An overview of a network design system

by W. E. BRACKER and B. R. KONSYNSKI
University of Arizona
Tucson, Arizona

INTRODUCTION

During the past few years there has been an increasing trend toward the development of on-line computer systems. By 1980 it is estimated that 70-80 percent of all larger computer installations will support some networking capability.[1] This trend has resulted in an increasing network user population and varying applications for computer-communication networks. Furthermore, due to economics, the prevalence of single-application networks is giving way to increasing numbers of multi-application networks.

This increase in the interconnection of computers has brought into focus the complexity of network design. While this is due partly to the size and diversity of computer networks, it is also due to the proliferation of available network hardware and facilities. As an example, there are over fifty different vendors (sales greater than one million/year) of data communication oriented hardware, and over twenty suppliers of data transmission facilities.[2]

NETWORK DESIGN

As with any system, computer communications networks are made up of various interrelated components, all of which are critical to the network design process. Some of these components (i.e. multiplexors, modems, terminals, etc.) are physical in nature, that is, they specify a piece of hardware or software with certain performance properties. Some network components, however, are not physical in nature but rather are considered to be logical components of network design. These include such design inputs as response time, security levels, and specification of user interactions. The logical design components are as critical to the design process as the various physical elements. Figure 1 presents a partial list of the physical and logical design components.

The main objective function in designing a computer communications network is the production of a minimum cost network which satisfies user performance requirements and design criteria. It is usually the case that an exact quantification of these parameters and constraints is not always possible. This is due to the unavailability of exact data regarding various network components. Estimation of the design parameters and constraints is accomplished by a study of user needs, by various statistical methods or by adopting industry standards. These approximations introduce errors in the design models employed during network design and analysis. The network planner also faces problems in deciding which network parameters are important to the design problem at hand; for example, some problems may require that modem turnaround time be specified exactly, while others may accept a rough approximation to this parameter.

Network design life cycle

The network design life cycle shown in Figure 2 reflects the general system design life cycle which has been presented by various authors.[3] It should be stressed that the difficult phases of the evaluation of user needs and design parameter determination in the design life cycle are often slighted by network designers.

Organizational impact

This design component is common to all systems design problems. We must completely analyze the impact of the proposed network or modifications to an existing network on the overall organizational structure. This impact analysis includes consideration of how much support upper management will give to the network during planning/design stages and eventual use of the system. In addition to analysis of the network impact on the current organizational structure, it is also necessary to measure the impact of the proposed system on future expansion and organizational goals.

Time-value of data

Data sent to or requested from the network must be processed within a given time period; therefore, it is necessary to evaluate network requirements based on response or other time performance criteria. This function is difficult to measure because it requires estimates by users who often have little or no idea about what they really want. Time-value of data also refers to the fact that some data may be made available almost instantaneously over a network but may not be used immediately. An example case is where a monthly status report is generated within five minutes of a request but is not used for a week after production.

TOPOLOGY/NETWORK ARCHITECTURES
MULTIPLEXING/CONCENTRATION
LINE TYPES
TRANSMISSION TYPE AND METHODS
MODEM TYPES
ERROR DETECTION AND CORRECTION
MESSAGE FORMATS AND ROUTING
MESSAGE SWITCHING TECHNIQUES
TERMINAL TYPES
PROTOCOLS
BACKUP CONFIGURATIONS
INTERFACE STANDARDS
USER INTERACTIONS
CARRIER SELECTION

Figure 1—Design components and parameters.

Traffic profiles

The definition of the information traffic profile is another area where quantification may not be possible; estimates of the type and arrival statistics of the incoming and outgoing data streams at a particular network node must be made. The network design process must provide some analysis and insight into the sensitivity of network performance due to variances of message structural estimates and assumptions.

Reliability analysis

Reliability normally is analyzed based on equipment failure rates (MTBF,MTTR) and data transmission error rates. Equipment failure can be measured on a statistical basis or on past experience while message error rates are a function of the error detection and correction features of the network and the line transmission error properties.

