BGP best practices for Day 1 & 2 Operations

VIRTUAL TECH DAY - 14.03.2023

Maximilian Wilhelm max@sdn.clinic

Agenda

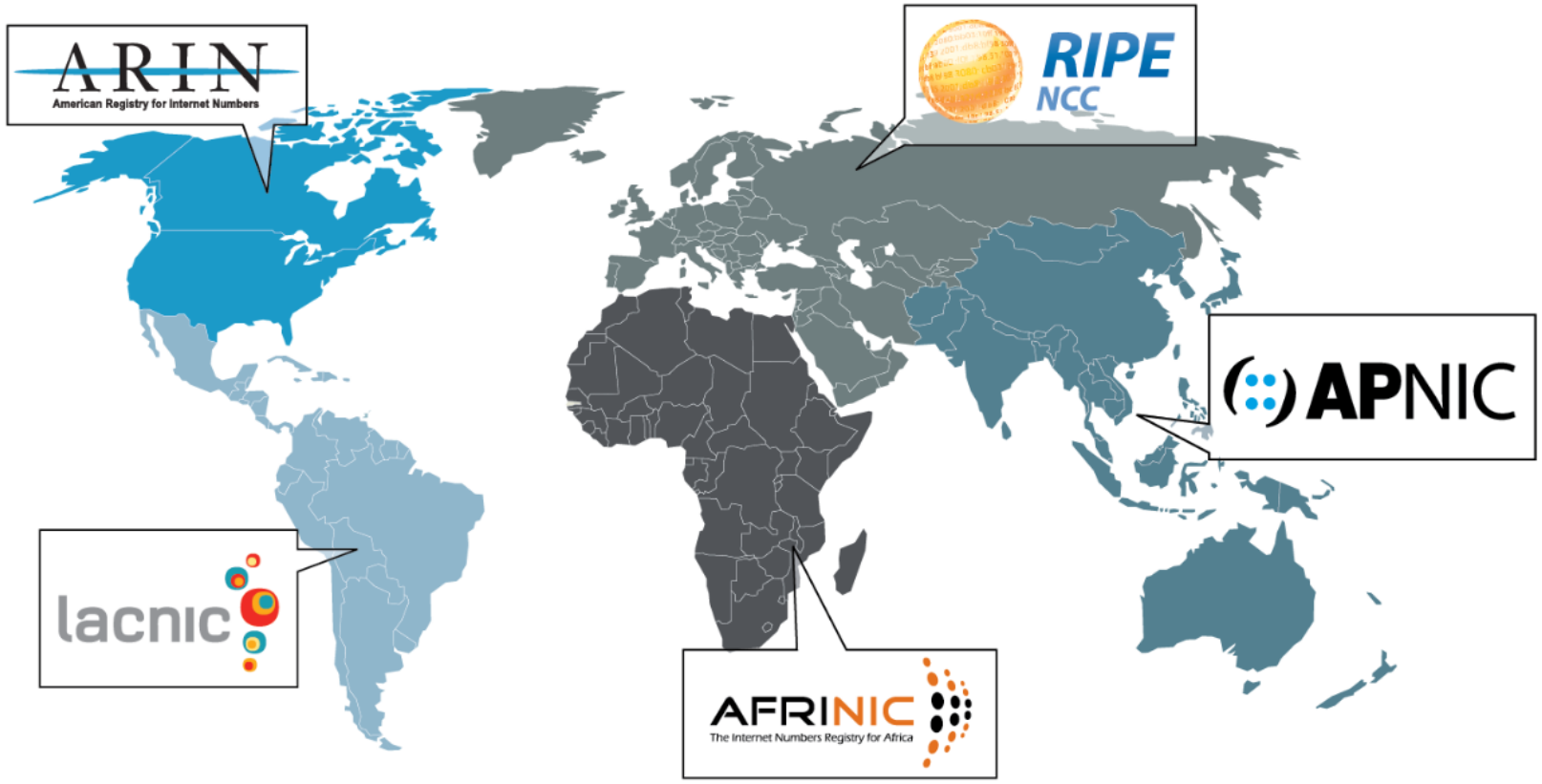
1. Who's who
2. Internet Fundamentals
    1. RIRs, LIRs, ASNs, IPs, and ASes
3. Routing Protocol Theory
4. BGP
    1. Paths & Attributes
    2. Best path decision
    3. Routingtables and their interaction
5. external vs. internal BGP
    1. Route Reflection
6. Best Current Operational Practices (BCOPs)
    1. Filtes, filters, filters
    2. Prefix-lists vs. Route-Maps/policies
    3. RPKI primer

Maximilian Wilhelm

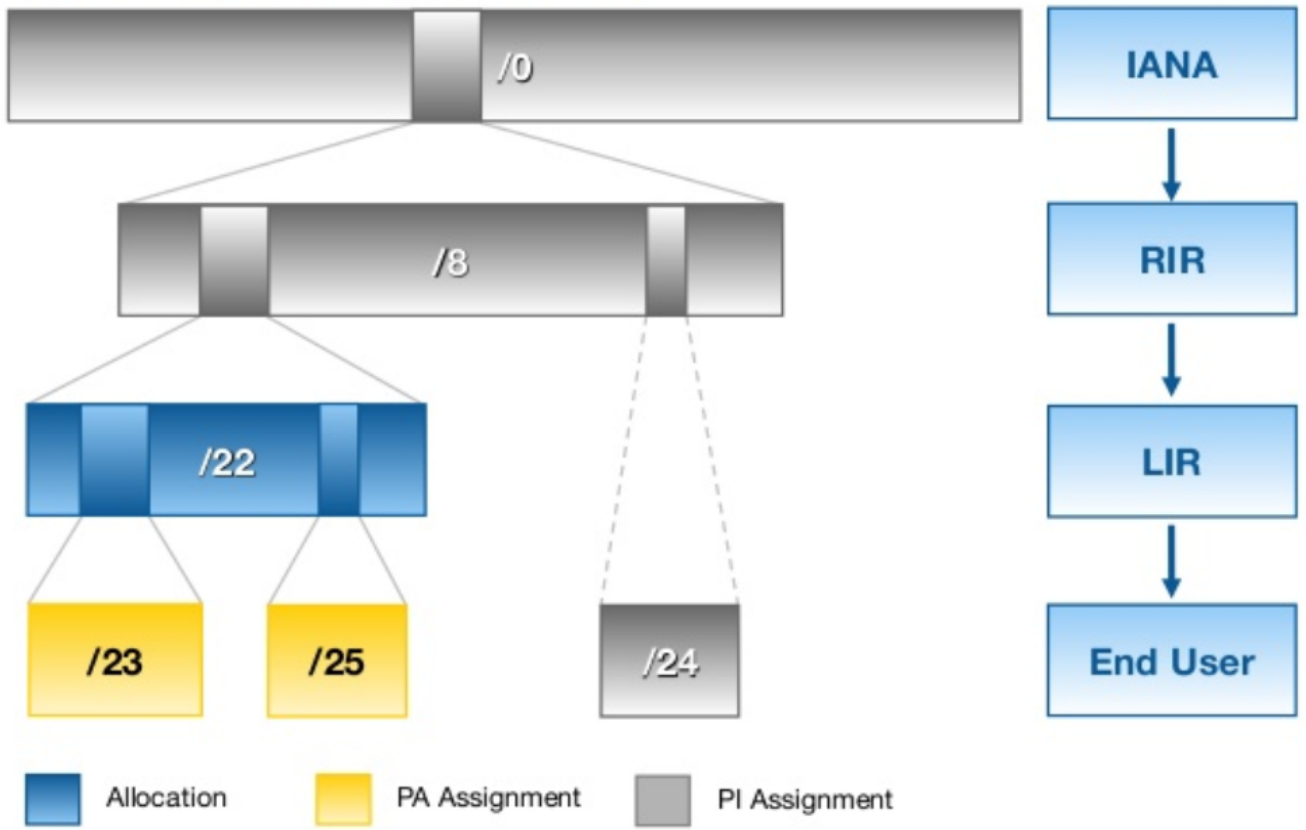
- Recovering Network Engineer
- OpenSource Hacker
- Holistic (Network) Automation Evangelist
- Fanboy of
  - (Debian) Linux
  - (Linux) networking
- Occupation:
  - By day: Network Automation Engineer at Cloudflare
  - By night: Infrastructure Archmage, Freifunk Hochstift
  - In between: Freelance Infrastructre Architect for hire
- Contact
  - @BarbarossaTM(@noc.social)
  - max@sdn.clinic

