

```
L.259      DO 255 J=1,8
L.260      DO 255 K=1,4
L.261      FIELD1(I,J,K)=0
L.262      FIELD2(I,J,K)=0
L.263      FIELD3(I,J,K)=0
L.264      255 CONTINUE
L.265      DO 256 J=1,8
L.266      FIELD1(I,J,5)=0
L.267      256 CONTINUE
L.268      220 CONTINUE
L.269      DO 275 I=1,NOICPS
L.270      DRSICP(I)=0
L.271      DO 275 J=1,CMAX
L.272      DO 275 K=1,8
L.273      PT1ICP(I,J,K)=0
L.274      PT2ICP(I,J,K)=0
L.275      275 CONTINUE
L.276      DO 280 I=1,NONDS
L.277      SWCONT(I)=1
L.278      SEMPH(I)=3
L.279      BUSSEM(I)=3
L.280      DATABS(I)=3
L.281      SEMPRC(I,3)=(-200)
L.282      DO 276 J=1,NOPCKS
L.283      DO 276 K=1,3
L.284      NODEST(I,J,K)=0
L.285      276 CONTINUE
L.286      DO 280 J=1,QUS
L.287      DO 280 K=1,8
L.288      QLIST(I,J,K)=0
L.289      IF (K.GT.2) GO TO 280
L.290      Q(I,J,K)=0
L.291      280 CONTINUE
L.292      DO 285 I=1,NONCPS
L.293      DO 285 M=1,8
L.294      FIELD2(I,M,1)=1
L.295      285 CONTINUE
L.296      DO 292 I=1,NOPCKS
L.297      DO 290 J=1,6
L.298      PACKET(I,J)=0
L.299      290 CONTINUE
L.300      DO 291 J=1,NONCPS
L.301      DUPLCT(I,J)=0
L.302      291 CONTINUE
L.303      292 CONTINUE
L.304      C
L.305      DO 298 I=1,N
L.306      PMEAN=IOP(I,8)
L.307      CALL GGUBS(DSEED,NRDN,R)
L.308      DO 294 K=1,NRDN
L.309      MSGLNZ(I,K)=(-PMEAN)*ALOG(1-R(K))
L.310      294 CONTINUE
L.311      PMEAN=IOP(I,9)
L.312      CALL GGUBS(DSEED,NRDN,R)
L.313      DO 296 K=1,NRDN
L.314      MSGEXP(I,K)=(-1.0/PMEAN)*ALOG(1-R(K))
L.315      296 CONTINUE
L.316      298 CONTINUE
```

BLOCK: MAIN EXECUTION

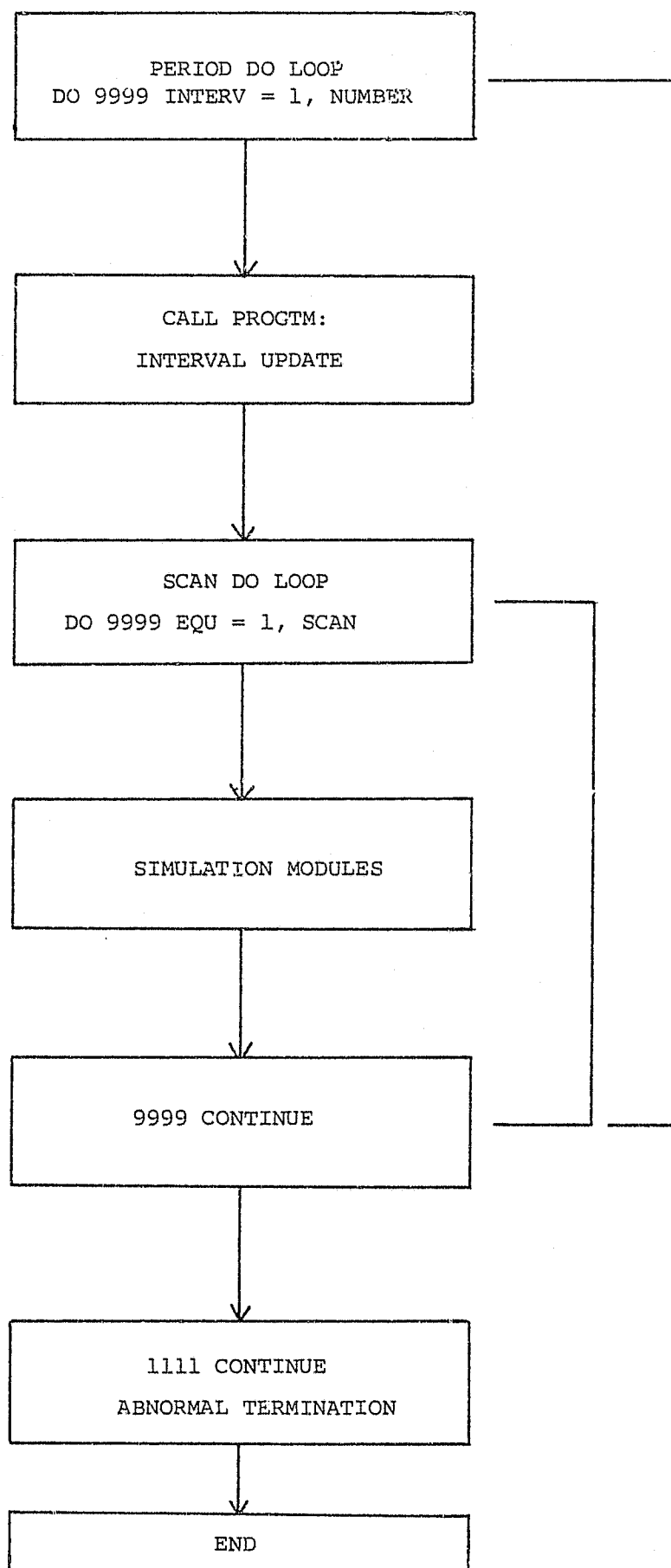


Figure B.5 Main Execution Block

BLOCK: MAIN EXECUTION

MODULE: PACKET INJECTION

Subroutine GGUBS (DSEED, NUMBER, W)

GGUBS is a routine called from the International Mathematical and Statistics Library (IMSL) for generating variates with a Poisson distribution.

DSEED: initial seed - double precision

NUMBER: number of variates generated

W(K): output array of subroutine, where K is greater than
or equal to the number of variates generated

Generation of interarrival times (and lengths) with a Poisson
distribution for packets:

$$p(K) = e^{-t/m} (t/m)^K / K!$$

p(K): probability of K arrivals in interval t

$$p(0) = e^{-t/m} \quad m: \text{mean}$$

probability of an arrival in time t:

$$p = 1 - p(0) = 1 - e^{-t/m}$$

$$t = -m \ln (1 - P)$$

P: is the variate generated by GGUBS

BLOCK: MAIN EXECUTION

MODULE: PACKET INJECTION

VIRTM (I) = PGM CONTROL : I VIRTUAL CIRCUIT NUMBER

INITIAL VALUE VIRTM (I) IS 1

DO FOR EACH VIRTUAL CIRCUIT (4080):

PGM CONTROL VIRTM (I):

IF VIRTM (I) = 1:

SET VIRTM (I) = 2;

GENERATE INTERARRIVAL INTERVAL;

PERFORM INTERVAL GENERATION ON CLOCK;

IF VIRTM (I) = 2:

IF INTERVAL HAS NOT YET ELAPSED GO TO LBL (4080);

SET VIRTM (I) = 1;

DO FOR PACKET ARRAY (4010):

SEARCH FOR FREE LOCATION AND

INSERT PACKET PARAMETERS

4010 CONTINUE

IF NO FREE LOCATION: ABNORMAL TERMINATION

DO FOR NODEST ARRAY (4040):

SEARCH FOR FREE LOCATION AND

INSERT PACKET PARAMETERS

4040 CONTINUE

IF NO FREE LOCATION: ABNORMAL TERMINATION

4080 CONTINUE

PACKET GENERATION FOR VIRTUAL CIRCUITS

```

I      L.      *****
L.318 C      *****
L.319 C
L.320 C      MODEL EXECUTION
L.321      DO 9999 INTERV=1,NUMBER
L.322 C
L.323      CNZ=4
L.324      CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L.325 C
L.326 C
L.327      DO 9999 KQU=1,SCAN
L.328 C
L.329 C      NETWORK PACKET INJECTION FACILITY
L.330      DO 4080 I=NTTLI,NTTL
L.331      NSTAT=VIRTM(I)
L.332      GOTO (4005,4008), NSTAT
L.333 4005 VIRTM(I)=2
L.334      ID=I
L.335      CALL GGUBS(DSEED,NRRN,R)
L.336      TIMERZ(ID)=(-PMEANC)*ALOG(1-R(1))
L.337      CNZ=2
L.338      CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L.339 4008 IF (SIGNTM(I).NE.3) GOTO 4080
L.340      VIRTM(I)=1
L.341 C
L.342      DO 4010 M=1,NOPCKS
L.343      IF (PACKET(M,1).NE.0) GOTO 4010
L.344      DO 4000 NRU=1,NONCPS
L.345      IF (DUPLCT(M,NRU).NE.0) GOTO 4010
L.346 4000 CONTINUE
L.347      MPKO=M
L.348      PACKET(M,1)=I
L.349      PML N=IOP(I,8)
L.350      CALL GGUBS(DSEED,NRRN,R)
L.351      PACKET(M,2)=(-PMEAN)*ALOG(1-R(1))
L.352      PACKET(M,3)=1
L.353      PACKET(M,4)=0
L.354      PACKET(M,5)=7
L.355      PACKET(M,6)=0
L.356      CNZ=5
L.357      CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L.358      GOTO 4030
L.359 4010 CONTINUE
L.360      WRITE (6,4020)
L.361 4020 FORMAT (T2,'ERROR - PACKET ARRAY OVERFLOW')
L.362      GOTO 1111
L.363 4030 DO 4040 MM=1,NOPCKS
L.364      IF (NODEST(IOP(I,2),MM,1).NE.0) GOTO 4040
L.365      NODEST(IOP(I,2),MM,1)=MPKO
L.366      NODEST(IOP(I,2),MM,2)=IOP(I,3)
L.367      NODEST(IOP(I,2),MM,3)=1
L.368      GOTO 4080
L.369 4040 CONTINUE
L.370      WRITE (6,4050)
L.371 4050 FORMAT (T2,'ERROR - NODEST ARRAY OVERFLOW')
L.372      GOTO 1111
L.373 4080 CONTINUE
L.374 C
```

BLOCK: MAIN EXECUTION

MODULES: SOURCE HOST - NETWORK COMMUNICATION
DESTINATION HOST - NETWORK COMMUNICATION

I : IS LOGICAL LINK NUMBER

DO FOR EACH LOGICAL LINK (8999) :

POISSON SOURCE GENERATOR

SOURCE-HOST

SOURCE MESSAGE GENERATION

N W COMM

ICP MESSAGE PROCESSING

DEST-HOST

REASSEMBLY

N W COMM

8999 CONTINUE

SUBMODULES OF ICP SIMULATION

BLOCK: . MAIN EXECUTION

MODULE: SOURCE HOST - NETWORK COMMUNICATION

SUBMODULE: POISSON SOURCE GENERATOR

DO FOR EACH LOGICAL LINK:

STATEG (I) = PGM CONTROL : I LOGICAL LINK NUMBER

INITIAL VALUE STATEG (I) = 2

IF QUOTA OF PACKETS GENERATED STATEG (I) = 1;

PROGRAM CONTROL STATEG (I);

IF STATEG (I) = 2:

SET STATEG (I) = 3;

GENERATE MESSAGE INTERARRIVAL TIME;

PERFORM INTERVAL GENERATION ON CLOCK;

IF STATEG (I) = 3:

IF INTERVAL HAS NOT YET ELAPSED GOTO LBL (320);

SET STATEG (I) = 2;

INCREMENT MESSAGE COUNTER TO INDICATE PACKET
HAS BEEN GENERATED;

DECREMENT COUNTER IOP (I,10) TO INDICATE NUMBER
OF MESSAGES STILL TO BE GENERATED;

IF STATEG (I) = 1:

320 CONTINUE

NO MORE MESSAGES GENERATED

POISSON SOURCE GENERATOR

```
L.375 C
L.376 C      USER - NETWORK INTERACTION
L.377 C
L.378      DO 8999 I=1,N
L.379 C
L.380 C      POISSON SOURCE GENERATOR
L.381      IF (IOP(I,10).EQ.0) STATEG(I)=1
L.382      NSTAT=STATEG(I)
L.383      GO TO (320,300,310), NSTAT
L.384 300     STATEG(I)=3
L.385         ID=I
L.386         TIMERZ(ID)=MSGEXP(I,RDNCTR(I))
L.387         CNZ=2
L.388         CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L.389 310     IF (SIGNTM(I).NE.3) GOTO 320
L.390         STATEG(I)=2
L.391         IOP(I,11)=IOP(I,11)+1
L.392         IOP(I,10)=IOP(I,10)-1
L.393 320     CONTINUE
L.394 C
```

BLOCK: MAIN EXECUTION
MODULE: SOURCE HOST-NETWORK COMMUNICATION
SUBMODULE: SOURCE MESSAGE GENERATION

DO FOR EACH LOGICAL LINK:

MESSAG (I) = PGM CONTROL : I LOGICAL LINK NUMBER;

INITIAL VALUE MESSAG (I) = 1

PROGRAM CONTROL, MESSAG (I)

IF MESSAG (I) = 1:

IF MESSAGE COUNTER ZERO GOTO LBL (460);
DECREMENT MESSAGE COUNTER;
INSERT LENGTH INTO MESSAGE BUFFER;
INCREMENT MESSAGE LENGTH GENERATION COUNT;
IF MESSAGE LENGTH COUNT EQUALS MAXIMUM
THEN WRITE ERROR MESSAGE;
GENERATE TIME REQUIRED TO TRANSMIT CALLES
PACKET OVER CHANNEL;
PERFORM INTERVAL GENERATION ON CLOCK;
SET MESSAG (I) = 1;

IF MESSAG (I) = 2:

IF INTERVAL HAS NOT YET ELAPSED GOTO LBL (460);
SIGNAL CALLES: CALL ESTABLISHMENT TO ICP;
SET MESSAG (I) = 3;

IF MESSAG (I) = 3:

LBL 415 IF "READY FOR NEXT SEGMENT" - RFNS - NOT YET RECEIVED

GOTO LBL (460);

IF MESSAGE SIZE LESS THAN OR EQUAL TO

SEGMENT SIZE GOTO LBL (430);

DECREASE MESSAGE LENGTH BY SEGMENT LENGTH;

INSERT SEGMENT LENGTH INTO ICP BUFFER;

GENERATE TIME REQUIRED TO TRANSMIT SEGMENT OVER CHANNEL;

PERFORM INTERVAL GENERATION ON CLOCK;

SET MESSAG (I) = 4;

IF MESSAG (I) = 4:

IF INTERVAL NOT ELAPSED GOTO LBL (460);

SIGNAL MSGRDY: MESSAGE SEGMENT READY;

SET MESSAG (I) = 3;

GOTO LBL 415

LBL 430 INSERT SEGMENT LENGTH INTO ICP BUFFER;

GENERATE TIME REQUIRED TO TRANSMIT SEGMENT OVER CHANNEL;

PERFORM INTERVAL GENERATION ON CLOCK;

SET MESSAG (I) = 5;

IF MESSAG (I) = 5:

IF INTERVAL NOT ELAPSED GOTO LBL (460);

SIGNAL EOM: END OF MESSAGE;

SET MESSAG (I) = 6;

IF MESSAG (I) = 6:

WAIT FOR SDEOM: SOURCE-DESTINATION END OF MSG;

IF NO SDEOM GOTO LBL (460);

SET MESSAG (I) = 1;

LBL 460 CONTINUE

SOURCE MESSAGE GENERATION

```
L.395 C
L.396 C      MESSAGE GENERATION BY SOURCE
L.397      MSGG=MESSAG(I)
L.398      GO TO (400,410,415,420,440,450), MSGG
L.399 400    IF (IOP(I,11).EQ.0) GO TO 460
L.400      IOP(I,11)=IOP(I,11)-1
L.401      IOP(I,13)=MSGLNZ(I,RDNCTR(I))
L.402      RDNCTR(I)=RDNCTR(I)+1
L.403      IF (RDNCTR(I).EQ.NRON) WRITE (6,405)
L.404 405    FORMAT (T2,'ERROR - NO.OF DEVIATES FOR RANDOM NO. GEN. EXC')
L.405      CALL TIMER (BITLNG,IOP(I,6),RIMR1)
L.406      ID=I+N
L.407      TIMERZ(ID)=RIMR1
L.408      CNZ=2
L.409      CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L.410      MESSAG(I)=2
L.411 410    IF (SIGNTM(I+N).NE.3) GOTO 460
L.412      CALLES=1
L.413      IOP(I,15)=CALLES
L.414      MESSAG(I)=3
L.415 415    RFNS=2
L.416      IF (IOP(I,15).NE.RFNS) GO TO 460
L.417      IF ((IOP(I,13)-SGMENT).LE.0) GO TO 430
L.418      IOP(I,13)=IOP(I,13)-SGMENT
L.419      IOP(I,14)=SGMENT
L.420      CALL TIMER (IOP(I,14),IOP(I,6),RIMR2)
L.421      ID=I+N
L.422      TIMERZ(ID)=RIMR2
L.423      CNZ=2
L.424      CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L.425      MESSAG(I)=4
L.426 420    IF (SIGNTM(I+N).NE.3) GOTO 460
L.427      MSGRDY=3
L.428      IOP(I,15)=MSGRDY
L.429      MESSAG(I)=3
L.430      GO TO 415
L.431 430    IOP(I,14)=IOP(I,13)
L.432      CALL TIMER (IOP(I,14),IOP(I,6),RIMR3)
L.433      ID=I+N
L.434      TIMERZ(ID)=RIMR3
L.435      CNZ=2
L.436      CALL PROGTM(TIMERZ,CLOCK,CNZ,I)
L.437      MESSAG(I)=5
L.438 440    IF (SIGNTM(I+N).NE.3) GOTO 460
L.439      EOM=4
L.440      IOP(I,15)=EOM
L.441      MESSAG(I)=6
L.442 450    SDEOM=5
L.443      IF (IOP(I,15).NE.SDEOM) GO TO 460
L.444      MESSAG(I)=1
L.445 460    CONTINUE
L.446 C
```

FLAGS:

'WAIT-STATE' FLAG	IOP (I, 15) = 11	SET
	IOP (I, 15) = 12	RESET
'FIRST PACKET' FLAG	IOP (I, 16) = 0/1	S/R
'MULTISEGMENT TRM' FLAG	IOP (I, 17) = 0/1	S/R
'TRM CONTROL' FLAG	IOP (I, 19) = 0/1	S/R

BLOCK: MAIN EXECUTION
MODULE: SOURCE HOST-NETWORK COMMUNICATION
SUBMODULE: ICP MESSAGE PROCESSING

DO FOR EACH LOGICAL LINK

SRCIO (I) = PGM CONTROL : I LOGICAL LINK NUMBER;
INITIAL VALUE SRCIO (I) = 1

PROGRAM CONTROL SRCIO (I)

IF SRCIO (I) = 1:

IF NO CALLES SIGNAL GOTO LBL (899);

CALL SUBR ACCESS TO ATTEMPT TO ALLOCATE
BUFFERS: - DOUBLE BUFFERING INITIALLY;

IF ACCESS GRANTED GO TO LBL (506);

CALL SUBR RESERV TO RESERVE CALL REQUEST;

IF RESERVATION REFUSED (ALL LOCATIONS FOR
REQUESTS OCCUPIED): SIGNAL SDEOM;
SET SRCIO (I) = 1;
GOTO LBL (899);

SET ICP FLAG 'WAIT-STATE';

SET SRCIO (I) = 2;

SET LOGICAL LINK FLAG;

IF SRCIO (I) = 2:

IF LOGICAL LINK FLAG NOT RESET GOTO LBL (899);

- INDICATES BUFFERS ALLOCATED

LBL 506 GENERATE TIME TO TRANSMIT RFNS PACKET
OVER CHANNEL TO SOURCE;

PERFORM INTERVAL GENERATION ON CLOCK;

SET SRCIO (I) = 3;

IF SRCIO (I) = 3:

IF INTERVAL NOT ELAPSED GOTO LBL (899);

SIGNAL RFNS;

SET SRCIO (I) = 4;

IF SRCIO (I) = 4:

IF AN EOM OR MSGRDY NOT RECEIVED GOTO LBL (899);

SET "FIRST PACKET" FLAG;

SET "MULTISEGMENT TRM" FLAG;

LBL 700: DIVIDE SEGMENT IN ICP BUFFER INTO PACKETS;

KNTR (I): SET TO NUMBER OF PACKETS;

LBL 715: SET TRM COUNTER TO KNTR (I) - 1;

SET VARIABLE 'NRE' TO NUMBER OF PACKETS
TO BE TRANSMITTED;

IF "FIRST PACKET" FLAG SET NRE TO 1;

LBL 720: DO FOR PACKET ARRAY (722) NRE:

SEARCH FOR FREE LOCATIONS AND

INSERT PACKET PARAMETERS;

722 CONTINUE

IF NO FREE LOCATION: ABNORMAL TERMINATION

DO FOR NODEST ARRAY (727) NRE:

SEARCH FOR FREE LOCATIONS AND

INSERT PACKET PARAMETERS;

727 CONTINUE

IF NO FREE LOCATION: ABNORMAL TERMINATION

RESET "FIRST PACKET" FLAG;

IF A SEGMENT HAS ALREADY BEEN SENT OR IF
EOM NOT RECEIVED (BUT MSGRDY) GOTO LBL (760);

RELEASE BUFFERS NOT USED BY MESSAGE (LESS THAN
ONE SEGMENT SIZE) BY CALLING SUBR ACCESS;

RESET "MULTISEGMENT TRM" FLAG;

LBL 760: RESET 'TRM CONTROL' FLAG;

IF EOM SET 'TRM CONTROL' FLAG;

IF 'FIRST PACKET' FLAG RESET AND 'TRM COUNTER' ZERO
(i.e. ALL PACKETS IN SEGMENT HAVE BEEN TRANSMITTED -
MULTISEGMENT) GO TO LBL (870);

IF 'FIRST PACKET' FLAG RESET AND EOM RECEIVED (ALL
PACKETS TRANSMITTED IN MESSAGE) GO TO LBL (870);

IF 'FIRST PACKET' FLAG RESET (i.e., FIRST PACKET HAS
BEEN TRANSMITTED AND REST TO FOLLOW) GOTO LBL (790);

SET SRCIO (I) = 5;

IF SRCIO (I) = 5:

IF 'NEGATIVE ACK' PACKET, OR 'READY FOR NEXT SEGMENT'
PACKET, OR 'BUFFER ACCESS ACK' PACKET NOT RECEIVED
GOTO LBL (899);

RESET 'SOURCE-TO-ICP TRM' FLAG;

IF 'NEGATIVE ACK' PACKET RECEIVED (NO DEST BUFFERS
AVAILABLE) GOTO LBL (850);

SET TRM COUNTER CONTENTS TO NRE: NO OF PACKETS TO BE SENT

IF 'BUFFER ACCESS ACK' PACKET RECEIVED
INCREMENT NRE (TRANSMIT ALL REMAINING
PACKETS AND RETRANSMIT THE FIRST PACKET);
GOTO LBL (720);

LBL 790: GENERATE TIME TO TRANSMIT PACKET OVER
CHANNEL TO SOURCE HOST; PERFORM INTERVAL GENERATION
ON CLOCK;

SET SRCIO (I) = 6

IF SRCIO (I) = 6:

IF INTERVAL HAS NOT ELAPSED GOTO LBL (899);
TRANSMIT 'RFNS' TO SOURCE ICP;
SET SRCIO (I) = 7;

IF SRCIO (I) = 7:

IF MSGRDY OR EOM: i.e., A SEGMENT NOT RECEIVED YET:
WAIT - GOTO LBL (899);
SET SRCIO (I) = 8;

IF SRCIO (I) = 8:

IF 'SOURCE-TO-ICP TRM' FLAG NOT SET: WAIT -
GOTO LBL (899);
RESET 'SOURCE-TO-ICP TRM' FLAG;
GO TO LBL 700, i.e., PREPARE TO TRM NEXT SEGM;

LBL 850 RELEASE ALL UNUSED BUFFERS THAT HAVE BEEN
ALLOCATED (DOUBLE BUFFERING IS PROVIDED AT
CALLES (CALL ESTABLISHMENT) OR MULTISGM TRM
GOTO LBL 875;

LBL 870 SET SRCIO (I) = 9;

IF SRCIO (I) = 9:

IF 'SDEOM' (ACK OF END OF TRM: SOURCE ICP-TO-DEST
ICP) OR 'TRMNCK' (CALL TERMINATION, NO BUFFERS
AT DEST ICP) WAIT: GOTO LBL (899);

LBL 875 SET SRCIO (I) = 10;

GENERATE TIME TO TRANSMIT SDEOM PACKET TO SOURCE
HOST BY SOURCE ICP (END OF CALL);

PERFORM INTERVAL GENERATION ON CLOCK;

IF SRCIO (I) = 10:

IF INTERVAL HAS NOT ELAPSED GOTO LBL (899);

SET SRCIO (I) = 1;

SIGNAL SDEOM

899 CONTINUE;

END;

SOURCE ICP PROCESSING

```

L.448 C      WAITING FOR MESSAGE ARRIVALS
L.449      MSRC=SRCIO(I)
L.450      GO TO (500,505,510,515,780,800,820,840,870,880), MSRC
L.451 500    CALLES=1
L.452      IF (IOP(I,15).NE.CALLES) GO TO 899
L.453 C      REQUEST FOR SOURCE BUFFER
L.454      RST=2
L.455      NRB=2*NSOURC
L.456      LINKNR=I
L.457      ICPNR=IOP(I,3)
L.458      TNR=1
L.459      CALL ACCESS(RST,NRB,LINKNR,ICPNR,TNR,ACC,SDM,NCR,NSOU,NSING,NMUL)
L.460      IF (ACC.EQ.1) GO TO 506
L.461      IF (ACC.NE.2) WRITE (6,501)
L.462 501    FORMAT (T2,'ERROR N08')
L.463      ST=3
L.464      ICP=IOP(I,3)
L.465      LNK=I
L.466      NO=2*NSOURC
L.467      CNZ=5
L.468      CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L.469      CL=CLOCK
L.470      TPE=1
L.471 C      REQUEST RESERVATION
L.472      CALL RESERV(ST,ICP,LNK,NO,CL,RES,TPE)
L.473      IF (RES.EQ.1) GO TO 503
L.474      IF (RES.NE.2) WRITE (6,502)
L.475 502    FORMAT (T2,'ERROR N09')
L.476 C      PURGE MESSAGE
L.477      IOP(I,15)=5
L.478      SRCIO(I)=1
L.479      GO TO 899
L.480 C      MESSAGE WAITING FOR BUFFER
L.481 503    WAITST=11
L.482      IOP(I,15)=WAITST
L.483      SRCIO(I)=2
L.484      LGLINK(I,1)=1
L.485 505    WAITFN=12
L.486      IF (LGLINK(I,1).NE.2) GOTO 899
L.487 506    CALL TIMER (BITLNG,IOP(I,6),RIMR4)
L.488      ID=I+(2*N)
L.489      TIMERZ(ID)=RIMR4
L.490      CNZ=2
L.491      CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L.492      SRCIO(I)=3
L.493 510    IF (SIGNTM(I+(2*N)).NE.3) GOTO 899
L.494 C      SOURCE NODE READY TO RECEIVE MESSAGE
L.495      RFNS=2
L.496      IOP(I,15)=RFNS
L.497      SRCIO(I)=4
L.498 515    IF (IOP(I,15).NE.3 .AND. IOP(I,15).NE.4) GO TO 899
L.499      IOP(I,16)=0
L.500      IOP(I,17)=0

```

```

L.501 700 KNTF(I)=0
L.502      KNS=IOP(I,14)
L.503      IOP(I,20)=IOP(I,14)
L.504 710 KNF=PSZNO-KNS
L.505      KNTR(I)=KNTR(I)+1
L.506      IF (KNF.GE.0) GO TO 715
L.507      KNS=KNS-PSZNO
L.508      GO TO 710
L.509 715 IOP(I,12)=KNTR(I)-1
L.510      NRE=KNTR(I)
L.511      IF (IOP(I,16).EQ.0) NRE=1
L.512 720 DO 728 NS=1,NRE
L.513      DO 722 M=1,NOPCKS
L.514      IF (PACKET(M,1).NE.0) GO TO 722
L.515      DO 721 NRY=1,NONCPS
L.516      IF (DUPLCT(M,NRY).NE.0) GOTO 722
L.517 721 CONTINUE
L.518      PACKET(M,1)=I
L.519      NTST=PSZNO-IOP(I,20)
L.520      IF (NTST.LT.0) PACKET(M,2)=PSZNO
L.521      IF (NTST.LT.0) IOP(I,20)=IOP(I,20)-PSZNO
L.522      IF (NTST.GE.0) PACKET(M,2)=IOP(I,20)
L.523      PACKET(M,3)=0
L.524      PACKET(M,4)=KNTR(I)
L.525      PACKET(M,5)=1
L.526      IF (IOP(I,15).EQ.4) PACKET(M,5)=7
L.527      PACKET(M,6)=0
L.528      MPK1=M
L.529      GOTO 726
L.530 722 CONTINUE
L.531      WRITE (6,723)
L.532 723 FORMAT (T2,'ERROR - PACKET ARRAY OVERFLOW')
L.533 726 DO 727 MM=1,NOPCKS
L.534      IF (NODEST(IOP(I,2),MM,1).NE.0) GO TO 727
L.535      NODEST(IOP(I,2),MM,1)=MPK1
L.536      NODEST(IOP(I,2),MM,2)=IOP(I,3)
L.537      NODEST(IOP(I,2),MM,3)=1
L.538      GO TO 728
L.539 727 CONTINUE
L.540      WRITE (6,729)
L.541 729 FORMAT (T2,'ERROR - NODEST ARRAY OVERFLOW')
L.542 728 CONTINUE
L.543      IOP(I,16)=IOP(I,16)+1
L.544      IF (IOP(I,17).EQ.1) GO TO 760
L.545      IF (IOP(I,15).EQ.3) GO TO 760
L.546      NX=0
L.547      ND=IOP(I,14)
L.548 750 ND=ND-BUFBLK
L.549      NX=NX+1
L.550      IF (ND.GT.0) GO TO 750
L.551      NRB=(2*NSOURC)-NX
L.552      RST=1
L.553      LINKNR=I
L.554      ICPNR=IOP(I,3)
L.555      TNR=1
L.556      CALL ACCESS(RST,NRB,LINKNR,ICPNR,TNR,ACC,SDM,NCR,NSOU,NSING,NMUL)

```

```

L.557      IOP(I,17)=1
L.558  760  IOP(I,19)=1
L.559      IF (IOP(I,15).EQ.4) IOP(I,19)=2
L.560      IF (IOP(I,16).EQ.1 .AND. IOP(I,12).EQ.0) GO TO 870
L.561      IF (IOP(I,16).GE.2 .AND. IOP(I,15).EQ.4) GO TO 870
L.562      IF (IOP(I,16).GE.2) GO TO 790
L.563      SRCIO(I)=5
L.564  780  IF (IOP(I,15).NE.9 .AND. IOP(I,18).NE.1 .AND. IOP(I,15).NE.23) GO T
CONT:  899
L.565      IOP(I,18)=0
L.566      IF (IOP(I,15).EQ.9) GO TO 850
L.567      NRE=IOP(I,12)
L.568      IF (IOP(I,15).EQ.23) NRE=NRE+1
L.569      GO TO 720
L.570  790  CALL TIMER(BITLNG,IOP(I,6),RIMR5)
L.571      ID=I+(2*N)
L.572      TIMERZ(ID)=RIMR5
L.573      CNZ=2
L.574      CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L.575      SRCIO(I)=6
L.576  800  IF (SIGNTM(I+(2*N)).NE.3) GOTO 899
L.577      IOP(I,15)=2
L.578      SRCIO(I)=7
L.579  820  IF (IOP(I,15).NE.3 .AND. IOP(I,15).NE.4) GO TO 899
L.580      SRCIO(I)=8
L.581  840  IF (IOP(I,18).NE.1) GO TO 899
L.582      IOP(I,18)=0
L.583      GO TO 700
L.584  850  NRB=0
L.585      IOP(I,14)=IOP(I,14)-PSZNO
L.586  860  IOP(I,14)=IOP(I,14)-BUFBLK
L.587      NRB=NRB+1
L.588      IF (IOP(I,14).GE.0) GO TO 860
L.589      IF (IOP(I,19).EQ.1) NRB=2*NSOURC
L.590      RST=1
L.591      LINKNR=I
L.592      ICPNR=IOP(I,3)
L.593      TNR=1
L.594      CALL ACCESS(RST,NRB,LINKNR,ICPNR,TNR,ACC,SDM,NCR,NSOU,NSING,NMUL)
L.595      GO TO 875
L.596  870  SRCIO(I)=9
L.597      IF (IOP(I,15).NE.10 .AND. IOP(I,15).NE.9) GO TO 899
L.598  875  SRCIO(I)=10
L.599      CALL TIMER(BITLNG,IOP(I,6),RTMZ)
L.600      ID=I+(2*N)
L.601      TIMERZ(ID)=RTMZ
L.602      CNZ=2
L.603      CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L.604  880  IF (SIGNTM(I+(2*N)).NE.3) GOTO 899
L.605      SRCIO(I)=1
L.606      IOP(I,15)=5
L.607  899  CONTINUE
L.608 C

```

BLOCK: MAIN EXECUTION

MODULE: DESTINATION HOST-NETWORK COMMUNICATION

DO FOR EACH LOGICAL LINK

DEST(I): PGM CONTROL: I LOGICAL LINK NUMBER

INITIAL VALUE DEST (I) = 1

PROGRAM CONTROL DEST(I)

