Ripcord: A Modular Platform for Data Center Networking

Brandon Heller,
David Erickson,
Nick McKeown
Stanford University
Stanford, CA, USA
{brandonh, derickso, nickm}
@stanford.edu

Rean Griffith,
Igor Ganichev,
Scott Shenker
University of California
Berkeley, CA, USA
{rean, igor}
@eecs.berkeley.edu
shenker@icsi.berkeley.edu

Kyriakos Zarifis
International Computer
Science Institute
Berkeley, CA, USA
zarifis@icsi.berkeley.edu

Daekyeong Moon
Nicira Networks
Palo Alto, CA, USA
dkmoon@nicira.com

Scott Whyte,
Stephen Stuart
Google, Inc.
Mountain View, CA, USA
{swhyte, sstuart}@google.com

ABSTRACT

In this demo, we present Ripcord, a modular platform for rapidly prototyping scale-out data center networks. Ripcord enables researchers to build and evaluate new network features and topologies, using only commercially available hardware and open-source software. The Ripcord demo will show three examples of custom network functions, operating together, on top of a 160-node cluster. The first is a routing engine that isolates classes of traffic. The second is a dynamic network manager than adjusts links and switch power states to reduce energy. The third is a statistics aggregator that supports network health monitoring and automatic alerts. The demo will be interactive, with a visualization of live parameters for each link and switch, such as bandwidth, drops, and power status, as well a control panel to modify the traffic load. We feel that an interactive demo is the best way to introduce the research community to Ripcord and get their feedback.

Categories and Subject Descriptors: C.2.1 [Network Architecture and Design]: Network communications; C.2.2 [Network Protocols] Routing protocols
General Terms: Design, Experimentation, Management
Keywords: Data center network, Ripcord, OpenFlow

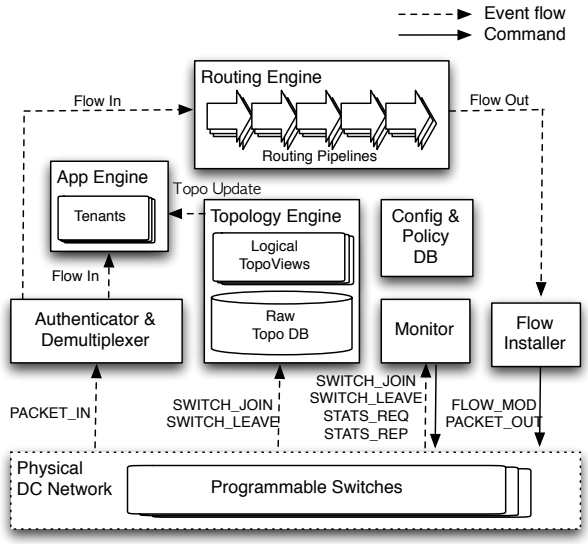

Figure 1: Ripcord Architecture

1. DETAILS

This section describes each major component of the proposed demo, including the base Ripcord platform, interactive dashboard, custom modules, and hardware setup. Note that we have a different goal from the SIGCOMM 2009 FlowVisor demo [6]. Instead of multiple controllers sharing an unstructured enterprise network, our goal is to show a single modular controller operating on structured data center networks.

1.1 Ripcord

See Figure 1 for an overview of Ripcord. The architecture includes a number of primitives for building new data center routing algorithms and management tools, and is intended to help researchers address key design challenges in the data center, including scalability, server migration, and forwarding state. Ripcord leverages NOX [2], an OpenFlow controller platform, to pass messages between modules and to modify and view switch state (such as flow entries and statistics).

The Ripcord prototype implements multiple data center routing engines, including ones with similar elements to VL2 [1] and PortLand [4], as described in [7]. Ripcord can simultaneously run multiple routing schemes on the same network, enabling side-to-side comparisons as well as distinct routing engines for different services. Each routing engine uses the structured topology representation; for example, PortLand-style routing can run on a VL2-style aggregated topology, with no code changes.

SIGCOMM'10, August 30–September 3, 2010, New Delhi, India.
ACM 978-1-4503-0201-2/10/08.

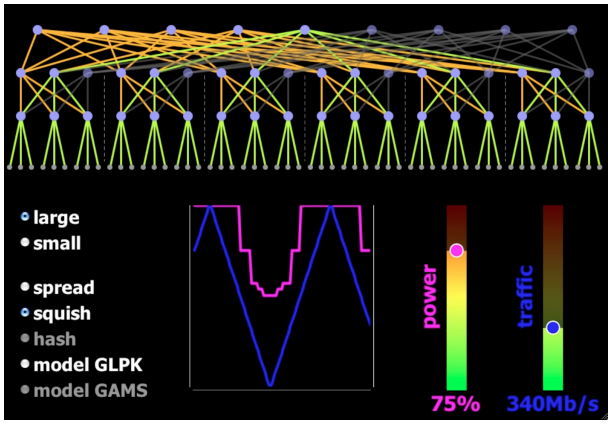

Figure 2: Example Dashboard. Here, ElasticTree has turned off some switches; the buttons are user-configurable parameters.

