

File created: 17-Aug-90 12:34:42 {DSK}<LISPFILERS>ETHERNET>TCP>NEW>TCPLLICMP.;2

changes to: (FNS PRINTICMP \ICMP.HANDLE.REDIRECT \ICMP.INPUT)

previous date: 6-Jan-89 16:38:06 {DSK}<LISPFILERS>ETHERNET>TCP>NEW>TCPLLICMP.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1985, 1986, 1987, 1988, 1989, 1990 by Xerox Corporation. All rights reserved.

(RPAQQ TCPLLICMPCOMS

```
[(COMS (* * ICMP functions)
  (DECLARE%: DONTCOPY (EXPORT (RECORDS ICMPADMASK ICMP ICMPPECHO ICMPDESTUN ICMPREDIRECT)
    (CONSTANTS * ICMPTYPES)
    (CONSTANTS * ICMPUNREACHABLES)
    (CONSTANTS * ICMPREDIRECTS)
    (CONSTANTS \ICMPOVLEN)
    (CONSTANTS \ICMP.PROTOCOL)
    (MACROS ICMPLENGTH)))
  (DECLARE%: EVAL@COMPILE DONTCOPY (MACROS IP.FROM.ICMP))
  (INITVARS * ICMPTIMEXS)
  (INITVARS (\ICMP.ECHO.REPLY.QUEUE (NCREATE 'SYSQUEUE))
    (\ICMP.ECHO.REPLY.EVENT (CREATE.EVENT "ICMP Echo reply"))
    (\ICMP.ECHOING))
  (GLOBALVARS \ICMP.ECHO.REPLY.QUEUE \ICMP.ECHO.REPLY.EVENT \ICMP.ECHOING)
  (FNS PRINTICMP \ICMP.DEST.UNREACHABLE \ICMP.REDIRECT \ICMP.ECHO.TEST \ICMP.HANDLE.ECHO.REPLY
    \ICMP.HANDLE.REDIRECT \ICMP.INPUT \ICMP.REPLY.TO.ECHO \ICMP.SETUPICMP \ICMP.TIME.EXCEEDED
    \ICMP.TRANSMIT)
  (FNS ICMP.HANDLE.ADDRESS.MASK \ICMP.INPUT \ICMP.REQUEST.ADDRESS.MASK)
  (ADDVARS (IPPRINTMACROS (1 . PRINTICMP))
```

(* * ICMP functions)

(DECLARE%: DONTCOPY

:: FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

```
[ACCESSFNS ICMPADMASK ((ICMPADMASKBASE (fetch (ICMP ICMPCONTENTS) of DATUM)))
  (BLOCKRECORD ICMPADMASKBASE ((ICMPADMASKID WORD)
    (ICMPADMASKSEQNO WORD)
    (ICMPADMASKADMASK FIXP]
```

```
[ACCESSFNS ICMP ((ICMPBASE (\IPDATABASE DATUM)))
  (BLOCKRECORD ICMPBASE ((ICMPSTYPE BYTE)
    (ICMPCODE BYTE)
    (ICMPCHECKSUM WORD)
    (ICMPDATASTART WORD)))
  (ACCESSFNS ICMP ((ICMPCONTENTS (LOC (fetch (ICMP ICMPDATASTART) of DATUM]
```

```
[ACCESSFNS ICMPPECHO ((ICMPPECHOBASE (fetch (ICMP ICMPCONTENTS) of DATUM)))
  (BLOCKRECORD ICMPPECHOBASE ((ICMPPECHOID WORD)
    (ICMPPECHOSEQNO WORD)
    (ICMPPECHODATA BYTE]
```

```
[ACCESSFNS ICMPDESTUN ((ICMPPECHOBASE (fetch (ICMP ICMPCONTENTS) of DATUM)))
  (BLOCKRECORD ICMPPECHOBASE ((NIL FIXP)
    (ICMPIPSTART WORD)))
  (ACCESSFNS ICMPDESTUN ((ICMPIPHEADER (LOC (fetch (ICMPDESTUN ICMPIPSTART) of DATUM]
```

```
[ACCESSFNS ICMPREDIRECT ((ICMPREDIRECTBASE (fetch (ICMP ICMPCONTENTS) of DATUM)))
  (BLOCKRECORD ICMPREDIRECTBASE ((ICMPGATEWAY FIXP)
    (ICMPIPSTART WORD)))
  (ACCESSFNS ICMPREDIRECT ((ICMPIPHEADER (LOC (fetch (ICMPREDIRECT ICMPIPSTART) of DATUM]
```

