At-scale experimentation with resource virtualization in a metro optical testbed.

ILIA BALDINE
Renaissance Computing Institute
JEFF CHASE
Duke University
GEORGE ROUSKAS
North Carolina State University
RUDRA DUTTA
North Carolina State University

In this paper we describe several efforts to explore integrated approaches to resource virtualization in a metro-scale optical testbed dubbed BEN (Breakable Experimental Network) located in Research Triangle Park, NC. The first one is an extension of work done at Duke on the ORCA (Open Resource Control Architecture) framework. The effort between RENCI and Duke seeks to expand the scope ORCA to enable it to function as management plane for the network as well as edge resources, acting as a GENI management plane. The second describes an attempt to redefine the architecture of the protocol stack with important implications to network virtualization. This is done using the SILO (Services Integration controL and Optimization) framework being jointly developed at RENCI and NCSU.

Categories and Subject Descriptors: ... [**...**]: ...

General Terms: Virtualization, Resource Provisioning, Optical Networks

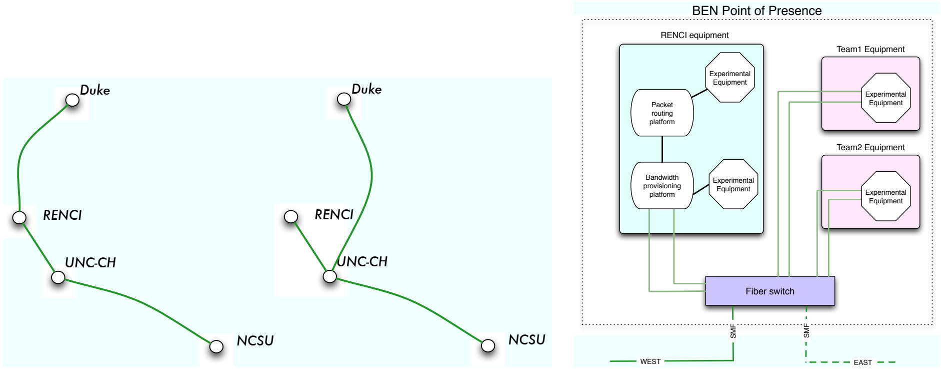
Additional Key Words and Phrases: breakable experimental network, silo, orca

1. INTRODUCTION

In 2008 Triangle Universities (UNC-CH, Duke and NCSU) in collaboration with RENCI (Renaissance Computing Institute) and MCNC began the rollout of a metro-scale optical testbed dubbed BEN Breakable Experimental Network. BEN consists of dark fiber (Figure 1(a)) provided by MCNC, which interconnects sites at the three Universities, RENCI and MCNC. It provides access for university researchers to a unique facility dedicated exclusively to experimentation with disruptive technologies (hence the term Breakable). Its unique feature is the ability to host researcher-owned equipment terminating the fiber at each site. The automatic switching in and out of equipment at the sites (BEN PoPs seen in Figure 1(b)) is

Author's address: I.Baldine, Renaissance Computing Institute, 100 Europa Dr., Chapel Hill, NC 27517

performed through the use of fiber switches. Experimenters also have the ability to monitor, access and reset their equipment remotely. The use of the facility is coarse-grain time scheduled based on experiment needs. The facility is managed by the representatives of the teams actively engaged in experimentation on it. RENCI acts as a caretaker of the facility and provides rack space and power for researcher equipment through its Engagement Sites on university campuses.



(a) BEN Topologies

(b) BEN PoP

Fig. 1. The BEN testbed consists of dedicated fiber and diverse edge resources maintained by the Triangle universities and RENCI, with access equipment installed and managed by RENCI through BEN PoPs.

2. DEMONSTRATING A NETWORK-WIDE VIRTUALIZATION MANAGEMENT PLANE.

BEN provides a unique platform for experimentation with resource virtualization. The ability to provide the researcher with unrestricted access to all networking layers starting with the physical and up, coupled with access from the facility to experimental compute and storage resources on individual campuses, allows to test various approaches to heterogeneous resource scheduling and virtualization. One example of such planned effort is RENCIs collaboration with Duke University on a recent proposal to the GENI Project Office.

