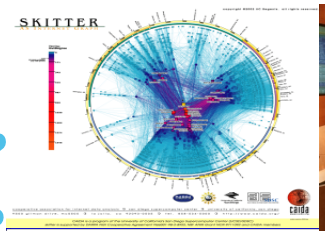
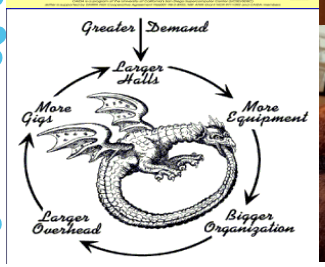
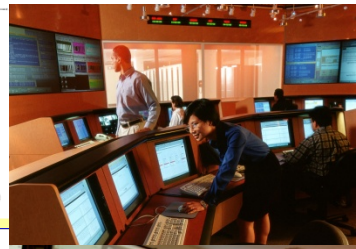
Remote Trigger Black Hole



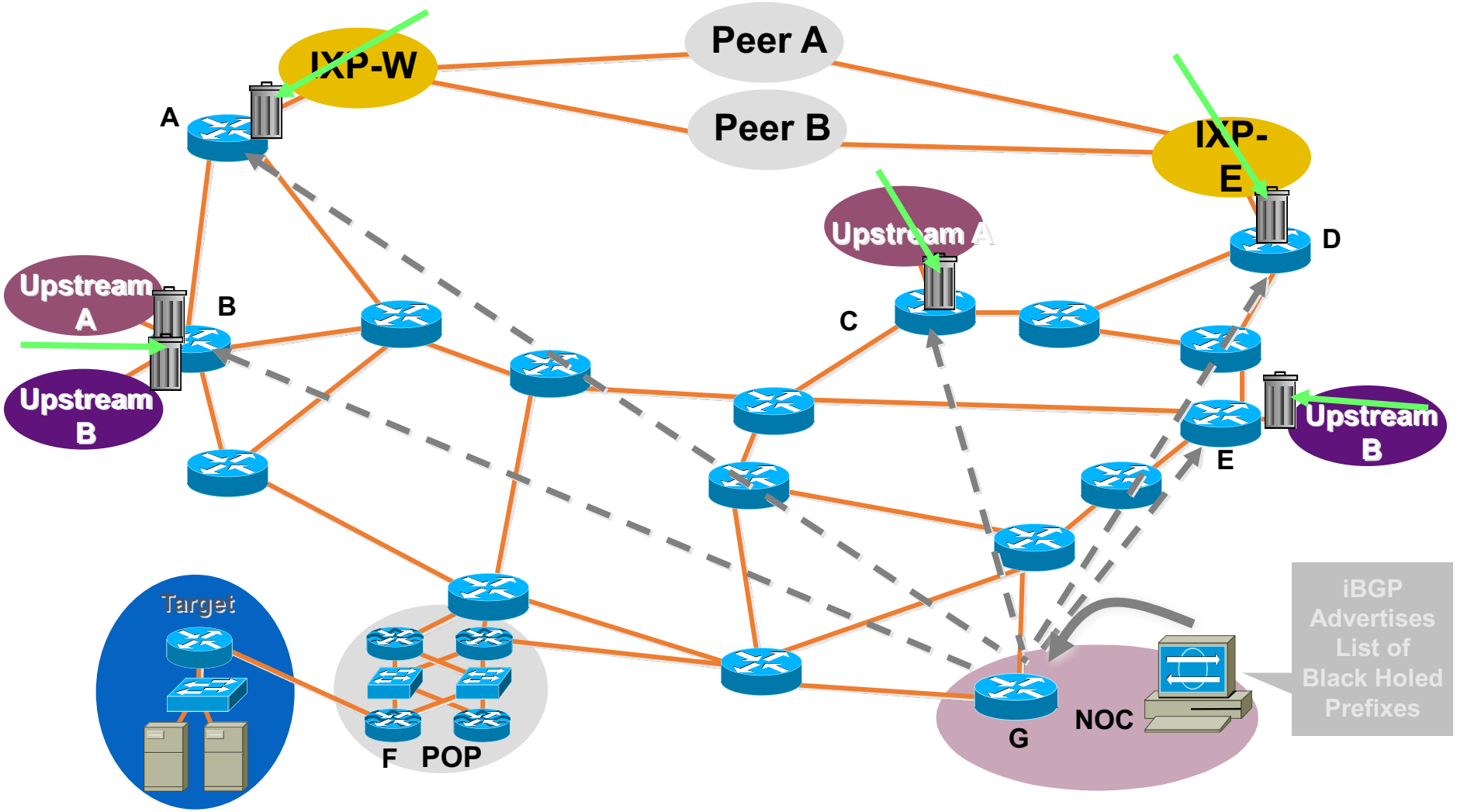




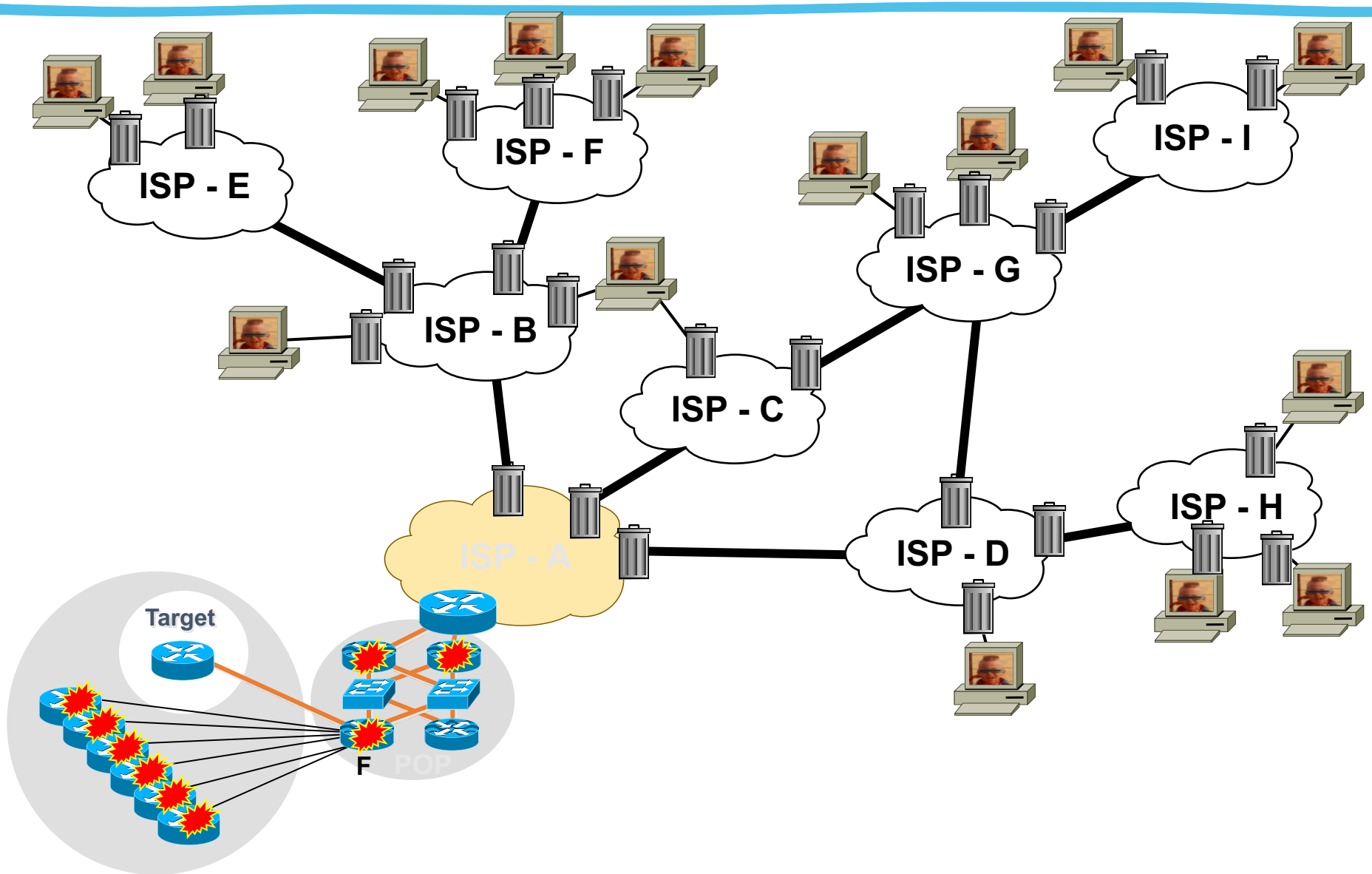
Remotely Triggered Black Hole Filtering

- We use BGP to trigger a network wide response to a range of attack flows.
- A simple static route and BGP will allow an SP to trigger network wide black holes as fast as iBGP can update the network.
- This provides SPs a tool that can be used to respond to security related events or used for DOS/DDOS Backscatter Tracebacks.

Customer is DOSed – After – Packet Drops Pushed to the Edge



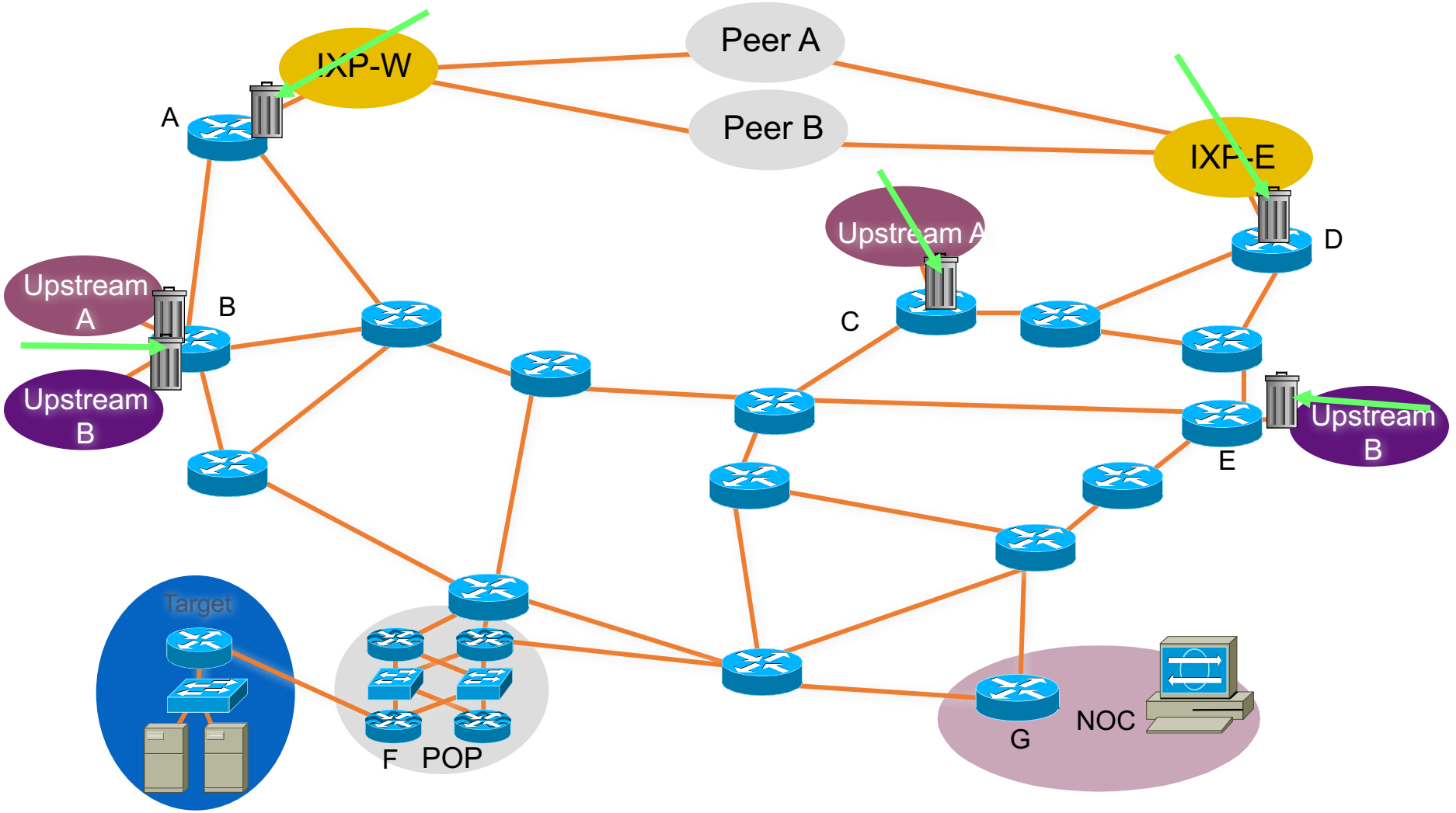
Inter-Provider Mitigation



What can you do to help?

- Remote Triggered Black Hole Filtering is the most common ISP DOS/DDOS mitigation tool.
- Prepare your network:
  - ftp://ftp-eng.cisco.com/cons/isp/essentials/ (has whitepaper)
  - ftp://ftp-eng.cisco.com/cons/isp/security/ (has PDF Presentations)
  - NANOG Tutorial:
    - http://www.nanog.org/mtg-0110/greene.html (has public VOD with UUNET)
  - APRICOT 2015
  - Turk, D., "Configuring BGP to Block Denial-of-Service Attacks", RFC 3882, September 2004.

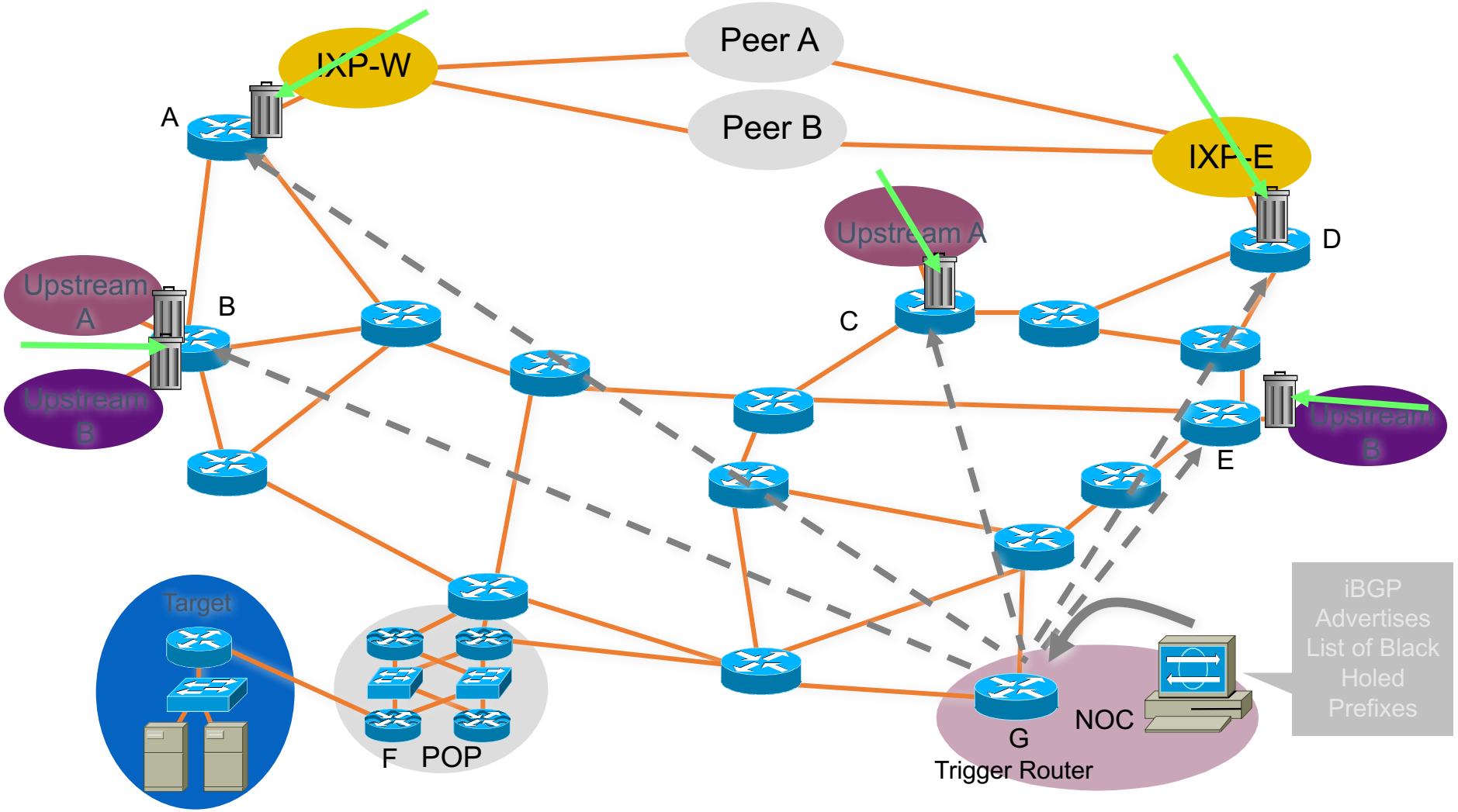
Blackhole Filtering



Remotely Triggered Blackhole Filtering

- We will use BGP to trigger a network wide response to an attack
- A simple static route and BGP will enable a network-wide destination address blackhole
  as fast as iBGP can update the network
- This provides a tool that can be used to respond to security related events and forms a foundation for other remote triggered uses
- Often referred to as RTBH

