Scaling AMS-IX Route Servers

David Garay

Supervisor: Stavros Konstantaras
Research Project 2, 2019
Motivation: Security

Zwitsers bedrijf routeerde KPN- en ander Europees verkeer via China Telecom

De Zwitserse colocationaanbieder Safe Host heeft donderdag twee uur lang verkeer van meerdere Europese telecomaanbieders per ongeluk omgeleid via het netwerk van China Telecom. Onder andere KPN-verkeer verliep via de omweg.

Het artikel beschrijft hoe Safe Host in het Duitse Frankfurt meer dan zeventigduizend routes via China Telecom liet verlopen. Daarbij ging het om dertienhonderd Nederlandse prefixes, waarvan veel van KPN. De omlaag begon voor KPN donderdagochtend rond 10.00, blijkens een grafiek van Oracle op basis van BGP Data. Na twee uur had Safe Host het probleem opgelost.

Donderdag bleek al dat KPN-klanten met internetproblemen te maken kregen. Ook zijn er ongewenste hinder van de storing. toen al gaven onder andere gebruikers op Gathering of Tweakers aan dat er sprake leek van bgp-hijacking. In een reactie zei KPN toen dat de fout lag bij een configuratiewijziging in het netwerk van een transitpartij in Zwitserland.

June 6, 2019

Large European Routing Leak Sends Traffic Through China Telecom

Beginning at 09:43 UTC today (6 June 2019), Swiss data center colocation company AS21217 leaked over 70,000 routes to China Telecom (AS4134) in Frankfurt, Germany. China Telecom then announced these routes on to the global internet redirecting large amounts of internet traffic destined for some of the largest European mobile networks through China Telecom’s network. Impacts were seen by some of Europe’s largest networks in Switzerland, Holland, and France among other countries.

... 4134 21217 21217 21217 21217 21217 21217 21217 13237 1136
### Motivation: Scalability

<table>
<thead>
<tr>
<th>IXP</th>
<th>Clients</th>
<th>Connected to Route Server</th>
<th>Update frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMS-IX (^1)</td>
<td>845</td>
<td>714</td>
<td>1 hour</td>
</tr>
<tr>
<td>DE-CX (^2,5) (Frankfurt)</td>
<td>870</td>
<td>846</td>
<td>6 hours</td>
</tr>
<tr>
<td>LINX (^2) (London)</td>
<td>819</td>
<td>640</td>
<td>At least 3 hours (^4)</td>
</tr>
</tbody>
</table>

* IPv4 only

**Security requires dynamic configuration capabilities**
Background Information

- Central point for exchange of network prefixes, alternative to full-mesh topology.
- It filters prefixes exchanged, following policies configured by network operators.
- A route server is not a route reflector.

Fig 1: What is a Route Server?
Background Information

Policies are periodically updated with dynamic data:

- **Internet Routing Registry DB:** source for whois information. Stores data using the *Routing Policy Specification Language* (RPSL).
- **Resource Public Key Infrastructure:** establishes the legitimacy of a prefix/autonomous system number (ASN) pairing.
- **Team Cymru:** maintains the bogon reference.

Fig 2: Data sources for a Route Server
Research Questions

● With regards to the route server’s policy update process, what are the performance and scalability performance indicators? And what are the bottlenecks of the process, and what is their impact?
  ○ How can we improve these indicators in a new, feasible design?
Related Research

**Problem Characterisation:**
Jenda Brands and Patrick de Niet looked at BGP Parallelization, as a way to overcome the CPU bottlenecks which cause long converge times, present in Route Servers BGP implementations.

**Solution Design:**
Gregor Hohpe present patterns in Enterprise Integration Patterns that help designing messaging systems.
Methodology

- Current utilization
- Current setup evaluation and experiment design.
  - What are the bottlenecks and their impact?
- Solution design
Utilization in the last 6 months

- With the help of RIPE’s STATs, we count every time an object `aut-num` and `route` change, and aggregate them per hour.
- Note: not every policy change and route/prefixes is relevant to our IXP.
- Only AMS-IX clients, and prefixes in the route servers where used.

Fig 3: Number of changes per hour of relevant objects
Utilization in the last 6 months

How often are relevant changes happening?

- Dimensioning decision based on monthly averages or peaks?

