Automated Inter-AS Traffic Engineering: An open source approach and operational considerations

Kostas Zorbadelos - Lead Network Architect

October 27, 2022

CANAL+ Telecom

https://www.peeringdb.com/asn/21351

Outline

Problem Statement & Design Goals

An open source automation tool for BGP Traffic Engineering

Considerations using on-demand BGP announcements

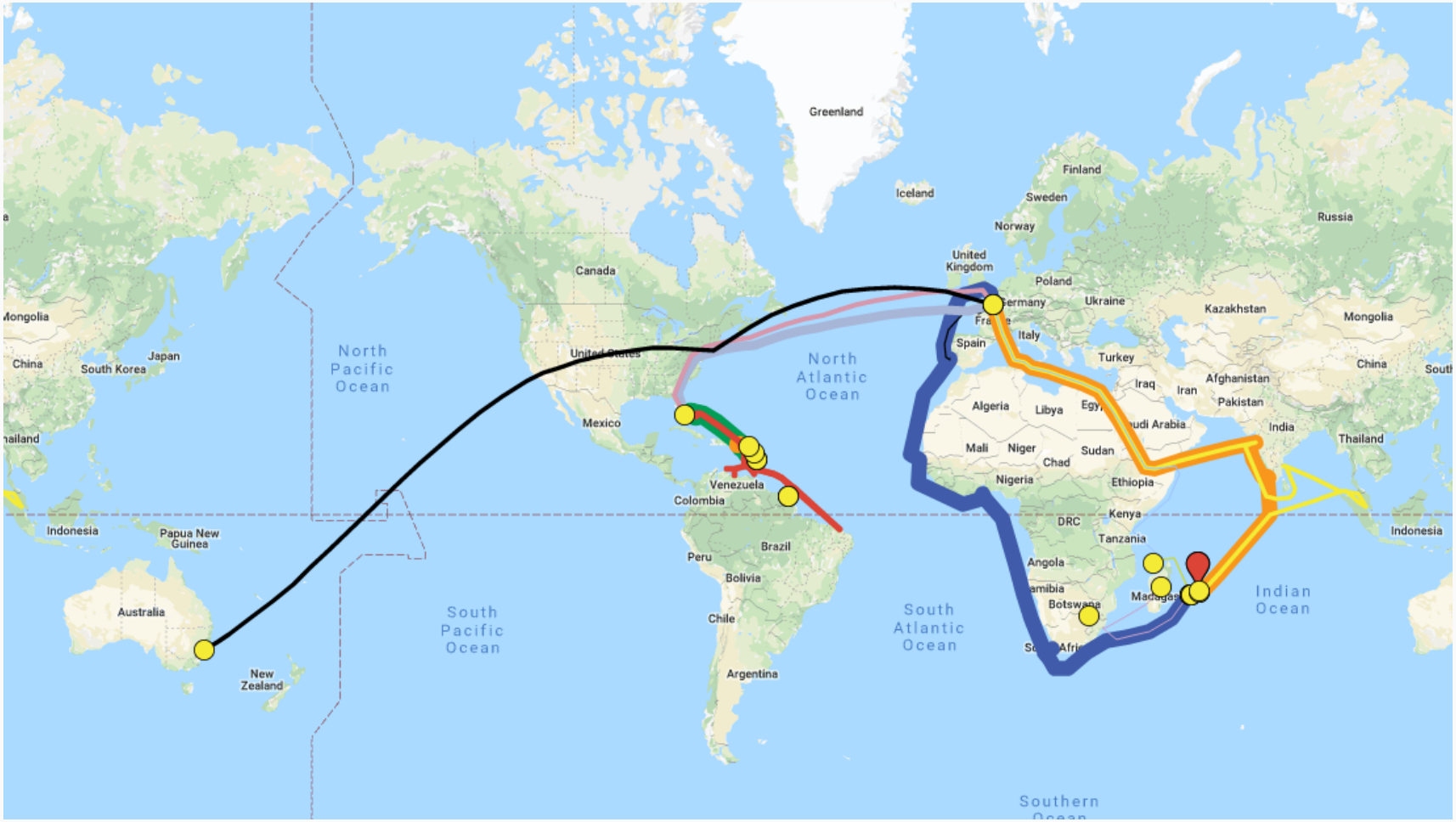
Problem Statement & Design Goals

Problem Description

- IP Network with multiple points of presense / geographically dispersed
