EVALUATING THE PERFORMANCE OF A UNIFIED SWITCHING NODE USING A SIMULATED NETWORK

Kenneth J. Bodzioch
Bernard E. Patrusky
RCA/Government Communications Systems Division

ABSTRACT

This paper describes a program which utilizes a discrete event simulation to drive a real switching node. Empirical measurements of various nodal performance characteristics are gathered and recorded during the exercise of this program to aid in the evaluation and design of candidate future nodal architectures.

Applications software and specialized hardware for a unified node which switches both digitized voice and data (packet) traffic were developed and tested in a flexible testbed facility at RCA in Camden, New Jersey. Both voice and data traffic are switched over common trunking facilities employing dynamic channel allocation. All packet transfers are made under the protocol set forth by the ANSI Advanced Data Communication Control Procedures (ADCCP).

To measure the performance characteristics of the unified switching node, software was developed to simulate the node/network and node/source terminal interfaces. Projected levels of voice and packet traffic were used as inputs to a discrete event simulation program which generates a source traffic tape in accordance with specified inter-arrival and holding time distributions. This traffic tape then serves as the traffic input to the network simulator software which examines and interprets each traffic event on the tape.

The network simulation software reacts to data arrivals by generating properly formatted packets, introducing these packets to the nodal applications program at the appropriate times, and providing all required network and source terminal protocol responses. Voice call arrival and departure events are used to alter the packet service rate on those trunks which employ dynamic channel allocation. Performance statistics and measurements are continuously collected by the network simulator and prepared for analysis by off-line data reduction programs.

INTRODUCTION

This paper outlines an ongoing investigation of the performance characteristics of a futuristic switching node. This "unified" node handles a variety of traffic classes including voice, packet data, and bulk data while offering several switching services such as circuit switching, message switching, and packet switching. The switch handles both local and tandem traffic.

The methodology developed to facilitate the measurement and evaluation of critical nodal performance parameters relies upon the credibility of the test node hardware and software configuration. Development of the test node utilized real hardware and real software elements which provides assurance that empirical factors have not been overlooked, and constitutes a useful hardware and software performance baseline against which candidate future nodal architectures can be compared. The steps taken to develop and validate the test node are enumerated below:

1) A system design was performed to implement the node functions within a single processor configuration.

2) Special front-end hardware designs were undertaken where such hardware development could obviously alleviate the processing burden.

3) A complete node applications software package was designed and implemented.

4) The hardware and software elements of the test node (see Figure 1) were partitioned to simulate a three node subnetwork by means of a re-entrant software architecture and back-to-back trunk connections. By utilizing three separate sets of tables and queues, the test node program assumes the identity of each subnetwork node as required.

5) The three node subnetwork was extensively tested using digital telephones and packet switched terminal devices. This testing effort serves to validate the applications software and specialized hardware by ascertaining that all nodal functions are operating as specified.

6) Network simulation software was developed to replace specialized front end hardware and to provide the necessary network/node and terminal/node interfaces (see Figure 2). The network simulation surrounds the test node applications software and permits the introduction of intensities of voice and data traffic which would

