



## White Paper

# DevOps for Network Engineers: The Implications for Network Automation

Prepared by

Caroline Chappell  
Practice Leader, Cloud & NFV, Heavy Reading  
[www.heavyreading.com](http://www.heavyreading.com)

on behalf of



[www.cisco.com](http://www.cisco.com)

June 2016

## Introduction

The automation of network operations is moving up the telco agenda. Operators have lagged other industries in the automation stakes because of their innate caution and the challenges involved in automating the management of a heterogeneous array of boxes accessible only through command line interfaces (CLIs). Now the programmability of the network is improving as more equipment adopts open application programming interfaces (APIs).

Competition over time to market is increasing as leading operators demonstrate the impact automation has on service delivery. Network virtualization is focusing attention on the need for and potentially huge benefits of automation. As a result, operators are increasingly comfortable with the idea of network automation, and are working on strategies to accelerate it.

How operators approach automation matters. Leading-edge operators, such as AT&T with its Domain 2.0 program, want to establish a standardized, agile and ultimately optimized automation culture and process that will apply across the network. They are taking a lean and agile approach to automation, following the collaborative principles of DevOps. DevOps was conceived as a means of creating a best practice automation culture, breaking down silos between the different organizational roles responsible for product/service delivery.

In a telco context, DevOps addresses the relationship between business development, product management, systems architecture and operations. Each role needs to expand its understanding of the others' domains, use common practices and work together in a more hands-on, practical and un-siloed way. For example, DevOps encourages the adoption of prototyping where multi-role teams collaboratively build and test ideas, including network operations issues. This provides faster feedback and ensures rapid alignment between business needs, architecture and practical operational considerations.

When it comes to automating network operations, DevOps puts tools and the responsibility for building automation directly into the hands of those with network knowledge and experience: its engineers. This contrasts with current practices where non-domain experts create automation and throw it back "over the wall" at the end of a slow and expensive waterfall process. This typically doesn't result in satisfied users.

This paper explores the applicability of DevOps to the automation of network operations. It discusses the three aspects of DevOps necessary to its success: creating a DevOps culture, understanding and applying DevOps practices and adopting the right set of tools for automation. It explains the way in which DevOps must be adapted to address network configuration issues, the automation of which differs from IT configuration requirements in key respects. The DevOps toolset must be extended with a service orchestration capability in order to automate short-lived, dynamic configurations to network devices in response to customer-facing service changes.

A growing number of operators are embracing network programmability, encouraging the adoption of an extreme automation mindset, adding coding to the skill-sets of network engineers and turning engineers' craft knowledge into automation scripts. This paper argues that a DevOps capability is increasingly critical to the successful, efficient and profitable delivery of network-based services.

# New Drivers for Network Automation

## The Telco Industry Is an Automation Laggard

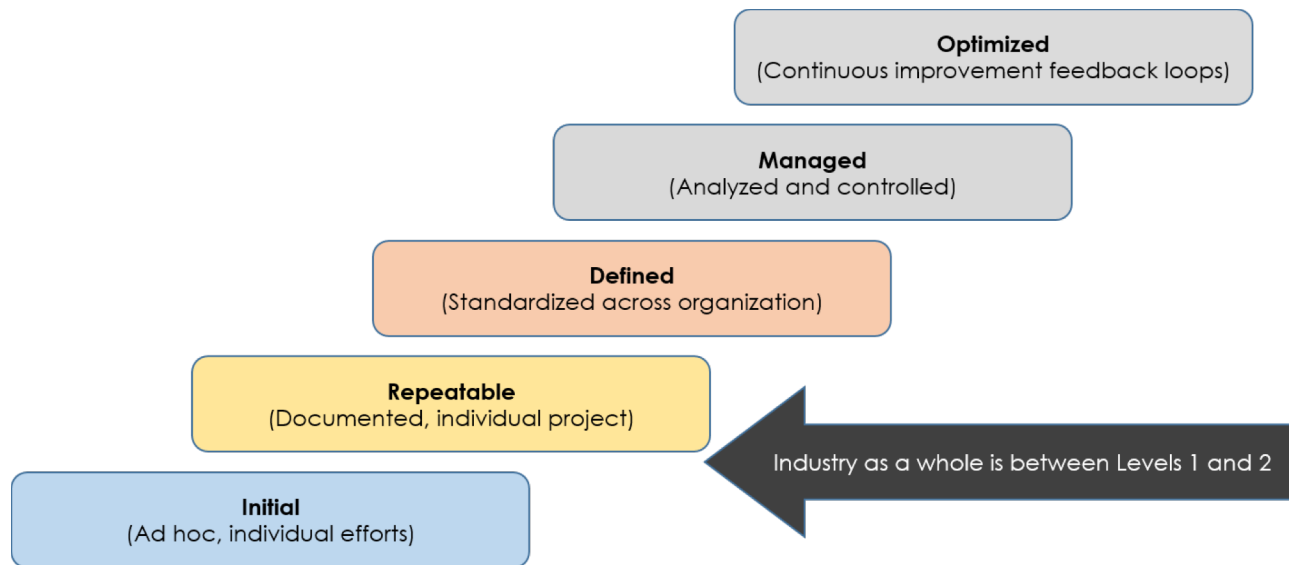
Aerospace, financial and manufacturing industries introduced software-driven, automated operations nearly 30 years ago. Over the past decade, the IT industry has transformed the way it automates infrastructure operations in the cloud. Cloud-native businesses are highly automated, and the Internet of Things (IoT) is spawning a new wave of software-based automation – for example, to support connected cars, remote surgery and smart cities. The world has proved that it can safely apply automation to highly mission critical and complex systems and that automation yields quality benefits, as well as those of speed and lower costs.