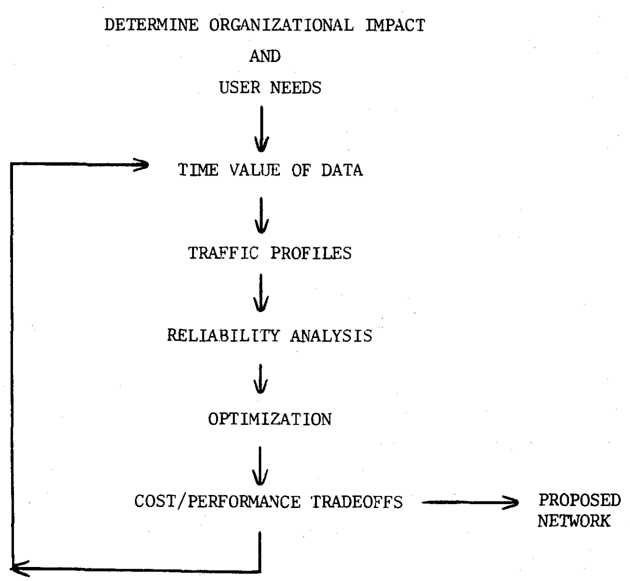

Figure 2—Network design life cycle.

Design optimization

There are two major steps in the design optimization process: (1) performance evaluation based on a given set of parameters and (2) determination of cost/performance tradeoffs. In the first step, a set of design parameters is specified and resultant performance is determined. The second step requires that a cost be determined for a given set of parameters and that cost/performance curves be established.

NETWORK DESIGN SYSTEM (NDS)

The Network Design System developed at the University of Arizona is an attempt to formalize the network design life cycle into a computer-aided design process (see Figure 3). NDS uses a Decision Support Philosophy[4] which provides the network planner/designer with maximum flexibility in the creation of an optimized data communication network that meets previously discussed design criteria and constraints. Using this methodology, the network planner/designer is concerned about what NDS can do in terms of network design and not the details about how it goes about its processing tasks. In particular, interfaces to the various design models are made as user transparent as possible.

Network planner/designer

The human interface to NDS is the network planner/designer; in most cases this consists of a group of individuals making up the planning/design team. These individuals state the network parameters and constraints to NDS using the

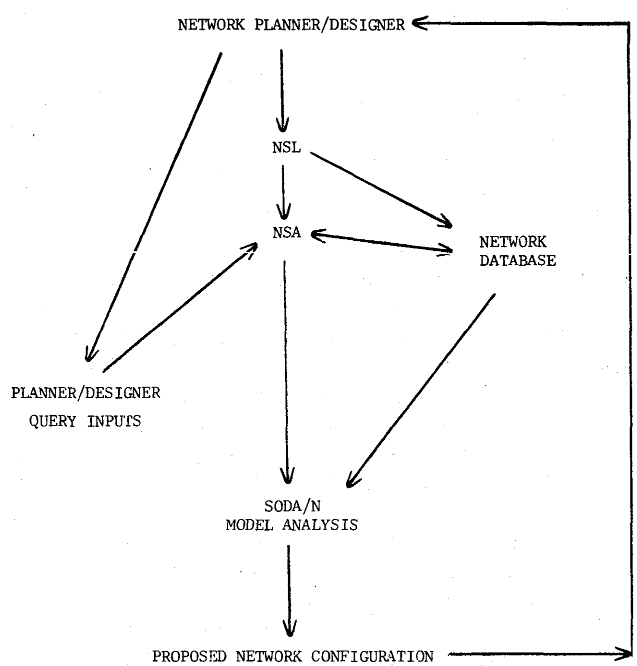

Figure 3—NDS Flow.

Network Specification Language (NSL) and the Network Statement Analyzer (NSA) query system.

