

File created: 7-Feb-89 23:16:44 {ERINYES}<LISPUSERS>KOTO>NSROUTINGHASH.;2

changes to: (RECORDS NSROUTINGINFO)
(VARS NSROUTINGHASHCOMS)

previous date: 11-Jan-88 21:27:31 {ERINYES}<LISPUSERS>KOTO>NSROUTINGHASH.;1

Read Table: OLD-INTERLISP-FILE

Package: INTERLISP

Format: XCCS

(* * Copyright (c) 1987, 1988, 1989 by Xerox Corporation. All rights reserved.)

(RPAQQ **NSROUTINGHASHCOMS**

```
[ (FNS \AGE.ROUTING.TABLE.HASH \HANDLE.NS.ROUTING.INFO.NEW \HANDLE.RAW.XIP.NEW \LOCATE.NSNET.NEW
  \FLUSHNDBS.NEW \MAP.ROUTING.TABLE.NEW \NSGATELISTENER.NEW \NSROUTING.HASHBITSFN \NSROUTING.EQUIVFN
  PRINTROUTINGTABLE)
  (GLOBALVARS \NS.ROUTING.TABLE)
  (* * LOADCOMP LLNS *before* loading this module so that this record declaration is in effect)
  (RECORDS NSROUTINGINFO)
  (FNS INSTALL UNINSTALL)
  (* installation utilities)
  (COMS (* debugging tools)
    (FNS ROUTINGPROBE))
  (DECLARE: DONTVAL@LOAD DOCOPY (P (UNINTERRUPTABLY
    (INSTALL (QUOTE \FLUSHNDBS))
    (INSTALL (QUOTE \MAP.ROUTING.TABLE))
    (INSTALL (QUOTE \HANDLE.NS.ROUTING.INFO))
    (INSTALL (QUOTE \LOCATE.NSNET))
    (INSTALL (QUOTE \HANDLE.RAW.XIP))
    (INSTALL (QUOTE \NSGATELISTENER))
    (RESTART.ETHER)
    (\LOCATE.NSNET -1)))])
```

(DEFINEQ

(\AGE.ROUTING.TABLE.HASH

```
[LAMBDA (TABLE)
  (MAPHASH TABLE (FUNCTION (LAMBDA (ENTRY KEY)
    (if (if (AND (NEQ (fetch RTHOPCOUNT of ENTRY)
      0)
      (TIMEREXPIRED? (fetch RTTIMER of ENTRY)))
    then (COND
      ((fetch RTRECENT of ENTRY)
        (* New entry, make it old)
        (replace RTRECENT of ENTRY with NIL)
        (SETUPTIMER \RT.TIMEOUTINTERVAL (fetch RTTIMER of ENTRY))
        NIL)
      (T \RT.PURGEFLG)))
    then (PUTHASH KEY NIL TABLE]))
```