The proposed work seeks to build on the ORCA (Open Resource Control Architecture) software platform developed by Jeff Chase of Duke University in order to create a GENI island using BEN. ORCA will play the role of the GENI management plane. ORCA emphasizes strong and general policy-neutral abstractions for dynamic sharing, resource control, and slice adaptation, with well-defined interfaces for modular policies. The proposal seeks to extend ORCA to scheduling and configuring of network resources (such as individual fibers, wavelengths, routes and tunnels) in addition to compute and storage resources, to enable end-to-end slicing (a term proposed in the GENI framework) of a network. An important partner in the proposal is Infinera Coroporation, whose DTN bandwidth provisioning platform RENCI is in the process of deploying on BEN. Infineras Bandwidth Virtualization features provide a unique capability enabling experimentation with network virtualization and slicing. RENCI and Duke University with engineering support from Infinera will extend ORCA to enable network resource virtualization (*slivering*) at the physical (fiber paths), bandwidth provisioning (DTN) and packet (Cisco and Juniper routers) layers and demonstrate the concept of end-to-end slicing using BEN infrastructure. Enabling end-to-end slicing is considered crucial to the success of the future GENI facility [Elliott 2007].

The ORCA project software is a mature research-grade prototype that has been in development since 2004. It comprises several modular components: a common leasing package (Shirako [Irwin et al. 2006]); security-related functions for accountable lease contracts and broker delegation (based on SHARP); and Cluster-on-Demand (COD), a back-end resource manager for cluster aggregates [Chase et al. 2003; Irwin et al. 2006]. Shirako is a Java-based toolkit for building resource leasing services using Web services technologies. It is structured as a common core with support for generic, recoverable lease state machines and well-defined plugin interfaces for extension modules that are configuration-specific (*handlers*), resource/component-specific (*drivers*), or policy-specific (*controllers*).

We plan to enable GENI sliver control for a number of resources beyond those supported in the current ORCA software. These include large-scale network storage, fiber switches enabling a reconfigurable optical plane, bandwidth provisioning platforms and MPLS-capable IP routers.

ORCA supports basic capacity slivering of local storage for Linux/Xen virtual machines using standard LVM (Logical Volume Management) mechanisms. Dynamic resizing of volumes containing filesystems is possible in principle, but is still problematic for common filesystems. ORCA also supports flash-cloning of root volumes for NFS-rooted virtual machines using NetApp and OpenSolaris/ZFS filers. For this project we will demonstrate capacity slivering, dynamic volume export/mounting, and approximate performance slivering of network storage, e.g., using bandwidth throttling such as a Request Windows technique [Jin et al. 2004].

Sliver control on BEN's networking equipment will be implemented in proxy ORCA component agents running plugin drivers for each type of equipment. Each agent runs on a blade server at the BEN node, under the direction of the authorized domain authority for its aggregate. Some pieces of equipment, notably the Cisco 6509 already provide useful pseudo-virtualization interfaces, e.g. the VRF-lite (VPN Routing/Forwarding Table) mechanism. In some cases there are several options available in each case to control individual devices, e.g., TL1 and SNMP interfaces. We will select on practicality and ease of implementation when necessary to meet our demo targets. In other cases the driver can issue configuration commands through the vendor-provided operator interface (e.g., CLI). We plan to allocate dynamic resources (e.g., IP address spaces) automatically, while relying on canned configuration descriptions in areas where automated generation of effective configurations is not yet available.

We propose to deploy an ORCA-based system to orchestrate virtualized slicing for the BEN testbed substrate and attached servers and storage. The BEN network substrate will serve as an interconnection between cluster sites at BEN points of presence. We plan to use COD to manage dynamic slivering of servers at the BEN switching points and the attached clusters at Duke and at RENCI. The slivering will be based on Xen virtual machines, for which we have well-developed support (handlers and drivers).

Initially, RENCI will operate a single broker/clearinghouse for the BEN testbed, which will control access to the attached clusters and storage incorporated into the testbed. Users will submit requests to the clearinghouse through a Web portal. The portal will be based on the Automat [Yumerefendi et al. 2007] portal from the ORCA software release. In addition to providing basic interfaces for operators and users, Automat allows users to upload and install custom *view* portlets and automated guest controllers to monitor and adapt their slices.