Network statement language (NSL)

NSL is similar in form and structure to PSL[5] which is designed to provide a user with methodology for the statement of requirements of an information processing system. NSL, in a form compatible with PSL, allows a user to state design requirements for computer communication networks. The language provides an interface with a set of design models and is the main user contact with NDS. The current implementation of NSL has fourteen sections and thirty-five connector words or individual statements acting as "adjectives" in describing their particular sections. NSL allows the description of the physical and logical network constructs discussed previously. NSL section types, individual statements, and connector words are shown in Table I.

NSL is specified using the META/Generalized Analyzer[6] methodology of the ISDOS project based at the University of Michigan. Both META and the Generalized Analyzer are in a significant prototype stage and are not currently available to the general public.

META analyzes the description of NSL and produces a database containing the language structure. After NSL has been specified and processed by META, the META Generalized Analyzer (GA) processes user-supplied NSL problem statements, analyzes the syntax and portions of the semantic relations and iteratively builds the Network Database. NSA uses the constructed database for its processing requirements.

Basic flow between NSL, META and the Generalized

TABLE I.—NSL Reserved Words

NSL SECTION TYPES

NODE	HOST
TOPOLOGY	DATABASE
TRAFFIC	MULTIPLEXOR
INFORMATION-USER	TERMINAL
TERMINAL-USER	INFORMATION-CENTER
APPLICATION	LINE
REPORT	DATA-SET

NSL STATEMENT TYPES

TERMINAL	INTELLIGENCE	HOST
MULTIPLEXOR	INFORMATION-USER	RESPONSE
TOPOLOGY	TERMINAL-USER	SIZE
HOST	SEE-MEMO	DATABASE
LINK	TRAFFIC	SEE-MEMO
FAN-IN	FLOW	HAPPENS
FAN-OUT	KEYING	LOCATION
PROTOCOL	PRIORITY	RELIABILITY
TYPE	GENERATES	COST
MODE	RECEIVES	SECURITY
ERROR	SYNONYMS	DESCRIPTION
KEYWORDS	APPLICATION	

NSL CONNECTOR WORDS

TO	BY
FOR	VIA
IS	ARE

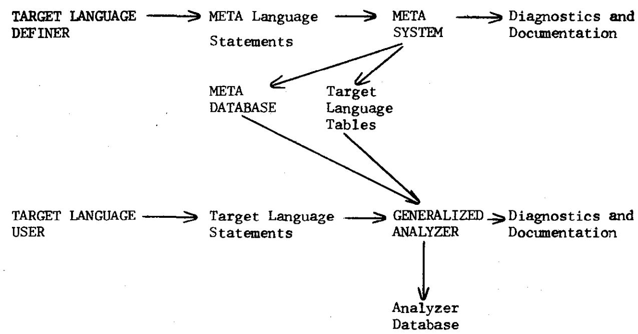

Figure 4—META system and generalized analyzer.

Analyzer is shown in Figure 4. The target language (NSL) is defined using META constructs; META then analyzes the target language and creates a database containing all target language objects types and their language inter-relationships. At this stage, META produces error diagnostics and various reports for further analysis and documentation.

After the target language has been successfully processed by META, sets of tables for the Generalized Analyzer are produced. The Generalized Analyzer enters the NSL network description into the network database and produces a series of reports which can be used by the network designer for documentation and/or analysis purposes.

Network statement analyzer

The Network Statement Analyzer is the NDS processor designed to accept network specifications produced by NSL and also obtain and process design inputs and report requests obtained by the query system. The NSA query processor produces a menu which allows the NDS user to select the supported NSA statement types as shown in Table II.

Consistency checks on the database produced by META/GA are performed by NSA. An NDS user can select checks based on various hardware connections: among these, whether all nodes in a proposed network can be reached by all other nodes, making sure that line and terminal speeds match, and verification of proper multiplexor/concentrator connections.

SODA/N

The Network System Optimization and Design Algorithms provide a set of models used to evaluate various design al-

TABLE II.—NSA Supported Statements

NDS/NSA STATEMENT TYPE	PURPOSE
QUERY	Activate NSA query processor
MODEL	Activate NDS supported models
CONSISTENCY	Perform consistency checks on the network database
MACRO	Activate the NSL macro pre-processor
INVENTORY	Various network reports

ternatives. In the current implementation of NDS, five design models are available; overall model implementation and integration philosophy is to create a design interface which requires minimum user interaction. Outputs from the models are also processed by NDS so that minimum user interpretation is needed; the user is only concerned about what information the models provide and not details on how they provide that information.