Remotely Triggered Blackhole Filtering



Remote Triggered Blackhole

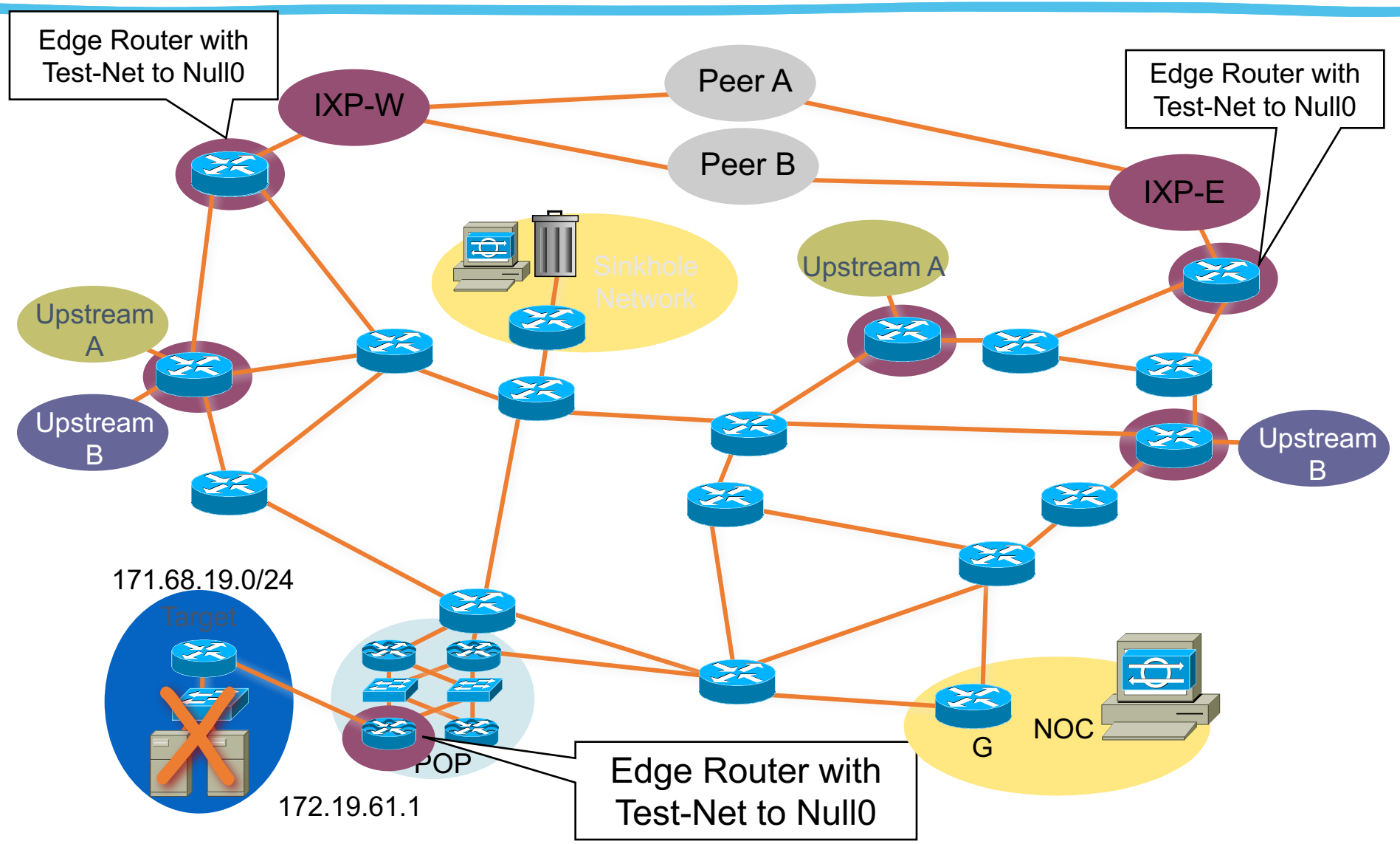
- Configure all edge routers with static route to Null0 (must use “reserved” network)
  - ip route 192.0.2.1 255.255.255.255 Null0
- Configure trigger router
  - Part of iBGP mesh
  - Dedicated router recommended
- Activate blackhole
  - Redistribute host route for victim into BGP with next-hop set to 192.0.2.1
  - Route is propagated using BGP to all BGP speaker and installed on routers with 192.0.2.1 route
  - All traffic to victim now sent to Null0

Step 1: Prepare All the Routers with Trigger

- Select a small block that will not be used for anything other than blackhole filtering; test
  Net (192.0.2.0/24) is optimal since it should not
  be in use
- Put a static route with a /32 from Test-Net—192.0.2.0/24 to Null 0 on every edge router on the network

```
ip route 192.0.2.1 255.255.255.255 Null0
```

Step 1: Prepare All the Routers with Trigger

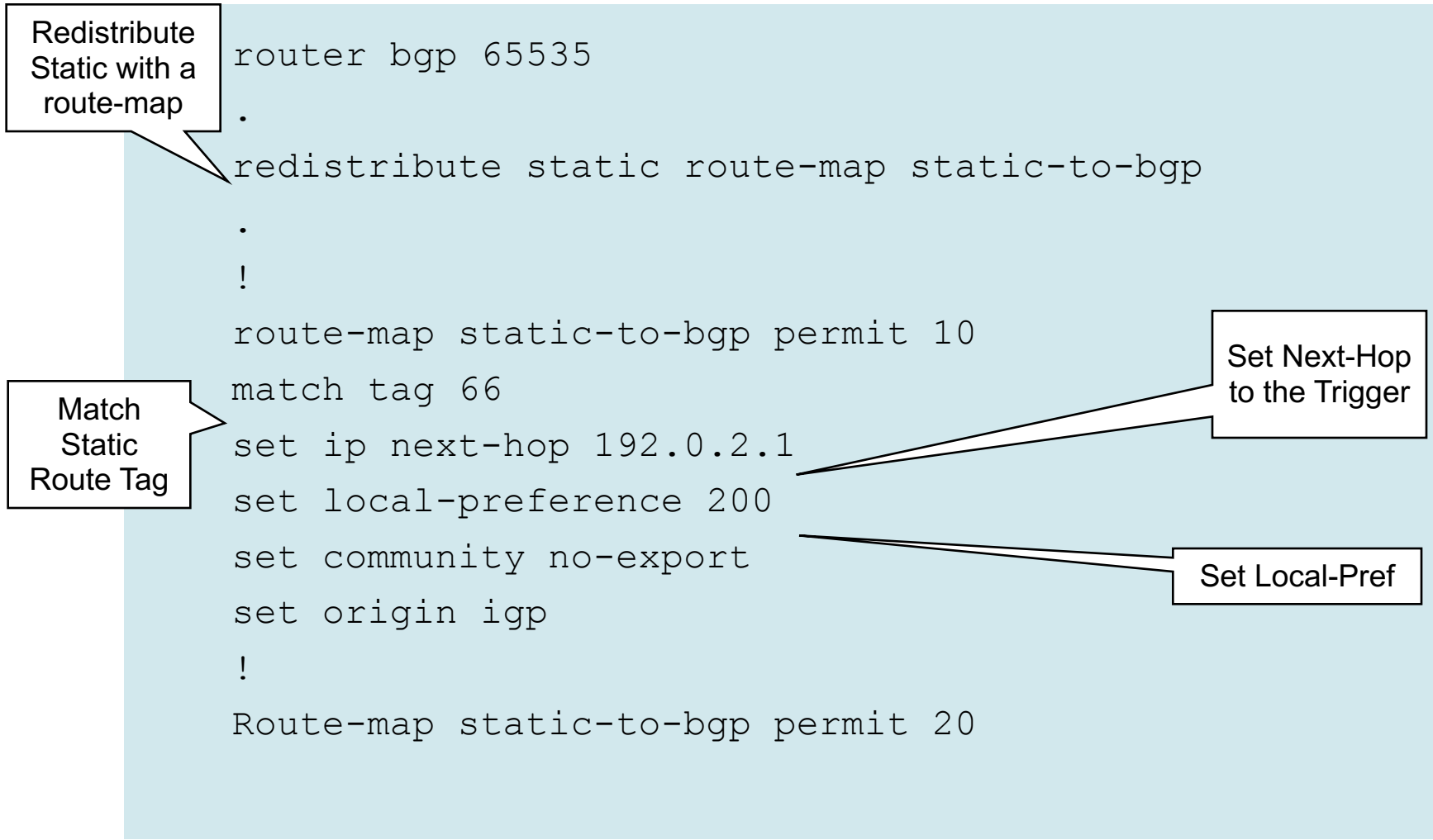


Step 2: Prepare the Trigger Router

The Trigger Router Is the Device That Will Inject the iBGP Announcement into the ISP's Network

- Should be part of the iBGP mesh—but does not have to accept routes
- Can be a separate router (recommended)
- Can be a production router
- Can be a workstation with Zebra/Quagga (interface with Perl scripts and other tools)

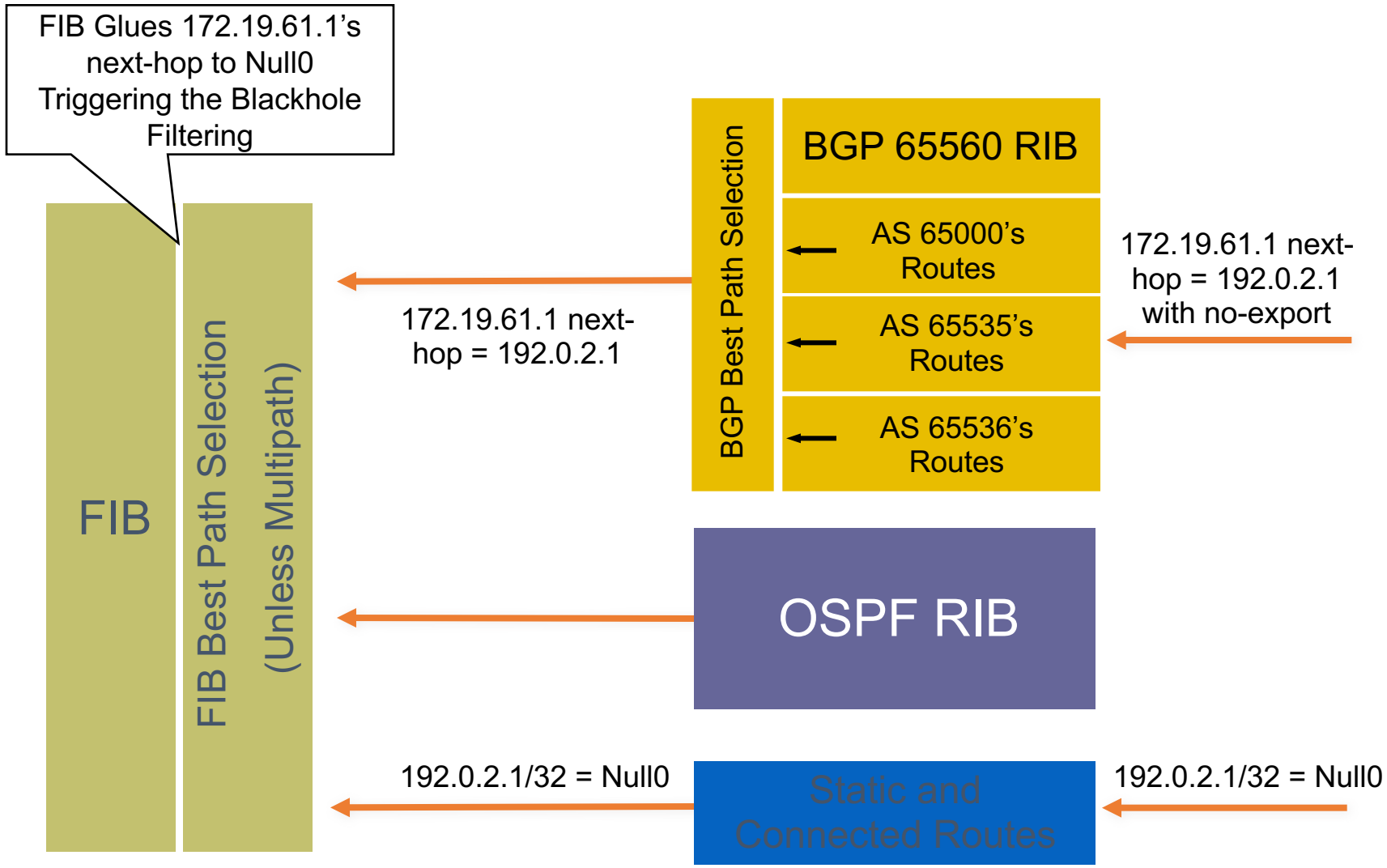
Trigger Router's Config



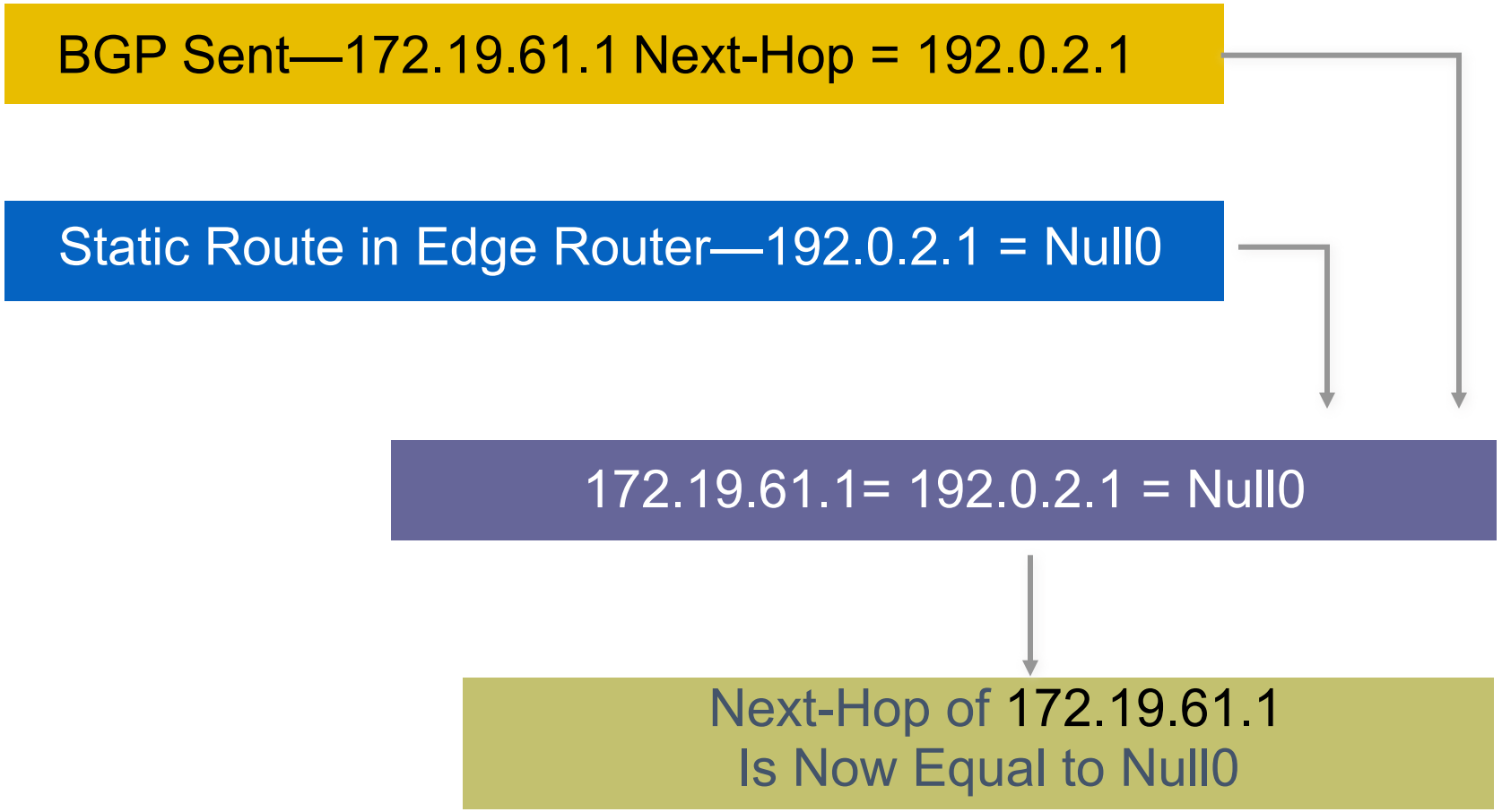
Step 3: Activate the Blackhole

- Add a static route to the destination to be blackholed; the static is added with the “tag 66” to keep it separate from other statics on the router
  - `ip route 172.19.61.1 255.255.255.255 Null0 Tag 66`
- BGP advertisement goes out to all BGP speaking routers
- Routers received BGP update, and “glue” it to the existing static route; due to recursion, the next-hop is now Null0