(RPAQQ ICMPTYPES

```
((\ICMP.ECHO.REPLY 0)
(\ICMP.DEST.UNREACHABLE 3)
(\ICMP.SOURCE.QUENCH 4)
(\ICMP.REDIRECT 5)
(\ICMP.ECHO 8)
(\ICMP.TIME.EXCEEDED 11)
(\ICMP.PARAMETER.PROBLEM 12)
(\ICMP.TIMESTAMP 13)
(\ICMP.TIMESTAMP.REPLY 14)
(\ICMP.INFO.REQUEST 15)
(\ICMP.INFO.REPLY 16)
(\ICMP.ADDRESS.MASK.REQUEST 17)
```

```

(\ICMP.ADDRESS.MASK.REPLY 18)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \ICMP.ECHO.REPLY 0)

(RPAQQ \ICMP.DEST.UNREACHABLE 3)

(RPAQQ \ICMP.SOURCE.QUENCH 4)

(RPAQQ \ICMP.REDIRECT 5)

(RPAQQ \ICMP.ECHO 8)

(RPAQQ \ICMP.TIME.EXCEEDED 11)

(RPAQQ \ICMP.PARAMETER.PROBLEM 12)

(RPAQQ \ICMP.TIMESTAMP 13)

(RPAQQ \ICMP.TIMESTAMP.REPLY 14)

(RPAQQ \ICMP.INFO.REQUEST 15)

(RPAQQ \ICMP.INFO.REPLY 16)

(RPAQQ \ICMP.ADDRESS.MASK.REQUEST 17)

(RPAQQ \ICMP.ADDRESS.MASK.REPLY 18)

(CONSTANTS (\ICMP.ECHO.REPLY 0)
  (\ICMP.DEST.UNREACHABLE 3)
  (\ICMP.SOURCE.QUENCH 4)
  (\ICMP.REDIRECT 5)
  (\ICMP.ECHO 8)
  (\ICMP.TIME.EXCEEDED 11)
  (\ICMP.PARAMETER.PROBLEM 12)
  (\ICMP.TIMESTAMP 13)
  (\ICMP.TIMESTAMP.REPLY 14)
  (\ICMP.INFO.REQUEST 15)
  (\ICMP.INFO.REPLY 16)
  (\ICMP.ADDRESS.MASK.REQUEST 17)
  (\ICMP.ADDRESS.MASK.REPLY 18))
)

(RPAQQ ICMPUNREACHABLES ((\ICMP.NET.UNREACHABLE 0)
  (\ICMP.HOST.UNREACHABLE 1)
  (\ICMP.PROTOCOL.UNREACHABLE 2)
  (\ICMP.PORT.UNREACHABLE 3)
  (\ICMP.CANT.FRAGMENT 4)
  (\ICMP.SOURCE.ROUTE 5)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \ICMP.NET.UNREACHABLE 0)

(RPAQQ \ICMP.HOST.UNREACHABLE 1)

(RPAQQ \ICMP.PROTOCOL.UNREACHABLE 2)

(RPAQQ \ICMP.PORT.UNREACHABLE 3)

(RPAQQ \ICMP.CANT.FRAGMENT 4)

(RPAQQ \ICMP.SOURCE.ROUTE 5)

(CONSTANTS (\ICMP.NET.UNREACHABLE 0)
  (\ICMP.HOST.UNREACHABLE 1)
  (\ICMP.PROTOCOL.UNREACHABLE 2)
  (\ICMP.PORT.UNREACHABLE 3)
  (\ICMP.CANT.FRAGMENT 4)
  (\ICMP.SOURCE.ROUTE 5))
)

(RPAQQ ICMPREDIRECTS ((\ICMP.REDIRECT.NET 0)
  (\ICMP.REDIRECT.HOST 1)
  (\ICMP.REDIRECT.SVC.AND.NET 2)
  (\ICMP.REDIRECT.SVC.AND.HOST 3)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \ICMP.REDIRECT.NET 0)

(RPAQQ \ICMP.REDIRECT.HOST 1)

(RPAQQ \ICMP.REDIRECT.SVC.AND.NET 2)

