A New Computer Program for Data Communications Network Simulation-SONET

Tsutomu Shiino*

1. INTRODUCTION

General-purpose simulators such as GPSS and SIMSCRIPT have been widely used for the traffic simulation of network systems such as data communication networks, computer systems, and traffic networks. Since all of them are language oriented programs, users have to build a system model by using a specified language. For this reason, their application requires considerably wide knowledge and experiences and much labour is imposed on the part of users. Another disadvantage is its inevitable complexity in the expression of network system which greatly decreases the simulation efficiency and requires long debugging time.

SONET described here is a data-oriented simulator designed to suit the network system simulation. Unlike that of GPSS and SIMSCRIPT, its input data is greatly simplified and it also has less number of steps. Also, SONET has a unique data structure suitable for the handling of network topology and has effective functions for network system simulation. All of these results in significant improvement in storage requirements and execution time.

The input data is divided into the following categories, each of which allows independent expression. These are input data for expressing network topology, input data giving attributes of the transaction, input data for giving processing contents, and input data for simulation control. Since it is provided with a routing function capable of selecting an optimum route for a given network condition, various kinds of routing algorithm can also be simulated.

Not only that it is effective to the traffic simulation of data communication systems, SONET can also be easily applied to the simulations of computer systems, traffic networks, and production plants, etc.

This program is expressed in FORTRAN by about 5,000 statements.

This paper first appeared in Japanese in Joho−Shori (Journal of the Information Processing Society of Japan), Vol. 14, No. 7 (1973), pp. 482～489.

* Software Systems Division, Oki Electric Industry Co., Ltd., Tokyo, Japan.

2. INPUT LANGUAGE AND SIMULATION PERFORMANCE

Input language of SONET consists of the following 9 types:

(1) topology of the model,
(2) attribute of transaction,
(3) transfer sequence,
(4) distribution function,
(5) network condition,
(6) sub-system,
(7) routing algorithm,
(8) simulation control,
(9) format and content of output.

2.1 EXPRESSION OF STRUCTURE OF MODEL

Topology of model is expressed by branches and nodes. There are two kinds of branches: L branch which is applied to transfer transactions only, and P branch which is applied to process transactions and vary the length of them. There are also two kinds of nodes: M node which has zero or only one branch starting from it, and R node which has two or more branches starting from it.

For example, L branch ℓ_1 is connected from M node m_1 to R node r_1. If the speed of transfer of the branch is v and the transfer is made in the unit of b packets, this branch is expressed as follows:

$$L\ell_1,\ Mm_1 - Rr_1, S = v, B = b.$$

A node is expressed as follows:

$$Mm_1, A = a_1$$

A denotes the capacity of the node as a storage. Any number of branches can be connected between two nodes, and some branches can start from a node and return to the same node. Most system structures can be equally expressed by use of the above-mentioned language.

2.2 EXPRESSION OF TRANSACTION

In SONET system, packets and blocks are considered as transactions. A transaction is first generated as a block which is split into packets at the beginning of transference. Attribute is given for a block as follows:

$$Dd_1,\ \text{'NAME'}, F=f_1,\ L=\ell_1,\ ST=t_1,\ N=n_1,\ VN=t_2,\ PATH=p$$

The symbol D means the attribute card, d_1 is the block number and NAME is the name given to the block. F means the generating interval and L indicates the block length. ST stands for the time when the block is first generated, and N limits the number of generation and VN indicates the life time of the transaction. PATH gives the route through which the transaction is transferred, and is expressed in three types of expression. In the first type of expression, transaction is transferred from node to node by selecting a route according to the values of the routing table. In the second type of expression, routing is made only between the several routes specified. In the third type of expression, transfer is made only through the specified single route.

2.3 RANDOM NUMBERS

The network parameters such as S, F and L can be changed by use of random numbers of specified distribution as follows:

(i) $X = x_1\ (x_2,\ x_3)$

(ii) $X = Fn_f(x_1)$

In expression (i), the random number is of uniform distribution within a range of $x_1 - x_2 \leqslant X \leqslant x_1 + x_2$ and takes on values of $x_1 \pm nx_3$ ($n = 1, 2, \ldots,$ and $x_2 \geqslant nx_3$). Expression (ii) indicates that the random number is given with distribution function Fn_f.