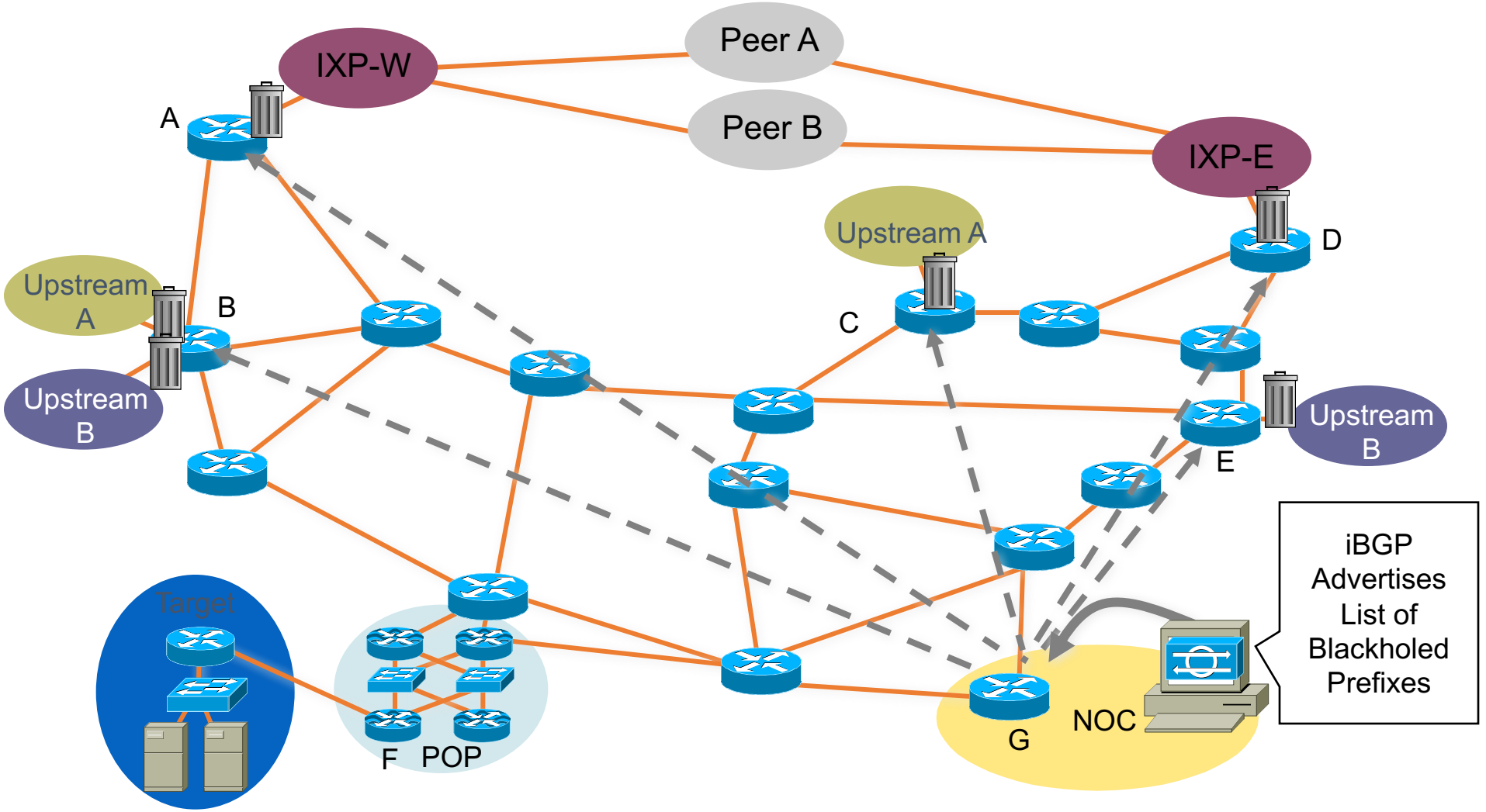
Step 3: Activate the Blackhole



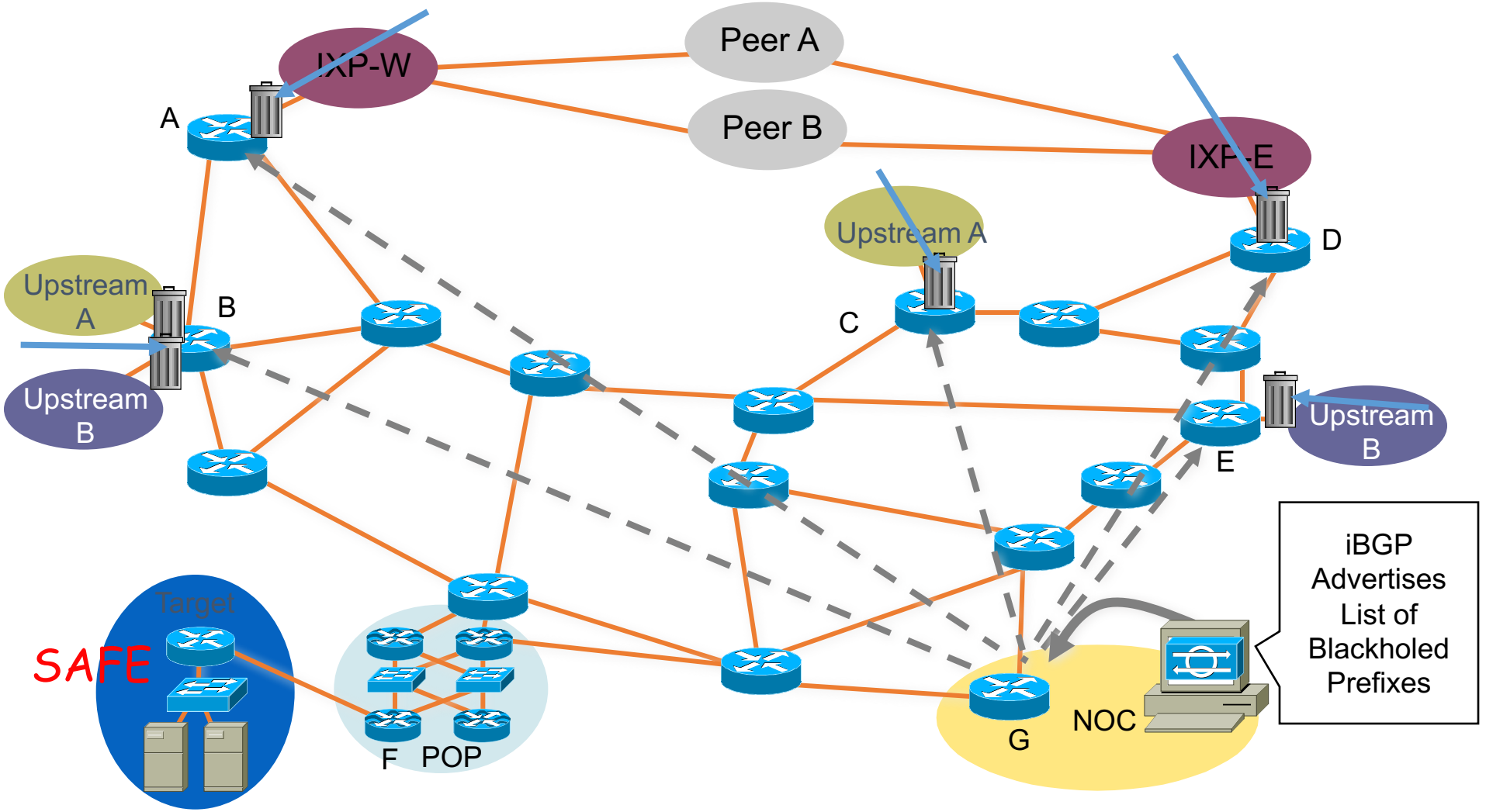
Step 3: Activate the Blackhole



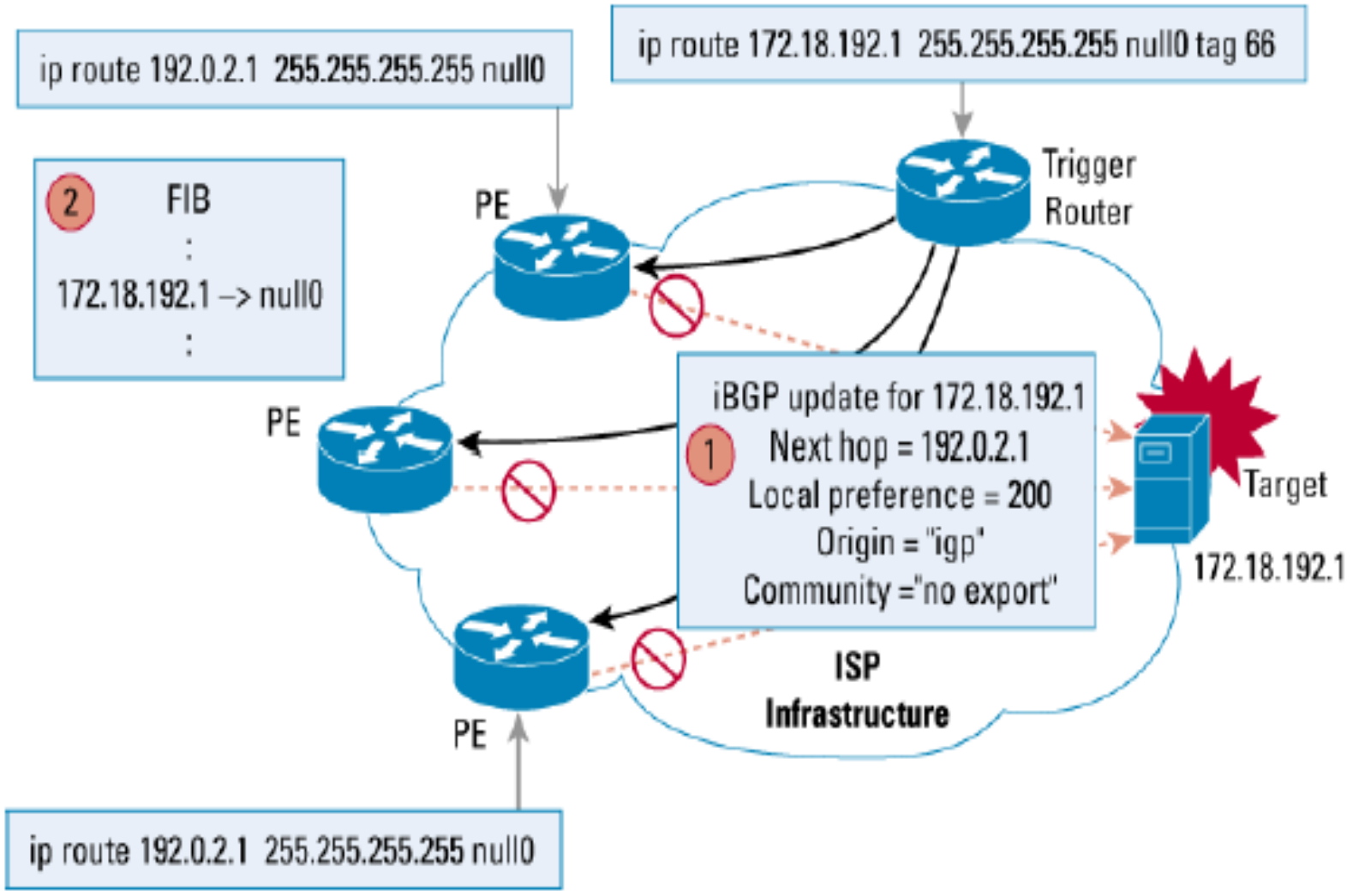
Step 3: Activate the Blackhole



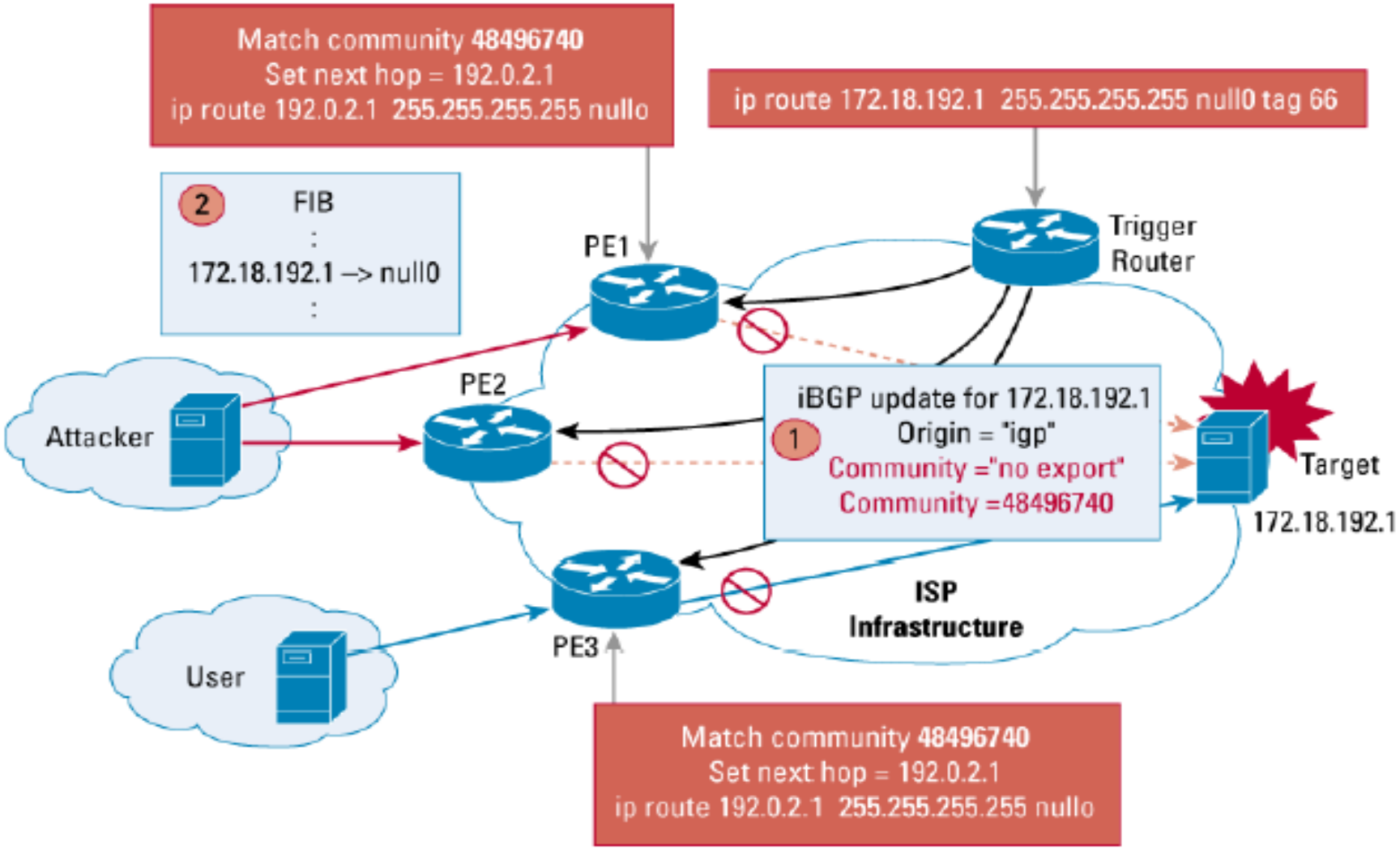
Customer Is DOSed (After) Packet Drops Pushed to the Edge



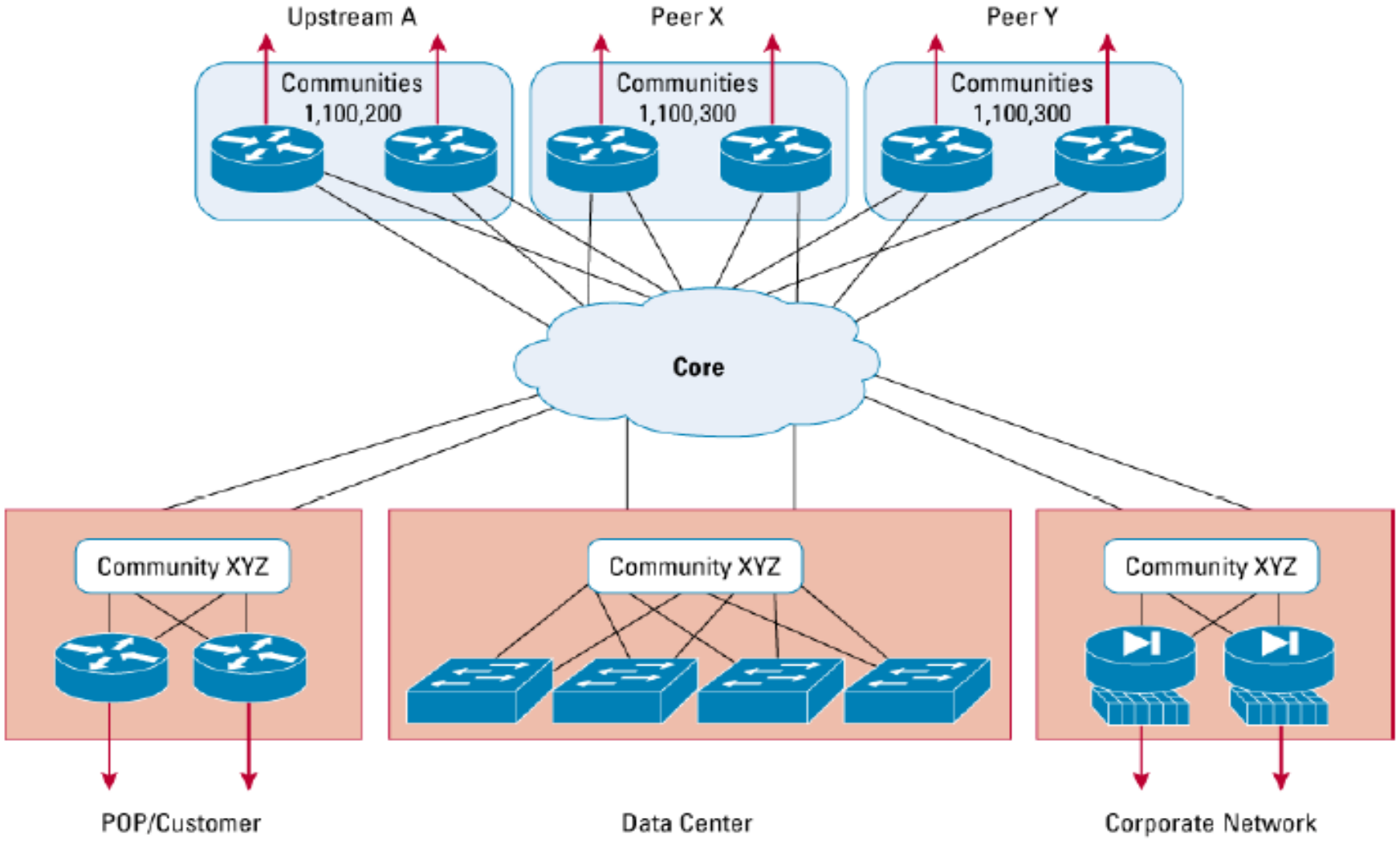
Deploying RTBH Filtering -- Next Hop Method