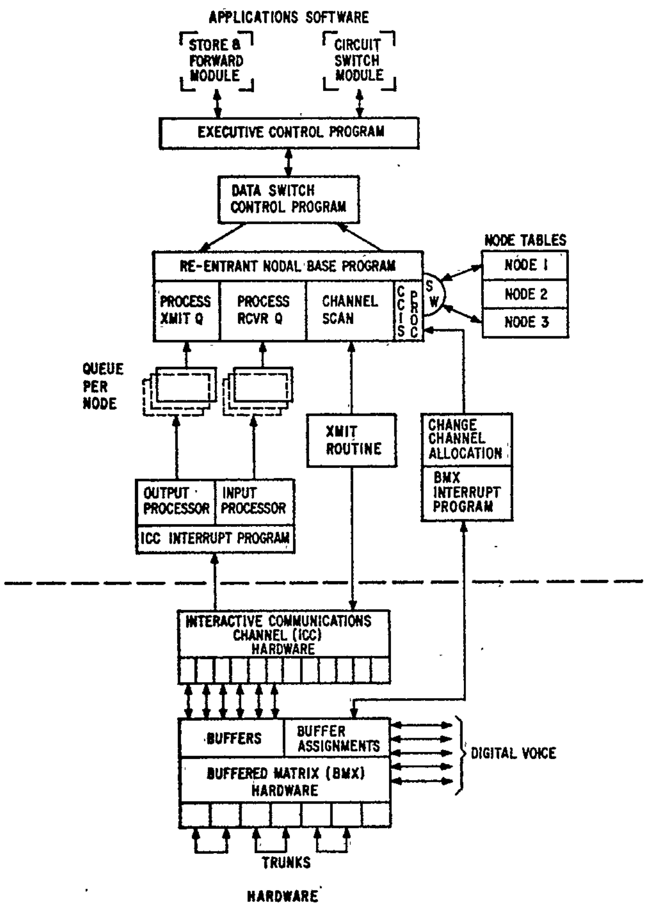

FIGURE 1 - TEST BED APPLICATIONS PROGRAM CONCEPT

otherwise require hundreds of phones, terminals, and operators to achieve.

7) A source traffic tape, previously generated off-line by a discrete event simulation program, serves as the traffic input to the network simulation software. Predetermined volumes of voice and data traffic are introduced to the network simulation software in time sequenced order. The network simulation software provides all network/node and terminal/node data interchanges and collects nodal performance statistics which are written to a record tape.

8) The recorded statistics are analyzed by an off-line data reduction program and various performance reports are automatically prepared. These reports are evaluated to determine the performance characteristics and traffic handling capabilities of the baseline test node configuration.

TEST NODE DESCRIPTION

Figure 1 is a functional block diagram which depicts the relationships between major software modules and specialized hardware within the test node. An executive control program schedules the Store-and-Forward, Circuit Switch, and Data Switch modules based upon demand. The Data Switch software processes all packet traffic and accounts for a major portion of the processing load.

The Data Switch software interfaces with the Interactive Communications Channel (ICC), a specialized front-end hardware device developed by RCA. The ICC transfers variably sized segments/packets between subscriber access lines, data trunks, and the processor memory via Direct Memory Access (DMA). All bit-and-byte-oriented operations required by the ADCCP link protocol are handled within the ICC, thus relieving the processor of a very heavy processing burden. The ICC delivers an interrupt to the processor only upon the complete reception or transmission of a data packet/segment.

Upon recognition of an ICC interrupt, software distributes received and transmitted packets to various queues associated with input or output link processing. The channel scan module then services these queues for each access line and data trunk.

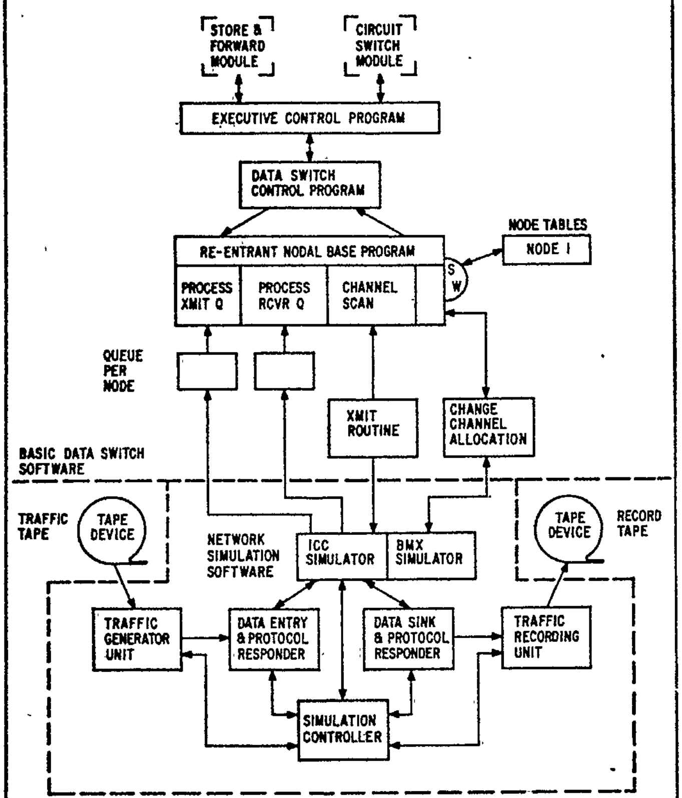

FIGURE 2 - ADDITION OF NETWORK SIMULATION SOFTWARE

The re-entrant structure of the Data Switch software permits the translation of ICC physical links to logical links which define the connectivity of a three node subnetwork. Using the same nodal applications program to sequentially process three sets of logical links permits functional verification of the test node applications software within the context of this three node subnetwork.

