Multiplexing BGP Sessions with BGP-Mux

Vytautas Valancius and Nick Feamster
School of Computer Science, Georgia Tech

1. INTRODUCTION

This paper describes a BGP-session multiplexer called **BGP-Mux**, which provides stable, on-demand access to global BGP route feeds. This gateway allows arbitrary and even transient client BGP connections to be provisioned and torn down on demand without affecting globally visible BGP sessions. BGP-Mux provides two capabilities: (1) the ability for a client network to receive *multiple* unfiltered routes per destination from a set of upstream ASes; and (2) the ability to provision BGP sessions without introducing global instability. Several applications could benefit from these features:

External connectivity for virtual networks. Architectures that host multiple virtual networks on a shared infrastructure, such as VINI [2] and GENI [4] require connectivity with external networks. In the case of experiments, external connectivity could allow real traffic to be routed over the virtual networks, allowing testing of new convergence algorithms or update types; generally speaking, any network that supports multiple virtual networks will have to redistribute routes from a one external BGP session to multiple virtual networks. This function requires multiple virtual networks on the same set of physical routers to *simultaneously* learn routes to external destinations via BGP.

Unfortunately, experimentation is inherently unstable: Virtual networks may be transient or unreliable, which could cause BGP sessions with neighboring networks to fluctuate rapidly and unnecessarily. Virtual networks should be able to come and go *without disrupting BGP sessions with neighboring networks*. BGP-Mux presents external networks with BGP sessions that are persistent, even when a virtual network's external BGP sessions come and go. Furthermore, a single set of external BGP sessions can provide routes to *all* virtual networks. The demand for such BGP session multiplexing will be even more in pronounced when virtual networks gain wider industry acceptance.

Real-time, on-demand BGP monitoring. Projects such as RouteViews [3] provide access to the history of route updates. Routes are logged periodically to a public server where they can be downloaded, but the infrastructure does not support real-time, on-demand monitoring for BGP routing updates, which might be useful for correlating routing updates with real-time data-plane measurements (*e.g.*, triggering data-plane measurements based on control-plane observations from RouteViews).

The only way to achieve real-time updates today in a non-intrusive fashion (*i.e.*, without connecting directly to each upstream ISP's router) would be to receive BGP updates from a route server (such as those maintained by RouteViews), but then one would only see updates that reflect the changes to the best route to each destination. In contrast, BGP-Mux could provide access to all BGP updates seen by a route server, not just whatever route the route server selects. Researchers can connect to BGP-Mux *on-demand*, whithout lengthy negotiations with the ISPs.

BGP monitoring in production networks. Currently, border routers that have several external BGP sessions propagate only the best route to their internal peers. Today's BGP monitoring techniques use iBGP sessions, which only propagate a single best route for each destination to the monitor. In contrast, BGP-Mux allows any client router, real or virtual, to receive *all* routes for each destination that may be learned over multiple external BGP sessions, without requiring neighboring networks to reconfigure their routers to establish multiple BGP sessions.

This selective propagation makes network wide BGP update monitoring hard. Instead, external BGP sessions could be terminated at BGP-Mux, collected for monitoring and passed to border routers unchanged.

2. DESIGN

Figure 1 shows the high-level operation of BGP-Mux. BGP-Mux has two types of connections: (1) *upstream* connections to ISPs and (2) *downstream* connections to clients, such as virtual networks or monitoring applications. The BGP-Mux has the following design requirements:

- *Transparent multiplexing of BGP routes to multiple downstream networks.* BGP-Mux presents the routes learned on each upstream eBGP session unmodified over client eBGP sessions. Downstream networks should have the illusion that they are connected directly to upstream networks.
- *Isolation of instability in downstream networks.* BGP-Mux can advertise certain routes as originated by itself, even if the routes were originated in unstable or transient client domains. As a result, routes announced by

CoNEXT'07, December 10–13, 2007, New York, NY, U.S.A.

