Demystifying Configuration Challenges and Trade-Offs in Network-based ISP Services

Theophilus Benson, Aditya Akella
University of Wisconsin, Madison
Madison, WI, USA
{tbenson, akella}@cs.wisc.edu

Aman Shaikh
AT&T Labs – Research
Florham Park, NJ, USA
ashaikh@research.att.com

ABSTRACT

ISPs are increasingly offering a variety of network-based services such as VPN, VPLS, VoIP, Virtual-Wire and DDoS protection. Although both enterprise and residential networks are rapidly adopting these services, there is little systematic work on the design challenges and trade-offs ISPs face in providing them. The goal of our paper is to understand the complexity underlying the layer-3 design of services and to highlight potential factors that hinder their introduction, evolution and management. Using daily snapshots of configuration and device metadata collected from a tier-1 ISP, we examine the logical dependencies and special cases in device configurations for five different network-based services. We find: (1) the design of the core data-plane is usually service-agnostic and simple, but the control-planes for different services become more complex as services evolve; (2) more crucially, the configuration at the service edge inevitably becomes more complex over time, potentially hindering key management issues such as service upgrades and troubleshooting; and (3) there are key service-specific issues that also contribute significantly to the overall design complexity. Thus, the high prevalent complexity could impede the adoption and growth of network-based services. We show initial evidence that some of the complexity can be mitigated systematically.

Categories and Subject Descriptors

C.2.3 [**COMPUTER-COMMUNICATION NETWORKS**]: Network Operations - Network management

General Terms

Design, Management, Measurement

Keywords

Network services, network modeling, configuration analysis

1. INTRODUCTION

Conventional ISP operations have historically focused mainly on providing Internet access to customers and ensuring global connectivity. In recent years, however, ISPs' focus has shifted toward providing large scale *network-based services*. These cater to groups of customers who need functions beyond best effort connectivity, such as virtual links or networks at layers 2 and 3, security, and quality-of-service. These services require the introduction of new layer-2/3 devices and/or significant modifications to the existing layer-2/3 setup. Customers of these services typically have fairly strict performance and availability requirements, which means ISPs have to enable the requisite support within their networks. Examples of network-based services include IPTV, Virtual-Wire (V-Wire), VPNs, VPLS, VoIP, DDoS protection, and teleconferencing.

The Internet is poised to experience rapid growth in the number of network-based services in the near future. There are several factors contributing to this. First, with growing traffic demands [1], merely providing IP connectivity is no longer a viable business model for ISPs; network-based services are seen as a crucial means to earn extra revenue. Second, router and switch vendors are responding to ISPs' desire to offer services by making their platforms virtualizable (e.g., through frameworks such as VRF [2, 3]) to ease creation of interesting new services. Finally, there is a growing demand for such services from both residential and enterprise customers. However, despite the growing importance and centrality of network-based services, few studies have examined the challenges ISPs face in offering them.

In this paper, we present a first-ever large-scale analysis of the *complexity* underlying the layer-3 design of a variety of network-based services in a tier-1 ISP. The central design issue in network-based services that motivates our work is how to make ISP networks *easier to evolve* to meet the growing demand for services and changing customer requirements. To a large extent, this hinges on the ease with which the infrastructure can be upgraded or reconfigured with new functionality.

Specifically, the focus of this paper is on the complexity arising from *device-level configurations*, which define the services and define how the devices are integrated into the network. We use configuration models to abstract away details of device configuration files and examine the difficulty in setting configuration *dependencies* (e.g., in setting up service-specific functionality on a router) and in creating the requisite *special cases* (e.g., to meet design constraints and differing customer requirements), which our prior work [7] shows to be indicative of the complexity in managing the underlying design and configuration. Based on our analysis, we then identify how to evaluate trade-offs in simplifying service designs to make them easier to evolve and manage.

There are several steps in creating and running a service over time, e.g., addition of layer-2/3 devices that have customer-, location- and service-specific configuration, integrating with the core, setting up the control-plane (e.g., routing between different sites of

SIGCOMM'11, August 15–19, 2011, Toronto, Ontario, Canada.

a VPN customer), and, occasionally, restructuring the network to achieve routing scalability as services expand. Using our configuration models, we analyze configuration snapshots collected from a large tier-1 ISP network over a period of several years. We study *five* services in all: VPN, VPLS, DDoS protection, Virtual Wire service and VoIP. We analyze the relative difficulty of all the above tasks, both within and across these services, and identifying what contributes most to complexity and why.

The key high-level findings from our analysis are as follows: (1) The design of the ISP core data-plane is usually service-agnostic and it appears simple to integrate new service infrastructures into it. Unfortuntely, the good news ends here. In particular, we find that the design of service control-planes becomes more complex over time with growing service size. A key cause for this is the complexity underlying the establishment of BGP sessions. (2) More crucially, the configuration of a service at the network edge could rapidly become more complex over time, which could severely hinder key management issues such as service upgrades, adding customers, and troubleshooting. We find that a central factor in the growing complexity is the diversity of customer-provisioning requirements and the changes that occur to them over time. (3) There are several service-specific issues that also contribute significantly to the overall design complexity (e.g., introduction of a new vendor and implementation of new policies); thus, it appears that service-specific mechanisms may be fundamentally unavoidable in managing service design and configuration.

Thus, it appears that the high prevalent complexity may impede the growth and adoption of services. A natural question is whether it is possible to *control or mitigate* the complexity. This issue cannot be addressed in isolation as the complexity depends on other key considerations including the choice of available vendors, cost, performance and resource constraints. In this paper, we take initial steps in this direction and show how to systematically control or mitigate (where possible) the complexity in customer provisioning and service control-plane design by making an informed choice of vendors and by picking appropriate routing substrates.

The rest of the paper is structured as follows. Section 2 provides an overview of network-based services and their configuration in ISPs. Section 3 describes the ISP network analyzed and presents the configuration models used. Section 4 presents our findings regarding the complexity of service designs. Section 5 analyzes and explains some of the key causes of the complexity. Section 6 shows how it may be possible to reduce such complexity by using alternate designs. Section 7 presents implications of our results, and discusses limitations of our study. Finally, Section 8 presents related work, and Section 9 concludes.

2. BACKGROUND

Our work focuses on the network services that are implemented within, and require support from, the provider's router and switch infrastructure. We refer to such services as being *network-based*. Examples include VPN, VPLS, VoIP and DDoS protection. Our work does not extend to other key services such as CDNs, cloud services, data center and hosting services which ISPs offer by placing racks of servers in strategic locations within their network.

We focus specifically on the configuration of devices required to run network-based services. Device configurations define both the service semantics, i.e., how customers can use a service, and operational aspects, such as how a service integrates with the rest of the network. Thus, configuration is central to the correct and effective functioning of a service and the underlying network. Furthermore, device configuration is perhaps one of the most time-consuming and error-prone tasks in ISP operations [10], often requiring significant manual intervention [8]. We discuss other factors relevant to services such as configuration of customer-premise routers that connect to the ISP in Section 7.

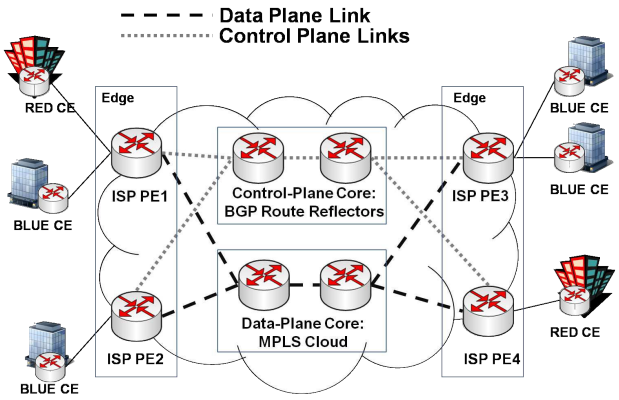


Figure 1: Architecture for a VPN service in an ISP.

Network-based services are realized through a combination of changes to layer-3 routing protocols and the deployment of new devices capable of supporting service-specific functionality. In this section, we present a brief overview of the roles played by various devices in supporting services, and then discuss how service introduction impacts the configuration of protocols within these devices. We also describe configuration steps required to provision a customer of a typical service.

2.1 ISP Network

In Figure 1, we show a typical tier-1 ISP network. The network devices are divided into the data-plane core, the control-plane core and the edge. The edge consists of a variety of specializable devices, known as *Provider Edge (PE)* devices. Each PE is configured to provide one or more services to the customers connected to it; customer devices that connect to PEs are called *Customer Edge (CE)* devices. For example, in Figure 1, we present an edge consisting of VPN PE devices which are used for creating virtual links and networks between the customer's CE devices. The data-plane core is a cloud of service-agnostic devices that provide transit for packets across the network in a highly efficient manner. The control-plane core consists of a set of BGP *route-reflectors (RRs)* [5] used for exachanging reachability information between Provider Edge devices (PEs).