Deploying RTBH Filtering -- Communities Method

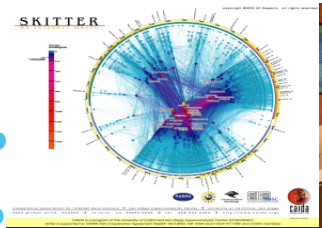


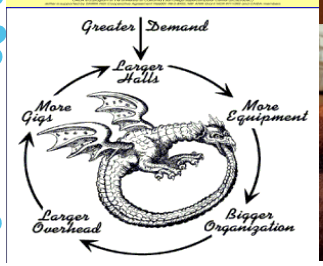
Deploying RTBH Filtering -- Communities Method

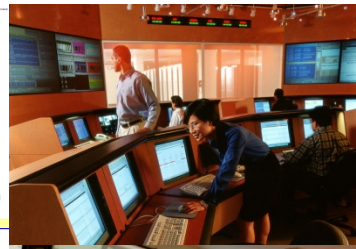


uRPF & Remote Trigger BlackHole

sRTBH







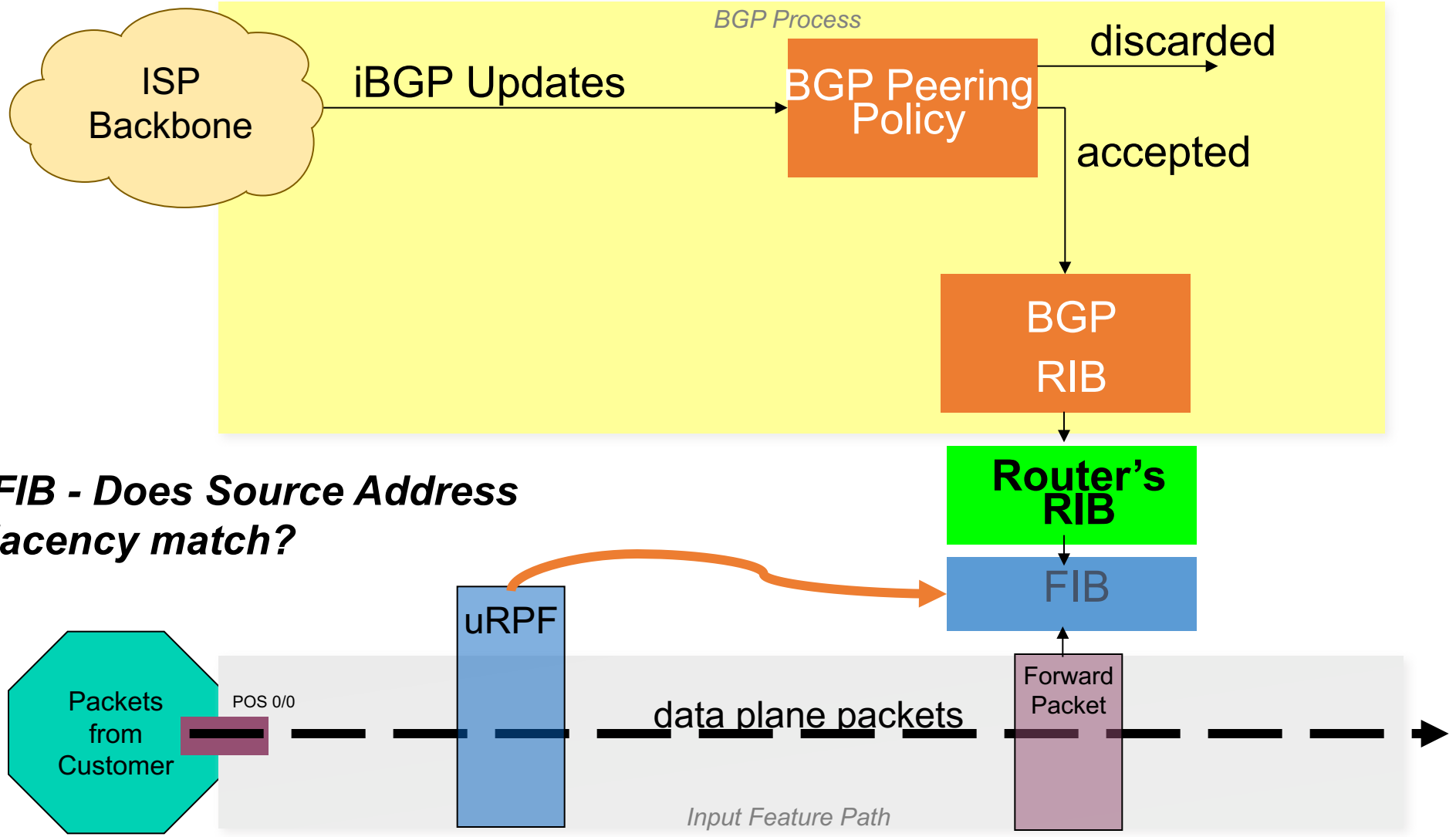
RTBH: Triggered Source Drops

- Dropping on destination is very important
  - Dropping on source is often what we really need
- Reacting using source address provides some interesting options:
  - Stop the attack without taking the destination offline
  - Filter command and control servers
  - Filter (contain) infected end stations
- Must be rapid and scalable
  - Leverage pervasive BGP again

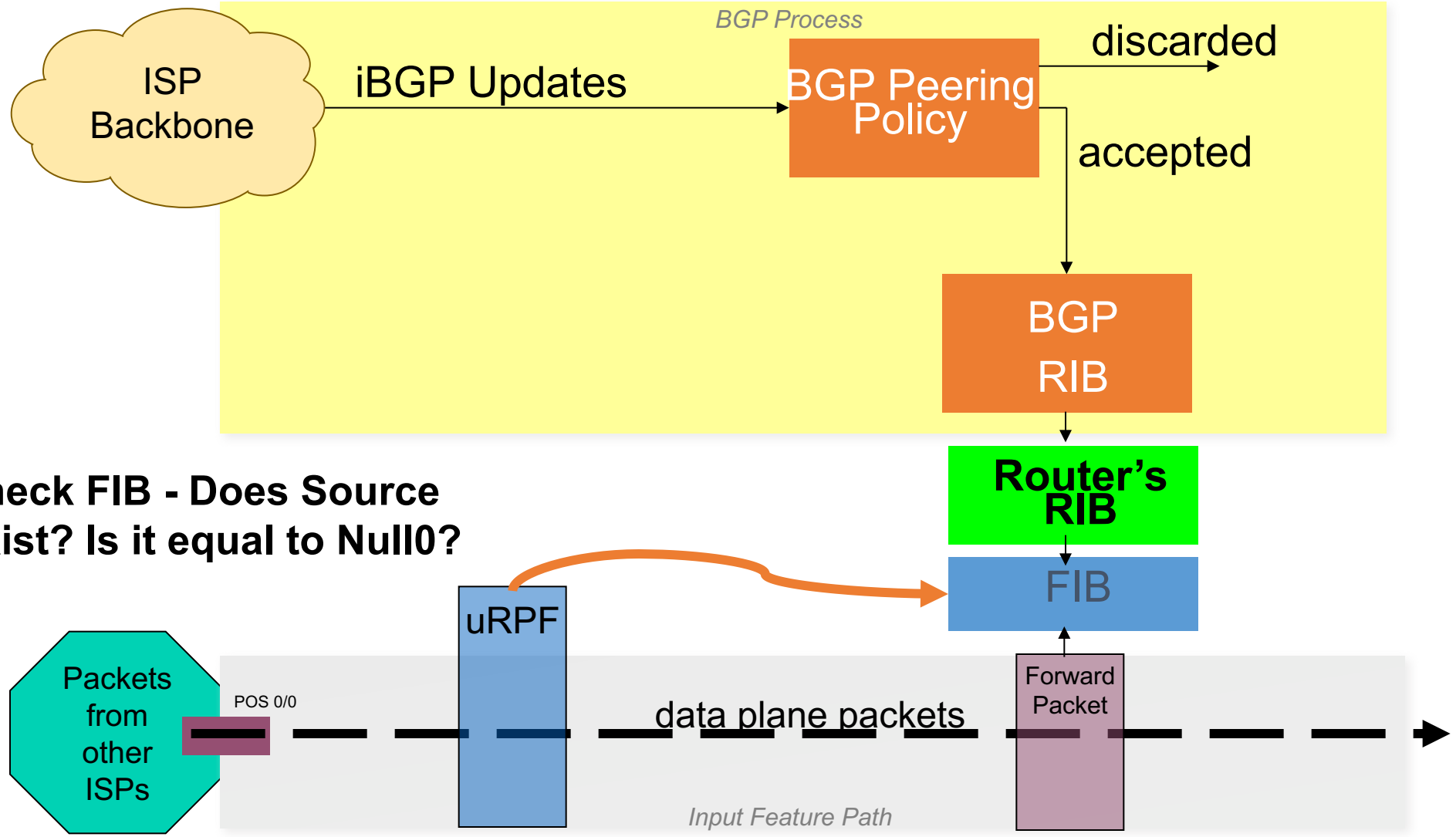
Unicast Reverse Path Forwarding (uRPF)

- uRPF is designed to drop traffic from sources which are:
  - Unreachable
  - Are received from “wrong” interface
- The purported source of ingress IP packets is checked to ensure that the route back to the source is “valid”
- Four flavors of uRPF:
  - Strict mode uRPF
  - Loose mode uRPF
  - Feasible Path uRPF
  - VRF Mode uRPF
- uRPF can be looked at as an dynamic filter because it always uses up-to-date information from the FIB table. ACL’s are static and have to be manually updated
- uRPF causes also minimal load on the router as it only adds one extra lookup in the FIB

