      icmp_dun.id_ts.its_otime
#define icmp_rtime      icmp_dun.id_ts.its_rtime
#define icmp_ttime      icmp_dun.id_ts.its_ttime
#define icmp_ip         icmp_dun.id_ip.idi_ip
#define icmp_radv       icmp_dun.id_radv
#define icmp_mask       icmp_dun.id_mask
#define icmp_data       icmp_dun.id_data
};

/*
 * Lower bounds on packet lengths for various types.
 * For the error advice packets must first insure that the
 * packet is large enough to contain the returned ip header.
 * Only then can we do the check to see if 64 bits of packet
 * data have been returned, since we need to check the returned
 * ip header length.
 */
#define ICMP_MINLEN     8                               /* abs minimum */
#define ICMP_TSLEN      (8 + 3 * sizeof (n_time))       /* timestamp */
#define ICMP_MASKLEN    12                              /* address mask */
#define ICMP_ADVLENMIN  (8 + sizeof (struct ip) + 8)    /* min */
#ifndef _IP_VHL
#define ICMP_ADVLEN(p)  (8 + ((p)->icmp_ip.ip_hl << 2) + 8)
        /* N.B.: must separately check that ip_hl >= 5 */
#else
#define ICMP_ADVLEN(p)  (8 + (IP_VHL_HL((p)->icmp_ip.ip_vhl) << 2) + 8)
        /* N.B.: must separately check that header length >= 5 */
#endif

/* Definition of type and code fields. */
/* defined above: ICMP_ECHOREPLY, ICMP_REDIRECT, ICMP_ECHO */
#define ICMP_UNREACH            3               /* dest unreachable, codes: */
#define ICMP_SOURCEQUENCH       4               /* packet lost, slow down */
#define ICMP_ROUTERADVERT       9               /* router advertisement */
#define ICMP_ROUTERSOLICIT      10              /* router solicitation */
#define ICMP_TIMXCEED           11              /* time exceeded, code: */
#define ICMP_PARAMPROB          12              /* ip header bad */
#define ICMP_TSTAMP             13              /* timestamp request */
#define ICMP_TSTAMPREPLY        14              /* timestamp reply */
#define ICMP_IREQ               15              /* information request */
#define ICMP_IREQREPLY          16              /* information reply */
#define ICMP_MASKREQ            17              /* address mask request */
#define ICMP_MASKREPLY          18              /* address mask reply */

#define ICMP_MAXTYPE            18

/* UNREACH codes */
#define ICMP_UNREACH_NET                0       /* bad net */
#define ICMP_UNREACH_HOST               1       /* bad host */
#define ICMP_UNREACH_PROTOCOL           2       /* bad protocol */
#define ICMP_UNREACH_PORT               3       /* bad port */
#define ICMP_UNREACH_NEEDFRAG           4       /* IP_DF caused drop */
#define ICMP_UNREACH_SRCFAIL            5       /* src route failed */
#define ICMP_UNREACH_NET_UNKNOWN        6       /* unknown net */
#define ICMP_UNREACH_HOST_UNKNOWN       7       /* unknown host */
#define ICMP_UNREACH_ISOLATED           8       /* src host isolated */
#define ICMP_UNREACH_NET_PROHIB         9       /* net denied */
#define ICMP_UNREACH_HOST_PROHIB        10      /* host denied */
#define ICMP_UNREACH_TOSNET             11      /* bad tos for net */
#define ICMP_UNREACH_TOSHOST            12      /* bad tos for host */
#define ICMP_UNREACH_FILTER_PROHIB      13      /* admin prohib */
#define ICMP_UNREACH_HOST_PRECEDENCE    14      /* host prec vio. */
#define ICMP_UNREACH_PRECEDENCE_CUTOFF  15      /* prec cutoff */

/* REDIRECT codes */
```

```
#define ICMP_REDIRECT_NET       0               /* for network */
#define ICMP_REDIRECT_HOST      1               /* for host */
#define ICMP_REDIRECT_TOSNET    2               /* for tos and net */
#define ICMP_REDIRECT_TOSHOST   3               /* for tos and host */

/* TIMEXCEED codes */
#define ICMP_TIMXCEED_INTRANS   0               /* ttl==0 in transit */
#define ICMP_TIMXCEED_REASS     1               /* ttl==0 in reass */

/* PARAMPROB code */
#define ICMP_PARAMPROB_OPTABSENT 1              /* req. opt. absent */

#define ICMP_INFOTYPE(type) \
        ((type) == ICMP_ECHOREPLY || (type) == ICMP_ECHO || \
        (type) == ICMP_ROUTERADVERT || (type) == ICMP_ROUTERSOLICIT || \
        (type) == ICMP_TSTAMP || (type) == ICMP_TSTAMPREPLY || \
        (type) == ICMP_IREQ || (type) == ICMP_IREQREPLY || \
        (type) == ICMP_MASKREQ || (type) == ICMP_MASKREPLY)

#endif /* __USE_BSD */

__END_DECLS

#endif /* netinet/ip_icmp.h */
```