; Edited 21-Jun-87 23:23 by BRIGGS

(\HANDLE.NS.ROUTING.INFO.NEW

```
[LAMBDA (XIP)
  (* edited: "11-Jan-88 20:48")
  ; Edited 21-Jun-87 23:11 by BRIGGS
  (* Processes a routing info XIP)

  [COND
    ((EQ (fetch XIPFIRSTDATAWORD of XIP)
      \XROUTINGINFO.OP.RESPONSE)
      (* Unless we're a gateway, we only handle responses)
      (PROG ((HOST (fetch XIPSOURCEHOST of XIP))
        (NDB (fetch EPNETWORK of XIP))
        (LENGTH (SUB1 (FOLDLO (IDIFFERENCE (fetch XIPLength of XIP)
          \XIPOVLEN)
          BYTESPERWORD))))
        (BASE (\ADDBASE (fetch XIPCONTENTS of XIP)
          1))
        ENTRY NET HOPS NETHASH)
      [COND
        ((NEQ (fetch NETTYPE of NDB)
          10)
          (OR (SETQ HOST (\TRANSLATE.10TO3 HOST NDB))
            (RETURN)
            (SETQ \NSROUTER.PROBECOUNT 0)
            (while (IGEQL LENGTH \NS.ROUTINGINFO.WORDS)
              do (SETQ HOPS (fetch (NSROUTINGINFO #HOPS) of BASE))
              [COND
                ((OR (SETQ ENTRY (GETHASH BASE \NS.ROUTING.TABLE))
                  (COND
                    ((ILEQ HOPS \NS.ROUTING.TABLE.RADIUS)
                     (SETQ NET (fetch (NSROUTINGINFO NET#) of BASE))
                     (PUTHASH NET (SETQ ENTRY (create ROUTING
                       RTNET# _ NET
                       RTTIMER _ (SETUPTIMER 0)))
                     \NS.ROUTING.TABLE)
                    (T)))
                (T)))
```

```

(COND
  ([AND (NEQ (fetch RTHOPCOUNT of ENTRY)
    0)
    (OR (NOT (fetch RTRECENT of ENTRY))
      (AND (EQUAL HOST (fetch RTGATEWAY# of ENTRY))
        (EQ NDB (fetch RTNDB of ENTRY)))
      (ILESSP HOPS (fetch RTHOPCOUNT of ENTRY]
    (replace RTGATEWAY# of ENTRY with HOST)
    (replace RTNDB of ENTRY with NDB)
    (replace RTHOPCOUNT of ENTRY with HOPS)
  (COND
    ((ILESSP HOPS \RT.INFINITY)
      (replace RTRECENT of ENTRY with T)
      (SETUPTIMER \RT.TIMEOUTINTERVAL (fetch RTTIMER of ENTRY])
    (SETQ LENGTH (IDIFFERENCE LENGTH \NS.ROUTINGINFO.WORDS))
    (SETQ BASE (\ADDBASE BASE \NS.ROUTINGINFO.WORDS]
  (PACKET XIP]))

```

(* edited: "11-Jan-88 20:47")
(* N.H.Briggs "21-Jun-87 23:53")

```
(COND
  ((EQ TYPE \EPT.XIP)
    [PROG (NSOC CSUM NDB DESTNET MYNET)
      [COND
        ((NULL \NS.READY)
          (RETURN (RELEASE.XIP XIP)]
        [COND
          ((AND (NOT (EQNSHOSTNUMBER (fetch XIPDESTHOST of XIP)
            \MY.NSHOSTNUMBER))
            (NOT (EQNSHOSTNUMBER (fetch XIPDESTHOST of XIP)
              BROADCASTNSHOSTNUMBER))) (* Not for us)
            (RETURN (\FORWARD.XIP XIP]
          (SETQ NDB (fetch EPNETWORK of XIP))
        [COND
          ((AND [NOT (IEQP (SETQ DESTNET (fetch XIPDESTNET of XIP))
            (SETQ MYNET (fetch NDBNSNET# of NDB]
