SIMULATION MODEL OF THE CABLE DATA NETWORK FOR THE ANALYSIS AND EVALUATION OF NETWORK PERFORMANCE

D. Gan, Computer Sciences Corporation
R. Paterson, GTE Sylvania Systems Group,
Strategic Systems Division

ABSTRACT

A Cable Data Network (CDN) simulation model was developed on VAX 11/780 computer facility in PASCAL as a part of the MX-C^3 system study. Its primary purpose was to supplement theoretical analysis and to evaluate the impact of changing the CDN (sub)system requirements on the performance measured primarily in terms of network reaction time and queue (buffer) buildup at the CDN nodes. The validated simulation model provided a powerful tool in rapidly determining the quantitative measures on both the network reaction times and the buffer seizing at the CDN nodes, in both the normal and degraded network operating environments. These quantitative measures have resulted in the following major contributions:

1. Optimum message set design and routing strategies
2. Recommendation of a faster processor at manned CDN nodes
3. Important considerations for minimum buffer sizing at unmanned CDN nodes.

INTRODUCTION

BACKGROUND AND PURPOSE

The CDN simulation model, developed by Computer Sciences Corporation (CSC) on an MX-C^3 subcontract with GTE Sylvania, was intended for primary use with the preattack communications within the MX-Wing in the shelter-basing mode of the MX-C^3 program. The double-ring topology of the CDN (Figure 1) evolved over a period of a year after a series of investigations on the alternatives of interconnecting the manned control/support centers and the unmanned shelters primarily to satisfy the CDN system requirements on connectivity and failure detection. The control hierarchy of the double-ring CDN had a three-tiered design. At the highest level there were two (2) manned Operations Control Centers (OCC). There were four (4) manned Area Support Centers (ASC) at the intermediate level. At the lowest level there were 200 linear clusters. Each linear cluster contains one MX-missile in one of a variable number (20 to 30 with a baseline value of 24) of unmanned Horizontal Shelter Sites (HSS). The CDN system message set design included the number and types of messages, message length and formats, and message propagation. All design efforts including topology design, control hierarchy design, and message set design, focused on the CDN system mission requirements (Table 1) in arriving at the optimum or near-optimum solutions.

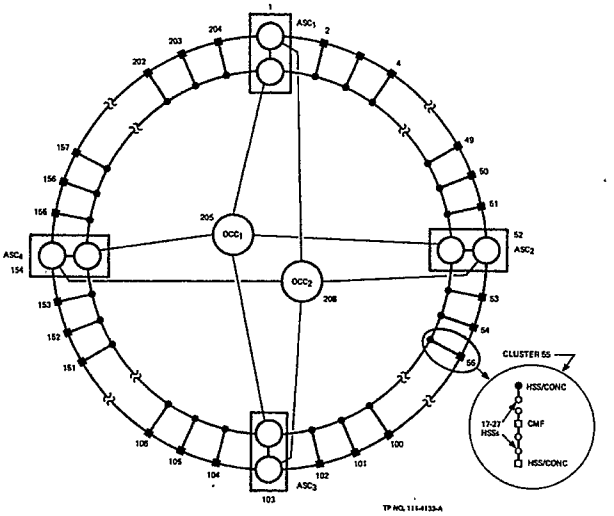

Figure 1. CDN Topology

The primary concern of the CDN system engineers and network designers, once the topology design was consolidated, was to determine how well the system would perform in envisioned operational environments. This was particularly true with regard to the network reaction times associated with message types and their propagation. The basic CDN of 4810 nodes

Proceedings of the 1982
Winter Simulation Conference
Highland * Chao * Madrigal, Editors

Table 1. CDN System Mission Requirements on Various Items

- o Connectivity
  - Single failure shall not isolate any shelter
  - Three failures shall not isolate more than one cluster from operational manned site
- o Failure Detection
  - Probability of false message propagation = $P_{fm} \leq 10^{-4}$
  - Probability of undetected message error = $P_{ume} \leq 10^{-10}$
  - Probability of detected message error = $P_{dme} \leq 10^{-4}$
  - Probability of missed message = $P_{mm} \leq 10^{-6}$
- o Reaction Time
  - Status collection/redistribution every two (2) minutes
  - Distribution of commands within five (5) seconds
- o Preservation of Location Uncertainty
  - Traffic analysis of encrypted messages shall not reveal missile launcher location
  - ASC shall not have access to missile launcher location.

was too large to analyze theoretically in predicting the expected queueing at many CDN nodes. There were also other key questions to be answered for hardware considerations including the buffer sizes and processing capabilities at a CDN node type, and the protocols at message and network levels. In addition, key decisions were required regarding the need for, and implementation of, error recovery and flow control. These considerations spurred the design and development of the CDN simulation model on CSC's VAX 11/780 computer facility in PASCAL. The following major goals served as the focus of the model effort.