2.2 Description of Services

As mentioned earlier, ISPs have traditionally provided Internet connectivity to business and residential customers. ISPs deploy networks consisting of routers, switches and links interconnecting them. These networks allow ISPs to provide other services to their customers. Two such services are VPN (Virtual Private Networks) and VPLS (Virtual Private LAN Service). These services are provided to enterprise customers, where an ISP uses its network to interconnect each customer's geographically distributed sites, thereby allowing the customer to form its own private intranet. With VPN, each customer gets a private IP (i.e., layer-3) network, allowing its sites to exchange IP packets amongst themselves. VPLS, on the other hand, provides a private ethernet (i.e., layer-2) network to the customer, thereby allowing its sites to exchange ethernet frames with one another. The implications of keeping these networks private are three-fold. First, the ISP needs to keep traffic of one customer separate from another customer. Second, the ISP needs to allow every customer to choose its own control-

plane mechanisms to route packets across its sites. Finally, each customer needs to have freedom in the way addresses are assigned to its devices even if this means overlap of addresses across different customers. For example, the red and blue customers might end up assigning the same addresses to their hosts in the Figure 1.

To satisfy these requirements, ISPs essentially create a "virtual slice" of their network for each customer. For the example shown in Figure 1, the ISP creates two virtual slices, one for the red and another for the blue customers. Since the underlying network infrastructure is shared amongst these slices, an ISP needs several capabilities to keep the slices separate from one another. The first capability allows the ISP to form tunnels between PEs attached to the same customer. This in turn allows the ISP to maintain traffic separation even when addresses overlap between different customers. Although various tunneling mechanisms are available, most tier-1 ISPs have converged on the use of Multiprotocol Label Switching (MPLS) and Label Switched Path (LSP) as the tunneling technology.

The second capability allows ISPs to use (I-)BGP and an infrastructure of BGP route-reflectors for exchanging routes between sites attached to a virtual slice. This is similar to what ISPs do for providing Internet connectivity. Unlike with Internet connectivity though, the routes do not always carry IPv4 addresses (e.g., VPLS uses MAC addresses). Therefore, a Multi-protocol version of BGP, known as MP-BGP [4], is used. MP-BGP allows BGP to carry routes for various kinds of address families.

Finally, an ISP needs to keep routes of one slice separate from other slices especially when addresses can overlap. To achieve this, PEs need the capability to keep slice-specific routes in their own separate routing tables. This is achieved by creating virtual routing tables on PEs using *Virtual Routing and Forwarding (VRF)* technology, such that there is one VRF for every customer connected to a given PE. For example, PE1 in Figure 1 has VRFs for blue and red customers, whereas PE2 only carries a single VRF for the blue customer. MP-BGP stitches the VRFs together by allowing VRFs of a given customer which could be strewn across multiple PEs to exchange routes amongst themselves. MP-BGP also ensures that routes in the VRFs of a given customer do not leak into another customer's VRFs.

In addition to providing "basic connectivity", the VPN and VPLS services often allow customers to segregate traffic into multiple classes for different treatment across the core (e.g., VoIP calls between customer sites would be given higher priority over e-mail traffic). This capability is known as *Class of Service (CoS)* routing. Supporing CoS requires the ability to classify and mark packets at PEs, and providing appropriate treatment (via queuing mechanisms) to each traffic class within the core.

2.2.1 Configuration

Configuring VRF: At the edge of the network, VRFs are configured on the Provider Edge (PE) devices to interface with the customer site and create the virtual network. In Figure 2, we present the configuration commands used for such a PE device. The VRF is configured in lines 1-4. The virtual network for a customer is defined as a set of VRFs that import the same set of routes. In a simple setup, this is achieved by setting the same "route-target" on all VRFs as is done in lines 3-4. More complex setups, such as hub-and-spokes [22, 14], may be achieved by using an identical set of "route-targets" on the spokes and using multiple different "route-targets" or "route-maps" on the hub. While these setups are more complex to configure, they provide the ISP with the benefit of reducing the number of routes installed by a VRF into a PE's forwarding table. We discuss these trade-offs in section 6.2. Lines

```
    ! Configuration for the VRF for a customer's virtual network.
 1  ip vrf blue
 2  rd 23234:100223
 3  route-target import 1000:1
 4  route-target export 1000:1
    !
    ! Configuration for a customer facing interface.
 5  interface ethernet1
 6  ip address 128.105.82.66 255.255.255.252
 7  ip vrf forwarding blue
 8  service-policy output policy1
 9  service-policy input policy2
    !
    ! Configuration for the CoS for a customer's virtual network.
10  policy-map policy1
11  class class1
12  police 10000 200 confirm-action transmit exceed-action drop
    !
13  class-map match-any class1
14  match mpls experimental 6
    !
    ! Configruation for BGP peering sessions with route-reflector.
15  router bgp 65000 ıconfiguration for BGP
16  router-id 192.168.6.6
17  neighbor 192.168.2.1 remote-as 1
18  neighbor 192.168.2.1 update-source Loopback0
19  !
20  address-family vpnv4
21  neighbor 192.168.2.1 activate
22  neighbor 192.168.2.1 send-community extended
23  exit-address-family
```

Figure 2: Configuration for a VPN PE.

5-9 present the configuration for the customer-facing interface on the PE. Specifically, the customer's interface is attached to the VRF in line 7.

Configuring MPLS: As shown in Figure 1, the PE devices are interconnected by a cloud of core devices. In this architecture, the PEs act as MPLS edge devices which perform classification of packets and add/remove LSP labels from packets based on this classification. The PEs add two labels to every packet: an outer label to be used by the core devices to switch to the destination PE, and an inner label for the destination PE to select the appropriate VRF. The core devices, acting as label switch routers, switch packets based on the outer label contained in their headers.

Configuring BGP: Reachability information between different PEs is exchanged using MP-BGP [4] as mentioned earlier. Due to the size of the ISP networks, PEs are configured to exchange reachability information indirectly through the use of BGP route-reflectors. In lines 15-23 of Figure 2, we present the configurations for the PE to communicate with a BGP route-reflector. This involves setting up neighbor commands (for BGP peering sessions) with the appropriate route-reflectors (in lines 20-23).

In Figure 3, we present the configuration for the corresponding route-reflector. The route-reflector contains neighbor commands for peering sessions to each of the other route-reflectors (lines 5-6) and its clients (lines 3-4). Commands in lines 7-13 then activate use of MP-BGP for VPN addresses for these neighbors. Each route-reflector should have N such set of neighboring commands, where N is the number of route-reflectors and the PEs taken together.

Configuring CoS: The Class-of-Service primitives include policy maps (Figure 2, lines 10-12) and class maps (lines 13-14) which limit the amount of traffic that an interface of the virtual network allows to enter or leave the network (the interface itself is configured to use these in lines 8-9).

```
1  router bgp 65000
2  router-id 192.168.2.1
3  neighbor 192.168.6.6 remote-as 1
4  neighbor 192.168.6.6 update-source Loopback0
5  neighbor 192.168.2.2 remote-as 1
6  neighbor 192.168.2.2 update-source Loopback0
   !
7  address-family vpnv4
8  neighbor 192.168.6.6 activate
9  neighbor 192.168.6.6 send-community extended
10 neighbor 192.168.6.6 route-reflector-client
11 neighbor 192.168.2.2 activate
12 neighbor 192.168.2.2 send-community extended
13 exit-address-family
```

Figure 3: BGP configuration for a route-reflector.

2.3 Customer Provisioning

A customer of a network-based service is set up within this framework by (1) physically connecting the customer site to one or more PE devices and setting up customer-facing interfaces (Figure 2, lines 5-9); (2) configuring a VRF instance (Figure 2, lines 1-4) on each of the connected PE devices; and (3) setting up the appropriate CoS primitives (Figure 2, lines 10-14). Different customers have different CoS requirements and thus different types of configuration. For example, some customers may not have any CoS polices in which case lines 8 and 9 are not required, while others may have more complex policies requiring the policy map in lines 10-12 to be significantly larger and to use multiple different class maps (similar to those in lines 13-14).

3. APPROACH

In this section, we describe the data-set and configuration models we employ to study the complexity underlying the layer-3 design of network-based services.

3.1 Data

From a tier-1 ISP network, we gathered *daily snapshots* of device configurations as well as documentation about service deployment, installation and upgrades over a *30 month* period from June 2008 to December 2010. We also obtained metadata regarding device role (e.g., core or edge), associated services (e.g., VPLS or VPN), device vendor, and OS version.

