I: Administrative

Proposal to ITO, the Defense Advanced Research Projects Agency

BAA 99-06: Extensible Networks and the Next Generation Internet
Technical Topic Area: Deeply Networked Systems

SEEDS: Simulation and Emulation Environment for Deeply Distributed Systems

by Information Sciences Institute
University of Southern California
Los Angeles, California 90089-1147

ISI Proposal 98-ISI-094

ISI Technical Contact: John Heidemann
(310) 822–1511 x708, fax: (310) 823–6714, johnh@isi.edu

ISI Administrative Contact: Margie Schroeder
(310) 822–1511, fax: (310) 823–6714, margiej@isi.edu

Summary of costs:
Y1 cost: $809,306
Y2 cost: $834,971
Y3 cost: $866,835
total: $2,511,113

Type of Business: other educational

Principal Investigators:
John Heidemann, research assistant professor, USC/ISI
Deborah Estrin, professor, USC

Approved for USC by

Lloyd Armstrong, Jr.
Provost and Senior Vice President for Academic Affairs
University of Southern California

The proposed work will be done jointly with personnel at University of California, Berkeley. They are submitting an identical technical proposal with technical contact Kevin Fall (kfall@cs.berkeley.edu).

II: Detailed Proposal Information

II.A. Innovative Claims

While traditional networking research has concentrated on the problems of communication between computers for the purposes of human interaction or distributed computation, new challenges are posed by the need for thousands of resource-constrained small systems to communicate and collaborate in a dynamically changing environment under substantial resource constraints. Development and understanding of these deeply networked systems requires both innovation and integration across several technologies: wireless networks, power management, mobile networking protocols, sensors, and algorithm design. Successful investigation of these areas requires the ability to rapidly explore and test the applicability of novel computational and communication methods involving a wide range of environments and resources. We claim that network *simulation* and *emulation* (interfacing simulation with live networks) provides a fundamentally needed tool in the exploration of this area, and that current simulation technology falls short in addressing this diverse set of needs. Furthermore, we believe DARPA must invest in a simulation infrastructure early enough to promote sharing of models and algorithms within the developing deep networking research community.

Current network simulation technology comprises both commercial and freely-available simulators supporting abstract network links and nodes, and in some cases wireless communications or multicast protocols. They generally lack the ability to model physical characteristics and constraints such as position, power levels, or sensor ranges, or they lack a rich set of Internet protocols. They also lack support for promising new coordination approaches such as data diffusion. While SEEDS will provide a basis for simulations involving these broad range of areas, we believe no single institution can provide complete coverage. Rather, we suggest the creation of an extensible *common* simulation platform to promote collaboration across all levels. SEEDS will approach this goal by extending the VINT/ns simulator, already in wide use for collaborative Internet research.

Deep networking and sensor networks are expected to contain a heterogeneous set of clusters of simple, homogeneous nodes. Simulations and experiments of such networks inherently require large numbers of nodes, physical distances, and time-scales. SEEDS will take advantage of the inherent structure of these networks, to enable large-scale parallel simulations over clustered computers through automatic problem partitioning. Emulation will allow us to extend the repeatability and automation of simulation into large-scale experiments mixing real-world and virtual (simulated) nodes.

In summary, SEEDS will provide the following innovations to current simulation technology:

- **Support for deep networking and sensor nets:** Physical and logical *location support* for nodes supporting service discovery and query routing. *Resource-aware networking*, including models of CPU, power, and radio limitations. A framework for simulation of *sensors and actuators*, including validated models of key sensors. *Internet protocols*, both standard and experimental.
- **Research in autonomous collaborative problem solving:** Libraries for data diffusion, a promising approach to collaborative algorithm design employing attribute-named data, caching, and gradient routing.
- **Emulation for controlled real-world experimentation and validation:** Transducers to convert between simulated and real-world traffic, soft real-time distributed scheduling to carry out distributed emulations with real-time delays.
- **Scalable simulation and experimentation with domain-specific resource allocation:** Algorithms for mapping problems in deep networking to cluster computers. Distributed, fault-tolerant simulations and emulations at large scale.
- **Hierarchical visualization of simulated and real-world phenomena:** Interactive visual debugging and evaluation of simulated and real-world traffic.

II.B. Technical Rationale, Approach and Plan

II.B.1. Challenges of Deeply Networked Systems

The proliferation of deeply networked systems has the power to change the physical world we live in through the power of mobile collaborative sensing and actuation. The enabling hardware technologies for this revolution are upon us: very inexpensive processors with enough power to run Internet-compatible networking and coordination protocols, small sensors covering a large range of physical properties, micro-actuators and MEMS technology, and position determination infrastructure (like GPS). Despite these advances, significant work remains in defining and understanding the operation of algorithms and network protocols that utilize these technologies. As an example, wireless communication protocols, distributed algorithms for coordination, and automatic network configuration must operate under restricted conditions such as limited communication availability and node power. Many of these new challenges are brought on by the requirements inherent in deeply networked systems: large numbers of autonomous networked nodes (1,000s to 10,000s per smart room or sensor net with loosely coupled aggregations of 100s of nets), diverse physical environments (some wired, many wireless, many limited by battery power), and the need for collaboration among nodes due to limited sensing and communication range and bandwidth.

These promises and challenges make it clear that substantial new research is required. Furthermore, the breadth of work implies that no single organization will be able to meet all the needs of this agenda. Prior experience in Internet protocol research shows that the first few years of research in a new area can be relatively inefficient as multiple organizations struggle with underdeveloped tools and then compare developments made in disparate environments. SEEDS proposes a *simulation and emulation environment* which can avoid this early inefficiency and duplication by providing robust tools for experimentation, collaboration, and comparison.

SEEDS will provide innovation in four key areas: deep networking simulation, emulation, scaling, and visualization. A common simulation environment allows individual researchers to rapidly explore algorithm and protocol design space while promoting collaboration across different levels of the problem (for example, independent researchers on diffusion algorithms and radio modems) and allowing comparisons at any given level (for example, comparing diffusion algorithms to traditional resource discovery). Emulation, with mixtures of real and virtual devices, is key to extending these benefits to real-world experiments—a requirement in this domain since experiments must traverse the scales required of deeply networked systems. Scaling is needed in order to help predict the overall behavior of many small systems interacting or interfering with one another. Finally, visualization, monitoring, and debugging tools are crucial in helping to understand both expected and unintended dynamic protocol and algorithm behaviors.

Our approach to the challenges of deeply networked systems is to provide a simulation environment that couples a rich Internet-style network simulator with the integrated support for physical characteristics needed by deeply networked systems research. In particular, we plan to build upon the successful VINT/ns simulator, which was designed for extensibility and advanced Internet protocol support. The resulting platform is expected to be useful for both deeply networked systems, and for other explorations into heterogeneous networking consistent with the DARPA NGI and SenseIT programs. This assertion is supported by the unplanned diverse applications of the ns simulator, which has been used for simulations including high-speed optical switching [35] and satellite networking [38].

II.B.2. Investigations in Deeply Networked Systems

SEEDS will provide unique capabilities for the simulation and emulation of deeply networked systems. To better understand the usefulness of these capabilities, we consider several areas of investigation that will be facilitated or significantly enhanced by SEEDS.

Tuning Algorithms With Very Large Parameter Spaces: Understanding the performance of auto-tuning, self-organizing systems of thousands of communicating nodes presents a need to explore a huge parameter space. The only possible way to explore this space and reach common understanding is through a range of simulations conducted by many different researchers.

As an example, directed diffusion will require investigation across a broad range of topologies and variants. Consider the issues raised: basic algorithms send requests in some limited broadcast, caching will likely occur in the network, and data transfer will reinforce gradients of interest. Specific questions to be answered include the definition of the methods of limited broadcast (e.g., in a wireless environment), the locus of caching and how caches are flushed and filled, the protocols used for data transfer, and the algorithms for determining interest gradients. SEEDS will bring to this investigation the network support for data migration and caching, data transfer protocols, gradient models, and examinations of diffusion algorithms. By providing well documented interfaces and a modular design, other researchers will be able to integrate their own caching and gradient determination algorithms.

While the task of developing new algorithms and the simulation infrastructure to support them may seem daunting, we have already gained valuable experience in earlier efforts using the VINT/ns simulator. Reliable Multicast protocols are an example where the freely available VINT/ns simulation infrastructure helped to nurture a very broad understanding of protocol behavior, from the original studies on simple topologies [13] to studies in the limiting cases of artificial topologies [29], changing topologies [37], and studies of variants and extensions [15, 23, 33]. By incorporating the physical characteristics of deeply networked systems, similar shared understanding and experimentation can be fostered in this new important area.

Selecting Which Prototypes to Build: Simulation alone is not sufficient for the broad development and deployment of deeply networked systems. Non-ideal physical properties such as sensor limitations, intermittent communication, and power levels can render simulation results useless for real-world application. Not all models can be validated and implemented with the degree of detail present in a live experiment. Even with limited models, SEEDS can provide a valuable service in reducing the search space by suggesting which prototypes are most promising to build.