```

```

{MEDLEY}<obsolete>tcp>TCPLLICMP.;1

(RPAQQ \ICMP.REDIRECT.SVC.AND.HOST 3)

(CONSTANTS (\ICMP.REDIRECT.NET 0)
  (\ICMP.REDIRECT.HOST 1)
  (\ICMP.REDIRECT.SVC.AND.NET 2)
  (\ICMP.REDIRECT.SVC.AND.HOST 3))
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \ICMPOVLEN 4)

(CONSTANTS \ICMPOVLEN)
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \ICMP.PROTOCOL 1)

(CONSTANTS \ICMP.PROTOCOL)
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS ICMPLENGTH MACRO [LAMBDA (ICMP)
  (IDIFFERENCE (fetch (IP IPTOTALLENGTH) of ICMP)
    (LLSH (fetch (IP IPHEADERLENGTH) of ICMP)
      2]))
)
)

;; END EXPORTED DEFINITIONS

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(PUTPROPS IP.FROM.ICMP MACRO [OPENLAMBDA (PKT)
  ;; Returns a pointer to the 'Internet header + 64 bits' found in an ICMP packet, offset so that it looks like an IP
  ;; record. I.e., add to the base the size of the IP header + ICMP header
  (\ADDBASE PKT (+ (UNFOLD (fetch (IP IPHEADERLENGTH) of PKT)
    WORDSPERCELL)
    (CONSTANT (+ (FOLDHI \ICMPOVLEN BYTESPERWORD)
      2]))
)
)

(RPAQQ ICMPTIMEXS ((\ICMP.TRANSIT.TIME.EXCEEDED 0)
  (\ICMP.FRAGMENT.TIME.EXCEEDED 1)))

(RPAQ? \ICMP.TRANSIT.TIME.EXCEEDED 0)

(RPAQ? \ICMP.FRAGMENT.TIME.EXCEEDED 1)

(RPAQ? \ICMP.ECHO.REPLY.QUEUE (NCREATE 'SYSQUEUE))

(RPAQ? \ICMP.ECHO.REPLY.EVENT (CREATE.EVENT "ICMP Echo reply"))

(RPAQ? \ICMP.ECHOING )

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \ICMP.ECHO.REPLY.QUEUE \ICMP.ECHO.REPLY.EVENT \ICMP.ECHOING)
)

(DEFINEQ

(PRINTICMP
  [LAMBDA (ICMP FILE)
    (LET ((*PRINT-BASE* 10)
      (TYPE (fetch (ICMP ICMPTYPE) of ICMP))
      (CODE (fetch (ICMP ICMPCODE) of ICMP)))
      (PRINTCONSTANT TYPE ICMPTYPES FILE "\ICMP.")
      (SPACES 1 FILE)
      (SELECTC TYPE
        (\ICMP.REDIRECT
          (PRINTCONSTANT CODE ICMPREDIRECTS FILE "\ICMP.REDIRECT.")
          (PRINTOUT FILE " " (\IP.ADDRESS.TO.STRING (fetch (ICMPREDIRECT ICMPGATEWAY) of ICMP))))
        (\ICMP.DEST.UNREACHABLE
          (PRINTCONSTANT CODE ICMPUNREACHABLES FILE "\ICMP.")
          (PRIN3 CODE FILE))
        (TERPRI FILE]))
    ; Edited 13-Sep-88 11:35 by bvm

\ICMP.DEST.UNREACHABLE
  [LAMBDA (PACKET CODE)
    (* ejs%: " 2-Feb-86 11:35")

```

(* * Returns an ICMP unreachable packet of proper code to sender)

```
(PROG ((ICMP (\ALLOCATE.ETHERPACKET))
  NWORDS)
  (\IP.SETUPIP ICMP (fetch (IP IPSOURCEADDRESS) of PACKET)
    0
    (\IP.FIND.PROTOCOL \ICMP.PROTOCOL))
  (\ICMP.SETUPICMP ICMP \ICMP.DEST.UNREACHABLE CODE)
  (SETQ NWORDS (IPLUS (FOLDHI 64 BITS PERWORD)
    (UNFOLD (fetch (IP IPHEADERLENGTH) of PACKET)
      WORDSPERCELL)))
  (\BLT (fetch (ICMPDESTUN ICMPIPHEADER) of ICMP)
    (fetch (IP IPBASE) of PACKET)
    NWORDS)
  (add (fetch (IP IPTOTALLENGTH) of ICMP)
    (UNFOLD NWORDS BYTESPERWORD))
  (\ICMP.TRANSMIT ICMP)
  (\RELEASE.ETHERPACKET PACKET])
```