We studied a total of **five** network-based services in this ISP: two of these – **VPN** and **VPLS** – were described in the previous section. Next, let us describe the remaining three.

The third service is **DDoS protection**, where the ISP offers mitigation to Denial of Service (DoS) attacks as a value-added service to enterprise customers that have subscribed to the Internet services. DDoS attacks are very prevalent in the Internet today. To mitigate such attacks, when the ISP detects an attack against a customer, it re-directs the customer's traffic to a "scrubbing farm". At the scrubbing farm the attack traffic is segregated from the normal traffic, and latter is sent back to the customer. Re-direction of the traffic to the scrubbing farm is achieved via a temporary BGP route announcement. The cleaned normal traffic is sent back over a dedicated VPN.

The fourth service, which we term **Virtual Wire**, is offered atop the VPLS infrastructure. It provides a customer with an "access link" to a PE offering layer-3 services like VPN and Internet access, and is realized by creating a point-to-point VPLS network – essentially a Virtual Wire – between the customer CE and the ISP's layer-3 PE.

The final service is a specific instantiation of VoIP targeted for the ISP's customers; we refer to this simply as **VoIP**. Traffic belonging to this service is routed between customer sites over a dedicated VPN.

Service	% of ISP Devices Used by the Service
VPN	48%
VPLS	27%
DDoS	31%
Virtual Wire (V-Wire)	25%
VoIP	5%

Table 1: Percentage of ISP devices used by the five services studied. The devices used by a service include both the PEs and the RRs, but not the MPLS core devices since the core devices are shared by all services.

We should mention that the DDoS protection, Virtual Wire, and VoIP services do not require the entire set of mechanisms and configuration steps outlined above, as these services are akin to customers of VPN and VPLS services. In particular, no special route-reflectors, MP-BGP setup or MPLS support is needed. However, they do require a group of VRFs with service specific configuration; the configuration of the VRFs may change over time to accommodate new customers of these services.

The services we study are quite diverse in terms of customer requirements, life-cycle, scale, and devices. The latter two aspects are captured in Table 1 where we show the percentage of overall devices used by each service. Note that the total of the first column exceeds 100% since a good fraction of devices are shared across services. Overall, we believe that this diversity offers a comprehensive view of network-based services.

3.2 Configuration Models

In order to understand the challenges of configuring services in ISP networks, we employ *configuration models*. These models allow us to abstract away the details of the underlying configuration files, while allowing us to systematically reason about the relative difficulty involved in various tasks such as configuring PE devices, core devices and control-plane mechanisms.

The models we use rely on first identifying *stanzas*, i.e., logical groupings of a set of configuration lines based on the functionality they directly define. For example, in Figure 2 lines 5-9, which specify properties for an interface are all grouped together into a single stanza. Various configuration languages provide different demarcators that can be used to identify stanzas.

Our models focus on two aspects of configuration: *creating and maintaining dependencies between stanzas*, and *creating special stanzas to cater to specific needs*.

Referential Graph. Since configuration files are modular in nature, when configuring a device, operators are often required to create references among different stanzas. For example, enabling a routing process on a router usually requires that the stanza defining the routing process refer to stanzas for the interfaces over which routing sessions are established. To model this dependency, we extract the referential graph for a device. Each node in the graph represents a configuration stanza. The edges between the nodes represent explicitly defined links between stanzas. To calculate the referential graph for a device, we developed a parser that reads through the configuration files, and determines stanzas and their names. Once all stanzas and their names are extracted, the parser re-processes the configuration file, creating links between stanzas everytime one stanza has a command explicitly referencing the name of another stanza.

The referential graph models the syntactic dependencies that operators must track in configuring services with various devices. We choose this model since, as prior work has shown, the higher the

number of referential links that must be established within the subgraph of the referential graph corresponding to a functionality, the greater the difficulty of setting up the functionality [7]. Referential graphs help us understand both how difficult it is to introduce a service and to modify it over time.

Templates. Templates are often used in configuring similar functions across devices of an ISP network [8]. Our template model identifies groups of stanzas in various devices with similar configurations (meaning that the stanzas could be configured using a single configuration template). Thus, templates and the devices that constitute them help track *uniformity* in configuration; the greater the non-uniformity, the more special cases there are in the design and the more difficult it is to set up and maintain a service [7]. We use copy-paste analysis tools [13] to derive sets of configuration stanzas that are identical across devices, reflecting functionality that is replicated across devices. We set the grammar and parameters of the analysis tool such that a group of stanzas identical in size and consisting of the same commands form a template. Using templates, we are able to understand the ease of provisioning a new PE for an existing or a new service as well as modifying or updating the functions performed by existing PEs.

4. CONFIGURING SERVICES

In this section, we analyze the complexity of the layer-3 design and configuration for the five network-based services. The more involved the service configuration is, the greater the amount of manual intervention and time required to instantiate, modify or update it over time, and the more error-prone it gets. In particular, we study the relative amounts of complexity contributed by three aspects that make up a service: (1) PE configuration, (2) PE-Core configuration, and (3) control-plane configuration. For each aspect, we provide a longitudinal analysis of how configuration difficulty changes with time and what contributes to the observed complexity. Also, while prior works have examined services in isolation, we provide a comparative analysis and examine reasons for the differences observed across services. We apply the models described in the previous section to study various services.

4.1 PE Configuration

We start by analyzing the configuration employed for setting up PEs for each service.

4.1.1 Referential Dependence

We use the referential graph model to quantify the dependencies within PE configurations, with a greater number of dependencies as an indication of greater configuration complexity.

In Figure 4 we present various high-level properties of the referential graphs for the five services we studied. From Figure 4 (a), we observe that the ratio of the aggregate referential graphs - in terms of number of referential links computed over all PEs on which the service is configured at any point relative to the inital number at the beginning of the study - increases monotonically for each service. This indicates that, in general, configurations of the services require more dependencies to be tracked as they evolve. In Figure 4 (b), we examine the worst referential graph size for each service; we see that this too steadily increases over time. In the case of VPLS, it increases by a factor of 2 over a 12-month timespan.

In Figure 4 (c), we examine the referential graph size for the median PE device for each service. We find that the median referential graph size for VPN and VPLS is not strictly monotonic. Figure 4 (d) shows the corresponding timeline for the number of PEs in each service. From this, we note that the drops in the referential graph sizes in Figure 4 (c) correspond to time instances when new PEs were added for the services: unlike existing devices, new devices do not have as many customers, and as such contain significantly simpler configurations. For other services, the median PE's referential graph increases in size over time.

We now examine the causes for the growing number of referential links in the services (Figures 4 (a) and (b)). To do this, in Figure 5, we present the relative numbers of various types of stanzas added within the overall referential graph of a service over time. Stanzas corresponding to the CoS include policers, filters, class maps and policy maps, while stanzas corresponding to the route control stanzas include route-maps, policy statements and community lists.

We make the following observations:

• For VPN (Figure 5 (a)) and VPLS (Figure 5 (b)), the changes in the referential graph sizes can be attributed to customer provisioning. With VPLS, this can be seen in the growing number of VRF stanzas – each VRF stanza indicates a different new customer site. In the case of VPN, the changes are due to a growing number of interfaces (whereas the proportion of VRF stays the same). This means existing customers are adding more sites to the same PEs.

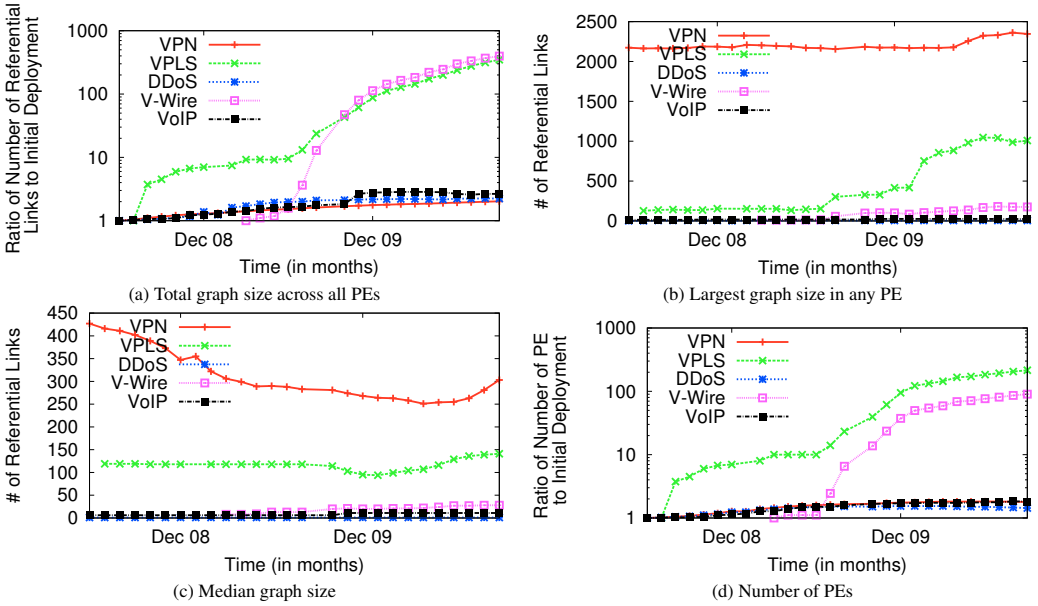
• In the case of the DDoS protection service (Figure 5 (c)), the growing complexity was because the ISP started deploying PE routers from another vendor. The DDoS-related configuration on these PEs requires multiple stanzas since the configuration language of the vendor is more structured and modular. In contrast, the configuration language of the original vendor allows specification of multiple options and parameters within a single stanza, resulting in fewer stanzas and less complex configuration on its PEs to achieve the same purpose.