Consider the example of adaptive fidelity in deep networking. We would like to design data acquisition algorithms where better results (increased fidelity) can be achieved simply by increasing the number of sensors, or where a given required level of service can be met over a longer duration by selectively enabling power-constrained sensors. Understanding the available options depends on sensor capabilities, power draws, node placement, and many other factors. Some of these are difficult to reproduce in simulation, yet even rough simulations present an important first step in understanding the salient issues and can often suggest which approaches appear most fruitful, encouraging *focused* prototyping of the two or three top designs.

Enabling Very Large Experiments: In addition to the large parameter space of individual deep-networking investigations, many such experiments are very important but very difficult to carry out. The following issues are raised when trying to design and develop deeply networked systems using real-world nodes:

- Experiments with large numbers of nodes can be logistically challenging to orchestrate and may be economically infeasible to construct.
- Multi-hop wireless networking requires testbeds several times the range of the wireless communication medium. As products with long ranges are incorporated, physical distance becomes a matter of practical importance.
- Power-conserving protocols are clearly important, but pushing the limits of power constraints can require hours or days of communication.

Simulation based on validated models and emulation-augmented experimentation are the only way to make these experiments tractable. With emulation, large numbers of nodes can be moved into simulation, reducing or eliminating the need to construct large scale testbeds.

Comparing Alternate Approaches: Deeply networked systems represent a set of new constraints and opportunities for designing future communication and computation methods. In particular, resource discovery and management are key requirements at all levels, from signal modulation to distributed learning algorithms. We expect a number of very different approaches will be considered:

- At the highest level, simulation can provide a natural ground to compare diffusion-based algorithms against optimization models based on economics, directory-based, or client/server-based approaches. Researchers will consider questions about query responsiveness, algorithm complexity, and protocol stability for each of these broad approaches in the context of a common environment of test scenarios. SEEDS will contribute to these protocol studies by providing this common environment and libraries of test scenarios.
- Many new protocols such as diffusion routing promise highly reliable delivery and robust configuration and fault tolerance, but at the cost of less constrained use of bandwidth as compared to a more static, less adaptive approach. Diffusion protocol studies will want to quantify the cost of adaptivity by comparing the additional bandwidth uses. SEEDS simplifies this effort through the common environment mentioned above and also with its existing library of traditional Internet protocols (TCP, multicast, routing, etc.).
- Mobile ad hoc routing has gathered significant momentum in recent years. As an example, the IETF working group on mobile ad hoc networks (MANET) is investigating at least seven unicast routing protocols (ZRP, TORA, DSR, CBRP, CEDAR, OLSRP) and three multicast-capable routing protocols (AMRoute, AODV, ODMRP). The large number of contending protocols cover a wide range of operational characteristics (for example, distance vector versus link state or other class of routing algorithm, use of source routing, need for reliable packet delivery, support of unidirectional links, etc.) The working group is currently beginning to consider the adoption of a common simulation framework, but this choice is made difficult by the lack of a single simulator combining advanced Internet protocols with real-world location, motion, and impaired connectivity dynamics. SEEDS will provide an attractive alternative to the parallel development of individual simulators in this area.

Multi-layer Interactions: Deeply networked systems and sensor networks will require research efforts at many layers of the problem: from collaboration algorithms and networking protocols to wireless communication hardware and sensor capabilities. Protocol studies will be undertaken to understand the interactions between these layers. For example:

- Interactions between transport- and link-layer data retransmission. Poorly done link-layer retransmission can add latency and reduce bandwidth, while a detailed understanding allows performance optimizations such as the Berkeley Snoop protocol [5].
- In deeply networked systems we expect interactions between all layers of the system to effect major design choices. For example, current protocols for reliable multicast depend on broadcasting control messages to all participants to suppress duplicates. In a multi-hop wireless network this approach may have much higher power requirements than other approaches such as router-supported multicast.

The vertical integration introduced in deeply networked systems and sensor networks within the real world make multi-layer protocol interactions increasingly important. SEEDS will address this need by supporting modular interfaces between layers, allowing early simple models at one layer to later be replaced with more accurate models. Researchers will be able to contribute to the models in areas where they are expert and re-use models at other layers. This collaboration allows increased confidence in model accuracy because of wide reuse and broad benefit as modules are vigorously validated.

Understanding Complex Protocol Behavior: There is a continuing need to gain more insight into the behavior of protocols and protocol interactions as they scale up. With the VINT network animator we have found visualizations offer insight in many situations:

debugging: visual display of protocol behavior early in development substantially eases debugging (examples include debugging SRM, layered multicast, rate-based pacing in TCP, web caching consistency)

quickly interpreting substantial data: color-animated network maps have proven helpful in understanding loss conditions in the multicast backbone (mbone)

detecting unusual occurrences in a large dataset: event/time maps are helpful to understanding timer behavior in SRM

correlating algorithm behavior with topology: graphical protocol representations are ideal for understanding how topology (and delay, bandwidth, and loss) affect algorithm behavior and performance. Examples include recovery behavior in reliable multicast protocols.

These approaches are going to be even more important in developing and understanding deep networking because they will require large numbers of interacting nodes and will employ radically new coordination techniques.

II.B.3. Simulation

The core of SEEDS is an advanced platform for simulating the operation of deeply networked systems using Internet-compatible protocols. Simulation provides a critical complement to implementation by supporting a more rapid exploration of the intended operational environment, a common basis for the comparison of research results, and a repeatable and automated testing capability. In addition, by augmenting simulation with emulation, many of the benefits of simulation can be applied to real-world implementations.

Deeply networked systems will stress existing protocols and algorithms by introducing new parameters relating to the physical world: nodes have physical and logical locations, sensors have detection ranges, connectivity is subject to interference, and nodes have power and computation constraints. Rapid experimentation with new protocols will require simulators supporting these concepts. The large number of physical operating parameters makes simulation prior to prototype construction particularly useful.

In considering a wireless, battery-operated sensing node, for example, a simulation would have to take into account the power to transmit and receive signals, transmission and reception ranges (as a function of power), the battery drain rate, maximum charge and minimum operating charge level, current operating environment and sensor capability. Many of these characteristics would not only require a model to drive the simulator, but may also be needed by algorithms as input parameters. In such cases, interfaces (e.g. the SMBus specification [14]) are required both in simulation and in prototypes.

Another example concerns the issue of security in deeply networked wireless systems with CPU and memory limitations. Such systems are especially vulnerable to subversion due to their low power levels and limited computational resources. Employing encryption on such devices introduces a tradeoff between the level of security (i.e., algorithmic complexity), CPU computation, and power utilization. Exploration of this tradeoff in simulation provides useful experience for designing secure remote sensors.

Existing network simulators have good support for current and experimental protocols in resource-rich wired environments, but exclude nearly all of these physical constraints mentioned above. Simulators that are constructed to examine system behavior with respect to one of these physical effects typically lack needed support for advanced networking protocols, making it extremely difficult to predict system behavior in a multi-protocol Internet-like environment. For deep networking to progress, the key SEEDS research challenge will be to provide a simulator providing *building blocks* for deeply networked systems, *libraries of common scenarios*, and *validated models* of key protocols.

Building blocks for deeply networked systems: The physical properties associated with deeply networked systems include physical location and movement, resource constraints, and physical detection and actuation capabilities. Physical location and highly accurate timing is now made practical by the deployment of GPS, and efforts are underway to design protocols to extend similar capabilities to nodes lacking the power or reception capabilities needed for GPS. Finally, equally important for indoor research is the notion of logical location, the room or area that a node occupies. Existing simulators' notions of location are quite simple by comparison and offer little support for these parameters.

Velocity, acceleration, and position (including altitude) provide sufficient information for locating a mobile node. However, these parameters do not capture the orientation of nodes. A set of orientation parameters commonly found in distributed interactive simulation such as bank, pitch, and yaw are needed by some simulations, yet are ordinarily not found even in simulators supporting wireless networking. As an example, these parameters have an important effect on nodes equipped with antennas and influence the reception and generation of RF signals. In addition, node orientation can affect the accuracy of sensors in measuring physical phenomena and the efficacy of actuators. SEEDS will include an extensible framework that includes movement models of nodes incorporating these characteristics. The interfaces to these models will be easily used by other models developed outside the SEEDS effort.

Equally important with node location and movement, the resource constraints of deeply networked systems must be reflected in simulation. Examples include limitations on CPU, memory, power, and connectivity. These limitations bring new challenges to the development of optimization algorithms which have traditionally made inherent assumptions about homogeneous capabilities. SEEDS will provide a framework to incorporate models of CPU capability, memory utilization, and power consumption. It will also provide models for common CPU/memory/battery combinations.

Remote sensing applications employing deeply networked systems are anticipated to include a number of physical sensors. Physical characteristics amenable to sensor technology may include sound, vibration, heat, light, motion, temperature, humidity, radiation, chemicals, and even taste [22]. While

SEEDS cannot develop direct models for all of this diverse set of properties, we will provide an extensible infrastructure where models developed by other researchers may by incorporated into the simulation and emulation framework. Included in this framework will be an interface describing the power utilization and effective range for various sensors and models of a few common sensors to exercise and demonstrate this capability.