            (NEQ MYNET 0)
            (NEQ DESTNET 0)) (* explicitly for a net other than us)
            (RETURN (\FORWARD.XIP XIP]
        [COND
          [[NULL (SETQ NSOC (\NSOCKET.FROM# (fetch XIPDESTSOCKET of XIP)
            (* Packets addressed to non-active sockets are just ignored.)
            (COND
              (XIPTRACEFLG (PRIN1 (QUOTE '&)
                XIPTRACEFILE)))
            (PROG (XIPBASE)
              (COND
                [(AND (EQ (fetch XIPTYPE of XIP)
                  \XIPT.ECHO)
                  (EQ (fetch XIPDESTSOCKET of XIP)
                    \NS.WKS.Echo)
                  (EQ (\GETBASE (SETQ XIPBASE (fetch XIPCONTENTS of XIP))
                    0)
                    \XECHO.OP.REQUEST)) (* Play echo server)
                  (COND
                    ([AND (NEQ (SETQ CSUM (fetch XIPCHECKSUM of XIP))
                      MASKWORD1'S)
                      (NEQ CSUM (\CHECKSUM (fetch XIPCHECKSUMBASE of XIP)
                        (SUB1 (FOLDHI (fetch XIPLength of XIP)
                          BYTESPERWORD]
                      (\XIPERROR XIP \XIPE.CHECKSUM))
                    (T (\PUTBASE XIPBASE 0 \XECHO.OP.REPLY)
                      (SWAPXIPADDRESSES XIP)
                      (replace EPREQUEUE of XIP with (QUOTE FREE))
                      (SENDXIP NIL XIP]
                    (T (\XIPERROR XIP \XIPE.NOSOCKET]
                  ((IGEQ (fetch (NSOCKET INQUEUELENGTH) of NSOC)
                    (fetch (NSOCKET NSOC#ALLOCATION) of NSOC))
                    (* Note that packets are just "dropped" when the queue
                      overflows.)
                    (\XIPERROR XIP \XIPE.SOCKETFULL))
                  ([AND \NS.CHECKSUMFLG (NEQ (SETQ CSUM (fetch XIPCHECKSUM of XIP))
                    MASKWORD1'S)
                    (NEQ CSUM (\CHECKSUM (fetch XIPCHECKSUMBASE of XIP)
                      (SUB1 (FOLDHI (fetch XIPLength of XIP)
                        BYTESPERWORD]
```

```

(\XIPERROR XIP \XIPE.CHECKSUM))
(T [COND
  ((EQ DESTNET 0)
    (* Fill in unspecified destination net
    (possibly redundantly with zero))
    (replace XIPDESTNET of XIP 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 NDBNSNET# of NDB with DESTNET)
    (replace NSNET of \MY.NSADDRESS with (SETQ \MY.NSNETNUMBER DESTNET))
    (PROG ((ENTRY (\LOCATE.NSNET DESTNET T))
      (OR ENTRY (PUTHASH DESTNET (SETQ ENTRY (create ROUTING
        RTNET# _ DESTNET))
          \NS.ROUTING.TABLE))
      (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 (NSOCKET INQUEUE) of NSOC)
        XIP)
      (add (fetch (NSOCKET INQUEUELENGTH) of NSOC)
        1)
      (NOTIFY.EVENT (fetch NSOCEVENT of NSOC)))])
  T])