• For the Virtual Wire service (Figure 5 (d)), the somewhat fluctuating pattern for the number of interfaces and other stanzas is due to device provisioning over time and due to the semantics of the service. Recall that the Virtual Wire service provides the abstraction of a point-to-point access link to a layer-3 PE. When a new VPLS PE is provisioned for this service, it contains configuration for setting up the basic abstraction, i.e. VRF and route control stanzas, – this is reflected in the higher proportion of VRF and route control stanzas in the initial months. When a new customer is provisioned, the ISP configures an interface for physical connectivity to the customer and then configures the virtual network to include this interface– this is reflected in the higher proportion of interface stanzas at later times.

• For VoIP (Figure 5 (e)), we find that over time more policy maps and class maps are being used within the service to configure additional CoS constraints (in fact, the same policy maps and class maps are being uniformly applied across the PEs).

We also note that despite different services using the same basic router functionality (i.e., VRF and MP-BGP), Figure 5 shows that the relative composition of different stanzas in the devices of various services appear to be quite different. This, along with the observations above, indicates that mechanisms designed to manage the configuration for one service may not directly apply to other services. Service-specific optimizations and solutions may be necessary.

To summarize, we find that the aggregate and worst-case sizes for referential dependency graphs grow for PEs of all services. The implication of the growing complexity is that some configuration-related management tasks, such as updating functionality on PEs, or even adding new PEs may become tricky over time. There are a variety of different reasons for the growing complexity, several of which are service-specific. However, one common culprit across many of the services, is customer provisioning. Finally, the existence of key differences among service requirements means that



(a) Total graph size across all PEs

(b) Largest graph size in any PE

(c) Median graph size

(d) Number of PEs

Figure 4: Longitudinal analysis of the referential graphs in service PE devices.

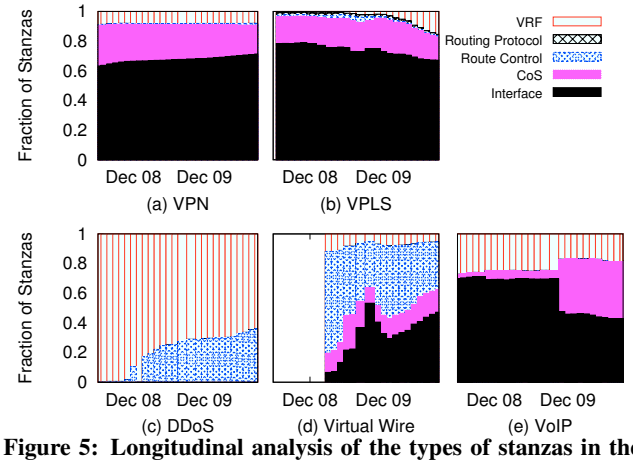


Figure 5: Longitudinal analysis of the types of stanzas in the referential graph (averaged over PEs).

one would require service-specific approaches for managing the complexity in designing and configuring services.

4.1.2 *Templates*

Next, we use templates to quantify the uniformity in the configuration between the devices being used for a particular service. Recall that greater uniformity (i.e., few templates and relatively even spread of templates across PEs) leads to simplicity in service configuration and ease in modifying or evolving it.

In Figure 6 (a), we show the number of templates across devices over time for various services. We observe that the number of templates in VPN-based services (i.e., VPN, DDoS and VoIP) grows more slowly over time compared to the VPLS-based services (i.e., VPLS and Virtual Wire). This can be explained by the fact that VPN and services based on it have been around longer than the VPLS-based services, and hence have had more time to "settle down".

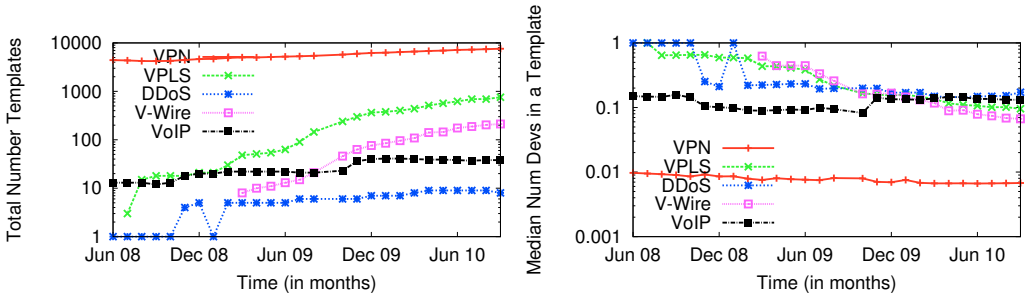
Figure 6 (b) presents the median number of devices that a template is copied on as a fraction of the total number of PE devices for the service. For both sets of services, as the number of templates rise, we observe a *decrease* in the median number of devices that share a given template, as one would expect.

Next we examine the cause for changes over time in the stanzas that make up templates and in the devices on which templates appear. We consider four types of changes: (1) *no change*: the template contains the same set of stanzas and devices, (2) *grow*: the stanzas within the template are replicated over more devices; (3) *new templates*: new stanzas are introduced to existing or new devices to form a new template, (4) *mutate*: exiting stanzas within the template are further specialized leading to a fragmentation of a template. For example, to add more customer sites to a customer's VPN, the VPN's VRF template is modified by adding configuration commands to specify the new interfaces. Note, while 'mutate' and 'new template' both result in essentially new templates, we differentiate between the two to highlight the number of new templates created by changing existing stanzas. Our observations are shown in Figure 7.

Across all the services, a significant fraction of templates remain unchanged over time. However, we note that significant template changes occur as well. Consider VPN whose relatively stable customer base explains the dominance of 'no change' templates. However, examining the other three categories shows that, over time, a large fraction (30-70%) of templates arise due to flux in the existing set of templates.

Upon closer examination, we find that tasks related to *provisioning of customers* (new or existing) contribute most to the flux in templates. Specifically, we find that 'new' templates are almost always for new customers' configurations which require specialization based on customers' locations; 'grow' corresponds to templates for shared functionality that gets replicated as new PE devices are provisioned (an example is configuration of management functions); finally, changes appear to be occurring in the requirements of existing customers over time, which contributes both to 'grow' and 'mutate' (most of these changes appear to be for customer CoS policies).

For the VPN service, the relative proportion of the four types of template changes remains mostly the same across time. However, the relative proportions change a lot over time for the VPLS and Virtual Wire services, and they reflect the service's evolution. Ini-



(a) Number of templates

(b) Number of instances of each template

Figure 6: Longitudinal analysis of configuration templates.

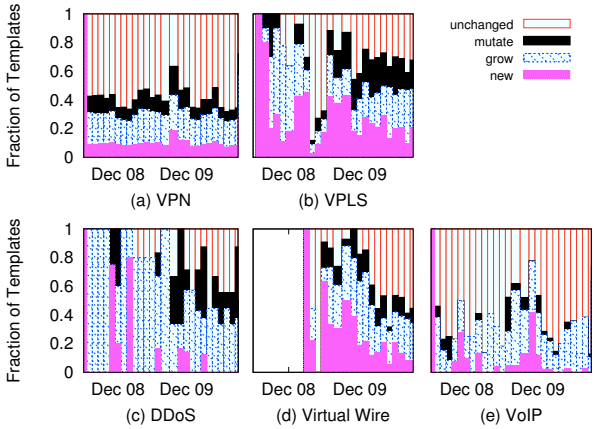


Figure 7: Longitudinal analysis of changes in templates.

tially, when new devices are being introduced, we find that 'new' templates dominate; when the service expands, 'grow' and 'mutate' dominate. Again, we find that customer provisioning contributes significantly to the overall flux in configuration templates.

