

- # Modern BGP Design

Simplify the BGP infrastructure

Wholesale Winery Tour - 04/2022

nicola modena - CCIE #19119 / JNCIE-SP #986

@nmodena

● Agenda

- Motivation
- Legacy/Traditional BGP design
- Moving RR off-path
- RR for Datacenter and POP
- One RR for ALL ?
- Platform selection
- Questions

- Motivation

- Most BGP design rely on classical behavior

New feature are usually presented alone

There is not a document like this

It's based on my own original design

Combine new feature to achieve a simple & modern solution

- **bgp PIC** with FRR/xLFA to minimize fault restoration delay
- **bgp ADD-PATH** for Path Diversity
- **bgp ORR** Optimize Route Reflections

1

Legacy Route Reflector design

Traditional Route Reflector design, as we learn from books

● Service Provider Backbone

Sample Scenario

- Multiple POP
- Multiple Transit Site
- Multiple DC Sites

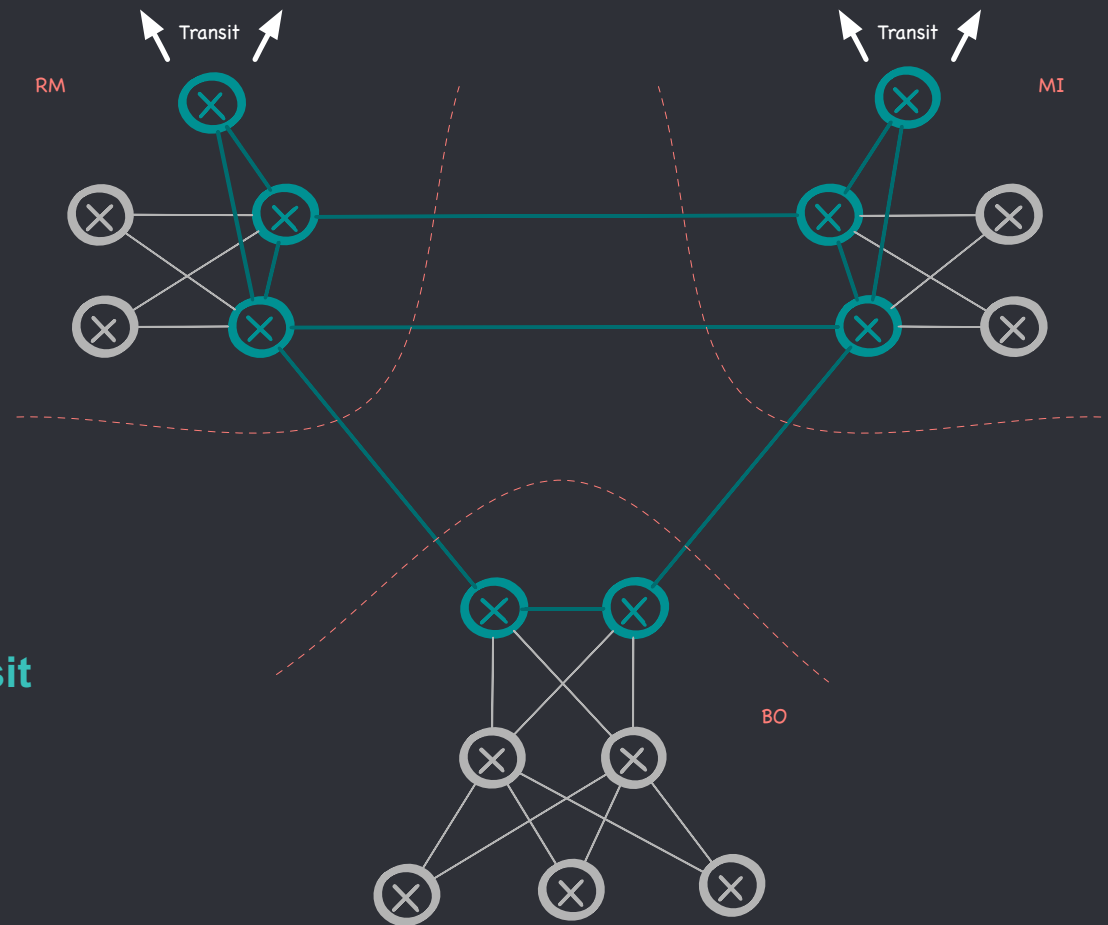
Requirement

- Optimal routing
- Load Balancing
- FIRT(*) confined in **CORE and Transit**
- default-route in POP/DC devices

