Computer Networks 61 (2014) 5–23

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

GENI: A federated testbed for innovative network experiments ☆

Mark Berman [a,*], Jeffrey S. Chase [b], Lawrence Landweber [c], Akihiro Nakao [d], Max Ott [e], Dipankar Raychaudhuri [f], Robert Ricci [g], Ivan Seskar [f]

[a] GENI Project Office, Raytheon BBN Technologies, 10 Moulton Street, Cambridge, MA 02138-1119, USA
[b] Department of Computer Science, Duke University, Durham, NC 27708-0129, USA
[c] GENI Project Office and Computer Science Department, University of Wisconsin, 1210 West Dayton Street, Madison, WI 53706, USA
[d] University of Tokyo, 4-6-1 Komaba, Meguro-ku, Tokyo 153-8904, Japan
[e] NICTA, Australian Technology Park, Level 5, 13 Garden Street, Eveleigh, NSW 2015, Australia
[f] Wireless Information Network Laboratory, Rutgers University, 671 Route 1 South, North Brunswick, NJ 08902-3390, USA
[g] School of Computing, University of Utah, 50 South Central Campus Drive, Room 3190, Salt Lake City, UT 84112, USA

ARTICLE INFO

Article history:
Received 5 July 2012
Received in revised form 22 November 2013
Accepted 20 December 2013
Available online 3 January 2014

Keywords:
GENI
Future Internet
Next generation networking
Future Internet testbeds
Distributed cloud computing

ABSTRACT

GENI, the Global Environment for Networking Innovation, is a distributed virtual laboratory for transformative, at-scale experiments in network science, services, and security. Designed in response to concerns over Internet ossification, GENI is enabling a wide variety of experiments in a range of areas, including clean-slate networking, protocol design and evaluation, distributed service offerings, social network integration, content management, and in-network service deployment. Recently, GENI has been leading an effort to explore the potential of its underlying technologies, SDN and GENI racks, in support of university campus network management and applications. With the concurrent deployment of these technologies on regional and national R&E backbones, this will result in a revolutionary new national-scale distributed architecture, bringing to the entire network the shared, deeply programmable environment that the cloud has brought to the datacenter. This deeply programmable environment will support the GENI research mission and as well as enabling research in a wide variety of application areas.

1. Introduction/purpose

GENI, the Global Environment for Networking Innovation, is a distributed virtual laboratory sponsored by the U.S. National Science Foundation (NSF) for the development, deployment, and validation of transformative, at-scale concepts in network science, services, and security. The key motivation for GENI is addressing widespread concerns about *Internet ossification*, the phenomenon whereby entrenched interests and infrastructure severely limit the potential for innovation in the global, public Internet [1,2]. In the middle of the first decade of the 2000s, virtualization began to emerge as a promising approach to enabling a large class of interesting experiments to run simultaneously at reasonable cost. Indeed [1], proposes the virtual testbed concept, and argues that it is well suited not only to testing, but also to deployment of new network architectures.

Today's GENI relies heavily on this basic concept, combining heterogeneous resource types, each virtualized along one or more suitable dimensions, to produce a single platform for network science researchers. GENI provides a venue that permits a growing number of experiments to move beyond simulation and into emulation and realistic

☆ GENI is funded by the U.S. National Science Foundation. Any opinions, findings, conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect the views of the National Science Foundation.
* Corresponding author. Tel.: +1 617 873 3675.
E-mail address: mberman@bbn.com (M. Berman).

1389-1286/$ - see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.bjp.2013.12.037

deployment environments. GENI is now progressing beyond its initial prototyping stage and into a period of wider deployment of increasingly standard components. Expansion is via GENI-enabled campuses, which incorporate these key GENI components:

- GENI racks: virtualized computation and storage resources.
- Software-defined networks (SDNs): virtualized, programmable network resources.
- WiMAX: virtualized cellular wireless communication (at selected campuses).

The current deployment phase is taking GENI from its initial "meso-scale" size of fourteen sites to more than fifty. (See Fig. 1 for candidate sites.) Plans for growth to 100–200 sites are currently under evaluation.

The degree of success of any laboratory, particularly one shared by a large community, depends heavily on two key factors: *utility* across a wide range of experiments, and *availability* to the relevant researchers. GENI's range of supported experiments includes:

- Size diversity ranging from simple, small experiments to "at scale" experiments involving large numbers of resources and/or large geographic extent (e.g., national scale).
- Protocol diversity including experiments compatible or incompatible with existing internet protocols, such as Internet Protocol (IP) and Transmission Control Protocol (TCP).
- Execution environment diversity, ranging from tightly controlled and highly repeatable laboratory-style experiments to "in the wild" service trials exposed to the vagaries of the global Internet.
- Opportunity for the inclusion of "real users," early adopters who use experimental services running on GENI and expose experimenters to real usage patterns.

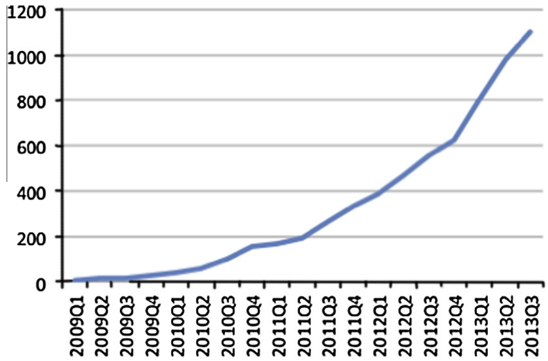


Fig. 2. Cumulative unique users creating GENI slivers.

This range of capabilities is intended not only to support a wide variety of research projects, but also to support the lifecycles of individual projects. Many projects start with small experiments in controlled environments, and expand to larger scales, more realistic deployment conditions, and even real users.

As an NSF-sponsored effort, GENI is widely available to researchers both inside and outside the U.S. The initial targets are the networking and distributed systems research communities. Results from early experiments carried out over the past two years indicate that a broad range of researchers are able to conduct investigations in a range of areas, including clean-slate networking, protocol design and evaluation, distributed service offerings, social network integration, content management, and in-network service deployment. A growing number of educators are employing GENI in their classes, both for laboratory assignments and research projects. Other likely beneficiaries include cloud computing researchers and a variety of domain scientists. At this time, over one thousand unique users have drawn upon GENI resources (see Fig. 2), reserving over 28,000 slivers.

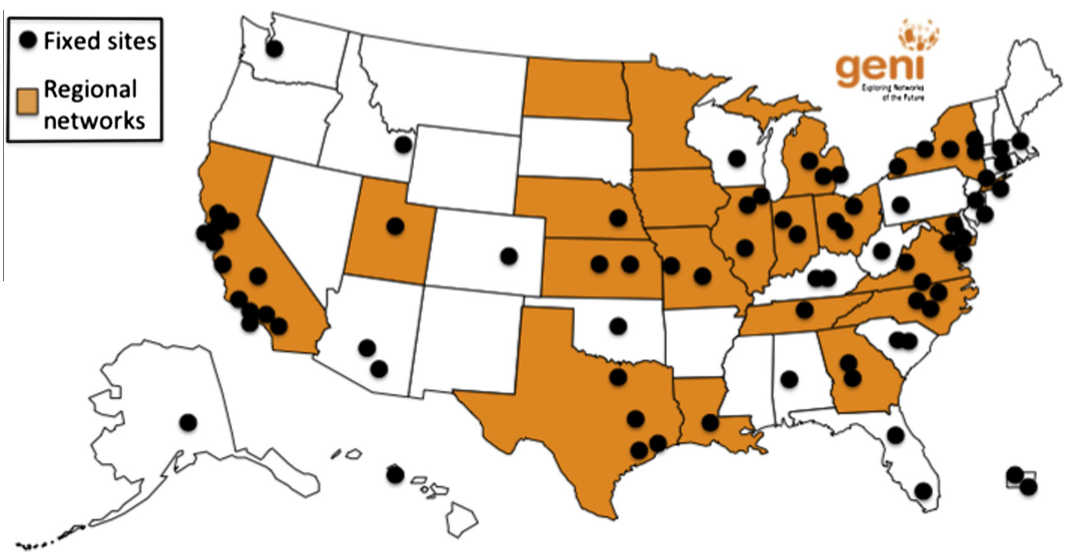


Fig. 1. Current phase candidate GENI expansion sites.

2. Key experimenter concepts and experiment life cycle

2.1. Experimenter concepts

Since the advent of timesharing, virtualization technologies have sought to present each user with the illusion of exclusive ownership of shared resources. In GENI, the exclusive resource illusion is that of a constellation of general-purpose computers, connected in experimenter-specified topologies via a programmable layer two network. Two key concepts contribute to the creation of this environment.

2.1.1. Sliceability

GENI borrows the concept of *sliceability* from the PlanetLab testbed [3]. Sliceability is the ability to support virtualization (simultaneous shared access to shared physical resources), while maintaining some degree of isolation for simultaneous experiments. The set of (virtualized) resources assigned to an experiment is called a *slice*, and each independently managed resource in a slice is a *sliver*. GENI slicing includes both compute and network resources.

Clearly, the selection of virtualization approach implies a tradeoff among performance, isolation, and cost. Maximum performance and complete isolation could be achieved with fully independent infrastructure, thus forgoing the cost benefits of virtualization. Because different experiments require different levels of performance and isolation, GENI supports multiple models of virtualization, even for a single resource type.

For example, GENI slices may incorporate compute resources from PlanetLab. These resources are implemented via a linux Vserver virtualization approach [4]. A slice may also include compute resources from ExoGENI racks, implemented as KVMs (kernel-based virtual machines) via OpenStack [5]; virtual machines from InstaGENI/ProtoGENI (OpenVZ virtual machines); or raw physical hosts from ProtoGENI [6] or ORBIT [7]. Each of these virtualization options provides different performance, isolation, and efficiency parameters, and any combination may be combined into a single GENI slice.

Similarly, network resources may be virtualized into GENI slices using two different paradigms. The most basic virtualization strategy is virtual local area networks (VLANs). VLANs provide a well-understood degree of data isolation, but they do not offer meaningful performance isolation or programmatic control. Experimenters desiring a more flexible network virtualization approach may choose to incorporate software defined networks (SDNs) in their GENI slices [8]. The current GENI implementation of SDN virtualization is via OpenFlow-enabled [9] network components.

2.1.2. Deep programmability

The concept of *deep programmability* is central to GENI's strategy for confronting ossification. Deep programmability refers to an experimenter's ability to influence the behavior of computing, storage, routing, and forwarding components deep inside the network, not just at or near the network edge. Traditional Internet development has not emphasized deep programmability as a desirable trait. In fact, many concerns, including equipment cost, performance, simplicity, interoperability, security, and others have often pushed in the opposite direction. The result is an environment that limits innovation. With the growing availability of hardware and software technologies that virtualize the network, it is no longer necessary to limit innovation to certain layers of the network: the time is ripe for introducing programmability, and thus enabling innovation in parts of the network that have long been considered "fixed points."

A typical GENI experiment's use of deep programmability involves multiple sites, each containing one or more programmable resources, communicating via standard and/or non-standard protocols, which are implemented in experimenter-provided software. A small experiment may require relatively few resources and might well be feasibly executed without recourse to physical GENI resources in the experimenter's own laboratory or in the laboratories of a handful of collaborating researchers. However, as an experiment's scale increases, the challenge of identifying additional resources and sites grows, and the need for a shared platform becomes clearer.

The effectiveness of deep programmability rests chiefly on two factors: (a) the extent (number, connectivity, and geography) of the programmable resources, and (b) the capabilities and performance of programmable behaviors. Although researchers might desire a fully programmable resource pool of the scale of the Internet, such a shadow infrastructure is clearly infeasible, for cost reasons if no other. Accordingly, GENI's growth strategy piggybacks on existing infrastructure investments to reach a large number of deep programmability sites (see Section 3 below).

