

Trinity: A Framework for Managing Wide-Area Virtual Networks

Yogesh Mundada[†], Murtaza Motiwala[†], Vytautas Valancius[†],
Andy Bavier^{*}, Nick Feamster[†], Larry Peterson^{*}, and Jennifer Rexford^{*}
^{*} Princeton University [†] Georgia Tech

1. Introduction

We describe *Trinity*, a wide-area virtual network management framework which acts as an *infrastructure provider* to manage the physical resources and *service provider* to build and manage virtual networks on the physical resources. As shown in Figure 1, Trinity treats VINI [1] as the infrastructure provider and Emulab [2] as the service provider.

2. Summary

Currently, ISPs function as both infrastructure providers (who manage routers, data centers) as well as service providers (who deploy network protocols) simultaneously. Cabo [3] (“Concurrent Architectures are Better than One”) proposes decoupling these two roles by using virtualization techniques to run services over equipment owned by different infrastructure providers.

VINI is a virtual network infrastructure which allows researchers to create their own arbitrary topologies and test the protocols under realistic but controlled environment. The software which is used to implement current version of VINI, Trellis, is in large parts derived from PlanetLab. Main difference between VINI and PlanetLab is that VINI allows experiments to create their own network topology which is overlaid on top of physical topology.

Emulab allows users to create their own network topologies which are then embedded on the physical topology. However, the main Emulab doesn’t use virtualization but allocates entire physical nodes to experiments. The Emulab management framework provides users an interface to create their experiments in familiar NS2 syntax and sets up the topology, allocates resources, and helps users manage experiments.

Trinity leverages the strengths of Emulab for building and managing experiments and VINI for instantiating virtual networks on a shared infrastructure. The integrated framework serves as an instance of the functionality split between infrastructure and service providers proposed by Cabo.

3. Challenges

Implementing Trinity entails several challenges:

- *Resource specification:* The global view of resources advertised by VINI and the virtual em-

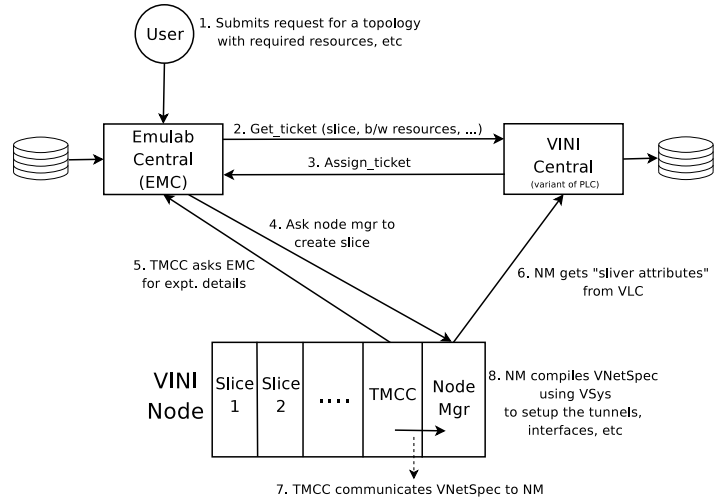


Figure 1: Initial prototype. Sequence of steps from user submitting a request for a topology to the actual creation of slices, tunnels, virtual interfaces, etc

bedding specified by Emulab to VINI should be in commonly agreed language.

- *Embedding of virtual topology over physical topology:* Trinity must determine how virtual nodes and links map to the underlying physical resources. This mapping is NP-hard [4].
- *Topology instantiation:* Once the resource requests are granted, Trinity must instantiate slices and tunnels on the physical nodes.
- *Resource management:* Trinity must manage the allocation, de-allocation, revocation and sub-leasing of physical resources.

REFERENCES

- [1] A. Bavier, N. Feamster, M. Huang, L. Peterson, and J. Rexford. In VINI Veritas: Realistic and Controlled Network Experimentation. In *Proc. ACM SIGCOMM*, Pisa, Italy, Aug. 2006.
- [2] Emulab. <http://www.emulab.net/>, 2006.
- [3] N. Feamster, L. Gao, and J. Rexford. How to lease the Internet in your spare time. Technical report, Georgia Institute of Technology, Aug. 2006. <http://www.cc.gatech.edu/~feamster/papers/cabo-tr.pdf>.
- [4] R. Ricci, C. Alfeld, and J. Lepreau. A solver for the network testbed mapping problem. *ACM Computer Communications Review*, 33(2), Apr. 2003.