Proceedings of the 1987 Winter Simulation Conference
A. Thesen, H. Grant, W. David Kelton (eds.)

Computer Simulation of Communications on the Space Station Data Management System

J. R. Agre, J. A. Clarke †, M. W. Atkinson, and I. H. Shahnawaz

Science Center, Rockwell International, Thousand Oaks, CA

Abstract

A discrete event simulation model for performance evaluation of various alternatives in the design of the communication system on the Data Management System (DMS) of the space station has been developed. DMS.SIM, the SIMSCRIPT-based model of DMS consists of two components: (I) The communication architecture model of multiple, interconnected, fiber-optic, local area networks (LANs) where the LAN access protocol is either token-bus or a version of CSMA/CD with deterministic collision resolution, and (II) the message workload generation module which allows the user to specify a realistic scenario for message generation and responses.

Several comparisons between the two access protocols are presented and a Space Station sample workload model constructed with this general representation is used as an example input model for simulation runs. The simulation program structure, the user interface for data input, and graphical display of output statistics from a simulation run are described.

1. INTRODUCTION

NASA is currently sponsoring a large effort to develop the designs for the new Space Station, a manned orbiting laboratory capable of supporting scientific, commercial, and defense applications and experiments. Associated with the Space Station itself may be space transportation systems, unmanned space platforms, satellites, other space vehicles, and ground systems. A collection of elements, called a payload, all of which interact to accomplish a given mission such as satellite recovery and repair or an astronomical experiment, must be supported and assisted by the Space Station facilities. The total communication and processing needs of the Space Station are handled by the Space Station Information System (SSIS) that consists of (1) interfacing the orbiting and ground-based communication elements into an end-to-end network for reliable delivery of command and control and user data, and (2) the Data Management System (DMS) which provides the hardware and software facilities for supporting the data processing and communication needs of the major subsystems onboard the Space Station. In addition, the DMS represents a common environment for interfacing the users and the operators with the operation and control of the Space Station systems. Further general information on the Space Station, SSIS, and DMS requirements can be found in [1,2]. An excellent review of the DMS Architecture and capabilities is presented in [3].

The DMS is envisioned as a distributed computing system consisting of a set of compatible computers or standard data processors (SDP), mass storage units (MSU), crew workstations called Multi-Purpose Application Consoles (MPAC), local area networks and their interfaces, and specialized equipment interfaces. All the software required to operate, monitor and control the Space Station, and all the services necessary to support data acquisition and distribution for the core subsystems and any payload experiments are included. There is a common network operating system for all SDPs and several databases must be supported. Some of these activities have real-time response requirements which must be handled by the DMS and others are mission-critical and of the highest priority [4].

In order to satisfy the various constraints of this environment effectively, the communication components within the DMS must accommodate the potentially high data rates of payload experiments as well as the response time needs of the real-time aspects of the station operations. These requirements are expected to be met by an architecture consisting of a mix of high-speed and lower-speed Local Area Networks (LANs). These networks will be interconnected using bridges, and the processors and workstations will connect to the networks via standard network interface units (NIUs) [5].

It is a DMS requirement that all communication protocols will follow the ISO OSI layered architecture concept [6], although specific protocols are not always specified. Within this framework for the DMS communication requirements, a myriad of possible architectures and protocols are candidates for actual implementation. It is extremely important to be able to assess the ability of a given design to meet the mission needs; computer-aided assessment tools are therefore necessary for accurate performance analysis. Simulation models have been successfully used to model the performance of many types of LAN protocols and architectures under a wide variety of conditions [7]. We feel that a computer simulation model offers the most flexible and accurate tool for examining competing communication architectures and protocols and comparing their performance. As a result, DMS.SIM, a discrete-event simulation of the communication system was developed and used to evaluate two fiber optic LAN protocols, a token-based logical ring and the Fiber Optic Data System (FODS) [8], a type of random access star-bus, under realistic message traffic conditions.

In Section 2, the structure of the DMS.SIM simulation program is outlined, the two LAN protocols under consideration are defined and the message workload model is described. In Section 3, the simulation program implementation is described and illustrated with a comparison of the performances of these two access protocols. The capability of the DMS.SIM to model explicit message traffic is shown by an example of a subsystem of DMS in Section 4. Lastly, some conclusions and further work are described in Section 5.

2. THE DMS COMMUNICATION ARCHITECTURE MODEL

Several models are available for performing high level anal-

---

† Rockwell Space Station Division, Downey, CA

ysis of the processing and communication system performance of the DMS. An analytical queueing model, called ADAM, that allows the inclusion of specific workload descriptions, has been employed to examine the performance of specific Space Station software, hardware, and communication architectures [9,10,11]. However, ADAM is a high level model and does not provide a detailed analysis of the communication system performance. Two simulation models specifically for DMS communication studies, called LANES II and LANES III [12,13], have been reported which examine the FODS and FDDI protocols, respectively. These models concentrate on the logical link and physical layer aspects of single LAN networks. However, these models do not support complex message traffic representations. To study the multiple network issues under realistic traffic scenarios, the DMS.SIM multi-Lan models were developed and implemented using SIMSCRIPT II.5, a discrete-event simulation language [14]. Separate, single LAN, detailed models of each protocol have been developed previously [15,16,17] and are used to verify the multi-LAN models.

The architecture model consists of a network of nodes, called stations, corresponding to the network interface units (NIUs) of the communicating components in the DMS. The stations are assumed to be connected to one or more LANs and the LANs are interconnected using special bridge stations. An NIU is assumed to employ the link-level protocol corresponding either to a Logical Ring or the FODS star-bus configuration, representing two of the candidate protocols of the DMS. Although there are desirable benefits in employing standard protocols, the DMS is not limited to considering only standard versions should special requirements not be satisfiable. Differing numbers of nodes may be attached to each LAN, but the LAN physical characteristics are constrained by the Space Station topology, e.g., the single or dual keel configurations and subsystem equipment locations.

Both the architectures considered and the model design adhere to a layered description. The link level protocol together with the physical parameters such as propagation delay and basic transmission rate, the network interconnection and the routing of the packets by the intermediate bridges, and the transport protocol layer that ensures end-to-end correct reception of messages, correspond to the first four layers of the OSI layered architecture model [6] as shown in Figure 1. These four layers constitute the 'communications' segment of the DMS.SIM. Message traffic offered to the communication segment by the session, presentation, and application (layers 5–7), are represented by a User layer using a 'message scheduling' model, which is described in detail in Section 2.3.

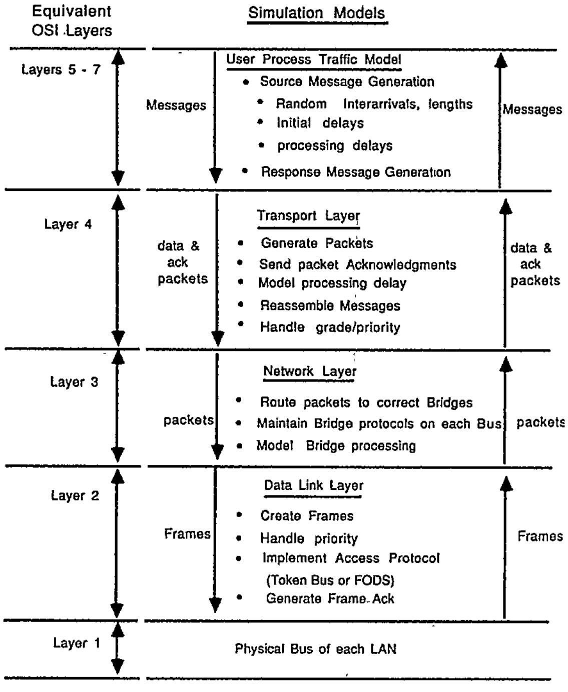


Figure 1: OSI Layer Representation

The general steps in message transmission are as follows: when a station on the LAN transmits a message to another, the message is handed by the sender station's User protocol layer to its (sender's) Transport protocol layer, which fragments the message into one or more smaller packets, adding to each a fixed 'overhead' bit string for identification and processing particular to this protocol layer. These packets are then passed to the sender's Link protocol layer where they are transformed into frames for transmission. In these models, there is no flow control and the Transport protocol layer sends all the packets to the Link layer disregarding the previous packets' status. Also, in this version of DMS.SIM, frame bit-errors, lost frames, time-outs, and duplicate transmissions of delayed but acknowledged frames are not included.