(\ICMP.REDIRECT

[LAMBDA (PACKET CODE)

(* ejs%: "2-Feb-86 12:13")

(* * Returns an ICMP unreachable packet of proper code to sender)

```
(PROG ((ICMP (\ALLOCATE.ETHERPACKET))
  NWORDS)
  (\IP.SETUPIP ICMP (fetch (IP IPSOURCEADDRESS) of PACKET)
    0
    (\IP.FIND.PROTOCOL \ICMP.PROTOCOL))
  (\ICMP.SETUPICMP ICMP \ICMP.REDIRECT CODE)
  (SETQ NWORDS (IPLUS (FOLDHI 64 BITS PERWORD)
    (UNFOLD (fetch (IP IPHEADERLENGTH) of PACKET)
      WORDSPERCELL)))
  (replace (ICMPREDIRECT ICMPGATEWAY) of ICMP with (OR \IP.DEFAULT.GATEWAY 0))
  (\BLT (fetch (ICMPREDIRECT ICMPIPHEADER) of ICMP)
    (fetch (IP IPBASE) of PACKET)
    NWORDS)
  (add (fetch (IP IPTOTALLENGTH) of ICMP)
    (UNFOLD NWORDS BYTESPERWORD))
  (\ICMP.TRANSMIT ICMP)
  (\RELEASE.ETHERPACKET PACKET])
```

(\ICMP.ECHO.TEST

[LAMBDA (IPADDRESS ECHOSTREAM DATALENGTH)

(* ejs%: "12-May-86 18:01")

(* * An ICMP echo tester)

```
(while (\QUEUEHEAD \ICMP.ECHO.REPLY.QUEUE) do (\RELEASE.ETHERPACKET (\DEQUEUE \ICMP.ECHO.REPLY.QUEUE)))
(resetvar \ICMP.ECHOING T (PROG (ICMP (IPSOCKET (\IP.FIND.PROTOCOL \ICMP.PROTOCOL)))
  (for SEQUENCE from 0 do ((SETQ ICMP (\ALLOCATE.ETHERPACKET))
    (\IP.SETUPIP ICMP (DODIP.HOSTP IPADDRESS)
      0 IPSOCKET)
    (\ICMP.SETUPICMP ICMP \ICMP.ECHO 0)
    (replace (ICMPECHO ICMPECHOID) of ICMP with 0)
    (replace (ICMPECHO ICMPECHOSEQNO) of ICMP with SEQUENCE
      )
    (add (fetch (IP IPTOTALLENGTH) of ICMP)
      4)
    (AND (NUMBERP DATALENGTH)
      (add (fetch (IP IPTOTALLENGTH) of ICMP)
        DATALENGTH))
    (printout ECHOSTREAM "!")
    (\ICMP.TRANSMIT ICMP)
    (AWAIT.EVENT \ICMP.ECHO.REPLY.EVENT \ETHERTIMEOUT)
    (COND
      [(SETQ ICMP (\DEQUEUE \ICMP.ECHO.REPLY.QUEUE))
        (COND
          ((IGREATERP (fetch (ICMPECHO ICMPECHOSEQNO)
            of ICMP)
            SEQUENCE)
            (printout T "ICMP echo out of sequence" T)
            (PRINTPACKET ICMP 'GET ECHOSTREAM)
            (RETURN ICMP))
          (T (printout ECHOSTREAM "+")
            (\RELEASE.ETHERPACKET ICMP)
            (T (printout ECHOSTREAM ".")))]
```

(\ICMP.HANDLE.ECHO.REPLY

[LAMBDA (ICMP)

(* ejs%: "28-Dec-84 09:02")

```
(COND
  (\ICMP.ECHOING (\ENQUEUE \ICMP.ECHO.REPLY.QUEUE ICMP)
```

```
(NOTIFY.EVENT \ICMP.ECHO.REPLY.EVENT))
(T (\RELEASE.ETHERPACKET ICMP])
```

(\ICMP.HANDLE.REDIRECT

[LAMBDA (ICMP)

