ZIVARO

···· BRILLIANT IT ····

Building Network Automation Maturity with Cisco Network Programmability

BJ Howell
Enterprise Network Architect

INTRODUCTION

The complexity of modern network technologies along with the speed of business and economic changes make it difficult for networks engineers to keep up with the rate of change at scale. Yes, many enterprise network engineers are hesitant to dive into the world of network programmability. Enterprise network administrators seldom have time to create customized network programmability scripts on their own, even if those scripts can save time in the long-run.

Zivaro recommends enterprises take a two-pronged approach to network programmability. First, organizations should work to educate the teams on the topic and start with small simple scripts based on readily-available examples. Second, they should use vendor-supplied network programmability software to quickly gain the efficiencies and enterprise-wide visibility and control. Along these lines, Zivaro provides a roadmap to enterprises to slowly and steadily build maturity with network programmability in a disciplined and intentional way.

NETWORKING CHALLENGES TODAY

Building and operating networks is hard enough, but when the pace of technological innovation is high, it makes it incredibly difficult to keep up with the rate of change. Businesses have high expectations of network availability and performance and the complexity of modern networks adds to the pressure on network administrators.

Network engineers have expressed being intimidated by these new software-driven configuration methods. They are familiar with using Secure Shell (SSH) command line configuration of network devices because they have spent years learning these commands and striving for product certifications that require this knowledge. Network engineers can be unfamiliar with the software-driven approaches that may require one to learn a computer programming language or write software from scratch.

While the network engineer is familiar with typing configuration commands, using network programmability methods that simply use the Command Line Interface (CLI) configuration commands is not necessarily "transformative". Using accessible methods like Ansible with a native CLI-command module doesn't necessarily leverage the power of software-driven approaches. While this may be a quick-and-easy approach to get something working, there may be more power in the web-based Representational State Transfer (REST) RESTful Application Programming Interface (API) methods. The reality is that APIs aren't necessarily built for humans and can be challenging for network engineers who lack software development experience. Network engineers tend to be task-oriented and

have difficulty achieving scalability when they perpetuate their legacy practice of manual device configuration. Network engineers are pressed for time and they don't have "leisure time" to learn these new software-driven approaches or to learn a programming language like python. They don't have time to build their own tools from scratch, much less have time to take an existing script they find on the Internet (or GitHub) and customize it for their environment.

Network engineers don't have the required skills to write complete python orchestration programs from scratch. It is more feasible for network engineers to re-reuse python scripts that others have already written. Network engineers may find many examples on using Netmiko or NAPALM well documented and easy to integrate into simple customized python scripts. For example, network engineer may not even create their own Ansible playbooks with Jinja2 templates, but they might if they are shown other examples they could follow. It would be easier for them to interface with a vendor's management system that is making the proper API calls to the network devices.