Goals

- Simplicity
- Scalability

*) Full Internet Routing Table



Core Full Mesh vs Route Reflections

FIRT in all core routers for optimal routing

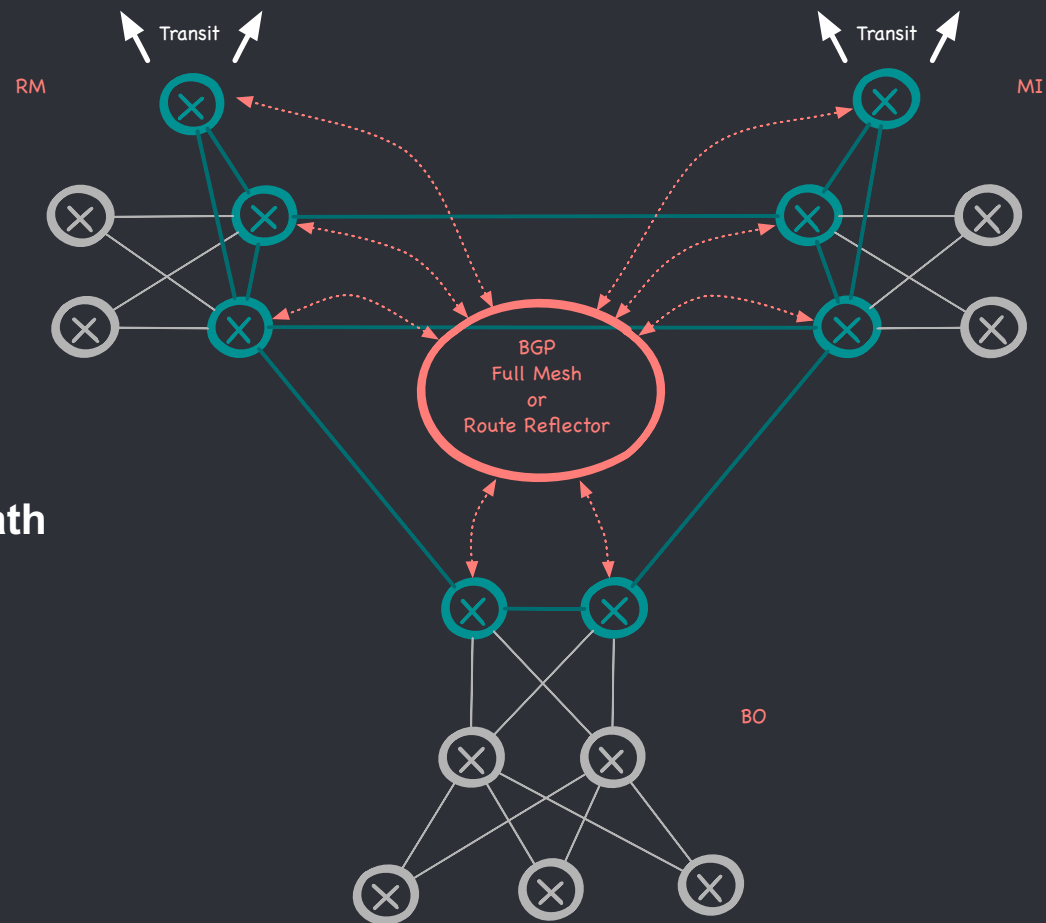
Full Mesh

- all routers receive all neighbors best path
- multiple path are possible
- not scalable

Route Reflector:

- Only Best Path it's reflected
- RR positioning it's important
- usually one RR per exit point

<https://blog.ipSPACE.net/2013/10/can-bgp-route-reflectors-really.html>



● Default route and RR hierarchy

FIRT it's not required inside POP/DC

default-route originated on:

- Transit : not optimal with MPLS
- Core : for LB and HA

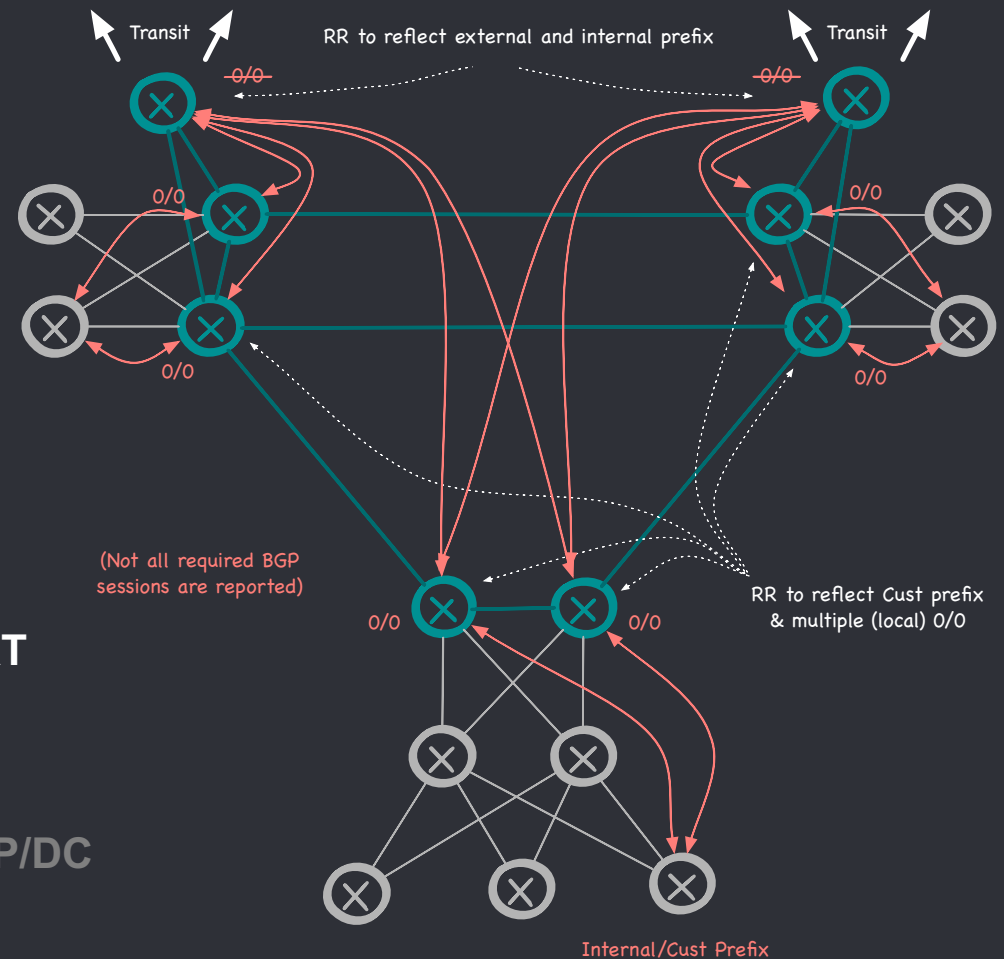
forwarding with «two stage lookup»