- Service Provider network with customer transit services
- Having multiple Internet transit providers and peers (in IXes or PNIs)
- Varying costs in transit capacity, submarine capacity can also be involved
- In traditional ISP networks downstream traffic is dominant

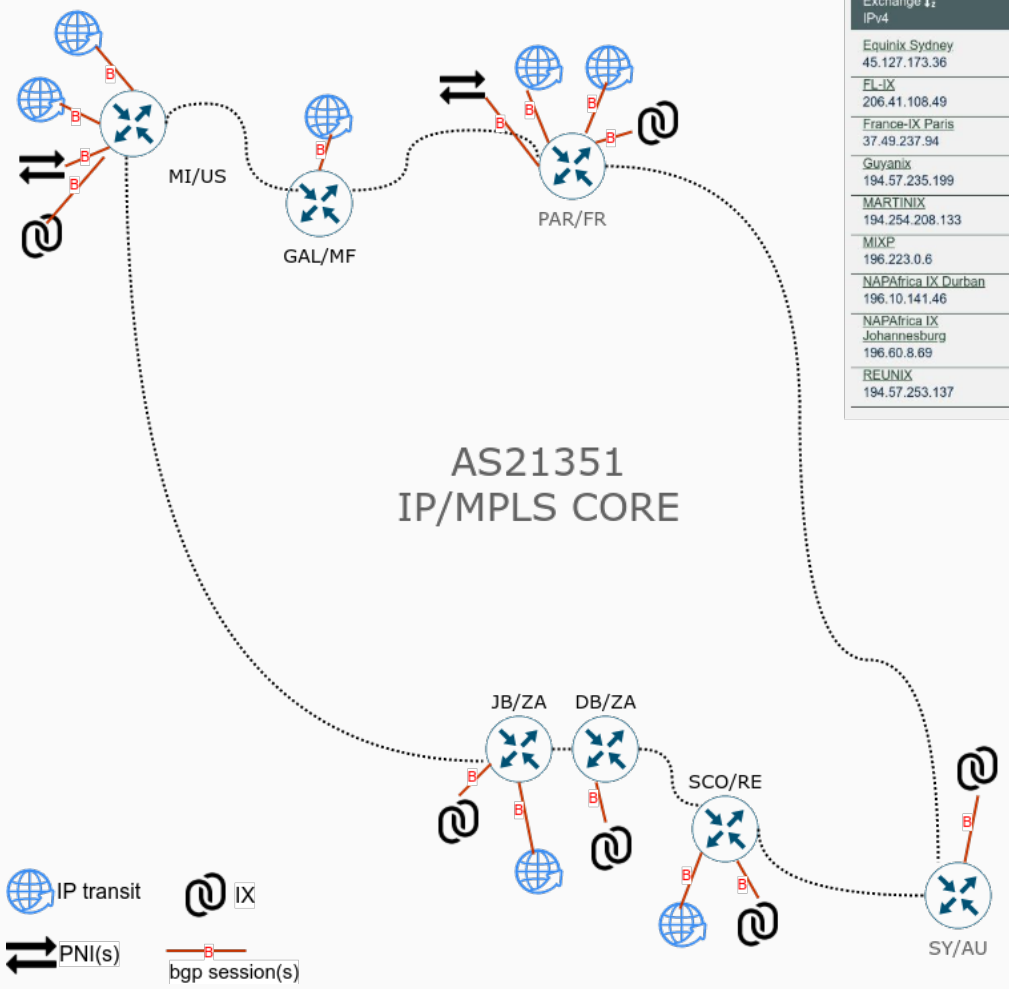
Need to optimize incoming traffic streams and distribute them among available capacity. Also need to divert traffic on-demand for security reasons (eg DDoS attacks).