A variety of deeply programmable components provide GENI experimenters with several options with different flexibility, ease of development, and performance characteristics. Rapid development and high flexibility is readily achieved by employing general-purpose computers as routing and forwarding components. A special case of this approach, which has proven highly effective in small-scale experiments, is to use a modular router package, such as Click [10]. Experiments with more specialized needs, for example [11], may directly program the required functionality. For experiments requiring higher performance in-network programmability, OpenFlow can be a good option, giving researchers flow-level control at line speeds, while requiring only a limited investment in programming and no special hardware. GENI also incorporates a modest number of special-purpose deeply programmable network components, including NetFPGA [12] and the Supercharged PlanetLab Platform [13].

2.1.3. User opt in

One lesson from the experiences of PlanetLab and other testbeds is that many interesting experiments are in fact service deployments, which rely on user interactions as a central research focus. For this reason, GENI incorporates the notion that end users who are not directly affiliated with an experiment team may *opt in* to the services provided by an experimental GENI slice. Because GENI slices are built on straightforward network virtualization concepts, they are particularly amenable to supporting opt in. When conducting experiments with opt in users, researchers are bound by standard ethical guidelines. In order to accept opt in users, an experimenter must proceed in two distinct phases. The first phase, experiment qualification, verifies that the experiment satisfies appropriate regulations and safeguards regarding safety and privacy. The qualification process generally involves collaboration between the experiment team and other campus authorities, such as an institutional review board (IRB) and CIO staff. Once the experiment is qualified, the experimenter and CIO team enable the network connectivity aspects of opt-in, and the experiment moves to the next phase.

It is during the second phase, enrollment, that end users actually have the opportunity to access experimental services. After being presented with appropriate information regarding information privacy controls, informed consent, etc., an end user opts in to an experimental service. At this point, selected traffic appropriate to the experimental service is diverted from its usual path into the public Internet and into the experimental slice. Both the precise mechanism used for redirection and the exact selection of which traffic is selected are dependent on the nature of the service being provided. Consider, for example, an end user who opts in to an experimental video streaming service. At enrollment time, the MAC address of her computer may be provided to the OpenFlow controller managing the experiment slice. So long as the user is connected to the campus OpenFlow network, traffic associated with the well-known service port for the video streaming application is now passed through the experiment slice, while all other traffic is unaffected. Note that different opt in mechanisms (e.g., explicitly changing service port, selecting a proxy server, end-host VLAN selection, or choosing a designated wireless SSID) may be chosen during the qualification phase, each with different implications for the end user experience.

2.1.4. Selected experimental applications

Taken together, sliceabiliity, deep programmability, and geographical diversity have shown substantial flexibility in enabling experimenters to challenge Internet ossification.

Novel Routing. One example is a series of experiments in adaptive source routing performed at the University of Illinois, Urbana-Champaign (UIUC) in 2010 and 2011. Pathlet routing [14] permits a limited form of source routing, in which networks advertise fragments of paths, called pathlets. Sources concatenate these into end-to-end source routes. Initial experiments, conducted on an overlay network running within PlanetLab, validated the basic pathlet concept, but with the unsatisfying restriction of running a new routing algorithm only over standard IP routing in the commodity Internet. By running an OpenFlow-based implementation of pathlet routing, the UIUC team was able to move path selection into the fast path of switching devices and test performance at line rates. These experiments enabled the evaluation of pathlet routing performance in a realistic network latency and packet loss environment, when contrasted with validation in simulation or in an overlay network. Future experiments are expected to include performance investigations incorporating multiple paths along the GENI backbone, optionally in combination with Internet paths.

Future Internet Protocols. In another example, the MobilityFirst Future Internet Architecture (FIA) project [15] has employed GENI-based configurations for a series of tests and experiments over several months to assist in development and validation of the basic algorithms and protocols underlying the mobility-focused architecture. Key components of the MobilityFirst architecture selected for early validation in GENI include a massively scalable global name resolution service [16] and generalized storage aware routing [17]. These test configurations enable the evaluation of MobilityFirst protocols in a realistic setting that would be difficult to create without the use of GENI. The current proof-of-concept network on GENI consists of 12 routers deployed on programmable GENI platforms (i.e. ProtoGENI nodes) spread across the US, with three edge networks (located at BBN, Cambridge, MA, WINLAB, Rutgers, North Brunswick, NJ and University of Utah, Salt Lake City, UT) having a WiMAX base station and/or WiFi access points for end-user mobile access. An additional node was deployed in Tokyo, Japan using the experimental FLARE programmable router platform recently developed by University of Tokyo researchers [18]. The corresponding topology (shown in Fig. 3) employs VLAN-based layer two stitching across multiple participating networks to establish a single layer two network across all deployed components. The configuration provides realistic wide-area RTT delays between router nodes, and combines a variety of link speeds and access technologies [19].

A test application featured in a March 2013 demonstration focused on mobile access to different delivery services offered by the MobilityFirst network, namely unicast, multicast and anycast. Mobile hosts accessing the network from the Utah demo site connected over WiFi and ran the MobilityFirst protocol stack with hop-by-hop transport, storage-aware routing and GUID (globally unique identifier) based services. The demo system deployed a weather service application that ran local instances of the service at different sites on the network – Salt Lake City, Utah, New Brunswick, NJ and Cambridge, MA – each providing the local weather information for the deployed physical location over an anycast service.

Other FIA projects have also incorporated GENI into their evaluation strategies. The eXpressive Internet Architecture (XIA) project [20] uses GENI for evaluation and demonstration of the XIA prototype software router and associated protocols and includes GENI as a target platform for software releases. The Named Data Networking project [21] uses a hybrid testbed configuration, incorporating both GENI and non-GENI resources, for performance testing.

Nationwide OpenFlow Network. Both of the experimental applications above, as well as a number of other experiments, are facilitated by a standing experimental OpenFlow deployment within the GENI network. The EnterpriseGENI project, a multi-university collaboration led by researchers at Stanford University, developed a control framework and deployment kit for OpenFlow-managed networks. Building on this platform, the

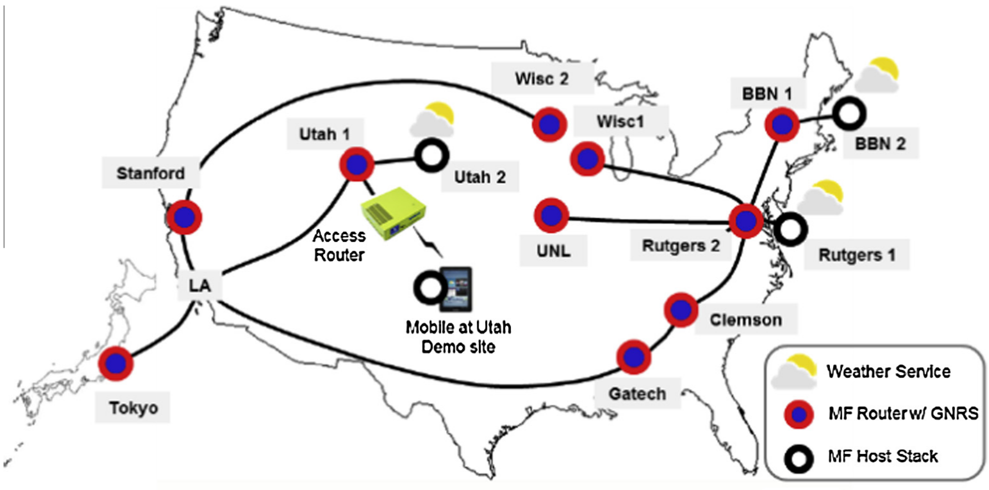


Fig. 3. Topology and deployed components for proof-of-concept MobilityFirst network demonstrated at GENI engineering conference (Salt Lake City, March 2013).

TangoGENI effort (or GENI mesoscale OpenFlow network) interconnects seven university campuses, the GENI Project Office (GPO), five National Lambda Rail (NLR), and five Internet2 (I2) sites, with multiple OpenFlow-controlled VLANs. Differing topologies are selected for the VLANs, providing experimenters with the opportunity to select different network performance characteristics by mapping their experiments onto an appropriate selection of network devices and general-purpose computers at these deep programmability sites. Fig. 4 shows the configuration of one of the TangoGENI VLANs.

2.2. Experiment lifecycle

The lifecycle stages of a GENI experiment support the wide range of experiment styles (Fig. 5). Experiments vary significantly in extent and duration, from short, highly controlled experiments to long-lived, "in the wild" service trials. Furthermore, GENI is both heterogeneous and federated, implying a need for sophisticated service discovery and configuration capabilities.

The wide range of experiment styles, sizes, and durations strongly suggests that there can be no satisfactory single experimenter interface to GENI. Rather than seek such a Swiss army knife approach, GENI employs open application programmer interfaces (APIs) to enable the development of multiple experimenter tools. This strategy is an explicit decision to give up the possible benefits of an enforced single user interface to GENI, with the expectation of encouraging a thriving community of interoperating experimenter tools. The initial results of this decision are promising, but it is still too early to draw a definitive conclusion. Early GENI tool development understandably focused on resource discovery and configuration and on basic experiment management, disproportionately populating these portions of the experiment lifecycle. Ongoing development in the GENI community continues to explore the benefits of simple, single-purpose tools and of larger, multipurpose tools and tool sets.

The GENI software architecture provides an API-level interface to mediate between resource providers and experimenters. Resource providers make their capabilities available in *aggregates*, collections of similar resources under the control of an aggregate manager. Experimenters discover, reserve and configure resources via tools. Tools are software that presents a user interface, generally graphical (GUI) or command-line (CLI) to the experimenter. When appropriately authorized by the experimenter, the tool acts as an agent, providing an abstraction layer above the API. By maintaining this open API, GENI creates an open market for tool development.

The aggregate manager API specifies the interface between tools and aggregate managers. An additional architectural component, the *clearinghouse* provides a set of services for GENI users, such as account creation and federation-wide resource tracking. The clearinghouse API specifies interfaces between tools and the clearinghouse and between the clearinghouse and aggregate managers [22].

2.2.1. Resource discovery and configuration

Resource discovery in GENI generally proceeds as a two-step process, often fully or semi-automated by experimenter tools. In the initial step, the clearinghouse or similar capability is queried to identify a list of aggregate managers with available resources. In the second step, the aggregate managers are queried for their available resources, using the *listresources* command. The aggregate managers reply with a resource specification, or *rspec*.[1] GENI supports three types of rspec. This type, which describes the resources an aggregate provides, is called an *advertisement rspec*.

Several experimenter tools exist to assist an experimenter in resource discovery, selection and configuration, and others are expected to become available. These tools

[1] A GENI rspec is an XML document intended to be generated and parsed by automated tools. The interested reader may find schema definitions for GENI rspecs at http://geni.net/resources/rspec.

