N95- 17577

111162

31982

p 7

EVALUATION OF NASA's END-TO-END DATA SYSTEMS USING DSDS+

Christopher Rouff
William Davenport
NASA Goddard Space Flight Center
Code 522.1
Greenbelt, MD 20771
(301) 286-2938
(301) 286-5149

Philip Message
Stanford Telecommunications Inc.
7501 Forbes Boulevard
Seabrook, MD 20706
(301) 464-8900

ABSTRACT

The Data Systems Dynamic Simulator (DSDS+) is a software tool being developed by the authors to evaluate candidate architectures for NASA's end-to-end data systems. Via modeling and simulation, we are able to quickly predict the performance characteristics of each architecture, to evaluate "what-if" scenarios, and to perform sensitivity analyses. As such, we are using modeling and simulation to help NASA select the optimal system configuration, and to quantify the performance characteristics of this system prior to its delivery.

This paper is divided into the following six sections:

I. The role of modeling and simulation in the systems engineering process. In this section, we briefly describe the different types of results obtained by modeling each phase of the systems engineering life cycle, from concept definition through operations and maintenance.

II. Recent applications of DSDS+. In this section, we describe ongoing applications of DSDS+ in support of the Earth Observing System (EOS), and we present some of the simulation results generated of candidate system designs. So far, we have modeled individual EOS subsystems (e.g. the Solid State Recorders used onboard the spacecraft), and we have also developed an integrated model of the EOS end-to-end data processing and data communications systems (from the payloads onboard to the principle investigator facilities on the ground).

III. Overview of DSDS+. In this section, we define what a discrete-event model is, and how it works. The discussion is presented relative to the DSDS+ simulation tool that we have developed, including it's run-time optimization algorithms that enables DSDS+ to execute substantially faster than comparable discrete-event simulation tools.

IV. Summary. In this section, we summarize our findings and "lessons learned" during the development and application of DSDS+ to model NASA's data systems.

V. Further Information.

VI. Acknowledgments.

I. THE ROLE OF MODELING AND SIMULATION IN THE SYSTEMS ENGINEERING PROCESS

As illustrated in Figure 1, modeling and simulation are invaluable tools throughout the systems engineering life cycle, as described in the following paragraphs.

During the concept definition phase, modeling is used to validate the operations concepts, and to derive preliminary estimates of system requirements. For example, an operations scenario for EOS entails recording of payload data generated onboard the

