

Packet transmission of data on a multibeam  
satellite system with high telephonic  
traffic.

Simulation of the system.

Program description.

Renzo Beltrame

Report C82-16

Istituto CNUCE  
PISA  
June 1982

1st Draft

DISTRIBUTION : ESA - Telespazio S.p.A.

NATURE : Report

PREVIOUS VERSIONS :

(c) Copyright CNUCE Pisa 1982

TABLE OF CONTENTS

1.0	Structure of the simulation program	1
1.1	Overall structure	1
2.0	Input format	9
3.0	Output format	14
4.0	Sketch of a simulation run	15
	References	21

1.0 STRUCTURE OF THE SIMULATION PROGRAM

This simulation program was derived from a pre-existing one which simulated a satellite network based on a single-transponder total-coverage telecommunication system.

Protocols of access to the single common channel were basically F-TOMA, R-TDMA and a S-ALOHA in the form implemented on SHTNET.

We modified this package, so that these basic MAPs (Multiple Access Protocols) can also work embedded in a multibeam satellite system on which a large part of the traffic was telephonic traffic.

To obtain this result we added in the ground stations a process which enables and disables the sender according to the time plan of spot-to-spot interconnection. The same process performs the swapping of the various element employed by the sender - queues, counters, tables, etc. - and related to a certain spot-to-spot interconnection from the active area to a wait area.

In order to preserve a good simulation of different link conditions we introduced the simulation of the up-links and the down-links, and, obviously, of the on-board switch matrix controller too.

In this way we can easily introduce in the future also the simulation of on-board memories and related processing of data.

1.1 OVERALL STRUCTURE

The overall structure of the program is the following:

```

*** Program SIMITLS1 - TDMA simulation on satellite channel ***
*** Multibeam system - Org/Dest variable                      ****
*** MAP: F-TDMA,S-ALOHA,R-TDMA                               ****
*** Version 1.1 - June 1982                                  ****
*** CNUCE - PISA - Renzo Beltrame                           ****;
begin
!*****
*                               GLOBALS DECLARATION                               *
*****;
READGLOBALS;
.....
.....
!*****
*                               SIMULATION PROGRAM BLOCK                               *

```

```

*****;
Simulation begin
  ref(UTILITY) UTI;
  ref(REPORT) REP;
  ref(MESSAGE) array RECMSG(0:NBEAM);
  ref(ONBOARDSWITCH) ONBOARDSW;
  ref(TRANSPIN) array UPTRANS(0:UPTRSP);
  ref(TRANSPORT) array DWTRANS(0:DWTRSP);
  ref(DESTBEAMPLAN) array UPBMPLANNER(0:NBEAM);
  ref(STATION) array UPSTATION(0:TNSTAT);
  ref(SINKMSG) array USINKMSG(0:TNSTAT);
!*****
*
*           CHANNEL PROCESSORS
*
*****;

!**** controller of on-board switching matrix ****;
Process class ONBOARDSWITCH(NSTATE,UPTRSP);
  short integer NSTATE,UPTRSP;
  .....
  .....

!**** statistics and collision ceck ****;
Process class PACKVALIDATION(CTRANS);
  .....
  .....
!**** up-link transponder ****;
Process class TRANSPIN(ID);
  !** id 1-origin ****;
  short integer ID;
  .....
  .....
Process class TRANSPORT(DB);
  !** db 1-origin ****;
  short integer DB;
  .....
  .....
end****OF CLASS TRANSPORT****;
!*****
*
*           PACKETS TYPE DEFINITION
*
*****;

Link class MESSAGE(STAT);
  ref(STATION) STAT;
  .....
end****OF CLASS MESSAGE****;

MESSAGE class RESVSLOT;
  .....
  .....
end****OF CLASS RESVSLOT****;

MESSAGE class DATASLOT;
  .....
end****OF CLASS DATASLOT****;

```

```

!*****
*                               HOST SIMULATION PROCESSORS                               *
*****

```

```

!*** generate packets *****;
Process class GENMSG (STAT,RATE,SD1,TYPE,NPACK,
                     MPACK,MPACKFLAG,DESTINA,DISTRIB);