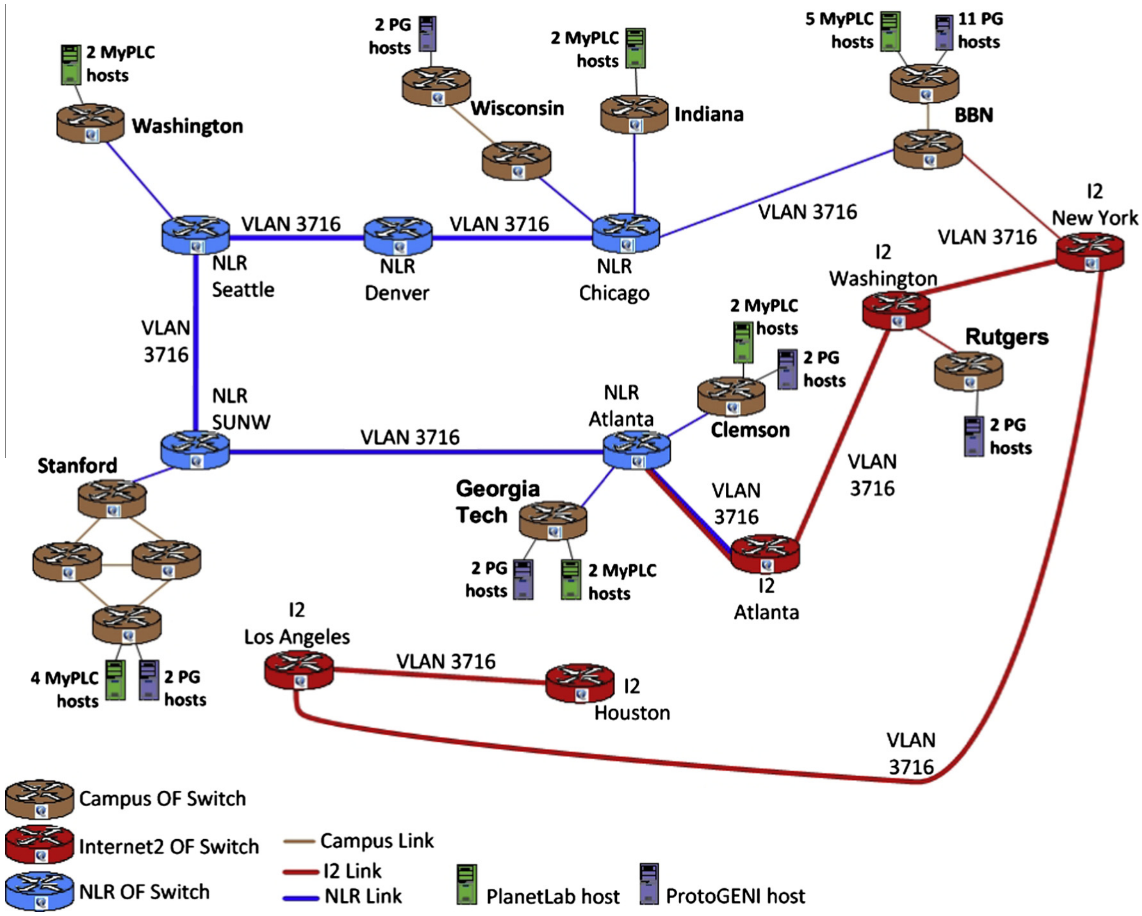


Fig. 4. Example nationwide OpenFlow-controlled VLAN.

differ chiefly with respect to the style of user interface presented, and the type(s) of resources each is best suited to manage. Example discovery and configuration tools in use by GENI experiments include the following.

- *Flack* is a web-based graphical resource discovery and experiment configuration tool, with a drag-and-drop interface for creating experimenter-specified topologies. Fig. 6 shows a Flack session. Boxes overlaid on the map show the locations of available resources, color-coded to indicate the aggregate manager offering these resources. The user may click through to more detailed information or drag-and-drop resources onto a canvas to design a slice [23].
- *omni* is a low-level tool, presenting a command-line interface that closely mimics the aggregate manager API calls and supporting scripted operation.
- *Myslice* is a web-based resource discovery tool, with an emphasis on identifying the compute resources best suited for a particular experiment and providing plugin-expandable query and interfaces.

Depending on the tool and resources, advertisement rspec data may be managed in several ways. A straightforward approach supported by several tools (including Flack, omni, and Myslice) is simply to list available resources for browsing and selection, possibly incorporating filtering capabilities to keep resource lists manageable. Another approach avoids presenting the experimenter with raw resource lists. Instead the experimenter provides a graphical or textual resource description, which the tool then attempts automatically to map onto available resources. Flack and omni both support this approach. GENI configuration tools also support more sophisticated experiment configurations that incorporate resource parameter settings, as well as software and data to be loaded onto experiment resources.

At this point, the tool produces a *request rspec* identifying desired experiment resources and their configuration. This request is provided to the affected aggregate managers, which respond with a *manifest rspec*, providing detailed information on resources that have been allocated to the experimenter's slice. A copy of the manifest is also provided to the GENI clearinghouse, for administrative use and incident response situations.

2.2.2. Experiment execution and maintenance

During the running lifetime of an experiment, researchers accomplish basic execution and maintenance tasks

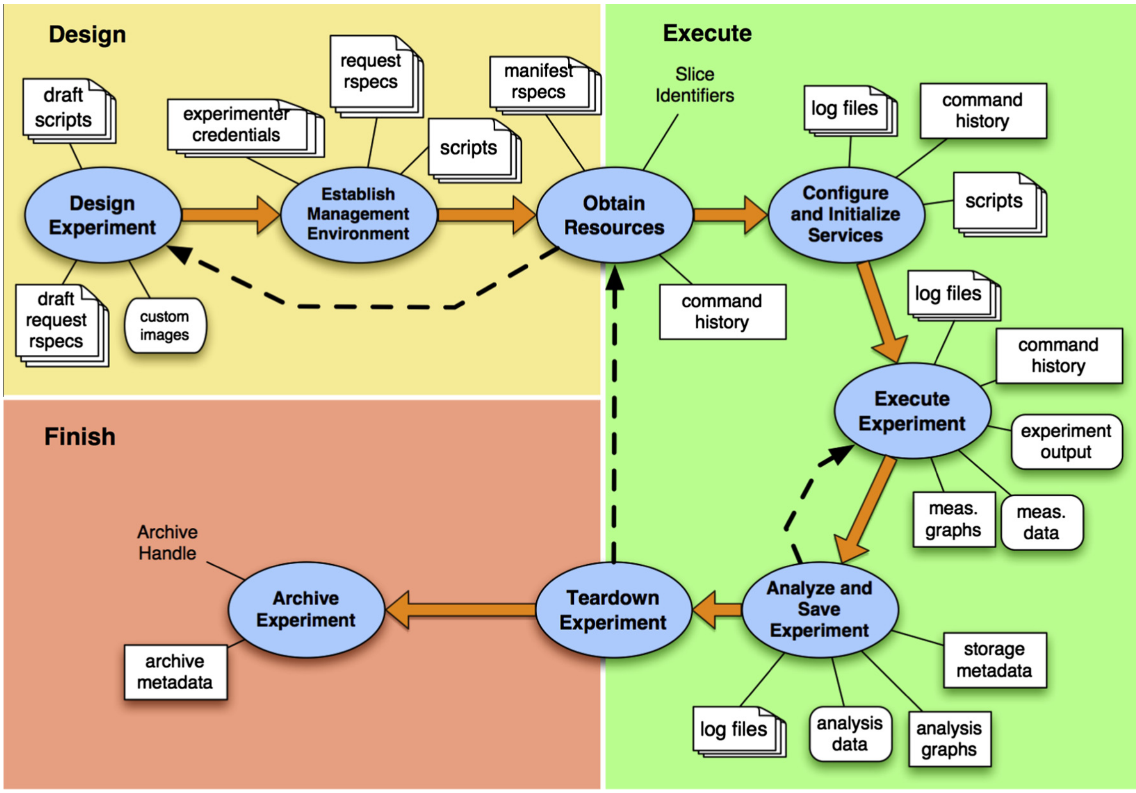


Fig. 5. GENI experiment lifecycle and work products.

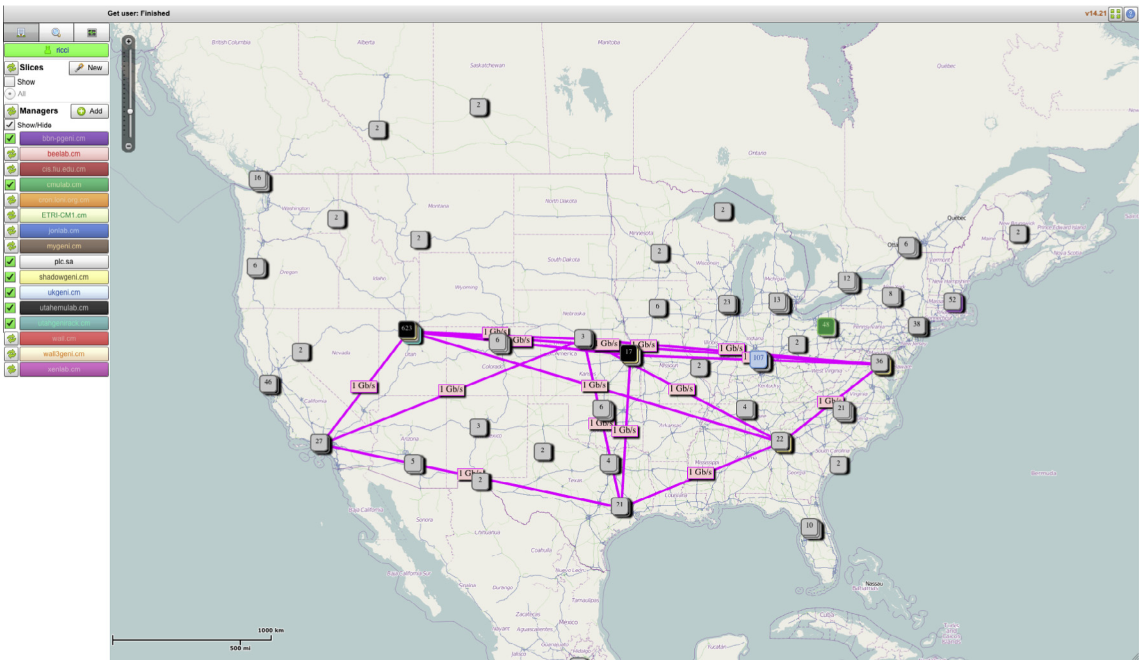


Fig. 6. Flack, a map-based resource discovery and configuration tool (courtesy Matt Strum, University of Utah).

through a combination of experiment support tools and direct access to resources.

Experiment management tools assist with potentially time-consuming and error-prone aspects inherent in running experiments and maintaining configurations across multiple resources. Similarly, they can be invaluable in reducing unintended configuration variations across multiple experimental trials. Example execution and maintenance tools include the following.

- Gush (GENI User Shell) is an execution management system, presenting experimenters with a command line or scripted interface for executing computation on a potentially large collection of compute devices [24].
- Stork is a tool set for experiment deployment and management, including capabilities to configure software installations on multiple computers in a slice [25].
- OMF is a control, measurement, and management framework for testbed-based experiments [26]. OMF is employed in several GENI aggregates and tools (e.g., LabWiki, see Fig. 7), as well as other testbeds.

While experiment management tools are helpful for many coordination and configuration tasks, researchers often find it convenient to perform simple execution and management tasks via direct access to experiment resources. The most common example is secure shell (ssh) access to a slice's compute resources. GENI resources generally permit standard user and administrator ("root") access within virtual containers, both via public key authentication. Experimenters may log in and simply run experiment commands. They may also install needed software, which may bootstrap further management tools by supporting additional services. It is a common usage pattern to install an Apache web server or mysql database within a slice, and then employ a standard web browser or mysql client to observe and control the experiment. In this fashion, GENI experimenters gain access to an open-ended universe of familiar tools.

3. Key deployment concepts and infrastructure support

The fundamental unit of GENI expansion is the *GENI-enabled campus*. To become GENI-enabled, a campus deploys the GENI technologies: GENI racks, OpenFlow/SDN, and possibly WiMAX on (a portion of) its campus and department backbones. The campus also agrees to certain administrative steps, including making resources available to local and remote researchers on an ongoing basis. These actions require active participation on campus from both research faculty and the office of the chief information officer (CIO).

3.1. Elements of a GENI campus deployment

The initial phases of GENI project execution involved the exploratory development of a relatively large number of candidate technologies for inclusion in the emerging GENI prototype deployment. Dozens of participating organizations worked together to identify, develop, and validate these candidate technologies, under the organizing framework of *clusters* (see Fig. 8). A cluster incorporated a number of collaborating projects, generally organized around a common *control framework*, the suite of software tools and interfaces that manage testbed resources.