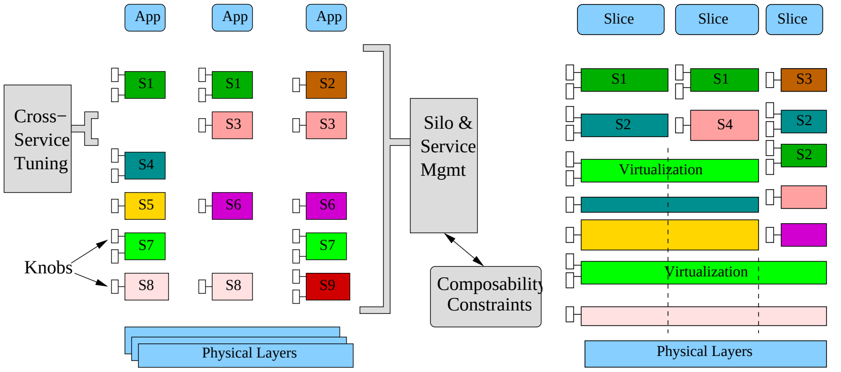
The clearinghouse is the policy enforcement point for resource allocation and scheduling policies for the testbed, which are implemented in a broker controller plugin. The initial policy will present each request for approval from a human administrator through the portal. Development of practical automated policy controllers is an important focus of synergistic research, but it is out of scope for this proposal. One such controller enforcing proportional sharing of a group of simple aggregates based on user identity [Grit and Chase 2008] and suitable for homogeneous resources has been prototyped by Chase. We will investigate adapting a similar approach to a single fiber with bandwidth virtualization, however devising effective automated slicing policies for a general multi-layered network is a significant research challenge out of scope for this effort.

The initial deployment will have a single connectivity provider: RENCI, which provides access to the fiber through the BEN equipment. The software mechanisms are sufficiently powerful to manage a network with multiple providers and some form of federation, e.g., providers peer with a common bandwidth broker and trust it to schedule allocations for subsets of their resources at specific times. Indeed, the BEN governance agreement requires that power, since the fiber is jointly owned by several institutions.

In order to prove the success of our approach to end-to-end slicing we will demonstrate experiments on slices created using ORCA. Examples of the experiments we envision are: (**a**) using Bittorrent to synchronize multiple distributed datastores on BEN, while varying the underlying topology, (**b**) running experiments involving scheduling of compute resources at several sites using Plush API on top of BEN.

3. ENABLING VIRTUALIZATION OF RESOURCES THROUGH A NOVEL NETWORK PROTOCOL FRAMEWORK.

Another example of a proposed experimentation with virtualization on BEN is RENCIs collaboration with NCSU (George Rouskas and Rudra Dutta) on an NSF FIND (Future Internet Design) funded SILO (Services Integration controL and Optimization) project. The SILO framework is an attempt to re-engineer a networking stack and escape from the current state of the protocol stack ossification. This is achieved by discarding a notion of rigid layers in a networking stack, and replacing it with vertically arranged sets of services, which are smaller in functionality than typical ISO layers. Such an arrangement of services is referred to as a silo. The advantage of SILO architecture is that silo structure can be individualized per-flow according to an application request. Each service within a silo has a well-defined interface to a service above and below it, as well as a set of knobs and gauges, needed to facilitate cross-layer interactions and optimizations. This flexible arrangement allows for a clean separation between processing of the datapath, cross layer interactions and optimization behavior, which in todays protocols are inextricably locked in a monolithic implementation (e.g. TCPs flow control). This separation encourages experimentation with individual protocols and optimization algorithms and helps advance the state of network science.



(a) Generalization of protocol layering (b) Successive virtualization

Fig. 2. SILO approach to protocol layering and realization of virtualization

Such an architecture has some unique implications for network resource virtualization. So far, network virtualization has been strongly coupled to the platform and hardware of the substrate. Logically, however, the virtualization of networks is really composed of many coordinated individual virtualization capabilities, distributed over networking elements, that share the common functionality of maintaining resource partitions and enforcing them. We can view these as separate and composable services, the most basic of which is splitting and merging of flows that provide isolation between users thus enabling virtualization. SILOs ability to construct custom service arrangement allow for selective virtualization of certain resources or hardware based on user requirements. SILO framework also allows for recursive virtualization or slicing thus enabling for generalized virtualization, where an owner of an individual network slice can seamlessly virtualize and slice it further for subsets of its users or customers.

