

i2HDH-

called if the I2H pcb.iocb word is nonzero.

1.) If it is, then a test is made to see if internal flow control permits the data to be accepted.

a) If not, the process times itself out in order to try again later.

b) If so, it attempts to allocate blocks to copy the IOCB into;

b1) if they are unavailable, it times itself out for a later retry.

b2) if available, the IOCB is copied into one or more blocks, with attention given to the EOF indication, then moved onto the I2H PUT queue (and I2H loaded.) This process is also repeated until either I2H puts up no more IOCBs or the interface is blocked due to flow control/lack of resources.

How i2HDH copies imp iocbs into aggregate structures-

HDH/I2H side - IMP messages are represented by either a single compound or simple block for message mode, or by a series of one or more blocks as above for packet mode.

> In message mode, I2H IOCBs are read contiguously into a simple block or a series of component blocks, taking care to leave space for the HDH and HDLC words in the first block.

> In packet mode,

1) The leader will be read into a simple block (with two words reserved),

2) Intermediate packets into compound blocks (with two words of the first component block reserved)

3) And the last packet into a simple or compound block (with two words reserved again).

While this copy routine has the appearance of requiring distinct sections for message and packet modes, it can be made fairly homogeneous by adding the simple test that in packet mode you start all over again at the end of each IOCB.

i2HDH:

ck if i2h iocb is waiting (i2h pcb.iocb non-zero)? n -> quit

abort current i2h iocb if abort flag set

HDOQ too long? y -> set T.O. flag, quit

link & pkt level status both up? n -> set T.O. flag, quit

allocate leader & sas blocks? n -> quit

if leader-only messages from the imp

if i2h host reset -> send hdh seq. brk

else call only-cpy

;only-cpy shares code with Houtcpy

;except EOM bit is on in header

elseif leader+data from imp

note mode (pkt or msg)

call Houtcpy

;Houtcpy is a general purpose iocb-to-AS copy routine which is smart
;enough to leave 2-word blanks in appropriate places for L2 & HDH headers.
;it saves status & length of msg whenever it jmps to HDHout so that it will
;resume in correct fashion

;counts pkts in pkt mode to insure no more than 9 are sent, abort if error
;it can only make valid packet sizes for the current mode:
;PKT MODE-even # bytes only in middle packets 2-126
;1-125 bytes in last packet

;MSG MODE- 0-1007 bytes only

;actual iocb-to-AS mechanisms are similar to x.25 L3

Houtcpy:

try to allocate AS blocks to copy iocb into? can't -> set T.0.flg,quit

if pkt mode, skip 2 words for header

copy iocb to CAS or SAS blks until EOM or sizelimit

EOM or sizelimit found? y -> note status, call HDHout

if end of iocb & msg mode,

call Hret.i2h ;routine to flush iocb & gpr i2h

ck if i2h has another iocb? n -> set T.0. flg, note status, quit

jmp Houtcpy for nxt iocb

if end of iocb & pkt mode

note status

call Hret.i2h

fall into HDHout

;HDHout jmps to Houtcpy when done if pkt mode
;to resume copying

;A <- pointer to AS blk, X <- Length count

HDHout:

adjust for HDH header at front & host padding at rear

get hdn header & customize using HDH status for this packet

enq on HDOQ

if pkt mode AND we have'nt just seen an EOM

jmp Houtcpy for nxt pkt

else jmp i2HDH for nxt msg

iH.xdv: implements swhxdv functions, runs at i2h priority in i2h process space

HDH Software XDV- The A register is used to determine which XDV function to emulate. This code is mostly identical to the x.25 equivalent.

