GENI Design Principles

GDD-06-08

GENI: Global Environment for Network Innovation

August 11, 2006

This document was prepared by the GENI Planning Group:

Larry Peterson, Princeton (Editor)

Tom Anderson, Washington

Dan Blumenthal, UCSB

Dean Casey, Ngenet Research

David Clark, MIT

Deborah Estrin, UCLA

Joe Evans, Kansas

Dipankar Raychaudhuri, Rutgers

Mike Reiter, CMU

Jennifer Rexford, Princeton

Scott Shenker, Berkeley

John Wroclawski, USC/ISI

Table of Contents

1 Introduction

Designing GENI is a multi-year process, with the 10-January-2006 *Project Execution Plan* (PEP) expected to be the first of a series of PEP snapshots. Even after construction begins, GENI will continue to evolve as new technologies become available and new user requirements come into focus. This naturally leads to a question of what capabilities GENI should and should not include. While the definition of GENI will continually change, we expect a set of design principles will guide that decision process, and a core set of value judgments will define GENI's ''center of gravity.''

Intimately linked to the design process is the question of how priorities will be established among the many possible uses of GENI once it has been constructed. We expect GENI will have an advisory board consisting of leading researchers, and that this board will establish policies about how resources are allocated to research projects. Such allocations are primarily based on research merit, but they are also influenced by how closely the proposed usage matches GENI's capabilities.

GENI's "center of gravity" cannot be characterized with a simple black-and-white statement. It is obviously in everyone's interest for GENI to be as useful to as many researchers as is possible—both in terms of functionality and in terms of capacity—but it is a matter of how to best manage and allocate limited resources (i.e., construction budget and facility capacity). This document attempts to define a clear set of principles and priorities that shape such decision processes. It is an attempt to capture the principles that have guided the planning group up to this point, and with community feedback, to serve as a working document to direct the process as we move forward.

2 Research Scope

GENI is an open, large-scale, realistic experimental facility that has the potential to revolutionize research in global communication networks. Its central goal is to *change the nature* of networked and distributed systems design: creating over time new paradigms that integrate rigorous theoretical understanding with compelling and thorough experimental validation. In doing so, we expect GENI will allow us to answer a variety of research questions:

- GENI will allow us to experimentally answer questions about complex network systems, giving us an increased fundamental understanding about their dynamics, stability, emergent behaviors, and related matters.

- GENI will allow us to evaluate alternative architectural structures, and reconcile the contradictory goals a network architecture must meet.

- GENI will allow us to evaluate engineering tradeoffs, and test theories about how different architectural elements might be designed.

GENI could also have broad impact by leading to artifacts that provide value to society. This might result in a better Internet, for example, one that is more secure, available, manageable, usable, and suitable for computing in the next decades. It might serve to

catalyze the distributed digitized world, helping provide users with control of their personal data, or supporting real-time sensing of the physical world. GENI might also provide value to the natural and social sciences, for example, by producing new networking services that enhance the scientific process.

Note that giving a full scientific rationale for GENI is beyond the scope of this document. This note outlines a value proposition for GENI—based on widely accepted criteria for evaluating research on networked and distributed systems—but it does not anchor its arguments in a specific set of research questions. We refer the reader to the PEP for example questions that researchers should be able to address on GENI. Also note that while specific research questions will likely drive the details of what technology to include in GENI, we believe the discussion presented in this document is orthogonal to such questions: this document is more focused on capabilities that bring value to the research methodology.

3 Requirements

GENI comprises a collection of hardware resources, including compute nodes, backbone links, tail circuits, storage capacity, customizable routers, wireless subnets, and so on. Each experiment using GENI will run on some subset of the GENI resources. We call the resources bound to a particular experiment a *slice.* GENI includes management software that is used to allocate resources to slices, embed slices in these resources, and ensure that slices do not interfere with each other.

GENI is intended to support two general kinds of activities: (1) deploying prototype network systems and learning from observations of how they behave under real usage, and (2) running controlled experiments to evaluate design, implementation, and engineering choices.