IF DEST (I) = 1:

IF NO PACKET IN INPUT BUFFER (FROM NETWORK)

GOTO LBL (8900)

IF PACKET IS DATA PACKET GOTO LBL (8030)

CONTROL PACKET SORTING:

IF PACKET CONTROL = 2 SET 'BUFFER ACCESS ACK' FLAG;

IF PACKET CONTROL = 4 SET 'SOURCE-TO-ICP TRM' FLAG;

IF PACKET CONTROL = 5 SET 'TRMNCK' - NO BUFFERS;

IF PACKET CONTROL = 6 SET 'READY FOR NEXT SGM';

GOTO LBL (8900)

LBL 8030 DATA PACKET RECEIVED: FIRST IN SEGMENT:

STORE PARAMETERS OF PACKETS;

IF EOM MESSAGE INDICATED SET 'BUFFERING' FLAG;

IF BUFFERING FLAG RESET CALL ACCESS TO PROVIDE
DOUBLE BUFFERING FOR CALL;

IF BUFFERING FLAG SET CALL ACCESS TO PROVIDE
REQUIRED NO BLOCKS TO STORE MSG;

IF BUFFERS PROVIDED GO TO LBL (8080);

NO BUFFERS AVAILABLE: CALL RESERV TO RESERVE
BUFFERS WHEN THESE BECOME FREE;

SET DEST (I) = 2;

SET 'WAIT' FLAG;

IF DEST (I) = 2:

IF 'WAIT' FLAG NOT RESET GO TO LBL (8900);

i.e. BUFFER BLOCKS NOT YET FREE

TRANSMIT CONTROL PACKET TO SOURCE ICP -

'BUFFER ACCESS ACK': ALL PACKETS TO BE
TRM INCLUDING RETRM OF FIRST (NOT STORED)

GOTO LBL (8090)

LBL (8080): TRANSMIT CONTROL PACKET TO SOURCE ICP -

'SOURCE-TO-ICP TRM': ALL REMAINING PACKETS TO
BE TRM, SIGNAL SOURCE HOST TO TRM
NEXT SEGMENT TO SOURCE ICP;

IF ONLY ONE PACKET IN MESSAGE (SINGLE PCK)

GOTO LBL 8220;

LBL 8090: DO FOR PACKET ARRAY (8092):

SEARCH FOR FREE LOCATIONS AND INSERT PACKET PARAMETERS;

8092 CONTINUE;

IF NO FREE LOCATION: ABNORMAL TERMINATION

DO FOR NODEST ARRAY (8100):

SEARCH FOR FREE LOCATIONS AND

INSERT PACKET PARAMETERS;

8100 CONTINUE;

NO BUFFERS AVAILABLE: CALL RESERV TO RESERVE
BUFFERS WHEN THESE BECOME FREE;

SET DEST (I) = 2;

SET 'WAIT' FLAG;

IF DEST (I) = 2:

IF 'WAIT' FLAG NOT RESET GO TO LBL (8900);

i.e. BUFFER BLOCKS NOT YET FREE

TRANSMIT CONTROL PACKET TO SOURCE ICP -

'BUFFER ACCESS ACK': ALL PACKETS TO BE
TRM INCLUDING RETRM OF FIRST (NOT STORED)

GOTO LBL (8090)

LBL (8080): TRANSMIT CONTROL PACKET TO SOURCE ICP -

'SOURCE-TO-ICP TRM': ALL REMAINING PACKETS TO
BE TRM, SIGNAL SOURCE HOST TO TRM
NEXT SEGMENT TO SOURCE ICP;

IF ONLY ONE PACKET IN MESSAGE (SINGLE PCK)

GOTO LBL 8220;

LBL 8090: DO FOR PACKET ARRAY (8092):

SEARCH FOR FREE LOCATIONS AND INSERT PACKET PARAMETERS;

8092 CONTINUE;

IF NO FREE LOCATION: ABNORMAL TERMINATION

DO FOR NODEST ARRAY (8100):

SEARCH FOR FREE LOCATIONS AND

INSERT PACKET PARAMETERS;

8100 CONTINUE;

IF 'MULTISEGMENT TRM' FLAG RESET, i.e. NO MORE SEGMENTS

TO BE RECEIVED GOTO LBL (8240);

SET DEST (I) = 3;

IF DEST (I) = 3:

WAIT FOR REST OF SEGMENTS CONSTITUTING SEGMENT:

IF NO PACKET GOTO LBL (8900);

STORE PACKET PARAMETERS;

RESET BUFFERING FLAG;

IF 'EOM' SET BUFFERING FLAG;

IF ALL PACKETS CONSTITUTING SEGMENT NOT YET RECEIVED

WAIT FOR REST: GOTO LBL (8900); IF BUFFERING FLAG

SET AND FLOW CONTROL FLAG RESET GOTO LBL 8200:

PREPARE TO TRM SGM TO HOST AND RELEASE BUFFERS;

SET DEST (I) = 4;

GENERATE TIME TO TRM SEGMENT TO DEST HOST;

PERFORM INTERVAL GENERATION ON CLOCK;

IF DEST (I) = 4:

IF INTERVAL NOT ELAPSED GO TO LBL (8900);

SET BUFFERING FLAG

TRANSMIT 'RFNS' PACKET TO SOURCE ICP

GO TO LBL 8090

LBL 8200 IF MULTISGM TRM FLAG RESET i.e. MESSAGE WAS
SINGLE SGM GO TO LBL (8220)
RELEASE UNUSED BUFFERS: BUFFERING REVERTS FROM
DOUBLE TO EXACT BLOCK ALLOCATION GENERATE TIME FOR
SEGMENT TRM TO DEST HOST;
PERFORM INTERVAL GENERATION ON CLOCK;
SET DEST (I) = 5;

IF DEST (I) = 5:

LBL (8220) IF INTERVAL NOT ELAPSED GOTO LBL (8900);
RELEASE ALL BUFFERS;
TRANSMIT FINAL ACK PACKET TO SOURCE ICP;
GO TO LBL (8090);
RESET ALL PARAMETER STORAGE LOCATIONS;
SET DEST (I) = 1;

8900 IF RESERVATION REQUESTED AT SOURCE ICP
SET ICP RESERV FLAG (1);
IF RESERVATION REQUESTED AT DEST ICP
SET ICP RESERV FLAG (2);

LBL 8999 CONTINUE END;

DESTINATION ICP PROCESSING

DO FOR ALL ICP's (5000):
IF ANY ICP RESERV FLAGS ARE SET THEN ACTIVATE
RESERVATION ALGORITHM BY CALLING SUBR RESERV;
RESET ICP RESERV FLAGS;

LBL 5000 CONTINUE;
END;

RESERV SUBR CALLING

```

      NDES=DF-1(I)
      GO TO (8010,8070,8100,8130,8160,8190),NDES
L.612 8010 IF (REASSM(I,1).EQ.0) GO TO 8900
L.613      PKNR=REASSM(I,1)
L.614      REASSM(I,1)=0
L.615      IF (PACKET(PKNR,5).EQ.1 .OR. PACKET(PKNR,5).EQ.7) GOTO 8030
L.616 8015 IF (PACKET(PKNR,5).EQ.4) IOP(I,18)=1
L.617      IF (PACKET(PKNR,5).EQ.5) IOP(I,15)=9
L.618      IF (PACKET(PKNR,5).EQ.6) IOP(I,15)=10
L.619      IF (PACKET(PKNR,5).EQ.2) IOP(I,15)=23
L.620      PACKET(PKNR,1)=0
L.621      GO TO 8900
L.622 8030 REASSM(I,2)=PACKET(PKNR,4)
L.623      REASSM(I,3)=1
L.624      REASSM(I,4)=PACKET(PKNR,2)
L.625      REASSM(I,7)=0
L.626      REASSM(I,6)=0
L.627      IF (PACKET(PKNR,5).EQ.7) REASSM(I,6)=1
L.628      PACKET(PKNR,1)=0
L.629      IF (REASSM(I,6).EQ.0) NRB=2*NSOURC
L.630      IF (REASSM(I,6).EQ.0) GOTO 8065
L.631      NRB=0
L.632      LISM=BUFBLK
L.633      LITM=PSZNO*REASSM(I,2)
L.634 8060 LMLQ=LISM-LITM
L.635      NRB=NRB+1
L.636      LITM=LITM-BUFBLK
L.637      IF (LMLQ.LT.0) GO TO 8060
L.638      REASSM(I,8)=NRB
L.639 8065 RST=2
L.640      LINKNR=I
L.641      ICPNR=IOF(I,5)
L.642      TNR=3
L.643      IF (REASSM(I,2).EQ.1) TNR=2
L.644      CALL ACCESS(RST,NRB,LINKNR,ICPNR,TNR,ACC,SDM,NCR,NSOU,NSING,NMUL)
L.645      IF (ACC.EQ.1) GOTO 8080
L.646      CNZ=5
L.647      CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L.648      CL=CLOCK
L.649      ST=3
L.650      ICP=IOF(I,5)
L.651      CALL RESERV(ST,ICP,I,NRB,CL,RES,TNR)
L.652      IF (RES.EQ.2) REASSM(I,5)=2
L.653      IF (RES.EQ.2) MSMG=5
L.654      IF (RES.EQ.2) GO TO 8090
L.655      DEST(I)=2
L.656      LGLINK(I,2)=1
L.657 8070 IF (LGLINK(I,2).NE.2) GOTO 8900
L.658 C      RESERVED BUFFERS ACCESSED
L.659      MSMG=2
L.660      REASSM(I,5)=1
L.661      REASSM(I,4)=0
L.662      REASSM(I,3)=0
L.663      GOTO 8090
L.664 8080 REASSM(I,5)=1
L.665      MSMG=4
L.666      IF (REASSM(I,2).EQ.1) GO TO 8220
L.667 8090 DO 8095 M=1,NOPCKS
L.668      IF (PACKET(M,1).NE.0) GOTO 8095
L.669      DO 8092 NRD=1,NONCPS
L.670      IF (DUPLCT(M,NRD).NE.0) GOTO 8095
L.671 8092 CONTINUE

```

```
L.672      PACKET(M,1)=1
L.673      PACKET(M,2)=FILLING
L.674      PACKET(M,3)=0
L.675      PACKET(M,4)=1
L.676      PACKET(M,5)=MSG
L.677      PACKET(M,6)=0
L.678      MPK2=M
L.679      GO TO 8097
L.680 8095 CONTINUE
L.681      WRITE (6,8096)
L.682 8096 FORMAT (T2,'ERROR - PACKET ARRAY OVERFLOW')
L.683 8097 DO 8100 M=1,NOPCKS
L.684      IF (NODEST(IOP(I,4),M,1).NE.0) GO TO 8100
L.685      NODEST(IOP(I,4),M,1)=MPK2
L.686      NODEST(IOP(I,4),M,2)=IOP(I,5)
L.687      NODEST(IOP(I,4),M,3)=1
L.688      GO TO 8120
L.689 8100 CONTINUE
L.690      WRITE (6,8110)
L.691 8110 FORMAT (T2,'ERROR - NODEST ARRAY OVERFLOW')
L.692 8120 IF (REASSM(I,5).EQ.1) GO TO 8140
L.693      IF (REASSM(I,5).EQ.2) GO TO 8240
L.694 8140 DEST(I)=3
L.695      IF (REASSM(I,1).EQ.0) GO TO 8900
L.696      PKNR=REASSM(I,1)
L.697      REASSM(I,1)=0
L.698      IF (PACKET(PKNR,5).NE.1 .AND. PACKET(PKNR,5).NE.7) GO TO 8015
L.699      REASSM(I,2)=PACKET(PKNR,4)
L.700      REASSM(I,3)=REASSM(I,3)+1
L.701      REASSM(I,4)=REASSM(I,4)+PACKET(PKNR,2)
L.702      REASSM(I,6)=0
L.703      IF (PACKET(PKNR,5).EQ.7) REASSM(I,6)=1
L.704      PACKET(PKNR,1)=0
L.705      IF (REASSM(I,2).NE.REASSM(I,3)) GO TO 8900
L.706      IF (REASSM(I,6).EQ.1 .AND. REASSM(I,7).EQ.0) GO TO 8200
L.707      IF (REASSM(I,7).EQ.0) GO TO 8170
L.708 8160 DEST(I)=4
L.709      IF (SIGNTM(I+(3*N)+NONCPS+NONDS).NE.3) GO TO 8900
L.710      IF (REASSM(I,6).EQ.1) GO TO 8200
L.711 8170 CALL TIMER(REASSM(I,4),IOP(I,7),RIMR9)
L.712      ID=I+(3*N)+NONCPS+NONDS
L.713      TIMERZ(ID)=RIMR9
L.714      CNZ=2
L.715      CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L.716      DO 8180 M=1,6
L.717      REASSM(I,M)=0
L.718 8180 CONTINUE
L.719      REASSM(I,7)=1
L.720      MSG=4
L.721      REASSM(I,5)=1
L.722      GO TO 8090
L.723 8200 IF (REASSM(I,7).NE.1) GO TO 8220
```

```
L.724      KCC=0
L.725      LISM=BUFBLK
L.726      LITM=PSZNO*REASSM(I,2)
L.727 8210  LMLQ=LISM-LITM
L.728      KCC=KCC+1
L.729      LITM=LITM-BUFBLK
L.730      IF (LMLQ.LT.0) GO TO 8210
L.731      REASSM(I,8)=KCC
L.732      NRB=(2*NSOURC)-REASSM(I,8)
L.733      RST=1
L.734      LINKNR=I
L.735      ICPNR=IOP(I,5)
L.736      TNR=3
L.737      CALL ACCESS(RST,NRB,LINKNR,ICPNR,TNR,ACC,SDM,NCR,NSOU,NSING,NMUL)
L.738 8220  CALL TIMER(REASSM(I,4),IOP(I,7),RIMR10)
L.739      ID=I+(3*N)+NONCPS+NONDS
L.740      TIMERZ(ID)=RIMR10
L.741      CNZ=2
L.742      CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L.743      DEST(I)=5
L.744 8230  IF (SIGNTM(I+(3*N)+NONCPS+NONDS).NE.3) GOTO 8900
L.745      NRB=REASSM(I,8)
L.746      RST=1
L.747      LINKNR=I
L.748      ICPNR=IOP(I,5)
L.749      TNR=3
L.750      IF (REASSM(I,2).EQ.1) TNR=2
L.751      IF (REASSM(I,7).EQ.1) TNR=3
L.752      CALL ACCESS(RST,NRB,LINKNR,ICPNR,TNR,ACC,SDM,NCR,NSOU,NSING,NMUL)
L.753      MSMG=6
L.754      REASSM(I,5)=2
L.755      GOTO 8090
L.756 8240  DO 8250 M=1,8
L.757      REASSM(I,M)=0
L.758 8250  CONTINUE
L.759      DEST(I)=1
L.760 C
L.761 8900  IF (IOP(I,15).EQ.11) DRSICP(IOP(I,3))=1
L.762      IF (DEST(I).EQ.2) DRSICP(IOP(I,5))=1
L.763 8999  CONTINUE
L.764 C
L.765      DO 5000 I=1,NOICPS
L.766      IF (DRSICP(I).EQ.0) GO TO 5000
L.767      ST=7
L.768      CALL RESERV(ST,I,LNK,NO,CL,RES,TPE)
L.769      DRSICP(I)=0
L.770 5000  CONTINUE
L.771 C
```

BLOCK: MAIN EXECUTION

MODULES: NODE-TO-NODE PROTOCOL

I IS NCP NUMBER

DO FOR EACH NCP (3999):

PACKET RECEPTION AND STORAGE FROM BUS

RETRM AND PARTITION TIMING

LINE TRANSMISSION: FIELD SETTING

LINE TRANSMISSION

SUP-MODULES

RECEIVING PHASE

BUFFER RELEASE

ACKNOWLEDGEMENT PROCESSING & STORAGE

3999 CONTINUE

SUBMODULES FOR INTERNODAL TRANSMISSION

BLOCK: MAIN EXECUTION
MODULE: INTERNODAL TRM
SUBMODULE: DATA BUS TO NCP PACKET TRM - STORAGE

DO FOR EACH NCP:

I: LOGICAL LINK NUMBER

IF INTERFACE BUFFER EMPTY i.e. NO PACKET TO BE
TRM TO THE DEST NODE GOTO LBL 3070;

A) DETERMINE PARTITION TO WHICH PACKET IS ALLOCATED:

STORE LOGICAL LINK ID: LOGLNO

DETERMINE FINAL DEST NODE: DESTND - CONTROL
PACKETS BELONGING TO A GIVEN LOGICAL LINK MOVE
IN THE OPPOSITE DIRECTION TO DATA PACKETS, THESE
THUS HAVE A DIFFERENT DEST.

DO LOOP (3006): DETERMINE PRESENT NODE

3008 THEN PARTITION: PARTON - MAY BE FOUND

B) ALLOCATE PRIORITY TO PACKET: TRANSMISSION
PRIORITY IS FCFS

DO LOOP (3002): FIND HIGHEST PRIORITY FACTOR
IN PARTITION BUFFERS

DO LOOP (3004): FIND HIGHEST PRIORITY FACTOR
IN COMMON BUFFERS

ALLOCATE NEXT HIGHEST PRIORITY FACTOR TO PACKET

NB. PACKET WITH LOWEST PRIORITY FACTORS TRANSMITTED FIRST

C) PACKET STORAGE

DO LOOP (3000): SEARCH FOR ADDRESS OF PARTITION
BUFFERS WHERE PACKET MAY BE STORED.
ATTEMPT TO STORE PACKET IN PARTITION
TRMBUF BUFFER, IF OCCUPIED THEN ATTEMPT TO STORE
PACKET IN PARTITION QBUF BUFFER, IF OCCUPIED
THEN STORE PACKET IN A COMMON BUFFER

END LBL 3070

NB. THE SWITCH PROCESSOR MAINTAINS A LIST OF THE
NUMBER OF FREE BUFFER BLOCKS IN EACH NCP. A PACKET
SENT TO AN NCP FROM WITHIN THE NODE MUST THEREFORE
ALWAYS BE ABLE TO FIND A FREE BUFFER (TRANSMISSION
AND RECEIVING BUFFERS ARE SEPARATED) OTHERWISE AN
ABNORMAL TERMINATION RESULTS

DATA BUS SIMULATION

```

L.773 I      INTERNODE PACKET TRANSMISSION: LOCAL PROTOCOLS
L.774 C      DO 3999 I=1,NONCPS
L.775      DATA BUS TO NCP BUFFERING INTERFACE
L.776      IF (IFBUFR(I).EQ.0) GO TO 3070
L.777      PCKNO=IFBUFR(I)
L.778      LOGLNO=PACKET(PCKNO,1)
L.779      DESTND=IOF(LOGLNO,4)
L.779      IF (PACKET(PCKNO,5).NE.1 .AND. PACKET(PCKNO,5).NE.7) DESTND=IOF(LOG
CONT: NO,2)
L.780      DO 3006 KM=1,NONDS
L.781      DO 3006 KN=1,NMAXIN
L.782      IF (MAPNCP(KM,KN).EQ.I) PRSNND=KM
L.783      IF (MAPNCP(KM,KN).EQ.I) GO TO 3008
L.784 3006 CONTINUE
L.785      WRITE (6,3007)
L.786 3007 FORMAT (T2,'ERROR - "PRESENT NODE" NOT FOUND')
L.787      GO TO 1111
L.788 3008 PARTON=NODPLT(PRSNND,DESTND,1)
L.789      KRRS=0
L.790      DO 3002 J=1,8
L.791      DO 3002 K=2,3
L.792      MPPK=PRINS1(I,J,K)
L.793      IF (MPPK.EQ.0) GOTO 3002
L.794      IF (PACKET(MPPK,6).GT.KRRS) KRRS=PACKET(MPPK,6)
L.795 3002 CONTINUE
L.796      DO 3004 K=1,MNOBFS
L.797      MPPK=BFP00L(I,K,1)
L.798      IF (MPPK.EQ.0) GOTO 3004
L.799      IF (PACKET(MPPK,6).GT.KRRS) KRRS=PACKET(MPPK,6)
L.800 3004 CONTINUE
L.801      KRRS=KRRS+1
L.802      PACKET(PCKNO,6)=KRRS
L.803      IF (DUPLCT(PCKNO,I).NE.0) WRITE (6,3014)
L.804      IF (DUPLCT(PCKNO,I).NE.0) GOTO 1111
L.805 3014 FORMAT (T2,'ERROR - DUPLICATION PROCEDURE  STM 3014 MAIN PGM')
L.806      DUPLCT(PCKNO,I)=1
L.807      DO 3000 K=1,8
L.808      IF (PRINS1(I,K,1).EQ.PARTON) KDUMM=K
L.809      IF (PRINS1(I,K,1).EQ.PARTON) GO TO 3010
L.810 3000 CONTINUE
L.811      WRITE (6,3015)
L.812      GO TO 1111
L.813 3015 FORMAT (T2,' ERROR NCP SIM1  STM 3015  MAIN PGM')
L.814 3010 IF (PRINS1(I,KDUMM,2).NE.0) GO TO 3030
L.815      PRINS1(I,KDUMM,2)=PCKNO
L.816      GO TO 3070
L.817 3030 IF (PRINS1(I,KDUMM,3).NE.0) GO TO 3040
L.818      PRINS1(I,KDUMM,3)=PCKNO
L.819      GO TO 3070
L.820 3040 DO 3050 J=1,MNOBFS
L.821      IF (BFP00L(I,J,1).EQ.0) JDUMM=J
L.822      IF (BFP00L(I,J,1).EQ.0) GO TO 3060
L.823 3050 CONTINUE
L.824      WRITE (6,3055)
L.825 3055 FORMAT(T2,' ERROR NCP SIM2')
L.826      GO TO 1111
L.827 3060 BFP00L(I,JDUMM,1)=PCKNO
L.828      BFP00L(I,JDUMM,2)=PARTON
L.829 C

```

BLOCK: MAIN EXECUTION

MODULE: INTERNODAL TRM

SUBMODULE: RETRM AND PARTITION TIMING

DO FOR EACH NCP

A) PERFORM RETRM TIMING

DO LOOP 3110: FOR EACH FIELD DO THE FOLLOWING:

IF A PACKET IS IN TRM THEN CHECK WHETHER
TIME-OUT INTERVAL ELAPSED,
IF ELAPSED SIGNAL 'TIME-OUT'

IF NO PACKET IN TRM THEN CHECK WHETHER
A TIME-OUT IS REQUIRED, IF NOT GO TO (B)

IF SO INITIALIZE CLOCKS

B) PERFORM PARTITION TIMING

DO LOOP 3140: FOR EACH PARTITION DO THE FOLLOWING:

IF A PARTITION NOT BLOCKED GO TO END

IF A PARTITION TIMED, CHECK WHETHER INTERVAL
ELAPSED, IF SO SIGNAL

OTHERWISE GO TO END

REQUEST FOR TIME-OUT: INITIALIZE CLOCKS

END

```
L.830 3070 IFBUFR(I)=0
L.831 C    UPDATE OF INTERVAL TIMERS
L.832      DO 3110 K=1,8
L.833      IF (FIELD3(I,K,1).NE.1) GOTO 3090
L.834      FIELD3(I,K,1)=0
L.835      FIELD3(I,K,2)=1
L.836      ID=K+(4*N)+NONCPS+(2*NONDS)
L.837      TIMERZ(ID)=TXMAX
L.838      CNZ=2
L.839      CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L.840 3090 IF (FIELD3(I,K,2).NE.1) GOTO 3110
L.841      IF (SIGNTM(K+(4*N)+NONCPS+(2*NONDS)).NE.3) GOTO 3110
L.842      FIELD3(I,K,2)=0
L.843      FIELD3(I,K,3)=1
L.844 3110 CONTINUE
L.845      DO 3140 K=1,8
L.846      IF (PRTNS2(I,K,5).NE.1) GOTO 3140
L.847      IF (PRTNS2(I,K,6).EQ.1) GOTO 3130
L.848      PRTNS2(I,K,6)=1
L.849      ID=K+8+(4*N)+NONCPS+(2*NONDS)
L.850      TIMERZ(ID)=PXMAX
L.851      CNZ=2
L.852      CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L.853 3130 IF (SIGNTM(K+8+(4*N)+NONCPS+(2*NONDS)).NE.3) GOTO 3140
L.854      PRTNS2(I,K,5)=0
L.855      PRTNS2(I,K,6)=0
L.856 3140 CONTINUE
L.857 C    END OF DATABUS-NCP TRANSFER PHASE
L.858 C
```

BLOCK: MAIN EXECUTION

MODULE: INTERNODAL TRM

SUBMODULE: LINE TRANSMISSION: FIELD SETTING

DO FOR EACH NCP

A) LINE TESTING

IF CHANNEL BUSY THEN GO TO LBL 3320 IN NEXT SUBMODULE

B) PACKET RETRANSMISSION

NB. PACKETS THAT REQUIRE TO BE RETRANSMITTED
ARE GIVEN PRIORITY

DO LOOP (3150): IF ANY FIELDS CONTAIN PACKETS
FOR RETRM CHOOSE FIRST PACKET ENCOUNTERED, OTHERWISE
GO TO NEXT SECTION (NEW PACKET TRM).

TRANSFER FIELD CONTENTS TO TRANSMISSION

LOCATIONS: PACTR; PARTITION MUST BE UNBLOCKED

INITIATE TIME-OUT INTERVAL

GO TO LINE TRM SUBMODULE

C) NEW PACKET TRANSMISSION

DETERMINE WHETHER FIELD AVAILABLE

IF NO FIELD GO TO NEXT SECTION (ACK TRM).

SELECT PACKET WITH LOWEST PRIORITY FACTOR FOR TRM.

CHECK PARTITION UNBLOCKED. TRANSFER FIELD CONTENTS

TO TRANSMISSION LOCATIONS: PACTR;

SET TRANSMISSION TIMER

GO TO NEXT MODULE

NB. Synchronization of transmitter and receiver, transfer of packets, acknowledgement, etc., is done via flags, these are discussed in Chapter 5.

D) ACK TRANSMISSION

IF NO PACKET RETRM OR TRM AND LINE NOT BUSY, DETERMINE WHETHER ACK CONTROL PACKET TO BE TRM.

IN THIS CASE ONLY FLAGS TRM

IT IS NOT NECESSARY TO MAKE USE OF A FIELD FOR CONTROL PACKET TRM

EACH TIME A PACKET IS RECEIVED AN ACK FLAG IS SET

IF THERE ARE NO DATA PACKETS THAT REQUIRE TO BE SENT A CONTROL PACKET IS SENT AND THE FLAG RESET

THIS METHOD IS NOT INFALLIBLE SINCE THE CONTROL PACKET MAY INCUR ERRORS AND BE DISCARDED, BUT IT DOES PROVIDE A MEANS FOR OBTAINING A REASONABLE RESPONSE TIME FOR STORE-AND-FORWARD BUFFERING

TRANSMISSION PHASE

- 307 -

```

L 859 C
L 860 C
L 861 C   LINE NOT BUSY
L 862     IF (LINE(I).EQ.1) GO TO 3320
L 863 C   LINE FREE: PACKET SELECTION
L 864 C   RETRANSMISSION DUE TO TIMEOUT
L 865     3145 DO 3150 K=1,8
L 866         IF (FIELD3(I,K,3).EQ.1) LLF=K
L 867         IF (FIELD3(I,K,3).EQ.1) GO TO 3160
L 868     3150 CONTINUE
L 869         GO TO 3180
L 870     3160 PACTR(I,1,1)=LLF
L 871         PACTR(I,1,2)=FIELD1(I,LLF,3)
L 872         PACTR(I,1,3)=FIELD1(I,LLF,4)
L 873         PACTR(I,1,4)=FIELD1(I,LLF,1)
L 874         PACTR(I,1,5)=FIELD1(I,LLF,2)
L 875         PACTR(I,1,6)=0
L 876         PACTR(I,1,7)=0
L 877         IF (PACTR(I,1,4).EQ.0 .AND. PACTR(I,1,5).EQ.0) FIELD3(I,LLF,3)=0
L 878         IF (PACTR(I,1,4).EQ.0 .AND. PACTR(I,1,5).EQ.0) GOTO 3145
L 879         DO 3170 M=1,8
L 880             PACTR(I,2,M)=FIELD2(I,M,3)
L 881             PACTR(I,3,M)=FIELD2(I,M,1)
L 882             PACTR(I,4,M)=FIELD2(I,M,2)
L 883     3170 CONTINUE
L 884         FIELD3(I,LLF,1)=1
L 885         FIELD3(I,LLF,3)=0
L 886         GO TO 3270
L 887 C   NEW PACKET SELECTION (FCFS)
L 888 C   DETERMINE WHETHER FIELD AVAILABLE
L 889     3180 DO 3200 K=1,8
L 890         IF (FIELD1(I,K,1).EQ.0) LLH=K
L 891         IF (FIELD1(I,K,1).EQ.0) GO TO 3220
L 892     3200 CONTINUE
L 893         GO TO 3250
L 894     3220 KPRTY=2147483647
L 895         LPM=0
L 896         LQM=0
L 897         DO 3230 J=1,8
L 898             DO 3230 K=2,3
L 899             MPPCK=PRINS1(I,J,K)
L 900             IF (MPPCK.EQ.0) GO TO 3230
L 901             IF (PACKET(MPPCK,6).GT.KPRTY .OR. DUPLCT(MPPCK,I).EQ.2 .OR. PRINS
CONT: ,J,5).EQ.1) GO TO 3230
L 902             KPRTY=PACKET(MPPCK,6)
L 903             LQM=PRINS1(I,J,K)
L 904             LPM=PRINS1(I,J,1)
L 905     3230 CONTINUE
L 906             DO 3240 J=1,MNOBFS
L 907             MPPCK=BFP00L(I,J,1)
L 908             MPARTN=BFP00L(I,J,2)
L 909             IF (MPPCK.EQ.0) GO TO 3240
L 910             IF (PACKET(MPPCK,6).GT.KPRTY .OR. DUPLCT(MPPCK,I).EQ.2) GO TO 324
L 911             DO 3235 K=1,8
L 912             IF (PRINS2(I,K,4).EQ.MPARTN .AND. PRINS2(I,K,5).EQ.1) GO TO 3240
L 913     3235 CONTINUE
L 914             KPRTY=PACKET(MPPCK,6)
L 915             LQM=BFP00L(I,J,1)
L 916             LPM=BFP00L(I,J,2)
L 917     3240 CONTINUE
L 918             IF (KPRTY.EQ.2147483647) GO TO 3250
L 919             DUPLCT(LQM,I)=2
L 920 C   LQM=SELECTED PACKET NUMBER
L 921 C   LPM=SELECTED PARTITION NUMBER

```

```
L.922 C
L.923 C      ALLOCATION OF TRANSMISSION FIELDS
L.924      IF (FIELD1(I,LLH,3).EQ.0 .AND. FIELD1(I,LLH,4).EQ.0) GO TO 3242
L.925      IF (FIELD1(I,LLH,3).EQ.0 .AND. FIELD1(I,LLH,4).EQ.1) GO TO 3243
L.926      WRITE (6,3241)
L.927 3241  FORMAT (T2,'ERROR - TRANSMISSION FIELDS')
L.928 3242  FIELD1(I,LLH,3)=0
L.929      FIELD1(I,LLH,4)=1
L.930      FIELD1(I,LLH,5)=1
L.931      GO TO 3244
L.932 3243  FIELD1(I,LLH,3)=0
L.933      FIELD1(I,LLH,4)=0
L.934      FIELD1(I,LLH,5)=1
L.935 C
L.936 C      TRANSFER PACKET AND PARTITION NUMBERS TO FIELDS
L.937 3244  FIELD1(I,LLH,1)=LQM
L.938      FIELD1(I,LLH,2)=LPM
L.939      FIELD3(I,LLH,1)=1
L.940 C      TRANSMISSION
L.941      PACTR(I,1,1)=LLH
L.942      PACTR(I,1,2)=FIELD1(I,LLH,3)
L.943      PACTR(I,1,3)=FIELD1(I,LLH,4)
L.944      PACTR(I,1,4)=FIELD1(I,LLH,1)
L.945      PACTR(I,1,5)=FIELD1(I,LLH,2)
L.946      PACTR(I,1,6)=0
L.947      PACTR(I,1,7)=0
L.948      DO 3245 M=1,8
L.949      PACTR(I,2,M)=FIELD2(I,M,3)
L.950      PACTR(I,3,M)=FIELD2(I,M,1)
L.951      PACTR(I,4,M)=FIELD2(I,M,2)
L.952 3245  CONTINUE
L.953 C      SET TRANSMISSION TIMER
L.954      FIELD3(I,LLH,1)=1
L.955      GO TO 3270
L.956 C
L.957 C      ACK TRANSMISSION - NO DATA PACKET SENT
L.958 3250  IF (ERRORG(I,3).NE.1) GO TO 3999
L.959      DO 3255 M=1,7
L.960      PACTR(I,1,M)=0
L.961 3255  CONTINUE
L.962      DO 3260 M=1,8
L.963      PACTR(I,2,M)=FIELD2(I,M,3)
L.964      PACTR(I,3,M)=FIELD2(I,M,1)
L.965      PACTR(I,4,M)=FIELD2(I,M,2)
L.966 3260  CONTINUE
L.967 C
```

BLOCK: MAIN EXECUTION
MODULE: INTERNODAL TRM
SUBMODULE: LINE TRANSMISSION

DO FOR EACH NCP

LBL 3270 SET LINE TO BUSY

CALCULATE TIME REQUIRED TO TRANSMIT PACKET
OVER CHANNEL

INCORPORATE ESTIMATED NODAL PROCESSING TIME
DETERMINE IF PACKET INCURS ERROR:

ERROR DISTRIBUTION IS ASSUMED TO BE POISSON AND
PERFORMED ON A LINE ERROR BIT RATE BASIS

NB. THE ACTUAL ERROR DISTRIBUTION ON THE
PCM CHANNEL IS NOT CONSIDERED IMPORTANT,
EMPHASIS IS ON WHETHER PROTOCOL CAN CAUSE
A RETRANSMISSION, AND DO SO WITHOUT
TRANSMITTING DUPLICATES OR PACKETS BEING LOST.
TIME THE INTERVAL BY INTERVAL GENERATION
PERFORMED ON CLOCK.