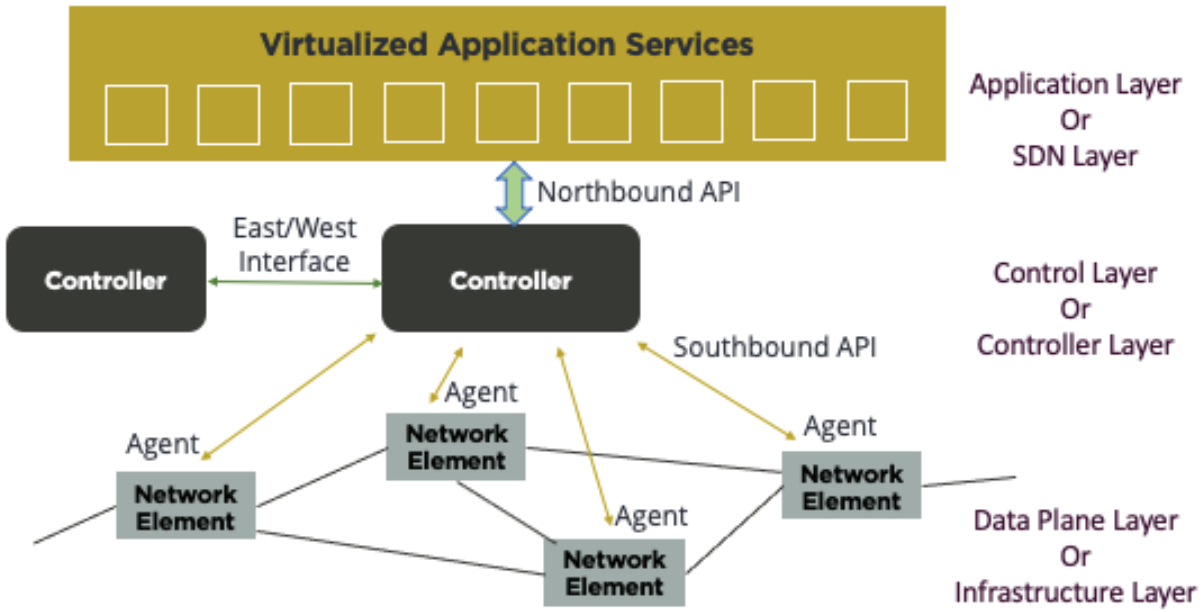
BACKGROUND

Most enterprises have made a significant investment in on-premises networks which are connected to remote data center facilities and are extending out to cloud infrastructure. For years, the networking teams have manually configured individual network devices by hand with specific configuration commands. Each network device is uniquely configured. Maintaining vast amounts of individual physical and virtual network, compute, storage, security systems, and all that encompasses creates surmounting technical debt.

Large enterprises are beginning to set goals to “automate everything” and strive to reduce human error, increase accuracy of configurations, reduce inconsistencies in network device configurations, and create repeatable processes for rapidly deploy new applications in support of the fast-moving global economy. Enterprises that are documenting goals of using a network programmability system often come up with a list like this.

- Reduce human error and inconsistencies in bespoke network device configurations and allow for configuration idempotence.
- Help identify configuration issues, devices with configurations that don't match the ideal "golden config". Finding the one device that isn't configured like the others and removing inconsistencies can increase Mean Time To Repair (MTTR).
- Save time, reduce time-to-market of new connectivity and speed up deployment of new applications.
- Improve the utilization of network administration staff to allow them time to focus on projects that drive more business benefit for the company, and relieve them of the burden of excessive administrative burden.
- Allow for rapid, zero-touch provisioning making it easier to build out new networks in new locations reducing travel requirements.
- Quickly push out changes to many devices simultaneously and allow for rapid roll-back facilitating operational simplicity.
- Help document the network and have constantly updated as-built information about the current network configurations.
- Achieve configuration flexibility, agility, creating a dynamically configured network that adapts to changing application traffic flows.

Network engineer's impressions of the term "network programmability" conjures up thoughts of Software Defined Networking (SDN). SDN is a new paradigm for network configuration as an approach to networking that separates the control plane from the forwarding plane to support virtualization. The controller has global visibility of the network elements and it is the central single-source-of-truth for the network design. Applications can interact with the northbound API (via Python, Java, C, REST, SNMP, etc.) as they request connectivity from the network or signal changes to the network. The controller uses a southbound API (via OpenFlow (OF), Interface to the Routing System (I2RS), Border Gateway Protocol (BGP), Diameter, Radius, NETCONF, Embedded Event Manager (EEM), One Platform Kit (OnePK), etc.) as the controller dynamically configures the network elements.



Network engineers don't want to build their own controller functionality or embark on year-long software development efforts leveraging the northbound API just to make a few tasks a few seconds quicker.

Network engineers can be intimidated by the wide variety of configuration methods and end up in "analysis paralysis" afraid of making a bad programming interface decision. They know that they should be wary of proprietary network programmability protocols, and instead, look for open standard protocols such as OpFlex, Yet Another Next Generation (YANG), Network Configuration Protocol (NETCONF), RESTCONF, and OpenFlow. Network engineers know to look for products that have RESTful APIs using Extensible Markup Language (XML) and JavaScript Object Notation (JSON) and YAML Ain't Markup Language (YAML) configuration file formats.

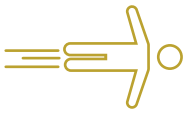
Enterprise network engineers may be intimidated by YAML configuration files and YANG data models. Although, it is more common for the manufacturers to create the YANG models, and the network engineers use a product that uses that configuration definition "behind the scenes". Shielding the network engineer from the complexities of the APIs may actually be the right decision for enterprises.

SOLUTION

The number of software-defined networking solutions available to enterprises is staggering. The choices can be overwhelming making enterprises unable to make a decision about how they should proceed. Zivaro provides recommendations and provide guidance to enterprises to develop a network programmability capability using a phased approach.

Building a Mature Network Automation Capability

Enterprises often don't have any software-defined IT capabilities yet they know that is the direction they would like to take their operations. It isn't feasible to jump right into intent-based networking if the network is a "brown-field" environment that has many underlying technical issues. Zivaro typically recommends enterprises slowly build maturity with network automation using a disciplined approach. It is more realistic to ease into the practice with a phased approach as detailed in the table below.

