Cisco Application-Centric Infrastructure: A Surprisingly Open Architecture

Introduction

Openness is an essential element of next-generation network architectures. Open network technologies can reduce costs, enable cross-domain infrastructure automation, and enhance monitoring, troubleshooting, and traffic engineering capabilities. By providing open, programmable access to their platforms, vendors can help network operators and developers build highly responsive, scale-out network architectures that are cheaper to build and operate. Despite criticism from open networking proponents who claim it is a closed, proprietary system, Cisco's Application Centric Infrastructure, its flagship software-defined networking solution for data centers, is a surprisingly open technology. Cisco's open APIs enable enterprises and technology providers to integrate third-party systems into the product. Also, the components of ACI are extremely programmable.

By providing open, programmable access to their platforms, vendors can help network operators and developers build highly responsive, scale-out network architectures that are cheaper to build and operate.

Openness Is the New Currency in Networking

To some extent, openness has always been an essential ingredient in networks. Vendors have long embraced open standards as a means of interoperating with competitors' products. But at the same time, network systems have long been closed platforms. Switches and routers may use open standards to communicate, but they often use proprietary extensions of those open standards. And the network platforms themselves feature proprietary vertical integration of hardware and software. Network equipment manufacturers use these extensions and this vertical integration as a means of differentiation and innovation. But such an approach has produced a market where network operators get locked into relationships with vendors. Moreover, network operators have very little control over their own futures. If they need new features or a new integration with third-party technologies and management tools, they usually have to lobby their vendor and then wait for a new product release that supports those features and integrations.

In the realm of software-defined networking (SDN), openness is the new currency. Myriad open standards, open source projects, and open architectures have emerged to meet the demands of enterprise and telecom engineers, who seek rapid innovation, customizability, network agility, and programmability. Many people believe that openness can prevent vendor lock-in, provide architectural modularity, and promote interoperability. The original architectural vision of SDN, with its separation of the network control plane and the data plan, promised a certain degree of openness, with controllers centralizing all high-level functionality in software and low-functioning, interchangeable switches and routers taking direction from those controllers via OpenFlow or another protocol. Furthermore, an ecosystem of bare-metal switches have emerged, where software and hardware providers pursue separate but parallel development paths in order to deliver faster innovation and the freedom to pick and choose among multiple hardware and software vendors.

When Cisco introduced Application-Centric Infrastructure (ACI), the company drew criticism from some quarters for building what critics called a proprietary closed architecture that eschewed open standards like OpenFlow in favor of new Cisco-invented open standards such as OpFlex. With ACI, Cisco declined to separate the control plane and data plane. It declined to offer disaggregated hardware and software. It continued to differentiate its products by offering a vertically integrated software and hardware stack.

In today's climate, where some networking vendors gain a certain degree of "street cred" by embracing openness, Cisco has been under fire for going in the opposite direction with ACI. But is this criticism

fair? After receiving multiple technical briefings, viewing multiple product demonstrations, and talking to several Cisco ACI customers, Enterprise Management Associates (EMA) believes that Cisco's ACI data center solution is far more open than people realize. In fact, it is may be the most open networking solution Cisco has ever produced.

The Openness of Application-Centric Infrastructure

Let's start with the basic building blocks. Like many network products from Cisco and its competitors, ACI relies on open standards in its basic architecture. Cisco uses the virtual extensible local area network (VXLAN) protocol to encapsulate traffic within ACI. VXLAN is used by many leading software-only network virtualization overlays to do the same thing. Cisco has added several extensions to VXLAN to support things like group-based policy. However, Cisco has taken the unusual step of open sourcing these VXLAN extensions and offering them to the Linux kernel. Those extensions have been accepted.

ACI also uses OpFlex, a Cisco-invented protocol that allows ACI's APIC controller to exchange policy information with the Nexus 9000 switches in an ACI fabric. Although no other vendors have publicly adopted this protocol, Cisco has made it open and proposed it as a standard to the Internet Engineering Task Force (IETF). It has created an open source OpFlex agent for Open Virtual Switch (OVS), the popular open source virtual switch used in many leading SDN architectures. The OpFlex agent was designed to be reusable so it can be integrated into other third-party networking devices as well. ACI also uses plenty of other open protocols, such as IS-IS.

Cisco has also created an open source project called Group-Based Policy (GBP) within the OpenStack community. GBP was designed to offer some of the benefits of ACI's application-centric policy model but it does so in a completely vendor-agnostic manner, supporting any Neutron plugin. In this case, rather than keep its policy model proprietary, Cisco has engaged with the community to open it up and encourage interoperability across vendors.

The Openness of ACI's APIC Controller

APIC, Cisco's policy controller for an ACI fabric, is the architectural element where Cisco really starts to distinguish itself for openness. Cisco has introduced an open REST-based application programming interface (API) that exposes the full object model of the APIC controller. This API gives everyone access to all the telemetry and configuration information inside an ACI fabric. Network operators and their third-party vendors can use the API to integrate a wide variety of tools into APIC for orchestration, automation, monitoring, diagnostics, and traffic analysis.

Cisco has introduced an open REST-based application programming interface (API) that exposes the full object model of the APIC controller.