```

```

    ref (STATION) STAT;
    real RATE,SD1,TYPE;
    short integer NPACK,MPACK;
    boolean MPACKFLAG;
    real array DISTRIB;
    boolean array DESTINA;

```

```
begin
```

```
.....
```

```
    procedure DESTINATION;
```

```
.....
```

```
!*** enques packets following destination *****;
```

```

    procedure ENQUEUE (MSG,STAT,K);
        ref (DATASLOT) MSG;
        ref (STATION) STAT;
        integer K;

```

```
.....
```

```
end****OF CLASS GENMSG****;
```

```

!*** sink packets *****;
Process class SINKMSG (STAT,SINKRATE);

```

```

    ref (STATION) STAT;
    real SINKRATE;

```

```
.....
```

```
.....
```

```
end****OF CLASS SINKMSG****;
```

```

!*****
*                               UTILITY PROCEDURES FOR STATIONS                               *
*****

```

```

class UTILITY;
begin

```

```

!*****
*                               SENDER SCHEDULING AND TRANSMISSIONS                               *
*****

```

```

!**** send scheduling*****;
    short integer procedure SENDSCHED (STAT,TM,TRLENGTH);
        name TRLENGTH;
        ref (STATION) STAT;
        real TM,TRLENGTH;

```

```
.....
```

```
.....
```

```

***transmit reservation slot*****;
procedure RTRANSMIT (STAT,TRTIME,TRLENGTH);
  name TRLENGTH;
  ref (STATION) STAT;
  real TRTIME,TRLENGTH;
.....
.....

*** transmit data slot from new queue or retrans queue ***;
procedure DTRANSMIT (TYPEQUEUE,STAT,TRTIME);
  ref (STATION) STAT;
  real TRTIME;
  short integer TYPEQUEUE;
.....
.....

***test buffer space available*****;
integer procedure BUFFLOAD (STAT);
  ref (STATION) STAT;
.....
.....

procedure STATQUEUE (IDQUEUE,STAT,CTIME);
  ref (STATION) STAT;
  real CTIME;
  short integer IDQUEUE;
.....
.....

*** statistics procedure *****;
procedure DATASTAT (SLOT,STAT,CURRTIME);
  ref (MESSAGE) SLOT;
  ref (STATION) STAT;
  real CURRTIME;
.....
.....

**** queue scheduling *****;
boolean procedure RTDMAQSCHED (STAT,CURRTIME);
  ref (STATION) STAT;
  real CURRTIME;
.....
.....
*****
*                               *
*                               *
*****

```

```

*** store reservation data *****;
procedure RESVSTORE (RSLOT,STAT);
  ref (RESVSLOT) RSLOT;
  ref (STATION) STAT;

```

```

.....
.....

!*** updates reservation table *****;
  procedure RSVUPDATE (STAT);
    ref (STATION) STAT;
.....
.....

!*** reset synchronism of scheduling *****;
  procedure SYNCRESET (STAT);
    ref (STATION) STAT;
  begin

    procedure DUMMYSEND;
.....
.....

.....

end *****OF CLASS UTILITY*****;
!*****
* STATISTICS AND REPORT PROCEDURES *
*****;

class DISTPACK (VBEG, VEND, STEPS, GEOFLAG);
  real VBEG, VEND;
  integer STEPS, GEOFLAG;
.....
.....

class REPORT;
begin
  text array TESTHEAD (1:4);
  ref (DISTPACK) array DISTRIBUTION (0:TSTATNUM, 0:8);
  real array REPSTAT (0:TNSTAT+UPTRSP+DWTRSP+2, 0:10, 0:7);
  !*** 0:tnstat stat.id
      *
      0 - effective input packets rate (bit/sec)
      1 - refused input packets rate "
      2 - effective output packets rate "
      3 - traffic "
      4 - throughput "
      5 - one-way delay (prdelay)
      6 - two-way delay "
      7 - retransmission queue (conditioned) (pkt)
report only (unconditioned) "
      8 - new packets queue (conditioned) "
report only (unconditioned) "
      9 - waiting for ack queue (conditioned) "
report only (unconditioned) "
      10 - output queue (conditioned) "
report only (unconditioned) "