Phase:	 CRAWL	 WALK	 RUN	 FLY
Network Programmability Tasks	Start learning to create simple scripts to gather information from devices. Assess network team's skills and the network device's programmability capabilities.	Use simple scripts to correct configuration inconsistencies, iterate with configuration checking scripts. Automate a few frequently repeated tasks to gain efficiency.	Use custom scripts, and vendor automation tools to perform deployments. Use RESTful API and template configuration methods.	Automate all new application deployments. Implement intent-based networking, and zero-touch provisioning. Hands off keyboard, no manual changes.

Zivaro helps clients create a plan for network automation, build the frameworks, and implement the tooling to get this accomplished. Our projects use an agile methodology where the requirements will be gathered, tasks will be prioritized, and the maximum value derived for the client as we iterate and help the client build maturity and move through these phases. We recommend enterprises follow Zivaro's "Steps to Success" for building a network programmability capability.

CRAWL

- Understand current network environment present in the enterprise network.
- Review vendor model, brand, and version in place in the networked environment and determine their ability to have their configurations automated with software.
- Review current network management systems used and their configuration management, and configuration automation capabilities.
- Survey the network administration teams and ascertain the tools they are most familiar.
- Create a near-term, mid-term, and long-term architecture and roadmap for network programmability and network device configuration management.
- Create a tactical plan and roadmap for increasing the configuration automation of network devices.

WALK

- Determine DevOps and automation methods used today and determine how best to integrate network configurations into rapid application deployments.
- Develop capabilities to programmatically read configuration settings and telemetry data from network devices.
- Work on building automation of common manual operational tasks that are frequently repeated. Create scripts for automating mundane repetitive tasks.
- Ensure configuration management is being performed on all network devices, archiving configurations, and comparing configuration versions to reveal change history.
- Create scripts that identify inconsistencies in bespoke network device configurations, striving for greater consistency of the entire network.

RUN

- Develop custom scripts, using open-source examples, to read configurations and start to push out changes to network devices.
- Build a repository of useful scripts and code sharing within the enterprise fostering a collaborative network programmability practice.
- Create scripts that can resolve inconsistencies in network devices to reduce complexity and diversity of configurations to adhere to "golden configurations".
- Use vendor-provided network programmability systems to gain efficiencies and achieve the enterprise goals without having to develop software.
- Seek to implement controller-based SDN-like capabilities of global policy configuration.
- Leverage DevOps principles and practices on physical or virtual network devices on premises, in hypervisors, or in cloud infrastructure.
- Start to work with RESTful APIs and use tools that have northbound configuration methods.

FLY

- Create automation scripts that can deploy network and applications simultaneously both on-premises and in cloud infrastructure.
- Let scripts, tools, and software automation configure the network and move away from manual configuration changes. No hands on keyboards; no more manual changes.
- Use Zero-Touch Provisioning methods for new sites and devices. Use controller-based and global policy-based configurations across the enterprise.
- Explore and start to implement intent-based network programmability capabilities leveraging products from established vendors.

CISCO NETWORK PROGRAMMABILITY SOLUTIONS

Network engineers can leverage solutions from their preferred vendor to help them accomplish their goals of network programmability, without having to code it themselves. Engineers can derive the benefits of automation and orchestration using a variety of Cisco management solutions that allow them to build up their network programmability capabilities over time.

Cisco Prime Infrastructure (PI) simplifies operations of the wired and wireless (converged access) network environments. This is a great place to start your network automation journey as PI includes proactive network management and configuration management and guided workflows for speeding up new deployments. PI has a configuration compliance engine that can compare yesterday's configuration with today's configuration, understand the changes and PI uses configuration templates to create "golden configurations" and identify device configurations that don't match.

Cisco Digital Network Architecture Center (DNA Center) is the network configuration automation and management platform that network administrators require. DNA Center provides configuration automation through defining user and device policies and deploy those into the network. DNA Center saves time with integrated workflows facilitating rapid zero-touch device provisioning (with Network Plug and Play), and automated software upgrades (with Software Image Management). Network admins can configure the change and propagate that change across the enterprise network without writing a single line of code or editing any YAML files. The DNA Center dashboard enables proactive and reactive troubleshooting leveraging Artificial Intelligence (AI) and Machine Learning (ML) algorithms that are pre-built into the product. DNA Center can also integrate with PI and import existing network designs and device specifications. The network engineer doesn't need to be a programmer to have the ability to create global configuration changes and have a global view of the network.

