A Network Management Information Plane for GENI: the architectural challenges

DRAFT

Karen Sollins, MIT CSAIL

August 18, 2010

1. Introduction

This paper outlines key architectural challenges in the provision of network management in the context of GENI. In order to do this, we will focus on three elements of the question: what is meant by network management, the architectural challenges presented by GENI itself, and the challenges of supporting information management and distribution as a substrate for network management.

The most widely held model of network management is that it is the set of tasks required to solve the problems of unexpected or incorrect behavior of the network. First, let us consider the breadth of unexpected or incorrect behaviors. These may not be limited only to the network itself and the resources that comprise it, but may validly be extended to the "clients" of those resources, whether distributed storage services such as peer-to-peer systems, email, or online social networking systems, and so forth. Thus, one might consider the behaviors of the distributed or networked "clients" to be within scope of management of the network; network management may include the requirements of the network itself, but also of those "end node" or "application" clients. Second, only addressing issues of unexpected or incorrect behavior is itself rather limiting. One might consider that configuration of networks, providing feedback to those "clients" on the behaviors of the network, prediction of behaviors of the network, tuning of the network to improve performance, or analysis for provisioning are examples of additional possible network management responsibilities. One of the reasons for expanding the definition of network management is that the resources involved are the same set of resources in each of these cases, and hence any information about how they are, or should be, behaving is also the same. It is quite likely critically important that these functions operate on the same information base to provide consistency and coordination.

In this context our next set of architectural challenges derive from GENI itself. Here we see two key architectural forces, layering and federation. We will consider these separately, although clearly within GENI they exist in a symbiotic relationship.

Let us consider layering first. Simply put, GENI defines a four layer architecture, with the Ops and Management Plane at the bottom and the Experiment Plane at the top. Above the Ops and Management Plane sits the Slice or Control Framework Plane, and between that and the Experiment Plane sits the Measurement Plane. Each layer uses the resources provided by the layer below it and provides its own resources to the layer above. Some of the resources, such as transport capabilities must be passed through a layer and up to its supported layer, while others may not. For the moment it is interesting to focus on those capabilities that are passed through the layers. We note that although a layer may "pass through" a resource such as packet transport, it is likely to reserve some of that resource for itself, providing to its upper layer only some of what was provided to it, for its own operation. Thus, for example the Slice Plane will provide transport among the slivers that comprise a slice, but it will also use some of the transport capacity provided to it to create and manage slices. This insight will play a role in our later analysis, in particular with respect to management.

The federation model proposed in Spiral 2, especially by the TIED project, reflects the realization that there will not be a centralized control model or mechanism for GENI. If it is to support not only clusters and other aggregates brought to the table, but peer networks, especially from other parts of the world, there will be no single central authority. It must operate effectively in a cooperative and distributed manner. This carries with it not only the need to clarify how collaborative decisions will be made, but also the incentives to encourage such collaboration.

With our expanded approach to network management and the double challenges of GENI's layered model and federated organization, we will then examine what we believe is the key substrate that will be required for effective network management, both across that federated environment and through the layers, an information substrate.

2. What do we mean by "management"

Let us now return in more detail to an examination of the term "management". We will begin with an examination of the most basic management function, the management of failures and misbehaviors, and then look beyond that.

The network manager spends a significant amount of time handling alerts and error reports, analyzing the causes of those signals in order to determine whether something in the management domain is not operating correctly, often doing triage as needed, and then figuring out what to do about those situations that require attention. This may take two forms, mitigation and repair. This set of problems is itself quite challenging. Problems that may be faced include:

- Processing incoming signals, both to eliminate duplicates and to rank order them in terms of priority and possibility effort to address
- Determination and acquisition of the information required to work backwards from the signal. It is worth noting that some information may be directly available either through a variety of techniques from ongoing to a priori collection, but some may only be available through inference on other information that can be acquired
- Working back from an alert or signal to the real source of the problem
- Acquisition of expected or correct behaviors
- Comparison and evaluation of expected and observed behaviors
- Evaluation of options for addressing the situation, considering a variety of costs (time, resources, money, etc.) in light of the criticality of the problem.
- Execution of a plan.