Fundamentals

The Five Regional Internet Registrars (RIRs)

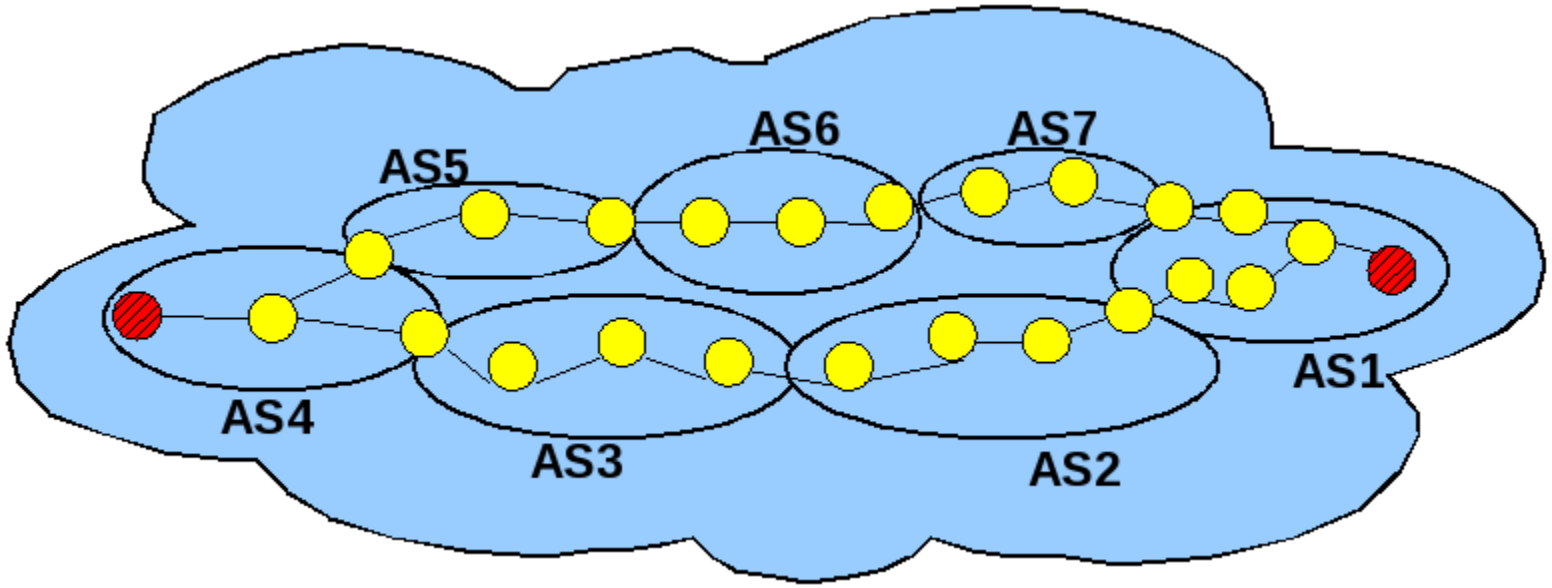


IP space allocation



Autonomous System(s)

- All routers and prefixes under the same routing policy / operator(s)
- Identified by a 16bit / 32bit AS Number (ASN)
- Assign by a LIR/RIR



Theory

Routing Protocol Theory

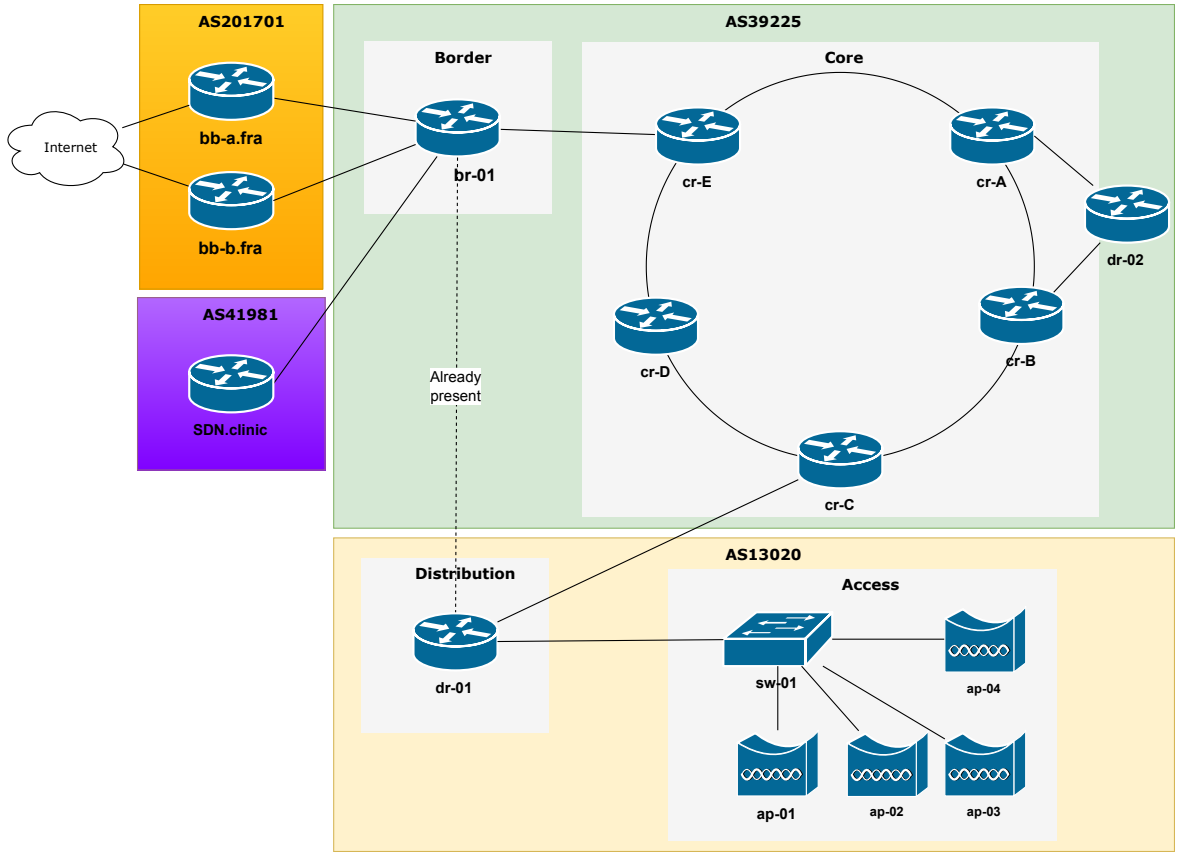
Theory

Home Scale Internet

- Single-homed to one ISP
- Local network 192.168.178.0/24
- One default router
- Only one path

Enterprise / Service Provider Scale Networking

Theory



Interior vs. Exterior Gateway Protocols

Exterior Gateway Protocols

- Exchange routing information between ASes / with untrusted peers
- De-facto standard protocol: BGP