The second specialized hardware device developed for the test node is called the Buffered Matrix (BMX). This device compiles a multiplexed frame which mixes real time data (e.g. digital voice) and packet data. The BMX is controlled by the processor which dynamically determines the makeup of each frame. Circuit switched lines and ICC data trunks are inputs to the BMX while the multiplexed internodal trunks are outputs on transmit; the inverse is true on the receive leg. The BMX requires processor intervention only when voice calls are added to or deleted from the frame. This approach clearly minimizes the processor load attributed to the control of internodal trunks. The multiplexing process just described is referred to as "dynamic channel allocation" and simplifies the implementation of this unified node concept.

DYNAMIC CHANNEL ALLOCATION

An important feature of the test node is the capability to accommodate real time traffic such as digitized voice, video graphics, and facsimile (termed "Class I" service) as well as data traffic such as packets and AUTODIN line blocks (labeled "Class II" service). To accomplish this, all internodal trunks employ a multiplexed frame as illustrated in Figure 3. The composition of a frame is dynamically altered as digitized voice or other Class I calls are added or deleted. Following the start of each frame, slots bearing Class I traffic are always arranged contiguously, leaving the remainder of the frame available for data packets or other Class II traffic awaiting transmission. Thus, the trunk bandwidth available for Class II data varies inversely with the proportion of Class I traffic occupying the trunk at any given instant.

A maximum limit is imposed on the total Class I bandwidth permitted on each trunk. Voice calls blocked on a given trunk are either rerouted to a less occupied trunk or cleared. This limit is determined by the required voice grade of service and insures that a specified portion of the frame is available for packet data and other Class II traffic.

The behavior of the trunk data queue is extremely difficult to accurately determine by analytic means because of the dynamic variation of the data service rate. However, the pertinent characteristics of the trunk data queue are readily monitored during the experimental simulation runs.

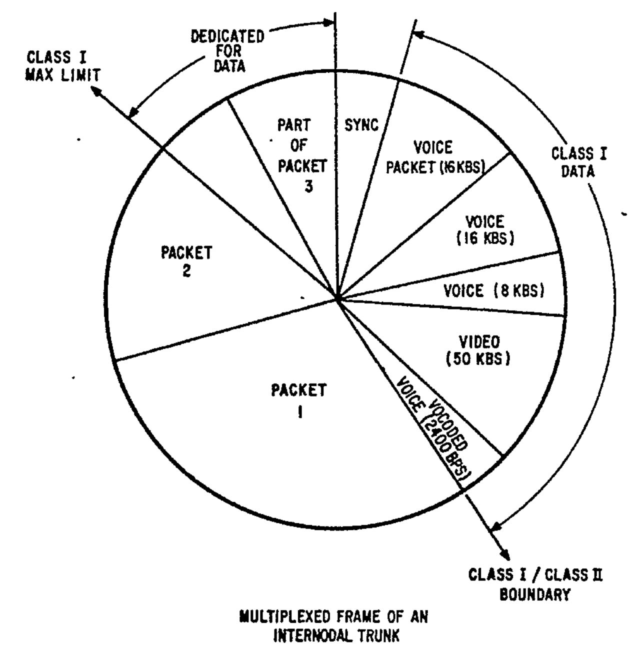

FIGURE 3 - DYNAMIC CHANNEL ALLOCATION CONCEPT

TRAFFIC GENERATION

In order to supply sufficiently large traffic loads to adequately stress the test node, compressed voice and data transactions are introduced by means of the pre-recorded source traffic tape. A typical traffic tape contains intermixed voice and data transactions ordered by desired time of entry. This input traffic tape is produced off-line by a discrete-event simulation program written in GPSS. The interarrival times, length/bandwidth distributions, precedence mix, holding times, etc., of the data and voice traffic are specified prior to the execution of the GPSS traffic simulation program. The resulting source traffic tape contains three types of transaction descriptors, each 32 bytes in length, namely: a) Voice Initiate, b) Voice Terminate, and c) Data. Each 32 byte entry on the traffic tape contains only transaction identification and control information; no text is actually included in the transaction entry. Transaction length (for data) or bandwidth (for voice) is specified within this 32 byte entry. The network simulation software uses the length of data transactions and the bandwidth of voice transactions to simulate the corresponding traffic load.

In the case of voice transactions, separate transaction entries indicate voice call "start" and voice call "termination". Voice transactions are used in the experiment to modulate the trunk data bandwidth available on incoming and outgoing trunk groups in the test node.