1. Using 0/0 from PE or DC to core
2. perform lookup and forward using FIRT

Hierarchical BGP design

- Transit as route-reflectors for Core
- Core/Border as route-reflectors for POP/DC
- How many RR ?

NOTE: if you are advertising cust prefix with IGP and then redistribute to BGP please don't!



2

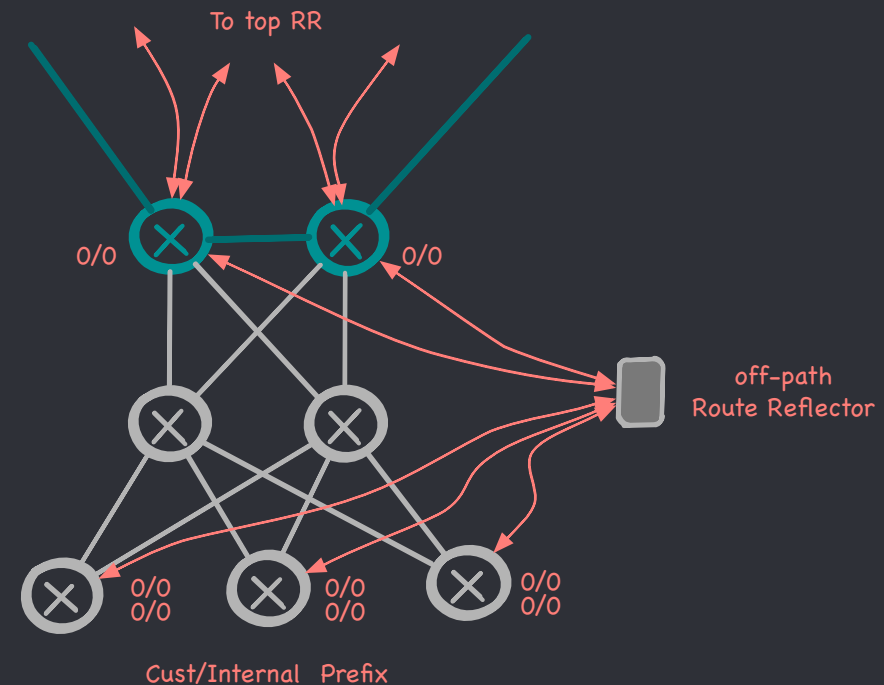
DataCenter/Pop Route Reflector

Detach Route Reflector role from border routers

- Move RR role from Core Borders to dedicate RR

Decoupling Control-Plane from Data-Plane:

- Redundant but also non optimal BGP prefix are useful to improve convergence time and achieve load-balancing
- Hierarchical FIB (BGP PIC) may combine local information and next-hop tracking: move convergence time from BGP to IGP.
- ADD-PATH enable advertisement of multiple path with different next-hop (and attributes)
-> rfc 7911 / Aug 2016



● ADD PATH Configuration example

ADD-PATH it's a negotiated capability
must be supported & configured

- **Session reset when enabled**
- **Independent Send and Receive capability**

In this simple design:

- **RR use only SEND**
- **Client use only RECEIVE**

route-reflector:

```
protocols {
  bgp {
    group iBGP {
      cluster 192.0.0.0;
      ...
      family inet {
        unicast {
          add-path send;
        }
      }
      neighbor 192.0.0.1;
      neighbor 192.0.0.2;
      [...]
    }
  }
}
```

clients:

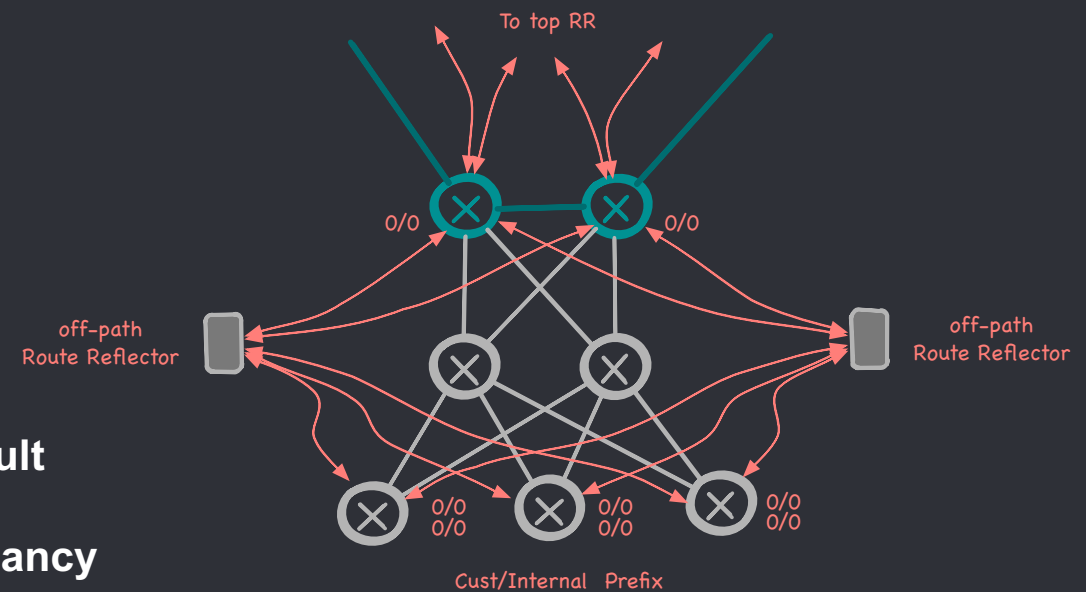
```
protocols {
  bgp {
    group RR {
      family inet {
        unicast {
          add-path receive;
        }
      }
      neighbor 192.0.0.254;
      neighbor 192.0.2.254;
    }
  }
}
```

● Route Reflector redundancy

- Redundancy must guarantee same functionality even in the event of a fault
- Do not abuse them, too much redundancy introduces complexity.

in this case:

- two path to cover LB and HA
- two copies to cover RR failure



- ADD PATH max-path

```
protocols {
  bgp {
    group iBGP {
      cluster 192.0.0.0;
      ...
      family inet {
        unicast {
          add-path {
            send {
              path-count 2;
            }
          }
        }
      }
      neighbor 192.0.0.1;
      neighbor 192.0.0.2;
      [...]
    }
  }
}
```

ADD-PATH configuration option to include max number of diverse path.

it's not valid on prefix basis: all or nothing.

3

Share Route Reflector

Share Route Reflectors between different Data-Center/Pop

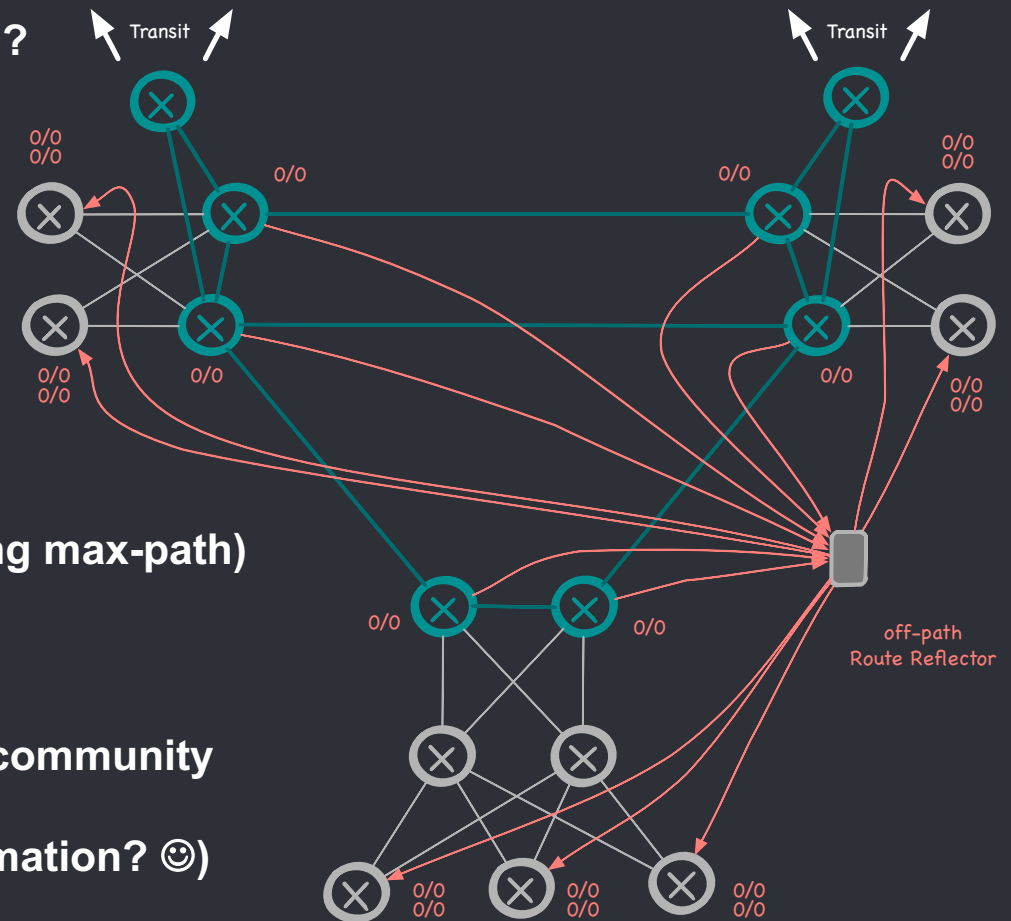
- Share same RR for all POP / DC

Can we use the same RR for all the SITES?

