

```
.lev var
;**must simulate buffer for tsbput
015610      V  b5typh: .block 1          ;message header
015611      V  b5blkp: .block 1          ;block pointer
015612      V  b5temp: .block 1
015613      V  b5seqh: .block 1
015614      V  b5pkth: .block 1
015615      V  b5dsth: .block 1
015616      V  b5midh: .block 1
015617      V  b5hmem: .block 1
015620      V  b5hcom: .block 1
015621      V  back5p: .block 1          ;round robin pointer
015622      V  back5i: .block 1          ;immediate action pointer
```

```
.stl TASK
.INCLUDE tsk.m4

.section pg16
.lev tsk

tskini:
016000 000401 7 tskedb: .enb tsk ;enter here if you happen to
016001 000103 7 tskdb: spr
016002 000000 7 %crash ;TSKDB: task awakened from NMF

tsk0:
016003 000401 7 tsk1: .enb tsk ;all tasks return here
016004 073657 7 ldx [stq]
016005 001001 7 .inh (all,h2i)
016006 120121 7 0 jst getqi i
016007 003000 7 0 jmp tskedb
016010 000401 7 0 .enb tsk
016011 032166 7 tskgbf: stx this ;is task queue empty?
016012 044011 7 lda srch x ;yes, debreak (but do ENB fir
016013 010172 7 sta source ;store buff addr in this
016014 044005 7 lda inch x ;save source imp
016015 011323 7 sta taskin ;save input channel
016016 100400 7 spl 0&hstmod ;from host?
016017 003062 7 jmp tskfor ;yes
016020 010000 7 sta 0 ;no, from modem, must duplicat
016021 145660 7 lda [mblock] ix ;calculate address of rsex wo
016022 011330 7 sta tinmbl ;save modem block pointer for
016023 010000 7 sta 0 ;old format line?

016024 044106 7 lda maxoctet x ;yes
016025 072166 7 ldx this ;no, get input channel number
016026 101040 7 snz ;from right half of neth
016027 003317 7 jmp tsk8ch ;channel number from either f
016030 044006 7 lda neth x ;mod 16
016031 141050 7 cal 0&chn128 ;address of bit mask for thi
016032 010001 7 tskgb1: sta ireg ;channel number div 8
016033 006072 7 ana [17] ;octet of input channel
016034 015661 7 add [bittab] ;octet number div 2
016035 011324 7 sta ackbit ;offset input modem block poi
016036 004001 7 lda ireg ;to access correct octet word
016037 040475 7 lgr 3 ;address of rsex word for thi
016040 011331 7 sta tinoct ;get odd-even bit from packet
016041 040477 7 lgr 1 ;jump if bit is 1
016042 015330 7 add tinmbl ;get rsex bits
016043 011332 7 sta tinptr ;mask with contents of bittab
016044 015662 7 add [rsex] ;if duplicate --send acks
016045 010002 7 sta jreg
016046 044006 7 lda neth x
016047 100400 7 spl 0&cdeven
016050 003315 7 jmp tskm3
016051 104002 7 lda jreg i
016052 140401 7 cma
016053 107324 7 tskm4: ana ackbit i
016054 100040 7 sze
016055 003451 7 jmp fadup
016056 044006 7 lda neth x
016057 006067 7 ana [dscpktr]
```

016060 100040 7
016061 003366 7

sze
jmp fqok

;to be discarded?
;yes, ack and flush

```
tskfor:  
016062 001001 7 .inh all  
016063 044011 7 0 lda srch x ;now check imp number  
016064 073323 7 0 ldx taskin ;(for trap report)  
016065 121663 7 0 jst [impchk] i ;check srch  
016066 021364 7 0 ;defhltn/73. invalid source imp number discovered in  
016067 072166 7 0 jst thltok  
016070 044014 7 0 ldx this  
016071 073323 7 0 ldx taskin ;range check DSTH field  
016072 121663 7 0 jst [impchk] i  
016073 021364 7 0 jst thltok ;bad imp number  
016074 010000 7 0 sta 0 ;set x=dst imp no.  
016075 144131 7 0 lda spftri ix ;get rut entry  
016076 006043 7 0 ana [spfded+route]  
016077 000401 7 0 .enb tsk  
016100 101040 7 snz 0&spf4us ;is it for us?  
016101 103664 7 jmp [forus] i ;yes  
016102 100400 7 spl 0&spfded  
016103 003310 7 jmp tskded ;for dead guy, rut wd was min  
.stl task: store and forward  
016104 006072 7 ana [route]  
016105 016052 7 sub one ;task store-and-forward  
016106 010000 7 sta 0 ;store shyfted route in x (0  
016107 011325 7 sta ourp  
016110 145660 7 lda [Emblock] ix ;modem block pointer for outp  
016111 010203 7 sta ourmb  
016112 010000 7 sta 0  
016113 044117 7 lda line x ;test for line test status  
016114 100040 7 sze ;is line dead?  
016115 003454 7 jmp fqng ;yes, wait till routing stable  
016116 001001 7 .inh (m2i,i2m,h2i)  
016117 121264 7 1 jst stats.dsfpri i ;** stats  
016120 072203 7 1 ldx ourmb ;yes  
016121 044027 7 1 lda m2ipcb.nchan x ;number of channels  
016122 056105 7 1 sub chcounter x ;subtract # of channels in us  
016123 101040 7 1 snz ;all channels in use?  
016124 003454 7 1 jmp fqng ;all in use, quit  
016125 044105 7 1 lda chcounter x ;any channels in use?  
016126 101040 7 1 snz  
016127 003145 7 1 jmp newchn ;none in use, skip tests  
016130 004216 7 1 lda nfa ;some in use, check buffer sp  
016131 016222 7 1 sub nfs ;# of free bufs  
016132 016221 7 1 sub nala ;excluding allocated  
016133 014225 7 1 add nals  
016134 016027 7 1 sub [minf]  
016135 100400 7 1 spl  
016136 003454 7 1 jmp fqng ;not enough room, quit
```

```

016137 004217 7 1 lda nsfa ;enough s/f room?
016140 016223 7 1 sub nsfs
016141 016156 7 1 sub maxs
016142 101400 7 1 smi
016143 003454 7 1 jmp fqng ;no, quit
016144 024217 7 1 irs nsfa ;count as s/f pkt
016145 004022 7 1 newchn: lda [-octmax+1)/2] ;number of chfree words
016146 010001 7 1 sta ireg
016147 044075 7 1 newch1: lda chfree x ;find a chfree word with a fr
016150 100040 7 1 sze ;any channels free in these c
016151 003156 7 1 jmp cfound ;yes, go choose one
016152 024000 7 1 irs 0 ;no, move on to next
016153 024001 7 1 irs ireg
016154 003147 7 1 jmp newch1
016155 000000 7 1 %crash
016156 033333 7 1 cfound: stx toutptr ;chcounter and chfree are inc
016157 140407 7 1 tca ;save offset modem block pair
016160 046075 7 1 ana chfree x ;single least significant bit
016161 011327 7 1 sta i2mbit ;for marking channel later
016162 052075 7 1 era chfree x ;now mark it assigned in chan
016163 050075 7 1 sta chfree x
016164 072203 7 1 ldx ourmb
016165 064105 7 1 irs chcounter x ;count channel busy
016166 000401 7 1 .enb tsk
016167 044026 7 lda m2ipcb.speed x ;calculate xmit delay on this
016170 041475 7 lgl 3 ;times 8 to get address of fi
016171 015665 7 add [delxmt] ;...word of delxmt for this sp
016172 011326 7 sta i2mslp ;(used as temp)
016173 072166 7 ldx this
016174 044004 7 lda wrdc x
016175 016065 7 sub [hdrl*400]
016176 040465 7 lgr 11. 0&pktsiz ;and with this packet length

016177 006013 7 ana [7]
016200 015326 7 add i2mslp ;now pointer to xmit delay pa
016201 011326 7 sta i2mslp ;arrival time
016202 044003 7 lda artm x ;subtract xmit delay since ar
016203 117326 7 sub i2mslp i ;subtracted from sent time at
016204 050003 7 sta artm x ;compute channel number of th
016205 005333 7 lda toutptr ;...lowest channel in the octe
016206 016203 7 sub ourmb ;...channel is in
016207 041474 7 lgl 4
016210 010000 7 sta 0
016211 005327 7 lda i2mbit
016212 024000 7 irs 0
016213 040477 7 lgr 1 ;next channel
016214 100040 7 sze ;bit shyfted out?
016215 003212 7 jmp .-3 ;no
016216 004203 7 lda ourmb ;x has channel number + 1
016217 026000 7 ima 0 ;modem block pointer in x
016220 016052 7 sub one ;compensate for extra irs
016221 010001 7 sta ireg ;store channel number in ireg
016222 054123 7 add i2mtab x ;get ptr to slot
016223 011326 7 sta i2mslp
016224 125326 7 irs i2mslp i ;mark slot in use
016225 044106 7 lda maxoctet x ;old format?
016226 101040 7 snz ;yes, move channel number to
016227 003304 7 jmp tskold ;get odd-even bit to send
016230 073333 7 newchn2: ldx toutptr

```

016231 005327 7

lda i2mbit

016232 046055 7	ana tsex x	;get o/e bit in tsex
016233 100040 7	sze	;was it zero?
016234 004071 7	lda [100000] 0&odeven	;no, set o/e bit
016235 012001 7	era ireg 0&chn128&chn8	;put channel no in a
016236 072203 7	ldx ourmb	
016237 052175 7	era zmastr x 0&endbit	;bit for lo-hi line
016240 072166 7	ldx this	;in order to detect looped line
016241 066006 7	ima neth x	;save it
016242 056006 7	sub neth x	;and adjust checksum
016243 054010 7	add chkh x	
016244 050010 7	sta chkh x	


```
.stl task: host status
.lev (tsk,t.osbck)

;get local host
;expects blk ptr in x, returns host # in a
016334 000000 5      hostno: 0
016335 044070 5      lda mbhst x
016336 141050 5      cal 0&mblhst          ;local host
016337 023666 5      cas [400-fh]         ;fake host?
016340 101000 5      nop
016341 017667 5      sub [400-fh-nh]       ;yes
016342 103334 5      jmp hostno i        ;yes, map above real hosts

.lev (tsk,bck)
;get host status, enter with phys host in x
;return reason in a, skip if host is up, preserves x
016343 000000 7      hstatr: 0
016344 144276 7      lda hosti ix          ;host up now?
016345 012052 7      era [hstup]
016346 100040 7      sze
016347 003354 7      jmp hstat1           ;no
016350 025343 7      irs hstatr          ;yes, skip return
016351 004072 7      lda [15.] 0&hstwhy    ;give funny reason
016352 013670 7      hstat0: era [hstday?hstr?hstmin?20] ;and unknown time
016353 103343 7      jmp hstatr i

016354 144307 7      hstat1: lda hdwni ix      ;do we know why host is down?
016355 006072 7      ana [hstwhy]
016356 101040 7      snz
016357 003362 7      jmp hstat2           ;no, use imp's reason
016360 144307 7      lda hdwni ix
016361 103343 7      jmp hstatr i

016362 144276 7      hstat2: lda hosti ix
016363 003352 7      jmp hstat0
```

```
.stl end-of-task
;end-of-task routines
;return a nack or ack to the sending modem, host, or back
    .lev tsk
    .lck all
016364 000000 7 0 thltok: 0                                ;report bug and give good ret
016365 120126 7 0           jst hltncc i
    .lev tsk
016366 072166 7 0 fqok:   ldx this                         ;free+quit, good return
016367 001001 7 0           inh fre
016370 120115 7 0           jst flushi i
    .lev tsk
goodm:
016371 001001 7 0           inh all
016372 005323 7 0           lda taskin
016373 100400 7 0           spl 0&hstmod
016374 003436 7 0           jmp goodh
016375 073332 7 0           ldx tinptr
016376 105324 7 0           lda ackbit i
016377 052065 7 0           era rsex x
016400 050065 7 0           sta rsex x
016401 073330 7 0 goodm2: ldx tinmbl
016402 005331 7 0           lda tinoct
016403 015661 7 0           add [bittab]
016404 010001 7 0           sta ireg
016405 104001 7 0           lda ireg i
016406 046110 7 0           ana snull x
016407 100040 7 0           sze
016410 003426 7 0           jmp goodm1
016411 104001 7 0           lda ireg i
016412 052110 7 0           era snull x
016413 050110 7 0           sta snull x
016414 044114 7 0           lda putoct x
016415 010001 7 0           sta ireg
016416 005331 7 0           lda tinoct
016417 110001 7 0           sta ireg i
016420 044114 7 0           lda putoct x
016421 141206 7 0           aoa
016422 062112 7 0           cas octend x
016423 000000 7 0           %crash
016424 044111 7 0           lda octtab x
016425 050114 7 0           sta putoct x
016426 044137 7 0 goodm1: lda i2mbsy x
016427 100040 7 0           sze
016430 003003 7 0           jmp tskl
016431 073323 7 0           ldx taskin
016432 144142 7 0           lda i2mpci ix
016433 010000 7 0           sta 0
016434 000043 7 0           gpr
016435 003003 7 0           jmp tskl

016436 140100 7 0 goodh: ssp 0&hstmod                  ;need to clear sign for hisb
016437 010000 7 0           sta 0 0&inpchn
016440 164306 7 0           irs tskfli ix
016441 164306 7 0           irs tskfli ix
016442 015671 7 0 goodh1: add [-th]
016443 101400 7 0           smi
016444 003003 7 0           jmp tskl
;double skip = good return
```