The receiving station's protocol layers remove from each arrived packet the 'overhead' bit string corresponding to this layer and pass the remaining data to the (receiver's) next protocol layer. At the transport level, an acknowlegement packet is formulated and transmitted back to the source. Packets from the same source and belonging to the same message are then regrouped and rejoined in their proper sequence to form the original message, which is then passed on to the receiving station's User protocol layer. If a message destination station is on a different LAN from the source station, special stations called bridges are employed which connect two LANs. The packet must be accepted on the input side of a bridge and then transmitted onto the other LAN. A bridge maintains the protocol on both of the LANs to which it is connected. The routing of a message is assumed to be static and is controlled by a user-specified routing table.

The communication topology and network operation are also user-specified. The program input allows a variable number of LANs, stations, bridges, and a routing matrix as well as a number of system parameters such as token size, bus transmission rates, packet lengths, buffer delays, and propagation times. Various statistics are collected by the simulation including message and packet delays, station queue sizes, and bus utilizations. The detailed NIU architecture and performance are being studied separately, but are considered as simple delays in DMS.SIM.

There are several variations to this procedure caused by message *grade of service* and *priority*. Three different message grades of service are included with the first (Grade 1) requiring a packet level acknowledgement and the most NIU processing. Grade 2 and Grade 3 require no acknowledgement and reduced processing requirements. Priority levels are user specified message attributes and are implemented by giving messages priority in the user-layer message queue, packets in the outgoing transport-layer packet queue, and frames in the link-layer outgoing queue.

2.1 THE LOGICAL RING, LINK-LEVEL MODEL

The first access scheme modeled is the IEEE 802.4 standard that defines a 10 megabits per second (Mb/s) token-bus protocol [18], although arbitrary transmission rates are allowed. This is a logical ring network configuration as opposed to a physical ring, such as the Fiber Digital Distributed Interface (FDDI), which is also being considered for the DMS. FDDI is 100 Mb/s optical fiber-based, token passing ring scheme [19,20], which is similar to the IEEE 802.5 token ring standard [21]. The performance of the logical ring (when operated at equal transmission rates) is expected to be quite similar to the token passing physical ring scheme. Several differences between the token bus and the FDDI include 1) FDDI propagation delay is the delay between stations, while in the token bus it is the total end-to-end cable propagation delay, 2) FDDI requires a NIU buffer delay of several bit-times, 3) numbers of priority queues and method of specification of the bandwidth allocation, and 4) FDDI employs a late count variable. Even with these differences, the basic performances of the two schemes in terms of message delay, utilization and throughput are expected to be close and thus, it is felt that the token bus is a reasonable approximation to FDDI.

The token-passing, logical ring data link layer ensures a measure of 'fairness' in transmission privileges to all the stations on the network through its access rules. Before being allowed to transmit, it requires that the station possess the *token* which is a short control frame that represents the right to transmit. Token holding timers (THTs) are explicitly provided in each NIU to control the time each station can retain the token. In the model under consideration, each NIU has two priority queues, the high priority or guaranteed bandwidth queue and the lower priority queue. Some high priority frames can be transmitted every time the NIU receives the token, while the lower priority can transmit only if there is time remaining on the THT.

In summary, the access method operates as follows: a user-specified, station dependent, target token rotation timer (TTRT) value is loaded into the station's token rotation timer (TRT) which begins counting down. When a station receives the token, the current value of the TRT is copied into the THT and the TRT is reset to the TTRT. The high priority queue is transmitted for the specified duration, and if there is time remaining on the THT, the lower priority queue is serviced. Frames being transmitted when the THT expires are allowed to complete.

An important improvement in the efficiency of the simulation was introduced to reduce execution time at the cost of a small approximation error. If no traffic is available for transmission, the token rotates from one station to the next, and this 'idle' condition leads to the creation of events which have no consequence and tends to slow the simulation considerably, especially if this condition is occurring on multiple LANs. This situation will occur during long quiescent periods in the workload model. In order to decrease the running time, we have eliminated the needless rotation of the token. The token is 'reactivated' when a packet appears in any of the queues of one of the stations connected to the ring, and the idle interval is converted to an equivalent number of rotations. This introduces a slight underestimate of delay in light loading.

2.2 THE FODS, LINK-LEVEL MODEL

The Fiber Optic Data System (FODS), a 32-node, 100 Mb/s, broadcast LAN based on a star topology, has been developed for space applications [8,22,23]. The protocol is being implemented as part of a larger NASA test bed effort [24,25]. At the physical layer, each node, through its NIU, is connected via a fiber to a central star coupler which mimics a broadcast bus by broadcasting whatever is received on any channel to all other channels. For the purposes of this paper, the star connection will be called the 'bus'. At the physical level, FODS is similar in function to the 802.3 Ethernet [26], and the NIUs are capable of performing carrier sense, collision detection, and generating jamming signals to insure collision detection at other NIUs.

The FODS access protocol is termed CSMA/CD/TS for Carrier Sense, Multiple Access with Collision Detection and Time-Slot collision resolution. The bus can be in one of two operating modes, *contention* and *time-slot*. In either mode, all NIUs constantly monitor the status of the bus. The bus is started in contention mode and switches to time-slot when a collision occurs. In contention mode, an NIU will first listen to the bus, and if it is busy (carrier sense) it will refrain from transmitting. When the bus has been sensed idle for at least a period of time called the *gap time*, the NIU will begin transmitting a data frame. After reception, the receiving NIU will generate a short acknowledgement frame (ACK) if the frame has no bit errors and a receive buffer was available. The ACK is transmitted before the expiration of the gap time, thereby assuring it collision free access to the bus (since all other NIUs are waiting for the gap time). The gap time is thus determined by the maximum propagation time and the time to sense an ACK.

If two or more NIUs sense the bus is idle and begin transmitting within the time necessary to detect the carrier, then a collision will occur and be detected by the transmitting stations. Upon detecting a collision, the NIU will cause a jamming noise burst on the bus which will alert all NIUs that a collision has occurred and that the bus is changing to time-slot mode. This time period, from sensing the carrier and jamming until all NIUs are alerted, is called the *collision detection interval*. In the time-slot mode, the NIUs are ordered and each is assigned a unique time-slot number in which it may transmit on the bus without contention. This effectively forces a logical polling of each of the NIUs allowing a single frame transmission. An NIU wishing to transmit will monitor the bus and count the number of idle periods of gap time length. When the count is equal to the assigned slot number, the NIU is permitted to transmit one frame. The time-slot durations are thus variable and are equal to the gap time for an idle station or the frame transmission plus the gap time. For the $k-th$ NIU, the delay until it is allowed to transmit during time-slot mode, $D(k)$, is

$$D(k) = D(k-1) + T_{k-1} + g$$

where T_{k-1} is equal to the transmission time of the frame from the $k-1$ station or zero if there were no waiting frames, and g is the gap time. After the slot of the last NIU has occurred, the LAN returns to the contention mode. However, any NIU with a pending frame will wait a random delay of at most the gap time before initiating a transmission.

