Calvin, James O., Joshua Seeger, Gregory D. Troxel, Daniel J. Van Hook,
"STOW Realtime Information Transfer and Networking System Architecture,"
95-12-061, Twelfth Workshop on Standards for the Interoperability of Distributed Simulations,
March 13-17, 1995.

STOW REALTIME INFORMATION TRANSFER AND NETWORKING SYSTEM ARCHITECTURE

James O. Calvin, `jcalvin@ll.mit.edu`, MIT LL
Joshua Seeger, `jseeger@bbn.com`, BBN STD
Gregory D. Troxel, `gtroxel@bbn.com`, BBN STD
Daniel J. Van Hook, `dvanhook@ll.mit.edu`, MIT LL
For the RITN program team

Keywords: Architecture, Bundling, Communications Architecture, Fidelity,
Local Area Network, Multicast, Wide Area Network

Abstract

ARPA's Synthetic Theater of War (STOW) program faces significant challenges as it builds upon the capabilities of today's distributed simulation systems and architecture. These challenges include requirements to economically support:

- greatly increased numbers of interacting entities
- an enriched synthetic environment supporting dynamic terrain, smoke, weather, electromagnetic effects
- more complex interactions, including those of command forces, EW/ECM, wide area viewers, rapidly steerable imaging systems, enhanced sensors

These and other requirements will severely stress information transfer and processing capabilities. Significant advances in Real-time Information Transfer and Networking (RITN) architectures and designs are needed to support STOW program goals. Initial steps towards a new architecture and systems design were taken and successfully demonstrated as part of the STOW-Europe (STOW-E) exercise, conducted in the fall of 1994. These initial steps were reported on at the 11th DIS Workshop.

This paper describes the approach being taken for the STOW RITN architecture, which builds upon the STOW-E approach. The specific topics addressed in the paper include:

- requirements and system constraints
- architectural vision: the longer range goal
- near term architectural approach (bi-level multicast)
- algorithms for managing information flow, such as data subscription and the use of IP multicast
- key simulation application interfaces, e.g., quality of service (QoS), resource reservation, etc.

The architecture and approaches developed in support of STOW will have a significant impact on DIS standards development.

0.0 Preface

While the paper lists a set of four authors, it is the combined work of the RITN team. Contributors to this paper include:

Dr. Stuart Milner
L. S. (Lee) Kollmorgen
Walter Milliken
R. Tanner
Carol Chiang
Dan Van Hook
Lou Berger
Kevin Russo
Ray Cole
Barth Root
Bibb Cain
Kevin E. Boner
Cynthia Keune
Adam H. Whitlock
CDR Gary Misch
Lester Foster
Joshua Seeger
Jim Calvin
Duncan C. Miller
Joshua E. Smith
Tom Tiernan
Steve Batsell
R. K. Nair
Larry Schuette
Kathleen Ashbaugh
Dave Fusco
Mike Newton

Dr. Milner is the ARPA program manager for RITN and CDR Misch is the DMSO program manager.

1.0 Background

Distributed Simulation is being scaled up along several different dimensions through ARPA's STOW (Synthetic Theater of War) program. Not only is the number of entities being dramatically increased, but the richness of the synthetic environment is being enhanced through addition of phenomenology such as smoke, dynamic terrain, and weather and through simulation of electronic warfare/countermeasures and radio communication. Introduction of command forces and C3I simulation will increase the complexity of interactions and behaviors. Finally, the network is being scaled in terms of the number of sites, the

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

geographic extent, and the degree of network sharing by independent users. These and similar factors result in large increases in complexity, interdependence, and information that needs to be exchanged. As a result, significant challenges for Distributed Simulation technology and systems exist in the areas of configuration management, operations, and real-time information transfer and networking. While all of these areas are of critical importance, this paper focuses on the last challenge listed: architectures and approaches for real-time information transfer and networking for STOW.

In the initial architecture for Distributed Simulation, developed under the SIMNET program, all simulation platforms receive all information broadcast by all the others. Each receiving node is responsible for sorting out what information is relevant to it. This simple broadcast architecture is no longer adequate. Even if sufficient network resources could be employed to carry all the broadcast traffic, it is unlikely that the simulation platforms would be able to economically process all the data they received. The scaleability of the current architecture is limited by this "firehose effect."

