

Facility Usage Scenarios

GDD-06-41

GENI: Global Environment for Network Innovations

December 22, 2006

Status: Draft (Version 0.1)

Note to the reader: this document is a work in progress and continues to evolve rapidly. Certain aspects of the GENI architecture are not yet addressed at all, and for those aspects that are addressed here, a number of unresolved issues are identified in the text. Further, due to the active development and editing process, some portions of the document may be logically inconsistent with others.

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

Nick McKeown, Stanford

Dipankar Raychaudhuri, Rutgers

Mike Reiter, CMU

Jennifer Rexford, Princeton

Scott Shenker, Berkeley

Amin Vahdat, UCSD

John Wroclawski, USC/ISI

This work is supported in part by NSF grants CNS-0540815 and CNS-0631422 .

Revision History

Version	Changes log	Date
v.01	Original version posted	12/22/06

Table of Contents

Revision History	3
Introduction	5
Scenario 1: Live Distributed Service.....	6
Scenario 2: Internet Probing Experiment	7
Scenario 3: Controlled Router Experiment	7
Scenario 4: Live Virtual ISP.....	8
Scenario 5: Dynamic Circuit Provisioning.....	8
Scenario 6: Enhanced Wireless Service	8
References.....	9

Introduction

This document identifies a collection of usage scenarios that we expect the GENI facility to support. The list is not intended to be exhaustive, but rather, it describes a set of reference points to help researchers see how their experiments might map onto GENI.

The discussion does not argue the merits of each experiment, or even describe the research question being addressed, except as necessary to understand what requirements the example scenarios place on the facility. For additional information placing these examples in the larger scientific agenda, the reader is referred to the GENI Research Plan [GDD-06-28].

We define each experiment (slice) along three dimensions, independent of the functionality the experiment might implement:

- the set of components the slice spans, including how those components are connected;
- the interface (level of abstraction) the slice requires on those components; and
- the expectations the slice has about resource guarantees.

With respect to the second point—the interface by which the experiment accesses the underlying network substrate—we assume components expose some combination of the following three interfaces:

- **Virtual Server Interface:** Experiments access the network through the standard socket interface. This interface allows inter-component communication is via TCP or UDP connections, or IP tunnels. The server interface does not imply a fixed topology—all components participating in the experiment are reachable (addressable) to the extent they are reachable via today’s Internet.
- **Virtual Router Interface:** Experiments access the network through a virtualized link interface. This interface exposes transmission queues and link failures, and the virtual links can be configured to support CBR guarantees. The router interface implies a fixed topology—the set of links (peers) available at each component is statically defined when the slice is created, and is **not** under control of the experiment running in the slice. There are two sub-cases of this interface: (1) the virtual link is point-to-point between a pair of components; and (2) the virtual link is multi-point, corresponding to a VLAN or a wireless network. We sometimes refer to the wireless version of the second case as a “Virtual MAC Interface.”
- **Virtual Switch Interface:** Experiments access the network through a virtualized control interface that permits the dynamic creation, termination, and provisioning of end-to-end circuits. This slice interface does not imply a fixed topology—some amount of physical link capacity (e.g., an optical wavelength) is allocated to a given experiment when the containing slice is created, but the set of end-to-end circuits multiplexed onto that capacity is under the control of the experiment. There are two sub-cases of this interface: (1) the interface includes framing, implying that it is packet-oriented; and (2) the interface does not include framing, implying that the experiment has access to a raw optical wavelength.

Note that one should not read too much into the names assigned to each interface type. They were chosen to reflect the protocol layer in today's Internet that would use the interface. For example, a router that forwards packets between (possibly heterogeneous) links would run on top of the router interface. Said another way, the router interface allows researchers to experiment with a virtual network consisting of a set of virtual routers. The functionality implemented by such virtual routers might be similar to or completely different from today's IP router, the important point being that both need some virtual topology among a set of nodes. For another discussion of the different levels at which researchers might want to access the physical substrate, the reader is referred to [GDD-06-26].

Scenario 1: Live Distributed Service

This scenario corresponds to a deployment study of a wide-area service that serves a live client population. Example services include content distribution and data dissemination networks, routing and indirection overlays, peer-to-peer networks, large file transfer and data streaming services, and so on. For this case, a slice includes a sliver at every edge site, each of which:

- takes advantage of connectivity to the legacy Internet to carry real user traffic;
- runs its service-specific functionality on top of the server interface (i.e., uses today's TCP/IP stack); and
- is able to *adapt to available resources*, that is:
 - is able to run with only a fair share of the available resources on each component
 - may want a minimal slice reservation as long as it's not limited to that reservation
 - if forced to make an upper-bound reservation, will repeatedly probe the component for additional resources.

Depending on the functionality provided by the service, each slice might also be allocated a significant amount of memory and disk storage capacity at each component.

Optionally, the slice might also include a sliver at each backbone PoP, each of which

- initially runs the standard IPv4 forwarding service (i.e., the backbone components merely provide high-bandwidth connectivity among edge sites);
- may eventually augment the standard IPv4 forwarding service with experiment-specific optimizations, in which case this modified forwarding service would run on top of the router interface; and
- may or may not want minimal guarantees, similar to the "adapt to available resources" requirement from above.

Note that instead of running the standard IPv4 forwarding service in its own slice (which gives the slice the option of modifying the forwarding behavior in service-specific ways) the slice may simply "opt in" to a backbone service that provides IPv4 packet forwarding, without any need for the slice to run its own routing protocol. That is, the logic and data structures for packet

forwarding in the backbone may be shared across many other slices needing a similar packet-delivery service.