In the case of data transactions, the 32 byte transaction entry includes basic packet header data and information specifying the number of

Evaluating Unified Node Performance (Continued)

packets and size of the last packet in the transaction. The transaction entry is used by the network simulation software to generate individual packets which are sent to the test node on access lines or trunks according to node data flow control commands (e.g., first packet/segment is sent in immediately while subsequent packets/segments are called for by the node as required).

NETWORK SIMULATION SOFTWARE

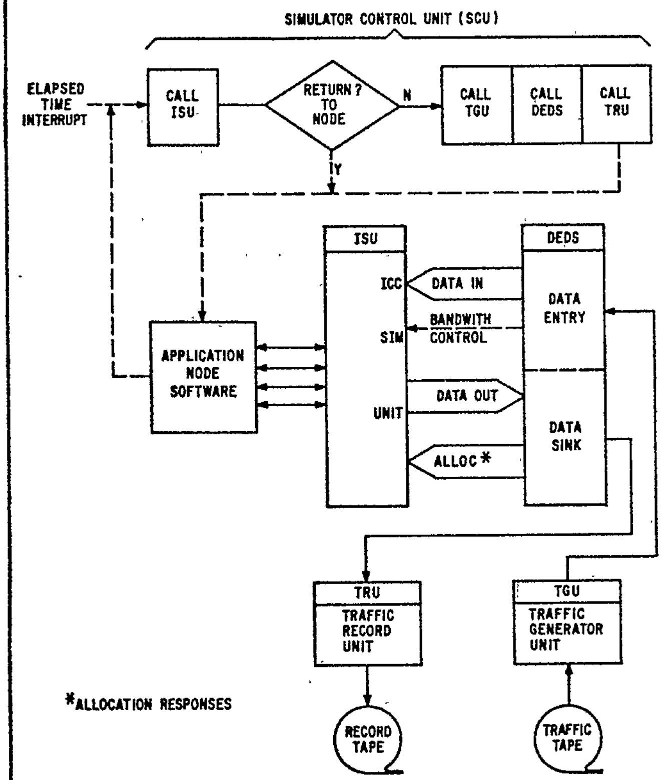

FIGURE 4 - EXPERIMENT SOFTWARE FLOW

As illustrated in Figure 4, the test node is surrounded by the network simulation software which provides the "illusion" of a connected network environment. Thus, the entire software package is logically divided into microscopic and macroscopic parts. The applications modules represent the detailed working functions of the microscopic portion, the test node, while the network simulation modules represent a macroscopic simulation of the surrounding network.

The network simulation modules are listed below:

a) Simulation Control Unit (SCU)

b) Traffic Generator Unit (TGU)

c) Data Entry/Data Sink (DEDS)

d) ICC Simulator Unit (ISU)

e) Traffic Recording Unit (TRU)

The functions of each of these network simulation software modules are outlined in the following sections.

SIMULATOR CONTROL UNIT (SCU)

This module coordinates the execution of the other network simulation modules, and manages buffer resources allocated to the network simulation software for storage of transactions in progress. In addition, SCU manipulates the elapsed time counter and the software clock. Part of this process involves passing control to other simulation modules and to the nodal applications program. Conceptually, the nodal applications program runs for a millisecond, and the network simulation program runs for as long as required to gather test data and perform required I/O operations. The "run time" of the test node is frozen while simulation programs are active. Thus simulation run times do not distort nodal delay measurements. Figure 5 depicts a typical sequence illustrating the internal progression of processing events within the processor. Hardware priority interrupts serve to transfer control from the application program to the simulation program at the expiration of the elapsed time counter.