IP Network - Geolocations



CANAL+ Telecom (AS21351) geolocations

AS21351 - Border / Peering Points



Public Peering Exchange Points

Filter

Exchange IPv4	ASN IPv6	Speed	RS Peer
Equinix Sydney 45.127.173.36	21351 2001:de8:6::2:1351:1	10G	✓
FL-IX 206.41.108.49	21351 2001:504:40:108::1:49	40G	✓
France-IX Paris 37.49.237.94	21351 2001:7f8:54::1:94	20G	✓
Guyanix 194.57.235.199	21351	100M	○
MARTINIX 194.254.208.133	21351	100M	○
MIXP 196.223.0.6	21351 2001:43f8:270:d0d0::6	1G	✓
NAPAfrica IX Durban 196.10.141.46	21351 2001:43f8:6d2::46	40G	○
NAPAfrica IX Johannesburg 196.60.8.69	21351 2001:43f8:6d0::69	60G	✓
REUNIX 194.57.253.137	21351	1G	○

Private Peering Facilities

Filter

Facility ASN	Country City
Equinix MI1 - Miami, NOTA 21351	United States of America Miami
Equinix SY1/SY2 - Sydney 21351	Australia Sydney
GNU Cayenne Collery 21351	French Guiana Cayenne
Interxion Paris 1 (Aubervilliers Cedex) 21351	France Paris
Interxion Paris 2 (Aubervilliers) 21351	France Paris
Interxion Paris 5 (St Denis) 21351	France Paris
MCN Le Lamentin 21351	Martinique Le Lamentin
Telehouse - Paris 2 (Voltaire - Léon Frot) 21351	France Paris
Teraco Durban, South Africa 21351	South Africa Durban
Teraco Johannesburg Campus, South Africa 21351	South Africa Johannesburg

Design Goals

- Manual configuration on routers is cumbersome
- Inconsistent configuration, error-prone
- Routers involved could be many, fast reaction not possible
- Configuration could be performed by network operators or even a program without human involvement
- Ideally vendor neutral (multiple vendor equipment in many networks)

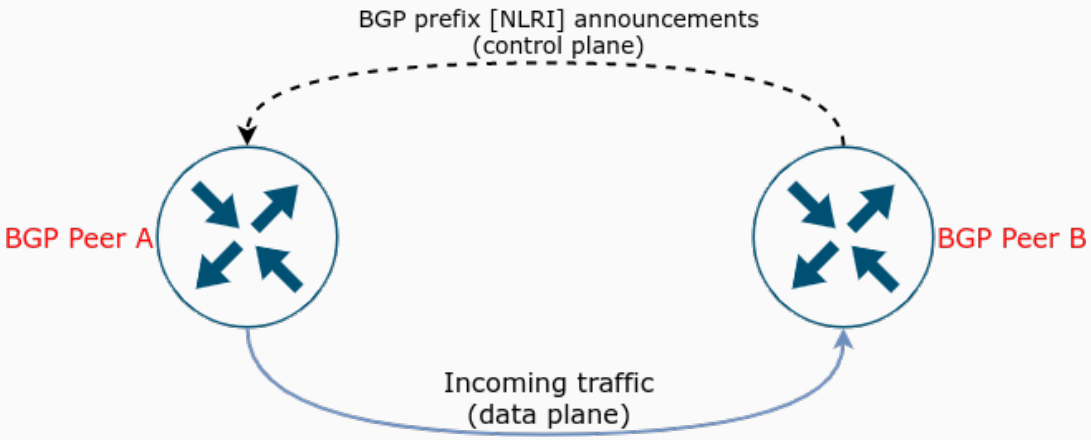
Design Goals (continued)