1. To perform rapid evaluations of the evolving message sets design and requirements in the CDN system
2. To perform investigations and feasibility studies that supplement the analyses
3. To evaluate simulation model results and make recommendations based on these results
4. To perform CDN system vulnerability and survivability analysis on requirements related to:
   a. Network reaction times
   b. Failure detection.

The following technical approach was used in the CDN simulation model design, development and evaluation, which spanned a period of roughly one (1) year. The effort terminated as a result of the October 3, 1981, presidential decision on the new fixed-silo basing mode for the MX-missiles.

TECHNICAL APPROACH

Our technical approach followed the work breakdown structure shown in Figure 2. The figure shows the evolution of the model effort in two parallel paths. One path was related to the basic model and the other to the enhanced model, as the CDN system design matured. Enhanced model implied the incorporation of enhancements in the basic simulation model to reflect additional CDN capabilities and design concepts. The details of the basic and enhanced model are described in later sections of this document under their respective headings. Both paths followed the pattern of similar work elements. The elements followed were: the system requirements; the software requirements; the model design and implementation; the simulation runs, the results obtained, and analysis; and recommendations. Inputs to the system requirements were obtained from the system engineering and analysis efforts as well as the technical discussions. Many technical notes defining software requirements, model design, and implementation were written as aides for the software development personnel. Formal technical reports were prepared on specific problems and their resolutions to record the progress of the effort. Results obtained from the runs of the simulation model and

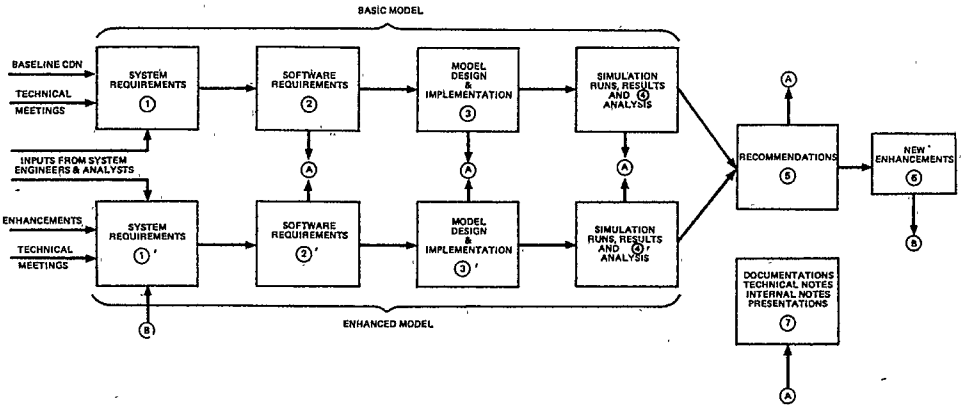

Figure 2. The CDN Simulation Model Work Breakdown Structure

recommendations based on the analysis of the results were also documented. Major recommendations contributing to the CDN system design are described later at greater length under the heading - simulation results and analysis. This effort also provided important inputs to the preliminary design report (PDR) on the cable plant engineering.

This paper discusses the basic operation of the CDN and the CDN simulation model. It also discusses the selected major results obtained through simulation model runs and their analysis in a section under that heading, which is followed by conclusions and recommendations.

CDN SIMULATION MODEL

CDN SYSTEM OPERATION

Before delving into the details of the model, it is essential to provide the following general background of the CDN system.

The entire CDN utilizes full-duplex, fiber-optic (FO) links that support a nominal data rate of 48 kilobits per second (kbps) for interconnections among nodes. The MX-Wing messages that are transmitted over the CDN consist of a nominal 1056 bits (66 sixteen-bit words) with 96 bits of header and 48 bits of trailer for the control and authentication information of each message (Figure 3). The first 48 bits of the header contain initialization data for the network cryptodevices, the second 48 bits of the header contain message type, destination address, duplicate suppression, direction/routing, and source authentication control information. The 48 bits of the trailer contain processing authentication data.

