AN INTRODUCTION TO MODELING WITH INS

Stephen D. Roberts
James Schier
Thom Sulanke

ABSTRACT

INS (the Integrated Network Simulation language) facilitates modeling by freeing the modeler from the mechanics of programming a simulation model. Instead, the modeler characterizes the system as a network containing easily defined nodes and branches. The network is easily translated to INS statements from which INS constructs and executes the simulation model. Output is automatically provided to reflect various perspectives of system behavior. Features of INS allow easy enhancement of statistics and embellishment of the model. INS is easy to learn and potentially applicable to a variety of environments.

MODELING WITH INS

Simulation modeling is a high level task in which the modeler seeks to represent the behavior of a system mathematically and statistically. The simulation model will usually require the construction of a computer program, often utilizing a simulation language. But if the simulation language requires the modeler to code events, manipulate files, generate random deviates, monitor status, collect statistics, and format reports, then modeling can become a secondary task to programming. For the novice or infrequent simulation user, the programming requirement may impede modeling because it can require considerable effort to translate the statement of the simulation problem into a simulation program ready for computer processing.

INS, the Integrated Network Simulation language, was created to provide a high level simulation language that would facilitate modeling. The INS modeler conceptualizes the simulation problem graphically as a network, employing a set of predefined modeling symbols or nodes. The modeling symbols represent such real world processes as queues and activities enabling the modeler to abstract directly from the real system to the INS network. The difficulties in translating the network into computer-readable form are minimized since the INS symbols correspond almost directly to INS statements which INS interprets. INS automatically executes the simulation and provides output. Thus the INS modeler can focus his attention on the important problems of modeling and ignore the mechanics of simulation programming.

Because modeling is a process of successive refinement, there is a need to be able to quickly produce a working "first cut" model and to embellish this model as greater insight is generated. INS facilitates embellishments because modifications are easily incorporated by changing specifications at a node or altering segments of the network. In this way the network becomes a modeling vehicle, useful not only in the creation of the simulation but also in giving structure to the modeling process.

Since INS is concise, easy to document, and visually appealing, INS models are also excellent vehicles for communication between modelers and their clients. Such communication can encourage clients and decision makers to participate more directly in the modeling process and consequently improves the chances of eventual implementation.

To illustrate the fundamentals of modeling with INS, we will consider an example. It will be enriched to demonstrate the ease of modifying INS models. Some discussion will also focus on INS output and its enhancement. Space does not permit a detailed discussion of many of these ideas and you are referred to the INS User's Manual (1).

A TV INSPECTION AND ADJUSTMENT EXAMPLE

Television sets which have just been assembled are routed to an inspection station for final testing. If they fail the inspection, they are sent to an adjustor who, after fixing them, routes them back for inspection again. It has been observed that 15% of the sets fail inspection, regardless of their previous adjustment. Suppose TVs arrive every 12 minutes and it takes 10 minutes to inspect, and 30 minutes to adjust a TV, if adjustment is required. Conceptually, we might depict the problem as the flow diagram in Figure 1.

The flow diagram is valuable because it provides a visual interpretation of the process. In fact, it is natural to define a problem by describing its processes or procedures. The diagram serves to interrelate the processes and remind us of the entire system. But the flow diagram is too ill-defined. Its expression is highly dependent on the individual and his choice of symbols to represent the problem.

INS NETWORK

INS builds on the notion that a graphic model is a natural way to describe a problem. To standardize the format of expression, INS provides the modeler with a set of parameterized symbols that specify elemental processes. The modeler selects appropriate symbols and integrates them into a network by connecting the nodes with branches. For example, an INS network corresponding to the previous problem is contained in Figure 2.

Several features should be noted. The INS network corresponds closely to the conceptual flow diagram. The INS symbols represent elemental but familiar processes. Node numbers are printed in bold letters and are located on the left side of the node symbol. Node 1 is a Source node and depicts the arrival of TVs. Nodes 2 and 4 are Activity nodes representing the delay of TVs in inspection and adjustment respectively. Node 3 is a Sink node and it depicts the departure of TVs. Notice that the data relevant to the problem is contained in the network, an improvement over the flow diagram. For example, TVs arrive with a 12 minute interarrival time at Source 1, adjustment takes 30 minutes at Activity 4, etc. Branches connect the nodes and represent the possible paths for the TVs. Branching in INS is determined by a branching mode specification located where the branch leaves the node. Only Activity 2 requires a branching decision and here branching is performed PRObabilistically. Specifications of the branching mode are listed on the branches, so we see that 85% of the TVs flow to Sink 3 and 15% flow to Activity 4.

The INS network translates immediately to INS statements as shown in Figure 3. It is these statements which INS interprets and checks for errors. By providing a little additional information on the creation of TVs and how to terminate the simulation, the above model can be simulated as it is written and we would obtain statistics which we could use to make further enhancements. Notice that the INS statements are themselves quite readable and the input is sparse. Employing the documentation potential, the INS model can be communicated easily to others.

Without any additional statements, INS provides default output, which in this case would be the average time TVs are in the network, the average time TVs are in each activity, and the average number of TVs in each activity. Also the number of times TVs encounter each node in the network would be printed. INS supplies values for most network specifications by default, printing all of these in an echo and generating a full array of output without special instructions. Our design philosophy is that INS should do as much for a user as possible and should allow the user to easily alter the model and output after examining a working simulation.

TRANSACTIONS

In a more generic sense, the units of traffic which flow through an INS network are called transactions. Transactions can be many and varied, all flowing in the same network. For example, the production system may manufacture different types of TVs.

To permit the distinction between transactions, each transaction has a set of system-defined attributes. Recognized by their first three characters, CREation time, and transaction TYPe are examples. These attributes become available to the modeler when the transaction is created. Furthermore, TYPe can be assigned by the modeler and can be changed during the transaction's flow through the network. For example, if two types of TVs arrive to the inspection station, we might alter the arrival of the TVs so that one Source node generates TVs of type 1 and the other TVs of type 2. Figure 4 contains these changes.