This exploratory process served to better articulate the key desired capabilities that should be emphasized in a larger GENI deployment. In many cases, these capabilities had earlier instantiations in the control frameworks and constituent projects comprising the project clusters. For example, the current InstaGENI implementation of GENI racks is built on ProtoGENI's software base and will also

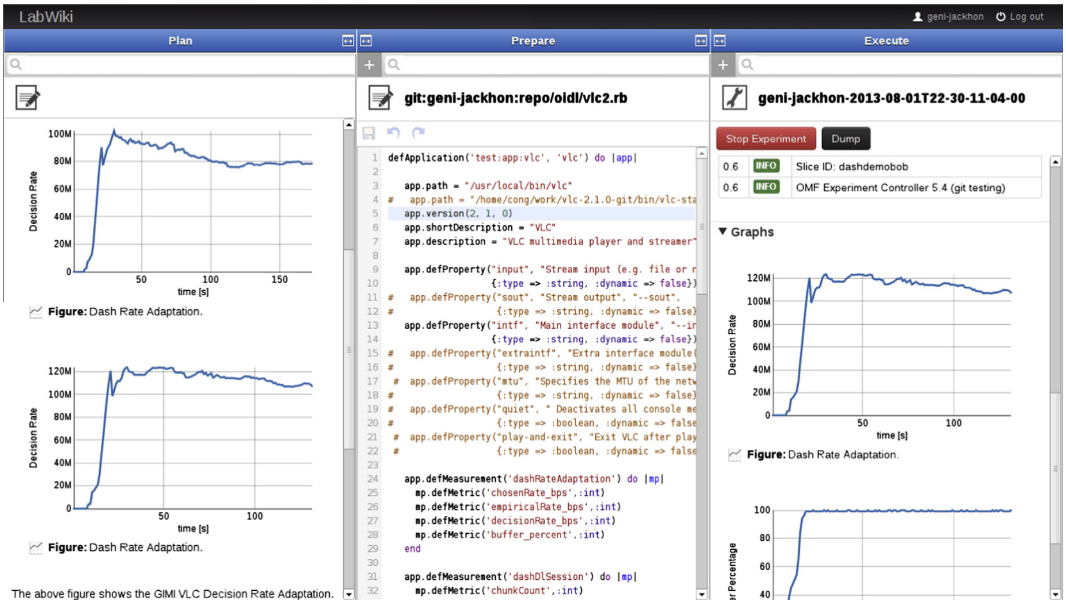


Fig. 7. Example experiment execution with LabWiki tool.

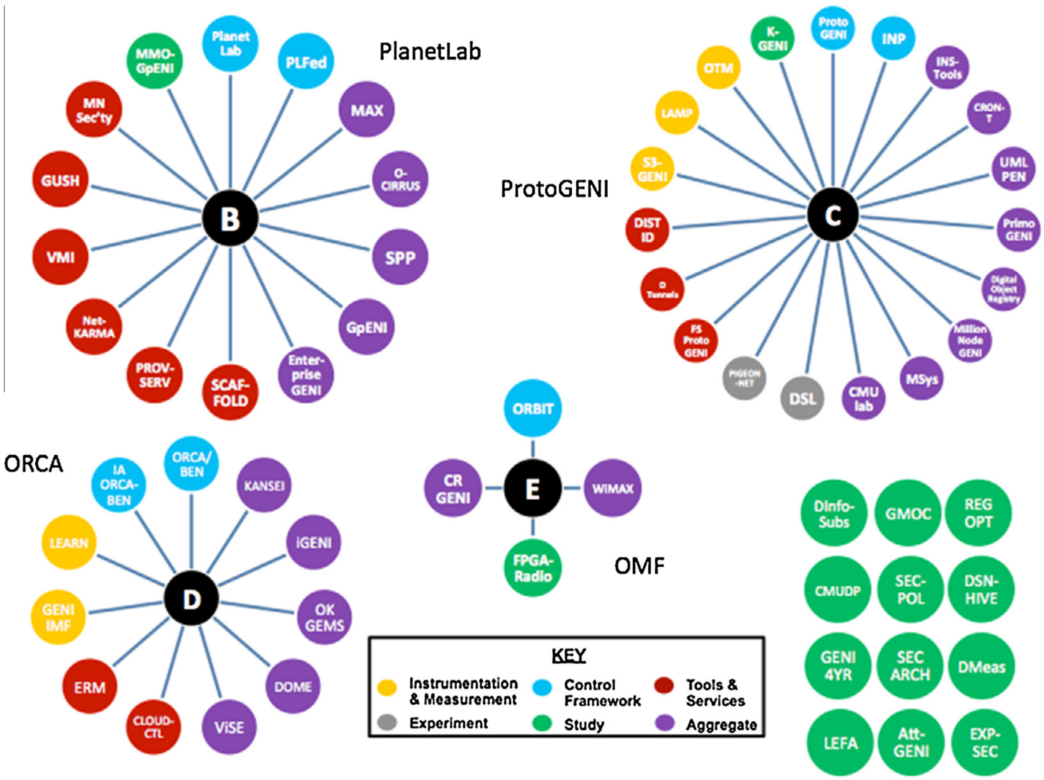


Fig. 8. GENI project clusters, circa 2010.

incorporate PlanetLab-style virtualization. Similarly, the ExoGENI rack implementation is built on the ORCA code base. Experimenter tools based on OMF are commonly used to manage GENI experiments.

In preparation for a larger rollout, the GENI development community focused additional effort on ensuring interoperability between control frameworks, and on consolidating the key capabilities into a smaller number of readily deployable components. This effort crystalized into the standard set of technologies now being deployed more widely. These elements are discussed in more detail in Section 3.4 below. This particular set of components was chosen to achieve a degree of consistency across the current GENI expansion. Future expansions may incorporate different capability in the deployed hardware base, while maintaining a base level of interoperability. By supporting an open software platform for tool developers, GENI expects to encourage continuous innovation and a growing ecosystem of software tools.

3.2. The role of the campus CIO

CIO collaboration is essential because end-to-end data paths between GENI components at different locations will have to traverse campus as well as regional/national R&E backbones. CIOs control the former and have key leadership roles in organizations that operate the latter. Hence, CIO participation is required in order to achieve the GENI goal of a national-scale widely distributed testbed for network science research.

To obtain support from CIOs, GENI has undertaken a program organized by one of the authors to educate university IT leadership on the potential benefits to their campuses of the GENI underlying technologies.[2] To date over 50 universities have participated in this program. They have been successful to a great extent because these technologies also offer the possibility of supporting enhanced campus network management capabilities as well as enabling more effective and cost-effective approaches to network security. These are but two of the potential benefits of OpenFlow/SDN campus network deployments.

Because it is understood that this technology is immature, the GENI program is encouraging initial experimental deployments of carefully monitored components of campus backbones. A key goal of GENI's current expansion phase is to achieve initial deployments of these technologies, in parallel with or integrated with production networks on over forty campuses. With the concurrent deployment of these technologies on regional and (inter)-national R&E backbones, this deployment will result in a revolutionary new programmable, virtualized, distributed collection of resources (networks, computers and storage) a global scale "deeply programmable cloud" that will support the GENI research mission and as well as enabling research and education in a wide variety of areas (e.g., domain science, big data/data mining, cloud-based applications, security).

[2] Dr. Landweber leads this effort. Selected archives are available at http://groups.geni.net/geni/wiki/OtherMeetings.

With NSF support, GENI has undertaken to increase the number of GENI-enabled campuses. The GENI program for accomplishing this includes a series of workshops for CIOs and other senior campus IT staff. In addition, teams consisting of a senior campus IT manager and a network engineer have been visiting campuses that have participated in the workshops to advise them on deployment strategies, network design, and equipment choices. The final component of the campus IT program has involved technical training on the GENI technologies for network engineers from these campuses.

The next phase of GENI expansion is still in the planning stage, but it is tentatively anticipated to overlap the current phase and increase the overall scale to 100–200 campuses.

3.3. Administrative steps to GENI-enabling

The key administrative goals in GENI-enabling a campus are:

- To establish between GENI and the campus a trust relationship, on which experimenters may rely.
- To establish and maintain resources available to the research community.
- To provide for local management and policy control over on-campus resources.
- To establish procedures for recognition of and response to incidents.

Steps toward these goals are laid out in an *aggregate provider's agreement*, which describes the relationship between GENI and a GENI-enabled campus [27]. These steps include providing an accurate catalog of resources available to experimenters, monitoring their security and stability, and communicating with other GENI-enabled campuses and the GENI Meta-Operations Center (GMOC) [28] to identify and respond to security incidents. By necessity, the campus CIO staff retains direct management authority over resources located on campus, both to protect the campus network and to ensure the availability of resources for local purposes.

The clear benefit to experimenters is the availability of additional resources. Each GENI-enabled campus provides new computation, storage, and network resources. In addition, each campus provides a geographically distinct deep programmability site, offering richer topology options for experiment designers.

GENI-enabled campuses benefit from the ability to "live in the future." By federating their existing computing and networking research infrastructures with GENI, they connect directly with other leading institutions. Researchers on campus may readily join forces with collaborating researchers at other institutions, as well as drawing upon GENI resources. If policy permits, the larger university community may have access to experimental services via end-user opt-in.

Both constituencies, experimenters and aggregate providers, benefit from the establishment of trust relationships between GENI and the other group. Reliance on these relationships helps reduce an intractable $m \times n$ web of agreements among m aggregates and n experimenters to a much more manageable $m + n$.

3.4. Technical steps to GENI-enabling

From a technology viewpoint, GENI-enabling a campus requires identifying the set of resources to be connected,

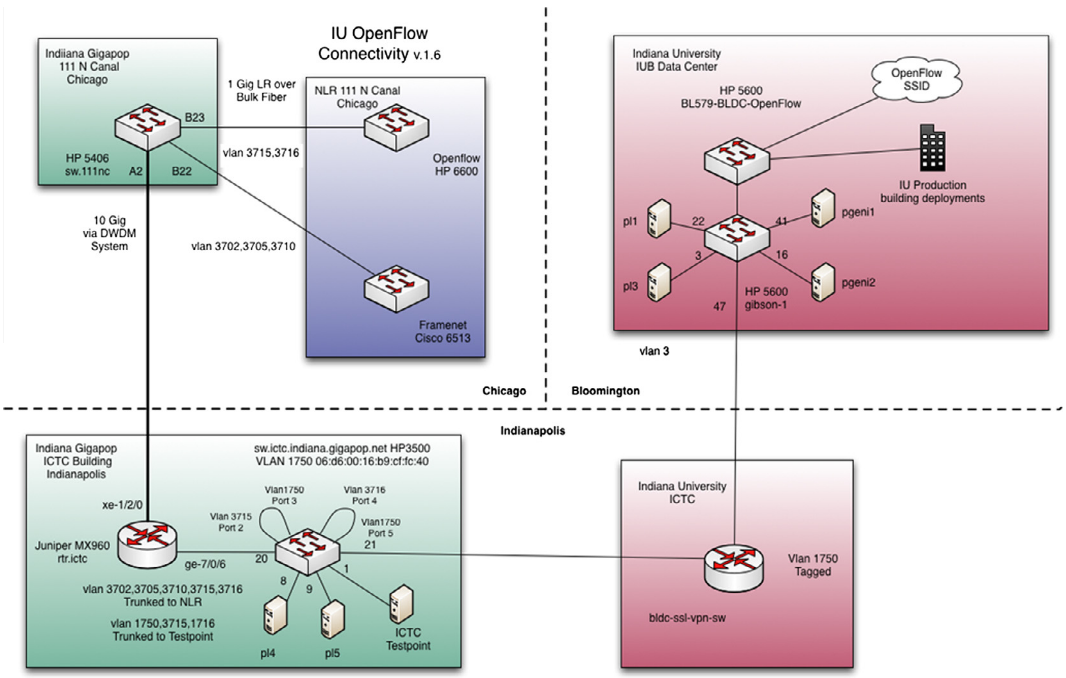


Fig. 9. GENI Deployment at Indiana University Campus (courtesy Chris Small, Indiana University).

