Transit Portal: Bringing Connectivity to the Cloud

Vytautas Valancius*, Yogesh Mundada*, Nick Feamster*, Jennifer Rexford†, Akihiro Nakao‡
*Georgia Tech †Princeton University ‡The University of Tokyo

1. Introduction

Computing in the cloud is becoming more common as companies migrate their applications into cloud computing platforms to reduce maintenance costs and increase availability. Cloud hosting platforms, such as Amazon's EC2, provide a virtual hosting environment that is easy for a service provider to provision in response to dynamically changing demands. This model lowers the barrier to entry for innovative Internet services and applications. In addition to hosting, services that are hosted in a cloud may also require a virtual *networking* environment to retain real-time, fine-grain control over how traffic enters the cloud. For example, given the global nature of many Internet applications, cloud data centers can be in many diverse geographical locations; a cloud service could expose some of these connectivity options and let the service provider to decide how the traffic is routed into and out of the cloud, as well as across the cloud for that service.

Various virtual network infrastructures (*e.g.*, VINI) already allow researchers to instantiate networks and run simultaneous network experiments on the same infrastructure [1]. In addition to the virtual network infrastructure itself, however, cloud networks also need external connectivity. Today, cloud hosting platforms provide only limited options for such external connectivity; the standard method is to provide bulk upstream connectivity to every application that is hosted on the cloud. Certain hosted applications may, however, need more control over how traffic exits and enters the cloud via its external connections. For example, some cloud users might prefer to select upstream providers based on cost while others might focus on performance.

Unfortunately, such flexibility is currently difficult to achieve. Cloud infrastructure providers try to balance performance and cost when it comes to their connections to the Internet, so customizing upstream connectivity for each service or virtual network in the cloud introduces prohibitive management and operational overhead, particularly if the service itself is short-lived or otherwise unstable. Virtual networks, on the other hand, face multiple hurdles if they want to peer directly with the Internet service providers, including: (1) negotiating direct contracts with the service providers, (2) obtaining IP addresses for proper peering with the ISPs, and (3) connecting the virtual network to the external legacy network. These challenges become even more significant when the virtual networks are short-lived (*e.g.*, in the case of researchers running experiments, or a virtual network that is provisioned for a specific event). In these cases, negotiating upstream connectivity from ISPs, who prefer stable, predictable BGP sessions, may be inconvenient or simply untenable.

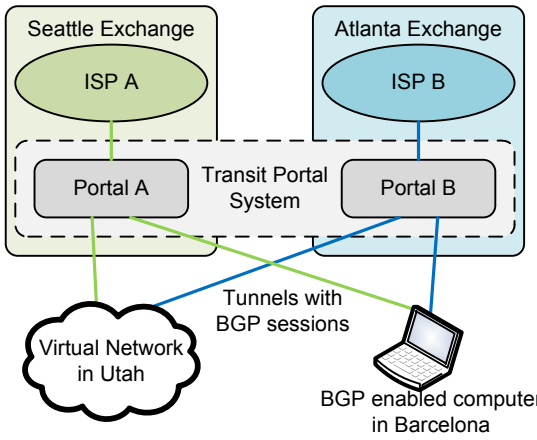

Figure 1: Transit Portal system demo setup.

This demonstration will present *Transit Portal*, a system that provides external, full fledged, on-demand Internet connectivity to downstream networks. Transit Portal offers a conventional Border Gateway Protocol (BGP) [2] session interface to downstream networks and presents an illusion of a direct connection to service provider, without requiring each virtual network to establish an explicit contract with each upstream provider. Transit Portal aggregates such customer sessions and provides a single, stable BGP session to the upstream provider. As shown in Figure 1, Transit Portal can be deployed in geographically distributed popular exchange points, close to the local ISPs. Virtual networks then peer with such upstream ISPs though BGP sessions that terminate on Transit Portal, using either dedicated transport or tunnels; this connectivity may be either local (*e.g.*, via a VLAN) or remote (*e.g.*, via a GRE tunnel).

We envision many possible applications for Transit Portal in addition to connecting virtual networks to the external Internet. For example, Transit Portal can provide connectivity to any geographically distributed service that requires control over routing. For example, Transit Portal could be used by Domain Name Service (DNS) providers to announce DNS server addresses though many ISPs to offer anycast service. Transit Portal could also be used by network operators to monitor their prefixes (*e.g.*, for detecting prefix hijacking). Researchers could also use Transit Portal to conduct experiments that rely on real-time BGP feeds.

Section 2 describes the design and implementation of Transit Portal. Section 3 explains the demonstration overview and goals.