```

```

*
0 - minimum
1 - maximum
2 - probability
3 - 1st order moment
4 - 2nd order moment
5 - actual value of stochastic variable
6 - cumulative value of 2nd order moment
7 - cumulative time value
*
integer I,J,REPDIM,INITQUEUE;

!*** data for report procedure *****;
procedure MEASURE(ID,TYPE,VAL,PROB);
  real VAL,PROB;
  short integer ID,TYPE;
.....
.....

!*** calculate moments and normalize *****
!*** write result on file REPORTS *****;
procedure CALCMOM;
.....
.....

!*** write report on file BUFREP *****;
procedure WRITEUP;
.....
.....

procedure HISTODUMP;
.....
.....

!*** clear all the recorded data *****;
procedure CLEARALL;
.....
.....

.....
.....

end ****OF CLASS REPORT****;
!*****
*      STATION PROCESSORS DEFINITION      *
!*****;

!**** controller of ground stations time-plan ****;
Process class DESTBEAMPLAN(GSTATE,IDBM);
  short integer GSTATE,IDBM;
.....
.....

```

```

Process class SEND (STAT);
  ref (STATION) STAT;
  .....
  .....

Process class RECEIVE (STAT);
  ref (STATION) STAT;
  .....
  .....

Process class STATION (ID,IDL,NSTAT,RGATE,NGATE,
                      RNOISE,SNOISE,MAXBUFF,LDATA,LPACK,CRATE);
  short integer ID,MAXBUFF,IDL,NSTAT;
  real RGATE,NGATE,RNOISE,SNOISE,LDATA,LPACK,CRATE;
begin
  .....
  .....

  real procedure UNOISEN;
  .....

  real procedure DNOISEN;
  .....

  procedure SWAPIN (BEAM);
    integer BEAM;
  .....

  procedure SWAPOUT (BEAM);
    integer BEAM;
  .....

  .....
  .....

end***OF CLASS STATION***;
!*****
* RECORDS DATA FOR TIME SERIES ANALYSIS *
*****;

Process class TIMESERIES (ID,TYPE,DTIME,FILENAM);
  value FILENAM;
  text FILENAM;
  real DTIME;
  short integer ID,TYPE;

  .....
  .....

end***** OF CLASS TIMESERIES *****;
!*****
* MAIN PROGRAM FOR THIS APPLICATION *
*****;

```

```

!*** reads data for time series recording (if any) *****;
procedure READTSERIES;
begin
  while not DATA.Endfile do
  begin
    .....
  end;
end*** OF PROCEDURE READTSERIES *****;

! ** main program body *****;

! ** initialize on-board switch-matrix *****;
.....
! ** initialize on-board transponders *****;
.....
! ** initialize timeplanners of ground-stations *****;
.....
end;
!*** reads data for station from data file *****;
! ** and generate ground stations *****;
.....
!*** reads data for traffic generators *****;
! ** and generate traffic sources *****;
.....
end;
.....
!*** calculate and write periodical reports ***;
!*** deactivate processors ***;
.....
end*** OF SIMULATION BLOCK *****;
.....
end ***** OF THE PROGRAM *****;

```

2.0 INPUT FORMAT

Here are listed the procedures through which input data for simulation run are given. Comments are self explanatory. All data are supplied in free format.

