Improving Network Management with Software Defined Networking

Hyojoon Kim and Nick Feamster, Georgia Institute of Technology

ABSTRACT

Network management is challenging. To operate, maintain, and secure a communication network, network operators must grapple with low-level vendor-specific configuration to implement complex high-level network policies. Despite many previous proposals to make networks easier to manage, many solutions to network management problems amount to stop-gap solutions because of the difficulty of changing the underlying infrastructure. The rigidity of the underlying infrastructure presents few possibilities for innovation or improvement, since network devices have generally been closed, proprietary, and vertically integrated. A new paradigm in networking, software defined networking (SDN), advocates separating the data plane and the control plane, making network switches in the data plane simple packet forwarding devices and leaving a logically centralized software program to control the behavior of the entire network. SDN introduces new possibilities for network management and configuration methods. In this article, we identify problems with the current state-of-the-art network configuration and management mechanisms and introduce mechanisms to improve various aspects of network management. We focus on three problems in network management: enabling frequent changes to network conditions and state, providing support for network configuration in a high-level language, and providing better visibility and control over tasks for performing network diagnosis and troubleshooting. The technologies we describe enable network operators to implement a wide range of network policies in a high-level policy language and easily determine sources of performance problems. In addition to the systems themselves, we describe various prototype deployments in campus and home networks that demonstrate how SDN can improve common network management tasks.

INTRODUCTION

Computer networks are dynamic and complex; unsurprisingly, as a result, configuring and managing them continues to be challenging. These networks typically comprise a large number of switches, routers, firewalls, and numerous types of middleboxes with many types of events occurring simultaneously. Network operators are responsible for configuring the network to enforce various high-level policies, and to respond to the wide range of network events (e.g., traffic shifts, intrusions) that may occur. Network configuration remains incredibly difficult because implementing these high-level policies requires specifying them in terms of distributed low-level configuration. Today's networks provide little or no mechanism for automatically responding to the wide range of events that may occur.

Today, network operators must implement increasingly sophisticated policies and complex tasks with a limited and highly constrained set of low-level device configuration commands in a command line interface (CLI) environment. Not only are network policies low-level, they are also not well equipped to react to continually changing network conditions. State-of-the-art network configuration methods can implement a network policy that deals with a single snapshot of the network state. However, network state changes continually, and operators must manually adjust network configuration in response to changing network conditions. Due to this limitation, operators use external tools, or even build ad hoc scripts to dynamically reconfigure network devices when events occur. As a result, configuration changes are frequent and unwieldy, leading to frequent misconfigurations [8].

Network operators need better ways to configure and manage their networks. Unfortunately, today's networks typically involve integration and interconnection of many proprietary, vertically integrated devices. This vertical integration makes it incredibly difficult for operators to specify high-level network-wide policies using current technologies. Innovation in network management has thus been limited to stop-gap techniques and measures, such as tools that analyze low-level configuration to detect errors or otherwise respond to network events. Proprietary software and closed development in network devices by a handful of vendors make it extremely difficult to introduce and deploy new protocols. Incremental "updates" to configuration methods and commands are generally dictated unilaterally by vendors. Meanwhile, operators' requirements for more functionality and increasingly complex network policies continue to expand.

0163-6804/13/$25.00 © 2013 IEEE