This scheme allows the advantages of low delay in a random access mode during lightly loaded periods and the stable behavior of a polling algorithm during periods of heavy loading. FODS is similar in concept to the 50 Mb/s Hyperchannel protocol [27]. There are two primary differences between FODS and the Hyperchannel protocol 1) the Hyperchannel time-slot mode is initiated after every transmission, and 2) during time-slot mode, the Hyperchannel always initiates the sequence with the first NIU, but an option for round robin ordering can be enforced with wait flip flops. These rules cause the Hyperchannel to remain in time-slot mode until a completely idle cycle has occurred. It is expected that the FODS and Hyperchannel in round robin mode will have similar performance, except that FODS has a small advantage under peak load.

2.3 MESSAGE WORKLOAD MODEL

A message workload generation module has been developed for modeling the message traffic found on the DMS. The module will produce the traffic inputs for the architecture simulation model of the DMS to determine the network performance characteristics. The objectives of the module are to specify the communication requirements using the terminology of the DMS environment and to allow for a diverse collection of message types, parameter distributions, and operational scenarios. The traffic model loads the network's stations with the various messages or signals that are transmitted during the course of a set of operation runs. The communications traffic is created using three primary structures: *operations*, *modes* and *message transactions*. Figure 2 gives a representation of the workload modeling process flow and where the input parameters are employed.

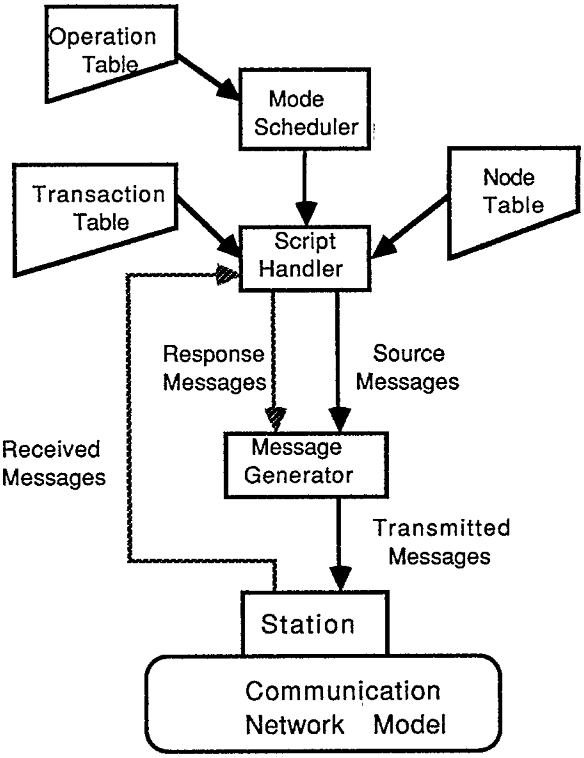


Figure 2: Workload Model Process Flow

An *operation* represents a high level task or job such as 'docking maneuver' or 'system check-out'. Each operation is associated with a scheduled start-time and a set of *modes*, where each mode represents a particular 'state' or 'milestone' of the associated operation and lasts for a period of time called the mode duration, e.g., start-up, normal, and completion modes. During a mode, certain messages may be transmitted to and from certain stations over the network.

Each mode causes the execution of a *message transaction* that is a list of *message descriptors*, containing all the messages to be generated during a particular mode. The message transaction is thought of as a script since it describes when, how often, how many, and where messages are generated. Several modes may employ a single transaction or each mode may have its own unique script. The Message Descriptor details the message contents, the time interval between successive messages, the message length, message source station, destination station, message type (i.e., whether the message has been initiated by the source station, or is a response to a previously received message), or whether a response message is desired from the destination, the execution time for the message, and finally the total number of such messages in case this message is a finite source.

The mode changes drive the simulation from the user layer. At each mode change, all message descriptors in the transaction of the current mode are used to generate the appropriate messages. Message sources of the previous mode are terminated. New sources are started and these messages are given to the source station queues for transmission. When a station has received a message from the network, the script handler examines the transaction data to determine the response. If a response is required, the descriptor of the response is used to generate the response message, which is given to the station and then transmitted. Before the response is generated, a processing delay is introduced. This represents the time needed by the user layer protocol process to formulate a response message.

There are several options in the message descriptor which support the modeling of stochastic traffic generators. The message attributes of initial delay, intergeneration time, length, and processing delay are specifiable as distributions and their parameters. Currently, the program supports the exponential, constant, and uniform distributions. In addition, special purpose message destination key words allow messages to be generated to random destination stations on the entire network (RAND) or to a specific LAN, (BUSi).

Using this general workload model it is possible to examine both steady state and transient network responses. An example of a steady state analysis is given in the next section and a transient analysis is given in an example Space Station workload in Section 4.

3. DMS.SIM SIMULATION MODEL FEATURES

The simulation model has been built with a front-end, user interface for data input and graphical output utilizing an IBM-PC/AT while the simulation is executed on either the IBM-PC/AT or a DEC VAX 11/780 for larger models. Since SIMSCRIPT is basically portable between the two machines, there is one version of the simulation software. Commercially available software packages were successfully integrated with the simulation for the user input interface and the graphical output, which resulted in reduced total model development time and allowed concentration on the simulation design.

3.1 USER INTERFACE FEATURES

The input portion of the simulation program is based upon a functional structure written in dBASE III PLUS [28] utilizing an IBM-PC/AT. It provides the user with an interactive, menu-driven input session and also maintains the input database. An input model is specified by means of several database files which collectively define the model, and may be named by the user when they are created. The user supplied model name then serves as a sub-directory which contains the member files constituting the model. When the user selects executes the simulation, the selected model files are converted (in background mode) into a SIMSCRIPT readable ASCII sequential file. The current active model is delivered to the simulation program and also becomes the default to the graphics program. The program will enter the PC SIMSCRIPT environment, SIMLAB, and execute the simulation there. In order to run the simulation on the VAX, the option 'Transfer Machines' has been provided to transfer the generated data file to the VAX and then run the simulation program. The simulation execution will produce several graphics output data files. If executing on the VAX, the graphics files must then be transferred back to the IBM-PC.

The user can select the option 'Graphics Output' from the main menu to graphically display statistics generated from the simulation on the active model. The graphics files of response times delays, bus utilizations, and message arrival times are created automatically by the simulation. The graphs are displayed using a commercially available, menu-driven graphics package called Energraphics [29]. This package allows the user to read the files containing the desired simulation output data and by means of various options edit, customize, and prepare it for graphical display. The user specified options are stored in a 'template' file which can be summoned for use on the graphical input files, enabling their preparation for graphical display. A user reads in the template first and then reads a new data file and edits graphics attributes such as scale(s), axis labels, and title to make a new graphics file. While it is possible to create entire standard graphs, allowing the user to customize the graphs through the use of the package was found to be preferable.