Now Source 1 generates TVs of TYPe = 1 and Source 5 generates TVs of TYPe = 2 both with interarrival times of 24 minutes. We have further shown how to route transactions based on type. At Route 6, the TVs are routed based on their ATtribute Value (ATV) so that transactions which have TYPe equal to 1 are sent to Sink 3 and those which have TYPe equal to 2 are sent to Sink 7. Attributes have many other uses. In addition, the user may assign and manipulate his own attributes and interpret them uniquely.

RESOURCES

We have shown how transactions flow through the INS network, being created, routed and destroyed, and possibly delayed in activities. But often the delay in an activity also represents the use of some resource. For example, TVs being inspected may involve the service of an inspector. In general, a transaction serviced at an activity may require several resources.

Resources in INS can be identified individually (such as resource number 6) and collectively (for example, resources of type 4). The user may also assign the arrival time of the resource, perhaps stochastically. The resources can be utilized at activities to respond to the demands of transactions. For example, at the inspection activity, we can model the use of one inspector as seen in Figure 5. We slightly modify the activity node by placing a resource symbol above it. In the resource symbol, we have specified that resource number 3 will service the inspection activity for 10 minutes. The specification of activity time changed from the activity node to the resource symbol for several reasons.

It is not uncommon for resources to be substitutable at an activity. For example, two inspectors may be available to perform the inspection. One inspector is new and takes 15 minutes while the other is experienced and requires 10 minutes. If both inspectors are available, a choice is needed since they are not identical. Hence INS employs a means of selecting among alternative resources, called the resource selection mode. Several selection modes are available in INS, the simplest being PRIority. For the sake of illustration, we will prefer the inexperienced inspector (resource number 4) to the experienced one (resource number 3). The revisions at the inspection activity are shown in Figure 6. The two resource symbols in one column above the inspection activity depict the two alternative resources that can satisfy the requirement for one resource. The preference for resource 4 (the top symbol) is noted by a priority number (lower numbers mean higher priority) and the PRI in the activity stipulates that resource selection is based on PRIority.

MULTIPLE RESOURCES AT AN ACTIVITY

If more than one resource is required at an activity, additional columns of resource symbols above the activity would be specified. Each column corresponds to the selection of a resource and therefore, a resource selection mode for each resource requirement is needed. If a testing machine (resource 1) is needed at the inspection activity, then the inspection activity would look like Figure 7. It is important to recognize that INS considers multiple resource requirements at activities simultaneously. If all resources are not available, the activity cannot be started. The capturing or holding of resources individually can be specified, but in general, multiple resource requirements will be considered together.

This unique feature of INS is particularly useful in the study of systems where several resources can constrain the processing of transactions. In production systems, typically both men and machines are required, and the simulation will reveal which are more critical. This promotes the realism of the simulation and expands the modeler's ability to represent very complex decision processes, ones which, nevertheless, are frequently encountered.

QUEUING AT ACTIVITIES

Since a resource may not be available every time a TV arrives for inspection, some TVs must wait for resources to become available. Representing queues in INS merely means the addition of queue nodes to the network. The inspection activity with a queue appears in Figure 8.

Queue 8 allows for queuing of TVs before Activity 2 to await an available resource. By default, the queue ranking is FIFO, but this is easily modified by specifying a ranking method (either HIGh or LOW) and a ranking attribute. For example, if transactions of TYPe 1 are to be prefered, Queue 8 would be specified as shown in Figure 9.

Figure 9 illustrates the multi-server single queue with a special queue discipline and server preference. Multi-queue, multi-server activities can be as simply represented in INS by adding additional queues at the activity. For example, we may want to separate the queues for the TVs being inspected for the first time and those which have been adjusted (Figure 10). Other features of the queuing capability in INS permit resources at an activity to serve only selected queues, permit reneging from the queue, and allow transactions to capture resources in the queue. These and other abilities are illustrated in the User's Manual.

USING RANDOM DEVIATES

To change from constant activity times and deterministic arrivals to stochastic variables in INS involves referencing any of eleven built-in distributions. User-specified probability density's can also be employed. Parameter sets are created using these distributions and each parameter set is identified by a number.

To obtain a sample from a distribution, the modeler simply uses the negative parameter set number. The PARAMETER statement defines the distribution but it is not an explicit part of the network. For example, the following statement defines an exponential whose mean is 12.

PARAMETER SET, 1, EXPONENTIAL, , 12

The parameter set number is 1 and it might be referenced on the Source node as in Figure 11 so that the interarrival of TVs from Source 1 will follow an exponential of mean 12.

SYSTEMS WITH ACTIVE RESOURCES

A particularly important and unique feature of INS is its ability to model systems where resources are themselves mobile in serving transactions at several activities within the network. Suppose our more experienced inspector can service transactions from either of two queues (numbers 8 and 9) at the inspection activity and furthermore can also perform adjustment (we will assume that Queue 10 exists before the adjustment Activity 4). Thus he is mobile, performing at several activities, serving several queues. This means that if a transaction enters any of these queues and this inspector is idle, then he can be used to begin their service. However, when this inspector finishes either inspecting or adjusting, there may be TVs in all three queues and he needs to decide which to service. By default, he will simply inspect whichever TV has waited the longest. But in many instances this may be inaccurate, for the inspector's job description may call for him to exercise choice. To model this choice process, INS permits the user to create a queue selector tree, composed of selector nodes and queue node references, arranged to depict a resource's decision-making process. For example, Figure 12 might be the selector tree for the experienced inspector (resource number 4). By referencing Selector 20 as the primary selector node for resource 4, the resource will employ this tree in determining which transaction to service. The selector node contains a selector rule that describes the selection process. Several rules are available in INS and the one in Selector 20 is called PREferred order while the one for Selector 21 is Longest Queue Wait. The selector tree is read left to right and top to bottom. Hence, the inspector will prefer (PRE) Selector 21 to a transaction in Queue 10, but at Selector 21, he will choose the transaction in Queues 8 and 9 which has been waiting the longest. If no transactions satisfy the algorithm, the resource becomes idle.

Notice that by referencing queues rather than activities in the selector tree, the modeling of activities with multiple queues is enhanced. Resources can then give priorties to queues which they serve. Combining this feature with the ability to rank transactions in the queue adds great flexibility to modeling complex queuing processes.