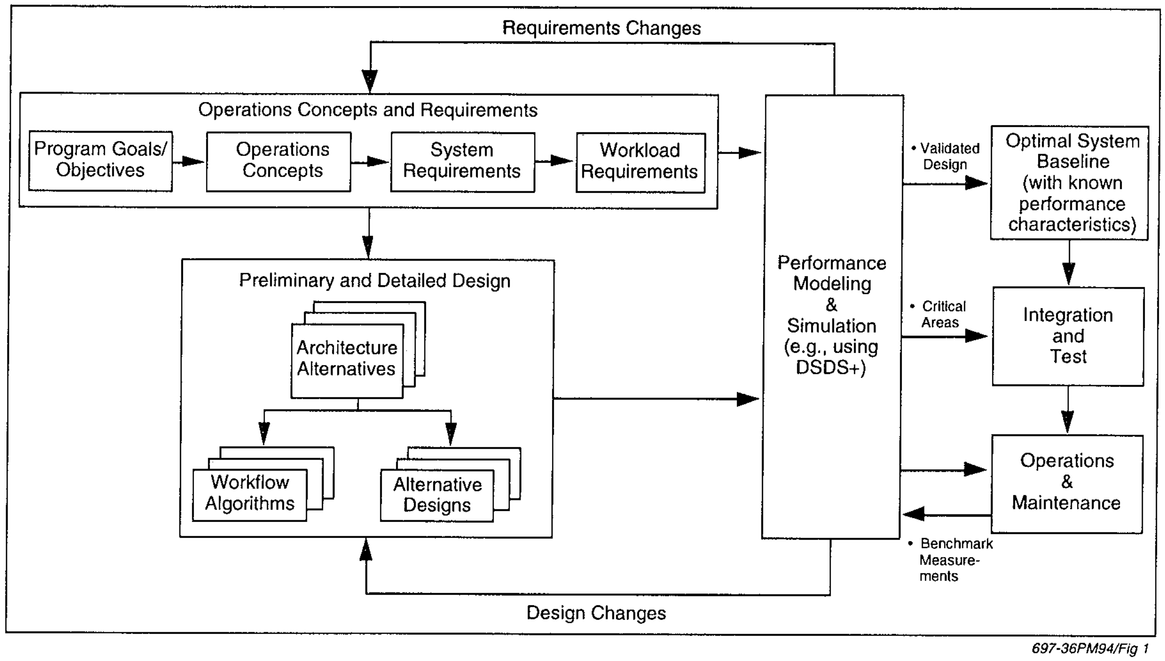

Figure 1. The Role of Modeling and Simulation in the Systems Engineering Life Cycle

spacecraft during each orbit, followed by periodic downlinking of the data during 10 minute contacts scheduled with the Tracking and Data Relay Satellite System (TDRSS). Modeling these scenarios provides estimates of the minimum onboard and ground-based storage requirements, and the minimum communications bandwidths necessary to distribute all of the data received during a downlink contact before data is received for the next contact period.

During the preliminary and detailed design phases, modeling is used to evaluate the performance of physical resources, configured in a certain topology to process the offered workload. The resources modeled include CPUs, busses, disks, networks, etc., and the workload includes software jobs/tasks to be executed, data to be processed/transferred, etc. Performance metrics generated by such a simulation include CPU utilization, queue sizes, network utilization, data latency, etc. Thus, simulation of the physical design adds an additional level of fidelity and insight into the anticipated behavior of the system, and the performance metrics generated reflect the practical constraints of the real system, above and beyond the theoretical minimums generated by modeling the operations scenarios.

During the integration and test phases, modeling is used to identify critical system functions and interfaces, and aspects of the system that have the smallest performance margins. Particular attention should be paid to these areas during testing, and the simulation results can be used to devise stress scenarios for subsequent testing.

During the operations and maintenance phase, modeling is used to evaluate the impact of any proposed changes to the system requirements or system design, such that the changes can be well-understood, and any side-effects identified. Further, performance benchmark measurements can be taken of the real system and compared against the simulated results generated in earlier life-cycle phases. These benchmark measurements can then be used to validate the simulation models (and, if necessary, to make refinements to the models), thereby enhancing the fidelity and level of confidence in subsequent simulation activities.

II. RECENT APPLICATIONS OF DSDS+

DSDS+ is currently being used at Goddard Space Flight Center (GSFC) to model the space and ground segments of the Earth Observing System, at Marshall Space Flight Center (MSFC) to model the Space Station Freedom Data Management System, and at

Johnson Space Center (JSC) to model the Space Station Freedom Control Center.

A major component of NASA's Mission to Planet Earth (MTPE) is the EOS program at GSFC. EOS encompasses many project boundaries, each responsible for different technical disciplines (e.g. spacecraft/instrument command and control, raw telemetry data processing, science data processing, data distribution, etc.); several of these organizations have utilized DSDS+ to conduct performance assessment studies germane to their areas of interest, and in addition, GSFC is sponsoring development of an end-to-end simulation model of EOS.

DSDS+ Model of End-to-End EOS System

The top-level schematic of the return-link, end-to-end data flows modeled for EOS is illustrated in Figure 2. The bullet-items listed to the right of each subsystem in the figure indicate those functions that have been modeled to-date. Other functions will be simulated in the near future, and the model will be updated as the EOS system definition evolves.

In addition to the wide range of functions noted on Figure 2, the following salient features of the model are worth pointing out:

- The simulation consists of a single, integrated model of three distinct segments of the EOS architecture: the EOS AM-1 spacecraft, the Space Network, and the EOS Data and Information System (EOSDIS).

- The end-to-end model is supplemented with more-detailed models of the Solid State Recorder, the Telemetry Processing Systems, and the network connecting the Science Data Processing Systems.

- The end-to-end model is being used to quantify the performance characteristics of the systems and sub-systems within each segment, as well as the performance impact of one segment on another.