Optionally, the slice might also include a sliver at each available wireless component, each of which

- initially runs the standard IPv4 forwarding service (i.e., the wireless components merely provide high-bandwidth connectivity among edge sites);
- may eventually augment the standard IPv4 forwarding service with experiment-specific optimizations (but still likely to run on top of IP); and
- may or may not want minimal guarantees, similar to the “adapt to available resources” requirement from above.

Scenario 2: Internet Probing Experiment

This scenario corresponds to an Internet measurement experiment, involving probes to sites in today’s Internet. For this case, a slice includes a sliver at every edge site, each of which

- takes advantage of connectivity to legacy Internet to send probe messages;
- runs network probes on top of the server interface (i.e., uses today's TCP/IP stack); and
- has periodic resource demands (i.e., probe/quiet/repeat), running in one of two modes
 - mode 1: minimal resource requirements, so a fair share allocation is fine,
 - mode 2: time-sensitive measurements, so latency guarantees are required.

Scenario 3: Controlled Router Experiment

This scenario corresponds to a controlled study of how a new algorithm or protocol compares to the standard Internet. The new strategy might modify only the control plane (e.g., test a new routing protocol) or it might involve modifying both the control and data planes (e.g., test a new congestion control algorithm that requires routers to mark packets forwarded on the data plane). For this case, a slice includes a sliver at every backbone PoP, each of which

- includes virtual links to some set of neighbors, forming some meaningful topology (no connectivity to the legacy Internet is required);
- runs the new algorithm or protocol on top of the router interface (requires the ability to inject faults), but may or may not include an “IP data plane” library; and
- requires fixed (upper and lower) reservations on CPU and link resources, although such guarantees are required only for fixed periods of time (i.e., while the experiment is running).

The slice likely also include slivers at some set of edge sites to serve as traffic sources and sinks. These slivers require some minimal resource reservation.

Optionally, the experiment might introduce modifications to the data plane (i.e., to move beyond IPv4 packet forwarding), but still run on top of the router interface.

Optionally, the experiment might include multiple backbone-wide slices, each of which emulates a distinct autonomous domain, perhaps each using a different topology. A sliver at select components—representing an Internet eXchange Point—might run a standard or experimental inter-domain routing protocol, and forward packets between virtual networks.

Scenario 4: Live Virtual ISP

This scenario corresponds to a deployment study of a Virtual ISP that offers an enhanced packet delivery service to a live client population. For example, the Virtual ISP might employ the new routing or congestion control algorithm validated in Scenario 3. For this case, a slice is similar to the previous configuration, except:

- slivers at edge sites and backbone nodes forward traffic between the slice and the legacy Internet (i.e., these edge slivers serve as ingress/egress nodes for the routing service), and maintain routing-protocol adjacencies with existing service providers (i.e., to learn how to reach external destinations and control how hosts participating in the experiment are reached);
- reservations for backbone slivers are continuous rather than periodic; and
- reservations for edge slivers are probably similar to those in Scenario 1.

Optionally, the slice might include a sliver running on each available wireless component, thereby bringing wireless access networks into the study.

Scenario 5: Dynamic Circuit Provisioning

This scenario corresponds to an experiment that creates a topology in which boundary routers are interconnected by a topology of circuits, where as the traffic matrix between the boundary routers changes, the experiment rapidly sets up (and removes) circuits between the routers. In this case, a slice includes a sliver at every backbone PoP, each of which

- includes dedicated bandwidth to physical neighbors;
- runs circuit management software on top of the switch interface; and
- requires fixed (upper and lower) reservations on CPU for fixed periods of time (i.e., while the experiment is running).

The circuits managed by the slice could be either electrical TDM circuits or optical WDM circuits, depending on whether the experimenter elects to include standard framing protocols.

Scenario 6: Enhanced Wireless Service

This scenario corresponds to an experiment that evaluates new functionality across one or more wireless networks. Examples include a service that routes packets to mobile devices based on geographic coordinates specified in a (non-IP) header, and a service that forwards large files on a hop-by-hop basis, accounting for the possibility that these files have to be stored for a period of time at each node so as to accommodate disrupted network connectivity. For this scenario, a

slice includes a sliver at every component in one or more wireless access networks (including both wireless nodes and nodes at the edge of the subnet), where

- a set of mobile client devices are configured with the application and protocol software required to opt into the experiment;
- a virtual topology is established between the wireless nodes and edge node associated with the subnet;
- the wireless nodes communicate with the edge nodes over the router interface;
- the wireless nodes communicate with each other and with mobile client devices over the router (virtual MAC) interface; and
- each sliver assumes CPU, link, memory, and storage guarantees so as to provide repeatable results.

Note that any given experiment will likely configure the slice differently. For example, an experiment involving geographic routing will assume a set of GPS-enabled mobile devices that clients use to opt into the experiment. Similarly, the disruption-tolerant forwarding service may require substantial memory and storage resources at each node (e.g., 1GB and 10-100GB, respectively).

Optionally, such experiments might use the service interface (running on top of TCP/IP) to access available global location services.

Optionally, such experiments might include slivers on the backbone, interconnecting multiple distinct wireless networks. These backbone components might be connected by virtual links (i.e., use the router interface), thereby emulating dedicated point-to-point circuits between wireless subnets. Alternatively, they might compose with one of the Virtual ISPs developed under scenario 4.

References

[GDD-06-28] David Clark and Scott Shenker (Chair), Aaron Falk (Ed). "GENI Research Plan," *GENI Design Document 06-28*, Research Coordination Working Group, September 2006.

[GDD-06-26] Dan Blumenthal and Nick McKeown, "Backbone Node: Requirements and Architecture," *GENI Design Document 06-26*, Backbone Working Group, November 2006.