In the networking context, virtualization is usually interpreted as implying two capabilities beyond simple sharing. The first is *isolation:* each user should be unaware and unaffected by the presence of other users, and should feel that it operates on a dedicated physical network. This is sometimes also called “slicing.” This can be broken down into two services: (i) slice maintenance, which keeps track of the various slices and the resources used by them, and (ii) access control, which polices the resource usage of each slice, as well as deciding whether new slices can be created or not; for example, rate control such as leaky bucket would be an access control function.

The second is *diversity:* each user should be able to use the substrate in any manner in which it can be used, rather than being restricted to use a single type of service (even if strictly timeshared). This is akin to the ability to run different operating systems on different virtual machines. In SILO, this capability is natively supported, through the composable nature of the stack. Not only do different silos naturally contain different sets of services, but the composability constrains provide a way to indicate what set of upper services may be chosen by different slices when buidling on a particular virtualized substrate.

Following the principle that a virtual slice of a network should be perceived just like the network itself by the user, we are led to the scenario that a slice of a network may be further virtualized. A provider who obtains a virtual slice and then supports different isolated customers may desire this scenario. The current virtualization approaches do not generalize gracefully to this possibility, because they depend on customized interfaces to a unique underlying hardware. If virtualization is expressed as services, however, it should be possible to design the services so that such generalization is possible simply by re-using the services (see Figure 2(b)).

It may appear from this discussion that in fact with per flow silo states, there is no need to virtualize, and in fact it is possible to extend all the slices to the very bottom (dotted lines in Figure 2(b)). However, the advantage lies precisely in state maintenance; a service which is not called upon to distinguish between multiple higher level users can afford to keep state only for a single silo, and the virtualization service encapsulates the state keeping for the various users.

BEN provides a convenient platform to test SILO ideas using the prototype software developed within this project. The researchers plan to implement virtualization-enabling services (e.g. splitting and merging) and test the concept of SILO-enabled virtualization using RENCIs equipment on BEN mentioned earlier.

To facilitate experimentation with the SILO framework on BEN Dell blade servers equipped with 10G XFP SR (Short Reach) NICs will be used to validate slicing approaches enabled by SILOs. Specifically, SILO prototype framework will be loaded on one or more servers at each BEN site, with servers communicating with each other over paths provisioned using Infinera DTN equipment. SILO-enabled slicing will be demonstrated with multiple silo arrangements facilitating bandwidth slicing and slicing of routing contexts.

4. CONCLUSIONS

BEN is a unique capability provided to Triangle researchers for the purposes of experimentation with disruptive technologies. Its flexibility makes it an ideal platform for testing of novel ideas and practical approaches to many of the exciting problem areas of today and the foreseeable future, such as virtualization. BENs existence allows some of the ideas to be tested at scale under real-world conditions and helps refine solutions, bridging the gap between lab experimentation and real world applications.

REFERENCES

Chase, J. S., Irwin, D. E., Grit, L. E., Moore, J. D., and Sprenkle, S. E. 2003. Dynamic Virtual Clusters in a Grid Site Manager. In *Proceedings of the Twelfth International Symposium on High Performance Distributed Computing (HPDC)*.

Elliott, C. 2007. Current top GENI risks. Presentation at the First GENI Engineering Conference.

Fu, Y., Chase, J., Chun, B., Schwab, S., and Vahdat, A. 2003. SHARP: An Architecture for Secure Resource Peering. In *Proceedings of the 19th ACM Symposium on Operating System Principles.*

Grit, L. and Chase, J. 2008. Weighted fair sharing for dynamic virtual clusters. In *SIGMETRICS 2008 (accepted as a poster).*

Irwin, D., Chase, J. S., Grit, L., Yumerefendi, A., Becker, D., and Yocum, K. G. 2006. Sharing Networked Resources with Brokered Leases. In *Proceedings of the USENIX Technical Conference.*

Jin, W., Chase, J. S., and Kaur, J. 2004. Interposed proportional sharing for a storage service utility. In *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems (ACM SIGMETRICS/Performance).*

Yumerefendi, A., Shivam, P., Irwin, D., Gunda, P., Grit, L., Demberel, A., Chase, J., and Babu, S. 2007. Towards an Autonomic Computing Testbed. In *Workshop on Hot Topics in Autonomic Computing (HotAC).*

...