- Every site must receive local default-route. This prevent sub-optimal routing with MPLS

Options:

- Send ALL the [default-] route (removing max-path) and let's IGP select locally.
Cons: Not scalable
- identify each site default route with a community and write a policy on RR for each site
CONS: complex, not scalable (... automation? 😊)
- ORR (?) what it's this ?



● ORR Configuration

Optimized Route Reflections
RFC 9107 / Aug 2021

Route Selection from a different IGP Location

leverage IGP running SPF based on client topology and **reflects best path(s) based on client position.**

Configurable on a peer-group basis

example: reflection optimized for RM and MI

```
protocols {
  bgp {
    group RM-NAMEX {
      type internal;
      cluster 192.0.0.0;
      ...
      optimal-route-reflection {
        igp-primary 192.0.0.1;
        igp-backup 192.0.0.2;
      }
      neighbor 192.0.0.1;
      neighbor 192.0.0.2;
      neighbor 192.0.0.3;
      neighbor 192.0.0.4;
    }
    group MI-MIX {
      type internal;
      cluster 192.0.0.0;
      ...
      optimal-route-reflection {
        igp-primary 192.0.2.1;
        igp-backup 192.0.2.2;
      }
      neighbor 192.0.2.1;
      neighbor 192.0.2.2;
      neighbor 192.0.2.3;
      neighbor 192.0.2.4;
    }
  }
}
```

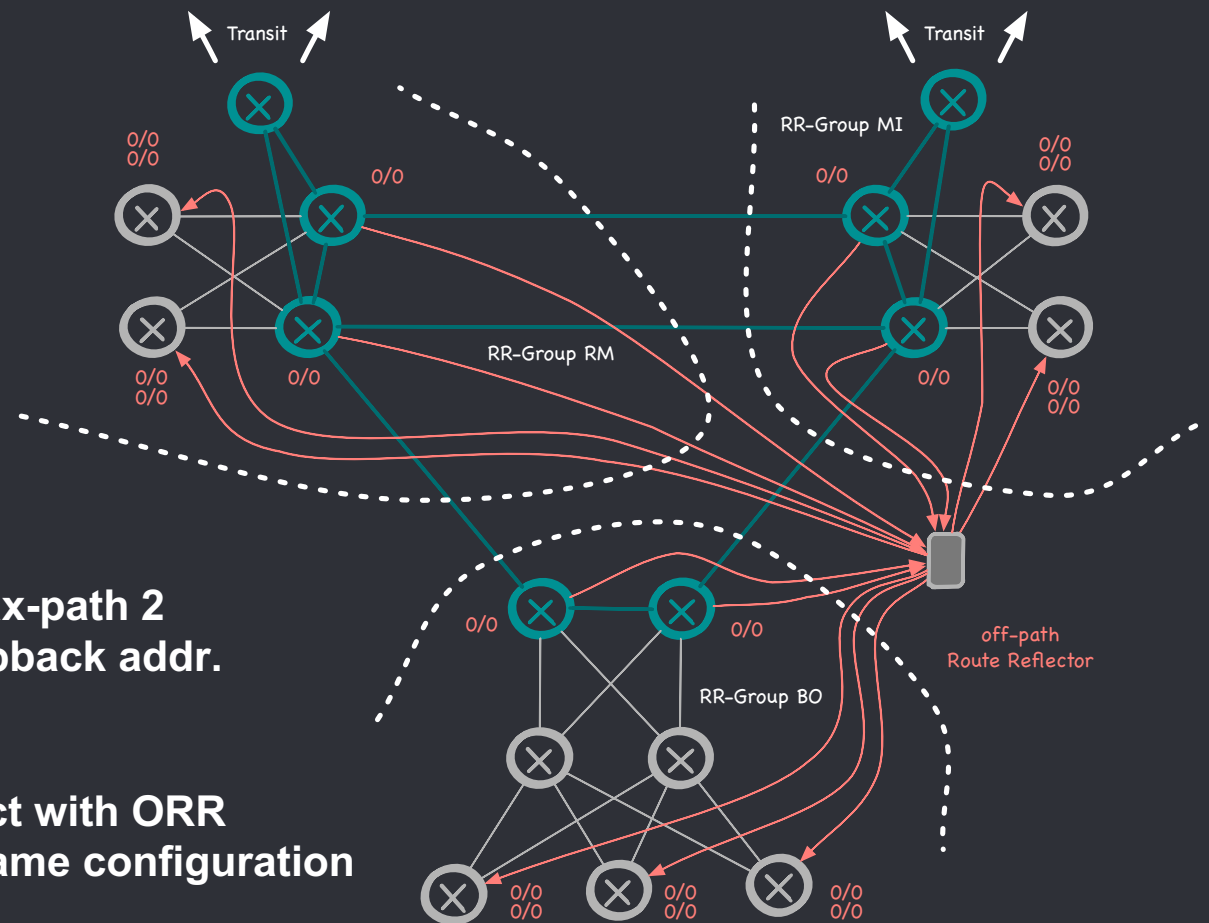
● Optimize route distribution with ORR

Solution:

- Create a peer-group per site
- Enable add-path send with max-path 2
- Enable ORR using border loopback addr.