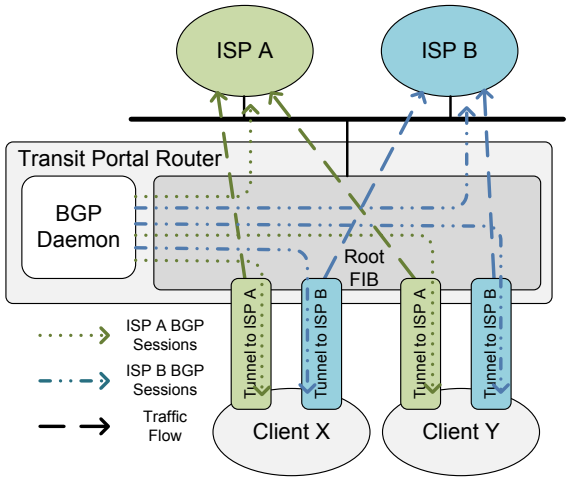

Figure 2: Transit Portal router control plane and data plane with two service providers.

2. Design

Transit Portal consists of the portal routers, which terminate control plane and data plane connections from client networks. The control plane of Transit Portal relies on a modified Quagga BGP routing daemon, and the data plane is implemented using tunneling and policy routing. Some of the function for Transit Portal is described in a CoNext poster [3]. The demonstration extends a demonstration of a "BGP Session Multiplexer", which was presented at the 4th GENI Engineering Conference and showed some of the control-plane functions of Transit Portal.

Control plane The control plane terminates BGP sessions from both the service providers and from Transit Portal users. Each service provider session is terminated into a separate BGP view. BGP view feature allows a BGP instance to have a separate routing table for each provider: The separation of instances is necessary so that clients connecting to Transit Portal receive routes unfiltered by the BGP selection process. A client willing to connect to two different ISPs has two different BGP sessions terminating in two different views. Transit Portal presents the illusion of the direct connectivity to an upstream provider for each such session: the client sees the AS number of the provider it connects to and the updates from providers are propagated with minimum delay and no modifications except for the next-hop address. Similarly, Transit Portal forwards unmodified BGP updates to the upstream providers. Additionally, Transit Portal allows client networks to announce routes to the global Internet to achieve upstream connectivity. The control plane setup is similar to one used by the BGP-Mux [3].

Data plane The data plane of Transit Portal router has two components: (1) a mechanism for delivering the traffic from downstream network to Transit Portal router, and (2) a mechanism for routing such traffic to the appropriate ISP. Transit Portal can deliver traffic in a varity of ways depending on the downstream client's geographical proximity (direct versus local connections) and transport type (tunnels or VLANs versus the dedicated circuit). In our demonstration, we show the data plane with GRE tunneling; in this setup, Transit Portal forms BGP sessions with the downstream networks over GRE tunnels, as shown in Figure 2.

Routing traffic in Transit Portal becomes more complicated when a portal router has more than one upstream provider. For outgoing traffic, Transit Portal uses policy routing rules that forward traffic based on an incoming interface. For example, when a client has two tunnels dedicated to two different providers, the traffic from each tunnel is sent to the appropriate ISP as shown in Figure 2. The Transit Portal installs these rules for each client connection.

Incoming traffic requires more advanced processing: Routing by source interface is insufficient because the same interface could be used by all providers to deliver traffic to all users. To map the incoming traffic to the proper tunnel, Transit Portal performs two-step policy routing based on source MAC address and destination IP address. The source MAC uniquely identifies the upstream ISP associated with the packet on a shared network; the destination IP address helps select the appropriate client network. Conventional routing is infeasible in this case because the downstream networks can advertise the same prefix over multiple tunnel interfaces. To enable downstream policy routing, we modified the Quagga BGP daemon to install the IP prefixes advertisements received from downstream networks to the policy routing database. No routes are installed in the main forwarding table (FIB).

3. Demonstration

The primary goal of the demonstration is to *show how Transit Portal can provide upstream connectivity to networks in a virtual network infrastructure or cloud setting*. These connections, from the perspective of the client network, should function as if the client is directly connected to upstream providers.

The demonstration has several secondary goals. First, we plan to showcase different tunneling technologies for connecting to the Transit Portal. Different tunneling technologies allow greater interoperability with existing networking equipment, which in turn allow customers to deploy new connections to external networks without changes to their existing networks. Second, we plan to show how Transit Portal improves the availability by using tunnel keep-alive messages and automated prefix withdrawals. Finally, we plan to show how Transit Portal provides transparency and stability to an upstream provider even when the downstream connections fluctuate.

REFERENCES

[1] A. Bavier, N. Feamster, M. Huang, L. Peterson, and J. Rexford. In VINI Veritas: Realistic and Controlled Network Experimentation. In *Proc. ACM SIGCOMM*, Pisa, Italy, Aug. 2006.

[2] Y. Rekhter, T. Li, and S. Hares. *A Border Gateway Protocol 4 (BGP-4)*. Internet Engineering Task Force, Jan. 2006. RFC 4271.

[3] V. Valancius and N. Feamster. Multiplexing bgp sessions with bgp-mux. In *CoNEXT '07: Proceedings of the 2007 ACM CoNEXT conference*, pages 1–2, New York, NY, USA, 2007. ACM.