- Do the traffic engineering reliably, without errors
- Easy to operate
- Do it quickly, even real time, depending on current traffic conditions or security incidents
- A link failure (especially in submarine capacity) would need proper action to bypass the failure
- Optimize economics (in transit services, capacities or peerings)
- Build a tool on a solid foundation that can grow in features
- Provide automation / scripting capabilities (future “self driving network”)

An open source automation tool for BGP Traffic Engineering

Inbound Inter-AS TE

- BGP is the exterior routing protocol between ASes



- BGP is used extensively for traffic engineering with various tricks (hacks?)
- Tool's purpose is to automate BGP announcements to peers, to affect in-bound traffic flow

Solution design

- Centralized configuration point
- Sources of truth, representing the *intended* state (peerings and announcements)
- Standardized BGP policy configuration generated by automation tools (OUT-bound policies)
- Tagging of prefixes (BGP communities) affects policy
- Only need to think (or generate) the proper tags in the routes to get the desired outcome
- Design flexibility, all traffic engineering tricks should be supported

BGP Large Communities

- RFC 8092 BGP large communities [3] a "recent" development (2017)
- 12 octets, three 4-byte integers (example `21351:602:6799`)
- Overcome policy design limitations with 32-bit ASNs
- RFC 8195 Use of BGP Large Communities [8], informational RFC giving excellent policy examples
- Informational and action communities
- An IETF "blessed" way to create policies!
- Our design was based on this

Informational vs action communities

- Following RFC 8195 paradigm, second number in the large community is a field that contains a *function* identifier
- Informational communities are labels for various attributes
- Action Communities are added as labels to request that a route be treated in a particular way within an AS

```
*Informational communities example*
<ASN>:3:<TYPE_OF_ROUTE>
Contains the type of a route (eg internal loopback,
internal b2b customer, transit customer route,
BGP announcement)

*Action community example*
<ASN>:40:<PEER_ASN>
*Do not* announce a route to a peer ASN
```

Supported TE actions

Action	Large community pattern
NO_ANNOUNCE_ANY_PEER	<LocalASN>:40:0
NO_ANNOUNCE_PEER	<LocalASN>:40:<PeerASN>
ANNOUNCE_PEER	<LocalASN>:41:<PeerASN>
PREPENDx[N]_PEER	<LocalASN>:6[N]:<PeerASN>
NO_ANNOUNCE_ANY_LOCATION	<LocalASN>:400:0
NO_ANNOUNCE_LOCATION	<LocalASN>:400:<LocationCode>
PREPENDx[N]_LOCATION	<LocalASN>:60[N]:<LocationCode>

Tool implementation - Sources of Truth

We utilize `NETBOX` [6] as an IPAM system. Each prefix announcement is tagged accordingly, using BGP large communities. `NETBOX` contains the intended state of all the announcements of our AS (prefixes and policy for them).

We utilize `Peering Manager` [5] to hold all the information regarding eBGP peerings with transit providers and peers. We document both PNIs and peerings via Internet Exchanges. From this the configuration management engine generates configuration for the peerings plus the standardized OUT-bound policies.

