xBGP: Faster Innovation in Routing Protocols

Thomas Wirtgen, Tom Rousseaux, Quentin De Coninck, Nicolas Rybowski, Randy Bush, Laurent Vanbever, Axel Legay, Olivier Bonaventure
Agenda

- Why bringing programmability to BGP?
- Inside xBGP
- Does using xBGP have an impact on router performances?
- Verifying xBGP extensions
- Conclusion
Routing on the Internet
Routing on the Internet

It’s a best practice to have routers from different vendors for stability & economical reason.
Routing on the Internet

It’s a best practice to have routers from different vendors for stability & economical reason.

All routers do not implement the same set of functionalities.
Networks are rapidly evolving

Operators constantly tune their networks

But they are limited:
1. By the Network OS interface (blackbox)
2. By the Standards (BGP + extensions)
Enhancing the visibility of the BGP control plane

Intra domain routers have no information about the exit router
Enhancing the visibility of the BGP control plane

Intra domain routers have no information about the exit router

I prefer European routes
Enhancing the visibility of the BGP control plane

Intra domain routers have no information about the exit router

I prefer European routes
Enhancing the visibility of the BGP control plane

Intra domain routers have no information about the exit router

I prefer European routes
Enhancing the visibility of the BGP control plane

Intra domain routers have no information about the exit router

I prefer European routes
A complex feature to achieve with classical routers

The Geographical Location TLV (GeoLoc TLV)

BGP Path Record Attribute: draft-raszuk-idr-bgp-pr-05
The Geographical Location TLV (GeoLoc TLV)

A complex feature to achieve with classical routers

BGP UPDATE a.b.c.0/24 via AS2

Add GeoLoc on the input edge routers

BGP Path Record Attribute: draft-raszuk-idr-bgp-pr-05
A complex feature to achieve with classical routers

The Geographical Location TLV (GeoLoc TLV)

1. Add GeoLoc on the input edge routers
2. Spread the GeoLoc inside the IGP

BGP UPDATE a.b.c.0/24 via AS2

BGP Path Record Attribute: draft-raszuk-idr-bgp-pr-05
A complex feature to achieve with classical routers

The Geographical Location TLV (GeoLoc TLV)

1. Add GeoLoc on the input edge routers
2. Spread the GeoLoc inside the IGP
3. Remove GeoLoc on the output edge routers

BGP Path Record Attribute: draft-raszuk-idr-bgp-pr-05
A complex feature to achieve with classical routers

The Geographical Location TLV (GeoLoc TLV)

1. Add GeoLoc on the input edge routers
2. Spread the GeoLoc inside the IGP
3. Remove GeoLoc on the output edge routers

⇒ This requires to have access to the router’s BGP implementation

BGP Path Record Attribute: draft-raszuk-idr-bgp-pr-05
All that remains is to ship the feature...

One does not simply ask to your routers vendor...

1. Standardisation of the new feature by the IETF
   (3.5 years in average for BGP & confirmed by another study [1])
2. Implementation on the vendor OSes
3. Update your routers

All that remains is to ship the feature...

One does not simply ask to your routers vendor...

1. Standardisation of the new feature by the IETF (3.5 years in average for BGP)
2. Implementation on the vendor OSes
3. Update your routers

You can not easily influence steps 1 and 2!

I would like to propose a new feature

We will think about how to standardize it if it adds value

Can you please update the router OS?

You don’t have the required support licence to ask us this
Current paradigm slows innovation

Problem #1: Routers from different vendors

Problem #2: Protocol extensions not implemented on all routers

Problem #3: Slow upgrade process

⇒ xBGP is designed to bring innovation & programmability to existing routing protocols
Why bringing programmability to BGP?

Inside xBGP

Does using xBGP have an impact on router performances?

