Verifying the Correctness of Wide-Area Internet Routing

Nick Feamster and Hari Balakrishnan
M.I.T. Computer Science and Artificial Intelligence Laboratory
{feamster,hari}@csail.mit.edu

http://nms.lcs.mit.edu/bgp/
BGP is Flexible

- Many options for implementing a variety of policies
  - Route injection, redistribution, aggregation
  - Import and export policies
  - Access control lists, filtering
  - AS Path prepending
  - Communities

- Flexibility for various network environments
  - Next-hop settings
  - Route flap damping
  - Timer settings

Wonderful!
But there’s a catch...
BGP Configuration Affects Correctness

- BGP has serious problems
  - Frequently misconfigured [Mahajan2002]
  - Forwarding loops [Dube1999]
  - Persistent route oscillation [Griffin1999, Varadhan2000]
  - Slow convergence/suppressed routes [Labovitz2001, Mao2002]

*BGP’s configuration determines whether the protocol behaves correctly or not.*

*These problems never happen in the "real world", right?*
"A number of Covad customers went out from 5pm today due to, supposedly, a DDOS (distributed denial of service attack) on a key Level3 data center, which later was described as a route leak (misconfiguration)."

-- dslreports.com

"A Level 3 spokesman would not confirm or deny that hardware was the source of the problem, nor would he elaborate on the nature of the issue."

-- news.com
10 Years of NANOG...

Reported problems:

- 93 filtering issues
- 62 leaked routes
- 139 problems with route visibility
- 108 blackholes
- 23 routing loops

... 

These problems haven’t gone away.
Stimulus-response Reasoning

"What happens if I tweak this import policy?"
"Let's just readjust this IGP weight..."
"New customer attachment point? Some cut-and-paste will fix that!"

Some time later, some "strange behavior" appears. (OOPS! Revert.)

- Operators have a terrible "programming environment".
  - Configuration is ad hoc and painful.
  - Wastes operator time.
  - Suboptimal performance, angry customers.

- Can't check for errors by "seeing what happens".
  - Won't catch misconfigured filters, redundant route reflectors, etc.
Possible Remedies

- Protocol is buggy. *Replace.*
  - What to fix?
  - "BGPv5" would have to be as flexible as BGPv4.
  - Will it be any less error-prone?

- Configuration language is too "low-level". *Redesign.*
  - Again, what are the flaws in today’s configuration languages?

*We must understand the problems in BGPv4 before proposing reasonable fixes.*
Our Approach

• Develop a tool that uses static analysis to analyze router configurations.

• Operators can make BGPv4 less error-prone.
  ▶ Find configuration problems before deployment.

• We can learn from the errors we find in today’s configurations.
**Needed: Higher-level Analysis**

- **Verify** the behavior of a particular configuration.
  - Check "correctness properties".
  - Check that the configuration conforms to intended behavior.

*More than a band-aid fix.*

*Useful for any router configuration language.*
Challenges

- Defining "correctness".
- Router configuration is distributed across multiple routers and ASes.
- Limitations of static analysis?
Contributions

- Correctness constraints for BGP routing.
- Design and implementation of rcc.
- Study of configuration errors in real-world networks.
- Recommended protocol and language changes.
How to Derive Correctness Constraints?

- Need a definition of correctness.
- Need a way to apply that definition.
Properties: The Routing Logic [FDNA 2003]

- **Validity**: Does it advertise invalid routes?
  - Bogus route origination, persistent forwarding loops, etc.

- **Visibility**: Does every valid path have a route?
  - Session resets, missing sessions, damped routes, etc.

- **Information-flow control**: Expose information?
  - Accidental route leaks to neighbors, etc.

- **Determinism**: Answer depend on orderings, etc.?
  - Irrelevant route alternatives can affect outcomes.