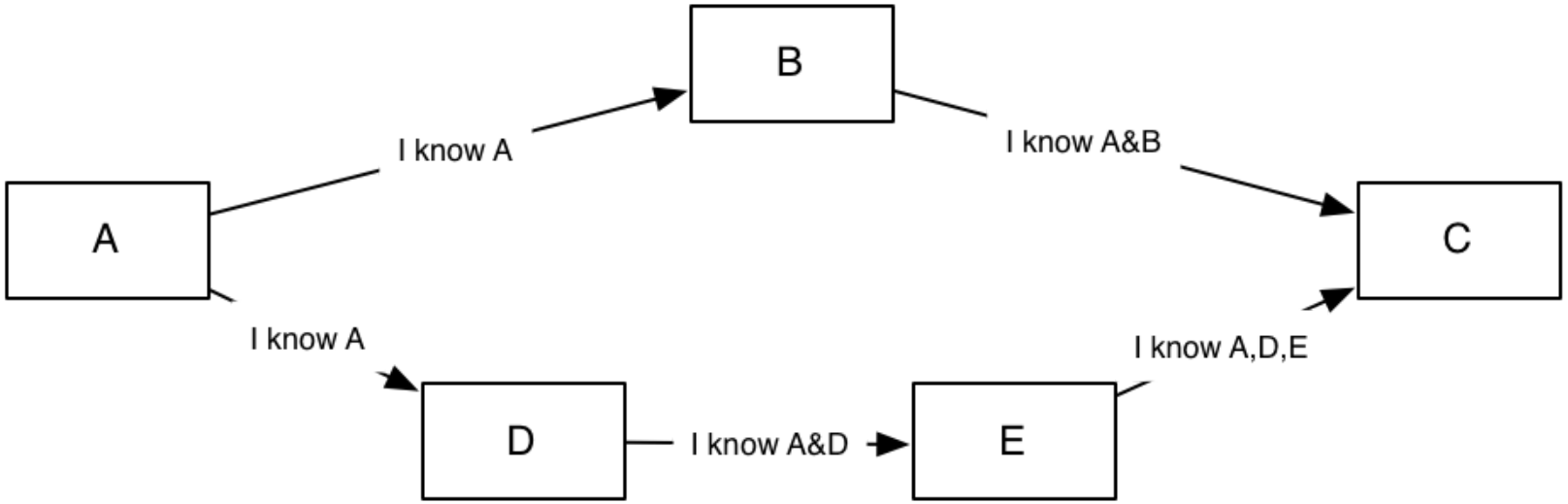
Interior Gateway Protocols

- Exchance routing information inside AS
- De-facto standard protocols:
  - OSPF or IS-IS
  - iBGP

Theory

Distance Vector

- Routers always send their complete routing table with increased metrics
- Metrics are counted to infinity where infinity = 15
- Example: Routing Information Protocol (*RIP*)

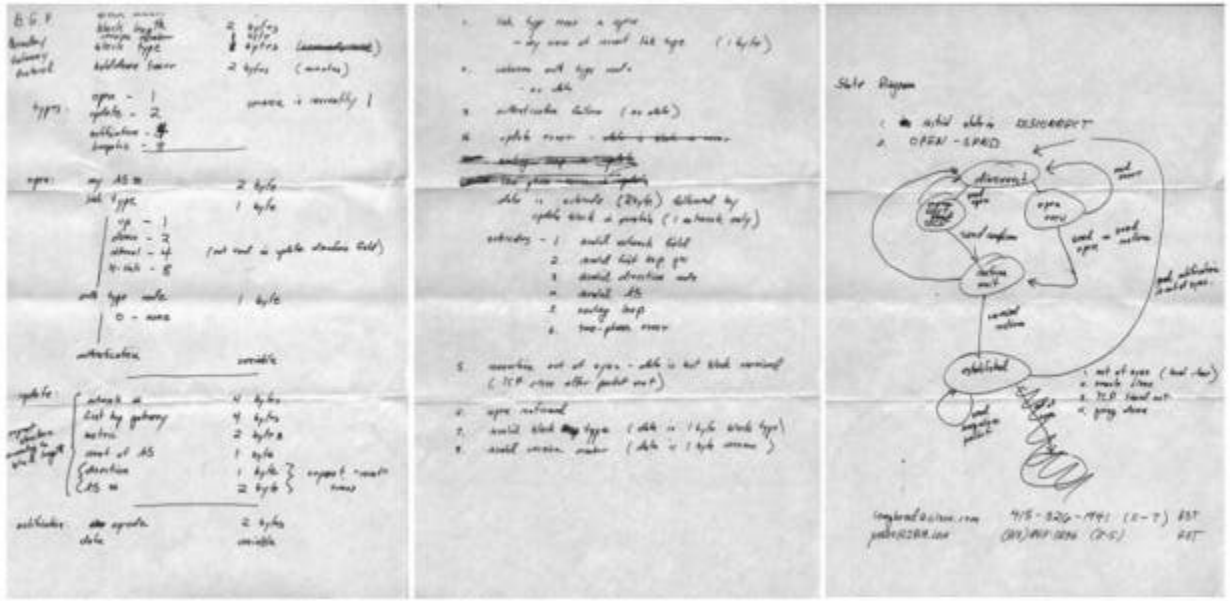


Path Vector

- Not only a next-hop but also a path to target
- Multiple attributes per route
- Multiple routes with different paths to a single destination possible
  - Enables equal cost multi path (ECMP)
- Only next-hop attributes are modified

Border Gateway Protocol

Three Napkin Protocol



Led to RFC "BGP-1" #1105 (1989/06)

History and BGP today

Four iterations of main BGP protocol

- RFC1105 (1989/06): BGP-1
- RFC1163 (1990/06): BGP-2
- RFC1267 (1991/10): BGP-3
- RFC1771 (1995/03): BGP-4 superseded by RFC4271 (2006/01)

Lots of additions (best of)

- RFC2918 (2000/09): Route Refresh Capability for BGP-4
- RFC4456 (2006/04): BGP-RR
- RFC4760 (2007/01): MP-BGP
- RFC6793 (2012/12): 32bit ASNs
- RFC7911 (2016/07): BGP AddPath
- RFC7999 (2016/10): BGP Blackhole Community

Holds the Internet together

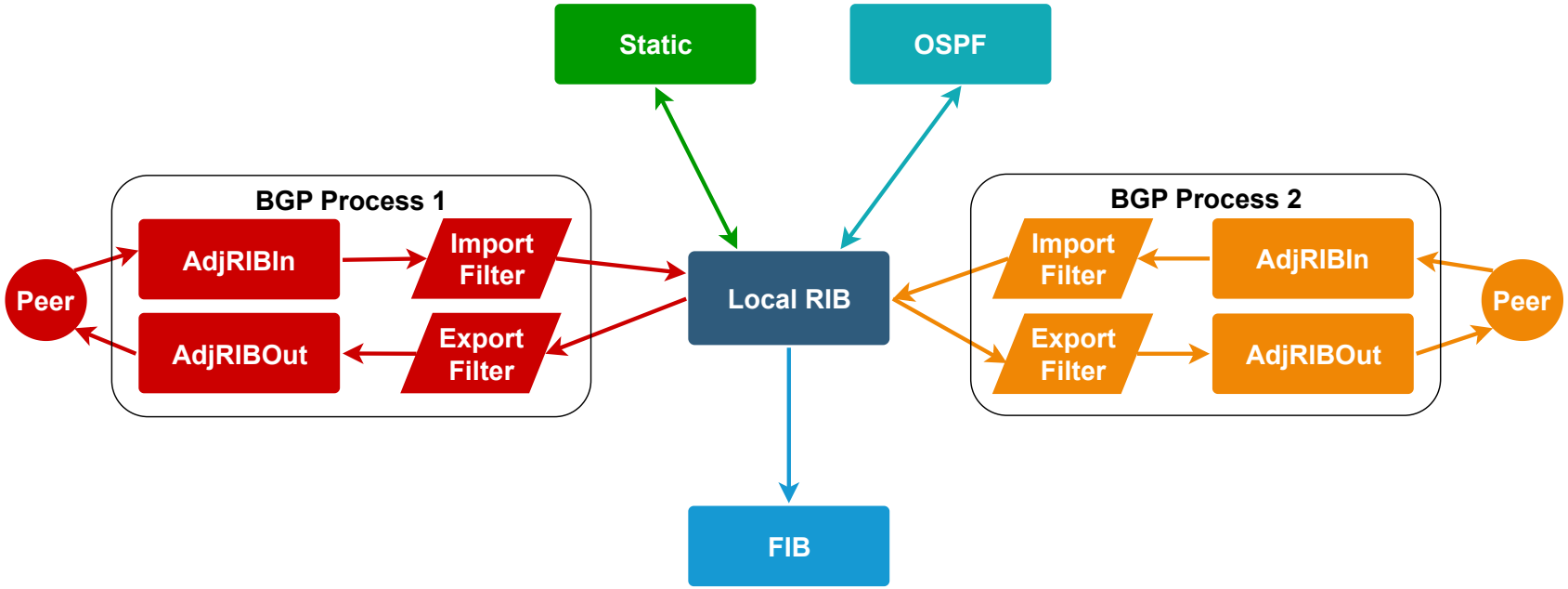
- Implemented in all major routing suites and vendors
- Interoperable

