SIMULATION OF THE DATA FLOW IN COMMON CONTROL SWITCHING SYSTEMS

Paul Hanglberger
Siemens A. G.
Munich, Federal Republic of Germany

ABSTRACT

In modern switching systems the buildup of connections is directly controlled by a computer. In the following a flexible modular simulation program which can be used for structural investigations of modern common control systems is explained.

1. INTRODUCTION

In modern switching systems the buildup of connections is directly controlled by a computer. A common-control switching system consists essentially of a central processor unit (computer), several peripheral control units and peripheral equipment (relay sets, switching matrix controllers, identifiers etc.) (see Fig. 1).

The peripheral control units perform the data exchange between the peripheral equipment and the central processor unit. The central processor unit processes the informations from the peripheral equipment with the aid of the switching programs and generates instructions for switching the switching network (via switching matrix controllers) or relay set instructions which are supplied to the periphery. The buildup and breaking-down of a connection consists of a defined sequence of informations and instructions. As the switching systems dealt with here have a totally different concept and layout to earlier systems, it is of fundamental importance to obtain data about their traffic theory performance. For the design of such common-control switching systems it is important to harmonize the capacities of the central processor unit and the periphery. Besides it is important that the system is not self-blocking under overload traffic conditions. A mathematical model which describes such a system can only be theoretically analysed and calculated by using considerable simplifications. For this reason an extensive system simulation on a digital computer is the only way of obtaining exact data about the traffic theory behaviour of such systems. In the following a flexible modular simulation program which can be used for structural investigations of modern common control systems is explained.

2. DESCRIPTION OF THE SIMULATION PROGRAM

When a program for a simulation of complex systems is prepared, one of the most important points that have to be clarified is what has to be taken into account and what may be negligible of the system, if as realistic results as possible are the goal. After the size of the program had been estimated, the problem-oriented programming language FORTRAN
was chosen. At the time the program was being prepared on the available digital computer, FORTRAN was the only higher-level programming language which was suitable for simulation and which permitted the programs to be segmented in a relatively simple way. The simulation program consists of about 8000 FORTRAN statements and is divided into three segments:

1. Input of the simulation data  
2. Simulation of the data flow in the switching systems  
3. Output of the simulation results.

2.1. Input of the Simulation Data  

The input data for the simulation program falls into two parts:  

a) General system parameters  
b) Characteristics of the traffic offered.

2.1.1. General System Parameters  

The switching system is simulated in the core memory of the digital computer by the simulation program as a model which consists of lists. In essence, the system is characterized by the following lists.

![Diagram of lists](image)

The general system parameters include data about the average execution time of the instructions of the central processor unit, the number of the peripheral control units, the number and length of the individual lists. In FORTRAN a list is characterized by declaring an array name A(I), where I is the number of computer words reserved for the given array. As FORTRAN does not provide the possibility of variable array declarations, no separate array name was chosen for each image of a list in the interest of a flexible structure of the simulation program.

If a common array name A is agreed upon for two lists, the first position on the second list is defined by the address A(M+1). If a maximum of M+N positions for the input/output array A in the FORTRAN program, the quantities M and N can be varied in the input data. This method was also applied to more than two lists and to two-dimensional arrays.

The general system parameters also include data about the number of instructions of the central processor unit for the input/output program which handles the data exchanges between the central processor unit and the peripheral control units. For convenient use of the simulation program, the system parameter input program was written so that only the number and the sequence of the data are laid down in the input card.

2.1.2. Characteristics of the Traffic Load

The simulation program simulated the processes in the correct-time sequence. The inter-arrival times of the calls (attempts to build up a connection), i.e. the spacing of the first information of the calls offered to the system is assumed to be a Poisson input. With the aid of an exponentially distributed random number Z and the mean inter-arrival time A of the calls, the instant of the arrival of the n-th call Xn is formed using the following formula:

\[ X_n = X_{n-1} + Z \times A \]  

where \( X_{n-1} \) is the instant of the arrival of the \((n-1)\)th call. The simulation program simulated all of the connection buildup and breakdown processes, i.e. it simulated the defined sequence of informations and instructions, which pass through the system between the periphery and the central processor unit. Informations are the data flow from the periphery to the central processor unit. Instructions are the data flow from the central processor unit to the periphery.

Example for the begin of the data flow of a connection buildup:

![Connection buildup diagram](image)

The buildup and breakdown of a connection consists of about 40 to 50 informations and instructions. One of the major difficulties in the preparation of the simulation program was to find a tabular form (identification table) suitable for the simulation of the call processing functions. In this table the informations and instructions belonging to one call are consecutively numbered and it essentially contains the following specifications about the informations and instructions:

For each information:

a) whether after a specified period of time a further information arrives in the periphery, belonging to the same call. If it is so the number of the new information and the time when it arrives is given.

b) whether there is a critical waiting time for the information so that they are not allowed to wait longer than a specified
period of time in the peripheral equipment. If it is so the critical waiting time is given.

c) the number of program instructions which must be handled in the central processor unit.

d) the number of the switching matrix and/or relay set instructions which as the result of the information processing are fed to the output lists by the central processor unit.

e) the sum and the numbers of the informations which must have arrived in the central processor unit, to process the informations.

For each instruction

a) whether an information is generated a specified time after the instruction has been issued to the peripheral equipment. If so, the number of the new information and the time when it is generated, is given. With the aid of the simulation program any desired combination with up to 8 different types of calls can be tested.