On the whole, we note that the configuration templates in use constantly grow in number and become increasingly specialized as a service grows and as customer requirements evolve; we study this issue in greater depth in Section 5. The rate of flux in templates depends on how long a service has been offerred – relatively new services tend to observe a far greater amount of flux.

Similar to our referential analysis, our analysis of templates indicates that a lot of the complexity arises from provisioning of customers, which includes adding new templates, and changing service definition for subsets of existing customers. The resulting high number of templates means that, for each candidate change to the service, a collection of templates across a small set of devices within the network would require consistent updates. As the service evolves, this would inevitably make key management tasks, especially those related to provisioning, more challenging.

4.2 PE-Core Configuration

Next, we examine configuration changes required to connect the PE devices to the MPLS core. This mainly involves configuring the interface on the core devices and thereby establishing the data plane for the services. We focus on VPLS and VPN. Since DDoS, Virtual Wire, and VoIP are built over VPLS and VPN, our results for VPN and VPLS include PEs used by these three services.

Figure 8 (a) shows the number of existing stanzas that must be changed to set up routing adjacencies over the PE-facing interface. Figure 8 (b) shows the number of interface-specific stanzas that must be set up to ensure that the appropriate data plane policies are applied to the interface, such as policing, queuing and access control. From (a), we note that, across all three services, most interfaces require configuration of just two stanzas to set up routing (corresponding to IGP and BGP).

From (b), we see that setting up the interface-specific data plane policies appears complex at first glance, requiring the configuration of as many as 20 or so different stanzas for all three services – these stanzas include CoS specifications (class maps and policy maps), and NetFlow configuration. However, as we show in Figure 8 (c), few of these stanzas are "new" across interfaces corresponding to PEs of a particular service. Thus, most core router interfaces appear to be reusing exactly the same data plane policies across PEs for a service. Furthermore, we examined a time series of the stanzas on the core router and found few changes to them over time.

Thus, we conclude from our analysis of the PE-core configuration that: (i) the configuration on the core routers is fairly service-agnostic, and (ii) it is reasonably simple to integrate PEs of new services.

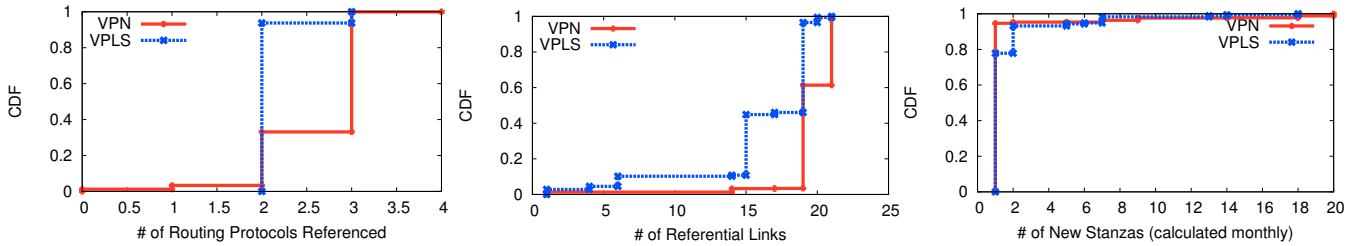
4.3 Control-Plane Configuration

As mentioned earlier, a common approach to designing the control-plane for network-based services is to use the MP-BGP protocol for redistributing routes between different service PEs. In what follows, we analyze the complexity in the configurations of the individual service-specific control-planes.

As shown in Section 2.2.1, BGP peering sessions among route-reflectors are set up by configuring different `neighbor` commands on the routers at both end-points of the peering session. We now examine the extent of referential dependencies underlying the configuration of the route-reflectors for various services.

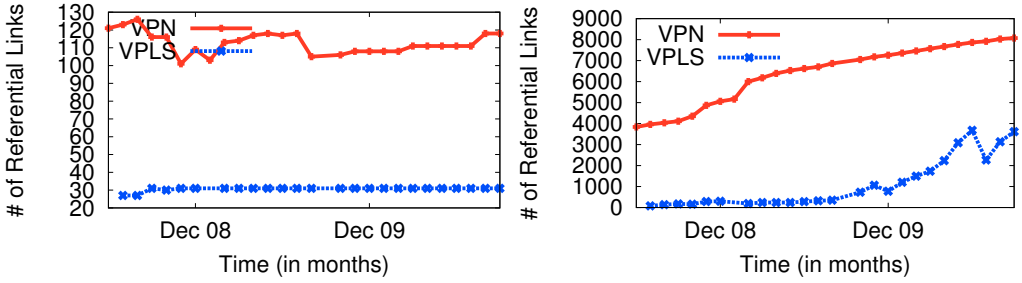
We observe that the median number of links in the route-reflector referential graph remains stable over time across the services (Figure 9 (a)), but the worst case referential graph link count (Figure 9 (b)) increases over time. This increase is dramatic in the case of the VPLS service. For this service, the growing complexity is purely due to the service growth – growing number of PEs necessitate constant changes to the route-reflectors to maintain connectivity and reachability between the newly added PEs. Further examination reveals that a predominant number of referential links in Figure 9 is specifically due to BGP neighbor commands to set up BGP sessions across PEs.

In Figure 10, we show the relative proportions of stanzas of different types in the route-reflectors for the two services. We see that the relative proportions of the different stanza types vary across the two services. VPLS route-reflectors have about 50% of the stanzas attributed to CoS policies while the VPN route reflectors have only 15% attributed to CoS policies. The differences arise due to the fact that the VPLS service imposes more policies to protect the route-reflectors. For example, while both services restrict access to management services, only VPLS throttles the number of connec-



(a) # of routing protocols configured to setup peering (b) # of interface-specific data plane stanzas (c) # of new interface-specific stanzas

Figure 8: Configuring core routers to integrate services.



(a) Median number of links (b) Worst-case number of links

Figure 9: Properties of the per-service RR referential graph.

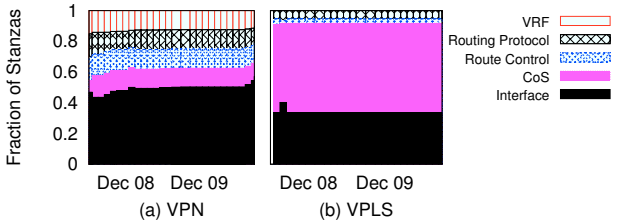


Figure 10: Longitudinal analysis of the types of stanzas in the referential graph (Averaged over RRs).

tions to the VPLS route-reflectors. The VPLS route reflectors are more safely guarded because the ratio of PEs to route-reflectors is much larger for the VPLS service than for the VPN service.

In conclusion, we find that the control-planes for the individual network services themselves are complex and distinct from each other, and the complexity grows as services themselves grow. The main reason is the establishment of BGP sessions required for route redistribution.

5. CUSTOMER PROVISIONING UNDER THE MICROSCOPE

In the previous section, we attributed the complexity in PE configurations mainly to customer-specific issues. In this section, we re-examine the referential graphs and templates for the PEs, focusing on the portions of the models created in response to the provisioning of the customers for a service. We compare these portions across different customers and over time. This comparison provides insights into why configuring the same services for different customers increases the complexity significantly rather than modestly. This section focuses on VPN and VPLS customers since customers for the other services are provisioned as part of the VPN/VPLS virtual networks.

Intuitively, we expect that provisioning the same services should entail only a slight change from one customer to the next. Due to the modularity of configuration languages, configuration across customers should differ mainly in a few parameters. Further, configuration stanzas can, in theory, be reused across different customers provisioned on the same devices. However, as we show below, we find that complexity arises because customers often require significantly different configuration due to different ways in which a service is actually used. We find that reuse does exist between customers, but is rather limited.

5.1 Configuration Complexity

We evaluate the complexity of provisioning customers for a service by examining the referential graph on a per PE basis, and the number of templates used by a customer across all PEs.

In Figure 11 (a), we present the number of links in the PE referential graph that pertain to provisioning a customer site for all customers of VPLS and VPN services. In the figure, we observe two small jumps in the CDF for the VPLS service. These two jumps correspond to customers employing a point-to-point service, requiring only four stanzas as shown in Figure 12 (a), and a point-to-multipoint VPLS service, requiring six stanzas as shown in Figure 12 (b). As shown in the figure, VRF and interface stanzas are common to both types of VPLS services. In addition, the point-to-multipoint service requires filters and policers for incoming as well as outgoing traffic. Since point-to-point service only has two end-points, filters and policers for outgoing traffic are not required once they are applied to the incoming traffic.