```

(\LOCATE.NSNET.NEW

[LAMBDA (NET DONTPROBE)

 (* edited: "11-Jan-88 20:49")
 (* N.H.Briggs "21-Jun-87 23:54")

```

(LET ((DATA (GETHASH NET \NS.ROUTING.TABLE)))
  (if DATA
    then (AND (ILESSP (fetch RTHOPCOUNT of DATA)
      \RT.INFINITY)
      DATA)
    elseif (NOT DONTPROBE)
      then (PUTHASH NET (create ROUTING
        RTNET# _ NET
        RTHOPCOUNT _ \RT.INFINITY
        RTIMER _ (SETUPTIMER 30000))
        \NS.ROUTING.TABLE)
      (* Insert an entry for the net, to be purged in 30 sec if router process hasn't filled it by then)
      (SETQ \NSROUTER.PROBECOUNT 5)
      (SETQ \NSROUTER.PROBETIMER (SETUPTIMER 0 \NSROUTER.PROBETIMER))
      (WAKE.PROCESS (QUOTE \NSGATELISTENER))
      (BLOCK)
      ;; return NIL in this case to indicate we didn't find it yet.
      NIL))

```

(\FLUSHNDBS.NEW

[LAMBDA (EVENT)

 (* edited: "11-Jan-88 21:20")
 (* bvm: "4-AUG-83 22:51")

```

[bind NDB QUEUE while (SETQ NDB \LOCALNDBS) do (SETQ \LOCALNDBS (fetch NDBNEXT of NDB))
  (replace NDBNEXT of NDB with NIL)
  (COND
    ((EQ EVENT (QUOTE RESTART))
      (APPLY* (fetch NDBETHERFLUSHER of NDB)
        NDB))
    (DEL.PROCESS (fetch NDBWATCHER of NDB))
    (replace NDBWATCHER of NDB
      with (replace NDBTRANSLATIONS of NDB with NIL))
    (COND
      ((SETQ QUEUE (fetch NDBTQ of NDB))
        (\FLUSH.NDB.QUEUE QUEUE EVENT (QUOTE OUTPUT))
        (* Don't do this just yet, because of possible race in
        \PUPGATELISTENER -
        (replace NDBTQ of NDB with NIL))
      )
    (COND
      ((SETQ QUEUE (fetch NDBIQ of NDB))
        (\FLUSH.NDB.QUEUE QUEUE EVENT (QUOTE INPUT))
        (replace NDBIQ of NDB with NIL))
      )
  )
  (SETQ \PUP.ROUTING.TABLE (CONS))
  (SETQ \NS.ROUTING.TABLE (HASHARRAY 100 50 (FUNCTION \NSROUTING.HASHBITSFN)
    (FUNCTION \NSROUTING.EQUIVFN))

```

(\MAP.ROUTING.TABLE.NEW

[LAMBDA (TABLE MAPFN)

 (* edited: "11-Jan-88 20:53")
 (* bvm: "22-SEP-83 14:21")

(if (HARRAYP TABLE)

```

    then (MAPHASH TABLE MAPFN)
    else (for ENTRY in (APPEND (CDR (OR TABLE \PUP.ROUTING.TABLE))) do (APPLY* MAPFN ENTRY))

```

(\NSGATELISTENER.NEW

[LAMBDA NIL

(* edited: "11-Jan-88 20:47")

; Edited 16-Jun-87 15:32 by BRIGGS

```

(PROG ((NSOC (OPENNSOCKET \NS.WKS.RoutingInformation T))
      (TIMER (SETUPTIMER 0))
      EVENT XIP BASE)
  (SETQ EVENT (fetch NSOC EVENT of NSOC))
  LP (COND
      ((SETQ XIP (GETXIP NSOC))
       (\HANDLE.NS.ROUTING.INFO XIP)
       (BLOCK))
      ((EQ (AWAIT.EVENT EVENT (COND
                                ((IGREATERP \NSROUTER.PROBECOUNT 0)
                                 \NSROUTER.PROBETIMER)
                                (T TIMER))
           T)
       EVENT)
      (GO LP)))
  (COND
    ((TIMEREXPIRED? TIMER)
     (\AGE.ROUTING.TABLE.HASH \NS.ROUTING.TABLE)
     (SETUPTIMER \RT.AGEINTERVAL TIMER)))
  [COND
    ((AND (IGREATERP \NSROUTER.PROBECOUNT 0)
          (TIMEREXPIRED? \NSROUTER.PROBETIMER))
     (* Routing info desired. Broadcast a routing request on each
        directly-connected net)
     [SETQ XIP (\FILLINKIP \XIPT.ROUTINGINFO NSOC BROADCASTNSHOSTNUMBER \NS.WKS.RoutingInformation 0
                          (IPLUS \XIPOVLEN BYTESPERWORD (UNFOLD \NS.ROUTINGINFO.WORDS BYTESPERWORD))
                          (replace XIPFIRSTDATAWORD of XIP with \XROUTINGINFO.OP.REQUEST)
                          (SETQ BASE (\ADDBASE (fetch XIPCONTENTS of XIP)
                                             1))
                          (replace (NSROUTINGINFO NET#) of BASE with -1)
                          (replace (NSROUTINGINFO #HOPS) of BASE with \RT.INFINITY)
                          (SENDXIP NSOC XIP)
                          (SETUPTIMER \NSROUTER.PROBEINTERVAL \NSROUTER.PROBETIMER)
                          (SETQ \NSROUTER.PROBECOUNT (SUB1 \NSROUTER.PROBECOUNT))
                          (GO LP)])
    ]

```

(\NSROUTING.HASHBITSFN

[LAMBDA (OBJECT)

; Edited 21-Jun-87 23:08 by BRIGGS

```

  (SELECTQ (TYPENAME OBJECT)
    (ETHERPACKET
      (LOGXOR (fetch (NSROUTINGINFO NET#-HI) of OBJECT)
              (fetch (NSROUTINGINFO NET#-LO) of OBJECT)))
    (SMALLP
      OBJECT)
    (FIXP
      (LOGXOR (\GETBASE OBJECT 0)
              (\GETBASE OBJECT 1)))
    (ERROR "Illegal arg (neither FIXP, SMALLP, nor ETHERPACKET)" OBJECT))