The models derive their inputs from two sources: NSL design specifications and NSA user interactions. The NSL design specifications produce a set of initial network conditions and assumptions to the design problem, while user interaction with NSA produces various design constraints and performance criteria. As an example, Table III shows the interaction between NSL, NSA and a capacity assignment model.

EXAMPLE

To illustrate the various components, the NSL section types NODE, TOPOLOGY, and TRAFFIC are implemented using the META methodology. The syntax description of these sections and associated statements are shown in Appendix A.

After the NSL syntax for the three sections and associated statements has been defined, the META representation for the abbreviated NSL is produced (Appendix B); this representation requires that all keywords, noise words and object types be defined. In addition, the relationship between objects must be specified along with template forms of the statements themselves. As was discussed previously, the result of this process is a database containing all the NSL language relationships and interface tables for the Generalized Analyzer. The system also produces a set of reports showing language structure and interrelations; a sample of this report type is shown in Appendix C.

NSL statements describing the proposed network design are input to the Generalized Analyzer. The Analyzer checks the incoming NSL syntax and places the NSL constructs into the META database. After all NSL descriptions have been processed, the Network Database is ready for access by NSA and its associated reports and models.

TABLE III.—Capacity Model Parameters

NSL-SUPPLIED PARAMETERS

	Parameter	Obtained From
T_{jk}	Messages/sec between two network nodes j,k	TRAFFIC-SECTION
N_{jk}	Connection matrix showing direct connections between $node_j$ and $node_k$	TOPOLOGY-SECTION
NSA-QUERY PARAMETERS		
U_i	Average length of messages over communication link i	
C	Overall network capacity	
R	Message Routing	

In order to show this process in greater detail, consider the small network shown in Figure 5 consisting of four NODEs (NODE1-NODE4) and five communication links (LINE1-LINE5). Traffic rates between any two node pairs are assumed to be symmetric and statistically independent of traffic between other node pairs. The NSL description of the simple example is shown in Appendix D.

Network topology showing direct connections between nodes is described in the TOPOLOGY-SECTION, while description of the individual nodes are shown in the NODE-SECTIONs. It is assumed that each node has two terminals, one multiplexor, and a host. Notice that the NODE-SECTIONs describe the hardware available at each node along with that node's location using a V/H coordinate scheme (LOCATION Statement).

Once the NSL description has been processed by the Generalized Analyzer and input to the Network Database, NSA is activated to produce database reports or activate various design/analysis models. In the example, we consider activation of a capacity assignment model.[7] This model establishes optimal link capacities based on network topology, message routing (assumed to be shortest path) and message traffic profiles. In the current implementation, Poisson Message arrival rates, exponential node service with infinite buffering are assumed. NSA accesses the Network Database and queries the user in order to establish the model input parameters. An example run of this model is shown in Appendix E assuming the Appendix D NSL description.

SUMMARY

The Network Design System provides an easy to use network planning and design tool; in addition, it allows a methodology of describing and evaluating existing networks. NSL statements are analyzed by the Network Statement Analyzer which, in turn, provides a Network Database, consistency checking, report generation, and model interfaces. Using the NDS approach, both existing and proposed systems are thoroughly analyzed. In addition, the top-down approach which is used with NDS allows the system planners/designers to maintain a perspective of the overall design

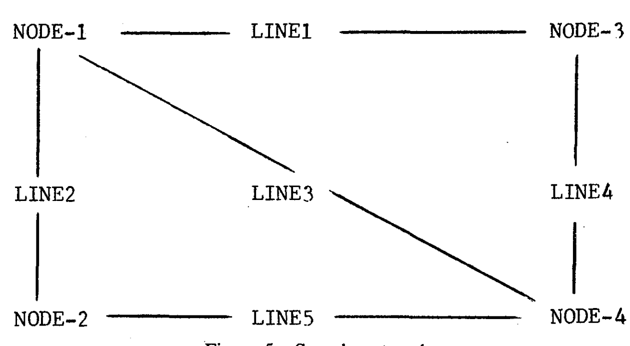

Figure 5—Sample network.

goals while at the same time allowing access to desired levels of detail in the design process.