An input model is defined by a group of eight tables (data base files): *node*, *operation*, *message description*, *bus*, *bridge*, *route*, *network parameters*, and *delay histogram* tables. The Node Table is used to define the names of each station and to assign a station to a particular LAN. The Bus Table defines properties of each LAN such as number of stations, bridges, propagation times, transmission rates and other data particular to the access method such as gap times or TTRT values. In the Bridge Table, an entry for each bridge describes the LANs which it interconnects (and TTRT values for ring LANs). Specific bridge assignments for routing messages between LANs are given in the Routing Table which provides a static allocation of the bridge which is the next-hop destination. Operations and their modes are scheduled in the Operation Table which assigns the associated script and mode duration times. The message sources are detailed in the Message Transaction Table which contains the necessary descriptor data described earlier. In this table, messages are given names and the source and destination addresses are node names which must match the entries in the Node table. A table of network specific parameters describing an assortment of characteristics such as token length, protocol overhead lengths, ACK lengths, bridge and NIU delays is input. Lastly, a table describing the desired histogram statistics is input. Histograms can be collected for message delays for tagged messages, message priority, and message grade-of-service.

3.2 Sample Inputs and Access Protocol Comparison

An example system with three LANs, thirty nodes (ten per LAN), and two bridges using the FODS protocol has been analyzed. Messages arrive at each station with an exponential interarrival time, a constant message length of 1000 bytes, and a random destination. The transmission rate is 100 Mb/s, the maximum packet length is 2100 bytes, and the propogation time, gap time and collision detection interval are .27, 1.2, and .56 microseconds (μs), respectively. Bridge 1 connects Bus 2 to Bus 1 and Bridge 2 connects Bus 2 to Bus 3. Lastly, some other network parameters are the number of overhead and ACK bytes is fifty, and the link-level processing delay is 4.42 μs.

This example was used as a basis to examine some of the relative performance measures between the logical ring and the FODS models. Similar data was input for the logical ring and several simulation runs were executed. The results comparing average message delay are displayed in Figure 3 and average acknowledged message delay are given in Figure 4. The average message delay is defined as the time from message creation until the last packet of the message is successfully received. The average acknowledged message delay is defined as the message delay plus the time to receive the acknowledge for the last packet.

Two logical ring models were considered, one with a long TTRT set to 10000 μs and the second with a short TTRT set to 100 μs, labeled in Figures 3-4 as **T-10K** and **T-100**, respectively. In the comparison, for each loading value, the above three models were run until the delay values stabilized. The loading of the network is given in terms of the number of arriving message data bits per second at a station. This results in a loading of Bus 2 that is approximately 40% greater than

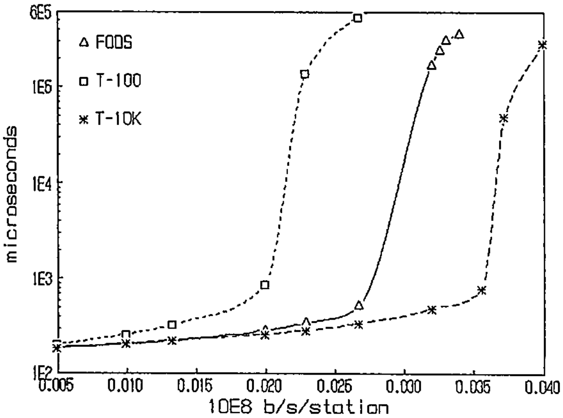


Figure 3: Average Message Delay

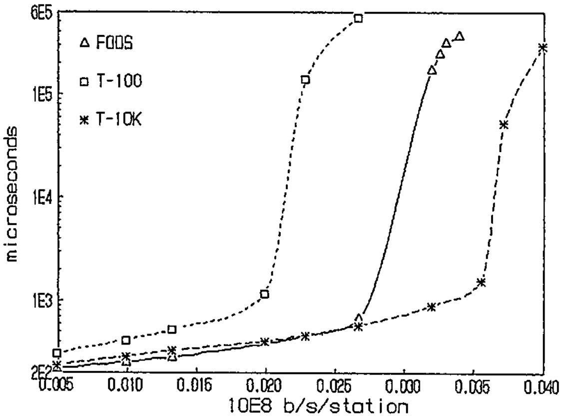


Figure 4: Average Acknowledged Message Delay

that of either Bus 1 or Bus 3. For example, the simulation run at 10^6 bits/second/station offers roughly a 17% load on Bus 1 and Bus 3 while Bus 2 must handle a 23% loading. Some of the effects of Bridge congestion are thus included.

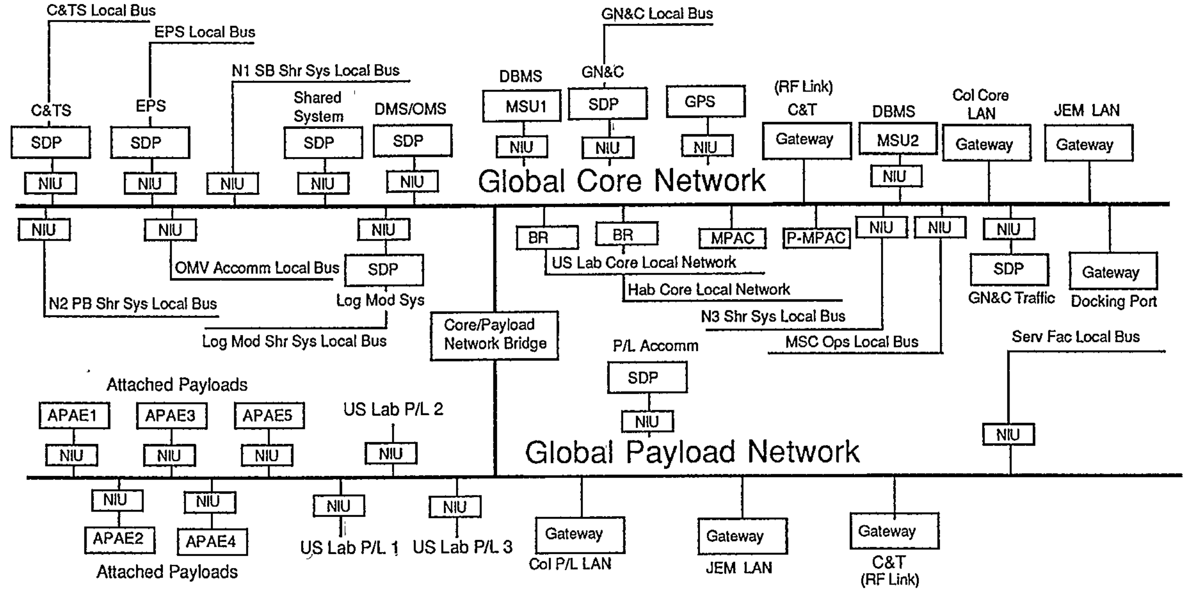
Examining the Figures 6-7, we notice that FODS and token bus with long TTRT (T-10K) systems yield similar delay characteristics under light and medium loads, but T-10K is more stable at higher loadings. The FODS scheme experiences significant collisions at the utilization levels above 40%. Due to the token bus model approximation, the delays are underestimated and the FODS is slightly superior, as expected. When the acknowledged message delay is considered, a dependency between a received message and the acknowledgement packet is introduced. FODS tends to allow a quicker response while the token bus tends to experience additional token revolutions before dispatching the ACK. The effects of the TTRT value are dramatic and underscore the importance of proper TTRT allocation. Networks with larger numbers of stations would tend to favor FODS in the lighter loading ranges since, in the token bus, the token revolution time is a function of the number of nodes. Both systems would be negatively effected by increasing the propagatation time value.