TRANSFER CONTENTS OF TRANSMISSION ARRAY
TO CONTENTS OF RECEIVER ARRAY.

PACTR → PACRC

```
L.968 C      LINE TRANSMISSION
L.969 3270 LINE(I)=1
L.970      ERRORG(I,3)=0
L.971      IF (PACTR(I,1,4).EQ.0) LBI=3=BITLNG
L.972      IF (PACTR(I,1,4).EQ.0) GO TO 3280
L.973      PKNN=PACTR(I,1,4)
L.974      LBITS=PACKET(PKNN,2)+BITLNG
L.975 3280 CONTINUE
L.976      LLBPS=15000
L.977      CALL TIMER (LBITS,LLBPS,RIMR6)
L.978      ID=I+(3*N)
L.979 C      NODAL PROCESSING TIME PER PACKET: NODPRC
L.980      TIMERZ(ID)=RIMR6+NODPRC
L.981      CNZ=2
L.982      CALL PROGTM (TIMERZ,CLOCK,CNZ,ID)
L.983 C      ERROR GENERATION
L.984 3300 ERRORG(I,2)=ERRORG(I,2)+LBITS
L.985      IF (ERRORG(I,2).LT.ERRORG(I,1)) GO TO 3320
L.986      NRY=1
L.987      PZZCX=100000
L.988      CALL GGUBS(DSEED,NRY,R)
L.989      ERRORG(I,1)=(PZZCX)*ALOG(1-R(1))
L.990      PACTR(I,1,6)=1
L.991      ERRORG(I,2)=0
L.992 C      PACKET IN TRANSMISSION
L.993 3320 IF (SIGNTM(I+(3*N)).NE.3) GOTO 3999
L.994      J=NCPNT(I)
L.995      DO 3330 M=1,7
L.996      PACRC(J,1,M)=PACTR(I,1,M)
L.997 3330 CONTINUE
L.998      DO 3335 M=1,8
L.999      PACRC(J,2,M)=PACTR(I,2,M)
L1000      PACRC(J,3,M)=PACTR(I,3,M)
L1001      PACRC(J,4,M)=PACTR(I,4,M)
L1002 3335 CONTINUE
L1003      LINE(I)=0
L1004      PACRC(J,1,7)=1
L1005 C      END OF TRANSMISSION PHASE
L1006 C
```

BLOCK: MAIN EXECUTION
MODULE: INTERNODAL TRM
SUBMODULE: RECEIVING PHASE

DO FOR EACH NCP

RECEIVE THE PACKET

IF PACKET HAS BIT ERRORS DELETE IT

THIS SUBMODULE OPERATES ON BOTH DATA AND CONTROL PACKETS

DO LOOP (3460) FOR ALL FIELDS:

BY COMPARING THE ARRIVED PACKET'S ACK FLAGS (PACRC)
WITH THE NCP'S TRANSMISSION FLAGS (FIELD 1), THE NCP
MAY DETERMINE THE STATUS OF ITS OWN PACKETS THAT IT
HAS SENT TO THE OTHER NCP, i.e. THE NCP WHICH
TRANSMITTED THE ABOVE ARRIVED PACKET.

THE PACKET STATUS OF EACH FIELD MAY BE

NO-OPERATION

RELEASE OF BUFFER (SUCCESSFUL TRM)

RETRANSMISSION (DUE TO BLOCKED PARTITIONS, i.e.
PARTITION TO WHICH SENT PACKET BELONGED, AND
COMMON BUFFERS ALL OCCUPIED)

END

THE ABOVE OPERATIONS ARE CONDUCTED IN THE NEXT SUBMODULE

```
L1007 C    RECEIVING PHASE
L1008 C
L1009 C    PACKETS WITH BIT ERRORS ARE DELETED
L1010 C    IF (PACRC(J,1,6).EQ.1) GO TO 3999
L1011 C    PROCESSING OF ACK INFORMATION
L1012 C    RELEASE OF BUFFERS
L1013 C    DO 3460 K=1,8
L1014 C    NO-OP
L1015 C    IF (FIELD1(J,K,3).EQ.0 .AND. FIELD1(J,K,4).EQ.0 .AND. PACRC(J,2,K).
CONT: Q.0 .AND. PACRC(J,3,K).EQ.1 .AND. PACRC(J,4,K).EQ.1) GO TO 3460
L1016 C    IF (FIELD1(J,K,3).EQ.0 .AND. FIELD1(J,K,4).EQ.1 .AND. PACRC(J,2,K).
CONT: Q.0 .AND. PACRC(J,3,K).EQ.1 .AND. PACRC(J,4,K).EQ.0) GO TO 3460
L1017 C
L1018 C    NO-OP
L1019 C    IF (FIELD1(J,K,3).EQ.0 .AND. FIELD1(J,K,4).EQ.0 .AND. PACRC(J,2,K).
CONT: Q.1 .AND. PACRC(J,3,K).EQ.1 .AND. PACRC(J,4,K).EQ.1) GO TO 3460
L1020 C    IF (FIELD1(J,K,3).EQ.0 .AND. FIELD1(J,K,4).EQ.1 .AND. PACRC(J,2,K).
CONT: Q.1 .AND. PACRC(J,3,K).EQ.1 .AND. PACRC(J,4,K).EQ.0) GO TO 3460
L1021 C
L1022 C    RELEASE OF BUFFERS
L1023 C    IF (FIELD1(J,K,3).EQ.0 .AND. FIELD1(J,K,4).EQ.0 .AND. PACRC(J,2,K).
CONT: Q.0 .AND. PACRC(J,3,K).EQ.1 .AND. PACRC(J,4,K).EQ.0) GO TO 3350
L1024 C    IF (FIELD1(J,K,3).EQ.0 .AND. FIELD1(J,K,4).EQ.1 .AND. PACRC(J,2,K).
CONT: Q.0 .AND. PACRC(J,3,K).EQ.1 .AND. PACRC(J,4,K).EQ.1) GO TO 3350
L1025 C
L1026 C    RETRANSMISSION
L1027 C    IF (FIELD1(J,K,3).EQ.0 .AND. FIELD1(J,K,4).EQ.0 .AND. PACRC(J,2,K).
CONT: Q.1 .AND. PACRC(J,3,K).EQ.1 .AND. PACRC(J,4,K).EQ.0) GO TO 3430
L1028 C    IF (FIELD1(J,K,3).EQ.0 .AND. FIELD1(J,K,4).EQ.1 .AND. PACRC(J,2,K).
CONT: Q.1 .AND. PACRC(J,3,K).EQ.1 .AND. PACRC(J,4,K).EQ.1) GO TO 3430
L1029 C
L1030 C    WRITE (6,3345)
L1031 C    3345 FORMAT(T2,'ERROR - FRAME TRANSMISSION')
```

BLOCK: MAIN EXECUTION
MODULE: INTERNODAL TRM
SUBMODULE: RELEASE OF BUFFERS

DO FOR EACH NCP

THIS SUBMODULE IS CONCERNED WITH THE RELEASE OF BUFFERS IN ALL
FIELDS DUE TO ACKS RECEIVED IN THE FLAGS OF THE ARRIVED PACKETS

DO FOR EACH FIELD

A) BUFFER RELEASE (LBL 3350)

RESET ALL FIELD FLAGS, INITIALIZING IT FOR THE NEXT
PACKET TRANSMISSION

DELETE THE INTERVAL IDENTIFIER STORED IN CLOCK -
RETRANSMISSION INTERVAL

SEARCH FOR PACKET IN PARTITION AND COMMON BUFFERS
AND DELETE IT.

GO TO END (LBL (3460)

B) RETRANSMISSION DUE TO BLOCKED PARTITION

RESET ALL FIELD FLAGS, INITIALIZING IT FOR THE NEXT
PACKET TRANSMISSION

DELETE THE INTERVAL IDENTIFIER STORED IN CLOCK -
RETRM INTERVAL

INITIALIZE PARTITION TIME-OUT

LBL 3460 END

```
L1032 L    RELEASE OF BUFFERS
L1033 3350 IF (FIELD1(J,K,5).NE.1) GOTO 3460
L1034      PCCNO=FIELD1(J,K,1)
L1035      PAANO=FIELD1(J,K,2)
L1036      FIELD1(J,K,1)=0
L1037      FIELD1(J,K,2)=0
L1038      FIELD1(J,K,5)=0
L1039      FIELD3(J,K,1)=0
L1040      FIELD3(J,K,2)=0
L1041      CNZ=3
L1042      ID=K+(4*N)+NONCPS+(2*NONDS)
L1043      CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L1044      SIGNTM(K+(4*N)+NONCPS+(2*NONDS))=1
L1045 C    SEARCH FOR PACKET AND DELETE
L1046 C    TRMBUF/QBUF
L1047      DO 3360 KD=1,8
L1048      DO 3360 LD=2,3
L1049      IF (PRTNS1(J,KD,LD).NE.PCCNO) GO TO 3360
L1050      PRTNS1(J,KD,LD)=0
L1051      DUPLCT(PCCNO,J)=0
L1052      GO TO 3460
L1053 3360 CONTINUE
L1054      DO 3370 JD=1,MNOBFS
L1055      IF (BFP00L(J,JD,1).NE.PCCNO) GO TO 3370
L1056      BFP00L(J,JD,1)=0
L1057      DUPLCT(PCCNO,J)=0
L1058      GO TO 3460
L1059 3370 CONTINUE
L1060      WRITE (6,3380)
L1061 3380 FORMAT (T2,'ERROR - PACKET (CHANNEL) RELEASE')
L1062 C    RETRANSMISSION DUE TO BLOCKED PARTITION: TIME-OUT
L1063 3430 IF (FIELD1(J,K,5).NE.1) GOTO 3460
L1064      PAANO=FIELD1(J,K,2)
L1065      NPCS=FIELD1(J,K,1)
L1066      FIELD1(J,K,1)=0
L1067      FIELD1(J,K,2)=0
L1068      FIELD1(J,K,5)=0
L1069      DUPLCT(NPCS,J)=1
L1070      FIELD3(J,K,1)=0
L1071      FIELD3(J,K,2)=0
L1072      CNZ=3
L1073      ID=K+(4*N)+NONCPS+(2*NONDS)
L1074      CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L1075      SIGNTM(K+(4*N)+NONCPS+(2*NONDS))=1
L1076      DO 3440 M=1,8
L1077      IF (PRTNS2(J,M,4).EQ.PAANO) LLC=M
L1078      IF (PRTNS2(J,M,4).EQ.PAANO) GO TO 3450
L1079 3440 CONTINUE
L1080      WRITE (6,3445)
L1081 3445 FORMAT (T2,' ERROR - PARTITION MISSING')
L1082 3450 PRTNS2(J,LLC,5)=1
L1083      PRTNS2(J,LLC,6)=0
L1084 3460 CONTINUE
L1085 C
```

BLOCK: MAIN EXECUTION
MODULE: INTERNODAL TRM
SUBMODULE: ACK PROCESSING AND STORAGE

THIS MODULE DETERMINES WHETHER THE PACKET IS TO BE RECEIVED
AND STORED PROPER, i.e. WHETHER IT IS TO BE TRM ON TO NEXT NODE,
OR WHETHER IT IS A DUPLICATE AND IS TO BE DELETED

DETERMINE WHETHER PACKET IS TO BE RECEIVED OR WHETHER DELETED -
DUPLICATE

DETERMINE NEW PARTITION OF PACKET

ATTEMPT TO STORE PACKET IN PARTITION OR COMMON BUFFERS

IF STORED INFORM NODAL SUPERVISOR OF PACKET PARAMETERS
AND REQUEST FOR SERVICE

IF NOT STORED SET APPROPRIATE PARTITION FLAG INFORMING
OF BLOCKED PARTITION

SET TRANSMITTER FLAGS ACCORDING TO WHETHER PACKET WAS
SUCCESSFULLY RECEIVED OR NOT

RESET PACRC PARAMETERS

END

```

L1086 I      ADI ONLY
L1087      IF (PACRC(J,1,1).EQ.0 .AND. PACRC(J,1,4).EQ.0) GO TO 3999
L1088 C      SYNCHRONIZE TRM AND REC FIELDS
L1089 C      DEPOSIT PACKET IN DESTINATION BUFFERS
L1090      FRMFLD=PACRC(J,1,1)
L1091 C      ACK: 1 1 TRANSMITTED PREVIOUSLY - RECEIVE PACKET
L1092      IF (FIELD2(J,FRMFLD,1).EQ.1 .AND. FIELD2(J,FRMFLD,2).EQ.1 .AND. F
CONT: C(J,1,2).EQ.0 .AND. PACRC(J,1,3).EQ.0) GO TO 3465
L1093 C      ACK: 0 1 TRANSMITTED PREVIOUSLY - RECEIVE PACKET
L1094      IF (FIELD2(J,FRMFLD,1).EQ.1 .AND. FIELD2(J,FRMFLD,2).EQ.0 .AND. F
CONT: C(J,1,2).EQ.0 .AND. PACRC(J,1,3).EQ.1) GO TO 3465
L1095 C      ACK: 1 1 FIELD:10 - DUPLICATE
L1096      IF (FIELD2(J,FRMFLD,1).EQ.1 .AND. FIELD2(J,FRMFLD,2).EQ.1 .AND. F
CONT: C(J,1,2).EQ.0 .AND. PACRC(J,1,3).EQ.1) ERRORG(J,3)=1
L1097      IF (FIELD2(J,FRMFLD,1).EQ.1 .AND. FIELD2(J,FRMFLD,2).EQ.1 .AND. F
CONT: C(J,1,2).EQ.0 .AND. PACRC(J,1,3).EQ.1) GO TO 3550
L1098 C      ACK: 0 1 FIELD:00 - DUPLICATE
L1099      IF (FIELD2(J,FRMFLD,1).EQ.1 .AND. FIELD2(J,FRMFLD,2).EQ.0 .AND. F
CONT: C(J,1,2).EQ.0 .AND. PACRC(J,1,3).EQ.0) ERRORG(J,3)=1
L1100      IF (FIELD2(J,FRMFLD,1).EQ.1 .AND. FIELD2(J,FRMFLD,2).EQ.0 .AND. F
CONT: C(J,1,2).EQ.0 .AND. PACRC(J,1,3).EQ.0) GO TO 3570
L1101      WRITE (6,3464)
L1102 3464 FORMAT(T2,' SYNCHRONIZATION ERROR')
L1103 C      STORAGE OF PACKET
L1104 3465 DO 3470 K=1,8
L1105      IF (PRINS2(J,K,1).EQ.PACRC(J,1,5)) LDD=K
L1106      IF (PRINS2(J,K,1).EQ.PACRC(J,1,5)) GO TO 3480
L1107 3470 CONTINUE
L1108      WRITE (6,3475)
L1109 3475 FORMAT (T2,'ERROR - PARTITIONS - RECEIVING')
L1110      GOTO 1111
L1111 C      TRMBUF
L1112 3480 IF (PRINS2(J,LDD,2).NE.0) GO TO 3490
L1113      PRINS2(J,LDD,2)=PACRC(J,1,4)
L1114      GO TO 3530
L1115 C      DBUF
L1116 3490 IF (PRINS2(J,LDD,3).NE.0) GO TO 3500
L1117      PRINS2(J,LDD,3)=PACRC(J,1,4)
L1118      GO TO 3530
L1119 C      BFPOLR
L1120 3500 CONTINUE
L1121      DO 3510 MKG=1,MNOBFS
L1122      IF (BFPOLR(J,MKG,1).EQ.0) LLFG=MKG
L1123      IF (BFPOLR(J,MKG,1).EQ.0) GO TO 3520
L1124 3510 CONTINUE
L1125 C      NO BUFFERS FREE
L1126      KSIGN=1
L1127      GO TO 3540
L1128 3520 BFPOLR(J,LLFG,1)=PACRC(J,1,4)
L1129      BFPOLR(J,LLFG,2)=PACRC(J,1,5)
L1130 3530 KSIGN=0

```

- 317 -

```

L1131 C      WRITE INFORMATION INTO NCPST ARRAY
L1132      DO 3531 MKG=1,NONDS
L1133      DO 3531 I=1,NMAXIN
L1134      IF (MAPNLF(MKG,I).EQ.J) NV=MKG
L1135      IF (MAPNCF(MKG,I).EQ.J) GO TO 3533
L1136      3531 CONTINUE
L1137      WRITE (6,3532)
L1138      3532 FORMAT (T2,'ERROR - NODAL NUMBER NOT FOUND STM 3532 ')
L1139      3533 DO 3534 MKG=1,NOPCKS
L1140      IF (NODEST(NV,MKG,1).NE.0) GO TO 3534
L1141      NODEST(NV,MKG,1)=PACRC(J,1,4)
L1142      NODEST(NV,MKG,2)=J
L1143      NODEST(NV,MKG,3)=2
L1144      GO TO 3540
L1145      3534 CONTINUE
L1146      WRITE (6,3535)
L1147      3535 FORMAT (T2,' ERROR - NODEST ARRAY OVERFLOW')
L1148      3540 ERRORG(J,3)=1
L1149 C
L1150 C      FRAME FIELD ACKNOWLEDGEMENT
L1151      IF (KSIGN.EQ.0 .AND. FIELD2(J,FRMFLD,1).EQ.1 .AND. FIELD2(J,FRMF
CONT: ).EQ.0) GO TO 3550
L1152      IF (KSIGN.EQ.1 .AND. FIELD2(J,FRMFLD,1).EQ.1 .AND. FIELD2(J,FRMF
CONT: ).EQ.0) GO TO 3560
L1153      IF (KSIGN.EQ.0 .AND. FIELD2(J,FRMFLD,1).EQ.1 .AND. FIELD2(J,FRMF
CONT: ).EQ.1) GO TO 3570
L1154      IF (KSIGN.EQ.1 .AND. FIELD2(J,FRMFLD,1).EQ.1 .AND. FIELD2(J,FRMF
CONT: ).EQ.1) GO TO 3580
L1155
L1156      WRITE (6,3545)
L1157      3545 FORMAT (T2,'ERROR - FRAME ACK')
L1158      3550 FIELD2(J,FRMFLD,1)=1
L1159      FIELD2(J,FRMFLD,2)=1
L1160      FIELD2(J,FRMFLD,3)=0
L1161      GO TO 3590
L1162      3560 FIELD2(J,FRMFLD,1)=1
L1163      FIELD2(J,FRMFLD,2)=1
L1164      FIELD2(J,FRMFLD,3)=1
L1165      GO TO 3590
L1166      3570 FIELD2(J,FRMFLD,1)=1
L1167      FIELD2(J,FRMFLD,2)=0
L1168      FIELD2(J,FRMFLD,3)=0
L1169      GO TO 3590
L1170      3580 FIELD2(J,FRMFLD,1)=1
L1171      FIELD2(J,FRMFLD,2)=0
L1172      FIELD2(J,FRMFLD,3)=1
L1173      3590 CONTINUE
L1174      KSIGN=0
L1175 C      RESET PACRC TO ZERO
L1176      DO 3595 M=1,7
L1177      PACRC(J,1,M)=0
L1178      3595 CONTINUE
L1179 C
L1180      3999 CONTINUE
L1181 C

```

- 317 -

```

L1131 C      WRITE INFORMATION INTO NCPST ARRAY
L1132      DO 3531 MKG=1,NONDS
L1133      DO 3531 F=1,NMAXIN
L1134      IF (MAPNLP(MKG,K).EQ.J) NV=MKG
L1135      IF (MAPNCP(MKG,K).EQ.J) GO TO 3533
L1136 3531 CONTINUE
L1137      WRITE (6,3532)
L1138 3532 FORMAT (T2,'ERROR - NODAL NUMBER NOT FOUND STM 3532 ')
L1139 3533 DO 3534 MKG=1,NOPCKS
L1140      IF (NODEST(NV,MKG,1).NE.0) GO TO 3534
L1141      NODEST(NV,MKG,1)=PACRC(J,1,4)
L1142      NODEST(NV,MKG,2)=J
L1143      NODEST(NV,MKG,3)=2
L1144      GO TO 3540
L1145 3534 CONTINUE
L1146      WRITE (6,3535)
L1147 3535 FORMAT (T2,' ERROR - NODEST ARRAY OVERFLOW')
L1148 3540 ERRORG(J,3)=1
L1149 C
L1150 C      FRAME FIELD ACKNOWLEDGEMENT
L1151      IF (KSIGN.EQ.0 .AND. FIELD2(J,FRMFLD,1).EQ.1 .AND. FIELD2(J,FRMF
CONT: ).EQ.0) GO TO 3550
L1152      IF (KSIGN.EQ.1 .AND. FIELD2(J,FRMFLD,1).EQ.1 .AND. FIELD2(J,FRMF
CONT: ).EQ.0) GO TO 3560
L1153      IF (KSIGN.EQ.0 .AND. FIELD2(J,FRMFLD,1).EQ.1 .AND. FIELD2(J,FRMF
CONT: ).EQ.1) GO TO 3570
L1154      IF (KSIGN.EQ.1 .AND. FIELD2(J,FRMFLD,1).EQ.1 .AND. FIELD2(J,FRMF
CONT: ).EQ.1) GO TO 3580
L1155
L1156      WRITE (6,3545)
L1157 3545 FORMAT (T2,'ERROR - FRAME ACK')
L1158 3550 FIELD2(J,FRMFLD,1)=1
L1159      FIELD2(J,FRMFLD,2)=1
L1160      FIELD2(J,FRMFLD,3)=0
L1161      GO TO 3590
L1162 3560 FIELD2(J,FRMFLD,1)=1
L1163      FIELD2(J,FRMFLD,2)=1
L1164      FIELD2(J,FRMFLD,3)=1
L1165      GO TO 3590
L1166 3570 FIELD2(J,FRMFLD,1)=1
L1167      FIELD2(J,FRMFLD,2)=0
L1168      FIELD2(J,FRMFLD,3)=0
L1169      GO TO 3590
L1170 3580 FIELD2(J,FRMFLD,1)=1
L1171      FIELD2(J,FRMFLD,2)=0
L1172      FIELD2(J,FRMFLD,3)=1
L1173 3590 CONTINUE
L1174      KSIGN=0
L1175 C      RESET PACRC TO ZERO
L1176      DO 3595 M=1,7
L1177      PACRC(J,1,M)=0
L1178 3595 CONTINUE
L1179 C
L1180 3999 CONTINUE
L1181 C

```

BLOCK: MAIN EXECUTION

MODULE: DATA BUS PACKET TRANSFERS

THIS MODULE SIMULATES THE TRANSFER OF PACKETS VIA THE DATA
BUS BETWEEN ICP's AND NCP's UNDER THE COMMAND DIRECTIVES
OF THE SWITCH PROCESSOR

DO FOR EACH NODE

DATABS (I): PGM CONTROL I: NODAL NUMBER

PROGRAM CONTROL DATABS (I)

IF DATABS (I) = 1:

WAIT FOR SYNCHRONIZATION SIGNAL BUSSEM = 1

TRANSFER THE COMMANDS FROM THE SWITCH TO THE
PROCESSORS PT2ICP → PT1ICP

PT2NCP → PT1NCP

(NB. TRANSFER TIME INCORPORATED INTO ESTIMATED
SWITCH ALGORITHM EXECUTION TIME)

SET CYCLE COUNTER TO 1 : NONCC

SEND SYNCHRONIZATION SIGNAL SEMPH = 3

RESET SYNCHRONIZATION SIGNAL BUSSEM = 3

(NB. THE TWO ABOVE SIGNALS ARE SIMULATION PROGRAM
EXECUTION VARIABLES, AND HAVE NO RELATION TO
THE SWITCH PROCESSOR AND ICP/NCP SYNCHRONIZATION.
THE FOLLOWING DO.)

SIGNAL (BY SWITCH) FOR STARTING PROCESSOR TO USE BUS:

SEMPRC (I,4) PROCESSOR

SEMPRC (I,5) ADDRESS

LBL 6015 IF TRANSMITTING PROCESSOR IS ICP:

IF PACKET IS A DATA PACKET, THEN RELEASE BUFFERS IN ICP
(BY REQUIRED NUMBER OF BLOCKS EQUIVALENT TO ONE PACKET),
PROVIDED MESSAGE TO WHICH PACKET BELONGS IS EOM..

IF TRANSMITTING PROCESSOR IS NCP:

RELEASE APPROPRIATE NUMBER OF BLOCKS IN NCP.

LBL 6245: ICP OR NCP:

GENERATE TIME REQUIRED TO TRANSMIT PACKET ACROSS BUS

PERFORM INTERVAL GENERATION ON CLOCK

SET DATABS (I) = 2

IF DATABS (I) = 2:

WAIT UNTIL INTERVAL HAS ELAPSED

DO LOOP (6105):

IF PROCESSOR TRANSMITTING IS ICP TRANSFER COMMANDS
TO SIMULATION VARIABLE VARTM

PTICP → VARTM

IF PROCESSOR TRANSMITTING IS NCP TRANSFER COMMANDS
TO SIMULATION VARIABLE VARTM

PTINCP → VARTM

USING VARTM:

IF PACKET BELONGS TO VIRTUAL CIRCUIT AND HAS REACHED
DEST NODE THEN DELETE

IF PACKET BELONGS TO LOGICAL LINK AND HAS REACHED DEST
NODE THEN STORE IN ICP BUFFER REASSM

IF PACKET AT INTERMEDIATE NODE THEN STORE IN NCP
BUFFER IFBUFR

IF END OF PERIOD REACHED, i.e. ALL PROCESSORS IN A PERIOD
HAVE FINISHED TRM THEN SET DATABS (I) = 1 AND GO TO END

IF END OF CYCLE REACHED THEN INCREMENT CYCLE COUNTER

TRANSFER ADDRESS OF NEXT PROCESSOR TO TRANSMIT

VARTM → SEMPRC

GO TO LBL 6015 (REPEAT ABOVE STPPS FOR EACH TRM
PROCESSOR UNTIL END-OF-PERIOD)

END

- 321 -

```

L1182 C      DATA BUS PACKET TRANSFERS
L1183      DO 9750 I=1,NONDS
L1184      NDATE=DATE$(I)
L1185      GO TO (6000,6250),NDATE
L1186 6000 IF (BUSSEM(I).NE.1) GO TO 6999
L1187      DO 6008 M=1,NMAXIC
L1188      NZW=MAPICP(I,M)
L1189      IF (MAPICP(I,M).EQ.0) GO TO 6008
L1190      DO 6005 K=1,CMAX
L1191      DO 6005 L=1,8
L1192      PT1ICP(NZW,K,L)=PT2ICP(NZW,K,L)
L1193 6005 CONTINUE
L1194 6008 CONTINUE
L1195      DO 6012 M=1,NMAXIN
L1196      NZQ=MAPNCP(I,M)
L1197      IF (MAPNCP(I,M).EQ.0) GO TO 6012
L1198      DO 6010 K=1,CMAX
L1199      DO 6010 L=1,8
L1200      PT1NCP(NZQ,K,L)=PT2NCP(NZQ,K,L)
L1201 6010 CONTINUE
L1202 6012 CONTINUE
L1203      NONCC(I)=1
L1204      SEMPH(I)=3
L1205      BUSSEM(I)=3
L1206      SEMPRC(I,4)=SEMPRC(I,1)
L1207      SEMPRC(I,5)=SEMPRC(I,2)
L1208 6015 IF (SEMPRC(I,5).EQ.2) GO TO 6025
L1209      PCCKNO=PT1ICP(SEMPRC(I,4),NONCC(I),3)
L1210      IF (IOP(PACKET(PCCKNO,1),19).EQ.1) GO TO 6245
L1211      IF (PACKET(PCCKNO,5).NE.1 .AND. PACKET(PCCKNO,5).NE.7) GO TO 6245
L1212      IF (PACKET(PCCKNO,3).EQ.1) GOTO 6245
L1213      NRB=0
L1214      LNM=PACKET(PCCKNO,2)
L1215 6020 LMLE=BUFBLK-LNM
L1216      NRB=NRB+1
L1217      LNM=LNM-BUFBLK
L1218      IF (LMLE.LT.0) GO TO 6020
L1219      RST=1
L1220      LINKNR=PACKET(PCCKNO,1)
L1221      ICPNR=IOP(PACKET(PCCKNO,1),3)
L1222      TNR=1
L1223      CALL ACCESS(RST,NRB,LINKNR,ICPNR,TNR,ACC,SDM,NCR,NSOU,NSING,NMUL)
L1224      GO TO 6245
L1225 6025 PCCKNO=PT1NCP(SEMPRC(I,4),NONCC(I),3)
L1226      DO 6030 M=1,8
L1227      DO 6030 LR=2,3
L1228      IF (PRTNS2(SEMPRC(I,4),M,LR).NE.PCCKNO) GO TO 6030
L1229      CRTNS2(SEMPRC(I,4),M,LR)=0
L1230      GO TO 6245
L1231 6030 CONTINUE
L1232      DO 6035 M=1,MNOBFS
L1233      IF (BFPOLR(SEMPRC(I,4),M,1).NE.PCCKNO) GO TO 6035
L1234      BFPOLR(SEMPRC(I,4),M,1)=0
L1235      GO TO 6245
L1236 6035 CONTINUE

```

```
L1237 6245 LLGXZ=PACKET(PCCKNO,2)
L1238      CALL TIMER(LLGX2,MUXRT,RIMR7)
L1239      ID=I+(3*N)+NONCPS
L1240      TIMERZ(ID)=RIMR7+NPZ
L1241      CNZ=2
L1242      CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L1243      DATABS(I)=2
L1244 6250 IF (SIGNTM(I+(3*N)+NONCPS).NE.3) GOTO 6999
L1245      DO 6105 J=1,7
L1246      IF (SEMPRC(I,5).EQ.1) VARTM(J)=PT1ICP(SEMPRC(I,4),NONCC(I),J)
L1247      IF (SEMPRC(I,5).EQ.2) VARTM(J)=PT1NCP(SEMPRC(I,4),NONCC(I),J)
L1248 6105 CONTINUE
L1249      IF (VARTM(5).EQ.1 .AND. PACKET(VARTM(3),3).EQ.1) GOTO 6255
L1250      IF (VARTM(5).EQ.1) REASSM(PACKET(VARTM(3),1),1)=VARTM(3)
L1251      IF (VARTM(5).EQ.2) IFBUFR(VARTM(4))=VARTM(3)
L1252      GOTO 6260
L1253 6255 IOP(PACKET(VARTM(3),1),10)=IOP(PACKET(VARTM(3),1),10)-1
L1254      PACKET(VARTM(3),1)=0
L1255 6260 IF (VARTM(6).EQ.SEMPRC(I,3)) GOTO 6200
L1256      IF (SEMPRC(I,5).EQ.1 .AND. PT1ICP(SEMPRC(I,4),NONCC(I),8).EQ.1) NON
CONT: C(I)=NONCC(I)+1
L1257      IF (SEMPRC(I,5).EQ.2 .AND. PT1NCP(SEMPRC(I,4),NONCC(I),8).EQ.1) NON
CONT: C(I)=NONCC(I)+1
L1258      SEMPRC(I,4)=VARTM(6)
L1259      SEMPRC(I,5)=VARTM(7)
L1260      GO TO 6015
L1261 6200 DATABS(I)=1
L1262 6999 CONTINUE
```

BLOCK: MAIN EXECUTION

MODULE: SWITCHING ALGORITHM

DO FOR EACH NODE

SWCONT (I): PGM CONTROL I: NODAL NUMBER

PROGRAM CONTROL SWCONT (I)

(A) SERVICE REQUESTS AND SORTING

IF SWCONT (I) = 1:

WAIT FOR SYNCHRONIZATION SIGNAL SEMPH = 3

SET SWCONT (I) = 2

RESET SYNCHRONIZATION SIGNAL SEMPH = 2

DETERMINE TOTAL NUMBER OF FREE BUFFER BLOCKS IN EACH
NCP AND STORE IN STO (TRANSMITTER SIDE OF NCP ONLY)

IF SWCONT (I) = 2:

STORE PACKET TRANSFER REQUESTS IN QLIST.