If we step back from this, we realize that, first this in itself is a non-trivial and often non-deterministic set of tasks. In addition, it is in practice only part of what network managers do. Two critically important other responsibilities are those of configuration and provisioning. Configuration reflects an often-complex set of choices about how to choose among the many options that may be available in organization and allocation of resources, frequently incorporating policy challenges as well. It is worth noting that in our diagnosis outline above, often the final plan that is executed will be a reconfiguration. Configuration and reconfiguration may require not only allocation or assignments, but timing, e.g. certain resources may need not only to be configured but also operational prior to others being configured and made operational. This all in itself is a complicated set of tasks, based on significant amounts of information and knowledge.

Again stepping back further, we note that management of network resources is done in the service of a set of client services, which themselves may be running on a distributed set of facilities, utilizing the network for communication. Once we bring these client services into the picture, we find a further set of requests on the network management facilities. Because most of the networking that is provided is not reserved and guaranteed, but rather statistical in some way and often best effort, these client services may make their own predictions of expected behaviors of the network. Increasingly we find third party services that try to aggregate these prediction services, both to provide increased service and to avoid duplication. Examples of this range from RON to iPlane. It is worth noting here that these predictions are often based on the same kinds of information that is required for both diagnosis and (re)configuration.

As we will discuss further below, in fact network management is increasingly made more complex because of the problems faced by usage that crosses management control boundaries. This is true not only in the network at large, but in GENI as well.

3. What is important in the GENI architecture

We now turn our attention to GENI itself and for reference include our understanding of the GENI architecture in Figure 1. We will examine it from two dimensions. First, we will work our way down through the layers. We will then address its federated nature and where that is both exposed and hidden, as well as considering the role of the GMOC in this context.