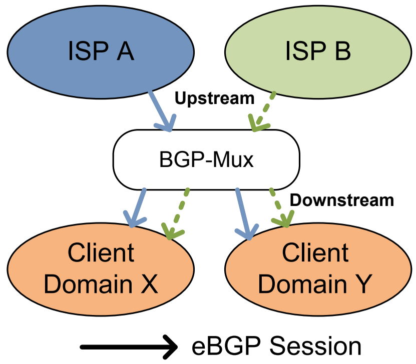

Figure 1: High-level BGP-Mux Operation. Each directed edge represents a BGP session. Client domains may either represent multiple virtual networks on a shared substrate or separate physical networks.

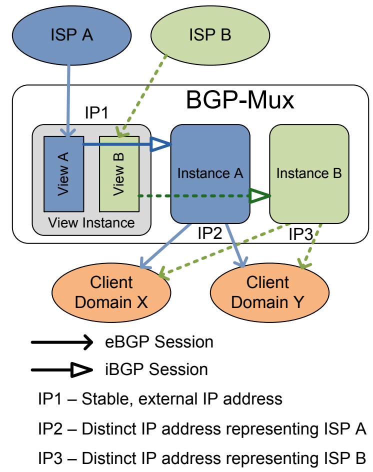

Figure 2: BGP-Mux Implementation

downstream networks need not create globally visible withdrawals or instability if the client goes down.

- *Downstream scalability.* BGP-Mux can be easily distributed across several physical machines and advertise routes to a large number of client networks.
- *Upstream scalability.* As with a route server, BGP-Mux can receive routes from many upstream ISPs.

To ensure *transparency*, the BGP-Mux must re-advertise *all routes* on all BGP sessions, rather than selecting a single best route per destination. *Isolation* requires BGP-Mux to rewrite updates from clients as its own (*e.g.*, removing private AS numbers and any other information specific to the client network). *Downstream scalability* (the ability to connect multiple clients to the BGP-Mux) is achieved by presenting to all downstream networks a separate IP address per upstream ISP. *Upstream scalability* mandates that the BGP-Mux provides a stable externally facing IP address for all upstream BGP sessions.

3. IMPLEMENTATION

We implemented BGP-Mux using the Quagga software router [1]. Figure 2 shows the BGP-Mux implementation in detail. An upstream network establishes a BGP session to a unique external BGP-Mux IP address. To allow all routes to be passed to clients, each BGP session terminates into a distinct *BGP-view*. BGP views provide isolation: Routes in two different BGP views are not directly compared.

Clients must perceive connections to BGP-Mux as direct sessions to upstream ISPs. BGP-Mux thus presents all clients with a different IP address for each ISP by instantiating one BGP instance per ISP, each with its own IP address. This configuration allows clients to connect to multiple BGP instances, each of which provides a BGP route from an upstream ISP. Each BGP instance receives routes from a corresponding view in the “view instance”. Each BGP-view in the view instance maintains a BGP session with one upstream ISP and propagates one route per destination ISP to the corresponding BGP instance via an internal BGP (iBGP) session. Propagating routes over iBGP also allows these instances to reside on separate physical machines, potentially improving scalability.

BGP-Mux processes routes differently depending on their origin. Routes from upstream networks enter into designated BGP-view and in turn forwarded to the corresponding BGP instance. This BGP instance removes the BGP-Mux’s AS number and forwards the route to connected clients. Upstream route processing depends on the policy applied to clients. For example, if no isolation is required between the clients, routes that enter a BGP instance from one client are subject to best-path selection and can reach other clients connected to the same instance.

4. SUMMARY AND FUTURE WORK

This paper presented the design and implementation of BGP-Mux, which provides complete BGP updates to short-lived or unstable domains while preserving global Internet routing stability. We described several possible application scenarios for BGP-Mux, including providing full routing tables to virtual networks, supporting real-time monitoring of routes at route servers, and more complete monitoring of the BGP routes received from neighboring networks at one ISP. We are establishing external BGP connectivity to VINI with an upstream ISP at two interconnection points to evaluate the performance of BGP-Mux in a realistic deployment scenario.

REFERENCES

[1] Quagga software router. http://www.quagga.net/, 2006.
[2] 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, August 2006.
[3] A. Broido and K. Claffy. Analysis of RouteViews BGP data: Policy atoms. In *Workshop on Network-Related Data Management*, May 2001.
[4] The GENI Initiative. http://www.nsf.gov/cise/geni/.