; Edited 24-Aug-88 16:16 by bvm

;;; Called when a gateway tells us a better route to the destination. There is a code for type of redirect, but it's not obviously meaningful

```
(LET* ((NDB (fetch EPNETWORK of ICMP))
  (GATEWAY (fetch (ICMPREDIRECT ICMPGATEWAY) of ICMP))
  (DESTADDRESS (fetch (IP IPDESTINATIONADDRESS) of (IP.FROM.ICMP ICMP)))
  (DESTNET (\IPNETADDRESS DESTADDRESS)))
;; Store the new route in the routing table
(COND
  [(= DESTNET (fetch (NDB NDBIPNET#) of NDB))
  (LET* ((SOURCEADDRESS (fetch (NDB NDBIPHOST#) of NDB))
    (SUBNETMASK (CDR (SASSOC SOURCEADDRESS \IP.SUBNET.MASKS)))
    (DESTSUBNET (LOGAND DESTADDRESS SUBNETMASK)))
    ;; The dest net is a local net. Either we fouled up in our routing, or the dest net is really a subnet
    (COND
      ((NOT (= DESTSUBNET (LOGAND SOURCEADDRESS SUBNETMASK)))
        ; Yes, this is a redirect for a subnet, if such is possible
        (SPUTASSOC DESTSUBNET GATEWAY \IP.ROUTING.TABLE]
      (T
        ; Non-local net
        (SPUTASSOC DESTNET GATEWAY \IP.ROUTING.TABLE)))
    ;; If it's a 10MB network, see if we have the 10MB address of this gateway, and if not, request the address
    (SELECTQ (fetch (NDB NETTYPE) of NDB)
      (10 (COND
        ((NOT (\AR.TRANSLATE.TO.10MB GATEWAY T))
          (\AR.TRANSLATE.TO.10MB GATEWAY)))
        NIL)
      (\RELEASE.ETHERPACKET ICMP])
```

(\ICMP.INPUT

[LAMBDA (ICMP)

; Edited 25-Aug-88 11:51 by bvm

;;; ICMP packet received

```
(COND
  ((\IP.CHECKSUM.OK (\IPCHECKSUM ICMP (fetch (ICMP ICMPBASE) of ICMP)
    (\IPDATALENGTH ICMP)))
  (SELECTC (fetch (ICMP ICMPTYPE) of ICMP)
    (\ICMP.ECHO.REPLY
      (\ICMP.HANDLE.ECHO.REPLY ICMP))
    (\ICMP.ECHO (\ICMP.REPLY.TO.ECHO ICMP))
    (\ICMP.DEST.UNREACHABLE
      ; Some packet couldn't reach its destination. Tell the protocol
      ; that sent the packet (found in the enclosed header)
      (LET* [(SEGMENT (IP.FROM.ICMP ICMP))
        (PROTOCOL (\IP.FIND.PROTOCOL (fetch (IP IPPROTOCOL) of SEGMENT]
        (COND
          (PROTOCOL (CL:FUNCALL (fetch (IPSOCKET IPSICMPFN) of PROTOCOL)
            ICMP SEGMENT PROTOCOL])
        (\ICMP.REDIRECT
          (\ICMP.HANDLE.REDIRECT ICMP))
        (\ICMP.ADDRESS.MASK.REPLY
          (\ICMP.HANDLE.ADDRESS.MASK ICMP))
        (\RELEASE.ETHERPACKET ICMP)))
    (T (AND IPTRACEFLG (PRINTPACKET ICMP 'ICMPGET IPTRACEFILE "[dropping packet--bad ICMP checksum]"))
```

(\ICMP.REPLY.TO.ECHO

[LAMBDA (ICMP)

(* ejs%: "12-May-86 17:34")

(* * Reply to an echo request)

```
(swap (fetch (IP IPSOURCEADDRESS) of ICMP)
  (fetch (IP IPDESTINATIONADDRESS) of ICMP))
(replace (ICMP ICMPTYPE) of ICMP with \ICMP.ECHO.REPLY)
(replace EPQUEUE of ICMP with 'FREE)
(\ICMP.TRANSMIT ICMP])
```

(\ICMP.SETUPICMP

[LAMBDA (ICMP TYPE CODE)

(* ejs%: "27-Dec-84 19:00")

```
(replace (ICMP ICMPTYPE) of ICMP with TYPE)
(replace (ICMP ICMPCODE) of ICMP with CODE)
(add (fetch (IP IPTOTALLENGTH) of ICMP)
  \ICMPOVLEN])
```