- The fidelity of the simulation results is improved by reading external instrument timelines which specify the exact data rates of each instrument at

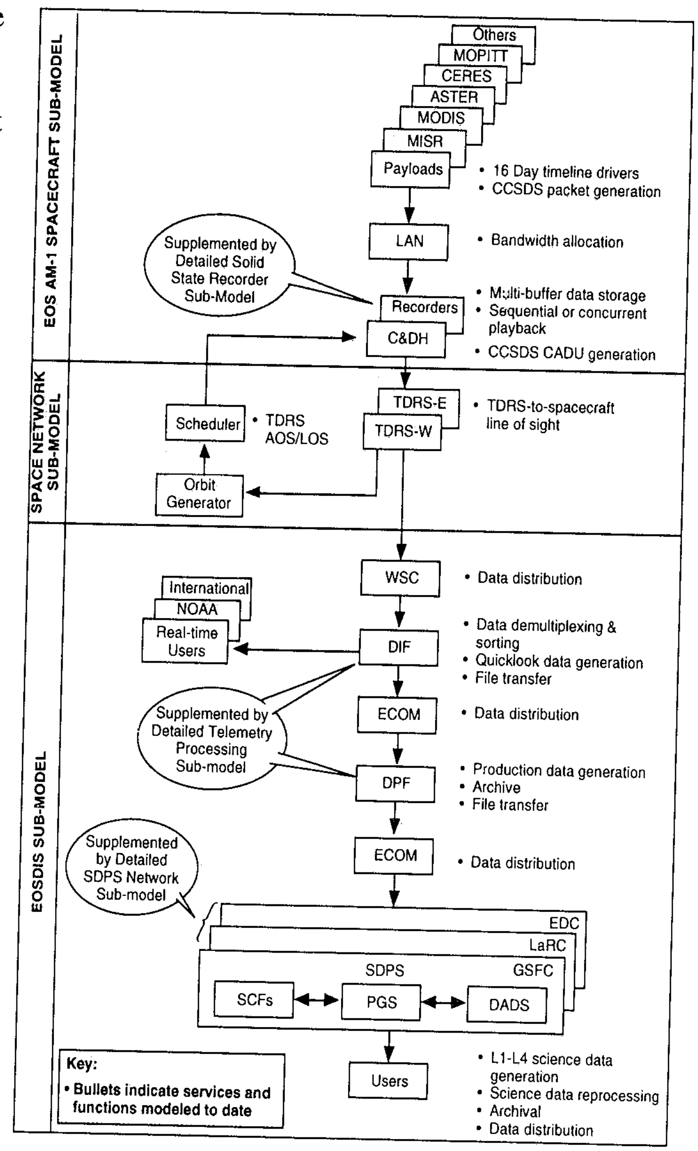

Figure 2. DSDS+ Model of End-to-End EOS AM-1 Architecture

each point in time throughout the 16-day cyclic period of the spacecraft. (The spacecraft makes successive orbits of the Earth, such that the entire surface area is viewed after 16 days, and then the cycle repeats.)

- Each iteration of the model (i.e. each "what-if" evaluation) is executed for a 16-day simulated period, corresponding to the spacecraft cyclic period. Each 16-day iteration takes less than 5 minutes to execute, due to the simulation optimization algorithms described in Section IV of this paper.

- The model generates hundreds of statistics that depict the performance characteristics from three perspectives: end-to-end, point-to-point, and sub-system by sub-system. For example, Figure 3 illustrates the end-to-end latency of NOAA data, assuming that there are no service interruptions in the system. As illustrated, in this scenario there is a 95% probability that NOAA will receive its data in 81 minutes or less, and none of its data will be delivered more than 127 minutes after the time of generation onboard.