In general, at a CDN node each message is checked after decryption for valid source before it is subjected to further processing, if the node has seen the message before it is suppressed. Otherwise the message is routed on the network based on the information contained in direction/routing control, the input link over which the message was received, and message type. The message, if addressed to the node, is process- authenticated before it undergoes any further processing. Advanced Data Communications Control Procedures (ADCCP) link control has been adopted for data link control. This increases the message length by an additional 48 bits as seen in Figure 3. A subset of ADCCP data link control codes was chosen for link initialization, message framing, error recovery, flow control, and recovery from abnormal conditions.

The CDN messages were generally of two types; commands that could only be originated from the manned nodes (OCCs and ASCs) and responses that, in general, were elicited from the unmanned nodes (HSSs and end points of the clusters which could serve as concentrators, CONC) in response to specific commands. Two major types of

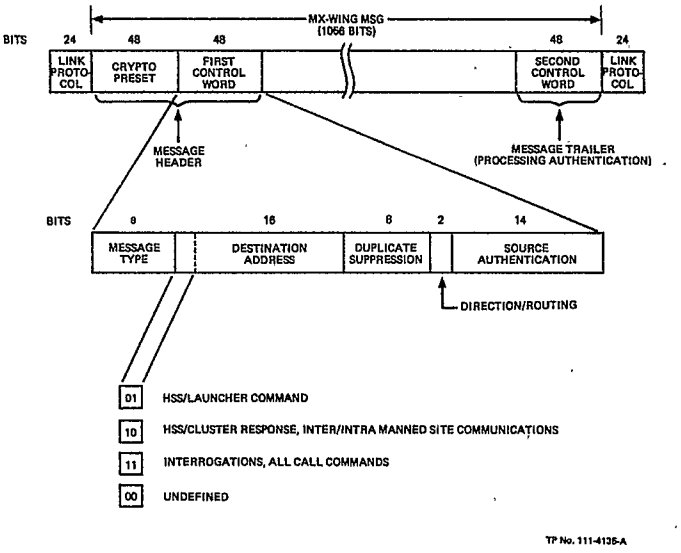

Figure 3. CDN Message Format

commands were allowed; an addressed (ADR) command, destined for a specific CDN node, and an all-call or broadcast (BC) command destined for every node in the network. A concept of flooding was used in command propagation; while a response generally traced back the path of the corresponding command. The primary goal of the preattack FO CDN was to periodically collect the operational and maintenance status of the HSSs. The CDN system specification allowed a total time of two (2) minutes for the collection and redistribution of this status data. This goal was accomplished by a suitable message set design that was utilized in the status collection and redistribution cycle. In the status cycle, a special class of BC command called status interrogation* was used. It could originate at a manned site and carry direction/routing information for its propagation on the inner or outer loop of the network in either the clockwise (CW) or counter-clockwise (CCW) direction. In the network, the interrogation is routed along the indicated loop on inter-cluster links until suppressed by the sequence number (duplicate suppression) check. At each cluster CONC, the interrogation enters the cluster and is retransmitted by the HSSs on intra-cluster links until it is suppressed by the CONC at the other end of the cluster. Each HSS, after retransmitting the interrogation, formats its own status message and transmits it in the direction from which the interrogation came.

*This interrogation was later modified to allow its propagation on any one loop only of the network in both the CW and CCW directions to satisfy the specification time for status collection and redistribution under facility (link and/or node) failure conditions.

The interrogation-entry cluster CONC waits a fixed time interval (a timeout) sufficient to receive status from all HSSs within that cluster. It then reformats the HSS status into cluster operational and maintenance status responses; and immediately transmits the cluster (operational) status response (CSR) to the source of interrogation in the direction from which the interrogation was received. The OCCs receive and process CSRs from the clusters and redistribute them to the other manned nodes. Two types of redistribution were considered in the message set design. In one type of redistribution, each OCC retransmitted all the CSRs (received from an ASC) to the remaining three (3) ASCs. In the other type of redistribution, each OCC retransmitted half of the CSRs to the remaining three (3) ASCs. The latter redistribution was called even/odd distribution because even numbered OCC (see Figure 1) retransmitted only the even numbered CSRs, while the odd numbered OCC retransmitted only the odd numbered CSRs. The status cycle consists of four subcycles, each corresponding to the CW or CCW direction with inner or outer loop propagation of the interrogation, or two subcycles for inner or outer loop propagation of the interrogation. Each subcycle in the four subcycle group and two subcycle group was triggered by status interrogation every 30 and 60 seconds, respectively.