(\ICMP.TIME.EXCEEDED

[LAMBDA (PACKET CODE)

(* ejs%: " 3-Feb-86 11:00")

(* * Returns an ICMP unreachable packet of proper code to sender)

```
(PROG ((ICMP (\ALLOCATE.ETHERPACKET))
  NWORDS)
  (\IP.SETUPIP ICMP (fetch (IP IPSOURCEADDRESS) of PACKET)
    0
    (\IP.FIND.PROTOCOL \ICMP.PROTOCOL))
  (\ICMP.SETUPICMP ICMP \ICMP.TIME.EXCEEDED CODE)
  (SETQ NWORDS (IPLUS (FOLDHI 64 BITSPERWORD)
    (UNFOLD (fetch (IP IPHEADERLENGTH) of PACKET)
      WORDSPERCELL)))
  (\BLT (fetch (ICMPDESTUN ICMPIPHEADER) of ICMP)
    (fetch (IP IPBASE) of PACKET)
    NWORDS)
  (add (fetch (IP IPTOTALLENGTH) of ICMP)
    (UNFOLD NWORDS BYTESPERWORD))
  (\ICMP.TRANSMIT ICMP])
```

(\ICMP.TRANSMIT

[LAMBDA (ICMP)

(* ejs%: "31-Dec-84 14:27")

(* * Checksum and transmit an ICMP packet)

```
(\IP.SET.CHECKSUM ICMP (fetch (ICMP ICMPBASE) of ICMP)
  (\IPDATALENGTH ICMP)
  (LOC (fetch (ICMP ICMPCHECKSUM) of ICMP)))
(\IP.TRANSMIT ICMP])
```

)

(DEFINEQ

(\ICMP.HANDLE.ADDRESS.MASK

[LAMBDA (ICMP)

; Edited 22-Mar-88 18:49 by eweaver

;; Called when an address-mask-reply icmp comes in.

```
(LET* ((FROM (fetch (IP IPSOURCEADDRESS) of ICMP))
  (DESTADDR (fetch (IP IPDESTINATIONADDRESS) of ICMP))
  (LOCALADDR (COND
    ((AND \3MBLOCALNDB (EQ (fetch NDBIPHOST# of \3MBLOCALNDB)
      DESTADDR))
      DESTADDR)
    ((AND \10MBLOCALNDB (EQ (fetch NDBIPHOST# of \10MBLOCALNDB)
      DESTADDR))
      DESTADDR)))
  (MASK (fetch (ICMPADMASK ICMPADMASKADMASK) of ICMP)))
```

(* ;; (CL:FORMAT PROMPTWINDOW "ICMP AdMask from ~a
mask ~a" (IP.ADDRESS.TO.STRING FROM)
(IP.ADDRESS.TO.STRING MASK)))

```
(COND
  ((NULL \IP.DEFAULT.GATEWAY)
   (SETQ \IP.DEFAULT.GATEWAY FROM)))
(COND
  ((NULL (SASSOC DESTADDR \IP.SUBNET.MASKS))
   (CL:PUSH (CONS DESTADDR MASK)
     \IP.SUBNET.MASKS])
```

(\ICMP.INPUT

[LAMBDA (ICMP)

; Edited 25-Aug-88 11:51 by bvm

;;; ICMP packet received

```
(COND
  ((\IP.CHECKSUM.OK (\IPCHECKSUM ICMP (fetch (ICMP ICMPBASE) of ICMP)
    (\IPDATALENGTH ICMP)))
  (SELECTC (fetch (ICMP ICMPTYPE) of ICMP)
    (\ICMP.ECHO.REPLY
     (\ICMP.HANDLE.ECHO.REPLY ICMP))
    (\ICMP.ECHO (\ICMP.REPLY.TO.ECHO ICMP))
    (\ICMP.DEST.UNREACHABLE
```

; Some packet couldn't reach its destination. Tell the protocol
; that sent the packet (found in the enclosed header)

```
(LET* [(SEGMENT (IP.FROM.ICMP ICMP))
  (PROTOCOL (\IP.FIND.PROTOCOL (fetch (IP IPPROTOCOL) of SEGMENT)
    (COND
      (PROTOCOL (CL:FUNCALL (fetch (IPSOCKET IPSICMPFN) of PROTOCOL)
        ICMP SEGMENT PROTOCOL))
    (\ICMP.REDIRECT
     (\ICMP.HANDLE.REDIRECT ICMP))
    (\ICMP.ADDRESS.MASK.REPLY
     (\ICMP.HANDLE.ADDRESS.MASK ICMP))
    (\RELEASE.ETHERPACKET ICMP)))
  (T (AND IPTRACEFLG (PRINTPACKET ICMP 'ICMPGET IPTRACEFILE "[dropping packet--bad ICMP checksum]")))
```