016445 144144 7 0

lda h2ipci ix

;get ptr to H2IPCB

```
016446 010000 7 0      sta 0
016447 000043 7 0      gpr
016450 003003 7 0      jmp tskl
                           ;poke it
                           ;and then we're done

.lev tsk
fqdup:
016451 001001 7        .inh fre
016452 120115 7 0      jst flushi i
016453 003401 7 0      jmp goodm2
                           ;flush buffer
                           ;set snull, poke i2m, etc
```

```
.lev tsk
016454 005323 7 fqng: lda taskin
016455 100400 7           spl 0&hstmod
016456 003463 7           jmp badh
016457 072166 7 fqmod: ldx this
016460 001001 7           .inh fre
016461 120115 7 0         jst flushi i
016462 003003 7 0         jmp tskl

.lev tsk
016463 140100 7 badh: ssp 0&hstmod
016464 010000 7           sta 0, 0&inpchn
016465 164306 7           irs tskfli ix
016466 003442 7           jmp goodh1

;free+quit, bad return
;from host?
;from modem, do not ack
;if we come here from FQNG, c
;yes, return a nack to host
;single skip = bad return
;then poke host (but not back
```

```
; .stl task: reassembly
; task reassembly begins
; .lev tsk
016467 072166 7 forus: ldx this ;pntr to pkt buff
016470 044013 7 lda pkth x
016471 006054 7 ana four 0&pktcod
016472 100040 7 sze
016473 005672 7 lda [tmbblk?rmbblk]
016474 013673 7 era [rmbblk]
016475 010167 7 sta thisb ;use block #0 for exceptional
016476 044012 7 lda seqh x
016477 010173 7 sta seqnum ;save our sequence no (mess/b
016500 141050 7 cal
016501 100040 7 sze
016502 022036 7 cas [nmb] ;blkno in range?
016503 015673 7 add [rmbblk] ;no
016504 003525 7 jmp forus1 ;no, block check fails
016505 014167 7 add thisb
016506 010001 7 sta ireg ;(temp) ptr to our block
016507 015674 7 add [mbtim]
016510 010174 7 sta mestb1 ;ptr to messno
016511 014036 7 add [Lmbmes-mbtim] 0&mbsta
016512 010175 7 sta mestb2 ;ptr to mesbts or rstate
016513 044013 7 lda pkth x
016514 072001 7 ldx ireg
016515 040474 7 lgr 4 0&usenum
016516 052250 7 era mbtim x
016517 006073 7 ana [mbusno] ;block check fails if
016520 010170 7 sta blkflg ;use numbers disagree
016521 044000 7 lda mbimp x
016522 012172 7 era source
016523 141340 7 ica ;block check fails
016524 014170 7 add blkflg ;if imp numbers disagree
016525 010170 7 forus1: sta blkflg
016526 072001 7 ldx ireg ;actual block
016527 101040 7 snz ;did check succeed?
016530 032167 7 stx thisb ;yes, save in thisb
016531 140040 7 cra
016532 010171 7 sta rngflg ;init range flag to be inrang
016533 001001 7 .inh t.o
016534 121675 7 5 jst [hostno] i ;get our local host
016535 000401 7 5 .enb tsk
016536 010176 7 sta lochst
```

```
016537 072166 7          ldx this
016540 044013 7          lda pkth x           ;trn or rep?
016541 006026 7          ana [14] 0&pktcod
016542 100040 7          sze
016543 004065 7          lda [10*mesnm1]      ;trn: mess->mess
016544 001001 7          .inh all            ;rep, type 1: mess->mess+8
016545 112174 7 0        era mestb1 i
016546 141044 7 0        car
016547 112174 7 0        era mestb1 i
016550 116174 7 0        sub mestb1 i
016551 014173 7 0        add seqnum
;spl /is mess no we got too high?
;irs rngflg /yes
016552 141140 7 0        icl
016553 022055 7 0        cas ten
016554 024171 7 0        irs rngflg
016555 024171 7 0        irs rngflg
016556 010000 7 0        sta 0
016557 044052 7 0        lda bittab x
016560 111676 7 0        sta [mesbit] i
016561 072166 7 0        ldx this
016562 044007 7 0        lda typx x
016563 006070 7 0        ana [nettyp] 0&dattyp
016564 100040 7 0        sze
016565 003610 7 0        jmp forusb
016566 044015 7 0        lda midh x
016567 006072 7 0        ana [subtyp]
016570 012011 7 0        era three 0&subunc
016571 101040 7 0        snz
016572 003627 7 0        jmp spfraw
016573 004171 7 0        lda rngflg
016574 014170 7 0        add blkflg
016575 100040 7 0        sze
016576 103677 7 0        jmp [fqok] i
016577 044013 7 0        lda pkth x
016600 006054 7 0        ana four 0&pktcod
016601 101040 7 0        snz
016602 103700 7 0        jmp [trngot] i
016603 105676 7 0        lda [mesbit] i
016604 106175 7 0        ana mestb2 i
016605 100040 7 0        sze
016606 103677 7 0        jmp [fqok] i
016607 103701 7 0        jmp [repgot] i
;is mess no we got too low?
;yes
;yes, but ok for incgot
;now convert diff between mes
;into a bit - 1,2,4, etc.
;data or network control pack
;network control
;data, is it uncontrolled?
;yes, first check if spf tagg
;in range?
;and with right block?
;no, discard
;yes, get packet code
;a transmission?
;yes
;no, a reply
;a duplicate?
;yes, discard
;no
```

```
016610 044013 7 0 forusb: lda pkth x ;get type 1 code
016611 006013 7 0 ana seven 0&pktcod
016612 010000 7 0 sta 0
016613 045617 7 0 lda fordis x ;form indirect pointer
016614 010001 7 0 sta ireg
016615 072166 7 0 ldx this
016616 102001 7 0 jmp ireg i ;dispatch

016617 017302 7 0 fordis: incgot 0&qercod
016620 021104 7 0 getblk 0&gtbcod
016621 021204 7 0 rstmmsg 0&rstcod
016622 016366 7 0 fqok
016623 020532 7 0 oorgot 0&oocod
016624 021216 7 0 gotblk 0&gtrcod
016625 021264 7 0 rstreq 0&rsqcod
016626 021307 7 0 rstrcpy 0&rsrcod
```

```
.lev tsk
.lck all
spfraw:
016627 000401 7 0     .enb tsk
016630 044007 7     lda typh x           ;is packet tagged?
016631 006062 7     ana [ldrtag]
016632 101040 7     snz
016633 102067 7     jmp [rawgot] i       ;no continue with raw packet

016634 044007 7     lda typh x           ;yes, redirect to tenex
016635 007702 7     ana [-1?ldrtag?ldrrut] ;turn off rut and tag bits
016636 050007 7     sta typh x           ;without further tagging
016637 005655 7     lda tagimp
016640 050014 7     sta dsth x           ;change dest imp
016641 005656 7     lda taghst
016642 141340 7     ica
016643 052012 7     era seqh x
016644 141044 7     car
016645 052012 7     era seqh x           ;and dest host
016646 050012 7     sta seqh x           ;compute checksum
016647 020002 7     jst cksum
016650 000000 7     %crash
016651 140407 7     tca
016652 054010 7     add chkh x
016653 050010 7     sta chkh x           ;real checksum
016654 103703 7     jmp [tskgbf] i       ;now resubmit to task

.lev con
016655 000005 C     tagimp: 5          ;imp 5
016656 000000 C     taghst: 0          ;host 0 (bbne)
```

```
.stil task: more reassembly
.section pg17
.lev tsk
.lck all
017000 044013 7 0 trngot: lda pkth x ;form message type
017001 041677 7 0 alr 1 0&mltpkt
017002 006013 7 0 ana seven 0&pktcod
017003 011542 7 0 sta messid
017004 004173 7 0 lda seqnum
017005 021346 7 0 jst rallyg ;get current type, state
017006 022011 7 0 cas three ;type>0?
017007 006011 7 0 ana three ;yes, state must be 0
017010 101040 7 0 snz ;state=3, n.g.
017011 100000 7 0 skp ;state 1,2, type=0 o.k.
017012 103701 7 0 jmp [fqok] i ;ignore message
017013 010000 7 0 sta 0
017014 044052 7 0 lda bittab x ;translate into bit
017015 073542 7 0 ldx messid
017016 047547 7 0 ana trntbl x ;corresponding bit on?
017017 101040 7 0 snz
017020 103701 7 0 jmp [fqok] i ;no, ignore message
017021 072167 7 0 ldx thisb
017022 121702 7 0 jst [blkage] i ;set age back to 4
017023 072172 7 0 ldx source
017024 144131 7 0 lda spftri ix ;get rut entry
017025 006043 7 0 ana [spfded+route]
017026 100400 7 0 spl
;defhlit(/5. recvd transmission from dead imp)
017027 121703 7 0 jst [thltok] i
017030 000401 7 0 .enb tsk
017031 005542 7 lda messid ;dispatch
017032 015704 7 add [trndis]
017033 010001 7 sta ireg
017034 072166 7 ldx this
017035 001001 7 .inh fre
017036 102001 7 0 jmp ireg i
```

017037 003120 7 0 trndis: jmp trngt1 ;single-pkt msg
017040 003120 7 0 jmp trngt8 ;multi-pkt msg
017041 003047 7 0 jmp reqgt1 ;single-pkt req
017042 003413 7 0 jmp gudral ;multi-pkt req
017043 003111 7 0 jmp gudgvb ;impossible now
017044 003111 7 0 jmp gudgvb ;giveback
017045 003413 7 0 jmp gudral ;single-pkt req inc
017046 003111 7 0 jmp gudgvb ;multi-pkt msg inc

017047 072176 7 0 reqgt1: ldx lochst ;get host for dead test
017050 144276 7 0 lda hosti ix
017051 012052 7 0 era [hstup]
017052 100040 7 0 sze
017053 003411 7 0 jmp trndr1 ;host alive?
017054 072166 7 0 ldx this ;no: req1 for dead host
017055 004053 7 0 lda two
017056 120117 7 0 jst maxchi i ;req 1 pkt, leave room for r
017057 003071 7 0 jmp nstor1 ;and all from back0
017060 005541 7 0 lda mesbit ;no storage for now
017061 100100 7 0 slz ;are we the next to go?
017062 003105 7 0 jmp gudtr0 ;yes, give to host
017063 024220 7 0 irs nrea ;no, take from reas count
017064 004166 7 0 lda this
017065 026161 7 0 ima reqstk ;put on req stk
017066 050000 7 0 sta ptrc x
017067 021377 7 0 jst ralput ;mark as req1 rcvd
017070 103705 7 0 jmp [goodm] i

017071 005541 7 0 nstor1: lda mesbit ;are we the next to go?
017072 101100 7 0 sln
017073 003413 7 0 jmp gudral ;no, quit and flush pkt
017074 004161 7 0 lda reqstk ;yes, try to flush a req
017075 101040 7 0 snz ;stack empty?
017076 003413 7 0 jmp gudral ;yes, quit and flush pkt
017077 010000 7 0 sta 0 ;no
017100 140040 7 0 cra
017101 066000 7 0 ima ptrc x ;splice req stk
017102 010161 7 0 sta reqstk ;free up buffer
017103 120115 7 0 jst flushi i ;take away from reas count
017104 024224 7 0 irs nres
017105 140040 7 0 gudtr0: cra ;make it look like non-req me
017106 011542 7 0 sta messid ;mark mesg state
017107 021377 7 0 jst ralput ;and fake reassembly block
017110 103706 7 0 jmp [t2h0] i

017111 021235 7 0 gudgvb: jstreasgt ;in reassembly?
017112 003116 7 0 jmp gudgv1 ;no, error
017113 101000 7 0 nop ;yes, no-name
017114 121707 7 0 jst [reasf] i ;yes, partial, free it up
017115 003413 7 0 jmp gudral

017116 120120 7 0 gudgv1: jst hltjst i ;defhlt(/45. no reasblk for mesg inc, inc?, or gvb)
017117 003413 7 0 jmp gudral

017120 044013 7 0 trngt1: ;single-packet msg for us
017121 011544 7 0 trngt8: lda pkth x ;multi-packet trans for us
017122 021235 7 0 sta tsktmp ;save current pkt no
017123 121703 7 0 jstreasgt ;find our guy in reas
017124 003212 7 0 ;defhlt(/37. no reas block for msg from imp=x)
017125 033543 7 0 jst [thltok] i ;no-name found, start a new m
jmp newmes ;save pntr to our reas block
stx orb