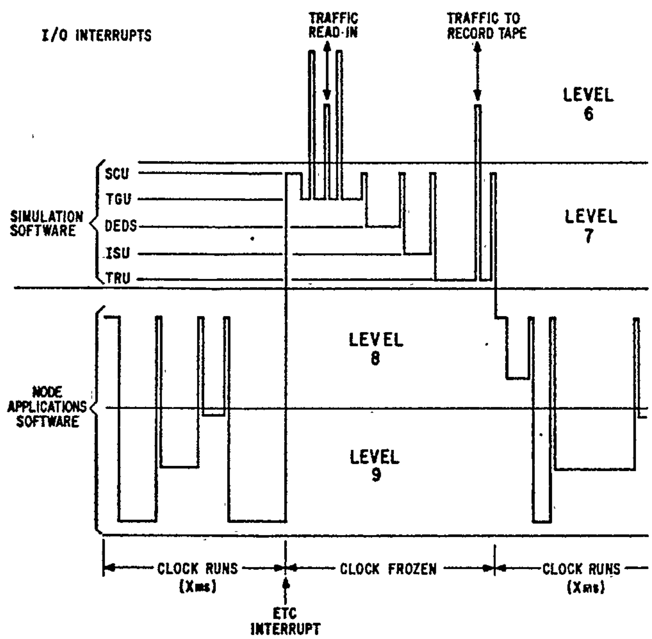

FIGURE 5 - EXPERIMENT SOFTWARE PROCESSING AND TIMING

TRAFFIC GENERATOR UNIT (TGU)

TGU is the network simulation software module which reads traffic blocks from the source traffic tape. These traffic blocks contain descriptors which represent voice and data transactions. TGU interprets each traffic descriptor, initiates both voice and data transactions at predetermined times, and terminates voice transactions as required.

TGU initiates voice transactions by subtracting the bandwidth specified in the traffic descriptor from the total bandwidth available on the appropriate internodal trunk(s). If sufficient bandwidth to complete the call is not available on the specified trunk(s), the call is blocked and cleared. TGU terminates voice transactions by adding the call bandwidth to the total bandwidth available on the appropriate internodal trunk(s).

At the appropriate time, TGU generates a properly formatted "Start of Message" packet or segment for all data transactions according to the information specified in the traffic descriptor. The first packet/segment is then queued to the selected access line or trunk for entry into the test node.

DATA ENTRY/DATA SINK (DEDS)

DEDS is the module which receives and interprets responses sent by the test node and generates properly formatted "Middle of Message" and "End of Message" packets or segments based upon these responses and information contained in the transaction descriptor. Thus, DEDS continues, and eventually terminates, all data transactions initiated by TGU.

Unlike TGU, which is driven by discrete traffic events recorded on the source traffic tape, DEDS reacts to actual traffic sent by the test node. Depending upon the packet or segment received, DEDS generates a command packet or the next sequential packet or segment of the data transaction, which is then queued to the proper access line or trunk for entry into the test node. Upon recognition of the last packet of a message, DEDS prepares a transaction termination record for output to the traffic record tape.

ICC SIMULATOR UNIT (ISU)

This module replaces the hardware Interactive Communications Channel (ICC) device whose function was to introduce and remove packets and/or segments to and from the test node. By simulating the ICC, physical limitations on traffic loading are removed. ISU provides the interface between the test node applications program and the network simulation software. Just as DEDS carried on the end-to-end protocol by responding to packets sent from the test node, ISU provides the ADCCP link protocol for all test node access lines and internodal trunks.

ISU functions in precisely the same manner as the hardware device it replaces. Upon the complete insertion or extraction of a data packet to/from the test node, a software controlled interrupt is generated. The nodal applications program services this interrupt as if the physical ICC generated it.

ISU also controls the delays associated with the simulated network and measures the processing delays incurred by packets sent to the test node. Packet delay records are prepared for output to the traffic record tape as packets exiting the test node are encountered.

TRAFFIC RECORD UNIT (TRU)

TRU is the vehicle by which measurements of critical nodal parameters are collected from the various other network simulation modules, formatted, and written to the traffic record tape. TRU handles both transaction termination records and periodic performance statistics. Traffic termination records occur whenever a voice or data transaction is completed (whether normally or abnormally). Performance statistics are compiled from various tables and counters throughout the network simulation software and written to the traffic record tape in prescribed format at periodic intervals which are specified by the test operator at program run time.

EXPERIMENT VARIABLES

The measurement of nodal thruput, delays, and similar nodal characteristics includes a range of situations produced by varying certain critical nodal and traffic parameters from one simulation run to the next. A comparison of the data generated over a series of runs yields information which can be interpreted as an indication of the sensitivity of the test node to the parameters involved. The parameters which are varied for these experiments include: traffic load, traffic composition, line and trunk rates, number of lines and trunks, voice grade of service, routing doctrine, and throttling rules. A further description of these variables follows.

NODAL CONFIGURATION VARIABLES

One fundamental group of variables involves setup of the nodal configuration to be tested. These programmable node configuration variables are established at the initiation of a given test and include:

a) Number of lines and trunks in the node.

b) Bandwidths of the lines and the trunks.

c) Maximum voice bandwidth on trunks.

d) Trunk routing algorithm.

e) Total buffers allotted for data.

f) Throttling thresholds.

g) Pre-emption/Priority rules.