Yet the telecom industry has largely resisted automating network operations. There is plenty of software in the network and, arguably, the network has been software-driven for a long time. But it hasn't been easily *programmable*. Network devices are accessed through proprietary CLIs, not open APIs. Network automation, where it does occur, is a difficult and expensive systems integration task typically carried out by IT developers within professional services organizations.

## Network Operations Begin to Embrace Automation

Only relatively recently, and beginning in the data center environment, have network engineers started to demand network programmability. Some engineers distrust automation; others have built their own automation artefacts in an *ad hoc* way, creating scripts and utilities for personal consumption. Few operators have organization-wide network automation programs at Level 3 and above of a network automation maturity model (see **Figure 1**).

**Figure 1: Network Automation Maturity Model**

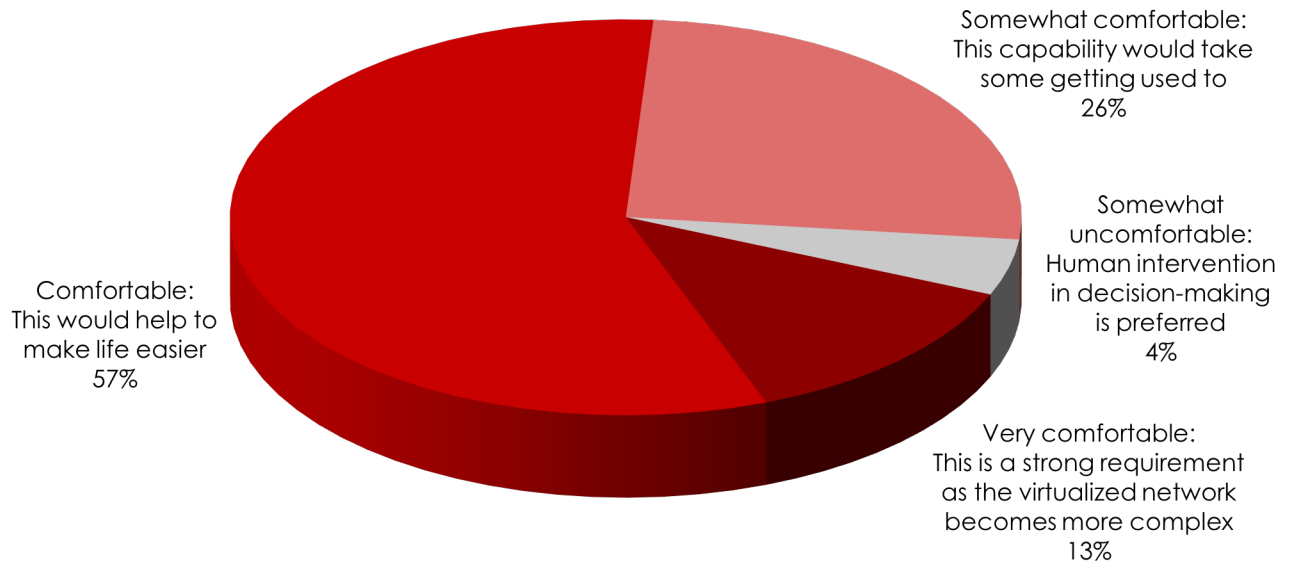


Source: Heavy Reading

But this situation is rapidly changing. Manual operations are severely limiting operators' ability to compete in a cloud-driven world where customers want network services on demand. Operators that have automated delivery of their virtual private network (VPN) and value-added Layer 4-7 services are putting pressure on those still taking weeks to provision manually new connectivity services and service chains. Software-defined networking (SDN) and network functions virtualization (NFV) are driving a new openness into the network, transforming its programmability.

The ability to automate network operations is rapidly becoming a key competitive differentiator for a telco and Heavy Reading research finds operators becoming increasingly comfortable with the idea of network automation (see **Figure 2**). They recognize that engineers interact manually with the network in idiosyncratic ways, so automation results in a more predictable and controllable network. They appreciate the agility that network automation brings, enabling network engineers to respond faster to configuration change requests and assurance issues. Operators also anticipate automation becoming a necessity due to the future complexity and scale of the virtualized network.

**Figure 2: Operator Levels of Comfort With Network Automation**



Source: Heavy Reading Spring 2015 NFV Service Assurance Survey; n=109

# Network Automation & DevOps

## What Is DevOps?

Leading-edge operators, such as AT&T with its Domain 2.0 program, want to move up the maturity curve to establish a standardized, agile and ultimately optimized automation culture and process that will apply across the network. So they are looking at ways of applying DevOps, the extreme automation mindset of cloud developers, to the automation of network operations.