Although some of the jumps in the number of referential links required for different customers of the VPN service (in Figure 11 (a)) arises for reasons similar to the VPLS services, remaining jumps are due to other reasons. The jumps at x equal to three and five are due to a VPN-equivalent of the VPLS point-to-point and point-to-multipoint services. However, the jumps at seven and fifteen are due to customers which require fine-grained control over CoS accorded to their traffic. The policies for these users divide traffic into different classes based on ToS/precedence fields, with each class getting a different treatment. For these customers, complexity increases because configuring the service requires references to a few policy maps which in turn reference class maps. The sharp jump at sixteen represents customers which require both fine-grained control over traffic as well as over how routes are redistributed in the

VPN. The customers use route-maps and community-list to filter out updates received from the route-reflectors. Finally, the tail is due to configuration of customer end-points on a PE from a different vendor. (We explore differences due to vendor configuration languages in greater detail in Section 6.1.)

Next, in Figure 11 (b), we examine the CDF of the number of templates used for provisioning the virtual networks of each customer. Most customers have a small number of templates: 90% of the customers have less than three templates. However, a small number of customers have a large number of templates. The differences between the tails for the two services arise because of the size of the customer's virtual networks. Customers with large virtual networks have sites with specialized configuration to overcome scale limitations; these require large number of templates resulting from the differences in the setup of VRFs across different sites.

To summarize, our findings are: (i) significant differences exist between customers of a service, (ii) these differences affect the difficulty of provisioning customers, and (iii) a small number of customers lead to high complexity in PE configuration.

5.2 Configuration Reuse

In this section, we aim to understand whether and how configuration reuse simplifies the action of customer provisioning. To do this, we examine the extent of such reuse both in provisioning individual customer sites as well as across the entire network. We start by analyzing reuse amongst the stanzas that exist within a PE. Next, we examine the amount of reuse amongst the stanzas in templates that are shared accross PEs.

In Figure 13 (a), we present the CDF of the fraction of the total number of stanzas in the referential graph that are reused for provisioning different customers of a service. Similar to the previous section, we limit out analysis to one customer -facing interface. We define a stanza in the referential graph to be reused, if at least two different stanzas explicitly refer to it, indicating that these two stanzas actively depend on it. We observe that although a significant amount of reuse exists (e.g., 40% of VPN customers reuse $\geq$ 75% of stanzas within a PE), there is never complete reuse in provisioning a customer. For VPLS, this occurs because at least two stanzas related to the configuration of the customer-facing interface and the VRF on it must be unique per interface. For VPN, limited reuse is due to the differences in policies that we described earlier in this section.

Using the templates, we extend the definition of reuse beyond a device to the entire network. We define a stanza in the referential graph as being reused if the stanza is in a template (indicating use across multiple devices). We observe, in Figure 13 (b), that using this definition of reuse brings downs the number of customers with no reuse (the jump at zero drops across all services). The CDFs shift down because while some policies are not shared by customers set up on a PE, these policies are shared by customers on different PEs. This indicates that certain customers have more in common with customers on other PEs than on their own PE. However, despite this sharing across customers, there is still a need for specialized stanzas as is evident from the fact that reuse is still limited.

Next in Figure 13 (c), we compare reuse over time for the VPN service. We observe that the fraction of unique stanzas grows over time indicating that customer provisioning does not become simpler, as the number of configuration stanzas being reused actually goes down. This observation also holds true for the VPLS service; however, we omit results due to space constraints.

Customers of the VPN service are in general more difficult to set up (i.e., referential graphs are larger) than customers of the VPLS service, especially for VPN customers employing some of the complex usage patterns discussed earlier in this section. Further, the larger number of usage patterns mean that reuse is less prevalent across VPN customers. The consequences of this are twofold: first, given the same number of customers, the configuration on a VPLS PE should grow much more slowly than on a VPN PE; and second, much of the complexity on VPN PEs is unavoidable, and required to implement large number of distinct usage patterns.

To summarize, we find that: (i) overlap exists between the configuration required to provision a customer for a particular service; (ii) configuring a customer still requires unique elements that other customers on the same PE or even across the entire network do not require; (iii) this kind of specialization leads to increased complexity in customer provisioning; and (iv) the extent of specialization increases with time.

6. TRADE-OFFS IN CONFIGURATION COMPLEXITY

Our focus so far has been on evaluating the complexity of the configurations underlying network-based services. We found customer provisioning and the service control-planes contribute significantly to the complexity. Since configuration is a crucial aspect of a service's design, it is important to control the complexity arising from the above two causes. Despite this, whether it is actually possible to mitigate complexity depends on other important factors. In this section, we examine two such factors – choice of vendor for devices and performance/scalability constraints – and show how to make systematic choices that result in simpler configurations (where possible, given the constraints) for customer provisioning and control-plane design.

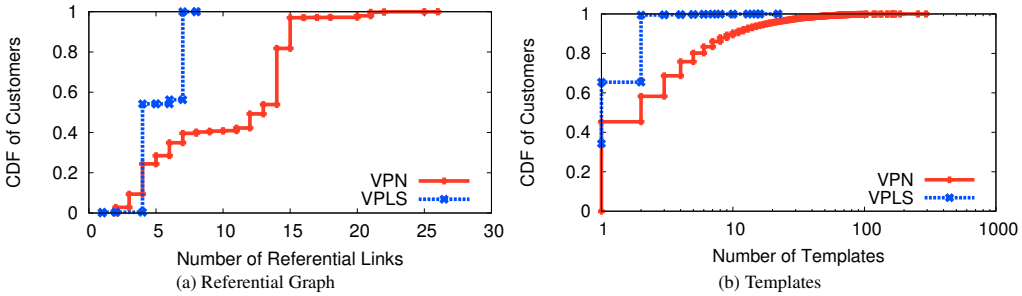
6.1 Choice of Vendor

Configuration languages of different vendors often vary a lot in language constructs, syntax and modularity. These differences affect the complexity of configuring the same functionality across devices from different vendors. Although configuration complexity has a huge bearing on service manageability, ISPs do not always choose their vendors solely based on ease or difficulty of configuring services on their devices. Other factors such as functionality and features, performance, price and management support play an equally important role.

To understand the impact of vendor on configuration complexity, we focus on the VPN service since it is a large service with PEs from different vendors. More specifically, we focus on customers whose sites are connected to PEs from different vendors. This allows us to perform a head-to-head comparison across vendors. The focus of this comparison is to examine the complexity of implementing the same policy across the two vendors and not to examine the complexity of the policy being implemented.

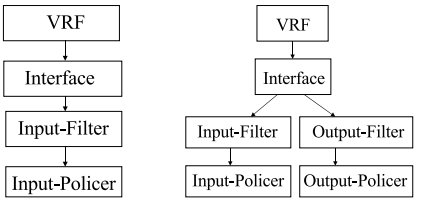
In Figure 14, we compare the configurations of the customers on the VPN service according to their referential dependencies. We use our model to compare the complexity of how configuration languages are used, and not the inherent complexity of the languages. Our results show that it is simpler to configure end-points on Vendor-1 than on Vendor-2: Vendor-2's configuration language requires more stanzas to express the exact same route control and CoS policies. For example, Vendor-1's language uses a combination of one route-map and several community lists to control the set of routes accepted into the VRF. Vendor-2, however, uses six different maps: two for input and four for output with each using on average two different community lists.

Since vendor configuration languages appear to play a role in



(a) Referential Graph (b) Templates

Figure 11: Per customer properties for all customers.



(a) point-to-point customer (b) point-to-multipoint customer

Figure 12: Referential graph for provisioning VPLS customers.

determining the configuration complexity of customer end-points, an appropriate choice of vendors could lead to a significantly simpler configuration during provisioning, thereby improving service manageability.

6.2 Choices Based on Resource Limits

Due to resource limitations, operators are often forced to trade-off a simple design for an efficient but more complex design. In this section, we present the implications of such design choices on a service's configuration, and highlight how certain choices, while substantially reducing the resource usage, introduce little complexity, whereas other choices involving a slight reduction in resource usage can significantly increase the complexity of the service. We illustrate this by (i) examining connectivity between customer sites, a key issue tied to provisioning; and (ii) the performance/scalability trade-offs made in configuring the per-service control-plane.

6.2.1 *Connectivity between Customer Sites*

The connectivity between customer sites in a VPN, i.e., which sites can communicate directly with one another, has a significant bearing on the resource usage. Typical choices available to ISPs are: (i) full mesh, where every site is allowed to communicate directly with every other site; (ii) hub and spoke, where one customer site is chosen as a hub, while the rest are chosen as spokes; communication between spoke sites is only allowed through the hub site; and (iii) multi-hub and spoke, where a few customer sites are chosen as hubs, while the rest are chosen as spokes and assigned to specific hubs. The mesh configuration is usually employed by default, but depending on the traffic matrix among customer sites, one of the other choices could be employed [22]. The design changes over time in response to changes in the traffic matrix.