(\ICMP.REQUEST.ADDRESS.MASK

```
[LAMBDA NIL ; Edited 8-Jan-88 15:15 by eweaver
;; Broadcast a request for the subnet mask. The reply is handled asynchronously by
;; \handle-icmp-address-mask.
(LET ((ICMP (\ALLOCATE.ETHERPACKET))
      (IPSOCKET (\IP.FIND.PROTOCOL \ICMP.PROTOCOL))
      (\IP.SETUPIP ICMP 0 0 IPSOCKET)
      (\ICMP.SETUPICMP ICMP \ICMP.ADDRESS.MASK.REQUEST 0)
      (replace (ICMPADMASK ICMPADMASKID) of ICMP with 0)
      (replace (ICMPADMASK ICMPADMASKSEQNO) of ICMP with 0)
      (add (fetch (IP IPTOTALLENGTH) of ICMP)
            4)
      (\ICMP.TRANSMIT ICMP]))
)
(ADDTOTVAR IPPRINTMACROS (1 . PRINTICMP))
(PUTPROPS TCPLLICMP COPYRIGHT ("Xerox Corporation" 1985 1986 1987 1988 1989 1990))
```

FUNCTION INDEX

ICMP.HANDLE.ADDRESS.MASK	6	\ICMP.HANDLE.REDIRECT	5	\ICMP.SETUPICMP	5
PRINTICMP	3	\ICMP.INPUT	5, 6	\ICMP.TIME.EXCEEDED	5
\ICMP.DEST.UNREACHABLE	3	\ICMP.REDIRECT	4	\ICMP.TRANSMIT	6
\ICMP.ECHO.TEST	4	\ICMP.REPLY.TO.ECHO	5		
\ICMP.HANDLE.ECHO.REPLY	4	\ICMP.REQUEST.ADDRESS.MASK	6		

CONSTANT INDEX

\ICMP.ADDRESS.MASK.REPLY	2	\ICMP.NET.UNREACHABLE	2	\ICMP.REDIRECT.SVC.AND.NET	3
\ICMP.ADDRESS.MASK.REQUEST	2	\ICMP.PARAMETER.PROBLEM	2	\ICMP.SOURCE.QUENCH	2
\ICMP.CANT.FRAGMENT	2	\ICMP.PORT.UNREACHABLE	2	\ICMP.SOURCE.ROUTE	2
\ICMP.DEST.UNREACHABLE	2	\ICMP.PROTOCOL	3	\ICMP.TIME.EXCEEDED	2
\ICMP.ECHO	2	\ICMP.PROTOCOL.UNREACHABLE	2	\ICMP.TIMESTAMP	2
\ICMP.ECHO.REPLY	2	\ICMP.REDIRECT	2	\ICMP.TIMESTAMP.REPLY	2
\ICMP.HOST.UNREACHABLE	2	\ICMP.REDIRECT.HOST	3	\ICMPOVLEN	3
\ICMP.INFO.REPLY	2	\ICMP.REDIRECT.NET	3		
\ICMP.INFO.REQUEST	2	\ICMP.REDIRECT.SVC.AND.HOST	3		

VARIABLE INDEX

ICMPREDIRECTS	2	IPPRINTMACROS	7	\ICMP.FRAGMENT.TIME.EXCEEDED	3
ICMPTIMEXS	3	\ICMP.ECHO.REPLY.EVENT	3	\ICMP.TRANSIT.TIME.EXCEEDED	3
ICMPTYPES	1	\ICMP.ECHO.REPLY.QUEUE	3		
ICMPUNREACHABLES	2	\ICMP.ECHOING	3		

RECORD INDEX

ICMP	1	ICMPADMASK	1	ICMPDESTUN	1	ICMPECHO	1	ICMPREDIRECT	1
------------	---	------------------	---	------------------	---	----------------	---	--------------------	---

MACRO INDEX

ICMPLENGTH	3	IP.FROM.ICMP	3
------------------	---	--------------------	---