Cisco Meraki Dashboard is the software-defined controller for Meraki deployments. It just so happens that the SDN controller, hosted in the cloud, facilitates configuration of multiple Meraki switches, wireless devices, security devices, and more. Most network administrators don't utilize the Meraki Dashboard has a RESTful API using JSON over HTTPS. There are integrations between Meraki devices and DNA Center further strengthening the management of networks that use both tools. The Meraki Dashboard is an example of a network programmability platform that makes it easy for a network administrator to speed up their tasks without having to develop their own code.

The Cisco Application Policy Infrastructure Controller (APIC) is a classic example of an easy-to-use application for rapidly configuring an entire data center fabric of Cisco Application Centric Infrastructure (ACI). The APIC is the highly-available data center network virtualization management and orchestration application that makes it easy to configure an array of data center spine/leaf switches simultaneously. Behind the scenes it uses RESTful APIs and XML to apply policies consistently across the ACI data center fabric. APIC is an example of a time-saving network programmability platform that makes for consistent and rapid deployment.

Cisco Intent-Based Networking (IBN) solutions are available when the enterprise is ready to take next step in their network programmability evolution. Network configurations must keep up with the rate

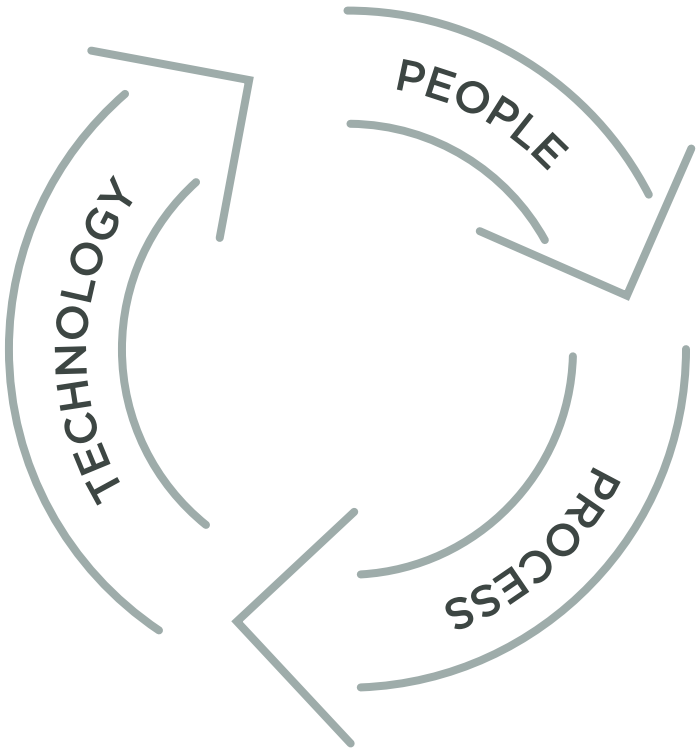
of change while keeping the environment continuously secure. Cisco's IBN systems are where the application access rules are documented in policies that are defined. These policies and rules are then instantiated into granular security configurations into the network devices. Examples of IBN functionality can be found in products like Cisco's Software-Defined Access (SD-Access) that provided end-to-end granular security policies facilitating network segmentation. Other examples of IBN are in Cisco's Software-Defined WAN (SD-WAN) solutions. Policies can be created and pushed to many locations simultaneously. Historically, that would have been a time-consuming endeavor performing remote configurations manually and were risky. If anything went wrong someone had to physically visit the site to remediate.

Cisco Network Services Orchestrator (NSO) is a YANG model-driven network orchestration system that can use network programmability functionality to configure Network Function Virtualization (NFV) via NETCONF across large-scale networks. Service providers, and extremely large enterprises, that need to deploy configurations across thousands of devices can use NSO, rather than spend years developing their own home-grown solution. NSO uses an idempotent algorithm called FASTMAP that pushes only the configuration changes, while maintaining the target ideal desired configuration state for network devices.

There are many solutions to chose as the enterprise continually and methodically builds their network programmability capabilities.

CONCLUSION

We know that to have a successful IT deployment integration of "People, Process, and Technology" is essential.



Investing in training people is key. If network engineers haven't yet began to learn about Cisco DevNet technologies, they can immediately start to obtain training, and consider pursuing DevNet certification exams. It can be easy to start to experiment in a simple physical or virtual lab environment. Using platforms like Cisco Virtual Internet Routing Lab (VIRL), Eve-NG or GNS3 , networkers can build a low-cost virtual lab and start to play with the programmability interfaces and protocols. There are even Cisco DevNet virtual labs that you can use to "kicking the tires" on some of these new programmability methods.

