Successful Deployment of Mesh Routers for Trans-Continental Urban Resolve Experiments

Thomas D. Gottschalk
CACR, Caltech
Pasadena, California
tdg@cacr.caltech.edu

Robert Lucas, Gene Wagenbreth, and John Tran
Information Sciences Inst. /Univ of So Calif
Marina del Rey, California
{rflucas, genew, jtran}@isi.edu

William Helfinstine
Lockheed Martin Info
Burlington, Mass
bill.helfinstine@lmco.co

ABSTRACT

This paper reports the successful deployment of a robust scalable interest-managed router architecture that has supported a series of trans-continental simulations, such as Urban Resolve. Previous architectures had served well over the years, but were conceptually limited both in scalability and in robustness, or fault-tolerance. The scalable router architecture had its inception in high performance parallel computing research and its initial application in a truly scalable architecture for inter-node communications on parallel supercomputers and Linux clusters. Its design provided both needed scalability and desirable robustness on the single platform meshes of several large parallel computers made up of hundreds of compute nodes. The scalable router was designed to integrate smoothly with other Urban Resolve software by reusing Run Time Infrastructures (RTI-s) components. In an effort to minimize communication latency, maximize use of available network bandwidth, and increase robustness of trans-continental (Virginia to Hawai'i) operations, Joint Forces Command's J9 directed that its wide-area router's offer the same characteristics of scalable and robust operations. That led to the wide-area deployment of the scalable routers. This paper sets forth the experience of that evolution, the non-disruptive incorporation of the new routers, the scalability of the interest-managed routing, and the performance of the new network. The assiduous factorization of the program, in order to optimize and temper the code, bore fruit during the implementation process and that factorization activity is explicated and analyzed. Further, the authors look to their experiences in high performance computing to lay out future capabilities and directions for additional development. The area of primary interest and importance is fault tolerance. A specific proposal for the design and fielding of a system impervious to the loss of individual router processes is presented.

ABOUT THE AUTHORS

Thomas D. Gottschalk is a Member of the Professional Staff, a Senior Research Scientist at the Center for Advanced Computing Research (CACR), and Lecturer in Physics all at the California Institute of Technology. Dr. Gottschalk has been active in parallel programming for nearly two decades, with efforts spanning integrated circuit design, intelligent agent simulations, theater missile defense, and physics modeling. He received a B.S. in Physics from Michigan State University and a Ph.D. in Theoretical Physics from the University of Wisconsin.

Robert F. Lucas, Gene Wagenbreth, and John J. Tran are all members of the JESPP team at the Computational Sciences Division of the Information Sciences Institute, Viterbi School of Engineering, University of Southern California. They all have been active in high performance computing research for a more than a decade. Dr. Lucas received BS, MS and PhD degrees from Stanford University. Gene Wagenbreth has a BS in Mathematics and Computer Science from the University of Illinois. John Tran has BS and MS degrees in Computer Science and Engineering from Notre Dame.

William Helfinstine is a senior research engineer in the Lockheed Martin Information Systems division. He has a long history of developing applications for the simulation community in the Department of Defense. He received a MS degree in Computer Science from the Massachusetts Institute of Technology.

Successful Deployment of Mesh Routers for Trans-Continental JSAF Simulations, Urban Resolve

Thomas D. Gottschalk
CACR, Caltech
Pasadena, California
tdg@cacr.caltech.edu

Robert Lucas, Gene Wagenbreth, and John Tran
Information Sciences Inst. /Univ of So Calif
Marina del Rey, California
{rflucas, genew, jtran}@isi.edu

William Helfinstine
Lockheed Martin Info
Burlington, Mass
bill.helfinstine@lmco.co

I. INTRODUCTION

The United States Joint Forces Command's Joint Experimentation Directorate (JFCOM J9) is conducting a series of experiments called Urban Resolve. These are focused on Joint Urban Operations (JUO) and involved human participants geographically distributed across the United States (Ceranowicz 2002). This in turn requires a real time, interest-filtered communication network.

