File created: 1-May-2021 19:49:18 **{DSK}<home>larry>ilisp>medley>sources>PUP.;2**

changes to: (FNS \PUP.SETTIME CANONICAL.HOSTNAME)
(VARS PUPCOMS)

previous date: 19-Jan-93 11:14:09 {DSK}<home>larry>ilisp>medley>sources>PUP.;1

Read Table: INTERLISP

Package: INTERLISP

Base: 10

Format: XCCS

```
(RPAQQ PUPCOMS
       ((COMS                                                    ; Low level pup
              (DECLARE%: DONTCOPY (EXPORT (RECORDS PUP PUPADDRESS)
                                          (MACROS \LOCALPUPADDRESS \LOCALPUPHOSTNUMBER \LOCALPUPNETNUMBER))
                     (GLOBALVARS \LOCALPUPNETHOST \OLDPUPHOST#))
              (FNS \STARTPUP ASSURE.PUP.READY \FIND.LOCALPUPHOSTNUMBER \PROMPT.FOR.PUP.NUMBER \HANDLE.RAW.PUP
                   \FORWARD.PUP \SETPUPCHECKSUM)
              (INITVARS (\PUP.CHECKSUMFLG T)
                     (\MAX.EPKTS.ON.PUPSOCKET 20Q)
                     (\LOCALPUPNETHOST)
                     (\OLDPUPHOST# 0)))
        (COMS                                                    ; Pup error stuff
              (DECLARE%: DONTCOPY (EXPORT (RECORDS ERRORPUP)
                                          (CONSTANTS * PUPERRORCODES))
                     (GLOBALVARS PUPERRORMESSAGES))
              (VARS PUPERRORMESSAGES)
              (FNS \PUPERROR))
        (COMS                                                    ; Pup utilities
              (FNS SETUPPUP SWAPPUPPORTS GETPUP SENDPUP EXCHANGEPUPS DISCARDPUPS GETPUPWORD \PUPINIT)
              (FNS ETHERHOSTNAME ETHERHOSTNUMBER ETHERPORT BESTPUPADDRESS NETDAYTIME0 \PUP.SETTIME \SETNEWTIME0
                   \NET.SETTIME NETDATE \LOOKUPPORT \PARSE.PORTCONSTANT \FIXLOCALNET)
              (FNS PORTSTRING OCTALSTRING)
              (INITVARS (\ETHERPORTS (HASHARRAY 24Q))
                     (\ETHERTIMEOUT 3720Q)
                     (\MAXETHERTRIES 4)
                     (\PUPCOUNTER 0))
              (GLOBALVARS \ETHERPORTS \PUPCOUNTER))
        (COMS                                                    ; Accessing a PUP's contents
              (FNS CLEARPUP PUTPUPWORD GETPUPBYTE PUTPUPBYTE GETPUPSTRING GETPUPSTREAM PUTPUPSTRING)
              (OPTIMIZERS GETPUPWORD PUTPUPWORD GETPUPBYTE PUTPUPBYTE))
        (COMS                                                    ; Reading property lists from streams
              (FNS READPLIST)
              (INITVARS \READPLIST.READTABLES)
              (GLOBALVARS \READPLIST.READTABLES))
        (COMS (FNS \CANONICAL.HOSTNAME \CANONICALIZE.PUP.HOSTNAME)
              (P                                                 ; Default this for when IP not loaded
                 (MOVD? 'NILL '\CANONICALIZE.IP.HOSTNAME NIL T))
              (ADDVARS (\HOSTNAMES)
                     (\SYSTEMCACHEVARS \HOSTNAMES))
              (GLOBALVARS \HOSTNAMES))
        [COMS                                                    ; PUP allocation
              (EXPORT (MACROS BINDPUPS)
                     (PROP INFO BINDPUPS)
                     (ALISTS (PRETTYPRINTMACROS BINDPUPS]
        (COMS                                                    ; Pup routing
              (FNS \PUPGATELISTENER \HANDLE.PUP.ROUTING.INFO \ROUTE.PUP \LOCATE.PUPNET SORT.PUPHOSTS.BY.DISTANCE
                   \PUPNET.CLOSERP PUPNET.DISTANCE)
              (INITVARS (\PUP.ROUTING.TABLE (CONS))
                     (\PUP.ROUTING.TABLE.RADIUS 2)
                     (\PUPROUTER.PROBECOUNT 0)
                     (\PUPROUTER.PROBETIMER)
                     (\PUPROUTER.PROBEINTERVAL 5670Q)
                     (\PUP.READY)
                     (\PUP.READY.EVENT (CREATE.EVENT "Pup Ready"))
                     (\PUP.READY.LOCK (CREATE.MONITORLOCK "Pup Ready")))
              (ADDVARS (\SYSTEMCACHEVARS \PUP.READY))
              (DECLARE%: DONTCOPY (RECORDS PUPROUTINGINFO)
                     (CONSTANTS \PUP.ROUTINGINFO.WORDS)
                     (GLOBALVARS \PUP.ROUTING.TABLE \PUP.ROUTING.TABLE.RADIUS \PUPROUTER.PROBECOUNT
                            \PUPROUTER.PROBETIMER \PUPROUTER.PROBEINTERVAL \PUP.READY \PUP.READY.EVENT
                            \PUP.READY.LOCK)))
        (COMS                                                    ; Sockets
              (DECLARE%: DONTCOPY (RECORDS PUPSOCKET)
                     (MACROS \PUPSOCKET.FROM#)
                     (GLOBALVARS \PUPSOCKETS.TABLE \MAX.EPKTS.ON.PUPSOCKET \PUP.CHECKSUMFLG))
              (INITRECORDS PUPSOCKET)
              (SYSRECORDS PUPSOCKET)
              (FNS OPENPUPSOCKET CLOSEPUPSOCKET PUPSOCKETNUMBER PUPSOCKETFROMNUMBER PUPSOCKETEVENT
                   \FLUSHPUPSOCQUEUE)
              (FNS \GETMISCSOCKET)
              (GLOBALVARS \MISC.SOCKET \PUPSOCKETS)
              (INITVARS (\MISC.SOCKET)
                     (\PUPSOCKETS)))
        (DECLARE%: DONTCOPY (EXPORT (RECORDS PORT ERRORPUP)
                                   (GLOBALVARS \ETHERWAIT1 \ETHERTIMEOUT \MAXETHERTRIES PUPTRACEFLG
                                          LOGINPASSWORDS)
                                   (GLOBALVARS PUPTRACEFILE PUPONLYTYPES PUPIGNORETYPES PUPPRINTMACROS)
                                   (CONSTANTS (\PUPOVLEN 26Q)
                                          (\MAX.PUPLENGTH 1024Q)
                                          (\TIME.GETPUP 5))
                                   (PROP VARTYPE PUPPRINTMACROS)
                                   (MACROS \GETPUPWORD \PUTPUPWORD \GETPUPBYTE \PUTPUPBYTE)
                                   (CONSTANTS * RAWPUPTYPES)
                                   (INITVARS (PUPTYPES RAWPUPTYPES))
                                   (CONSTANTS * WELLKNOWNPUPSOCKETS))
               (CONSTANTS * PUPCONSTANTS)
               (MACROS PUPDEBUGGING)
               (ALISTS (PUPPRINTMACROS 210Q 214Q 211Q 213Q 201Q 30Q))
               (RECORDS TIMEPUPCONTENTS))
        (COMS                                                    ; echo utilities
              (FNS PUP.ECHOSERVER PUP.ECHOUSER))
        (COMS                                                    ; Peeking
              (FNS \PEEKPUP \MAYBEPEEKPUP)
              (INITVARS (\PEEKPUPNUMBER))
              (DECLARE%: EVAL@COMPILE DONTCOPY (CONSTANTS \ETHERHOSTLOC)
                     (GLOBALVARS \PEEKPUPNUMBER)))
        (COMS                                                    ; Debugging assistance
              (FNS PRINTPUP PRINTPUPROUTE PRINTPUPDATA PRINTERRORPUP PUPTRACE PRINTCONSTANT)
              (INITVARS (PUPTRACEFLG)
                     (PUPTRACEFILE T)
                     (PUPTRACETIME))
              (GLOBALVARS PUPTRACETIME)
              (ADDVARS (PUPPRINTMACROS)
                     (PUPONLYTYPES)
                     (PUPIGNORETYPES))
              (ALISTS (PUPPRINTMACROS 4 220Q 221Q 223Q 224Q)))
        (DECLARE%: DONTEVAL@LOAD (P (\PUPINIT)))
        (DECLARE%: EVAL@COMPILE DONTCOPY (LOCALVARS . T)
               (FILES (LOADCOMP)
                      LLETHER))
        (PROP (MAKEFILE-ENVIRONMENT FILETYPE)
              PUP)))

;; Low level pup

(DECLARE%: DONTCOPY

;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

(ACCESSFNS PUP [(PUPBASE (LOCF (fetch (ETHERPACKET EPBODY) of DATUM]
       [BLOCKRECORD PUPBASE ((PUPLENGTH WORD)
                             (PUPTCONTROL BYTE)
                             (PUPTYPE BYTE)
                             (PUPID FIXP)
                             (PUPDEST WORD)
                             (PUPDESTSOCKET FIXP)
                             (PUPSOURCE WORD)
                             (PUPSOURCESOCKET FIXP)
                             (PUPDATASTART 412Q WORD))
              (BLOCKRECORD PUPBASE ((NIL WORD)
                                    (TYPEWORD WORD)
                                    (PUPIDHI WORD)
                                    (PUPIDLO WORD)
                                    (PUPDESTNET BYTE)
                                    (PUPDESTHOST BYTE)
                                    (PUPDESTSOCKETHI WORD)
                                    (PUPDESTSOCKETLO WORD)
                                    (PUPSOURCENET BYTE)
                                    (PUPSOURCEHOST BYTE)
                                    (PUPSOURCESOCKETHI WORD)
                                    (PUPSOURCESOCKETLO WORD))    ; Temporary extra synonyms
                     (SYNONYM PUPDESTNET (DESTNET))
                     (SYNONYM PUPDESTHOST (DESTHOST))
                     (SYNONYM PUPDESTSOCKETHI (DESTSKTHI))
                     (SYNONYM PUPDESTSOCKETLO (DESTSKTLO))
                     (SYNONYM PUPSOURCENET (SOURCENET))
                     (SYNONYM PUPSOURCEHOST (SOURCEHOST))
                     (SYNONYM PUPSOURCESOCKETHI (SOURCESKTHI))
                     (SYNONYM PUPSOURCESOCKETLO (SOURCESKTLO)))
              (SYNONYM PUPDEST (DEST))
              (SYNONYM PUPDESTSOCKET (DESTSKT))
              (SYNONYM PUPSOURCE (SOURCE))
```

```
               (SYNONYM PUPSOURCESOCKET (SOURCESKT))
               (ACCESSFNS PUPDATASTART ((PUPCONTENTS (LOCF DATUM]
       [ACCESSFNS PUP [(PUPCHECKSUMBASE (fetch PUPBASE of DATUM))
                        (PUPCHECKSUMLOC (\ADDBASE (fetch PUPBASE of DATUM)
                                                (FOLDLO (SUB1 (fetch PUPLENGTH of DATUM))
                                                      BYTESPERWORD]
               (BLOCKRECORD PUPCHECKSUMLOC ((PUPCHECKSUM WORD]
       (TYPE? (type? ETHERPACKET DATUM)))

(ACCESSFNS PUPADDRESS ((PUPNET# (LRSH DATUM 10Q))
                        (PUPHOST# (LOGAND DATUM 377Q)))
       (CREATE (IPLUS (LLSH PUPNET# 10Q)
                       PUPHOST#)))
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS \LOCALPUPADDRESS MACRO (NIL \LOCALPUPNETHOST))

(PUTPROPS \LOCALPUPHOSTNUMBER MACRO (NIL (fetch PUPHOST# of \LOCALPUPNETHOST)))

(PUTPROPS \LOCALPUPNETNUMBER MACRO (NIL (fetch PUPNET# of \LOCALPUPNETHOST)))
)

;; END EXPORTED DEFINITIONS

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \LOCALPUPNETHOST \OLDPUPHOST#)
)
)

(DEFINEQ

(\STARTPUP
  [LAMBDA (EVENT)                                                  ; Edited 15-Jan-88 01:04 by bvm
    (for SOC in \PUPSOCKETS do ;; Flush any pups waiting on existing sockets.  Not only are they stale, but they will have the wrong NDB
                               (\FLUSHPUPSOCQUEUE SOC))
    (ASSURE.PUP.READY EVENT])

(ASSURE.PUP.READY
  [LAMBDA (QUIET)                                                  ; Edited 15-Jan-88 01:03 by bvm
    ;; Assures that Pup software is enabled.  PUP is turned off after exit until somebody indicates a need for it
    (WITH.MONITOR \PUP.READY.LOCK
        [COND
           ((NULL \PUP.READY)
            (PROG ((NDB \LOCALNDBS)
                   (PROC (FIND.PROCESS '\PUPGATELISTENER))
                   MYHOST#)
                  (SETQ \PUP.ROUTING.TABLE (\CLEAR.ROUTING.TABLE \PUP.ROUTING.TABLE))
                  (CLRHASH \ETHERPORTS)
                  (COND
                     ((NULL NDB)
                      (SETQ \PUP.READY 'NO)
                      (SETQ \LOCALPUPNETHOST 0)
                      (AND PROC (DEL.PROCESS PROC))
                      (RETURN)))
              LP  (COND
                     ((NEQ (fetch NDBPUPHOST# of NDB)
                           0)
                      (SETQ MYHOST# (fetch NDBPUPHOST# of NDB)))
                     ([NULL (OR MYHOST# (SETQ MYHOST# (\FIND.LOCALPUPHOSTNUMBER NDB QUIET QUIET]
                      (SETQ \LOCALPUPNETHOST 0)                    ; Don't know our pup number yet, so wait until somebody actually
                                                                   ; asks for pup service
                      (\DEL.PACKET.FILTER (FUNCTION \HANDLE.RAW.PUP))
                      (AND PROC (DEL.PROCESS PROC))
                      (RETURN))
                     (T (replace NDBPUPHOST# of NDB with MYHOST#)))
                  (COND
                     ((SETQ NDB (fetch NDBNEXT of NDB))
                      (GO LP)))
                  (SETQ \LOCALPUPNETHOST (create PUPADDRESS
                                                 PUPNET# _ (fetch NDBPUPNET# of \LOCALNDBS)
                                                 PUPHOST# _ MYHOST#))
                  (SETQ \OLDPUPHOST# MYHOST#)
                  [COND
                     (\10MBFLG (\ADD.PACKET.FILTER (FUNCTION \HANDLE.RAW.3TO10)))
                     (T (\DEL.PACKET.FILTER (FUNCTION \HANDLE.RAW.3TO10]
                  (SETQ \PUPROUTER.PROBECOUNT 5)
                  (SETQ \PUPROUTER.PROBETIMER (SETUPTIMER 0 \PUPROUTER.PROBETIMER))
                                                                   ; This will get gate listener to probe for gateways when it starts
                                                                   ; up.
                  (COND
                     (\GATEWAYFLG (AND PROC (DEL.PROCESS PROC)))
                     (PROC                                        ; Restart proc because it contains local timer that is now garbage
                           (RESTART.PROCESS PROC))
                     (T (ADD.PROCESS '(\PUPGATELISTENER)
                               'RESTARTABLE
                               'SYSTEM
                               'AFTEREXIT \PUP.READY.EVENT)))
                  (\ADD.PACKET.FILTER (FUNCTION \HANDLE.RAW.PUP))
                  (SETQ \PUP.READY T)
                  (NOTIFY.EVENT \PUP.READY.EVENT)
                  (BLOCK)
                  (RETURN T])])
```

```
(\FIND.LOCALPUPHOSTNUMBER
  [LAMBDA (NDB EVENT QUIET)                                       (* bvm%: "26-Jul-84 16:27")

    ;; Finds out our pup address on this 10mb NDB

    (PROG (NEWNUMBER)
          [COND
             [(SETQ NEWNUMBER (\LOOKUPPUPNUMBER \MY.NSHOSTNUMBER NDB))
              (COND
                 (PUPTRACEFLG (printout PUPTRACEFILE "My pup address = " (fetch PUPNET# of NEWNUMBER)
                                     "#"
                                     (fetch PUPHOST# of NEWNUMBER)
                                     "#" T]
             (QUIET (RETURN NIL))
             (T (SETQ NEWNUMBER (\PROMPT.FOR.PUP.NUMBER (AND (EQ EVENT 'AFTERLOGOUT)
                                                               (NEQ \OLDPUPHOST# 0)
                                                               (OCTALSTRING \OLDPUPHOST#]

     ;; Only rely on the host number part of reply.  There is confusion for machines that exist on more than one net

          (RETURN (fetch PUPHOST# of NEWNUMBER])
```

```
(\PROMPT.FOR.PUP.NUMBER
  [LAMBDA (DEFAULT)                                               (* bvm%: "26-Jul-84 16:30")
    (RESETLST
        (PROVIDE.PROMPTING.WINDOW "Ethernet info needed")
        (RESETBUFS (PROG (NEWNUMBER)
                     LP  (SETQ NEWNUMBER (PACK* (PROMPTFORWORD "Please enter my pup host number (in octal):"
                                                       DEFAULT NIL NIL NIL T)
                                                'Q))
                         (TERPRI T)
                         (COND
                            ((OR (NOT (FIXP NEWNUMBER))
                                 (ILEQ NEWNUMBER 0)
                                 (IGEQ NEWNUMBER 376Q))
                             (printout T T "Pup host number must be between 1 and 376" T T)
                             (CLEARBUF T)
                             (FLASHWINDOW (TTYDISPLAYSTREAM))
                             (GO LP)))
                         (RETURN NEWNUMBER))))])
```

```
(\HANDLE.RAW.PUP
  [LAMBDA (PACKET TYPE)                                           ; Edited 26-Feb-91 12:03 by jds

    ;; Handles the arrival of a PUP.  If it is destined for a local socket that has room, queues it there, else releases it

    (COND
       ((EQ TYPE \EPT.PUP)
        [COND
           ((NULL \PUP.READY)
            (RELEASE.PUP PACKET))
           (T (PROG ((NDB (fetch EPNETWORK of PACKET))
                     CSUM PUPSOC DESTNET MYNET)
                    (COND
                       ((NULL NDB)

                        ;; Somehow, there's no network descriptor for this, so punt:

                        (RELEASE.PUP PACKET)
                        (RETURN)))
                    [COND
                       ((AND (NEQ (fetch PUPDESTHOST of PACKET)
                                  (fetch NDBPUPHOST# of NDB))
                             (NEQ (fetch PUPDESTHOST of PACKET)
                                  0))
                        (RETURN (\FORWARD.PUP PACKET]
                    [COND
                       ((AND (NEQ (SETQ DESTNET (fetch PUPDESTNET of PACKET))
                                  (SETQ MYNET (fetch NDBPUPNET# of NDB)))
                             (NEQ MYNET 0)
                             (NEQ DESTNET 0))                     ; Destination net is not us, so packet not for us
                        (RETURN (\FORWARD.PUP PACKET]
                    (COND
                       [[NULL (SETQ PUPSOC (\PUPSOCKET.FROM# (fetch PUPDESTSOCKETHI of PACKET)
                                                   (fetch PUPDESTSOCKETLO of PACKET]
                                                                  ; Packets addressed to non-active sockets are just ignored.
                        (COND
                           (PUPTRACEFLG (PRIN1 '& PUPTRACEFILE)))
                        (COND
                           ((AND (EQ (fetch PUPTYPE of PACKET)
                                     \PT.ECHOME)
                                 (EQ (fetch PUPDESTSOCKETLO of PACKET)
                                     \PUPSOCKET.ECHO)
                                 (EQ (fetch PUPDESTSOCKETHI of PACKET)
                                     0))                         ; Play echo server
                            (replace TYPEWORD of PACKET
                               with (COND
                                       ([AND (NEQ (SETQ CSUM (fetch PUPCHECKSUM of PACKET))
                                                  MASKWORD1'S)
                                             (NEQ CSUM (\CHECKSUM (fetch PUPCHECKSUMBASE of PACKET)
                                                              (SUB1 (FOLDHI (fetch PUPLENGTH of PACKET)
                                                                           BYTESPERWORD]
                                        \PT.IAMBADECHO)
                                       (T \PT.IAMECHO)))
                            (SWAPPUPPORTS PACKET)
                            (replace EPREQUEUE of PACKET with 'FREE)
                            (SENDPUP NIL PACKET))
                           (T (\PUPERROR PACKET \PUPE.NOSOCKET]
                       ((IGEQ (fetch (PUPSOCKET INQUEUELENGTH) of PUPSOC)
                              (fetch (PUPSOCKET PUPSOC#ALLOCATION) of PUPSOC))
                                                                 ; Note that packets are just 'dropped' when the queue overflows.
                        (\PUPERROR PACKET \PUPE.SOCKETFULL))
                       ([AND \PUP.CHECKSUMFLG (NEQ (SETQ CSUM (fetch PUPCHECKSUM of PACKET))
                                                   MASKWORD1'S)
                             (NEQ CSUM (\CHECKSUM (fetch PUPCHECKSUMBASE of PACKET)
                                              (SUB1 (FOLDHI (fetch PUPLENGTH of PACKET)
                                                            BYTESPERWORD]
                        (\PUPERROR PACKET \PUPE.CHECKSUM))
                       (T [COND
                             ((EQ DESTNET 0)                     ; Fill in unspecified destination net (possibly redundantly with
                                                                 ; zero)
                              (replace PUPDESTNET of PACKET with MYNET))
                             ((EQ MYNET 0)

                              ;; Packet of specific destination net has arrived on a socket that we listen to.  If we don't know our own net
                              ;; number, assume sender is telling the truth

                              (replace NDBPUPNET# of NDB with DESTNET)

                              ;; But don't try to set our \LOCALPUPNETHOST if the NDB doesn't know its pup host number.  This can
                              ;; happen when a pup arrives in the interval after the NDB was created and before a \LOOKUPPUPNUMBER
                              ;; call has succeeded.

                              [COND
                                 ((NEQ 0 (fetch NDBPUPHOST# of NDB))
                                  (SETQ \LOCALPUPNETHOST (create PUPADDRESS
                                                                 PUPNET# _ DESTNET
                                                                 PUPHOST# _ (fetch NDBPUPHOST# of NDB]
                                                                 ; This variable only for backward compatibility.  Delete it some
                                                                 ; day
                              (PROG [(ENTRY (OR (\LOCATE.PUPNET DESTNET T)
                                                  (\ADD.ROUTING.TABLE.ENTRY \PUP.ROUTING.TABLE
                                                         (create ROUTING
                                                                RTNET# _ DESTNET]
                                    (replace RTHOPCOUNT of ENTRY with 0)
                                    (replace RTGATEWAY# of ENTRY with NIL)
                                    (replace RTNDB of ENTRY with NDB)
                                    (replace RTRECENT of ENTRY with T]
                          (UNINTERRUPTABLY
                              (\ENQUEUE (fetch (PUPSOCKET INQUEUE) of PUPSOC)
                                     PACKET)
                              (add (fetch (PUPSOCKET INQUEUELENGTH) of PUPSOC)
                                   1)
                              (NOTIFY.EVENT (fetch PUPSOCEVENT of PUPSOC)))]
        T])
```