AT SAME TIME DETERMINE PARTITION AND DESTINATION OF
PACKET, AND SELECT APPROPRIATE ICP OR NCP TO TRANSFER
PACKET TO - STORE IN QLIST

IF MORE THAN ONE CHANNEL TO A NEXT NODE - SELECT NCP
WITH MOST FREE BUFFERS, i.e. SMALLEST QUEUE

(B) WEIGHTING FACTOR ALLOCATION

A PACKET IS ALLOCATED A WEIGHTING FACTOR ACCORDING TO
THE CLASS IT BELONGS TO:

ICP-TO-ICP : NON NETWORK TRAFFIC

ICP-TO-NCP : TRAFFIC ENTERING NETWORK

NCP-TO-NCP : TRANSIT TRAFFIC

NCP-TO-ICP : TRAFFIC LEAVING NETWORK

QUEUE PACKET IDENTIFIER (POINTER FROM QLIST) IN Q
ACCORDING TO PRIORITY
(INITIAL PRIORITY SET EQUAL TO WEIGHTING FACTOR)

(C) FORMING SWITCH MATRIX

DETERMINE THE SWITCH MATRIX FOR DATA GIVEN IN QLIST
GIVING PRIORITY TO PACKETS FIRST IN Q

DETERMINE POINTER ALLOCATIONS FOR COMMANDS

INCREMENT PRIORITIES

GENERATE SWITCH ALG EXECUTION INTERVAL SET SWCONT (I) = 3

IF SWCONT (I) = 3:

IF INTERVAL NOT YET ELAPSED WAIT

SET SWCONT (I) = 1

SET SYNCHRONIZATION SIGNAL SEMPH = 1

SET SYNCHRONIZATION SIGNAL BUSSEM = 1

```
L1263 C
L1264 C      SWITCHING ALGORITHM
L1265      NSWC=SWCONT(I)
L1266      GOTO (9050,9080,9720), NSWC
L1267 9050 IF (SEMPH(I).NE.3) GOTO 9750
L1268      SWCONT(I)=2
L1269      SEMPH(I)=2
L1270      DO 9070 KS=1,NMAXIN
L1271      NZA=MAPNCP(I,KS)
L1272      IF (NZA.EQ.0) GOTO 9070
L1273      DO 9060 K=1,8
L1274      IF (PRTNS1(NZA,K,1).NE.0 .AND. PRTNS1(NZA,K,2).EQ.0) STO(NZA,1)=STO
CONT: NZA,1)+1
L1275      IF (PRTNS1(NZA,K,1).NE.0 .AND. PRTNS1(NZA,K,3).EQ.0) STO(NZA,1)=STO
CONT: NZA,1)+1
L1276      PARTNP(NZA,K,2)=PRTNS2(NZA,K,5)
L1277 9060 CONTINUE
L1278      DO 9065 K=1,MNOBFS
L1279      IF (BFP00L(NZA,K,1).EQ.0) STO(NZA,1)=STO(NZA,1)+1
L1280 9065 CONTINUE
L1281      STO(NZA,2)=STO(NZA,1)
L1282 9070 CONTINUE
L1283 9080 DO 9200 J=1,NOPCKS
L1284      IF (NODEST(I,J,1).EQ.0) GOTO 9200
L1285      DO 9100 M=1,QUS
L1286      IF (QLIST(I,M,5).EQ.0) LMB=M
L1287      IF (QLIST(I,M,5).EQ.0) GOTO 9120
L1288 9100 CONTINUE
L1289      WRITE (6,9110)
L1290      GOTO 1111
L1291 9110 FORMAT (T2,'ERROR - QLIST ARRAY OVERFLOW')
L1292 9120 IDNODE=IOP(PACKET(NODEST(I,J,1),1),4)
L1293      IF (PACKET(NODEST(I,J,1),5).NE.1 .AND. PACKET(NODEST(I,J,1),5).NE.7
CONT: IDNODE=IOP(PACKET(NODEST(I,J,1),1),2)
L1294      QLIST(I,LMB,1)=NODEST(I,J,2)
L1295      QLIST(I,LMB,2)=NODEST(I,J,3)
L1296      QLIST(I,LMB,5)=NODEST(I,J,1)
L1297      IF (I.NE.IDNODE) GOTO 9140
L1298      QLIST(I,LMB,3)=IOP(PACKET(NODEST(I,J,1),1),5)
L1299      IF (PACKET(NODEST(I,J,1),5).NE.1 .AND. PACKET(NODEST(I,J,1),5).NE.7
CONT: QLIST(I,LMB,3)=IOP(PACKET(NODEST(I,J,1),1),3)
L1300      QLIST(I,LMB,4)=1
L1301      QLIST(I,LMB,6)=0
L1302      GO TO 9160
L1303 9140 ICNT=0
L1304      NOKAN=MAXCH+1
L1305      DO 9150 M=2,NOKAN
L1306      IF (NODPLT(I,IDNODE,M).EQ.0) GOTO 9150
L1307      IF (STO(NODPLT(I,IDNODE,M),1).GE.ICNT) LMY=M
L1308      IF (STO(NODPLT(I,IDNODE,M),1).GE.ICNT) ICNT=STO(NODPLT(I,IDNODE,M),
CONT: )
L1309 9150 CONTINUE
L1310      STO(NODPLT(I,IDNODE,LMY),1)=STO(NODPLT(I,IDNODE,LMY),1)-1
L1311      QLIST(I,LMB,3)=NODPLT(I,IDNODE,LMY)
L1312      QLIST(I,LMB,4)=2
L1313      QLIST(I,LMB,6)=NODPLT(I,IDNODE,1)
L1314 C
```

```

L1315 C      ICP-ICP:NON NW TRAFFIC

L1316 9160 IF (QLIST(I,LMB,2).EQ.1 .AND. QLIST(I,LMB,4).EQ.1) QLIST(I,LMB,7)=2
L1317 C      ICP-NCP:TRAFFIC ENTERING NW
L1318 IF (QLIST(I,LMB,2).EQ.1 .AND. QLIST(I,LMB,4).EQ.2) QLIST(I,LMB,7)=1.
L1319 C      NCP-NCP:TRANSIT TRAFFIC
L1320 IF (QLIST(I,LMB,2).EQ.2 .AND. QLIST(I,LMB,4).EQ.2) QLIST(I,LMB,7)=5
L1321 C      NCP-ICP:TRAFFIC LEAVING NW
L1322 IF (QLIST(I,LMB,2).EQ.2 .AND. QLIST(I,LMB,4).EQ.1) QLIST(I,LMB,7)=8
L1323 C
L1324 DO 9170 M=1,3
L1325 NODEST(I,J,M)=0
L1326 9170 CONTINUE
L1327 DO 9180 ML=1,QUS
L1328 IF (QLIST(I,LMB,7).GT.Q(I,ML,2)) MK=ML
L1329 IF (QLIST(I,LMB,7).GT.Q(I,ML,2)) GOTO 9185
L1330 9180 CONTINUE
L1331 9185 MX1=LMB
L1332 MX2=QLIST(I,LMB,7)
L1333 DO 9190 M=MK,QUS
L1334 MY1=Q(I,M,1)
L1335 MY2=Q(I,M,2)
L1336 Q(I,M,1)=MX1
L1337 Q(I,M,2)=MX2
L1338 QLIST(I,Q(I,M,1),8)=M
L1339 IF (MY1.EQ.0) GOTO 9200
L1340 MX1=MY1
L1341 MX2=MY2
L1342 9190 CONTINUE
L1343 9200 CONTINUE
L1344 C
L1345 9450 IF (Q(I,1,1).EQ.0) GOTO 9750
L1346 NOCYC=0
L1347 9480 NOCYC=NOCYC+1
L1348 MCNT=0
L1349 DO 9500 K=1,NOI
L1350 DO 9500 J=1,5
L1351 SWITCH(K,J)=0
L1352 9500 CONTINUE
L1353 QCNTR=0
L1354 J=0
L1355 J=J+1
L1356 9510 IF (Q(I,J,1).EQ.0) GOTO 9620
L1357 DO 9520 M=1,NOI
L1358 IF (SWITCH(M,1).EQ.0) KMQ=M
L1359 IF (SWITCH(M,1).EQ.0) GOTO 9540
L1360 IF (SWITCH(M,1).EQ.QLIST(I,Q(I,J,1),1) .AND. SWITCH(M,2).EQ.QLIST(I,
CONT: Q(I,J,1),2)) GO TO 9600
L1361 IF (SWITCH(M,3).EQ.QLIST(I,Q(I,J,1),3) .AND. SWITCH(M,4).EQ.QLIST(I,
CONT: Q(I,J,1),4)) GO TO 9600
L1362 9520 CONTINUE
L1363 9540 IF (QLIST(I,Q(I,J,1),4).EQ.1) GOTO 9575
L1364 DO 9560 MX=1,8
L1365 IF (PARTNP(QLIST(I,Q(I,J,1),3),MX,1).EQ.QLIST(I,Q(I,J,1),6)) LLL=MX
L1366 IF (PARTNP(QLIST(I,Q(I,J,1),3),MX,1).EQ.QLIST(I,Q(I,J,1),6)) GOTO 95
CONT: 70
L1367 9560 CONTINUE
L1368 WRITE (6,9562)
L1369 9562 FORMAT (T2,'ERROR - MAIN PGM PARTITION NUMBER NOT FOUND STM 9570')
L1370 GOTO 1111

```

```
L1371 9570 IF (PARTNP(QLIST(I,Q(I,J,1),3),LLL,2).EQ.1) GOTO 9600
L1372      IF (STO(QLIST(I,Q(I,J,1),3),2).LT.1) GOTO 9600
L1373      STO(QLIST(I,Q(I,J,1),3),2)=STO(QLIST(I,Q(I,J,1),3),2)-1
L1374 9575 DO 9580 M=1,5
L1375      SWITCH(KMQ,M)=QLIST(I,Q(I,J,1),M)
L1376 9580 CONTINUE
L1377      DO 9584 M=1,8
L1378      QLIST(I,Q(I,J,1),M)=0
L1379 9584 CONTINUE
L1380      MKS=QUS-1
L1381      DO 9585 M=J,MKS
L1382      Q(I,M,1)=Q(I,(M+1),1)
L1383      Q(I,M,2)=Q(I,(M+1),2)
L1384      IF (Q(I,M,1).EQ.0) GOTO 9590
L1385      QLIST(I,Q(I,M,1),8)=M
L1386 9585 CONTINUE
L1387 9590 Q(I,QUS,1)=0
L1388      Q(I,QUS,2)=0
L1389      QCNTR=QCNTR+1
L1390      J=J-1
L1391      IF (QCNTR.EQ.PKMUX(I)) GOTO 9620
L1392 9600 J=J+1
L1393      IF (J.LE.QUS) GOTO 9510
L1394 9620 IF (QCNTR.EQ.0) SEMPRC(I,1)=(-100)
L1395      IF (QCNTR.EQ.0) GOTO 9712
L1396      DO 9650 M=1,NOI
L1397      DO 9650 MM=1,5
L1398      IF (SWITCH(M,MM).EQ.0) SWITCH(M,MM)=(-1)
L1399 9650 CONTINUE
L1400 C
L1401      KW1=NOI-1
L1402      DO 9675 JV=1,KW1
L1403      IF (SWITCH((JV+1),1).EQ.(-1)) GOTO 9680
L1404      KW2=JV+1
L1405      DO 9660 JW=KW2,NOI
L1406      IF (SWITCH(JV,3).EQ.SWITCH(JW,1) .AND. SWITCH(JV,4).EQ.SWITCH(JW,2))
CONT:  JU=JW
L1407      IF (SWITCH(JV,3).EQ.SWITCH(JW,1) .AND. SWITCH(JV,4).EQ.SWITCH(JW,2))
CONT:  GOTO 9665
L1408 9660 CONTINUE
L1409      GOTO 9675
L1410 9665 DO 9670 KA=1,5
L1411      KW3=SWITCH((JV+1),KA)
L1412      SWITCH((JV+1),KA)=SWITCH(JU,KA)
L1413      SWITCH(JU,KA)=KW3
L1414 9670 CONTINUE
L1415 9675 CONTINUE
L1416 C
```

- 328 -

```

L1417 C      POINTER ALLOCATION
L1418 9680 KK=NOCYC
L1419      DO 9700 M=1,NOI
L1420      IF (SWITCH(M,3).EQ.(-1)) GO TO 9700
L1421      MCNT=MCNT+1
L1422      IF (MCNT.EQ.1 .AND. NOCYC.EQ.1) SEMPRC(I,1)=SWITCH(M,1)
L1423      IF (MCNT.EQ.1 .AND. NOCYC.EQ.1) SEMPRC(I,2)=SWITCH(M,2)
L1424      IF (MCNT.EQ.1 .AND. NOCYC.EQ.1) LIN1=SWITCH(M,1)
L1425      IF (MCNT.EQ.1 .AND. NOCYC.EQ.1) LIN2=SWITCH(M,2)
L1426      IF (SWITCH(M,2).EQ.2) GO TO 9685
L1427 C
L1428 C      ICP
L1429      PT2ICP(SWITCH(M,1),KK,1)=LIN1
L1430      PT2ICP(SWITCH(M,1),KK,2)=LIN2
L1431      PT2ICP(SWITCH(M,1),KK,3)=SWITCH(M,5)
L1432      PT2ICP(SWITCH(M,1),KK,4)=SWITCH(M,3)
L1433      PT2ICP(SWITCH(M,1),KK,5)=SWITCH(M,4)
L1434      GO TO 9690
L1435 C
L1436 C      NCP
L1437 9685 PT2NCP(SWITCH(M,1),KK,1)=LIN1
L1438      PT2NCP(SWITCH(M,1),KK,2)=LIN2
L1439      PT2NCP(SWITCH(M,1),KK,3)=SWITCH(M,5)
L1440      PT2NCP(SWITCH(M,1),KK,4)=SWITCH(M,3)
L1441      PT2NCP(SWITCH(M,1),KK,5)=SWITCH(M,4)
L1442 C
L1443 9690 IF (MCNT.EQ.1 .AND. NOCYC.EQ.1) GO TO 9695
L1444      IF (MCNT.EQ.1) GO TO 9692
L1445 C
L1446      IF (LIN2.EQ.1) PT2ICP(LIN1,KK,6)=SWITCH(M,1)
L1447      IF (LIN2.EQ.1) PT2ICP(LIN1,KK,7)=SWITCH(M,2)
L1448      IF (LIN2.EQ.2) PT2NCP(LIN1,KK,6)=SWITCH(M,1)
L1449      IF (LIN2.EQ.2) PT2NCP(LIN1,KK,7)=SWITCH(M,2)
L1450      GO TO 9694
L1451 9692 IF (LIN2.EQ.1) PT2ICP(LIN1,KK-1,6)=SWITCH(M,1)
L1452      IF (LIN2.EQ.1) PT2ICP(LIN1,KK-1,7)=SWITCH(M,2)
L1453      IF (LIN2.EQ.2) PT2NCP(LIN1,KK-1,6)=SWITCH(M,1)
L1454      IF (LIN2.EQ.2) PT2NCP(LIN1,KK-1,7)=SWITCH(M,2)
L1455 9694 CONTINUE
L1456 C
L1457      LIN1=SWITCH(M,1)
L1458      LIN2=SWITCH(M,2)
L1459 C
L1460 9695 CONTINUE
L1461 C
L1462      IF (SWITCH(M,2).EQ.1) PT2ICP(SWITCH(M,1),KK,8)=0
L1463      IF (SWITCH(M,2).EQ.2) PT2NCP(SWITCH(M,1),KK,8)=0
L1464      LZID=SWITCH(M,2)
L1465      LZNO=SWITCH(M,1)
L1466      LZCYC=KK
L1467      LZZZ=' '
L1468 C
L1469 9700 CONTINUE
L1470 C
L1471 C      END OF CYCLE
L1472      IF (SWITCH(LZZZ,2).EQ.1) PT2ICP(SWITCH(LZZZ,1),KK,8)=1
L1473      IF (SWITCH(LZZZ,2).EQ.2) PT2NCP(SWITCH(LZZZ,1),KK,8)=1
L1474 C
L1475      IF (NOCYC.EQ.CMAX) GO TO 9710
L1476      NOLEV=0
L1477      GO TO 9480
L1478 C
L1479 C      END OF PERIOD: SET LAST PROCESSOR POINTER TO (-200)
L1480 9710 IF (LZID.EQ.1) PT2ICP(LZNO,LZCYC,6)=(-200)
L1481      IF (LZID.EQ.2) PT2NCP(LZNO,LZCYC,6)=(-200)
L1482 C

```

```
L1483 C      INCREMENT PRIORITIES
L1484 9712 DO 9715 J=1,QUS
L1485        IF (Q(I,J,1).EQ.0) GO TO 9716
L1486        Q(I,J,2)=Q(I,J,2)+QLIST(I,Q(I,J,1),7)
L1487        M1=J
L1488        N1=J
L1489 9714 M1=M1-1
L1490        IF (M1.EQ.0) GOTO 9715
L1491        IF (Q(I,N1,2).LE.Q(I,M1,2)) GOTO 9715
L1492        L1=Q(I,M1,1)
L1493        L2=Q(I,M1,2)
L1494        Q(I,M1,1)=Q(I,N1,1)
L1495        Q(I,M1,2)=Q(I,N1,2)
L1496        QLIST(I,Q(I,M1,1),8)=M1
L1497        Q(I,N1,1)=L1
L1498        Q(I,N1,2)=L2
L1499        QLIST(I,Q(I,N1,1),8)=N1
L1500        N1=N1-1
L1501        GO TO 9714
L1502 9715 CONTINUE
L1503 C
L1504 9716 ID=I+(4*N)+NONCPS+NONDS
L1505        TIMERZ(ID)=SWPROC
L1506        CN7=2
L1507        CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L1508        SWCONT(I)=3
L1509 9720 IF (SIGNTM(I+(4*N)+NONCPS+NONDS).NE.3) GOTO 9750
L1510        SWCONT(I)=1
L1511        SEMPH(I)=1
L1512        BUSSEM(I)=1
L1513 9750 CONTINUE
L1514 9999 CONTINUE
L1515 1111 CONTINUE
L1516        STOP
L1517        END
L1518 C
```

PROGRAM SUBROUTINE : ACCESS

RLQ = 1, 2, 3, 4 : PROGRAM CONTROL VARIABLE

NUMBER: IS NUMBER OF BUFFER BLOCKS TO BE RELEASED OR ACCESSED

LINKNO: IS LOGICAL LINK NUMBER

ICPNO: IS ICP NUMBER

TYP: IS TYPE OF BUFFERS REQUIRED OR RELEASED

ACCSS: SIGNAL: = 1 BUFFERS GRANTED
 = 2 BUFFERS REFUSED

SDEOM: SIGNAL: = 1 ALL BUFFERS RELEASED BY LINK
 = 2 BUFFERS STILL OCCUPIED

NCO: NUMBER OF COMMON BUFFER BLOCKS FREE

NSI: NUMBER OF SINGLE PACKET BUFFER BLOCKS FREE

NSO: NUMBER OF SOURCE PACKET BUFFER BLOCKS FREE

NMU: NUMBER OF MULTIPACKET BUFFER BLOCKS FREE

SUBROUTINE ACCESS (RLQ, NUMBR, LINKNO, ICPNO, TYP, ACCSS,
 SDEOM, NCO, NSO, NSI, NMU)

PROGRAM CONTROL VARIABLE: RLQ

IF RLQ = 1:

RELEASE OF BUFFER SPACE AT ICP

FIND LOCATIONS OF LINK NUMBER AND TYPE OF BUFFERS ACQUIRED

RELEASE THESE

IF RLQ = 2:

REQUEST FOR BUFFER SPACE AT ICP

DETERMINE WHETHER TOTAL NUMBER OF REQUIRED BUFFER BLOCKS
CAN BE ALLOCATED.

IF NOT SIGNAL ACCSS = 2

OTHERWISE STORE RELEVANT LOGICAL LINK AND BUFFER TYPE
PARAMETERS AND SIGNAL ACCSS = 1

IF RLQ = 3:

DETERMINE NUMBER OF FREE BUFFER BLOCKS FOR EACH TYPE:
NSO, NSI, NMV, NCO

IF RLQ = 4:

INITIALIZATION OF SIMULATION: SET NUMBER OF BUFFER
BLOCKS FOR EACH TYPE

END

```
L1519 C
L1520 C      PROGRAM SUBROUTINES
L1521      SUBROUTINE ACCESS(RLQ,NUMBR,LINKNO,ICPNO,TYP,ACCSS,SDEOM,NC0,NS0,NSI
CONT: ,NMU)
L1522      INTEGER BLOCK(10,8), ALLOC(10,20,4)
L1523      INTEGER LFTTYP,LFTCOM,TOTAL,VAR1,VAR2,RLQ,NUMBR,LINKNO
L1524      INTEGER ICPNO,TYP,ACCSS,SDEOM,NTF,NC
L1525      INTEGER RELTYP,RELCOM,NOTAL,VAR3,VAR4
L1526 C
L1527 C      NUMBER OF ICPS IN NETWORK
L1528      NOICPS=10
L1529 C
L1530 C      NUMBER OF PORTS IN NETWORK
L1531      N=20
L1532 C
L1533      GO TO (100,200,300,400), RLQ
L1534 C
L1535 C      RELEASE OF BUFFER SPACE AT ICP
L1536 100  I=ICPNO
L1537      DO 105 J=1,N
L1538      IF (ALLOC(I,J,1).NE.LINKNO .OR. ALLOC(I,J,2).NE.TYP) GO TO 105
L1539      RELTYP=ALLOC(I,J,3)
L1540      RELCOM=ALLOC(I,J,4)
L1541      NOTAL=RELTYP+RELCOM
L1542      IF (NUMBR.GT.NOTAL) WRITE (6,445)
L1543      VAR3=NUMBR-RELCOM
L1544      VAR4=RELCOM
L1545      ALLOC(I,J,3)=ALLOC(I,J,3)-VAR3
L1546      ALLOC(I,J,4)=ALLOC(I,J,4)-VAR4
L1547      IF (ALLOC(I,J,3).LT.0 .OR. ALLOC(I,J,4).LT.0) WRITE (6,450)
L1548      SDEOM=2
L1549      IF (ALLOC(I,J,3).EQ.0 .AND. ALLOC(I,J,4).EQ.0) SDEOM=1
L1550      IF (ALLOC(I,J,3).EQ.0 .AND. ALLOC(I,J,4).EQ.0) ALLOC(I,J,1)=0
L1551      IF (ALLOC(I,J,3).EQ.0 .AND. ALLOC(I,J,4).EQ.0) ALLOC(I,J,2)=0
L1552      GO TO 110
L1553 105  CONTINUE
L1554      WRITE (6,455)
L1555 110  J=TYP
L1556      K=TYP+4
L1557      BLOCK(I,J)=BLOCK(I,J)-VAR3
L1558      BLOCK(I,4)=BLOCK(I,4)-VAR4
L1559      IF (BLOCK(I,J).LT.0 .OR. BLOCK(I,4).LT.0) WRITE (6,460)
L1560      GO TO 435
L1561
```

```
L1562 C
L1563 C      REQUEST FOR BUFFER SPACE AT ICF
L1564 200    I=ICPNO
L1565        J=TYP
L1566        K=TYP+4
L1567        LFTTYP=BLOCK(I,K)-BLOCK(I,J)
L1568        LFTCOM=BLOCK(I,8)-BLOCK(I,4)
L1569        TOTAL=LFTTYP+LFTCOM
L1570        IF (NUMBR. .TOTAL) GO TO 220
L1571        VAR1=LFTTYP-NUMBR
L1572        IF (VAR1.GE.0) VAR2=0
L1573        IF (VAR1.GE.0) VAR1=NUMBR
L1574        IF (VAR1.LT.0) VAR2=- (VAR1)
L1575        IF (VAR1.LT.0) VAR1=LFTTYP
L1576        BLOCK(I,J)=BLOCK(I,J)+VAR1
L1577        BLOCK(I,4)=BLOCK(I,4)+VAR2
L1578 C      HERE J IS NUMBER OF LOGICAL LINKS FOR ALLOC(ARRAY)
L1579        DO 205 J=1,N
L1580        IF (ALLOC(I,J,1).NE.LINKNO .OR. ALLOC(I,J,2).NE.TYP) GO TO 205
L1581        ALLOC(I,J,3)=ALLOC(I,J,3)+VAR1
L1582        ALLOC(I,J,4)=ALLOC(I,J,4)+VAR2
L1583        ACCSS=1
L1584        GO TO 225
L1585 205    CONTINUE
L1586        DO 215 J=1,N
L1587        IF (ALLOC(I,J,1).NE.0) GO TO 215
L1588 210    ALLOC(I,J,1)=LINKNO
L1589        ALLOC(I,J,2)=TYP
L1590        ALLOC(I,J,3)=VAR1
L1591        ALLOC(I,J,4)=VAR2
L1592        ACCSS=1
L1593        GO TO 225
L1594 215    CONTINUE
L1595        WRITE (6,440)
L1596 220    ACCSS=2
L1597 225    GO TO 435
L1598 C
```

```
L1599 C      NUMBER OF BLOCKS LEFT IN ICP MEMORY
L1600 300     I=ICPNO
L1601        NSO=BLOCK(I,5)-BLOCK(I,1)
L1602        NSI=BLOCK(I,6)-BLOCK(I,2)
L1603        NMU=BLOCK(I,7)-BLOCK(I,3)
L1604        NCO=BLOCK(I,8)-BLOCK(I,4)
L1605        GO TO 435
L1606 C
L1607 C      INITIALIZATION OF ICP MEMORY
L1608 400     DO 410 I=1,NOICPS
L1609        DO 410 J=1,4
L1610        BLOCK(I,J)=0
L1611 410     CONTINUE
L1612        DO 415 I=1,NOICPS
L1613        BLOCK(I,5)=8
L1614        BLOCK(I,6)=5
L1615        BLOCK(I,7)=8
L1616        BLOCK(I,8)=50
L1617 415     CONTINUE
L1618        DO 420 I=1,NOICPS
L1619        DO 420 J=1,N
L1620        DO 420 K=1,4
L1621        ALLOC(I,J,K)=0
L1622 420     CONTINUE
L1623 C
L1624 C      END OF SUBROUTINE ACCESS
L1625 435     CONTINUE
L1626 440     FORMAT(T2,'ERROR NO1')
L1627 445     FORMAT(T2,'ERROR NO2')
L1628 450     FORMAT(T2,'ERROR NO3')
L1629 455     FORMAT(T2,'ERROR NO4')
L1630 460     FORMAT(T2,'ERROR NO5')
L1631        RETURN
L1632        END
L1633 C
```

PROGRAM SUBROUTINE: RESERV

STAT = 3, 4, 5, 7 : PROGRAM CONTROL VARIABLE

ICPRES: ICP NUMBER

LNKRES: LOGICAL LINK NUMBER

NOBUF: NUMBER OF BUFFERS REQUIRED

CLK: CLOCK VALUE

RESU: RESULT - 1 REQUEST GRANTED
- 2 REQUEST REFUSED

TP: TYPE OF BUFFERS REQUIRED

SUBROUTINE RESERV (STAT, ICPRES, LNKRES, NOBUF, CLK, RESU, TP)

PROGRAM CONTROL VARIABLE: STAT

IF STAT = 3:

REQUEST TO BE PLACED ON WAITING LIST FOR ICP BUFFERS

IF REQUEST GRANTED, STORE ALL RELEVANT PARAMETERS IN WAITLN
AND SIGNAL RESU = 1

IF REQUEST REFUSED - WAITING LIST LOCATIONS ALL OCCUPIED -
SIGNAL RESU = 2

IF STAT = 4:

DETERMINE PARAMETERS OF A LOGICAL LINK THAT IS WAITING FOR BUFFERS,

i.e., NOBUF : NUMBER OF BUFFERS REQUIRED

CLX : TIME AT WHICH REQUEST WAS MADE

TP : TYPE OF BUFFERS REQUIRED

RESU = 3: DUMMY VARIABLE

IF STAT = 5

INITIALIZATION OF SUBROUTINE

SET ALL WAITING LIST LOCATIONS TO ZERO (CONTENTS THEREOF)

IF STAT = 7

THIS SECTION EXECUTES THE WAITING-STATE ALGORITHM THAT
DETERMINES WHICH LOGICAL LINKS ARE TO OBTAIN THE
REQUIRED BUFFERS DEPENDING ON THE TYPE OF BUFFERS SOUGHT

THIS SECTION IS EXECUTED EVERY PERIOD IF THERE ARE REQUESTS
CALL SUEER ACCESS TO DETERMINE NO OF FREE BUFFERS OF
EACH TYPE

NCR : NO OF COMMON

NSOV : NO OF SOURCE

NSING : NO OF SINGLE PACKET

NMUL : NO OF MULTIPACKET

TYPES OF REQUESTS:

SOURCE BLOCKS: NSOU + NCR

SINGLE PACKET BLOCKS: NSING + NCR

MULTIPACKET BLOCKS: NMUL + NCR

LMA (I): I = ICP NUMBER

: REQUEST BEING SERVICED - LOGICAL LINK

ONCE A REQUEST IS CHOSEN FOR SERVICE, NO OTHER LOGICAL LINK
IN THE ICP IS SERVICED UNTIL THE LINK BEING SERVICED HAS
BEEN ALLOCATED ALL ITS BUFFERS

LMA (I) = 0: NO REQUEST SERVICED

DETERMINE REQUEST WITH HIGHEST PRIORITY AND SERVICE IT
PROVIDED THERE ARE SOME BUFFERS FREE IN ITS TYPE

i.e. (NCR + NMUC) $\neq$ 0

OR (NCR + NSOU) $\neq$ 0

OR (NCR + NSING) $\neq$ 0

ALLOCATE THE AVAILABLE BUFFERS

DECREASE TYPE (NCR + NMUL)

(NCR + SOU)

(NCR + NSING)

ACCORDING TO NUMBER OF BLOCKS ALLOCATED

CALL ACCESS SUBROUTINE TO ALLOCATE BLOCKS.

SIGNAL: LGLINK = 1 THAT REQUEST HAS BEEN GRANTED

GO BACK AND REPEAT FOR A NEW REQUEST

SIGNAL: LGLINK = 2 FOR OTHER LOGICAL LINKS

INDICATING THAT SERVICE REQUESTS STILL WAITING

INCREASE PRIORITIES OF REQUESTS

NPRSIN: SINGLE PACKET PRIORITY INCREMENT

NPRMUL: MULTIPACKET PRIORITY INCREMENT

NPRSOU: SOURCE PRIORITY INCREMENT

END

```
L1634 C
L1635 C
L1636 SUBROUTINE RESERV(STAT,ICPRES,LNKRES,NOBUF,CLK,RESU,TP)
L1637 INTEGER SIGNTM(130),LGLINK(20,2)
L1638 INTEGER WAITLN(10,10,5), LMA(10)
L1639 INTEGER STAT,ICPRES,LNKRES,NOBUF,CLK,RESU,TP
L1640 INTEGER RST,NRB,LINKNR,ICPNR,TNR,ACC,SDM,NTR,NCR
L1641 COMMON SIGNTM,LGLINK
L1642 I=ICPRES
L1643 NRSV=10
L1644 NOICPS=10
L1645 NOLNK=20
L1646 GO TO (800,800,300,400,500,800,700), STAT
L1647 C WAITING LIST FOR ICP BUFFERS
L1648 300 DO 305 J=1,NRSV
L1649 IF (WAITLN(I,J,1).NE.0) GO TO 305
L1650 WAITLN(I,J,1)=LNKRES
L1651 WAITLN(I,J,2)=NOBUF
L1652 WAITLN(I,J,3)=CLK
L1653 WAITLN(I,J,4)=TP
L1654 WAITLN(I,J,5)=0
L1655 RESU=1
L1656 GO TO 800
L1657 305 CONTINUE
L1658 RESU=2
L1659 GOTO 800
L1660 C
L1661 400 DO 405 J=1,NRSV
L1662 IF (WAITLN(I,J,1).NE.LNKRES) GO TO 405
L1663 NOBUF=WAITLN(I,J,2)
L1664 CLK=WAITLN(I,J,3)
L1665 TP=WAITLN(I,J,4)
L1666 DO 404 KRS=1,5
L1667 WAITLN(I,J,KRS)=0
L1668 404 CONTINUE
L1669 RESU=3
L1670 GO TO 800
L1671 405 CONTINUE
L1672 WRITE (6,810)
L1673 C
L1674 C INITIALIZATION OF SUBROUTINE
L1675 500 DO 505 I=1,NOICPS
L1676 LMA(I)=0
L1677 DO 505 J=1,NRSV
L1678 DO 505 K=1,5
L1679 WAITLN(I,J,K)=5555555
L1680 505 CONTINUE
L1681 GO TO 800
L1682 C
```

- 339 -

```

L1683 C   WAITING STATE ALGORITHM
L1684 700 I=ICPRES
L1685 710 RST=3
L1686     CALL ACCESS(RST,NRB,LINKNR,I,TNR,ACC,SDM,NCR,NSOU,NSING,NMUL)
L1687     IF (NCR.EQ.0 .AND. NSOU.EQ.0 .AND. NSING.EQ.0 .AND. NMUL.EQ.0) GOTO
CONT: 790
L1688     LPS=0
L1689     IF (LMA(I).NE.0) GOTO 775
L1690     IF ((NCR+NSOU).EQ.0) GO TO 730
L1691     DO 720 J=1,NRSV
L1692     IF (WAITLN(I,J,1).NE.0 .AND. WAITLN(I,J,4).EQ.1 .AND. WAITLN(I,J,5).
CONT: GE.LPS) LMA(I)=J
L1693     IF (WAITLN(I,J,1).NE.0 .AND. WAITLN(I,J,4).EQ.1 .AND. WAITLN(I,J,5).
CONT: GE.LPS) LPS=WAITLN(I,J,5)
L1694 720 CONTINUE
L1695 730 IF ((NCR+NMUL).EQ.0) GO TO 750
L1696     DO 740 J=1,NRSV
L1697     IF (WAITLN(I,J,1).NE.0 .AND. WAITLN(I,J,4).EQ.3 .AND. WAITLN(I,J,5).
CONT: GE.LPS) LMA(I)=J
L1698     IF (WAITLN(I,J,1).NE.0 .AND. WAITLN(I,J,4).EQ.3 .AND. WAITLN(I,J,5).
CONT: GE.LPS) LPS=WAITLN(I,J,5)
L1699 740 CONTINUE
L1700 750 IF ((NCR+NSING).EQ.0) GO TO 770
L1701     DO 760 J=1,NRSV
L1702     IF (WAITLN(I,J,1).NE.0 .AND. WAITLN(I,J,4).EQ.2 .AND. WAITLN(I,J,5).
CONT: GE.LPS) LMA(I)=J
L1703     IF (WAITLN(I,J,1).NE.0 .AND. WAITLN(I,J,4).EQ.2 .AND. WAITLN(I,J,5).
CONT: GE.LPS) LPS=WAITLN(I,J,5)
L1704 760 CONTINUE
L1705 770 IF (LMA(I).EQ.0) GO TO 790
L1706 775 RST=2
L1707     ICPNR=I
L1708     LINKNR=WAITLN(I,LMA(I),1)
L1709     IF (WAITLN(I,LMA(I),4).EQ.1) GO TO 761
L1710     IF (WAITLN(I,LMA(I),4).EQ.2) GO TO 762
L1711     IF (WAITLN(I,LMA(I),4).EQ.3) GO TO 763
L1712 761 NZLFT=(NCR+NSOU)-WAITLN(I,LMA(I),2)
L1713     IF (NZLFT.GE.0) NRB=WAITLN(I,LMA(I),2)
L1714     IF (NZLFT.LT.0) NRB=(NCR+NSOU)
L1715     TNR=1
L1716     GO TO 765
L1717 762 NZLFT=(NCR+NSING)-WAITLN(I,LMA(I),2)
L1718     IF (NZLFT.GE.0) NRB=WAITLN(I,LMA(I),2)
L1719     IF (NZLFT.LT.0) NRB=(NCR+NSING)
L1720     TNR=2
L1721     GO TO 765
L1722 763 NZLFT=(NCR+NMUL)-WAITLN(I,LMA(I),2)
L1723     IF (NZLFT.GE.0) NRB=WAITLN(I,LMA(I),2)
L1724     IF (NZLFT.LT.0) NRB=(NCR+NMUL)
L1725     TNR=3
L1726 765 WAITLN(I,LMA(I),2)=WAITLN(I,LMA(I),2)-NRB
L1727     CALL ACCESS(RST,NRB,LINKNR,ICPNR,TNR,ACC,SDM,NCR,NSOU,NSING,NMUL)
L1728     LGLINK(WAITLN(I,LMA(I),1),1)=1
L1729     LGLINK(WAITLN(I,LMA(I),1),2)=1
L1730     IF (WAITLN(I,LMA(I),2).EQ.0 .AND. WAITLN(I,LMA(I),4).EQ.1) LGLINK(W
CONT: ITLN(I,LMA(I),1),1)=2
L1731     IF (WAITLN(I,LMA(I),2).EQ.0 .AND. WAITLN(I,LMA(I),4).EQ.2) LGLINK(W
CONT: ITLN(I,LMA(I),1),2)=2
L1732     IF (WAITLN(I,LMA(I),2).EQ.0 .AND. WAITLN(I,LMA(I),4).EQ.3) LGLINK(W
CONT: ITLN(I,LMA(I),1),2)=2
L1733     IF (WAITLN(I,LMA(I),2).EQ.0) WAITLN(I,LMA(I),1)=0
L1734     IF (WAITLN(I,LMA(I),2).EQ.0) LMA(I)=0
L1735     IF (WAITLN(I,LMA(I),2).EQ.0) GOTO 710