MORE ADVANCED INS CONCEPTS

Space does not permit the discussion of additional INS concepts and our purpose here is simply to convey some fundamentals of the INS modeling philosophy. However, three features merit mentioning.

One is user-assigned attributes. INS offers the convenient capability for the assignment of several types of attributes. These attributes can be used as variables for introducing counters, switches and totals into the network. Furthermore, extensive arithmetic is available to permit complex manipulation of attribute values. Attributes can be used for indirect specification within the network. For example, attributes can refer to parameter sets and resources. Attributes can be used to route transactions in the network, assign resource requirements at an activity, rank transactions in a queue, and serve as the basis for the assignment of other attributes. In addition, statistics can be collected on the attributes. Attributes are changed within the network as the situation dictates.

A second important feature of INS is its ability to model preemptive relationships among activities using resources. This feature allows the modeler to say that certain activities can preempt less important activities for resources. Thus, if adjustment of TVs is given absolute preference to inspection, the adjustment activity can be specified to preempt resources from inspection when needed. Such a modeling capability further enhances the active nature of resources within the network.

The third feature that merits mentioning is the ability to model processing dependencies among transactions. Such dependencies occur when the continued processing of one transaction depends upon another transaction. If TVs were loaded into a truck rather than leaving at Sink 3, a truck-full of TVs would need to be collected before the truck could leave. Hence a transaction may wait because of a synchronization requirement. This added complexity can be handled with the addition of one statement, thus encouraging the modeling of a broader variety of queuing circumstances in addition to those that arise because of resource unavailability.

REMODELING THE TV INSPECTION AND ADJUSTMENT EXAMPLE

To cap our discussion of modeling with INS, we will reconsider the TV Inspection and Adjustment example. In addition to our previous description of the problem, we add the following embellishments. Interarrival times of the TVs follow an exponential distribution, the inspection activity times are taken from Erlangs of order 3, and the adjustment activity time is lognormal. Four resources are available; one is a testing machine to be used at inspection; one is an adjustor; and the other two are inspectors, one is new and the other experienced. The new inspector arrives on the job 30 minutes after the start of the simulation. The experienced inspector can inspect and help out at the adjustment station, but should give preference to inspection. At the inspection activity, the newer inspector should be used first if both inspectors are available, while at the adjustment activity the adjustor should be prefered. Since resources may not always be available at the activities, queues are needed. Two queues exist at the inspection station, one for newly arriving TVs and a second for TVs being reinspected. Those TVs which have been in the inspect-adjust system the longest should be given priority at the point of reinspection. Also, with the addition of the testing machine, the inspection activity is divided into two parts. In the first part, the testing machine is used to diagnose circuit malfunctions. In the second part, whoever is inspecting the TV continues to inspect without using the machine. The complete INS network for the embellished problem is given in Figure 13. The network illustrates how the embellishments which have been described are easily integrated into a new INS network. The inspection is modeled with two activity nodes, the second of which has a resource selection mode that calls for the resource CURrently in use in the inspection operation. This specification frees the testing machine for diagnosis on another TV while retaining the inspector. Since we continue to utilize a resource, no queue is needed as the TV will never need to wait for the resource to become available.

Even though the problem has been greatly extended, the network retains its intuitive appeal with the nodes and the branches providing a close analogue to the physical system. The depiction is concise, and sufficient alphabetic and numeric information exists so that the simulation model is fully documented from information contained in the INS network.

TRANSLATING THE NETWORK TO INS STATEMENTS

The network of Figure 13 has been translated into a complete set of INS statements in Figure 14. This is the information from which INS constructs and executes the simulation. The free form nature of the statements when accompanied by suitable comments (comments in INS can be placed in any line after the $ symbol) render the input very readable, even by persons unfamiliar with INS. This further enhances model documentation, and structures the model, so that even the modeler will find it helpful as modifications and enhancements of the model are made.

The input statements can be separated into two parts. The first part serves as a preamble to the description of the graphic network. The preamble usually begins with a GENERAL statement (statements are defined by their first word) that contains the simulation project title and other overall information. In this case, five runs of the simulation are requested, each run lasting 480 minutes (or one day). Other preamble statements define the structure of the simulation model. For this problem, parameter sets and resources are needed. Notice that the resource's queue selector trees, as defined by the SELECTOR statements, are also included in the preamble.

The second portion of the INS statements describes the network of transaction flow. Here the correspondence of the graphic network to the set of statements is virtually identical where each symbol has its own statement. In general, the rule is that each node corresponds to a statement, therefore changes to nodes or even the deletion or addition of nodes are isolated in one statement. This means that working models are easily changed.

Notice that the input involves no FORTRAN, ALGOL, or other general purpose programming. Furthermore, the modeler need not be concerned about the mechanics of the simulation, i.e. event handling, file manipulation, statistics collection, and random deviate generation. This lack of mechanical involvement encourages the modeler to focus his creative attention on the model building by placing very few requirements on him to translate and program the simulation model once the problem is formulated through the INS network. He simply works at the problem definition stage, leaving to INS the details of the simulation. These features are especially important to the naive or infrequent simulation modeler.

INS OUTPUT

Without any special instructions, INS automatically provides output on the construction and execution of the simulation model and output which statistically summarizes the behavior of the model. The output will include a listing of the input statements and an echo of the simulation variables both user- and system-supplied. If any errors are found by INS in construction or execution, INS prints an appropriate error message. In general, INS attempts to execute the model whenever possible, supplying variables by default, and warning the user of suspicious circumstances. The echo, therefore, totally defines the simulation problem as INS understands it and is particularly useful to the modeler for debugging.

The summary output is the statistical description of the model results. Four points of view of the simulation are given by default. First, overall statistics about the run are given, such as the number of runs included in the report and the time during which statistics are collected.

Second, statistics relating to nodes are provided. Node counts indicate the number of times each node has been encountered by transactions. Node statistics are generally related to occupancy of each node by transactions. For example, queue nodes and activity nodes provide statistical data (such as mean, standard deviation, minimum, and maximum) on the average number and average time transactions spent in each queue and in each activity. Queue time statistics are subdivided to include and exclude those transactions which do not wait but pass through the queue in zero time. Sink node statistics provide total time in the system for transactions which exit the system at the particular sink node and this time is divided into the total time spent in queues and activities.