Reaction time for a status subcycle is defined from the time of status interrogation origin to the receipt of the last possible cluster operational status response at the origin. Reaction time for a command is defined as the time between the origination of the command at the source and the command receipt at the intended destination.

Within the envisioned normal operation of the CDN communication, the steady state message traffic within the status cycle would be interspersed with other message traffic such as ADRs, BCs and radar data collection. In such a traffic mix, the reaction times, buffer sizes at a CDN node, and impact of selected flow-control protocols, are some of the major parameters or issues that cannot be determined analytically for a network of the size described. This is particularly so since the message traffic mix varies according to the environment and the demands on the network.

In the following, we present a brief overview of the CDN simulation model followed by a concise simulation model description. Finally, some major results are presented and analyzed with some of the derived major recommendations.

OVERVIEW OF THE CDN SIMULATION MODEL

The model is a next-event (event-driven) simulation in which the message propagation activity and the facility (nodes and/or links) failure conditions in the network are represented as message events and facility failure events, respectively. The model simulates the entire network and is capable of selecting a variable number of shelters or HSSs per cluster. Accurate message movement in the network is simulated and can be displayed on the screen for validation or analysis. The model allows the collection of the following additional data for the CDN performance evaluation:

1. Time history at selected nodes - for reaction time computations
2. Queue buildup and density reports at each node in input and output buffers - for buffer sizing at the CDN nodes
3. Processor and link resource utilization - for loading evaluations.

Inputs to the simulation are submitted in an interactive mode and utilize the nine screen displays in selecting various optional parameters for the simulated CDN environment:

1. Manned node connectivity matrix (which can be changed)
2. Network operational parameters (such as nodes per cluster), message length, and link protocol overhead (OH)
3. Site processing times by message type
4. Simulation control parameters (including priority in queues), error recovery - yes or no, flow control (that is finite or infinite buffer size) - yes or no, and type of status cycle used
5. Error recovery parameters including bit error rate (BER) on links and various related timeouts
6. Network loading in terms of various message types mix
7. Facility failure scenario - to indicate time and duration of failed nodes and links
8. Flow control parameters on buffer sizes-to initiate flow control messages
9. Listings of nodes where all data including message traffic activity and buffer buildup has to be collected for a reports program.

The simulation can be executed in either the interactive or batch mode. A separate utility program called the reports program can be utilized to output up to seven different reports at the user's option:

1. Selected node statistics
2. Time organized queue statistics
3. Node organized queue statistics
4. Processor (at a node) utilization
5. Link utilization
6. Queue density report (at selected nodes)
7. Time weighted queue density report (at selected nodes).

The simulation model, as noted earlier, has evolved in two phases - the basic model and the enhanced model. Both are described below.

BASIC CDN SIMULATION MODEL

The basic model incorporates the entire CDN and is capable of selecting a variable number of

shelters per cluster. A CDN node is modeled as a four-part entity; one input buffer, one processor, as many output buffers as the number of outgoing links, and outgoing links (Figure 4). All buffers are assumed infinite in size. A CDN link is incorporated in the model as a transmission delay which is based on the link data rate, the message length, and the link protocol overhead, including the effect of bit stuffing. Bit stuffing is required in the ADCCP message frames to ensure data transparency. Our investigation, supported by a small simulation, indicated that the average increase in the MX-wing message as a result of bit stuffing was 18. This effectively increased the MX-wing message frame length to 1056+48+18 = 1122 bits in the simulation model for reaction time computations. The model simulates all the necessary message propagation in the complete status collection cycle and command propagation for addressed (ADR) or broadcast (BC - all call) messages in accordance with the input scenario. The following types of status collection cycles are simulated in the model:

1. Status cycle with four subcycle operation (inner or outer loop and CW or CCW propagation of interrogation). Two methods of collecting the shelter (HSS) status at a concentrator (CONC) are included in this operation.
2. Status cycle with two subcycle operation (inner or outer loop propagation of interrogation).