017126 033545 7 0 oldmes: stx tend ;find our slot
017127 173545 7 0 ldx tend i 0&rch&ptrc ;get next pkt ptr
017130 044013 7 0 lda pkth x ;and its pkt no
017131 023544 7 0 cas tsktmp ;compare with ours
017132 003126 7 0 jmp oldmes ;his>ours, keep looking
017133 103701 7 0 jmp [fqok] i ;his=ours, duplicate
017134 004166 7 0 lda this ;his<ours, insert
017135 127545 7 0 ima tend i 0&rch&ptrc
017136 110166 7 0 sta this i ;insert our packet
017137 073543 7 0 ldx orb ;restore reass ptr
017140 005544 7 0 lda tsktmp ;is this last packet?
017141 041477 7 0 lgl 1 081stpkt
017142 101400 7 0 smi
017143 003157 7 0 jmp notl ;no
017144 041475 7 0 lgl 3 ;yes, get packet number
017145 006074 7 0 ana [pktnum<<4]
017146 052100 7 0 era rct x ;get into lh of rct
017147 141044 7 0 car
017150 052100 7 0 era rct x ;save, get 16 + #alloc - 1
017151 066100 7 0 ima rct x ;diff = 16 + #alloc - 1 - pkt#
017152 056100 7 0 sub rct x ;-16 = surplus...
017153 006074 7 0 ana [pktnum<<4] ;/...to readjust reass. by
017154 141140 7 0 icl
017155 014224 7 0 add nres
017156 010224 7 0 sta nres
017157 044100 7 0 notl: lda rct x ;adjust reass count for surplu
017160 141340 7 0 ica
017161 052100 7 0 era rct x ;is reassembly done?
017162 101040 7 0 snz
017163 003166 7 0 jmp done ;yes
017164 064100 7 0 irs rct x 0&rsf ;no, bump count
017165 103705 7 0 jmp [goodm] i

```
017166 044100 7 0 done: lda rct x ;compute -# of pkts
017167 141050 7 0 cal
017170 140401 7 0 cma
017171 050100 7 0 sta rct x ;save inreas blk
017172 033546 7 0 stx ready ;savereas blk ptr
017173 072176 7 0 ldx lochst ;for dead test
017174 144276 7 0 lda hosti ix
017175 012052 7 0 era [hstup]
017176 100040 7 0 sze ;is host alive?
017177 003205 7 0 jmp dondoned ;no, flush all
017200 021377 7 0 jst ralput ;mark mesg state
017201 005541 7 0 lda mesbit
017202 100100 7 0 slz ;is mess no the one we want?

017203 103710 7 0 jmp [t2h] i ;yes, give to host
017204 103705 7 0 jmp [goodm] i ;no, quit

017205 073546 7 0 dondoned: ldx ready ;get pointer
017206 121707 7 0 jst [reasf] i ;flush block and packets
017207 004054 7 0 lda four
017210 011542 7 0 sta messid ;mark as dead
017211 003415 7 0 jmp gudrdd

017212 044040 7 0 newmes: lda rms x ;compute # allocated
017213 141140 7 0 icl
017214 141206 7 0 aoa
017215 011543 7 0 sta orb ;temp
017216 014220 7 0 add nrea
017217 010220 7 0 sta nrea
017220 005543 7 0 lda orb
017221 014225 7 0 add nals
017222 010225 7 0 sta nals
017223 004067 7 0 lda [(pktnum+pktnm1)*40] ;init to 16 pkts (impossible
017224 054040 7 0 add rms x ;+# allocated - 1
017225 141044 7 0 car
017226 050100 7 0 sta rct x
017227 004173 7 0 lda seqnum ;set full id
017230 050040 7 0 sta rms x
017231 033543 7 0 stx orb
017232 021377 7 0 jst ralput ;mark mesg state
017233 073543 7 0 ldx orb
017234 003126 7 0 jmp oldmes
```

```
;get reassembly block
;r1 - none found
;r2 - no-name found (rms=16+pkts allocated-1/200+blk)
;r3 - ours found (rms=mess/blk)
017235 000000 7 0reasgt: 0
017236 072166 7 0          ldx this           ;set up no-name test
017237 044013 7 0          lda pkth x        ;multi-pkt?
017240 006071 7 0          ana sign 0&m1tpkt
017241 100400 7 0          spl                ;if no, use 0 allocates
017242 004041 7 0          lda [7<<8.] 0&pktnum ;otherwise, use 7
017243 012173 7 0          era seqnum
017244 141044 7 0          car
017245 012173 7 0          era seqnum       ;+ block number
017246 012061 7 0          era [200]         ;+ no-name bit
017247 011301 7 0          sta reasgn
017250 140040 7 0          cra                ;init never saw no-name
017251 011300 7 0          sta reasge
017252 073711 7 0          ldx [-nreab+1]    ;search reas table
017253 145712 7 0          relook: lda [reasbt+rms+nreab] ix ;get id
017254 012173 7 0          era seqnum       ;compare with ours
017255 101040 7 0          snz
017256 003275 7 0          jmp reasg2      ;partial found
017257 012173 7 0          era seqnum
017260 013301 7 0          era reasgn
017261 101040 7 0          snz
017262 033300 7 0          stx reasge      ;no-name found, set flag
017263 024000 7 0          irs 0
017264 003253 7 0          jmp relook      ;loop
017265 005300 7 0          lda reasge      ;no full match
017266 072172 7 0          ldx source       ;in case of trap
017267 101040 7 0          snz      ;any no-name seen?
017270 103235 7 0          jmp reasgt i   ;no, r1 ==> none found
017271 015713 7 0          reasg1: add [reasbt+rch+nreab] ;form ptr to block
017272 010000 7 0          sta 0
017273 025235 7 0          irs reasgt      ;r2 ==> no-name found
017274 103235 7 0          jmp reasgt i
017275 004000 7 0          reasg2: lda 0      ;get reas blk index
017276 025235 7 0          irs reasgt      ;r3 ==> ours found
017277 003271 7 0          jmp reasg1
```

```

017300      V    .lev var
017301      V   reasge: .block 1
;no-name seen flag and end pt
;no-name test word

;come here on receipt of incomplete query
    .lev tsk
    .lck all

017302 004170 7 0 incgot: lda blkflg
017303 100040 7 0           sze
;block check ok?
017304 103714 7 0           jmp [outofr] i
;no, send out-of-range
017305 004173 7 0           lda seqnum
;use message to get
017306 021346 7 0           jst rallyg
;state and type info
017307 011542 7 0           sta messid
;save type*4+state
017310 004171 7 0           lda rngflg
;is mess no we got
017311 022052 7 0           cas one
;next expected-8<=<next exp
017312 103714 7 0           jmp [outofr] i
;<next expected-8 or >next ex
017313 103715 7 0           jmp [increp] i
;=next expected
017314 005542 7 0           lda messid
;in range, get state
017315 006011 7 0           ana three
017316 022053 7 0           cas two
017317 003345 7 0           jmp incign
;reply, leave alone
017320 003341 7 0           jmp incmsg
;state=2, check further

017321 100040 7 0           sze
017322 103714 7 0           jmp [outofr] i
;request
017323 121716 7 0           jst [swroom] i
;room for reply?
017324 005542 7 0           lda messid
;idle, get previous type
017325 040476 7 0           lgr 2
017326 014054 7 0           add four 0&rfncod&allcod&dedcod&irfcod
017327 011542 7 0           sta messid
;save pkt code
017330 072166 7 0           ldx this
;set up for swprep
017331 044012 7 0           lda seqh x
;will be swapped into midh
017332 141044 7 0           car 0&subtyp
017333 050012 7 0           sta seqh x
017334 004066 7 0           lda [datotyp+datcpt+prirty]
017335 111717 7 0           sta [swptyp] i
017336 072172 7 0           ldx source
017337 004173 7 0           lda seqnum
;defhlt(/49. sending duplicate reply to imp=x, messno
017340 121720 7 0           jst [hltrep] i
;send duplicate reply
; * above does not return *

017341 005542 7 0 incmsg: lda messid
;get type
017342 040476 7 0           lgr 2
017343 101000 7 0           nop
;do outofr in every case, for
;   sze
;is it all1 sent/msg rcvd ?
017344 103714 7 0           jmp [outofr] i
;no, send out-of-range
;defhlt(/66. ignoring inc?)
017345 121703 7 0 incign: jst [thltok] i
;reply, leave alone
; * the above does not return

```

```
;get rally entry, expects mess no in a,  
; returns type*4 + state in a  
    .lev tsk  
    .lck all  
017346 000000 7 0 rallyg: 0  
017347 072167 7 0          ldx thisb  
017350 052250 7 0          era mbtim x  
017351 141044 7 0          car  
017352 052250 7 0          era mbtim x  
017353 056250 7 0          sub mbtim x  
017354 141140 7 0          icl 0&mesnum      ;mess n0 - rmess  
017355 006013 7 0          ana seven  
017356 041677 7 0          alr 1  
017357 140407 7 0          tca  
017360 015376 7 0          add sftcn3  
017361 011364 7 0          sta ralgs1      ;shyft to low order bits  
017362 011371 7 0          sta ralgs2  
017363 044430 7 0          lda mbtyp x      ;get type  
017364      7 0 ralgs1: .block 1  
017365 006011 7 0          ana three  
017366 041476 7 0          lgl 2          ;*4  
017367 011375 7 0          sta ralgt1  
017370 044340 7 0          lda mbsta x      ;get state  
017371      7 0 ralgs2: .block 1  
017372 006011 7 0          ana three  
017373 013375 7 0          era ralgt1  
017374 103346 7 0          jmp rallyg i  
  
017375      .lev var  
              V ralgt1: .block 1          ;temp  
  
017376 040700 C          .lev con  
                          sftcn3: arr 0      ;shyft instr constant  
  
                          .lev tsk  
                          .lck all  
;set state, type into rally  
017377 000000 7 0 ralput: 0  
017400 073542 7 0          ldx messid  
017401 045547 7 0          lda trntbl x  
017402 141140 7 0          icl  
017403 012173 7 0          era seqnum  
017404 141050 7 0          cal  
017405 012173 7 0          era seqnum  
017406 072167 7 0          ldx thisb  
017407 121721 7 0          jst [raltyp] i  
017410 103377 7 0          jmp ralput i
```

017411 004054 7 0 trndr1: lda four ;req1 for dead: mark in rally
017412 011542 7 0 sta messid ;flush pkt
017413 072166 7 0 gudral: ldx this ;set new state, type
017414 120115 7 0 jst flushi i ;get state
017415 021377 7 0 gudrdd: jst ralput
017416 044340 7 0 nxtmes: lda mbsta x
017417 006011 7 0 ana three
017420 022053 7 0 cas two
017421 103705 7 0 jmp [goodm] i ;reply, quit
017422 003454 7 0 jmp nxtms1 ;message
017423 101040 7 0 snz ;idle, quit
017424 103705 7 0 jmp [goodm] i ;request, get type
017425 044430 7 0 lda mbtyp x
017426 006053 7 0 ana two
017427 100040 7 0 sze
017430 103705 7 0 jmp [goodm] i ;req rcvd?
017431 121722 7 0 jst [b0setf] i ;nos quit
017432 004053 7 0 lda two ;make rally active
017433 052430 7 0 era mbtyp x ;yes, make it all to be sent
017434 050430 7 0 sta mbtyp x
017435 101100 7 0 sln ;all1?
017436 003474 7 0 jmp nxtms3 ;yes
017437 044250 7 0 upmess: lda mbtim x ;increase rmess
017440 014062 7 0 add [mesnm1] 0&mesnum
017441 050250 7 0 sta mbtim x
017442 004173 7 0 lda seqnum ;increase sequence no
017443 014062 7 0 add [mesnm1]
017444 010173 7 0 sta seqnum
017445 044340 7 0 lda mbsta x ;rotate rstate and rtype
017446 040676 7 0 arr 2
017447 050340 7 0 sta mbsta x
017450 044430 7 0 lda mbtyp x
017451 040676 7 0 arr 2
017452 050430 7 0 sta mbtyp x
017453 003416 7 0 jmp nxtmes ;go on to next one
017454 044430 7 0 nxtms1: lda mbtyp x ;get type for message state
017455 006011 7 0 ana three
017456 022052 7 0 cas one
017457 003463 7 0 jmp nxtms2 ;dead and inc, same type for
017460 100000 7 0 skp ;gvb, type=rfnm for reply
017461 003531 7 0 jmp nxtms6 ;message, go search reassembly
017462 140040 7 0 cra
017463 052430 7 0 nxtms2: era mbtyp x ;set new type
017464 006011 7 0 ana three
017465 052430 7 0 era mbtyp x
017466 050430 7 0 sta mbtyp x
017467 044340 7 0 lda mbsta x ;change state to reply
017470 012052 7 0 era one
017471 050340 7 0 sta mbsta x
017472 121722 7 0 jst [b0setf] i ;make rally active
017473 003437 7 0 jmp upmess ;and bump mess no

```

017474 121723 7 0 nxtms3: jst [srcstk] i ;in req stk?
017475 003515 7 0 jmp nxtms5 ;yes
017476 004173 7 0 nxtms4: lda seqnum ;no, look at next mess no
017477 014062 7 0 add [mesnm1]
017500 010173 7 0 sta seqnum
017501 021346 7 0 jst rallyg
017502 012052 7 0 era one ;req1 rcvd?
017503 100040 7 0 sze
017504 103705 7 0 jmp [goodm] i ;no, quit
017505 121723 7 0 jst [srcstk] i ;yes, is req on stack?
017506 103705 7 0 jmp [goodm] i ;yes, quit
017507 004173 7 0 lda seqnum ;req1 rcvd, not on stack
017510 141044 7 0 car
017511 013724 7 0 era [47]
017512 072167 7 0 ldx thisb
017513 121721 7 0 jst [rallyp] i ;make it all1 to be sent
017514 003476 7 0 jmp nxtms4 ;loop

017515 072167 7 0 nxtms5: ldx thisb ;change from all1 to be sent
017516 044340 7 0 lda mbsta x
017517 012011 7 0 era three
017520 050340 7 0 sta mbsta x
017521 044430 7 0 lda mbtyp x
017522 012053 7 0 era two
017523 050430 7 0 sta mbtyp x
017524 073546 7 0 ldx ready
017525 140040 7 0 cra ;splice stack
017526 066000 7 0 ima ptrc x
017527 127545 7 0 ima tend i 0&ptrc ;give to host
017530 103725 7 0 jmp [t2h1] i

017531 021235 7 0 nxtms6: jstreasgt ;search reassembly
017532 103705 7 0 jmp [goodm] i ;none found, quit
017533 103705 7 0 jmp [goodm] i ;no-name found, quit
017534 044100 7 0 lda rct x ;match, is it complete?
017535 101400 7 0 smi
017536 103705 7 0 jmp [goodm] i ;no, quit
017537 033546 7 0 stx ready ;yes, savereas block ptr
017540 103710 7 0 jmp [t2h] i ;give to host