Configuration management engine

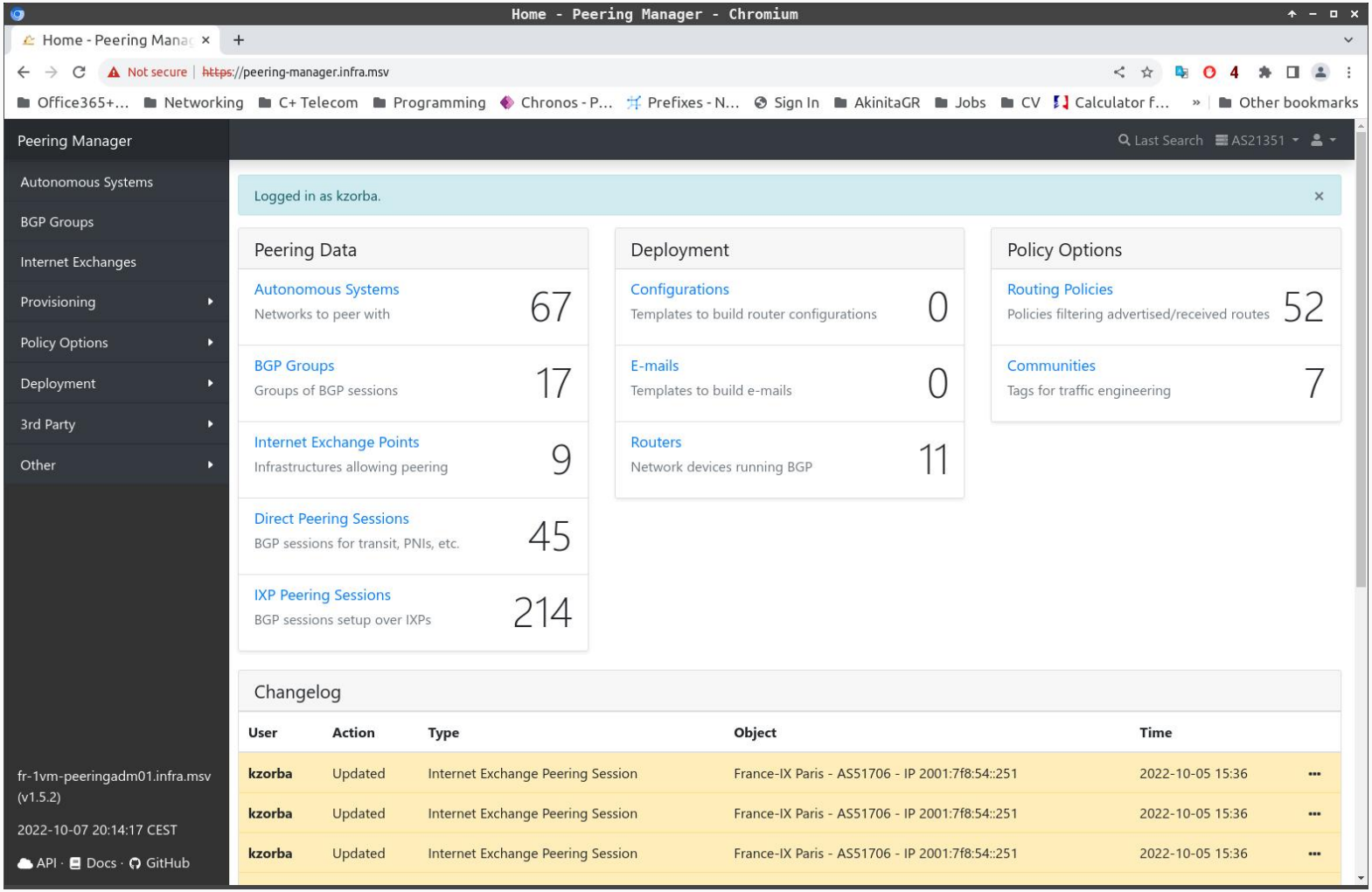
- A lot of open source configuration management frameworks
- Salt (sometimes referred to as SaltStack) an open-source software for event-driven IT automation, remote task execution, and configuration management
- NAPALM is a vendor neutral, cross-platform open source project that provides a unified API to network devices
- All Python based
- Development based on Salt/NAPALM using Jinja templates [7]

github repo

`https://github.com/kzorba/bgp-te-tool` [9]