(\FORWARD.PUP

```
  [LAMBDA (PUP)                                                  (* bvm%: "22-SEP-83 14:24")

    ;; Called when we receive a PUP not addressed to us.  Unless we are a gateway, dump it

    (COND
       (\PEEKPUPNUMBER (\MAYBEPEEKPUP PUP))
       (\GATEWAYFLG (\GATEWAY.FORWARD.PUP PUP))
       (T (COND
             (PUPTRACEFLG (PRINTPUP PUP 'GET NIL "PUP not addressed to this host: ")))
          (\RELEASE.ETHERPACKET PUP])
```

(\SETPUPCHECKSUM

```
  [LAMBDA (PUP)                                                  (* bvm%: "11-FEB-83 12:28")

    ;; Sets the PUPCHECKSUM field of PUP to checksum over its current contents

    (replace PUPCHECKSUM of PUP with (COND
                                        [\PUP.CHECKSUMFLG (\CHECKSUM (fetch PUPCHECKSUMBASE of PUP)
                                                              (SUB1 (FOLDHI (fetch PUPLENGTH of PUP)
                                                                           BYTESPERWORD]
                                     (T \NULLCHECKSUM)))
    T])
)

(RPAQ? \PUP.CHECKSUMFLG T)

(RPAQ? \MAX.EPKTS.ON.PUPSOCKET 20Q)

(RPAQ? \LOCALPUPNETHOST )

(RPAQ? \OLDPUPHOST# 0)

;; Pup error stuff

(DECLARE%: DONTCOPY

;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

[ACCESSFNS ERRORPUP ((ERRORPUPBASE (fetch PUPCONTENTS of DATUM)))
       (BLOCKRECORD ERRORPUPBASE ((ERRORPUPCOPY 12Q WORD)       ; Copy of pup header
                                  (ERRORPUPCODE WORD)
                                  (ERRORPUPARG WORD)            ; Usually zero
                                  (ERRORPUPSTRINGBASE WORD)     ; Human readable message

                                  ]
)

(RPAQQ PUPERRORCODES
       ((\PUPE.CHECKSUM 1)
        (\PUPE.NOSOCKET 2)
        (\PUPE.SOCKETFULL 3)
        (\PUPE.GATEWAY.BADPUP 1001Q)
        (\PUPE.NOROUTE 1002Q)
        (\PUPE.NOHOST 1003Q)
        (\PUPE.LOOPED 1004Q)
        (\PUPE.TOOLARGE 1005Q)
        (\PUPE.WRONG.GATEWAY 1006Q)
        (\PUPE.GATEWAYFULL 1007Q)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \PUPE.CHECKSUM 1)

(RPAQQ \PUPE.NOSOCKET 2)

(RPAQQ \PUPE.SOCKETFULL 3)

(RPAQQ \PUPE.GATEWAY.BADPUP 1001Q)

(RPAQQ \PUPE.NOROUTE 1002Q)

(RPAQQ \PUPE.NOHOST 1003Q)

(RPAQQ \PUPE.LOOPED 1004Q)

(RPAQQ \PUPE.TOOLARGE 1005Q)

(RPAQQ \PUPE.WRONG.GATEWAY 1006Q)

(RPAQQ \PUPE.GATEWAYFULL 1007Q)

(CONSTANTS (\PUPE.CHECKSUM 1)
       (\PUPE.NOSOCKET 2)
       (\PUPE.SOCKETFULL 3)
       (\PUPE.GATEWAY.BADPUP 1001Q)
       (\PUPE.NOROUTE 1002Q)
       (\PUPE.NOHOST 1003Q)
       (\PUPE.LOOPED 1004Q)
       (\PUPE.TOOLARGE 1005Q)
       (\PUPE.WRONG.GATEWAY 1006Q)
       (\PUPE.GATEWAYFULL 1007Q))
)

;; END EXPORTED DEFINITIONS

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS PUPERRORMESSAGES)
)
)
```

```
(RPAQQ PUPERRORMESSAGES
       ((1 "Bad Checksum")
        (2 "No such socket")
        (3 "Socket full")
        (1001Q "Inconsistent pup")
        (1002Q "No route to that host")
        (1003Q "Host is down")
        (1004Q "Too many hops")
        (1005Q "Pup too long")
        (1006Q "Wrong gateway for that host")
        (1007Q "Gateway IQ full")))

(DEFINEQ

(\PUPERROR
  [LAMBDA (PUP ERRCODE MSG)                                     (* bvm%: " 5-Jan-85 23:33")

;;; Turn packet around into an error packet with given error

    (COND
       (\PEEKPUPNUMBER (\MAYBEPEEKPUP PUP))
       ((AND (NEQ (fetch PUPDESTHOST of PUP)
                  0)
             (NEQ (fetch PUPTYPE of PUP)
                  \PT.ERROR))                                   ; Don't respond to errors or to broadcasts!
        [COND
           ((AND PUPTRACEFLG (NEQ PUPTRACEFLG 'PEEK))
            (printout PUPTRACEFILE "Incoming packet dropped because: " (OR (CADR (ASSOC ERRCODE PUPERRORMESSAGES
                                                                                        ))
                                                                           ERRCODE)
                   T)
            (OR (EQ PUPTRACEFLG 'RAW)
                (PRINTPUP PUP]
        (\BLT (fetch PUPCONTENTS of PUP)
              (fetch PUPBASE of PUP)
              (FOLDLO \PUPHEADERLEN BYTESPERWORD))              ; Copy pup header into body
        (replace ERRORPUPCODE of PUP with ERRCODE)
        (replace ERRORPUPARG of PUP with 0)
        [replace PUPLENGTH of PUP with (IPLUS \PUPOVLEN \PUPHEADERLEN (ITIMES 2 BYTESPERWORD)
                                              (\PUTBASESTRING (LOCF (fetch ERRORPUPSTRINGBASE of PUP))
                                                     0
                                                     (OR MSG (CADR (ASSOC ERRCODE PUPERRORMESSAGES))
                                                         ""]
        (replace PUPTYPE of PUP with \PT.ERROR)
        (SWAPPUPPORTS PUP)
        (replace EPREQUEUE of PUP with 'FREE)
        (SENDPUP NIL PUP))
       (T (\RELEASE.ETHERPACKET PUP])
)

;; Pup utilities

(DEFINEQ

(SETUPPUP
  [LAMBDA (PUP DESTHOST DESTSOCKET TYPE ID SOC REQUEUE)         (* bvm%: " 5-Jan-85 23:34")

    ;; Initialize pup header PUP with indicated destination HOST, DESTSOCKET and TYPE.  A local socket and ID (if not supplied) are assigned.
    ;; Returns a 'socket' datum

    (OR \PUP.READY (ASSURE.PUP.READY))
    (replace PUPLENGTH of (SETQ PUP (\DTEST PUP 'ETHERPACKET)) with \PUPOVLEN)
                                                                ; pup data initially empty
    (replace (PUP TYPEWORD) of PUP with TYPE)                   ; Clears PUPTCONTROL
    [replace PUPID of PUP with (OR ID (SETQ \PUPCOUNTER (COND
                                                           ((IGEQ \PUPCOUNTER 177777Q)
                                                            1)
                                                           (T (ADD1 \PUPCOUNTER]
    (replace PUPDEST of PUP with (OR (FIXP (SETQ DESTHOST (ETHERPORT DESTHOST T)))
                                     (CAR DESTHOST)))
    (replace PUPDESTSOCKET of PUP with (COND
                                          ((AND (LISTP DESTHOST)
                                                (NEQ (CDR DESTHOST)
                                                     0))
                                           (CDR DESTHOST))
                                          (T DESTSOCKET)))
    (AND REQUEUE (replace EPREQUEUE of PUP with REQUEUE))
    (OR SOC (SETQ SOC (OPENPUPSOCKET])

(SWAPPUPPORTS
  [LAMBDA (PUP)                                                 (* bvm%: "12-FEB-83 16:21")
    (swap (fetch PUPSOURCE of PUP)
          (fetch DEST of PUP))
    (swap (fetch PUPSOURCESOCKETHI of PUP)
          (fetch DESTSKTHI of PUP))
    (swap (fetch PUPSOURCESOCKETLO of PUP)
          (fetch DESTSKTLO of PUP])

(GETPUP
  [LAMBDA (PUPSOC WAIT)                                             (* bvm%: "24-MAY-83 17:42")
    (SETQ PUPSOC (\DTEST PUPSOC 'PUPSOCKET))
    (PROG (PUP TIMER)
      LP  (UNINTERRUPTABLY
              (COND
                 ((SETQ PUP (\DEQUEUE (ffetch (PUPSOCKET INQUEUE) of PUPSOC)))
                  (add (ffetch (PUPSOCKET INQUEUELENGTH) of PUPSOC)
                       -1))))
          (COND
             [(NULL PUP)
              (COND
                 (WAIT (COND
                          ((EQ WAIT T))
                          [TIMER (COND
                                    ((TIMEREXPIRED? TIMER)
                                     (RETURN]
                          (T (OR (FIXP WAIT)
                                 (LISPERROR "NON-NUMERIC ARG" WAIT))
                             (SETQ TIMER (SETUPTIMER WAIT))
                             T))
                       (AWAIT.EVENT (ffetch PUPSOCEVENT of PUPSOC)
                              TIMER T)
                       (GO LP))
                 (T                                                 ; Let ether procs run
                    (BLOCK]
             [(EQ \EPT.PUP (fetch EPTYPE of PUP))
              (AND PUPTRACEFLG (\MAYBEPRINTPACKET PUP 'GET]
             (T (AND PUPTRACEFLG (printout PUPTRACEFILE T "Non-PUP packet " PUP " arrived on " PUPSOC T))
                (SETQ PUP)))
          (RETURN PUP])

(SENDPUP
  [LAMBDA (PUPSOC PUP)                                              (* bvm%: " 5-Jan-85 23:34")

    ;; Returns the PUP arg iff packet can be sent;  returns a litatom explaining error otherwise.

    (SETQ PUP (\DTEST PUP 'ETHERPACKET))
    [AND PUPSOC (SETQ PUPSOC (\DTEST PUPSOC 'PUPSOCKET]
    (replace EPTYPE of PUP with \EPT.PUP)
    (replace PUPTCONTROL of PUP with 0)
    (PROG (NDB)
          (\RCLK (LOCF (fetch EPTIMESTAMP of PUP)))
          [COND
             ((AND PUPSOC (EQ (fetch PUPSOURCESOCKETLO of PUP)
                              0)
                   (EQ (fetch PUPSOURCESOCKETHI of PUP)
                       0))
              (replace PUPSOURCESOCKETHI of PUP with (fetch PSOCKETHI of PUPSOC))
              (replace PUPSOURCESOCKETLO of PUP with (fetch PSOCKETLO of PUPSOC]
          (RETURN (COND
                     ((NEQ (OR \PUP.READY (ASSURE.PUP.READY))
                           T)                                       ; No PUP?
                      (\REQUEUE.ETHERPACKET PUP)
                      'NoEther)
                     ((fetch EPTRANSMITTING of PUP)
                      (AND PUPTRACEFLG (printout PUPTRACEFILE "[Packet not sent--already being transmitted]" T))
                      'AlreadyQueued)
                     ((NULL (SETQ NDB (\ROUTE.PUP PUP)))
                      (AND PUPTRACEFLG (PRINTPUPROUTE PUP "[Put fails: no routing]" PUPTRACEFILE))
                      (\REQUEUE.ETHERPACKET PUP)
                      'NoRouting)
                     (T (\SETPUPCHECKSUM PUP)
                        (AND PUPTRACEFLG (\MAYBEPRINTPACKET PUP 'PUT))
                        (TRANSMIT.ETHERPACKET NDB PUP)
                        (BLOCK)
                        NIL])

(EXCHANGEPUPS
  [LAMBDA (SOC OUTPUP DUMMY IDFILTER TIMEOUT)                       (* bvm%: "24-MAY-83 23:19")

    ;; Sends out OUTPUP on SOC and waits for a reply, which it puts in INPUP.  If IDFILTER is true, only replies with the same ID are accepted.
    ;; Returns input pup on success, or NIL on failure.  TIMEOUT overrides the default timeout.

    (DISCARDPUPS (\DTEST SOC 'PUPSOCKET))                          ; Flush any pups waiting on this socket
    (SENDPUP SOC OUTPUP)
    (bind INPUT (TIMER _ (SETUPTIMER (OR TIMEOUT \ETHERTIMEOUT)))
          (EVENT _ (ffetch PUPSOCEVENT of SOC)) do (COND
                                                     ([AND (SETQ INPUP (GETPUP SOC))
                                                           (OR (NOT IDFILTER)
                                                               (IEQP (fetch PUPID of INPUP)
                                                                     (fetch PUPID of OUTPUP]
                                                      (RETURN INPUP))
                                                     (T (AWAIT.EVENT EVENT TIMER T)))
```

```
       repeatuntil (TIMEREXPIRED? TIMER])

(DISCARDPUPS
  [LAMBDA (SOC)                                                   (* bvm%: " 5-MAY-83 23:51")
    (SETQ SOC (\DTEST SOC 'PUPSOCKET))
    (UNINTERRUPTABLY
        (PROG1 (fetch (PUPSOCKET INQUEUELENGTH) of SOC)
            (\FLUSH.PACKET.QUEUE (fetch (PUPSOCKET INQUEUE) of SOC))
            (replace (PUPSOCKET INQUEUELENGTH) of SOC with 0)))])

(GETPUPWORD
  [LAMBDA (PUP WORD#)                                             (* bvm%: "31-JAN-83 15:27")
    (\GETBASE [fetch PUPCONTENTS of (SETQ PUP (\DTEST PUP 'ETHERPACKET]
           WORD#])

(\PUPINIT
  [LAMBDA NIL                                                     ; Edited 13-Feb-89 15:25 by snow
    (for FN in '(SETUPPUP EXCHANGEPUPS GETPUP SENDPUP CLEARPUP GETPUPSTRING PUTPUPSTRING ALLOCATE.PUP
                      RELEASE.PUP CREATESOCKET FLUSHSOCKET)
       bind NEWFN unless (GETD (SETQ NEWFN (PACK* '\ FN))) do     ; make dummy defs for old \ fns
                                                              (PUTD NEWFN (GETD FN)
                                                                    T))
    (OR (EQ \MACHINETYPE \MAIKO)
        (INITPUPLEVEL1 T])

)

(DEFINEQ

(ETHERHOSTNAME
  [LAMBDA (PORT USE.OCTAL.DEFAULT)                                (* bvm%: "25-Apr-86 12:40")

;;; Looks up the name of the host at address PORT.  PORT may be a numeric address, or (host . socket) as returned by ETHERPORT

    (PROG ((SOC (\GETMISCSOCKET))
           (SOCKET# 0)
           (OPUP (ALLOCATE.PUP))
           NETHOST RESULT BUF IPUP)
          (OR (EQ (OR \PUP.READY (ASSURE.PUP.READY))
                  T)
              (RETURN))
          [SETQ NETHOST (COND
                            ((NULL PORT)
                             (\LOCALPUPHOSTNUMBER))
                            ((FIXP PORT))
                            [(AND (LISTP PORT)
                                  (FIXP (SETQ SOCKET# (CDR PORT)))
                                  (FIXP (CAR PORT]
                            ((AND (NLISTP PORT)
                                  (SETQ NETHOST (\PARSE.PORTCONSTANT PORT)))
                             (SETQ SOCKET# (CDR NETHOST))
                             (CAR NETHOST))
                            (T (LISPERROR "ILLEGAL ARG" PORT]
          [COND
             ((EQ (fetch PUPNET# of NETHOST)
                  0)                                              ; Net not specified, default to local net
              (SETQ NETHOST (create PUPADDRESS
                                    PUPNET# _ (\LOCALPUPNETNUMBER)
                                    PUPHOST# _ NETHOST]
          (SETUPPUP OPUP 0 \PUPSOCKET.MISCSERVICES \PT.ADDRLOOKUP NIL SOC T)
          (add (fetch PUPLENGTH of OPUP)
               6)                                                 ; port is 6 bytes long
          (replace (PORT NETHOST) of (SETQ BUF (fetch PUPCONTENTS of OPUP)) with NETHOST)
          (replace (PORT SOCKET) of BUF with SOCKET#)
          [to \MAXETHERTRIES when (SETQ IPUP (EXCHANGEPUPS SOC OPUP NIL T))
             do (SELECTC (fetch PUPTYPE of IPUP)
                     (\PT.ADDRRESPONSE
                          (SETQ RESULT (GETPUPSTRING IPUP))
                          (COND
                             ([for C instring RESULT always (AND (ILESSP C (CHARCODE 10Q))
                                                                 (IGEQ C (CHARCODE 0]

                              ;; Not really a name, but a Dandelion processor ID.  Pretend is NIL so as not to confuse rest of world with
                              ;; uninvertable name

                              (SETQ RESULT NIL)))