- no community, no policy
- just enable add-path and select with ORR
- all clients shares exactly the same configuration

SIMPLE and AUTO-OPTIMIZED!

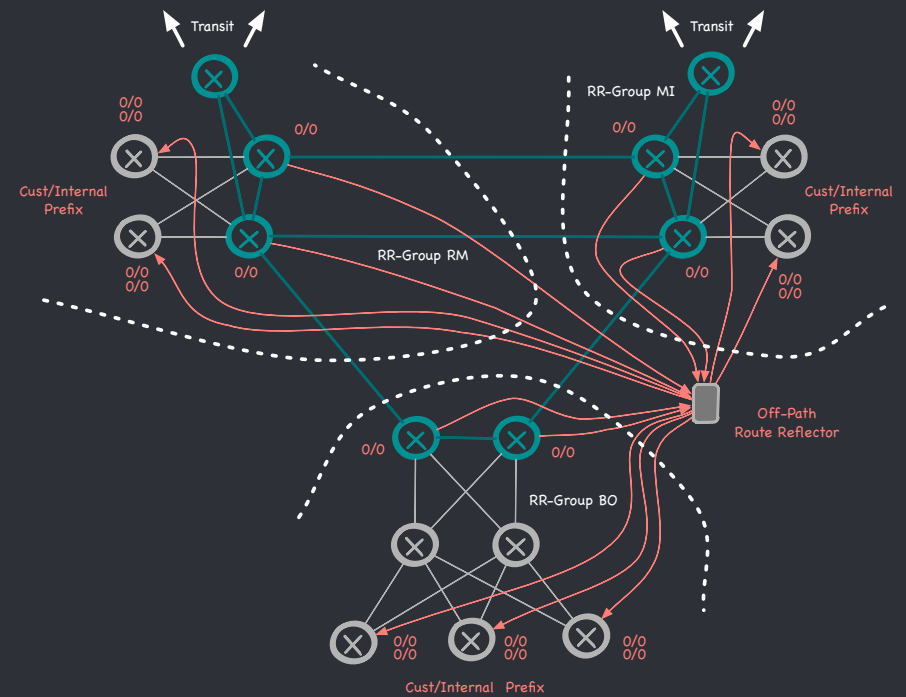
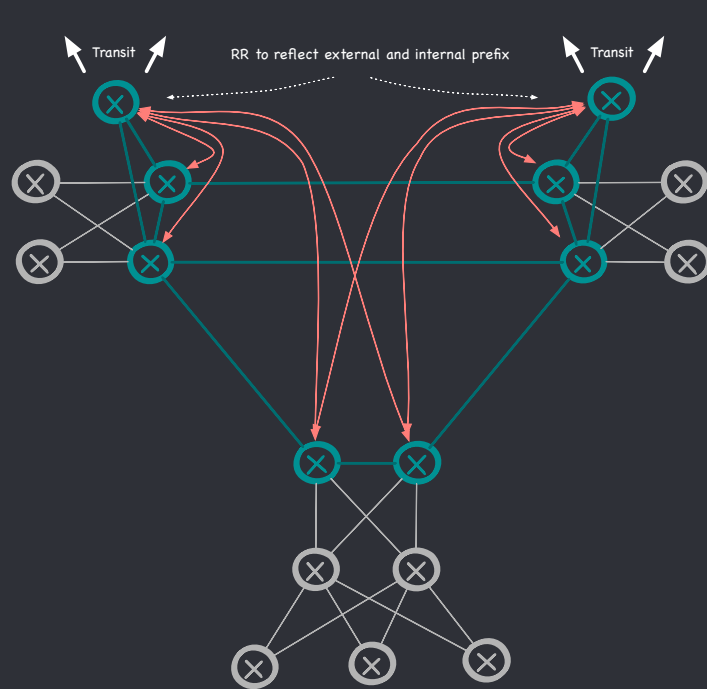