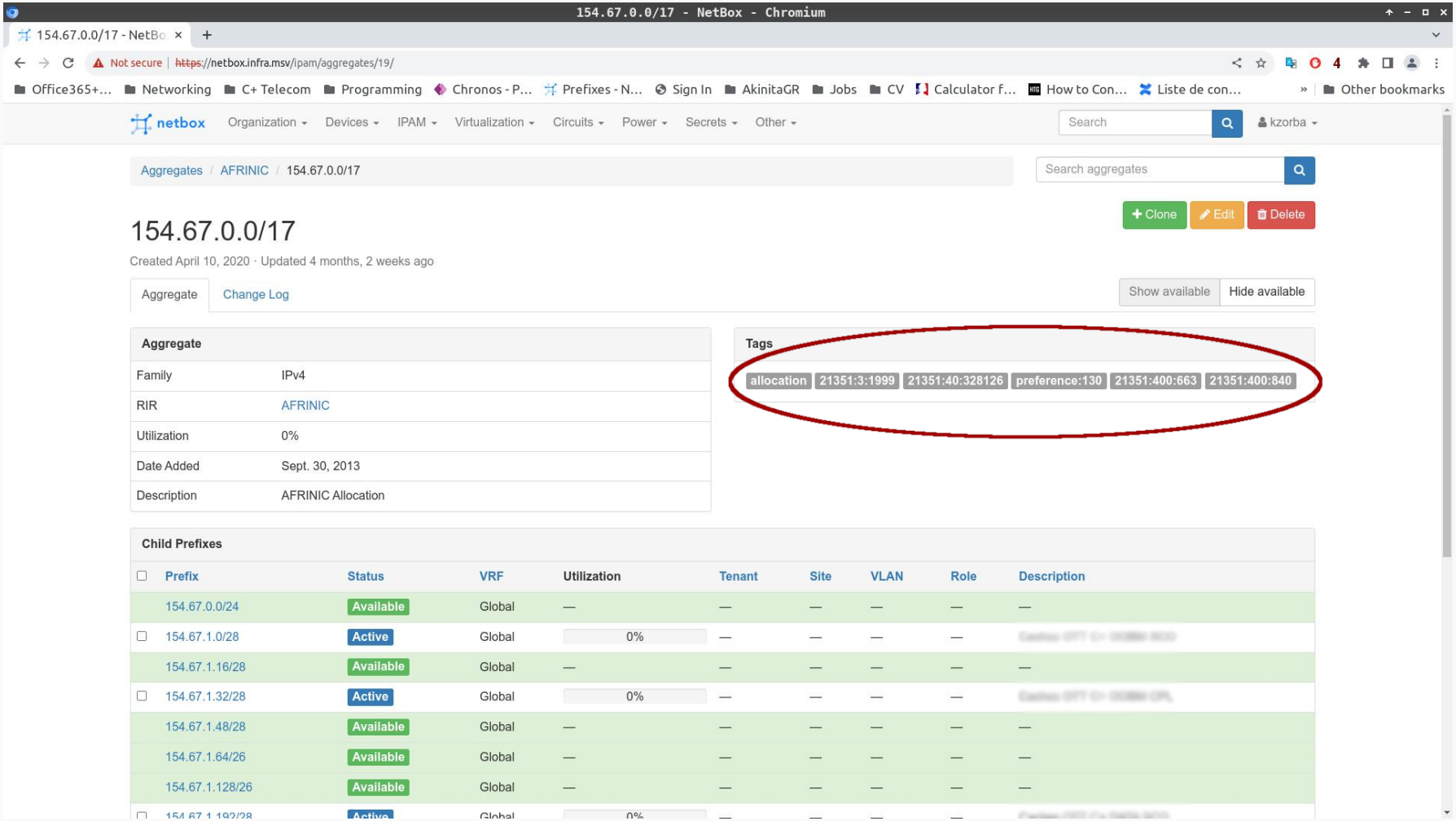
- Tool code, documentation and demonstration
- Simulated network using docker containers and `docker-compose`
- `goBGP` containers simulate peers and transit providers
- Current implementation supports Juniper routers (JunOS jinja templates)
- Contributions (eg other vendor support) highly welcome!

