Statement of Work

University of Utah and Princeton University Proposal: "Exploring Federation of Testbeds with Diverse Models"

1 Introduction

Our proposed work will focus on prototyping federation across testbeds whose models and resources differ in many ways, and demonstrating the utility of the result. Our work will be in the context of the existing Emulab and PlanetLab testbeds, but with an eye towards future experimental and deployment networks such as GENI.

Many of the main differences between the Emulab and PlanetLab environments are due to their most common usage models. A typical Emulab *experiment* is used for development, debugging, and emulation of networked and distributed applications, and will last hours or days, and a user relinquishes control of resources while he or she is not actively working. Some Emulab "experiments" are in fact system or user-provided services, especially those that incorporate PlanetLab nodes in their topologies, and run indefinitely. A PlanetLab *slice* may live for months or years, and many provide "always on" services to the PlanetLab user community or to Internet users at large. Most Emulab experiments primarily contact other nodes inside of Emulab; in contrast, many PlanetLab slices provide services to or measure areas of the Internet outside PlanetLab.

The life cycle differences between Emulab experiments and PlanetLab slices lead to other differences, such as the testbeds' models for granting users access to resources. On PlanetLab, a physical node can be shared by an arbitrary number of users simultaneously. Currently, all users get a roughly equal share of resources such as CPU time, though PlanetLab is working on guarantees and other types of reservations. In Emulab, on the other hand, most resources are allocated to a single user at a time. Thus, when Emulab is full, new users cannot enter the system. PlanetLab does not have a "full" capacity, but each user's share of the resources degrades as more users enter the system. Both approaches have their advantages, and each results in a different type of unavailability under load.

Because of its general-purpose nature, GENI will need to be able to support components that follow both models or some combination of them. Thus, our work on federating Emulab and PlanetLab will result in knowledge valuable for building GENI. Furthermore, the resulting artifact may prove useful as a "departure point" for an actual GENI federation implementation.

Beyond informing the design of GENI federation, prototyping federation in the Emulab context provides an additional benefit. Utah's Emulab testbed site is an instance of a "purpose-built" facility, and integrating such facilities is a goal of GENI federation. By federating Utah's Emulab testbed and PlanetLab, through the design and implementation of software on both sides, we will enable the ability to first test a service in Emulab and then easily migrate that service to PlanetLab. In the GENI context, one could imagine a GENI facility that will manage the life cycle of a developing service, moving it from development on a simulation facility, to testing on an emulation facility such as Emulab, to deployment in the "real world" via the GENI core.

There are two themes to the proposed work. The first is about reconciling how users (and groups of users) are tracked and acquire access to resources across the federated testbeds. The second is about bridging the gap between Emulab-style and PlanetLab-style activities, and new modes of use that combine the two.

2 Bridging Principals and Authentication

Emulab today has a portal to PlanetLab, allowing Emulab users to allocate slices on PlanetLab. This gives users the ability to integrate Emulab resources (such as cluster nodes, emulated links, and wireless devices) with PlanetLab resources (a large number of vantage points in the production Internet). Combining these resources allows for much richer experimentation and deployment than any of them do by themselves.

The current portal is inadequate for true federation, however, for two reasons. First, the portal is one-way: Emulab users can acquire PlanetLab resources, but not vice versa. We discuss our approach to this issue in Section 3. Second, users (principals) are not well-tracked across the portal. When a slice is created on behalf of an Emulab user, it is done using Utah's site credentials. This has the advantage that it provides a "single sign-on" capability for users going through the Emulab interface. It has a fundamental disadvantage, however, in that this model differs from the PlanetLab authentication model. PlanetLab requires strong auditing capabilities so that traffic that leaves PlanetLab can be traced back to the principal responsible for that traffic. Because experiments on PlanetLab occasionally set off intrusion detection systems and their services have been abused in the past for illegal purposes, accountability is vital.

We will investigate several different mechanisms for better handling principals and accountability between Emulab and PlanetLab, and implement at least one and probably two of them. These are the ones we are currently considering but further study during this project, and the concurrent evolution of the GENI security architecture, may cause changes.

The first, a highly efficient, convenient, and user-friendly approach, is to merge (for authentication and authorization purposes, not physically merge) the principal databases of the two testbeds. Today the two databases keep similar data and are of similar scale. Emulab will need to change to a an email address as a unique ID. Clearly this approach will be highly effective but does not necessarily scale to large federated environments. It might scale well enough the demands of GENI in its initial years, however.

The second, a short-term solution, would be to have Emulab request tickets for the real PIs from the PlanetLab Core (PLC). These tickets are used to create properly authenticated slices. A significant downside is that if not done cleverly, and with a delegation mechansism for the PIs, it will remove the single sign-on property of the portal. PIs (or their automated proxy) will need to pre-create or on-demand create slices for any of the users in their Emulab project. They will have to create tickets to allow access to those slices, and communicate those tickets to Emulab to allow it to manage the slice. The process of building this "manual" bridge between the Emulab and PlanetLab authentication mechanisms will inform the design of our third approach, described next.