Global data are read by the following procedure:

```

procedure READGLOBALS;
begin
  *** simulation data file - record length 80 characters ****;
  DATA:-new Infile("DATA");
  DATA.Open(Blanks(80));

  *** reference bit rate (bps) ****;
  REFSPEED:=DATA.Inreal;
  *** propagation delay (sec) ****
  *** average - equal for all stations ****;
  PRDELAYS:=DATA.Inreal;
  *** virtual slot length (bit) ****;
  LVSLOT:=DATA.Inreal;
  *** number of beams ****;
  BEAMNUM:=DATA.Inint;
  *** max. number of stations per beam ****;
  STAPBEAM:=DATA.Inint;
  *** number of up-link transponders ***;
  UPTRSP:=DATA.Inint;
  *** simulation time (sec) ****;
  SIMTIMES:=DATA.Inreal;
  *** load reduction factor ****;
  LOADFAC:=DATA.Inreal;
  *** rate of reports production (prdelay) **;
  REPRATE:=DATA.Inint;
  *** new line ****;
  DATA.Inimage;
  *** number of switch states ****;
  SSTATE:=DATA.Inint;
  *** number of equivalent states ****;
  SEQSTATE:=DATA.Inint;
  *** frame scaling factor ****;
  LFRAMEF:=DATA.Inreal;
  *** data/traffic ratio ****;
  DTRATIO:=DATA.Inreal;
  *** data/traffic ratio for time plan ****;
  DTPLAN:=DATA.Inreal;
  Outtext("Global data read");
  Outimage;
end*** OF PROCEDURE READGLOBALS ****;

```

Data for the on-board switch controller are read by the following procedure:

```

!*** reads data for switch matrix *****;
procedure READSWCHPLAN;
begin
  short integer I,J;

  if DATA.Endfile then
  begin
    Outtext("error in input data "); Outimage;
    ERROR("user err. 01");
  end;
  for I:=0 step 1 until SSTATE-1 do
  begin
    !*** every beam block separated by a line ***;
    DATA.Inimage;
    for J:=1 step 1 until UPTRSP do
      !*** a set of data per line *****;
      !*** down transponder read *****;
      ONBOARDSW.TRANSINTERCONN(I,J):=DATA.Inint;
    !** delay for the next state *****;
    ONBOARDSW.TRANSTIMEPLAN(I):=DATA.Inreal*LSSTATE;
  end;
  Outtext("On-board matrix data read");
  Outimage;
end *** OF PROCEDURE READSWCHPLAN *****;

```

Data for ground station time plan controller are read by the following procedure:

```

!*** reads data for beam plan *****;
procedure READBEAMPLAN(IDP);
  short integer IDP;
begin
  short integer I,J,K;

  !*** new line *****;
  DATA.Inimage;
  if DATA.Endfile then
  begin
    Outtext("error in input data "); Outimage;
    ERROR("user err. 01");
  end;
  !** number of states for ground-stations planner **:
  K:=DATA.Inint-1;
  UPBMPLANNER(IDP):-new DESTBEAMPLANNER(K, IDP);
  for I:=0 step 1 until K do
  begin
    !*** every beam block separated by a line ***;
    DATA.Inimage;
    for J:=1,2 do
      !*** spot to be connected *****;
      UPBMPLANNER(IDP).DSTBMINTERCONN(I,J):=DATA.Inint;
    !*** window for transmission ****;
    UPBMPLANNER(IDP).DSTBMTIMEPLAN(I,1):=DATA.Inreal*
      DTPLAN*LSSTATE;
  end;
end;

```

```

    *** delay for the next state ***;
    UPBMPANNER (IDP) .DSTBMTIMEPLAN (I,2) :=DATA .Inreal*
                                                LSSTATE;
end;
Outtext("Ground stations time plan read");
Outimage;
end*** OF PROCEDURE READBEAMPLAN *****;