```

```
L1736 C
L1737 C      NPRSIN=SINGLE PACKET PRIORITY INCREMENT
L1738 C      NPRMUL=MULTIPACKET PRIORITY INCREMENT
L1739 C      NPRSOU=SOURCE PRIORITY INCREMENT
L1740 790    NPRSIN=1
L1741        NPRMUL=1
L1742        NPRSOU=1
L1743        DO 795 J=1,NRSV
L1744        IF (WAITLN(I,J,1).EQ.0) GO TO 795
L1745        IF (WAITLN(I,J,4).EQ.1) WAITLN(I,J,5)=WAITLN(I,J,5)+NPRSOU
L1746        IF (WAITLN(I,J,4).EQ.2) WAITLN(I,J,5)=WAITLN(I,J,5)+NPRSIN
L1747        IF (WAITLN(I,J,4).EQ.3) WAITLN(I,J,5)=WAITLN(I,J,5)+NPRMUL
L1748 795    CONTINUE
L1749        GO TO 800
L1750 C
L1751 C      END OF SUBROUTINE
L1752 800    CONTINUE
L1753 805    FORMAT (T2,'ERROR NO6')
L1754 810    FORMAT (T2,'ERROR NO7')
L1755        RETURN
L1756        END
L1757 C
```

PROGRAM SUBROUTINE: PROGTM

TIMERZ : TIME INTERVAL

CLOCK : CLOCK VALUE

CNZ : 1, 2, 3, 4, 5, 6 : PROGRAM CONTROL

ID : IDENTIFIER

SUBROUTINE PROGTM (TIMERZ, CLOCK, CNZ, ID)

IF CNZ = 1:

NO OPERATION

IF CNZ = 2:

INTERVAL GENERATION

THE INTERVAL TIMERZ GENERATED BY IDENTIFIER ID
IS STORED IN ORDER OF EVENTS

IF CNZ = 3:

INTERVAL DELETION

REMOVE THE INTERVAL FROM THE QUEUE AS IDENTIFIED BY ID

IF CNZ = 4:

INTERVAL UPDATE

EVERY PERIOD THE INTERVAL AT THE HEAD-OF-QUEUE
BECOMES THE NEW CLOCK VALUE

IF CNZ = 5:

TRANSFER CLOCK VALUE TO MAIN PGM

IF CNZ = 6:

INITIALIZATION OF TIMER AND IDENTIFIER QUEUES

END

PROGRAM SUBROUTINE: TIMER

LNBITS : INTEGER, LENGTH OF MSG IN BITS

SPDBITS : INTEGER, LINE RATE IN BPS

CONVERTS ABOVE VARIABLES TO REAL AND DIVIDES TO
DETERMINE TIME TAKEN FOR MSG TO TRAVERSE CHANNEL

RSTV : REAL, DURATION

SUBROUTINE TIMER (LNBITS, SPDBITS, RSTV)

```
L1758 C      INTERVAL TIMER CONTROL
L1759          SUBROUTINE PROGTM(TIMERZ,CLOCK,CNZ,ID)
L1760          REAL TIMERZ(130), EVENT(130), CLOCK
L1761          INTEGER SIGNTM(130), LGLINK(20,2), IDTINV(130) , CNZ, ID
L1762          COMMON SIGNTM,LGLINK
L1763 C
L1764 C      NOLABL: NUMBER OF IDENTIFIERS
L1765          NOLABL=130
L1766 C
L1767          GO TO (100,200,300,400,500,600), CNZ
L1768 C
L1769 C      IDLE
L1770 100      CONTINUE
L1771          GO TO 700
L1772 C
L1773 C      INTERVAL GENERATION
L1774 200      SIGNTM(ID)=2
L1775          RESUL=CLOCK+TIMERZ(ID)
L1776          DO 210 I=1,NOLABL
L1777              IF (RESUL.LT.EVENT(I)) MKR=I
L1778              IF (RESUL.LT.EVENT(I)) GO TO 230
L1779 210      CONTINUE
L1780          WRITE (6,220)
L1781 220      FORMAT (T2,'ERROR - INTERVAL GENERATION  STM 220  SUBR:PROGTM')
L1782 230      IF (MKR.EQ.NOLABL) GO TO 260
L1783          XX1=RESUL
L1784          MX1=ID
L1785          DO 240 I=MKR,NOLABL
L1786              XY1=EVENT(I)
L1787              MY1=IDTINV(I)
L1788              EVENT(I)=XX1
L1789              IDTINV(I)=MX1
L1790              XX1=XY1
L1791              MX1=MY1
L1792 240      CONTINUE
L1793 260      EVENT(MKR)=RESUL
L1794          IDTINV(MKR)=ID
L1795          GO TO 700
L1796 C
L1797 C      INTERVAL DELETION
L1798 300      DO 310 K=1,NOLABL
L1799          IF (IDTINV(K).NE.ID) GOTO 310
L1800          KMR=K
L1801          GOTO 330
L1802 310      CONTINUE
L1803          GOTO 700
L1804 330      KMZ=NOLABL-1
L1805          IF (KMR.EQ.NOLABL) GOTO 360
L1806          DO 340 K=KMR,KMZ
L1807              EVENT(K)=EVENT(K+1)
L1808              IDTINV(K)=IDTINV(K+1)
L1809 340      CONTINUE
L1810 360      EVENT(NOLABL)=1.0E+75
L1811          IDTINV(NOLABL)=0
L1812          GOTO 700
L1813 C
```

```
L1814 C      INTERVAL UPDATE
L1815 400     IF (CLOCK.GT.1.0E+73) WRITE (6,410)
L1816 410     FORMAT (T2,'ERROR - MAXIMUM CLOCK VALUE EXCEEDED')
L1817        IF (EVENT(1).GT.1.0E+74) GO TO 700
L1818        CLOCK=EVENT(1)
L1819 420     SIGNTM(IDTINV(1))=3
L1820        KSM=NOLABL-1
L1821        DO 430 I=1,KSM
L1822        EVENT(I)=EVENT(I+1)
L1823        IDTINV(I)=IDTINV(I+1)
L1824 430     CONTINUE
L1825        EVENT(NOLABL)=1.0E+75
L1826        IDTINV(NOLABL)=0
L1827        IF (EVENT(1).EQ.CLOCK) GO TO 420
L1828        GO TO 700
L1829 C
L1830 C      CLOCK VALUE
L1831 500     CLOCK=CLOCK
L1832        GO TO 700
L1833 C
L1834 C      INITIALIZATION
L1835 600     DO 610 I=1,NOLABL
L1836        EVENT(I)=1.0E+75
L1837        IDTINV(I)=0
L1838        TIMERZ(I)=0.0
L1839        SIGNTM(I)=1
L1840 610     CONTINUE
L1841        CLOCK=0.0
L1842        GO TO 700
L1843 C
L1844 700     CONTINUE
L1845        RETURN
L1846        END
L1847 C
L1848 C
L1849        SUBROUTINE TIMER(LNBITS,SPDBTS,RSTV)
L1850        INTEGER LNBITS,SPDBTS
L1851        REAL BITS,BPS,RSTV
L1852        BITS=LNBITS
L1853        BPS=SPDBTS
L1854        RSTV=(BITS/BPS)
L1855        RETURN
L1856        END
```

```

L1148 C *****
L1149 IF (INTERV.EQ.1000 .OR. INTERV.EQ.2000 .OR. INTERV.EQ.3000 .OR. IN
CONT: RV.EQ.4000) GOTO 1198
L1150 IF (INTERV.EQ.5000 .OR. INTERV.EQ.6000) GOTO 1198
L1151 GOTO 1199
L1152 1198 CONTINUE
L1153 WRITE (6,7753) ICPICP,NCPNCP
L1154 7753 FORMAT (T2,'ICPICP,NCPNCP',T30,2I8)
L1155 WRITE (6,1119) (LCVT(I),I=1,6)
L1156 1119 FORMAT (T2,'NODE THROUGHPUT',T20,6I8)
L1157 WRITE (6,1118) AVERAG,VARIAN,NUMAVG
L1158 1118 FORMAT (T2,'MEAN,MEANSQUARE,NUMBER',T35,2F15.4,T70,I8)
L1159 WRITE (6,9991) BMEAN,BVAR,NBEG
L1160 9991 FORMAT (T2,'SOURC:MEAN,MEAN2,N0',T35,2F15.4,T70,I8)
L1161 CNZ=5
L1162 CALL PROGTM(TIMERZ,CLOCK,CNZ,ID)
L1163 WRITE (6,1114) CLOCK
L1164 1114 FORMAT (T2,'CLOCK VALUE',T30,F16.4)
L1165 WRITE (6,1171) (NOCHL(I),I=1,7)
L1166 WRITE (6,1172) (NOCHL(I),I=8,14)
L1167 WRITE (6,1176) (NOCHL(I),I=15,20)
L1168 1176 FORMAT (T2,6I8)
L1169 WRITE (6,1173) PMEANC
L1170 1173 FORMAT (T2,'PMEANC',T20,1F15.3)
L1171 1171 FORMAT (T2,7I8)
L1172 1172 FORMAT (T2,7I8)
L1173 WRITE (6,1178)
L1174 1178 FORMAT (T2,///)
L1175 1199 CONTINUE
L1176 C *****

```

ERROR MESSAGES

Subroutine ACCESS

'ERROR NO2' : Attempt to release more blocks than
were allocated

'ERROR NO3' : More blocks released than allocated
for a logical link (ALLOC ARRAY)

'ERROR NO4' : Logical link number for which buffer
release requested not found

'ERROR NO5' : More blocks released than allocated
for a logical link (BLOCK ARRAY)

Subroutine RESERV

'ERROR NO7' : Logical link number for which reservation
requested, not found

Subroutine PROGTM

'ERROR-INTERVAL GENERATION' : Identifier unable to be
stored in event queue

'MAXIMUM CLOCK VALUE EXCEEDED' : Simulation clock value
greater than 1,0E + 74 seconds

Main Program

'ERROR-PACKET ARRAY OVERFLOW' :

'ERROR-NODEST ARRAY OVERFLOW' :

'ERROR-QLIST ARRAY OVERFLOW' :

Above arrays are used to store the simulation data. An overflow indicates that the dimensions of the array were not enough to store all the data. Note that the number of locations used by the program may vary during the simulation.

'ERROR-NO OF DEVIATES FOR RANDOM NO GEN EXC' :

The simulation has used its quota of deviates used for generating messages with random lengths and interarrival times.

'ERROR NO8' : The value of variable ACC when calling subroutine ACCESS was neither one nor two (buffers granted, refused resp.).

'ERROR NO9' : The value of variable RES when calling subroutine RESERV was neither one nor two (requested granted, refused resp.).

Internodal Packet Transmission

'ERROR-PRESENT NODE NOT FOUND' : present nodal number cannot
be found in MAPNCP array.

'ERROR-DUPLICATION PROCEDURE' : duplicates of a packet are
found in array PACKET, which is supposed to
contain one copy only of each packet.

'ERROR NCP SIM 1' : Packet partition number cannot be found in NCP

'ERROR NCP SIM 2' : No storage for packet at NCP. A packet
trm from the data bus to the NCP under control
of the switch processor must always find free
buffer blocks.

'ERROR-TRANSMISSION FIELDS' : Trm field flags are set to
invalid values.

'ERROR-FRAME TRANSMISSION' : Receiver field flags are set to
invalid values.

'ERROR-PACKET (CHANNEL) RELEASE' : Acknowledged packet cannot
be found for release of buffers.

'ERROR-PARTITION MISSING' : Partition number at receiver
cannot be found.

'ERROR-SYNCHRONIZATION' : Packet field flags and NCP transmitter
field flags do not make a valid combination.

'ERROR-NODAL NUMBER NOT FOUND' : nodal number to which NCP
belongs not found in MAPNCP array.

'ERROR-FRAME ACKNOWLEDGEMENT' : Acknowledgement flag settings
are invalid.

A listing for the topologies of three networks is contained in the following pages. This list is the input data necessary to define and to simulate the networks.

NET1 defines the topology for the Network Simulation, NODD for the Nodal Simulation, and IOPD for a network used to conduct tests on the source node-to-destination node protocols. These tests were conducted and the results displayed in Chapter 4.

10 NET1

LD Ref	1	5000	2	500	8	4000	5	40	14
L.001	500	100	1	16	70	12	1	450	1
L.002	8	4	250	50					
L.003	1	1	1	7	7200	7200	400	00	0
L.004	1	1	1	7	7200	7200	400	00	0
L.005	1	2	1	8	7200	7200	400	00	0
L.006	1	1	2	9	7200	7200	400	00	0
L.007	1	2	2	10	7200	7200	400	00	0
L.008	1	3	3	17	7200	7200	400	00	0
L.009	1	4	3	18	7200	7200	400	00	0
L.010	1	5	4	25	7200	7200	400	00	0
L.011	1	6	4	26	7200	7200	400	00	0
L.012	1	7	5	33	7200	7200	400	00	0
L.013	1	8	5	34	7200	7200	400	00	0
L.014	2	9	1	1	7200	7200	400	00	0
L.015	2	10	1	2	7200	7200	400	00	0
L.016	2	9	2	15	7200	7200	400	00	0
L.017	2	10	2	16	7200	7200	400	00	0
L.018	2	11	3	19	7200	7200	400	00	0
L.019	2	12	3	20	7200	7200	400	00	0
L.020	2	13	4	27	7200	7200	400	00	0
L.021	2	14	4	28	7200	7200	400	00	0
L.022	2	15	5	35	7200	7200	400	00	0
L.023	2	16	5	36	7200	7200	400	00	0
L.024	3	17	1	3	7200	7200	400	00	0
L.025	3	18	1	4	7200	7200	400	00	0
L.026	3	17	2	11	7200	7200	400	00	0
L.027	3	18	2	12	7200	7200	400	00	0
L.028	3	19	3	23	7200	7200	400	00	0
L.029	3	20	3	24	7200	7200	400	00	0
L.030	3	21	4	29	7200	7200	400	00	0
L.031	3	22	4	30	7200	7200	400	00	0
L.032	3	23	5	37	7200	7200	400	00	0
L.033	3	24	5	38	7200	7200	400	00	0
L.034	4	25	1	5	7200	7200	400	00	0
L.035	4	26	1	6	7200	7200	400	00	0
L.036	4	25	2	13	7200	7200	400	00	0
L.037	4	26	2	14	7200	7200	400	00	0
L.038	4	27	3	21	7200	7200	400	00	0
L.039	4	28	3	22	7200	7200	400	00	0
L.040	4	29	4	31	7200	7200	400	00	0
L.041	4	30	4	32	7200	7200	400	00	0
L.042	4	31	5	39	7200	7200	400	00	0
L.043	4	32	5	40	7200	7200	400	00	0
L.044	5	33	1	7	7200	7200	400	00	0
L.045	5	34	1	8	7200	7200	400	00	0
L.046	5	33	2	15	7200	7200	400	00	0
L.047	5	34	2	16	7200	7200	400	00	0
L.048	5	35	3	23	7200	7200	400	00	0
L.049	5	36	3	24	7200	7200	400	00	0
L.050	5	37	4	31	7200	7200	400	00	0
L.051	5	38	4	32	7200	7200	400	00	0
L.052	5	39	5	33	7200	7200	400	00	0
L.053	5	40	5	34	7200	7200	400	00	0
L.054									

L.055	3	5	0	0	0	0	0	0
L.056	4	0	0	0	0	0	0	0
L.057	2	0	0	0	0	0	0	0
L.058	1	2	0	0	0	0	0	0
L.059	4	0	0	0	0	0	0	0
L.060	5	0	0	0	0	0	0	0
L.061	1	3	0	0	0	0	0	0
L.062	4	5	0	0	0	0	0	0
L.063	2	0	0	0	0	0	0	0
L.064	1	0	0	0	0	0	0	0
L.065	3	0	0	0	0	0	0	0
L.066	5	0	0	0	0	0	0	0
L.067	1	3	0	0	0	0	0	0
L.068	2	4	0	0	0	0	0	0
L.069	0	2	3	4	3			
L.070	1	0	1	4	4			
L.071	1	1	0	4	5			
L.072	1	2	3	0	5			
L.073	3	4	3	4	0			
L.074	4	10	7	1	11	13	3	
L.075	9	8	2	5	14	6	12	
L.076	11	10	11	12	10			
L.077	1	2	3	4	5	6	7	8
L.078	9	10	11	12	13	14	15	16
L.079	17	18	19	20	21	22	23	24
L.080	25	26	27	28	29	30	31	32
L.081	33	34	35	36	37	38	39	40
L.082	1	2	3	0				
L.083	7	8	0	0				
L.084	4	5	6	0				
L.085	9	10	11	12				
L.086	13	14	0	0				
L.087	0	3	1	2	1			
L.088	7	0	7	8	8			
L.089	4	4	0	5	6			
L.090	10	9	11	0	12			
L.091	13	14	13	14	0			

/D NODD									
L.001	01	0001	2	500	8	4000	6	40	20
L.002	500	100	1	16	70	25	1	450	1
L.003	15	10	250	50					
L.004	1	1	2	16	7200	7200	400	0	000
L.005	1	1	2	16	7200	7200	400	0	000
L.006	1	2	2	17	7200	7200	400	0	000
L.007	1	3	2	18	7200	7200	400	0	000
L.008	1	4	3	21	7200	7200	400	0	000
L.009	1	5	3	22	7200	7200	400	0	000
L.010	1	6	3	23	7200	7200	400	0	000
L.011	1	7	4	26	7200	7200	400	0	000
L.012	1	8	4	27	7200	7200	400	0	000
L.013	1	9	4	28	7200	7200	400	0	000
L.014	1	10	5	31	7200	7200	400	0	000
L.015	1	11	5	32	7200	7200	400	0	000
L.016	1	12	5	33	7200	7200	400	0	000
L.017	1	13	6	36	7200	7200	400	0	000
L.018	1	14	6	37	7200	7200	400	0	000
L.019	1	15	6	38	7200	7200	400	0	000
L.020	2	16	1	1	7200	7200	400	0	000
L.021	2	17	3	24	7200	7200	400	0	000
L.022	2	18	4	29	7200	7200	400	0	000
L.023	2	19	5	34	7200	7200	400	0	000
L.024	2	20	6	39	7200	7200	400	0	000
L.025	3	21	1	2	7200	7200	400	0	000
L.026	3	22	2	20	7200	7200	400	0	000
L.027	3	23	4	30	7200	7200	400	0	000
L.028	3	24	5	35	7200	7200	400	0	000
L.029	3	25	6	40	7200	7200	400	0	000
L.030	4	26	1	3	7200	7200	400	0	000
L.031	4	27	2	16	7200	7200	400	0	000
L.032	4	28	3	21	7200	7200	400	0	000
L.033	4	29	5	31	7200	7200	400	0	000
L.034	4	30	6	36	7200	7200	400	0	000
L.035	5	31	1	4	7200	7200	400	0	000
L.036	5	32	2	17	7200	7200	400	0	000
L.037	5	33	3	22	7200	7200	400	0	000
L.038	5	34	4	27	7200	7200	400	0	000
L.039	5	35	6	37	7200	7200	400	0	000
L.040	6	36	1	4	7200	7200	400	0	000
L.041	6	37	2	18	7200	7200	400	0	000
L.042	6	38	3	22	7200	7200	400	0	000
L.043	6	39	4	28	7200	7200	400	0	000
L.044	6	40	5	32	7200	7200	400	0	000
L.045	6	40	5	32	7200	7200	400	0	000
L.046	6	39	4	28	7200	7200	400	0	000
L.047	6	38	3	22	7200	7200	400	0	000
L.048	6	37	2	18	7200	7200	400	0	000
L.049	6	36	1	4	7200	7200	400	0	000
L.050	5	35	6	37	7200	7200	400	0	000
L.051	5	34	4	27	7200	7200	400	0	000
L.052	5	33	3	22	7200	7200	400	0	000
L.053	5	32	2	17	7200	7200	400	0	000
L.054	5	31	1	4	7200	7200	400	0	000

L.055	2	0	0	0	0	0	0	0		
L.056	2	0	0	0	0	0	0	0		
L.057	3	0	0	0	0	0	0	0		
L.058	3	0	0	0	0	0	0	0		
L.059	4	0	0	0	0	0	0	0		
L.060	4	0	0	0	0	0	0	0		
L.061	5	0	0	0	0	0	0	0		
L.062	5	0	0	0	0	0	0	0		
L.063	6	0	0	0	0	0	0	0		
L.064	6	0	0	0	0	0	0	0		
L.065	1	3	4	5	6	0	0	0		
L.066	1	3	4	5	6	0	0	0		
L.067	1	2	4	5	6	0	0	0		
L.068	1	2	4	5	6	0	0	0		
L.069	1	2	3	5	6	0	0	0		
L.070	1	2	3	5	6	0	0	0		
L.071	1	2	3	4	6	0	0	0		
L.072	1	2	3	4	6	0	0	0		
L.073	1	2	3	4	5	0	0	0		
L.074	1	2	3	4	5	0	0	0		
L.075	0	2	3	4	5	6				
L.076	1	0	1	1	1	1				
L.077	1	1	0	1	1	1				
L.078	1	1	1	0	1	1				
L.079	1	1	1	1	0	1				
L.080	1	1	1	1	1	0				
L.081	11	12	13	14	15	16	17	18	19	20
L.082	1	2	3	4	5	6	7	8	9	10
L.083	25	10	10	10	10	10				
L.084	16	17	18	19	20					
L.085	21	22	23	24	25					
L.086	26	27	28	29	30					
L.087	31	32	33	34	35					
L.088	36	37	38	39	40					
L.089	1	2	3	4	5	6	7	8	9	10
L.090	11	12	0	0	0	0	0	0	0	0
L.091	13	14	0	0	0	0	0	0	0	0
L.092	15	16	0	0	0	0	0	0	0	0
L.093	17	18	0	0	0	0	0	0	0	0
L.094	19	20	0	0	0	0	0	0	0	0
L.095	0	1	3	5	7	9				
L.096	11	0	11	11	11	11				
L.097	13	13	0	13	13	13				
L.098	15	15	15	0	15	15				
L.099	17	17	17	17	0	17				
L.100	19	19	19	19	19	0				
L.101	0	2	4	6	8	10				
L.102	12	0	12	12	12	12				
L.103	14	14	0	14	14	14				
L.104	16	16	16	0	16	16				
L.105	18	18	18	18	0	18				
L.106	20	20	20	20	20	0				

/D I O P D									
L.001	20	0250	2	500	8	4000	6	9	16
L.002	500	100	1	60	70	7	2	100	1
L.003	2	5	20	2					
L.004	1	1	3	4	9600	9600	800	6	100
L.005	1	1	3	4	9600	9600	800	6	100
L.006	1	1	3	4	9600	9600	800	6	100
L.007	1	1	3	4	9600	9600	800	6	100
L.008	1	1	3	4	9600	9600	800	6	100
L.009	1	1	3	4	9600	9600	800	6	100
L.010	1	1	3	4	9600	9600	800	6	100
L.011	1	1	3	4	9600	9600	800	6	100
L.012	1	1	3	4	9600	9600	800	6	100
L.013	1	1	3	4	9600	9600	800	6	100
L.014	3	4	1	1	9600	9600	800	6	100
L.015	3	4	1	1	9600	9600	800	6	100
L.016	3	4	1	1	9600	9600	800	6	100
L.017	3	4	1	1	9600	9600	800	6	100
L.018	3	4	1	1	9600	9600	800	6	100
L.019	3	4	1	1	9600	9600	800	6	100
L.020	3	4	1	1	9600	9600	800	6	100
L.021	3	4	1	1	9600	9600	800	6	100
L.022	3	4	1	1	9600	9600	800	6	100
L.023	3	4	1	1	9600	9600	800	6	100
L.024	5	7	1	2	0	0	250	0	0
L.025	2	3	6	9	0	0	250	0	0
L.026	2	3	4	0	0	0	0	0	
L.027	1	0	0	0	0	0	0	0	
L.028	2	0	0	0	0	0	0	0	
L.029	4	5	6	0	0	0	0	0	
L.030	4	5	6	0	0	0	0	0	
L.031	1	3	4	0	0	0	0	0	
L.032	1	2	3	0	0	0	0	0	
L.033	1	2	3	0	0	0	0	0	
L.034	6	0	0	0	0	0	0	0	
L.035	6	0	0	0	0	0	0	0	
L.036	5	0	0	0	0	0	0	0	
L.037	3	4	0	0	0	0	0	0	
L.038	3	4	0	0	0	0	0	0	
L.039	5	0	0	0	0	0	0	0	
L.040	6	0	0	0	0	0	0	0	
L.041	3	4	0	0	0	0	0	0	
L.042	0	2	3	4	4	4			
L.043	1	0	3	4	4	4			
L.044	1	2	0	4	5	6			
L.045	1	2	3	0	5	6			
L.046	3	3	3	4	0	6			
L.047	3	3	3	4	5	0			
L.048	2	1	6	8	7	3	5	4	
L.049	12	13	16	9	10	15	14	11	
L.050	3	2	5	7	3	6			
L.051	1	2							
L.052	3	0							
L.053	4	0							
L.054	5	6							
L.055	7	0							
L.056	8	9							
L.057	1	0	0	0	0				
L.058	6	0	0	0	0				
L.059	2	3	4	5	0				
L.060	7	8	9	10	11				
L.061	15	16	0	0	0				
L.062	12	13	14	0	0				

L.063	0	1	1	1	1	1
L.064	6	0	6	6	6	6
L.065	2	3	0	4	4	4
L.066	7	7	7	0	11	10
L.067	16	16	16	16	0	15
L.068	12	12	12	12	14	0
L.069	0	0	0	0	0	0
L.070	0	0	0	0	0	0
L.071	0	0	0	5	5	5
L.072	8	8	8	0	0	9
L.073	0	0	0	0	0	0
L.074	13	13	13	13	0	0

SRCNO SRCID DSTNO DSTID SRCPC DSTPC

- 357 -

LINE#						LENGTH	MARKVL	NO. MSG
1	1	1	3	4	9600	9600	800	6
2	1	1	3	4	9600	9600	800	6
3	1	1	3	4	9600	9600	800	6
4	1	1	3	4	9600	9600	800	6
5	1	1	3	4	9600	9600	800	6
6	1	1	3	4	9600	9600	800	6
7	1	1	3	4	9600	9600	800	6
8	1	1	3	4	9600	9600	800	6
9	1	1	3	4	9600	9600	800	6
10	1	1	3	4	9600	9600	800	6
11	3	4	1	1	9600	9600	800	6
12	3	4	1	1	9600	9600	800	6
13	3	4	1	1	9600	9600	800	6
14	3	4	1	1	9600	9600	800	6
15	3	4	1	1	9600	9600	800	6
16	3	4	1	1	9600	9600	800	6
17	3	4	1	1	9600	9600	800	6
18	3	4	1	1	9600	9600	800	6
19	3	4	1	1	9600	9600	800	6
20	3	4	1	1	9600	9600	800	6

INTERNAL PACKET GENERATION

21	5	7	1	2	0	0	250	0	0
22	2	3	6	9	0	0	250	0	0

PROCESSOR PARTITION MATRIX

2	3	4	0	0	0	0	0
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
4	5	6	0	0	0	0	0
4	5	6	0	0	0	0	0
1	3	4	0	0	0	0	0
1	2	3	0	0	0	0	0
1	2	3	0	0	0	0	0
6	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
3	4	0	0	0	0	0	0
3	4	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
3	4	0	0	0	0	0	0

NODE FLOT MATRIX, PARTITIONS

0	2	3	4	4	4
1	0	3	4	4	4
1	2	0	4	5	6
1	2	3	0	5	6
3	3	3	4	0	6
3	3	3	4	5	0

NCP TO NCP CONNECTION

2	1	6	8	7	3	5	4	12	13	16	9	10	15	14	11
---	---	---	---	---	---	---	---	----	----	----	---	----	----	----	----

PACKET MULTIPLEXING ON BUS

3	2	5	7	3	6
---	---	---	---	---	---

MAPPING: NODE TO ICP

1	2
3	0
4	0
5	6
7	0
8	9

MAPPING: NODE TO NCP

1	0	0	0	0
6	0	0	0	0
2	3	4	5	0
7	8	9	10	11
15	16	0	0	0
12	13	14	0	0

ROUTING DATA: FIRST CHANNEL

0	1	1	1	1	1
6	0	6	6	6	6
2	3	0	4	4	4
7	7	7	0	11	10
16	16	16	16	0	15
12	12	12	12	14	0

ROUTING DATA: SECOND CHANNEL

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	5	5	5
8	8	8	0	0	9
0	0	0	0	0	0
13	13	13	13	0	0

20	250	2	500	8	4000	6	9	16
500	100	1	60	70	7	2	100	1
2	5	20	2					

APPENDIX C

PARAMETERS FOR NETWORK SIMULATION

Logical Link Data

Number	SRCNOD	SRCICP	DSTNOD	DSTICP	
1	1	1	1	7	
2	1	2	1	8	
3	1	1	2	9	
4	1	2	2	10	
5	1	3	3	17	
6	1	4	3	18	
7	1	5	4	25	
8	1	6	4	26	
9	1	7	5	33	SRCCHC
10	1	8	5	34	DSTCHC
11	2	9	1	1	7200
12	2	10	1	2	BPS
13	2	9	2	15	
14	2	10	2	16	
15	2	11	3	19	
16	2	12	3	20	
17	2	13	4	27	
18	2	14	4	28	
19	2	15	5	35	
20	2	16	5	36	
21	3	17	1	3	
22	3	18	1	4	
23	3	17	2	11	
24	3	18	2	12	

SRCCHC : SOURCE CHANNEL CAPACITY

DSTCHC : DESTINATION CHANNEL CAPACITY

Logical Link Data (Contd.)