Coordination of sensor nodes represents a new challenge to system and network protocol designers. In particular, the need to discover computational, power, and sensing resources under adverse conditions suggests that algorithms targeted at autonomous, distributed operation which do not strive to compute a global optimum are more appropriate than cooperative algorithms aimed at computing a global optimal solution and assuming symmetric communication and limitless computational resources. Several phenomena have been suggested as a basis for development of such algorithms, including economics (market economies), physical systems (diffusion), living systems (genetic algorithms and neural networks), and mathematics (stochastic and fuzzy feedback/control systems). SEEDS will address the exploration of these algorithms by providing primitives for time-sensitive events, data cache management, and physically-limited data transmission and sensing. Data diffusion algorithms show particular promise as a model for coordinating deeply networked nodes and sensor nets. We will provide explicit support for these algorithms, including support for attribute-named data and attribute matching. Data aggregation will be required of power conserving nodes; we will include upcalls needed to allow nodes to initiate aggregation in response to congestion.

Scenario libraries: Simulation *scenarios* provide a common set of simulation configurations used for repeated experimentation. While Internet-style scenarios have been developed in the context of the VINT/ns and other projects, these scenarios do not capture the physical properties of deeply networked systems. SEEDS will develop an evolving set of simulation scenarios incorporating physical properties which may be reproduced in simulation or the real world. These libraries will include movement models where appropriate.

Validated models: With few exceptions, simulations are only useful to the extent they provide insight into some real-world phenomena. A thorough understanding of what behavior is captured (and excluded) in simulator models is important to gauge the applicability of a model to the real world. The VINT/ns effort has made extensive use of expert evaluation to ensure that simulation models are appropriate for some aspects of real-world behavior. In SEEDS, we will continue this direction and couple it with quantitative comparisons of simulation against real world network traces where possible and appropriate. In addition, SEEDS will institute a repository of model descriptions describing the applicability limits of each supported model.

Even validated simulators may begin to diverge when extended with new functionality into new operating domains. Simple maintenance of a large simulator alone poses a substantial software engineering problem. VINT/ns currently includes nearly 100,000 lines of code in the core, and is twice as large when examples and support code are included. We have instituted an automated regression testing and validation procedure with the VINT/ns code base and intend to continue and refine these procedures for the SEEDS effort.

II.B.4. Emulation

Simulation provides the ability to evaluate and develop experimental systems in a controlled environment where details deemed unnecessary may be eliminated by careful choice of abstraction. To validate the experience gained with simulated systems, an increase in detail provided by live implementation is

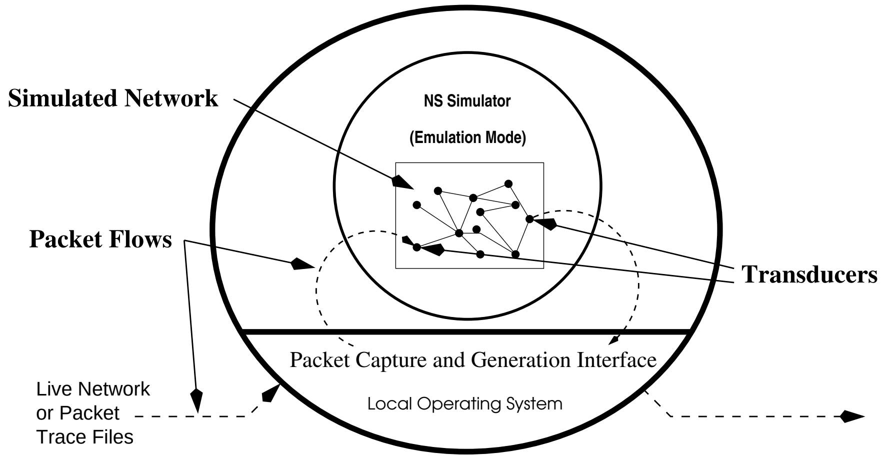

Figure 1: Interaction between network traffic and a simulated network through emulation.

often necessary in order to better judge the appropriateness of introducing new systems into the real world. Through *emulation*, we can extend the benefits of simulation to the physical world, allowing real-world devices to be examined with the speed, control, and reproducibility previously available only in a simulation. Combined with tracing facilities, emulation can be used to capture and replay the traffic generated by either simulated or real-world scenarios. Furthermore, emulation represents the only plausible approach to enable large-scale experiments critical to evaluating systems for deeply networked systems early in their development cycle.

A network emulator is a hybrid between a network simulator and a real-world protocol implementation. In an emulator, software "transducers" within the simulator translate between standard network protocol packets (from either trace files or a live network) and those used within the simulator, enabling simulated protocol modules to interact with standard implementations. Figure 1 shows this interaction. Traffic can cross the simulator from one transducer to another, or it can originate or terminate in a protocol module within the simulator. It can be traced and replayed using either standard network monitoring tools or simulator tracing facilities.

Emulation attempts to address the deficiencies of simulation while retaining its strengths. While simulation may lead to false conclusions due to unintended abstraction of operational details, it has the desirable properties of repeatability and ease of configuration. Emulation capability is thus important in fostering the development of deeply networked systems because it allows a simulator framework to be used for:

- Evaluating correctness of protocol implementations and stressing proposed algorithms.
- Extending the scale of real-world experiments with partially virtual scenarios.
- Replacing a portion of the network between real world nodes.
- Supporting large-scale simulation via parallel federated simulations over a network of workstations.

Figure 2 demonstrates each of these cases.

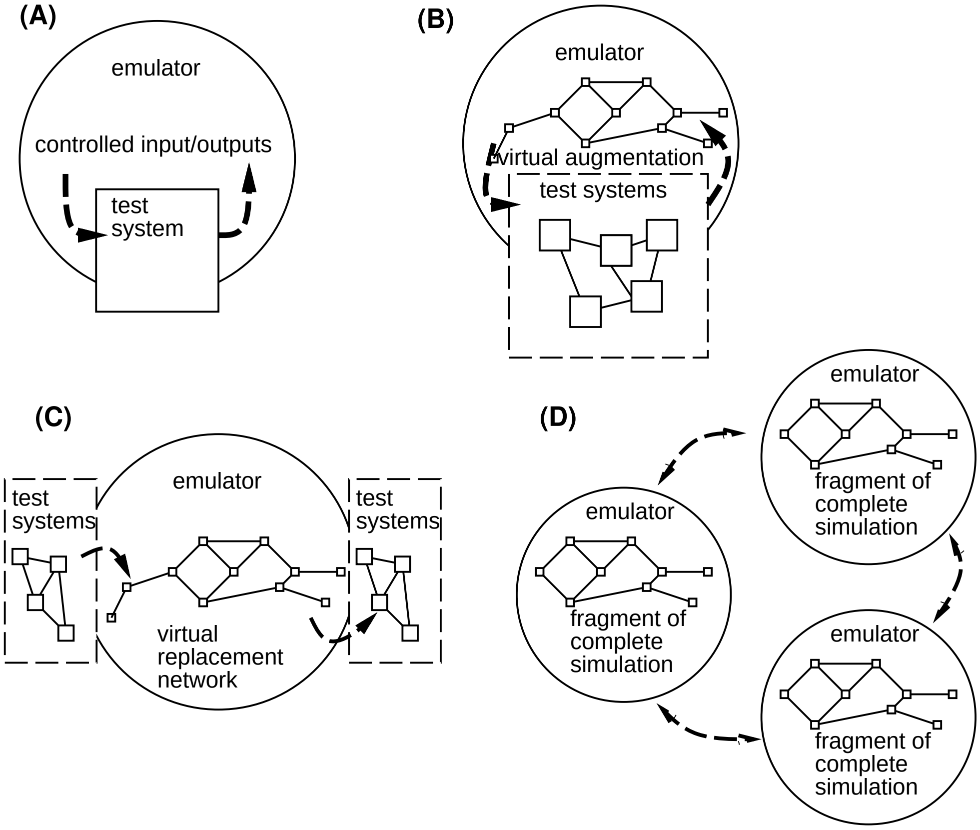

Figure 2: Possible uses of the SEEDS emulation facility.

Implementation Validation via Emulation: A great need exists for tools to aid in verifying the correctness of protocol implementations. As an example, the venerable TCP protocol has been implemented by many commercial vendors, yet many of the implementations contain bugs or other errors [26, 11]. These problems arise not only from simple implementation errors, but also from protocol under-specification, unexpected interactions of multiple protocols, or the aggregation of many protocol instances. Robustness in the face of large scale relies on correct specification and implementation of network protocols. By directing network traffic through an emulator, it may be manipulated to induce otherwise unrevealed behavior in end-node protocol implementations (Figure 2 parts (A) and (B).) By passing real world traffic through a simulator, SEEDS allows very sophisticated but controlled experiments. For example, while other systems such as NISTNet [9] and DummyNet [30] can subject traffic to controlled delay and losses generated by simple models, SEEDS emulation can subject real-world traffic to simulated cross-traffic. This innovation is critical to accurate experiments because simulated cross-traffic will be reactive to congestion and loss.