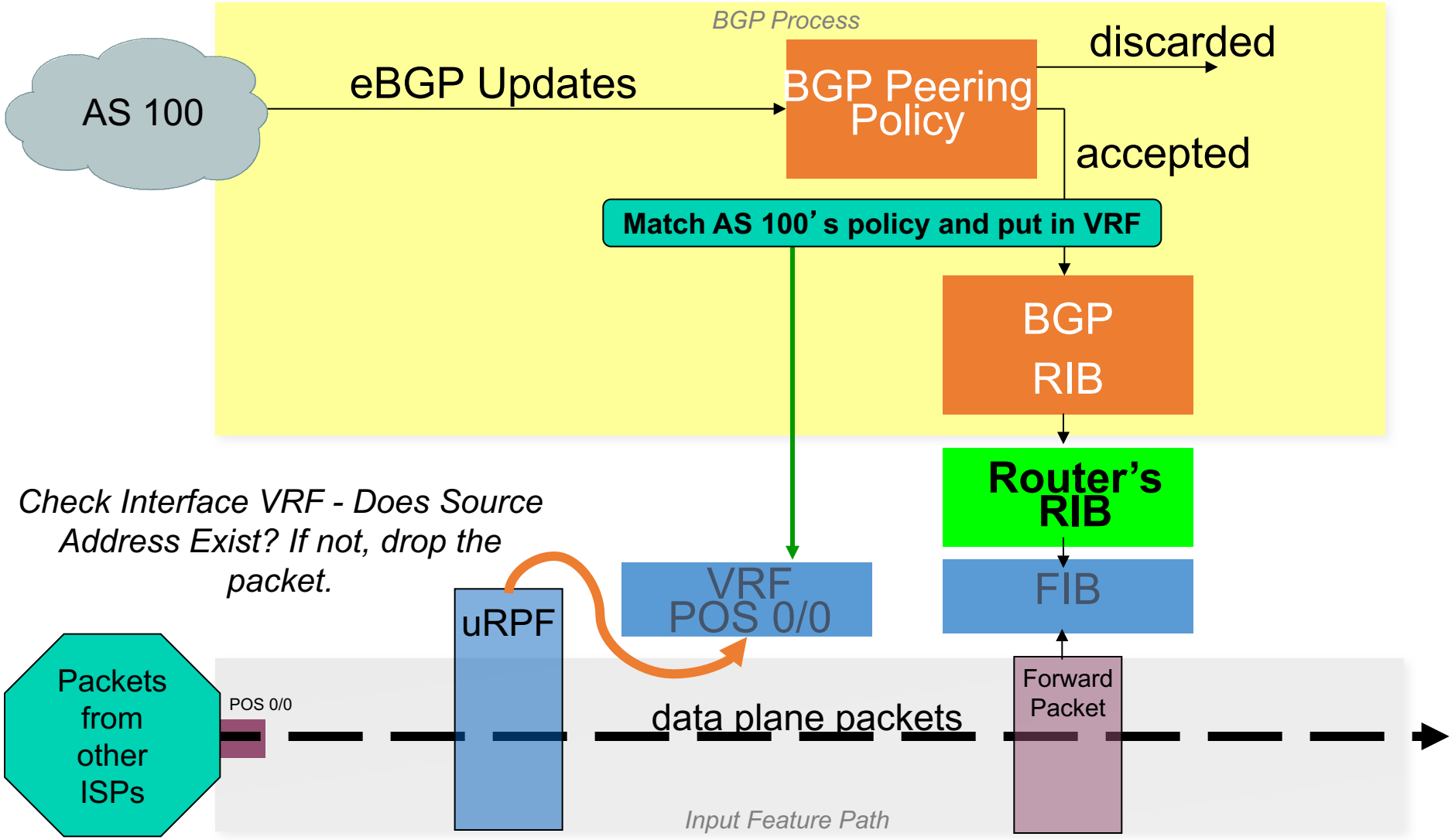
uRPF Strict Mode – BCP 38 Filtering



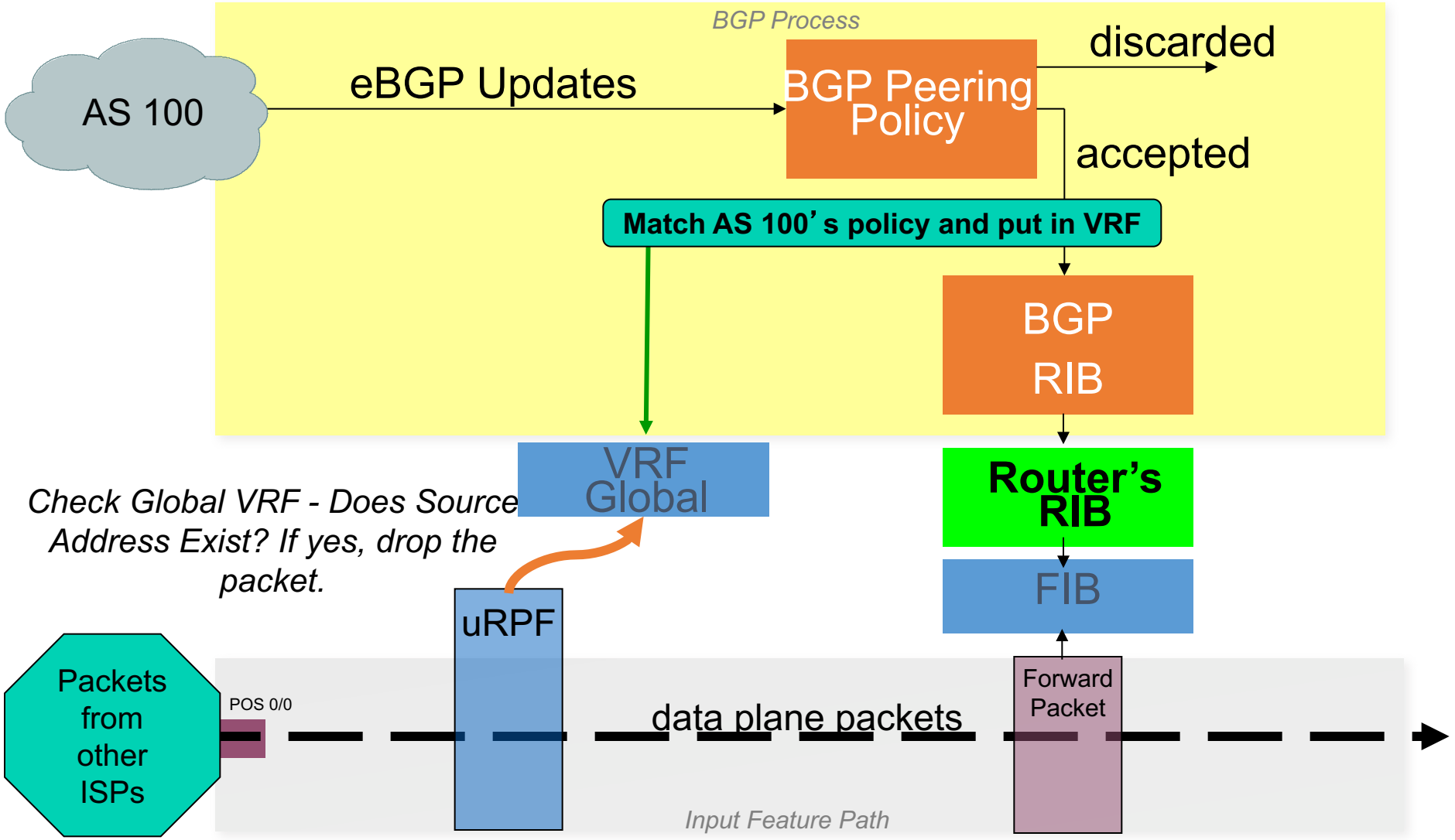
uRPF Loose Check



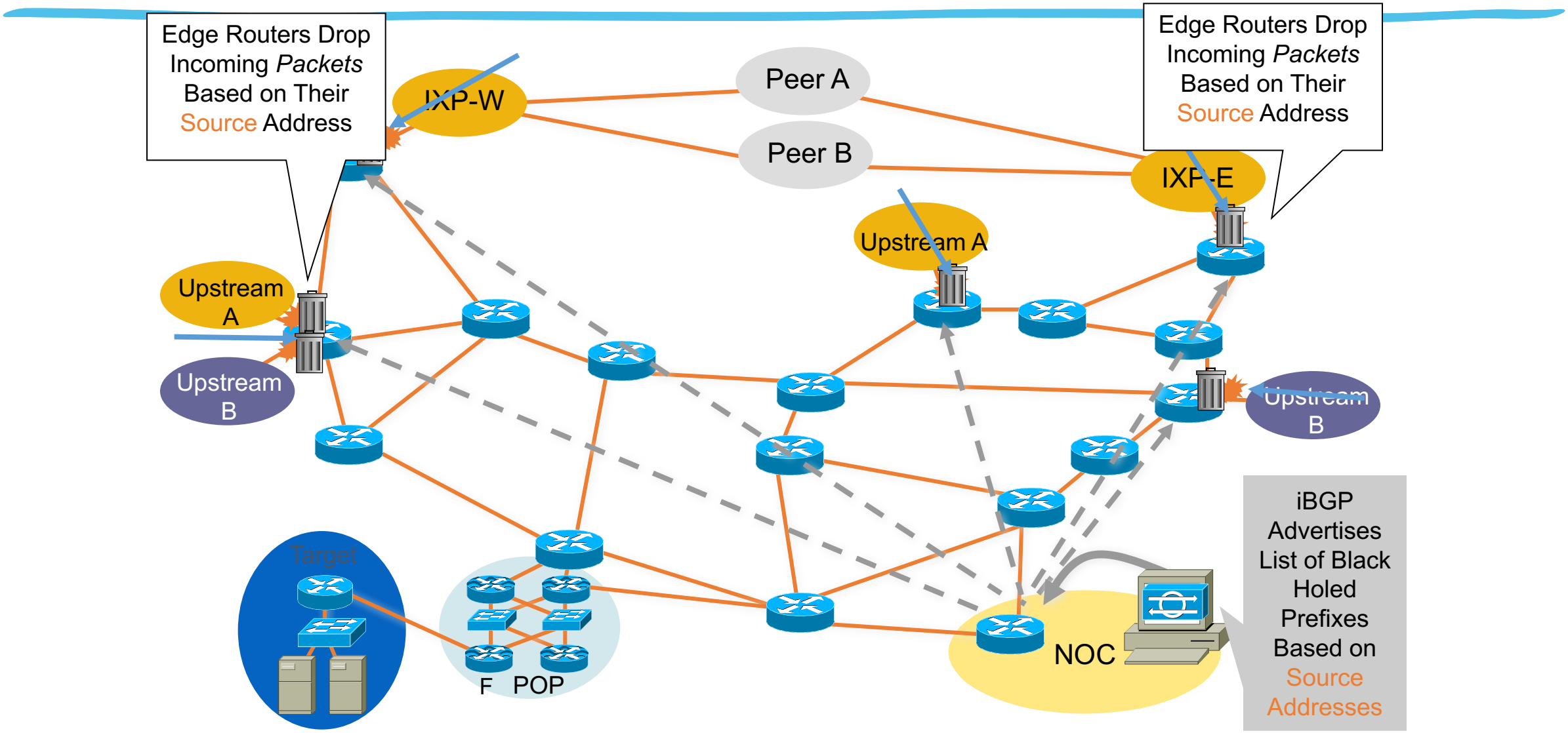
uRPF VRF Mode #1 – If not Exist – Drop



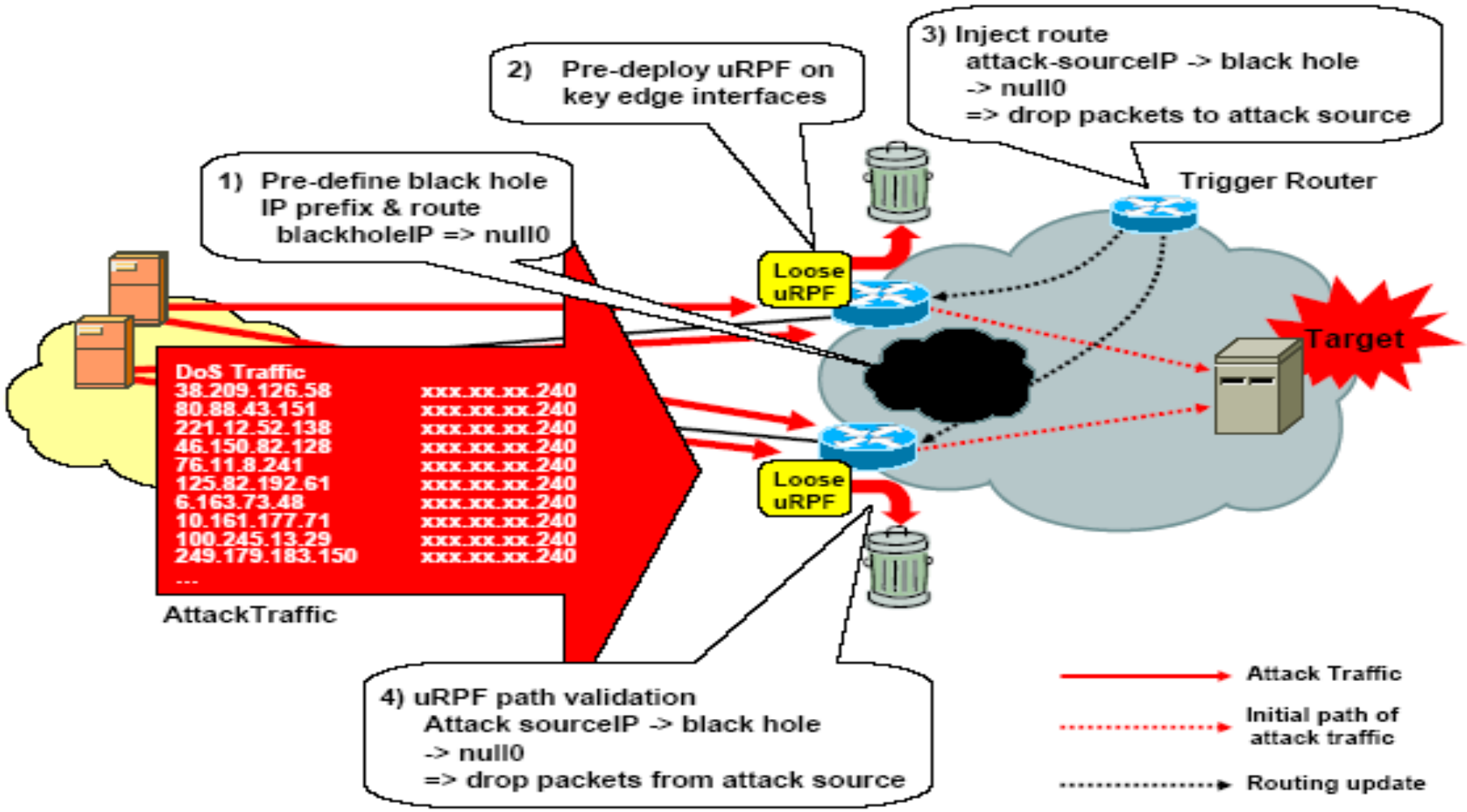
uRPF VRF Mode #2 – If Exist – Drop



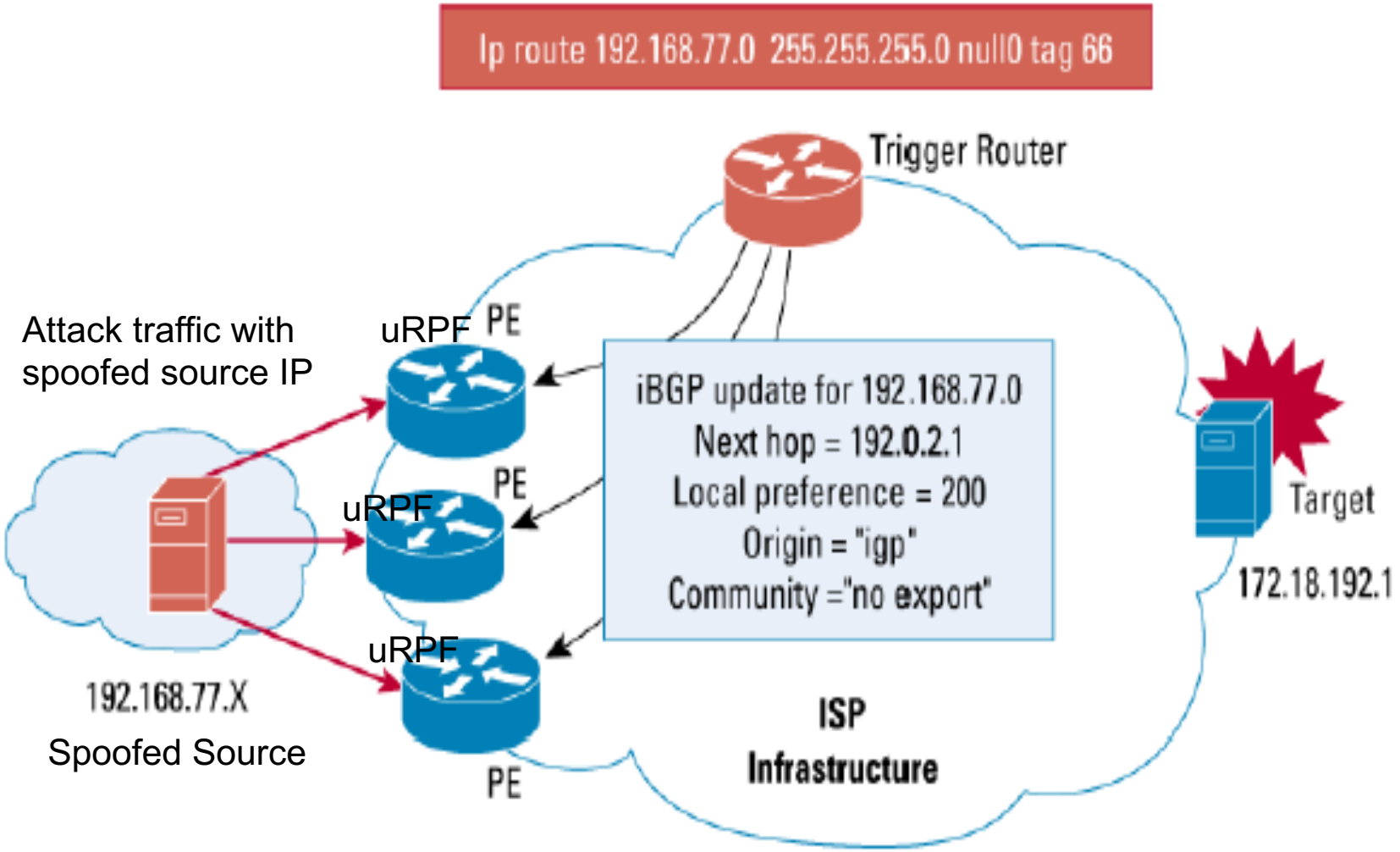
Customer Is DOSed (After) *Packet* Drops Pushed to the Edge



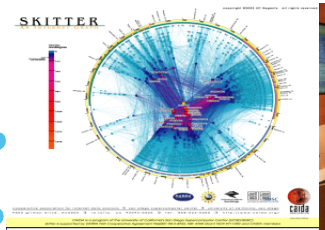
Source IP Based Remote Trigger Black Hole Filtering

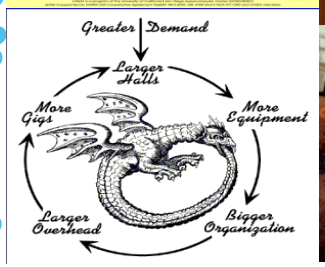


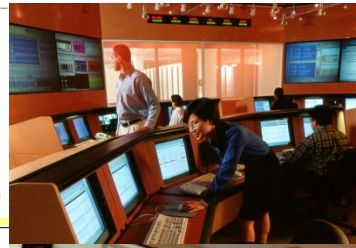
Source IP Based RTBH Filtering Example



Operational Example - NTT







What is wrong with Normal RTBH?

- Classic RTBH is an “all or nothing proposition:
- You throw away revenue generated by the victim IP address, in order to avoid congesting your upstream links

Damage Control is not Mitigation

- Selective RTBH should be considered as yet another tool in the toolbox when under duress.

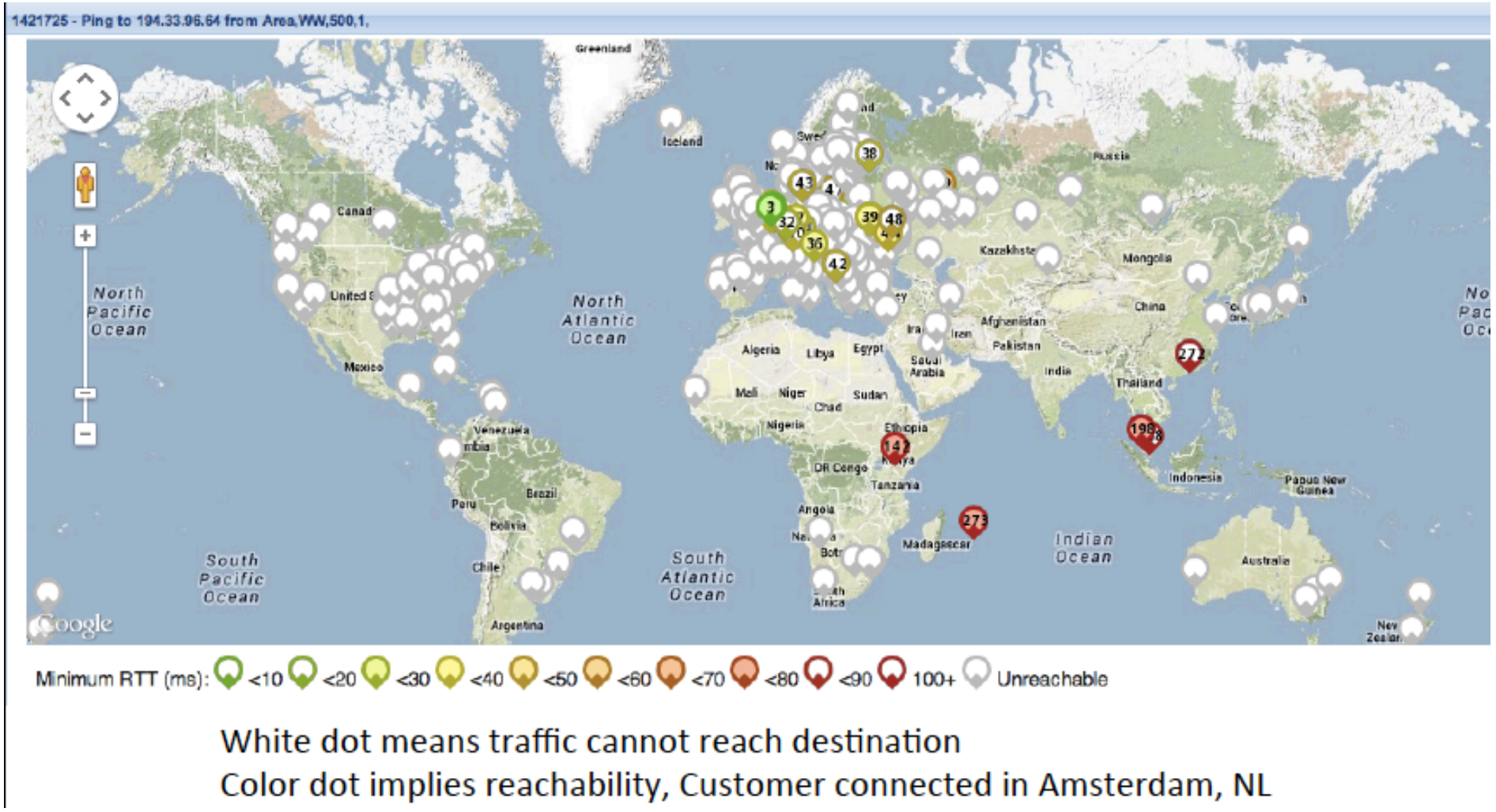
- Assertion #1: ”*it is better to remain partially reachable than not reachable during an all out DOS attack*.”