                          (RETURN))
                     (\PT.NAME/ADDRERROR
                          (COND
                             (PUPTRACEFLG (printout PUPTRACEFILE "Address lookup error for " (PORTSTRING NETHOST
                                                                                                     SOCKET#)
                                                 ": "
                                                 (GETPUPSTRING IPUP)
                                                 T)))
                          (RETURN))
```

```
                    NIL)
             finally (COND
                      (PUPTRACEFLG (printout PUPTRACEFILE "Address lookup timed out" T]
          (AND IPUP (RELEASE.PUP IPUP))
          (RELEASE.PUP OPUP)
          (RETURN (OR RESULT (AND USE.OCTAL.DEFAULT (PORTSTRING NETHOST (AND (NEQ SOCKET# 0)
                                                                             SOCKET#])

(ETHERHOSTNUMBER
  [LAMBDA (NAME)                                                ; Edited 31-Mar-87 18:26 by bvm:
    (OR \PUP.READY (ASSURE.PUP.READY))
    (COND
       ((NULL NAME)
        (\LOCALPUPADDRESS))
       (T (CAR (BESTPUPADDRESS NAME])

(ETHERPORT
  [LAMBDA (NAME ERRORFLG MULTFLG)                               (* bvm%: "16-NOV-83 11:40")

;;; Returns net address of NAME as (nethost . socket), or list of same if MULTFLG is true . Caches results locally so doesn't have to look all the time. If
;;; ERRORFLG is true, generates error on failure.

;;; If MULTFLG is nonNIL, returns a list of results --- singleton unless perhaps from \LOOKUPPORT

    (PROG (VAL)
          (RETURN (COND
                     ([SETQ VAL (COND
                                   ((FIXP NAME)                 ; A host number. Give it socket zero
                                    (\FIXLOCALNET (CONS NAME 0)))
                                   [(LISTP NAME)                ; An existing port structure
                                    (COND
                                       ((AND (FIXP (CAR NAME))
                                             (FIXP (CDR NAME)))
                                        (\FIXLOCALNET NAME))
                                       (ERRORFLG (\ILLEGAL.ARG NAME))
                                       (T (RETURN]
                                   (T (\PARSE.PORTCONSTANT NAME]
                      (COND
                         (MULTFLG (LIST VAL))
                         (T VAL)))
                     [(SETQ VAL (OR (GETHASH NAME \ETHERPORTS)
                                    (PUTHASH NAME (\LOOKUPPORT NAME)
                                           \ETHERPORTS)))       ; note we always save multiple values in case they are ever
                                                                ; wanted
                      (COND
                         (MULTFLG VAL)
                         (T (CAR VAL]
                     (ERRORFLG (ERROR "host not found" NAME])

(BESTPUPADDRESS
  [LAMBDA (HOST ERRORSTREAM)                                    (* bvm%: " 5-Jan-85 23:36")
    ;; Returns a pup port for HOST, selecting the one of possibly multiple ports that is closest, returning NIL if there is no route or name lookup fails. If
    ;; ERRORSTREAM = ERROR, causes error on failure; otherwise ERRORSTREAM is a stream to print an appropriate error message to before
    ;; returning NIL
    (PROG (PORT NET MSG)
          (OR (EQ (OR \PUP.READY (ASSURE.PUP.READY))
                  T)
              (RETURN))
      RETRY
          (COND
             [[SETQ PORT (COND
                            ((FIXP HOST)                        ; A host number. Give it socket zero
                             (\FIXLOCALNET (CONS HOST 0)))
                            [(LISTP HOST)                       ; An existing port structure
                             (COND
                                ((AND (FIXP (CAR HOST))
                                      (FIXP (CDR HOST)))
                                 (\FIXLOCALNET HOST))
                                (ERRORSTREAM (SETQ MSG "Invalid port specification")
                                       (GO ERROR))
                                (T (RETURN]
                            (T (\PARSE.PORTCONSTANT HOST]
              (COND
                 ((OR (EQ (SETQ NET (fetch PUPNET# of (CAR PORT)))
                          0)
                      (EQ NET (\LOCALPUPNETNUMBER)))
                  (RETURN PORT))
                 (T (SETQ PORT (LIST PORT]
             ((SETQ PORT (OR (GETHASH HOST \ETHERPORTS)
                             (PUTHASH HOST (\LOOKUPPORT HOST)
                                    \ETHERPORTS)))              ; note we always save multiple values in case they are ever
                                                                ; wanted
                 )
                (ERRORSTREAM (SETQ MSG "Host not found")
                        (GO ERROR))
                (T (RETURN)))
            [RETURN (for TRY from 1 to 5 bind NOTLOOKEDUP HOPS BESTHOPS BESTPORT ROUTE
                         do (SETQ BESTHOPS \RT.INFINITY)
                             (SETQ NOTLOOKEDUP (SETQ BESTPORT NIL))
                             [for PAIR in PORT do (COND
                                                         ((OR [NOT (SETQ ROUTE (\LOCATE.PUPNET (fetch PUPNET#
                                                                                                           of (CAR PAIR]
                                                               (IGEQ (SETQ HOPS (fetch RTHOPCOUNT of ROUTE))
                                                                       \RT.INFINITY))
                                                          (SETQ NOTLOOKEDUP T))
                                                         ((ILESSP HOPS BESTHOPS)
                                                          (SETQ BESTHOPS HOPS)
                                                          (SETQ BESTPORT PAIR]
                                                                              ; Enter request for routing for all hosts
                             (COND
                                ((AND BESTPORT (OR (NOT NOTLOOKEDUP)
                                                        (ILEQ BESTHOPS \PUP.ROUTING.TABLE.RADIUS)
                                                        (IGREATERP TRY 1)))
                                  (RETURN BESTPORT)))
                             (BLOCK \ETHERTIMEOUT)
                         finally (COND
                                    (ERRORSTREAM (SETQ MSG "No route to host")
                                            (GO ERROR]
       ERROR
            (COND
                ((EQ ERRORSTREAM 'ERROR)
                 (ERROR MSG HOST)
                 (GO RETRY))
                (T (printout ERRORSTREAM T MSG ": " HOST)
                    (RETURN])
```

```
(NETDAYTIME0
  [LAMBDA NIL                                                                   (* bvm%: "26-Jul-84 15:26")

;;; Returns a 32-bit unsigned alto time from the network, if possible

     (\NET.SETTIME T])
```

```
(\PUP.SETTIME
  [LAMBDA (RETFLG)                                                              ; Edited 13-May-88 15:22 by MASINTER
     (CL:UNLESS (AND RETFLG (NOT (STRINGP RETFLG)))
          (SETQ \TimeZoneComp (SUBRCALL GETUNIXTIME 10Q NIL)))
     (\PROCESS.RESET.TIMERS)
     (DAYTIME])
```

```
(\SETNEWTIME0
  [LAMBDA (NEWTIME)                                                             (* bvm%: "26-Jul-84 15:23")
     (PROG [(OLDTIME (\DAYTIME0 (create FIXP]
            (\SETDAYTIME0 NEWTIME)
            (COND
                ((IGREATERP (IABS (IDIFFERENCE NEWTIME OLDTIME))
                         454Q)                                                  ; Time changed by more than 5 minutes, maybe mention it
                 (printout PROMPTWINDOW T "[Time reset to " (DATE (DATEFORMAT TIME.ZONE))
                         "]"])
```

```
(\NET.SETTIME
  [LAMBDA (RETFLG)                                                              (* bvm%: "26-Jul-84 15:25")

;;; Sets the time from local network time server, or just returns said time if RETFLG is true

     (if \LOCALNDBS
          then (SELECTQ (fetch (NDB NETTYPE) of \LOCALNDBS)
                     (3 (OR (\PUP.SETTIME RETFLG)
                             (\NS.SETTIME RETFLG)))
                     (12Q (OR (\NS.SETTIME RETFLG)
                               (AND \PUP.READY (\PUP.SETTIME RETFLG))))
                    NIL])
```

```
(NETDATE
  [LAMBDA NIL                                                                   (* bvm%: "25-Apr-86 12:46")
     (GDATE (ALTO.TO.LISP.DATE (OR (NETDAYTIME0)
                                         (\DAYTIME0 (create FIXP])
```

```
(\LOOKUPPORT
  [LAMBDA (NAME)                                                                ; Edited  1-Apr-87 12:37 by bvm:

;;; Looks up the ether address of NAME, returning a list of dotted pairs (nethost  . socket), or NIL on failure
```

```
    (AND NAME (PROG ((SOC (\GETMISCSOCKET))
                      (OPUP (ALLOCATE.PUP))
                      RESULT BUF LEN IPUP)
                     (SETUPPUP OPUP 0 \PUPSOCKET.MISCSERVICES \PT.NAMELOOKUP NIL SOC T)
                     (PUTPUPSTRING OPUP NAME)
                     [to \MAXETHERTRIES when (SETQ IPUP (EXCHANGEPUPS SOC OPUP NIL T))
                        do (SELECTC (fetch PUPTYPE of IPUP)
                                (\PT.NAMERESPONSE
                                    [COND
                                       ((> (SETQ LEN (IQUOTIENT (FOLDLO (- (fetch PUPLENGTH of IPUP)
                                                                            \PUPOVLEN)
                                                                        BYTESPERWORD)
                                                             \PORTIDLEN))
                                           1)
                                        (COND
                                           (PUPTRACEFLG (printout PUPTRACEFILE "Multiple response received for "
                                                               NAME T]
                                    [RETURN (SETQ RESULT (from 1 to LEN as (PTR _ (fetch PUPCONTENTS of IPUP))
                                                             by (\ADDBASE PTR \PORTIDLEN)
                                                             collect (CONS (fetch (PORT NETHOST) of PTR)
                                                                           (fetch (PORT SOCKET) of PTR])
                                (\PT.NAME/ADDRERROR
                                    (COND
                                       (PUPTRACEFLG (printout PUPTRACEFILE "Name lookup error for " NAME ": "
                                                            (GETPUPSTRING IPUP)
                                                            T)))
                                    (RETURN))
                               NIL)
                        finally (COND
                                   (PUPTRACEFLG (printout PUPTRACEFILE "Name lookup timed out" T]
                     (AND IPUP (RELEASE.PUP IPUP))
                     (RELEASE.PUP OPUP)
                     (RETURN RESULT])

(\PARSE.PORTCONSTANT
  [LAMBDA (STR)                                                   (* bvm%: "16-NOV-83 12:01")

;;; If STR is a constant ether address of form net#host#socket, returns a port, else NIL

    (for CH instring (OR (STRINGP STR)
                         (SETQ STR (MKSTRING STR)))
       bind NET HOST VAL do (COND
                                [(AND (IGEQ CH (CHARCODE 0))
                                      (ILEQ CH (CHARCODE 7)))    ; Add octal digit into value
                                 (SETQ VAL (IPLUS (COND
                                                     (VAL (LLSH VAL 3))
                                                     (T 0))
                                                  (IDIFFERENCE CH (CHARCODE 0]
                                ((EQ CH (CHARCODE %#))           ; # terminates net or host number
                                 (COND
                                    (NET (RETURN)))
                                 (SETQ NET HOST)
                                 (SETQ HOST (OR VAL 0))
                                 (SETQ VAL NIL))
                                (T (RETURN)))
       finally                                                    ; Ran out of chars.  Save last value parsed, make sure we have
                                                                  ; at least a net and host
             (RETURN (AND (OR HOST VAL)
                          (CONS (LOGOR (OR HOST 0)
                                       (COND
                                          (NET (LLSH NET 10Q))
                                          (T 0)))
                                (OR VAL 0])

(\FIXLOCALNET
  [LAMBDA (PORT)                                                  (* bvm%: " 5-Jan-85 23:37")

    ;; Port is a dotted pair (nethost  . socket).  We force the nethost to have a nonzero net if we know our net by now.  Returns the possibly modified
    ;; PORT

    [PROG (NET)
          (COND
             ((AND (ILESSP (CAR PORT)
                           400Q)
                   (NEQ (CAR PORT)
                        0)
                   \LOCALNDBS
                   (SETQ NET (fetch NDBPUPNET# of \LOCALNDBS))
                   (NEQ NET 0))
              (RPLACA PORT (create PUPADDRESS
                                   PUPNET# _ NET
                                   PUPHOST# _ (CAR PORT]
    PORT])

)
```

```
(DEFINEQ

(PORTSTRING
  [LAMBDA (NETHOST SOCKET)                                         (* bvm%: " 5-Jan-85 23:40")
    [COND
       ((LISTP NETHOST)
        (SETQ SOCKET (CDR NETHOST))
        (COND
           ((EQ SOCKET 0)
            (SETQ SOCKET NIL)))
        (SETQ NETHOST (CAR NETHOST]
    (CONCAT (OCTALSTRING (LRSH NETHOST 10Q))
           '%#
           (OCTALSTRING (LOGAND NETHOST 377Q))
           '%#
           (COND
              (SOCKET (OCTALSTRING SOCKET))
              (T ""])

(OCTALSTRING
  [LAMBDA (N)                                                      (* bvm%: "21-JUL-81 12:16")
    (GLOBALRESOURCE (\NUMSTR \NUMSTR1)
           (CONCAT (\CONVERTNUMBER N 10Q NIL NIL \NUMSTR \NUMSTR1])
)

(RPAQ? \ETHERPORTS (HASHARRAY 24Q))

(RPAQ? \ETHERTIMEOUT 3720Q)

(RPAQ? \MAXETHERTRIES 4)

(RPAQ? \PUPCOUNTER 0)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \ETHERPORTS \PUPCOUNTER)
)
```