These are two very different activities. Classical science equates its experimentation with the latter, but Computer Science is different. We benefit from building and running prototypes because building something and watching it run helps us to identify implicit assumptions, the need for different functionality, surprising behavior, unexpected limitations, and so on. In this sense, such "experimental systems" work is like constructing a building—engineering principles tell you whether it is sound design, but you need to build it and use it to decide how well it serves its purpose.

GENI must support both types of activities, and in fact, GENI should make it easy to first perform a sequence of controlled experiments on a new network system, and then subject the system to real user traffic as part of a longer-term deployment study.

With these two experimental modalities in mind, the core set of requirements that GENI must satisfy include:

- **Sliceability:** To be cost-effective, GENI must be a shared facility that can be used to support multiple experiments running on behalf of many independent research groups. Virtualization is a key technology that supports this goal since it allows the facility to be multiplexed across multiple researchers. A second approach is to partition resources among researchers, either in time (analogous to astronomers sharing a telescope) or in space (a single researcher is given exclusive access to some subset of resources). To set a general expectation, we imagine on the order of a thousand researchers utilizing GENI.

- **Generality:** GENI should give each experimenter the flexibility needed to perform the desired experiment. This means that each component should be programmable, so that researchers are not limited to experimenting with small changes to pre-existing functionality. Researchers should not be required to program their entire experiment from scratch — they should be able to take advantage of previously defined functionality and abstractions — but they should not be restricted by such existing functionality.

- **Fidelity:** GENI should permit experiments that correlate to what one might expect in a real network. This means individual components must expose functionality at the right level of abstraction, and it must be possible to arrange these components into a representative network. Clearly, much responsibility for conducting a meaningful experiment falls to the researcher, but it is a goal that the facility, itself, not unduly limit a researcher's ability to conduct such an experiment.

- **User Access:** To support meaningful deployment studies, GENI must make it easy for a broad mix of users to "opt in'' to experimental services. This means providing physical connectivity to a large user community, along with mechanisms that make it easy for users to join one or more experimental services; allowing experiments to run continuously (as no user will want to use a service that is up for only a limited period of time each day); and connecting GENI to the legacy Internet (to gain the positive benefit of interacting with existing Internet services and their users).

- **Controlled Isolation:** GENI must support strong isolation between slices so that experiments do not interfere with each other. GENI's isolation mechanisms should be sufficiently robust to make reproducible experiments possible, and to the extent they are not, it should provide enough feedback about what resources a slice actually receives to enable researchers to evaluate the validity of their results. At the same time, GENI must support controlled interconnection of slices to each other and to the current Internet, allowing researchers to build directly on each other's work, and to draw on existing Internet users and resources. This implies mechanisms that enable user opt-in and desirable data exchange between slices, while keeping undesired outside factors from interfering with GENI experiments and containing GENI experiments so that they do not adversely affect the rest of the Internet.

- **Diversity & Extensibility:** GENI must include a wide class of networking technologies, spanning the spectrum of wired and wireless technologies available today. GENI must also be extensible — with explicitly defined procedures and system interfaces — making it easy to incorporate additional technologies, including those that *do not exist* today. This will allow GENI to be useful to a broad range of researchers, remain useful over a much longer lifespan, support GENI's role as a low-friction vehicle for deployment of new technologies by both academic researchers and industrial partners, and foster close collaboration between "device researchers" and "systems researchers."

- **Wide Deployment:** GENI must have as wide a reach as possible. This is necessary to support experimentation at scale, and to maximize the opportunity to attract real users. Access cannot be limited to only those few sites that host backbone nodes. Wide deployment also implies a rich interconnection of the facility to the legacy Internet.

- **Observability:** GENI must offer strong support for measurement-based quantitative research. This means that the GENI resources, along with all the network systems deployed on it, must be heavily instrumented. The generated data must be collected and archived, and analysis tools developed.