This paper presents a discussion of integration of a new communication network, the MeshRouter, into Urban Resolve to handle the wide area communications. The MeshRouter has initially been developed for use within the two Linux clusters the DOD High Performance Computing Modernization Program made available for J9 (Barrett 2004a). Extending it for use in the wide area brought with it new opportunities for J9 including both lower latency and more effective use of network bandwidth. It also involved a new set of challenges, and this paper discusses how these were resolved.

The MeshRouter, like the existing "tree" router that is also used in UR, has the property that if a router node fails, then all of its clients are cut off from the rest of the experiment, and in general the experiment must be brought to a halt. Given that hundreds of people may be involved and the importance of the insights gained in these experiments, the authors believe that the communication network used to support UR should ultimately be made more fault tolerant, such that no single failure of a router node can halt a UR experiment. A specific proposal for how to accomplish this end is also presented in this paper.

The remainder of this paper is organized as follows. Section II provides a brief review of the MeshRouter, which was first developed for use within scalable parallel processor (SPP) systems such as Linux clusters. Section III discusses how a wide area MeshRouter was integrated into Urban Resolve. Plans for making the Urban Resolve communication network fault tolerant are discussed in Section IV. The paper ends with concluding remarks in Section V.

II. BRIEF REVIEW OF THE MESHROUTER

The MeshRouter provides a general framework for interested-filtered message exchanges in very large-scale distributed discrete event simulations. Details are provided in (Gottschalk 2004ab) and (Barrett 2004ab). For the purposes of this paper, the important aspects of the framework are as follows:

1. Isolation of application-specific components (e.g., "interest" and "message") into carefully factored objects within the overall software framework.
2. General, high-level router processes that manage message traffic, given the application-specific components.
3. Support for a wide variety of communication topologies linking individual routers into a full communications networks.

A schematic illustration of a basic "Router Process" within this framework is shown in Fig.(1).

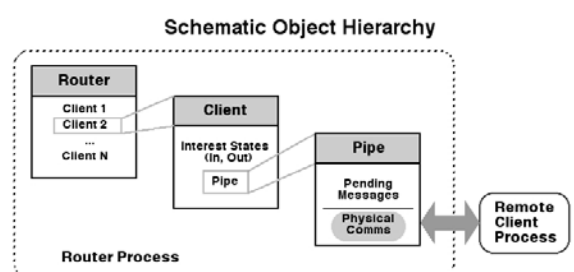

Figure 1, Fundamental objects in the MeshRouter system.

The "Router" object in Fig.(1) manages a list of communications clients. Each client object is characterized by collective interest states and by a particular "Pipe" object that manages the actual "bit-on-a-wire" communications process.

The router processes of Fig.(1) can be tied together in a variety of ways to provide specific communications networks for Urban Resolve. The simplest example is the Tree network illustrated in Fig.(2).

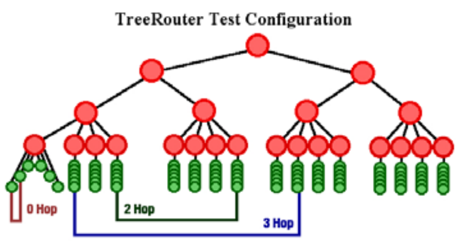

Figure 2, Communications network for a Tree router network.

The specifics of Fig.(2) – in particular, the "N-Hop" labels – are discussed below. In terms of general procedures, the important characterizations of a tree network are as follows:

1. Application processes (the green disks) are organized into fixed groups associated with a given, "lowest-level" router (red disc). All communications to/from an application process go through these low-level routers.
2. A hierarchical tree of additional router processes provides communications among the low-level {Router,Client} groups.

Tree configurations such as Fig.(2) have a number of standard problems. Most notably, as the overall simulation size increases, the depth of the tree increases, meaning that:

1. The communication volume and processing load at the highest-level (root) router increases.
2. The longest communications path within the network, and hence latency, also increases.