- **Safety**: Will it converge to a unique, stable answer?
  - Policy-induced oscillation
1. **Origination**: A router "originates" a route.
Application: BGP Route Propagation

2. Export: Router advertises route to other routers.
3. **Import:** Other routers learn those routes.
Application: BGP Route Propagation

3. Import: Other routers learn those routes.
Application: BGP Route Propagation

6. Intra-AS Propagation: Propagates route within the AS.
Applying Correctness Definitions to BGP

1. Origination: A router "originates" a route.
2. Export: Router advertises route to other routers.
3. Import: Other routers learn those routes.
6. Propagation: Propagates route within the AS.
Putting it together

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Origination</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Export</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>3. Import</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>4. Selection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>5. Modification</td>
<td>•</td>
<td></td>
<td></td>
<td></td>
<td>•</td>
</tr>
<tr>
<td>6. Intra-AS Prop.</td>
<td>•</td>
<td>•</td>
<td></td>
<td></td>
<td>•</td>
</tr>
</tbody>
</table>

- Determine which aspects of correctness apply at each stage of BGP’s operation.
- Express constraints.
- Try to test constraints with static analysis.
Example: Validity

• Incorrect Origin AS (Origination)

• Incorrect AS Path (Export)

• Incorrect or Missing Filters (Export/Import)

• Incorrect "next-hop" attribute (Modification)

• Intra-AS Inconsistencies (Intra-AS Propagation)
Validity: Incorrect "next-hop" attribute

- One necessary, commonly violated condition: next-hop reachability

Routes from AS 1 have next-hop e.f.g.h. If e.f.g.h not injected into IGP, some routes from within AS will fail.
Example: Visibility

- Failure to install valid routes (Import)
- iBGP Signaling (Intra-AS Propagation)
Visibility: iBGP Signaling Overview

- Default: don’t readvertise iBGP-learned routes
  - Complete propagation requires "full-mesh" iBGP.
  - Doesn’t scale.

- "Route reflection" improves scaling (RFC 2796)
  - **Client**: re-advertise as usual
  - **Route reflector**: reflect non-client routes to all clients, client routes to non-clients and other clients.
Visibility: iBGP Signaling

Route Reflector (RR)

W

Client

X

Y

Z

RR

Client
Visibility: iBGP Signaling

route $r_2$ to $d$
Visibility: iBGP Signaling

Route Reflector (RR)

route \( r_1 \) to \( d \)

\( W \)

\( X \)

\( Y \)

\( Z \)

route \( r_2 \) to \( d \)

iBGP Signaling Partition!
Theorem.

Suppose the iBGP reflector-client relationship graph contains no cycles.

Then, the AS’s configuration satisfies visibility if, and only if, the set of routers that are not route reflector clients forms a full mesh.

*Condition is easy to check with static analysis.*
Verification requires a specification of intended policy. (We don’t have this today, but we can make reasonable assumptions.)

- Controlled export (Export)
- Consistent export (Export)
- Consistent import (Import)

These conditions are difficult to "eyeball" in practice, but easy to check with static analysis.
Information-flow Control: Controlled Export

Simple rule: don’t advertise routes from one peer to other peers.

A \xrightarrow{\text{Peer}} B \xleftarrow{\text{Peer}} C

“Announce p”
Bad: Specifying Policy with Mechanism

```plaintext
neighbor 10.0.0.1 route-map EXPORT-A out
neighbor 192.168.0.1 route-map IMPORT-C in
  ip community-list 1 permit 0:1000
route-map IMPORT-C permit 10
  set community 0:1000
!
route-map EXPORT-A permit 10
  match community 1
!
```

"Announce p"
Common practice: make routes to neighboring peers look "equally good".

If AS 1 and AS 2 are peers, the AS path length and MED on both of these sessions should normally be identical.

Again, much easier with static analysis.
Where Static Analysis is Less Helpful

- **Determinism:** Persistent route oscillation
- **Safety:** Policy disputes [Griffin 2001]

We suggest protocol modifications to make BGP more verifiably correct.
Determinism: Persistent route oscillation