REFERENCES

1. McLaughlin, R. A., "Piecing Together The Datacom Industry," *Datamation*, July, 1979, pps. 110-137.
2. Bracker, W. E., "On the Automated Design of Data Communication Networks," Working Paper, Univ. of Arizona, Dept. of MIS, 1979, unpublished.
3. Cougar, D., "Evolution of Business System Analysis Techniques," *Computing Surveys*, September 1973, reprinted in Cougar and Knapp, *System Analysis Techniques*, Wiley, 1974.
4. Konsynski, B. and Bracker, W., Computer-Aided System Design Tools Integration," MIS Technical Report, University of Arizona, 1979.
5. ISDOS Project, "Problem Statement Language (PSL): Introduction and User's Manual," ISDOS Project Report, University of Michigan, 1979.
6. Yamamoto, Y., "ISDOS META System Memorandum META-1,2," ISDOS Project, Dept. of Industrial and Operations Engineering, The University of Michigan, 1978.
7. Schwartz, M., *Computer-Communication Network Design and Analysis*, Ch. 4, Prentice Hall, Englewood Cliffs, N.J., 1977.
8. Konsynski, B., "A Model of Computer-Aided Definition and Analysis of Information System Requirements," Ph.D. Dissertation, Purdue University, 1976.

APPENDIX A: SAMPLE NSL SYNTAX

```
########### NODE-SECTION ##########

NODE-SECTION name(s);

SYNONYMS ARE synonym-name(s);

DESCRIPTION;
comment-entry;

KEYWORDS ARE keyword-name(s);

SEE-MEMO memo-name(s);

TERMINALS ARE terminal-name(s);

MULTIPLEXORS ARE multiplexor-name(s);

HOSTS ARE host-name(s);

LOCATION IS system-parameter, system-parameter;

#########TOPOLOGY-SECTION #########

TOPOLOGY-SECTION name(s);

SYNONYMS ARE synonym-name(s);

DESCRIPTION;
comment-entry;

KEYWORDS ARE keyword-name(s);

SEE-MEMO memo-name(s);

LINK {node-name ! terminal-name ! multiplexor-name}
     TO
     {node-name ! terminal-name ! multiplexor-name}
     VIA line-name;

##########TRAFFIC-SECTION ##########

TRAFFIC-SECTION name(s);

SYNONYMS ARE synonym-name(s);

DESCRIPTION;
comment-entry;

KEYWORDS ARE keyword-name(s);

SEE-MEMO memo-name(s);

TRAFFIC node-name TO node-name FLOW system-
parameter;
```

APPENDIX B: SAMPLE NSL META REPRESENTATION

```
KEYWORD LOCATION;
SYNONYMS LOC;

KEYWORD TRAFFIC;
SYNONYMS TRAF;

KEYWORD FLOW;
SYNONYMS FL;

KEYWORD LINK;
SYNONYMS LI;

NOISE-WORD ARE;
NOISE-WORD IS;
NOISE-WORD PER;
NOISE-WORD TO;
NOISE-WORD VIA;

OBJECT NODE-SECTION;
SYNONYMS N-S, NS;
NMCODE NMNODE 1;

OBJECT TRAFFIC-SECTION;
SYNONYMS T-S, TS;
NMCODE NMTRAF 2;

OBJECT TOPOLOGY-SECTION;
SYNONYMS TOP-S, TOPS;
NMCODE NMTOPO 3;

PROPERTY INTEGER-VALUE;
APPLIES ALL;
VALUES INTEGER;
```