BGP at a glance

- Neighbors have to be configured explicitly*
  - TCP connection on port 179
- Neighbors exchange Network Layer Reachability Information (NLRI)
- Router selects best route(s)
- Selected routes are installed in Local-RIB / FIB

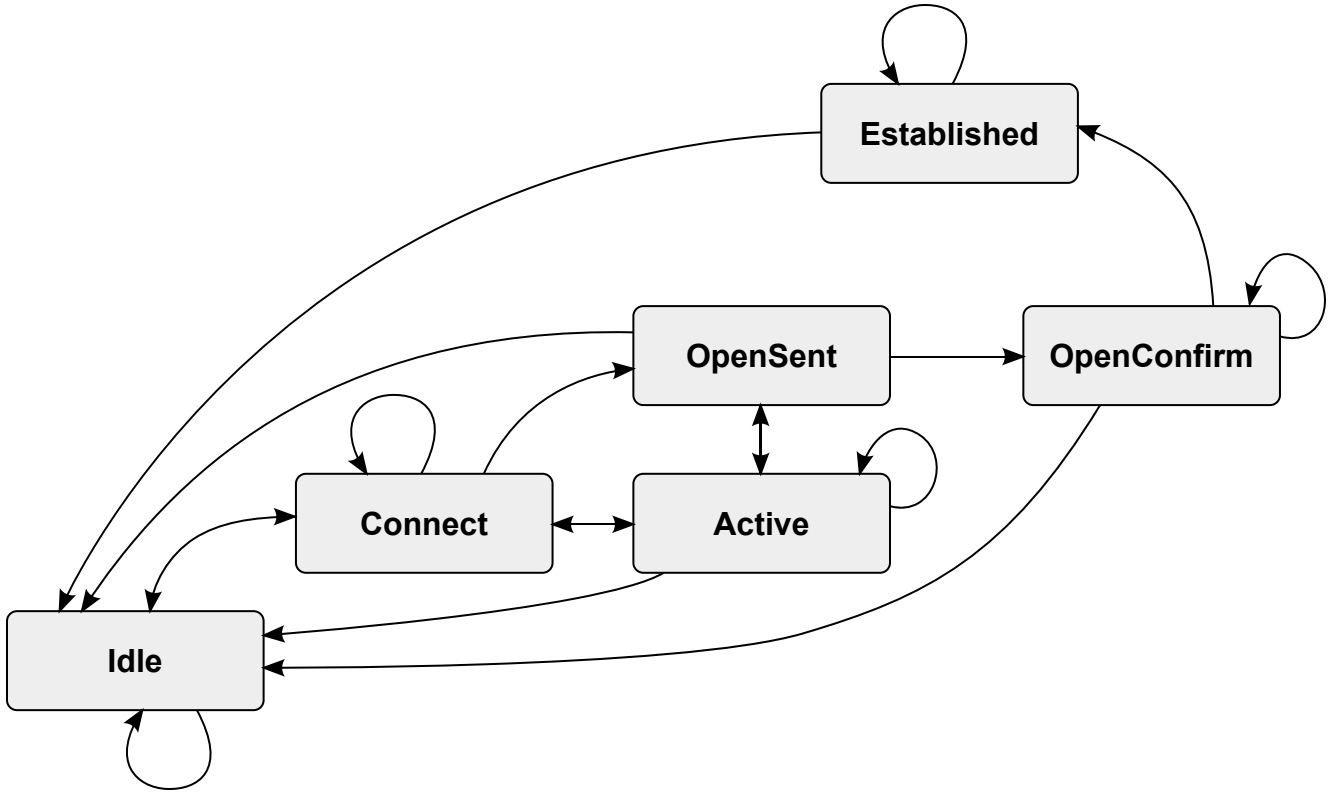
* Let's ignore DC fabrics for the moment

What's this routing table mess about?



- *Local Routing Information Base (RIB)* or *Loc-RIB*
- *Forwarding Informatio Base (FIB)*
- *Adjacency RIB In* (*AdjRIBIn*)
- *Adjecency RIB Out* (*AdjRIBOut*)

Statemachine



Quelle: Wikimedia Commons

BGP Path

List of all Autonomous Systems (ASes) a route has gone trough

- Used to prevent routing loops
- Shorter is better

- Lab network as seen at DE-CIX:
  - Path: 201701 39225 13020
  - Translation: FFRL, SDN-CLINIC-LAB, CCCV-AS

- May be prepended for traffic engineering purposes:
  - Path: 50629 201701 201701 201701 201701 39225 13020
  - Translation: LWLcom, (3x prepend) FFRL, SDN-CLINIC-LAB, CCCV-AS

Path Attributes

Each prefix* carries a set of attributes:

- Next-Hop
- Local Preference
- AS Path
- Origin
- Multi Exit Discriminator (MED, optional)
- Communities (optional)
- Large Communities (optional)

Attributes can be:

- *Transitive* or *non-transitive*
  - Will be forwarded to the next BGP-speaker - or not
- *Mandatory* or *optional*

* or set thereof

BGP Next-Hop

A route has to point somewhere

IP address of the next router on a path

- Can be adjacent router
- Can be a router somewhere else
  - Look up route to remote router in local routing table
  - Will probably be resolved via IGP (e.g. OSPF)

Is the next hop not present in the local routing table the route cannot be used

Will be installed with an unreachable next hop in RIB

BGP Local Preference

- 32bit unsigned int
- Default value: 100
- Meaningful within one AS
- Higher is better
- Real world usage: Rank prefixes of customers, peers, transits, …
  - Direct/prioritize where our traffic is sent

Example values from Freifunk Rheinland e.V AS201701 Backbone:

Local Pref	Prefix type
5	Remote Transits
500	Transits
1.000	Peerings
10.000	Downstreams

Further watching: https://media.ccc.de/v/denog13-12617-local-pref-considered-evil

BGP Multi Exit Discriminator

- 32bit unsigned int
- Default value: 0
- Meaningful between two adjacent ASes
  - non-transitive
- Lower is better
- Real world usage: Direct multi-homed peer where to ingest traffic to us
- Some ASes ignore/overwrite MED values
  - Talk to peer

BGP Communities

- Numerical route attributes / labels
- Meaningful only to local AS specifications

Classical Communities (RFC1997, Aug. 1996):

- 32bit unsigned int
  - Usually split up as: 16bit ASN, 16bit value
- Default value: non set
- Canonical representation: <ASN>:<value>

Well-Known communities

- NO_EXPORT
- NO_ADVERTISE
- BLACKHOLE (RFC7999, Oct. 2016)

BGP Large Communities

16bit ASN part to small for 32bit ASNs

Large Communities (RFC8092, Feb. 2017):

- 96bit unsigned int
  - 32bit ASN
  - 32bit Local-Data1
  - 32bit Local-Data2
- Canonical Representation: <ASN>:<Local Data 1>:<Local Data 2>

BGP Route Decision (eBGP)

1. Is the Next Hop reachable?
2. Higher Local Preference
3. Shorter AS_PATH
4. Smaller ORIGIN
5. Smaller MED, if multiple prefixes with same next AS
6. Smaller Router ID
7. Smaller Neighbor IP address