```

```
.lev var
017541    V  mesbit: .block 1          ;bit corresponding to our mes
017542    V  messid: .block 1          ;message type code
017543    V  orb: .block 1            ;our reas block
017544    V  tsktmp: .block 1          ;our pkt no.
017545    V  tend: .block 1            ;temp end pntr
017546    V  ready: .block 1          ;pntr to packet to give to ho

.lev con
;transmit type table
017547 005404 C  trntbl: 4+0+b0tr+b0sm+b03m  ;0- msg1
017550 005405 C              1+4+b0tr+b0sm+b03m  ;1- msg8
017551 003401 C              1+0+b0tr+b0sa+b03m  ;2- req1
017552 013401 C              1+0+b0ta+b0sa+b03m  ;3- req8
017553 025400 C              0+0+b0td+b0sm+b03m  ;4- dead
017554 015401 C              1+0+b0ta+b0sm+b03m  ;5- givb
017555 035401 C              1+0+b0ti+b0sm+b03m  ;6- inc1
017556 035405 C              1+4+b0ti+b0sm+b03m  ;7- inc8
```

```
.lev tsk
.lck all
017557 000000 7 0 srcstk: 0
017560 073726 7 0           ldx [reqstk]
017561 133727 7 0 srcstl: stx [tend] i      ;search request stack
017562 044000 7 0           lda ptrc x       ;save last
017563 101040 7 0           snz             ;get next
017564 003574 7 0           jmp srcsta      ;is stack empty?
017565 010000 7 0           sta 0            ;yes, quit
017566 044012 7 0           lda seqh x       ;compare messno/block
017567 012173 7 0           era seqnum
017570 100040 7 0           sze
017571 003561 7 0           jmp srcstl      ;no match, loop
017572 133730 7 0           stx [ready] i     ;match
017573 103557 7 0           jmp srcstk i      ;yes, non-skip return, found

017574 025557 7 0 srcstq: irs srcstk          ;skip return, not found
017575 103557 7 0           jmp srcstk i
```

```
;free reassembly block and any packets, x=reas block
.lev (t.o,tsk)
.lck all

017576 000000 5 0 reASF: 0
017577 140040 5 0           cra
017600 066100 5 0           ima rct x
017601 100040 5 0           sze
017602 003612 5 0           jmp reASF1
017603 050000 5 0           sta rch x
017604 066040 5 0           ima rms x
017605 141140 5 0           icl
017606 141206 5 0           aoa
017607 014225 5 0           add nals
017610 010225 5 0           sta nals
017611 103576 5 0           jmp reASF i

017612 101400 5 0 reASF1: smi          ;partial?
017613 006074 5 0           ana [pktnum*20] ;yes, give back pkts allocate
017614 101400 5 0           smi          ;or last pktno
017615 141140 5 0           icl
017616 100400 5 0           spl
017617 140401 5 0           cma
017620 141206 5 0           aoa
017621 014224 5 0           add nres
017622 010224 5 0           sta nres
017623 140040 5 0           cra
017624 050040 5 0           sta rms x
017625 066000 5 0           ima rch x
017626 010000 5 0           sta 0
017627 044000 5 0           lda ptrc x
017630 011637 5 0           sta reASFx
017631 120115 5 0           jst flushi i
017632 005637 5 0           lda reASFx
;**note** address comparison below ***
017633 023731 5 0           cas [sign-pkth] ;end of chain?
017634 003626 5 0           jmp reASF2   ;no
017635 103576 5 0           jmp reASF i   ;yes
017636 003626 5 0           jmp reASF2   ;no

.lev var
017637 V reASFx: .block 1
```

```
.lev tsk
;subr to check hac and get lochst, skips if ok
;expects hacmem in a, haccom in x
017640 000000 7    tskhac: 0
017641 011677 7      sta tskhc1           ;save hacmem and haccom
017642 033700 7      stx tskhc2
017643 072166 7      ldx this            ;get phys local host
017644 044012 7      lda seqh x
                           ;defplc(vha - task hook)
017645 141140 7      icl 0&dsthost        ;get local host field
017646 022056 7      cas [nh]
017647 003672 7      jmp tskhfh
017650 103640 7      jmp tskhac i
017651 010000 7    tskhas: sta 0
017652 010176 7      sta lochst          ;save local host
017653 144144 7      lda h2ipci ix
017654 010000 7      sta 0
017655 044044 7      lda h2ipcb.hacmem x
017656 007677 7      ana tskhc1          ;set index there
017657 101040 7      snz
                           ;our mem
                           ;and its mem
                           ;ok?
017660 005700 7      lda tskhc2          ;no, its com
017661 046044 7      ana h2ipcb.hacmem x
017662 100040 7      sze
                           ;and our mem (again)
                           ;ok?
017663 003667 7      jmp tskhag
017664 044045 7      lda h2ipcb.haccom x
017665 007677 7      ana tskhc1          ;yes
017666 100040 7      sze
                           ;no, our com
                           ;and its mem
                           ;ok?
017667 025640 7    tskhag: irs tskhac
017670 072166 7      ldx this
017671 103640 7      jmp tskhac i
                           ;yes, skip return
                           ;no

017672 017732 7    tskhfh: sub [400-fh]    ;is it a fake host?
017673 100400 7      spl
017674 103640 7      jmp tskhac i
017675 014056 7      add [nh]
017676 003651 7      jmp tskhas
                           ;no, access denied
                           ;yes, map it above real hosts

017677   V    tskhc1: .block 1           ;foreign hacmem
017700   V    tskhc2: .block 1           ;foreign haccom
```

```
.lev tsk
.lck all
.section pg20
rawgot:
020000 000401 7 0      .enb tsk
020001 044004 7      lda wrdc x           ;reduce pkt size
020002 016062 7      sub [pktsi1]       ;by haccom word
020003 050004 7      sta wrdc x
020004 141140 7      icl 0&pktsiz
020005 014031 7      add [hadr]
020006 014000 7      add 0             ;and point to it
020007 010167 7      sta thisb
020010 104167 7      lda thisb i       ;fix checksum
020011 054010 7      add chkh x
020012 016052 7      sub one           ;incl length
020013 050010 7      sta chkh x
020014 044013 7      lda pkth x        ;get hacmem
020015 172167 7      ldx thisb i       ;get haccom
020016 121542 7      jst [tskhac] i     ;do hac check and get lochst

020017 103543 7      jmp [fqok] i       ;host access not allowed
020020 072176 7      ldx lochst         ;check host state
020021 144276 7      lda hosti ix
020022 012052 7      era [hstup]
020023 100040 7      sze               ;is it up?
020024 103543 7      jmp [fqok] i       ;no, discard packet
020025 010167 7      sta thisb
020026 004053 7      lda two            ;room for one raw packet?
020027 001001 7      .inh fre
020030 120117 7 0     jst maxchi i
020031 103543 7 0     jmp [fqok] i
020032 024220 7 0 t2h0: irs nrea        ;no
020033 004166 7 0     lda this          ;take it
020034 073544 7 0 t2h1: ldx [reastb]    ;fake reassembly block
020035 050000 7 0     sta rch x
020036 004014 7 0     lda minus1
020037 050100 7 0     sta rct x
020040 133545 7 0     stx [ready] i     ;put on host queue
```

```

020041 173545 7 0 t2h:    ldx [ready] i      ;load the reassembly pntr
020042 140040 7 0          cra
020043 011153 7 0          sta t2hsiz
020044 044000 7 0          lda rch x
020045 011151 7 0          sta readye
020046 140040 7 0          cra
020047 066100 7 0          ima rct x
020050 011152 7 0          sta t2hcnt
020051 140040 7 0          cra
020052 111545 7 0          sta [ready] i      ;initialze
020053 050040 7 0          sta rms x
020054 066000 7 0          ima rch x
020055 000401 7 0          .enb tsk
020056 010000 7 0          t2h11:   sta 0      ;ordered packet chain
020057 044004 7 0          lda wrdc x
020060 141140 7 0          icl 0&pktsiz
020061 016055 7 0          sub [hdrl]
020062 041474 7 0          lgl 4      ;free reassembl block
020063 015153 7 0          add t2hsiz
020064 011153 7 0          sta t2hsiz
020065 004152 7 0          lda times
020066 016076 7 0          sub m30sec
020067 066005 7 0          ima inch x
020070 120150 7 0          jst trace.task i
020071 105545 7 0          lda [ready] i      ;time in slow ticks
020072 066000 7 0          ima ptrc x
020073 133545 7 0          stx [ready] i      ;give it 30 secs to live
020074 025152 7 0          irs t2hcnt
020075 003056 7 0          jmp t2h11
020076 073151 7 0          ldx readye
020077 005153 7 0          lda t2hsiz
020100 140407 7 0          tca
020101 011153 7 0          sta t2hsiz
020102 044004 7 0          lda wrdc x
020103 141140 7 0          icl 0&pktsiz
020104 014012 7 0          add [hdrl-1]
020105 014000 7 0          add 0      ;new ordered chain head
020106 010000 7 0          sta 0      ;ptr to last pkt
020107 044000 7 0          lda 0 x
020110 101040 7 0          snz
020111 003117 7 0          jmp t2h13
020112 025153 7 0          t2h12:   irs t2hsiz
020113 100100 7 0          slz
020114 003117 7 0          jmp t2h13
020115 040477 7 0          lgr 1
020116 003112 7 0          jmp t2h12

```

```
020117 173545 7 t2h13: idx [ready] i ;get first pkt ptr
020120 005153 7 lda t2hsiz ;and save message size
020121 140407 7 tca
020122 066006 7 ima neth x ;in neth word
020123 056006 7 sub neth x
020124 054010 7 add chkh x ;and fix checksum
020125 050010 7 sta chkh x
020126 044007 7 lda typh x
020127 006066 7 ana [priorty]
020130 100040 7 sze ;pick pri or reg queue
020131 005546 7 lda [ehpq-ehq]
020132 014176 7 add lochst ;add offset generated in host
020133 010000 7 sta 0 ;put in x
020134 001001 7 .inh (i2h,h2i,t.o,all)
020135 145547 7 0 lda [ehq] ix 0&ptrc
020136 010001 7 0 sta ireg
020137 105545 7 0 lda [ready] i
020140 110001 7 0 sta ireg i ;put on host queue
020141 005151 7 0 lda readye
020142 151547 7 0 sta [ehq] ix
020143 021517 7 0 jst task2h ;poke host if necessary
020144 004167 7 0 lda thisb
020145 101040 7 0 snz ;raw pkt?
020146 103550 7 0 jmp [goodm] i ;yes, we're done, no messno s
020147 072167 7 0 idx thisb ;go back for more
020150 103551 7 0 jmp [upmess] i

        .lev var
V  readye: .block 1 ;end pnter for ready
V  t2hcmt: .block 1 ;remaining pkt counter
V  t2hsiz: .block 1 ;message size (in bits)
```

```
;send incomplete reply if inc mesnum=rmess
;give back reas blk if inc mess1 or mess8, inc gvb
    .lev tsk
    .lck all
020154 072172 7 0 increp: ldx source
020155 004173 7 0           lda seqnum
                            ;defhlt(/50. sending inc reply to imp=x, messno=a)
020156 120120 7 0           jst hltjst i
020157 072166 7 0           ldx this
020160 044013 7 0           lda pkth x
020161 100400 7 0           spl 0&mltpkt      ;multi-pkt?
020162 003171 7 0           jmp incrpl      ;yes
020163 004031 7 0           lda [6]          ;single, set mess code
020164 127552 7 0           ima [messid] i   ;and get state
020165 012053 7 0           era two         ;all1 sent?
020166 101040 7 0           snz
020167 103553 7 0           jmp [gudgvb] i   ;yes, flush reassembly
020170 103554 7 0           jmp [gudral] i

020171 004013 7 0 incrpl: lda seven        ;set mess code
020172 111552 7 0           sta [messid] i
020173 103553 7 0           jmp [gudgvb] i   ;flush reassembly

    .lev tsk
    .lck all
;send back out-of-range
020174 021254 7 0 outcfr: jst swroom      ;room for reply?
020175 004040 7 0           lda [nettyp+netcpt+prirty]
020176 011253 7 0           sta swptyp
020177 004026 7 0           lda [ocrcod]
020200 111552 7 0           sta [messid] i
020201 004173 7 0           lda seqnum
020202 072172 7 0           ldx source
                            ;defhlt(/13. sending out of range to imp=x, messno=a)
020203 021204 7 0           jst hltrep
```

```
.lev tsk
.lck all
020204 000000 7 0 hltrep: 0
020205 120126 7 0         jst hltncc i
020206 105552 7 0         lda [messid] i      ;get pkt code
                           swprep:
020207 000401 7 0         .enb tsk
020210 072166 7           ldx this
020211 052015 7           era midh x        ;get foreign use no
020212 007555 7           ana [177777?usenum]
020213 052015 7           era midh x
020214 066013 7           ima pkth x        ;set pkth, get local use no
020215 006074 7           ana [usenum]
020216 052012 7           era seqh x        ;and local block no
020217 141044 7           car 0&blknum
020220 052012 7           era seqh x
020221 066015 7           ima midh x        ;set midh, get foreign block
020222 052012 7           era seqh x
020223 141050 7           cal 0&mesnum
020224 052012 7           era seqh x
020225 050012 7           sta seqh x
020226 044011 7           lda srch x      ;swap source and destination

020227 066014 7           ima dsth x
020230 050011 7           sta srch x
020231 140040 7           cra
020232 050006 7           sta neth x
020233 005253 7           lda swptyp
020234 050007 7           sta typf x
020235 044004 7           lda wrdc x        ;set standard packet size
020236 006013 7           ana [usecnt] 0&pktsiz&trcpkt&trcrct
020237 013556 7           era [(hdrl+2)<<8.]
020240 050004 7           sta wrdc x
020241 020002 7           jst cksum        ;checksum
020242 000000 7           %crash       ;buffer word count smashed - t
020243 140407 7           tca
020244 054010 7           add chkh x
020245 050010 7           sta chkh x        ;true checksum
020246 105557 7           lda [erpq] i
020247 010001 7           sta ireg
020250 132001 7           stx ireg i 0&ptrc ;put on
020251 133557 7           stx [erpq] i      ;reply queue
020252 103550 7           jmp [goodm] i      ;give good return

.lev var
020253 V swptyp: .block 1          ;typf for swprep
```

```

        .lev tsk
        .lck all
020254 000000 7 0 swroom: 0 ;check if room
020255 140040 7 0         cra ;for task reply
020256 120117 7 0         jst maxchi i
020257 103543 7 0         jmp [fqok] i ;none, quit & ignore
020260 024220 7 0         irs nrea ;count in reassembly
020261 103254 7 0         jmp swroom i

        .lev tsk
        .lck all
020262 044013 7 0 repgot: lda pkth x ;get type of reply
020263 006011 7 0         ana three 0&pktcod
020264 111552 7 0         sta [messid] i
020265 044012 7 0         lda seqh x ;find transaction block
020266 121560 7 0         jst [tsbget] i
020267 003312 7 0         jmp repmis
020270 033531 7 0         stx tsbptr ;not there, error
020271 044110 7 0         lda lcd x ;save tsb pointer
020272 141140 7 0         icl ;get tsb type
020273 006013 7 0         ana [7]
020274 022012 7 0         cas [tsbgvb/400] ;is it giveback?
020275 003306 7 0         jmp repbad ;(6,7)no, illegal
020276 003466 7 0         jmp repgvb ;(5)yes
020277 010000 7 0         sta 0 ;(0-4)no, other
020300 105552 7 0         lda [messid] i ;and reply type
020301 043302 7 0         jmp repdis x ;dispatch

020302 003363 7 0 repdis: jmp repms1 0&tsbms1 ;0- mes1 sent
020303 003407 7 0         jmp repms8 0&tsbms8 ;1- mes8 sent
020304 003316 7 0         jmp reprq1 0&tsbrq1 ;2- req1 sent
020305 003367 7 0         jmp reprq8 0&tsbrq8 ;3- req8 sent
020306 072172 7 0 repbad: idx source ;4- illeg
020307 004173 7 0         lda seqnum
                                ;defhlt(/67. bad transaction block for imp=x, messno=
020310 120120 7 0         jst hltjst i
020311 003466 7 0         jmp repgvb

020312 072172 7 0 repmis: idx source
020313 004173 7 0         lda seqnum
                                ;defhlt(/68. missing transaction block for imp=x, mes
020314 120120 7 0         jst hltjst i
020315 003471 7 0         jmp repdon

