SDN, OpenFlow and NFV Workshop

OpenFlow, Software Defined Networks (SDN) and Network Function Virtualization (NFV) technologies fuel the latest hype bubble in the networking industry and service provider environments.

Based on vendor and industry press promises, well-published OpenFlow deployment with Google's internal network, and numerous other industry initiatives, these technologies became an unavoidable boardroom discussion as service providers and enterprises try to seek new revenue streams or optimize their costs.

On the other hand, many engineers are left wondering what's really going on behind the scenes and how useful these technologies might be in real-life networks.

This workshop describes the technology fundamentals of Software Defined Networking (SDN), OpenFlow and Network Function Virtualization (NFV), their advantages and pitfalls, and the potential use cases including a brief overview of some existing deployments. The focus of the presentation is on real-life deployment scenarios and design discussions.

Topics Covered

- The need for Software Defined Networking (SDN)
- SDN Explained
- Introduction to OpenFlow
- OpenFlow scalability challenges
- Benefits of Network Function Virtualization (NFV)
- BGP-based SDN solutions
- Software Defined WAN (SD-WAN)
- Network programmability with NETCONF and YANG
- Network automation with Ansible
- SDN and controller-based networking deployment considerations
- Real-life SDN use cases

See Workshop Contents below for more details.

Target Audience

Network architects, designers and implementation engineers working in environments that are evaluating the viability or plan to deploy SDN solutions based on OpenFlow, BGP, NFV or other related technologies.

Availability

SDN, OpenFlow and NFV is a 3-day on-site workshop. The workshop can be extended by an extra day dedicated to customer's design challenges or SDN/NFV deployment strategy.

About the Author

Ivan Pepelnjak, CCIE#1354 (Emeritus), has been designing and implementing large-scale service provider and enterprise networks using advanced and emerging technologies since 1990.

Ivan started analyzing OpenFlow-based solutions and writing about OpenFlow technology and SDN concepts in early 2011. He was moderating the first-ever OpenFlow symposium in Silicon Valley in September 2011, had SDN presentations at RIPE and other regional ISP meetings, ran full-day SDN workshops at Interop and Troopers, and created OpenFlow/SDN webinars for NEC, VMware and Nuage Networks.

Ivan published two books on SDN and Openflow in 2014, and helped large multinational organizations and equipment vendors familiarize themselves with SDN concepts, evaluate their SDN strategies, and plan and design SDN pilots. He's also author of several Cisco Press books, prolific blogger at blog.ipspace.net and author of a series of highly successful webinars.

Workshop Contents

The Need for Software Defined Networking

While the whole IT industry has been moving toward highly automated solutions in the last decade, networking has remained stuck – most networking engineers are still manually configuring individual devices.

There's high time we change the deployment and operational processes and reduce the amount of time spent doing repetitive manual tasks; this part of the workshop will give you some high-level guidelines and explore the high-level aspects of SDN including:

- Centralized control plane;
- Software (x86-based) switching;
- Custom network operating systems and whitebox switching;
- Network device APIs;
- Network programmability and orchestration.

The second part of this section focuses on technologies underlying SDN and NFV – OpenFlow, NETCONF, APIs, virtualization and virtual appliances. It will also try to answer the fundamental questions: *When, Why and How should you program your network*.

Software Defined Networking Architectures

Software defined networking is not a new technology – we've been using the concepts of programmable networks for decades.

This section describes common SDN architectures and deployment scenarios including:

- Device and service provisioning;
- Routing and forwarding adjustment controllers;
- Centralized control plane.

Introduction to OpenFlow

This section describes the concepts of OpenFlow, a new protocol used to decouple control plane (topology discovery, path calculation…) from data plane (packet forwarding). It covers the following topics:

- Traditional forwarding with distributed routing protocols
- Controller-based forwarding
- Basics of OpenFlow protocol
- Benefits and drawbacks of OpenFlow

OpenFlow Scalability Challenges

OpenFlow concepts are not new and share scalability challenges with similar technologies and architectures including Frame Relay, ATM, ForCES and MPLS-TP. This section discusses the major OpenFlow scalability challenges:

- Hardware limitations
- Proactive and reactive forwarding table setup
- Hop-by-hop and path-based forwarding
- Control-plane scalability and lack of shared fate

BGP-based SDN

Numerous SDN solutions use BGP as the controller-to-device communication protocol. This section explains the basics of BGP-based SDN, documents several typical use cases and gives practical deployment guidelines, including sample open-source-based controller implementation.