In the STOW-E (STOW-Europe) exercise, conducted in the Fall of 1994, a step was taken away from the broadcast architecture towards a new architecture. The STOW-E exercise presented a number of real-time information transfer and networking problems, including wide area viewers, severe throughput limitations on encryption devices and tail circuits, large numbers of entities, limited processing capabilities of many participating simulation applications, and limited wide area multicast support. To address these problems, an Application Gateway (AG) [1] was fielded at each network site. The AG, which was interposed between each site LAN and the DSI WAN, applied a number of algorithms and techniques for managing traffic flowing to and from each site. The algorithms employed in the AG included the following:

- **Culling** blocks PDUs considered unnecessary for the purposes of STOW-E from transmission over the WAN. Examples include transmitter PDUs, Persistent Object Protocol transmissions, and PDUs indicating collisions between entities on the same LAN.
- **PICA** (Protocol Independent Compression Algorithm) removes redundancy. Bit-pattern differences from a reference are transmitted from each sender to the receivers. The reference is conveyed using a reliability mechanism that ensures consistency across the system.
- **Grid filtering** partitions the terrain into regions for which updates are sent at different rates in order to accommodate wide area viewers. Rate control is accomplished through the rethresholding algorithm, which uses multiple dead reckoned models for each entity in conjunction with transmit threshold control to modulate update rates. Regions of the terrain for which different update rates are needed are communicated between AGs by sending grid cell lists. In addition to controlling WAN-bound packet rates, a **LAN filter** that uses grids to select and forward only relevant updates to each LAN is employed.
- **QES** (Quiescent Entity Service) removes redundancy due to the transmit timeout (default is five seconds). QES detects when an entity becomes quiescent, informs all remote AGs, and stops sending updates over the WAN. The AGs at each site then emit updates for the quiescent entities onto their attached LANs at the transmit timeout rate. Should a quiescent entity become active, its updates are again forwarded over the WAN.
- **Rethresholding** controls state update rates over the WAN. Multiple dead reckoned models are maintained for each entity. Transmit thresholds (position, orientation, time) are relaxed to reduce packet rates for selected entities. This algorithm is used in conjunction with grid filtering to support wide area viewers as well as for graceful degradation in the event of overload.
- **Bundling** combines PDUs into larger packets in order to reduce packet rates. A packet is transmitted when either a timer expires or the packet reaches a maximum size. As a side effect, bit rates are reduced since fewer packet headers must be transmitted.
- Overload management and graceful degradation are accomplished through **load leveling**, which spreads packet transmissions out in time and **rethresholding**, which controls update rates by relaxing dead reckoning thresholds.

DOCKET ALARM Find authenticated court documents without watermarks at docketalarm.com.

These algorithms proved to be very effective: traffic was reduced by more than an order of magnitude, permitting the STOW-E exercise to be successfully supported over a constrained network and system infrastructure.

While the success of the approach taken for STOW-E is impressive, much work remains to be done in support of STOW. As an initial prototype and proof of concept, the STOW-E AG and its algorithms must evolve in several directions in order to meet the requirements of the STOW program. First, the High Performance Application Gateway (HPAG) must be able to take advantage of network technologies such as multicasting, resource reservation, and broadband services that will be needed to support STOW. A necessary role of the HPAG is isolating simulation applications from particular interface and performance details of the network, and vice versa. This “impedance matching” role decouples simulation applications and the network and permits independent evolution. Second, new and enhanced information flow management algorithms need to be developed and incorporated into the HPAG and/or simulation hosts in order to handle the significant increases in traffic levels and new traffic types expected. Along similar lines, evolution of the DIS protocol will require support for Agents [2] that function as enablers for simulation applications. The HPAG, along with other systems, will be an execution platform for these Agents. Finally, since the STOW program at its culmination will deliver an operational training system, the Application Control Techniques (ACT which includes the HPAG) must be productized and extended. ACT must flexibly support legacy as well as new simulations. It must also permit migration of its functions so as to make best overall use of system resources. A significant challenge to be faced is the need to construct systems that are useful and economical in the short term but flexible enough to permit growth to address the problems of the longer term.