```

For every ground station the following data are to be supplied:

```

    *** reads data for stations *****;
    procedure READSTATDATA (I) ;
        short integer I;
    begin
        real NGATE, RGATE, ERBITEC, RNOISE, CRATE;
        integer LRFRAME, STATRES, MAXRES, UPDSLOT,
                LRESVSLOT, LRESVPACK, LDATASLOT, LDATAPACK, MAXBUFF;
        boolean RTDMAFLAG, SALOHAFLAG, ERRDATAFLAG;

        *** new line *****;
        DATA .Inimage;
        if DATA.Endfile then
            begin
                Outtext("error in input data "); Outimage;
                ERROR ("user err. 02");
            end;
        *** channel speed rate (bps) *****;
        CRATE:=DATA.Inreal;
        !*****
            rtdmaflag  salohaflag
                1      0      R-TDMA/F-TDMA
                1      1      R-TDMA/S-ALOHA
                0      0      F-TDMA
                0      1      S-ALOHA
            *****;
        RTDMAFLAG:=1 eq DATA.Inint;
        SALOHAFLAG:=1 eq DATA.Inint;
        *** max. number of data slots per frame ***;
        LRFRAME:=DATA.Inint;
        *** reservation slots per routing frame *****;
        STATRES:=DATA.Inint;
        *** reservation slot length (slot) *****;
        LRESVSLOT:=DATA.Inreal;
        *** reservation packet length (bit) *****;
        LRESVPACK:=1+Entier(LRESVSLOT*CRATE*LVSLOT);
        *** data slot length (virtual slot) *****;
        LDATASLOT:=DATA.Inreal;
        ERRDATAFLAG:=ERRDATAFLAG or (LDATASLOT gt LRFRAMEF);
        ! or (remainder(lframef,ldataslot) lt fuzz);
        *** data packet length (bit) *****;
        LDATAPACK:=1+Entier(LDATASLOT*CRATE*LVSLOT);
        *** rgate and ngate values for S-ALOHA *****;
        RGATE:=DATA.Inreal; NGATE:=DATA.Inreal;
        *** up-link bit error rate *****;
    end;

```

```

ERRBITC:=DATA.Inreal;
!*** receiver error bit rate *****;
RNOISE:=DATA.Inreal;
!*** buffer pool allocation *****;
MAXBUFF:=DATA.Inint;
!*** number of traffic generators *****;
K1:=DATA.Inint;
.....
if ERRDATAFLAG then
begin
  Outtext("check compatibility of switch data vs packet length");
  Outimage;
end;
end*** OF PROCEDURE READSTATDATA *****;

```

For each traffic generator the following data are supplied:

```

!*** reads data for traffic generators *****;
procedure READGENTRAF(I);
  short integer I;
begin
  boolean array DESTINA(0:TNSTAT,0:TNSTAT);
  real array DISTRIB(0:TNSTAT);
  real DISTFACT;
  short integer K,J,J1;

  !*** new line *****;
  DATA.Inimage;
  !*** input packets distribution *****
      0 - erlang distribution of interarrival time
      1 - fixed rate *****;
  TYPE:=DATA.Inint;
  !*** distribution factor *****;
  DISTFACT:=DATA.Inreal;
  !*** input packets rate (bit/sec) ****;
  RATE1:=LOADFAC*DTRATIO*DISTFACT*DATA.Inreal;
  !*** interarrival time distribution S.D. ****;
  SD1:=LOADFAC*DTRATIO*DISTFACT*DATA.Inreal;
  !*** number of packets per generation *;
  NPACK:=DATA.Inreal;
  MPACKFLAG:=NPACK gt 1;
  !*** distribution of the number of packets **
      per generation:
      0 - exp. distribution
      1 - fixed rate *****;
  MPACK:=DATA.Inint;
  !*** new line *****;
  DATA.Inimage;
  !*** number of destination sets *****;
  J1:=DATA.Inint;
  for J:=1 step 1 until J1 do
  begin
    !*** new line *****;
    DATA.Inimage;

```

```

    *** set destination stations ***;
    DISTRIB(J-1):=DATA.Inreal;
    for K:=DATA.Inint while (K gt 0) do
        DESTINA(J-1,K-1):=true;
    end;
    activate new GENMSG(UPSTATION(I),RATE1,SD1,TYPE,
        NPACK,MPACK,MPACKFLAG,DESTINA,DISTRIB);
    Outtext(" data for traffic generator read");
    Outimage;
end*** OF PROCEDURE READGENTRAF *****;

```

At each ground station we suppose a process which spill traffic at a prefixed rate. The data for this process are supplied as follows:

```

*** reads data for sink *****;
procedure READSINKTRAF;
begin
    *** new line *****;
    DATA.Inimage;
    *** output packets rate (bit/sec) ***;
    RATE2:=DATA.Inreal;
    Outtext(" data for traffic sink read");
    Outimage;
end*** OF PROCEDURE READSINKTRAF *****;

```

### 3.0 OUTPUT FORMAT

We have three files on output.