```

```

020316 073531 7 0 reprq1: idx tsbptr
020317 012052 7 0 era one
020320 101040 7 0 snz
020321 003332 7 0 jmp repal1
020322 140040 7 0 cra
020323 066264 7 0 ima lpk x
020324 101040 7 0 snz
020325 003412 7 0 jmp sndrfn
020326 010000 7 0 sta 0
020327 120115 7 0 jst flushi i
020330 024224 7 0 irs nres
                                         ;and give back to reassembly

020331 003412 7 0 jmp sndrfn
                                         ;and send rfnm

020332 140040 7 0 repal1: cra
020333 066264 7 0 ima lpk x
020334 010000 7 0 sta 0
020335 044004 7 0 lda wrdc x
020336 101100 7 0 sln 0&usecnt
020337 003357 7 0 jmp repal2
020340 044013 7 0 lda pkth x
020341 016052 7 0 sub one 0&msgcod&reqcod
020342 050013 7 0 sta pkth x
020343 044010 7 0 lda chkh x
020344 014052 7 0 add one
020345 050010 7 0 sta chkh x
020346 105557 7 0 lda [erpq] i
020347 010001 7 0 sta ireg
020350 132001 7 0 stx ireg i 0&ptrc
020351 133557 7 0 stx [erpq] i
020352 073531 7 0 idx tsbptr
020353 044110 7 0 lda lcd x
020354 012063 7 0 era [tsbms1?tsbrq1]
020355 050110 7 0 sta lcd x
020356 103543 7 0 jmp [fqok] i
                                         ;make tsb code mes1 sent

020357 004000 7 0 repal2: lda 0
020360 073531 7 0 idx tsbptr
020361 050264 7 0 sta lpk x
020362 103561 7 0 jmp [fqng] i
                                         ;and reject for now

020363 012052 7 0 repms1: era one
020364 101040 7 0 snz
020365 103543 7 0 jmp [fqok] i
020366 003412 7 0 jmp sndrfn
                                         ;was reply all1?
                                         ;yes, duplicate, reject
                                         ;no, send rfnm

020367 012052 7 0 reprq8: era one
020370 101040 7 0 snz
020371 021373 7 0 jst gotall
020372 003466 7 0 jmp repgvb
                                         ;was reply all8?
                                         ;yes, count allocate
                                         ;no

```

```
020373 000000 7 0 gotall: 0
020374 072167 7 0 ldx thisb ;increment allocate count
020375 044340 7 0 lda mbmes x 0&mballc
020376 014064 7 0 add [mball1]
020377 050340 7 0 sta mbmes x 0&mballc
020400 072176 7 0 ldx lochst
020401 144256 7 0 lda hiali ix
020402 101040 7 0 snz ;timer running? if so, leave
020403 005562 7 0 lda [173000] ;otherwise set for 10. fast t
020404 141206 7 0 aoa ;increment allocate count
020405 150256 7 0 sta hiali ix
020406 103373 7 0 jmp gotall i

020407 012052 7 0 repms8: era one ;was reply all8?
020410 101040 7 0 snz
020411 021373 7 0 jst gotall ;yes, count allocate
020412 073531 7 0 sndrfn: ldx tsbptr
020413 105552 7 0 lda [messid] i ;what type reply?
020414 022053 7 0 cas two
020415 003436 7 0 jmp sndinc 0&inccod ;incomplete
020416 003453 7 0 jmp sndded 0&dedcod ;dead
020417 005563 7 0 lda [crfnm] 0&rfncod&allcod ;normal
020420 050044 7 0 sndrf1: sta lms x ;set msg type
020421 000010 7 0 %rdclock
020422 056000 7 0 sub lch x
020423 121450 7 0 jst stats.hstimi i ;**stats
020424 140040 7 0 cra ;clear chain ptr word
020425 111531 7 0 sta tsbptr i 0&lch
020426 005531 7 0 lda tsbptr ;put tsb block
020427 072176 7 0 ldx lochst ;on host reply queue
020430 167564 7 0 ima [ehrq] ix
020431 010001 7 0 sta ireg
020432 005531 7 0 lda tsbptr
020433 110001 7 0 sta ireg i ;poke host
020434 021517 7 0 jst task2h
020435 003471 7 0 jmp repdon

020436 044220 7 0 sndinc: lda lid x ;was this marked as an error
020437 006072 7 0 ana [subtyp]
020440 012054 7 0 era [cerror]
020441 100040 7 0 sze
020442 003451 7 0 jmp sndin1 ;no
020443 044220 7 0 lda lid x ;yes, different ih message t
020444 012054 7 0 era [cerror] ;sub type=0
020445 050220 7 0 sta lid x
020446 004065 7 0 lda [cerrdt] ;type=error in data
020447 003420 7 0 jmp sndrf1

stats.hstimi:
020450 016265 7 0 jsttsk

020451 005565 7 0 sndin1: lda [cinctr] ;type=incomplete
020452 003420 7 0 jmp sndrf1
```

020453 072166 7 0 sndded: ldx this ;get reason
020454 044016 7 0 lda data x
020455 073531 7 0 ldx tsbptr
020456 050264 7 0 sta lpk x ;save it in tsb
020457 064220 7 0 irs lid x 0&chstd ;subtype=host dead
020460 104175 7 0 lda mestb2 i ;put in stopping mode (init,s
020461 007566 7 0 ana [177777?mbrst?mbinit?mbstp]
020462 013567 7 0 era [mbinit+mbstp] ;let outstanding mesgs go
020463 110175 7 0 sta mestb2 i ;and then reset
020464 004041 7 0 lda [cdestd] ;type=dead
020465 003420 7 0 jmp sndrf1

020466 073531 7 0 repgvb: ldx tsbptr ;free up transaction block
020467 140040 7 0 cra
020470 050044 7 0 sta lms x
020471 104174 7 0 repdon: lda mestb1 i ;turn on got.it bit in tmess

020472 012173 7 0 era seqnum
020473 141044 7 0 car
020474 012173 7 0 era seqnum
020475 016173 7 0 sub seqnum
020476 141140 7 0 icl
020477 140407 7 0 tca
020500 010000 7 0 sta 0
020501 044062 7 0 lda bittab+8. x
020502 112175 7 0 era mestb2 i
020503 007570 7 0 ana [177777?mbinct] ;turn off timer
020504 012034 7 0 era [mbinct]
020505 110175 7 0 sta mestb2 i
020506 072176 7 0 ldx lochst
020507 140040 7 0 cra
020510 166250 7 0 ima hirwi ix
020511 101040 7 0 snz ;is host hung on some resourc
020512 103543 7 0 jmp [fqok] i ;is HIRW set?
020513 144144 7 0 lda h2ipci ix ;yes, get host pointer
020514 010000 7 0 sta 0
020515 000043 7 0 gpr ;poke it
020516 103543 7 0 jmp [fqok] i ;all done

.lev tsk
020517 000000 7 task2h: 0 ;got something for host output
020520 072176 7 ldx lochst
020521 144300 7 lda ihspi ix
020522 100040 7 sze ;is host idle?
020523 103517 7 jmp task2h i
020524 144143 7 lda i2hpc i ;get ptr to PCB
020525 010000 7 sta 0 ;poke it
020526 000043 7 gpr
020527 072176 7 ldx lochst ;return
020530 103517 7 jmp task2h i

.lev var
020531 V tsbptr: .block 1 ;pointer to our transaction b

```
.lev tsk
.lck all
020532 072167 7 0 oorgot: ldx thisb ;got an out of range
020533 044340 7 0 lda mbmes x ;save allocate count
020534 006042 7 0 ana [mbstp+mballc]
020535 013571 7 0 era [mbrst+mesbts] 0&mbinit&mbinct
020536 050340 7 0 sta mbmes x ;turn on rst, no messages ou
020537 004173 7 0 lda seqnum
020540 072172 7 0 ldx source
;defhlt(/42. got out of range from imp=x, messno=a)
020541 121572 7 0 jst [thltok] i
```

.stl task: message block routines

```

;search for local block routine
;runs under software lock **blkslc** in order to lock
;out h2i if tsk is using it
;called with handling type,imp no. in a,
;t/r host pair in blkslc,
;and second trn/rcv block in x.
;returns: locked, blkslc reset to sign.
    ;r1 ==> no block match, no block free
    ;r2 ==> block match, ptr in x, a destroyed
    ;r3 ==> no block match, first free block grabbed, with
    ;      imp no & hosts in mbimp and mbhst, use no
    ;      incremented and age cleared. block
    ;      ptr in x, a destroyed.

.section pg21
.lev (h2i,tsk)
blksrc: 0
    sta blksip          ;save imp #
    ana [177777?mbhand]
    ima blksip          ;and handling type
    era blksip
    sta blkshd
    lda minus1           ;start free count
    sta blksfr
    lda [-nmb+1]          ;at -1
    sta blkcnt            ;search whole table except bl
    blksl1: lda mbimp x
    cas blksip           ;get next block impno
    ;is it ours?
    jmp blksl2           ;>, no, and cannot be free
    jmp blksth           ;=, check hosts
    snz                  ;<, might be free
    irs blksfr           ;yes, bump counter
    skp                  ;not first free
    stx blksx2            ;save first free found
    blksl2: irs 0          ;loop over whole table
    irs blkcnt
    jmp blksl1
    lda blksfr           ;match not found, get #free-1
    spl                  ;any free?
    jmp blksl4           ;no, r1
    lda 0                 ;yes, decrement free count
    idx [b2fcnt]          ;assume receive block
    cas [tmbblk+nmb]      ;is it transmit?
    nop
    ldx [b1fcnt]           ;yes, different count
    lda 0 x               ;do not drop below 0
    sze
    sub one
    sta 0 x               ;get its ptr
    ldx blksx2
    lda blkslc             ;set up hosts
    sta mbhst x
    lda mbtim x            ;goose use no
    add [mbuse1]
    ana [mbusno] 0&mbage
    snz                  ;use #=0 not allowed
    add [mbuse1]
    sta mbtim x 0&mesnum&mbage ;and clr ages, mess no
    sti

```

```

021052 005077 4      lda blkshd           ;set handling type
021053 050160 4      sta mbfor x
021054 005076 4      lda blksip           ;set foreign imp
021055 050000 4      sta mbimp x
021056 025000 4      irs blksrc
021057 025000 4      blksl3: irs blksrc
021060 004071 4      blksl4: lda sign
021061 001001 4      lnh all
021062 011103 4 0    sta blkslc
021063 103000 4 0    jmp blksrc i      ;and return

        .lev (h2i,tsk)
021064 005103 4      blksth: lda blkslc
021065 052070 4      era mbhst x
021066 100040 4      sze
021067 003022 4      jmp blksl2
021070 005077 4      lda blkshd
021071 052160 4      era mbfor x
021072 006075 4      ana [mbhand]
021073 100040 4      sze
021074 003022 4      jmp blksl2
021075 003057 4      jmp blksl3      ;no, try some more
                                         ;yes, get desired handling ty
                                         ;match?

        .lev var
021076      V     blksip: .block 1      ;imp number
021077      V     blkshd: .block 1      ;handling type
021100      V     blkcnt: .block 1      ;search count
021101      V     blksx2: .block 1      ;first free block ptr
021102      V     blksfr: .block 1      ;# free blocks - 1

021103      V     blkslc: .block 1      ;software lock, sign=free
                                         ; host pair ==> used

;got a get-a-block message
        .lev tsk
        .lck all
021104 121336 7 0  getblk: jst [swroom] i  ;room for reply?
021105 000401 7 0  .enb tsk
                                         ;yes

021106 044017 7      lda data+1 x
021107 010001 7      sta ireg
021110 044016 7      lda data x
021111 072001 7      ldx ireg
021112 121337 7      jst [tskhac] i
021113 003153 7      jmp getblk
021114 072176 7      ldx lochst
021115 121340 7      jst [hstatr] i
021116 003156 7      jmp getblk
021117 072166 7      ldx this
021120 044012 7      lda seqh x
021121 141340 7      ica

021122 011103 7      sta blkslc      ;locks blksrc routine
021123 044015 7      lda midh x
021124 006075 7      ana [hndtyp]
021125 012172 7      era source
021126 073341 7      ldx [rmblk+1]
021127 021000 7      jst blksrc      ;get handling type
                                         ;and imp no
                                         ;first usable receive block
                                         ;find receive block

        .lck all
021130 003164 7 0    jmp getblk      ;no match, no free blocks