Number	SRCNOD	SRCICP	DSTNOP	DSTICP
25	3	19	3	23
26	3	20	3	24
27	3	21	4	29
28	3	22	4	30
29	3	23	5	37
30	3	24	5	38
31	4	25	1	5
32	4	26	1	6
33	4	25	2	13
34	4	26	2	14
35	4	27	3	21
36	4	28	3	22
37	4	29	4	31
38	4	30	4	32
39	4	31	5	39
40	4	32	5	40
41	5	33	1	7
42	5	34	1	8
43	5	33	2	15
44	5	34	2	16
45	5	35	3	23
46	5	36	3	24
47	5	37	4	31
48	5	38	4	32
49	5	39	5	33
50	5	40	5	34

Processor Partition Matrix

Network	Partition							
Processor	1	2	3	4	5	6	7	8
1	3	5	0	0	0	0	0	0
2	4	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	0
4	1	2	0	0	0	0	0	0
5	4	0	0	0	0	0	0	0
6	5	0	0	0	0	0	0	0
7	1	3	0	0	0	0	0	0
8	4	5	0	0	0	0	0	0
9	2	0	0	0	0	0	0	0
10	1	0	0	0	0	0	0	0
11	3	0	0	0	0	0	0	0
12	5	0	0	0	0	0	0	0
13	1	3	0	0	0	0	0	0
14	2	4	0	0	0	0	0	0

Packet Multiplexing on Bus

Node	1	2	3	4	5
No Paks MUX	11	10	11	12	10

Mapping: Node to ICP

Node	ICP's							
1	1	2	3	4	5	6	7	8
2	9	10	11	12	13	14	15	16
3	17	18	19	20	21	22	23	24
4	25	26	27	28	29	30	31	32
5	33	34	35	36	37	38	39	40

Mapping: Node to NCP

Node	NCP's							
1	1	2	3	0	-	-	-	-
2	7	8	0	0	-	-	-	-
3	4	5	6	0	-	-	-	-
4	9	10	11	12	-	-	-	-
5	13	14	0	0	-	-	-	-

Node Plot Matrix, Partitions

		DESTINATION NODE				
		1	2	3	4	5
SOURCE NODE	1	0	2	3	4	3
	2	1	0	1	4	4
	3	1	1	0	4	5
	4	1	2	3	0	5
	5	3	4	3	4	0

NCP to NCP connection

Source NCP	1	2	3	4	5	6
Destination NCP	4	10	7	1	11	13

Source NCP	7	8	9	10	11	12
Destination NCP	3	9	8	2	5	14

Source NCP	13	14
Destination NCP	6	12

Route Map: First Channel

		DESTINATION NODE				
		1	2	3	4	5
PRESENT NODE	1	0	3	1	2	1
	2	7	0	7	8	8
	3	4	4	0	5	6
	4	10	9	11	0	12
	5	13	14	13	14	0

Network Parameters

N	50	BITLNG	70
NUMBER	-	NOI	12
SCAN	2	MAXCH	1
BUFBLK	500	NOPCKS	450
NSOURC	8	CMAx	1
SGMENT	4000	NMAXIC	8
NONDS	5	NMAXIN	4
NOICPS	40	QUS	250
NONCPS	14	NVC	50
PSZNO	500	SWPROC	0,01
NRDN	100	MUXRT	64000
NRRN	1	TXMAX	0,040
MNOBFS	16	PXMAX	0,030
NPZ	0,0005	NODPRC	0,0005

APPENDIX D

PARAMETERS FOR NODAL SIMULATION

Logical Link Data

Number	SRCNOD	SRCICP	DSTNOD	DSTICP
1	1	1	2	16
2	1	2	2	17
3	1	3	2	18
4	1	4	3	21
5	1	5	3	22
6	1	6	3	23
7	1	7	4	26
8	1	8	4	27
9	1	9	4	28
10	1	10	5	31
11	1	11	5	32
12	1	12	5	33
13	1	13	6	36
14	1	14	6	37
15	1	15	6	38
16	2	16	1	1
17	2	17	3	24
18	2	18	4	29
19	2	19	5	34
20	2	20	6	39
21	3	21	1	2
22	3	22	2	20
23	3	23	4	30
24	3	24	5	35
25	3	25	6	40

SRCCHC
DSTCHC
7200 BPS

SRCCHC : SOURCE CHANNEL CAPACITY

DSTCHC : DESTINATION CHANNEL CAPACITY

Logical Link Data (Contd.)

Number	SRCNOD	SRCICP	DSTNOD	DSTICP
26	4	26	1	3
27	4	27	2	16
28	4	28	3	21
29	4	29	5	31
30	4	30	6	36
31	5	31	1	4
32	5	32	2	17
33	5	33	3	22
34	5	34	4	27
35	5	35	6	37
36	6	36	1	4
37	6	37	2	18
38	6	38	3	22
39	6	39	4	28
40	6	40	5	32
41	6	40	5	32
42	6	39	4	28
43	6	38	3	22
44	6	37	2	18
45	6	36	1	4
46	5	35	6	37
47	5	34	4	27
48	5	33	3	22
49	5	32	2	17
50	5	31	1	4

Processor Partition Matrix

Network Processor	Partitions							
	1	2	3	4	5	6	7	8
1	2	0	0	0	0	0	0	0
2	2	0	0	0	0	0	0	0
3	3	0	0	0	0	0	0	0
4	3	0	0	0	0	0	0	0
5	4	0	0	0	0	0	0	0
6	4	0	0	0	0	0	0	0
7	5	0	0	0	0	0	0	0
8	5	0	0	0	0	0	0	0
9	6	0	0	0	0	0	0	0
10	6	0	0	0	0	0	0	0
11	1	3	4	5	6	0	0	0
12	1	3	4	5	6	0	0	0
13	1	2	4	5	6	0	0	0
14	1	2	4	5	6	0	0	0
15	1	2	3	5	6	0	0	0
16	1	2	3	5	6	0	0	0
17	1	2	3	4	6	0	0	0
18	1	2	3	4	6	0	0	0
19	1	2	3	4	5	0	0	0
20	1	2	3	4	5	0	0	0

Packet Multiplexing on Bus

Node	1	2	3	4	5	6
No Pak MUX	25	7	7	7	7	7

Mapping: Node to ICP

Node	ICP's														
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	16	17	18	19	20	0	0	0	0	0	0	0	0	0	0
3	21	22	23	24	25	0	0	0	0	0	0	0	0	0	0
4	26	27	28	29	30	0	0	0	0	0	0	0	0	0	0
5	31	32	33	34	35	0	0	0	0	0	0	0	0	0	0
6	36	37	38	39	40	0	0	0	0	0	0	0	0	0	0

Mapping: Node to NCP

Node	NCP's									
1	1	2	3	4	5	6	7	8	9	10
2	11	12	0	0	0	0	0	0	0	0
3	13	14	0	0	0	0	0	0	0	0
4	15	16	0	0	0	0	0	0	0	0
5	17	18	0	0	0	0	0	0	0	0
6	19	20	0	0	0	0	0	0	0	0

Node Plot Matrix Partition

		DESTINATION NODE					
		1	2	3	4	5	6
SOURCE NODE	1	0	2	3	4	5	6
	2	1	0	1	1	1	1
	3	1	1	0	1	1	1
	4	1	1	1	0	1	1
	5	1	1	1	1	0	1
	6	1	1	1	1	1	0

NCP to NCP Connection

SOURCE NCP	1	2	3	4	5	6	7	8	9	10
DEST NCP	11	12	13	14	15	16	17	18	19	20

SOURCE NCP	11	12	13	14	15	16	17	18	19	20
DEST NCP	1	2	3	4	5	6	7	8	9	10

Route Map: First Channel

	DESTINATION NODE					
	1	2	3	4	5	6
PRESENT NODE	1	0	1	3	5	7
	2	11	0	11	11	11
	3	13	13	0	13	13
	4	15	15	15	0	15
	5	17	17	17	17	0
	6	19	19	19	19	0

Route Map: Second Channel

	DESTINATION NODE					
	1	2	3	4	5	6
PRESENT NODE	1	0	2	4	6	10
	2	12	0	12	12	12
	3	14	14	0	14	14
	4	16	16	16	0	16
	5	18	18	18	18	0
	6	20	20	20	20	0

Network Parameters

N	50	BITLNG	70
NUMBER	-	NOI	25
SCAN	2	MAXCH	2
BUFBLK	500	NOPCKS	450
NSOURC	8	CMAX	1
SGMENT	4000	NMAXIC	15
NONDS	6	NMAXIN	10
NOICPS	40	QUS	250
NONCPS	20	NVC	50
PSZNO	500	SWPROC	0,01
NRDN	100	MUXRT	640000
NRRN	1	TXMAX	0,040
MNORFS	16	PXMAX	0,030
NPZ	0,0005	NODPRL	0,0005

APPENDIX E

SIMULATION RESULTS

This appendix contains the tables of all simulation performance results for the nodal and network simulations.

CONTENTS

NETWORK SIMULATION: EXTENDED RUNS -
TABLES E2 - E12

NETWORK SIMULATION: DELAY VALUES -
TABLES E13 - E15

NODAL SIMULATION: DELAY VALUES -
TABLES E16 - E19

NETWORK AND NODAL SIMULATION: RECORDED
THROUGHPUT - TABLES E20 - E21

NETWORK SIMULATION: PRIORITY VALUE VARIATION
TABLES E22 - E23

NODAL SIMULATION: RETRM INTERVAL VARIATION
TABLES E24 - E26

NODAL SIMULATION: BUS RATE & ALG EXEC
TABLES E27 - E29

NETWORK SIMULATION: ALG EXEC TIME
TABLES E30 - E31

NODAL SIMULATION: SINGLE CHANNEL CONNECTION
TABLES E32 - E33

TABLE E1 APPENDIX CONTENTS

NETWORK SIMULATION: EXTENDED RUNS
CYCLE AND SIMULATED TIME VALUES (SECONDS)

CYCLES (2000) (4000) (6000) (8000) (10000) (12000) (14000) (16000)

INPUT LEVEL IS 50 PACKETS/SEC

PRIORITY(ENTER) = 1	PRIORITY(TRANSIT) = 5							
5.85	11.31	16.92	22.52	28.38	34.50	40.38	46.23	
PRIORITY(ENTER) = 5	PRIORITY(TRANSIT) = 1							
5.85	11.31	16.92	22.52	28.38	34.50	40.38	46.23	

INPUT LEVEL IS 525 PACKETS/SEC

PRIORITY(ENTER) = 1	PRIORITY(TRANSIT) = 5							
0.73	1.37	2.01	2.68	3.34	4.01	4.63	5.08	
PRIORITY(ENTER) = 5	PRIORITY(TRANSIT) = 1							
0.72	1.37	1.98	2.65	3.33	4.02	4.72	5.32	

INPUT LEVEL IS 625 PACKETS/SEC

PRIORITY(ENTER) = 1	PRIORITY(TRANSIT) = 5							
0.62	1.16	1.74	2.31	2.91	3.50	4.05	4.56	
PRIORITY(ENTER) = 5	PRIORITY(TRANSIT) = 1							
0.60	1.18	1.76	2.36	2.93	3.51	4.01	4.50	

INPUT LEVEL IS 750 PACKETS/SEC

PRIORITY(ENTER) = 1	PRIORITY(TRANSIT) = 5							
0.54	1.02	1.50	1.98	2.49	3.00	3.42	3.76	
PRIORITY(ENTER) = 5	PRIORITY(TRANSIT) = 1							
0.54	1.02	1.51	2.02	2.50	2.95	3.32	3.64	

INPUT LEVEL IS 835 PACKETS/SEC

PRIORITY(ENTER) = 1	PRIORITY(TRANSIT) = 5							
0.50	0.96	1.40	1.78	2.20	2.59	2.92	3.25	
PRIORITY(ENTER) = 5	PRIORITY(TRANSIT) = 1							
0.50	0.96	1.42	1.88	2.33	2.79	3.17	3.55	

TABLE E-2

NETWORK SIMULATION: EXTENDED RUNS

INPUT LEVEL IS 50 PACKETS/SEC

CYCLES	TOTAL DELAY		ENTRY DELAY		TRANSIT DELAY	THROUGHPUT
	MEAN	DEV	MEAN	DEV	MEAN	PCKS/SEC
	MILLISECONDS					
PRIORITY(ENTER) = 1		PRIORITY(TRANSIT) = 5				
2000	31.43	17.17	11.43	2.33	20.00	49
4000	31.59	16.32	11.42	2.26	20.17	50
6000	31.02	16.38	11.51	2.33	19.51	51
8000	31.16	16.07	11.54	2.36	19.62	51
10000	31.19	15.81	11.51	2.34	19.68	50
12000	31.09	15.82	11.50	2.34	19.59	49
14000	31.01	15.67	11.53	2.38	19.48	49
16000	31.65	15.67	11.52	2.36	19.53	50

PRIORITY(ENTER) = 5			PRIORITY(TRANSIT) = 1			
2000	31.43	17.16	11.42	2.30	20.01	49
4000	31.59	16.32	11.41	2.22	20.18	50
6000	31.35	16.62	11.48	2.29	19.87	51
8000	31.58	16.68	11.49	2.26	20.09	50
10000	31.71	16.60	11.54	2.34	20.17	50
12000	31.53	17.04	11.51	2.31	20.01	50
14000	31.60	17.07	11.58	2.45	20.02	50
16000	31.77	17.05	11.60	2.45	20.17	50

BUS UTILIZATION FACTORS

CYCLES	NODAL NUMBERS				
	(1)	(2)	(3)	(4)	(5)
PRIORITY(ENTER) = 1			PRIORITY(TRANSIT) = 5		
2000	0.02	0.01	0.02	0.01	0.01
4000	0.02	0.01	0.02	0.02	0.01
6000	0.02	0.01	0.02	0.02	0.01
8000	0.02	0.01	0.02	0.02	0.01
10000	0.02	0.01	0.02	0.02	0.01
12000	0.02	0.01	0.02	0.02	0.01
14000	0.02	0.01	0.02	0.02	0.01
16000	0.02	0.01	0.02	0.02	0.01
2000	0.02	0.01	0.02	0.01	0.01
4000	0.02	0.01	0.02	0.02	0.01
6000	0.02	0.01	0.02	0.02	0.01
8000	0.02	0.01	0.02	0.02	0.01
10000	0.02	0.01	0.02	0.02	0.01
12000	0.02	0.01	0.02	0.02	0.01
14000	0.02	0.01	0.02	0.02	0.01
16000	0.02	0.01	0.02	0.02	0.01

TABLE E-3

NETWORK SIMULATION: EXTENDED RUNS
LINE UTILIZATION FACTORS

CHANNELS	CYCLES							
	(2000)	(4000)	(6000)	(8000)	(10000)	(1200)	(14000)	(16000)
50 PACKETS/SEC								
PRIORITY(ENTER) = 1 PRIORITY(TRANSIT) = 5								
(1)	0.06	0.05	0.05	0.05	0.05	0.05	0.05	0.05
(2)	0.01	0.01	0.01	0.02	0.02	0.02	0.02	0.02
(3)	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03
(4)	0.07	0.05	0.05	0.05	0.05	0.05	0.05	0.05
(5)	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
(6)	0.04	0.04	0.04	0.04	0.04	0.03	0.03	0.03
(7)	0.03	0.02	0.03	0.03	0.03	0.03	0.03	0.03
(8)	0.03	0.03	0.03	0.03	0.04	0.04	0.03	0.04
(9)	0.04	0.04	0.04	0.03	0.03	0.03	0.03	0.03
(10)	0.02	0.01	0.02	0.02	0.02	0.02	0.02	0.02
(11)	0.02	0.02	0.01	0.02	0.02	0.02	0.02	0.01
(12)	0.03	0.05	0.04	0.04	0.04	0.04	0.04	0.04
(13)	0.04	0.04	0.04	0.04	0.03	0.03	0.03	0.03
(14)	0.03	0.04	0.04	0.03	0.03	0.03	0.03	0.03
PRIORITY(ENTER) = 5 PRIORITY(TRANSIT) = 1								
(1)	0.06	0.05	0.06	0.05	0.05	0.05	0.05	0.05
(2)	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.02
(3)	0.03	0.03	0.03	0.03	0.03	0.03	0.04	0.04
(4)	0.07	0.05	0.05	0.05	0.05	0.05	0.05	0.05
(5)	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
(6)	0.04	0.04	0.04	0.03	0.03	0.04	0.04	0.04
(7)	0.03	0.02	0.03	0.03	0.03	0.03	0.03	0.03
(8)	0.03	0.03	0.03	0.03	0.04	0.03	0.04	0.04
(9)	0.04	0.04	0.04	0.04	0.04	0.03	0.03	0.04
(10)	0.02	0.01	0.01	0.02	0.01	0.01	0.02	0.02
(11)	0.02	0.02	0.02	0.01	0.02	0.02	0.02	0.02
(12)	0.03	0.05	0.04	0.04	0.04	0.04	0.04	0.04
(13)	0.04	0.04	0.04	0.04	0.04	0.03	0.03	0.03
(14)	0.03	0.04	0.03	0.03	0.03	0.03	0.03	0.03

TABLE E-4

NETWORK SIMULATION: EXTENDED RUNS
INPUT LEVEL IS 25 PACKETS/SEC

CYCLES	TOTAL DELAY		ENTRY DELAY		TRANSIT DELAY	THROUGHPUT
	MEAN	DEV	MEAN	DEV	MEAN	
			MILLISECONDS			PCKS/SEC
PRIORITY(ENTER) = 1			PRIORITY(TRANSIT) = 5			
2000	46.67	20.71	20.87	8.82	25.80	453
4000	48.38	23.23	21.06	10.02	27.32	497
6000	48.62	23.84	20.89	9.55	27.73	509
8000	49.16	24.57	20.95	9.42	28.21	512
10000	49.12	24.65	20.66	9.06	28.46	513
12000	49.55	25.30	20.63	9.03	28.92	517
14000	51.06	29.32	20.66	8.93	30.40	518
16000	51.83	28.74	20.57	8.80	31.26	518
PRIORITY(ENTER) = 5			PRIORITY(TRANSIT) = 1			
2000	47.25	25.26	17.20	4.96	30.05	455
4000	48.10	26.17	17.73	5.61	30.37	496
6000	50.60	27.62	18.07	5.66	32.53	511
8000	50.47	28.16	18.17	5.77	32.30	511
10000	50.09	27.81	18.08	5.76	32.01	509
12000	50.45	28.10	18.04	5.85	32.41	508
14000	49.69	27.85	17.99	5.80	31.70	505
16000	51.41	30.64	17.94	5.71	33.47	506

BUS UTILIZATION FACTORS

CYCLES	NODAL NUMBERS				
	(1)	(2)	(3)	(4)	(5)
PRIORITY(ENTER) = 1			PRIORITY(TRANSIT) = 5		
2000	0.15	0.13	0.15	0.16	0.12
4000	0.17	0.13	0.17	0.16	0.13
6000	0.17	0.13	0.18	0.17	0.13
8000	0.17	0.13	0.17	0.17	0.13
10000	0.17	0.13	0.17	0.17	0.14
12000	0.17	0.13	0.17	0.17	0.14
14000	0.17	0.13	0.18	0.17	0.14
16000	0.17	0.13	0.17	0.17	0.14
PRIORITY(ENTER) = 5			PRIORITY(TRANSIT) = 1		
2000	0.15	0.13	0.14	0.16	0.12
4000	0.16	0.13	0.16	0.17	0.13
6000	0.17	0.14	0.16	0.17	0.14
8000	0.17	0.14	0.17	0.17	0.13
10000	0.17	0.14	0.16	0.17	0.14
12000	0.17	0.14	0.17	0.17	0.13
14000	0.17	0.13	0.16	0.17	0.13
16000	0.16	0.14	0.16	0.17	0.13

TABLE E-5

NETWORK SIMULATION: EXTENDED RUNS

INPUT LEVEL IS 525 PACKETS/SEC LINE UTILIZATION FACTORS

CHANNELS	CYCLES							
	(2000)	(4000)	(6000)	(8000)	(10000)	(12000)	(14000)	(16000)
PRIORITY(ENTER) = 1 PRIORITY(TRANSIT) = 5								
(1)	0.41	0.51	0.49	0.52	0.53	0.55	0.55	0.56
(2)	0.08	0.12	0.17	0.16	0.16	0.17	0.18	0.18
(3)	0.37	0.38	0.34	0.35	0.34	0.34	0.37	0.37
(4)	0.50	0.50	0.50	0.50	0.50	0.51	0.51	0.51
(5)	0.18	0.17	0.19	0.19	0.19	0.19	0.20	0.20
(6)	0.26	0.30	0.31	0.33	0.35	0.34	0.34	0.35
(7)	0.36	0.33	0.34	0.32	0.34	0.37	0.39	0.39
(8)	0.28	0.26	0.30	0.34	0.37	0.35	0.35	0.38
(9)	0.35	0.30	0.32	0.32	0.32	0.35	0.37	0.38
(10)	0.16	0.20	0.18	0.18	0.18	0.18	0.18	0.18
(11)	0.16	0.19	0.21	0.19	0.18	0.18	0.21	0.22
(12)	0.35	0.30	0.31	0.31	0.35	0.35	0.35	0.34
(13)	0.33	0.33	0.33	0.33	0.33	0.34	0.37	0.37
(14)	0.29	0.31	0.34	0.35	0.34	0.34	0.33	0.36

PRIORITY(ENTER) = 5 PRIORITY(TRANSIT) = 1								
(1)	0.44	0.43	0.45	0.45	0.43	0.46	0.46	0.45
(2)	0.14	0.17	0.17	0.16	0.16	0.16	0.17	0.17
(3)	0.29	0.30	0.34	0.34	0.32	0.32	0.31	0.34
(4)	0.48	0.49	0.54	0.54	0.51	0.52	0.52	0.53
(5)	0.18	0.19	0.19	0.19	0.18	0.17	0.17	0.18
(6)	0.24	0.33	0.31	0.31	0.31	0.31	0.31	0.33
(7)	0.45	0.37	0.36	0.37	0.34	0.33	0.33	0.36
(8)	0.31	0.31	0.35	0.35	0.34	0.34	0.32	0.33
(9)	0.26	0.29	0.34	0.33	0.36	0.36	0.34	0.35
(10)	0.11	0.16	0.16	0.18	0.19	0.17	0.17	0.19
(11)	0.18	0.19	0.18	0.20	0.23	0.22	0.23	0.24
(12)	0.47	0.41	0.44	0.39	0.39	0.38	0.37	0.38
(13)	0.24	0.21	0.36	0.37	0.38	0.37	0.37	0.37
(14)	0.30	0.27	0.28	0.27	0.32	0.33	0.32	0.33

TABLE E-6

NETWORK SIMULATION: EXTENDED RUNS

INPUT LEVEL IS 625 PACKETS/SEC

CYCLES	TOTAL DELAY		ENTRY DELAY		TRANSIT DELAY	THROUGHPUT
	MEAN	DEV	MEAN	DEV	MEAN	
MILLISECONDS						PCKS/SEC
PRIORITY(ENTER) = 1 PRIORITY(TRANSIT) =5						
2000	52.43	32.56	24.80	14.97	27.63	557
4000	53.79	31.07	23.16	12.93	30.63	611
6000	53.07	29.83	22.87	11.65	30.20	623
8000	52.81	28.79	22.76	11.38	30.05	624
10000	52.06	27.84	22.63	11.44	29.43	617
12000	52.98	29.69	22.67	11.21	30.31	616
14000	53.07	29.63	22.76	11.47	30.31	623
16000	53.85	30.11	23.09	13.02	30.76	618
PRIORITY(ENTER) = 5 PRIORITY(TRANSIT) = 1						
2000	50.01	28.58	17.93	5.42	32.08	554
4000	51.07	29.46	18.51	6.49	32.56	596
6000	51.85	29.61	19.46	5.45	32.39	604
8000	51.70	29.60	19.08	6.34	32.62	602
10000	51.60	29.49	19.14	6.26	32.46	607
12000	52.88	31.28	19.22	6.25	33.66	603
14000	55.01	37.16	19.50	6.78	33.51	605
16000	58.38	43.80	19.63	6.99	38.78	605

BUS UTILIZATION FACTORS

CYCLES	NODAL NUMBERS				
	(1)	(2)	(3)	(4)	(5)
PRIORITY(ENTER) = 1 PRIORITY(TRANSIT) = 5					
2000	0.19	0.15	0.19	0.18	0.14
4000	0.19	0.15	0.20	0.21	0.16
6000	0.19	0.15	0.21	0.21	0.16
8000	0.19	0.15	0.21	0.21	0.16
10000	0.19	0.15	0.20	0.21	0.16
12000	0.20	0.15	0.20	0.20	0.16
14000	0.20	0.15	0.20	0.20	0.16
16000	0.20	0.15	0.20	0.20	0.16
PRIORITY(ENTER) = 5 PRIORITY(TRANSIT) = 1					
2000	0.16	0.17	0.17	0.20	0.15
4000	0.19	0.17	0.19	0.19	0.16
6000	0.19	0.17	0.19	0.19	0.16
8000	0.18	0.17	0.19	0.20	0.16
10000	0.19	0.17	0.19	0.20	0.16
12000	0.19	0.17	0.19	0.20	0.16
14000	0.19	0.17	0.19	0.20	0.16
16000	0.19	0.17	0.19	0.20	0.16

TABLE E-7

NETWORK SIMULATION: EXTENDED RUNS
 INPUT LEVEL IS 625 PACKETS/SEC LINE UTILIZATION FACTORS
 CHANNELS CYCLES
 (2000) (4000) (6000) (8000) (10000) (12000) (14000) (16000)

PRIORITY(ENTER) = 1					PRIORITY(TRANSIT) = 5			
(1)	0.68	0.61	0.59	0.56	0.62	0.62	0.62	0.66
(2)	0.09	0.17	0.15	0.21	0.22	0.21	0.24	0.26
(3)	0.42	0.45	0.46	0.43	0.40	0.41	0.41	0.44
(4)	0.61	0.64	0.63	0.61	0.59	0.60	0.61	0.63
(5)	0.22	0.23	0.24	0.24	0.23	0.23	0.22	0.25
(6)	0.38	0.38	0.40	0.40	0.38	0.42	0.43	0.47
(7)	0.47	0.48	0.46	0.45	0.45	0.43	0.42	0.45
(8)	0.36	0.37	0.40	0.41	0.42	0.41	0.41	0.42
(9)	0.33	0.34	0.35	0.37	0.36	0.36	0.34	0.36
(10)	0.11	0.15	0.15	0.16	0.19	0.20	0.21	0.22
(11)	0.25	0.23	0.27	0.26	0.26	0.24	0.23	0.24
(12)	0.35	0.41	0.46	0.46	0.48	0.44	0.43	0.43
(13)	0.31	0.35	0.38	0.38	0.35	0.36	0.36	0.41
(14)	0.34	0.51	0.50	0.46	0.44	0.41	0.40	0.43

PRIORITY(ENTER) = 5					PRIORITY(TRANSIT) = 1			
(1)	0.35	0.46	0.51	0.49	0.50	0.55	0.61	0.61
(2)	0.19	0.21	0.17	0.16	0.19	0.18	0.19	0.20
(3)	0.45	0.42	0.34	0.33	0.34	0.37	0.40	0.39
(4)	0.55	0.57	0.55	0.53	0.55	0.59	0.60	0.61
(5)	0.21	0.24	0.26	0.23	0.23	0.25	0.26	0.25
(6)	0.25	0.40	0.41	0.35	0.38	0.37	0.42	0.44
(7)	0.47	0.44	0.45	0.42	0.41	0.43	0.45	0.45
(8)	0.35	0.37	0.42	0.42	0.44	0.43	0.41	0.41
(9)	0.48	0.44	0.45	0.45	0.43	0.43	0.44	0.43
(10)	0.19	0.18	0.21	0.21	0.21	0.20	0.20	0.19
(11)	0.24	0.21	0.23	0.26	0.27	0.28	0.28	0.28
(12)	0.46	0.41	0.39	0.39	0.41	0.39	0.38	0.39
(13)	0.37	0.43	0.44	0.40	0.37	0.39	0.41	0.42
(14)	0.43	0.39	0.41	0.44	0.42	0.42	0.44	0.43

TABLE E-8

NETWORK SIMULATION: EXTENDED RUNS

INPUT LEVEL IS 750 PACKETS/SEC

CYCLES	TOTAL DELAY		ENTRY DELAY		TRANSIT DELAY	THROUGHPUT PKTS/SEC
	MEAN	DEV	MEAN	DEV	MEAN	
	MILLISECONDS					

PRIORITY(ENTER) = 1 . PRIORITY(TRANSIT) = 5						
2000	53.62	30.51	24.06	13.93	29.56	637
4000	58.29	30.59	26.38	15.12	31.91	696
6000	60.99	36.52	28.66	21.84	32.33	710
8000	63.02	37.70	29.65	21.83	33.37	724
10000	63.91	38.57	29.67	21.51	34.24	732
12000	63.41	37.60	28.93	20.27	34.48	730
14000	65.14	46.13	28.93	19.84	36.30	726
16000	69.53	56.57	28.77	20.00	40.76	718

PRIORITY(ENTER) = 5 PRIORITY(TRANSIT) = 1						
2000	54.29	33.80	20.67	8.79	33.62	606
4000	64.82	46.00	21.91	9.30	42.91	693
6000	67.44	48.51	22.68	10.64	44.76	714
8000	69.73	54.59	23.10	11.65	46.63	715
10000	72.13	57.80	23.07	11.52	49.06	723
12000	73.47	63.71	23.00	11.33	50.47	725
14000	74.90	64.66	23.08	11.28	51.82	718
16000	78.31	69.60	23.04	11.09	55.27	708

BUS UTILIZATION FACTORS

CYCLES	NODAL NUMBERS				
	(1)	(2)	(3)	(4)	(5)
PRIORITY(ENTER) = 1 PRIORITY(TRANSIT) = 5					
2000	0.22	0.19	0.20	0.21	0.17
4000	0.23	0.20	0.22	0.23	0.19
6000	0.23	0.20	0.23	0.24	0.20
8000	0.23	0.20	0.23	0.24	0.21
10000	0.23	0.20	0.23	0.24	0.21
12000	0.23	0.20	0.23	0.24	0.21
14000	0.23	0.20	0.23	0.24	0.21
16000	0.23	0.20	0.23	0.24	0.20
PRIORITY(ENTER) = 5 PRIORITY(TRANSIT) = 1					
2000	0.21	0.18	0.21	0.21	0.17
4000	0.23	0.20	0.23	0.23	0.20
6000	0.23	0.20	0.24	0.24	0.20
8000	0.23	0.20	0.24	0.24	0.19
10000	0.23	0.20	0.24	0.24	0.19
12000	0.23	0.19	0.24	0.24	0.20
14000	0.23	0.19	0.23	0.25	0.19
16000	0.23	0.19	0.23	0.25	0.19

TABLE E-9

NETWORK SIMULATION: EXTENDED RUNS

INPUT LEVEL IS 750 PACKETS/SEC LINE UTILIZATION FACTORS

CHANNELS CYCLES
(2000) (4000) (6000) (8000) (10000) (12000) (14000) (16000)

PRIORITY(ENTER) = 1 PRIORITY(TRANSIT) = 5

(1)	0.52	0.57	0.61	0.65	0.63	0.63	0.67	0.70
(2)	0.24	0.25	0.25	0.24	0.25	0.24	0.26	0.29
(3)	0.54	0.58	0.59	0.48	0.48	0.53	0.52	0.55
(4)	0.56	0.64	0.64	0.64	0.65	0.65	0.69	0.72
(5)	0.24	0.21	0.26	0.23	0.24	0.22	0.23	0.26
(6)	0.38	0.46	0.52	0.59	0.59	0.58	0.58	0.60
(7)	0.58	0.50	0.49	0.49	0.47	0.49	0.52	0.56
(8)	0.51	0.53	0.56	0.57	0.56	0.54	0.54	0.59
(9)	0.45	0.53	0.48	0.46	0.49	0.52	0.55	0.60
(10)	0.18	0.24	0.27	0.28	0.27	0.27	0.28	0.30
(11)	0.23	0.22	0.21	0.22	0.24	0.22	0.26	0.30
(12)	0.46	0.55	0.57	0.57	0.56	0.55	0.58	0.62
(13)	0.35	0.40	0.43	0.49	0.49	0.49	0.52	0.57
(14)	0.36	0.42	0.44	0.47	0.50	0.52	0.56	0.60

PRIORITY(ENTER) = 5 PRIORITY(TRANSIT) = 1

(1)	0.62	0.67	0.69	0.72	0.76	0.80	0.82	0.83
(2)	0.23	0.24	0.23	0.25	0.30	0.30	0.32	0.36
(3)	0.48	0.48	0.46	0.47	0.48	0.44	0.50	0.54
(4)	0.60	0.69	0.69	0.74	0.76	0.74	0.77	0.79
(5)	0.27	0.26	0.28	0.24	0.27	0.26	0.31	0.36
(6)	0.40	0.52	0.50	0.49	0.47	0.55	0.60	0.63
(7)	0.66	0.56	0.57	0.56	0.56	0.54	0.59	0.63
(8)	0.36	0.44	0.53	0.52	0.53	0.57	0.61	0.65
(9)	0.39	0.49	0.49	0.51	0.51	0.53	0.55	0.59
(10)	0.10	0.24	0.22	0.26	0.25	0.26	0.28	0.32
(11)	0.28	0.26	0.24	0.25	0.25	0.26	0.29	0.35
(12)	0.49	0.49	0.52	0.54	0.52	0.56	0.56	0.58
(13)	0.41	0.44	0.44	0.41	0.40	0.43	0.49	0.54
(14)	0.38	0.45	0.53	0.52	0.54	0.56	0.56	0.59

TABLE E-10

NETWORK SIMULATION: EXTENDED RUNS
 INPUT LEVEL IS 835 PACKETS/SEC

CYCLES	TOTAL DELAY		ENTRY DELAY		TRANSIT DELAY	THROUGHPUT
	MEAN	DEV	MEAN	DEV	MEAN	

MILLISECONDS						PCKS/SEC
PRIORITY(ENTER) =1	PRIORITY(TRANSIT) =5					
2000	58.95	35.31	28.43	21.19	30.52	656
4000	71.19	53.87	34.23	33.24	36.96	749
6000	78.15	61.75	37.38	39.43	40.77	772
8000	82.03	67.03	38.24	40.88	43.82	787
10000	83.61	70.43	39.50	45.58	44.11	790
12000	86.95	81.05	42.35	55.24	44.60	771
14000	93.49	94.09	44.41	64.87	49.08	756
16000	101.03	108.55	46.68	73.39	54.35	741

PRIORITY(ENTER) =5	PRIORITY(TRANSIT) = 1					
2000	54.23	33.36	21.28	8.28	32.95	696
4000	66.91	56.93	23.71	11.85	43.20	762
6000	71.64	60.22	24.53	13.03	47.11	775
8000	76.07	66.46	25.62	14.63	50.45	777
10000	80.36	73.12	26.60	17.07	53.76	789
12000	83.17	77.52	26.14	16.30	57.03	788
14000	83.94	79.68	26.33	16.08	57.61	782
16000	91.19	91.33	26.49	16.22	64.70	780

BUS UTILIZATION FACTORS

CYCLES	NODAL NUMBERS				
	(1)	(2)	(3)	(4)	(5)

PRIORITY(ENTER) = 1	PRIORITY(TRANSIT) = 5				
2000	0.21	0.20	0.21	0.24	0.20
4000	0.23	0.22	0.24	0.25	0.21
6000	0.24	0.21	0.26	0.26	0.21
8000	0.26	0.21	0.26	0.26	0.22
10000	0.26	0.22	0.25	0.26	0.21
12000	0.25	0.21	0.25	0.26	0.21
14000	0.25	0.21	0.25	0.26	0.21
16000	0.25	0.21	0.24	0.26	0.21