An alternative communications architecture available within the MeshRouter framework is based on the "Synthetic Forces Express" (SFExpress) (Brunett 1998, Messina 1998) scalable implementation of ModSAF done for DARPA in 1995-1997. Collections of SAF simulators exchange messages with specified Primary Router processors, and additional "Pop-Up" and "Pull-Down" router layers provided scalable, interest-limited message exchange among the various sets of simulators. The SFExpress network is illustrated in Fig.(3).

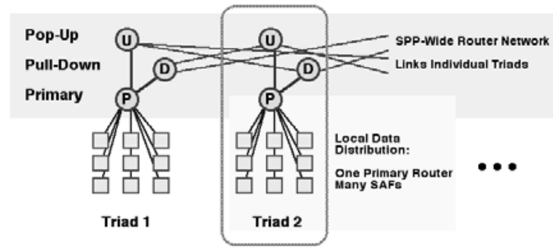

Figure 3, Schematic illustration of the SFExpress communications architecture.

The communications architecture in Fig.(3) is demonstrably scalable provided that the number of messages of interest to any individual applications process is bounded. When connected in a one-dimensional mesh, as depicted in Fig.(3) the maximum number of communications links is three, independent of the overall simulation problem size.

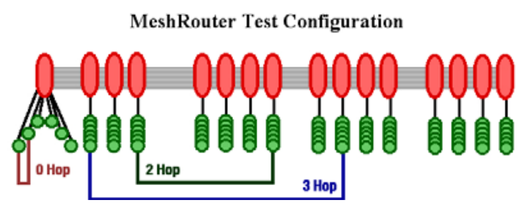

Figure 4, One dimensional mesh topology within the MeshRouter architecture.

The MeshRouter framework supports a straightforward generalization of the SFExpress framework, as illustrated in the schematic of Fig.(4). The three distinct routers for the triads of Fig.(3) are here instantiated as three distinct router processes (in the sense of Fig,(1)) all executing on a single processor.

Performance comparisons for the Tree Router and Mesh Router architectures of Figs.(2,4) were presented in (Amburn 2005). These studies involved comparisons of message exchange times between identified pairs of processors. The pairs were chosen for fixed sets of message path lengths in the tree configuration (this is the origin/meaning of the "N-Hop" labels in Figs.(2,4). The observed ratios of [Tree/Mesh] message times are shown in Fig.(4). The improved performance for the MeshRouter configuration is largely due to the shorter message paths within the mesh.

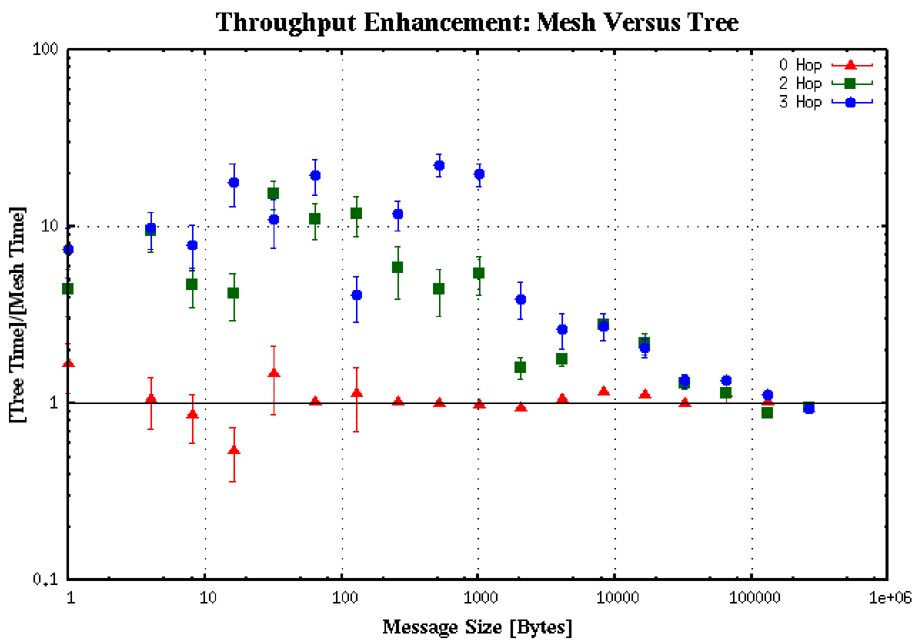

Figure 5, Ratios (Tree/Mesh) of mean communications times versus message sizes.