The third view reflected by the output is the transaction statistics. These statistics indicate the number and time of transactions in the network and also state the number of transactions created.

The fourth portion of the summary output gives resource utilization statistics. For each resource, resource utilization by resource state is given. Resource states include the time the resource is not in the network, the time it is idle, and the time it is busy at an activity. Results include the percentage of their utilization in each of the states as well as the average time they occupied that state.

Thus, the INS modeler will be able from the INS output to extract several perspectives on the results. The diagnostics will aid in verifying the model while the summary should reinforce the focus on modeling by generating data that should validate the modeler's view of the system's operation. This insight will hopefully lead to problem solutions and the testing of alternatives.

ENHANCING THE INS OUTPUT

INS has a number of provisions for expanding or even constraining the default output so more or less detailed information can be obtained. The execution of the simulation can be monitored in several ways. A system state trace can be activated to produce a motion picture-like presentation of the motion of transactions as they flow from node to node and the behavior of resources as they change their states. The trace can be constrained to display only certain resources or only particular transactions. Static pictures of the network during execution are also available, triggered either by the flow of transactions in the network or by time. These pictures may range from portions of the summary report to information on the current status of transactions and resources.

Summary report statistics can be enhanced in several ways. Additional statistical nodes can be added to the network to compute transit times between points in the network or interarrival times at some point in the network. Resource utilization statistics can be extended to yield data by individual activity and queue nodes. Furthermore, node statistics and resource utilization can be enhanced by the display of a histogram of the statistical distribution or a table of statistics broken down by transaction type. The table and histogram in Figure 15 were produced by the TABLE statement in Figure 14. It required only one statement to produce a table combining the statistics on the number of TVs waiting in Queues 8 and 9. Tables and histograms for resources, attributes, activities, sinks, and other nodes can be equally simply created.

The extended output thus gives the modeler a broad, but convenient range of outputs. He can examine the detailed processing to debug or to otherwise verify his model and he can create sophisticated statistics to gain additional insight into the system's behavior.

APPLICATION AND AVAILABILITY OF INS

INS has been applied in several environments.

The thirty examples in the User's Manual (1) illustrate the range of applications for INS including queuing, inventory, computer, manufacturing, and service systems.

INS has been successfully used in simulation courses for industrial engineering students. Graduate students utilized it as a new network-based language while it has been used as the principle simulation methodology in an undergraduate course in systems analysis. The language has also been used as the basis of a senior elective in medical school. Medical students have used INS to create models for clinical investigation. These instances of its use in a teaching environment have demonstrated that, not only can the language be quickly learned and applied by persons with minimal technical expertise, but it can also be valuable to those with deeper appreciation of simulation.

INS is written in ANSI FORTRAN making it portable and has been implemented on a number of machines. The User's Manual and INS are available from the Regenstrief Institute, 1001 West Tenth Street, Indianapolis, Indiana, 46202. Because the language has been developed in a non-profit institute, INS is available at the current price of $400. User's Manuals are available separately. INS is actively being maintained by the Institute and is the subject of continuing improvement both to increase the efficiency of its execution and to extend its applications.

CONCLUSION

Simulation modeling involves constructing a model by abstracting data and processes from the system being studied and implementing the model on a computer. INS facilitates model construction by providing a convenient set of symbols representing processes such as queues and activities which enable the modeler to formulate the model quickly yet precisely. By a simple translation from the symbols to INS statements, the model is implemented without effort. INS automatically handles all details of the simulation including file handling, random deviate generation, and statistics collection and reporting, thus relieving the modeler of these tasks. This dual approach in a high level language allows the modeler to focus attention on the important tasks of model creation and validation. INS is easy to learn, to use, and to modify as has been shown by the example we have developed. Since INS is visually appealing and easily documented, it is an excellent vehicle for communication between the modeler and the client, encouraging cooperative participation. Last but not least, INS is written in ANSI standard FORTRAN and is available for a variety of computers.

REFERENCES

1. Roberts, S.D., and J.S. Schier, "INS User's Manual for Version 3", Regenstrief Institute for Health Care, Indianapolis, Indiana, 1978.