Stressing algorithms, both in implementations and under simulation, is uniquely enabled by emulation. For example, burst packet losses, re-ordering, duplication, and congestion created by traffic internal to the simulator can act upon traffic routed through it in order to trigger and evaluate the congestion behavior of real-world implementations. Using simulated traffic sources to affect real-world traffic flows provides the ability to evaluate the robustness of protocol implementations when faced with newly proposed traffic sources that have yet to be rendered to implementation (Figure 2 parts (B) and (C)). In addition, traffic generated by real-world sources are not easily characterized by simple models [27], making the evaluation of new packet scheduling and buffer management algorithms more difficult.

Expanding the Scale of Live Experiments: The physical constraints of real world devices makes testing extraordinarily difficult in deeply networked systems. Whereas experiments on Internet protocols like TCP required a few nodes, and multicast requires 10s to 100s of nodes, scaling deep networking is difficult in at least two dimensions:

large numbers of nodes Hundreds or thousands of nodes is a defining characteristic of deeply networked systems, but testing experimental algorithms with that many physical nodes is nearly impossible because of the labor required to deploy and monitor each experimental apparatus. Furthermore, even modest changes to the test environment may induce a massive effort in reconfiguration.

physics of communication Wireless networking and power-constrained networking will play a large role in deeply networked systems, but robust protocols require a detailed understanding of boundary conditions. Even short-range (20m) wireless protocols limit experiments to a few wireless hops before experiments become physically too large to perform in a building or reasonably sized area. Similarly, we must understand power requirements for systems with day- or week-long battery lives while conducting experiments that conclude in a reasonable amount of time.

We will address these issues by using emulation to allow experiments to seamlessly mix virtual and real nodes. Emulation can either augment an existing testbed (Figure 2 part (B)) or actually replace portions of a network (Figure 2 part (C)).

Specific Emulation Requirements and Approach: Work in the area of emulation has generally proceeded along two paths: *network emulation* and *environment emulation*. Network emulation aims to allow simulated components to communicate with protocol implementations in the real-world. Environment emulation extends the model further in an attempt to build an implementation environment in which a real-world protocol implementation may be executed directly within the simulator. Environment emulation generally requires the emulator to implement an environment specific to a particular operating system. In SEEDS, we intend to pursue a network emulation approach, due to its operating system independence, compatibility with the existing VINT/ns framework, and because the environment emulation approach is being pursued by other researchers in a commercial venture [18].

Preliminary support for emulation exists the VINT/ns simulator [4]. Ns includes primarily low-level network functions (packet filters based on the Berkeley Packet Filter [24], ARP, ICMP, raw IP, etc.) and a method for capturing and generating live traffic. SEEDS will substantially expand these services to include at least TCP and deep networking protocols such as data diffusion. It will ensure that the implementation is scalable and will characterize its performance under load. Simulator protocols for TCP, multicast, and directed diffusion will be enhanced so that they can serve as end-points for external (real-world) traffic. Transducers for these protocols must be provided to map between their real-world and simulation-world packet formats.

These facilities enable protocol validation via emulation. To demonstrate this innovative ability to the community and fully develop this capability, we will demonstrate validation of key algorithms in one or more protocols such as TCP or directed diffusion.

To support scalable emulation we will enhance the ns real-time scheduler to coordinate multiple parallel simulators and real nodes. When interacting with real nodes, these schedulers must track real-time. When serving as glue between parallel simulation they must efficiently skip over periods where there is no interaction.

Finally, we will enhance emulation to demonstrate tests with mixes of hundreds of simulated and a small number (dozens) of real nodes. The potential utility of a scalable emulation environment is enormous. As a partial list of benefits, it could be employed to: evaluate the interaction of many concurrent network flows generated by live protocol implementations, investigate the performance of mobile ad-hoc routing, etc.

II.B.5. Large Scale Simulation and Emulation

Interaction with the physical world is the defining characteristic of deeply networked systems, and physical limitations of sensor and actuator ranges imply very large numbers of interacting nodes to cover a large geographical region. SEEDS will pursue three complementary approaches to push the scale of simulation and real-world experimentation: parallelism, use of abstraction, and efficient implementation techniques. We will provide for large simulation sizes by optimizing simulation behavior for large numbers of homogeneous nodes and with coarse-grained parallel simulation using a specialized cluster computer. We will also use these approaches with emulation in supporting mixed experiments of physical and virtual nodes to expand real-world experiment size.

Parallelism Deeply networked systems are characterized by a set or cluster of homogeneous systems cooperatively acting to solve a common problem. They may be completely autonomous, or driven by a centralized manager. The interconnection of clusters (e.g. via the Internet) forms a large heterogeneous system of homogeneous components. We expect the scale of the homogeneous components to number in the thousands or tens of thousands with the entire system comprising millions of nodes. The need for a parallel simulation infrastructure is immediately apparent at such large scale, yet most parallel simulators do not generally scale to this large size. Emulation exacerbates the problem because it requires the overall distributed simulation to execute in near real time.

Given the specialized nature of deeply network systems, where nodes are expected to be weakly connected and conservative in their use of resources, the achievable scale of detailed (i.e. packet-level) simulations and emulations can be significantly enhanced through careful assignment of simulation components to processors. The Berkeley Millennium project [7] is constructing a campus-wide computing infrastructure comprising a cluster of clusters interconnected by a high-speed, low-latency network. Individual clusters contain a combination of uniprocessors and multiprocessors running a variety of operating systems. SEEDS will leverage this distributed computational resource to carry out parallel emulations capable of scaling to thousands of nodes on conventional (non-real-time) operating systems (Figure 2 part (D)). While this approach does not guarantee deterministic real-time response, it is capable of approximating real-time as required by most emulation scenarios. The hierarchy and clustering present in network topologies suggest a natural approach to decompose large simulations for coarse-grained parallel simulation. We will develop tools to construct and control these simulations in NOW environments.

Abstraction Protocol abstraction has proven very successful at providing two orders of magnitude scaling improvement in VINT/ns [17]. Detailed ns scenarios scale to 100s of nodes on a single machine (one CPU, 512MB memory), while abstract simulations reach 20,000 nodes. There we developed a methodology for discovery and evaluation of protocol abstractions:

1. Identify key bottleneck. For example, memory usage due to a large number of concurrent multicast receivers.

2. Design a protocol abstraction which addresses that bottleneck. For example, centralize handling of multicast routing.

3. Validate the abstraction at small scales by comparing detailed and abstract protocol representations in the same simulator. Quantify abstraction accuracy and document regions of inaccuracy. For example, centralized multicast has different transient behavior as nodes join, but steady state behavior is not affected. Important application metrics (such as SRM repair time) are not statistically altered.

4. Perform large scale simulations possible only via abstraction

Efficient Implementation Techniques An emphasis of VINT/ns simulation has been fine-grained protocol composition. While ideal for small scale experiments and when considering protocol variations, this approach is not optimal when scaling simulations up to thousands or millions of nodes. We will develop new simulation techniques which leverage this homogeneity. Promising approaches here include selecting protocol representations which minimize the amount of duplicated state and abstracting away protocol details not critical to large scale simulations—we will look at both simulator implementation and algorithms. As an example of the savings that can come from reviewing protocol representation in VINT/ns we found enormous state duplication in configuration support. Changing this representation resulted in a savings of 1/3 of simulation memory for 200-node TCP simulations, allowing 50% larger simulations on a given machine. Similar opportunities will occur in deep networking. For example, replacing duplicate representations of attributes for request diffusion with references.

We plan to apply these approaches to key deep networking components. A change of representation using abstraction enables the replacement of complex protocol state with a simple finite-state automata, minimizing memory and processing requirements. We will use the detailed simulator to evaluate full protocol state (for example, TCP congestion window behavior) for sample inputs. A full state-table expansion would be enormous, but judicious use of approximation will allow similar state to be collapsed and merged. We can then validate this approach at small scale and use it for very-large scale.

II.B.6. Interpreting Results

Understanding the results of network protocol experiments can be difficult: mountains of data can be generated without capturing the important components. In addition, simple tools such as time-event plots may fail to highlight dynamic interactions. For these reasons, visualization using animations represents the preferred means to help in reasoning about experimental results. During protocol development, visualization can dramatically ease protocol debugging and understanding. Consider identifying errors in a TCP fast-retransmit algorithm with packet logs, protocol graphs, or packet animations (Figure 3). Graphs and animations greatly simplify the task. Visualization is also key to interpreting large amounts of data to understand protocol behavior and detect anomalies. The eye can easily watch the flow of hundreds of concurrent packets and through color markers identify their types and functions; the same analysis would be impossible using packet logs.