4

Combine Core and POP Route Reflectors

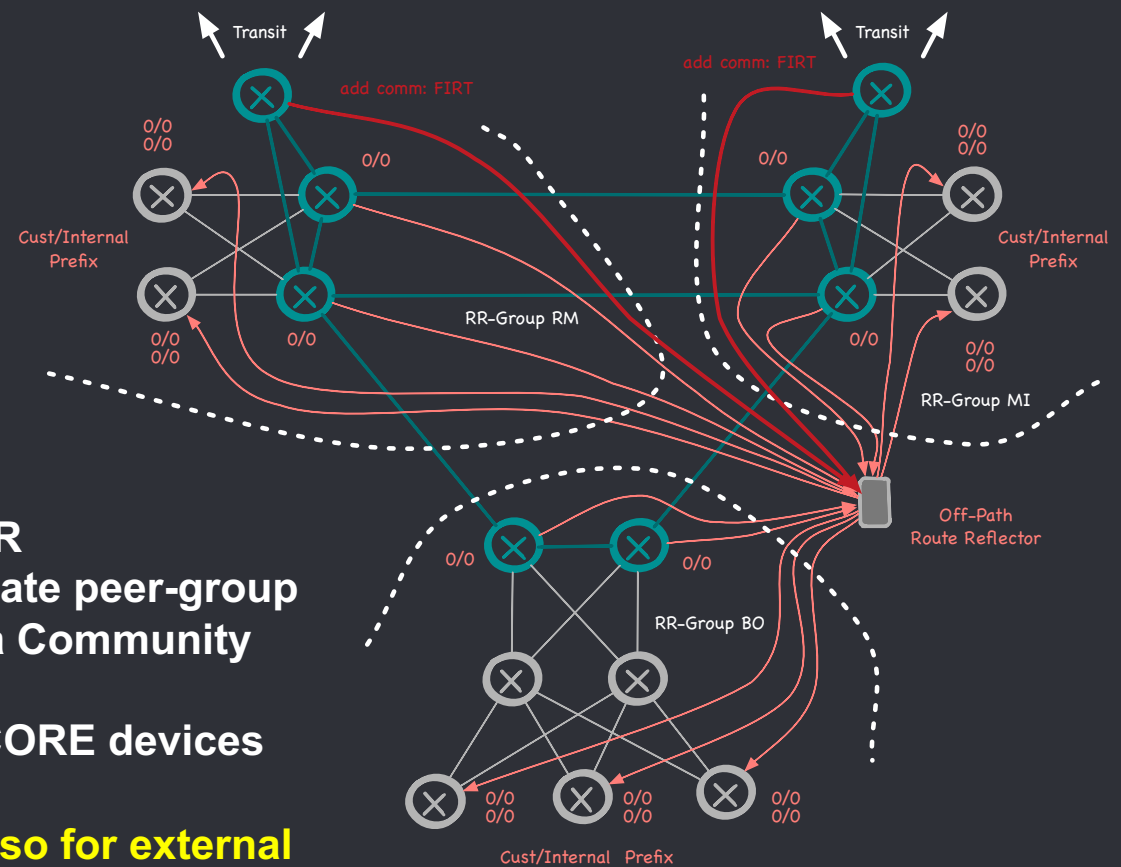
How combine DC/POP and Core RR

● Core vs POP/DC Route Reflecotors : almost different informations



- Core RR (Transit) holds the FIRT and Customer/Internal prefix
- DC/POP RR holds multiple default-routes and Customer/Internal prefix
- It's possible to combine the two infrastructure ? how ?

- Share same RR for all the routing information



Solution:

- Peer also Transit Routers with RR
- Configure Transit in the appropriate peer-group
- Mark all the external prefix with a Community
- Send EXTERNAL prefix only to CORE devices
- Leverage ADD-PATH and ORR also for external prefix and for all the sites -> HA (BGP PIC) and LB

● Complete RR Configuration

One peer-group per site

On Transit mark all received external prefix with a «FIRT» custom community

RR may use add-path to send multiple prefix/NH (when available) for both internal and external destinations

ORR will automatically select the two optimal prefix based on client IGP topology