AS 2

A, B, C => A, B, C

AS 3

A, B => B
A, B, C => A
A > C > B

Cisco 7500 Series

Cisco Systems

 Ethernet 1000 Series

Cisco 7600 Series
Determinism: Persistent route oscillation

AS 2

A

B

AS 3

C

A, B => B
A, B, C => A
A > C > B
Determinism: Persistent route oscillation

A,B,C => A
A,B => B
A,C => B

Cisco Systems Cisco 7500 Series
Determinism: Persistent route oscillation

A, B, C => A
A, B => B
A, B, C => A
A > C > B

Cisco Systems Cisco 7500 Series
Determinism: Persistent route oscillation

A, B, C => A
A, B => B
A, C > B

AS 2

AS 3

A, B => B
A, B, C => A
A > C > B

Cisco Systems Cisco 7500 Series
Determinism: Persistent route oscillation

AS 2

A

B

C

AS 3

A, B => B
A, B, C => A
A > C > B

Static analysis is no help.
rcc Overview

- Expand macros, etc.
- Generate intermediate format
- Verify using queries against intermediate format

Extensible design.
Errors Happen

- **Serious Errors (1st Class)**
  - Incorrect or missing filters (~ 50 sessions)
  - iBGP signaling partitions (10 instances)
  - Unintentional transit (3 instances)

- **Annoyances (2nd Class)**
  - Inconsistent export (3 instances)
  - Nondeterministic settings (34 routers)
  - Failure to install valid routes (3 routers)

- **Cleanup (3rd Class)**
  - Sessions with undefined policies (2 sessions)
  - Policies with undefined distribute lists, etc. (30 policies)
  - Incomplete iBGP sessions (76 sessions)
Errors in Every AS

- prefix adv. w/o route
- incomplete iBGP sessions
- policy w/undefined ACL
- policy w/undefined community
- policy w/undefined AS path
- session w/undefined policy
- router w/synchronization
- nondeterministic tiebreak
- router w/o determ. med
- inconsistent export to peer
- unintentional transit
- duplicate loopbacks
- RR cluster partition
- route reflector partition
- non-RR iBGP partition
- missing prefix in filters
- session w/undefined filters
- eBGP session w/no filters

Number of ASes

- First-class
  - (5)
  - (2)
  - (5)
  - (2)
  - (2)
- Second-class
  - (2)
  - (2)
  - (2)
- Third-class
  - (5)
  - (2)
  - (2)

Number of ASes: 0 1 2 3 4 5 6 7 8
Curiosities

- Historical relationships between ASes.
  "I don’t know what the status of that relationship is these days. Perhaps it is still active---at least in the configs!"

- Inbound AS path prepending.

- Intentional cold-potato routing.

- Next-hop settings.
Why are these errors happening?

• Ad hoc process
  ▶ **Example:** Filtering is rarely (if ever) done correctly.
  ▶ **Solution:** Automation; build validity into BGP (e.g., S-BGP).
"The real solution to this problem is to make it possible for ISPs to closely track RIR allocations in their filters in a semi-automated way. There may still be a few days of delay before a new allocation is fully routable but ISPs can compensate for that with internal processes."

-- Michael Dillon, NANOG, 2/24/2004

"...all this bogon or related filtering is not a long-term solution. We need it now, but the long term solution is some kind of authentication that will allow only the rightful owner of a block to announce it."

-- Michael Py, NANOG, 2/24/2004
Why are these errors happening?

- **Ad hoc process**
  - *Example*: Filtering is rarely (if ever) done correctly.
  - *Solution*: Automation; build validity into BGP (e.g., S-BGP).

- **Obscure mechanisms**
  - *Example*: iBGP signaling partitions
  - *Solution*: Redesign iBGP

- **Indirect specification**
  - *Example*: Incorrect implementation of information flow policies
  - *Solution*: Better configuration languages
Ongoing and Future Work

- Protocol design work
  - intra-AS route propagation
  - policy/protocol restrictions to guarantee safety on fast timescales

- Constraint specification is not easy (yet).
  - *Idea*: statistical beliefs of "correctness"

- Verifying constraints across multiple ASes.

- Towards *intent-based* configuration languages.
  - Figuring out how to express operator intent.
  - Operator should specify *intended goals*, not the mechanism.
Conclusion

- BGP needs systematic verification techniques, regardless of configuration language.

- Our contributions:
  - Correctness constraints for configuration.
  - Design and implementation of rcc.
  - Study of configuration errors in real-world networks.
  - Recommended protocol and language changes.

- Early version of rcc is available.
  - More than 30 operators have downloaded the tool.
  - Tested configurations of 9 ASes (and counting).

http://nms.lcs.mit.edu/bgp/