Heavy Reading defines DevOps as: "a culture and practices that support collaboration between those who design software and those responsible for running it to ensure its predictability, reliability, manageability, cost-efficiency and security in a production environment."

In other words, software is designed from the outset *to be managed* – it is no longer "thrown over the wall" to systems administrators who have to figure out how to operate it for themselves. Because an "automate first" mindset is an intrinsic part of the DevOps culture, participants naturally expect new software to be managed *in an automated way*. IT operations staff create upfront the operational checklist(s) and standardized, automated platform(s) in which software applications will live, in conjunction with, and as guidance for, software developers. Developers can, therefore, make frequent changes to applications in the certainty that each change is operationally safe. DevOps de-risks the introduction of new software into a live environment and supports the subsequent maintenance, upgrade and extension of that software. It complements agile development methods with automated operations.

## The DevOps Shift: From OSS-Driven to Network-Driven Automation

DevOps is associated with software development and IT. As networks are increasingly software-driven, DevOps becomes highly applicable in a network context, too. It addresses the relationship between network service designers and the network engineers responsible for applying operational changes to those services to satisfy user and customer demands. DevOps puts responsibility for automating network operations directly into the hands of those with knowledge and experience of the network: its engineers.

Traditionally, IT departments run network automation projects using a waterfall model. Such projects involve non-domain experts (programmers) soliciting requirements from network engineers, creating the automation and then throwing it back "over the wall" at the end of the project. This is typically a slow and expensive process. And because it is implemented by people who lack domain expertise, it doesn't always result in satisfied users.

Operators that re-skill their network engineers in the ways of DevOps can eliminate the systems integration costs and the time it takes to create automation artefacts. DevOps drives automation from the network in an agile fashion using the knowledge of domain experts.

## Three Aspects of DevOps

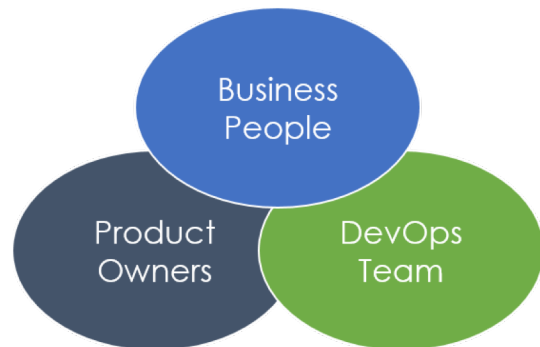
In order to implement DevOps in a network context, operators must understand its three facets:

- **People, process, culture:** DevOps is all about creating among its adherents a certain culture, characterized by: a willingness to collaborate across roles and departments based on business objectives – the antithesis of many companies' dependence on organizational silos and their technical focus; a strong belief in automation; and an open attitude to experimentation that encourages sharing and reuse, rather than the not-invented-here syndrome prevalent today. Creating a DevOps culture is the hardest aspect of its introduction. It requires dedicated programs, such as AT&T's Domain 2.0, and can take several years to roll out across a large telco organization.
- **Practices:** DevOps encompasses a range of practices, such as the ability to treat infrastructure as code, the sharing of tooling by development and operations teams, continuous development of new features and functions and their continuous integration into live code without taking down systems. DevOps practices include rigorous automated testing, release management and deployment processes to de-risk and control such integration and the ability to roll changes back and forward automatically. Advanced automated monitoring and scaling capabilities create data that is used to optimize feedback loops and improve operational quality. De-provisioning is as important as provisioning to streamline resource consumption. DevOps assumes that operations supports a culture of customer/user on-demand self-service so automation is driven from that perspective.
- **Tools:** A huge variety of tools, many of them open and open source, are available to support DevOps practices. Many network engineers are already familiar with such tools and many write scripts in Python, Perl and other languages today, as [a useful blog points out](#). Tooling efforts need to be supported, systematized and brought into alignment with an operator's overall automation goals.

## Introducing DevOps Culture to the Organization

As a collaborative process, DevOps requires the involvement of all the roles engaged in delivering products to customers, including business development, product management, systems architecture and operations. In order to learn from and share with one another, all roles need to expand their understanding of each other's domains, use common practices and gain a knowledge of programming so that they can use common tools.

**Figure 3: Change Across Roles**



Source: Cisco

Every role needs to change:

- Meet in same practices
- Use same tools
- Learn from and educate each other
- More hands-on prototyping

They also need to work together in a more hands-on, practical and un-siloed way. For example, instead of systems architects spending large amounts of time creating PowerPoint architectures, they should adopt a prototyping approach where they collaboratively build and test architectural ideas with network operations. This provides faster feedback and ensures rapid alignment between business needs, architecture and practical operational considerations.

### **Introducing DevOps Culture to Network Operations**

As a rule of thumb, it appears to be easier to train network engineers in DevOps practices and tools than it is for IT operations staff to understand the network and qualify for its automation. Operators such as AT&T are encouraging DevOps adoption through programs that reskill employees, but they can't transform the industry completely alone. They need support from academic institutions, for example: network engineering courses at university level should teach programming and software development courses should encourage an understanding of networking and the impact of the network on application behavior in the cloud. Operators also need their employees on both the IT and networking sides of the house to step up. In the future, telcos are unlikely to employ "pure" IT programmers or network engineers – they will need people with hybrid skills.

To start the journey toward network automation, leading operators are identifying their best engineers with the right mindset for DevOps:

- An existing enthusiasm for, or interest in learning about, business objectives and programming.
- A positive attitude toward collaboration with other disciplines, including IT and business.
- An ability to think in abstractions and to generalize and model the tasks they carry out.
- A willingness to share data models and automation scripts, rather than keeping them as their "secret sauce," and to encourage reuse by others.

Such engineers can become evangelists for DevOps, demonstrating the value of network automation and the high level of job satisfaction that is gained from en-



gaging in it. Many network engineers fear that DevOps, and particularly the introduction of IT into network management processes, will de-skill and displace them. On the contrary, DevOps requires deep levels of knowledge about what to do in the network, in what sequence and why – experience that only network engineers can bring to bear on automation.