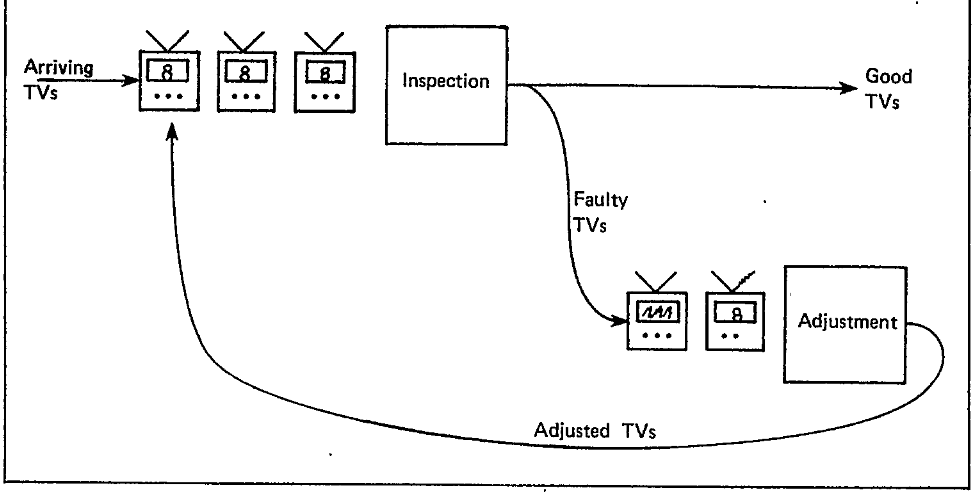


Figure 1: Flow Diagram of TV Inspection and Adjustment Problem

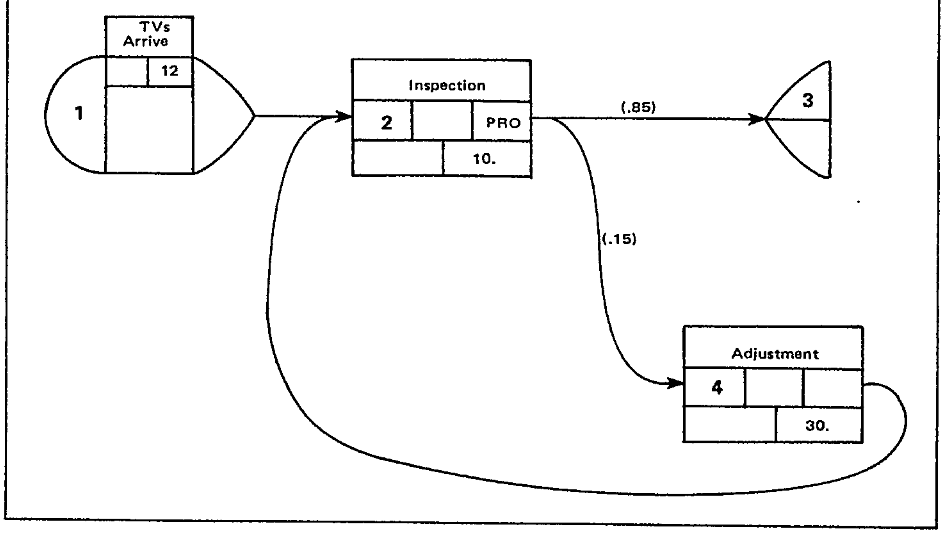


Figure 2: Initial INS Network

Figure 3: INS Statements

```
SOURCE, 1, , 12                     $ Arrival of TVs
     BRANCH TO, 2                   $ Go to Inspect
ACTIVITY, 2, PRO, (8)10.            $ Inspect TVs
     BRANCH TO, 3, .85              $ 85% pass
     BRANCH TO, 4, .15              $ 15% fail
ACTIVITY, 4, (8)30.                 $ Adjust TVs
     BRANCH TO, 2                   $ Reinspect TVs
SINK, 3                             $ Good TVs leave
```