Tool usage - Peerings



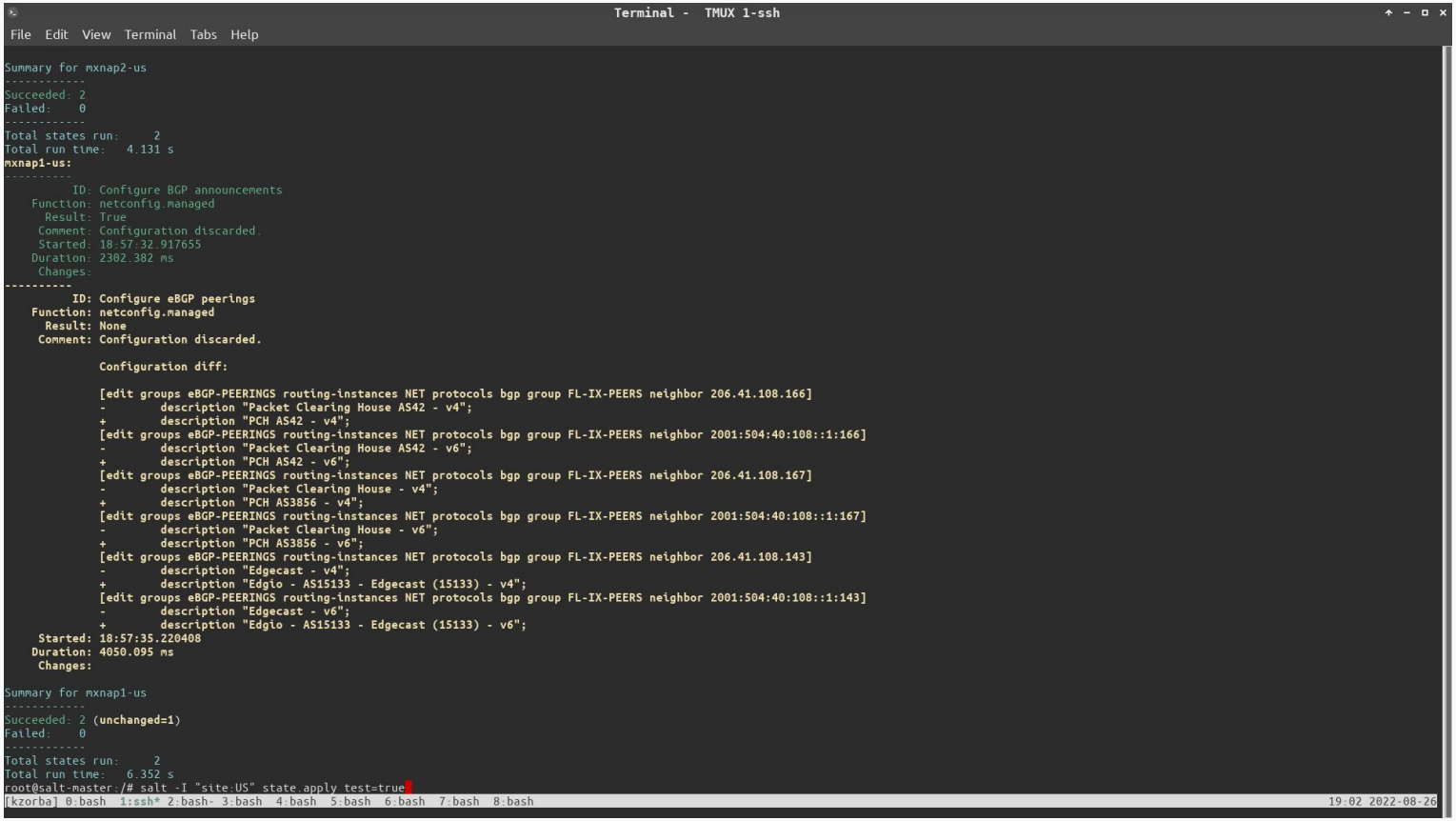
BGP information in Peering Manager

Tool usage - BGP announcements in IPAM



Prefix large community tagging in netbox

Network configuration with Salt



Applying state in a subset of routers

Production Network rollout

- Gradual deployment in AS21351 (router by router and location by location)
- Operations currently handled by the engineering team, a very small circle
- Training to operations teams will follow
- Possibility to rollout the tool POP by POP and in each (geo) location at a time was very beneficial for controlled deployment
- Up until now, rollout was smooth and controlled
- Tool currently handles ~260 peerings in 6 diverse geo-locations and 6 IXes

