Challenges in Unifying Control of Middlebox Traversals and Functionality

Aaron Gember, Theophilus Benson, Aditya Akella
University of Wisconsin – Madison
{agember,tbenson,akella}@cs.wisc.edu

1. INTRODUCTION

Network services appliances, i.e., middleboxes, are a key component of enterprise networks. Through examination and modification of network traffic, middleboxes help ensure security, optimize performance, and facilitate remote access. A diverse array of middleboxes exist, both in terms of functionality and vendor, requiring distinct, distributed configuration across the enterprise [8]. Furthermore, the network must be configured (physically or via routing changes) to direct traffic through the appropriate middleboxes.

Including middleboxes in network topologies has become easier and more flexible with the advent of software defined networking (SDN). SDN enables middleboxes to be placed anywhere within the network while still ensuring that specific subsets of traffic traverse the desired set of middleboxes [3, 6]. SDN is especially useful for middlebox deployments in clouds: tenants and providers can leverage SDN to direct traffic between application and middlebox VMs [4].

While SDN enables control over middlebox traversals, the configuration of the middleboxes themselves remains an out-of-band activity. Each middlebox must be individually configured with the appropriate policies, rulesets, etc. Such distributed, manual configuration, separate from control over traffic forwarding, makes reasoning about and verifying middlebox deployments challenging. Additionally, changes in network topology–which can occur frequently in overlays connecting VMs in the cloud–or changes in the underlying middlebox software/hardware–which can occur when enterprises move services from a local data center to the cloud–requires reconfiguration of middleboxes. These issues are even worse in networks with 100s of middleboxes [8].

We argue that configuration of both middlebox traversals and middlebox functionality should be *unified* under a single *centralized* control plane (Section 2). This (*i*) enables easy verification of objectives, (*ii*) decreases errors due to distributed configurations and topology changes, and (*iii*) permits seamless migration of middleboxes between different underlying substrates (e.g., local data center to cloud).

There are several challenges and trade-offs in designing a centralized unified middlebox control plane (Section 3). First, the examination, modification, and forwarding applied to specific traffic subsets should be specifiable in a flexible, concise manner. Second, the objectives need to be reconciled with the constraints of the underlying infrastructure.

2. MOTIVATION

Today, deploying middleboxes in enterprise networks and clouds requires managing the flow of traffic through middleboxes separately from managing the functionality and policies of the middleboxes themselves. Several problems, especially at large scale, arise from this control strategy:

- **Configurations are topology dependent, and distributed.** Enterprises begin with high-level security and performance objectives, but these objectives rapidly devolve into individual, topology-dependent configurations for each middlebox instance—e.g., one Intrusion Prevention System (IPS) may be configured with a ruleset for protecting web servers, while another IPS is configured to protect internal file servers. Moreover, the traffic subsets reaching these instances are based on completely separate configurations in network elements—e.g., traffic for the web servers may come from any of several links, requiring careful forwarding to ensure the IPS is traversed. Additions/removals of end-hosts (which can occur frequently in elastic cloud environments) or changes in high-level objectives require modifications to configurations across many middleboxes (e.g., several IPSs) and/or networking components (e.g., routers or SDN controllers) to ensure correct behavior. Distributed configurations also make verifying and reasoning about objectives difficult.
- **Configurations are tied to specific infrastructure.** The type, features, and capabilities of a middlebox also impact its configuration—e.g., an IPS with a slow CPU will be configured with smaller rulesets. Changing to an environment with a different piece of middlebox hardware or software means re-writing configurations to adapt to different constraints. Furthermore, configuration syntax differs between middlebox types despite significant commonalities—e.g., a firewall applies forward/drop actions based on packet header fields and an IPS applies examination actions based on header fields. These issues make it difficult to seamlessly move application and middlebox deployments between enter-

prise data centers and clouds, and they constraint administrator and tool flexibility.

These issues motivate the adoption of a *centralized, unified control plane* for managing both the flow of traffic through middleboxes and the functionality/policies implemented by each middlebox. This enables seamless changes in end-hosts, objectives, and infrastructure without errors or deficiencies in examination/modification of traffic by middleboxes. Next, we discuss the challenges in designing such a control plane.