Figure 4: Network Modification for TV Types

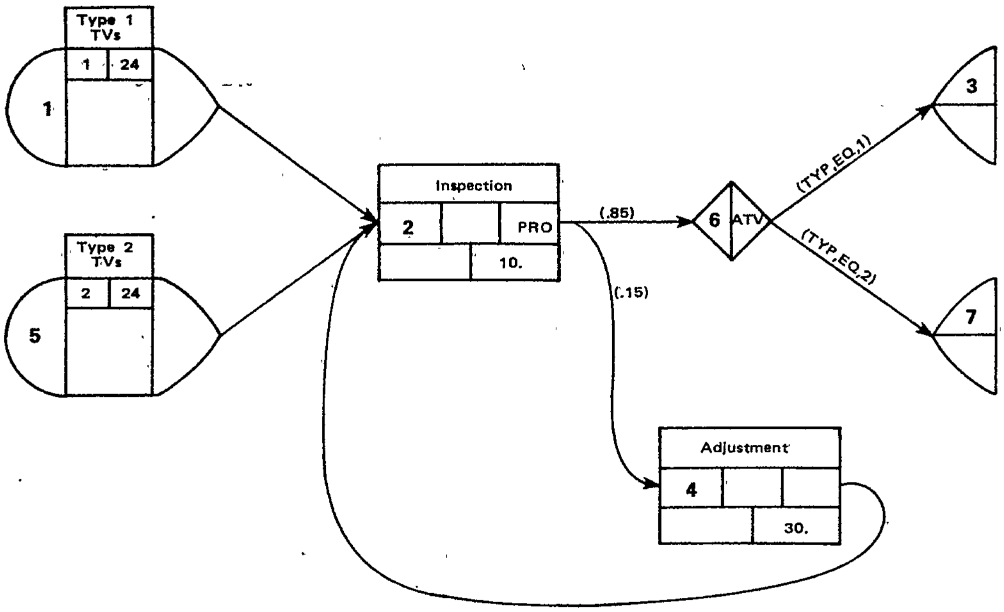


Figure 5: Using an Inspector

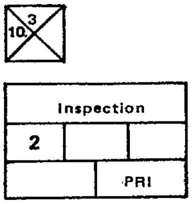


Figure 6: Alternative Inspectors

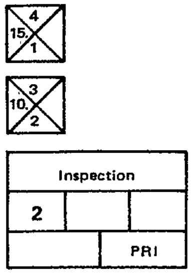


Figure 7: Two Resource Requirements

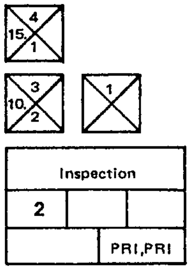


Figure 8: Queuing for Inspection

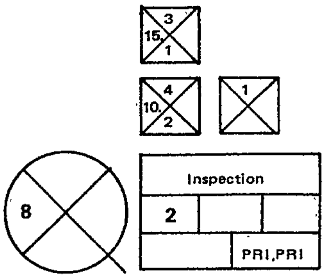


Figure 9: Queuing in Order of TYPe

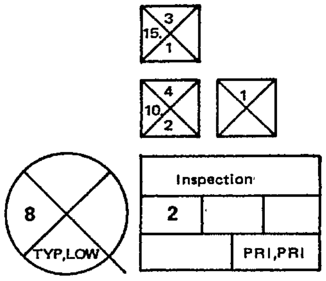


Figure 10: Multiple Queues

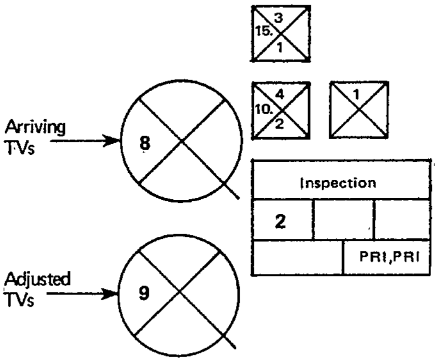


Figure 11: Stochastic Interarrival Times

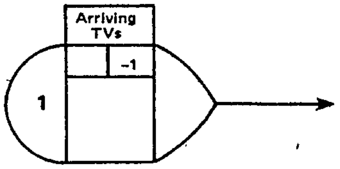


Figure 12: Queue Selector Tree

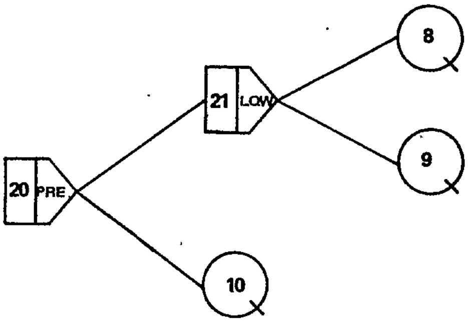
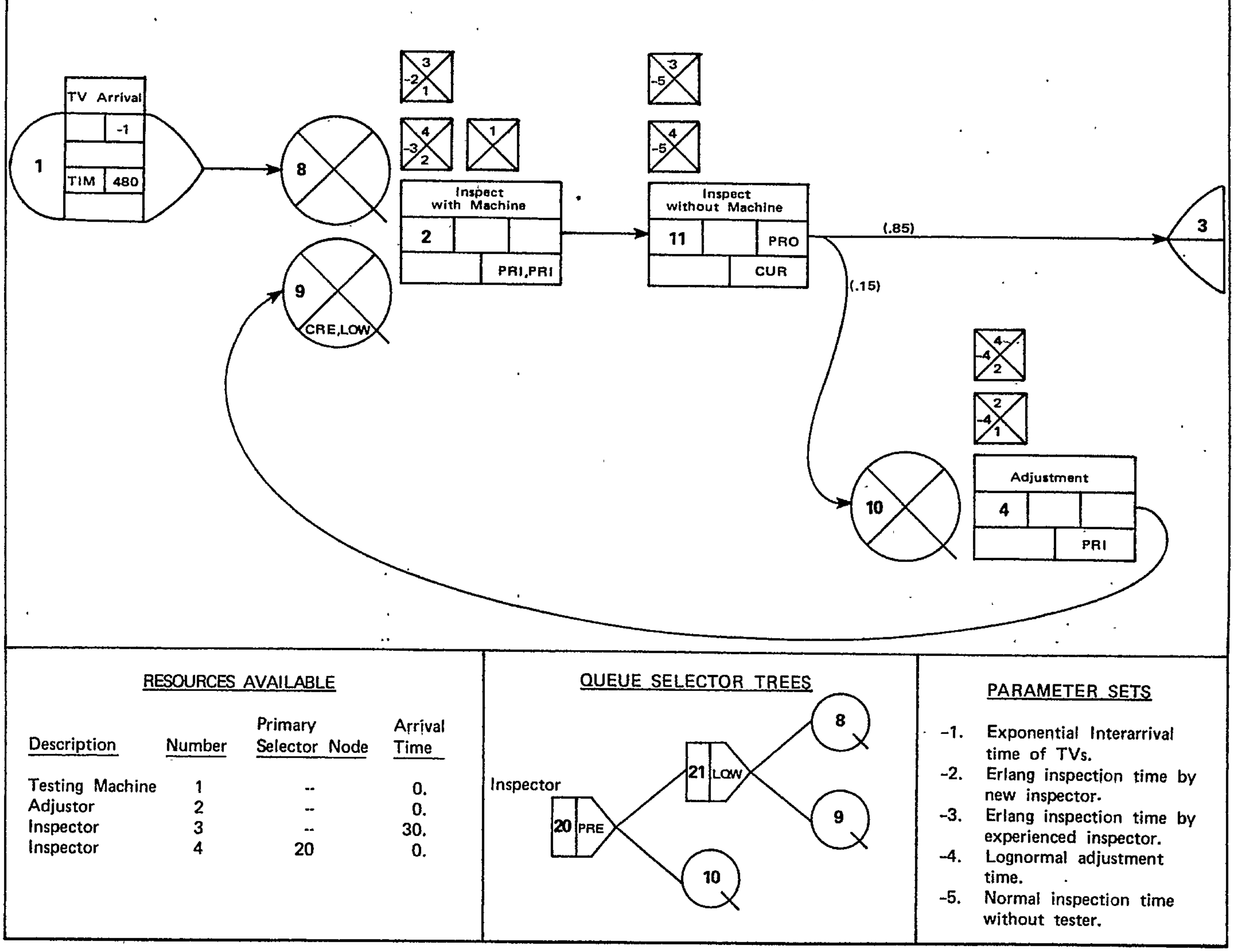