Although visualization represents an important tool for analysis, post-experiment protocol animations do not generalize to the scales required for deep networking where thousands of nodes, each with dozens of parameters, interact. Current visualization systems for network data are deficient in several areas. Generally, they have:

- limited support for multi-level visualization and abstraction
- limited interactivity between the visualization tools and data source (simulation or data source)
- limited abilities to cope with large amounts of data

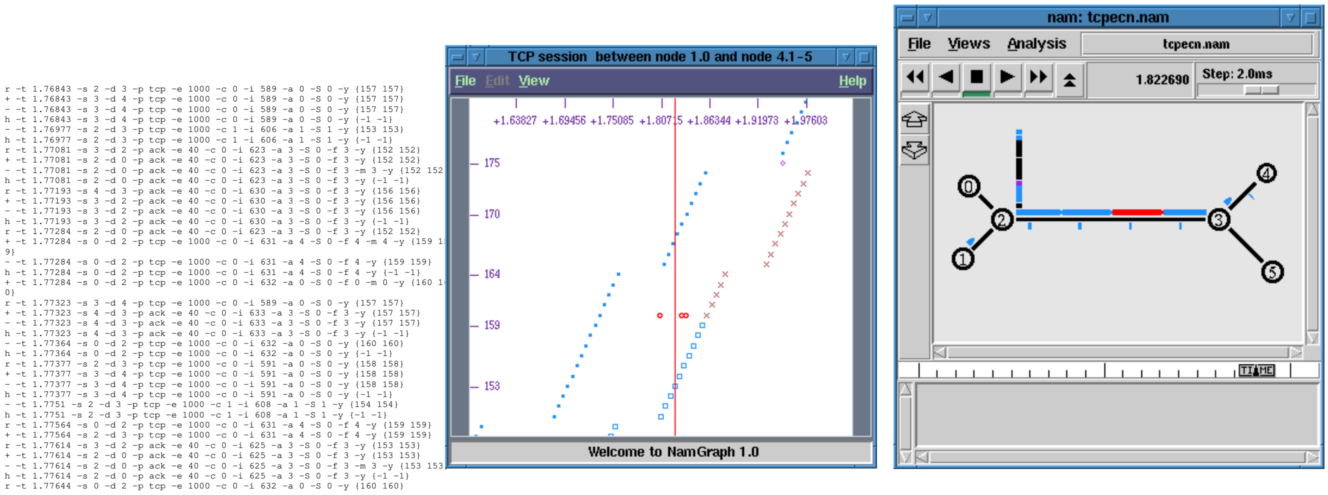

Figure 3: Visualization examples: packet logs, protocol graphs, or packet animations.

- limited applicability
- weak support for application-specific visualizations
- weak support for non-packet-level visualizations (timers, groups, hierarchies)

These areas fall into three classifications: the need for scalable visualization, the need for end-user customizable visualization of multiple data types, and the need for better integration between visualization, simulation, and experimentation.

Scalable, Interactive Visualization: Even large-scale experiments with existing Internet protocols pose a significant visualization challenge. Deeply networked systems raise this bar even further. SEEDS will meet this challenge in two complementary ways. First, we will extend the VINT/nam animation tool to provide hierarchical visualization. For large scenarios, nodes will automatically cluster into "supernodes" which can be interactively expanded as necessary. Second, we will provide selective visualization at several levels. In the simulator, better control over what details are captured and logged will be provided both by direct control and filtering. In the animator, the user will be able to select which nodes are shown both hierarchically (as above) and which are simply not of interest.

Finally, we will tightly couple the simulator and animator to allow interactive control of what data is monitored in the simulator. This represents a substantial change in current VINT ns and nam architecture, but will provide a better user experience by speeding the feedback loop between what might be interesting and what is shown.

End-user Customization of Visualization: Several different innovative approaches are being considered to support self-configuring networking in deeply distributed networking. Data diffusion and other approaches to self-organization and resource discovery will require new protocols and packet formats. Thus, end-user customization becomes a requirement of the visualization system. To meet this requirement, SEEDS will provide end-user programmability support in SEEDS/nam through the use of Tcl scripts and drawing primitives. We will define specific APIs allowing users to specify actions based on packet data and network events.

Network visualization needs to move beyond simple packet animations by providing capabilities to aid in understanding the dynamic and boundary-condition behaviors of algorithms. We will expand packet-level visualization with support for other data types common to network protocols, including timers, hierarchy and group formation, and node states. For example, timers can be represented by bars (or perhaps burning fuses) at each node which decrease as they expire. Multi-hierarchies and node states can be represented by combinations of node color, shape, line thickness, or annotations such as letters, numbers, or colored balls near the node. Some of these are possible in VINT/nam today with substantial effort, but significant improvement is possible. Finally, better support is required to make visualizations self-contained. All visualization features should be self-identifying or identified in labels or legends; currently these services are not provided.

Visualization of Different Data Sources: Visualization tools can process input from any data source capable of providing properly-formatted trace data. Regrettably, few tools exist today to map raw network data into the necessary formats. We will provide tools to support visualization of both simulated and live network streams. One promising approach to easily accomplish this is to pass live data into the simulator through the emulation service, thereby leveraging the existing tracing and logging facilities.

II.B.7. Collaborative Plan

This proposal is a collaboration between USC's Information Sciences Institute and the University of California, Berkeley. The involved personnel have collaborated closely on the VINT project, producing VINT/ns and VINT/nam. We expect this collaboration to be equally successful on SEEDS, due to the natural partitioning of tasks and a proven set of tools to aid collaboration.

The core of the SEEDS effort will be closely coordinated between the two organizations. This work includes improvements to the simulation engine, development, documentation, and validation of new modules. ISI will lead efforts in applying the simulator to collaboration algorithms for deep networking (such as data diffusion), location determination, and hierarchical visualization. ISI technical staff will also coordinate release engineering and support. UC Berkeley will lead the emulation work and parallelization using the Berkeley Millennium infrastructure. Although each individual area will be led at a different institution, substantial sharing will also occur. For example, ISI will use emulation to conduct large scale experiments of mixed physical and virtual nodes, while UCB will use visualization tools to observe live traces through the emulator.

This kind of close distributed collaboration is made possible due to Internet collaboration tools. Both groups will share a common software repository (using CVS), and UC Berkeley will provide remote access to their computing facilities. The collaboration tools developed within the DARPA MASH project will be used to conduct joint weekly meetings. We will also hold periodic (about every 6 months) project-wide meetings and retreats.

II.C. Deliverables

SEEDS project deliverables:

- We will develop the SEEDS environment, to include
  - a simulation engine including interfaces and basic modules for simulating mobile, resource-constrained networked systems
  - an emulation service allowing packets to pass between the simulator and the real world
  - modules using the emulator to validate important real-world protocols
  - support for parallel simulation over a cluster of computers
  - a set of simulator validation and regression tests including an automated facility to execute them
  - one or more models of collaboration protocols, including data diffusion
  - one or more models of location-awareness services
  - one or more models of physical sensors
  - one or more models for CPU, memory, and power capabilities
  - a visualization tool capable of indicating physical location, motion, logical location, and orientation of nodes in three dimensions plus packet flow
- We will provide full documentation of the simulation and emulation environment, as well as all associated algorithm, interface, and protocol designs.
- We will provide the software implementation of the simulation and emulation environment in source code form. The University of California and University of Southern California will retain copyright to developed software. With suitable attribution as to the origination of any code used in derivative software, it is our intention that *no-fee, non-exclusive* licensing agreements be freely available for educational institutions, government sponsors, and industrial collaborators.
- We will provide DARPA with all related technical publications, including technical reports, conference and journal publications, and student theses.
- We will provide DARPA with twice yearly project summaries and status reports.
- We will provide one or more tutorials per year about the SEEDS environment to the DARPA and Internet community. We will also hold working meetings as needed to coordinate the integration of other researchers' simulation modules.
- We will participate in Internet working groups as appropriate to influence future simulation activities, and will work with other DARPA-sponsored groups as requested.
- We will demonstrate enhancements to the simulation base at certain key points in its development, as indicated in the Milestones section of this proposal (II.E.)

See the Statement of Work (II.D.) for a more detailed list of tasks. We expect to make no proprietary claims on the results of this research.

II.D. Statement of Work

Tasks that will be led at ISI or UCB are indicated; where not indicated the effort will be shared.

II.D.1. Simulator and Simulation Modules

1. Evaluate VINT/ns for use as base simulator.
2. Evaluation of existing wireless networking capabilities; integration of modules from other researches such as UCLA and CMU.
3. Simulator extensions for deeply networked systems:
   - support physical and logical location determination, model of location reception and accuracy (ISI)
   - support for node location, position and orientation
   - modeling of node resource constraints (CPU capability, power)
   - framework for modeling sensors and actuators and demonstration models of key sensors
   - support for collaboration algorithms such as data diffusion, including attribute naming and matching, gradients (ISI)
   - support for data aggregation (ISI)
4. Scenario libraries pertinent to deeply networked systems and sensor nets.
5. Documentation for new modules.
6. Validations of key sensor modules and deep networking protocols.
7. Regression test suite for core simulator.
8. Abstraction mechanisms and optimizations supporting large-scale simulations with homogeneous nodes. (ISI)

II.D.2. Emulation Engine

1. Demonstration of large scale experiments using combinations of real and simulated nodes.
2. Demonstration of protocol validation of a complete protocol implementation through the emulation interface. (UCB)
3. Transducers for important Internet and deep networking protocols. (UCB)
4. Support for coarse-grained parallel simulation in a NOW environment. (UCB)
5. Evaluation of use of UCLA and GaTech support of parallel simulation. (UCB)

II.D.3. Visualization

1. Tools to support visualization of real-world network traces (either via emulation or stand-alone).
2. Tight integration of simulation and visualization tools to support interactive visualization. (ISI)
3. Support for hierarchical visualization. (ISI)
4. Support for visualization of 3-D structures (for example, multi-floor buildings). (ISI)
5. APIs for user-customization of visualization. (ISI)

II.D.4. Simulation Studies

1. Simulation studies of key protocols for deeply networked systems. Studies will demonstrate comparisons of alternate approaches and exploration of parameter space of a given protocol.