```

(\NSROUTING.EQUIVFN

[LAMBDA (X Y)

(* N.H.Briggs "22-Jun-87 14:34")

```

  (SELECTQ (TYPENAME X)
    (ETHERPACKET (SELECTQ (TYPENAME Y)
      (SMALLP (AND (EQ (fetch (NSROUTINGINFO NET#-HI) of X)
                      0)
                   (EQ (fetch (NSROUTINGINFO NET#-LO) of X)
                       Y)))
      (FIXP (AND (EQ (fetch (NSROUTINGINFO NET#-HI) of X)
                    (\GETBASE Y 0))
                (EQ (fetch (NSROUTINGINFO NET#-LO) of X)
                    (\GETBASE Y 1))))
      (ETHERPACKET (AND (EQ (fetch (NSROUTINGINFO NET#-HI) of X)
                            (fetch (NSROUTINGINFO NET#-HI) of Y))
                       (EQ (fetch (NSROUTINGINFO NET#-LO) of X)
                            (fetch (NSROUTINGINFO NET#-LO) of Y))))
      NIL))
    (SMALLP (SELECTQ (TYPENAME Y)
      (SMALLP (EQ X Y))
      (FIXP (EQUAL X Y))
      (ETHERPACKET (AND (EQ (fetch (NSROUTINGINFO NET#-HI) of Y)
                            0)
                       (EQ (fetch (NSROUTINGINFO NET#-LO) of Y)
                           X)))
      NIL))
    (FIXP (SELECTQ (TYPENAME Y)
      (SMALLP FIXP)
      (EQUAL X Y))
      (ETHERPACKET (AND (EQ (fetch (NSROUTINGINFO NET#-HI) of Y)
                            (\GETBASE X 0))

```

```
(EQ (fetch (NSROUTINGINFO NET#-LO) of Y)
  (\GETBASE X 1)))
```

```
  NIL))
NIL])
```

(PRINTROUTINGTABLE

```
[LAMBDA (TABLE SORT? FILE)
```

```
(* edited: "11-Jan-88 21:25")
```

```
(* N.H.Briggs "14-Dec-87 12:17")
```

```
(PROG (HASHENTRIES)
  (SELECTQ TABLE
    (NS [MAPHASH \NS.ROUTING.TABLE (FUNCTION (LAMBDA (X)
      (push HASHENTRIES X]
      (SETQ TABLE (CONS NIL HASHENTRIES)))
    ((NIL PUP)
      (SETQ TABLE \PUP.ROUTING.TABLE))
    NIL)
  (RESETFORM (RADIX 8)
    (printout FILE " Net# Gateway #Hops Recent?" T)
    (for ENTRY in (COND
      (SORT? (SORT (APPEND (CDR TABLE))
        (if (EQ SORT? (QUOTE HOPS))
          then [FUNCTION (LAMBDA (X Y)
            (ILESSP (fetch RTHOPCOUNT of X)
              (fetch RTHOPCOUNT of Y]
            else T)))
      (T (CDR TABLE)))
    (bind GATE do (printout FILE .I6.8 (fetch RTNET# of ENTRY))
      (COND
        ((NOT (SETQ GATE (fetch RTGATEWAY# of ENTRY)))
          (PRIN1 " --- " FILE))
        ((FIXP GATE)
          (printout FILE .I9.8 GATE))
        (T (SPACES 2 FILE)
          (PRINTNSHOSTNUMBER GATE FILE)))
      (printout FILE 30 .I2 (fetch RTHOPCOUNT of ENTRY)
        (COND
          ((fetch RTRECENT of ENTRY)
            " Yes")
          ((TIMEREXPIRED? (fetch RTTIMER of ENTRY))
            " timed out")
          (T " No"))
        T))
    (TERPRI FILE])
```

```
)
```

```
(DECLARE: DOEVAL@COMPILE DONTCOPY
```

```
(GLOBALVARS \NS.ROUTING.TABLE)
)
```

```
(* * LOADCOMP LLNS *before* loading this module so that this record declaration is in effect)
```

```
(DECLARE: EVAL@COMPILE
```

```
[BLOCKRECORD NSROUTINGINFO (
```

```
(* Format of each entry in a routing info packet, the hashing code relies on the fact that the net number comes first.)
```

```
(NET#-HI WORD)
(NET#-LO WORD)
(#HOPS WORD))
(AccessFNS ((NET# (\GETBASEFIXP DATUM 0)
  (\PUTBASEFIXP DATUM 0 NEWVALUE]
```

```
)
```

```
(DEFINEQ
```