;; Accessing a PUP's contents

```
(DEFINEQ

(CLEARPUP
  [LAMBDA (PUP)                                                    (* bvm%: "31-JAN-83 15:31")
    (replace EPLINK of (SETQ PUP (\DTEST PUP 'ETHERPACKET)) with NIL)
                                                                   ; Clear the pointer fields
    [replace EPUSERFIELD of PUP with (replace EPPLIST of PUP with (replace EPREQUEUE of PUP
                                                                   with (replace EPSOCKET of PUP
                                                                          with (replace EPNETWORK of PUP
                                                                                 with NIL]
    (\ZEROWORDS (fetch PUPBASE of PUP)
           (\ADDBASE (LOCF (fetch SOURCESKT of PUP))
                  1))
    (\ZEROBYTES (fetch PUPCONTENTS of PUP)
           0
           (SUB1 \MAX.PUPLENGTH])

(PUTPUPWORD
  [LAMBDA (PUP WORD# VALUE)                                        (* bvm%: "31-JAN-83 15:31")
    (\PUTBASE [fetch PUPCONTENTS of (SETQ PUP (\DTEST PUP 'ETHERPACKET]
           WORD# VALUE])

(GETPUPBYTE
  [LAMBDA (PUP BYTE#)                                              (* bvm%: "31-JAN-83 15:31")
    (\GETBASEBYTE [fetch PUPCONTENTS of (SETQ PUP (\DTEST PUP 'ETHERPACKET]
           BYTE#])

(PUTPUPBYTE
  [LAMBDA (PUP BYTE# VALUE)                                        (* bvm%: "31-JAN-83 15:31")
    (\PUTBASEBYTE [fetch PUPCONTENTS of (SETQ PUP (\DTEST PUP 'ETHERPACKET]
           BYTE# VALUE])

(GETPUPSTRING
  [LAMBDA (PUP OFFSET)                                             (* bvm%: "26-Apr-84 10:04")
    (PROG [(NC (IDIFFERENCE (IDIFFERENCE [ffetch PUPLENGTH of (SETQ PUP (\DTEST PUP 'ETHERPACKET]
                                    \PUPOVLEN)
                      (OR OFFSET (SETQ OFFSET 0]
          (RETURN (COND
                     ((IGREATERP NC 0)
                      (\GETBASESTRING (ffetch PUPCONTENTS of PUP)
                             OFFSET NC))
                     (T                                          ; Could give error if length negative, but the empty string is a
                                                                 ; reasonable thing to return
                        (ALLOCSTRING 0])

(GETPUPSTREAM
  [LAMBDA (PUP OFFSET LENGTH ACCESS WRITEXTENSIONFN)             (* bvm%: "26-OCT-83 12:10")
    (\MAKEBASEBYTESTREAM [fetch PUPCONTENTS of (SETQ PUP (\DTEST PUP 'ETHERPACKET]
           (OR OFFSET (SETQ OFFSET 0))
           (OR LENGTH (IDIFFERENCE (IDIFFERENCE (fetch PUPLENGTH of PUP)
                                          \PUPOVLEN)
                             OFFSET))
           (OR ACCESS 'INPUT)
           WRITEXTENSIONFN])

(PUTPUPSTRING
  [LAMBDA (PUP STR)                                              (* bvm%: "31-JAN-83 15:35")
    (add [fetch PUPLENGTH of (SETQ PUP (\DTEST PUP 'ETHERPACKET]
         (\PUTBASESTRING (fetch PUPCONTENTS of PUP)
                (IDIFFERENCE (fetch PUPLENGTH of PUP)
                       \PUPOVLEN)
                STR])
)

(DEFOPTIMIZER GETPUPWORD (PUPARG WORD#)
                         `(\GETBASE (fetch PUPCONTENTS of (\DTEST ,PUPARG 'ETHERPACKET))
                                 ,WORD#))

(DEFOPTIMIZER PUTPUPWORD (PUPARG WORD# VALUE)
                         `(\PUTBASE (fetch PUPCONTENTS of (\DTEST ,PUPARG 'ETHERPACKET))
                                 ,WORD#
                                 ,VALUE))

(DEFOPTIMIZER GETPUPBYTE (PUPARG BYTE#)
                         `(\GETBASEBYTE (fetch PUPCONTENTS of (\DTEST ,PUPARG 'ETHERPACKET))
                                 ,BYTE#))

(DEFOPTIMIZER PUTPUPBYTE (PUPARG BYTE# VALUE)
                         `(\PUTBASEBYTE (fetch PUPCONTENTS of (\DTEST ,PUPARG 'ETHERPACKET))
                                 ,BYTE#
                                 ,VALUE))

;; Reading property lists from streams

(DEFINEQ

(READPLIST
  [LAMBDA (STREAM NOERRORFLG)                                    (* bvm%: " 6-Oct-86 14:14")

;;; Reads an FTP-style property list from STREAM.  If the plist is malformed, causes an error unless NOERRORFLG is true.  FTP-style plists look like
;;; lists of two elements in a very rigid syntax: each element of the list is (property value);  spaces are significant except the one immediately following
;;; property.  READPLIST returns the property names as uppercase atoms, the values as strings

    (PROG ([READTABLES (OR (LISTP \READPLIST.READTABLES)
                           (SETQ \READPLIST.READTABLES (LET ((TAB1 (COPYREADTABLE 'ORIG))
                                                             TAB2)
                                                            (SETSEPR NIL NIL TAB1)

                                      ;; Want to set up two readtables to read properties.  Both read tables use ' as escape character.  The
                                      ;; first read table reads the property; it terminates on space and is case-insensitive.  The second read
                                      ;; table reads the value; it terminates on right paren.

                                                            (SETSYNTAX '%' 'ESCAPE TAB1)
                                                            (SETSYNTAX '%% 'OTHER TAB1)
                                                            (SETQ TAB2 (COPYREADTABLE TAB1))
                                                            (SETBRK (CHARCODE (")"))
                                                                   NIL TAB2)
                                                            (SETBRK (CHARCODE (SPACE))
                                                                   NIL TAB1)
                                                            (READTABLEPROP TAB1 'CASEINSENSITIVE T)
                                                            (CONS TAB1 TAB2]
           PLIST)
          (OR (EQ (BIN STREAM)
                  (CHARCODE "("))
              (GO ERROR))
          [RETURN (bind CH while (EQ (SETQ CH (BIN STREAM))
                                     (CHARCODE "("))
                     collect                                     ; Another element
                           (PROG1 [LIST (RATOM STREAM (CAR READTABLES))
                                        (PROGN (BIN STREAM)      ; Skip over the space
                                               (RSTRING STREAM (CDR READTABLES]
                               (COND
                                   ((NEQ (BIN STREAM)
                                         (CHARCODE ")"))
                                    (GO ERROR))))
                     finally (COND
                              ((NEQ CH (CHARCODE ")"))
                               (GO ERROR]
      ERROR
          (OR NOERRORFLG (ERROR "Malformed property list in stream" STREAM))
          (RETURN NIL])

)

(RPAQ? \READPLIST.READTABLES NIL)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \READPLIST.READTABLES)
)

(DEFINEQ

(\CANONICAL.HOSTNAME
  [LAMBDA (NAME)                                                 ; Edited 11-Mar-88 12:09 by bvm

;;; Returns the canonical name of a given hostname, in case a server has synonyms

    (if (NUMBERP NAME)
        then (AND (SMALLP NAME)
                  (< NAME 377Q)
                  NAME)
      else (if (NOT (LITATOM NAME))
              then (SETQ NAME (MKATOM NAME)))
          (OR (CDR (FASSOC NAME \HOSTNAMES))
              (\CANONICALIZE.PUP.HOSTNAME NAME])

(\CANONICALIZE.PUP.HOSTNAME
  [LAMBDA (NAME)                                                 ; Edited 11-Mar-88 12:09 by bvm
    (DECLARE (GLOBALVARS FIXSPELLREL))
    (LET ((PORT (ETHERPORT NAME))
          OFFICIALNAME)
         (COND
            (PORT (if [NOT (LITATOM (SETQ OFFICIALNAME (MKATOM (U-CASE (OR (ETHERHOSTNAME PORT)
                                                                           NAME]
                      then                                       ; DLions with no real name come out as large integers, not
                                                                 ; litatoms, so use name given
                           (SETQ OFFICIALNAME (U-CASE NAME)))
                  (push \HOSTNAMES (CONS NAME OFFICIALNAME))     ; If no name in database, take what was given
                  OFFICIALNAME)
            (\HOSTNAMES (FIXSPELL NAME FIXSPELLREL \HOSTNAMES T])

)

;; Default this for when IP not loaded

(MOVD? 'NILL '\CANONICALIZE.IP.HOSTNAME NIL T)

(ADDTOVAR \HOSTNAMES )

(ADDTOVAR \SYSTEMCACHEVARS \HOSTNAMES)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \HOSTNAMES)
)

;; PUP allocation
;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

(PUTPROPS BINDPUPS MACRO [X (CONS (LIST 'LAMBDA (CAR X)
                                           (CONS 'PROGN (CDR X)))
                                    (in (CAR X) collect (LIST 'ALLOCATE.PUP])
)

(PUTPROPS BINDPUPS INFO BINDS)

(ADDTOVAR PRETTYPRINTMACROS
          (BINDPUPS
LAMBDA
  (FORM)
  (PROG [(POS (IPLUS 2 (POSITION]
        (PRIN1 "(")
        (PRIN2 (CAR FORM))
        (SPACES 1)
        (PRINTDEF (CADR FORM)
               (POSITION))
        (OR [EQ COMMENTFLG (CAAR (SETQ FORM (CDDR FORM]
            (TAB POS 0))
        (PRINTDEF FORM POS T T FNSLST)
        (PRIN1 ")"))))
```

