Active BGP Measurement with BGP-Mux

Ethan Katz-Bassett (USC)
with testbed and some slides hijacked from Nick Feamster and Valas Valancius
Before I Start

- Georgia Tech system, I am just an enthusiastic user
  - Nick Feamster and his students:
    - Valas Valancius
    - Bharath Ravi

- Questions for the audience:
  - What would you use this system for? What should we use it for?
  - How do we get more ASes to connect to us?
    - Getting them to agree to peer
    - Then, getting the connection to work
Networks Use BGP to Interconnect

- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic
Networks Use BGP to Interconnect

- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic
Networks Use BGP to Interconnect

- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic
Networks Use BGP to Interconnect

- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic
Networks Use BGP to Interconnect

- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic
Networks Use BGP to Interconnect

- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic
Networks Use BGP to Interconnect

- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic
Networks Use BGP to Interconnect

- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic
Networks Use BGP to Interconnect

- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic
Networks Use BGP to Interconnect

- BGP sessions
- Route advertisements
- Traffic over those routes
- BGP controls both inbound and outbound traffic
Virtual Networks Need BGP, too

Say I have some neat new routing ideas. I want to test them:

- Emulate the type of AS (CDN, stub, etc) of my choice
  - Choose a set of providers, peers, and customers
- Inbound:
  - Choose routes from those providers
  - Send traffic along those routes
- Outbound:
  - Announce my prefix(es) to neighbors of choice, with communities, etc
  - Receive traffic to prefix(es)
- And everyone else should be able to do this, also
Traditionally, BGP Experiments are Hard

I have some neat new routing ideas. How do I test them?

- Passive observation
  - E.g., RouteViews, RIPE
  - Receive feeds only
- Limited “active” measurements
  - E.g., Beacons
  - Generally, regular announcements and withdrawals
- Know the right people
  - Negotiate the ability to make announcements
  - High overhead, limited deployment

All limit what you can do
What I Need to Get What I Want

- Resources
  - IP address space
  - AS number

- Connectivity & contracts
  - BGP peering with real ASes
  - Data plane forwarding

- Time and money
BGP-Mux Provides All This For You

- **Resources**
  - IP address space
    - 184.164.224.0/19
  - AS number
    - AS47065

- **Connectivity & contracts**
  - BGP peering with real ASes
    - 5 Universities as providers
  - Data plane forwarding
    - Send & receive traffic

- **Time and money**
  - One-time cost

---

Active BGP Measurement with **BGP-Mux**
Design Requirements

- **Session transparency**: BGP updates should appear as they would with direction connection
- **Session stability**: Upstreams should not see transient behavior
- **Isolation**: Individual networks should be able to set their own policies, forward independently, etc
- **Scalability**: BGP-Mux should support many networks
A Project Using BGP-Mux

LIFEGUARD: Locating Internet Failures Effectively and Generating Usable Alternate Routes Dynamically

- Locate the ISP / link causing the problem

- Suggest that other ISPs reroute around the problem
  - What would we like to add to BGP to enable this?
  - What can we deploy today, using only available protocols and router support?
Our Goal for Failure Avoidance

- Enable content / service providers to repair persistent routing problems affecting them, regardless of which ISP is causing them

Setting

- Assume we can locate problem
- Assume we are multi-homed / have multiple data centers
- Assume we speak BGP

- We use BGP-Mux to speak BGP to the real Internet: 5 US universities as providers
Self-Repair of Forward Paths

Straightforward: Choose a path that avoids the problem.
Self-Repair of Forward Paths

Straightforward: Choose a path that avoids the problem.
A Mechanism for Failure Avoidance

Forward path: Choose route that avoids ISP or ISP-ISP link

Reverse path: Want others to choose paths to my prefix P that avoid ISP or ISP-ISP link X
  ▶ Want a BGP announcement AVOID(X,P):
    ▶ Any ISP with a route to P that avoids X uses such a route
    ▶ Any ISP not using X need only pass on the announcement
Ideal Self-Repair of Reverse Paths
Ideal Self-Repair of Reverse Paths
Ideal Self-Repair of Reverse Paths
Ideal Self-Repair of Reverse Paths

AVOID(L3,WS)

AVOID(L3,WS)

AVOID(L3,WS)
Ideal Self-Repair of Reverse Paths

AVOID(L3,WS)

AVOID(L3,WS)

AVOID(L3,WS)
Practical Self-Repair of Reverse Paths
Practical Self-Repair of Reverse Paths
Practical Self-Repair of Reverse Paths
Practical Self-Repair of Reverse Paths
Practical Self-Repair of Reverse Paths

1. UW → L3 → ATT → WS
2. L3 → ATT → WS
3. Sprint → Qwest → WS
4. AISP → Qwest → WS
5. Qwest → WS
6. WS

**LIFEGUARD: Practical Repair of Persistent Route Failures**
Practical Self-Repair of Reverse Paths

LIFEGUARD: Practical Repair of Persistent Route Failures

UW → L3 → ATT → WS

L3 → ATT → WS

ATT → WS

Sprint → Qwest → WS

AISP → Qwest → WS

Qwest → WS

Some Web Server
Practical Self-Repair of Reverse Paths

UW → L3 → ATT → WS
L3 → ATT → WS
ATT → WS
Sprint → Qwest → WS
AISP → Qwest → WS
Qwest → WS
Practical Self-Repair of Reverse Paths

\[ UW \rightarrow L3 \rightarrow ATT \rightarrow WS \]
\[ L3 \rightarrow ATT \rightarrow WS \]
\[ ATT \rightarrow WS \]
\[ Sprint \rightarrow Qwest \rightarrow WS \]
\[ AISP \rightarrow Qwest \rightarrow WS \]
\[ Qwest \rightarrow WS \]

avoid(L3, WS)
Practical Self-Repair of Reverse Paths

BGP loop prevention encourages switch to working path.
Practical Self-Repair of Reverse Paths