Verifying xBGP extensions

Conclusion
GeoLoc needs to alter the BGP Workflow

RFC 4271 BGP Workflow
GeoLoc needs to alter the BGP Workflow

BGP Messages
From Peers

BGP Control Plane

Data Plane

FIB

RFC 4271 BGP Workflow
GeoLoc needs to alter the BGP Workflow

RFC 4271 BGP Workflow
GeoLoc needs to alter the BGP Workflow

RFC 4271 BGP Workflow
GeoLoc needs to alter the BGP Workflow

RFC 4271 BGP Workflow
GeoLoc needs to alter the BGP Workflow

RFC 4271 BGP Workflow
GeoLoc needs to alter the BGP Workflow

RFC 4271 BGP Workflow
GeoLoc needs to alter the BGP Workflow

RFC 4271 BGP Workflow
GeoLoc needs to alter the BGP Workflow

RFC 4271 BGP Workflow
Traditional BGP implementations are opaque.
BGP workflow are now exposed with xBGP
GeoLoc needs to alter the BGP Workflow

RFC 4271 BGP Workflow
GeoLoc needs to alter the BGP Workflow

RFC 4271 BGP Workflow
GeoLoc needs to alter the BGP Workflow

1. BGP Messages From Peers
   - Adj-RIB-IN
   - Import Filters

2. Loc-RIB
   - BGP Decision Process

3. BGP Control Plane
   - Data Plane
   - Export Filters
   - Adj-RIB-OUT

4. FIB

5. BGP Messages To Peers
   - Adj-RIB-IN
   - Import Filters

RFC 4271 BGP Workflow
GeoLoc needs to alter the BGP Workflow

RFC 4271 BGP Workflow

GeoLoc Ext.

---

1. Decode GeoLoc
2. Add GeoLoc
3. Use GeoLoc
4. Delete GeoLoc
5. Encode GeoLoc

BGP Messages From Peers

BGP Messages To Peers

BGP Decision Process

FIB

VM

Adj-RIB-IN

Loc-RIB

Adj-RIB-OUT

Import Filters

Export Filters

BGP Control Plane

Data Plane
xBGP: a paradigm shift

Operators can now add extension codes to their routers
Operators can now add extension codes to their routers.

Routers execute the GeoLoc code with an eBPF based VM.
xBGP makes the link between Router and extensions

Provided by operators

xBGP Programs

Provided by our paper

Network OS Router

Provided by vendors

xBGP

RIB  Peer States  Memory  Network IO  BGP State
xBGP makes the link between Router and extensions

Provided by operators

xBGP Programs

Provided by our paper

Network OS Router

Provided by vendors

xBGP Programs

Provided by operators

xBGP Programs

Provided by our paper

Network OS Router

Provided by vendors

xBGP Programs
Demonstrating the programmability of xBGP

xBGP requires a little adaptation to the host BGP implementation.

We have adapted both FRRouting and BIRD to be xBGP compliant

<table>
<thead>
<tr>
<th></th>
<th>FRRouting (LoC)</th>
<th>BIRD Routing (LoC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modification to the codebase</td>
<td>30</td>
<td>10</td>
</tr>
<tr>
<td>Building Insertion Points</td>
<td>73</td>
<td>66</td>
</tr>
<tr>
<td>Plugin API</td>
<td>624</td>
<td>415</td>
</tr>
<tr>
<td>libxbgp</td>
<td>3004 + dependencies</td>
<td></td>
</tr>
<tr>
<td>User Space eBPF VM</td>
<td>2776</td>
<td></td>
</tr>
</tbody>
</table>

https://www.pluginized-protocols.org/xbgp
Other use cases

<table>
<thead>
<tr>
<th>xBGP Extension</th>
<th>LoC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geographical Location</td>
<td>388</td>
</tr>
<tr>
<td>Valley free routes</td>
<td>143</td>
</tr>
<tr>
<td>Filtering routes by IGP cost</td>
<td>36</td>
</tr>
<tr>
<td>Scanning for BGP zombies</td>
<td>1071</td>
</tr>
<tr>
<td>Influence remote BGP Decision Process</td>
<td>62</td>
</tr>
<tr>
<td>Monitoring the routes propagation time</td>
<td>806</td>
</tr>
</tbody>
</table>