;; END EXPORTED DEFINITIONS

;; Pup routing

```
(DEFINEQ

(\PUPGATELISTENER
  [LAMBDA NIL                                                       ; Edited 15-Jan-88 03:00 by bvm
    (PROG ((SOCKET (OPENPUPSOCKET \PUPSOCKET.ROUTING T))
           (TIMER (SETUPTIMER 0))
           PUP EVENT BASE)
          (PROCESSPROP (THIS.PROCESS)
                 'INFOHOOK
                 (FUNCTION \ROUTINGTABLE.INFOHOOK))                 ; For info, print our routing table
          (PROCESSPROP (THIS.PROCESS)
                 :PROTOCOL
                 'PUP)
          (SETQ EVENT (fetch PUPSOCEVENT of SOCKET))
      LP  (COND
             ((SETQ PUP (GETPUP SOCKET))
              (\HANDLE.PUP.ROUTING.INFO PUP)
              (BLOCK))
             ((EQ (AWAIT.EVENT EVENT (COND
                                        ((> \PUPROUTER.PROBECOUNT 0)
                                         \PUPROUTER.PROBETIMER)
                                        (T TIMER))
                         T)
                  EVENT)                                            ; Waiting for pup to arrive or timer to expire--pup arrived.
              (GO LP)))
          (COND
             ((TIMEREXPIRED? TIMER)
              (\AGE.ROUTING.TABLE \PUP.ROUTING.TABLE)
              (SETUPTIMER \RT.AGEINTERVAL TIMER)))
          [COND
             ((AND (> \PUPROUTER.PROBECOUNT 0)
                   (TIMEREXPIRED? \PUPROUTER.PROBETIMER))           ; Routing info desired.  Broadcast a routing request on each
                                                                    ; directly-connected net
              (SETUPPUP (SETQ PUP (ALLOCATE.PUP))
                     0 \PUPSOCKET.ROUTING \PT.GATEWAYREQUEST NIL SOCKET)
              (SENDPUP SOCKET PUP)
              (SETUPTIMER \PUPROUTER.PROBEINTERVAL \PUPROUTER.PROBETIMER)
              (SETQ \PUPROUTER.PROBECOUNT (SUB1 \PUPROUTER.PROBECOUNT]
          (GO LP])

(\HANDLE.PUP.ROUTING.INFO
  [LAMBDA (PUP)                                                     ; Edited 15-Jan-88 01:15 by bvm
                                                                    ; Processes a routing info PUP

    [COND
       ((EQ (fetch PUPTYPE of PUP)
            \PT.GATEWAYRESPONSE)                                    ; Unless we're a gateway, we only handle responses
        (PROG ((HOST (fetch PUPSOURCEHOST of PUP))
               (NDB (fetch EPNETWORK of PUP))
               (LENGTH (FOLDLO (IDIFFERENCE (fetch PUPLENGTH of PUP)
                                       \PUPOVLEN)
                              BYTESPERWORD))
               (BASE (fetch PUPCONTENTS of PUP))
               (TABLE \PUP.ROUTING.TABLE)
               (MASK \ROUTING.TABLE.MASK)
               (RADIUS \PUP.ROUTING.TABLE.RADIUS)
               ENTRY NET HOPS OLDHOPS BUCKET NEWTIMER)
              [COND
                 ((NEQ (fetch NETTYPE of NDB)
                       3)                                           ; For PUP on 10mb net, get translated address
                  (OR (SETQ HOST (\TRANSLATE.3TO10 HOST NDB))
                      (RETURN]
              (SETQ \PUPROUTER.PROBECOUNT 0)                        ; We info from somewhere, so can stop probing
              (while (>= LENGTH \PUP.ROUTINGINFO.WORDS)
                 do (SETQ HOPS (ADD1 (fetch (PUPROUTINGINFO %#HOPS) of BASE)))
                    (SETQ NET (fetch (PUPROUTINGINFO NET#) of BASE))
                    [COND
                       ((OR [AND (SETQ BUCKET (\GETBASEPTR TABLE (UNFOLD (LOGAND NET MASK)
                                                                         WORDSPERCELL)))
                                 (when (EQ (fetch RTNET# of (SETQ ENTRY (CAR BUCKET)))
                                           NET)
                                    do (RETURN T) repeatwhile (SETQ BUCKET (CDR BUCKET]
                            (COND
                               ((<= HOPS RADIUS)
                                [\ADD.ROUTING.TABLE.ENTRY TABLE (SETQ ENTRY (create ROUTING
                                                                                     RTNET# _ NET
```

```
                                                                                    RTTIMER _ (SETUPTIMER 0]
                                T)))
                        ;; Have an entry for this net.  Shall we accept the new info?
                        (COND
                           ((EQ (SETQ OLDHOPS (fetch RTHOPCOUNT of ENTRY))
                                0)                               ; Don't touch the directly connected net

                            )
                           ((COND
                               ((AND (EQ NDB (fetch RTNDB of ENTRY))
                                     (EQ HOST (fetch RTGATEWAY# of ENTRY)))
                                ;; Same net and gateway, so we'll want to update the hop count
                                T)
                               ((OR (NOT (fetch RTRECENT of ENTRY))
                                    (< HOPS OLDHOPS))
                                ;; Shorter route than we had, or the old route was getting out of date.  Note we only smash these fields on
                                ;; this arm of the cond, since they're unchanged on the other arm.  Smashing there would be slow, especially
                                ;; since NDB's tend to have overflowed ref counts.  Also note OLDHOPS is NIL for brand new entry, which is
                                ;; why we check RECENT first.
                                (replace RTGATEWAY# of ENTRY with HOST)
                                (replace RTNDB of ENTRY with NDB)
                                T))
                            (replace RTHOPCOUNT of ENTRY with HOPS)
                            (COND
                               ((< HOPS \RT.INFINITY)            ; Hops at infinity means inaccessible, so don't encourage this
                                                                 ; entry to stick around.
                                (replace RTRECENT of ENTRY with T)
                                (COND
                                   (NEWTIMER                     ; Save repeatedly calling the clock--everyone can get the same
                                                                 ; timer.
                                          (\BLT (fetch RTTIMER of ENTRY)
                                                NEWTIMER WORDSPERCELL))
                                   (T (SETQ NEWTIMER (SETUPTIMER \RT.TIMEOUTINTERVAL (fetch RTTIMER of ENTRY]
                    (SETQ LENGTH (- LENGTH \PUP.ROUTINGINFO.WORDS))
                    (SETQ BASE (\ADDBASE BASE \PUP.ROUTINGINFO.WORDS]
    (\RELEASE.ETHERPACKET PUP])
```