BGP loop prevention encourages switch to working path.
Practical Self-Repair of Reverse Paths

BGP loop prevention encourages switch to working path.
Practical Self-Repair of Reverse Paths

BGP loop prevention encourages switch to working path.
Practical Self-Repair of Reverse Paths

BGP loop prevention encourages switch to working path.
Practical Self-Repair of Reverse Paths

BGP loop prevention encourages switch to working path.
Practical Self-Repair of Reverse Paths

BGP loop prevention encourages switch to working path.
Practical Self-Repair of Reverse Paths

BGP loop prevention encourages switch to working path.
Naive Poisoning Causes Transient Loss

- Some ISPs may have working paths that avoid problem ISP X
- Naively, poisoning causes path exploration even for these ISPs
- Path exploration causes transient loss
Naive Poisoning Causes Transient Loss

- Some ISPs may have working paths that avoid problem ISP X
- Naively, poisoning causes path exploration even for these ISPs
- Path exploration causes transient loss
Naive Poisoning Causes Transient Loss

- Some ISPs may have working paths that avoid problem ISP X
- Naively, poisoning causes path exploration even for these ISPs
- Path exploration causes transient loss
Naive Poisoning Causes Transient Loss

- Some ISPs may have working paths that avoid problem ISP X
- Naively, poisoning causes path exploration even for these ISPs
- Path exploration causes transient loss
Naive Poisoning Causes Transient Loss

- Some ISPs may have working paths that avoid problem ISP X
- Naively, poisoning causes path exploration even for these ISPs
- Path exploration causes transient loss
Naive Poisoning Causes Transient Loss

- Some ISPs may have working paths that avoid problem ISP X
- Naively, poisoning causes path exploration even for these ISPs
- Path exploration causes transient loss
Naive Poisoning Causes Transient Loss

- Some ISPs may have working paths that avoid problem ISP X
- Naively, poisoning causes path exploration even for these ISPs
- Path exploration causes transient loss
Naive Poisoning Causes Transient Loss

- Some ISPs may have working paths that avoid problem ISP X
- Naively, poisoning causes path exploration even for these ISPs
- Path exploration causes transient loss
Prepend to Reduce Path Exploration

- Most routing decisions based on:
  (1) next hop ISP
  (2) path length
- Keep these fixed to speed convergence
- Prepending prepares ISPs for later poison
Prepend to Reduce Path Exploration

- Most routing decisions based on:
  1. next hop ISP
  2. path length
- Keep these fixed to speed convergence
- Prepending prepares ISPs for later poison
Prepend to Reduce Path Exploration

- Most routing decisions based on:
  1. next hop ISP
  2. path length
- Keep these fixed to speed convergence
- Prepending prepares ISPs for later poison
Prepend to Reduce Path Exploration

- Most routing decisions based on:
  1. next hop ISP
  2. path length
- Keep these fixed to speed convergence
- Prepending prepares ISPs for later poison
Prepend to Reduce Path Exploration

- Most routing decisions based on:
  1. next hop ISP
  2. path length
- Keep these fixed to speed convergence
- Prepending prepares ISPs for later poison

Active BGP Measurement with BGP-Mux
Prepend to Reduce Path Exploration

- Most routing decisions based on:
  1. next hop ISP
  2. path length
- Keep these fixed to speed convergence
- Prepending prepares ISPs for later poison
Tested Idea Using BGP-Mux

- With no prepend, only 65% of unaffected ISPs converge instantly
- With prepping, 95% of unaffected ISPs re-converge instantly, 98%<1/2 min.
- Also speeds convergence to new paths for affected peers
Summary

- BGP-Mux lets researchers experiment with BGP in the wild
  - Transparent to experiments and stable to upstream
- Initial experiments using it:
  - LIFEGUARD: reroute around ASes or links
  - PoiRoot: root cause analysis of BGP path changes
    - Expose routing preferences
    - Induce changes to use as ground truth
  - PECAN: joint content and network routing
    - Measure performance of alternate paths
Those Three Questions

- Data sharing
  - Reverse traceroute data now online
  - Other researchers passively observed our active BGP updates
  - Use the testbed yourself
- Visualization: http://tp.gtnoise.net/
Active BGP Measurement with BGP-Mux
Conclusion

- BGP-Mux lets researchers experiment with BGP in the wild
  - Transparent to experiments and stable to upstream
  - Georgia Tech system, I am just an enthusiastic user

- **LIFEGUARD**: Let edge networks reroute around failures

- Questions for the audience:
  - What would you use this system for? What should we use it for?
  - How do we get more ASes to connect to us?
    - Getting them to agree to
    - Then, getting the connection to work
      - VLAN between BGP-Mux and border router
      - Ability to advertise BGP routes