APIC will respond to REST API calls with XML- or JSON-based data outputs that contain a complete object hierarchy detailing telemetry and configuration data by tenant, application profiles, endpoint groups, and so on. Network operators and developers can refine these API requests by adding new objects that can search for and retrieve data according to very specific parameters. For instance, they can pull all the data for bridges associated with a specific tenant and display that data in whatever management tool they are using.

Cisco has released a Python software development kit (SDK) and an ACI toolkit to encourage its customers to leverage the REST API. The ACI toolkit is an open source API wrapper that gives standard REST clients simple tools for collecting from APIC a subset of the most popular capabilities that Cisco has identified. The Python SDK is a more comprehensive set of tools that helps developers define objects for extracting any conceivable data sets from APIC.

While Cisco has multiple technology partnerships with management vendors—including CA, Splunk, Puppet Labs, and CFEngine—that take advantage of this openness, developers of any management tool can

work with the APIC API. In fact, EMA spoke with an ACI customer that successfully lobbied the customer's preferred infrastructure monitoring vendor to integrate its product with APIC to extract configuration and telemetry data, even though that management vendor had no formal technical partnership with Cisco. This data is far more rich and detailed than anything that could be collected via flow records or Simple Network Management Protocol (SNMP) polling and traps.

Cisco's New Approach to API-Based Switch Programmability

The Nexus 9000 switch family is the hardware foundation of ACI. It runs an ACI-mode operating system optimized for the SDN fabric. Network engineers can also run the Nexus 9000 in "standalone" mode by booting up a streamlined version of Cisco's NX-OS operating system, which is optimized for leaf-and-spine networking. Cisco recently updated the version of NX-OS that runs on the Nexus 9000 to make these switches more open and programmable than any other switch Cisco currently offers. Although Nexus 9000 switches running in NX-OS standalone mode are not part of an ACI fabric, this new level of programmability warrants mentioning in this context. It gives Cisco customers another option for taking advantage of open networking in a Cisco environment.

Cisco has offered some level of programmability on its Nexus switches for several years, but until now there had been limitations. Nexus switches offered Python and NETCONF APIs that were mapped to the command line interface (CLI) that engineers used to configure Nexus switches and extract information from them. As network engineers know, CLI is essentially a prescriptive set of instructions defined by Cisco developers that enable very specific interactions between a network engineer and NX-OS. API calls that had to go through CLI instead of directly accessing NX-OS objects were inefficient and limited in functionality. They also required complicated NX-OS-specific formatting that was beyond the skillset of most mainstream enterprise developers working with APIs.

Now Cisco has introduced new REST APIs that are directly exposed to the objects within NX-OS. This enhancement lowers the bar for third-party integration with a Nexus 9000 switch. Any tool that has a web services architecture can integrate with a Nexus 9000 switch using this new API. A log analytics tool, for instance, could poll the API on every switch and extract machine data, index it, and make it immediately searchable. Though such an operation was possible over previous Cisco APIs, it was much more difficult to perform. As the formatting requirements of those legacy APIs were not available in standard programming tools, a developer needed a very specialized skillset to collect such data. With Cisco's new REST API for Nexus 9000 switches, mainstream developers can program and integrate with the switches much more rapidly and effectively.

With Cisco's new REST API for Nexus 9000 switches, mainstream developers can program and integrate with the switches much more rapidly and effectively.

Cisco has further enhanced NX-OS for Nexus 9000 by exposing direct access to Linux utilities. This gives network operators a Linux-based alternative to CLI management. These utilities—such as RPMs, bash shell, and tcpdump—are now native to the operating system. Enterprises now have the option to essentially manage a Nexus 9000 like a Linux server. Data center operators can consolidate server and network management and use systems administration tools to manage the network, if that is their preferred operating model. This feature directly addresses the needs of data center operators who turn to bare-metal switch software provider for such integrated management capabilities.

Cisco has further opened the platform by exposing the ability to run Linux containers directly on the Nexus 9000 switch. This allows network operators or third-party party vendors to directly embed their own applications or agents on the switch for tighter integration with the surrounding environment.

EMA Perspective

In both its research and its interactions with enterprise IT organizations, EMA has found that open, programmable architectures are a top priority for the data center. Last year EMA surveyed enterprises that were adopting aspects of a software-defined data center (SDDC) and found that 70% of them had initiatives in place to enable developers to take better advantage of data center infrastructure.[1] They were doing this both by adding programmability to existing infrastructure and by adopting new programmable infrastructure. These emerging requirements demand open and programmable products across all technology domains, but particularly in the network, which has traditionally lagged behind the rest of the IT industry in this regard.

The open networking movement, characterized by SDN, bare-metal switching, new open standards, and open source software, is seeking to address these requirements. It is EMA's assessment that the openness and programmability of Cisco ACI addresses these many of these requirements. Cisco ACI is far more open than the industry gives it credit for. EMA recommends that enterprises looking for open programmable solutions assess ACI to determine whether it meets their requirements. Its approach to providing an open and programmable network may contrast sharply with the approaches adopted by advocates of bare-metal switching and open source network software, but Cisco is offering a degree of openness that was previously unheard of in Cisco networks.

Cisco is offering a degree of openness that was previously unheard of in Cisco networks.

[1] EMA, "Obstacles and Priorities on the Journey to the Software-Defined Data Center," January 2014.