(\ROUTE.PUP

```
  [LAMBDA (PUP READONLY)                                         (* bvm%: "15-Feb-85 22:21")
    ;; Encapsulates PUP, choosing the right network and immediate destination host.  Returns an NDB for the transmission.  Defaults the pup source
    ;; fields, unless READONLY is set
    (PROG ((NET (fetch PUPDESTNET of PUP))
           (HOST (fetch PUPDESTHOST of PUP))
           PDH ROUTE NDB)
          (COND
             [(EQ NET 0)
              (COND
                 ((NOT (SETQ NDB \LOCALNDBS))
                  (RETURN]
             ((SETQ ROUTE (\LOCATE.PUPNET NET))
              (SETQ NDB (fetch RTNDB of ROUTE)))
             (T (RETURN)))
          [SETQ PDH (COND
                       ((AND ROUTE (NEQ (fetch RTHOPCOUNT of ROUTE)
                                        0))
                        (fetch RTGATEWAY# of ROUTE))
                       ((EQ (fetch NETTYPE of NDB)
                            3)
                        HOST)
                       ((EQ HOST 0)                              ; Broadcast
                        BROADCASTNSHOSTNUMBER)
                       ((\TRANSLATE.3TO10 HOST NDB))
                       (T (RETURN]
          (replace EPNETWORK of PUP with NDB)
          (ENCAPSULATE.ETHERPACKET NDB PUP PDH (fetch PUPLENGTH of PUP)
                 (ffetch NDBPUPTYPE of NDB))
          [COND
             ((NOT READONLY)
              [COND
                 ((EQ NET 0)
                  (replace PUPDESTNET of PUP with (fetch NDBPUPNET# of NDB]
              (replace PUPSOURCENET of PUP with (fetch NDBPUPNET# of NDB))
              (COND
                 ((EQ (fetch PUPSOURCEHOST of PUP)
                      0)
                  (replace PUPSOURCEHOST of PUP with (fetch NDBPUPHOST# of NDB]
          (RETURN NDB])
```