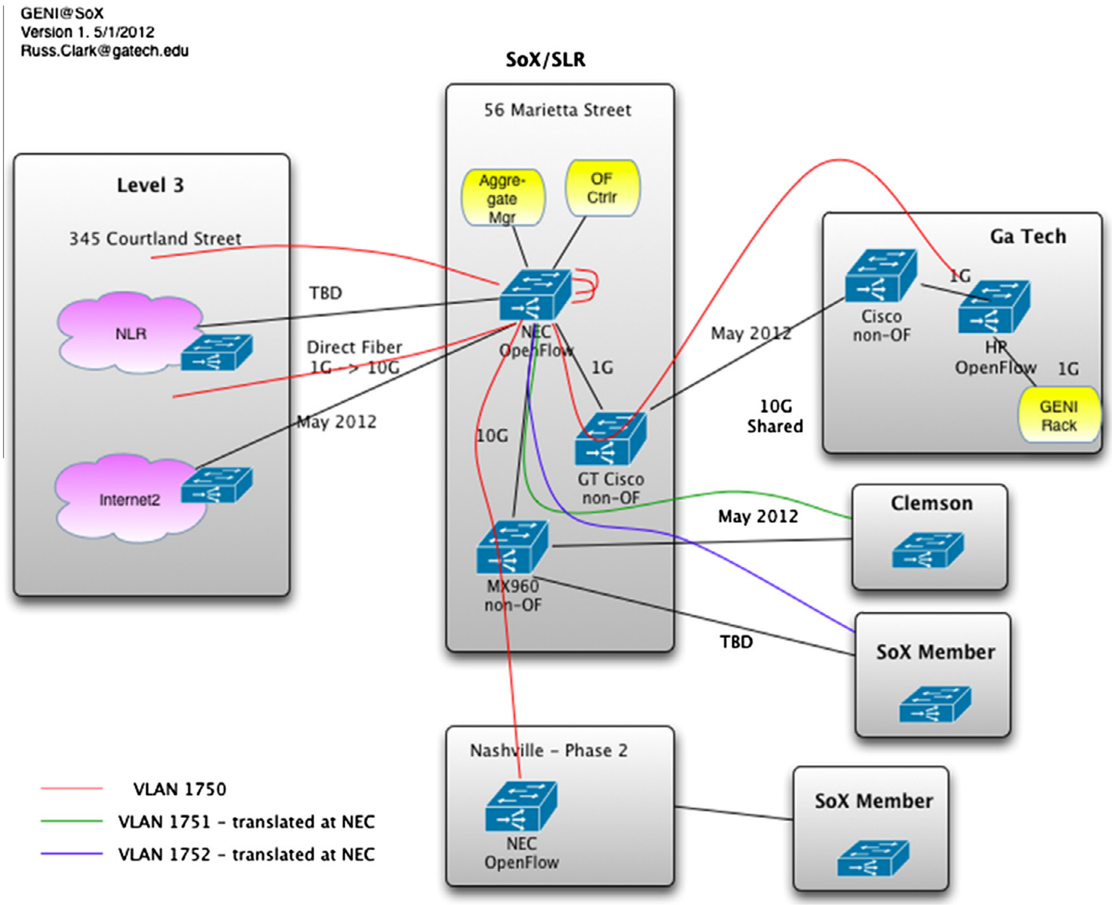


Fig. 10. GENI campus connections to Georgia Institute of Technology and Clemson University via Southern Crossroads (SoX) Regional Network (courtesy Russ Clark, Georgia Institute of Technology).

and engineering a suitable network configuration. Because each campus has unique needs, a collaborative design process is used to arrive at a suitable design, such as that shown in Fig. 9. Different GENI resources are deployed at different GENI-enabled campuses, however the key resource types driving campus expansion are GENI racks, SDNs, and WiMAX, discussed below.

At each GENI campus, arrangements must be made to connect GENI resources on a campus to the campus edge. One common design pattern emerging in the current deployment is placing a GENI rack, and possibly additional OpenFlow-capable switches, within or adjacent to a "Science DMZ," part of the campus network optimized for high-performance science applications. Another approach, currently being pursued at the University of Utah, is to adopt a sliced architecture for all or part of the campus backbone. This slicing can be accomplished using emerging technologies such as SDN or with older technologies such as VLANs or MPLS. No matter which strategy is employed, the end goal is to establish GENI resource connectivity to the edge of campus with the "cleanest" path possible.

In addition to designing the campus network configuration, a GENI-enabled campus must also arrange connectivity to other GENI sites. Typically, a campus will connect to one or both of the research backbones through a regional network. An example configuration is shown in Fig. 10.

3.4.1. GENI racks

A GENI rack (see Fig. 11) is the basic deployment unit of GENI computation and storage resources in a GENI-enabled campus. From a hardware standpoint, a rack comprises multiple compute nodes, disk-based persistent storage, and OpenFlow switches. As the name implies, these components are housed in a single rack, with appropriate power distribution and cabling.

Over this hardware substrate, the rack runs a software stack that presents the AM API to experimenter tools. The Aggregate Manager and associated administrative functions are housed on one physical host in the rack. While clearinghouse functions within GENI are conceptually centralized, they may be distributed for redundancy of for administrative purposes. The current GENI deployment employs two primary clearinghouses, in Salt Lake City, Utah and Cambridge, Massachusetts, respectively. Control plane connectivity among an experimenter tool, a GENI rack, and a clearinghouse is via Internet. Additional information on the software components of GENI and their relationships is given in Section 3.5 below.

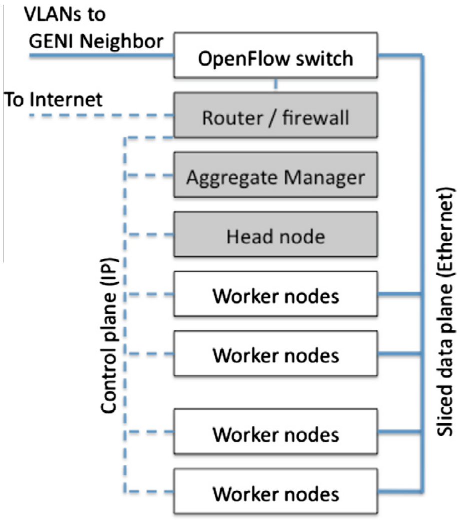


Fig. 11. GENI rack schematic.

The basic resources exported to experimenters are bare metal or virtual machines, with the specific virtualization mechanism varying depending on the rack implementation and experimenter choice. Today, experimenters reach the control interfaces of these virtual machines via standard Internet connectivity, possibly including network address translation (NAT) at the rack's control switch. Allocated virtual machines may be interconnected along the data plane via an experimenter-selected combination of IP, vanilla VLANs, and OpenFlow-managed VLANs.

Slices typically include resources from multiple GENI aggregates. The process of *stitching* interconnects these resources into a wide range of potentially large network topologies at layer two. Initial GENI rack deployments incorporate only a limited form of stitching, generally requiring manual intervention or restricted to slices that include only a subset of GENI resource types. An automated GENI-wide stitching design is currently underway.

There are two current implementations of GENI racks. They share the fundamental characteristics described above. The ExoGENI design, built by a team lead by RENCI (Renaissance Computing Institute), is emphasizing performance, building on IBM hardware and an OpenStack software platform [5]. The InstaGENI rack, built by a HP Labs-led team, is emphasizing affordability and building on HP hardware and a combined ProtoGENI and PlanetLab software base [29,30].

3.4.2. Software-defined networks

In addition to the SDN capability found in GENI racks, GENI-enabled campuses also deploy within the campus network a number of switches supporting SDNs. The size and specifics of the SDN-capable network on campus varies to meet site-specific goals, and its design is often the most challenging part of the planning process for a GENI-enabled campus. However, there are certain common elements.

The campus SDN-capable network must, on the one hand, reach experimenters and potential opt-in users, and on the other hand, connect the campus to the external GENI SDN infrastructure. This latter goal is achieved by providing for connectivity to GENI backbone nodes on one or both of the Internet2 and National Lambda Rail research networks, typically via a regional network. The former goal, SDN reach into the campus, can be achieved incrementally, beginning with a modest network focused on likely research users, and eventually expanding to reach more of the campus. SDN deployment is generally in the form of standard commercial switches running OpenFlow firmware, which are available from several major vendors.

Campus network resources are grouped into GENI aggregates, and an aggregate manager mediates experimenter access to virtualized network resources. Typically, a slice includes regions of *flow space*, sets of actual or potential network flows through these network devices [31]. The flow space is high dimensional, but common dimensions used for slicing flow space include parameters at layer one (e.g. switch port), layer two (e.g. source and destination MAC address, eth_type), layer three (e.g. source and destination IP address) and layer four (e.g. source and destination TCP port number). Flows falling within the slice's flow space are treated in accordance with the directives of an OpenFlow controller, which may run on a compute resource within the slice or elsewhere [32].

Fallback options are available to extend programmable network features to resources on or off campus that are not directly connected to the SDN-capable portion of the campus network. The most straightforward option is to extend a VLAN or tunnel connecting these resources to an OpenFlow-controlled portion of the network. An experimenter may then exert OpenFlow-based control over the traffic once it enters the SDN-capable network. An experimenter may augment this approach by employing software-based routers (e.g. Click or XORP modular software routers) in the slice's data path, at the cost of significantly reduced performance.

3.4.3. WiMAX

Some GENI-enabled campuses deploy WiMAX (Worldwide Interoperability for Microwave Access) base stations and sector antennae. WiMAX provides an avenue for the inclusion of researcher-owned and operated cellular systems in GENI. These resources have proven suitable for a range of experiments. These include wireless-specific research, network research with a focus on mobility, and experiments in which wireless just happens to be an available connectivity option.

As with other GENI resources, access to WiMAX sources is virtualized throughout. Virtualization in the wireless network is via WiMAX channels, to which specific mobile devices are mapped on the mobile side. Virtualization software within the GENI WiMAX base station maps a slice's WiMAX channels onto VLANs and then to virtual machines, which provide the desired service to the mobile device [33]. The resulting effect is essentially a sliced cellular system extending into a GENI slice.

3.5. Federation

GENI is built using a *federated* model: that is, it is built out of resources that are owned and operated by different providers. Confining experimental infrastructure to a single owner or administrative domain necessarily limits its scale and diversity; by using a federated model, in which multiple domains cooperate to provide a coherent facility, GENI embraces contributions from a range of networks. Each resource provider maintains local autonomy, and the ability to set policy for use of its own resources. At the same time, GENI's federation architecture provides a framework for establishment of mutual trust and collaboration.

3.5.1. Assembling GENI via federation

GENI's internal architecture is fundamentally one of collaboration between tools and aggregates [34]. Tools act as proxies for experimenters, managing and controlling GENI resources. Aggregates represent the ultimate owners of GENI resources, lending them to experimenters for incorporation into slices.

Two additional players are introduced into the architecture to assist in providing protection assurances desired by experimenters and operations staff. The *GENI Meta-Operations Center* (GMOC) staff monitor the health and proper operation of GENI and facilitate response to incidents. The *clearinghouse* provides required trust, authentication, accountability, and authorization services for GENI. The relationships among these entities and services are presented in Fig. 12.