```

021131 003166 7 0

jmp getblm

;match, might be duplicate

021132 032167 7 0	stx thisb	;got a free block
021133 140040 7 0	cra	
021134 050340 7 0	sta mbsta x	;init rstate and rtype
021135 050430 7 0	sta mbtyp x	;to be idle
021136 021254 7 0	jst blknew	;init foreign block, use numb
021137 044250 7 0	getbll: lda mbtim x	;get local use no
021140 006073 7 0	ana [mbusno]	
021141 041474 7 0	lgl 4 0&usenum	
021142 072166 7 0	ldx this	
021143 050013 7 0	sta pkth x	;will be swapped to midh
021144 004167 7 0	lda thisb	;get local block no
021145 017342 7 0	sub [rmblk]	
021146 050012 7 0	sta seqh x	;will be swapped into midh
021147 004040 7 0	lda [nettyp+netcpt+prirty]	
021150 111343 7 0	sta [swptyp] i	;set up typ h

```
.lev tsk
021151 004027 7 getblr: lda [gtrcod] ;send getblk reply
021152 103344 7 jmp [swprep] i

021153 005345 7 getblh: lda [nettyp+netcpt+prirty+hacrej+blkrej]
                           ;host access reject
021154 111343 7 getbln: sta [swptyp] i ;send back got-no-block
021155 003151 7 jmp getblr

021156 072166 7 getbld: ldx this ;save host status
021157 050016 7 sta data x
021160 005346 7 lda [nettyp+netcpt+prirty+blkrej] ;host dead reject
021161 003154 7 jmp getbln

021162 005347 7 getblf: lda [nettyp+netcpt+prirty+hacrej] ;host format reject
021163 003154 7 jmp getbln

.lev tsk
.lck all
021164 024224 7 0 getble: irs nres ;no task reply to be sent
021165 103350 7 0 jmp [fqok] i ;ignore get-a-block

021166 032167 7 0 getblm: stx thisb
021167 044160 7 0 lda mbfor x ;check for duplicate
021170 072166 7 0 ldx this
021171 052015 7 0 era midh x ;with same hand,use,block
021172 072167 7 0 ldx thisb
021173 101040 7 0 snz
021174 003137 7 0 jmp getbll ;duplicate, send got-a-block

021175 024224 7 0 irs nres ;not a dup, no task reply to
021176 044250 7 0 lda mbtim x ;age block to max
021177 006032 7 0 ana [177777?mbage]
021200 012072 7 0 era [mbage]
021201 050250 7 0 sta mbtim x
021202 004172 7 0 lda source
                           ;defhlt(/14. rcvd non-dup getblk from imp=a for activ
021203 121351 7 0 jst [thltok] i
```

```
;got a reset message
    .lev tsk
    .lck all
021204 121336 7 0 rstmsg: jst [swroom] i           ;room for reply?
021205 004170 7 0 lda blkflg                      ;yes
021206 100040 7 0 sze                            ;block check ok?
021207 003212 7 0 jmp blkrs1                     ;no, just send reset reply
021210 021315 7 0 jst blkrss                     ;do reset-r
021211 024352 7 0 resetr
    .ret tsk
021212 004040 7 0 blkrs1: lda [nettyp+netcpt+prirty]
021213 111343 7 0 sta [swtyp] i                  ;set typ
021214 004072 7 0 lda [rsrcod]                   ;send reset reply
021215 103344 7 0 jmp [swprep] i

;got a got-a-block message
    .lev tsk
    .lck all
021216 004170 7 0 gotblk: lda blkflg
021217 100040 7 0 sze                           ;duplicate?
021220 103350 7 0 jmp [fqok] i                  ;yes
021221 072167 7 0 ldx thisb
021222 044340 7 0 lda mbmes x
021223 006035 7 0 ana [mbrst+mbinit+mbstp] ;got-a-block expected?
021224 012070 7 0 era [mbinit]
021225 100040 7 0 sze
021226 103350 7 0 jmp [fqok] i                  ;no, duplicate
021227 072166 7 0 ldx this
021230 044007 7 0 lda typh x
021231 006034 7 0 ana [hacrej+blkrej] 0&cnhac&chstd
021232 100040 7 0 sze                           ;pos or neg reply?
021233 003240 7 0 jmp gotblk                    ;negative, got-no-block
021234 021254 7 0 jst blknew                     ;initialize foreign stuff
021235 005352 7 0 lda [mbinct+mesbts] 0&mbrst&mbinit&mbstp&mballc
021236 050340 7 0 sta mbmes x                  ;clear init, initialize inc t
                                         ; allocates, and messages
021237 103350 7 0 jmp [fqok] i

021240 013353 7 0 gotblk: era [mbrst+mbinit+mesbts] 0&mbstp&mbinct
021241 110175 7 0 sta mestb2 i                 ;turn on rst+init, save subty
021242 044016 7 0 lda data x                   ;get host status
021243 141044 7 0 car                          ;save left half
021244 112174 7 0 era mestb1 i 0&mesnum     ;in tmess
021245 110174 7 0 sta mestb1 i
021246 044016 7 0 lda data x
021247 141050 7 0 cal                          ;and right half
021250 072167 7 0 ldx thisb
021251 052160 7 0 era mbfor x 0&mbfrnb      ;in blknum
021252 050160 7 0 sta mbfor x
021253 103350 7 0 jmp [fqok] i

    .lev tsk
    .lck all
021254 000000 7 0 blknew: 0                   ;init blk with foreign block
021255 072166 7 0 ldx this
021256 044015 7 0 lda midh x
021257 072167 7 0 ldx thisb
021260 006045 7 0 ana [usenum+blknum] 0&mbfrnu&mbfrnb
021261 052160 7 0 era mbfor x 0&mbhand
```

021262 050160 7 0 sta mbfor x
021263 103254 7 0 jmp blknew i

```

;got a reset request message
    .lev tsk
    .lck all
021264 004170 7 0 rstreq: lda blkflg           ;duplicate?
021265 100040 7 0 sze                           ;yes, send automatic reset
021266 003302 7 0 jmp rstrel                      ;no, reset this block
021267 072167 7 0 ldx thisb
021270 044340 7 0 lda mbmes x                   ;any messages outstanding?
021271 006007 7 0 ana [mesbts]
021272 012007 7 0 era [mesbts]
021273 100040 7 0 sze
021274 103350 7 0 jmp [fqok] i                 ;yes, can't reset now
021275 044340 7 0 lda mbmes x
021276 140500 7 0 ssm 0&mbrst                  ;reset bit
021277 007354 7 0 ana [177777?mbinit?mbinct] ;and timer to max
021300 050340 7 0 sta mbmes x
021301 103350 7 0 jmp [fqok] i

021302 121336 7 0 rstrel: jst [swroom] i       ;room for reset message?
021303 004040 7 0 lda [nettyp+netcpt+prirty]
021304 111343 7 0 sta [swptyp] i               ;set typ
021305 004025 7 0 lda [rstcod]                  ;return a reset from here
021306 103344 7 0 jmp [swprep] i

;got a reset reply message
    .lev tsk
    .lck all
021307 004170 7 0 rstrep: lda blkflg           ;duplicate?
021310 100040 7 0 sze
021311 103350 7 0 jmp [fqok] i                 ;yes, ignore
021312 021315 7 0 jst blkrss                  ;no, do reset-t
021313 024162 7 0 resett
    .ret tsk
021314 103350 7 0 jmp [fqok] i

    .lev tsk
    .lck all
021315 000000 7 0 blkrss: 0                   ;perform resets in task
021316 024160 7 0 irs tsklck                 ;keep timeout from calling re
021317 105315 7 0 lda blkrss i               ;get function (t or r)
021320 011331 7 0 sta blkrst
021321 004172 7 0 lda source
021322 072167 7 0 ldx thisb
021323 121331 7 0 jst blkrst i              ;im# # argument
021324 000401 7 0 .enb tsk
021325 140040 7 0 cra
021326 010160 7 0 sta tsklck
021327 025315 7 0 irs blkrss
021330 103315 7 0 jmp blkrss i             ;block # argument
                                                ;let timeout in again
                                                ;skip subr arg

021331     .lev var
V      blkrst: .block 1                     ;reset routine pointer

```

```
000306 041457 V .section pg0
000307 041510 V tskfli: tskflg
000307 041510 V hdowni: hdown

041457 V .section heap
041510 V tskflg: .block th+bh
041510 V hdown: .block nh
```

.sttl LINE UP/DOWN
.INCLUDE zud.m4

;NMFS notes:

```
022000      .section pg22
V   zpntr: .block 1          ;pointer for k-of-n counters
022001      V   zlnc:  .block 1    ;line number zded1 is doing
022002      V   zrflag: .block 1  ;copied zrcv
022003      V   thd:   .block 1   ;channel to min number imp
022004      V   ztemp:  .block 1  ;temporary for clupln
022005      V   mismatch: .block 1;set to modem number of line

;bits in typh word of protocol packet (zrcv)
     .lev und
000001      zhliho=1 0&typh   ;1 ==> hello, 0 ==> ihy
000004      zhliup=4           ;1 ==> line up
001000      zhltyp=1000        ;subtype for protocol packet
```

```
;timecut logic for line up/down
.lev t.o
022006 000000 5 zdedl: 0 ;called for each imp line
022007 032204 5 stx zmb1 ;save modem block pointer
022010 044025 5 lda pcb.num x ;get modem number
022011 011001 5 sta zlno
022012 140040 5 cra
022013 066156 5 ima zrcv x ;clear and copy receive bits

022014 011002 5 sta zrflag
022015 041477 5 lgl 1 0&swtyp&pactyp ;type bit into carry bit
022016 044117 5 lda line x ;get line state
022017 100400 5 spl ;silent?
022020 003135 5 jmp zdslnt ;yes
022021 101040 5 snz ;up?
022022 003053 5 jmp zdup ;yes
022023 022011 5 cas [3] ;waiting state?
022024 003045 5 jmp zdwait ;yes
022025 101000 5 nop ;no, state=3
022026 100001 5 src ;anything received?
022027 003172 5 jmp zddwn2 ;yes, (state=down or coming up)
022030 084174 5 irs zzcmup x ;count ticks to come up
022031 101000 5 nop
022032 044117 5 zdcmup: lda line x ;has line started up yet?
022033 100100 5 slz
022034 003071 5 jmp zdedx1 ;no, since nothing in just ex
022035 064162 5 zdcmng: irs zkcnt x ;slave up or cmup, master cmu
022036 003151 5 jmp zddwn1 ;have not missed k in a row

zslvup:
022037 001001 5 .inh (m2i,i2m)
022040 044117 5 1 lda line x ;was line up?
022041 101040 5 1 snz ;yes -- kill it now (slave on
022042 003116 5 1 jmp zdedkl ;no, reset to silent state
022043 021261 5 1 jst zkill ;will exit after enable
022044 003125 5 1 jmp zdkill
```

```
.lev t.o
022045 064161 5 zdwait: irs zncnt x ;waiting for flooding delay
022046 003053 5 jmp zdup ;wait some more
022047 001001 5 .inh m2i
;waiting over, bring line up
022050 121703 5 1 jst [lnupst] i
022051 000401 5 1 .enb t.o
022052 072204 5 ldx zmbl
022053 044175 5 zdup: lda zmastr x ;master?
022054 101040 5 snz ^
022055 003177 5 jmp zdslup ;no
022056 021300 5 jst zknset ;set pointer for ticking k-of
022057 145000 5 zdtic1: lda zpntr ix ;count out any of the counter
022060 100040 5 sze ;not yet up to zero
022061 165000 5 irs zpntr ix
022062 101000 5 nop ;cycle over k-1 counters
022063 024000 5 irs 0
022064 003057 5 jmp zdtic1 ;was ihy received?
022065 005002 5 lda zrflag ;(hello's filtered out at m2i
022066 101040 5 snz ;no, setup for counting ano
022067 003100 5 jmp zdmiss ;restore line number
022070 072204 5 zdedlx: ldx zmbl ;tell i2m to send
022071 064160 5 zdedx1: irs zsend x ;cnt one sent (slave cnts tic
022072 064200 5 zdedlc: irs rtssnt x
022073 101000 5 nop ;exit
022074 103006 5 jmp zdedl i ;reset k consecutive counter
022075 044172 5 zdedsl: lda zk x ;and exit
022076 050162 5 sta zkcnt x ;set pointer for k-of-n count
022077 003072 5 jmp zdedlc ;zrflag used as temp
022100 021300 5 zdmiss: jst zknset ;look for an empty counter
022101 004071 5 lda sign ;found one?
022102 011002 5 sta zrflag ;yes, go set it
022103 145000 5 zdmis1: lda zpntr ix ;no, record largest in case n
022104 101040 5 snz ;this larger than previous
022105 003127 5 jmp zdtic2 ;cycle over k-1 counters
022106 023002 5 cas zrflag ;get largest counter
022107 011002 5 sta zrflag
022110 101000 5 nop
022111 024000 5 irs 0
022112 003103 5 jmp zdmis1 ;get neighbor number for trap
022113 005002 5 lda zrflag
022114 072204 5 ldx zmbl
022115 001001 5 .inh all
022116 010001 5 0 zdedkl: sta ireg ;restore a
022117 044116 5 0 lda neighb x ;defhlrt/72. line killed in zdedl, x=neighbor number)
022120 010000 5 0 sta 0 ;give info trap
022121 004001 5 0 lda ireg ;restore modem block pointer
022122 120120 5 0 jst hltjst i ;kill line
022123 072204 5 0 ldx zmbl
022124 120123 5 0 jst killii i
022125 000401 5 0 .enb t.o
022126 103006 5 0 jmp zdedl i ;exit
```