4. DMS Traffic Model

An example traffic workload model for the DMS was developed and implemented in the DMS.SIM format based on available NASA documents and other public domain information given in the references. The example assumes certain configurations and operational scenarios and as such it should be viewed as one of a battery of DMS workloads that would be used to assess a particular architecture. In particular, the workload will serve to illustrate the flexibility of the model in examining transient message traffic that is associated with Space Station events, such as a computer failure or a docking maneuver.

4.1 Space Station DMS Architecture

The backbone of the DMS architecture consists of two LANs called the Global Core and Global Payload Networks as shown in Figure 5. The Core Network is used for data traffic from the Space Station core systems (e.g., Guidance Navigation and Control (GN&C), Electrical Power System (EPS)) while the Payload Network is dedicated to data traffic from the onboard payloads. These were separated to prevent possibly erroneous payload data from corrupting one of the core systems. The global networks are intended to span the entire Space Station and interconnect the LANs which are confined to specific regions or modules. Bridge interface units are employed to interconnect the global and local networks using common protocols, and gateway nodes are used to connect networks with different protocols. DMS processing elements and customer payloads can connect to either global or local networks through an NIU. There are other special purpose local buses which transport data from the sensors/effectors and the controlling SDP and may also attach to a global or local bus through an NIU. This Space Station configuration (see



Abbreviation Definitions

C&TS	Communication & Tracking System	Log Mod Shr Sys	Logistics Module Shared Systems
EPS	Electrical Power System	US Lab	US Laboratory Module
N1 SB Shr Sys	Node 1&Starboard Beam Shared Systems	Hab	Habitation Module
DMS	Data Management System	N3 Shr Sys	Node 3 Shared Systems
OMS	Operation Management System	MSC Ops	Mobile Servicing Center Operations
DBMS	Database Management System	GN&C Traffic	GN&C Traffic Management
GN&C	Guidance Navigation&Control	Docking Port	Space Shuttle Docking Port
GPS	Global Positioning System	APAE	Attached Payload Accommodations Equipment
Col	Columbus Module (ESA)	P/L	Payload
JEM	Japanese Experiment Module	LAN	Local Area Network
N2 PB Shr Sys	Node 2&Port Beam Shared Systems	P/L Accomm	Payload Accommodations
OMV Accomm	Orbiting Maneuvering Vehicle Accommodations	Serv Fac	Servicing Facility

Figure 5: Example Space Station DMS Architecture

Fig. 5) contains two global networks (Core and Payload), five local networks (Habitation Module Core, US Laboratory Module Core, Columbus Module Payload, Columbus Module Core and Japanese Payload), and nine local buses (GN&C, Shared System, EPS, etc.) [5,30].

One goal of a DMS architecture is to isolate the specific data traffic associated with a particular region to the local bus. In a 'good' partition of the system, most of the DMS traffic will use the local buses and most of the local traffic will consist of periodic, small messages from high frequency control loops. The traffic on the global networks tends to be aperiodic, larger messages from file transfers or other data management functions. This is only a general rule as there may be some cases of control loops across the global networks and files moving across the local networks. The DMS will provide four priorities for messages which are classified in order of increasing priority as: background, normal, isochronous, and emergency [4]. These priorities are nonpreemptive and determine the order of service in message, packet, and frame queues of the NIU. This mix of periodic, aperiodic, and prioritized traffic creates a complicated situation, the analysis of which is aided by the simulation model.

4.2 Example DMS Architecture Analysis

The DMS.SIM model developed for this paper focusses on the Global Core and Payload Networks since these global networks are the backbone of the DMS and their performance is critical in the overall operations. In addition, they will contain an interesting traffic mix of the types described above. A future goal is to include all the local networks and buses. In DMS.SIM each global network is modeled as a LAN and the Core and Payload interfaces are modeled as stations on the LANs. From the perspective of the global networks, each of the local networks and buses can be simply viewed as either message sources or sinks coming to or from a single stations. The Core network (LAN_1) is configured with 23 stations which represent 7 system SDPs, 6 Local Buses, 2 MSUs, 2 MPACs, 4 Gateways and 2 Bridges. The Payload network comprises 13 stations, including 5 Attached Payload Accommodation Equipment Systems (clusters of payloads), 5 US Laboratory payloads, Columbus and Japanese Modules, a Payload Accommodation SDP, a Gateway, and a Local Bus. The one interconnecting bridge station, called the Core/Payload Bridge, is also attached to both of the LANs. For this hardware suite, the resulting Node Table that was input to DMS.SIM is shown in Table 1.

A data traffic scenario was developed which consists of a mix of two types of traffic: steady state and event messages. The steady state traffic reflects the typical mix of synchronous and asynchronous traffic such as control loops, health and monitor functions, computer data transfers, and telemetry downlinked to ground, while the event traffic consists of messages that arrive infrequently (with respect to the steady state) and are associated with events such as a docking maneuver or a Space Station reboost operation. The specific numbers used to represent the message traffic are given in the Message Transaction Table in Table 2.

The steady state traffic represents a typical operations load derived from references [2,4,5,31,32] and is a mix of normal and background priority messages. The normal priority messages are non-critical messages that with non-strict delay requirements (e.g., system status report) and the background priority are those generated by a background process with no specified delay requirement (e.g., electronic mail, file backup). For the Core Network, there is a mix of 503 normal and 60 background messages/second transmitted by the 23 stations with an average message size of 1024 bytes [32] resulting in a total data loading of 4.6 Mb/s (without the DMS protocol overhead). The maximum frequency results from the Shared System SDP which outputs 75 messages/second. On the Payload network, the steady state traffic is estimated at 735 normal messages per second and no background messages. Since 690 of the 735 messages are downlink telemetry from the payloads, their average message length is assumed to be 1100 bytes, the data size of the standard CCSDS (Consultative Committee for Space Data Systems) telemetry data packet. The remaining 45 messages have an average length of 1024 bytes. These result in a 7.6 Mb/s data traffic load on the Payload Network. The maximum traffic load comes from the Columbus Module which outputs 244 messages/second of downlink telemetry.

NODE TABLE

Node type 0 => simple nodes

node id	name	bus
1	C&T_SDP	1
2	EPS_SDP	1
3	NODE1_STARBOAR	1
4	SHARED_SYS_SDP	1
5	DMS/OMS_SDP	1
6	MSU1	1
7	GN&C_SDP	1
8	TCS_SDP	1
9	C&T_GW	1
10	NODE2_SHARED_S	1
11	OMV_ACCOM_BIA	1
12	LOG_MOD_SYS_SD	1
13	LAB_CORE_NET_B	1
14	HAB_CORE_NET_B	1
15	NODE3_MPAC	1
16	NODE3_P-MPAC	1
17	NODE3_SHARED_S	1
18	MSC_OPS_BIA	1
19	GN&C_TRAFFIC_S	1
20	DOCK_PORT_GW	1
21	MSU2	1
22	ESA_MOD_GW	1
23	JEM_MOD_GW	1
24	APAE1	2
25	APAE2	2
26	APAE3	2
27	APAE4	2
28	APAE5	2
29	US_LAB_P/L1	2
30	US_LAB_P/L2	2
31	US_LAB_P/L3	2
32	ESA_P/L1	2
33	JEM_P/L1	2
34	P/L_ACCOM_SDP	2
35	P/L_GW	2
36	SERV_FAC	2

Table 1: DMS Example - Node Table

The event traffic scenario was chosen to depict a drastic transient increase in the DMS traffic. Two message types will be represented, isochronous and emergency, where an isochronous message has a strict delay requirement (i.e., control loop or other time-critical traffic) and an emergency represents a Space Station crisis. The isochronous event is a control loop process and assumes that a crewmember is using an MPAC joystick to control a robotic arm connected to the Servicing Facility. The MPAC is connected via an NIU to the Core LAN while the Servicing Facility is connected via the Servicing Facility Local Bus to an NIU on the Payload LAN. These are isochronous since a low delay is required, and they are transmitted on the Core network, across the Core/Payload Bridge and onto the Payload LAN, and vice versa. The robot arm control loop sampling rate is 60 Hz and message size is 100 bytes, resulting in a .096 Mb/s data traffic load per LAN (messages 15 and 36 in Table 2c).