A longer-term approach to handling principals would be to give Emulab the role of a "slice authority" (SA) for PlanetLab. In this role, Emulab will be responsible for its own users, creating slices for them directly on PlanetLab nodes. This will allow Emulab to implement its own policies and usage model, but makes it responsible for the actions of its users. Emulab may be responsible for auditing slices it creates and ensuring that they do not violate its own policies or the PlanetLab Acceptable Use Policy, or, better, it may be able to delegate much of that to PlanetLab (which would correspond to a component manager in the GMC). The Emulab SA will also be able to authenticate PlanetLab users who want to allocate resources within Emulab; thus, it will be an automated and bidirectional bridge between the two domains.

The fourth alternative is to use a version of the GENI security architecture when it is sufficiently well defined and implemented. That has the advantage of more power, more flexibility, more security, and more longevity.

3 Bridging the Resource Models

Emulab contains a vast array of resources that could be valuable to PlanetLab users, including wired cluster PCs that can run essentially arbitrary operating systems including Windows, Linux, and FreeBSD; physically distributed 802.11 wireless PCs; network processors; sensor boards; and software radio platforms. Just as PlanetLab resources will be useful to Emulab users, these Emulab resources will be useful to PlanetLab users. Of particular value, in either direction, is the ability of all users to leverage Emulab's emulation capabilities for prototyping services.

The primary challenge in federating Emulab and PlanetLab is reconciling the resource usage models. Given PlanetLab's model of resources (slices), it is relatively straightforward to allocate PlanetLab resources to Emulab users: the primary challenge is reducing latency for short-term experiments. The opposite direction is more challenging. Since Emulab allocates resources to a single user at a time, it would be impractical to dedicate significant Emulab resources to PlanetLab full-time. Thus, one of our goals is to explore how to make Emulab resources available to PlanetLab on a part-time basis, either by introducing the notion of transient resources to PlanetLab or by hiding the transient nature of Emulab resources behind a virtualization layer provided by an alternative "private PlanetLab" that interoperates with the public PlanetLab.

One way to present Emulab resources would be as a set of traditional PlanetLab nodes that are hosted at Utah. The unique aspect is that the nodes would be virtual and would have variable resources. At one extreme, it might appear that there is a single "super node" that, at various times, could host anywhere from zero to thousands of virtual machines ("slivers") depending on how many physical machines are available. A virtual machine presented by the Emulab interface could be an actual physical machine or a conventional virtual running on a virtual machine monitor (e.g., Xen or VMware), depending on the resource requirements of the slice.

To implement this federation model, a number of issues will need to be addressed and a number of current and proposed PlanetLab interfaces refined. Primary in the former is how to deal with critical, long-running infrastructure services. For the latter, we would need to implement an alternative PLC (in its management and slice authority roles) and implement multiple VM types.

The overarching theme is to use Emulab for what it does well: resource allocation and mapping, network emulation, and large-scale automated experiment setup, with an eye toward creating "entities" with external network connections that appear to PlanetLab and its users as "nodes."

4 Deliverables

This proposal assumes 1.5 staff FTE at Utah and at least .25 FTE at Princeton. Both parties will work together to understand (especially) and extend the code bases.

For the proposed work, we will integrate Emulab and PlanetLab as described in the previous sections. Concurrently, we will develop a federation interface for the two testbeds. In doing this, we will not only produce an artifact useful for both Emulab and PlanetLab users, but also gain a deeper understanding of integrating services and code from multiple sources into a coherent user experience. These lessons will help inform and guide the definition of the GENI architecture. More concretely, the results of this work will be twofold.

First, the Emulab portal to PlanetLab will be improved, giving users a uniform and convenient interface to a diverse set of resources while respecting the autonomy of both testbeds. Specific tasks include studying the options for authentication and authorization, implementing and deploying probably at least two of them (short and long-term solutions), with the ultimate goal of evolving the prototype to a GENI provisioning service (PS). The initial work (principal merging) should be complete within the first two months. The SA and other prototypes will follow.

Second, PlanetLab users will be able to integrate Emulab resources into their slices via PlanetLab interfaces. The initial work is to prototype a proxy PLC to run on Emulab and federate with the master PLC. Specific tasks include: complete the design of the PLC federation API, refine the virtual machine resource specification, implement the API on the PlanetLab side, design and implement virtual node manager and auditing services and the proxy PLC at Emulab, define an AUP for Emulab resources, and make the resulting federation available to users. We expect to have a demonstrable version of PLC federation, allocating physical Emulab PCs, within the first nine months.

At the completion of the work, we will demonstrate the utility of PlanetLab resources embedded in an Emulab experiment, and Emulab resources embedded in a PlanetLab slice.

Completion of this federation will also give Emulab capabilities similar to a GENI Component Manager (CM), as outlined in the GMC draft specification. A CM aggregates a set of components into a single autonomous unit. In the current GENI vision, many such CMs will cooperate, along with higher-level services, to produce GENI itself. A CM is responsible for some set of components, instantiating slices on and managing those components. By acting as a Management Authority, a CM makes the components it manages available to the rest of GENI.

Though the exact GMC interface is still being defined, we feel that a federated Emulab will provide the basic functionality required from a CM and will test some of the interfaces defined by the GMC. Thus, the proposed work will provide early insight into what will be required of the GMC interface to enable the inclusion of Emulab-style facilities in GENI.