Abstract: A new SDN control plane architecture for hybrid optical/electrical switch networks is demonstrated. Compared to traditional VLAN-based control, our OpenFlow implementation is more scalable and enables faster reconfiguration of multihop or multipath networks.

1. Introduction
Hybrid optical/electrical networks, formed by adding an optical circuit switched (OCS) network in parallel to a conventional electrical (packet-switched) network, have been extensively studied in recent years in order to alleviate bandwidth and latency bottlenecks at potentially lower cost/power than with pure electrical switch networks. While a number of workload classes require optical switch times on the order of microseconds [1], recent studies have revealed that many other data center workloads, such as VM migration, VM check-pointing [2] and big data computation, generate concentrated traffic which can benefit from high bandwidth optical circuits with tens of millisecond switching time, which are offered by off-the-shelf large radix optical switches based on 3D MEMS mirrors. Most past work has limited routing to loopless rack or POD matchings (bipartite routing graphs) [3], [4]. While effective in avoiding the pain of an efficient electronic control-plane, this approach leaves two important techniques out of the design space: a) routing over multiple paths (multi-path) and b) supporting opto-electronic circuits spanning multiple Top-Of-Rack (TOR) switches (multi-hop). Multi-hop routing extends the connectivity degree of a rack beyond the set of racks it is optically paired with (1-hop reachable), thereby relaxing the need for frequent optical reconfiguration [5]. Multi-path routing can be used to further increase throughput by fully utilizing available single-/multi-hop circuits. Both techniques are mentioned but not described in detail in the OSA architecture [6].

This paper outlines a control-plane supporting loop-free multi-hop and multi-path routing in a hybrid optical/electrical datacenter network using two alternative technologies: 802.1q [7] (VLAN-based) forwarding and OpenFlow (OF [8]). We have implemented these control planes and present evaluation results obtained on a downsized yet representative real test-bed. Our preliminary measurements manifest that OF-based control of edge switches yields a significant reduction in the time overhead required to manipulate forwarding state as part of dynamic optical circuit management, leading to faster affinity of the topology to workload requirements and thus higher throughput.

2. Forwarding on the network edge
The role of the control-plane in a hybrid datacenter network architecture (cf. left side of Figure 1A) is to constantly adapt the reconfigurable topology to traffic requirements by jointly cross-connecting optical switches and manipulating forwarding states at electronic switches that carry traffic through the optical circuits. The network controller implements a set of desired optical circuits using the underlying forwarding substrate. Given the need to route over a redundant Ethernet topology, forwarding on VLAN-tags satisfies functionality requirements, as
exemplified on the right hand side of Figure 1A and extensively shown in [9]. However, the limited number of VLAN-tags combined with the need for scalable systems forces a solution involving dynamic VLAN-tag manipulation on edge switches. Implementing the same routing using OF is straightforward by having the controller install required flow rules to TOR switches; this is elementary functionality in Software Defined Networking (SDN).

3. Evaluation

We implemented both solutions in the form of network controllers: a) a controller that dynamically manipulates VLAN-ids on TOR switches via SNMP calls and b) a Python application that manipulates flow rules at Openflow-1.0.0 compatible TOR switches through communication with the Floodlight OF controller [10]. The two implementations share among others the ability to dynamically cross-connect MEMS switches, as well as a multi-threaded implementation of all code dealing with network device manipulation, having a dedicated thread deal with a single network device for the purpose of faster completion of a full reconfiguration cycle through parallelization.

Figure 2 presents performance results we obtained through evaluation of the two solutions using the experimental setup shown in Figure 1B. All timing results correspond to end-to-end delay measured at the controller. We observe that setting the VLAN-tag on a single port (corresponding to a single optical circuit setup) takes 750ms, while synchronous circuit manipulation (i.e. touching all circuits on every reconfiguration cycle [1][2]) can cost up to 1.1 seconds (assuming that the 64 ports of the TOR switch are roughly equally split to server- resp. core-facing ports). On the contrary, we observe on the right-hand side of Figure 2 that forwarding state installation in OpenFlow is much faster, starting at 57 ms for a single bidirectional flow installation. This number is the end-to-end delay that could be reduced with a different controller implementation; the rule installations themselves can be well below 10ms and have low delay jitter [11]. Although it is beyond the scope of this paper to present an efficient flow rule creation scheme for efficient forwarding in a hybrid network, we note that a bidirectional wildcard entry suffices for re-routing a server's flow from the electrical network to an optical circuit.

Assuming 40 servers/rack, we scaled measurements up to 40 rule installations, observing throughout that the circuit setup delay scales linearly to the number of rules installed. We also plot the total reconfiguration delay, including the MEMS-mirror switching times. We note that a misconfiguration in our Python code calling the MEMS switch driver had the effect of doubling the MEMS reconfiguration times (~400ms) perceived at the controller, compared to the switching times perceived at the native switch driver (211ms). We observe that within the budget of roughly 20 bidirectional rules, optical reconfiguration governs the overhead of reconfiguration, while beyond that it is edge state installation that becomes the bottleneck. We acknowledge that this is sensitive to our test-bed setup, noting that OF may well become the bottleneck even for a single flow redirection, as optical switches based on 3D-MEMS mirror arrays can be optimized for switching times on the order of few milliseconds.

4. Conclusions

We implemented an OpenFlow-based control scheme on a hybrid optical/electrical switch network enabling multihop and multipath routing. Our measurements show a significant reduction in network reconfiguration time compared to traditional VLAN-based control. Our findings are in support of the software-defined networking concept as a performance enabler in hybrid networks, in addition to the arguments of convenience (ease of programming) and native flexibility (ability to build arbitrary routing graphs over a physical infrastructure).

References