- Assertion #2: “*I can take a percentage of the DOS traffic, but not all.*”

NTT Effects: Discard outside of 1000 km radius

Customer connects in Amsterdam, Netherlands
White dot means traffic cannot reach destination
Colored dot implies reachability

NTT Effect: Discard outside this country

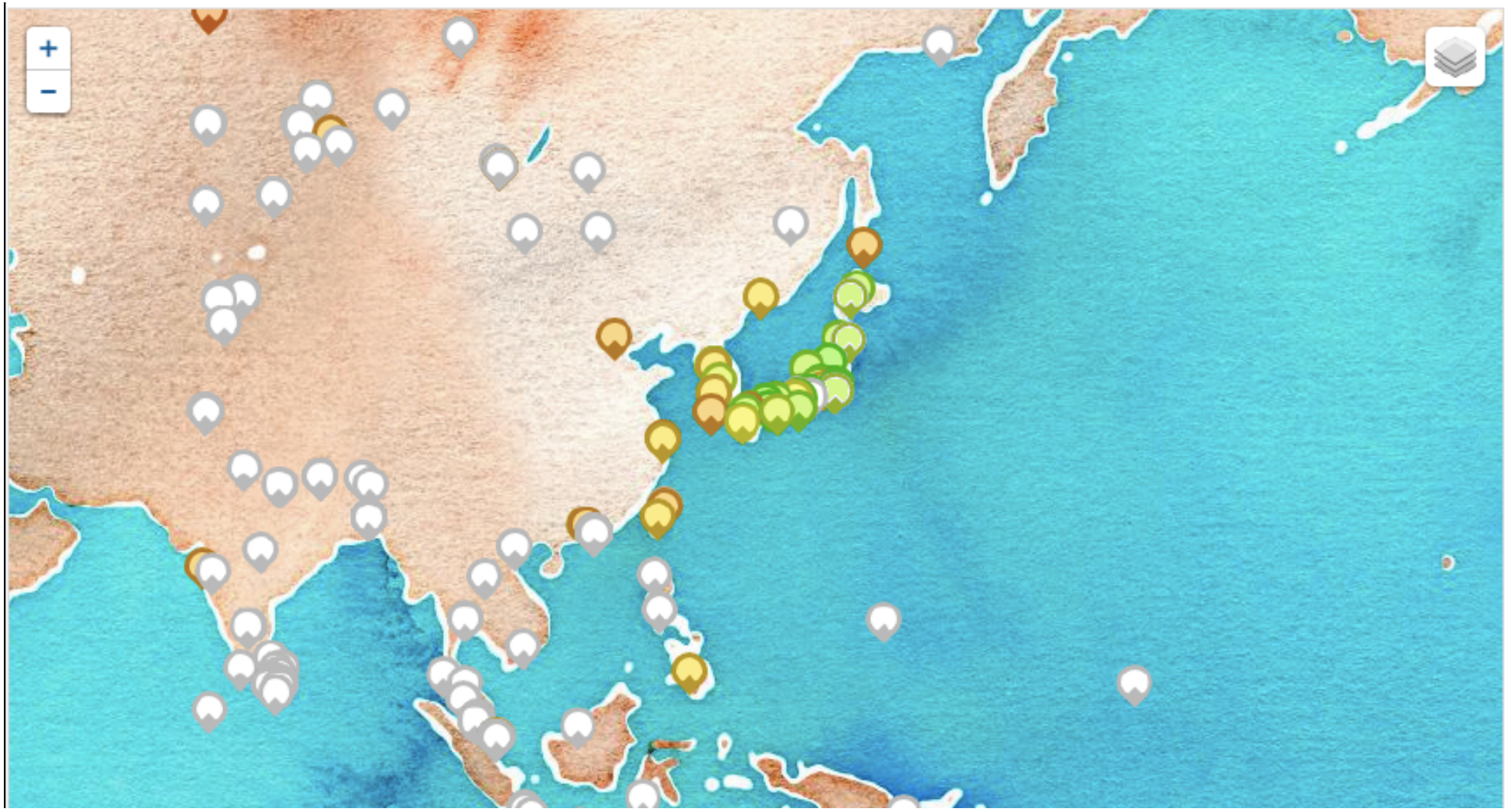


White dot means traffic cannot reach destination

Color dot implies reachability, Customer connected in Amsterdam, NL

“Discard outside of NL” allows for NL Reachability

21914:664 “Only blackhole outside the country the announcement originated”



How NTT Sets this up (Carrier Example)

- Focus on four features:

Scope	End-user BGP community
Outside 'This' country	15562:664
Outside 'This' continent	15562:660
Outside 1000 KM radius	15562:663
Outside 2500 KM radius	15562:662

- "This" means "where the customer interconnection is located"
- Distance from Edge router to Edge router in the SP's network "as the crow flies" (not the actual fiber path length). Can only be guaranteed for NTT's backbone.

Assign your routers some integers ….

- A detailed naming scheme for your network elements which means something make life easier to troubleshoot.

name	Continent id	ISO31661	City ID	Latitude, Longitude
tky.jp	3	392	46	35.65671, 139.80342
sjo.us	1	840	29	37.44569,-122.16111
dal.us	1	840	33	32.80096, -96.81962
nyc.us	1	840	26	40.71780, -74.00885
lon.uk	2	276	23	51.51173, -0.00197
ams.nl	2	528	20	52.35600, 4.95068
sto.se	2	752	22	59.36264, 17.95560

Router Specific Configurations (Ciscoish)

nyc.us:

```
ip community-list THIS:METRO     seq 5 permit 65123:10026
ip community-list THIS:COUNTRY   seq 5 permit 65123:840
ip community-list THIS:CONTINENT seq 5 permit 65123:1000
```

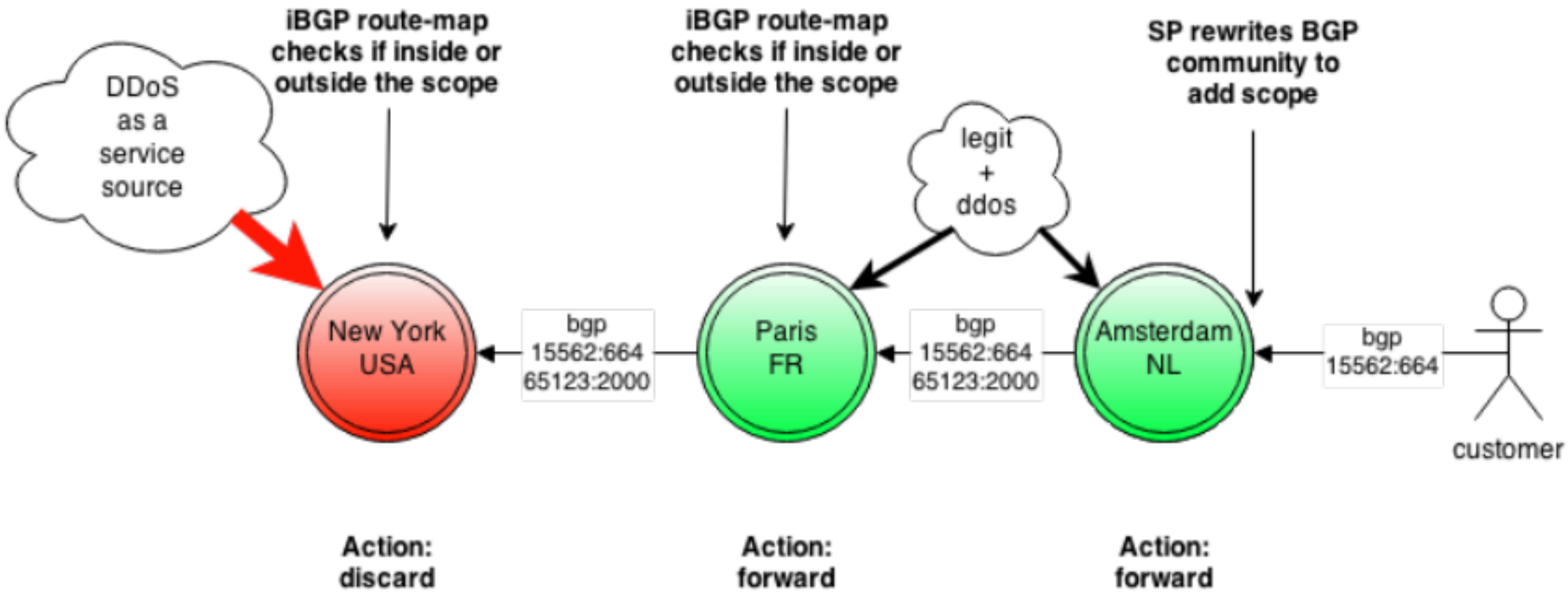
lon.uk:

```
ip community-list THIS:METRO     seq 5 permit 65123:20023
ip community-list THIS:COUNTRY   seq 5 permit 65123:276
ip community-list THIS:CONTINENT seq 5 permit 65123:2000
```

ams.nl:

```
ip community-list THIS:METRO     seq 5 permit 65123:20020
ip community-list THIS:COUNTRY   seq 5 permit 65123:528
ip community-list THIS:CONTINENT seq 5 permit 65123:2000
```

What happens where …..



iBGP Inbound route-map

```
ip route 10.0.0.1 255.255.255.255 null0

route-map INBOUND-IBGP permit 100
  match community  15562:666          ! classic blackhole community
  set ip next-hop 10.0.0.1            ! discard

route-map INBOUND-IBGP permit 200
  match community  15562:660 15562:662 15562:663 15562:664
  continue 1100                       ! Jump over regular 'accept' @ 1000
                                      ! towards scope checking

route-map INBOUND-IBGP permit 1000
                                      ! No match statement == accept anything

route-map INBOUND-IBGP permit 1100
  match community  THIS:METRO THIS:COUNTRY THIS:CONTINENT
                                      ! If match is found, accept prefix and stop
                                      ! evaluating the route-map


route-map INBOUND-IBGP permit 1101    ! Anything that arrives here: discard
  set ip next-hop 10.0.0.1
```

Customer Facing Route-Map

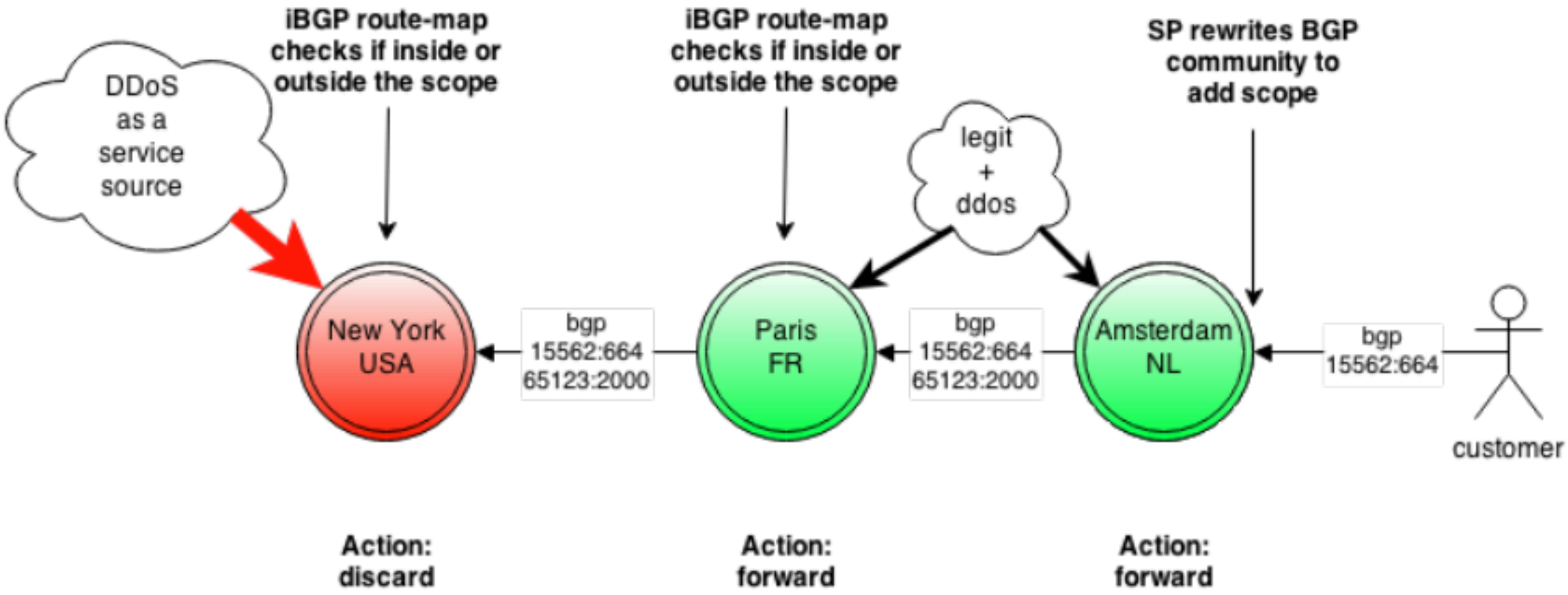
```
01.  route-map IMPORT:FROM:CUSTOMER-A permit 200
02.    match ip address prefix-list CUSTOMER-A-PREFIXES
03.    match community 15562:666
04.    set community  no-export additive
05.    set ip next-hop 10.0.0.1

06.  route-map IMPORT:FROM:CUSTOMER-A permit 300
07.    match ip address prefix-list CUSTOMER-A-PREFIXES
08.    match community SCOPED:ACTION
09.    continue 600              ! Remember this jump !

10.  route-map IMPORT:FROM:CUSTOMER-A permit 400
11.    match ip address prefix-list CUSTOMER-A-PREFIXES
12.    set local-preference 650

13.  route-map IMPORT:FROM:CUSTOMER-A deny 500
```

Customer Facing Route-Map (continued)

Add/Rewrite scoping information when a ‘scoped action’ is used

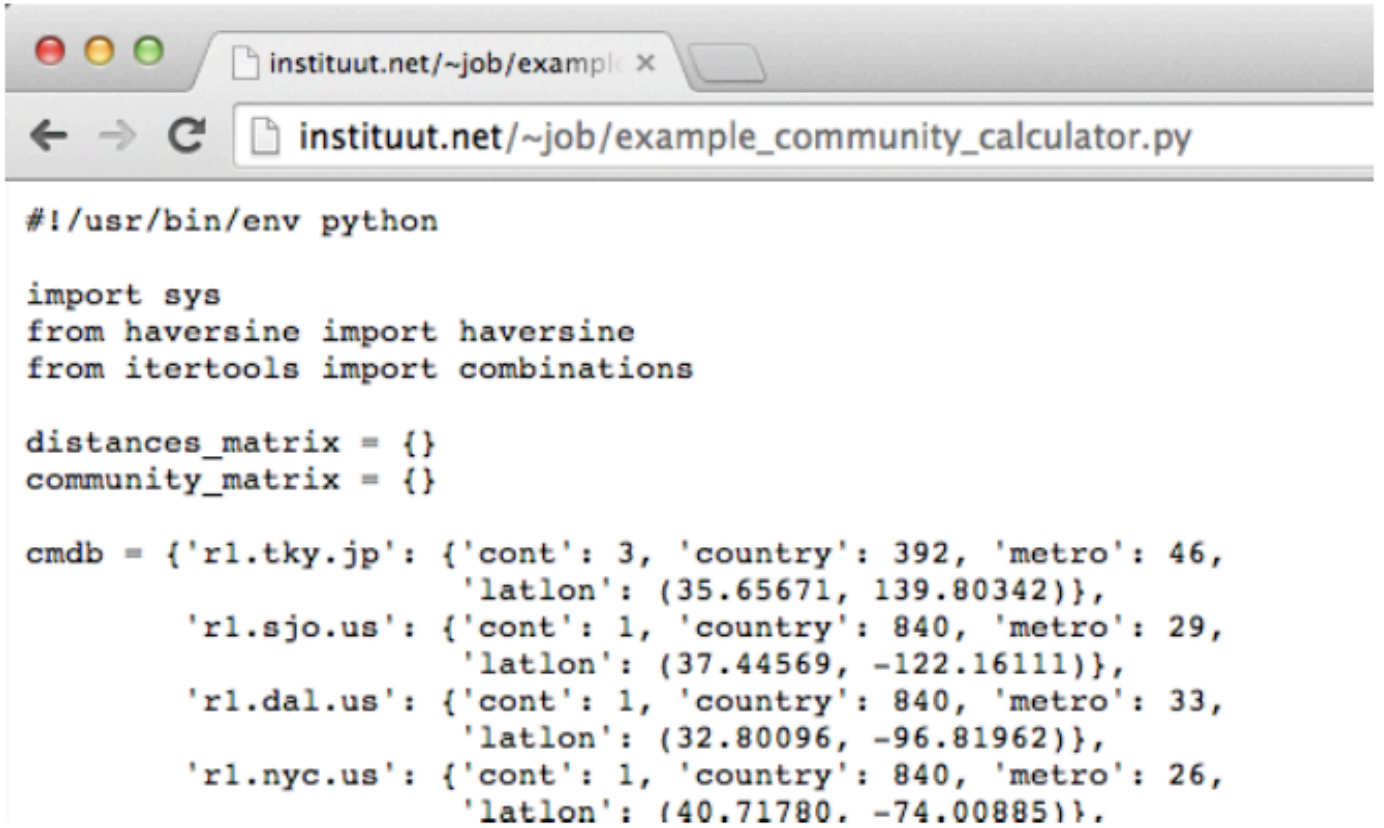
```
14.  route-map IMPORT:FROM:CUSTOMER-A permit 600          ! Here is 600 again
15.    match community  OUTSIDE:1000KM:RADIUS:DISCARD    ! 15562:663
16.    set community  65123:10029 additive

17.  route-map IMPORT:FROM:CUSTOMER-A permit 700
18.    match community  OUTSIDE:2500KM:RADIUS:DISCARD    ! 15562:662
19.    set community  65123:10033 65123:10029 additive

20.  route-map IMPORT:FROM:CUSTOMER-A permit 900
21.    match community  OUTSIDE:THIS:COUNTRY:DISCARD     ! 15562:664
22.    set community  65123:840 additive

23.  route-map IMPORT:FROM:CUSTOMER-A permit 1100
24.    match community  OUTSIDE:THIS:CONTINENT:DISCARD   ! 15562:660
25.    set community  65123:1000 additive
```

What happens where (customer facing) …



But wait a minute …

- How do you figure out what needs to be rewritten?