## Extending DevOps Practices Into Network Operations

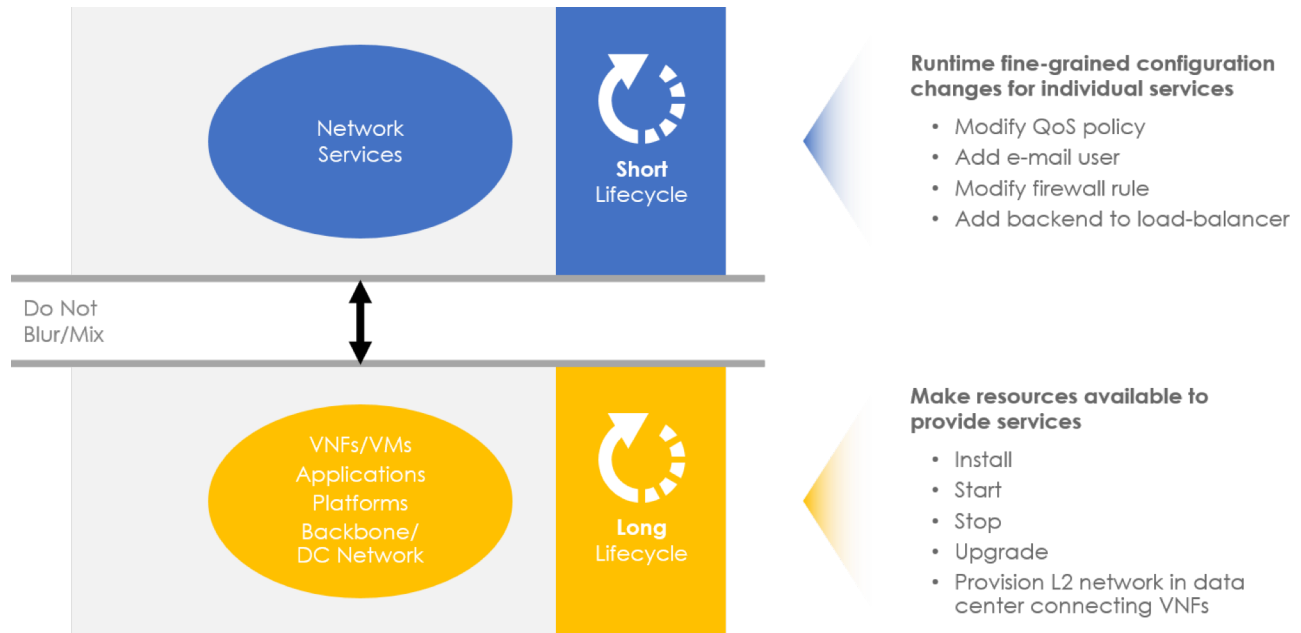
There is a key difference between IT DevOps and DevOps for network engineers. This is due to the nature of the operational changes to *network services* that network engineers need to deal with compared with operational changes to IT applications, the domain of IT DevOps. IT operations staff play two DevOps roles:

- **Supporting the introduction and ongoing maintenance of packaged third-party applications**, such as a MySQL or WordPress server. DevOps focuses here on onboarding such applications, applying patches, upgrades and managing new releases. The frequency of interaction with an individual piece of software is relatively low, for example, weekly, and operations staff automate these interactions using well-known DevOps tools, such as Chef, Puppet, Ansible and Jenkins for testing.
- **Supporting the introduction of in-house developed code**, where the frequency of interactions is much higher, possibly multiple times a day as developers roll out new features. Operations staff may use further tools to develop automated, continuous integration scripts that regression test the interaction between new and existing features.

In both roles, IT operations staff work with application templates that apply relatively long-lived changes to applications, such as install, start, stop and upgrade. Templates – and the tools that produce them – vary in terms of the level of complexity they can apply to these changes. Some templates only express static, "golden" configurations. These are appropriate in the case of applications that are extremely long-lived and require minimal change. Such templates can contain hard-coded variables because of their longevity and they can be stored in a GIT repository from which they can be retrieved over and over again. More sophisticated templates may allow variable substitution and the setting of loops and if-statements, giving DevOps finer-grained control over changes to applications.

The same kinds of templates apply to virtual network functions (VNFs) deployed in a data center: Like applications, VNFs are software packages that need to be installed, started, stopped and upgraded and the same DevOps tools and templates can be used to automate these functions.

**Figure 4: Network Service Configuration & IT Application Configuration Are Different**



Source: Cisco

However, network engineers have a further set of VNF changes to contend with: the extremely fine-grained and frequent configuration changes needed to fulfill new service order and change requests in the network. Here network engineers may deal with thousands of changes to network elements every day. The service-related configuration of a VNF is, therefore, anything but static and long-lived. It may have a Day Zero configuration laid down by a template when it is first deployed, but after this, its configurations are extremely short-lived. They may change every few minutes as new customer-facing services using the element are added, modified and terminated.

At this point, the template approach taken by IT DevOps tools breaks. Service-related configurations, such as the runtime reconfiguration of a virtual firewall, requires a stateful knowledge of the current configuration and a script that talks to an API to change it. Such scripts can't pre-exist in templates: they must be rendered at runtime. Configuring a multi-tenanted network element to support a customer-facing service that is continually changing in an on-demand environment is a step beyond IT DevOps tools and draws heavily on network engineering experience and device knowledge.