By basing its architecture on a collection of open interfaces and trust relationships, GENI remains open to innovation by tool and aggregate developers, rather than mandating a single, "winning" control framework or toolset. This opportunity for innovation extends even to the degree of coupling among components. One example is the step of mapping an experimenter's desired network topology onto available GENI resources. In addition to fully automated embedding algorithms contained within some aggregate managers, various GENI tool and aggregate combinations currently support at least three different types of control over this process. In the simplest and least flexible approach, an experimenter identifies specific resources, by name, in a *fully bound* request rspec. If the resources are not available, the request fails. In an *unbound* rspec, only the type of resource is identified, and the aggregate manager searches for any appropriate resources. A completely different approach relies on resource elasticity. For some kinds of virtualized resources, e.g. lightweight virtual machines, an aggregate manager may simply mint a fresh resource as needed. Not only is each of these approaches supported by the GENI architecture, but it is also possible for an innovative developer to implement a new tool that implements an improved approach to resource selection. Such a tool could internally transform an unbound rspec or other high-level description to a fully or partially bound rspec for fulfillment by one or more aggregate managers. Overall, GENI's open architecture has been successful in promoting a lively bazaar of development, with multiple teams working, sometimes collaboratively, sometimes independently on new tools for resource selection, experiment management, instrumentation, and even a general aggregate manager framework.

3.5.2. GENI in collaborative federations

The GENI project is working closely with peer projects worldwide to pursue collaborative federation efforts. Through these efforts, testbed developers and users are better able to identify those areas where federation is well understood and those areas where additional design efforts and coordination are required. GENI participates in significant international federations, such as TransCloud and the Slice Around the World effort. TransCloud interconnects multiple sites in the US, Germany, and the Netherlands, using connectivity provided by CAVEWave, StarLight [35], NetherLight [36], DFN (German National Research and Education Network) [37], NLR (National Lambda Rail) [38], and GLIF (Global Lambda Integrated Facility) [39] into a prototype shared, distributed cloud computation service [40]. The Slice Around the World initiative is beginning work to federate network research centers and research labs that are participating in multiple next generation networking activities worldwide.

With production cloud computing services widely available through commercial providers, it is increasingly important to include these cloud service providers in research testbed plans for federation. The Data Intensive Cloud (DICloud) project applies such an approach, federating cloud computing resources provided by Amazon's Elas-

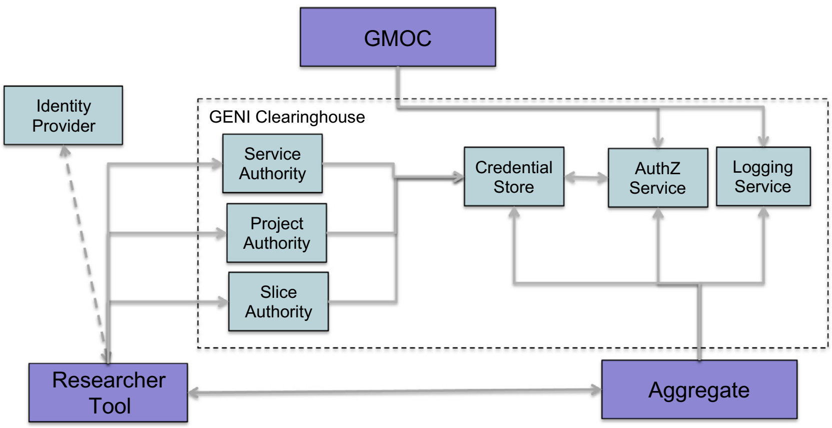


Fig. 12. GENI federation services relationships.

tic Compute Cloud (EC2) and Simple Storage Service (S3) with GENI sensing, computation, and network resources to develop a flexible processing approach for weather radar with the ability to scale adaptively in the case of severe weather [41].

4. Related work

To the extent possible, the GENI project has worked to make the development, deployment, and support of a working GENI testbed chiefly an engineering effort, relying heavily on concepts developed in other testbed efforts. Furthermore, because GENI's deployment strategy is one of federation, the current GENI testbed reuses not only ideas from other testbed efforts, but also the actual hardware and software components developed by federated testbeds. In addition to efforts discussed below, there are other peer projects in progress worldwide, notably including projects in Europe, Korea, Japan, and Brazil, that are important collaborators for GENI.

4.1. PlanetLab

PlanetLab is a global research network that supports the development of new network services [42]. It is managed by a consortium of academic, industry, and government institutions for the benefit of the research community. Many key GENI components trace their origins, either in code or in concept, back to roots in PlanetLab [3]. The Slice-Based Federation Architecture (SFA), an approach to federation growing primarily out of the PlanetLab experience, is the basis for much of the GENI API [43]. In particular, the aggregate manager API is adapted from the SFA.

Other ideas pioneered in PlanetLab and prominent in GENI include:

- Use of a common overlay network infrastructure as a mechanism for validating and deploying novel network services.
- Providing a widely available leasing service for networked compute resources.
- The slice concept.
- Use of lightweight host virtualization for efficient support of many long-lived services.

As an overlay network deployed over the global Internet, PlanetLab is intended to support experiments that run at or above layer three. This design decision permits PlanetLab to present a simple, well-understood, and powerful network abstraction to potential experimenters. More recently, some PlanetLab variants, such as VINI (Virtual Network Infrastructure) [44], VICCI [45], and GpENI (Great Plains Environment for Network Innovation) [46,47] are reaching into lower layers of the network stack.

In addition to architectural contributions, PlanetLab also shares resources with GENI by supporting the AM API and exporting a GENI aggregate. Via this interface, experimenters have access to PlanetLab's worldwide network of virtualized compute resources. Some experimenter tools originally developed with PlanetLab in mind were readily generalized into a GENI context. Examples include Gush (GENI User Shell) [24] and the Stork [25] and Raven [48] experiment management tools.

4.2. Emulab and ProtoGENI

Emulab is a network testbed, incorporating a number of centrally managed general-purpose computers and network resources [49]. The initial and largest Emulab, which includes several hundred nodes, is located at the University of Utah, where it is managed by the Flux Research Group. Emulab software is available on an open source basis, and has been used to stand up dozens of Emulabs worldwide. The Emulab software also forms the basis for ProtoGENI, a GENI prototype and control framework, which extends the Emulab model to bridge multiple physical sites and support the GENI API [6].

The fundamental GENI abstraction of multiple general-purpose computers interconnected by layer two networks in experimenter-specified topologies draws directly on the basic capabilities of Emulab. In addition, several of the specifics of the GENI API derive basic ideas and/or implementation from Emulab and ProtoGENI. Notable among these are the content and format of GENI rspecs, and portions of the experiment lifecycle. In addition, the ProtoGENI implementation of multiple sites interconnected at layer two over Internet2 links is a major influence on GENI stitching.

The primary ProtoGENI site at Utah and several other ProtoGENI sites also share resources with GENI by exporting aggregates that speak the GENI API. The primary resources exported are bare metal computers connected by layer two and layer three networks, although virtual machines are also available. One of the GENI rack implementations, InstaGENI, is built on Emulab/ProtoGENI software.

4.3. OpenFlow and SDN

Software-Defined Networking (SDN) provides an open standard mechanism for separating the control and data planes and exerting flow-specific control over network behavior [8]. OpenFlow is a vendor-neutral SDN protocol that provides for the implementation of this control [9,50].

The OpenFlow standard has gained substantial traction, and most major vendors of network equipment now offer OpenFlow-enabled products. Several testbeds have been developed or are currently under development, connecting significant numbers of devices into meaningful OpenFlow-controlled networks. One example is the TangoGENI network discussed in Section 2.1.4 above. Another is a 10 Gbps, multi-vendor OpenFlow network testbed deployed within the SCinet research sandbox during the SC11 conference in November 2011. This network supported multiple disruptive experiments across a nationwide network, extending into the conference facility. Currently under development, the Network Development and Deployment Initiative (NDDI) is a combined effort by Stanford University, Indiana University, and Internet2 to deliver SDN capabilities across a wide subset of the Internet2 backbone and member institutions.

Deployment of OpenFlow in a GENI-enabled campus has proven to be a very appealing proposition. Because

OpenFlow provides mechanisms for experimental and production traffic to coexist within the same network, these deployments offer benefits to both the experimenter and CIO staff constituencies on campus. By federating OpenFlow deployments across multiple campuses and backbone resources, researchers gain access to more realistic and capable networks on which to run their experiments.

4.4. ORBIT

ORBIT (Open Access Research Testbed for Next-Generation Wireless Networks) is a wireless network testbed with both emulation and field-trial capabilities [7]. The hardware layout includes a grid of 400 WiFi nodes. Because ORBIT is located at Rutgers, a GENI-enabled campus, GENI experimenters are able to include associated resources in their experiments, such as the MobilityFirst experiments discussed above.

Because of the inherent variability associated with wireless experiments, ORBIT has devoted substantial effort toward reproducibility. The radio grid emulator enables reproducibility in the wireless networking aspects of experiments. Equally important in a unique and often oversubscribed testbed, the Orbit Management Framework (OMF) provides mechanisms to rapidly and reliably deploy and execute experiments [51] and gather the resulting measurement data. Key aspects of OMF are finding their way into the GENI experiment lifecycle, chiefly via experimenter tools and instrumentation features [52].

4.5. FIRE

The FIRE (Future Internet Research and Experimentation) Initiative was launched at the beginning of 2007 as part of Framework Programme 7 (FP7) and consists of several affiliated projects. Several of these projects offer experimentation testbeds to network researchers [53].

The FEDERICA project is a European project to create a scalable, Europe-wide, clean slate, infrastructure to support future Internet experiments. FEDERICA and GENI support very similar virtualization models, with sliced infrastructure extending through computational and networking resources [54,55].

The OFELIA (OpenFlow in Europe: Linking Infrastructure and Applications) project is deploying a federated, heterogeneous testbed environment, with OpenFlow as a common control paradigm [56,57].

The BonFIRE cloud experimentation facility includes six geographically distributed testbeds across Europe, offering heterogeneous computing, storage, and networking resources. The BonFIRE Virtual Wall emulation environment is based on Emulab control software. The resource model is Infrastructure as a Service (IaaS), and is intended for experiments running in cloud environments [58,59].

4.6. G-Lab

The experimental facility of the German-Lab (G-Lab) project spans six German universities: Würzburg, Kaiserslautern, Berlin, München, Karlsruhe, and Darmstadt [60,61]. G-Lab includes approximately two hundred general-purpose computers, along with high-quality IP connectivity among the various sites. Experimenters generally construct their topologies of interest by connecting virtual machines over IP. Multiple virtualization techniques are supported, with the default vserver virtualization model provided by a private PlanetLab, which is controlled from the G-Lab Kaiserslautern site. The facility also supports deployment of custom boot images for access to bare metal computational resources. Because of the multiple virtualization options, experimenters have the choice of essentially a high-performance PlanetLab distribution, or more customized topology controls. G-Lab has also been incorporated into federated configurations with other testbeds, including GENI, for larger, international experimental topologies.

4.7. JGN-X, VNode, and StarBED3

The Japan Gigabit Network (JGN) network testbed and StarBED (Hokuriku Research Center) emulation testbed are collaborating efforts intended to support innovation in next-generation networks. The most recent versions of these testbeds are JGN-X (JGN – Extreme) and StarBED3 (StarBED Cubic), respectively. The VNode project is a joint effort among the University of Tokyo, NICT, NTT, NEC, Hitachi, and Fujitsu to enable network virtualization using hardware based on production routers.

JGN is a testbed environment for implementation and test deployment of next-generation network technologies developed by NICT (National Information and Communications Technology). JGN-X supports experimenter access at multiple layers. Layer 3 service includes both IPv4 and IPv6 at bandwidths from 100 Mbps to 10 Gbps, with traffic separation available via VLAN. Layer 2 Ethernet service with similar capabilities is available for non-IP experiments. Layer 1 optical testbed services (dark fiber) are available across a total of approximately six hundred kilometers of fiber [62].

The VNode project is approaching network virtualization based on a new slicing mechanism that is based on lessons learned from PlanetLab and CoreLab experiences. The VNode architecture provides for explicit separation between router programmability and network connectivity. Four VNodes have been deployed in JGN-X since September 2010. This infrastructure supports a deep programmability notion, and is capable of supporting multiple simultaneous slices, running independent protocols [63].