- **Federation & Sustainability:** GENI must be designed for a 15-20 year lifetime, going well beyond a 5-7 year construction phase. To ensure the sustainability, it should be possible for participating institutions (including countries) to contribute resources in return for access to the resources of the GENI as a whole. It should also be possible for new research communities to "opt-in" by connecting their purpose-built networks (including dedicated transmission pipes and sensor networks) into GENI and running their applications and services in a slice of GENI. Both of these scenarios imply the need to support *federation*. In addition, GENI must be designed with *operational costs* in mind, including hardware upgrades, software maintenance, and ongoing operational support.

- **Ease of Use:** GENI must remove as many practical barriers as possible to researchers being able to make full use of the facility. A typical network or distributed systems research project is conducted by a single principal investigator along with a single student. For GENI to be practical for these users, the overhead of understanding how to map their intended experiment onto GENI must be within reach. This means GENI needs to provide a rich set of tools for configuring, monitoring, and debugging experiments, a rich set of common utilities to be used by experimenters, and predictable and repeatable behavior for experiments running on the system. At the same time, GENI will also need to provide access to the full set of capabilities of the system for "power users."

- **Security:** GENI must be secure, so that its resources cannot accidently or maliciously be used to attack today's Internet. To this end, GENI should be designed to operate in a "do no harm" posture: an experiment should run within a "bounding box" that limits what it can do; it must be possible to trace network activity back to the responsible experiment (and experimenter), so that any problems or complaints can be addressed; and should GENI enter a period where activities of some components cannot be adequately monitored or controlled, GENI should restrict those activities by other means to a point where safety can be assured (e.g., by shutting down a slice or bringing GENI as a whole into a safe state).

While it would be difficult to argue against any one of these requirements in isolation, what makes GENI a unique and compelling instrument is how it balances these requirements to support research that simply cannot be done today. This balancing act has two aspects. First, it involves resolving conflicts among requirements; these tensions are discussed in the next section. Second, it involves recognizing the specific combination of capabilities that are unique to GENI—capabilities that are not available in a more limited facility (e.g., in a single researcher's lab or a smaller more special-purpose testbed). They include: (1) wide-spread deployment, (2) a diverse and extensible collection of network technologies, and (3) support for real user traffic. These three properties effectively define GENI's value proposition.

4 Tensions

Many of the requirements outlined in the previous section are synergistic. For example, a wide-spread deployment naturally supports greater user access, and making GENI extensible (so it can accommodate new technologies) is consistent with its support for federation (so new communities and partners can add their resources to GENI).

On the other hand, there are intrinsic tensions among some of these requirements, as well as between different types of experiments that value the requirements differently. This section identifies several of these tensions, and offers guidance as to how conflicts should be resolved.

4.1 Sliceability vs Fidelity

Balancing sliceability and fidelity is one of the most fundamental challenges facing GENI. On the one hand, virtualizing the underlying hardware allows many researchers to share a common set of resources, and can increase flexibility by synthesizing multiple and/or higher-function virtual environments from a single physical resource. On the other hand, virtualization has two potential limits: (1) it allows for the possibility that one experiment might interfere with another experiment, and (2) it potentially hides certain capabilities and properties of the underlying hardware. Both give the facility less fidelity than if a researcher had the resources all to him or herself. Note that virtualization does *not* imply that all slices equally share the available resources, and hence, subjected to unpredictable performance. An admission control mechanism can be used to limit the number of active slices at any given time, and resource guarantees can be made to certain slices. Still the possibility of interference exists.

On the surface, this particular conflict is easy to resolve—GENI should provide strong isolation between slices and the lowest level of virtualization that the technology allows. Any given component may not provide the desired level on day one, but advancing the state-of-art in virtualization over GENI's lifetime is an ongoing objective. Note that higher levels of abstraction should also be retained for those experiments that do not want to be exposed to low-level details, but virtualization should be pushed as "low" as technically possible (cost allowing).

However, there will be those that argue that any amount of virtualization is too much, and that their research requires access to "bare metal." This might be because of the need for access to a component-specific feature, or because virtualization introduces too much unpredictability in timing measurements. There may also be resources that simply cannot be virtualized. GENI does not preclude the possibility that raw hardware elements can be allocated to some slices—as mentioned in the previous section, partitioning is another way of implementing slices—but doing so is likely to come at one of two costs.