The first - usually spooled on the printer - contains the log of the different processes initialization and activation.

The 2nd contains data for the histogram of the amount of storage used in each ground station.

The 3rd contains statistics. In each ground station we record:

- total input packet rate (effective and refused) from traffic sources
- total output packet rate, vs. terrestrial network
- traffic of outgoing packets, vs. satellite system
- throughput for the outgoing packets
- one-way delay
- two-way delay if we have an error recovery protocol
- queues length (new arrival, retransmission, wait for ACK and output queues)

In each link we record traffic, throughput, delay and a statistics of collisions. We record minimum, maximum, expectation and variance for every stochastic variable. In the statistics of queues length we record both conditional and unconditional probabilities.

Data are recorded on files with a record length of 80 characters. They are in free format character representation, which is typical of Simula67 outputs.

Reports are produced at a fixed rate during simulation time. The rate is another parameter that can be supplied at every simulation run.

A shortened form of the outputs is reported in the next chapter.

4.0 SKETCH OF A SIMULATION RUN

We list here input data and results of a simulation run we used to test the program. To have a good test we simulated a simplified but realistic situation of ten transponder and one ground station per transponder. Time plan, traffic values and distributions between the various destinations were obtained from a traffic analysis for the Italian domestic satellite ITALSAT.

Input data for the simulation run are listed below:

```

1.2E+8 0.250 01024 10 001 010 004.70 9.8E-1 04  ** GLOBALS DATA **
042 62 1.0000E+0 5.0E-2 5.0E-2
 8  2  7  5  4  3  6  1  9 10  1          ** 0 **      **ON-BOARD MATRIX
 8  2  7  5  4  3  6  1 10  9  1          ** 1 **
 8  2  7  5  4  3  6  1  0  0  4          ** 5 **
 8  0  9  5  4 10  6  1  0  0  1          ** 6 **
 3  8  9  1  6  5  7  2  4  0  2          ** 8 **
 3  8  0  1  6  5  7  2  4  0  1          ** 9 **
.....
.....
0  5  4  3  2  6  1 10  0  8  1          ** 61 **
** GROUND STATIONS PLAN ***
10
1  8  7  7
1  3  5  5
1  4  6  7
1  2  5  5
1  1  6  7
1 10  1  1
1  9  2  2
1  5  7 15
1  6  6  6
1  7  6  7
10
2  2  6  7
2  8  6  6
2 10  2  2
2  6  6  6
2  7  6  6
2  3  5  7
2  4  6 10
2  9  2  2
2  1  5 10
2  5  6  6
10
.....
.....
10
10 10  1  1
10  9  1  9
10  4  2  2
10  3  1  5

```

```

10 5 1 5
10 1 1 4
10 2 2 4
10 6 1 22
10 7 2 8
10 8 2 2
1.200E+8 0 0 0100 00 01 1.00 0.0E-0 0.0E-0 0.00 0.00 05000 1
0 1.00E-0 9.87096E+7 9.87906E+7 1 1
10
0.1176470588 1 0
0.09803921569 2 0
0.09803921569 3 0
0.1176470588 4 0
0.137254902 5 0
0.1176470588 6 0
0.1176470588 7 0
0.137254902 8 0
0.03921568627 9 0
0.01960784314 10 0
1.200E+8 1.200E+8 ***END STATION 1***
1.200E+8 0 0 0100 00 01 1.00 0.0E-0 0.0E-0 0.00 0.00 05000 1
0 1.00E-0 9.6774E+7 9.6774E+7 1 1
10
0.1 1 0
0.12 2 0
.....
.....
1.200E+8 0 0 0100 00 01 1.00 0.0E-0 0.0E-0 0.00 0.00 05000 1
0 1.00E-0 2.709E+7 2.709E+7 1 1
10
0.07142857143 1 0
0.1428571429 2 0
0.07142857143 3 0
0.1428571429 4 0
0.07142857143 5 0
0.07142857143 6 0
0.1428571429 7 0
0.1428571429 8 0
0.07142857143 9 0
0.07142857143 10 0
1.200E+8 1.200E+8 ***END STATION 10**