The studies in (Amburn 2005) were all based on simple "rtiPerf" federates that simply sent/received fixed size messages. Over the past year, these toy federates have been replaced by "real" JSAF federates, as used in ongoing JFCOM Urban Resolve experiments. This is described in the next section.

III. INTEGRATING A WIDE-AREA MESHROUTER NETWORK INTO URBAN RESOLVE

Figure 6, Software routing topology for the urban Resolve experiments.

The wide-area network typically used for Urban Resolve experiments is shown in Fig.(6). The three primary sites are JFCOM in Suffolk, VA, TEC at Ft. Belvoir, VA, and SPAWAR in San Diego, CA. In addition, two large Linux clusters at the Aeronautical Systems Center (ASC) at Wright-Patterson Air Force Base in Ohio and the Maui High Performance Computing Center in Hawaii are also incorporated. As shown in Fig.(6), the five sites are connected by a high-level tree, rooted at JFCOM, joining site-specific tree networks.

It is not feasible to join all the processors at the five sites into a single mesh configuration for two reasons:

1. There are only a limited number of (secure) physical communications links joining the distinct sites.
2. Only a (very) small number of the individual processors within the SPP assets can be accessed directly from the outside world.

Instead, the strategy used to adapt the MeshRouter framework is hierarchical:

1. The five disjoint sites are connected through a single "Transcontinental Mesh", with the nominal local root routers at each site replaced by a full MeshRouter triad processor.
2. Computational assets at each site connect to the appropriate router using either a local tree router or a local mesh router configuration.

The overall configuration is illustrated in Fig.(7).

The practical difficulties associated with implementing and executing the communications model of Fig.(7) were primarily related to specifying the system through use of configuration files. Since these specifications are germane to the discussions of robust routing networks in Section IV, a brief review of the standard configuration specifications for Urban Resolve's applications, using RTI-s, is in order.

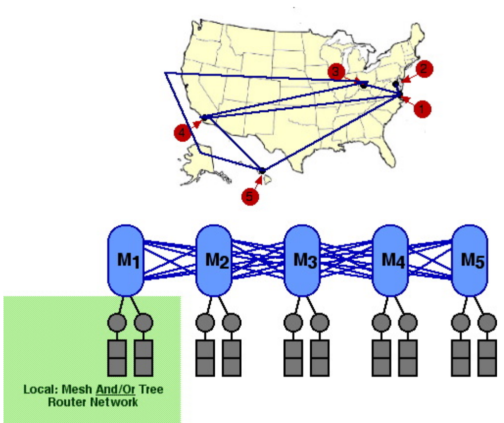

Figure 7, Schematic illustration of a hybrid "TransContinental Mesh" communications framework.

Among other information, the standard RTI-s Runtime Infrastructure Defaults (RID) file contains connectivity specifications of the form:

```
(node, client, type, server)
```

where "client" and "server" specify addresses of a client process and its associated communications server (i.e., associated router) and "type" specifies the communications type (TCP, UDP, etc.). At initialization, each processor finds the RID entry with "client" matching its IP address and then attempts to initialize a communications link of the specified type to the appropriate server. This same mechanism is used to specify the connection of each individual client in Fig.(4) to its associated router.