Benefits of Network Function Virtualization

If you open a firewall, load balancer, WAN accelerator or almost any other network services appliance, you'll find one or more x86 processors, standard GE/10GE NICs and some custom packet handling logic. Is there any reason we have to be tied to physical hardware? Wouldn't it be better to deploy the same services in virtual machine format and make them flexible? That's the fundamental concept of Network Function Virtualization.

Does it really make sense to replace physical network services appliances with virtual machines? What are the benefits and drawbacks of NFV approach? This section will give you the answers you need to start evaluating applicability of NFV in your environment.

Network Programmability with NETCONF and YANG

NETCONF is a protocol widely used to configure networking devices (it's supported by Brocade, Cisco, Juniper and other vendors). This section describes NETCONF and YANG (the data model description language used by NETCONF), their benefits and shortcomings, and the vendor-specific implementation details. It includes the following topics:

- What is NETCONF and YANG
- Why are SNMP, CLI and REST not good enough?
- Where did NETCONF and YANG come from?
- How does NETCONF work over XML?
- How does YANG work?
- Why would you write a YANG module? Is it useful?

- I want to deploy a service like MPLS/VPN - are NETCONF and YANG useful?
- Tools you can use to test your NETCONF code
- Differences in NETCONF implementations
- Deployment examples

Network Automation Overview

This section describes typical network automation scenarios, from device provisioning to automated troubleshooting and acceptance tests and guides you on a journey from manually-operated networks of today through network state abstraction toward automated provisioning and failure remediation.

On that journey you'll also identify the common reasons for network automation, meet CLIs and APIs, and learn about typical caveats.

Network Automation Tools

Chef, Puppet, and Ansible are the most popular server configuration management tools, and all of them get used in network automation solutions.

This section describes the fundamentals of all three tools, their typical implementation on network devices, and the potential benefits and drawbacks of using them. It then focuses on Ansible is one of, which is commonly the tool-of-choice due to its agentless design.

SDN and Controller-Based Networking Deployment Considerations

Networking solutions with centralized network intelligence or control plane have existed for almost half a century (IBM SNA, ATM, Frame Relay, Ipsilon Flow Management Protocols).

Not surprisingly, novel SDN architectures using centralized controller clusters exhibit similar challenges:

- Single points of failure;
- Impact of network partitions;
- Balance between tightly- and loosely-coupled elements;
- Control plane and controller security;
- Impact of data plane activity on control-plane performance (punting to control plane);
- Control plane denial of service (DoS) attacks.

This section describes typical SDN deployment considerations, ranging from architectural and design challenges to security and operational considerations.

Real-Life SDN Use Cases

Service providers and enterprises are already deploying SDN, using NETCONF, BGP or OpenFlow as the implementation technology. This section describes numerous use cases based on real-life deployments:

- Data center fabrics (Arista XMPP, Juniper QFabric, NEC ProgrammableFlow, Plexxi, Big Cloud Fabric)
- Data center network automation
- Overlay virtual networks
- Microsegmentation (VMware NSX)
- Forwarding optimizations and exception routing with BGP (Microsoft)
- Optimized WAN edge forwarding (Spotify/Arista)
- Software-defined WAN
- Centralized traffic engineering (Juniper Northstar, Google OpenFlow-based solution, Fibbing)
- Programmable network taps and tap aggregation networks (Arista, NEC, Big Switch, Cisco)
- Network monitoring (Plexxi Control, HP SDN VAN controller, x86-based solutions)
- Network services insertion (NEC ProgrammableFlow, segment routing, virtualization solutions)
- Scale-out load balancing (NEC/Riverbed, Coho Data, Microsoft Azure) and firewalling (Arista/Palo Alto)
- Scale-out intrusion detection system (University of Indiana)
- DoS mitigation tools (Remote-triggered black holes, BGP Flowspec, NEC/Radware)
- Edge policy enforcement