2.0 STOW Network System Requirements

2.1 Background

The primary requirement of the STOW network system is to deliver the necessary data to the appropriate applications with a minimum of latency while consuming a minimum of network bandwidth. The STOW network system must do this without adversely compromising validity. A complicating factor for this requirement is that technology underlying the STOW network system is evolving while the nature of the data to be delivered is expanding and changing.

2.2 High Level Requirements

In this section we will enumerate the high level requirements of the STOW network system. These requirements will address both the short and long term requirements STOW network system. When necessary, requirement modifications for the 1995 STOW network system will be noted.

2.2.1 The STOW network system must support large exercises with elements supplied from multiple sites from around the world. Exercise targets appear in table 2.2.1-1.

Year	Target Entities	Target sites
1995	5,000	6
1997	50,000	30
2000	100,000	50

Table 2.2.1-1: Entity–site support targets for the STOW network system

2.2.2 The STOW network system must permit simulation applications to be built and run independent of exercise size (specified entity and object count and density). What is meant by this statement is that simulation applications are specified to operate up to a maximum local entity count and density. So long as an exercise scenario does not exceed these *local* limits, the simulation application can function independent of total exercise size.

2.2.3 The STOW network system must provide mechanisms to allow simulation applications to control rejection of irrelevant simulation data as close to the data source as possible.

2.2.4 The STOW network system must support a heterogeneous simulation environment. This requirement includes, but is not limited to, multiple simulation platform types (e.g., Silicon Graphics based, or RS-6000 with Evans & Sutherland image generators), multiple network types (e.g., ATM, FDDI, Ethernet), and multiple simulation types (e.g., live, constructive,

Find authenticated court documents without watermarks at docketalarm.com.

and virtual) operating simultaneously to function as a single system.

2.2.5 The STOW network system must provide a layer of abstraction between the simulation application and the actual network. This abstraction will permit the network to evolve while minimizing changes to the simulation applications.

2.2.6 The STOW network system must provide parametric data that will allow better utilization of the network by all components. Examples of data to be provided include the preferred packet size, number of multicast groups available, maximum packet rate supported, etc.

2.2.7 All three components of the system, network, agents, and applications, must provide health status information to exercise and network monitoring facilities.

2.2.8 All three components, simulation applications, networks, and agents, of the system must provide performance monitoring and logging facilities.

2.2.9 It must be possible to reallocate an ACT function as required. This will allow various portions of the STOW network system to be placed into the computational platform where they provide the best performance for the overall system. Decisions regarding allocation of function will be based on criteria such as cost (recurring and non-recurring), latency, robustness, and throughput.

2.2.10 For those applications requiring it (e.g., interacting manned simulators), the STOW network system must provide application-to-application (or process-to-process) data transfer within 100ms, 80ms of which is allocated to the network (LAN and WAN).

2.2.11 Based on current estimates [3] (that will be revised as new information becomes available), the STOW network system must be prepared to support peak offered loads (aggregate) of 140,000 packets per second and 230Mb per second. It is anticipated that the various Application Control Techniques will reduce these offered loads by some, as yet unknown, factor.

2.2.12 Essential services (software and hardware) which represent single points of failure for the STOW network system must employ mechanisms to assure rapid recovery from failures.

2.2.13 The STOW network system must facilitate current and future DIS protocols.

2.2.14 The STOW network system must support non-DIS data transfer (e.g., ftp and VTC).

2.2.15 The STOW network system must support the following security requirements:

2.2.15.1 Confidentiality of user data is a requirement. User data must not be released to or accessed by an unauthorized person, application, subnet, or network. Authorization is based on clearance level, community of interest, and need-to-know. Both classified and unclassified but sensitive user data must be protected (at an appropriate level) from unintentional or intentional disclosure.

2.2.15.2 The network management system has associated security requirements. These requirements include protection against attacks that could affect availability and performance of network elements. System managers must be authenticated before configuration changes are allowed. Network management protocol data units (PDUs) that result in any action must be authenticated.