II.E. Milestones

See the Statement of Work (II.D.) for a more detailed list of tasks.

Simulator: Public simulator releases will occur every 6 months and will be tested on key development platforms. Daily snapshots of the evolving simulator code-base will be constantly available.

- Months 1-6: Evaluate VINT/ns applicability to deep networking, sensor nets, and wireless networking. Add and enhance ns support for location, movement, orientation.
- Months 7-12: Extend location support to include physical and logical locations and models of GPS. Provide initial APIs and libraries for collaboration algorithms (including attribute naming, matching, and gradients) and for sensor models.
- Months 13-18: Demonstrate support for data diffusion and other algorithms for distributed collaboration. Demonstrate preliminary sensor models. Add support for resource-constrained nodes (including CPU and power).
- Months 19-24: Validate key sensor models; provide APIs for modeling actuation. Provide preliminary abstractions for scaling.
- Months 25-30: Provide scenarios library for deep networking and sensor nets.
- Months 31-36: Demonstrate simulation scalability using combinations of abstraction and federated simulation.

Emulation:

- Months 1-6: Extend and enhance current VINT/ns emulation facility with validation tests and low-level support.
- Months 7-12: Migrate emulator addressing architecture to be Internet compatible.
  Develop a test/protocol validation framework for external implementations with emulation; demonstrate validation of multiple independent implementations of a mature protocol (like TCP).
  Begin integration of ns with distributed support services.
- Months 13-18: Demonstrate mixed real and virtual node experiments with deep networking or sensor network nodes
  Use distributed support services from Berkeley NOW project with ns to perform small-scale federated simulations.
- Months 19-24: Quantify and tune performance of federated simulation over NOW-style clusters.
- Months 25-30: Demonstrate functionality on Berkeley Millennium cluster.

Visualization:

- Months 1-6: Evaluate nam for user-customization; design user APIs.
  Extend visualization tool to include support for coordination algorithms and location.
- Months 7-12: Provide preliminary user customization service.
  Demonstrate visualization of live network traces.
  Experiment with hierarchical approaches to large-scale visualization.
- Months 13-18: Tightly integrate simulator and visualization program to provide better interactivity.
- Months 19-24: Evolve hierarchical visualization and support for developing coordination algorithms.

II.F. Technology Transfer

The transferable results of SEEDS will include:

techniques and methods: simulation, emulation, and visualization of deeply networked systems

software: the core simulator, validated modules, interfaces, and libraries for deep networking, emulation libraries for important protocols, scenarios to compare similar protocols

protocol development: simulation studies and validations (comparing simulated to experimental measurements) of key deep networking protocols.

We will promote the use of SEEDS tools through several paths:

frequent software releases: daily public snapshots of the code base (alpha quality), with regular (at least every 6 months) public releases tested on platforms important to the DARPA and Internet research communities

interaction with the SEEDS user community: we will maintain and respond on mailing lists, and provide periodic workshops and tutorials. We will use these forums to obtain early feedback from current and potential users and to aid the integration of other researchers' simulation modules.

demonstrations of SEEDS simulation and emulation tools: to the research and DARPA communities via papers, demonstrations, tutorials, usage in protocol studies and validations

publication : reports and technical papers describing SEEDS techniques, software, protocols and analyses

II.G. Comparison with Other Ongoing Research

SEEDS will build on work in several fields, including network simulation, scalable simulation, simulation of deep networking and sensor nets, and protocol visualization.

II.G.1. Network Simulators

Four network simulation packages are currently in common use within the network research community: VINT/ns [4], Parsec/Glomosim [3], OpNet, and S3/SSF [16]. VINT/ns includes a broad base of advanced Internet protocols (reliable multicast, experimental TCP models, etc) and uses fine-grain objects to enhance extensibility. It has attracted a very large user community (more than 230 institutions, many with several users as of October 1998, with about 200 people attending three tutorials), and has incorporated a large amount of community-contributed code. Parsec emphasizes fine-grain parallel simulation and incorporates its own simulation language and libraries of Internet protocols. It includes advanced wireless channel models and has also captured a large user community (about 300 academic and industrial sites). OpNet is a commercial product with a large user community (over 1,000 sites). It focuses on performance prediction and modeling for the deployment of current-generation networks. Mil3 Corporation, the developers of OpNet, do not release source code for their protocol models, but they do provide commercial support. S3/SSF is a recently developing simulation engine and network protocol class library from DIMACS. It includes a new fine-grain parallel simulation engine and a federated simulator based on ns.

We plan to build SEEDS as an enhancement to the VINT/ns simulator. We believe it provides the most promising base for simulation of deep networking for several reasons. On a technical basis, ns provides a broad library of Internet protocols such as IP multicast plus wired and wireless networking protocols which have promising applications in sensor networks. These protocols are backed up by a regression test suite, so we expect to be able to make substantial changes without breaking existing services. Most importantly, the ns approach of fine-grained composable modules allows substantial code reuse in new protocols and experiments. Ns also contains a preliminary implementation of emulation.

Equally importantly, ns has a proven track record as a vehicle for collaboration, with substantial libraries of contributed code from other institutions and broad impact on research (for example, it was the most widely used simulator by the authors of papers at SIGCOMM '98). We think we can safely say that VINT/ns is the most and perhaps only successful example of a widely adopted simulator with code contributed by many individuals from around the community, in addition to the principal developers. We have demonstrated the importance of investing significant effort into specifying and refining APIs to allow for fine-grain composability, and in building tools and environments that serious designers need. Moreover, we have productively invested in validation tests and scenario generators that are essential to making a truly collaborative design environment feasible. Finally, ns's very open approach to source code with frequent releases and daily snapshots helps foster this environment.

To achieve parallel simulation we intend to leverage of the work of others such as Parsec and Georgia Tech's federated ns. In parallel simulation, both Parsec and S3 provide fine-grain parallelism, while S3 also supports coarse-grain (federated simulation). SEEDS will go beyond basic simulation parallelism in two important ways. First, we will build coarse-grained parallelism over a specialized cluster of cluster computers and automate the task of problem assignment across multiple computers. This assignment will employ domain-specific information available in deep networking simulations. For example, simulated nodes which are physical close will share the same processing node. Second, we will extend the benefits of simulation and parallelism into real world experiments through emulation. This will allow substantial scale-ups in experiments using a mixture of physical and virtual nodes.

While we believe parallelism is important to some researchers, we believe that to achieve advances in the area of deep networking, research must be driven by the development environment that provides flexibility and promotes collaboration, rather than having a primary emphasis on parallelism. Simulation writing time is as often the bottleneck as simulation running time. DARPA's vision in deeply distributed systems will require the concerted efforts of a wide range of researchers at different levels of the system: applications, protocols, hardware design, signaling, devices, and so on. With this breadth of work required collaboration is required if simulation writing time is not to become more of a bottleneck. We believe that by applying the VINT/ns framework and approach to this new problem domain, SEEDS can enable this kind of collaboration and so rapidly advance networking research.

II.G.2. Network Emulation

Specialized forms of emulation have been around for many years in the form of link delay emulators such as Hitbox [1], DummyNet [30], and NIST/net [9]. These systems typically support emulating delay and in some cases controlled loss. SEEDS goes substantially further than these approaches by passing traffic through a simulator. Thus in addition to delay and loss, we can subject real-world traffic to controlled cross-traffic. Since simulated cross-traffic will be responsive to congestion (for example) this approach can support a much wider range of protocol experiments.

Rather than passing real-world packets into the simulator, several systems directly embed real-world code in the simulator. For example, X-sim is a simulation and run-time environment for network protocol development [8]. Simphony virtualizes all networking and supporting kernel interfaces to allow real code to run in a virtual network of operating systems [18]. SEEDS takes the opposite of these approaches: we allow simulators to interact with the real world at the packet (not code) level. SEEDS takes this approach to maximize support for protocol experimentation through fine-grain composition [4]. We have found that many of the details required in kernel networking code (such buffer management, fragmentation, error recovery, etc.) obscure and complicate protocol exploration. For new fields such as deep networking and sensor networks, the need for rapid exploration is paramount.