Gratis download!
http://instituut.net/~job/example_community_calculator.py

Proof: Software is cool – SDN finally arrived!

```
derp:~ job$ wget -q http://instituut.net/~job/example_community_calculator.py

derp:~ job$ python example_community_calculator.py

r1.lon.uk - rewrite targets:
1000 km: 65123:20020 65123:276
2500 km: 65123:2000

r1.dal.us - rewrite targets:
1000 km: 65123:10033
2500 km: 65123:840

r1.sjo.us - rewrite targets:
1000 km: 65123:10029
2500 km: 65123:10033 65123:10029

r1.nyc.us - rewrite targets:
1000 km: 65123:10026
2500 km: 65123:10026 65123:10033

<snip>
```

The router integers are essential ….

… provides grouping of routers, which the software/route-maps use ….

COMM /RTR	:276	:2000	:10033	:840	:10029	:10026	:30046	:20022	:20020
ams.nl		X							X
lon.uk	X	X							
sto.se		X						X	
nyc.us				X		X			
dal.us			X	X					
sjo.us				X	X				
tky.jp							X		

(incomplete table, but you get the gist…)

Considerations

- Automate ***all*** route-map deployments (actually, automate everything!)
- Use or make a CMDB where you store integers
- Selective Blackholing is a pretty advanced feature..
  with very little router specific configuration ☺
- Can be deployed on any vendor. Crappy vendors are not an excuse. This requires no extra CAPEX
- Customers don't ask for this feature because they don't know it exists (yet)
- Saves **both** the service provider and customer money: win/win

NTT Today ..

BGP Community	Routing policy
2914:666	Global discard / classic blackhole
2914:660	Only blackhole outside the region
2914:664	Only blackhole outside the country
2914:661	Only blackhole inside the region
2914:663	Only blackhole inside the country

NTT Customers

Announce the selective blackhole prefix but don't blackhole it yourself! To achive "blackhole outside this region"

This is probably bad:

```
    set routing-options rib inet6.0 static route
2001:67c:208c::2/128 discard community 2914:660
```

Something like this is better:

```
 set routing-options rib inet6.0 static route
2001:67c:208c::2/128 no-install community 2914:660
```

Questions – Peers helping Peers

Job Snijders

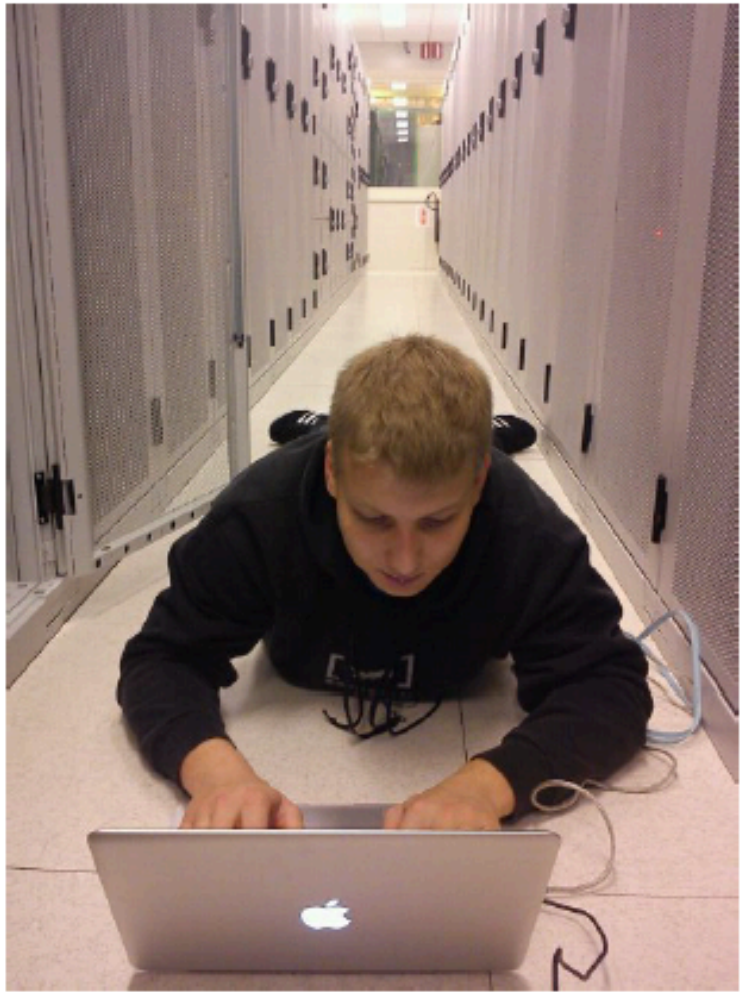
NTT Communications

Founder of NLNOG RING

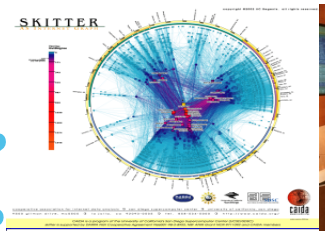
Twitter: @JobSnijders
Email: job@instituut.net

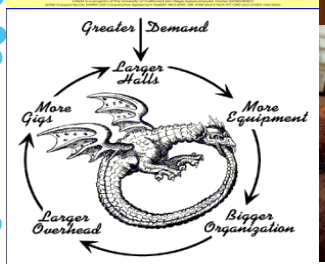
Hobbies: IP Routing, LISP, MPLS, IPv6, RPSL

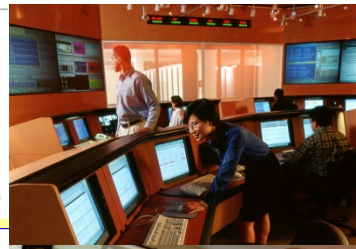
Shoe size: 45/EU



The Essential RTBH and BGP FlowSpec

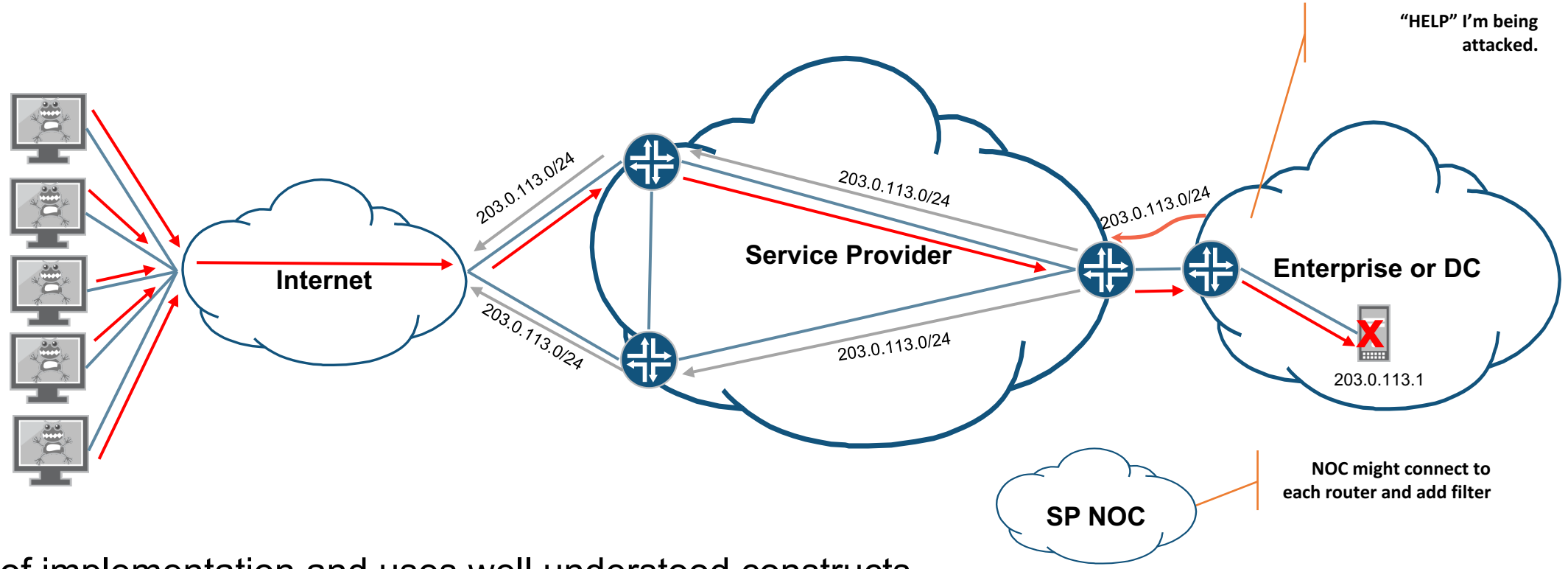






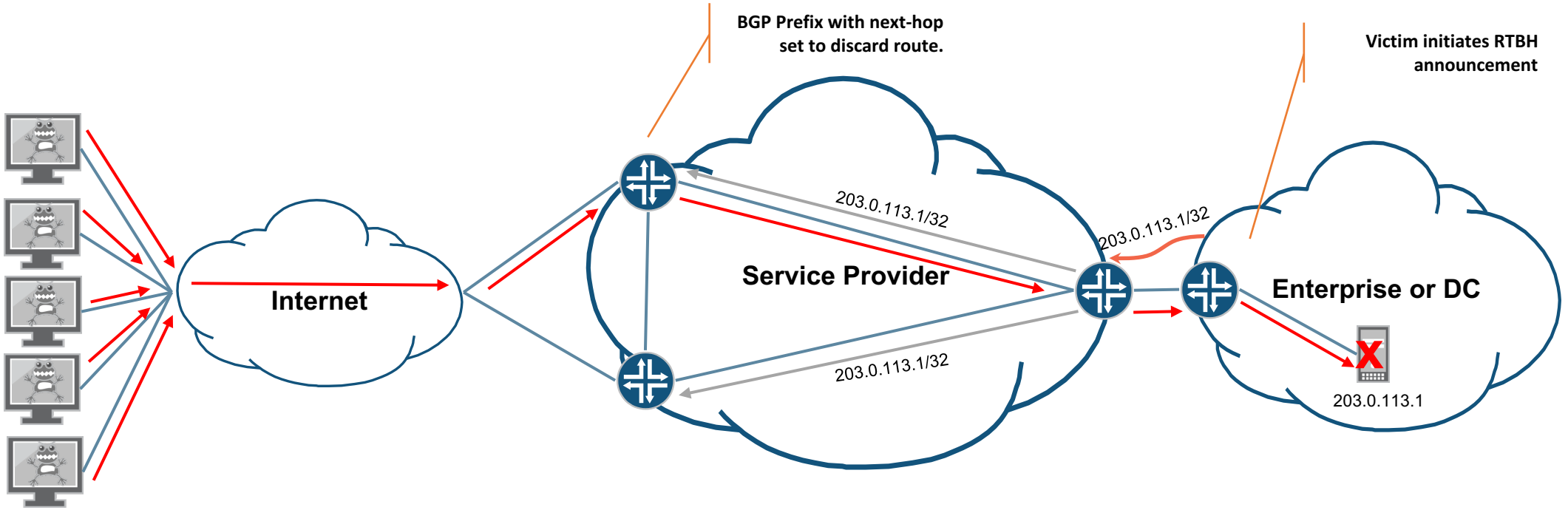
Slide Contribution from Justin Ryburn
Senior System Engineer
jryburn@juniper.net

Blocking DDoS in the “Old” Days



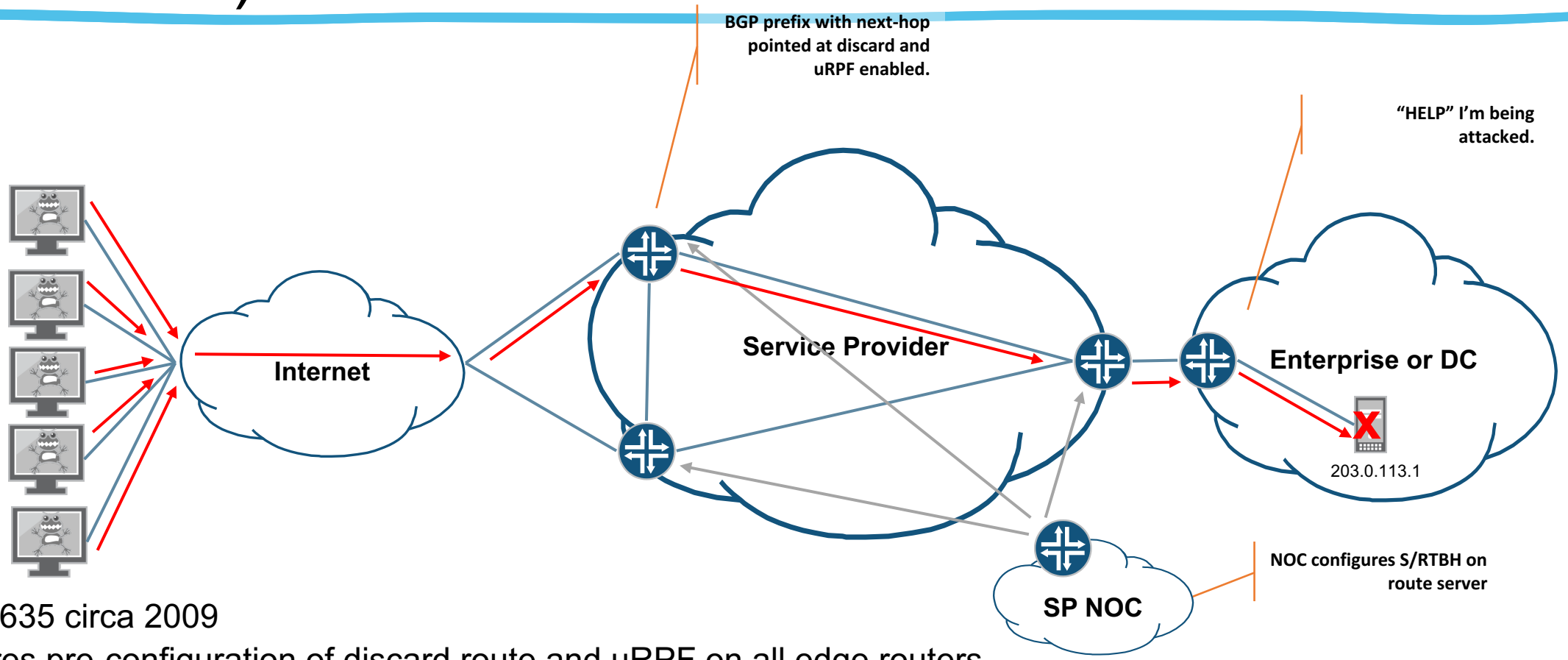
- Ease of implementation and uses well understood constructs
- Requires high degree of co-ordination between customer and provider
- Cumbersome to scale in a large network perimeter
- Mis-configuration possible and expensive

Destination Remotely Triggered Black Hole (D/RTBH)



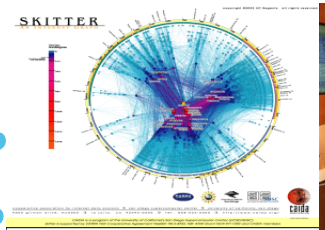
- RFC 3882 circa 2004
- Requires pre-configuration of discard route on all edge routers
- Victim's destination address is completely unreachable but attack (and collateral damage) is stopped.