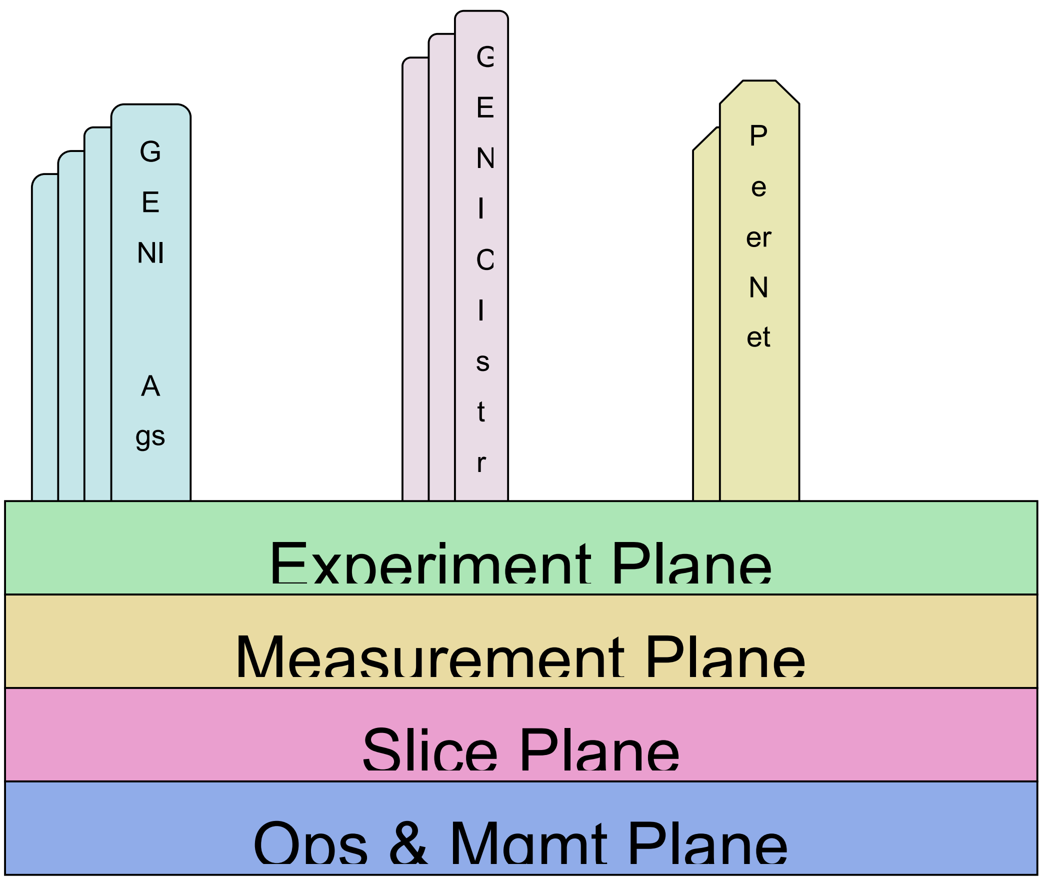

Figure 1: The GENI Architecture and Federation

The layered GENI architecture

The top level objective of the GENI effort is to provide network researchers with apparently isolated environments in which to run independent and potentially parallel experiments. From the perspectives of these users of GENI, they are running their experiments in isolated networks, that are not affected by either the other GENI users or the GENI infrastructure itself. It is worth noting that this kind of isolation does not require that they be isolated from the broader Internet, because in many cases that may be an important aspect of their experimentation. Instead, to the extent they are using GENI provided resources, from their perspective they have complete access to those resources, apparently without effects from other experimenters. Of course, this may be mitigated by resources that are not guaranteed to them. An example may be bandwidth. If an experiment is provided only statistical bandwidth between two slivers, the path may be influenced by other experiments with which it shares the links. If the experiment is provided with two slivers that communicate via the Internet, then the bandwidth may be influenced by outside activity as well. The observant reader will notice that if a slice is being supported by slivers provided by peering networks (e.g. GENI and Onelab), the "links" between nodes in the experimental network will most likely be over the public Internet and therefore built on statistically shared resources. That all said, the abstraction that is provided to the experimenter here is a unified network as defined by the experimenter for purposes of experimentation.

In the GENI layered model, the measurement plane sits below the experiment. This suggests that the measurement plane actually provides two things. One is what one would expect, the framework for measuring and monitoring experiments, without having an impact on the experiments themselves. This suggests that the resources used to do and manage measurements must be allocated outside the set of resources allocated to the experiments themselves. The second thing that the measurement plane must be doing is passing through the resources from the supporting layer (what we have labeled the slice plane) that will be composed into the experimental network itself. For example, if a network node is to provide storage for the experiment, and the measurement plane also requires storage, access and usage of storage available to the measurement plane must be split, so that the experimental node is provided with a virtual file system without conflicts from the measurement activities.

There is a different approach that could have been taken here, but from the GENI documents appears not to have been. It could have been the case that the measurement and control or slice planes both sit in parallel on top of the operations and management plane and the experiment is configured to sit on top of both of them. This would then suggest that the resources derived from the measurement plane would be provided separately from those derived from slice and control activities, specifially the resources used to create and manage the experimental networks. We will return to this discussion in discussing the challenges for network management in this context.

The lowest plane architected by GENI is the operations and management plane. It is the responsibility of this plane to provide access to the supporting resources, both in the form of configuration (organization) of the resources and management support when this substrate is not operating as expected or intended. In other words, it is in this plane that the focus has been placed on adequate or expected performance, and what to do when those goals are not being met. This is a fairly standard approach, because it is well understood that if the physical and lower layers in a network are not operating as intended, then any abstractions sitting above them also may not be meeting expectations. As we will note below, unfortunately, this is not the whole problem. The measurement plane gives us a hint of that. The architecture has a home for figuring out whether the low level resources are working adequately and a home for figuring out how an experiment is performing. We notice here, that there is a gap and furthermore, no real opportunity to drill down from behaviors in the experiment through the planes to understand the sources of targeted (mis)behaviors.