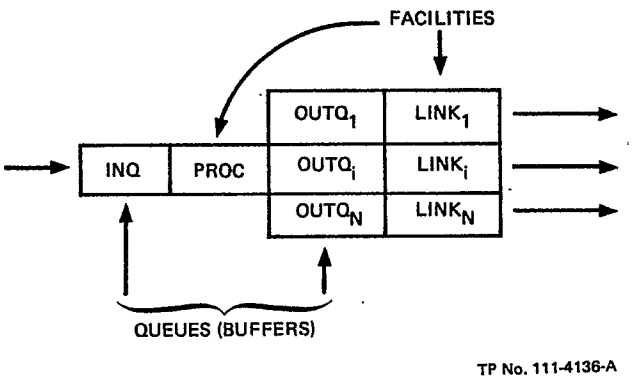

Figure 4. Model of a CDN Node

In the model, a type one interrogation (OSI_1) triggers the collection of type one (inter) cluster status response (CSR_1) containing HSS Operational Status Responses (OSR) (for up to 30 nodes) plus nine (9) HSS/Cluster Maintenance Facility (HSS/CMF) Maintenance Status Responses (MSR). The model also simulates the type two operational status interrogation (OSI_2) for the collection of type two and type three (inter) cluster status responses (CSR_2 and CSR_3) containing respectively the launcher status and HSS/CMF MSR data. The two remaining major features of the model are:

1. Type-dependent message processing at a CDN node. Two processing times are used, PT_1 is the time required to process a message for retransmission and PT_2 is the time required to perform additional processing functions at the node after message retransmission - such as assembling the status response.
2. Sequence number checking for duplicate suppresion. Applies to both commands and responses. Checking is performed by message type and source.

ENHANCED CDN SIMULATION MODEL

The enhanced model contains all the features of the basic model as well as the additional capabilities described in the following:

1. Simulation of the selected ADCCP link protocols for the following functions:
   a. Data transparency (in basic model)
   b. Error recovery
   c. Flow control
2. Buffer related topics:
   a. Finite buffer size
   b. Buffer thresholds for flow control messages
   c. Priority in buffers for message handling
3. Faster processing at manned nodes
4. Timeouts associated with events.

As explained in the previous section, data transparency is modeled in the simulation by the inclusion of an additional overhead that corresponds to the average number of stuff bits expected.

Error recovery implies retransmission of a message received in error. In the simulation, a message in error (on a link with a specified message error rate) is detected via a uniform random number draw at the receiver. When incorrect reception occurs, the I-frame transfer timeout will be utilized in the retransmission.

The model simulates the flow control that results from finite size buffers primarily by using receiver-ready (RR)/receiver-not-ready (RNR) control signals (S-frames of the ADCCP link protocol). We analyzed these flow control protocols, and of the three alternatives, we decided to implement the following for its simplicity. If an input buffer full condition is encountered at a CDN node, the node sends an RNR to all nodes to which it is connected followed by I-frames (if any to send exist) until an RR can be sent. Thresholds can be defined for the number of employed buffers by node type (OCC or ASC or CONC or HSS) at which the RNR and RR will be sent. If an output buffer full condition is encountered at a CDN node, no processing is allowed at the node processor other than acknowledgment/nonacknowledgment (ACK/NAK) of the incoming messages (thus allowing the node's input buffer to be filled) until there is space available in the affected output buffer. When space becomes available in the affected output buffer, the processing of messages is again enabled and an appropriate control signal (RR or RNR) is sent to the connected nodes based on the state of the node's input buffer. The model also allows message handling by priority (defined by message type) for certain message types.

The model can simulate faster processing at manned nodes to reflect the use of a miniprocessor rather than a microprocessor.

The model has three specific timeouts implemented, which are triggered by the

following events:

1. Timeout at an entry concentrator when an OSI_1 enters a cluster
2. Timeout associated with the exchange of I-frames
3. Timeouts associated with node and link failure conditions.

Each timeout can be controlled (by the user) in the simulation using an additional delta time so that the optimal value of each timeout can be determined using the simulation run results.

CDN SIMULATION RESULTS AND ANALYSIS

Simulation results related to network reaction times, buffering, and flow control are presented in this section. Major factors influencing the network reaction time results are:

1. Link protocol overhead reflected in transmission delay
2. PT_1, PT_2 variation in their fixed sum PT
3. Faster processing at manned nodes
4. Error recovery
5. Facility failures
6. Finite buffer size and flow control.

All simulation results are based upon the 1122-bit message (1056-bit message plus 66 bits for link protocol overhead and bit stuffing) and the 48 kbps data transmission rate on the links. Simulation results on network performance for simple input scenarios (such as status subcycle operation only) were validated by analysis. This provided a basis for predicting the network performance using simulation results in complex input scenarios (such as status subcycle operation under additional traffic loading and/or facility failure condition) which could not be analyzed primarily as a result of expected queues at many CDN nodes.

Infinite Buffer Size Results

The following results assumed infinite buffer sizes at the CDN nodes and consequently no flow control.