The tree is created from a text file called the connectivity map which lists, for each host, the host which an application or router should connect to. It is not required that the connectivity map be the same on each host, though in practice it usually is. It is possible to create inconsistent connectivity maps, which do not correspond to a tree. Possible errors are a collection of disjoint trees, or a tree with an extraneous connection where the intended root of the tree erroneously connects to an intermediate node, creating a loop. There is no central global map, which can be checked for consistency. There are no internal checks in the routers or applications to detect and diagnose errors. An interactive program called MARCI implements a GUI to allow an operator to create a global tree. MARCI creates the connectivity map from the virtual tree and distributes the map to all the hosts. In practice, when using only tree routers, the MARCI program is stable and error free, and these errors do not occur.

Connectivity within the individual routers of the mesh configuration of Fig.(4) is defined by a new entry in the RID file, called the "mesh map". The syntax used for this is as follows:

```
(mesh_map
  (mesh "tcp"
    (triad router_address)
    (triad router_address)
     ...
    (triad router_address)
  )
)
```

The individual "triad" lines specify the addresses of the individual routers within the mesh. The initialization procedures automatically generate all inter-router communications links for full mesh communications.

Deployment of a wide-area MeshRouter was initially done during an Urban Resolve practice trial. This was the only opportunity to test it using several hundred hosts at multiple sites, with high data rates. The mesh map gives the host names of the collection of nodes in each mesh. The use of the connectivity map is unchanged, allowing connections other than the mesh connections. The MARCI interface for creating the mesh map was simple. The topology selected for the tests was a mesh of four geographic locations, J9, TEC, SPAWAR and ASC, with the same trees at each site that were normally used without the mesh routers, yielding a mesh of trees. Initial results were negative. There was either no connectivity between some hosts, or, more perplexing, the applications would perform well for a few minutes before the network became saturated with excessive message traffic and the simulation failed to operate. In these cases testing and development of applications and procedures by all the developers was halted due to the router problems.

Investigation revealed that the problems were inconsistent mesh maps and connectivity maps. One problem in particular was that most of the tree router connectivity map was retained for the mesh router tests. It contained entries directing the tree routers at

TEC, SPAWAR and GLENN to connect to the J9 router as clients. When the mesh routers were used these connections were inadvertently left in. This resulted in dual connections from TEC, SPAWAR and ASC to J9, once as clients and once as members of the mesh. The result was that messages would loop and never leave the system, even though they had been delivered to the appropriate client, gradually saturating the network. These errors were diagnosed by examining the maps by hand. A consistency check was put into the mesh routers to never connect as a client to a router, which was a member of the same mesh. A warning is generated, but execution is continued. This and some scrutiny of the MARCI operation allowed subsequent MeshRouter tests to succeed. The MeshRouter now performs as expected and is stable for practice trials and Urban Resolve experiments.

To exploit the MeshRouter, the J9 network team reconfigured the physical routers at each of the participating sites so the network is a fully connected network. In addition to having no single point of failure, this new network configuration exploits greater communication throughput and with the MeshRouter, reduces latency by establishing direct single-hop connectivity between all the geographically distributed centers. For example, a message from ASC, at Wright Patterson AFB OH, is no longer required to visit the J9 router, the head of the tree, before being forwarded to SPAWAR, San Diego.

With its new fully meshed network topology, J9 can expect the following improvements to network traffic and operational capabilities: (1) J9 itself is not a single point of failure, (2) the overarching DREN network is responsible for packet traffic between sites as opposed to a single J9 router, (3) two or more sites can simultaneously exchange data without having a bottleneck at a single site, and (4) the fully meshed network is backward compatible to support the tree-router architecture.

IV: GENERALIZATIONS FOR FAULT-TOLERANT COMMUNICATIONS

The router networks used in the current Urban Resolve experiments are not fault-tolerant in the sense that failures of individual router processes disrupt communications within the global, distributed simulation. For the tree-router architecture of Fig.(2), failures at the root router divide the entire simulation into two non-communicating halves. Failures on a mesh-router triad are a bit more localized (only the simulators directly below that router are lost), but the disruptions are still unacceptable.