II.G.3. Deep Networking and Sensor Nets

Simulation of distributed sensor networks includes the DS3 product from Logicon Corporation. It focuses on the modeling of sensors and how to perform logical data collection between them. This system, and others like it, generally fail to include Internet-compatible protocols.

Recent work such as the Simple Systems ISAT study [12] and DARPA's smart spaces workshop have suggested promising new approaches to making deep networking and sensor networks possible. Approaches such as data diffusion [36, 34, 19] seem promising but require additional work to apply to networking.

II.G.4. Protocol Visualization

Network protocol visualization has a long history, including the SeeNet system for 2-D visualization [6], Simphony/Ensim provides protocol visualization [18], and VINT/nam which includes packet-level animation and application-specific time-event plots [4]. CMU's Ad Hockey tool is targeted at mobile computing and supports GUI-based scenario input and visualization [28]. SEEDS will extend VINT/nam, since that system provides good support for both packet-level and application-specific events. Our approaches to hierarchical visualization and end-user customization will extend the scale of this tool.

Work on large-scale network visualization includes traffic visualization at Stanford and NLANR [25], web traffic visualization [20, 32], and network topology visualization of Cox et al. at AT&T [10]. Many others have examined approaches to visualize large amounts of data, for example SeeData [2] and hyperbolic views [31, 21]. We will consider these approaches in our work, but will focus on closer integration with the simulator and experimental apparatus rather than post-experiment viewing.

II.H. Personnel

Key Personnel	Level of Effort
John Heidemann	75%
Deborah Estrin	20%
Kevin Fall	65%
Steven McCanne	(summer)
Randy Katz	(summer)

II.H.1. ISI Personnel

Pursuant to FAR, Subpart 9.5, the University of Southern California/Information Sciences Institute does not support any DARPA offices and therefore has no Organizational Conflict of Interest.

John Heidemann, Ph.D.—Principal Investigator.

John Heidemann is a research assistant professor at USC/ISI where he has worked since 1995. He has been conducting ARPA-sponsored operating systems and network research since 1989. His stackable filing interface is deployed in the 4.4BSD operating system. His networking research has included studies of HTTP and TCP interactions, TCP optimizations, and currently network simulation issues with the VINT project. As a part of VINT he has focused on simulation abstraction to improve scaling and extension of the simulator to new domains. He his a member of ACM, IEEE, and Usenix, is an active reviewer for IEEE/ACM ToN, SIGCOMM, NSF, and is currently co-chair of the USENIX '99 invited talks track.

John Heidemann is currently supported at 100% effort on the VINT project at ISI. He is involved in several pending proposals with DARPA, including SEEDS SCADDS (co-PI at 25% effort with Deborah Estrin and Ramesh Govindan), SNST (co-PI at 25% time with Allison Mankin and Deborah Estrin), all in the area of deeply embedded networking.

Deborah Estrin, Ph.D.—Principal Investigator.

Deborah Estrin is a Professor of Computer Science at the University of Southern California in Los Angeles where she joined the faculty in 1986. Estrin received her Ph.D. (1985) and M.S.(1982) from the Massachusetts Institute of Technology and her B.S. (1980) from U.C. Berkeley. In 1987, Estrin received the National Science Foundation, Presidential Young Investigator Award for her research in network interconnection and security. Estrin is a co-PI on the DARPA Virtual Internet Testbed (VINT) project and the NSF Routing Arbiter project at USC's Information Sciences Institute where she spends much of her time supervising doctoral student research. While she continues her research related to protocol scaling and multicast, most recently she has begun to focus on problems related to networking and coordination among very large numbers of physically-embedded devices (sensors, actuators). Estrin is an active member of the IETF multicast routing related working groups and a long-time member of the End-to-End research group. Estrin is a member of the ACM, IEEE,and AAAS. She has served on numerous panels for The National Science Foundation, The National Academy of Engineering, and DARPA's Information Systems and Technology (ISAT) advisory board. Estrin has served as an editor for the ACM/IEEE Transaction on Networks and the Wiley Journal of Internetworking Research and Experience, and as a member of the program committee for many IEEE Infocom and ACM Sigcomm conferences, and she was program co-chair of ACM Sigcomm '96.

Deborah Estrin is currently supported at 25% effort on the VINT project, 5% effort on the NSF RA project at ISI, and 15% effort on the STRESS project on USC campus. She is involved in several pending proposals with DARPA, including SEEDS 20% (with John Heidemann and Kevin Fall), SCADDS (PI at 45% effort with Ramesh Govindan and John Heidemann), SNST (co-PI at 10% time with Allison Mankin and John Heidemann), all in the area of deeply embedded networking.

The VINT and RA funding will be ending around the time that the currently applied for funding would be available.

II.H.2. UC Berkeley Personel

Summaries of Kevin Fall, Steven McCanne, Randy Katz, the key UCB personnel, appear in the UC Berkeley full proposal.

II.I. Facilities

II.I.1. USC/ISI

ISI plays a key role in the support of networking for research in the Los Angeles area through the development and management of the Los Nettos regional network. This includes the presence of top-of-the-line routers and other networking equipment at ISI.

ISI maintains a large University research Information Processing Center (IPC). ISI has built a reputation for excellence and efficiency in both experimental computing services and production services. The IPC contains a variety of computers, from mainframes to microcomputers. The computers are networked together on an Ethernet-based local area network (100Mbps backbone and server connections, 100Mbps and 10Mbps workstation connections), and all are provided high-speed access to the Internet via the Los Nettos regional network (also managed by ISI).

The Computer Networks Division also maintains research infrastructure in two lab areas. The WAN networking lab includes shared OC-3c ATM links provided and primarily utilized by the CAIRN project and the GTE SCAN consortium, connected to multiple Fore ASX200 ATM switches and an Ascend/Netstar Gigarouter. The High-Performance Networking Lab contains 10 high performance PCs running FreeBSD, connected by 155Mbps ATM, 100Mbps fast Ethernet, and 1.2Gbps Myrinet links. The HPN Lab resources are shared via our Web-based reservation system. The division also uses the largest production installation Myrinet LAN, with 56 office ports currently supporting 35 office workstations, interconnected by a redundant mesh of 10 8-port switches, with 640Mbps links.

II.I.2. U.C. Berkeley Computer Science Division

Soda Hall provides a state-of-the-art environment for investigating innovative networking technologies and novel network services. The building was designed with future network research in mind: the entire building is on raised floor with cabling channel emanating from large vertically stacked communication rooms. The design allows experimental networks to be deployed on a building wide scale, in parallel with a large production network. The complex has grown smoothly from about 500 IP hosts to over 2000 hosts without any wiring. Currently we have several large experimental networks connected through a research router to the campus network. Three of these are substantial ATM clouds with direct connections into servers and expansion through switched Ethernet to a large number of nodes. In addition, there is an experimental wireless network (used by the DARPA-sponsored BARWAN Project, ISDN network, and Gigabit Ethernet components).

Through the DARPA-sponsored Networks of Workstations (NOW) Project, NSF Titan Project, and DOE-sponsored CLUMPS project, we have constructed very large clustered servers with scalable multi-Gigabit system area networks. The NOW prototype consists of 100 Sun Ultra 150s with 12.8 GB of memory, 460 GB of disk, and 30 GFLOPS, collectively, connected through the first very large Myricom network (160 MB/s per link). Using Active Messages we are able to sustain more than 2 GB/s through the fabric and obtain user-to-user communication below 15 μs. A similar network with multiple NICs per node is used for the cluster of four Sun Enterprise 5000 8-way SMPs. In addition, Berkeley has recently received a $6 million donation over three years to construct multiple clusters of Pentium-based SMPs within the department and across campus, including a very large cluster of some 300 processors interconnected with Gigabit Ethernet.

In addition, through such associated research projects as the DARPA/NSF/NASA-funded Digital Libraries Project (Robert Wilensky, Principal Investigator), the Berkeley Multimedia Research Center (Lawrence Rowe, Director), the DARPA-sponsored MASH project, and the Partners in Advanced Transit and Highways (PATH) project, the Berkeley Computer Science Division already has a very rich infrastructure of compute servers, video servers, and mass storage systems upon which we can construct advanced services and widely deploy useful sensors and actuators.