Federation

If we start with the Oxford English Dictionary, we find two primary definitions of the work "federation", with different implications. The first reflects the formation of a "unity" from a set of otherwise independent organizations, in which each retains control of its internal or local affairs. The second includes not only the existence of the league of federated organizations, but also the concept of a parent or central control, especially with respect to external relationships between the federation and other entities with which it interacts. We can see these two concepts reflected in the Spiral 2 vision of federation in GENI, as is being described in the recent versions of the **Slice-based Framework Architecture** document.

If we return to Figure 1, we have depicted sets of GENI clusters, GENI aggregates, and Peer nets. Each example of each of these can be considered a member of the federation, with different degrees of coordination and collaboration among them. Thus, for example, a set of Emulab based facilities might become a federation as has been done in some of the TIED demonstrations. That federation may itself federate with the US-based PlanetLab cluster within the US-based GENI federation. But internationally, US GENI may federate with OneLab or K-Lab (Korea) or J-Lab

(Japan). In the federated model, each federant retains control over the allocation and usage of the resources it contributes.[1] In contrast, from the outside the experimental facility provides an appearance and control structure that allows the experimenter or other user of the facility to interact with the facility as a whole, and not have to negotiate individually with each federant.

Perhaps even more importantly, from the perspective of this work, management also needs to be federated. We see the beginnings of this, but only in its most basic form in the GMOC. The GMOC is intended to collect statistics and information about operational activity across the whole federation and expose that to the user and management community. That said, it will be completely dependent on the federants providing the information it intends to collate and provide. Because of the federated nature of the facilities, the GMOC can only request information, and the participants can determine the extent to which they will comply with those requests.

To conclude, we propose that there are two orthogonal architectural principles of GENI that provide the architectural context in which management must reside. The first is the layering or GENI plane architecture, which supports increasingly rich layers of abstraction, but also suggest that for those resources that are "passed through" a layer, some may be allocated to the use of the layer itself and some to the layer above. The second principle is that of federation, which is reflected in the combination of local control over local resources with a common, unified front to external users.

4. Challenges in supporting information for management

With our broad definition of management and the architectural principles of GENI described in the previous two sections, we can now consider the question of where management fits into the architecture and the challenges for it. We will examine the need for management in the two orthogonal dimensions of the GENI architecture and then look for the points of commonality and those of necessary diversity.

Let us begin with the current layered "plane" structure. We argue here that each plane itself needs to be managed, to configure it or reconfigure it, analyze it for correct or adequate behavior and address situations where that is not true, and possibly to predict behaviors, in order to provide

[1] The Emergency Stop effort apparently ran into exactly this issue. As was discussed in the last GEC, in July, 2010, when it is determined that an emergency stop should be executed, apparently there is no recourse if a federant chooses not to abide by the emergency stop. It can make its own internal decisions, with little or no recourse.

improved resource usage by the layers above. We choose here to consider this bottom up. The Ops and Management Plane will provide, for example, the paths between facilities components, the services for discovering resources available in remote facilities, and so forth. Hence, in order for the Slice Plane to operate effectively, it must be able to depend on adequate and correct behavior from the Ops and Management Layer. If the supporting services, whether routing among resources in different aggregates or providing correct DNS lookup for researcher authentication services, or a large number of other capabilities, the Ops and Management Plane itself must be managed. As mentioned in Section 2, the kinds of management activities may vary widely, but they will be based on observation of behaviors of the resources and some model (information) about the definitions of behaviors. Configuration and reconfiguration may be more dependent on the latter, diagnosis of failures dependent on both, and prediction perhaps mostly on the former.