The second event message represents an emergency and assumes that a Payload Accommodations SDP on the Payload network has crashed during a critical experiment and must be rebooted and reloaded with the application software. The necessary files are stored in an MSU on the Core network and must be transferred over the Core Network to the Core/Payload Bridge, over the Payload network to the dead SDP. A file transfer of about 3.4 megabytes (application software plus operating system parameters) must occur within 30 seconds. This event causes high priority messages to be generated at the rate of 111 messages per second, each equal to the packet size of 1024 bytes (assumed MSU block size), placing a total load of .91 Mb/s on each LAN (message 37 in Table 2).

The purpose of this scenario is to observe whether or not the emergency message delay requirement can be realized and to assess the effects on the delay distributions of the four priority classes. The specific characteristics of the messages used in the simulation such as source, message size, intergeneration time, and destinations are given in Table 2. Most of the steady state traffic is modeled with exponential interarrival times, with rates equal to those described above, to allow for random effects. These messages are given random destinations on the Core (BUS1) or the Payload (BUS2) buses. The event messages are assigned to specific source-destination pairs as they are of particular interest. The two scenarios are scheduled using two operations, the first for the steady state traffic with the control loop and the second to start the emergency messages. In this example, the steady state traffic and the emergency file transfer messages both consist of a single mode which begins at time 0 and lasts for 10 seconds, the duration of the simulation. For comparison purposes, a separate run without the emergency messages (single operation) was also obtained.

4.3 Simulation Results for the DMS Model

The simulation model was run for two cases of the Core and Payload Networks using the FODS protocol at 100 Mb/s (Case I) and 20 Mb/s (Case II). Both cases include significant NIU processing delays (800 μs) and bridge delay (400 μs) representing estimated link layer function in order to observe its effect on message delay. These results are summarized in Table 3 (Case I) and Table 4 (Case II). Each case was run for two different workloads consisting of the steady state and control loop traffic with the Emergency messages (Workload I) and without the Emergency messages (Workload II). The tables include the mean, 95-th percentile, and standard deviation of the message delay for eight classes of messages. The eight message types are associated for statistics collection by using the tag fields of Table 2. Each message priority class was tagged and within each class, messages were tagged to differentiate between sources (e.g., Background from MSU1) and the buses on which they were sent (i.e., Core, Payload, Both). Tables 3-4 also list the utilization of the networks.

NOTE : msg = message   scr = message script (action list no.)

sid = message script id. no.
spt = message script pointer address
p = mode (p = 1=> start)
t = message type
= 1=> message from source
= 2=> response from dest
dest = destination (node)
srce = source (node)
grd = message grade
pri = message priority

i = message initial delay duration expressed as a distribution
= 0=> infinitely generating source
= 1=> constant generation with frequency=(1/parm1), parm2 = 0
= 2=> Uniform distribution with parameters parm1, parm2
= 3=> Exponential distribution with parameters parm1, parm2
g = message inter-generation time duration expressed as a distribution (see 'i', g = 1,2,3)
l = message length expressed as a distribution (see 'i',l =1,2,3)
d = message processing delay expressed as a distribution (see 'd',l =1,2,3)

scr	spt	sid	p	t	msg. name	ref.	srce. node	dest. node	tag	grd	pri	i	parm1	parm2	g	parm1	parm2	l	parm1	parm2	d	parm1	parm2
1	455696	1	1	1	MSG1	A	C&T_SDP	BUS1	1	1	2	1	0.	0.	3	0.042	0.	2	1.0	2048.0	1	0.	0.
2	455808	1	1	1	MSG2	A	EPS_SDP	BUS1	1	1	2	1	0.	0.	3	0.040	0.	2	1.0	2048.0	1	0.	0.
3	455920	1	1	1	MSG3	A	NODE1_ST	BUS1	1	1	2	1	0.	0.	3	0.100	0.	2	1.0	2048.0	1	0.	0.
4	456032	1	1	1	MSG4	A	SHARED_S	BUS1	1	1	2	1	0.	0.	3	0.013	0.	2	1.0	2048.0	1	0.	0.
5	456208	1	1	1	MSG5	A	DMS/OMS_	BUS1	1	1	2	1	0.	0.	3	0.014	0.	2	1.0	2048.0	1	0.	0.
6	456320	1	1	1	MSG6	A	MSU1	BUS1	1	1	2	1	0.	0.	3	0.033	0.	2	1.0	2048.0	1	0.	0.
7	456432	1	1	1	MSG7	A	GN&C_SDP	BUS1	1	1	2	1	0.	0.	3	0.022	0.	2	1.0	2048.0	1	0.	0.
8	456544	1	1	1	MSG8	A	TCS_SDP	BUS1	1	1	2	1	0.	0.	3	0.040	0.	2	1.0	2048.0	1	0.	0.
9	456720	1	1	1	MSG9	A	C&T_GW	BUS1	1	1	2	1	0.	0.	3	0.042	0.	2	1.0	2048.0	1	0.	0.
10	456832	1	1	1	MSG10	A	NODE2_SH	BUS1	1	1	2	1	0.	0.	3	0.100	0.	2	1.0	2048.0	1	0.	0.
11	456944	1	1	1	MSG11	A	OMV_ACCO	BUS1	1	1	2	1	0.	0.	3	1.000	0.	2	1.0	2048.0	1	0.	0.
12	457056	1	1	1	MSG12	A	LOG_MOD_	BUS1	1	1	2	1	0.	0.	3	0.100	0.	2	1.0	2048.0	1	0.	0.
13	457232	1	1	1	MSG13	A	LAB_CORE	BUS1	1	1	2	1	0.	0.	3	0.067	0.	2	1.0	2048.0	1	0.	0.
14	457344	1	1	1	MSG14	A	HAB_CORE	BUS1	1	1	2	1	0.	0.	3	0.067	0.	2	1.0	2048.0	1	0.	0.
15	457456	1	1	1	MSG15	A	NODE3_MP	BUS2	3	1	3	1	0.	0.	1	0.017	0.	1	100.0	0.	1	0.	0.
16	457568	1	1	1	MSG16	A	NODE3_P-	BUS1	1	1	2	1	0.	0.	3	0.167	0.	2	1.0	2048.0	1	0.	0.
17	458256	1	1	1	MSG17	A	NODE3_SH	BUS1	1	1	2	1	0.	0.	3	0.100	0.	2	1.0	2048.0	1	0.	0.
18	458368	1	1	1	MSG18	A	MSC_OPS_	BUS1	1	1	2	1	0.	0.	3	1.000	0.	2	1.0	2048.0	1	0.	0.
19	458480	1	1	1	MSG19	A	GN&C_TRA	BUS1	1	1	2	1	0.	0.	3	0.048	0.	2	1.0	2048.0	1	0.	0.
20	458592	1	1	1	MSG20	A	DOCK_POR	BUS1	1	1	2	1	0.	0.	3	1.000	0.	2	1.0	2048.0	1	0.	0.
21	458768	1	1	1	MSG21	A	MSU2	BUS1	1	1	2	1	0.	0.	3	0.033	0.	2	1.0	2048.0	1	0.	0.
22	458880	1	1	1	MSG22	A	ESA_MOD_	BUS1	1	1	2	1	0.	0.	3	0.067	0.	2	1.0	2048.0	1	0.	0.
23	458992	1	1	1	MSG23	A	JEM_MOD_	BUS1	1	1	2	1	0.	0.	3	0.067	0.	2	1.0	2048.0	1	0.	0.
24	459104	1	1	1	MSG24	A	APAE1	P/L_GW	2	1	2	1	0.	0.	3	1.0000	0.	1	1100.0	0.	1	0.	0.
25	459792	1	1	1	MSG25	A	APAE2	P/L_GW	2	1	2	1	0.	0.	3	0.0125	0.	1	1100.0	0.	1	0.	0.
26	459904	1	1	1	MSG26	A	APAE3	P/L_GW	2	1	2	1	0.	0.	3	1.0000	0.	1	1100.0	0.	1	0.	0.
27	460016	1	1	1	MSG27	A	APAE4	P/L_GW	2	1	2	1	0.	0.	3	1.0000	0.	1	1100.0	0.	1	0.	0.
28	460128	1	1	1	MSG28	A	APAE5	P/L_GW	2	1	2	1	0.	0.	3	0.0046	0.	1	1100.0	0.	1	0.	0.
29	460304	1	1	1	MSG29	A	US_LAB_P	P/L_GW	2	1	2	1	0.	0.	3	1.0000	0.	1	1100.0	0.	1	0.	0.
30	460416	1	1	1	MSG30	A	US_LAB_P	P/L_GW	2	1	2	1	0.	0.	3	0.0670	0.	1	1100.0	0.	1	0.	0.
31	460528	1	1	1	MSG31	A	US_LAB_P	P/L_GW	2	1	2	1	0.	0.	3	0.5000	0.	1	1100.0	0.	1	0.	0.
32	460640	1	1	1	MSG32	A	ESA_P/L1	P/L_GW	2	1	2	1	0.	0.	3	0.0040	0.	1	1100.0	0.	1	0.	0.
33	460816	1	1	1	MSG33	A	JEM_P/L1	P/L_GW	2	1	2	1	0.	0.	3	0.0078	0.	1	1100.0	0.	1	0.	0.
34	460928	1	1	1	MSG34	A	P/L_ACCO	BUS1	4	1	2	1	0.	0.	3	0.0400	0.	2	1.0	2048.0	1	0.	0.
35	461040	1	1	1	MSG35	A	MSU1	BUS1	5	1	1	1	0.	0.	3	0.0330	0.	2	1.0	2048.0	1	0.	0.
36	461152	1	1	1	MSG36	A	SERV_FAC	BUS1	6	1	3	1	0.	0.	1	0.0170	0.	1	100.0	0.	1	0.	0.
37	461328	2	1	1	MSG37	A	MSU1	P/L_ACCO	7	1	4	1	0.	0.	3	0.0090	0.	1	1024.0	0.	1	0.	0.
38	461440	1	1	1	MSG38	A	MSU2	BUS1	8	1	1	1	0.	0.	3	0.0330	0.	2	1.0	2048.0	1	0.	0.
39	461552	1	1	1	MSG39	A	P/L_GW	BUS2	2	1	2	1	0.	0.	3	0.1000	0.	2	1.0	2048.0	1	0.	0.
40	461664	1	1	1	MSG40	A	P/L_ACCO	BUS2	2	1	2	1	0.	0.	3	0.1000	0.	2	1.0	2048.0	1	0.	0.