Figure 5 shows the reaction time for a simple input scenario of a status subcycle (in two subcycle operation of the status cycle) as a function of processing time PT (corresponding to unmanned nodes) with R and 1/s as parameters. R is the ratio of PT_1 to PT_2 and 1/s is the ratio of processing speed at the manned node to processing speed at unmanned node. Linear plots and breakpoints on them for this simple scenario were validated analytically. Breakpoints on the plots occur as input queues start to build up at the OCC. The figure clearly indicates the reduced status subcycle reaction time advantage of using a miniprocessor in place of a microprocessor at the OCC (compare 1/s=1 and 1/s=1.875 plots for same values of R).

Figure 6 shows the input queue buildup at the OCC during the simple scenario of a status subcycle as a function of time and this was validated by simple analysis. As shown in the figure, the approximate queue duration and maximum number in the queue can be analytically

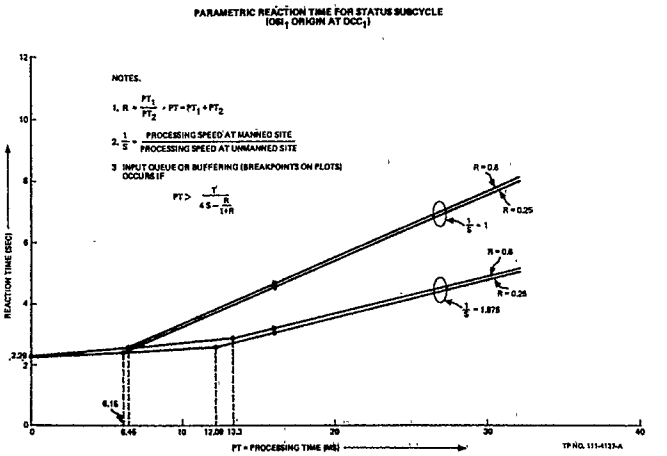

Figure 5. Parametric Status Subcycle Reaction Time

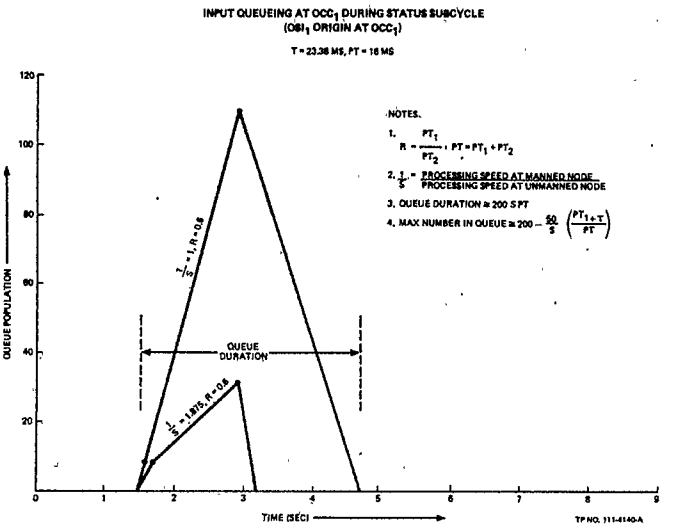

Figure 6. Input Queue Buildup

predicted. The maximum number occurs in the queue (approximately) when the last CSR_1s arrive at the OCC. A maximum queue of 110 CSR_1s develops at the OCC using a microprocessor. When this is replaced by a miniprocessor, the queue number is reduced to 31 reflecting faster service time.

For nominal values of PT_1 = 6 ms and PT_2 = 10 ms and with no flow control (or infinite buffers) and assuming no facility failures on the CDN and microprocessors at all nodes (1/s=1), the status subcycle reaction time is 4.68 seconds. With miniprocessors at the manned nodes (1/s=1.875) the status subcycle reaction time is reduced to 3.18 seconds. The reduction is primarily because of a smaller input queue buildup at the OCC as a result of faster processing speed at the manned node of OCC.

One example of a complex scenario which cannot be analyzed is additional traffic loading with the status subcycle. With an additional loading of all-call (BC) command injection at the expected traffic rate of 100 milliseconds and miniprocessors at the manned nodes, the status subcycle reaction time increases to 3.31 seconds

and 3.51 seconds, respectively, for injection at one OCC and at both OCCs. From the buffer count observations of the simulation runs it was determined that this increase in the status subcycle reaction time from 3.18 seconds was primarily as a result of the increased number of messages in the input queue at the OCC. Maximum input queueing of 45 and 66 is respectively observed at the OCC of OSI_1 origination; these queues contain, in addition to CSR_1s, the injected and retransmitted commands.