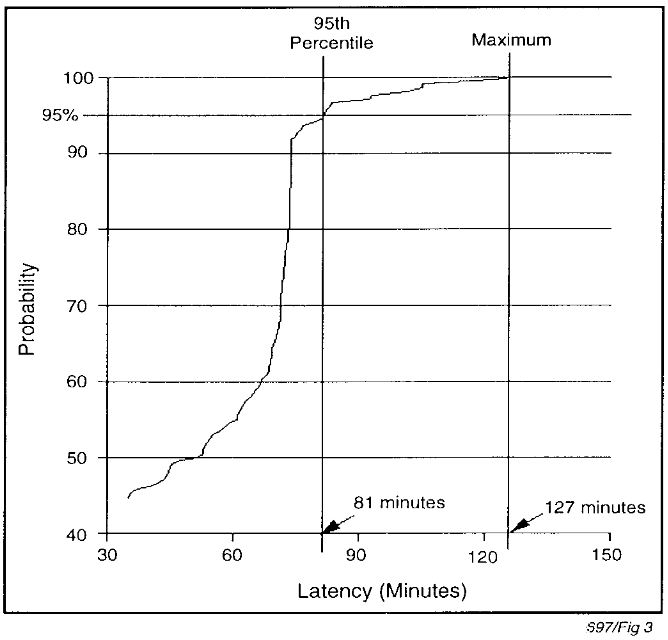

Figure 3: End-to-End Latency for NOAA Data

DSDS+ Model of EOS Solid State Recorder

During the last five years, several different technologies and management schemes have been proposed for implementation of the data recorders onboard the EOS spacecraft. The particular solutions proposed have had widely differing effects on cost, size, weight, shelf-life, maintainability, and performance. During this period, we have applied DSDS+ to evaluate the performance metrics of these different technologies, and we have determined factors such as: the number of recorders required, their capacities, their latencies, their required recording and playback rates, their impact on the ground data processing system, etc.

The most recent advances in technology now support high capacity, space-qualified, solid state recording devices (i.e. memory chips), with significant performance benefits. For example, these devices enable the different payload data streams to be written to different physical partitions, that can then be played back sequentially (thereby enabling high-priority data sources to be transmitted first), or they can be played back concurrently (thereby providing each payload with equal access to the downlink channel).

The DSDS+ results recently obtained by modeling the Solid State Recorders are illustrated in Figure 4. As indicated, the maximum buffer size required to support the EOS-AM1 payloads is approximately 122.5 Gbits, well below the planned capacity of 140 Gbits. However, these results are contingent upon the assumption that there are "near-perfect" operations throughout the end-to-end system. A more realistic assumption is that there are occasional service interruptions: for example, missed contact periods between the spacecraft and TDRSS due to loss of signal. The EOS-AM1 spacecraft makes 233 orbits during each 16-day cycle, and it is scheduled to receive two contacts with TDRSS during each orbit; i.e. it receives a total of 466 contacts per 16 day cycle. Therefore, we re-ran the Solid State Recorder model 466 times, missing a different TDRSS contact each time. As each simulation executed, we obtained the maximum buffer size observed during the 16 day simulated period; we then plotted the results, which are given in Figure 5.