prevent FIRT distribution on non-core device with a simple export policy

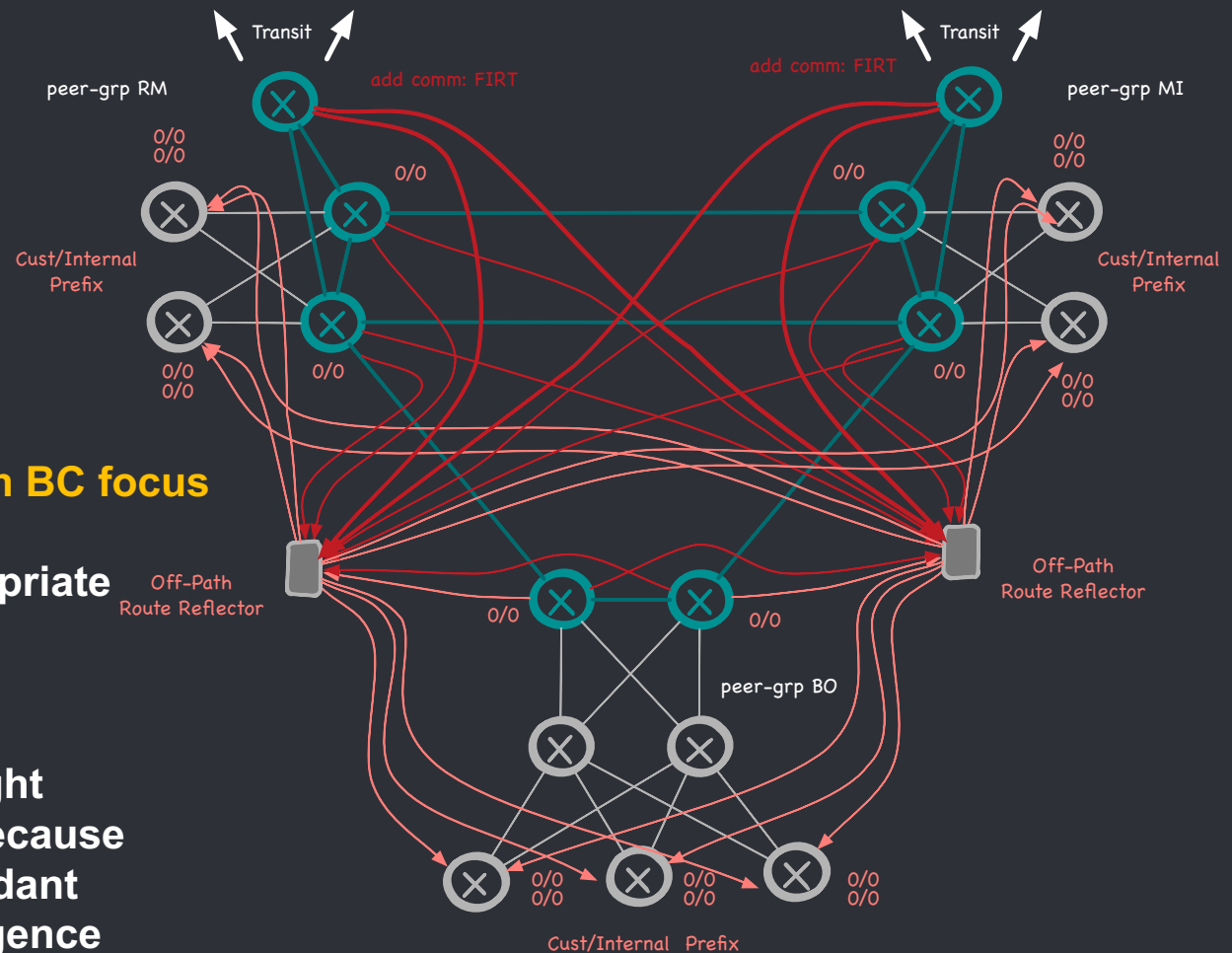
```
protocols {
  bgp {
    group RM-NAMEX {
      type internal;
      cluster 192.0.0.0;
      family inet {
        unicast {
          add-path {
            send {
              path-count 2;
            }
          }
        }
      }
    }
    ...
    optimal-route-reflection {
      igp-primary 192.0.0.1;
      igp-backup 192.0.0.2;
    }
    neighbor 192.0.0.1; // CORE
    neighbor 192.0.0.2; // CORE
    neighbor 192.0.0.3 export NO-FIRT; // PE
    neighbor 192.0.0.4 export NO-FIRT; // PE
  }
}

policy-options {
  policy-statement NO-FIRT {
    term reject-external-prefix {
      from community FIRT;
      then reject;
    }
  }
}
```

● How Many Route Reflectors ??

- **Number & Location of RR with BC focus**
- **No more than 2 RR per client**
- **DC location it's usually appropriate**
- **Prevent circular dependency**

- **In this case two RR are enough**
- **You can locate it anywhere because latency it's not critical: redundant information and local convergence**





RR Platform

- **Route Reflector it's not a router anymore**
- **Modern BGP implementations are optimized for multi-core and multi-thread**
- **Use VM with multiple core and high memory**
- **Server sizing based on nr. of client and nr. of prefixes**
- **More RR just to scale more clients and cover Business Continuity requirements**
- **ORR still not available in open/free implementations**

- **for IGP adjacency use a dedicated Interface/VLAN or a GRE Tunnel**

5

Summay

pros of modern bgp design

● Traditional vs Modern BGP design

key point of modern BGP design:

- one RR for all, two just for redundancy
- Flat BGP infrastructure for fast convergence
- RR are now Server and not Router
- Separated Control-Plane and Data-Plane
- Located anywhere for Infrastructure and Cost optimization
- **All client configuration are identical and without policy**
- All and redundant routing information in the clients but only necessary
- **Path Diversity**
- Load balancing (bgp **multipath**)
- High Availability based on **local convergence** (IGP)
- **All routing policy and optimization performed (almost automatically) on RR**
- Only one policy required on Route Reflector to select FIRT or NO-FIRT clients
- **Simple and Scalable**

● Migration from Traditional Design

Q: This is beautiful but **how to migrate** from a traditional BGP design ?

A: Obviously depends on how many customization/tricks you have deployed in your backbone but:

You can deploy the new infrastructure on top of the existing:

- ✓ Add the two new RR
- ✓ On core device check RIB capacity for new FIRT copies
- ✓ Peer all clients with the new RR
- ✓ use high AD/Preference on received prefix to prevent FIB install over existing
- ✓ compare old and new BGP prefixes to compare convergence

Use with route AD/Preference and progressively remove the old BGP cfg

- Key point for any design

*"**Simple can be harder than complex:** You have to work hard to get your thinking clean to make it simple. But it's worth it in the end because once you get there, you can move mountains." - Steve Jobs*

0

THANK YOU

Questions ?

Nicola Modena - CCIE #19119 JNCIE-SP #986

[linkedin.com/in/nmodena](https://www.linkedin.com/in/nmodena) - @nmodena

nicola@modena.to

disclaimer:
This is an original design performed during my consultancy activity
you can share and use just citing the source

a special tanks to:
Ivan Pepelnjak & Tiziano Tofoni for review and motivation for continuous improvement