When we consider the Slice or Control Framework Plane, again, without management this plane will be operating in free fall. The resources to some extent will be different, including slices and the slivers assigned to them and the access control over them, etc., but if, for example, the Slice Framework creates slices, it will be important that the system be able to analyze whether the experimenter actually received the slices as designated, or the slivers were assigned correctly, and so forth.

When an experimenter runs an experiment, not only is it important that the measurement plane provide measurements of what occurred within the context of that experiment, but that the experimenter be able to be confident that the measurement plane itself was configured and operating as expected. This may, for example, mean that the measurement plane must be provided with both enough storage and enough transmission capacity to store the measurements where and when expected. If measurements are stored on a remote storage server, it may be critical to know that that server has adequate capacity and that the network path and disk access both have enough bandwidth to store the information. Again, information is at the core of this ability to have confidence in the behavior of the plane in question. As discussed, the measurement plane itself is likely to provide significant support for management of the experiment plane, although it may be in the service not only of collecting experimental results, but, as is often important for an experiment, may be important in collecting information about the context of the experiment. This contextual information that may also be critical to management at this plane itself.

To complement this layered model of management, there will also need to be a federated approach to management. As discussed in Section 3, one of the key features of a federation is the retention of local control over local resources. Control is reflective of management. Thus, as the SFA proposes, an individual aggregate or peer network will retain allocation and access control over the resources that it makes available. This needs to be reflected in local management as well, although as with federated control, cooperation and coordination will also be critical. Thus, a management architecture must support not only local control, but local choice to collaborate and cooperate not only in the provision of resources but in the management of the collection or federation of them. It is well understood that just because a each element of set of resources is operating within specification, one cannot conclude necessarily that the composition of the resources is operating as intended or desired. Thus, for the federation to operate successfully, not only will some of the management responsibility remain under local control, but it will also be necessary to support an approach to cooperation and sharing of information, knowledge and expertise in management.[2]

We argue here that an information substrate will be required both within each plane of the GENI architecture and under the local control of each federant, but more than that, coordination both across the planes and among federants will be key to the success of that management framework. As with the issue of sharing some amount of information across federants, it will also be critical for effective management to make management information, and especially knowledge inferred from the underlying data accessible among the planes. We therefore suggest that at the core of an effective approach to management will be an information plane, supported orthogonally to the previously defined architectural planes and across the federation, with both a means of identifying the information with respect to its nature, type or ontology, and with access control adequate to the requirements of the federants. This combination of abstraction control and access control will be reflective of the two dimensions of the GENI architecture. Keys to the design of such a capability will be availability to each element of each plane, discovery of the availability of the

---

[2] The canonical example I use of this problem is to imagine a path in a network composed of a set of nodes and links, in which each link operates within bounds by dropping no more than 5% of its traffic. Being able to confirm that each link under independent management is operating correctly within bounds does not allow one to conclude that the composed path is even operational, not to mention providing adequate in performance.

information, adaptability to changing resource available, efficiency in usage of resources, scalability, provenance of the information, and support for reasoning.

It is important to realize that this is separate from the reasoning capabilities itself. We suggest here that the reasoning and analysis is an aspect of management that must be extensible and hence unconstrained, as much as possible. As new monitoring and management approaches develop, it should be straight forward to include them in the management framework. Hence we focus on the information plane that will support but not limit such extensibility.

This paper is the first in a series of two. The second paper, if funded, will examine a proposed design for such a network management information plane. In addition to meeting the challenges identified above, it will be important to address questions of the sources of information including not only monitoring and measurement, but also transformations to achieving either efficiency of storage and access, but also possibly some controlled degree of information hiding, the placement (and caching of information) for policy support, efficiency, and availability, discovery and access to information, and a framework for evaluating these choices simultaneously, as will be needed to make coherent and consistent choices about availability and access.