⇒ Check the paper for those use cases
Agenda

- Why bringing programmability to BGP?
- Inside xBGP
- **Does using xBGP have an impact on router performances?**
- Verifying xBGP extensions
- Conclusion
Using a Virtual Machine inside BGP

Use Case

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Convergence Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>xFRR</td>
<td>xBIRD</td>
</tr>
</tbody>
</table>

Full IPv4 & IPv6 routing table (1M routes) - xBGP router - Downstream
Using a Virtual Machine inside BGP

Full IPv4 & IPv6 routing table (1M routes)

xBGP router

Downstream

Additional overhead due to the xBGP internals

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Convergence Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>xFR</td>
<td>+1.05%</td>
</tr>
<tr>
<td>xBIRD</td>
<td>+1.60%</td>
</tr>
</tbody>
</table>
Using a Virtual Machine inside BGP

Full IPv4 & IPv6 routing table (1M routes)

xBGP router

Downstream

Additional overhead due to the xBGP internals

Worst case involving all insertion points

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Convergence Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>No xBGP program</td>
<td>xFRR: +1.05%</td>
</tr>
<tr>
<td></td>
<td>xBIRD: +1.60%</td>
</tr>
<tr>
<td>Route reflection</td>
<td>xFRR: +12.97%</td>
</tr>
<tr>
<td></td>
<td>xBIRD: +7.43%</td>
</tr>
</tbody>
</table>
Using a Virtual Machine inside BGP

Additional overhead due to the xBGP internals

Worst case involving all insertion points

Data serialization is more costly in FRR

The “JIT” compiler is not efficient as native machine code

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Convergence Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>xFRR</td>
<td>xBIRD</td>
</tr>
<tr>
<td>No xBGP program</td>
<td>+1.05%</td>
</tr>
<tr>
<td>Route reflection</td>
<td>+12.97%</td>
</tr>
</tbody>
</table>
Agenda

- Why bringing programmability to BGP?
- Inside xBGP
- Does using xBGP have an impact on router performances?
- **Verifying xBGP extensions**
- Conclusion
The code executed by xBGP is **untrusted**

The code should be annotated, and then passed to the verification tools.
The right tool to the right property

- **T2**: termination
- **CBMC**: memory safety
- **libxbgp**: VM isolation & API restriction
- **Seahorn**: BGP properties
Verifying the BGP syntax of GeoLoc

If the xBGP extension adds Geographic coordinates, it must respect the TLV format defined in the draft.

```
|  Attr. Flags  |Attr. Type Code|Attr. Length (8 or 16 bits) |Attr. Length (8 or 16 bits) |
+-----------------+-----------------+-----------------+-----------------|
| latitude (64 bits) | longitude (64 bits) |
```
Verifying the BGP syntax of GeoLoc

If the xBGP extension adds Geographic coordinates, it must respect the TLV format defined in the draft.

<table>
<thead>
<tr>
<th>Flags</th>
<th>Attr. Flags</th>
<th>Attr. Type Code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Flags MUST be 0x8</td>
<td>Code MUST be 0x2a</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Length</th>
<th>Attr. Length (8 or 16 bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length MUST be 8</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Latitude</th>
<th>Latitude (64 bits)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Longitude</th>
<th>Longitude (64 bits)</th>
</tr>
</thead>
</table>

<Latitude, Longitude> MUST be a valid geographic location
Conclusion

With xBGP, BGP implementations can become truly extensible.

See [https://www.pluginized-protocols.org/xbgp](https://www.pluginized-protocols.org/xbgp) for running source code.

xBGP provides new opportunities with other routing protocols.

[pluginized-protocols.org](https://pluginized-protocols.org)

[thomas.wirtgen@uclouvain.be](mailto:thomas.wirtgen@uclouvain.be)