The hub and spoke design can offer significant savings in the number of routes stored by a VRF in the PE's forwarding table, which is often a critical resource for ISPs. This is because routes learned from a spoke site need to be propagated only to the hub site and not to other spoke sites. Thus, PEs with the spoke VRFs only need to store routes learned from the attached VRFs and a default route from all hubs in the VRF's virtual network. However, the savings in VRF space comes at the expense of additional latency the customer incurs for inter-spoke communication as well as extra bandwidth consumed by such traffic on the ISP's network.

In Table 2, we present the trade-offs between different choices in terms of the amount of state and the number of templates required to realize different configurations. From the table, we observe that *given the choice between a mesh and hub and spoke design, the operator should choose hub and spoke* as it provides a large reduction in state with minimal configuration overhead. However, when *given the choice between mesh and multi-hub and spoke, the choice is not as clear, and depends on the properties of the virtual network*.

	mesh	hub and spoke	multi-hub and spoke
# Routes (Spoke)	CE	$1+C$	$1+C$
# Routes (Hub)	n/a	CE	CE
# Routes (Total)	CE^2	$CE+(C+1)(E-1)$	$CEH+(C+1)(E-H)$
# Templates	1	2	$2H$
# Referential Links	0	0	0

Table 2: Number of routes and difficulty for maintaining each of the different VPN designs. Here, C is the number of customer routes, E is the number of customer sites, and H is the number of hubs.

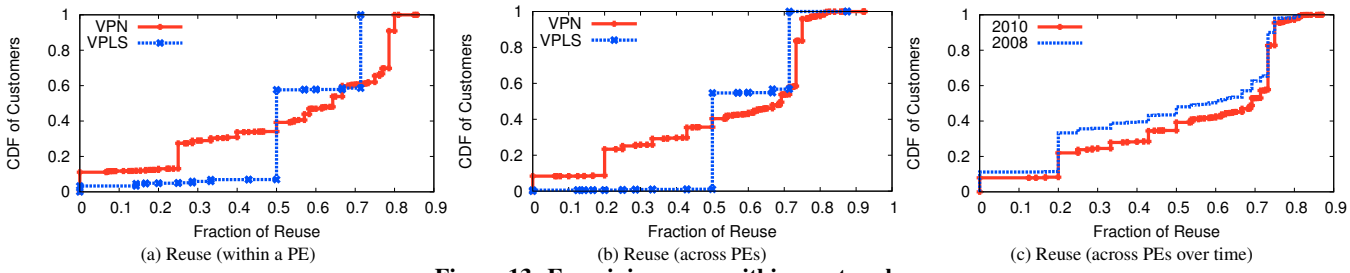
To illustrate, we present a simple example of a customer VPN with ten sites and 40 routes advertised by each customer's site. For this customer, full mesh ensures a single template but requires 400 routes in the VRF for each site. Transitioning to a hub and spoke reduces the VRF size to 41 at all but one PE, and increases the number of templates only to two. Finally, a transition from hub and spoke to a multi-hub and spoke with five hubs and one spoke per hub, increases the number of templates from two to ten and increases the VRF sizes to 400 at four sites. Thus, there is a significant increase in complexity but not a commensurate improvement in state. If the customer's traffic patterns allows a set up containing 3 hubs and 2-3 spokes per hub, then using multi-hub and spoke can result in significant state improvement relative to mesh (4000 vs 1487 routes per VRF) at low added complexity (1 vs 6 templates).

Conversations with the operators of the network indicate that the discussion above indeed crystallizes a trade-off they are facing in reality: for several VPN customers, the ISP is considering using multi-hub and spoke, but the design complexity is preventing operators from implementing it.

6.2.2 *MP-BGP Route Distribution*

The most common question for the MP-BGP control-plane is how to interconnect PEs for exchanging routes. Typical choices available to an ISP are: (i) full mesh, where every PE is connected to every other PE; (ii) route-reflectors, where every PE is connected to a set of route-reflectors (RRs), and routes between PEs are exchanged via RRs; and (iii) multi-plane route-reflectors [19], a recent proposal where RRs are divided into "planes", and those in a given plane only learn and distribute a subset of routes.

While making a choice, operators examine the number of peering sessions that PEs and RRs need to maintain, as well as the number of routes that RRs need to maintain. Depending on these parameters, operators choose a design that minimizes resource consump-



(a) Reuse (within a PE) (b) Reuse (across PEs) (c) Reuse (across PEs over time)

Figure 13: Examining reuse within a network.

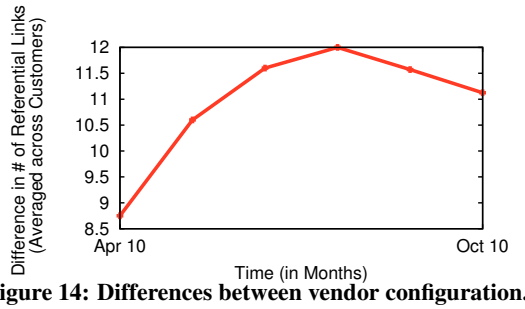


Figure 14: Differences between vendor configuration.

	mesh	Route-Reflectors	Multiplane Route-Reflectors
# Peering (PE)	N	2	2
# Peering (RR)	n/a	N	N
# Peering (Total)	$N(N-1)$	$R(R-1)+2N$	$M(R/M*(R/M-1))$ $+2N+M(M-1)$
# Routes (PE)	T	T	T/M
# Routes (RR)	T	T	T/M
# Templates	1	2	$2M$
# Referential Links	$N(N-1)$	$R(R-1)+2N$	$M(R/M*(R/M-1))$ $+2N+M(M-1)$

Table 3: Number of peering sessions, routes and difficulty for maintaining each of the different BGP control-plane designs. Here, N is the number of PEs, R is the number of route-reflectors, M is the number of different planes, and T the total number of routes contributed by all customers. We also assume that each PE is connected to two RRs for robustness purposes.

tion on PEs for maintaining peering sessions, including exchange of routing state.

In Table 3, we present the trade-offs between different designs. We examine the trade-off not only in the traditional terms of the number of peering sessions, the number of devices, and the size of the routing tables transferred, but also in terms of the configuration complexity (i.e., the number of templates and the size of the referential graphs).

From Table 3, we observe that a tremendous amount of reduction in the number of peering sessions arises by switching from the mesh to a single plane route-reflector setup, with only a modest change in the number of templates from one to two. Thus, *mesh is clearly inferior to using route-reflectors*.

However, with the multi-plane route reflector setup, the complexity may increase significantly. We illustrate this with a simple example. Consider an ISP service with 2,000 PEs, 50 RRs in a single plane, and 20,000 customer routes. For a single plane route-reflector design, the resulting number of peering sessions is 6,450 and each RR has to maintain 20,000 routes. However, if the ISP switches to a multi-plane solution with ten planes, the total number of peering sessions goes down to 4,290 while the number of routes in each RR is reduced to 2,000. This reduction in memory and peering sessions comes at a price though as the configuration complexity increases due to loss in uniformity – the number of templates used grows from 2 to 20 in this case. Thus, *the choice of multi-plane and its set up (i.e., number of planes) must be made carefully to ensure that the increase in the configuration complexity remains low.*

7. DISCUSSION

We describe key implications of our work on other proposals for network-based service design, as well as on tools for service configuration management. We then describe some limitations of our work.

Implications for other work on network-based services. Several prior works on network-based services have focused on issues related to the performance of the services (see for example, prior work which explores the reduction of VPN memory consumption [22, 14, 23, 6], diagnoses VPN problems [26], or models the usage patterns of IPTV networks [20, 21]). Our study is complimentary as it shows that many of the proposed changes may require intricate (re)configuration which could make future changes to services hard. This aspect of service design should be considered as an equally important requirement to understand if the proposals are attractive in practice.

Configuration management. Our study also has implications for common configuration management tools. For example, our results show that template-based tools such as Presto [9], while effective for general management of an ISP's devices may prove inadequate for configuration management of certain services. For example, Presto may not really simplify PE configuration in VPNs, where the number of templates is very large (Section 4.1.2). In general, given the complexity underlying most services, a simpler alternative in the long term may be to configure and manage services through the use of new architectures, e.g, 4D [12], that eliminate the need for low-level configuration by allowing operators to specify high-level network policies and invariants.

Applicability to other ISPs. While we studied a single tier-1 ISP, we believe that our insights have a general applicability to the deployment of network-based services in other ISPs since most tier-1 ISPs utilize similar technologies to realize network-based services. For example, in deploying telepresence, a regional ISP on the west coast uses Cisco H323 gatekeepers as PEs to differentiate customers and employs similar trade-offs as discussed in Section 6.2 to configure the control-plane between these gatekeepers.