Figure 13: INS Network for TV Inspect and Adjust



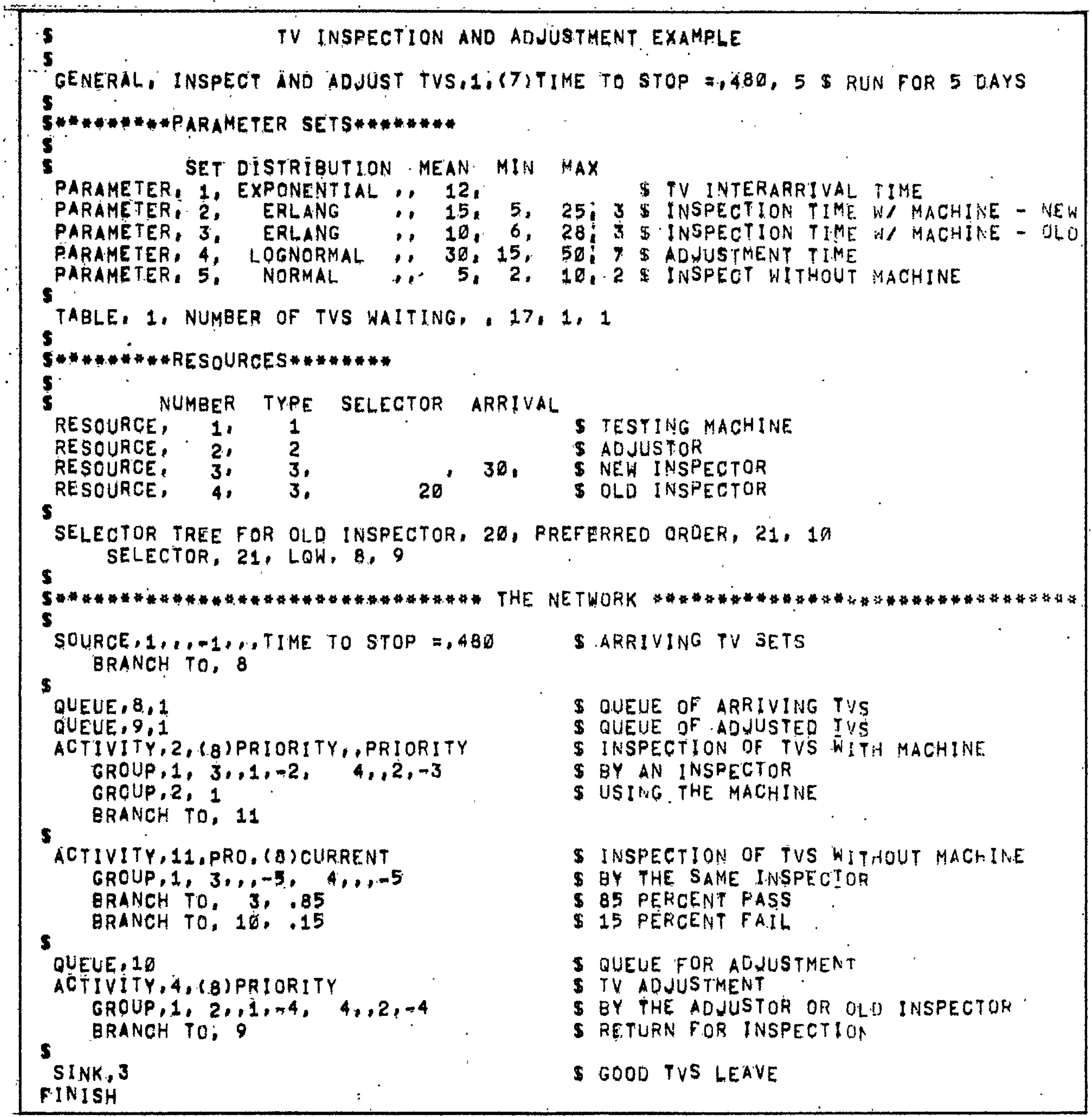
RESOURCES AVAILABLE

Description	Number	Primary Selector Node	Arrival Time
Testing Machine	1	--	0.
Adjustor	2	--	0.
Inspector	3	--	30.
Inspector	4	20	0.

PARAMETER SETS

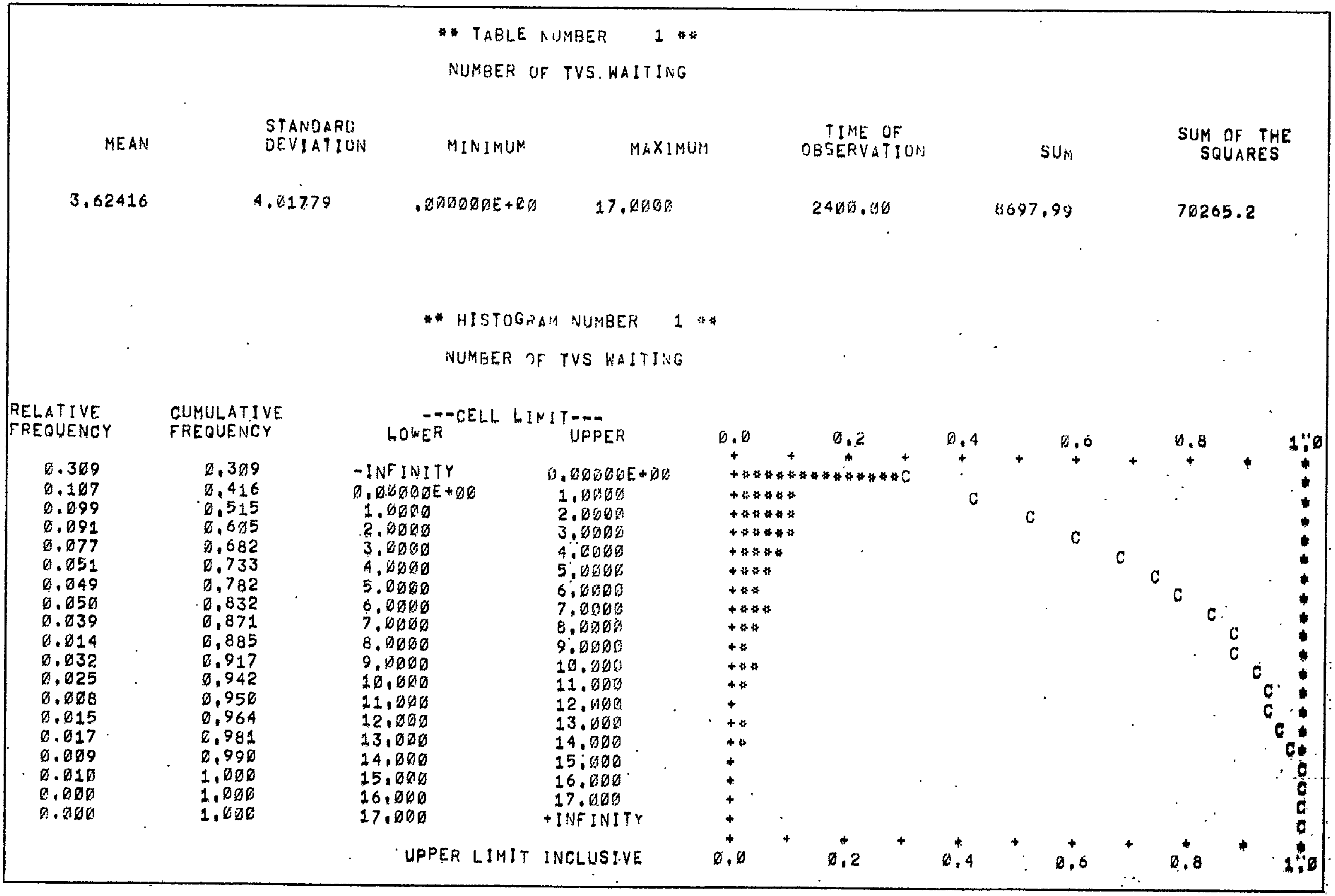
-1. Exponential Interarrival time of TVs.
-2. Erlang inspection time by new inspector.
-3. Erlang inspection time by experienced inspector.
-4. Lognormal adjustment time.
-5. Normal inspection time without tester.