StarBED (Hokuriku Research Center) is a simulation/emulation testbed incorporating approximately one thousand general-purpose computers, with redundant connectivity into a large switch cluster. Experimenters acquire compute resources as uninitialized PCs. These resources are then imaged from scratch and experimental software launched under the control of a specialized experiment scripting capability, SpringOS. Virtualization of compute resources is available via VMWare [64].

4.8. KREONET and K-GENI

KREONET (Korea Research Environment Open NETwork) is a national R&D network, managed by KISTI (Korea

Institute of Science and Technology Information). KREONET connects twelve regional network centers and over two hundred participating institutions. The K-GENI project, a collaboration between GENI and KREONET researchers was initiated in 2009. It has explored international testbed federation, as well as the challenges of wide-area distributed network monitoring and meta-operations [65].

5. Conclusion and future work

The GENI virtual laboratory provides a flexible and capable environment for the exploration of disruptive networking technology. The key concepts of sliceability and deep programmability significantly expand the opportunities for experiments in network science, services, security, and related fields.

The current GENI deployment is actively growing to over fifty sites in the current expansion phase and is expected to continue growing to an at scale size of one hundred to two hundred sites. As GENI gains additional deeply programmable sites, the sophistication of available network topologies and the range of possible at scale experiments grow dramatically. Future efforts will continue along the growth path through the GENI-enabled campus model.

In addition, experimenters and operators will need more capable management tools to make best use of the growing GENI-enabled infrastructure. A core set of basic experimenter tools is currently available. However, additional work is needed to solidify this base and expand it to provide broader capability.

Initial planning is also beginning for the longer-term future oversight of GENI, likely based on a community governance model.

References

[1] T. Anderson, L. Peterson, S. Shenker, J. Turner, Overcoming the Internet impasse through virtualization, Computer 38 (4) (2005) 34–41.

[2] J.S. Turner, D.E. Taylor, Diversifying the Internet, in: Global Telecommunications Conference, vol. 2, 2005.

[3] Larry Peterson, Tom Anderson, David Culler, Timothy Roscoe, A blueprint for introducing disruptive technology into the Internet, in: ACM SIGCOMM Computer Communication Review, vol. 33, no. 1, January 2003, pp. 59–64.

[4] Brent Chun, David Culler, Timothy Roscoe, Andy Bavier, Larry Peterson, Mike Wawrzoniak, Mic Bowman, PlanetLab: an overlay testbed for broad-coverage services, in: ACM SIGCOMM Computer Communication Review, vol. 33, no. 3, July 2003, pp. 3–12.

[5] Ilia Baldine, Yufeng Xin, Anirban Mandal, Paul Ruth, Chris Heerman, Jeff Chase, ExoGENI: a multi-domain infrastructure-as-a-service testbed, in: 8th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM 2012).

[6] Robert Ricci, Jonathon Duerig, Leigh Stoller, Gary Wong, Srikanth Chikkulapelly, Woojin Seok, Designing a federated testbed as a distributed system, in: Proceedings of the 8th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, 2012.

[7] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, M. Singh, Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols, in: Wireless Communications and Networking Conference, vol. 3, 2005, pp. 1664–1669.

[8] Scott Shenker, The future of networking, and the past of protocols (Keynote Presentation), in: Open Networking Summit, Stanford, October 2011.

[9] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, Jonathan Turner, OpenFlow: enabling innovation in campus networks, in: SIGCOMM Computer Communication Review, vol. 38, no. 2, April 2008, pp. 69–74.

[10] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, M. Frans Kaashoek, The click modular router, ACM Trans. Comput. Syst. 18 (3) (2000) 263–297. August 2000.

[11] Jae Woo Lee, R. Francescangeli, J. Janak, S. Srinivasan, S.A. Baset, H. Schulzrinne, Z. Despotovic, W Kellerer, NetServ: Active Networking 2.0, in: IEEE International Conference on Communications Workshops, 2011.

[12] Jad Naous, Glen Gibb, Sara Bolouki, Nick McKeown, NetFPGA: reusable router architecture for experimental research, in: Proceedings of the ACM Workshop on Programmable Routers for Extensible Services of Tomorrow (PRESTO '08), New York, 2008, pp. 1–7.

[13] Jonathan S. Turner, Patrick Crowley, John DeHart, Amy Freestone, Brandon Heller, Fred Kuhns, Sailesh Kumar, John Lockwood, Jing Lu, Michael Wilson, Charles Wiseman, nd David Zar, Supercharging planetlab: a high performance, multi-application, overlay network platform, in: SIGCOMM Computer Communications Review, New York, vol. 37, 2007, pp. 85–96.

[14] P. Brighten Godfrey, Igor Ganichev, Scott Shenker, Ion Stoica, Pathlet routing, in: Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication (SIGCOMM '09), New York, 2009, pp. 111–122.

[15] Dipankar Raychaudhuri, Kiran Nagaraja, Arun Venkataramani, MobilityFirst: a robust and trustworthy mobility-centric architecture for the future Internet, ACM Mobile Comput. Commun. Rev. 16 (3) (2012) 2–13.

[16] S. Nelson, G. Bhanage, D Raychaudhuri, GSTAR: generalized storage-aware routing for MobilityFirst in the future mobile Internet, in: Proceedings of the 6th ACM International Workshop on Mobility in the Evolving Internet Architecture (MobiArch), New York, 2011, pp. 19–24.

[17] T. Vu, A. Zhang, Y. Baid, T. Nguyen, J. Fukuyama, R. Martin, D. Raychaudhuri, DMap: a shared hosting scheme for dynamic identifier to locator mappings in the global internet, in: Proceedings of the 32nd International Conference on Distributed Computing Systems (ICDCS 2012), Macao, 2011.

[18] Akihiro Nakao, Deeply Programmable Network Through Advanced Network Virtualization, in: IEICE International Symposium on Network Virtualization, 2012.

[19] Ivan Seskar, Kiran Nagaraja, Sam Nelson, Dipankar Raychaudhuri, MobilityFirst future internet architecture project, in: Proceedings of the 7th Asian Internet Engineering Conference (AINTEC '11), New York, 2011, pp. 1–3.

[20] Ashok Anand, Fahad Dogar, Dongsu Han, Boyan Li, Hyeontaek Lim, Michel Machado, Wenfei Wu, Aditya Akella, David G. Andersen, John W. Byers, Srinivasan Seshan, Peter Steenkiste, XIA: an architecture for an evolvable and trustworthy internet, in: Proceedings of the 10th ACM Workshop on Hot Topics in Networks (HotNets-X), New York.

[21] The NDN Project Team, Named Data Networking (NDN) Project, 2010, October. <http://www.named-data.net/ndn-proj.pdf>.

[22] GENI Aggregate Manager API Version 3. <http://groups.geni.net/geni/wiki/GAPI_AM_API_V3>.

[23] Jonathon Duerig, Robert Ricci, Leigh Stoller, Matt Strum, Gary Wong, Charles Carpenter, Zongming Fei, James Griffioen, Hussamuddin Nasir, Jeremy Reed, Xiongqi Wu, Getting started with GENI: a user tutorial, in: SIGCOMM Computer Communications Review, vol. 42, no. 1, January 2012, pp. 72–77.

[24] Jeannie Albrecht, Danny Yuxing Huang, Managing distributed applications using gush, in: Testbeds and Research Infrastructures. Development of Networks and Communities 6th International ICST Conference, TridentCom 2010, Berlin, Germany, May 18–20, 2010, Revised Selected Papers, 2010, pp. 401–411.

[25] Stork Project. <http://www.cs.arizona.edu/stork>.

[26] Thierry Rakotoarivelo, Maximilian Ott, Guillaume Jourjon, Ivan Seskar, OMF: a control and management framework for networking testbeds, in: SIGOPS Operating Systems Review, vol. 43, no. 4, January 2010, pp. 54–59.

[27] GENI Project Web Site, October 2010. <http://groups.geni.net/geni/attachment/wiki/ComprehensiveSecurityPgm/Aggregate%20Provider%20Agreement%20v3.pdf>.

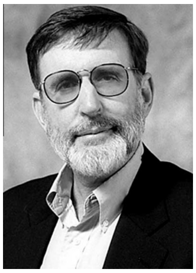
[28] GENI Meta-Operations Center, GENI Meta-Operations Center, 2013. <http://gmoc.grnoc.iu.edu>.

[29] Rick McGeer, Jessica Blaine, Nicki Watts, N. Krishnan, Andy Bavier, Robert Ricci, Joe Mambretti, GENI Project Wiki. <http://