Limitations. End-to-end configuration of a service includes not only provider side mechanisms but also customer side configuration. Due to data limitations, our study focused mainly on the former. We leave a study of customer side (CE) configuration issues for future work. We also note that our work focuses on the configurations of devices used in network services, and related factors such as the vendors in use and resource limitations faced. A more comprehensive study of challenges and trade-offs in services, however, requires considering cost, ease of troubleshooting in alternate

designs, and limitations imposed due to available functionality in various devices.

8. RELATED WORK

Our study is motivated by, and complimentary to, prior work that highlights the complexity in configuring Class-of-Service [24] and BGP policies [16, 10] in ISP networks. The latter also highlights the relatively high incidence of errors in BGP configuration, and blames the poorly designed router configuration languages for many of the errors. Note that our study examines complexity *given* commonly-used configuration languages, but it does not examine how much of this complexity arises from the basic design of these languages themselves. We leave this for future work.

While we focus on complexity within an ISP's network configuration, prior works [7, 15, 18, 11] have focused on the causes for complexity in enterprises. These studies found core enterprise network designs to be complex due to the usage of VLANs, route-redistribution commands, and network security and filtering policies. For ISP network services, we find that a significant amount of complexity can be attributed to the configuration of customer virtual networks at the edge, which involves the usage of CoS, VRFs, and route control stanzas, while the core network itself is simple.

The models we employed in this study are similar to those used in [7] to study enterprise networks, but with key extensions to focus on network-based services and the unique functionality found in ISP networks (e.g., the extensive use of route-reflectors, MPLS and router virtualization).

Finally, our study adds to a growing body of work that has found mining configurations to be a useful way to obtain insights about networks. E.g., recent works [17, 25] have used configuration files to perform root cause analysis, and troubleshooting of anomalies and performance problems in ISPs.

9. CONCLUSIONS

The usage and deployment of network-based services is growing as ISPs aim to provide advanced features to residential and enterprise customers. In this paper, we present the first large-scale analysis of these services, focusing on their design and configuration. We study several years' worth of configuration data corresponding to five services in a tier-1 ISP. We decompose the configurations into various tasks performed by operators and systematically highlight the complexity underlying each. We show that complexity grows over time. We find that while the exact causes of complexity may differ across different services, two factors – customer provisioning differences and control-plane (BGP) configuration – consistently make the overall designs complex. We conclude the study by exploring ways to reduce complexity, specifically, by considering the alternatives in, and the trade-offs imposed by, factors such as the choice of vendors and resource scaling.

We hope that our findings can lead to a broader discussion on understanding service configurations, on the design of both service-specific and service-agnostic support mechanisms, and potentially on alternate network architectures and router mechanisms with much better intrinsic support for network-based services.

10. ACKNOWLEDGMENTS

We would like to thank the operations team at the tier-1 ISP for providing the data and documentation about the services. We would also like to thank Katerina Argyraki (our shepherd), Aaron Gember, Seungjoon Lee, Kobus van der Merwe, Dan Pei, Kevin D'Souza, Adrian Cepleanu, Stephen Hutnik, Maria Napierala and the anonymous reviewers for their insightful feedback. This work is supported in part by NSF grants CNS-0746531, CNS-1017545 and CNS-1050170. Theophilus Benson is supported by an IBM PhD Fellowship.

11. REFERENCES

[1] Cisco visual networking index: Forecast and methodology, 2009-2014. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html.

[2] Virtual routing and forwarding. http://www.cisco.com/en/US/docs/net_mgmt/active_network_abstraction/3.7/reference/guide/vrf.html.

[3] Virtual routing and forwarding. http://www.juniper.net/techpubs/software/junos/junos61/swconfig61-routing/html/instance-overview.html#1017937.

[4] T. Bates, R. Chandra, D. Katz, and Y. Rekhter. Multiprotocol Extensions for BGP-4. RFC 4760 (Draft Standard), Jan. 2007.

[5] T. Bates, E. Chen, and R. Chandra. BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP). RFC 4456 (Draft Standard), Apr. 2006.

[6] Z. ben Houidi and M. Meulle. A new VPN routing approach for large scale networks. In *Proc. IEEE ICNP*, 2010.

[7] T. Benson, A. Akella, and D. A. Maltz. Unraveling the complexity of network management. In *NSDI*, April 2009.

[8] D. Caldwell, A. Gilbert, J. Gottlieb, A. Greenberg, G. Hjalmtysson, and J. Rexford. The cutting edge of IP router configuration. In *In Proc. of Hotnets-II*, 2003.

[9] W. Enck, P. Mcdaniel, A. Greenberg, S. Sen, P. Sebos, S. Spoerel, and S. Rao. Configuration management at massive scale: System design and experience. In *In 2007 USENIX ATC*, pages 73–86, 2007.

[10] N. Feamster and H. Balakrishnan. Detecting BGP configuration faults with static analysis. In *Proceedings of USENIX NSDI*, pages 43–56, Berkeley, CA, USA, 2005.

[11] P. Garimella, Y.-W. E. Sung, N. Zhang, and S. Rao. Characterizing VLAN usage in an operational network. In *ACM INM '07*, pages 305–306, New York, NY, USA, 2007.

[12] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang. A clean slate 4D approach to network control and management. *SIGCOMM Comput. Commun. Rev.*, 35(5):41–54, 2005.

[13] T. Kamiya, S. Kusumoto, and K. Inoue. Ccfinder: a multilinguistic token-based code clone detection system for large scale source code. *IEEE Trans. Softw. Eng.*, 28(7), 2002.

[14] C. Kim, A. Gerber, C. Lund, D. Pei, and S. Sen. Scalable VPN routing via relaying. In *Proceedings of SIGMETRICS*, pages 61–72, New York, NY, USA, 2008. ACM.

[15] F. Le, G. G. Xie, D. Pei, J. Wang, and H. Zhang. Shedding light on the glue logic of the Internet routing architecture. In *Proceedings of ACM SIGCOMM*, pages 39–50, New York, NY, USA, 2008.

[16] R. Mahajan, D. Wetherall, and T. Anderson. Understanding BGP misconfiguration. In *Proceedings of ACM SIGCOMM*, pages 3–16, New York, NY, USA, 2002.

[17] A. A. Mahimkar, H. H. Song, Z. Ge, A. Shaikh, J. Wang, J. Yates, Y. Zhang, and J. Emmons. Detecting the performance impact of upgrades in large operational networks. In *Proceedings of ACM SIGCOMM*, pages 303–314, New York, NY, USA, 2010.

[18] D. A. Maltz, G. Xie, J. Zhan, H. Zhang, G. Hjálmtýsson, and A. Greenberg. Routing design in operational networks: a look from the inside. In *Proceedings of ACM SIGCOMM*, pages 27–40, New York, NY, USA, 2004.

[19] M. Napierala. AT&T MPLS network and VPN services. PLNOG, 2008.

[20] T. Qiu, Z. Ge, S. Lee, J. Wang, J. Xu, and Q. Zhao. Modeling user activities in a large IPTV system. In *Proceedings of ACM IMC*, pages 430–441, New York, NY, USA, 2009.

[21] T. Qiu, Z. Ge, S. Lee, J. Wang, Q. Zhao, and J. Xu. Modeling channel popularity dynamics in a large IPTV system. In *Proceedings of ACM SIGMETRICS*, pages 275–286, New York, NY, USA, 2009.

[22] S. Raghunath and K. K. Ramakrishnan. Trade-offs in resource management for Virtual Private Networks. In *Proc. IEEE INFOCOM*, 2005.

[23] S. Raghunath, K. K. Ramakrishnan, and S. Kalyanaraman. Measurement-based characterization of IP VPNs. *IEEE/ACM Trans. Netw.*, 15:1428–1441, December 2007.

[24] Y.-W. E. Sung, C. Lund, M. Lyn, S. G. Rao, and S. Sen. Modeling and understanding end-to-end class of service policies in operational networks. In *Proceedings of SIGCOMM*, pages 219–230, New York, NY, USA, 2009. ACM.

[25] D. Turner, K. Levchenko, A. C. Snoeren, and S. Savage. California fault lines: understanding the causes and impact of network failures. In *Proceedings of ACM SIGCOMM*, pages 315–326, New York, NY, USA, 2010.

[26] Y. Zhao, Z. Zhu, Y. Chen, D. Pei, and J. Wang. Towards efficient large-scale VPN monitoring and diagnosis under operational constraints. In *Proc. IEEE INFOCOM*, pages 531–539, 2009.