(\LOCATE.PUPNET

```
  [LAMBDA (NET DONTPROBE)                                        ; Edited 29-Sep-89 10:28 by jds
    ;; Returning routing info entry for NET, or NIL if not in table.  If not found, initiates a probe for the net, unless DONTPROBE is true.
    (OR (SMALLP NET)
        (HELP "Bad network number" NET))
    (OR \PUP.READY (ASSURE.PUP.READY))
    (LET [(BUCKET (\GETBASEPTR \PUP.ROUTING.TABLE (UNFOLD (LOGAND NET \ROUTING.TABLE.MASK)
                                                        WORDSPERCELL]
         (for DATA in BUCKET when [OR (EQL (fetch (ROUTING RTNET#) of DATA)
                                          NET)
                                     (AND (EQ 0 NET)
                                          (EQ 0 (fetch (ROUTING RTHOPCOUNT) of DATA]
            do (RETURN (AND (< (fetch RTHOPCOUNT of DATA)
                               \RT.INFINITY)
                            DATA))
            finally (COND
                     ((EQ 0 NET)                                  ; Net is 0 -- the local net, so return a routing showing 0 hops to
                                                                  ; that net.
                      (RETURN (create ROUTING
                                     RTNET# _ NET
                                     RTHOPCOUNT _ 0)))
                     ((NOT DONTPROBE)                             ; Insert an entry for the net, to be purged in 30 sec if router
                                                                  ; process hasn't filled it by then
                      (\RPLPTR \PUP.ROUTING.TABLE (UNFOLD (LOGAND NET \ROUTING.TABLE.MASK)
                                                        WORDSPERCELL)
                             (CONS (create ROUTING
                                          RTNET# _ NET
                                          RTHOPCOUNT _ \RT.INFINITY
                                          RTTIMER _ (SETUPTIMER 72460Q))
                                   BUCKET))
                      (SETQ \PUPROUTER.PROBECOUNT 5)
                      (SETQ \PUPROUTER.PROBETIMER (SETUPTIMER 0 \PUPROUTER.PROBETIMER))
                      (WAKE.PROCESS '\PUPGATELISTENER)
                      (BLOCK])
```

```
(SORT.PUPHOSTS.BY.DISTANCE
  [LAMBDA (HOSTLIST)                                              (* bvm%: " 6-MAY-83 00:18")
    (COND
       ((NULL (CDR (LISTP HOSTLIST)))
        HOSTLIST)
       (T ;; HOSTLIST is a list each of whose elements has a pup nethost in its CAR and anything in its CDR.  In particular, standard pup PORT
          ;; pairs work
          [for PAIR in HOSTLIST do (\LOCATE.PUPNET (fetch PUPNET# of (CAR PAIR]
                                                                  ; Enter request for routing for all hosts
          (BLOCK)
          (COND
             ((NOT (for PAIR in HOSTLIST always (\LOCATE.PUPNET (fetch PUPNET# of (CAR PAIR))
                                                     T)))
              (BLOCK \ETHERTIMEOUT)))
          (SORT HOSTLIST (FUNCTION \PUPNET.CLOSERP])
```

```
(\PUPNET.CLOSERP
  [LAMBDA (X Y)                                                   (* edited%: "12-APR-83 12:44")
    (PROG ((ROUTEX (\LOCATE.PUPNET (fetch PUPNET# of (CAR X))
                          T))
           ROUTEY)
          (RETURN (COND
                     ((NULL ROUTEX)
                      NIL)
                     ((SETQ ROUTEY (\LOCATE.PUPNET (fetch PUPNET# of (CAR Y))
                                          T))
                      (ILESSP (fetch RTHOPCOUNT of ROUTEX)
                             (fetch RTHOPCOUNT of ROUTEY)))
                     (T T])
```

```
(PUPNET.DISTANCE
  [LAMBDA (NET#)                                                  (* bvm%: " 1-MAR-83 16:15")
    (PROG ((ROUTE (\LOCATE.PUPNET NET#)))
          [COND
             ((NULL ROUTE)
              (to 4 do (BLOCK \ETHERTIMEOUT) repeatuntil (SETQ ROUTE (\LOCATE.PUPNET NET#]
          (RETURN (COND
                     (ROUTE (fetch RTHOPCOUNT of ROUTE])
)
```

```
(RPAQ? \PUP.ROUTING.TABLE (CONS))

(RPAQ? \PUP.ROUTING.TABLE.RADIUS 2)

(RPAQ? \PUPROUTER.PROBECOUNT 0)

(RPAQ? \PUPROUTER.PROBETIMER )

(RPAQ? \PUPROUTER.PROBEINTERVAL 5670Q)
```

```
(RPAQ? \PUP.READY )

(RPAQ? \PUP.READY.EVENT (CREATE.EVENT "Pup Ready"))

(RPAQ? \PUP.READY.LOCK (CREATE.MONITORLOCK "Pup Ready"))

(ADDTOVAR \SYSTEMCACHEVARS \PUP.READY)

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(BLOCKRECORD PUPROUTINGINFO (                                    ; Format of each entry in a pup routing info packet.  We only
                                                                 ; actually use NET# and #HOPS
                             (NET# BYTE)
                             (GATENET# BYTE)
                             (GATEHOST# BYTE)
                             (%#HOPS BYTE)))
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \PUP.ROUTINGINFO.WORDS 2)

(CONSTANTS \PUP.ROUTINGINFO.WORDS)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \PUP.ROUTING.TABLE \PUP.ROUTING.TABLE.RADIUS \PUPROUTER.PROBECOUNT \PUPROUTER.PROBETIMER
       \PUPROUTER.PROBEINTERVAL \PUP.READY \PUP.READY.EVENT \PUP.READY.LOCK)
)
)


;; Sockets

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(DATATYPE PUPSOCKET ((NIL BITS 4)
                     (PUPSOCLINK POINTER)                        ; So that we can Queue them
                     (PSOCKET# FIXP)
                     (INQUEUE POINTER)
                     (INQUEUELENGTH WORD)
                     (PUPSOC#ALLOCATION WORD)
                     (PUPSOCHANDLE WORD)                         ; Back-fitting for Bcpl
                     (PUPSOCPUPADDRESS WORD)                     ; Local net/host
                     (NIL BITS 4)
                     (PUPSOCEVENT POINTER)                       ; Event that is notified when a pup arrives on this socket
                     (NIL BITS 4)
                     (NIL POINTER))
       (BLOCKRECORD PUPSOCKET ((NIL BITS 4)
                               (NIL POINTER)
                               (PSOCKETHI WORD)
                               (PSOCKETLO WORD)))
       INQUEUE _ (create SYSQUEUE)
       PUPSOC#ALLOCATION _ \MAX.EPKTS.ON.PUPSOCKET)
)

(/DECLAREDATATYPE 'PUPSOCKET '((BITS 4)
                               POINTER FIXP POINTER WORD WORD WORD WORD (BITS 4)
                               POINTER
                               (BITS 4)
                               POINTER)

       ;; ---field descriptor list elided by lister---

       '16Q)

(DECLARE%: EVAL@COMPILE

(PUTPROPS \PUPSOCKET.FROM# MACRO (OPENLAMBDA (SOCHI SOCLO)
                                      (for SOC in \PUPSOCKETS when (AND (EQ (fetch PSOCKETLO of SOC)
                                                                            SOCLO)
                                                                        (EQ (fetch PSOCKETHI of SOC)
                                                                            SOCHI))
                                         do (RETURN SOC))))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \PUPSOCKETS.TABLE \MAX.EPKTS.ON.PUPSOCKET \PUP.CHECKSUMFLG)
)
)

(/DECLAREDATATYPE 'PUPSOCKET '((BITS 4)
                               POINTER FIXP POINTER WORD WORD WORD WORD (BITS 4)
                               POINTER
                               (BITS 4)
                               POINTER)
       ;; ---field descriptor list elided by lister---
       '16Q)

(ADDTOVAR SYSTEMRECLST
          (DATATYPE PUPSOCKET ((NIL BITS 4)
                               (PUPSOCLINK POINTER)
                               (PSOCKET# FIXP)
                               (INQUEUE POINTER)
                               (INQUEUELENGTH WORD)
                               (PUPSOC#ALLOCATION WORD)
                               (PUPSOCHANDLE WORD)
                               (PUPSOCPUPADDRESS WORD)
                               (NIL BITS 4)
                               (PUPSOCEVENT POINTER)
                               (NIL BITS 4)
                               (NIL POINTER))))

(DEFINEQ

(OPENPUPSOCKET
  [LAMBDA (SKT# IFCLASH)                                          (* bvm%: "21-JUL-83 10:36")

    ;; Creates a new local PUPSOCKET If SKT# is supplied, it is the identifying number (32-bit) of the socket, and an error occurs if that socket is
    ;; already in use.

    (PROG ((ID#EXPLICIT? (FIXP SKT#))
           PUPSOC CLASHP SOCHI SOCLO)
          [COND
             [(type? PUPSOCKET SKT#)
              (SETQ PUPSOC SKT#)
              (\FLUSHPUPSOCQUEUE PUPSOC)
              (COND
                 ((NEQ PUPSOC (\PUPSOCKET.FROM# (fetch PSOCKETHI of PUPSOC)
                                                (fetch PSOCKETLO of PUPSOC)))
                  (ERROR PUPSOC "Attempt to re-open a released PUPSOCKET."]
             (T (COND
                   (ID#EXPLICIT? (SETQ SOCHI (\HINUM SKT#))
                          (SETQ SOCLO (\LONUM SKT#)))
                   (T                                             ; Pick a socket that is reasonably random but won't conflict with
                                                                  ; well-known sockets
                      [SETQ SOCLO (LOGOR 100000Q (\LONUM (DAYTIME]
                      (SETQ SOCHI 1)))
                (UNINTERRUPTABLY
                    [do (COND
                           ((NOT (SETQ CLASHP (\PUPSOCKET.FROM# SOCHI SOCLO)))
                            (SETQ PUPSOC (create PUPSOCKET
                                                PSOCKETHI _ SOCHI
                                                PSOCKETLO _ SOCLO))
                            (replace PUPSOCEVENT of PUPSOC with (CREATE.EVENT PUPSOC))
                            (push \PUPSOCKETS PUPSOC)
                            (RETURN))
                           [(NOT ID#EXPLICIT?)
                            (SETQ SOCLO (LOGOR 100000Q (ADD1 (LOGAND SOCLO 77777Q]
                           (T (RETURN])
                (COND
                   (CLASHP (SELECTQ IFCLASH
                                ((T ACCEPT)
                                     (\FLUSHPUPSOCQUEUE (SETQ PUPSOC CLASHP)))
                                ((DON'T FAIL)
                                     (RETURN NIL))
                                (ERROR "Socket number is already in use" SKT#]
          (RETURN PUPSOC])

(CLOSEPUPSOCKET
  [LAMBDA (PUPSOC NOERRORFLG)                                     (* bvm%: " 5-MAY-83 23:58")

    ;; Closes a local PUPSOCKET -- argument = T means close all sockets

    (COND
      ((EQ PUPSOC T)
        (while \PUPSOCKETS do (\FLUSHPUPSOCQUEUE (SETQ PUPSOC (pop \PUPSOCKETS)))
                              (replace PUPSOCEVENT of PUPSOC with NIL)))
       (T (\FLUSHPUPSOCQUEUE (\DTEST PUPSOC 'PUPSOCKET))
          (PROG1 (COND
                    ((FMEMB PUPSOC \PUPSOCKETS)
                     (SETQ \PUPSOCKETS (DREMOVE PUPSOC \PUPSOCKETS))
                     T)
                    ((NOT NOERRORFLG)
                     (ERROR PUPSOC "not an open PUP socket")))
                 (replace PUPSOCEVENT of PUPSOC with NIL])

(PUPSOCKETNUMBER
  [LAMBDA (PUPSOC)                                                  (* bvm%: "14-FEB-83 15:21")
    (fetch PSOCKET# of PUPSOC])

(PUPSOCKETFROMNUMBER
  [LAMBDA (SOC#orSOCLO SOCHI)                                       (* bvm%: "21-JUL-83 11:39")
    [COND
       ((NULL SOCHI)
        (SETQ SOCHI (\HINUM SOC#orSOCLO))
        (SETQ SOC#orSOCLO (LOGAND SOC#orSOCLO 177777Q]
    (\PUPSOCKET.FROM# SOCHI SOC#orSOCLO])

(PUPSOCKETEVENT
  [LAMBDA (PUPSOC)                                                  (* bvm%: "10-MAY-83 22:32")
    (ffetch PUPSOCEVENT of (\DTEST PUPSOC 'PUPSOCKET])

(\FLUSHPUPSOCQUEUE
  [LAMBDA (PUPSOC)                                                  (* bvm%: "11-FEB-83 12:55")
    (\FLUSH.PACKET.QUEUE (fetch (PUPSOCKET INQUEUE) of PUPSOC))
    (replace (PUPSOCKET INQUEUELENGTH) of PUPSOC with 0)
    PUPSOC])
)

(DEFINEQ

(\GETMISCSOCKET
  [LAMBDA NIL                                                       (* bvm%: "14-FEB-83 15:29")

    ;; Opens a socket for miscellaneous services, if we don't have it open yet

    (COND
       ((AND \MISC.SOCKET (FMEMB \MISC.SOCKET \PUPSOCKETS))
        \MISC.SOCKET)
       (T (SETQ \MISC.SOCKET (OPENPUPSOCKET])
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \MISC.SOCKET \PUPSOCKETS)
)

(RPAQ? \MISC.SOCKET )

(RPAQ? \PUPSOCKETS )

(DECLARE%: DONTCOPY

;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

[BLOCKRECORD PORT ((NETHOST WORD)
                   (SOCKET FIXP))
       (BLOCKRECORD PORT ((NET BYTE)
                          (HOST BYTE)
                          (SOCKETHI WORD)
                          (SOCKETLO WORD]

[ACCESSFNS ERRORPUP ((ERRORPUPBASE (fetch PUPCONTENTS of DATUM)))
       (BLOCKRECORD ERRORPUPBASE ((ERRORPUPCOPY 12Q WORD)           ; Copy of pup header
                                  (ERRORPUPCODE WORD)
                                  (ERRORPUPARG WORD)                ; Usually zero
                                  (ERRORPUPSTRINGBASE WORD)         ; Human readable message

                                  ]
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \ETHERWAIT1 \ETHERTIMEOUT \MAXETHERTRIES PUPTRACEFLG LOGINPASSWORDS)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS PUPTRACEFILE PUPONLYTYPES PUPIGNORETYPES PUPPRINTMACROS)
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \PUPOVLEN 26Q)

(RPAQQ \MAX.PUPLENGTH 1024Q)

(RPAQQ \TIME.GETPUP 5)
```

```
(CONSTANTS (\PUPOVLEN 26Q)
       (\MAX.PUPLENGTH 1024Q)
       (\TIME.GETPUP 5))
)

(PUTPROPS PUPPRINTMACROS VARTYPE ALIST)

(DECLARE%: EVAL@COMPILE

(PUTPROPS \GETPUPWORD DMACRO ((PUP WORD#)
                              (\GETBASE (fetch PUPCONTENTS of PUP)
                                     WORD#)))

(PUTPROPS \PUTPUPWORD DMACRO ((PUP WORD# VALUE)
                              (\PUTBASE (fetch PUPCONTENTS of PUP)
                                     WORD# VALUE)))

(PUTPROPS \GETPUPBYTE DMACRO ((PUP BYTE#)
                              (\GETBASEBYTE (fetch PUPCONTENTS of PUP)
                                     BYTE#)))

(PUTPROPS \PUTPUPBYTE DMACRO ((PUP BYTE# VALUE)
                              (\PUTBASEBYTE (fetch PUPCONTENTS of PUP)
                                     BYTE# VALUE)))
)

(RPAQQ RAWPUPTYPES
       ((\PT.ECHOME 1)
        (\PT.IAMECHO 2)
        (\PT.IAMBADECHO 3)
        (\PT.ERROR 4)
        (\PT.RFC 10Q)
        (\PT.ABORT 11Q)
        (\PT.END 12Q)
        (\PT.ENDREPLY 13Q)
        (\PT.DATA 20Q)
        (\PT.ADATA 21Q)
        (\PT.ACK 22Q)
        (\PT.MARK 23Q)
        (\PT.INTERRUPT 24Q)
        (\PT.INTERRUPTREPLY 25Q)
        (\PT.AMARK 26Q)
        (\PT.GATEWAYREQUEST 200Q)
        (\PT.GATEWAYRESPONSE 201Q)
        (\PT.ALTOTIMEREQUEST 206Q)
        (\PT.ALTOTIMERESPONSE 207Q)
        (\PT.MSGCHECK 210Q)
        (\PT.NEWMAIL 211Q)
        (\PT.NONEWMAIL 212Q)
        (\PT.NOMAILBOX 213Q)
        (\PT.LAURELCHECK 214Q)
        (\PT.NAMELOOKUP 220Q)
        (\PT.NAMERESPONSE 221Q)
        (\PT.NAME/ADDRERROR 222Q)
        (\PT.ADDRLOOKUP 223Q)
        (\PT.ADDRRESPONSE 224Q)
        (\PT.PRINTERSTATUS 200Q)
        (\PT.STATUSRESPONSE 201Q)
        (\PT.PRINTERCAPABILITY 202Q)
        (\PT.CAPABILITYRESPONSE 203Q)
        (\PT.PRINTJOBSTATUS 204Q)
        (\PT.PRINTJOBRESPONSE 205Q)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \PT.ECHOME 1)

(RPAQQ \PT.IAMECHO 2)

(RPAQQ \PT.IAMBADECHO 3)

(RPAQQ \PT.ERROR 4)

(RPAQQ \PT.RFC 10Q)

(RPAQQ \PT.ABORT 11Q)

(RPAQQ \PT.END 12Q)

(RPAQQ \PT.ENDREPLY 13Q)

(RPAQQ \PT.DATA 20Q)

(RPAQQ \PT.ADATA 21Q)

(RPAQQ \PT.ACK 22Q)
```

```
(RPAQQ \PT.MARK 23Q)

(RPAQQ \PT.INTERRUPT 24Q)

(RPAQQ \PT.INTERRUPTREPLY 25Q)

(RPAQQ \PT.AMARK 26Q)

(RPAQQ \PT.GATEWAYREQUEST 200Q)

(RPAQQ \PT.GATEWAYRESPONSE 201Q)

(RPAQQ \PT.ALTOTIMEREQUEST 206Q)

(RPAQQ \PT.ALTOTIMERESPONSE 207Q)

(RPAQQ \PT.MSGCHECK 210Q)

(RPAQQ \PT.NEWMAIL 211Q)

(RPAQQ \PT.NONEWMAIL 212Q)

(RPAQQ \PT.NOMAILBOX 213Q)

(RPAQQ \PT.LAURELCHECK 214Q)

(RPAQQ \PT.NAMELOOKUP 220Q)

(RPAQQ \PT.NAMERESPONSE 221Q)

(RPAQQ \PT.NAME/ADDRERROR 222Q)

(RPAQQ \PT.ADDRLOOKUP 223Q)

(RPAQQ \PT.ADDRRESPONSE 224Q)

(RPAQQ \PT.PRINTERSTATUS 200Q)

(RPAQQ \PT.STATUSRESPONSE 201Q)

(RPAQQ \PT.PRINTERCAPABILITY 202Q)

(RPAQQ \PT.CAPABILITYRESPONSE 203Q)

(RPAQQ \PT.PRINTJOBSTATUS 204Q)

(RPAQQ \PT.PRINTJOBRESPONSE 205Q)

(CONSTANTS (\PT.ECHOME 1)
       (\PT.IAMECHO 2)
       (\PT.IAMBADECHO 3)
       (\PT.ERROR 4)
       (\PT.RFC 10Q)
       (\PT.ABORT 11Q)
       (\PT.END 12Q)
       (\PT.ENDREPLY 13Q)
       (\PT.DATA 20Q)
       (\PT.ADATA 21Q)
       (\PT.ACK 22Q)
       (\PT.MARK 23Q)
       (\PT.INTERRUPT 24Q)
       (\PT.INTERRUPTREPLY 25Q)
       (\PT.AMARK 26Q)
       (\PT.GATEWAYREQUEST 200Q)
       (\PT.GATEWAYRESPONSE 201Q)
       (\PT.ALTOTIMEREQUEST 206Q)
       (\PT.ALTOTIMERESPONSE 207Q)
       (\PT.MSGCHECK 210Q)
       (\PT.NEWMAIL 211Q)
       (\PT.NONEWMAIL 212Q)