022127 033002 5 zdtic2: stx zrflag ;save index to counter
022130 072204 5 ldx zmbl ;get line number
022131 044173 5 lda zn x ;get proper n for this line
022132 073002 5 ldx zrflag ;restore pointer to counter
022133 151000 5 sta zpntr ix ;set it to -n
022134 003070 5 jmp zdedlx ;exit sending hello

022135 064174 5 zdsln1: irs zzcmup x ;count tics to come up
022136 101000 5 nop
022137 064162 5 irs zkcnt x ;silent period over?
022140 003071 5 jmp zdedx1 ;no, send hello and quit
022141 004052 5 lda [1]
022142 050117 5 sta line x ;yes, change to down state
022143 010154 5 sta ludflg ;set line up/down flg for bck
022144 044172 5 lda zk x ;reset consecutive k counter

022145 050162 5 sta zkcnt x ;do line clean up now
022146 121704 5 jst [clupln] i ;setup k-of-n pointer
022147 021300 5 jst zknset ;clear the k-1 counters
022150 021163 5 jst zdclr ;reset n consecutive counter
022151 005606 5 zddwn1: lda znup
022152 050161 5 sta zncnt x
022153 044206 5 lda rt pend x ;zero retransmission timers
022154 011000 5 sta zpntr
022155 073705 5 ldx [-nnodes/8.-1]
022156 021163 5 jst zdclr ;zero the counters
022157 044175 5 lda zmastr x ;master or slave?
022160 100040 5 sze
022161 003071 5 jmp zdedx1 ;master, send hello and exit
022162 003072 5 jmp zdedlc ;slave, just count and exit

022163 000000 5 zdclr: 0 ;line up/down subroutine
022164 140040 5 cra
022165 151000 5 sta zpntr ix
022166 024000 5 irs 0
022167 003165 5 jmp .-2
022170 072204 5 ldx zmbl
022171 103163 5 jmp zdclr i

022172 004053 5 zddwn2: lda [2]
022173 066117 5 ima line x
022174 100100 5 slz
022175 003234 5 jmp zdcmu0
022176 003204 5 jmp zdcmu1 ;mark line coming up
;when something received
;was line dead before?
;yes, fix parameters
;continue with coming up

022177 100001 5 zdslup: src
022200 003204 5 jmp zdcmu1
022201 064162 5 irs zkcnt x
022202 003072 5 jmp zdedlc
022203 003037 5 jmp zslvup
022204 064174 5 zdcmu1: irs zzcmup x
022205 101000 5 nop
022206 044175 5 lda zmastr x
022207 101040 5 snz
022210 003075 5 jmp zdeds1
022211 005002 5 lda zrflag
022212 100100 5 slz 08zhlihlo
022213 003035 5 jmp zdcmng
022214 044172 5 lda zk x
022215 050162 5 sta zkcnt x ;slave up, anything received?
;yes
;no, missed k in row?
;no
;yes, kill line start over
;count ticks to come up
;(superfluous if slave up)

022216 064161 5 irs zncnt x ;slave?
;yes, just count tick and exit
;no, master
;was the received packet an i
;no, tick k counter
;yes
;reset k consecutive counter
;count another n consecutive

022217 003071 5 jmp zdedx1
022220 004054 5 lda [4]
022221 050117 5 sta line x
022222 044176 5 lda lnclkp x
022223 017706 5 sub [clockf]
022224 010000 5 sta 0
022225 045600 5 lda wattic x
022226 072204 5 ldx zmb1
022227 017606 5 sub znup
022230 101400 5 smi
022231 004014 5 lda minus1
022232 050161 5 sta zncnt x
022233 003070 5 jmp zdedlx ;line not up yet, send hello
;line has come up, go into wait
;while spf updates flood across

;clock indeks for this line
;speed tells how long to wait
;ticks to equal one minute

;have already waited this many
;enough already?
;yes, wait one more

;come here when moving from down state to coming up. set up the
;k and n counters and the line clock. then, decide on the number
;of octets to use on this line (minimum of our config value and
;our neighbor's config value, which was sent in neth word of
;packet and is saved in zoct) and initialize i2mtab and octtab
;for that value. if we're using our neighbor's value, trap.

022234 121707 5 zdcmu0: jst [setnk] i ;yes, read config card for K/
022235 044157 5 lda zoct x ;neighbor's highest octet number
022236 062107 5 cas cfactet x ;compare to ours
022237 044107 5 lda cfactet x ;neighbor will lower to our n
022240 101000 5 nop ;we agree
022241 050106 5 sta maxoctet x ;if we have more than neighbor's
022242 141206 5 aoa
022243 041475 5 lgl 3
022244 050027 5 sta m2ipcb.nchan x ;fix nchan too
022245 121710 5 jst [octini] i ;fix size of i2mtab and octtab
022246 044107 5 lda cfactet x ;trap if we lowered our number
022247 056106 5 sub maxoctet x ;of channels
022250 101040 5 snz
022251 003204 5 jmp zdcmu1 ;using config number of channels
022252 001001 5 .inh m2i

022253 044027 5 1 lda m2ipcb.nchan x ;get new number of channels f
022254 073001 5 1 ldx zlno ;get modem number for trap

022255 120120 5 1 ;defhlt(83. lowering to A channels on line X)
022256 000401 5 1 jst hltjst i
022257 072204 5 .enb t.o
022260 003204 5 ldx zmb1
jmp zdcmu1 ;continue with coming up

```
.lev bck
    .lck (m2i,i2m,t.o,tsk)
;Enter and exit with x=modem block pointer
022261 000000 9 1 zkill: 0
022262 044172 9 1 lda zk x
022263 050162 9 1 sta zkcnt x
022264 050117 9 1 sta line x
022265 050175 9 1 sta zmstr x
022266 103261 9 1 jmp zkill i

.lev bck
    .lck (m2i,i2m,t.o,tsk)
;Enter and exit with x = modem block pointer
022267 000000 9 1 Killin: 0
022270 140040 9 1 cra
022271 050116 9 1 sta neighb x
022272 050174 9 1 sta zzcmup x
022273 121711 9 1 jst [delclr] i
022274 105712 9 1 lda [rtclks+2] i

022275 050207 9 1 sta rtclkp x
022276 021261 9 1 jst zkill
022277 103267 9 1 jmp killin i

.lev t.o
022300 000000 5 zknset: 0
022301 073001 5 ldx zlno
022302 145713 5 lda [mblock] ix
022303 015714 5 add [zkofn+7]
022304 011000 5 sta zpntr
022305 072204 5 ldx zmb1
022306 044172 5 lda zk x
022307 010000 5 sta 0
022310 024000 5 irs 0
022311 103300 5 jmp zknset i

;(z)kill a line
;set up silent period
;mark line reset
;mark us master

;subr to stop input and output
;clear tics to come up counter
;call subr to clear delay var
;set spf retr. clock pointer

;init protocol variables

;set pointer and x for k-of-n
;calculate end pointer for
;ticking k of n counters
;zpntr=modem block pointer+
;offset of zkofn words + 7

;x must be k-1
```

```
;line up/down - m2i logic
    .lev m2i
022312 072233 1     m2izpk: ldx m2isp           ;get pointer to packet
022313 105715 1     lda [thd] i               ;pick up chan to lowest no in
022314 012232 1     era mp                  ;compare with this line no
022315 100040 1     sze                      ;is it this line?
022316 003321 1     jmp m2izp1             ;no
022317 044012 1     lda seqh x              ;yes, so copy in his value fo
022320 010153 1     sta sync                 ;keeping global time througho
022321 044006 1     m2izp1: lda neth x       ;neighbor's number of octets

022322 010001 1     sta ireg                ;save for later
022323 044007 1     lda typh x              ;get flag bits
022324 072201 1     ldx mbl                 ;restore line number
022325 050156 1     sta zrcv x              ;save for t.o to process
022326 004001 1     lda ireg                ;save neth for t.o
022327 050157 1     sta zoct x              ;it has neighbor's highest oc
022330 044117 1     lda line x              ;line silent?
022331 100400 1     spl                     ;yes, quit now
022332 103716 1     jmp [m2ifr1] i          ;count hello/ihy received
022333 064177 1     irs rtrcvd x         ;check if this imp elsewhere
022334 101000 1     nop                     ;first, clear lnei entry for
022335 140040 1     cra                     ;save source imp number
022336 050120 1     sta lnei x              ;loop thru lnei table
022337 072233 1     ldx m2isp               ;is this an imp modem?
022340 044011 1     lda srch x              ;no
022341 010002 1     sta jreg                ;get source imp number
022342 073717 1     ldx [-ch]                ;a match?
022343 032001 1     m2izd1: stx ireg          ;yes, go check it out
022344 145720 1     lda [mblock+ch] ix      ;restore line number for irs
022345 101040 1     snz                     ;continue thru lnei table
022346 003355 1     jmp m2izd2             ;get source imp number in A
022347 010000 1     sta 0                   ;free of duplicates now
022350 004002 1     lda jreg                ;set and get neighbor number
022351 052120 1     era lnei x              ;don't kill line if it was d
022352 101040 1     snz                     ;else, check for neighbor cha
022353 003406 1     jmp m2izd3             ;kill line if neighbor change
022354 072001 1     m2izd4: ldx ireg          ;get imp number
022355 024000 1     m2izd2: irs 0            ;set last neighbor table
022356 003343 1     jmp m2izd1             ;are we master?
022357 004002 1     lda jreg                ;no... slave
022360 072201 1     ldx mbl                 ;yes and no... looped
022361 066116 1     ima neighb x           ;yes, mark master and hi end

022362 100040 1     sze                     ;is line up?
022363 052116 1     era neighb x           ;no, just exit
022364 100040 1     sze
022365 003446 1     jmp neikil             ;get line up?
022366 044116 1     lda neighb x           ;no, just exit
022367 050120 1     sta lnei x              ;yes and no... looped
022370 022106 1     cas mine               ;yes, mark master and hi end
022371 003421 1     jmp m2izsl             ;is line up?
022372 003443 1     jmp m2izlp             ;no, just exit
022373 004070 1     m2izms: lda [endbit]        ;get line up?
022374 050175 1     sta zmastr x           ;no, just exit
022375 044117 1     lda line x              ;yes and no... looped
022376 100040 1     sze
022377 103716 1     jmp [m2ifr1] i          ;yes, mark master and hi end
```

022400 044156 1 lda zrcv x ;check nice-stop kill line bi
022401 101100 1 sln ;(i.e. kill line if it's a ho
022402 003451 1 jmp m2imis ;check for octet mismatch on

022403 072232 1
022404 120120 1
022405 003435 1

;defhlt(/79. master kills line due to slave says hel
ldx mp ;get line number for trap
jst hltjst i
jmp m2izkl ;go kill line

022406 044117 1 m2izd3: lda line x ;is dup. neighbor dead?
022407 100040 1 sze
022410 003416 1 jmp m2ided
022411 044120 1 lda lnei x ;yes, clear its lnei entry
022412 022106 1 cas mine ;see if looped modem
022413 100000 1 skp ;no
022414 003354 1 jmp m2izd4 ;yes continue
022415 103716 1 jmp [m2ifr1] i ;no
022416 140040 1 m2ided: cra ;yes, clear its lnei entry
022417 066120 1 ima lnei x ;(and restore imp no. in a-rr
022420 003354 1 jmp m2izd4 ;keep searching lnei table

022421 140040 1 m2izsl: cra 0&endbit ;mark slave, low end of line
022422 050175 1 sta zmastr x

022423 064160 1 irs zsend x ;i2m should send ihy
022424 044156 1 lda zrcv x
022425 040475 1 lgr 3 C&zhloup ;received line state into c-b
022426 044117 1 lda line x ;our line state in a
022427 100040 1 sze
022430 003440 1 jmp m2izdn ;line was down
022431 100001 1 src ;was up, now reported up?
022432 003451 1 jmp m2imis ;yes, check for octet mismatch
022433 072232 1 ldx mp ;get line number for trap
022434 120120 1 ;defhlt(/71. slave kills line due to master down)
022435 072201 1 jst hltjst i
022436 120123 1 m2izkl: ldx mbl ;no!, kill it
022437 103716 1 jst killii i
022438 103716 1 jmp [m2ifr1] i

022440 100001 1 m2izdn: src ;was down, still reported down
022441 121703 1 jst [lnupst] i ;no, now bring it up
022442 103716 1 jmp [m2ifr1] i

022443 004071 1 m2izlp: lda [swttyp] 0&zhlhlo ;say we received an ihy
022444 050156 1 sta zrcv x ;and claim we are master
022445 003373 1 jmp m2izms

022446 072232 1 neikil: ldx mp ;get line number for trap
022447 120120 1 jst hltjst i
022450 003435 1 jmp m2izkl ;kill line

;come here when the line is up and the modem-in part of line up/down
;processing has been done. check for agreement between our neighbor's
;highest octet number and the highest octet number we're using. if
;they don't match, kill the line.
022451 044157 1 m2imis: lda zoct x ;neighbor's value
022452 052106 1 era maxoctet x ;our value
022453 101040 1 snz ;agree?
022454 103716 1 jmp [m2ifr1] i ;yes
022455 044027 1 lda m2ipcb.nchan x ;no, trap and kill line
022456 072232 1 ldx mp
;defhlt(/77. killing line X due to channel mismatch)