In the function descriptions below, the software host-supplied XDV routine is called swhxdv, and it is called with the same A register value supplied to the software XDV routine. The swhxdv routine does not return any results. The functions performed are as follows:

A:

```
; Function 0 - abort
ip.xd0: sta dt.ipk x      ; set ItoL3 abort flag nonzero
ip.xdr: gpr               ; goad L3 process
        pcb              ; restore I2H PCB
ip.xdx: lda ireg          ; restore function code
        jmp ip.xdv i      ; return
```

```
; Function 1 - raise ready line
ip.xd1: cra 0&dtimpd      ; set IMP status to UP
ip.xdc: sta dt.rdy x      ; in L3 PCB
        jmp ip.xdr        ; goad L3 and exit
```

```
; Function 2 - lower ready line
ip.xd2: lda [dtimpd]       ; set IMP status to DOWN
        ; HDH line down sequence
        jmp ip.xdc        ; in L3 PCB and goad
```

```
; Function 3 - get status
ip.xd3=ip.xdx              ; ignore and just restore A
```

```
; Function 4 - loop
ip.xd4=ip.xdx              ; ignore and just restore A
```

```
; Function 5 - unloop and clear/abort
ip.xd5=ip.xd0              ; do the same as function 0
```

```

12toHDL:
a)process I frames from lev. 2
b)process control messages from lev. 2
c)process HDH control blocks from lev. 2

if hdl blocked? y -> set T.O.,quit
any input on my hdl? n -> restore pcb,quit
see if AS is data or control
    hdl control? y -> hdl.cntl
    HDH control? y -> HDH.cntl
data AS
    alloc sas for 1822 leader? n -> set t.o. flg,quit
    is HDHQ too long? y-> set T.O.,quit
count 12 thru put
setup pointers
    HDH.as ptr to AS
    HDH.ldr ldr blk pointer
    HDH.pak packet pointer
verify packet size valid for mode
    if bad,reset routine,irs errcnt & discard,try again
ck if SEQ break? y -> flush current msg,continue
is host state discarding? y -> discard
ck packet type valid,note status
enq packet on HDHQ ;less HDH header
jmp 12toHDL ;try for more

;we ascertain the type of HDH control packet & act accordingly
HDH.cntl:
get hdl header
did loop state change? y -> note status change,continue [trap?]
ck h/i bit-looped? y -> hdlhop
IHY r'cvd? -> set IHY flag in hdl status
line up status? n -> hdlrel
header lin bit set? n -> hdlldn (host took line down,note & trap if status chng
illegal? y -> irs errcnt,fall into hdlrel
hdlrel: release AS & quit
jmp to 12toHDL

;ascertain hdlc command and translate to HDH event
hdl.cntl:
try to alloc large SAS? can't -> set t.o. flg,try later
get current leader
decode command
    get control word,index
    get command to event translation
dispatch to action routine
jmp 12toHDL ;back for more

command/event translation table

link up -> link up,record status
link down -> link down,record status,call HDHldwn
break1 -> HDHtol2 break, " " ,call outflush ;flush outgoing msgs,
;snd SEQ break pkt.
break2 -> 12toHDL break " " ,call inflush ;flush ingoing msgs until
;next SOM found

```

HCHpoll:

- a)check 15-sec tick flag
- b)if ticked, process active hcb timers
- c)ck HCB for events & process them
 - ck if hello should be sent

get tick flag set by timeout process

set? n -> go ck dte events

process active hcb timer

is timer running? n -> go ck dte events
decrement

0? n - go ck dte events

set timeout event

process dte events

ck line up/down status & send SEQ brk if line just declared up

ck if hello should be sent & process

other misc. events

return

HDH2i - subroutine

- a) abort current iocb if requested
- b) process messages on the hdhq for the imp

come here if the pcb.iocb word of the associated H2I PCB is nonzero, HDH process checks its queue of messages going to the IMP.

1.) If HDHQ is empty, return

2.) If the queue is nonempty, then as much of the current

message as possible is copied into the IOCB. If the current message comes to an end, the EOF flag of the IOCB is set. The IOCB is then moved to the PUT queue and H2I is goaded. This process is repeated until either H2I puts up no more IOCBs or the queue to the IMP becomes empty.