We consider here two distinct strategies for generalizing the basic router framework to tolerate individual router failures:

Strategy 1: Dynamic Reconfiguration.
Strategy 2: Shadow Routers

Before exploring these, it is necessary to reconsider and generalize the manner in which communication links are specified and initialized within the current RTI-s.

Communications Directors

We assume in this discussion that an individual application process is "smart enough" to recognize that an individual communications link has failed. In such instances, it is assumed that processors will attempt to reestablish communications, essentially repeating the original communication initialization procedures and then declaring the current processor interest state, as appropriate.

The problem with this procedure (already implemented in JSAF/RTI-s) lies in the essentially static/fixed specification of communications partners. As noted above, the RID file includes specification lines of the form

```
(node, client, type, server).
```

These entries are generated through the standard MARCI software prior to simulation execution. However, these entries do not change during execution, and the RID entry above is of no value to the simulator when the specified router process is the point of failure.

A procedure for specifying dynamic communications partners is needed. One reasonably straightforward procedure would be as follows.

1. The existing, static RID specifications lines above would be replaced by entries of the form:

   ```
   (node, client, type, dir1, dir2)
   ```

2. The address "dir1" is not the address of the client's router but the address of an auxiliary process that will return the needed server address to the client.

3. The address “dir2” is a backup address server to be used by the client in the event of “dir1” failure.

This simple modification of the existing RID syntax is sufficient to support dynamic Client⇔Router associations, as needed in any fault-tolerant router network.

The issue, then, becomes that of specifying the operations in the new “Communications Director” processes specified in the generalized RID connectivity entries. In general, it is required that

1. Individual director processes will be associated with some specified collection of (logical) router processes.
2. Directors will receive/maintain health and status updates from the associated routers.
3. Individual Client⇔Router assignments returned by the director will reflect router ensemble status.
4. The state information maintained in the “primary director” for a set of routers will be replicated in the secondary director, as a guard against director failures.

The following subsections explore two models for the communications directors, as appropriate for the (very different) communications models within components of Fig.(7).

Dynamic Router-Client Reconfiguration

Consider the basic task of associating client processes to triads within a fully connected mesh (e.g., simulators and triads all within a single SPP as in the schematic of Fig.(4)). The triad/router nodes are (trivially) aware of already connected clients. More generally, it can be assumed that each triad knows not only its number of attached clients but can also estimate some sort of load factor associated with servicing those clients. A typical load factor would be the total number of bytes transferred to and from all lower clients within the past N seconds.

Assuming that the director process receives and maintains the load rates from all (active) triads, the simplest procedure for the director on receipt of a new connection request would be:

Assign the newly received communications request to the least-loaded triad within the system.

There are several obvious advantages to this procedure:

1. No backup router triads (“hot spares”) are required. Clients formerly associated with a now broken/lost triad are simply redistributed to the remaining, healthy triads.
2. The procedure provides a simple mechanism for dynamic balancing of the communications load. For example, an overloaded triad could simply break connections to some of its clients, letting the director subsequently reassign the clients to less-loaded triads.

The director processed in this scheme must be moderately “intelligent”, receiving and maintaining health and load information from all associated triads.

Shadow Routers

The dynamic client reassignment model just described is quite feasible for communications within a single SPP or LAN since all point-to-point communications within such systems are (largely) equivalent. However, this model is not applicable for wide area communications networks such as the “Transcontinental Mesh” system of Fig.(7). The reasons are fairly obvious:

1. There are, in general, only a limited number of physical communication links available between/among geographically distinct sites (particularly when secure communications are required).
2. An SPP has only a limited number of associated processors that can be accessed from remote (i.e., non-SPP) processors.

In these situations, the nominal Site⇔Triad associations of the nominal network must be maintained. The easiest implementation of fault-tolerant communications in such cases comes through replicated/redundant triads at each site.

In the example of Fig.(7), this would involve maintaining two separate/dedicated router triads at each site in the transcontinental mesh – the nominal triad process and a backup triad that remains unused unless/until the primary triad processor fails.