Fig 4: Number of changes per hour of relevant objects
We monitored the effects of policy updates on CPU, memory and traffic. We designed three experiments:

- Route server reconfigurations with different file sizes;
- Route server reconfigurations, where BGP updates were triggered;
- Route server peering with a large number of peers (>1100).
## Results

<table>
<thead>
<tr>
<th>Experiments</th>
<th>Result</th>
<th>Tooling / Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reconfiguration time as result of file size</td>
<td>~0.3s per 10MB file size increase</td>
<td>ars issue #48</td>
</tr>
<tr>
<td>Reconfiguration time as result of BGP update traffic</td>
<td>~ 0.5s per additional peer</td>
<td></td>
</tr>
<tr>
<td>CPU utilization as result of the number of peers</td>
<td>Crash at 1013 peers in our setup</td>
<td>Ulimit configuration - insufficient system resources.</td>
</tr>
</tbody>
</table>
Reconfiguration time vs Number of Peers

Fig 7: Reconfiguration time vs number of peers sending BPG updates as result of policy change, contribution per peer
Summary of challenges

- Policy updates are **not applied in real-time**.
- Updates cause high CPU utilization, **blocking** the Route Server to new tasks.
  - If moving to a information Push model, route server might be busy.
- Network load increase as result of updates
Application Integration Alternatives

Data Transfer:
File Transfer and Shared Database.

Disadvantages: stale data, or if polling in use, inefficient use of resources.

Invoke remote functionality:
Remote Procedure Invocation (RPI) and Messaging.

Fig 8: Integration alternatives
Application Integration Alternatives

With RPI, we have up to \textbf{NxM} IXPs and ASNs, simultaneous processes at the data source.

- Addressing, failures and performance are not transparent.

- \textbf{Messaging} offers loose-coupling asynchronous communications.

Fig 8: Integration alternatives
With a Messaging system, broadcast of messages is more efficiently.

- In a **Publish-Subscriber** channel, clients receive real-time notifications about topics they have subscribed to.

- In our example, when AS65020 changes its policy, interested IXPs can receive it immediately.

- Messages remain in the system until consumed, or expire.
Proposed design: New functionalities

Fig 10: Sequence diagram - Policy updates push model

Modifications required:

- Message Gateway.
- Messaging system.
Example: Google PubSub

Fig 11: Messaging system example (left) and client (right)
To receive policy change notifications, a client subscribes to the topic of the respective ASN.

- Transport options depend on Messaging System implementation, and message format remain RPSL to leverage existing tools.

Fig 12: Sequence diagram - Policy updates push model
Proposed design: Policy updates procedure

Notifications are received in real-time.

- Duplicated messages policy, throttling and parallelization are handled at the client’s Messaging Gateway.

Fig 13: Sequence diagram - Policy updates push model
Architecture Vision

Fig 14: Architecture vision
Discussion

● Design
  ○ Does it address the real-time and throttling requirements?
  ○ Is the design future proof?
  ○ Is there justification for a Message System?

● Limitations in our methodology
  ○ Limited use cases evaluated
  ○ Validation against production statistics, simulation in scale.
In our experiments, we found that the route server **blocks** as result of policy updates. The blocking time depends on the file size and on the amount of peers undergoing BGP Update procedures.

We propose a **messaging** based design which addresses the lack of real-time policy updates, we discuss the component required and discuss how **throttling and queueing** can help alleviate the impact of the BGP policy updates.

Our statistics regarding rate of policy updates are limited in the amount of objects monitored, and we recommend IXPs to **perform measurements in production** on policy changes to assess their impact on the network.
Future Work

- Improve Bird’s reconfiguration efficiency by evaluating Binary configuration formats
- Study other use cases (e.g. Policy implementation feedback)
- Extend statistical investigation to include IPv6 objects, and other objects.
Backup
Reconfiguration time vs Number of Peers

Fig 7: Reconfiguration time vs number of peers sending BPG updates
Erlang B: 28 arrivals, ~16s processing, 1 server

source
Utilization in the last 6 months

Where are the events coming from?

These are the percentage of networks doing 0-100 changes, 101-200...; in the last 6 months.

- Most relevant events come from few network operators.

Fig 4: Frequency of changes, in ranges of 100, in the last 6 months
Who is using arouteserver?

- BharatIX, BIRD.
- CATNIX, BIRD.
- IX-Denver, BIRD.
- MBIX, BIRD.
- PIT-IX, BIRD.
- SwissIX, OpenBGPD.
- Unmetered.Exchange, BIRD.
- VANIX.
- YXEIX, BIRD.
- YYCIX, OpenBGPD.

Are you using it? Do you want to be listed here? Drop me a message!

Fig: Frequency of changes, in ranges of 100, in the last 6 months
Reconfiguration time vs File size

Fig 6: Reconfiguration time vs file size