PRIORITY(ENTER) = 5	PRIORITY(TRANSIT) = 1				
2000	0.22	0.19	0.23	0.24	0.19
4000	0.24	0.20	0.26	0.25	0.20
6000	0.25	0.21	0.26	0.26	0.20
8000	0.25	0.22	0.26	0.26	0.20
10000	0.25	0.22	0.26	0.26	0.21
12000	0.25	0.21	0.26	0.26	0.21
14000	0.25	0.21	0.26	0.26	0.21
16000	0.25	0.21	0.26	0.26	0.21

TABLE E-11

NETWORK SIMULATION: EXTENDED RUNS
 INPUT LEVEL IS 835 PACKETS/SEC LINE UTILIZATION FACTORS
 CHANNELS CYCLES
 (2000) (4000) (6000) (8000) (10000) (12000) (14000) (16000)

PRIORITY(ENTER) = 1		PRIORITY(TRANSIT) = 5						
(1)	0.64	0.75	0.81	0.82	0.86	0.88	0.89	0.90
(2)	0.18	0.15	0.21	0.26	0.28	0.27	0.30	0.36
(3)	0.54	0.55	0.51	0.55	0.59	0.63	0.67	0.70
(4)	0.76	0.81	0.86	0.88	0.89	0.91	0.92	0.93
(5)	0.28	0.33	0.32	0.28	0.27	0.33	0.37	0.41
(6)	0.38	0.43	0.47	0.50	0.52	0.56	0.61	0.63
(7)	0.57	0.64	0.61	0.62	0.66	0.71	0.74	0.77
(8)	0.51	0.52	0.60	0.65	0.62	0.61	0.65	0.69
(9)	0.56	0.54	0.60	0.62	0.63	0.65	0.68	0.71
(10)	0.28	0.23	0.21	0.27	0.28	0.27	0.28	0.32
(11)	0.27	0.28	0.28	0.27	0.30	0.33	0.39	0.42
(12)	0.56	0.52	0.58	0.64	0.62	0.66	0.69	0.73
(13)	0.42	0.51	0.53	0.55	0.57	0.60	0.65	0.68
(14)	0.59	0.52	0.60	0.61	0.61	0.66	0.70	0.73

PRIORITY(ENTER) = 5		PRIORITY(TRANSIT) = 1						
(1)	0.62	0.70	0.73	0.72	0.74	0.74	0.77	0.79
(2)	0.22	0.26	0.21	0.22	0.22	0.23	0.21	0.22
(3)	0.44	0.46	0.49	0.48	0.48	0.50	0.55	0.60
(4)	0.70	0.67	0.67	0.72	0.76	0.77	0.80	0.82
(5)	0.30	0.37	0.35	0.34	0.33	0.32	0.32	0.33
(6)	0.34	0.48	0.45	0.45	0.48	0.49	0.54	0.57
(7)	0.57	0.61	0.60	0.57	0.57	0.57	0.60	0.63
(8)	0.47	0.46	0.49	0.52	0.51	0.54	0.55	0.60
(9)	0.49	0.57	0.62	0.66	0.64	0.63	0.67	0.71
(10)	0.25	0.23	0.26	0.30	0.33	0.33	0.31	0.31
(11)	0.25	0.24	0.25	0.24	0.25	0.24	0.28	0.26
(12)	0.56	0.45	0.47	0.50	0.51	0.51	0.52	0.55
(13)	0.37	0.35	0.43	0.45	0.47	0.50	0.54	0.57
(14)	0.58	0.65	0.60	0.61	0.61	0.58	0.60	0.63

TABLE E-12

DELAY VALUES FOR NETWORK SIMULATION IN MILLISECONDS

- 394 -

	TOTAL DELAY		ENTRY DELAY		TRANSIT DELAY
	MEAN	$\sqrt{VAR}$	MEAN	$\sqrt{VAR}$	MEAN
	PRIORITY(ENTER)=1		PRIORITY(TRANSIT)=5		100 PACKETS/SEC
RUN ONE	33.21	16.91	11.77	3.68	20.44
RUN TWO	34.61	17.94	12.94	3.65	21.67
RUN THREE	33.33	16.91	12.45	3.24	20.88
	PRIORITY(ENTER)=5		PRIORITY(TRANSIT)=1		100 PACKETS/SEC
RUN ONE	33.30	16.25	12.63	3.34	20.67
RUN TWO	34.16	17.19	12.60	3.37	21.56
RUN THREE	33.21	17.16	12.49	3.11	20.72
	PRIORITY(ENTER)=1		PRIORITY(TRANSIT)=5		200 PACKETS/SEC
RUN ONE	36.63	17.27	15.20	4.90	21.43
RUN TWO	39.38	18.91	14.91	4.52	24.47
RUN THREE	38.09	19.03	14.90	4.63	23.19
	PRIORITY(ENTER)=5		PRIORITY(TRANSIT)=1		200 PACKETS/SEC
RUN ONE	36.07	18.00	14.54	4.01	21.53
RUN TWO	40.28	19.95	14.73	4.21	25.55
RUN THREE	36.53	19.33	14.47	4.05	22.06
	PRIORITY(ENTER)=1		PRIORITY(TRANSIT)=5		500 PACKETS/SEC
RUN ONE	47.76	22.35	20.49	7.74	27.27
RUN TWO	46.76	22.22	19.47	6.83	27.29
RUN THREE	47.48	23.33	19.76	7.79	27.72
	PRIORITY(ENTER)=5		PRIORITY(TRANSIT)=1		500 PACKETS/SEC
RUN ONE	50.76	31.79	18.40	6.03	32.36
RUN TWO	49.51	28.14	17.90	5.28	31.61
RUN THREE	48.54	26.26	17.88	5.99	30.66
	PRIORITY(ENTER)=1		PRIORITY(TRANSIT)=5		625 PACKETS/SEC
RUN ONE	60.42	43.15	26.05	20.18	34.37
RUN TWO	63.65	40.50	26.36	17.11	40.29
RUN THREE	53.64	30.81	22.73	12.07	30.91
	PRIORITY(ENTER)=5		PRIORITY(TRANSIT)=1		625 PACKETS/SEC
RUN ONE	60.36	43.42	20.45	7.98	39.91
RUN TWO	63.02	42.53	20.95	9.65	42.07
RUN THREE	52.31	30.26	18.81	5.94	33.50
	PRIORITY(ENTER)=1		PRIORITY(TRANSIT)=5		835 PACKETS/SEC
RUN ONE	96.83	86.64	41.18	73.82	35.65
RUN TWO	117.31	91.51	73.63	61.74	43.68
RUN THREE	89.03	73.35	53.87	60.87	35.16
	PRIORITY(ENTER)=5		PRIORITY(TRANSIT)=1		835 PACKETS/SEC
RUN ONE	94.78	97.30	52.53	58.81	42.25
RUN TWO	94.68	98.44	49.26	54.86	45.42
RUN THREE	83.03	76.86	40.24	40.66	42.79

TABLE E-13

NETWORK SIMULATION: BUS UTILIZATION FACTORS

NODAL NUMBERS	(1)	(2)	(3)	(4)	(5)
INPUT LEVEL IS 100 PACKETS/SEC					
	PRIORITY(ENTER)=1		PRIORITY(TRANSIT)=5		
RUN ONE	0.03	0.03	0.03	0.03	0.03
RUN TWO	0.03	0.03	0.03	0.03	0.03
RUN THREE	0.03	0.03	0.03	0.03	0.03
	PRIORITY(ENTER)=5		PRIORITY(TRANSIT)=1		
RUN ONE	0.03	0.03	0.03	0.03	0.03
RUN TWO	0.03	0.03	0.03	0.03	0.03
RUN THREE	0.03	0.03	0.03	0.03	0.03
INPUT LEVEL IS 200 PACKETS/SEC					
	PRIORITY(ENTER)=1		PRIORITY(TRANSIT)=5		
RUN ONE	0.07	0.05	0.06	0.06	0.05
RUN TWO	0.07	0.06	0.07	0.06	0.05
RUN THREE	0.06	0.05	0.07	0.07	0.05
	PRIORITY(ENTER)=5		PRIORITY(TRANSIT)=1		
RUN ONE	0.06	0.05	0.06	0.06	0.05
RUN TWO	0.07	0.05	0.07	0.07	0.05
RUN THREE	0.06	0.06	0.07	0.07	0.05
INPUT LEVEL IS 500 PACKETS/SEC					
	PRIORITY(ENTER)=1		PRIORITY(TRANSIT)=5		
RUN ONE	0.16	0.14	0.15	0.16	0.12
RUN TWO	0.18	0.14	0.17	0.16	0.13
RUN THREE	0.16	0.13	0.16	0.16	0.12
	PRIORITY(ENTER)=5		PRIORITY(TRANSIT)=1		
RUN ONE	0.15	0.13	0.15	0.17	0.13
RUN TWO	0.16	0.14	0.17	0.16	0.14
RUN THREE	0.16	0.13	0.16	0.16	0.13
INPUT LEVEL IS 625 PACKETS/SEC					
	PRIORITY(ENTER)=1		PRIORITY(TRANSIT)=5		
RUN ONE	0.19	0.15	0.20	0.21	0.16
RUN TWO	0.21	0.17	0.22	0.19	0.17
RUN THREE	0.21	0.17	0.20	0.20	0.17
	PRIORITY(ENTER)=5		PRIORITY(TRANSIT)=1		
RUN ONE	0.19	0.17	0.19	0.19	0.16
RUN TWO	0.21	0.18	0.21	0.21	0.17
RUN THREE	0.22	0.18	0.20	0.19	0.16
INPUT LEVEL IS 835 PACKETS/SEC					
	PRIORITY(ENTER)=1		PRIORITY(TRANSIT)=5		
RUN ONE	0.30	0.27	0.31	0.37	0.27
RUN TWO	0.28	0.29	0.33	0.38	0.28
RUN THREE	0.30	0.31	0.31	0.36	0.28
	PRIORITY(ENTER)=5		PRIORITY(TRANSIT)=1		
RUN ONE	0.29	0.29	0.29	0.35	0.26
RUN TWO	0.28	0.26	0.32	0.36	0.30
RUN THREE	0.30	0.26	0.29	0.37	0.26

TABLE E-14

NETWORK SIMULATION: LINE UTILIZATION FACTORS

CHANNELS	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12	-13	-14
INPUT LEVEL IS 100 PACKETS/SEC														
	PRIORITY(ENTER)=1							PRIORITY(TRANSIT)=5						
RUN ONE	.11	.03	.06	.10	.03	.07	.08	.06	.05	.04	.03	.07	.08	.05
RUN TWO	.09	.04	.07	.09	.03	.07	.06	.06	.07	.03	.03	.06	.06	.07
RUN THREE	.10	.03	.06	.10	.04	.07	.07	.07	.07	.03	.04	.07	.08	.06
	PRIORITY(ENTER)=5							PRIORITY(TRANSIT)=1						
RUN ONE	.11	.03	.07	.10	.04	.07	.07	.06	.05	.04	.03	.07	.06	.06
RUN TWO	.08	.04	.07	.10	.03	.07	.05	.06	.08	.03	.03	.07	.06	.07
RUN THREE	.10	.04	.07	.10	.04	.07	.06	.06	.07	.03	.03	.07	.07	.06
INPUT LEVEL IS 200 PACKETS/SEC														
	PRIORITY(ENTER)=1							PRIORITY(TRANSIT)=5						
RUN ONE	.16	.07	.19	.23	.07	.11	.15	.11	.12	.07	.06	.15	.13	.15
RUN TWO	.21	.05	.13	.20	.05	.13	.13	.10	.12	.06	.04	.13	.14	.15
RUN THREE	.21	.10	.11	.19	.09	.14	.13	.14	.18	.07	.08	.14	.12	.13
	PRIORITY(ENTER)=5							PRIORITY(TRANSIT)=1						
RUN ONE	.15	.06	.14	.20	.07	.11	.12	.12	.13	.07	.08	.15	.13	.14
RUN TWO	.19	.07	.14	.20	.06	.12	.12	.10	.15	.06	.07	.11	.12	.15
RUN THREE	.20	.10	.16	.20	.10	.15	.16	.14	.13	.08	.08	.13	.16	.16
INPUT LEVEL IS 500 PACKETS/SEC														
	PRIORITY(ENTER)=1							PRIORITY(TRANSIT)=5						
RUN ONE	.41	.10	.041	.48	.17	.27	.45	.27	.37	.18	.17	.33	.27	.38
RUN TWO	.42	.20	.44	.56	.20	.31	.29	.25	.37	.11	.13	.35	.30	.30
RUN THREE	.47	.20	.31	.45	.20	.34	.27	.40	.25	.17	.20	.31	.36	.34
	PRIORITY(ENTER)=5							PRIORITY(TRANSIT)=1						
RUN ONE	.41	.16	.31	.54	.14	.26	.32	.39	.28	.18	.18	.44	.35	.28
RUN TWO	.47	.15	.37	.47	.17	.33	.42	.29	.31	.10	.17	.36	.26	.35
RUN THREE	.53	.22	.34	.50	.17	.36	.37	.43	.23	.15	.23	.36	.32	.30
INPUT LEVEL IS 625 PACKETS/SEC														
	PRIORITY(ENTER)=1							PRIORITY(TRANSIT)=5						
RUN ONE	.61	.15	.46	.65	.26	.39	.47	.37	.36	.15	.26	.42	.35	.51
RUN TWO	.60	.16	.43	.66	.24	.48	.36	.41	.42	.18	.16	.45	.46	.42
RUN THREE	.62	.18	.45	.68	.23	.43	.43	.42	.43	.16	.22	.43	.43	.49
	PRIORITY(ENTER)=5							PRIORITY(TRANSIT)=1						
RUN ONE	.47	.19	.37	.55	.28	.38	.46	.38	.44	.20	.20	.41	.45	.43
RUN TWO	.56	.22	.48	.59	.21	.34	.47	.45	.33	.18	.21	.46	.31	.36
RUN THREE	.57	.23	.42	.62	.21	.40	.49	.43	.41	.21	.21	.44	.44	.41
INPUT LEVEL IS 835 PACKETS/SEC														
	PRIORITY(ENTER)=1							PRIORITY(TRANSIT)=5						
RUN ONE	.78	.30	.60	.78	.55	.61	.66	.69	.75	.45	.36	.63	.72	.72
RUN TWO	.81	.39	.76	.73	.34	.53	.88	.84	.81	.33	.46	.81	.75	.84
RUN THREE	.64	.35	.46	.72	.39	.61	.60	.76	.81	.34	.45	.69	.68	.84
	PRIORITY(ENTER)=5							PRIORITY(TRANSIT)=1						
RUN ONE	.76	.47	.86	.80	.39	.78	.75	.81	.70	.33	.36	.76	.68	.78
RUN TWO	.75	.38	.60	.80	.47	.68	.66	.72	.63	.35	.38	.63	.63	.77
RUN THREE	.87	.27	.89	.84	.55	.57	.82	.82	.76	.39	.40	.69	.56	.66

TABLE E-15

DELAY VALUES FOR NODAL SIMULATION IN MILLISECONDS						
TOTAL DELAY			ENTRY DELAY		TRANSIT DELAY	
	MEAN	✓VAR	MEAN	✓VAR	MEAN	
50.0 PACKETS/SEC INPUT						
PRIORITY(ENTRY)=1			PRIORITY(TRANSIT)=5			
RUN ONE	41.94		14.81	11.77	2.75	30.17
RUN TWO	43.03		15.93	12.12	3.08	30.91
RUN THREE	42.00		16.26	11.82	2.86	30.18
50.0 PACKETS/SEC INPUT						
PRIORITY(ENTRY)=5			PRIORITY(TRANSIT)=1			
RUN ONE	41.48		14.56	11.97	2.88	29.51
RUN TWO	42.07		16.08	11.91	2.72	30.16
RUN THREE	42.00		15.77	11.80	2.85	30.20
500.0 PACKETS/SEC						
PRIORITY(ENTRY)=1			PRIORITY(TRANSIT)=5			
RUN ONE	59.88		18.09	19.32	7.20	40.56
RUN TWO	65.45		21.93	22.64	11.19	42.81
RUN THREE	64.58		23.38	20.53	8.70	44.05
500.0 PACKETS/SEC						
PRIORITY(ENTRY)=5			PRIORITY(TRANSIT)=1			
RUN ONE	61.97		21.80	17.45	4.97	44.52
RUN TWO	65.70		25.04	18.71	6.50	46.99
RUN THREE	62.87		23.96	17.60	5.61	45.27
625.0 PACKETS/SEC						
PRIORITY(ENTRY)=1			PRIORITY(TRANSIT)=5			
RUN ONE	71.40		25.24	24.74	14.15	46.66
RUN TWO	76.27		27.58	26.10	13.41	50.17
RUN THREE	75.59		29.08	24.28	12.67	51.31
625.0 PACKET/SEC						
PRIORITY(ENTRY)=5			PRIORITY(TRANSIT)=1			
RUN ONE	67.73		26.12	19.76	7.53	47.97
RUN TWO	73.82		31.86	20.37	7.85	53.45
RUN THREE	70.91		30.10	20.12	9.22	50.79
835.0 PACKETS/SEC						
PRIORITY(ENTRY)=1			PRIORITY(TRANSIT)=5			
RUN ONE	146.78		81.35	79.11	75.92	67.67
RUN TWO	148.97		84.78	75.99	71.27	72.98
RUN THREE	143.67		85.96	73.03	71.89	70.64
835.0 PACKETS/SEC						
PRIORITY(ENTRY)=5			PRIORITY(TRANSIT)=1			
RUN ONE	142.02		98.76	55.83	55.58	86.19
RUN TWO	144.25		96.78	61.74	69.01	82.51
RUN THREE	140.15		97.03	58.61	62.23	81.54

TABLE E-16

NODAL SIMULATION: BUS UTILIZATION FACTORS

NODAL NUMBERS	(1)	(2)	(3)	(4)	(5)	(6)
INPUT LEVEL IS 50 PACKETS/SEC						
	PRIORITY(ENTER)=1		PRIORITY(TRANSIT)=5			
RUN ONE	0.04	0.01	0.01	0.01	0.01	0.01
RUN TWO	0.04	0.01	0.01	0.01	0.01	0.01
RUN THREE	0.03	0.01	0.01	0.01	0.01	0.01
	PRIORITY(ENTER)=5		PRIORITY(TRANSIT)=1			
RUN ONE	0.04	0.01	0.01	0.01	0.01	0.01
RUN TWO	0.04	0.01	0.01	0.01	0.01	0.01
RUN THREE	0.03	0.01	0.01	0.01	0.01	0.01
INPUT LEVEL IS 500 PACKETS/SEC						
	PRIORITY(ENTER)=1		PRIORITY(TRANSIT)=5			
RUN ONE	0.35	0.11	0.10	0.10	0.12	0.12
RUN TWO	0.35	0.10	0.09	0.09	0.13	0.13
RUN THREE	0.33	0.09	0.09	0.09	0.13	0.12
	PRIORITY(ENTER)=5		PRIORITY(TRANSIT)=1			
RUN ONE	0.35	0.09	0.11	0.10	0.13	0.12
RUN TWO	0.36	0.09	0.11	0.09	0.13	0.13
RUN THREE	0.33	0.08	0.09	0.11	0.12	0.11
INPUT LEVEL IS 625 PACKETS/SEC						
	PRIORITY(ENTER)=1		PRIORITY(TRANSIT)=5			
RUN ONE	0.44	0.12	0.12	0.12	0.16	0.16
RUN TWO	0.45	0.13	0.11	0.11	0.18	0.15
RUN THREE	0.42	0.11	0.11	0.13	0.15	0.15
	PRIORITY(ENTER)=5		PRIORITY(TRANSIT)=1			
RUN ONE	0.44	0.11	0.13	0.13	0.16	0.14
RUN TWO	0.44	0.12	0.12	0.11	0.17	0.15
RUN THREE	0.41	0.11	0.12	0.11	0.16	0.15
INPUT LEVEL IS 835 PACKETS/SEC						
	PRIORITY(ENTER)=1		PRIORITY(TRANSIT)=5			
RUN ONE	0.70	0.20	0.21	0.20	0.25	0.25
RUN TWO	0.70	0.20	0.20	0.19	0.25	0.24
RUN THREE	0.65	0.19	0.19	0.21	0.24	0.25
	PRIORITY(ENTER)=5		PRIORITY(TRANSIT)=1			
RUN ONE	0.69	0.20	0.19	0.22	0.25	0.24
RUN TWO	0.69	0.20	0.20	0.20	0.26	0.24
RUN THREE	0.67	0.20	0.20	0.19	0.23	0.24

TABLE E-17

NODAL SIMULATION: LINE UTILIZATION FACTORS

CHANNEL NUMBERS	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
	PRIORITY(ENTER)=1				PRIORITY(TRANSIT)=5				50 PACKETS/SEC	
RUN ONE	0.03	0.03	0.04	0.04	0.04	0.04	0.03	0.02	0.04	0.04
RUN TWO	0.04	0.04	0.03	0.04	0.03	0.03	0.04	0.05	0.03	0.03
RUN THREE	0.03	0.03	0.04	0.03	0.04	0.03	0.03	0.03	0.04	0.03
	PRIORITY(ENTER)=5				PRIORITY(TRANSIT)=1					
RUN ONE	0.04	0.03	0.03	0.03	0.04	0.03	0.03	0.03	0.04	0.04
RUN TWO	0.03	0.04	0.03	0.03	0.04	0.03	0.04	0.04	0.03	0.03
RUN THREE	0.03	0.03	0.03	0.04	0.04	0.04	0.03	0.03	0.03	0.03
	PRIORITY(ENTER)=1				PRIORITY(TRANSIT)=5				500 PACKETS/SEC	
RUN ONE	0.36	0.31	0.29	0.33	0.31	0.30	0.33	0.31	0.34	0.36
RUN TWO	0.37	0.37	0.31	0.30	0.33	0.31	0.38	0.41	0.30	0.27
RUN THREE	0.29	0.27	0.32	0.29	0.30	0.30	0.32	0.32	0.32	0.32
	PRIORITY(ENTER)=5				PRIORITY(TRANSIT)=1				500 PACKETS/SEC	
RUN ONE	0.28	0.32	0.31	0.32	0.39	0.35	0.33	0.29	0.31	0.31
RUN TWO	0.29	0.34	0.41	0.36	0.30	0.33	0.41	0.39	0.33	0.39
RUN THREE	0.30	0.29	0.33	0.36	0.38	0.39	0.29	0.33	0.27	0.21
	PRIORITY(ENTER)=1				PRIORITY(TRANSIT)=5				625 PACKETS/SEC	
RUN ONE	0.32	0.38	0.38	0.38	0.37	0.38	0.39	0.42	0.40	0.46
RUN TWO	0.45	0.47	0.39	0.43	0.44	0.43	0.51	0.52	0.40	0.35
RUN THREE	0.27	0.32	0.40	0.37	0.45	0.46	0.43	0.47	0.46	0.39
	PRIORITY(ENTER)=5				PRIORITY(TRANSIT)=1				625 PACKETS/SEC	
RUN ONE	0.39	0.38	0.36	0.43	0.45	0.46	0.40	0.42	0.33	0.34
RUN TWO	0.44	0.45	0.45	0.50	0.37	0.37	0.41	0.41	0.46	0.44
RUN THREE	0.39	0.36	0.38	0.42	0.33	0.40	0.37	0.39	0.38	0.38
	PRIORITY(ENTER)=1				PRIORITY(TRANSIT)=5				835 PACKETS/SEC	
RUN ONE	0.58	0.70	0.68	0.63	0.66	0.63	0.62	0.68	0.62	0.62
RUN TWO	0.70	0.66	0.66	0.70	0.63	0.52	0.71	0.67	0.64	0.61
RUN THREE	0.63	0.57	0.54	0.51	0.65	0.69	0.60	0.61	0.67	0.61
	PRIORITY(ENTER)=5				PRIORITY(TRANSIT)=1				835 PACKETS/SEC	
RUN ONE	0.63	0.61	0.73	0.68	0.55	0.64	0.69	0.62	0.56	0.55
RUN TWO	0.67	0.55	0.65	0.63	0.61	0.56	0.78	0.78	0.65	0.67
RUN THREE	0.61	0.65	0.68	0.72	0.69	0.75	0.67	0.59	0.59	0.69

TABLE E-18

NODAL SIMULATION: LINE UTILIZATION FACTORS

CHANNEL NUMBERS (11) (12) (13) (14) (15) (16) (17) (18) (19) (20)										
PRIORITY(ENTER)=1 PRIORITY(TRANSIT)=5 50 PACKETS/SEC										
RUN ONE	0.03	0.02	0.03	0.02	0.02	0.02	0.05	0.04	0.04	0.04
RUN TWO	0.03	0.03	0.02	0.02	0.02	0.02	0.04	0.03	0.06	0.05
RUN THREE	0.02	0.02	0.03	0.03	0.02	0.02	0.04	0.04	0.03	0.04
PRIORITY(ENTER)=5 PRIORITY(TRANSIT)=5 50 PACKETS/SEC										
RUN ONE	0.03	0.02	0.02	0.03	0.02	0.02	0.04	0.04	0.04	0.03
RUN TWO	0.03	0.02	0.03	0.03	0.02	0.02	0.05	0.04	0.04	0.04
RUN THREE	0.01	0.01	0.02	0.02	0.02	0.03	0.05	0.04	0.04	0.04
PRIORITY(ENTER)=1 PRIORITY(TRANSIT)=5 500 PACKETS/SEC										
RUN ONE	0.27	0.32	0.23	0.24	0.19	0.20	0.30	0.32	0.34	0.36
RUN TWO	0.20	0.20	0.18	0.18	0.18	0.25	0.41	0.42	0.41	0.40
RUN THREE	0.19	0.22	0.19	0.22	0.20	0.21	0.40	0.37	0.38	0.39
PRIORITY(ENTER)=5 PRIORITY(TRANSIT)=1 500 PACKETS/SEC										
RUN ONE	0.21	0.20	0.26	0.22	0.20	0.23	0.37	0.41	0.35	0.29
RUN TWO	0.23	0.20	0.21	0.20	0.21	0.25	0.38	0.44	0.42	0.37
RUN THREE	0.21	0.17	0.18	0.18	0.30	0.25	0.36	0.34	0.39	0.40
PRIORITY(ENTER)=1 PRIORITY(TRANSIT)=5 625 PACKETS/SEC										
RUN ONE	0.33	0.34	0.26	0.28	0.23	0.27	0.47	0.44	0.39	0.44
RUN TWO	0.32	0.36	0.25	0.21	0.24	0.24	0.51	0.51	0.51	0.60
RUN THREE	0.25	0.23	0.30	0.26	0.27	0.34	0.47	0.45	0.53	0.54
PRIORITY(ENTER)=5 PRIORITY(TRANSIT)=1 625 PACKETS/SEC										
RUN ONE	0.25	0.29	0.29	0.25	0.24	0.24	0.44	0.47	0.39	0.39
RUN TWO	0.30	0.27	0.23	0.24	0.26	0.31	0.53	0.55	0.42	0.45
RUN THREE	0.24	0.21	0.29	0.27	0.26	0.26	0.49	0.52	0.52	0.52
PRIORITY(ENTER)=1 PRIORITY(TRANSIT)=5 835 PACKETS/SEC										
RUN ONE	0.56	0.49	0.52	0.62	0.57	0.47	0.75	0.67	0.67	0.64
RUN TWO	0.43	0.51	0.54	0.56	0.50	0.51	0.72	0.70	0.76	0.76
RUN THREE	0.55	0.57	0.51	0.50	0.59	0.51	0.72	0.67	0.71	0.76
PRIORITY(ENTER)=5 PRIORITY(TRANSIT)=1 835 PACKETS/SEC										
RUN ONE	0.54	0.56	0.36	0.43	0.58	0.57	0.63	0.70	0.69	0.72
RUN TWO	0.47	0.52	0.54	0.47	0.54	0.68	0.72	0.71	0.68	0.69
RUN THREE	0.49	0.57	0.56	0.54	0.52	0.59	0.72	0.76	0.76	0.73

TABLE 1 19

NETWORK SIMULATION: RECORDED THROUGHPUT (PCKS/SEC)

I/P LEVEL	RUN 1	RUN 2	RUN 3	MEAN
PRIORITY(ENTER)=1		PRIORITY(TRANSIT)=5		
100	99	104	102	101.7
200	196	213	200	203.0
500	482	518	489	496.3
625	601	644	616	620.3
835	770	796	782	782.7
PRIORITY(ENTER)=5		PRIORITY(TRANSIT)=1		
100	100	104	100	101.3
200	198	212	198	202.7
500	483	515	484	494.0
625	614	620	613	615.7
835	773	793	757	774.3

NODAL SIMULATION: RECORDED THROUGHPUT (PCKS/SEC)

I/P LEVEL	RUN 1	RUN 2	RUN 3	MEAN
PRIORITY(ENTER)=1		PRIORITY(TRANSIT)=5		
50	51	51	46	49.3
500	478	482	445	468.3
625	582	587	555	574.7
835	767	735	702	734.7
PRIORITY(ENTER)=5		PRIORITY(TRANSIT)=1		
50	50	49	46	48.3
500	475	476	442	464.3
625	583	580	551	571.3
835	738	726	690	718.0

TABLE E-20

RECORDED NODAL THROUGHPUT IN PACKETS PER SECOND ON DATA BUS

NODAL NUMBERS		(1)	(2)	(3)	(4)	(5)	(6)
INPUT LEVEL IS 50 PACKETS PER SECOND							
CAPACITY	50.0	14.0	14.0	14.0	18.0	18.0	
	PRIORITY(ENTER)=1			PRIORITY(TRANSIT)=5			
RUN ONE	51.2	15.0	14.6	12.7	19.6	19.7	
RUN TWO	46.2	12.5	14.6	13.1	16.6	16.0	
RUN THREE	47.2	12.7	11.8	13.0	16.5	17.4	
MEAN	48.2	13.4	13.7	12.9	17.6	17.7	
	PRIORITY(ENTER)=5			PRIORITY(TRANSIT)=1			
RUN ONE	49.5	13.2	15.4	13.4	18.9	16.8	
RUN TWO	46.4	11.9	13.0	13.8	17.7	16.6	
RUN THREE	47.6	11.7	11.9	13.6	17.1	18.0	
MEAN	47.8	12.3	13.4	13.6	17.9	17.1	
INPUT LEVEL IS 500 PACKETS PER SECOND							
CAPACITY	500.0	140.0	140.0	140.0	180.0	180.0	
	PRIORITY(ENTER)=1			PRIORITY(TRANSIT)=5			
RUN ONE	492.2	131.7	128.1	125.2	181.4	178.5	
RUN TWO	447.7	118.2	124.2	119.5	175.6	158.0	
RUN THREE	475.1	133.3	126.9	141.1	166.7	166.7	
MEAN	471.7	127.7	126.4	128.6	174.6	167.7	
	PRIORITY(ENTER)=5			PRIORITY(TRANSIT)=1			
RUN ONE	482.6	125.3	144.3	129.0	182.2	181.4	
RUN TWO	448.4	113.5	127.1	147.6	160.6	155.8	
RUN THREE	464.8	121.3	132.5	136.0	162.7	169.0	
MEAN	465.3	120.0	134.6	137.5	168.5	168.7	
INPUT LEVEL IS 625 PACKETS PER SECOND							
CAPACITY	625.0	175.0	175.0	175.0	225.0	225.0	
	PRIORITY(ENTER)=1			PRIORITY(TRANSIT)=5			
RUN ONE	606.2	179.2	144.4	153.9	249.6	202.6	
RUN TWO	565.6	144.7	155.6	179.0	209.2	210.0	
RUN THREE	581.4	161.2	164.6	177.5	194.6	228.1	
MEAN	584.4	161.7	154.9	170.1	217.8	213.6	
	PRIORITY(ENTER)=5			PRIORITY(TRANSIT)=1			
RUN ONE	596.0	168.4	164.1	154.6	226.3	204.7	
RUN TWO	564.0	151.5	166.3	145.7	217.4	198.4	
RUN THREE	580.3	172.2	152.5	174.8	216.8	197.9	
MEAN	580.1	164.0	161.0	158.4	220.2	200.3	
INPUT LEVEL IS 835 PACKETS PER SECOND							
CAPACITY	835.0	233.8	233.8	233.8	300.6	300.6	
	PRIORITY(ENTER)=1			PRIORITY(TRANSIT)=5			
RUN ONE	769.8	203.1	217.0	202.1	309.0	276.9	
RUN TWO	725.0	199.7	192.7	203.8	255.2	264.3	
RUN THREE	765.4	202.7	192.1	222.8	287.2	271.3	
MEAN	753.4	201.8	200.6	209.6	283.8	270.8	
	PRIORITY(ENTER)=5			PRIORITY(TRANSIT)=1			
RUN ONE	755.0	211.8	219.3	186.4	279.6	278.6	
RUN TWO	711.2	216.0	187.6	200.8	256.5	257.6	
RUN THREE	743.2	179.8	199.6	201.7	296.9	271.8	
MEAN	736.5	202.5	202.2	196.3	277.7	269.3	

TABLE E-21

VARIATION OF SWITCHING ALGORITHM PRIORITIES

PRIORITY DELAY(TOTAL) DELAY(ENTRY) DELAY(TRANSIT)
ENTRY TRANSIT MEAN $\sqrt{\text{VAR}}$ MEAN $\sqrt{\text{VAR}}$ MEAN
DELAY FIGURES IN MILLISECONDS INPUT LEVEL 750 PCKS/SEC