Enterprises can build their network programmability capabilities by looking at their normal operational processes and look for "quick-wins" to ease burdensome tasks and automate them. Organizations should create an internal program of developing network programmability. Creating a continuous-improvement process of increasing the network programmability capabilities and put these methods into common practice will make the process sustainable. Zivaro recommends organizations start small and not try to "boil the ocean" attempting to write a piece of software do everything imaginable. Instead, start with small tasks that you repeat frequently and build up from there as you develop your coding skills.

Organizations can the build up their network automation technology as a similar pace as they are developing their people and processes. They can start to test other Cisco network programmability platforms that remove the heavy lifting and let you do your job easier. These will reduce friction and streamline operational responsibilities as you build your experience. Your confidence will grow and you will feel empowered to take on more adventurous software-driven techniques. Enterprises can purchase network programmability technology at the right time so that it can be immediately effective. Don't buy the technology until you are ready to put it to its full use, to release the maximum benefit to the business resulting from the investment in network programmability products.

Before you know it, Cisco's network programmability solutions will be the jet pack beneath your organization allowing you to reach heights you never imagined.

REFERENCES

Following are additional references to other information for those who want to delve deeper into these topics.

Cisco Cloud Management and Network Programmability Systems:

https://www.cisco.com/c/en/us/products/cloud-systems-management/index.html

Cisco Prime Infrastructure
https://www.cisco.com/c/en/us/products/cloud-systems-management/prime-infrastructure/index.html

Cisco DNA Center
https://www.cisco.com/c/en/us/products/cloud-systems-management/dna-center/index.html

Cisco Application Policy Infrastructure Controller (APIC)
https://www.cisco.com/c/en/us/products/cloud-systems-management/application-policy-infrastructure-controller-apic/index.html

Cisco Intent-Based Networking (IBN)
https://www.cisco.com/c/en/us/solutions/intent-based-networking.html

Cisco Network Services Orchestrator (NSO)
https://www.cisco.com/c/en/us/products/cloud-systems-management/network-services-orchestrator/index.html
https://developer.cisco.com/docs/nso/#!nso-fundamentals

Cisco DevNet:

https://developer.cisco.com/

Cisco DevNet Training and Certification

https://developer.cisco.com/certification/

https://www.cisco.com/c/en/us/training-events/training-certifications/certifications/devnet.html

Network Programmability Books:

Programming and Automating Cisco Networks: A guide to network programmability and automation in the data center, campus, and WAN, By Ryan Tischer, Jason Gooley, Cisco Press, Sept 9, 2016.

https://www.ciscopress.com/store/programming-and-automating-cisco-networks-a-guide-to-9781587144653

Network Programmability Fundamentals LiveLessons, By Kevin Wallace, Cisco Press, Oct 2, 2017.

https://www.ciscopress.com/store/network-programmability-fundamentals-livelessons-9780134840628

Cisco Digital Network Architecture: Intent-based Networking for the Enterprise, By Tim Szigeti, David Zacks, Matthias Falkner, Simone Arena, Cisco Press, Jan 3, 2019.

https://www.ciscopress.com/store/cisco-digital-network-architecture-intent-based-networking-9781587147050

Transforming Campus Networks to Intent-Based Networking, By Pieter-Jan Nefkens, Cisco Press, Dec 20, 2019.

https://www.ciscopress.com/store/transforming-campus-networks-to-intent-based-networking-9780135466339

Cisco Software-Defined Access, By Srilatha Vemula, Jason Gooley, Roddie Hasan, Cisco Press, Aug 2, 2020.

https://www.ciscopress.com/store/cisco-software-defined-access-9780136448389

Cisco Certified DevNet Associate DEVASC 200-901 Official Cert Guide, By Chris Jackson, Jason Gooley, Adrian Iliesiu, Ashutosh Malegaonkar, Cisco Press, Oct 1, 2020.

https://www.ciscopress.com/store/cisco-certified-devnet-associate-devasc-200-901-official-9780136642961

New book coming out in a few months:

Network Programmability and Automation Fundamentals, By Khaled Abuelenain, Jeff Doyle, Anton Karneliuk, Vinit Jain, Cisco Press, May 10, 2021.

https://www.ciscopress.com/store/network-programmability-and-automation-fundamentals-9780135183656