See RFC4271, Section 9.1.2 for more details

external vs. internal BGP

External BGP (eBGP):

- Routers in two different ASes
- Session established between IP addresses on transfer network

Internal BGP (iBGP):

- Routers within the same AS
- Sessions usually established between router's loopback addresses
  - Requires working IGP (e.g. OSPF)
  - Session will stay up as long as on path within the network exists
- No path attributes altered in iBGP
  - Routes received via iBGP MUST NOT be sent to an iBGP neighbor
  - iBGP full mesh required*

* unless iBGP Route Reflection is used

BGP Route Decision (complete)

1. Is the Next Hop reachable?
2. Higher Local Preference
3. Shorter AS_PATH
4. Smaller ORIGIN
5. Smaller MED, if multiple prefixes with same next AS
6. eBGP > iBGP
7. Smaller IGP cost
8. Smaller Router ID
9. Smaller Neighbor IP address

See RFC4271, Section 9.1.2 for more details

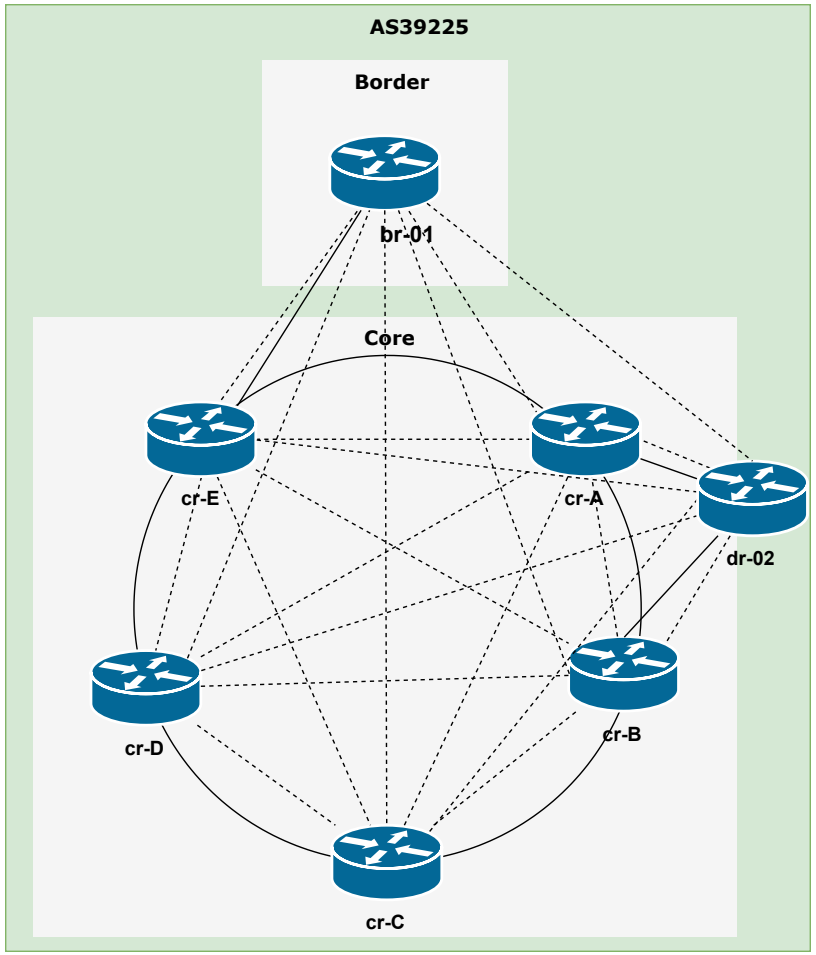
iBGP Full Mesh

Rule: Every router within an AS has to have an iBGP session to every other router.

- O(n²) scale. D'oh.
- Management & memory headache

But there's help:

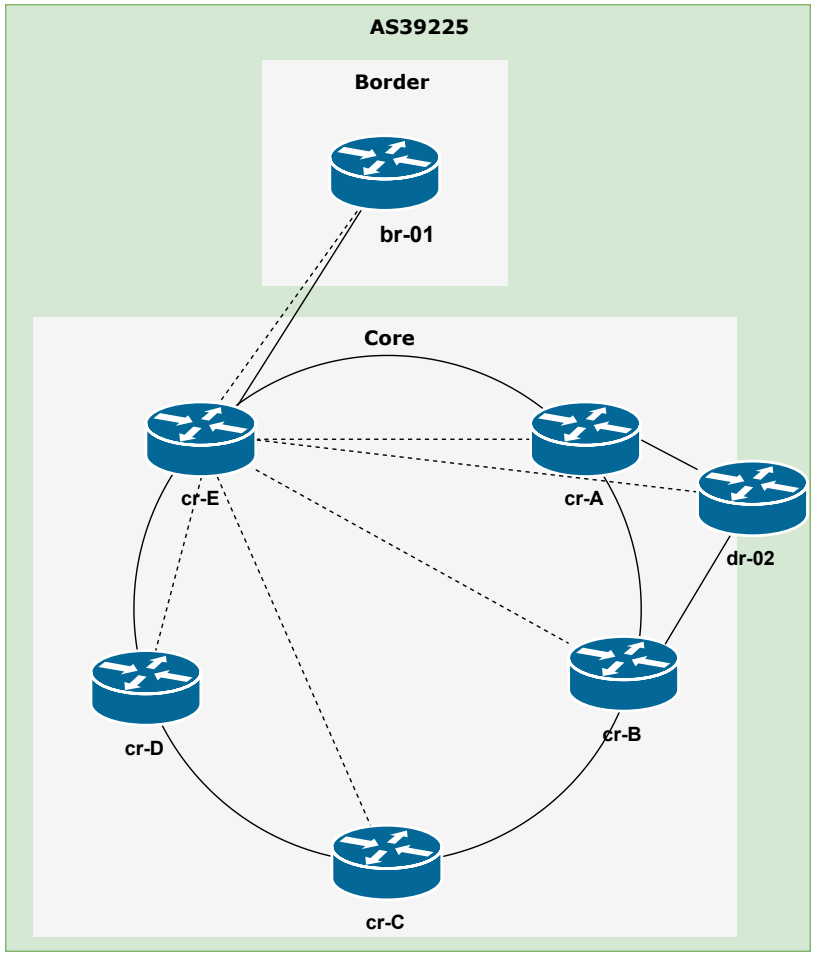
- Route Reflection!



iBGP Route Reflection

RFC4456 (Apr. 2006)

- BGP extension to work around full mesh requirement
- *Cluster ID* prevents routing loops
  - Think "internal Path"
- Route Reflector (RR) advertises eBGP + iBGP routes to iBGP peers
- Special configuration only on RR