3. CENTRALIZED, UNIFIED CONTROL

Our proposed middlebox control plane manages both the flow of traffic through middleboxes and the policies implemented by the middleboxes themselves. We assume specific types of physical or virtual appliances (e.g., firewalls, IPSs) are deployed in an enterprise network or cloud based on high-level objectives. Furthermore, we assume these middleboxes are connected by a software controlled communication substrate, e.g., a network composed of hardware- or software-based OpenFlow switches with physical wiring or virtual tunnels between them. A control plane that spans these components requires several key design considerations.

Specifying Objectives. Control planes are responsible for coordinating the behavior of nodes to meet high-level objectives. For example, a NOX [5] controller installs appropriate flow rules in OpenFlow switches to meet the basic objective that two end-hosts can communicate. In the case of middleboxes, the objectives are more complex: packets should be *examined, modified, and forwarded* in different ways depending on different *traffic characteristics*. Furthermore, objectives will change based on addition/removal of end-hosts and applications, new enterprise needs, etc.

Capturing these objectives for middlebox deployments is a significant challenge. Proposed frameworks enable specification of either which types of middleboxes specific traffic should traverse [6] or which actions middleboxes should apply [2]. We need to capture both, and connect them, with carefully designed abstractions. Enterprises should be able to define their objectives without regard to the separation between network and middlebox and without consideration of the physical network topology. Unifying all configurations under a single abstraction will enable enterprises to easily manage and verify deployments at large scale.

Existing SDN [1] and middlebox [2] standards provide a starting point for designing such an abstraction. Commonalities already exist between the two for some types of middleboxes: e.g., flow rules in OpenFlow are similar to the rules in a simple stateless firewalls. However, other middleboxes (e.g., WAN optimizers) have significantly more configuration complexity that substantially differs from SDN control messages. We must carefully trade-off support for middlebox diversity with clarity and conciseness of the abstraction.

Considering Underlying Infrastructure. Given the objective specifications, the control plane must appropriately configure the forwarding behavior of each switch and the examination/modification behavior of each middlebox. More importantly, configuration should happen dynamically based on current network topology, middlebox features, and switch and middlebox capabilities.

The challenge lies in appropriately reconciling objectives with the constraints of the underlying infrastructure. Existing works have focused on this challenge for specific middlebox types [7] and for entirely software-based middleboxes [8]. We seek to flexibly coordinate component behavior across a diverse infrastructure (assuming some basic standards support across vendors, e.g., OpenFlow [1] compatibility). This enables enterprises to use physical or virtual network appliances and network components of their choosing, providing the flexibility for migration of applications and middleboxes between local data centers and clouds. Additionally, multiple virtual or physical appliances can be leveraged to meet specific objectives in a distributed way, enabling large scale application deployments.

Flexibility and Extensions. A centralized, unified control plane for managing both middlebox traversals and policies also enables rich new capabilities: A middlebox deployment can be made more fault tolerant by duplicating policies and re-routing traffic in the event of network or middlebox failures. Some network components or middleboxes can be shut down under light load to conserve energy. (This can be problematic for stateful middleboxes, so our control plane needs to be augmented to account for middlebox and network state, and middleboxes need to expose this state via a common interface.) Additional virtual middleboxes could automatically be instantiated to handle transient loads that exceed the capacity of pre-provisioned physical appliances.

In summary, adopting a *centralized, unified control plane* for managing both the flow of traffic through middleboxes and the functionality/policies implemented by each middlebox has many benefits over the distributed, infrastructure-dependent configuration done today. However, achieving this goal requires drawing important connections between middlebox and network configurations and carefully considering the constraints of the underlying infrastructure.

4. REFERENCES

[1] Openflow switch specification 1.1.0. `http://openflow.org/documents/openflow-spec-v1.1.0.pdf`.

[2] RFC 5189:Middlebox Communication (MIDCOM) Protocol Semantics. `http://tools.ietf.org/html/rfc5189`.

[3] M. Casado et al. Ethane: taking control of the enterprise. In *SIGCOMM*, 2007.

[4] A. Gember et al. Stratos: Virtual middleboxes as first-class entities. Technical Report TR1771, University of Wisconsin-Madison, 2012.

[5] N. Gude et al. NOX: towards an operating system for networks. *ACM SIGCOMM Computer Communication Review*, 38(3):105–110, 2008.

[6] D. A. Joseph et al. A policy-aware switching layer for data centers. In *SIGCOMM*, 2008.

[7] V. Sekar et al. Network-wide deployment of intrusion detection and prevetion systems. In *CoNEXT*, 2010.

[8] V. Sekar et al. Design and implementation of a consolidated middlebox architecture. In *NSDI*, 2012.