1	1	87.18	73.77	42.37	39.28	44.81
1	2	75.81	56.24	36.83	39.72	38.91
2	1	76.45	66.24	35.17	32.37	41.28
1	3	85.14	69.27	46.43	51.71	43.86
3	1	76.75	67.30	35.42	26.95	41.33
1	5	75.57	57.37	39.23	40.07	36.34
5	1	71.70	58.70	25.25	11.23	46.45
1	8	76.52	55.31	41.75	38.96	34.77
8	1	75.37	85.48	30.72	27.22	44.65
1	10	80.23	61.94	45.18	49.85	35.05
10	1	77.75	83.46	27.93	19.01	49.82
1	15	80.89	58.87	43.93	42.51	36.96
15	1	77.60	76.07	30.17	23.72	47.43

BUS UTILIZATION FACTORS

PRIORITY	NODAL NUMBERS					
ENTRY TRANSIT	(1)	(2)	(3)	(4)	(5)	
1	1	0.27	0.24	0.28	0.32	0.25
1	2	0.27	0.25	0.26	0.30	0.24
2	1	0.26	0.23	0.29	0.31	0.24
1	3	0.27	0.25	0.26	0.31	0.24
3	1	0.28	0.25	0.27	0.28	0.24
1	5	0.27	0.24	0.27	0.30	0.23
5	1	0.27	0.24	0.28	0.31	0.26
1	8	0.27	0.25	0.28	0.33	0.25
8	1	0.27	0.21	0.28	0.32	0.23
1	10	0.26	0.25	0.28	0.33	0.25
10	1	0.27	0.24	0.28	0.31	0.23
1	15	0.27	0.25	0.28	0.32	0.24
15	1	0.25	0.24	0.26	0.30	0.24

TABLE E-22

LINE UTILIZATION FACTORS

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)
PRIORITY(ENTRY)=1		PRIORITY(TRANSIT)=1											
0.74	0.27	0.54	0.71	0.40	0.65	0.57	0.65	0.66	0.39	0.40	0.64	0.58	0.66
PRIORITY(ENTRY)=1		PRIORITY(TRANSIT)=2											
0.72	0.30	0.63	0.75	0.32	0.62	0.68	0.69	0.60	0.29	0.25	0.69	0.55	0.61
PRIORITY(ENTRY)=2		PRIORITY(TRANSIT)=1											
0.73	0.30	0.55	0.69	0.39	0.51	0.57	0.66	0.51	0.27	0.34	0.64	0.64	0.64
PRIORITY(ENTRY)=1		PRIORITY(TRANSIT)=3											
0.74	0.33	0.54	0.71	0.32	0.55	0.70	0.62	0.56	0.36	0.30	0.52	0.67	0.62
PRIORITY(ENTRY)=3		PRIORITY(TRANSIT)=1											
0.73	0.28	0.61	0.70	0.34	0.61	0.65	0.51	0.54	0.40	0.30	0.49	0.51	0.63
PRIORITY(ENTRY)=1		PRIORITY(TRANSIT)=5											
0.71	0.19	0.59	0.76	0.39	0.49	0.59	0.64	0.66	0.35	0.32	0.55	0.59	0.62
PRIORITY(ENTRY)=5		PRIORITY(TRANSIT)=1											
0.78	0.25	0.52	0.74	0.34	0.56	0.67	0.59	0.51	0.29	0.33	0.58	0.61	0.68
PRIORITY(ENTRY)=1		PRIORITY(TRANSIT)=8											
0.76	0.22	0.58	0.75	0.34	0.58	0.70	0.68	0.66	0.25	0.29	0.65	0.61	0.66
PRIORITY(ENTRY)=8		PRIORITY(TRANSIT)=1											
0.76	0.33	0.62	0.76	0.48	0.62	0.60	0.60	0.52	0.37	0.34	0.56	0.55	0.64
PRIORITY(ENTRY)=1		PRIORITY(TRANSIT)=10											
0.76	0.28	0.47	0.73	0.45	0.49	0.71	0.65	0.70	0.28	0.37	0.58	0.53	0.78
PRIORITY(ENTRY)=10		PRIORITY(TRANSIT)=1											
0.66	0.28	0.60	0.78	0.40	0.54	0.58	0.69	0.62	0.29	0.34	0.51	0.56	0.72
PRIORITY(ENTRY)=1		PRIORITY(TRANSIT)=15											
0.78	0.34	0.60	0.72	0.49	0.56	0.63	0.69	0.67	0.26	0.32	0.50	0.50	0.64
PRIORITY(ENTRY)=15		PRIORITY(TRANSIT)=1											
0.78	0.24	0.70	0.77	0.35	0.77	0.76	0.67	0.65	0.37	0.29	0.61	0.59	0.71

TABLE E-23

NODAL SIMULATION: RETRANSMISSION INTERVAL VARIATION

RETRM INTERVL	TOTAL DELAY		ENTRY DELAY		TRANSIT DEL MEAN	THROUGHPUT PKTS/SEC
	MEAN MILLISECONDS	VAR	MEAN	VAR		

INPUT LEVEL IS 625 PACKETS/SEC						
5	133.28	71.19	30.34	21.95	102.94	446
10	116.24	59.81	30.12	18.35	86.12	476
20	108.19	61.41	32.48	21.21	75.71	491
30	68.15	23.05	22.93	10.92	45.22	579
50	64.46	22.90	23.87	11.87	40.59	576
100	64.46	22.90	23.87	11.87	40.59	576

INPUT LEVEL IS 750 PACKETS/SEC						
5	142.31	85.78	38.20	26.94	104.11	505
50	96.42	36.73	35.18	22.96	61.24	743
500	93.76	44.18	38.73	36.05	55.03	732
1000	93.76	44.18	38.73	36.05	55.03	732

INPUT LEVEL IS 835 PACKETS/SEC						
5	171.56	93.73	60.00	56.08	111.56	477
10	169.23	86.71	60.73	49.40	108.50	479
20	162.49	90.79	66.23	55.44	96.26	500
30	157.92	85.37	82.89	72.75	75.03	811
50	164.79	95.63	91.09	85.77	73.70	789
100	164.79	95.63	91.09	85.77	73.70	789

BUS UTILIZATION FACTORS

RETRM INTRVL	NODAL NUMBERS					
	(1)	(2)	(3)	(4)	(5)	(6)

INPUT LEVEL IS 625 PACKETS/SEC						
5	0.36	0.10	0.11	0.09	0.13	0.14
10	0.39	0.11	0.10	0.12	0.14	0.13
20	0.38	0.10	0.10	0.11	0.15	0.13
30	0.43	0.13	0.13	0.12	0.14	0.16
50	0.43	0.13	0.11	0.12	0.16	0.15
100	0.43	0.13	0.11	0.12	0.16	0.15

INPUT LEVEL IS 750 PACKETS/SEC						
5	0.45	0.13	0.12	0.12	0.15	0.16
50	0.56	0.16	0.15	0.17	0.21	0.21
500	0.55	0.16	0.15	0.16	0.20	0.19
1000	0.55	0.16	0.15	0.16	0.20	0.19

INPUT LEVEL IS 835 PACKETS/SEC						
5	0.47	0.14	0.13	0.15	0.18	0.18
10	0.51	0.14	0.13	0.15	0.19	0.19
20	0.52	0.15	0.13	0.14	0.19	0.20
30	0.70	0.21	0.19	0.21	0.25	0.25
50	0.70	0.21	0.18	0.22	0.25	0.24
100	0.70	0.21	0.18	0.22	0.25	0.24

TABLE E-24

NODAL SIMULATION: RETRANSMISSION INTERVAL VARIATION

LINE UTILIZATION FACTORS

CHANNEL NOS		LINE UTILIZATION FACTORS (MILLISECONDS)				
RETRM. INTERVAL	100	50	30	20	10	5
INPUT LEVEL IS 625 PACKETS/SEC						
1	0.39	0.39	0.36	0.77	0.84	0.87
2	0.44	0.44	0.42	0.77	0.70	0.72
3	0.28	0.28	0.40	0.67	0.74	0.75
4	0.34	0.34	0.42	0.71	0.76	0.84
5	0.45	0.45	0.41	0.72	0.82	0.86
6	0.44	0.44	0.44	0.69	0.81	0.80
7	0.36	0.36	0.34	0.77	0.89	0.85
8	0.36	0.36	0.37	0.77	0.83	0.83
9	0.38	0.38	0.47	0.71	0.93	0.96
10	0.40	0.40	0.46	0.79	0.83	0.88
11	0.25	0.25	0.24	0.58	0.66	0.54
12	0.26	0.26	0.23	0.60	0.64	0.59
13	0.26	0.26	0.28	0.48	0.53	0.65
14	0.26	0.26	0.30	0.49	0.53	0.71
15	0.22	0.22	0.30	0.59	0.65	0.56
16	0.21	0.21	0.24	0.65	0.69	0.66
17	0.51	0.51	0.44	0.89	0.86	0.96
18	0.52	0.52	0.44	0.84	0.96	0.98
19	0.42	0.42	0.44	0.81	0.90	0.92
20	0.44	0.44	0.51	0.81	0.95	0.95
INPUT LEVEL IS 835 PACKETS/SEC						
1	0.54	0.54	0.63	0.84	0.93	0.93
2	0.61	0.61	0.61	0.90	0.89	0.89
3	0.61	0.61	0.68	0.84	0.96	0.96
4	0.71	0.71	0.70	0.89	0.88	0.88
5	0.64	0.64	0.74	0.91	0.92	0.94
6	0.68	0.68	0.72	0.89	0.95	0.94
7	0.75	0.75	0.70	0.90	0.95	0.95
8	0.67	0.67	0.67	0.92	0.99	0.97
9	0.50	0.50	0.58	0.89	0.96	0.96
10	0.55	0.55	0.62	0.83	0.96	0.96
11	0.57	0.57	0.56	0.90	0.94	0.94
12	0.54	0.54	0.54	0.92	0.93	0.94
13	0.39	0.39	0.50	0.85	0.87	0.87
14	0.46	0.46	0.45	0.85	0.81	0.85
15	0.46	0.46	0.52	0.80	0.92	0.92
16	0.49	0.49	0.57	0.87	0.95	0.92
17	0.65	0.65	0.62	0.93	0.96	0.96
18	0.75	0.75	0.76	0.88	0.96	0.99
19	0.77	0.77	0.65	0.91	0.94	0.94
20	0.70	0.70	0.77	0.97	1.00	0.99

TABLE E-25

NODAL SIMULATION: RETRANSMISSION INTERVAL VARIATION

LINE UTILIZATION FACTORS

CHANNEL NOS RETRM INTERVAL 1000	LINE UTILIZATION FACTORS (MILLISECONDS)			
	500	50	5	
INPUT LEVEL IS 750 PACKETS/SEC				
1	0.53	0.53	0.53	0.82
2	0.54	0.54	0.52	0.86
3	0.48	0.48	0.51	0.81
4	0.53	0.53	0.56	0.78
5	0.49	0.49	0.47	0.75
6	0.50	0.50	0.48	0.74
7	0.41	0.41	0.49	0.95
8	0.50	0.50	0.54	0.97
9	0.51	0.51	0.53	0.96
10	0.41	0.41	0.45	0.93
11	0.33	0.33	0.34	0.70
12	0.37	0.37	0.39	0.72
13	0.31	0.31	0.34	0.69
14	0.32	0.32	0.28	0.71
15	0.33	0.33	0.36	0.73
16	0.36	0.36	0.37	0.79
17	0.63	0.63	0.58	0.98
18	0.57	0.57	0.62	0.99
19	0.55	0.55	0.64	0.91
20	0.56	0.56	0.60	0.96

TABLE E-26

NODAL SIMULATION: BUS RATE & ALG. EXEC. TIME VARIATION						
BUS RATE	TOTAL DELAY		ENTRY DELAY		TRANSIT DELAY	THROUGHPUT
BPS	MEAN	DEV	MEAN	DEV	MEAN	
BPS	MILLISECONDS					PKTS/SEC
SWITCH ALG. EXECUTION TIME = 0.5 mSECS						
8 E 5	17.49	12.91	1.29	0.78	16.20	464
8 E 5	16.74	13.79	1.06	0.53	15.68	474
10E 5	17.89	14.39	0.98	0.51	16.91	479
15E 5	16.19	12.68	0.81	0.23	15.38	463
SWITCH ALG. EXECUTION TIME = 1.0 mSECS						
1 E 5	120.99	59.98	26.30	45.72	94.69	210
4 E 5	21.28	14.97	2.56	1.87	18.72	460
6 E 5	19.90	15.18	1.93	1.06	17.97	476
8 E 5	19.12	13.49	1.68	0.71	17.44	477
10E 5	17.31	12.28	1.53	0.51	15.78	468
15E 5	17.66	13.05	1.40	0.38	16.26	464
SWITCH ALG. EXECUTION TIME = 5.0 mSECS						
1 E 5	181.75	98.19	43.59	87.74	138.16	218
4 E 5	36.33	16.22	9.12	29.77	27.21	464
6 E 5	34.47	15.93	8.64	3.12	25.83	466
8 E 5	34.92	15.08	8.39	3.06	26.53	466
10E 5	34.44	15.19	8.62	3.04	25.82	474
15E 5	33.54	15.29	7.88	2.61	25.66	478
SWITCH ALG. EXECUTION TIME = 10.0 mSECS						
1 E 5	166.34	80.27	45.78	63.66	120.56	181
4 E 5	65.33	22.26	21.81	8.88	43.52	476
6 E 5	62.77	21.85	21.01	9.63	41.76	462
8 E 5	59.73	20.78	19.89	8.53	39.84	457
10E 5	60.93	20.85	19.75	8.02	41.18	465
15E 5	58.68	19.12	18.95	7.14	39.73	480
SWITCH ALG. EXECUTION TIME = 15.0 mSECS						
1 E 5	154.04	52.23	55.14	39.05	98.90	170
4 E 5	99.58	30.73	36.57	17.98	63.01	451
6 E 5	92.55	28.74	34.72	15.56	58.03	449
8 E 5	91.35	30.43	34.18	19.27	56.57	448
10E 5	99.51	36.50	36.13	19.87	63.38	448
15E 5	90.07	30.71	33.60	17.25	56.47	461

LOG FORM FOR BUS RATE VARIATION (BUS RATE IS 1.5 MBPS)

ALG EXEC TIME	10LOGx	MEAN TOTAL DELAY (mSECS)
15 mSECS	-18.24	90.07
10 mSECS	-20.00	58.68
5 mSECS	-23.01	33.54
1 mSECS	-30.00	17.66
0.5mSECS	-33.01	16.19

TABLE E-27

BUS UTILIZATION FACTORS

NODAL SIMULATION: BUS RATE & ALG. EXEC. TIME VARIATION

BUS RATE BPS	(1)	(2)	(3)	(4)	(5)	(6)
SWITCH ALG. EXECUTION TIME = 0.5 mSECS						
6 E 5	0.37	0.11	0.09	0.11	0.13	0.14
8 E 5	0.28	0.08	0.08	0.08	0.10	0.10
10E 5	0.23	0.07	0.05	0.07	0.09	0.08
15E 5	0.15	0.04	0.04	0.04	0.05	0.05
SWITCH ALG. EXECUTION TIME = 1.0 mSECS						
1 E 5	0.99	0.32	0.34	0.41	0.60	0.59
4 E 5	0.55	0.16	0.16	0.16	0.19	0.19
6 E 5	0.38	0.11	0.11	0.11	0.13	0.14
8 E 5	0.28	0.08	0.08	0.08	0.11	0.10
10E 5	0.31	0.08	0.08	0.09	0.11	0.11
15E 5	0.15	0.04	0.04	0.04	0.06	0.06
SWITCH ALG. EXECUTION TIME = 5.0 mSECS						
1 E 5	0.99	0.34	0.40	0.39	0.59	0.54
4 E 5	0.55	0.14	0.16	0.16	0.20	0.20
6 E 5	0.37	0.10	0.10	0.11	0.13	0.12
8 E 5	0.28	0.08	0.07	0.08	0.10	0.10
10E 5	0.23	0.06	0.06	0.06	0.09	0.08
15E 5	0.15	0.04	0.04	0.04	0.06	0.05
SWITCH ALG. EXECUTION TIME = 10.0 mSECS						
1 E 5	0.89	0.38	0.30	0.39	0.53	0.52
4 E 5	0.57	0.16	0.15	0.15	0.22	0.20
6 E 5	0.37	0.10	0.11	0.10	0.13	0.13
8 E 5	0.27	0.07	0.07	0.08	0.10	0.10
10E 5	0.22	0.06	0.07	0.06	0.08	0.08
15E 5	0.15	0.05	0.04	0.04	0.06	0.05
SWITCH ALG. EXECUTION TIME = 15.0 mSECS						
1 E 5	0.94	0.30	0.32	0.32	0.57	0.48
4 E 5	0.54	0.15	0.14	0.16	0.19	0.21
6 E 5	0.36	0.10	0.09	0.10	0.13	0.13
8 E 5	0.27	0.07	0.07	0.08	0.10	0.10
10E 5	0.22	0.06	0.06	0.06	0.08	0.08
15E 5	0.15	0.04	0.04	0.04	0.05	0.05

TABLE E-28

NODAL SIMULATION: BUS RATE & ALG. EXEC. TIME VARIATION
LINE UTILIZATION FACTORS

BUS RATE (MBPS)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
CHANNEL NUMBERS																				
SWITCH ALG. EXECUTION TIME = 0.5 mSECS																				
0.6	.36	.38	.29	.27	.33	.33	.33	.32	.26	.27	.15	.21	.16	.16	.20	.20	.35	.32	.43	.41
0.8	.31	.33	.34	.28	.33	.31	.26	.29	.37	.38	.22	.26	.21	.22	.21	.23	.39	.40	.27	.30
1.0	.30	.32	.32	.23	.42	.33	.33	.38	.40	.33	.21	.21	.17	.19	.20	.23	.49	.36	.35	.38
1.5	.30	.28	.32	.31	.37	.33	.31	.29	.29	.36	.17	.19	.19	.21	.16	.20	.40	.34	.40	.38
SWITCH alg. execution time = 1.6 mSECS																				
0.1	.18	.20	.20	.17	.15	.15	.23	.17	.18	.12	.11	.17	.19	.25	.28	.23	.37	.29	.33	.36
0.4	.36	.32	.27	.29	.34	.35	.26	.30	.34	.29	.21	.20	.23	.24	.18	.21	.42	.32	.36	.31
0.6	.41	.36	.31	.32	.32	.33	.30	.34	.29	.34	.22	.20	.22	.25	.23	.19	.40	.37	.42	.38
0.8	.39	.35	.29	.28	.32	.31	.41	.32	.27	.29	.19	.21	.27	.21	.15	.26	.40	.41	.36	.36
1.0	.31	.34	.31	.29	.30	.27	.28	.33	.31	.29	.13	.17	.20	.21	.23	.27	.27	.36	.37	.34
1.5	.32	.32	.37	.35	.26	.26	.26	.33	.29	.36	.19	.22	.24	.19	.20	.21	.45	.39	.41	.39
SWITCH ALG. EXECUTION TIME = 5.0 mSECS																				
0.1	.13	.12	.18	.23	.19	.16	.22	.19	.21	.23	.13	.16	.14	.15	.22	.22	.27	.26	.27	.32
0.4	.30	.29	.35	.34	.33	.32	.32	.29	.30	.34	.19	.14	.21	.23	.18	.20	.34	.34	.39	.40
0.6	.29	.28	.35	.34	.35	.31	.34	.33	.31	.28	.19	.20	.10	.21	.22	.20	.37	.35	.30	.36
0.8	.30	.31	.34	.32	.34	.36	.30	.29	.31	.32	.23	.23	.21	.20	.18	.16	.37	.37	.36	.37
1.0	.30	.32	.33	.33	.34	.36	.32	.32	.37	.36	.22	.19	.21	.21	.18	.18	.39	.37	.35	.32
1.5	.28	.31	.32	.31	.33	.32	.31	.39	.36	.36	.20	.23	.24	.30	.19	.24	.44	.38	.30	.30
SWITCH ALG. EXECUTION TIME = 10.6 mSECS																				
0.1	.12	.21	.20	.20	.17	.18	.20	.18	.16	.17	.23	.14	.11	.17	.21	.21	.41	.28	.26	.35
0.4	.29	.31	.31	.32	.39	.37	.34	.33	.36	.36	.24	.28	.25	.22	.19	.17	.38	.42	.32	.33
0.6	.31	.30	.36	.35	.28	.31	.32	.32	.28	.31	.19	.21	.20	.26	.23	.19	.35	.35	.35	.37
0.8	.24	.28	.27	.28	.32	.33	.32	.32	.35	.35	.20	.20	.16	.20	.19	.18	.38	.34	.38	.38
1.0	.31	.31	.36	.37	.34	.33	.35	.31	.25	.28	.24	.24	.20	.23	.20	.20	.36	.40	.36	.33
1.5	.33	.31	.39	.38	.39	.26	.34	.38	.37	.37	.27	.24	.23	.22	.21	.26	.35	.42	.33	.32
SWITCH ALG. EXECUTION TIME = 15.0 mSECS																				
0.1	.12	.14	.11	.17	.15	.14	.29	.22	.25	.16	.17	.14	.18	.15	.20	.19	.34	.32	.26	.29
0.4	.29	.30	.32	.33	.32	.32	.30	.29	.31	.32	.18	.20	.18	.18	.21	.21	.40	.35	.37	.41
0.6	.28	.28	.26	.26	.33	.30	.35	.38	.28	.28	.24	.25	.16	.17	.21	.22	.31	.37	.36	.39
0.8	.32	.30	.29	.26	.34	.37	.30	.31	.32	.27	.18	.22	.20	.19	.21	.22	.37	.41	.32	.39
1.0	.29	.31	.30	.29	.30	.36	.27	.30	.35	.30	.23	.23	.20	.23	.21	.20	.34	.32	.35	.40
1.5	.27	.28	.32	.30	.32	.30	.33	.28	.34	.32	.20	.18	.21	.22	.21	.20	.37	.40	.37	.33

TABLE E-29

NETWORK SIMULATION: SWITCHING ALG EXEC TIME IS 1MSEC

I/P LEVEL PKGS/SEC	TOTAL DELAY		ENTRY DELAY		TRANSIT DEL MEAN
	MEAN	$\sqrt{\text{VAR}}$ MILLISECONDS	MEAN	$\sqrt{\text{VAR}}$	
625	20.41	24.31	1.87	0.75	18.54
835	20.27	22.53	2.03	0.93	18.24
1000	39.58	40.95	2.18	2.11	37.40

BUS UTILIZATION FACTORS

I/P LEVEL	NODAL NUMBERS				
	(1)	(2)	(3)	(4)	(5)
625	0.27	0.21	0.25	0.26	0.20
835	0.30	0.25	0.31	0.32	0.25
1000	0.37	0.32	0.39	0.43	0.31

LINE UTILIZATION FACTORS

CHANNEL	INPUT TRAFFIC IN PACKETS/SEC		
	625	835	1000
1	0.83	0.80	0.93
2	0.20	0.28	0.40
3	0.50	0.57	0.72
4	0.67	0.83	0.92
5	0.25	0.33	0.48
6	0.50	0.63	0.75
7	0.61	0.67	0.60
8	0.43	0.66	0.90
9	0.44	0.52	0.72
10	0.33	0.21	0.53
11	0.28	0.30	0.65
12	0.47	0.70	0.82
13	0.51	0.56	0.75
14	0.46	0.53	0.75

TABLE E-30

NODAL SIMULATION: SWITCHING ALG EXEC TIME IS 1MSEC DOUBLE CHANNEL

I/P LEVEL PKGS/SEC	TOTAL DELAY		ENTRY DELAY		TRANSIT DEL MEAN
	MEAN	VAR MILLISECONDS	MEAN	VAR	
500	19.90	15.18	1.93	1.06	17.97
625	23.74	16.68	1.95	0.93	21.67
835	30.06	23.52	2.13	1.14	27.93
1000	41.33	28.00	2.45	1.54	38.88

BUS UTILIZATION FACTORS

I/P LEVEL	NODAL NUMBERS					
	(1)	(2)	(3)	(4)	(5)	(6)
500	0.35	0.10	0.10	0.10	0.12	0.13
625	0.57	0.16	0.15	0.18	0.19	0.21
835	0.68	0.20	0.18	0.20	0.25	0.25
1000	0.84	0.25	0.25	0.22	0.32	0.31

LINE UTILIZATION FACTORS

CHANNEL	INPUT TRAFFIC IN PACKETS/SEC			
	500	625	835	1000
1	0.41	0.48	0.58	0.76
2	0.36	0.46	0.65	0.70
3	0.31	0.53	0.60	0.85
4	0.32	0.58	0.60	0.76
5	0.32	0.50	0.60	0.71
6	0.33	0.61	0.74	0.71
7	0.30	0.42	0.54	0.73
8	0.34	0.47	0.55	0.74
9	0.29	0.48	0.55	0.65
10	0.34	0.51	0.59	0.66
11	0.22	0.31	0.42	0.62
12	0.20	0.35	0.39	0.47
13	0.27	0.28	0.33	0.52
14	0.25	0.30	0.32	0.55
15	0.23	0.33	0.48	0.49
16	0.19	0.36	0.42	0.50
17	0.40	0.59	0.67	0.89
18	0.37	0.57	0.65	0.96
19	0.42	0.59	0.69	0.81
20	0.38	0.55	0.75	0.81

TABLE E-31

NODAL SIMULATION: SINGLE CHANNEL CONNECTION

DELAY VALUES FOR NETWORK SIMULATION IN MILLISECONDS

I/P LEVEL	TOTAL DELAY		ENTRY DELAY		TRANSIT DEL	THROUGHPUT
PCKS/SEC	MEAN	$\sqrt{\text{VAR}}$	MEAN	$\sqrt{\text{VAR}}$	MEAN	PCKS/SEC
	MILLISECONDS					

ALGORITHM EXECUTION TIME 1MSECS

100	16.56	13.71	1.60	0.58	14.96	96
200	28.86	20.47	1.67	0.67	27.19	191
400	51.31	51.54	1.74	0.69	49.57	338
500	155.53	118.84	1.82	0.80	153.71	364

ALGORITHM EXECUTION TIME 10MSECS

100	45.85	16.67	13.33	4.36	32.52	98
200	54.86	19.58	16.56	6.94	38.30	195
400	89.95	47.70	27.33	23.35	62.62	373
500	186.66	128.09	99.24	100.97	87.42	398

BUS UTILIZATION FACTORS

I/P LEVEL	NODAL NUMBERS					
PCKS/SEC	(1)	(2)	(3)	(4)	(5)	(6)

ALGORITHM EXECUTION TIME 1MSECS

100	0.07	0.02	0.02	0.02	0.02	0.03
200	0.14	0.04	0.04	0.04	0.05	0.05
400	0.26	0.07	0.07	0.07	0.10	0.10
500	0.37	0.09	0.08	0.09	0.14	0.13

ALGORITHM EXECUTION TIME 10MSECS

100	0.07	0.02	0.02	0.02	0.03	0.03
200	0.14	0.04	0.04	0.04	0.05	0.05
400	0.28	0.08	0.08	0.08	0.10	0.10
500	0.38	0.11	0.10	0.11	0.14	0.13

TABLE E-32

NODAL SIMULATION: SINGLE CHANNEL CONNECTION

LINE UTILIZATION FACTORS

CHANNEL	TRAFFIC INPUT IN PACKETS/SEC			
	100	200	400	500
	ALGORITHM EXECUTION TIME 1MSECS			
1	0.14	0.36	0.64	0.96
2	0	0	0	0
3	0.16	0.55	0.66	0.95
4	0	0	0	0
5	0.13	0.53	0.63	0.93
6	0	0	0	0
7	0.11	0.64	0.67	0.96
8	0	0	0	0
9	0.12	0.52	0.66	0.94
10	0	0	0	0
11	0.08	0.35	0.53	0.82
12	0	0	0	0
13	0.08	0.45	0.54	0.87
14	0	0	0	0
15	0.09	0.29	0.53	0.87
16	0	0	0	0
17	0.17	0.67	0.70	0.98
18	0	0	0	0
19	0.15	0.42	0.65	0.97
20	0	0	0	0
	ALGORITHM EXECUTION TIME 10MSECS			
1	0.14	0.24	0.53	0.75
2	0	0	0	0
3	0.14	0.32	0.48	0.74
4	0	0	0	0
5	0.12	0.28	0.57	0.69
6	0	0	0	0
7	0.13	0.24	0.57	0.73
8	0	0	0	0
9	0.13	0.28	0.52	0.74
10	0	0	0	0
11	0.09	0.16	0.31	0.49
12	0	0	0	0
13	0.08	0.22	0.33	0.53
14	0	0	0	0
15	0.08	0.17	0.37	0.57
16	0	0	0	0
17	0.14	0.36	0.66	0.80
18	0	0	0	0
19	0.17	0.26	0.62	0.85
20	0	0	0	0

TABLE E-33

REFERENCES

REFERENCES

1. Martin, J
Telecommunications and the Computer
Prentice-Hall - 1976
2. Christiansen, H M and Senft, F
'PCM 30/32, a Time-Division Multiplex Transmission System
for Local and Short-Haul Networks'
Siemens Review - 38, pp. 190 - 194, 1971
3. Davies, D W and Barber, D L
Communication Networks for Computers
Wiley - 1973
4. Roberts, L G
'The Evolution of Packet-Switching'
Proceedings of the IEEE - Vol. 66, No. 11, pp. 1307 - 1313, 1978
5. Fick, H and Kammerl, A
'Multiplexing and Switching of Data in Synchronous
Networks and the Realization in the EDS system'
IEEE Transactions on Communication - Vol. 61, No. 8,
pp. 925 - 927, 1973
6. Schwartz, M
Computer Communication Network Design and Analysis
Prentice-Hall - 1977
7. Gray, P J
'Network Services in Systems Network Architecture'
IEEE Transactions on Communication - Vol. COMM-25, No. 1,
pp 104 - 116, 1977

8. Fischer, M J and Harris, P C
'A Model for Evaluating the Performance of an Integrated
Circuit and Packet Switched Multiplex Structure'
IEEE Transaction on Communication - Vol. 24, No. 2,
pp. 195 - 202, 196
9. Jenny, C J and Kunmerle
'Distributed Processing Within an Integrated Circuit, Packet
Switching Node'
IEEE Transactions on Communications - Vol. 24, No. 10,
pp. 1089 - 1100, 1976
10. Miyahara, H, Teshigawara, T, Hasegawa, T
'Delay and Throughput Evaluation of Switching Methods in
Computer Communication Networks'
IEEE Transactions in Communications - Vol. 26, No. 3,
pp. 337 - 343, 1978
11. Newport, C B and Ryzlak, Y
'Communication Processors'
Proceedings of the IEEE - Vol. 60, No. 11, pp. 1321 - 1332, 1972
12. Mills, D L
'Communication Software'
Proceedings of the IEEE - Vol. 60, No. 11, pp. 1333 - 1341, 1972
13. Steele, J and Mattson, R C
'Architecture of a Universal Communications Processor'
Computer Design - No. 11, pp. 63 - 68, 1973

14. Mann, W F, et al
'A network-oriented multiprocessor'
Proc. AFIPS National Computer Conference
Montvale, Vol. 45, pp. 533 - 540, 1976
15. Roberts, L G
'Packet Network Design - the third generation'
IFIP Congress Proceedings
North Holland Publishing, pp. 541 - 545, 1977
16. Sarch, R
'Communication Processors - Trends'
Data Communications - Special Report, Jan. 1979
17. Fuchs and Jackson, I E
'Estimates of Distributions of Random Variables for Certain
Computer Communication Traffic Models'
Communications of the ACM - No. 12, Vol. 13, pp. 752 - 757, 1970
18. Crowther, W R, et al
'Issues in Packet-Switching Network Design'
AFIPS Conference Proceedings
Montvale, Vol. 44, pp. 161 - 175, 1975
19. Kleinrock, L
'Principles and Lessons in Packet Communications'
Proceedings of the IEEE - Vol. 66, No. 11, pp. 1320 - 1329, 1978
20. Frank, H and Kahn, R E
'Computer Communication Network Design - Experience with
Theory and Practice'
Proc. of AFIPS Spring Joint Computer Conference
Montvale, Vol. 37, pp. 255 - 270, 1972

21. Kleinrock, L
Queueing Systems, Theory and Application, Vol. 2
Wiley, 1976
22. Davis, D W
'The Control of Congestion in Packet-Switch Networks'
IEEE Transactions on Communications
Vol. 20, No. 3, pp. 546 - 556, 1972
23. Kahn, R E and Crowther, P
'Flow Control in a Resource-Sharing Network'
IEEE Transactions on Communications
Vol. 20, No. 3, pp. 539 - 565, 1972
24. Lain, S S and Reiser, M
'Congestion Control by Input Buffer Limits - An Analysis'
IEEE Transactions on Communications
Vol. 27, No. 1, pp. 127 - 133, 1979
25. Pennotti, M C and Schwarz, M
'Congestion Control in Store-and-Forward Links'
IEEE Transactions on Communications
Vol. 23, No. 12, pp. 1434 - 1443, 1975
26. Pauzin, L and Zimmerman, H
'A Tutorial on Protocols'
Proceedings of the IEEE
Vol. 16, No. 11, p. 1346 - 1370, 1978

27. Gray, J P
 'Line Control Procedures'
 Proceedings of the IEEE
 Vol. 60, No. 11, pp. 1301 - 1312, 1972
28. Ferrari, D
 Computer Systems Performance Evaluation
 Prentice-Hall, 1978

Author Reinink K

Name of thesis Data Adaptor unit for an electronic exchange: system design and simulation study 1981

PUBLISHER:

University of the Witwatersrand, Johannesburg

©2013

LEGAL NOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed, or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page) for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.

Author Reinink K

Name of thesis Data Adaptor unit for an electronic exchange: system design and simulation study 1981

PUBLISHER:

University of the Witwatersrand, Johannesburg

©2013

LEGAL NOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed, or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page) for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.