022457 120120 1 jst hltjst i
022460 003435 1 jmp m2izkl

```

        .sttl line up/down: line cleanup
        .lev t.o

022461 000000 5    clupln: 0                                ;line cleanup after silent pe
022462 001001 5      .inh (m2i,fre)
022463 005721 5 0    lda [smpq]
022464 021607 5 0    jst jsrts
022465 005722 5 0    lda [smq]
022466 021607 5 0    jst jsrts
022467 004204 5 0    lda zmb1
022470 015723 5 0    add [tsex]                            ;zero all tsex words
022471 010001 5 0    sta ireg
022472 072022 5 0    ldx [- (octmax+1)/2]            ;(octmax+1)/2 of them
022473 140040 5 0    cra
022474 110001 5 0    clup11:  sta ireg i
022475 024001 5 0    irs ireg
022476 024000 5 0    irs 0
022477 003474 5 0    jmp clup11
022500 004014 5 0    lda [1777777]
022501 072032 5 0    ldx [- (octmax+1)]
022502 110001 5 0    clup12:  sta ireg i
022503 024001 5 0    irs ireg
022504 024000 5 0    irs 0
022505 003502 5 0    jmp clup12
022506 072204 5 0    ldx zmb1
022507 140040 5 0    cra
022510 066105 5 0    ima chcounter x
022511 052105 5 0    era chcounter x
022512 100040 5 0    sze
022513 024217 5 0    irs nsfa
022514 044123 5 0    lda i2mtab x
022515 062112 5 0    nack1:   cas octend x
022516 101000 5 0    nop
022517 003526 5 0    jmp nack10
022520 011004 5 0    sta ztemp
022521 140040 5 0    cra
022522 111004 5 0    sta ztemp i
022523 005004 5 0    lda ztemp
022524 141206 5 0    aoa
022525 003515 5 0    jmp nack1
022526 044125 5 0    nack10:  lda retrqi x
022527 011004 5 0    sta ztemp
022530 000022 5 0    nack11:  deq
022531 003535 5 0    jmp nack12
022532 021543 5 0    jst rqsub
022533 005004 5 0    lda ztemp
022534 003530 5 0    jmp nack11
022535 072204 5 0    nack12:  ldx zmb1
022536 140040 5 0    cra
022537 050110 5 0    sta snull x
022540 050142 5 0    sta i2mnxt x
022541 000401 5 0    .enb t.o
022542 103461 5     jmp clupln i
022543 000000 5 0    .lck fre
022544 044006 5 0    rqsub:  0
022545 006067 5 0    lda neth x
                                         ana [dscpkt]          ;discard bit set?

```

```
022546 100040 5 0           sze
022547 003571 5 0           jmp rqflu          ;yes, go flush pkt
022550 105724 5 0           lda [errrq] i
022551 010001 5 0           sta ireg
022552 132001 5 0           stx ireg i
022553 133724 5 0           stx [errrq] i
022554 004060 5 0           lda [trcrprt]
022555 052004 5 0           era wrdc x      ;mark rerouted
022556 050004 5 0           sta wrdc x
022557 141340 5 0           ica
022560 101400 5 0           smi 0&trcpkt
022561 103543 5 0           jmp rqsub i
022562 005001 5 0           lda zlno
022563 120145 5 0           jst trace.m2idone i    ;**trace
022564 173724 5 0           idx [errrq] i
022565 044004 5 0           lda wrdc x      ;turn off any trace bits
022566 007725 5 0           ana [177777?trcpkt]
022567 050004 5 0           sta wrdc x
022570 103543 5 0           jmp rqsub i

022571 120115 5 0 rqflu:   jst flushi i      ;flush buffer
022572 024223 5 0           irs nsfs
022573 103543 5 0           jmp rqsub i      ;uncount it
```

022574 023062 5 0 inclki: clockf+rstslo-rstclf ;50kb - once per slow
022575 023064 5 0 clockf+rstmed-rstclf ;230kb - five per slow
022576 023061 5 0 clockf+rst050-rstclf ;19.2kb - every other slow
022577 023060 5 0 clockf+rst125-rstclf ;9.6kb - every 5 slow

022600 177767 5 0 wattic: -9. ;ticks to wait to equal one m
022601 177756 5 0 -18. ;corresponds to fast clock tr.
022602 177723 5 0 -45. 0&nfclks
022603 177646 5 0 -90.
022604 177437 5 0 -225. 0&rst2md
022605 177076 5 0 -450. ;note: no entry for single fa
022606 177704 5 0 znup: -60. ;same for all lines

```
.lev t.o
.lck fre
;used by cluplin to reroute packets on dead line
022607 000000 5 0 jsrts: 0
022610 014204 5 0 add zmb1
022611 011617 5 0 sta jsrtq
022612 073617 5 0 jsrts1: ldx jsrtq
022613 120122 5 0 jst mgetqi i
022614 103607 5 0 jmp jsrts i
022615 021543 5 0 jst rqsub ;not ok to reroute
022616 003612 5 0 jmp jsrts1

.lev var
022617 V jsrtq: .block 1
```

```
.stl1 line up subroutine

.lev (m2i,t.o)
022620 000000 1 lnupst: 0 ;do line up stuff
022621 105726 1 lda [smoshf] i ;build shyft for propagation
022622 006072 1 ana [17]
022623 140407 1 tca
022624 014046 1 add [ligr 0]
022625 011664 1 sta lnupsh
022626 021662 1 jst lnupsb ;get line prop delay
022627 101040 1 snz
022630 141206 1 aca ;don't let delay = 0
022631 023727 1 cas [dlinf-1] ;clamp delay to dlinf-1
022632 101000 1 nop
022633 005727 1 lda [dlinf-1] ;init current and base delays
022634 050215 1 sta delbas x
022635 050214 1 sta delcur x
022636 111730 1 sta [rupsnd] i ;send real spf update
022637 006017 1 ana [177774] ;is prop delay > 4*8ms.?
022640 100040 1 sze ;if so, treat as long delay 1
022641 005667 1 lda rtclks+1 ;yes, use med ticks
022642 101040 1 snz
022643 005666 1 lda rtclks ;no, use fast tick
022644 050207 1 sta rtclkp x
022645 010154 1 sta ludflg ;set line up/down flg for bch
022646 140040 1 cra
022647 050117 1 sta line x ;mark line up
022650 044116 1 lda neighb x
022651 012106 1 era mine
022652 100040 1 sze
022653 111731 1 sta [iniflg] i ;set when first unlooped line
022654 044116 1 lda neighb x ;get neighbor number for trap
022655 010001 1 sta ireg
022656 044174 1 lda zzcmup x
022657 072001 1 ldx ireg
;defhlt(/74. line up, a=ticks to come up, x=neighbor
022660 120120 1 jst hltjst i
022661 103620 1 jmp lnupst i

022662 000000 1 lnupsb: 0 ;get line's prop delay
022663 044030 1 lda m2ipcb.delay x ;shyft prop delay
022664 1 lnupsh: .block 1
022665 103662 1 jmp lnupsb i

.lev con
022666 023065 C rtclks: clockfrstfst-rstclf ;retr. clk for land lines
022667 023063 C clockfrst2md-rstclf ;retr. clk for sat lines
022670 023062 C clockfrstslo-rstclf ;retr. clk for dead lines
```

```
; delclr: subroutine to reset delay variables for killin

022671 000000 C    delclr: 0                      ;reset delay variables
022672 010315 C    sta thresh
022673 050211 C    sta delsum x                 ;zero counters and sums
022674 050212 C    sta delsmo x
022675 050213 C    sta delcnt x
022676 004007 C    lda [dlinf]                  ;report down delay values
022677 050214 C    sta delcur x
022700 050215 C    sta delbas x
022701 111730 C    sta [crupsnd] i             ;send real spf update
022702 103671 C    jmp delclr i                ;return to killin

        .stl timeout: line drivers
        .section pg24
        .lev t.o
        .lck all
        rstdout:
024000 000401 S 0    .enb t.o
024001 073713 S      ldx [-ch]
024002 033076 S    clklp1: stx rstctm          ;check for line protocol call
024003 145714 S      lda [mblock+ch] ix       ;save minus line number
024004 101040 S      snz                         ;is this an imp modem?
024005 003016 S      jmp clklp5               ;(i.e. is there an M2I proces
024006 010000 S      sta 0                        ; no, go on to next line
024007 044176 S      lda lnclkp x            ;get clock for this line
024010 010001 S      sta ireg
024011 104001 S      lda ireg i
024012 141206 S      aoa
024013 101040 S      snz                         ;is it about to fire?
024014 121715 S      jst [zdedl] i           ; yes, do line up/down
024015 073076 S      ldx rstctm
024016 024000 S    clklp5: irs 0
024017 003002 S      jmp clklp1
024020 102124 S      jmp toret i

        .lev (t.o,bck)
;Subroutine to set up the K and N counters for a line. Enter with X
;equal to the line number in question. It finds the line speed from the
;config area and resets m2ipcb.speed (thus the speed of a line
;can be changed without restarting the imp). It then sets up
;line-speed dependant parameters: k, n, line clock, and retransmission
;timer.
;
;bit coding of config area and m2ipcb.speed:
;
; 00 50kb
; 01 230kb
; 10 5kb
; 11 9.6kb
;
;Uses IREG. Preserves X. (X=modem block pointer)

024021 000000 S    setnk: 0                      ;set K/N values for line
024022 044025 S      lda pcb.num x             ;get modem number
024023 041475 S      lgl 3 0&cf.size          ;8 words in each config block
024024 015716 S      add [configtrunk+cf.speed]; addr of speed for this l
024025 010001 S      sta ireg
```

```

024026 104001 5 lda ireg i
024027 050026 5 sta m2ipcb.speed x
024030 041475 5 lgl 3
024031 015717 5 add [delxmt+7]
024032 010002 5 sta jreg
024033 044030 5 lda m2ipcb.delay x
024034 041475 5 lgl 3
024035 114002 5 add jreg i
024036 041477 5 lgl 1
024037 023720 5 cas [1150.]
024040 003067 5 jmp setmid
024041 101000 5 nop
024042 005721 5 lda [1250.]
024043 050136 5 setrtt: sta i2mrtt x
024044 044026 5 lda m2ipcb.speed x
024045 015722 5 add [lnclk1]
024046 010002 5 sta jreg
024047 104002 5 lda jreg i
024050 050176 5 sta lnclkp x
024051 044026 5 lda m2ipcb.speed x
024052 012052 5 era [1]
024053 101040 5 snz
024054 003062 5 jmp set230
024055 005101 5 lda zpn1
024056 050173 5 sta zn x
024057 005077 5 lda zpk1
024060 050172 5 sta zk x
024061 103021 5 jmp setnk i

024062 005102 5 set230: lda zpn2 ;use special values
024063 050173 5 sta zn x
024064 005100 5 lda zpk2
024065 050172 5 sta zk x
024066 103021 5 jmp setnk i

024067 023723 5 setmid: cas [3250.] ;mid-range round-trip time?
024070 003074 5 jmp sethi ;no, use high value
024071 101000 5 nop
024072 005724 5 lda [3500.]
024073 003043 5 jmp setrtt ;350 msec
;and continue

024074 005725 5 sethi: lda [10000.] ;1 second
024075 003043 5 jmp setrtt

024076 V .lev var
          rstctm: .block 1 ;loop counter for stdout

024077 177774 C .lev con
          zpk1: -4
024100 177773 C zpk2: -5
024101 177754 C zpn1: -20.
024102 177773 C zpn2: -5

```

```
.stl TIMEOUT
.INCLUDE tim.m4
```

```
;t.o
```

```
;This is timeout initialization. We reset all the timers and fall into the
;normal timeout code.
```

```
.section pg24
```

```
.lev t.o
```

```
024103 004015 5 timini: lda minus2
024104 073671 5           ldx tikcnt
```

```
;minus number of fast clocks
```

```
024105 151726 5 toil1: sta [clockn] ix
024106 024000 5           irs 0
```

```
024107 003105 5           jmp toil1
```

```
024110 004014 5           lda minus1
024111 011712 5           sta toslow
024112 072030 5           ldx [clocks-clockf]
```

```
;initialize slow timers
```

```
;minus number of slow clocks
```

```
024113 151727 5 toil2: sta [clockf] ix
024114 024000 5           irs 0
```

```
024115 003113 5           jmp toil2
```

```
024116 005730 5           lda [-500.]
024117 111731 5           sta [wdtime] i
```

```
;init software WDT
```

```
;Here to debreak for 25.6 milliseconds
```

```
024120 003124 5           jmp todwu
024121 000103 5           tod:   spr
024122 100000 5           skp
024123 000000 5           %crash
```

```
;first time through, skip the
```

```
;TODB: got poked
```

```
024124 004056 5           todwu: lda [%25.6ms]
024125 000020 5           pcb
024126 000203 5           tpr
024127 072006 5           ldx zero
024130 025712 5           irs toslow
```

```
;default=0 (fast tick)
```

```
;is this really a slow tick?
```

```
024131 003135 5           jmp to1
024132 005705 5           lda rstslo
024133 011712 5           sta toslow
024134 072052 5           ldx one
024135 024151 5           to1:   irs time
024136 101000 5           nop
024137 024153 5           irs sync
024140 101000 5           nop
```

```
;no
```

```
;yes, reset clock
```

```
;every 25th tick = 640ms
```

```
024141 045667 5           lda dspcnt x
024142 011711 5           sta todisp
024143 045671 5           lda tikcnt x
024144 010000 5           sta 0
```

```
;count local time
```

```
024145 045711 5           to2:   lda rstcln x
024146 165726 5           irs [clockn] ix
024147 100000 5           skp
024150 151726 5           sta [clockn] ix
024151 024000 5           irs 0
```

```
;select both or just fast tic
```

```
; indices for dispatch
```

```
;and for ticking clocks
```

```
;get ready to reset
```

```
;reset this clock
```

```
024152 003145 5      jmp to2                      ;tick next clock
                     .lev t.o
024153 073711 5      to4:   idx todisp            ;*dispatch t.o routines*
```