The model for the director process in this case is fairly trivial: The director maintains health/status information on both the primary and backup routers. Communications are shifted to the backup triad only when the primary triad fails.

Remarks

The strategy for creating a fault tolerant network will likely involve both dynamic reconfiguration as well as shadow routers. As discussed above, each has its place. The initial challenge will involve extending RTI-s to recognize directors and shadow routers. In addition, when a client has to reconnect following the loss of its router, it will also have to declare its entire interest space. MARCI will then have to be enhanced to incorporate them. Once these changes are made, J9 will have a practical level of fault tolerance in its communication network. The "fire and forget" message passing paradigm used within RTI means there will be packet losses when routers fail, but the number should be small enough that an experiment can continue.

V: CONCLUDING REMARKS

This paper has described the successful integration of the MeshRouter to support interest-filtered communication in JFCOM J9's Urban Resolve experiments. Once the initial configuration errors had been corrected, operation of the routers was indistinguishable from the previous communication network J9 had deployed. The MeshRouter will enable J9 to maximize its use of network bandwidth while simultaneously reducing communication latency among the geographically distributed participants. Both the MeshRouter and J9's earlier "tree" network are now available for J9 to use at its discretion.

Looking to the future, the authors believe that the next significant improvement in J9's communication network should be to make it fault tolerant. This would minimize the loss of Urban Resolve experiment time due to the failure of routers, or the computers they run on. A strategy of dynamically reconfiguring router-client pairs appears suitable for local area networks and SPPs, where the networks are relatively homogenous. Shadow routers appear to be a better solution for wide-area networks, where geographically distributed routers cannot be reconfigured without introducing significant additional network latency and bandwidth.

ACKNOWLEDGEMENTS

We thank the Joint Forces Command (JFCOM) and Joint Experimentation Directorate (J9) for support and encouragement during this work. This material is based on research sponsored by the Air Force Research Laboratory under agreement number FA8750-05-2-0204. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government.

REFERENCES

Amburn, P. & Gottschalk, T. (2005), Extending the MeshRouter Framework for Distributed Simulations, *Proceedings of the 2005 I/ITSEC Conference*, Orlando Florida.

Barrett, B. & Gottschalk, T. (2004a), Advanced Message Routing for Scalable Distributed Simulations, *Proceedings of the 2004 I/ITSEC Conference*, Orlando Florida.

Barrett, B. and Gottschalk, T. (2004b), Pipes and Connections, *Caltech/CACR Technical Report 2004.213*

Brunett, S. & Gottschalk, T. (1998) A Large-Scale Metacomputing Framework for the ModSAF Real-time Simulation, *Parallel Computing* 24.

Ceranowicz, A., Torpey, M., Hellfinstine, W., Evans, J. & Hines, J. (2002), Reflections on Building the Joint Experimentation Federation, Proceedings of the 2002 *I/ITSEC Conference*, Orlando Florida.

Gottschalk, T. (2004a), The MeshRouter Architecture, *Caltech/CACR Technical Report 2004.212*.

Gottschalk, T. (2004b), MeshRouter Primitives: Messages, Interest, and Interpreters, *Caltech/CACR Technical Report 2004.214*.

Graebener, R., Rafuse, G., Miller, R., & Yao, K.-T. (2003), The Road to Successful Joint Experimentation Starts at the Data Collection Trail, *Proceedings of the 2003 I/ITSEC Conference*, Orlando Florida.

Lucas R., & Davis, D. (2003) Joint Experimentation on Scalable Parallel Processors (JESPP), *Proceedings of the 2003 I/ITSEC Conference*, Orlando Florida.

Messina, P., Brunett, S., Davis, D., & Gottschalk, T. (1998), Distributed Interactive Simulations for Synthetic Forces, Proceedings of the International *Parallel Processing Symposium*, Geneva, Switzerland.

Synthetic Theater of War (STOW 1996) Engineering Document 1A (ED-1A) Analysis Report, *NRaD Report*, 1996