NETWORK TURNAROUND DELAY

When a source node transmits packets to a destination node, the source node must receive responses returned by the destination node, which indicate new resource allocations, so that the source may continue to send. The delay of the destination node response is a function of network congestion and the number of nodes traversed. This response delay affects the distribution of incoming packet traffic. Short delays result in fast transaction handling or bursty packet traffic. Long delays tend (1) to throttle traffic and produce a more even traffic pattern. The network simulation program generates a dynamic value for the turnaround delay which is based upon two factors: the nodal delay currently measured in the test node for a packet of the priority in question, and the estimated total propagation delay.

TRAFFIC VARIABLES

The primary traffic variables for voice and data are enumerated below. All traffic is generated with Poisson arrival distributions and exponential length distributions.

VOICE TRAFFIC VARIABLES

1) Mean voice intensity level (measured in erlangs).

2) Percentage distribution of voice call types categorized as: Local (line/line), Tandem (trunk/trunk), Incoming (trunk/line), and Outgoing (line/trunk).

3) Mean voice call holding time.

4) Voice call bandwidth distributions (calls of various digital rates can be accommodated by the node. The mean percentage of calls for each rate present in the traffic mix is specified.)

DATA TRAFFIC VARIABLES

1) Traffic volume (stated in terms of transactions per hour).

2) Percentage distribution of transaction types categorized as: Local (line/line), Tandem (trunk/trunk), Incoming (trunk/line), and Outgoing (line/trunk).

3) Transaction Precedence/Category distributions.

4) Transaction length distributions (Mean lengths of transactions are expressed in bytes for each transaction category. Interactive, Narrative/Record, Bulk 1, and Bulk 2 data transactions have different mean lengths).

(1) Satellite links in the transmission path, for example.

DATA COLLECTION

In order to monitor the performance of the test node, the experiments develop output tape records for use by the offline data reduction program. The nodal parameters recorded for each test run are:

a) Incident traffic statistics (numbers of data and voice transactions in progress, etc.).

b) Voice grade of service.

c) Transaction acceptance and delivery delay statistics.

d) Thruput statistics (Packets/sec into and out of the node).

e) Buffer utilization statistics.

f) Trunk group utilization statistics (% occupancy of trunk group bandwidth).

g) Presence of saturation conditions.

h) Queue statistics.

i) Packet nodal delay statistics.

The basic method of collection consists of gathering a series of statistical data at periodic measurement intervals (i.e., once per second) and recording the data gathered. In addition, a termination record is prepared and recorded each time a voice or data call is terminated. The output record tape is then processed by an offline data reduction program to prepare the data for evaluation and analysis.

OUTPUT TAPE FORMAT

A single output record tape is prepared during the course of each test run. Data transferred to the record tape is written in fixed blocks of 520 bytes. Four different types of data blocks are recorded: 1) Label Block, 2) Packet Delay Block, 3) Completed Transaction Block, and 4) Major Data Block. A functional description of each block type merits further discussion.

LABEL BLOCK

This block is written on the output tape to identify the source traffic tape, test node configuration parameters, and network turnaround delay parameters used for the test run. This label is printed with all data reduction reports in order to correlate the test run with the collected data.

PACKET DELAY BLOCK

The packet delay block format is illustrated in Figure 6. Each entry indicates the delay in the test node measured for each packet leaving the node. The entry also indicates packet priority and type. (Packet type indicates whether packet is tandem, local, incoming or outgoing.) The primary purpose of this record is to obtain packet delay statistics (mean, maximum, variance, etc.) for each test run.