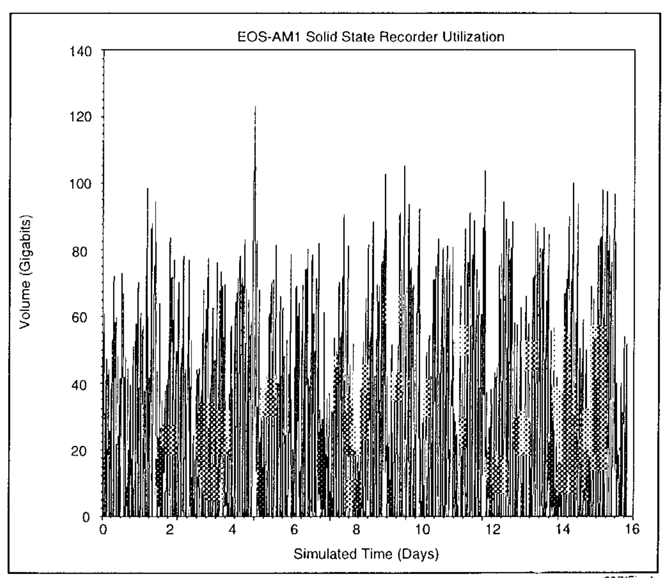

Figure 4. EOS AM-1 Solid State Recorder Utilization

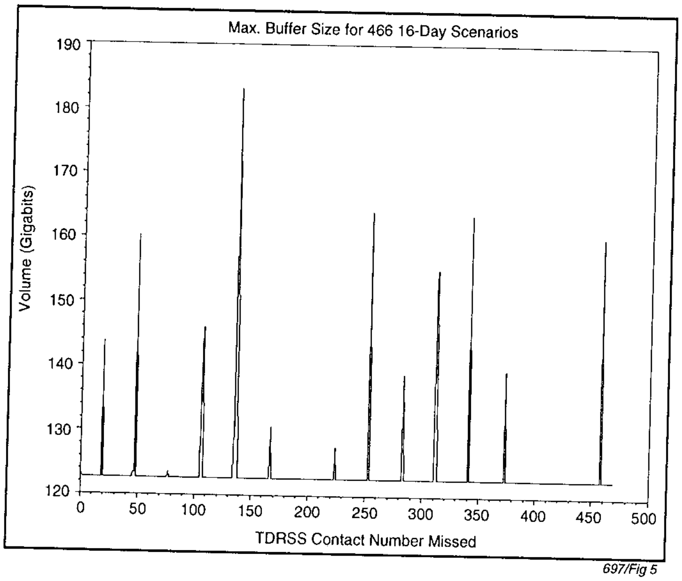

Figure 5. Maximum EOS AM-1 Solid State Recorder Utilization

As indicated in Figure 5, the volume of data buffered exceeded the Solid State Recorder capacity of 140 Gbits on eight occasions (e.g. when TDRSS contact number 19 was missed, when contact number 48 was missed, etc.). Therefore, there is approximately a 2% probability (8/466*100) that data will be lost if a TDRSS contact is missed. Also, it is worth noting that a TDRSS contact can be missed in the majority of cases without impacting the maximum volume of data that has to be recorded (i.e., the volume remains constant at 122.5 Gbits because the worse-case buffering occurs at some other point in the 16-day cycle, and is not related to the TDRSS contact that was missed).

III. DSDS+ OVERVIEW

The Data Systems Dynamic Simulator (DSDS+) is a general-purpose, discrete-event simulation tool. It contains an extensive library of pre-programmed simulation elements that are connected together by the user to represent the real system being modeled. Examples of the pre-programmed elements include: data generators and sinks, data processors (e.g. CPUs with various service disciplines), buffers and queues, and data switches and routers. Each of these elements simulates a particular function or service, which may be tailored by the user to represent the specific application being modeled. For example, the data generator has a list of parameters associated with it that enable the user to define characteristics such as the packet sizes to be generated, their inter-arrival times, their priorities, etc. If desired, multiple instances of an element may be included in the model (e.g. multiple data generators), and each instance will have its own set of parameters defining the specific operations being simulated.

Models are developed pictorially in DSDS+, using a graphical user interface that provides close correlation between the model representation and the real system. Further, the model drawings can be developed hierarchically, to any depth required, so that complex models can be decomposed into a series of detailed sub-level models, as illustrated in Figure 6.

As illustrated in the figure, events (i.e. messages) flow from element to element within discrete-event models. When the event arrives at an element, the underlying code associated with the element is executed, and some action is taken to simulate the operations of the real system. For example, an element that simulates the TDRSS propagation delay might hold the event for a quarter of a second before forwarding it to the next element in the model. A slightly more complex element might calculate the transmission delay by dividing the bandwidth (input as a user-supplied parameter associated with the element) by the size of the incoming event to be transmitted. As the model executes, simulation results can then be collected automatically, as a function of time, simply by observing the flow of events in the system, or by observing the sizes of the internal queues, etc.

It should be noted that DSDS+ events do not carry the real data with them in the model, but rather, they carry attributes that define the characteristics of the real data (such as the packet size). As illustrated in Figure 6, the events are held on a chronologically ordered list (called an event calendar) that is maintained by the scheduling engine. The engine removes the event from the top of the list, it instantaneously advances the simulation clock time to the new scheduled time, and it then forwards the event to the appropriate element for subsequent execution. Thus, there is no relationship between wall-clock time and simulated time, and the next event might be scheduled for processing in a (simulated) nano-second or a (simulated) day.