Figure 14



```
$                    TV INSPECTION AND ADJUSTMENT EXAMPLE
$
 GENERAL, INSPECT AND ADJUST TVS,1,(7)TIME TO STOP =,480, 5 $ RUN FOR 5 DAYS
$
$*********PARAMETER SETS********
$
$           SET DISTRIBUTION  MEAN  MIN  MAX
 PARAMETER, 1, EXPONENTIAL ,,  12,                $ TV INTERARRIVAL TIME
 PARAMETER, 2,   ERLANG    ,,  15,  5,  25; 3 $ INSPECTION TIME W/ MACHINE - NEW
 PARAMETER, 3,   ERLANG    ,,  10,  6,  28; 3 $ INSPECTION TIME W/ MACHINE - OLD
 PARAMETER, 4,  LOGNORMAL  ,,  30, 15,  50; 7 $ ADJUSTMENT TIME
 PARAMETER, 5,   NORMAL    ,,   5,  2,  10, 2 $ INSPECT WITHOUT MACHINE
$
 TABLE, 1, NUMBER OF TVS WAITING, , 17, 1, 1
$
$*********RESOURCES********
$
$         NUMBER  TYPE  SELECTOR  ARRIVAL
 RESOURCE,   1,    1                          $ TESTING MACHINE
 RESOURCE,   2,    2                          $ ADJUSTOR
 RESOURCE,   3,    3,           ,  30,        $ NEW INSPECTOR
 RESOURCE,   4,    3,        20               $ OLD INSPECTOR
$
 SELECTOR TREE FOR OLD INSPECTOR, 20, PREFERRED ORDER, 21, 10
     SELECTOR, 21, LOW, 8, 9
$
$******************************** THE NETWORK ********************************
$
 SOURCE,1,,,-1,,,TIME TO STOP =,480       $ ARRIVING TV SETS
    BRANCH TO, 8
$
 QUEUE,8,1                                $ QUEUE OF ARRIVING TVS
 QUEUE,9,1                                $ QUEUE OF ADJUSTED TVS
 ACTIVITY,2,(8)PRIORITY,,PRIORITY         $ INSPECTION OF TVS WITH MACHINE
    GROUP,1, 3,,1,-2,   4,,2,-3           $ BY AN INSPECTOR
    GROUP,2, 1                            $ USING THE MACHINE
    BRANCH TO, 11
$
 ACTIVITY,11,PRO,(8)CURRENT               $ INSPECTION OF TVS WITHOUT MACHINE
    GROUP,1, 3,,,-5,  4,,,-5              $ BY THE SAME INSPECTOR
    BRANCH TO,  3, .85                    $ 85 PERCENT PASS
    BRANCH TO, 10, .15                    $ 15 PERCENT FAIL
$
 QUEUE,10                                 $ QUEUE FOR ADJUSTMENT
 ACTIVITY,4,(8)PRIORITY                   $ TV ADJUSTMENT
    GROUP,1, 2,,1,-4,  4,,2,-4            $ BY THE ADJUSTOR OR OLD INSPECTOR
    BRANCH TO, 9                          $ RETURN FOR INSPECTION
$
 SINK,3                                   $ GOOD TVS LEAVE
FINISH
```

Figure 15: Output From TV Inspection and Adjustment Example



```
                                ** TABLE NUMBER    1 **
                                 NUMBER OF TVS WAITING

                   STANDARD                                          TIME OF                      SUM OF THE
     MEAN          DEVIATION        MINIMUM          MAXIMUM       OBSERVATION          SUM         SQUARES

   3.62416         4.01779        .000000E+00        17.0000         2400.00          8697.99       70265.2



                                ** HISTOGRAM NUMBER   1 **
                                 NUMBER OF TVS WAITING

RELATIVE      CUMULATIVE          ---CELL LIMIT---
FREQUENCY     FREQUENCY         LOWER          UPPER       0.0       0.2       0.4       0.6       0.8       1.0
                                                            +    +    +    +    +    +    +    +    +    +    +
  0.309       0.309        -INFINITY    0.00000E+00         +**************C                                  *
  0.107       0.416      0.00000E+00         1.0000         +*****               C                            *
  0.099       0.515           1.0000         2.0000         +*****                    C                       *
  0.091       0.605           2.0000         3.0000         +*****                        C                   *
  0.077       0.682           3.0000         4.0000         +****                             C               *
  0.051       0.733           4.0000         5.0000         +***                                 C            *
  0.049       0.782           5.0000         6.0000         +**                                    C          *
  0.050       0.832           6.0000         7.0000         +***                                      C       *
  0.039       0.871           7.0000         8.0000         +**                                         C     *
  0.014       0.885           8.0000         9.0000         +*                                          C     *
  0.032       0.917           9.0000         10.000         +**                                           C   *
  0.025       0.942           10.000         11.000         +*                                             C  *
  0.008       0.950           11.000         12.000         +                                              C  *
  0.015       0.964           12.000         13.000         +*                                              C *
  0.017       0.981           13.000         14.000         +*                                               C*
  0.009       0.990           14.000         15.000         +                                                 C
  0.010       1.000           15.000         16.000         +                                                 C
  0.000       1.000           16.000         17.000         +                                                 C
  0.000       1.000           17.000      +INFINITY         +                                                 C
                                                            +    +    +    +    +    +    +    +    +    +    +
                    UPPER LIMIT INCLUSIVE                  0.0       0.2       0.4       0.6       0.8       1.0
```