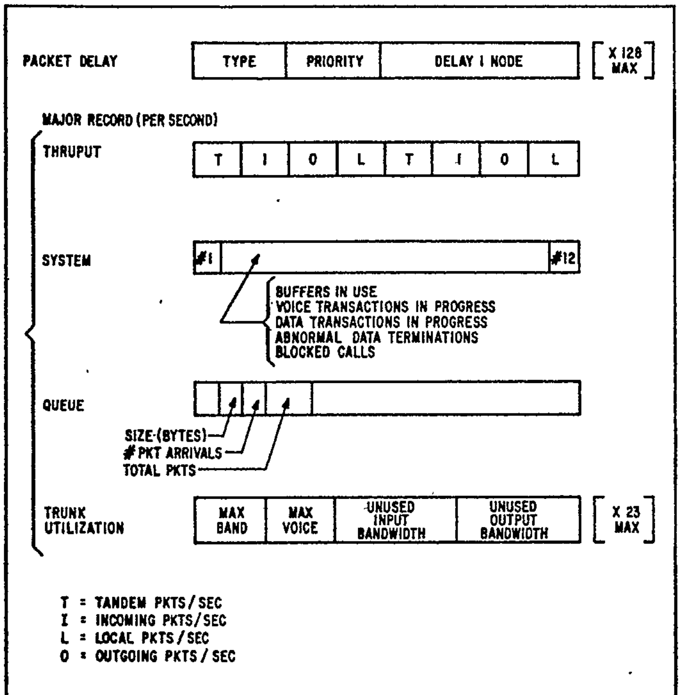

FIGURE 6 - TYPICAL RECORD TAPE DATA ENTRY FORMATS

COMPLETED TRANSACTION BLOCK

Each completed transaction entry contains all transaction descriptors and certain termination data. In the case of voice transactions, normal or "blocked" termination is indicated. In the case of data, "normal" or "abnormal" termination is indicated together with measured acceptance delay. Delivery delay and length of the data transaction are also available from each entry. Transaction acceptance and other delay statistics are derived from the data contained in this type of record to assess node performance.

MAJOR DATA BLOCK

The Major Data Block is composed of four types of data entries as shown in Figure 6. This block is recorded at fixed measurement intervals (i.e., once per second) to gather node status data on a dynamic basis.

The Thruput entry contains counts of packets entering and leaving the node during the measurement interval. Separate counts are recorded for each packet type (incoming, tandem, etc.), further subdivided by priority.

The System record represents the system state at the end of each measured interval and indicates the degree of congestion of the test node in terms of buffers utilized and number of voice and data transactions in progress. Instances of saturation (i.e., abnormally terminated data transactions) during the measurement interval are also recorded.

The Queue entry provides an in-depth examination of the dynamics of a preselected set of queues. Queue length, the number of packets in the queue, and the number of packet arrivals for each queue per measurement interval, are recorded. This permits analysis of data queue behavior for various mixes of voice and data traffic.

The Trunk Utilization entries for each trunk (at the end of each measurement interval) represent sufficient utilization data to permit determination of the current occupancy of the trunk for each class of traffic.

DATA REDUCTION

The output tape for each test run is processed, off line, to develop printouts of data statistics. The Data Reduction program develops statistical data for specific parameters of interest.

The Data Reduction program contains three basic subprograms. Each subprogram processes the raw data on the output tape to develop a different set of printouts. The subprograms are titled: 1) Data Edit, 2) Basic Statistics, and 3) Advanced Statistics.

DATA EDIT

This program screens and prints selected records from the output tape. Printouts are formatted so that successive records are time ordered and column aligned. The histogram of any specific parameter can therefore be readily observed. The edit output also serves as a quick check that the test run operated correctly. In order to limit the volume of printed data, the operator can select run time boundaries for start and stop of the edit printout.

BASIC STATISTICS

This program develops and prints the important statistical quantities associated with the dynamic parameters measured during a test run. The parameter categories examined by the basic statistics program for which mean, min, max and standard deviation are calculated, are listed below:

1) Packet Delay

2) Thruput

3) Queue Size

4) Number of Packets/Sec arriving to Queue

5) Number of Packets in Queue

6) Trunk Utilization

7) Voice Utilization

8) Buffer Utilization

9) Acceptance Delay

In addition, the relative frequency of throttling and other saturation conditions, average processor utilization, and the percentage of blocked voice calls are also determined and printed.

ADVANCED STATISTICS

This program develops and prints a probability density table and cumulative probability distribution table for any selected parameter previously analyzed by the Basic Statistics program.

In order to evaluate the behavior of the test node, it is now possible to plot mean, max, and 99% delays as a function of incident data load for given voice load and grades of service. Typical plots which are developed include:

1) Packet Delay vs. % Data Traffic

2) Acceptance Delay vs. % Data Traffic

3) Processor Utilization vs. % Data Traffic

4) Buffer Utilization vs. % Data Traffic

5) Thruput (Pkts/Sec) vs. % Data Traffic

6) Transactions/Sec vs. % Data Traffic

7) Pkts/Sec vs. Packet Delay

8) Trunk Utilization vs. % Data Traffic

In order to interpret the plots developed, those points where saturation, throttling, or other anomalies existed during the test run must be identified. Any one point plotted on any given curve is normally obtained from one test run.