The queue of messages going to the IMP will consist of a series of simple and compound blocks representing complete messages or fragments thereof.

> Message mode messages will be represented by a single compound or simple block.

> Packet mode messages will be represented by a series of one or more blocks, containing at least a simple block for the leader, compound blocks for intermediate packets, and simple or compound blocks for the last packet.

The copy routine will merely move data from the series of blocks into the H2I IOCBs, observing these rules:

a) when finished with a simple or compound block, finish the IOCB also, and

b) look at the first data word of each simple and compound IOCB, which will be the HDH header. Do not copy this word, but make note of the EOF flag, and set the EOF flag of the IOCB that receives the last data from the block.

Thus the copy routine should serve equally well both message and packet modes.

HDH2i:

if we are aborted?

call Hret.h2i ;flush current iocb

abort i/o

reset hdh status

ck if HDHQ has data? n -> quit

ck if h2i iocb available? n -> set T.O., quit

deq 1st blk

simple or compound?

get first word (XPGAC), ck for eof & set eof flg in iocb if found

copy SAS or CAS data into iocb

call Hret.h2i ;iocb -> h2i putq

flush SAS or CAS blks

jmp HDH2i for more

restart,background,initialization

;
; The HDH restart subroutine is called from the HDH init hook of the
; IMP initialization process when the IMP (re)starts. It sets the global
; HDH initialization flag, which will cause the background process to
; run the HDH initialization routine after the IMP comes up.

;
; The HDH background subroutine is called from the HDH background hook
; of the IMP background process each background loop. It performs the
; following functions:

- ;
; a) Check the HDH init flag and call i3init to initialize.
; b) Check the HDH shutdown flag and call i3shut to shut down.
; c) Check the NCC command location (nccc) and call i3nccc to
; take appropriate action.

HDHinit:

- a) clear global variable area
b) search imp config block for HDH hosts
c) for each HDH host, find the pcb corresponding to the HDH process.
Initialize the pcb with local (HDH) information including line speed and
HDH mode found in config word, L2 information and i2h, h2i information. Then
start up the l3 process and make it known to L2 & i2h/h2i.
d) create HDHQ (l2-to-l3) for each HDH host

Much of the code in this section is identical to the c30 x25 L3 version except
that the variable names will change to reflect the HDH equivalents & HDH
specific functionality.

Library Routines:

outflush ;flushes outgoing msgs ,sends HDH SEQUENCE break
inflush ;flushes incoming msgs,set status to ignore new messages until
;next SOM found.

Hret.h2i ;puts current iocb on h2i putq & gprs h2i
Hret.i2h ; " " " i2h " " i2h

;
; The HDH timeout process is global to all the HDH hosts. It is woken up
; nh times every 15 seconds, and each time sets the tick flag and goads
; a different host.
;

; see if there is a HDH process to poke. Look in our own private
; process table (not L2's), since we only want HDH L3s and not someone
; else's. This code is mostly the same as X.25 level 3.

<*>

L J SSSS
L J S
L J SSSS
L J S
L J J S S
LLLLLL JJJJ SSSS

Wed Oct 27 22:26:46 1982

Listing file imp.lst for Linda Seamonson at bbns.

Listing file imp.lst for Linda Seamonson at bbns.

Listing file imp.lst for Linda Seamonson at bbns.

Wed Oct 27 22:26:46 1982

L J SSSS
L J S
L J SSSS
L J S
L J J S S
LLLLLL JJJJ SSSS

L J SSSS
L J S
L J SSSS
L J S
L J S S
L J J S S
LLLLLL JJJJ SSSS

Wed Oct 27 22:26:46 1982

Listing file imp.lst for Linda Seamonson at bbns.

Listing file imp.lst for Linda Seamonson at bbns.

Listing file imp.lst for Linda Seamonson at bbns.

Wed Oct 27 22:26:46 1982

L J SSSS
L J S
L J SSSS
L J S
L J S S
L J J S S
LLLLLL JJJJ SSSS