2.4 FUNCTION

Function gives distribution of random numbers. Independent variables are deviations from the central value and the probabilities having these values are dependent variables. There are two kinds of function; one which makes linear interpolation between given points and the other which only takes the values of given points. These functions can be referred to by function number and central value.

2.5 CONDITION

Generation and disappearance of a transaction, update of routing table, output of results and termination of simulation can be effected at the time when a condition is satisfied. The condition that can be specified for such purposes are listed below.

The number of the nodes which a message passed has amounted to "n".

The time "t" has elapsed since generation of a transaction.

Waiting time has reached "t".

A message passed the same node "n" times.

No transaction arrived at node "m" last "t" time.

Transaction "d" has arrived at node "m".

Transaction "d" has generated.

Transaction "d" has disappeared.

Constant time interval "t" has passed.

These conditions can be described either in parallel or series.

They can be specified by CONDITION n_c. Where Nc is a condition number.

This condition is to be refered to as below.

VN = CONDITION 1

This example means that a transaction disappears at the time when condition 1 is satisfied.

2.6 SUB-SYSTEM

When many units of the same type or several parts of the same structure exist in a system model, they are defined as sub-systems. In the main system these are expressed only with sub-system numbers for the purpose of simplifying the input data. Connection of sub-systems to the main system is made at nodes by use of node numbers as arguments.

2.7 SELECTION OF ROUTE

Selection of route is made to the branch which takes the minimum value in the routing table. The routing table is given by the following expression.

$$T(n_n, n_e, n_b) = t$$

Symbol n_n is the node number where the transaction stays at that moment, n_e is the number of nodes which will be the address of the transaction, and n_b is the number of the branch to be selected. The values of the routing table are given by the initial values and update formulas, and these can be expressed in the same manner as the arithmetic expression of FORTRAN. Using these expressions, various routing algorithm can be simulated.

2.8 SIMULATION CONTROL

The following are statements for control of simulation:

TIME,STEP = t_d, TIME,START = t_s, TIME,END = t_e

RESET, CLEAR, MODIFY, MODEL, EXECUTE, END

The first line indicates three kinds of time statements which give time interval, starting time and ending time of simulation respectively. RESET erases all the statistical quantity up to the moment, and CLEAR erases all the results except the random number. MODIFY changes the parameter values of the model. MODEL is the statement which indicates the beginning expression of the system model. EXECUTE is the command which indicates the start of simulation and END is the statement which indicates the completion of input data. By using the statements mentioned above separately or in combination in a proper way, various simulations can be made for many problems. For example, problems in breakdown of units or transmission lines, and change of traffic by time etc.

2.9 COLLECTION OF STATISTICAL QUANTITY AND SPECIFICATION OF OUTPUT

Collection of statistical quantity and the output of results are made by TABLE statements. TABLE specifies the content of the output and its format. The outputs are presented in such forms as means value, maximum value, minimum value, standard deviation, and frequency distribution table. Statistical quantities are regarded such as rates of utilization, utilization times of branches and nodes, queue, arriving frequency, arriving interval, transit time of a transaction, and number of nodes which a transaction passed etc.

2.10 REQUIREMENT OF MAIN MEMORY OF COMPUTER AND LIMIT OF INPUT MODEL

In SONET system, according to the capacity of memory of the computer in use and the requirements for simulation speed, the best system can be composed by overlaying the modules or propoer selection of capacity of the arrays. The following is an example of the limit of input model in the case of main memory capacity of 128 kilo bytes:

Transaction (packet)	1,000	Function	20
Kinds of block	100	Condition	20
Branch	200	Sub-system	10
Node	100	Routing table	200

3. EXAMPLE OF APPLICATION

Figure 1 shows an example of input data of SONET. The model is a system in which the computer is connected, through a terminal control unit, with four terminal units with full duplex lines of 200 bit/s.