groups.geni.net/geni/attachment/wiki/GENIRacksHome/Instageni-Racks/InstaGENI%20Design%20Document%20Rick%202-15%282%29.pdf>.
[30] Nicholas Bastin, Andy Bavier, Jessica Blaine, Jim H. Chen, Narayan K. Krishnan, Joe Mambretti, Robert Ricci, Nicki J. Watts, The InstaGENI Initiative: An Architecture for Distributed Systems and Advanced Programmable Networks, Comput. Netw. 61 (2014) 24–38.
[31] Rob Sherwood, Michael Chan, Adam Covington, Glen Gibb, Mario Flajslik, Nikhil Handigol, Te-Yuan Huang, Peyman Kazemian, Masayoshi Kobayashi, Jad Naous, Srinivasan Seetharaman, David Underhill, Tatsuya Yabe, Kok-Kiong Yap, Yiannis Yiakoumis, Hongyi Zeng, Guido Appenzeller, Ramesh Johari, Nick McKeown, Guru Parulkar, Carving research slices out of your production networks with OpenFlow, in: SIGCOMM Computer Communications Review, New York, vol. 40, 2010, pp. 129–130.
[32] OpenFlow Web Site, February, 2011. <http://www.openflow.org/documents/openflow-spec-v1.1.0.pdf>.
[33] G. Bhanage, I. Seskar, D Raychaudhuri, A service oriented experimentation framework for virtualized WiMAX systems, in: Proceedings of the 7th International ICST Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM 2011), Shanghai, 2011.
[34] GENI Architecture Team (Marshall Brinn and Robert Ricci, chairs), GENI Federation Software Architecture Document, March, 2012. <http://groups.geni.net/geni/attachment/wiki/GeniArchitectTeam/GENI%20Software%20Architecture%20v1.0.pdf>.
[35] Joel Mambretti, Overview of StarLight: An Advanced Facility for Research and Education at Northwestern University, April, 2004. <http://www.it.northwestern.edu/bin/docs/starlight_overview.pdf>.
[36] SURFnet, NetherLight, 2012. <http://www.surfnet.nl/en/thema/netherlight>.
[37] J. Pattloch, DFN: Deutsche Forschungsnetz, 2007–2013. <https://www.dfn.de/>.
[38] National LambdaRail, NLR: National LambdaRail, 2005–2010. <http://www.nlr.net>.
[39] Joe Mambretti, Jim Chen, Fei Yeh, Creating environments for innovation: designing and implementing advanced experimental network research testbeds based on the global lambda integrated facility and the StarLight exchange, Comput. Netw. 61 (2014) 118–131.
[40] Rick McGeer, Joe Mambretti, Paul Mueller, Chris Matthews, Chris Pearson, Yvonne Coady, Jim Chen, Fei Yeh, Andy Bavier, Marco Yuen, Jessica Blaine, Alvin AuYoung, Alex Snoeren, Justin Cappos, TransCloud: a distributed environment based on dynamic networking, in: Tenth GENI Engineering Conference, San Juan. <http://groups.geni.net/geni/attachment/wiki/IGENI/TransCloud%20Demo%20Presentation%20for%20GEC%2010%20Draft%20Version%2071.pdf>.
[41] Michael Zink, Prashant Shenoy, Jim Kurose, David Irwin, Emmanuel Cecchet, DiCloud Project Status Report, University of Massachusetts, Amherst, MA, 2010.
[42] Andy Bavier, Mic Bowman, Brent Chun, David Culler, Scott Karlin, Steve Muir, Larry Peterson, Timothy Roscoe, Tammo Spaline, Mike Wawrzoniak, Operating system support for planetary-scale network services, in: Proceedings of the 1st Conference on Networked Systems Design and Implementation (NSDI '04), Berkeley, vol. 1, 2004, pp. 19.
[43] Larry Peterson, Soner Sevinc, Jay Lepreau, Robert Ricci, John Wroclawski, Ted Faber, Stephen Schwab, and Scott Baker, Slice-Based Facility Architecture, April, 2009. <http://svn.planet-lab.org/attachment/wiki/WikiStart/sfa.pdf>.
[44] VINI: A Virtual Network Infrastructure. <http://vini-veritas.net/>.
[45] Larry Peterson, Andy Bavier, Sapan Bhatia, VICCI: A Programmable Cloud-Computing Research Testbed, Department of Computer Science, Princeton University, Princeton, NJ, Technical report TR-912-11, 2011.
[46] James P.G. Sterbenz, Deep Medhi, Byrav Ramamurthy, Caterina Scoglio, David Hutchison, Bernhard Plattner, Tricha Anjali, Andrew Scott, Cort Buffington, Gregory E. Monaco, Don Gruenbacher, Rick McMullen, Justin P. Rohrer, John Sherrell, Pragatheeswaran Angu, Ramkumar Cherukuri, Haiyang Qian, Nidhi Tare, The great plains environment for network innovation (GpENI): a programmable testbed for future Internet architecture research, in: Testbeds and Research Infrastructures. Development of Networks and Communities 6th International ICST Conference, TridentCom 2010, Berlin, Germany, May 18–20, 2010, Revised Selected Papers, 2010, pp. 428–441.
[47] Deep Medhi, Byrav Ramamurthy, Caterina Scoglio, Justin P. Rohrer, Egemen K. Çetinkaya, Ramkumar Cherukuri, Suan Liu, Pragatheeswaran Angu, Andy Bavier, Cort Buffington, James P.G. Sterbenz, The GpENI Testbed: network infrastructure, implementation experience, and experimentation, Comput. Netw. 61 (2014) 51–74.
[48] John Hartman, Scott Baker, A Provisioning Service for Long-Term GENI Experiments, June, 2011. <http://groups.geni.net/geni/wiki/ProvisioningService>.
[49] Brian White, Jay Lepreau, Leigh Stoller, Robert Ricci, Shashi Guruprasad, Mac Newbold, Mike Hibler, Chad Barb, Abhijeet Joglekar, An integrated experimental environment for distributed systems and networks, in: SIGOPS Operating Systems Review, vol. 36, no. SI, December 2002, pp. 255–270.
[50] Srini Seetharaman, Masayoshi Kobayashi, Guru Parulkar, Guido Appenzeller, Joseph Little, Johan van Reijendam, Paul Weissmann, Nick McKeown, Maturing of OpenFlow and software defined networking through deployments, Comput. Netw. 61 (2014) 151–175.
[51] M. Ott, I. Seskar, R. Siraccusa, M. Singh, ORBIT testbed software architecture: supporting experiments as a service, in: First International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM 2005), pp. 136–145.
[52] Jolyon White, Guillaume Jourjon, Thierry Rakotoarivelo, Max Ott, Measurement architectures for network experiments with disconnected mobile nodes, in: Testbeds and Research Infrastructures. Development of Networks and Communities 6th International ICST Conference, TridentCom 2010, Berlin, Germany, May 18–20, 2010, Revised Selected Papers, 2010, pp. 315–330.
[53] Serge Fdida, Jerker Wilander, Timur Friedman, Anastasiu Gavras, Leandro Navarro, Mike Boniface, Sophia MacKeith, Susanna Avéssta, Martin Potts, FIRE Roadmap Report 1 – Part II, Future Internet Research and Experimentation (FIRE), 2011.
[54] Mauro Campanella, The FEDERICA project: creating cloud infrastructures, in: Cloud Computing: First International Conference, CloudComp 2009, Munich, Germany, October 19–21, 2009, Revised Selected Papers, 2010.
[55] Mauro Campanella, Fabio Farina, The FEDERICA Infrastructure and Experience, Comput. Netw. 61 (2014) 176–183.
[56] Andreas Köpsel, Hagen Woesner, OFELIA – Pan-European Test Facility for OpenFlow Experimentation, in: Witold Abramowicz, Ignacio M. Llorente, Mike Surridge, Andrea Zisman, Julien Vayssière (Eds.), Towards a Service-Based Internet, Lecture Notes in Computer Science, vol. 6994, Springer, Berlin Heidelberg, pp. 311–312.
[57] Marc Suñé, Leonardo Bergesio, Hagen Woesner, Tom Rothe, Andreas Köpsel, Didier Colle, Bart Puype, Dimitra Simeonidou, Reza Nejabati, Mayur Channegowda, Mario Kind, Thomas Dietz, Achim Autenrieth, Vasileios Kotronis, Elio Salvadori, Stefano Salsano, Marc Körner, Sachin Sharma, Design and implementation of the OFELIA FP7 facility: THE European OpenFlow testbed, Comput. Netw. 61 (2014) 132–150.
[58] BonFIRE Project, BonFIRE web site. <http://www.bonfire-project.eu>.
[59] Jordi Jofre, Celia Velayos, Giada Landi, Michał Giertych, Alastair C. Hume, Gareth Francis, Albert Vico Oton, Federation of the BonFIRE multi-cloud infrastructure with networking facilities, Comput. Netw. 61 (2014) 184–196.
[60] Dennis Schwerdel, Daniel Günther, Robert Henjes, Bernd Reuther, Paul Müller, German-lab experimental facility, in: Proceedings of the Third Future Internet Conference on Future Internet (FIS'10), Berlin, Heidelberg, pp. 1–10.
[61] Dennis Schwerdel, Bernd Reuther, Paul Müller, Thomas Zinner, Phuoc Tran-Gia, Future Internet research and experimentation: the G-lab approach, Comput. Netw. 61 (2014) 102–117.
[62] JGN-X Web Site. <http://www.jgn.nict.go.jp/english/index.html>.
[63] Akihiro Nakao, Architectures and testbeds enabled through advanced network virtualization: CoreLab and VNode projects, in: The 3rd EU Japan Symposium on Future Internet.
[64] Toshiyuki Miyachi, Junya Nakata, Razvan Beuran, Ken-ichi Chinen, Kenji Masui, Satoshi Uda, Yasuo Tan, Yoichi Shinoda, Realistic simulation of Internet in experiment, in: Systems Modeling and Simulation: Theory and Applications, Asia Simulation Conference 2006, 2007.
[65] Dongkyun Kim, Joobum Kim, Gicheol Wang, Jin-Hyung Park, Seung-Hae Kim, K-GENI testbed deployment and federated meta operations experiment over GENI and KREONET, Comput. Netw.61 (2014) 39–50.

Mark Berman is Vice President for Technology Development at Raytheon BBN Technologies and GENI Project Director. Mark works to bring technical innovations into practical use. Mark's research interests are in the area of complex distributed systems and their usability. As manager of BBN's Intelligent Computing business, he exercised management oversight over numerous research and technology transition efforts across a diverse set of disciplines and advocated for an open source approach to technology transition. Mark holds A.B. and S.M. degrees in computer science from Harvard.

Jeffrey S. Chase is a Professor of Computer Science at Duke University and a Visiting Scientist at the Renaissance Computing Institute (RENCI). He leads the Open Resource Control Architecture (ORCA) control framework project and has held various leadership positions in the GENI initiative.



Lawrence Landweber is the John Morgridge Professor Emeritus at the University of Wisconsin and an advisor to the GENI Project Office. Since 1980 he has been a contributor to the development of the global Internet. Major US projects in which he has participated have included CSNET, the Gigabit Testbed Project, the 100 Megabits to 100 Million Homes Project and GENI. He is a Fellow of the ACM, the recipient of the IEEE Award on International Communication and of the Internet Society's Jonathan B. Postel Service Award. He was recently inducted by the Internet Society into the "Internet Hall of Fame."

Akihiro Nakao received B.S. (1991) in Physics, M.E. (1994) in Information Engineering from the University of Tokyo. He was at IBM Yamato Laboratory, at Tokyo Research Laboratory, and at IBM Texas Austin from 1994 to 2005. He received M.S. (2001) and Ph.D. (2005) in Computer Science from Princeton University. He has been teaching as an Associate Professor in Applied Computer Science, at Interfaculty Initiative in Information Studies, Graduate School of Interdisciplinary Information Studies, the University of Tokyo since 2005. He has also been an expert visiting scholar and a project leader at National Institute of Information and Communications Technology (NICT) since 2007.

Max Ott is the Research Group Leader of the Networked Systems Theme at NICTA since January 2006. His group is active in various international testbed activities, such as FIRE in Europe and GENI in the US. Before joining NICTA he co-founded Semandex, which is a pioneer provider of Content-Based Networks, a new generation of Enterprise Information Integration and Knowledge Management systems. He also holds a Research Professor appointment at WINLAB, Rutgers University, where he is responsible for the software architecture of the ORBIT testbed. He holds a Ph.D. in electrical engineering from Tokyo University, Japan, and an M.S. from the Technical University, Vienna, Austria.

Dipankar Raychaudhuri is Professor-II, Electrical & Computer Engineering and Director, WINLAB (Wireless Information Network Lab.) at Rutgers University. As WINLAB's Director, he is responsible for an internationally recognized industry-university research center specializing in wireless technology. His research group at WINLAB has been working on design and implementation of next-generation wireless networks covering a number of emerging usage scenarios such as ad hoc mesh, vehicular, cognitive radio, 4G and mobile Internet. He is the principal investigator for several multi-institutional projects funded by the US National Science Foundation (NSF) including: "ORBIT" open-access next-generation wireless network testbed, "CogNet" cognitive radio network, and most recently, the "MobilityFirst" future Internet architecture (FIA) project. He also helped to initiate the ongoing GENI program for deployment of a global-scale experimental infrastructure for Internet research, and is currently involved with the the "Open GENI Base Station" project aimed at deploying programmable 4G wireless networks at several university campuses across the US. Dr. Raychaudhuri has previously held progressively responsible corporate R&D positions in the telecom/networking industry including: Chief Scientist, Iospan Wireless (2000–2001), Assistant General Manager & Dept Head-Systems Architecture, NEC USA C&C Research Laboratories (1993–1999) and Head, Broadband Communications Research, Sarnoff Corp (1990–1992). He obtained his B.Tech (Hons) from the Indian Institute of Technology, Kharagpur in 1976 and the M.S. and Ph.D degrees from SUNY, Stony Brook in 1978, 79. He is a Fellow of the IEEE.

Robert Ricci is a Research Assistant Professor in the University of Utah's School of Computing. He has been been building network testbeds for over a decade, beginning with the Emulab facility and continuing with the National Science Foundation's GENI and PRObE efforts. He has done research on a variety of topics regarding testbed design, construction, and use, including resource mapping, control system design, and emulation of realistic network conditions.

Ivan Seskar is Associate Director at WINLAB, Rutgers University responsible for experimental systems and prototyping projects. His current research interests are in the areas of wireless testbeds, mobile networking and software defined radios. Mr. Seskar's research group at WINLAB developed one of the first software defined radio (SDR) prototypes in the late-1990's, and more recently worked on a project which led to the development of WiNC2R, a high-performance network-centric cognitive radio platform. He is also the lead project engineer for ORBIT, with responsibility for development, integration and deployment of the radio grid emulator system which became operational in 2005. He is currently PI or co-PI for wireless GENI projects at Rutgers, including the Open WiMAX base station, OpenFlow campus network and the CRKit cognitive radio platform.