Table 2: DMS Example - Message Transaction Tables

Table 3 indicates that the addition of the emergency high priority file transfer to the workload has a neglible effect on the delays of other messages sent across the DMS. All the delays shown for both workloads are well within the DMS requirements. The greatest effects were on the Background from MSU1 which must compete at the source with the file transfer. The Normal (Both) and Isochronous (MPAC) messages also experienced a slight increase since they use the same Core-Payload bridge station as the emergency messages. For this particular workload scenario, the results indicate that the 100 Mb/s bus provides a wide margin of growth since even at the peak load (Workload II) the utilizations are under 10%. Of course much more analysis, including assessment of DMS growth is needed before a conclusion on the necessity of a 100 Mb/s bus can be reached.

Table 4 shows the results for Case II using the 20 Mb/s buses. As expected the emergency messages have a greater impact in this case. The delay increase ranges from 3% for the Normal messages to almost 10% for the Isochronous (MPAC), however the delays are still within the DMS requirements. The messages experiencing the greatest delay increase are those that are competing with the emergency messages at the source or bridge station. At peak loads, bus utilizations of less than 48% were achieved, indicating that the 20 Mb/s bus is adequate and contains some spare bandwidth. However, as shown in the previous section, the FODS protocol suffers performance degradation at slightly higher loading. The question of the whether this bandwidth reserve is enough is still under investigation.

5. CONCLUSIONS

The DMS.SIM simulation tool for performance assessment of multiple LAN communication architectures for the Space Station DMS has been presented. Two data link level protocols have been simulated, a token passing logical ring and FODS, a randam access scheme, both of which allow a number of parametrically variable configurations and operating rates. Using the table-driven inputs to the simulation program, it is possible to create specific message workloads for analysis of the transient responses of the networks to particular events. Several example cases were run using the DMS.SIM models and used to compare the performance of the token bus and the FODS protocol under various loadings. Lastly, an example using a realistic Space Station architecture and message workload was presented, and sample simulation results were used to illustrate the types of output available.

The use of DMS.SIM to model the DMS architecture and workload displayed the effectiveness and ease of use of the simulation. Representing the DMS Core and Payload Network topology was simple because DMS.SIM allows any topology consisting of multiple buses connected by bridges. The model is flexible enough to consider differing bus rates such as 20 Mb/s local buses connected to 100 Mb/s global buses through bridges. Perhaps the most powerful aspect of DMS.SIM is the workload model. Realistic scenarios were created that combined actual and statistical characterizations of the traffic. For example, many of the messages can be specified as having random destination on a particular network, while others have precise destinations. The message tagging facility was of great use in defining statistics that allowed separate measurement of message types, rather than overall averages. In summary, DMS.SIM was effective due to the generic network topology, user specified protocol parameters, realistic workload model and integrated statistics facilities.

	WORKLOAD I †			WORKLOAD II ‡		
Message Type and -Tag	Mean (10^{-3}s)	95% (10^{-3}s)	Std. Dev. (10^{-3})	Mean (10^{-3}s)	95% (10^{-3}s)	Std. Dev. (10^{-3})
Background(MSU1)-5	0.90	1.05	0.08	0.93	1.15	0.11
Background(MSU2)-8	0.91	1.05	0.08	0.91	0.95	0.08
Normal(Both)-4	1.40	1.59	0.11	1.41	1.60	0.13
Normal(Core)-1	0.91	1.03	0.08	0.91	1.03	0.08
Normal(Payload)-2	0.98	1.28	0.17	0.99	1.30	0.18
Isochronous(MPAC)-3	1.25	1.33	0.06	1.27	1.43	0.08
Isochronous(Robot)-6	1.25	1.35	0.06	1.26	1.35	0.07
Emergency (Both)-7	-	-	-	1.42	1.65	0.09