(INSTALL

```
[LAMBDA (FN)
  (if [NOT (GETD (MKATOM (CONCAT FN ".OLD")
    then (MOVD FN (MKATOM (CONCAT FN ".OLD"))
      NIL T))
  (MOVD (MKATOM (CONCAT FN ".NEW"))
    FN NIL T])
```

```
; Edited 21-Jun-87 22:08 by BRIGGS
```

(UNINSTALL

```
[LAMBDA (FN)
  (if (GETD (MKATOM (CONCAT FN ".OLD")))
    then (MOVD (MKATOM (CONCAT FN ".OLD"))
      FN NIL T])
```

```
; Edited 21-Jun-87 22:08 by BRIGGS
```

```
)
```

(* * installation utilities)

(* * debugging tools)

(DEFINEQ

(ROUTINGPROBE

[LAMBDA NIL

; Edited 17-Jun-87 18:16 by BRIGGS

(LET ((NSOC (OPENNSOCKET \NS.WKS.RoutingInformation T))
XIP BASE)

[SETQ XIP (\FILLINXIP \XIPT.ROUTINGINFO NSOC BROADCASTNSHOSTNUMBER \NS.WKS.RoutingInformation 0
(IPLUS \XIPOVLEN BYTESPERWORD (UNFOLD \NS.ROUTINGINFO.WORDS BYTESPERWORD]

(**replace** XIPFIRSTDATAWORD **of** XIP **with** \XROUTINGINFO.OP.REQUEST)

(SETQ BASE (\ADDBASE (**fetch** XIPCONTENTS **of** XIP)
1))

(**replace** (NSROUTINGINFO NET#) **of** BASE **with** -1)

(**replace** (NSROUTINGINFO #HOPS) **of** BASE **with** \RT.INFINITY)

(SENDXIP NSOC XIP])

)

(DECLARE: DONTVAL@LOAD DOCOPY

(UNINTERRUPTABLY

(**INSTALL** (QUOTE \FLUSHNDBS))

(**INSTALL** (QUOTE \MAP.ROUTING.TABLE))

(**INSTALL** (QUOTE \HANDLE.NS.ROUTING.INFO))

(**INSTALL** (QUOTE \LOCATE.NSNET))

(**INSTALL** (QUOTE \HANDLE.RAW.XIP))

(**INSTALL** (QUOTE \NSGATELISTENER))

(RESTART.ETHER)

(\LOCATE.NSNET -1))

)

(PUTPROPS **NSROUTINGHASH COPYRIGHT** ("Xerox Corporation" 1987 1988 1989))

FUNCTION INDEX

INSTALL	5	\FLUSHNDBS.NEW	3	\NSGATELISTENER.NEW	4
PRINTROUTINGTABLE	5	\HANDLE.NS.ROUTING.INFO.NEW	1	\NSROUTING.EQUIVFN	4
ROUTINGPROBE	6	\HANDLE.RAW.XIP.NEW	2	\NSROUTING.HASHBITSFN	4
UNINSTALL	5	\LOCATE.NSNET.NEW	3		
\AGE.ROUTING.TABLE.HASH	1	\MAP.ROUTING.TABLE.NEW	3		

RECORD INDEX

NSROUTINGINFO	5
---------------------	---