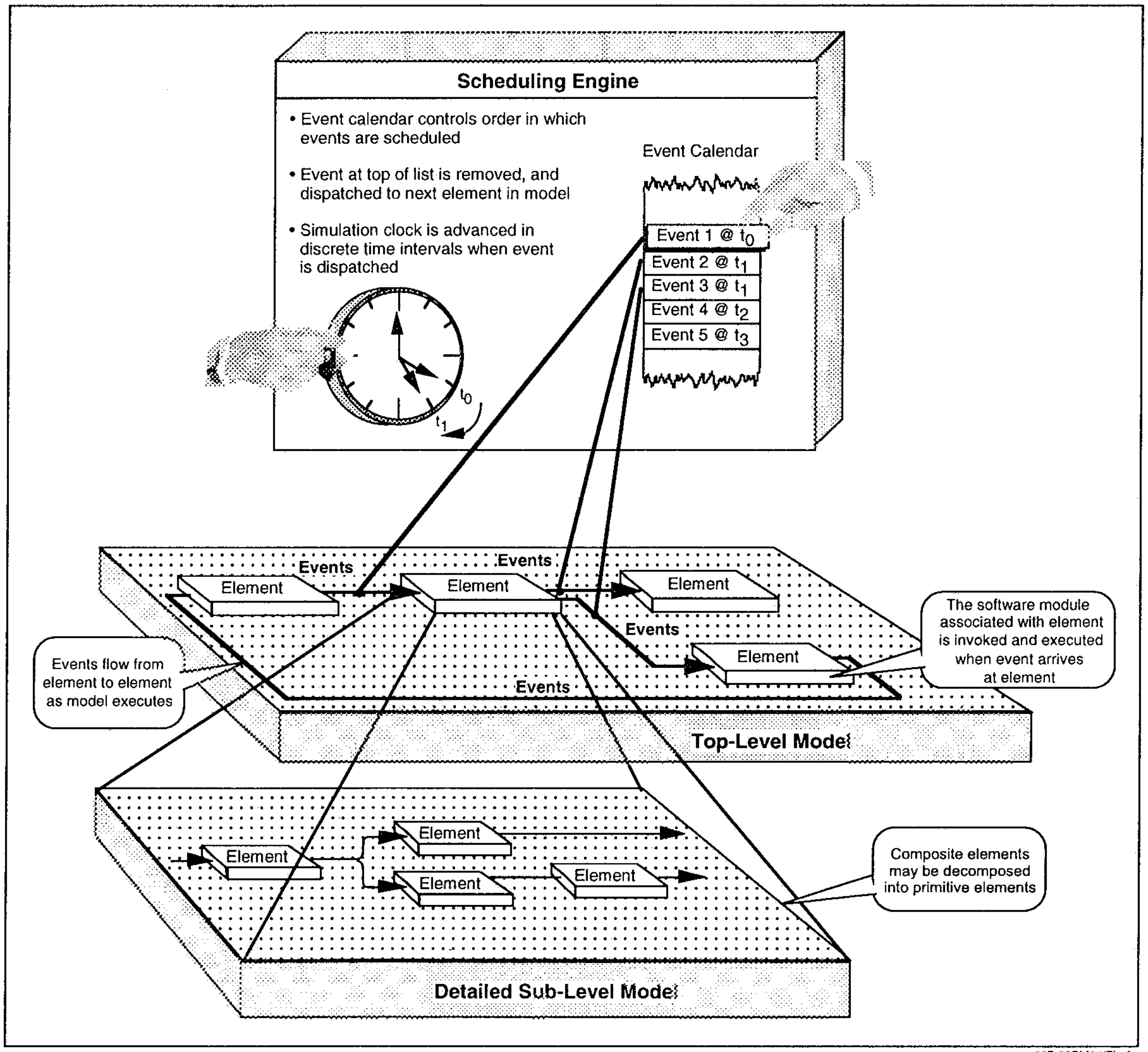

Figure 6. DSDS+ Simulation Concepts

However, the time required for a discrete-event model to terminate will increase with the total number of events to be processed. If each packet is modeled as an event, then end-to-end models of NASA's high data rate systems will require many months to terminate, even when executed on high performance workstation-class computers. The reason is obvious: the real system will be implemented by multiple "super-computers" distributed throughout the space and ground segments, each processing tens of thousands of packets per second. Therefore, how can a simulation model keep pace, since it is hosted on a single computer? We have implemented a solution to this problem within DSDS+, using a hybrid continuous-flow and discrete-event technique that we call "data streams". Briefly, the data stream methodology takes advantage of the fact that successive packets flow through a data system at a constant data rate, with relatively infrequent changes in the rate. Thus, the system can be modeled by considering the impact of what happens when the rate changes, without regard to the individual packets that constitute the data flow. For example, if during some time interval, a data source temporarily generates data at a rate that exceeds the processing capacity, then the queue size (and resultant queuing delay) will increase linearly with time until the source stops generating data, and then the queue size will decrease linearly with time (although the queuing delay will continue to increase linearly with time until the queue is empty).

The data stream approach is ideally suited to model NASA's data systems, since many of the science instruments generate data at a constant rate during each duty cycle, with relatively infrequent rate changes. Therefore, a data stream model is required to process relatively few events (each of which represent a change in data rate), and it doesn't matter that the data rates themselves are extremely high (typically, up to 150 Mbps). As a result, we are able to utilize DSDS+ to model complex, end-to-end data systems, at a detailed-level, for very long periods of simulated time and yet generate the results within just a few minutes (for example, the 16 day simulations of EOS require less than 5 minutes to terminate).

IV. SUMMARY

The preceding sections have demonstrated that modeling and simulation are invaluable systems engineering tools to help define and select the optimal system configuration. Further, the performance characteristics of this system will be known prior to its delivery. This is not just because simulation results have been generated, but also because modeling is a two-way street, and the questions asked in order to develop a model usually prompt the systems engineer to resolve ambiguities or incomplete specifications that would otherwise have gone un-noticed. Therefore, it is our belief that the steps required to develop a model should be undertaken, even if the model itself is never actually constructed.