Another complex scenario, difficult to analyze, involves cases of facility failures with the status subcycle. The simulation results indicated that even with major facility failures*, meaning either the links interconnecting the manned nodes or manned nodes themselves, the observed status subcycle reaction time was well within the specification value. The worst observed status subcycle reaction time then was 7.31 seconds with both OCCs failed and an ASC taking over the role of the OSI_1 origination. The increase in the status subcycle reaction time was primarily due to the doubling of the links on which the OSI_1 had to travel to elicit OSRs.

Finite Buffer Size Results

The following limited results are with finite buffer sizes at the CDN nodes and with selected flow control protocols. Efforts on detailed results and refinements and recommendation on protocols were curtailed due to the presidential decision mentioned previously. By using a suitable facility failure scenario, a stressed network situation was created to observe if the linear cluster would break down for addressed (ADR) commands when flow control was invoked using finite buffer sizes at the CDN nodes. The observed results are summarized in Figures 7 and 8.

Figure 7 shows the V-shaped plot that indicates the network congestion occurring at about the center of cluster 24 wherein the addressed commands injected from both OCCs are destined for HSS number 12. As a result of the input and output buffer size of the one (1) used at each shelter (HSS) and the arrival of two (2) ADRs from adjacent HSSs, the input queue at (cluster 24, HSS 12) or (24,12) gets full and (24,12) sends RNRs to adjacent HSSs, (24,11) and (24,13), whose output queues in turn get full thereby forcing input queues at (24,11) and (24,13) to get full. This chain reaction continues to concentrators, (24,1) and (24,24). As a result, of the 25 addressed commands that enter the cluster from either concentrators, only seven make it to the center of the cluster, while the remaining are deadlocked in either the input or output queues at the intermediate HSSs with no message traffic movement in or out of the nodes.

Figure 8 shows the expected asymptotic behavior of the reaction time for ADR addressed to (24,12). In this example the slope of the plot is roughly the transmission time of one message (a message in the output queue to be

*A maximum of a combination of up to three link and/or node failure is permitted as stated in the CDN system specification.

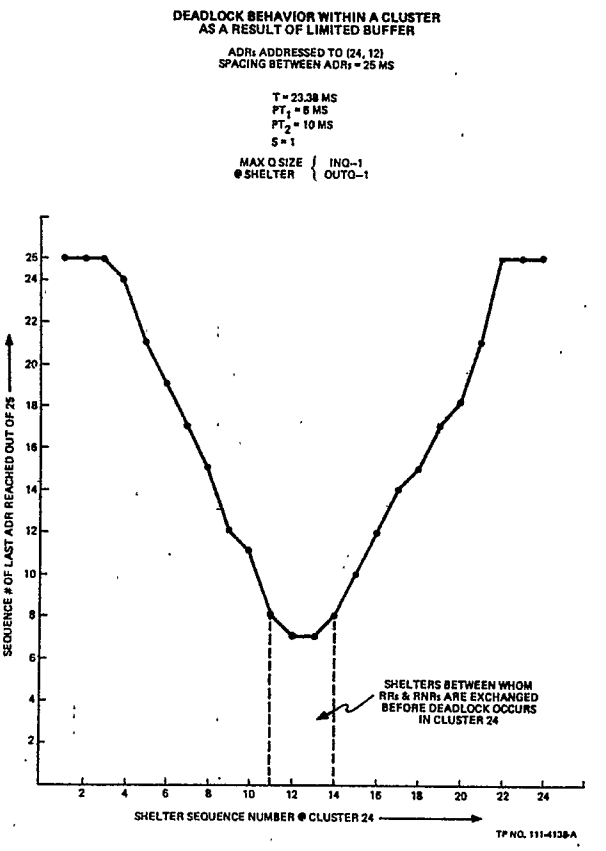

Figure 7. Network Congestion Within Linear Cluster

serviced by link server at this rate) until the asymptote is encountered.

These observed results indicate a potential congestion problem within linear clusters in the stressed network situation. This problem may be alleviated by increased buffer size at the HSSs.

CONCLUSIONS

The CDN simulation model provided a powerful tool for the CDN system design and development and rapid CDN system performance evaluation. Before any performance evaluation, each added capability to the validated basic simulation was verified with appropriate analytical checks. The following major contributions and recommendations resulted from this effort on the model development and use.