       (\PT.NOMAILBOX 213Q)
       (\PT.LAURELCHECK 214Q)
       (\PT.NAMELOOKUP 220Q)
       (\PT.NAMERESPONSE 221Q)
       (\PT.NAME/ADDRERROR 222Q)
       (\PT.ADDRLOOKUP 223Q)
       (\PT.ADDRRESPONSE 224Q)
       (\PT.PRINTERSTATUS 200Q)
       (\PT.STATUSRESPONSE 201Q)
       (\PT.PRINTERCAPABILITY 202Q)
       (\PT.CAPABILITYRESPONSE 203Q)
       (\PT.PRINTJOBSTATUS 204Q)
       (\PT.PRINTJOBRESPONSE 205Q))
)

(RPAQ? PUPTYPES RAWPUPTYPES)
```

```
(RPAQQ WELLKNOWNPUPSOCKETS ((\PUPSOCKET.TELNET 1)
                            (\PUPSOCKET.ROUTING 2)
                            (\PUPSOCKET.FTP 3)
                            (\PUPSOCKET.MISCSERVICES 4)
                            (\PUPSOCKET.ECHO 5)
                            (\PUPSOCKET.EFTP 20Q)
                            (\PUPSOCKET.PRINTERSTATUS 21Q)
                            (\PUPSOCKET.LEAF 43Q)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \PUPSOCKET.TELNET 1)

(RPAQQ \PUPSOCKET.ROUTING 2)

(RPAQQ \PUPSOCKET.FTP 3)

(RPAQQ \PUPSOCKET.MISCSERVICES 4)

(RPAQQ \PUPSOCKET.ECHO 5)

(RPAQQ \PUPSOCKET.EFTP 20Q)

(RPAQQ \PUPSOCKET.PRINTERSTATUS 21Q)

(RPAQQ \PUPSOCKET.LEAF 43Q)

(CONSTANTS (\PUPSOCKET.TELNET 1)
       (\PUPSOCKET.ROUTING 2)
       (\PUPSOCKET.FTP 3)
       (\PUPSOCKET.MISCSERVICES 4)
       (\PUPSOCKET.ECHO 5)
       (\PUPSOCKET.EFTP 20Q)
       (\PUPSOCKET.PRINTERSTATUS 21Q)
       (\PUPSOCKET.LEAF 43Q))
)

;; END EXPORTED DEFINITIONS

(RPAQQ PUPCONSTANTS ((\PUPHEADERLEN 24Q)
                     (\NetMask 177400Q)
                     (\HILOCALSOCKET 1)
                     (\PORTIDLEN 3)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \PUPHEADERLEN 24Q)

(RPAQQ \NetMask 177400Q)

(RPAQQ \HILOCALSOCKET 1)

(RPAQQ \PORTIDLEN 3)

(CONSTANTS (\PUPHEADERLEN 24Q)
       (\NetMask 177400Q)
       (\HILOCALSOCKET 1)
       (\PORTIDLEN 3))
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS PUPDEBUGGING MACRO [(X . Y)
                              (COND
                                 (PUPTRACEFLG (printout PUPTRACEFILE X . Y])
)

(ADDTOVAR PUPPRINTMACROS (210Q CHARS)
                         (214Q CHARS)
                         (211Q CHARS)
                         (213Q CHARS)
                         (201Q WORDS 2 CHARS 24Q |...|)
                         (30Q CHARS))

(DECLARE%: EVAL@COMPILE

(BLOCKRECORD TIMEPUPCONTENTS ((TIMEPUPVALUEHI WORD)
                              (TIMEPUPVALUELO WORD)
                              (TIMEPUPEASTP FLAG)
                              (TIMEPUPHOURS BITS 7)
                              (TIMEPUPMINUTES BITS 10Q)
                              (TIMEPUPBEGINDST WORD)
                              (TIMEPUPENDDST WORD))              ; format of alto time response

       )
)
```

```
)

;; echo utilities

(DEFINEQ

(PUP.ECHOSERVER
  [LAMBDA (ECHOWINDOW FLG)                                          (* bvm%: " 7-AUG-83 01:11")
    (RESETLST
        (PROG ((SOC (OPENPUPSOCKET \PUPSOCKET.ECHO T))
                PUP EVENT ISGOOD)
               (RESETSAVE NIL (LIST 'CLOSEPUPSOCKET SOC))
               (OR FLG (SETQ FLG 'PEEK))
               (SETQ EVENT (fetch PUPSOCEVENT of SOC))
          LP   (COND
                  ((SETQ PUP (GETPUP SOC))
                   (SETQ ISGOOD (EQ (fetch PUPTYPE of PUP)
                                    \PT.ECHOME))
                   [COND
                      (ECHOWINDOW (SELECTQ FLG
                                        (NIL)
                                        (PEEK (PRIN1 (COND
                                                         (ISGOOD '!)
                                                         (T '?))
                                                     ECHOWINDOW))
                                        (PRINTPUP PUP NIL ECHOWINDOW]
                   (COND
                      (ISGOOD (replace TYPEWORD of PUP with \PT.IAMECHO)
                              (SWAPPUPPORTS PUP)
                              (replace EPREQUEUE of PUP with 'FREE)
                              (SENDPUP SOC PUP))
                      (T (RELEASE.PUP PUP)))
                   (BLOCK))
                  (T (AWAIT.EVENT EVENT)))
               (GO LP)))])

(PUP.ECHOUSER
  [LAMBDA (HOST ECHOSTREAM INTERVAL NTIMES)                         (* bvm%: " 1-NOV-83 15:31")
    (RESETLST
        [PROG ((OPUP (ALLOCATE.PUP))
                (PORT (BESTPUPADDRESS HOST (OR ECHOSTREAM PROMPTWINDOW)))
                (SOC (OPENPUPSOCKET))
                (TIMER (SETUPTIMER 0))
                IPUP EVENT ECHOPUPLENGTH I)
               (RESETSAVE NIL (LIST 'CLOSEPUPSOCKET SOC))
               (OR PORT (RETURN))
               (OR INTERVAL (SETQ INTERVAL 1750Q))
               (OR NTIMES (SETQ NTIMES 1750Q))
               (SETQ ECHOSTREAM (GETSTREAM (OR ECHOSTREAM T)
                                        'OUTPUT))
               (SETUPPUP OPUP PORT \PUPSOCKET.ECHO \PT.ECHOME NIL SOC T)
               (PUTPUPWORD OPUP 0 (SETQ I 1))
               (add (fetch PUPLENGTH of OPUP)
                    BYTESPERWORD)
               (PUTPUPSTRING OPUP "Random string for echo")
               (SETQ ECHOPUPLENGTH (fetch PUPLENGTH of OPUP))
               (SETQ EVENT (fetch PUPSOCEVENT of SOC))
          LP   (SENDPUP SOC OPUP)
               (PRIN1 '! ECHOSTREAM)
               (SETUPTIMER INTERVAL TIMER)
               (do (COND
                      [(SETQ IPUP (GETPUP SOC))
                       (COND
                          ((PROG1 (SELECTC (fetch PUPTYPE of IPUP)
                                        (\PT.IAMBADECHO
                                             (PRIN1 'x ECHOSTREAM))
                                        (\PT.IAMECHO (COND
                                                         ((NOT (AND (EQ (fetch PUPIDHI of IPUP)
                                                                        (fetch PUPIDHI of OPUP))
                                                                    (EQ (fetch PUPIDLO of IPUP)
                                                                        (fetch PUPIDLO of OPUP))
                                                                    (EQ (fetch PUPLENGTH of IPUP)
                                                                        ECHOPUPLENGTH)))
                                                          (PRIN1 '? ECHOSTREAM)
                                                          NIL)
                                                         ((IEQP (GETPUPWORD IPUP 0)
                                                                I)
                                                          (PRIN1 '+ ECHOSTREAM))
                                                         (T (PRIN1 "(late)" ECHOSTREAM)
                                                            NIL)))
                                        (\PT.ERROR (PRINTERRORPUP IPUP ECHOSTREAM)
                                                   NIL)
                                        (PROGN (PRIN1 '? ECHOSTREAM)
                                               NIL))
                                   (RELEASE.PUP IPUP))
                           (RETURN]
                      (T (AWAIT.EVENT EVENT TIMER T)))
                 repeatuntil (TIMEREXPIRED? TIMER) finally (COND
                                                           ((fetch EPTRANSMITTING of OPUP)
                                                            (PRIN1 "[not yet transmitted; maybe transmitter is
                                                                   off]" ECHOSTREAM)))
                                                        (PRIN1 '%. ECHOSTREAM))
              (COND
                  ((IGREATERP (OR (EQ NTIMES T)
                                  (add NTIMES -1))
                          0)
                   (PUTPUPWORD OPUP 0 (add I 1))
                   (GO LP])])
)

;; Peeking

(DEFINEQ

(\PEEKPUP
  [LAMBDA (HOST FILE)                                                (* bvm%: " 1-NOV-83 15:32")
    (PROG (NETHOST L)
          [COND
             ((NULL HOST)
              (SELECTQ (fetch NETTYPE of \LOCALNDBS)
                   (3 (\PUTBASE (EMADDRESS \ETHERHOSTLOC)
                             0
                             (fetch NDBPUPHOST# of \LOCALNDBS)))
                   (12Q)
                   NIL)
              (RPTQ 24Q (BLOCK))                                     ; empty the pipe
              (SETQ \PEEKPUPNUMBER))
             (T [COND
                    ((EQ HOST T)
                     (SETQ \PEEKPUPNUMBER T))
                    (T [SETQ L (for H inside HOST
                                  collect (PROGN (SETQ NETHOST (CAR (BESTPUPADDRESS H PROMPTWINDOW)))
                                                 (COND
                                                    ([AND NETHOST (OR (EQ (fetch PUPNET# of NETHOST)
                                                                          0)
                                                                      (EQ (fetch PUPNET# of NETHOST)
                                                                          (\LOCALPUPNETNUMBER]
                                                     (fetch PUPHOST# of NETHOST))
                                                    (T (ERROR H "not a host on local network"]
                       (SETQ \PEEKPUPNUMBER (COND
                                               ((CDR L)
                                                L)
                                               (T (CAR L]                    ; Now make us promiscuous
                 (SELECTQ (fetch NETTYPE of \LOCALNDBS)
                     (3 (\PUTBASE (EMADDRESS \ETHERHOSTLOC)
                               0 0))
                     (12Q)
                     NIL)
                 [COND
                    (FILE (SETQ PUPTRACEFILE (OR (OPENP FILE 'OUTPUT)
                                                  (OPENFILE FILE 'OUTPUT]
                 (OR PUPTRACEFLG (SETQ PUPTRACEFLG T]
          (RETURN \PEEKPUPNUMBER])

(\MAYBEPEEKPUP
  [LAMBDA (PUP)                                                      (* bvm%: " 5-Jan-85 23:39")
    [COND
       ((AND \PEEKPUPNUMBER PUPTRACEFLG)
        (PROG (DIRECTION)
              (COND
                  ([OR (EQ \PEEKPUPNUMBER T)
                       (EQ (fetch PUPDESTHOST of PUP)
                           0)
                       (for HOST inside \PEEKPUPNUMBER thereis (OR [COND
                                                                      ((EQ (fetch PUPSOURCEHOST of PUP)
                                                                           HOST)
                                                                       (SETQ DIRECTION 'PUT]
                                                                   (COND
                                                                      ((EQ (fetch PUPDESTHOST of PUP)
                                                                           HOST)
                                                                       (SETQ DIRECTION 'GET]
                   (PRINTPUP PUP DIRECTION PUPTRACEFILE NIL T]
    (\RELEASE.ETHERPACKET PUP])
)

(RPAQ? \PEEKPUPNUMBER )

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE
(RPAQQ \ETHERHOSTLOC 610Q)

(CONSTANTS \ETHERHOSTLOC)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \PEEKPUPNUMBER)
)
)

;; Debugging assistance

(DEFINEQ

(PRINTPUP
  [LAMBDA (PACKET CALLER FILE PRE.NOTE DOFILTER)                 (* bvm%: " 5-Jan-85 23:40")
    (\DTEST PACKET 'ETHERPACKET)
    (OR FILE (SETQ FILE PUPTRACEFILE))
    (PROG ((TYPE (fetch PUPTYPE of PACKET))
           MACRO LENGTH)
          (COND
             ([AND DOFILTER (COND
                                  (PUPONLYTYPES (NOT (FMEMB TYPE PUPONLYTYPES)))
                                  (PUPIGNORETYPES (FMEMB TYPE PUPIGNORETYPES]
              (PRIN1 (SELECTQ CALLER
                          ((GET RAWGET)
                              (COND
                                 ((EQ (fetch PUPDESTHOST of PACKET)
                                      0)                         ; Broadcast
                                  '*)
                                 (T '+)))
                          ((PUT RAWPUT)
                              '!)
                          '?)
                     PUPTRACEFILE)
              (RETURN)))
          (AND PRE.NOTE (PRIN1 PRE.NOTE FILE))
          (PRINTPUPROUTE PACKET CALLER FILE)
          [COND
             ((SETQ MACRO (CDR (FASSOC TYPE PUPPRINTMACROS)))
              (COND
                 ((NLISTP MACRO)
                  (RETURN (RESETFORM (OUTPUT FILE)
                                 (APPLY* MACRO PACKET FILE]
          (printout FILE "Length = " .P2 (SETQ LENGTH (fetch PUPLENGTH of PACKET))
                 " bytes" " (header + " .P2 (IDIFFERENCE LENGTH \PUPOVLEN)
                 ")" T "Type = ")
          (PRINTCONSTANT TYPE PUPTYPES FILE "\PT.")
          (printout FILE ", ID = " .P2 (fetch PUPID of PACKET)
                 T)
          (COND
             ((IGREATERP LENGTH \PUPOVLEN)                       ; Tells how to print data.  Consists of elements in pairs: a byte
                                                                 ; offset followed by a type
              (PRIN1 "Contents: " FILE)
              (PRINTPACKETDATA (fetch PUPCONTENTS of PACKET)
                     0
                     (OR MACRO '(BYTES 14Q |...|))
                     (IDIFFERENCE LENGTH \PUPOVLEN)
                     FILE)))
          (TERPRI FILE))
    PACKET])

(PRINTPUPROUTE
  [LAMBDA (PACKET CALLER FILE)                                   (* bvm%: "26-OCT-83 15:33")
    (TAB 0 0 FILE)
    (AND CALLER (printout FILE CALLER ":  "))
    (PROG ((CONTROL (fetch PUPTCONTROL of PACKET))
           CSECS)
          (printout FILE "From " (PORTSTRING (fetch PUPSOURCE of PACKET)
                                        (fetch PUPSOURCESOCKET of PACKET))
                 " to "
                 (PORTSTRING (fetch PUPDEST of PACKET)
                        (fetch PUPDESTSOCKET of PACKET)))
          [COND
             ((NEQ CONTROL 0)
              (printout FILE ", Hops = " .P2 (LRSH CONTROL 4]
          (COND
             (PUPTRACETIME (printout FILE " [" .I4 (IQUOTIENT (SETQ CSECS (\CENTICLOCK PACKET))
                                                          144Q)
                                  '%. .I2..T (IREMAINDER CSECS 144Q)
                                  "]")))
          (TERPRI FILE])
```

```
(PRINTPUPDATA
  [LAMBDA (PUP MACRO OFFSET FILE)                                 (* bvm%: "26-MAY-83 12:13")
    (PRINTPACKETDATA (fetch PUPCONTENTS of PUP)
           OFFSET MACRO (IDIFFERENCE (fetch PUPLENGTH of PUP)
                               \PUPOVLEN)
           FILE])

(PRINTERRORPUP
  [LAMBDA (PUP FILE)                                              (* bvm%: "12-FEB-83 16:24")
    (printout FILE "From " (PORTSTRING (fetch PUPSOURCE of PUP))
           ": [Error " .P2 (fetch ERRORPUPCODE of PUP)
           "] "
           (GETPUPSTRING PUP 30Q)
           T])

(PUPTRACE
  [LAMBDA (FLG REGION)                                            ; Edited 14-Jan-88 18:06 by bvm
    (MAKE-NETWORK-TRACE-WINDOW 'PUPTRACEFLG 'PUPTRACEFILE "Pup traffic" REGION FLG])

(PRINTCONSTANT
  [LAMBDA (VAR CONSTANTLIST FILE PREFIX)                          (* bvm%: " 4-APR-83 16:11")
    (PRIN2 VAR FILE)
    (COND
       ((LISTP CONSTANTLIST)
        (PRIN1 " (" FILE)
        (PRIN1 (OR [for X in CONSTANTLIST when (EQ (CADR X)
                                                   VAR)
                      do (RETURN (COND
                                    [(AND PREFIX (STRPOS PREFIX (CAR X)
                                                        1 NIL T))
                                     (SUBSTRING (CAR X)
                                            (ADD1 (NCHARS PREFIX]
                                    (T (CAR X]
                    '?)
               FILE)
        (PRIN1 ")" FILE])
)

(RPAQ? PUPTRACEFLG )

(RPAQ? PUPTRACEFILE T)

(RPAQ? PUPTRACETIME )

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS PUPTRACETIME)
)

(ADDTOVAR PUPPRINTMACROS )

(ADDTOVAR PUPONLYTYPES )

(ADDTOVAR PUPIGNORETYPES )

(ADDTOVAR PUPPRINTMACROS (4 . PRINTERRORPUP)
                           (220Q CHARS)
                           (221Q REPEAT BYTES -2 WORDS -4)
                           (223Q BYTES -2 WORDS)
                           (224Q CHARS))

(DECLARE%: DONTEVAL@LOAD

(\PUPINIT)
)

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)

(FILESLOAD (LOADCOMP)
       LLETHER)
)

(PUTPROPS PUP MAKEFILE-ENVIRONMENT (:READTABLE "INTERLISP" :PACKAGE "INTERLISP" :BASE 10Q))

(PUTPROPS PUP FILETYPE CL:COMPILE-FILE)

(PUTPROPS PUP COPYRIGHT ("Venue & Xerox Corporation" 3676Q 3677Q 3700Q 3701Q 3702Q 3703Q 3704Q 3705Q 3706Q
                                 3707Q 3710Q 3711Q 3745Q))
```

FUNCTION INDEX

CONSTANT INDEX

VARIABLE INDEX

MACRO INDEX

RECORD INDEX

OPTIMIZER INDEX

PROPERTY INDEX