2.2.15.3 Key management is an operational requirement. The key distribution system must protect the keys from disclosure to unauthorized entities, ensure that the keys were received from the authorized entity and were not changed during transmission, and also ensure that keys are available when needed.

2.2.15.4 There is a requirement for some observers to obtain access to data at or below their security level (i.e., the stealth viewer).

DOCKET ALARM Find authenticated court documents without watermarks at docketalarm.com.

2.2.16 The STOW network system must be capable of supporting multiple, independent, simultaneous exercises. The total traffic resulting from multiple exercises may not exceed the design goals for a single exercise as shown in table 2.2.1-1.

2.3 User requirements and system constraints

Some user requirements pose difficult challenges for the STOW network architecture and the techniques used to reduce bandwidth demands of simulation applications. In this section, some of these user requirements are enumerated. Solving many of these problems is not strictly a part of the HPAG design, or the STOW network system. However, the STOW network system and HPAG must be designed with these constraints in mind to assure that the user requirements can be adequately met. Experiments must be conducted (with user input or participation) to quantify any adverse effects imposed by the bandwidth-reduction techniques. The final selection of Application Control Techniques will be based on these findings.

2.3.1 Validity

No technique used by the STOW network system or HPAG should adversely affect the validity of an exercise. However, some techniques, such as dead reckoning threshold control and load leveling, can have an effect on validity. It is not clear, however, what the effect of a small amount of threshold control and load-leveling is on an overall exercise.

Further, the validity constraints will not be the same for all types of exercises. Training exercises will typically have a less severe validity constraints than a test and evaluation exercise. Experiments must be performed to determine how these techniques may be used and thereby insure that use of threshold control and load-leveling do not invalidate the results of an experiment.

2.3.2 All data available everywhere

The STOW-E exercise did not deliver full-fidelity[1] data for all entities to all sites. Some of the user community viewed this a deficiency of the system. However, a user requirement for all data to be available everywhere is a response to a perceived problem rather than a hard requirement. The STOW network system must deliver the necessary data to allow each player and observer in an exercise to perform as if all the data were available everywhere. Examples of consumers of such data are plan-view displays and after-action review (AAR) systems. The requirements and constraints of each of these (and other systems) will be addressed in later sections.

2.3.3 Plan-View Displays

Plan-view displays (PVDs) provide a two dimensional view of the exercise playbox. The PVD is an important tool that can provide the user with a view ranging from an overview of all or major portions of the exercise playbox, to a narrow view approximating that of a manned simulator. When the PVD operator uses the broad view, it would appear that all, or nearly all, data is required to support the PVD. However, with this broad view, the PVD cannot display the position of entities with the same accuracy of a manned simulator or when the PVD is used with a narrower field of view. As such, the PVD requires data about all (or many) of the entities in the exercise, but that data can be provided at a reduced frequency of delivery. When the PVD is operated with a more narrow field of view, the normal ACT filtering on high fidelity data can be used to support the PVD. The PVD simulation application will change the type of data it uses based on the operator’s chosen field of view.

2.3.4 Stealths

Another important tool for exercises is the Stealth. The Stealth vehicle allows an exercise observer to move to any point in the exercise playbox and observe the exercise. As such, the Stealth behaves like any other vehicle (with the exception of being invisible to the exercise participants) and should be supported adequately by the mechanisms that support all other entities.

2.3.5 Logging

In exercises prior to STOW-E, all PDUs sent during an exercise were recorded. In STOW-E, this was not so easily accomplished because the AG did not forward all PDUs to all sites. Thus, each LAN had a unique world view. The solution in STOW-E was to record traffic on multiple LANs, and then combine the log files after the exercise.

---

[1] Full-fidelity is defined as the lowest uncertainty data available within the simulation system. The thresholds for this full-fidelity data are set at exercise initialization time, and are traditionally on the order of one meter of positional uncertainty and three degrees of orientation uncertainty.

DOCKET ALARM Find authenticated court documents without watermarks at docketalarm.com.

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**