CORD: Central Office Re-architected as a Datacenter

Larry Peterson and Andy Bavier, *Open Networking Lab (ON.LAB)*

Network operators want to make their networks efficient, programmable, elastic and agile to meet the challenges of user bandwidth demands, as well as to create new revenue streams with innovative services. They want to benefit from both economies of scale (infrastructure constructed from a few commodity building blocks) and agility (ability to rapidly deploy and elastically scale services) that cloud providers enjoy today. These cloud-inspired benefits are especially needed at the edge of operator network: in the **Telco Central Office (CO)**. These facilities contain 50 years' worth of legacy purpose-built devices, making them both a source of significant CAPEX and OPEX and a potential barrier to rapid innovation.

In response to these challenges, network operators are re-inventing their networks. One such effort is **CORD**, a collaborative effort between **AT&T** and the **Open Networking Lab**. CORD re-architects the Central Office as a datacenter. The basic approach centers on unifying three related but distinct threads: **SDN**, which is about separating the network's control and data planes, and making the control plane open and programmable; **NFV**, which is about moving the data plane from hardware appliances to virtual machines; and the **Cloud**, which defines the state-of-the-art in building scalable services—leveraging software-based solutions, micro-service architecture, virtualized commodity platforms, elastic scaling, and service composition, to enable network operators to rapidly innovate. The goal of CORD is not only to replace today's purpose-built hardware devices with their more agile software-based counterparts, but also to make the central office an integral part of every Telco's larger cloud strategy and to enable them to support more attractive and meaningful networking services.

The target hardware for CORD consists of a collection of commodity servers and storage, interconnected by a leaf-spine fabric constructed from white-box switches. It is similar in design (albeit on a smaller scale) to a conventional datacenter; however, the switching fabric—organized as a *leaf-spine* topology—is optimized for traffic flowing east-to-west, between the access network that connects customers to the central office and the upstream links that connects the central office to the operator's backbone. There is no north-south traffic in the conventional sense.

With respect to software, our reference implementation of CORD exploits three open source projects. **OpenStack** is the cluster management platform, providing the core IaaS capability. **ONOS** is the network operating system that manages the underlying white-box switching fabric. It also hosts a collection of control applications that implement services on behalf of Telco subscribers. ONOS is also responsible for embedding virtual networks in the underlying fabric, which is in turn accessed via OpenStack's Neutron API. **XOS** is a service abstraction layer that unifies infrastructure services (provided by OpenStack), control plane services (provided by ONOS), and any data plane or cloud services (running in OpenStack-provided virtual machines). It provides explicit support for multi-tenant services, making it possible to create, name, operationalize, manage and compose services as first-class operations.

Just as all the devices in a hardware-based Central Office must be wired together in a meaningful way, their software counterparts must also be managed as a collective. This process is often called *service orchestration*, but if network operators are to enjoy the same agility as cloud providers, the abstractions that underlie the orchestration framework must fully embrace (1) the elastic scale-out of the resulting virtualized functionality, and (2) the composition of the resulting disaggregated (unbundled) functionality. Our approach is to adopt *Everything-as-a-Service (XaaS)* as a unifying principle. This brings the disparate functionality introduced by virtualizing the hardware devices under a single coherent model. The control functions run as scalable services (these functions run on top of ONOS, a scalable network operating system), the data plane functions run as scalable services (these functions scale across a set of VMs), the commodity infrastructure is itself managed as a service (this service is commonly known by the generic name IaaS), and various other global cloud services running in the central office are also managed as scalable services. No matter the role it plays in the overall CORD architecture, each service is structured in exactly the same way: it supports a logically centralized interface, called a *service controller;* it is elastically scale across a set of *service instances* (corresponding to VMs and OpenFlow switches); and it is multi-tenant with an associated *tenant abstraction*.

The plan is to build a fully functional reference implementation model, followed by validating the architecture through lab trials and eventually field trials, and hardening the system along the way based on trial data. Though CORD is still in the lab, we have already bumped into many research challenges. A few examples: new system abstractions for multi-layer, multi-cloud service orchestration; primitives for composing new Cloud services using multi-tenant building-block services; NFV function placement based on service requirements and real-world constraints; isolating services, and tenants within multi-tenant services; and reducing the attack surface of services that leverage Clouds, SDN, and NFV in combination. We believe that building new breeds of Clouds and Cloud applications to satisfy real-world requirements is a fruitful area for future research.

CORD development has leveraged the **CloudLab** facility. However it's not clear that existing testbeds are adequate to support a large push into this space. CloudLab has provided value both through the ability to easily spin up OpenStack clusters, as well as to allocate bare-metal machines. This has increased our development speed and made it easier to collaborate with distributed team members. On the other hand we have sometimes been hampered by node shortages and transient failures on CloudLab. We would also like to see modern SDN capabilities (e.g., OpenFlow 1.3) widely available within and between test facilities. Finally we think that the value of Cloud resources outside of the datacenter—e.g., at the network edge (like PlanetLab) and in the core (VINI)—will become more apparent as the research community explores new paradigms; we encourage investment in this area.