DATA EVALUATION

The key performance criterion for node design of a voice system is achievement of the required grade of service for the specified voice traffic load. Similarly, the node design for data is predicated on achieving the required acceptance delay and delivery times for the specified precedence mix and traffic load. In order to size a node employing dynamically allocated trunks, the test data obtained will be evaluated to determine the data and voice capacity that can be supported by the single processor test node while achieving the required voice grade of service and data delays. Upperbound values for processor capability, buffer size, and trunk capacity can then be estimated for heavier traffic requirements. The design parameters thus developed will serve as guidance for developing a multiprocessor node architecture for larger traffic loads.

SUMMARY

A flexible tool has been developed to test and optimize the hardware and software elements of a unified node. The particular nodal performance parameters investigated are sufficiently sensitive to routing strategies, data and voice preemption/priorities, throttling rules, buffer size, etc., that manipulation and calibration of these algorithms and thresholds is necessary to maximize nodal thruput, minimize data delays, and meet voice grade of service requirements. Mathematical analysis holds little promise of accurate results primarily because of the complexity of a model which accounts for all variables.

Many of the variables involve discrete events which cannot be accurately described in analytical form. A Markovian analysis can be used to determine the probabilities of expected waiting times for very simplified trunk queue models. However, when dealing with mixed voice rates on a trunk, multiple trunks, changeable priority rules (i.e., where data may preempt voice when data queues reach certain thresholds), multiple data priorities, and other complexities, accurate analysis is extremely difficult.

A pure simulation would be more amenable to handling all the complexities introduced by this multiplicity of variables; however, a pure simulation employing postulated rather than actual nodal hardware and software would not provide an accurate indication of attainable thruput or required buffer memory size, since these parameters are sensitive to computer processing delays. Furthermore, the concurrent development of actual tested hardware and software provides a usable residue which can be captured and used in the planned unified node for the future. The final stage of this complex project is the actual conduct and evaluation of the experiments which are expected to be completed in the early part of 1977.

ACKNOWLEDGMENTS

This investigation is being accomplished under Government contract for the Rome Air Development Center (RADC) in conjunction with the Defense Communications Agency (DCA). The authors gratefully acknowledge the support provided by Mr. F. Kozien of RADC, and especially wish to thank the following people for their guidance and review: Mr. I. Susskind, RCA Program Technical Director; Mr. J. Swaim, RCA Program Manager; Mr. A. Mack, RCA Staff Consultant; and Mr. W. Gesek, Software Consultant affiliated with DMS Computing.

BIBLIOGRAPHY

1. ANSI Document X3S34/589 (Draft 5), "Proposed American National Standard for Advanced Data Communication Control Procedures (ADCCP) (Independent Numbering)," June 1976.

2. Coviello, G.J., Vena, P.A., "Concept for an Integrated Circuit and Packet Switched Telecommunications System," Defense Communications Engineering Center Technical Comment No. 8-75, January 1975.

3. Fischer, M.J., Harris, T.C., "An Analysis of an Integrated Circuit- and Packet-Switched Telecommunications System," NTC, pp. 42-18 - 42-23, December 1975.

4. Halfin, S., Segal, M., "A Priority Queueing Model for a Mixture of Two Types of Customers," SIAM Journal of Applied Mathematics, Vol. 23, pp. 369-379, November 1972.

5. Kimbleton, S. R., Schnieder, G.M., "Computer Communications Networks: Approaches, Objectives, and Performance Considerations," Computing Surveys, Vol. 1, No. 3, pp. 127-173, September 1975.

6. Kummerle, K., "Multiplexor Performance for Integrated Line and Packet Switched Traffic," ICCC, Stockholm, Sweden, pp. 507-515, August 1974.

7. Maiwald, D., Muller, H.R., Rudin, H.R., West, C., "Integrated Communication System Performance," ICCC, Seattle, Washington, pp. 24-13 - 24-21, June 1973.

8. Svobodova, L., "Online System Performance Measurements with Software and Hybrid Monitors," Proceedings ACM SIGOPS Fourth Symposium on Operating System Principles, pp. 45-53, October 1973.

9. Zafiropulo, P., "Flexible Multiplexing for Networks Supporting Line Switched and Packet Switched Data Traffic," ICCC, Stockholm, Sweden, pp. 517-523, August 1974.