PROPERTY STRING-VALUE;
APPLIES ALL;
VALUES ANY-VALUE;

PROPERTY NUMBER-VALUE;
APPLIES ALL;
VALUES ANY-VALUE;

RELATION LOCATION-RELATION;
PARTS LOCATION-OBJECT-PART, LOCATION-PART-1, LOCATION-PART-2;

COMBINATION LOCATION-OBJECT-PART NODE-SECTION WITH LOCATION-PART-1 VALUE-FOR INTEGER-VALUE WITH LOCATION-PART-2 VALUE-FOR INTEGER-VALUE;

CONNECTIVITY MANY LOCATION-OBJECT-PART ONE LOCATION-PART-1, LOCATION-PART-2;

CONNECTION-TYPE T5;
RTCODE RTLOCA 50;
STORED LOCATION-OBJECT-PART 3, LOCATION-PART-1 1, LOCATION-PART-2, 2;

STATEMENT LOCATION-STATEMENT;
USED LOCATION-OBJECT-PART LOCATION-RELATION;
FORM LOCATION IS LOCATION-PART-1, LOCATION-PART-2;

RELATION TRAFFIC-RELATION;
PARTS TRAFFIC-OBJECT-PART, TRAFFIC-PART-1, TRAFFIC-PART-2, TRAFFIC-PART-3;
COMBINATION TRAFFIC-OBJECT-PART TRAFFIC-SECTION WITH TRAFFIC-PART-1 NODE-SECTION WITH TRAFFIC-PART-2 NODE-SECTION WITH TRAFFIC-PART-3 VALUE-FOR NUMBER-VALUE;

CONNECTIVITY ONE TRAFFIC-OBJECT-PART MANY TRAFFIC-PART-1, TRAFFIC-PART-2, TRAFFIC-PART-3;

CONNECTION-TYPE F4;
RTCODE RTTRAA 55;
STORED TRAFFIC-OBJECT-PART 1, TRAFFIC-PART-1 2, TRAFFIC-PART-2 3, TRAFFIC-PART-3 4;

STATEMENT TRAFFIC-STATEMENT;
USED TRAFFIC-OBJECT-PART TRAFFIC-RELATION;
FORM TRAFFIC TRAFFIC-PART-1 TO TRAFFIC-PART-2 IS TRAFFIC-PART-3;

RELATION TOPOLOGY-RELATION;
PARTS TOPOLOGY-OBJECT-PART, TOPOLOGY-PART-1, TOPOLOGY-PART-2, TOPOLOGY-PART-
COMBINATION TOPOLOGY-OBJECT-PART TOPOLOGY-SECTION WITH TOPOLOGY-PART-1 TOPOLOGY-SECTION WITH TOPOLOGY-PART-2 TOPOLOGY-SECTION WITH TOPOLOGY-PART-3 VALUE-FOR STRING-VALUE;

CONNECTIVITY ONE TOPOLOGY-OBJECT-PART MANY TOPOLOGY-PART-1, TOPOLOGY-PART-2, TOPOLOGY-PART-3;

CONNECTION-TYPE F4;
RTCODE RTTOPA 60;
STORED TOPOLOGY-OBJECT-PART 1, TOPOLOGY-PART-1 2, TOPOLOGY-PART-2 3, TOPOLOGY-PART-3 4;

STATEMENT TOPOLOGY-STATEMENT;
USED TOPOLOGY-OBJECT-PART TOPOLOGY-RELATION;
FORM LINK TOPOLOGY-PART-1 TO TOPOLOGY-PART-2 VIA TOPOLOGY-PART-3;

APPENDIX C: META SAMPLE REPORT-OBJECT SUMMARIES

Object name = NODE-SECTION Synonym(s) = N-S, NS

Relation name = LOCATION-RELATION
Part name = LOCATION-OBJECT-PART
Statement name = LOCATION-STATEMENT
Form = 1: LOCATION IS LOCATION-PART-1, LOCATION-PART-2 ;