**Figure 5: Changes in Different Domains**

Domain	Examples	Changes	Change Rate
Applications	Latest code base for Web shop	In-house developed code Software changes	Daily
IT Platforms	New patch on WordPress server New database server	External software releases and patches	Weekly
	mysql config file	Config file contents	
Network Functions	New VPN Add leg to VPN Add new VAS to VPN Change QoS parameters for VPN	Configuration changes	Thousands per day
	New device OS version	New and upgraded devices	Quarterly

Source: Cisco

### Adding Service Orchestration to the DevOps Armory

Network engineers need to work with DevOps automation tools, but they also need a further automation tool that can apply short-lived, dynamic configurations to network devices in response to customer-facing service changes. This is the remit of a **service orchestration tool** that supports:

- Standardized service and device data models
- Open protocols, such as NETCONF
- State convergence as a means of efficiently automating the runtime configuration of devices

Service orchestration is defined in a previous white paper: **Service Orchestration and Network Virtualization: A Lifecycle View**.

There are two different scenarios for using service orchestration with DevOps tools, depending on whether an operator is only delivering network services or selling IT (cloud) services with which connectivity may be bundled:

- In the first scenario, the service orchestration tool is the main interaction point, orchestrating calls to the DevOps tool(s).
- In the second scenario, DevOps tools are the main point of interaction, calling on the automation within the service orchestration tool only when network service delivery is involved.

Service orchestration fits into the emerging open ecosystem of DevOps tools that operators are starting to use to deploy and manage the lifecycle of VNFs in the cloud. These include Puppet, Chef, Vagrant, CFEngine and Bcfg2 for application management, Make, Jenkins, Maven, CruiseControl and Hudson for continuous

integration and test, GIT for version control and a raft of application and infrastructure monitoring tools, including Nagios, Munin, Zabbix, Sensu, LogStash, CloudWatch, Splunk and New Relic.

### **What Needs to Be Automated?**

The new and necessary drive to automate network operations begs the question of what to automate and where. The basic rule of automation says: if you carry out an action more than once, automate it. The extreme automation mindset associated with cloud automation is summed up in the phrase, "Use things you can program and program things you can use."

Operators must become hard-nosed about automation and programmability in their procurement choices: for example, if one router only comes with a CLI, select another that is API-enabled. They also need to encourage network engineers to write down their manual tasks and script them, whether these pertain to the traditional physical or virtualized network. Much of the knowledge of an operator's network environment lives in the heads of its network engineers and is based on custom and craft: it is never codified and is, therefore, currently inaccessible to automation.

## Conclusion

The speed of the modern business environment and the growth and increasing complexity of the network are putting pressure on operators to automate network operations. Automation increases operator agility and results in a more predictable, reliable network. Telcos can take their inspiration from the extreme automation practiced in the cloud and the DevOps culture and large array of tools this is producing. As the network itself becomes "cloudified," this culture and a DevOps set of tools becomes highly relevant to operators and, specifically, to their network engineers.

However, operators must understand the limits of DevOps tools and the particular challenges involved in automating the network. Automating the hybrid network or a virtualized network with multi-tenanted VNFs used by more than one customer-facing service requires the scripting of runtime configurations, which are very different from the static, template-based configurations created for IT applications. To automate the network fully, operators will need to cultivate a hybrid form of DevOps, which brings the automation capabilities of service orchestration tools into the DevOps mix.

Heavy Reading expects that a growing number of operators will seek to climb up the automation maturity curve as they embrace network programmability and SDN and NFV technologies. They will put in place automation programs that encourage an extreme automation mindset, add coding to the skillsets of network engineers and begin to turn engineers' craft knowledge into repeatable, managed and eventually, optimized, automation scripts. This is a marathon journey that requires organization-wide governance and planning to determine what should be automated and when. Leading-edge operators have already started this journey and are beginning to reap rewards: those who resist risk becoming uncompetitive in their ability to deliver network-based services.

## About Cisco

Cisco (NASDAQ: CSCO) is the worldwide leader in IT that helps companies seize the opportunities of tomorrow by proving that amazing things can happen when you connect the previously unconnected. For ongoing news, please go to [newsroom.cisco.com](http://newsroom.cisco.com).

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third-party trademarks mentioned are the property of their respective owners.