```

The log of initialization and activation of the various processes has the following form:

```

Global data read
Propagation delay is 2.92960E+04 time units
Switching equivalent state is 2.00000E+01 time units
Frame or multiframe length is 1.24000E+03 time units,
and 1.05813E-02 sec
On-board matrix data read
On-board matrix controller initialized
Up-link transponder n. 0 initialized
.....

```

```

Up-link transponder n. 10 initialized
Down-link transponder n. 0 initialized
.....
Down-link transponder n. 10 initialized
Ground stations time plan read
Ground station time planner n. 1 initialized
.....
Ground stations time plan read
Ground station time planner n. 10 initialized
Data for station 1 read
Station n. 1 initialized
  data for traffic generator read
  data for traffic sink read
.....
.....
Data for station 10 read
Station n. 10 initialized
  data for traffic generator read
  data for traffic sink read
Station n. 1 activated
  receiver n. 1 initialized and activated
.....
.....
Station n. 10 activated
  receiver n. 10 initialized and activated
Ground station time planner n. 1 activated
.....
Ground station time planner n. 10 activated
  sender n. 1 initialized and activated
  .....
  sender n. 10 initialized and activated
On-board matrix controller activated
Down-link transponder n. 9 activated
Down-link transponder n. 1 activated
Down-link transponder n. 5 activated
Down-link transponder n. 3 activated
Down-link transponder n. 8 activated
Down-link transponder n. 7 activated
Down-link transponder n. 6 activated
Down-link transponder n. 4 activated
Down-link transponder n. 2 activated
Down-link transponder n. 10 activated
Report written

```

Histogram of memory occupancy in the ground stations has the following form:

```

0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000

```

0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000  
0.000 1.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000

The periodic output during simulation run has the following form:

5.41913E+05	1.92000E+09	4.68768E+06	8.03084E-03
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
6.66666E+05	6.00000E+06	4.84295E+06	2.78669E-04
4.01338E+05	1.20000E+08	4.75349E+06	3.01525E-02
4.01338E+05	1.20000E+08	4.80436E+06	3.05673E-02
1.00007E+00	1.52991E+00	1.14631E+00	1.19905E-02
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
1.00000E+00	7.60000E+01	1.70459E+01	1.23759E+02
1.00000E+00	7.60000E+01	1.70457E+01	1.23760E+02
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	7.00000E+00	1.92018E+00	2.47103E+00
0.00000E+00	7.00000E+00	1.92008E+00	2.47109E+00

.....  
.....  
.....

1.83014E+05	1.92000E+09	1.30536E+06	6.90212E-04
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
1.66667E+05	6.00000E+06	1.30449E+06	2.01286E-04
1.87500E+05	1.20000E+08	1.28685E+06	2.98156E-03
1.87500E+05	1.20000E+08	1.28583E+06	2.92510E-03
1.00080E+00	1.75532E+00	1.15873E+00	1.82910E-02
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
1.00000E+00	1.80000E+01	5.31728E+00	9.00788E+00
1.00000E+00	1.80000E+01	5.31709E+00	9.00854E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	2.00000E+00	1.52904E+00	2.49158E-01
0.00000E+00	2.00000E+00	1.52844E+00	2.49978E-01

0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00



0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	1.00000E+00	3.48248E-02	2.67680E-02
0.00000E+00	1.55044E+00	1.11362E+00	1.00867E-02
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

.....  
.....  
.....

0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	1.00000E+00	9.38646E-03	2.69138E-03
0.00000E+00	2.00719E+00	1.14754E+00	2.16845E-02
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

REFERENCES

R.1 R. Beltrame, "Simulazione di schemi di accesso a divisione di tempo su canale comune via satellite. Algoritmi di SATNET", Rapporto interno CNUCE C80-10, Dicembre 1980.