Relation name = TRAFFIC-RELATION
Part name = TRAFFIC-PART-2
*** No usages

Relation name = TRAFFIC-RELATION
Part name = TRAFFIC-PART-1
*** No usages

Object name = TOPOLOGY-SECTION
Synonym(s) = TOP-S, TOPS

Relation name = TOPOLOGY-RELATION
Part name = TOPOLOGY-OBJECT-PART
Statement name = TOPOLOGY-STATEMENT
Form = 1: LINK TOPOLOGY-PART-1 TO TOPOLOGY-PART-2 VIA TOPOLOGY-PART-3 ;

Relation name = TOPOLOGY-RELATION
Part name = TOPOLOGY-PART-2
*** No usages

Relation name = TOPOLOGY-RELATION
Part name = TOPOLOGY-PART-1
*** No usages

Object name = TRAFFIC-SECTION Synonym(s) = T-S, TS

Relation name = TRAFFIC-RELATION
Part name = TRAFFIC-OBJECT-PART
Statement name = TRAFFIC-STATEMENT
Form = 1: TRAFFIC TRAFFIC-PART-1 TO
TRAFFIC-PART-2 IS TRAFFIC-PART-3 ;

APPENDIX D: NSL OF EXAMPLE NETWORK

```
TOPOLOGY-SECTION TOP 1;

LINK NODE-1 TO NODE-3 VIA LINE1;
LINK NODE-1 TO NODE-2 VIA LINE2;
LINK NODE-1 TO NODE-4 VIA LINE3;
LINK NODE-2 TO NODE-3 VIA LINE4;
LINK NODE-2 TO NODE-4 VIA LINE5;

TRAFFIC-SECTION TRAFFIC1;

LINK NODE-1 TO NODE-2 FLOW 9.05;
LINK NODE-1 TO NODE-3 FLOW 6.12;
LINK NODE-1 TO NODE-4 FLOW 3.00;
LINK NODE-2 TO NODE-3 FLOW 4.50;
LINK NODE-2 TO NODE-4 FLOW 1.00;
LINK NODE-3 TO NODE-4 FLOW 10.8;

NODE-SECTION NODE-1;
TERMINALS ARE T1-1, T2-1;
MULTIPLEXOR IS MUX-1;
HOST IS CPU1;
LOCATION IS 10, 25;

(REPEAT FOR NODE-2 ... NODE-4)
```

APPENDIX E: NDS EXAMPLE

```
NDS—UNIVERSITY OF ARIZONA VERSION 1.0 8/3/
79 10:20

→NSL = TEST. NSL (File created in Appendix D)
→LISTING→TEST.LST (Source/Diagnostics File)
→DATABASE→TEST.DB (META Produced Database)
** NO DIAGNOSTICS **
  SECTIONS PROCESSED:6
  STATEMENTS PROCESSED:55
  OUTPUT FILE AND DATABASE FILE WRITTEN

** NDS/NSL COMPLETE 8/3/79 **

** NDS/NSA VERSION 1.0 8/3/79 **
OPTIONS:
1. EXIT
2. MODELS
3. REPORTS
OPTION→2
** NSA MODEL ANALYSIS **
ACTIVE MODELS:
1. CONCENTRATOR LOCATOR
2. CAPACITY ASSIGNMENT
3. TERMINAL LOCATOR
4. SPANNING TREE
MODEL→2
** CAPACITY ASSIGNMENT **
REQUIRED SECTIONS:
TOPOLOGY-SECTION
TRAFFIC-SECTION
** ALL REQUIRED SECTIONS CONSISTENT **
ROUTING→SHORT
CAPACITY→1000
** MODEL COMPLETE:CAPACITY **
RESULT FILE→TTY
MODEL:CAPACITY
ROUTING:SHORT
CAPACITY:1000
LINK      CAPACITY(BPS)
-----     -------------
LINE1     200
LINE2     450
LINE3     110
LINE4     100
LINE5     120
** ALL CONSTRAINTS MET

** NDS/NSA VERSION 1.0 TERMINATED 8/3/79 **
```