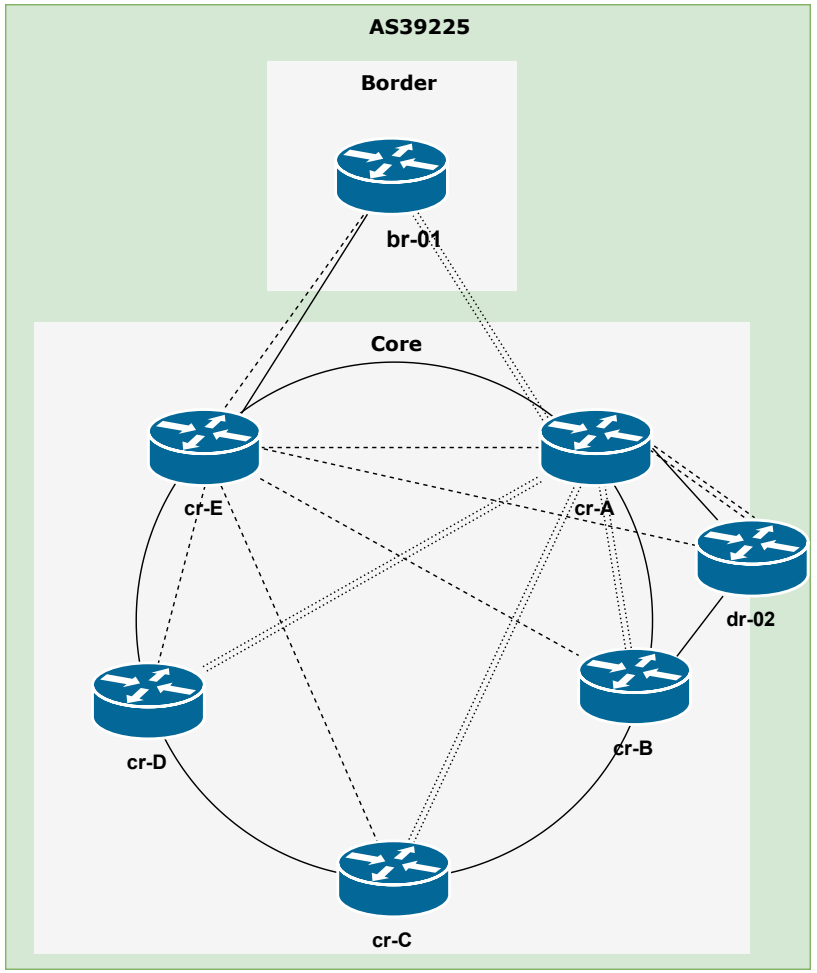
Redundant iBGP Route Reflection

Obvious SPoF can be alleviated by redundant RRs

Make sure to place RRs well distributed in your network

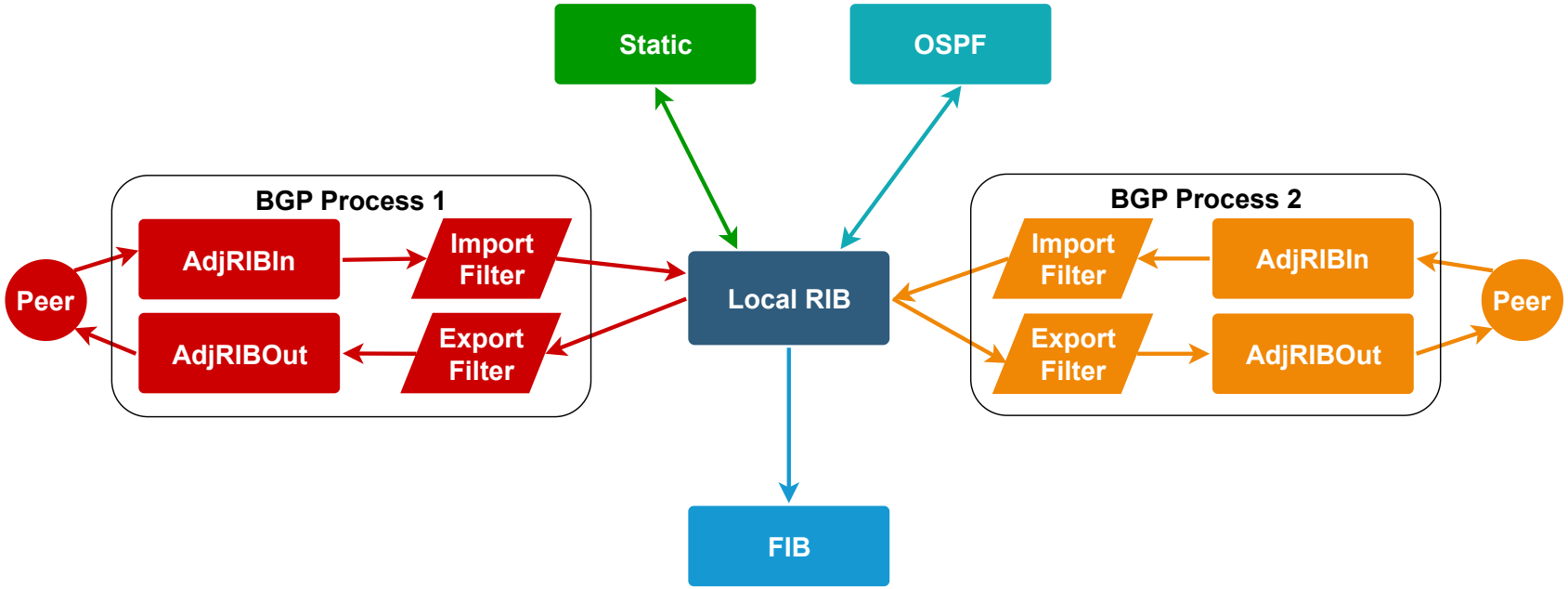
Further reading:

- Route Reflection RFC4456
- Optimal Route Reflection: RFC9107



BGP in the real world

Routing Tables - (combined) RIB

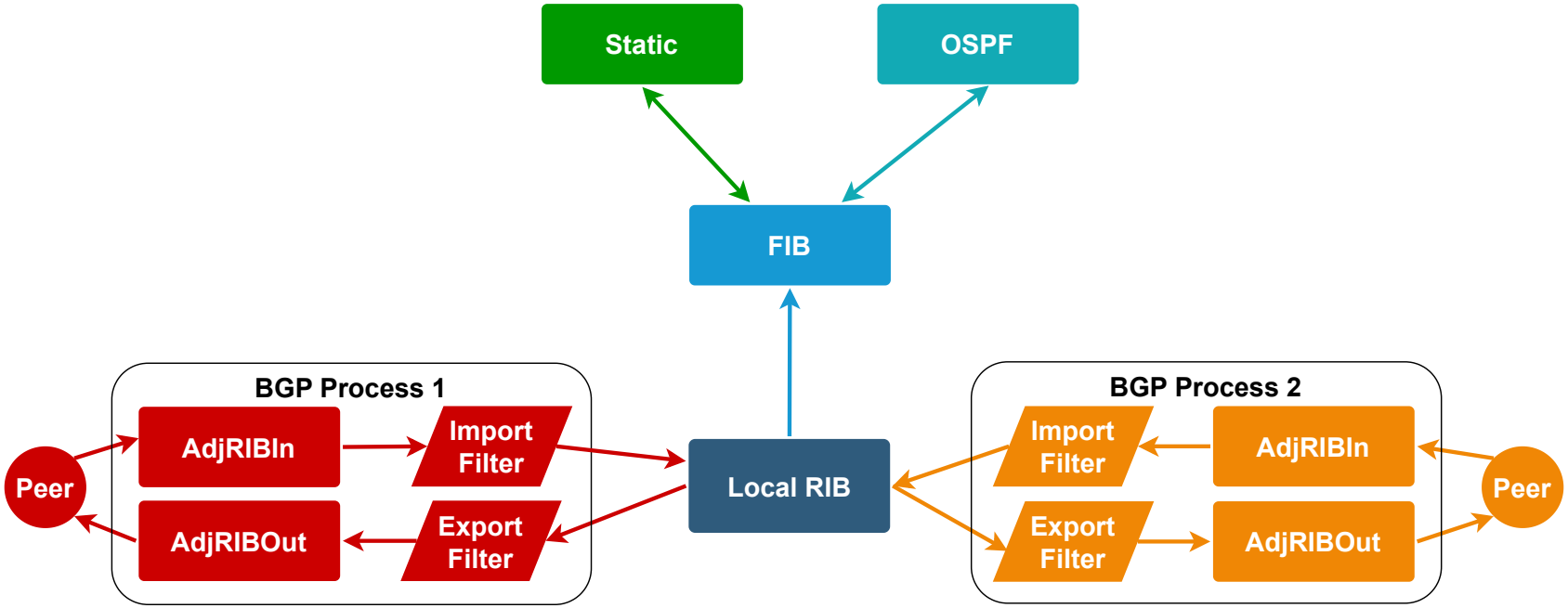


- May require strict import/export filters for all protocols

Used by (e.g.):

- bird
- bio-routing
- Juniper

Routing Tables - per-protocol RIBs



- Routes have to be explicitly *redistributed* between protocols

Used by (e.g.):

- Arista
- Cisco (IOS*, NX-OS)
- FRR
- Huawei

A Word on Defaults

Annoyance

- Default IPv4 unicast peer for any AF on Cisco-esk NOSes
  - no bgp default ipv4-unicast

Security

- Default eBGP Route Propagation Behavior without Policies (RFC8212)
  - List of compliance NOSes: https://github.com/bgp/RFC8212
  - Check your NOS (version)!