Partitioning resources in time means that the resource cannot sustain a real user workload, and hence, limits its appropriateness for deployment studies. Some fraction of GENI's resources can be shared in this way, as long as sufficient capacity is available to support deployment studies. (As noted above, even when virtualization is employed, an admission control mechanism may be used to limit the number of slices that can be active any given time, analogous to time-based partitioning of resources.)

Partitioning resources in space may mean that only a limited number of researchers can include a given resource in their slice. This may be necessary for certain high-cost resources that cannot be easily virtualized, in which case it will be necessary for the community to either prioritize their research or find ways to synthesize their many experimental systems into a few comprehensive systems. While we might imagine a thousand researchers sharing GENI as a whole, we might see perhaps only tens of research projects sharing access to any high-cost/non-virtualizble resource in this way.

Independent of the technique used to slice resources, a GENI policy committee will necessarily be involved in prioritizing resource allocation decisions.

4.2   Generality vs Fidelity

Designing GENI to be general (programmable) is potentially at odds with perfect fidelity. For example, a researcher could argue that to faithfully evaluate a new function or protocol it is necessary experiment with a commercial implementation, or possibly with a function-specific hardware implementation. In practice, however, such an implementation is likely to expose a limited interface rather than be generally programmable. Such a device has perfect fidelity for a narrow set of experiments, but less value to the larger research community. On the other hand, an open source, software-based implementation of the same function or protocol might run on a general-purpose component that other experimenters can share, but without the performance or fidelity of the special-purpose implementation.

Clearly, it should be possible to make a merit-based case for the special-purpose component that benefits a narrow set of researchers, but it is generally expected that some amount of fidelity will be sacrificed to support a general-purpose facility that serves a wide-range of research. We also note that more narrowly defined communities should be allowed to connect their special-purpose components to GENI, and make them available to interested researchers.

Related to the issue of generality versus fidelity is the issue of simplicity: researchers want to work at a low enough level of abstraction so that important system details are not hidden, but at the same time, they do not want to work at such a low level that they have to reinvent uninteresting (to them) layers of software just to create an environment that allows them to address their specific research problem. This is actually a unique opportunity for GENI—it should support multiple levels of abstraction, and over time, build up a suite of shared code for commonly used functions. Researchers should be able to work at whatever level of abstraction best matches their needs.

4.3   Architectural Design vs Technology Development

We expect an on-going tension between researchers wanting to use GENI to test and evaluate new networking technologies, and those wanting to evaluate new architectural designs that (among other things) take the capabilities of new technologies into account. The former tend to focus on single components, while the latter must take a more comprehensive (end-to-end) perspective. GENI's policies should favor architectural research (broadly defined) that takes advantage of the fact that it spans a diverse collection of hardware resources. This is because no individual technology is fully validated until it has been shown to work with real users in a given context, but also

because we are interested in exploring alternative architectures that are capable of integrating a diverse set of technologies.

We note, however, that there is value to component developers being able to evaluate their technology in the context of end-to-end architectures and under the realistic workloads GENI is expected to generate. GENI should allow such technologies to be plugged into the facility once they are mature enough to support GENI users, but we expect early-stage technology development (both hardware and software) to happen outside of GENI. (There is also likely to be a transition path whereby a new technology is made available to early adopters in a subset of GENI.) To make a case for adding a new component to GENI, it will need to support the interfaces defined by the management framework, be sufficiently programmable to give researchers the flexibility they need, and to the extent possible (see the above discussion), be sharable by multiple slices.

Note that this discussion does not directly address the question of what technologies are initially included in GENI. This decision is driven largely by the requirements of the specific research to be conducted on GENI, and so is beyond the scope of this document. In general, however, we observe that the overriding goal is to include a diversity of technologies that stress the "corner cases" of comprehensive network architectures.