† Utilizations: Core = 6.0%, Payload = 7.3%

‡ Utilizations: Core = 8.3%, Payload = 9.4%

Table 3: CASE I Example: Message Delay With 100 Mb/s LANs

	WORKLOAD I †			WORKLOAD II ‡		
Message Type and -Tag	Mean (10^{-3}s)	95% (10^{-3}s)	Std. Dev. (10^{-3})	Mean (10^{-3}s)	95% (10^{-3}s)	Std. Dev. (10^{-3})
Background(MSU1)-5	1.37	2.08	0.39	1.47	2.25	0.53
Background(MSU2)-8	1.41	2.30	0.42	1.50	2.50	0.46
Normal(Both)-4	2.53	3.95	0.75	2.64	4.20	0.89
Normal(Core)-1	1.39	2.08	0.36	1.43	2 23	0.42
Normal(Payload)-2	1.48	2.18	0.34	1.53	2.30	0.43
Isochronous(MPAC)-3	1.75	2.90	0.53	1.93	3.55	0.82
Isochronous(Robot)-6	1.72	2.75	0.50	1.86	3.20	0.68
Emergency(Both)-7	-	-	-	2.57	3.70	0 59

† Utilizations: Core = 29.9%, Payload = 36.5%

‡ Utilizations: Core = 41.6%, Payload = 47.1%

Table 4: CASE II Example: Message Delay With 20 Mb/s LANs

Future investigations into the effects of the bridge delays, NIU buffering, and higher layer protocol functions in the NIU are necessary to complete the DMS evaluation. Preliminary models of detailed NIU architectures and protocol software are being developed. Further work on the simulation program is concentrating on including a transport layer module that implements more of the transport level features such as time-outs, flow control, duplicate packet checking, and the associated NIU processing overhead. Other future studies will involve linking the DMS.SIM to the overall SSIS communications to incorporate explicitly the air-to-ground, satellite links, and orbital aspects. In addition, a distributed system simulation program has been developed and is being used to integrate the computer processing systems with the DMS communication for application level performance evaluation [33]. The method for coupling the processing model and the communications oriented DMS.SIM is under investigation.

REFERENCES

[1]. Space Station Data System Analysis/Architecture Study, Draft Task 1 Report (Rev. 1), SSDS-TR-101-5-85, TRW Corp., Mar., 1985.

[2]. Space Station Data System Analysis/Architecture Study, Draft Task 4 IOC Preliminary Design Report, SSDS-TR-104-8-85, TRW Corp., Aug., 1985.

[3]. W. E. Mallary and V. A. Whitelaw, "Space Station Data Management System Architecture," Proceedings of the IEEE, Vol. 75, No. 3, pp. 320-328, March, 1987.

[4]. Space Station Projects Definition and Requirements Document, NASA JSC-31000, Johnson Space Center, Houston, TX, Nov., 1986.

[5]. Architecture Control Document: Data Management System, NASA JSC-30261, Johnson Space Center, Houston, TX, Jan., 1987.

[6]. International Standards Organization (ISO), " Information Processing Systems - Open Systems Interconnection - Basic Reference Model," Draft Standard ISO/DIS 7498, 1982.

[7]. R. L. Larsen, J. R. Agre, and A. K. Agrawala, "Comparative Evaluation of Local Area Communication Technology," CPEUG 1980, Orlando, FL, 1980, pp. 87-97.

[8]. M. P. Varga, "A Robust 100 MBPS Network For Avionics Applications," NAECON Proceedings, 85CH2189-9, Vol. 1, May, 1985, pp. 103-111.

[9]. W. R. Jones, "A Computer Modeling Methodology and Tool for Assessing Design Concepts for the Space Station Data Management System," NASA Tech. Memo #87647, Langley Research Center, Hampton, VA, April, 1986.

[10]. A. Vallone, "Computer Aided Assessment Model Development for Space Station Data Management System Concepts: Space Station Representative Data Base (Revision 1)," CSC/TM-85/6124, Computer Sciences Corp., Jan., 1986.

[11]. M. Z. Wu, "Computer Aided Assessment Model Development for Space Station Data Management System Concepts: Automated Distributed Architecture Modeling (ADAM) User's Guide", CSC/TM-85/6050, Computer Sciences Corp., Dec., 1985.

[12]. K. Richardson, "LANES II User Manual," NASA Technical Note TN-85-7104-925-49, Ames Research Center, Moffett Field, CA, Dec., 1985.

[13]. K. Richardson, "LANES III User Manual," NASA Technical Note, Ames Research Center, Moffett Field, CA, 1986.

[14]. SIMSCRIPT II.5 Programming Language, CACI Inc., Los Angeles, CA, 1987.

[15]. A. R. K. Sastry and M. W. Atkinson, "Simulation of the IEEE 802.4 Token Passing Bus Protocol Using SIMSCRIPT," NBS Workshop on Analytic and Simulation Modeling of IEEE 802.4 Token Bus, Gaithersburg, MD, Apr., 1985.

[16]. J. R. Agre, I. H. Shahnawaz, M. W. Atkinson, and A. R. K. Sastry, "Simulation of a Local Area Network For a Newspaper Production System," Winter Simulation Conference, Dec., 1987.

[17]. M. W. Atkinson and A. R. K. Sastry, "A Simulation Model For the FDDI Token Passing Ring Scheme," International Communications Conference, Seattle, WA, June, 1987, pp. 1300-1304.

[18]. Token Passing Bus Access Method and Physical Layer Specifications, IEEE Standard 802.4, 1985.

[19]. FDDI Token Ring Media Access Control, Draft Proposed American National Standard, ASC X3T9.5 Rev. 8, March 1, 1985.

[20]. M. Johnson, " Reliability Mechanism of the FDDI High Bandwidth Token Ring Protocol," RIACS, NASA Ames Research Center, May 1985.

[21]. Token Passing Ring Access Method and Physical Layer Specifications, ANSI/IEEE 802.5, 1985.

[22]. Space Station Work Package 2 Advanced Development: BIU Definition Report, SSS 87-0079, Rockwell/Sperry, Jan., 1987.

[23]. Space Station Work Package 2 Advanced Development: FODS Upper Layer Development Report, SSS 87-0077, Rockwell/Sperry, Jan., 1987.

[24]. N. R. Howes, "Tinman User's Manual For the Data Management System Test Bed," Lockheed Engineering and Management Co., Dec., 1985.

[25]. Space Station Information System Test Bed: User Data Flow Task S-160 Interim Report, S403, Ford Aerospace & Communications Corp., Aug., 1986.

[26]. IEEE Project 802, Local Area Network Standards, "Ethernet Access Method and Physical Layer Specifications," IEEE Standard 802.3, 1985.

[27]. W. R. Franta and J. R. Heath, "Measurement and Analysis of HYPERchannel Networks," IEEE Trans. on Computers, Vol. C-33, No. 3, Mar., 1984.

[28]. dBase III Plus, Ashton-Tate, Torrance, CA, 1985.

[29]. ENERGRAPHICS, Enertronics Research, Inc., St. Louis, MO, 1985.

[30]. Space Station Work Package 2 Definition and Preliminary Design Phase Data Management System Report DR-02, SSS 86-0073 Vol. II, Rockwell International Corp., Dec., 1986.

[31]. Mission Requirements Database, NASA, Johnson Space Center, Houston, TX, Jan., 1987.

[32]. Space Station Definition and Preliminary Design, Preliminary Analysis and Design Document (DR-02) Data Management System Book 12, MDC H2028A, McDonnell Douglas Corp., June, 1986.

[33]. J.R. Agre, M. W. Atkinson, and C. Y. Wang, "Performance Modeling of Distributed Simulation Through Simulation," International Communications Conference, Seattle, WA, June, 1987, pp. 1321-1327.