BGP Mux: GEC 7 Demo: OpenVPN Clients and BGP Route Control

Yogesh Mundada, Vytautas Valancius, Nick Feamster
Georgia Tech

Overview

This demonstration shows the interconnection of the BGP session multiplexer ("BGP Mux") with an upstream "provider" at Georgia Tech, the integration of both the control and data planes with the BGP Mux, and a downstream virtual network. This demonstration corresponds to a problem set that is being offered in Georgia Tech's CS 8803 "Next-Generation Networking" Class. That problem set is available here: `http://www.gtnoise.net/classes/cs8803/spring_2010/psets/ps2/ps2.pdf`

The main goal of this demonstration is to show the following in the context of a course assignment:

- *Upstream connectivity.* A BGP Mux in Atlanta is connected to an upstream "provider" (i.e., the border routers of the campus at Georgia Tech).
- *Downstream connectivity to virtual networks.* A downstream client network, connected to the BGP muxes via OpenVPN, with full BGP connectivity from both the Atlanta mux. This downstream connectivity feature is new and shows that *any* host that is capable of running an OpenVPN client can connect to the BGP Mux to get data plane connectivity.
- *Data plane.* A working data plane from a downstream client virtual network—packets flowing to and from the experiment from the Internet.

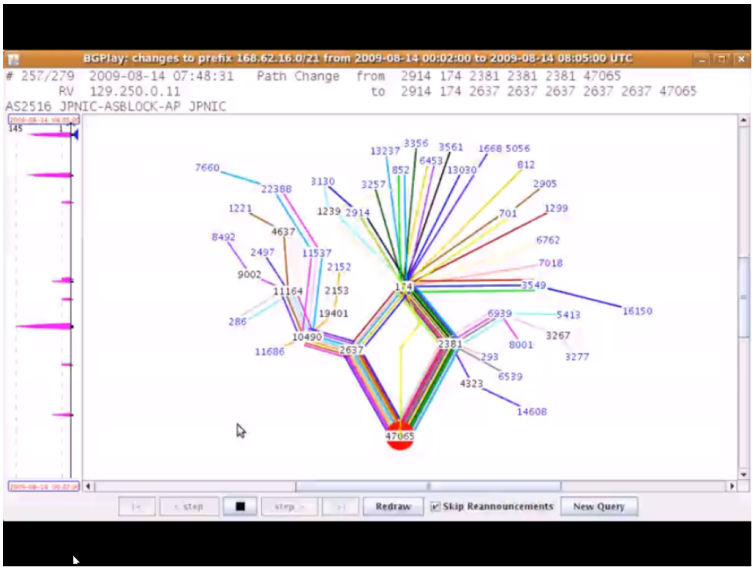

Figure 1: Demo Setup

Demonstration Details

In this section, we describe details of the overall demonstration setup, and next steps for integration.

Upstream connectivity We presently have upstream BGP connectivity via three upstream providers. We have upstream connectivity through Georgia Tech (AS 2637), via which we have direct connectivity to NLR and Internet2, as well as the commodity Internet. Second, we have upstream connectivity through

PSGNet (AS 3130), which has upstream connectivity via Verio (AS 2914) and Sprint (AS 1239). Finally, we have connectivity via the University of Wisconsin at Madison (AS 2381).

The BGP Muxes at both Georgia Tech and Wisconsin are connected via a VLAN with the border router on the respective campus networks. On each site, a dedicated server on the local network acts as the BGP Mux and has a dedicated interface for the VLAN to the border router. This point-to-point link is numbered with a /30 IP prefix, with one address for each side of the connection.

This demonstration shows upstream connectivity to the Georgia Tech network only.

Connectivity between virtual networks and the BGP Mux These nodes run the *DTunnels* Linux kernel, which allows the Mux to establish downstream connections to the client network via Ethernet GRE (EGRE) tunnels. This EGRE-based downstream connectivity reflects progress over the demonstration at GEC 4, where these downstream connections were established over layer 3 (eBGP multihop). This direct connectivity is only currently possible at Georgia Tech and University of Wisconsin, where we have direct control over the kernel on the BGP session multiplexer itself.

OpenVPN Setup. We have configured a virtual private server in Emulab that runs Quagga and OpenVPN. For the purposes of the demonstration, this setup is running on the Emulab node itself, but this could also be running inside a virtual machine. We have uploaded a sample `openvpn.conf` file to the DTunnels wiki to show how to configure the client and server for this part of the setup.

Data plane This demonstration shows the a functional BGP Mux data plane that can be connected to *any* downstream client network that runs an OpenVPN client that connects to the BGP Mux. The current setup shows data traffic (i.e., pings) traveling between the downstream virtual network and the hotel gateway for GEC 7.

BGP Route Control The accompanying video shows a downstream client that connects to the Internet via two upstream networks: one at Georgia Tech and one at the University of Wisconsin. The client alters its outbound route announcements by adding BGP "communities" to cause the outbound client to prefer one upstream ISP for inbound traffic over the other. The video shows how various ASes shift from one ISP to the other after learning the new route. Similarly, any experiment could control its inbound and outbound traffic as if it were directly connected to the respective upstream ISPs.

Next Steps

We continue to tackle the following issues:

1. **Multiple downstream client networks.** Currently, we have a single virtual network downstream from the BGP Muxes at both Georgia Tech and Wisconsin. We next plan to test the BGP Mux's ability to exchange traffic with multiple client networks, which will involve forwarding incoming traffic based on the client network's assigned IP address space. For outbound traffic, the BGP Mux must forward traffic to the upstream ISP that the client has selected based on the client network's MAC address.
2. **Multiple upstream ISPs.** We also plan to extend the BGP Mux so that it connects to multiple upstream ISPs in at least one location. This will allow us to test: (1) whether an experimental network can connect to a subset of upstream ISPs; (2) whether the mux can correctly forward traffic to the client network's chosen upstream ISP, given multiple possible choices for upstreams.
3. **Increase the number of BGP Mux deployments.** We are planning to add BGP Mux deployments in KDDI (Japan), Utah, Princeton, and several other locations to increase the locations from which experiments can obtain upstream BGP connectivity.