2.2. Simulation of the Data Flow in the Switching System

In the main part of the simulation program the data exchange between the periphery and the central processor unit via the peripheral control units as well as the processing of the informations in the central processor unit are simulated. Central control switching systems in which the central processor unit starts the I/O program at fixed intervals, can be examined with the simulation program. The following is an example of one of the I/O strategies which can be examined with the simulation program:

Each peripheral device served by a peripheral control unit is simulated with the aid of a list into which all informations reaching the system are written. An entry into this list consists of the number of the call buildup, the number of the informations as well as the instant of the arrival of the information.

Example 3

| 120 | 3 | 1500.30 |

120th call buildup
3rd information
1500.30 instant of the arrival of the information.

If a peripheral control unit is not busy, it can accept an information from the periphery independently of the central processor unit and will be busy during the transfer time \( T_A \). At the beginning of each I/O intervals, i.e. prior to the start of the I/O programs, the status of all peripheral control units is tested in the simulation program. If the list of the peripheral control unit is not empty an information is transferred and the peripheral control unit is busy up to the instant

\[ T = \text{Max}(T_A/T_B) + T_U \]

where

\( T_A \) is the instant of the arrival of the information in the peripheral equipment

\( T_B \) is the instant at which the peripheral control unit was last busy.

This is done for all peripheral control units. Before an information is transferred a test is made with the aid of the identification table to see whether a critical waiting time has been exceeded. If it is so, the call is lost.

Also, a test is made on each information to see whether at a later instant \( T_0 \) another information of this call arrives in the system. If so, the new information is written into an auxiliary list and is transferred to the peripheral device list at instant \( T_0 \).

The I/O programs take over the informations which are completely in the peripheral control units into an common input list of the central processor. Subsequently, if present, an instruction of an output list is given to the peripheral control unit. The peripheral unit is busy for the time \( T_0 \), in which the instruction is given to an peripheral device.

After the I/O program started of fixed intervals has ended, the information processing begins. The informations of the input list are dealt with by the first in - first out method. Only the dynamic number of program instructions required for the processing of the information is of importance for the simulation. The central control unit is busy for the time of the dynamic program run.

The processing of information, however, depends on the call progress status. For this reason, a call status table is used to keep a record in the simulation program about the status of all calls in the system everyday. If an information of the input list arrives in the central processor unit, the arrival is stored in the call status store. A compare between the identification table and the call status store decides whether the information can processed immediately. If the compare is negative, because not all informations are available, the information is written in an auxiliary list (time list) and is processed when all informations are available. During the processing of an information, instructions can be written into the output lists. There are several output lists (for the various types of peripheral devices) which are handled with various priorities by the I/O program.

2.3. Output of the Simulation Results

The results of the simulation can be divided into four groups:

a) Loss calls

b) waiting time and response time distributions

c) loading of the central processor unit and the peripheral control units.

d) data about the input/output lists.

2.3.1. Lost Calls

Lost calls, i.e. unsuccessful attempts to build up a connection, occur as a result of critical waiting times being exceeded when specific informations (for instance digits) are fetched from the peripheral devices for transfer to the peripheral control units. The percentage of these lost calls is specified for each simulated type of offered traffic.

2.3.2. Waiting Time and Response Time Distributions

Waiting time occurs when informations are fetched from the peripheral devices and transferred to the peripheral control units, prior to the processing of the informations in the input list by the central processor unit as well as prior to the issue of the output list instructions to the control units. The distribution of these waiting times is stated. The simulation program offers the possibility of examining various input/output strategies for a specific system configuration so as to
minimize the waiting times (and hence the number of lost calls). In addition, so-called response times which provide data about the quality of service of a system can be measured with the simulation program. The response time is the time difference between the instant $T_R$ and instant $T_A$, where e.g. instant $T_R$ is the instant of arrival of information $A$ in an peripheral device and instant $T_A$ is the instant at which instruction $B$ is issued to a peripheral control unit.

2.3.3. Loading of the Central Processor Unit and the Peripheral Control Units

As the I/O routine of the central processor unit is genuinely simulated, the load of the central processor unit, due to the routine control of the peripheral control units and due to the transfer of informations and the issue of instructions, can be exactly measured. Moreover, the load of the central processor unit by processing the informations is given. If calls are lost because specific informations were not fetched from the periphery and transferred to the peripheral control unit within a specified time, an unproductive load will occur in the central processor unit. The central processor unit instructions performed between the start of the connection buildup and the fetching of these informations were performed in vain. The simulation program determines the percentage of this unproductive load component when overload traffic is offered to the system.

For each peripheral control unit the number of informations which are transferred to the input list, as well as the number of instructions which are transferred from the output lists to the peripheral equipment is given.

2.3.4. Data about the Input/Output Lists

The normal operational sequence of the central processor unit is disturbed if instructions are to be written into the output lists and the space available there is insufficient. The central processor unit then runs in a waiting loop, interrupted by the I/O program until sufficient free positions are available.

The dependence of this time span on the length of the output lists can be examined by varying the list lengths. Moreover the number of the intervalls, during which the control units were blocked to the transfer of informations from the periphery for lack of free positions in the input list is stated.

3. CONCLUSION

The simulation program is a powerful tool for the system design. With the aid of the simulation program various input/output strategies were tested to minimize the waiting times and the percentage of loss calls for a given traffic offer. The following results are important:

a) The choice of an I/O strategy is essentially dependent on the relation of the number of informations to the number of instructions of the calls.

b) The length of the I/O interval and the number of the peripheral control units must be chosen so that the load of the peripheral control units is in the range of the load of the central processor unit.