4.4 Performance vs Function

A question often asked about a network is "how fast does it go?" Asking this question of GENI raises the question of performance goals within GENI's design. In the past, performance-related objectives have often defined network testbeds, with speed becoming the key measure of success.

In contrast, GENI's research objectives are broad, and its success metrics focus on properties other than speed. As a result, GENI's design is not focused on performance, and in fact many of the mechanisms used within GENI dramatically increase the challenge of achieving high performance. Despite this, performance cannot be neglected; if GENI does not offer sufficient performance to be useful, it will not be used.

Unfortunately, performance is not a single metric. Rather "performance" encompasses a number of metrics, considered along at least three dimensions. Each of these dimensions affects a different class of experimenters and users of GENI:

- *Relative performance* is the ratio of performance at one point in the network to performance at other points, or of one performance metric to another performance metric at some point in the net. Relative performance ratios may have a strong effect on network architecture, as well as determining the types of operations that can be performed on data within a network.

- *Absolute aggregate performance* is the level of performance available to meet overall system demand at any given place and time. Absolute aggregate performance is important to supporting applications such as content distribution and flash crowd management.

- *Absolute single-flow performance* is the level of performance available to a single application session. Absolute single-flow performance is important to supporting new high-demand applications, such as HDTV video or 3-D data visualization.

In each of these dimensions, there is tension between performance, function, and cost. This tension is strengthened by GENI's objective of providing programmable and sliceable substrate across a range of technologies. Performance levels that are simple to reach in a tuned, fixed-function component are often expensive or difficult to attain within a more general-purpose, flexible system element. Further, reasoning about GENI performance metrics is made difficult because GENI's performance objective is the more nebulous "good enough to meet GENI's research support goals", rather than a simpler, more specific one such as "as fast as possible" or "100 Gbps."

A final tradeoff related to GENI performance concerns how the system evolves over time. It is clear that performance levels sufficient for the first phase of GENI deployed in the near future will be insufficient for the lifetime of the facility. For this reason, performance goals in the near term must be related to longer-term plans for ongoing upgrade and improvement of the facility.

4.5   Networking vs Applications Research

GENI is neutral about what level of the network researchers focus their efforts, and so does not draw sharp lines between network low-level protocols, high-level network services, and end-user applications. Any research that benefits from wide-spread deployment, diverse network technologies, and support for realistic network conditions should be supported.

The critical point-of-tension is that GENI is designed to support research in networking and distributed systems—as opposed to simply providing bandwidth to end users—yet it also benefits from traffic generated by real users. It will be necessary to evaluate the research value of traffic generated by a given slice to decide if allocating resources to that slice is warranted, rather than merely providing an infrastructure service to some user community. We can imagine three ways in which a research group justifies the value of traffic they are carrying: (1) by making traffic traces available to other researchers, (2) by providing a novel network service whose efficacy needs to be evaluated, and (3) by offering to run as part of (on top of) a novel network architecture.

Note that new communities that find value in some capability of GENI—or some innovative service deployed on GENI—are free to augment GENI with enough capacity to carry their user traffic, independent of other research considerations.

4.6   Design Studies vs Measurement Studies

GENI is being designed primarily to allow researchers to experiment with new network architectures and services not available today, and this purpose will be the primary factor used to prioritize among various design choices and resource allocation decisions. Our hope and intention, however, is that the GENI facility will also provide a new capability for monitoring the current Internet. We believe such dual-use is possible because both capabilities require wide deployment, rich interconnection to the existing Internet, and heavy instrumentation. Using the GENI facility as a platform to monitor the current Internet is a secondary goal that will also inform its design.

4.7   Deployment Studies vs Controlled Experiments

We do not view the two primary usage models as being in conflict—a research group

might naturally progress from a series of controlled experiments to a long-term deployment study—but there is an important difference in how the two models stress the facility. Both are related to security.

A controlled experiment attempts to both eliminate all outside (uncontrolled) influences from affecting the experiment, and keep the experiment from impacting the rest of the world. The latter requires strong *containment* mechanisms, so that for example, an experiment that measures the effectiveness of a new malware-prevention architecture is not allowed to escape onto the Internet. Because such a breach of containment could have a catastrophic effect, it is likely that experiments will need to be reviewed to evaluate such risks.