Experience share

- Design (mostly) and implementation not trivial
- Very limited human resources and day to day operations required attention
- Very demanding preparations tasks in the network to prepare the ground for the tool introduction
- 90% of the time was low level details and thinking
- Operations now require a paradigm shift (not easy)
- After all the work, the basis is there for future development as well

Considerations using on-demand BGP announcements

IRR / RPKI ROAs

- Good MANRS: Route policy, contacts and intended announcements SHOULD be documented in IRR. Accurate route filtering necessary
- What about on-demand announcements?
- Sometimes, not practical to pre-provision every possible route/route6 object
- Updating the IRR on the time of need leads to late reaction (upstream filter updates?)
- RPKI ROAs another source of validation

ROA maxLength?

- RPKI ROA maxLength can provide the necessary flexibility for TE
- Security implications (forged-origin prefix/subprefix hijack)
- ~~draft-ietf-sidrops-rpkimaxlen~~ RFC 9319/BCP 185: The Use of maxLength in the Resource Public Key Infrastructure (RPKI) [2]
- Greater harm for non announced address space
- Minimal ROAs recommended (whenever possible)

Experiences with IP transit providers

- Different levels of flexibility in transit provider networks
- From completely manual communication in case of need for a new announcement to various automated options in-between
- Respected providers care a lot about security and stability of the routing system (of course)
- Some providers give options to announce sub-prefixes and are responsive
- Automated generation of filters takes time and is performed a limited number of times within the day
- Flexibility required from customers

Monitoring of BGP Announcements

- Need a way to monitor our current BGP announcements
  1. as seen from the outside world (Internet)
  2. as sent from our own routers to peers
- For (1) various services exist (eg RIPE RIS live feed)
- For (2) best solution is BMP (BGP Monitoring Protocol [Adj-RIB-Out] [1], supported in pmacct [4] tool)

- Configurable differences between (1) and (2) should give alerts (eg possible hijack detection)
- Currently work in progress

Discussion points

- Best practices (in a document form) for operators and transit providers regarding on-demand announcements?
- Are the security implications with RPKI ROA maxLength a blocking point?
- Flexibility/operation agility vs security tradeoff
- How “real-time” should traffic engineering actions be and how often?
- AntiDDoS defences and big failures?

Thank You!

THANK YOU!

A special thanks to the authors/contibutors of the great open source tools available and the relevant communities.

QUESTIONS?

References

[1] T. Evens et al. *RFC 8671: Support for Adj-RIB-Out in the BGP Monitoring Protocol (BMP)*. https://datatracker.ietf.org/doc/html/rfc8671. Nov. 2019.

[2] Y. Gilad et al. *The Use of maxLength in the Resource Public Key Infrastructure (RPKI)*. https://datatracker.ietf.org/doc/rfc9319/.

References ii

[3] J. Heitz et al. *RFC 8092: BGP Large Communities Attribute*. https://datatracker.ietf.org/doc/html/rfc8092. Feb. 2017.

[4] Paolo Lucente. *pmacct BMP daemon*. http://www.pmacct.net/.

[5] Guillaume Mazoyer and Contributors. *Peering Manager*. https://peering-manager.net.

[6] *NETBOX*. https://github.com/netbox-community/netbox.

[7] *Network Automation with Salt*. https://docs.saltproject.io/en/3002/topics/network_automation/index.html.

References iii

[8] J. Snijders, J. Heasley, and M. Schmidt. *RFC 8195: Use of BGP Large Communities*. https://datatracker.ietf.org/doc/html/rfc8195. June 2017.

[9] Kostas Zorbadelos. *A tool for automated BGP Traffic Engineering*. https://github.com/kzorba/bgp-te-tool.