1. The CDN message sets and the routing went through many evolutions of design before arriving at optimal message set and routing strategies. For each design, the network performance analysis was supplemented by the results from the runs of the simulation model (called simulation results henceforth) on parameters including network reaction times and buffer (queue) buildup (see Figures 5 and 6).

2. Analytical studies supplemented by the simulation results show the feasibility of various alternatives. The simulation results aided the selection of the best alternatives as shown in the following examples:

a. Redistribution of operational cluster responses from the manned sites considered two alternative techniques; flooding and even/odd distribution. Simulation model results showing

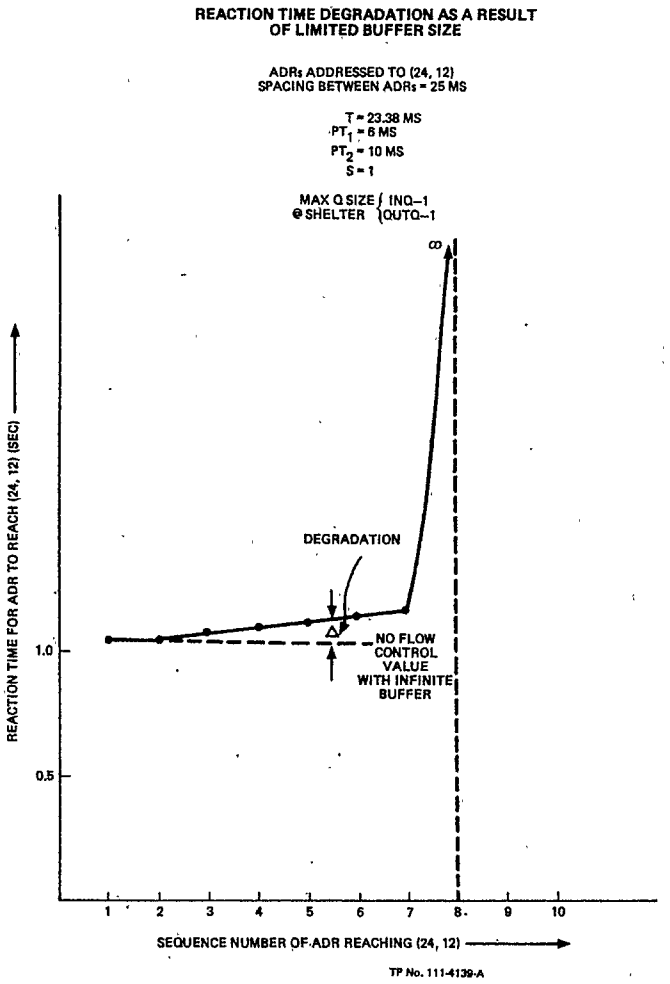

Figure 8. Reaction Time for ADR Showing Degradation with Finite Buffers

reduced input queueing with essentially the same status subcycle reaction time clearly demonstrated the superiority of the even/odd distribution technique.

b. Two alternatives of processing speeds were considered; one used (slower) microprocessors at all CDN nodes, the other used (faster) miniprocessors only at the manned nodes (OCCs and ASCs). The simulation results clearly demonstrated a superior network performance when faster miniprocessors were used at the manned nodes (see Figure 5 and 6).

c. For buffer sizing at a shelter (HSS) the simulation results in Figures 7 and 8 clearly demonstrate that a buffer size of one (1) each for input and output queues is insufficient for avoiding congestion in the cluster in a stressed CDN network; and resultant delays in message deliveries. A pool of buffers may be utilized at a CDN node, with two buffers as minimum provided for each output link to allow for a message to be held in the pool until its correct receipt at the next node is received by some means. The cost of buffers is another major consideration in seizing the buffers.

3. The following analytical studies, and protocol definitions and evaluation efforts were validated and/or improved by the analysis of the simulation results.

a. A technical report on Error Recovery was prepared that summarized analytical results that were supported by simulation observations.

b. A technical report on various timeouts was prepared that proposed the use of simulation to optimize the timeout values.

c. Flow control techniques and related protocol definition and development was supplemented by simulation results.

d. Efforts on bit stuffing required to ensure data transparency in ADCCP frames, helped to pinpoint the average number of bit stuffed to be used in the network performance computations.

ACKNOWLEDGMENT

The authors wish to acknowledge the dedicated support of the many engineers and programmers of CSC and GTE who contributed to the development of the CDN simulation model and software. Particular thanks are due to C. Albright and S. Golas of CSC, who were the principal designers of the simulation software.