In contrast, a deployment study necessarily involves an experimental service interacting with real users, including both individuals that are trying to abuse the network in some way, and individuals that are trying to use the network to transport illegal content. GENI must be willing to carry such traffic; it cannot be isolated for the sake of security. Thus, GENI is expected to behave like an ISP in today's Internet in that it must be responsive to complaints when they are raised. This means it must include auditing mechanisms that allow operators to identify badly behaving experiments, so that they can be quickly isolated or shut down. In general, it must be possible to rapidly bring the facility as a whole into a safe and controlled state.

5 Engineering Principles

GENI will be a complex distributed system that includes a wide-array of computing and networking technologies, and a sophisticated collection of software services. Everyone understands that building GENI will be a significant undertaking, yet it is important that the facility be constructed on-time and on-budget. While sound management practices will certainly need to be adopted, we also identify a set of engineering design principles that we consider essential to the successful construction of the GENI facility.

- **Start with a well-crafted system architecture**. The more complex the factorization of the system into a set of component building blocks, the greater the risk that the inter-dependencies among components will become unmanageable. The success of the Internet itself can be traced in large part to the fact that its architecture allowed components to evolve independently of each other. The GENI architecture is guided by the same design principle, whereby independent technologies can be plugged into the management framework with virtually no dependency on each other, and independent distributed services to be developed without heavy-weight coordination. This technical solution is critical to the ability to manage the construction process.

- **Build only what you know how to build**. Because software is plastic, there is a tendency towards feature creep; it is easier to specify the features a system "must" have, than it is to make those features work together. Left unchecked, this can result in systems that are simply too complex to work. There will be those who will complain that we are doing too little, beyond what we already understand. Our answer is, exactly, but the *synthesis* of these elements is revolutionary.

- **Build incrementally, taking experience and user feedback into account**. It is a well known result of computer science research that in software or hardware construction

efforts, errors are cheapest to fix when they are caught early. The best way to do that is to put the system into active use at the earliest possible moment, gain live experience with the system, and incrementally evolve the system based on what you learn.

- **Design open protocols and software, not stovepipes**. A huge point of leverage for us, versus other examples of large scale software systems construction, is that the users of the facility—the computer science research community—are themselves capable of fixing and enhancing the system, if we give them the right tools. This is unique to the case where we build systems for ourselves, versus building systems for other people; project meltdown is much more likely if the result is take it or leave it. We aim to build a system that continues to evolve in meaningful ways after GENI construction is complete. All of the successful examples of large-scale systems being successfully delivered by the computer science research community have the property that they continued to be modified by their user community, well after initial delivery.

- **Leverage existing software**. While some aspects of GENI will need to be implemented from scratch, we expect to be able to leverage significant amounts of existing software. It is essential that we take advantage of such software, and to the extent possible, do so in a way that allows us to also leverage the support systems already in place to keep this software up-to-date. Even adapting, rather than directly using an off-the-shelf software package takes time, and raises the question of who now supports the modified package. Similar arguments favor commercially available hardware.

- **Leverage existing infrastructure.** While GENI will provide an environment for experimenting with networking concepts that are not constrained by today's Internet, GENI must be able to leverage today's Internet as a ``bootstrapping'' mechanism. GENI should not ``architect in'' any assumptions about today's Internet—so that newly designed alternatives can be substituted for today's protocols as they become available—but neither should we be apologetic about using today's (sometimes imperfect) network to make the construction of GENI feasible. The alternative strategy of having GENI depend on yet-to-be-demonstrated network architectures or capabilities is a far greater risk to its success.

Note that while this discussion focuses on the application of sound engineering practices during GENI's construction, it is equally important to consider the on-going operational costs of managing and maintaining the GENI facility after construction is complete. We believe the same set of engineering principles that make the construction process manageable also lend themselves a sustainable system.