Simulation models are also relatively inexpensive to develop - far less than the cost of trying to correct performance problems subsequently found in the as-built system! For example, the DSDS+ simulation models of the EOS Solid State Recorder were developed in just a few staff-weeks, and yet their pay-off has been tremendous: the EOS project has decided to increase the recorder capacity to 200 Gbits to prevent loss of the science data.

Finally, we believe that the unique run-time optimization algorithms in DSDS+ make it the most suitable tool available to model NASA's end-to-end data systems. While there are many excellent commercial tools on the market, none contain any optimization methodologies; therefore, practical constraints limit their use to evaluation of localized systems, simulated for short time durations.

V. FURTHER INFORMATION

This paper is presented in conjunction with an online demonstration of DSDS+, including the simulation models developed recently of NASA's end-to-end data system.

DSDS+ is a NASA-owned tool, and therefore it is available free of charge to any NASA organization or support contractor. For further information, please contact Bill Davenport at (301) 286-5149, or at the address given at the top of this paper.

VI. ACKNOWLEDGMENTS

The Data Systems Technology Division, GSFC Code 522.1, is managing the development of DSDS+, and Stanford Telecommunications, Inc. is the implementation contractor, under the Systems Engineering and Analysis Support (SEAS) contract.

The following NASA organizations and programs have sponsored the development and/or application of DSDS+ to-date[1]:

GSFC Code 500 (1992 - 1993): Chief Engineer's Office
GSFC Code 502 (1986 - 1991): Customer Data and Operations System (CDOS) Project
GSFC Code 502 (1988 - 1990): Earth Observing System Data and Information System (EOSDIS) Project
GSFC Code 504 (1993 - 1994): Systems Engineering Office
GSFC Code 505 (1993 - Present): Earth Science Data and Information Systems (ESDIS) Project
GSFC Code 520 (1986 - Present): Data Systems Technology Division
GSFC Code 560 (1993 - Present): EOS Data and Operations System (EDOS) Project
HQ Code R (1991 - 1993): Office of Aeronautics and Space Technology (RTOP Project)
HQ Code S (1989 - 1991): Space Station Freedom Program

[1]Several of these projects/organizations have since been reorganized and renamed, but the affiliation names listed are the ones in effect when the work was sponsored.