1.2 Dashboard

Visualizing network state in real time is the key to demonstrating - and understanding - Ripcord. Attendees will see a GUI like the one shown in Figure 2. The upper part of the dashboard displays the network topology. Atop this background, one can see the utilization of every link, with traffic-light color-coding: green for lightly loaded links, yellow for moderately loaded links, and red for highly loaded links. If a node or link is intentionally disabled by a module - or by the demo attendee - the node appears with an X. Each application has its own parameters to control, which might be the duration of each collection epoch for the statistics aggregator, or fault tolerance and link utilization parameters for ElasticTree. The bottom part of the dashboard displays high-level statistics. A graph communicates changes in power, throughput, and latency over time.

1.3 Integrated TE and QoS

The Traffic Engineering (TE) management tool, built on top of Ripcord, gives routing preference to specific classes of traffic, such as video streams. Flows for these classes are routed along paths reserved by a network administrator, ahead-of-time. In addition, the TE tool maintains sets of pre-calculated re-routing actions to take, should a link or switch fail. We will demonstrate tunnel setup based on different network requirements and optimization functions as well as the failover procedure, measuring response times and network disruption caused by tunnel preemptions.

1.4 ElasticTree

ElasticTree is a dynamic optimizer that tries to shut off as many unneeded network elements as possible, while respecting performance and fault tolerance constraints [3]. Given a traffic matrix and network topology, ElasticTree generates the set of switches and links that need to be active. A range of optimizers have been implemented, which vary in optimality, generality, and solution time. This demo will extend the ElasticTree paper to a larger system with more realistic application traffic. Demo attendees are encouraged to play with the system to explore tradeoffs between energy, performance, and fault tolerance.

1.5 Aggregated Statistics

The Aggregated Statistics application collects flow, switch and link statistics from the network and provides the underlying data to both ElasticTree and the visualization. It also enables interactive queries of displayed network elements. For example, users can obtain detailed metrics on individual (or aggregated) flows in the network. They can gather and track individual port measurements from switches, or track link utilization trends to assess the impact of changing traffic engineering and power management decisions.

1.6 Data Center Platform

The expected platform is a 160-node cluster. The network is organized as a three-layer fat tree with four-port switches, except that instead of using two 10 Gbps downlinks, each edge switch uses 20 1 Gbps downlinks to hosts. Each switch runs a port of OpenFlow, an open-source, vendor-neutral, flow-based API added to commercial switches, routers and access points [5]. When an OpenFlow-enabled switch receives a packet for which there is no matching flow entry, it is sent to a controller that makes a decision to add flow entries in switches as needed to set up a path. Alternatively, as is done by some Ripcord routing engines, paths can be set up in advance.

Even though the prototype is not production quality, we believe that Ripcord presents a compelling framework for researchers to implement, evaluate, and (eventually) deploy new ideas. The three modules created for this demo show its flexibility, and we expect the visualization to be useful both in debugging new Ripcord modules, as well as understanding the traffic patterns of data center applications.

2. REFERENCES

[1] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta. VL2: A Scalable and Flexible Data Center Network. In *Proc. of SIGCOMM*, Barcelona, Spain, 2009.

[2] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, and N. McKeown. Nox: Towards an operating system for networks. In *ACM SIGCOMM CCR: Editorial note*, July 2008.

[3] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown. Elastictree: Reducing energy in data center networks. In *NSDI'10: Proceedings of the 7th USENIX Symposium on Networked Systems Design and Implementation*, 2010.

[4] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat. Portland: a scalable fault-tolerant layer 2 data center network fabric. In *SIGCOMM '09: Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, pages 39–50, New York, NY, USA, 2009. ACM.

[5] The openflow switch. `http://www.openflowswitch.org`.

[6] R. Sherwood, M. Chan, A. Covington, G. Gibb, M. Flajslik, N. Handigol, T. Huang, P. Kazemian, M. Kobayashi, J. Naous, et al. Carving research slices out of your production networks with OpenFlow. *ACM SIGCOMM Computer Communication Review*, 40(1):129–130, 2010.

[7] A. Tavakoli, M. Casado, T. Koponen, and S. Shenker. Applying NOX to the Datacenter. In *8th ACM Workshop on Hot Topics in Networking (Hotnets)*, New York City, NY, October 2009.