Best Current Operational Practices (BCOPs)

Big Thanks and Shoutout

A lot of the following slides are inspired or entirely borrowed (with permission :)) from **Fiona** and **Vincentz**, who gave an awesome workshop on *Network Architecture In Practice* at DENOG14 (Slide deck) and/or from router configs I have been working with.

The examples are for Junos, as it's fairly readable

- Focussing on the concepts here
- Think of snippets are buildings block / puzzle pieces
- You may need to adapt to bird, IOS-XR, EOS, ...

Ideally this will eventually end up in the DENOG Routing Guide with more examples for different platforms, when we all have a little spare time :)

Also see the NLNOG BGP Filter Guide for a lot of config examples.

The Basics

- PeeringDB entry, so others know where to peer with you
- IRR DB entries, so others can build filters
  - ROUTE / ROUTE6 + AS-SETs
- Sign you prefixes with RPKI
  - AfricNIC's RPKI: https://afrinic.net/resource-certification
- Have good MANRS
  - "*Mutually Agreed Norms for Routing Security (MANRS) is a global initiative that helps reduce the most common routing threats.*"
  - MANRS-Network-Implementation-Guide.pdf

The Basics - ROUTE objects

A lot of networks will automatically build filters base on route objects for your AS

- Maybe including your upstreams / peers
- It may be worth having ROUTE objects for more specifics ready for traffic engineering use cases

```
$ whois 2a07:ed07:fc13::42

...
inet6num:       2a07:ed07:fc13::/48
netname:        SDN-CLINIC-FrOSCon-Network-Track
descr:          FrOSCon Network Track - Infrastructure Network
country:        DE
...

% Information related to '2a07:ed07:fc13::/48AS39225'

route6:         2a07:ed07:fc13::/48
origin:         AS39225
mnt-by:         MW7511-MNT
created:        2018-07-30T14:10:59Z
last-modified:  2018-07-30T14:10:59Z
source:         RIPE
```

The Basics - AS-Set

An AS-Set allows listing all your ASes and those of your customers

- Set up the *customer cone*
- Allows easy filtering for your upstreams / peers

```
$ whois AS-SDN-CLINIC-LAB
as-set:         AS-SDN-CLINIC-LAB
descr:          SDN.clinic LAB - currently in use at FrOSCon Network Track
tech-c:         MAX7511-RIPE
admin-c:        MAX7511-RIPE
mnt-by:         MW7511-MNT
created:        2018-07-31T09:57:40Z
last-modified:  2018-07-31T10:48:31Z
source:         RIPE
members:        AS39225
members:        AS13020
```

Filters, Filters, Filters

Speak after me:

I will filter my external BGP sessions,
only announce my prefixes and those of my customers,
and only accept what is right.

You don't want to be the person who caused this :)

https://blog.cloudflare.com/route-leaks-and-confirmation-biases/

Quick recap

Transit

- Provides access to the whole Internet
- Maybe a *Full Table*, may be *Default Route* only

Peer

- Direct connection between two networks
- Can either be PNI or via an IXP

Customer/Downstream

- We are transit for them
- We provide a default route or Full Table

What Should We Export?

	To Customer	To Peer	To Transit
Own Aggregates	Yes	Yes	Yes
From Customer	Yes	Yes	Yes
From Peer	Yes	No	No
From Upstream	Yes	No	No

Our prefixes + customer prefixes = *Customer cone*

Now, remember those Communities? Let's use them for controlling our announcements!

Community Controlled Prefix Announcements

Allows to control what we announce when we learn it

- Wait what?!

Simplifies configuration significantly

- No need to update export filters everywhere if new customer prefixes are added

Community	**Announce ...**
ASN:100	To iBGP
ASN:101	To Transits
ASN:102	To Customers
ASN:103	To Peers

```
community TO_INTERNAL members <ASN>:100;
...
```

Our Prefixes

Our prefixes (aggregates) need to be in the RIB to be announced

- Either as aggregate routes (only becomes active if more specific is present)
- Or static discard route

Inject into network in core, not on the edge

- If an edge gets isolated and still announces aggregates it will blackhole traffic

```
[edit routing-options aggregate]
route 192.0.2.0/24 {
    community [ TO_INTERNAL TO_PEERING TO_CUSTOMER TO_UPSTREAM ];
    discard;
}
```

or

```
[edit routing-options static]
route 192.0.2.0/24 {
    community [ TO_INTERNAL TO_PEERING TO_CUSTOMER TO_UPSTREAM ];
    discard;
}
```

RPKI - Drop RPKI Invalids

RPKI invalids = prefixes announced with wrong origin ASN and/or allowed prefix-length.

- Likely caused by a hijack or route-leak
- Most certainly not intended by the rightful owner, so ignore those prefixes

Common software suites for RTR servers: Routinator and/or OctoRPKI

```
[edit routing-options validation]
group RPKI {
    max-sessions 4;
    session 2001:db8::23::1 {
        port 3323;
    local-address 2001:db:23::42;
    }
    session 2001:db8::23::2 {
        port 3323;
    local-address 2001:db:23::42;
    }
}
```

Remermber to allow traffic in CoPP :)

RPKI - Drop Invalids (Contd.)

```
[edit policy-options policy-statement 6-BASE-IN]
term MARK-RPKI-VALID {
    from validation-database valid;
    then {
        validation-state valid;
    }
}
term REJECT-RPKI-INVALID {
    from validation-database invalid;
    then {
        validation-state invalid;
    reject;
    }
}
```

(Same for IPv4)

Filter inbound, on all eBGP sessions

IXP Peering LANs

IXP Peering LANs should never be announced or accepted via eBGP

- Including more specifics
- At least for IXPs you're connected to
- This may lead to blackholing (temporary or continous)

Can get especially bad if you peer with the IXPs Route Server and those are still reachable

IXP Peering LANs - config

```
[edit policy-options]
prefix-list 4-PEERING-LANS {
    196.60.8.0/22;      # NAPA Africa Jburg
}
prefix-list 6-PEERING-LANS {
    2001:43f8:6d0::/64;  # NAPA Africa Jburg
}

[edit policy-options policy-statement 4-BASE-IN]
term REJECT-PEERING-LANS {
    from {
        prefix-list-filter 4-PEERING-LANS orlonger;
    }
    then reject;
}
[edit policy-options policy-statement 6-BASE-IN]
term REJECT-PEERING-LANS {
    from {
        prefix-list-filter 6-PEERING-LANS orlonger;
    }
    then reject;
}
```

Filter inbound, on all eBGP sessions

Invalid Prefix Lengths - Size Does Matter!

Longest prefixes to be expected in the DFZ

- /24 for IPv4
- /48 for IPv6

Usually considered too short:

- IPv4: /0 - /7
- IPv6: /0 - /2

Exceptions:

- Default route from upstream (if desired)
- Shorter customer prefixes (e.g. from your PA space)
- RTBH

Invalid Prefix Lengths - Config

```
[edit policy-options policy-statement 4-BASE-IN]
term REJECT-INVALID-PREFIX-LENGHT {
    from {
        route-filter 0.0.0.0/0 prefix-length-range /0-/7;
        route-filter 0.0.0.0/0 prefix-length-range /25-/32;
    }
    then reject;
}

[edit policy-options policy-statement 6-BASE-IN]
term REJECT-INVALID-PREFIX-LENGHT {
    from {
        route-filter ::/0 prefix-length-range /0-/2;
        route-filter ::/0 prefix-length-range /49-/128;
    }
    then reject;
}
```

Filter in/outbound, on all eBGP sessions

Bogon Prefixes

BOGONs should never be seen on the Internet

- Private IP space (RFC1918)
- CGNAT space (RFC6598)
- Documentation prefixes
- Multicast space
- "Class E"
- ...