4. CONCLUSION

Recently there is increasing demand for computer simulation systems for estimation of system performances and for design of network systems.

SONET was developed as a system with which simulation can be made simplier and correctly in a shorter time. As network systems become more complicated they become more difficult to express the model with the conventional general purpose simulation languages; or the expression has to be more complicated in many cases.

With SONET system, users can express various network model very simply with less erros. The number of input steps is very small. This number is reduced to several tens of a percent or less than ten percent of the case with GPSS. It requires a very short execution time, and in the case of the same model applied with IBM360/195 the execution time is reduced by about one-tenth to one quarter as compared to the simulation by GPSS/360.

Various route selection techniques have been studied from the point of view of effective network utilization. SONET has suitable data structure for route selection, so it realizes efficient simulation in this field.

```
MODEL
L1,M1-R1,S=200
L2,R1-M1,S=200
L3,M2-R2,S=200
L4,R2-M2,S=200
L5,M3-R3,S=200
L6,R3-M3,S=200
L7,M4-R4,S=200
L8,R4-M4,S=200
L9,R1-M5
L10,R2-M5
L11,R3-M5
L12,R4-M5
L13,R5-R1
L14,R5-R2
L15,R5-R3
L16,R5-R4
L17,M5-R6,S=200
L18,R6-R5,S=200
P1,R6-R6,G=11
P2,R6-R6,G=13.5
H1,L1-L2
H2,L3-L4
H3,L5-L6
H4,L7-L8
M1,A=1000
M2,A=1000
M3,A=1000
M4,A=1000
R6,A=1000
D1,'DATA-01',F=600,L=480,ST=400,PATH=M1,R6,L1,L9,L17
D2,'DATA-02',F=840,L=560,PATH=M1,R6,L1,L9,L17
D3,'DATA-03',F=881,L=450,ST=300,PATH=M2,R6,L3,L10,L17
D4,'DATA-04',F=941,L=380,PATH=M2,R6,L3,L10,L17
D5,'DATA-05',F=209,L=410,PATH=M3,R6,L5,L11,L17
D6,'DATA-06',F=1655,L=200,ST=900,PATH=M3,R6,L5,L11,L17
D7,'DATA-07',F=769,L=400,PATH=M4,R6,L7,L12,L17
D8,'DATA-08',F=150,L=340,ST=50,PATH=M4,R6,L7,L12,L17
D9,'DATA-09',F=6825,L=1240,PATH=R6,M1,L18,L13,L2
D10,'DATA-10',F=606,L=720,ST=200,PATH=R6,M1,L18,L13,L2
D11,'DATA-11',F=129,L=6800,PATH=R6,M2,L18,L14,L4
D12,'DATA-12',F=1655,L=2200,ST=800,PATH=R6,M3,L18,L15,L6
D13,'DATA-13',F=941,L=3960,PATH=R6,M4,L18,L16,L8
D14,'DATA-14',F=1071,L=280,PATH=M3,M3,L5,L11,L17,P1,L18,L15,L6
D15,'DATA-15',F=170,L=220,ST=20,PATH=M4,M4,L7,L12,L17,P2,L18,L16,L8
PACKET,L=1020
PRIORITY,(7,15),(5,14),3,1,8,6,4,2,13,12,9,10,11
PRIORITY,H1=L1,H2=L3,H3=L5,H4=L7
INTERRUPTION,REST=2
INTERRUPTION,L2=EOB,L4=EOB,L6=EOB,L8=EOB
CONDITION1=PR(1800)
PRINT=CONDITION1
TABLE,U,L1,L3,L5,L7,L17,L18,R1,R2,R3,R4,R5,M5
TABLE,Q,L1=1,2,L3=3,4,L5=5,6,14,L7=7,8,15
TABLE,4,L18,D10,0,2,40
TIME,END=7800
EXECUTE
CLEAR
MODIFY
D1,F=1200,L=640
TIME,END=7800
EXECUTE
END
```

Fig. 1 An example of SONET input data list