II.J. Cost by Task

1/5/99 Budget Summary to Form 1411

ISI Proposal No. 1998-ISI-094

SEEDS: Simulation and Emulation Environment for Deeply Distributed Systems

Project Leader: Heidemann, Estrin

Period of Performance: 9/1/99 - 8/31/2002

Cost Categories	YEAR 1	YEAR 2	YEAR 3	TOTAL
Salaries & Wages	208,587	216,931	225,608	651,127
Fringe Benefits	58,613	60,958	63,396	182,967
GRA Stipends	92,418	96,115	99,959	288,492
Materials & Supplies	14,921	15,667	16,450	47,038
Computer Supplies	2,251	2,251	2,251	6,752
Communications	2,522	2,428	448	5,398
Software	540	540	540	1,621
Travel	32,736	31,594	36,450	100,780
ISI Common Costs	70,736	73,566	76,508	220,811
IPC Costs	23,830	24,561	25,298	73,690
*SUBTOTAL (MTDC)	507,156	524,610	546,909	1,578,675
Facilities & Admin	122,732	126,956	132,352	382,039
Equipment	50,939	50,939	50,939	152,818
Tuition Assistance	46,010	48,311	50,726	145,047
Division Admin Costs	42,141	43,826	45,579	131,547
ISI Facilities	40,329	40,329	40,329	120,987
**TOTAL	809,306	834,971	866,835	2,511,113

MINOR DIFFERENCES MAY EXIST DUE TO ROUNDING

Proposed Effort:	5.45	5.45	5.45
J. Heidemann, Comp Sci/	75.00%	75.00%	75.00%
D. Estrin, Professor	20.00%	20.00%	20.00%
Y. Xu, Programmer Analy	100.00%	100.00%	100.00%
P. Haldar, Programmer A	100.00%	100.00%	100.00%
TBH, GRA III	62.50%	62.50%	62.50%
TBH, GRA III	62.50%	62.50%	62.50%

TBH, GRA III	62.50%	62.50%	62.50%
TBH, GRA III	62.50%	62.50%	62.50%

GFY Funding Summary:

GFY 1999	117,326
GFY 2000	801,062
GFY 2001	864,756
GFY 2002	727,969

The workstations included in this budget will be used for software development and experimentation directly related to executing this contract. It is USC's policy to direct charge research projects for any equipment that must be acquired in order to conduct the work. As a nonprofit, educational institution, the university is not ordinarily able to provide cost-sharing for the acquisition of equipment.

III: Additional Information

References

[1] J.S. Ahn, Peter B. Danzig, Z. Liu, and L. Yan. Evaluation of TCP Vegas: Emulation and experiment. In *Proceedings of the ACM SIGCOMM*, pages 185–195, Cambridge, Massachusetts, August 1995. ACM.

[2] J.M. Antis, S.G. Eick, and J.D. Pyrce. Visualizing the structure of large relational databases. *IEEE Software*, 13(1):72–9, January 1996.

[3] Rajive Bagrodia, Richard Meyer, Mineo Takai, Yu an Chen, Xian Zeng, Jay Martin, and Ha Yoon Song. PARSEC: A parallel simulation environment for complex systems. *IEEE Computer*, 31(10):77–85, October 1998.

[4] Sandeep Bajaj, Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, Padma Haldar, Mark Handley, Ahmed Helmy, John Heidemann, Polly Huang, Satish Kumar, Steven McCanne, Reza Rejaie, Puneet Sharma, Scott Shenker, Kannan Varadhan, Haobo Yu, Ya Xu, and Daniel Zappala. Virtual InterNetwork Testbed: Status and research agenda. Technical Report 98-678, University of Southern California, July 1998.

[5] Hari Balakrishnan, Srinivasan Seshan, Elan Amir, and Randy H. Katz. Improving TCP/IP performance over wireless networks. In *Proceedings of the First ACM Conference on Mobile Computing and Networking*. ACM, November 1995.

[6] Richard A. Becker, Stephen G. Eick, and Allan R. Wilks. Visualizing network data. *IEEE Transactions on Visualization and Computer Graphics*, 1(1):16–28, March 1995.

[7] UC Berkeley and Computer Science Division. Millennium: A campus-wide cluster of clusters. web page at http://www.millennium.berkeley.edu/, 1997.

[8] L. Brakmo and L. Peterson. Experiences with network simulation. In *Proceedings of the ACM SIGMETRICS*. ACM, 1996.

[9] Mark Carson. NIST net home page. http://www.antd.nist.gov/itg/nistnet/, January 1998.

[10] Kenneth C. Cox, Stephen G. Eick, and Taosong He. 3d geographic network displays. *ACM SIGMOD Record*, 24(4), December 1996.

[11] V. Paxson (editor), M. Allman, S. Dawson, I. Heavens, and B. Volz. Known TCP implementation problems. Work in progress (Internet draft draft-ietf-tcpimpl-prob-02.txt.txt), November 1997.

[12] Deborah Estrin. Colonies of small, specialized devices. ISAT final report, August 1998.

[13] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang. A reliable multicast framework for light-weight sessions and application level framing. In *Proceedings of the ACM SIGCOMM*, pages 342–356, Cambridge, Massachusetts, August 1995. ACM.

[14] SBS Implementer's Forum. *System Management Bus Specification, version 1.1*. December 1998.

[15] Ramesh Govindan, Haobo Yu, and Deborah Estrin. Scalable non-transactional replication in the internet. Technical Report 98-682, University of Southern California, September 1998.

[16] S3 Research Group. Scalable self-organizing simulations. web page http://dimacs.rutgers.edu/Projects/Simulations/darpa/S3_home.html, 1998.

[17] Polly Huang, Deborah Estrin, and John Heidemann. Enabling large-scale simulations: selective abstraction approach to the study of multicast protocols. In *Proceedings of the International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 241–248, Montreal, Canada, July 1998. IEEE.

[18] X. W. Huang, R. Sharma, and S. Keshav. The Simphony protocol development environment. submitted for publication to Infocom '99, July 1998.

[19] Van Jacobson. How to kill the internet. Presentation given at the SGICOMM '95 Middleware Workshop, Cambridge, MA, August 1995.

[20] Stephen E. Lamm and Daniel A. Reed. Real-time geographic visualization of world wide web traffic. In *Proceedings of the Fifth International World Wide Web Conference*, May 1996.

[21] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proceedings of the ACM Conference on Human Factors in Software (CHI)*, pages 401–408, Denver, Colorado, May 1995. ACM.

[22] J. J. Lavigne, S. Savoy, M. B. Clevenger, J. E. Ritchie, B. McDoniel, S.-J. Yoo, E. V. Anslyn, o. T. McDevitt, J. B. Shear, and D. Neikirk. Solution-based analysis of multiple analytes by a sensor array: Toward the development of an "electronic tongue". *Journal of the American Chemical Society*, 120:6429–6430, July 1998.

[23] C. Liu, D. Estrin, S. Shenker, and L. Zhang. Recovery timer adaptation in srm. In *Proceedings of the IEEE Infocom*. IEEE, 1998.

[24] Steven McCanne and Van Jacobson. The BSD packet filter: A new architecture for user-level packet capture. In *USENIX Conference Proceedings*, pages 259–269. USENIX, January 1993.

[25] T. Munzner, E. Hoffman, K. Claffy, and B. Fenner. Visualizing the global topology of the mbone. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 85–92, San Francisco, CA, USA, October 1996. IEEE.

[26] Vern Paxson. End-to-end internet packet dynamics. *ACM/IEEE Transactions on Networking*, 5(5):601–615, October 1997. (Revised version of SIGCOMM paper).

[27] Vern Paxson and Sally Floyd. Why we don't know how to simulate the Internet. In *Winter Simulation Conference*, Atlanta, Georgia, December 1997.

[28] The CMU Monarch Project. *The CMU Monarch Project's ad-hockey visualization Tool for ns scenario and trace files*. Carnegie-Mellon University, August 1998.

[29] Suchitra Raman, Steven McCanne, and Scott Shenker. Asysmptotic scaling behavior of global recovery in SRM. In *Proceedings of the ACM SIGMETRICS*, pages 90–99, Madison, WI, USA, June 1998. ACM.

[30] Luigi Rizzo. Dummynet and forward error correction. In *Proceedings of the USENIX Conference Proceedings*, New Orleans, USA, June 1998. USENIX.

[31] George G. Robertson, Stuart K. Card, and Jack D. Mackinlay. Information visualization using 3d interactive animation. *Communications of the ACM*, 36(4):56–71, April 1993.

[32] Will H. Scullin, Thomas T. Kwan, and Daniel A. Reed. Real-time visualization of world wide web traffic. In *Proceedings of the Symposium on Visualizing Time-Varying Data*, September 1995.

[33] Puneet Sharma, Deobrah Estring, Sally Floyd, and Lixia Zhang. Scalablne session messages in SRM. USC technical report, August 1997.

[34] Alexander Semyon Sherstinsky. *M-Lattice: A System for Signal Synthesis and Processing Based on Reaction-Diffusion*. PhD thesis, Massachusetts Institute of Technology, May 1994.

[35] Stephen Suryaputra, Joseph Bannister, Alan Wilner, and Joe Touch. System architecture and performance evaluation of packet over wavelength. USC/ISI Technical Report, in preparation, December 1998.

[36] A. M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of Royal Society of London*, 237(B):37–72, August 1952.

[37] Kannan Varadhan, Deborah Estrin, and Sally Floyd. Impact of network dynamics on end-to-end protocols: Case studies in reliable multicast. In *Proceedings of the 3rd IEEE Symposium on Computers and Communication Systems*, pages 147–153, Athens, Greece, June 1998. IEEE.

[38] L. Wood, H. Cruickshank, and Z. Sun. Supporting group applications via satellite constellations with multicast. In *Proceedings of the 6th IEE Conference on Telecommunications*, pages 190–194, Edinburgh, United Kingdom, March 1998. Institution of Electrical Engineers.