You shouldn't accept nor announce these publically

https://www.team-cymru.com/bogon-networks

Bogon Prefixes - Config IPv4

```
[edit policy-options]
prefix-list 4-BOGON-PREFIXES {
    0.0.0.0/8;
    10.0.0.0/8;
    100.64.0.0/10;
    127.0.0.0/8;
    169.254.0.0/16;
    172.16.0.0/12;
    192.0.0.0/24;
    192.0.2.0/24;
    192.168.0.0/16;
    198.18.0.0/15;
    198.51.100.0/24;
    203.0.113.0/24;
    224.0.0.0/3;
}

[edit policy-options policy-statement 4-BASE-IN]
term REJECT-BOGONS {
    from {
        prefix-list-filter 4-BOGON-PREFIXES orlonger;
    }
    then reject;
}
```

Filter in/outbound, on all eBGP sessions

Bogon Prefixes - Config IPv6

```
[edit policy-options]
prefix-list 6-BOGON-PREFIXES {
    ::/8;
    100::/8;
    200::/7;
    400::/6;
    800::/5;
    1000::/4
    2000::/16;
    2001::/16;
    2002::/16;
    4000::/2;
    8000::/1;
}

[edit policy-options policy-statement 6-BASE-IN]
term REJECT-BOGONS {
    from {
        prefix-list-filter 6-BOGON-PREFIXES orlonger;
    }
    then reject;
}
```

Filter in/outbound, on all eBGP sessions

Bogon ASNs

There are also some ASNs which should not be seen in the wild, mainly

- Private ASNs
- Special purpose ASNs

You may want to allow them for customers which don't have a public ASN :)

Resource: https://bgpfilterguide.nlnog.net/guides/bogon_asns/

Bogon ASNs - Config

```
[edit policy-options]
as-path-group BOGON-ASNS {
    as-path ZERO ".* 0 .*";
    as-path AS_TRANS ".* 23456 .*";
    as-path BOGON16 ".* [64496-131071] .*";
    as-path BOGON32 ".* [4200000000-4294967294] .*";
    as-path LAST32 ".* 4294967295 .*";
}

[edit policy-options policy-statement 4-BASE-IN]
term REJECT-BOGON-ASNS {
    from as-path-group BOGON-ASNS;
    then reject;
}

[edit policy-options policy-statement 6-BASE-IN]
term REJECT-BOGON-ASNS {
    from as-path-group BOGON-ASNS;
    then reject;
}
```

Filter in/outbound, on all eBGP sessions

Accept or Reject Own Prefixes?

Depends on your network topology!

Lots of island?

- Only drop local prefixes

Have you own backbone?

- Filter your prefixes on all edges

Accept or Reject Own Prefixes? - Config

```
[edit policy-options]
prefix-list 4-OWN-PREFIXES {
    192.0.2.0/24;
    198.18.0.0/15;
}

[edit policy-options policy-statement 4-BASE-IN]
term REJECT-MY-PREFIXES {
    from {
        prefix-list-filter 4-OWN-PREFIXES orlonger;
    }
    then reject;
}
```

Filter inbound, on all eBGP sessions

Purge our communities

- If we use communits to control BGP announcements - and thereby the behaviour of our network - we must protect them!
  - Strip all "our" communities from prefixes received from external peers

```
[edit policy-options]
community OUR-COMMUNITIES members <OUR ASN>:*;

[edit policy-options policy-statement 4-<PEER>-IN]
term SCRUB-COMMUNITIES {
    then community delete OUR-COMMUNITIES;
}
```

The same applies for Large Communities, if in use

Filter in/outbound, on all eBGP sessions

Maintenance Switch

Sometimes a router needs to be taken out of service

- Ideally we drain all traffic before we do that

Simple version:

```
[edit policy-options policy-statement MAINTENANCE-MODE]
inactive: term ACTIVATE-MAINTENANCE {
    then reject;
}
```

Activate maintenance with

```
activate policy-options policy-statement MAINTENANCE-MODE term ACTIVATE-MAINTENANCE
```

The more fancy way would be to use apply-group

Apply this policy to all sessions which need to be drained

Putting it together - a transit

```
[edit policy-options policy-statement 4-TRANSIT-IN]
term ACCEPT-FULL-TABLE {
    then {
        community add TO_INTERNAL;
        community add TO_CUSTOMER;
        accept;
    }
}

[edit policy-options policy-statement 4-TRANSIT-OUT]
term EXPORT-TO-TRANSIT {
    from {
        community TO_UPSTREAM;
    }
    then {
        next-hop self;
        accept;
    }
}

[edit protocols bgp group 4-TRANSIT-DTAG]
neighbor 192.0.2.42 {
    description "Deutsche Telekom";
    import [ MAINTENANCE-MODE 4-BASE-IN 4-TRANSIT-IN REJECT-ALL ];
    export [ MAINTENANCE-MODE 4-BASE-OUT 4-TRANSIT-OUT REJECT-ALL ];
    peer-as 3320;
}
```

A Peer - Simple Setup with Prefix-Limit

At least have a prefix-list:

```
[edit protocols bgp group 4-NAPA-JBURG neighbor a.b.c.d]
description "Some nice peer";
import [ MAINTENANCE-MODE 4-BASE-IN 4-<PEER>-IN REJECT-ALL ];
export [ MAINTENANCE-MODE 4-BASE-OUT 4-<PEER>-OUT REJECT-ALL ];
family inet {
    unicast {
        prefix-limit {
            maximum 200;
            teardown {
                idle-timeout 1440;
            }
        }
    }
}

peer-as <PEER ASN>
```

Check PeeringDB for meaningful limit and apply some headroom, or ask peer.

On eBGP sessions with peers and maybe customers

c# A Peer - Standard Setup with Strict Filters

Query peer prefixes via bgpq4 or simlar tool fro IIRDB

```
[edit policy-option policy-statement 4-<PEER>-IN]
term PEER-PREFIXES {
    from {
        prefix-list-filter 4-<PEER>-PREFIXES;
    }
    then accept;
}

[edit protocols bgp group 4-NAPA-JBURG neighbor a.b.c.d]
description "Some nice peer";
import [ MAINTENANCE-MODE 4-BASE-IN 4-<PEER>-IN REJECT-ALL ];
export [ MAINTENANCE-MODE 4-BASE-OUT 4-PEER-OUT REJECT-ALL ];
family inet {
    unicast {
        prefix-limit {
            maximum 200;
            teardown {
                idle-timeout 1440;
            }
        }
    }
}

peer-as <PEER ASN>
```

Further checks and useful things

Skipped for brevity and time reasons:

- Seeing Tier1 ASNs in AS path via peering usually indicates a route-leak
- Very long AS paths are usually unused and consume resources
- Rejecting prefixes from direct peers via IXP Route Servers
- Graceful Shutdown (RFC8326)
- Adding informational communities, where a route was learned

See slides from DENOG workshop linked above for more details and examples

One last thing: BGP Roles

Define BGP relationship as part of BGP configuration

- Provider <-> Customer
- Peer <-> Peer
- Route Server Client <-> Route Server

Provides additional safeguard against misconfigurations

- Vendor adoption still ongoing - push yours :)

Specified in (RFC9234)

DENOG presentation

Further Reading

RFC4271: A Border Gateway Protocol 4 (BGP-4) - Rekhter, Li, Hares, 2006

BGP - Building Reliable Networks with the Border Gateway Protocol - I. van Beijnum, 2002

BGP Design and Implementation - Randy Zhang & Micah Bartell, 2003

Routing TCP/IP Volume II - Jeff Doyle & Jennifer DeHaven Carroll, 2016

Network Automation with Go

Fundamentals

Theory

BGP

Real World

BCOP

Links

Questions?

Questions?