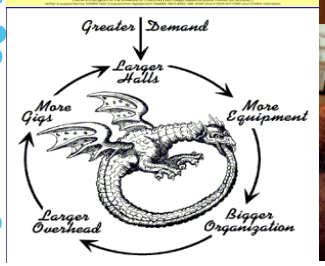
Source Remotely Triggered Black Hole (S/RTBH)

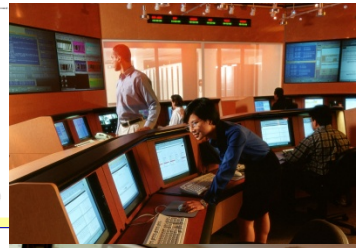


- RFC 5635 circa 2009
- Requires pre-configuration of discard route and uRPF on all edge routers
- Victim's destination address is still useable
- Only works for single (or small number) source.

BGP FlowSpec Overview







BGP Flow Specification

- Specific information about a flow can now be distributed using a BGP NLRI defined in RFC 5575 [5] circa 2009
  - AFI/SAFI = 1/133: Unicast Traffic Filtering Applications
  - AFI/SAFI = 1/134: VPN Traffic Filtering Applications
- Flow routes are automatically validated against unicast routing information or via routing policy framework.
  - Must belong to the longest match unicast prefix.
- Once validated, firewall filter is created based on match and action criteria.

BGP Flow Specification

- BGP Flowspec can include the following information:
  - Type 1 - Destination Prefix
  - Type 2 - Source Prefix
  - Type 3 - IP Protocol
  - Type 4 – Source or Destination Port
  - Type 5 – Destination Port
  - Type 6 - Source Port
  - Type 7 – ICMP Type
  - Type 8 – ICMP Code
  - Type 9 - TCP flags
  - Type 10 - Packet length
  - Type 11 – DSCP
  - Type 12 - Fragment Encoding

BGP Flow Specification

- Actions are defined using BGP Extended Communities:
  - 0x8006 – traffic-rate (set to 0 to drop all traffic)
  - 0x8007 – traffic-action (sampling)
  - 0x8008 – redirect to VRF (route target)
  - 0x8009 – traffic-marking (DSCP value)

Vendor Support

- DDoS Detection Vendors:
  - Arbor Peakflow SP 3.5
  - Accumuli DDoS Secure
- Router Vendors:
  - Alcatel-Lucent SR OS 9.0R1
  - Juniper JUNOS 7.3
  - Cisco 5.2.0 for ASR and CRS [6]
- OpenSource BGP Software:
  - ExaBGP

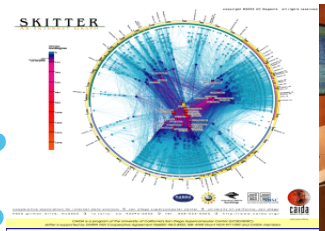
What Makes BGP Flowspec Better?

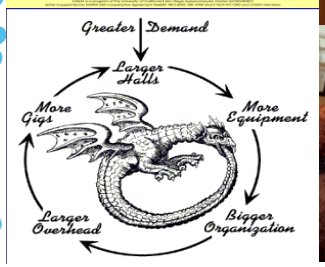
- Same granularity as ACLs
  - Based on n-tuple matching
- Same automation as RTBH
  - Much easier to propagate filters to all edge routers in large networks
- Leverages BGP best practices and policy controls
  - Same filtering and best practices used for RTBH can be applied to BGP Flowspec

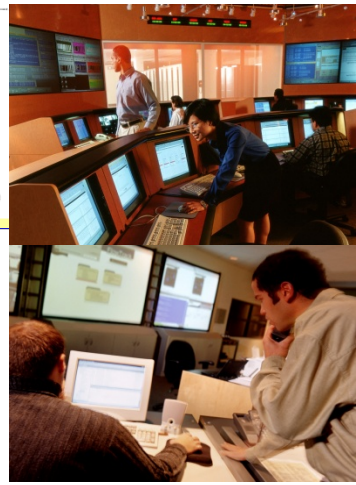
Caveats

- Forwarding Plane resources
  - Creating dynamic firewall filters that use these resources
  - More complex FS routes/filters will use more resources
  - Need to test your vendors limits and what happens when it is hit
  - Usually ways to limit the number and complexity of filters to avoid issues
- Not a replacement technology
  - Should be ADDED to existing mitigation methods and not replace them
- When it goes wrong (bugs) it goes wrong fast
  - Cloudflare outage: https://blog.cloudflare.com/todays-outage-post-mortem-82515/

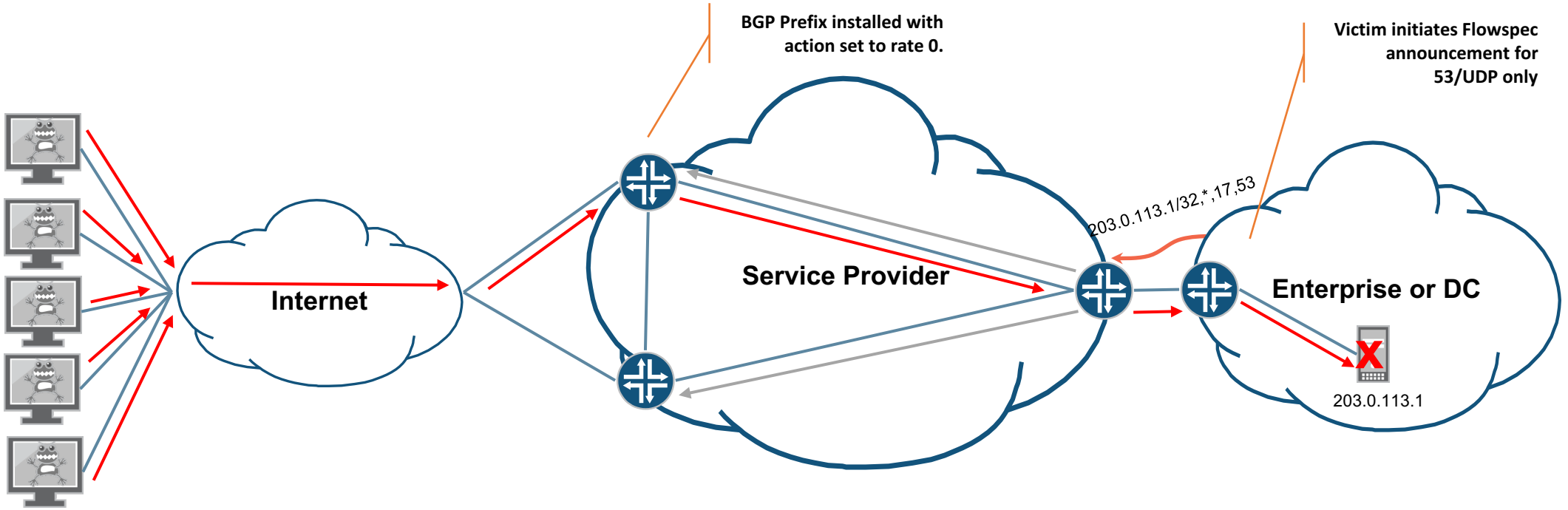
Use Case Examples







Inter-domain DDoS Mitigation Using Flowspec



- Allows ISP customer to initiate the filter.
- Requires sane filtering at customer edge.

Edge Router Configuration

Alcatel-Lucent

```
router
    autonomous-system 64496
    bgp
        group "CUST-FLOWSPEC"
            neighbor 192.0.2.1
                family ipv4 flow-ipv4
                peer-as 64511
                no flowspec-validate
            exit
        exit
        no shutdown
    exit
Exit
```

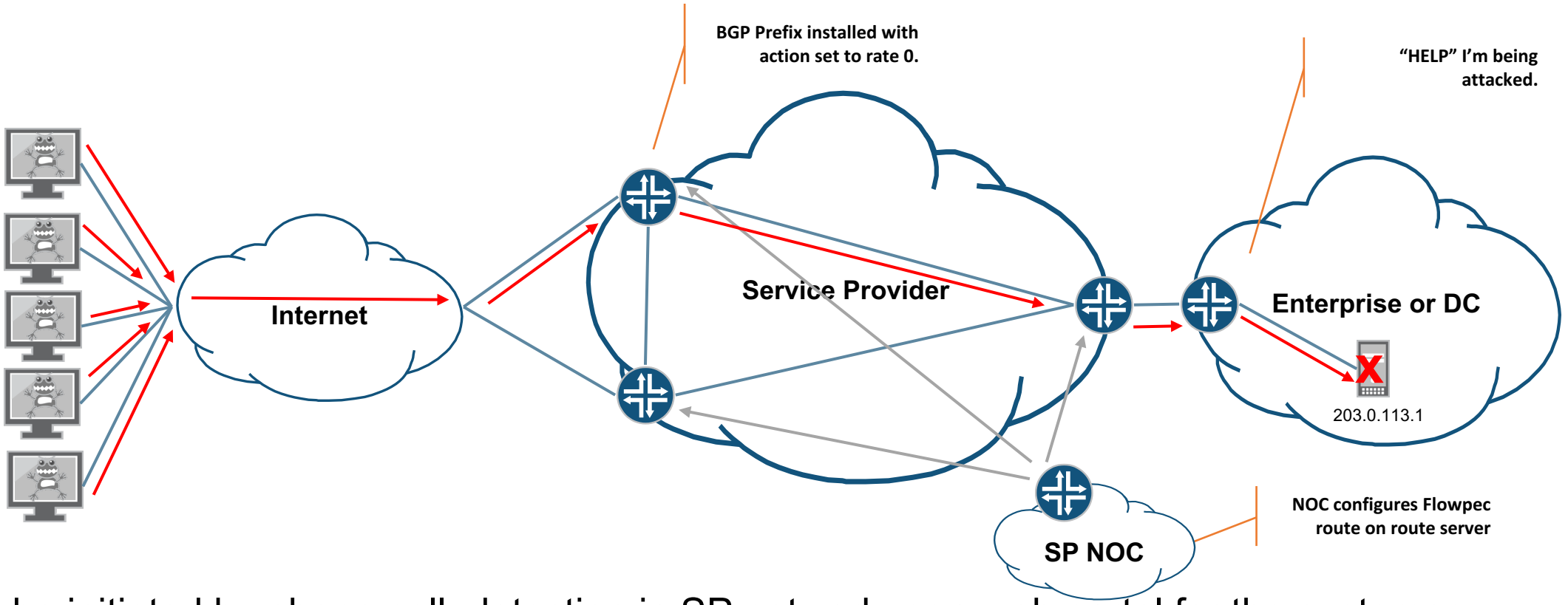
Cisco [7]

```
router bgp 64496
  ! Initializes the global address family
  address-family ipv4 flowspec
!
  neighbor 192.0.2.1
    remote-as 64511
    ! Ties it to a neighbor configuration
    address-family ipv4 flowspec
```

Juniper

```
protocols {
    bgp {
        group CUST-FLOWSPEC {
            peer-as 64511;
            neighbor 192.0.2.1 {
                family inet {
                    flow;
                }
            }
        }
    }
}
routing-options {
    flow {
        term-order standard;
    }
}
```

Intra-domain DDoS Mitigation Using Flowspec



- Could be initiated by phone call, detection in SP network, or a web portal for the customer.
- Requires co-ordination between customer and provider.

Edge Router Configuration

Alcatel-Lucent

```
router
    autonomous-system 64496
    bgp
        group "RR-CLIENT-FLOWSPEC"
            neighbor 198.51.100.1
                family ipv4 flow-ipv4
                peer-as 64496
            exit
        exit
        no shutdown
    exit
exit
```

Cisco [7]

```
router bgp 64496
  ! Initializes the global address family
  address-family ipv4 flowspec
!
  neighbor 198.51.100.1
    remote-as 64496
    ! Ties it to a neighbor configuration
    address-family ipv4 flowspec
```

Juniper

```
protocols {
    bgp {
        group RR-CLIENT-FLOWSPEC {
            type internal;
            neighbor 198.51.100.1 {
                family inet {
                    flow;
                }
            }
        }
    }
}
routing-options {
    flow {
        term-order standard;
    }
}
```

Route Server Configuration

Alcatel-Lucent

```
router
    autonomous-system 64496
    bgp
        group "RR-CLIENT-FLOWSPEC"
            neighbor 198.51.100.2
                family ipv4 flow-ipv4
                peer-as 64496
            exit
        exit
        no shutdown
    exit
exit
```

Cisco [7]

```
router bgp 64496
  ! Initializes the global address family
  address-family ipv4 flowspec
!
  neighbor 198.51.100.2
    remote-as 64496
    ! Ties it to a neighbor configuration
    address-family ipv4 flowspec
```

Juniper

```
protocols {
    bgp {
        group RR-CLIENT-FLOWSPEC {
            type internal;
            neighbor 198.51.100.2 {
                family inet {
                    flow;
                }
                export FLOWROUTES_OUT;
            }
        }
    }
}
```

Route Server Configuration

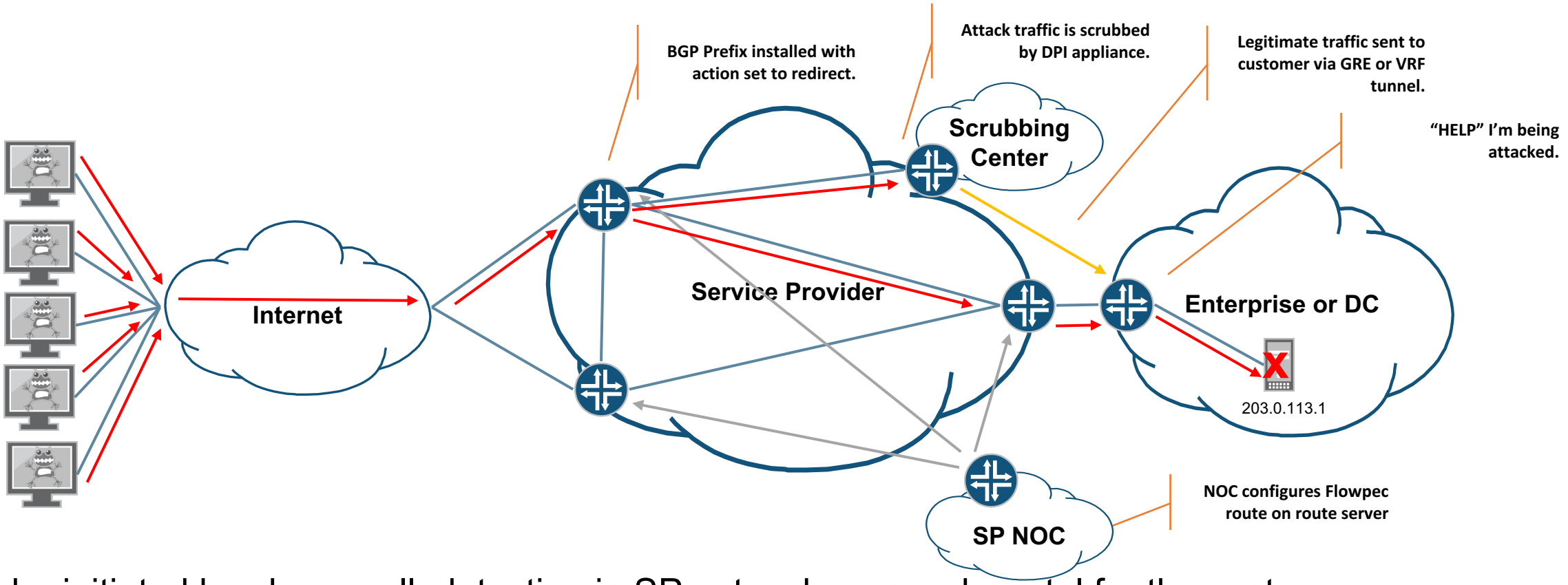
Cisco [7]

```
class-map type traffic match-all attack_fs
 match destination-address ipv4 203.0.113.1/32
 match protocol 17
 match destination-port 53
end-class-map
!
policy-map type pbr attack_pbr
 class type traffic attack_fs
  drop
 class class-default
end-policy-map
!
flowspec
 address-family ipv4
  service-policy type pbr attack_pbr
exit
```

Juniper

```
routing-options {
    flow {
        term-order standard;
        route attack_fs {
            match {
                destination 203.0.113.1/32
                protocol udp;
                destination-port 53;
            }
            then discard;
        }
    }
}
policy-options {
    policy-statement FLOWROUTES_OUT {
        from {
            rib inetflow.0;
        }
        then accept;
    }
}
```

DDoS Mitigation Using Scrubbing Center



- Could be initiated by phone call, detection in SP network, or a web portal for the customer.
- Allows for mitigating application layer attacks without completing the attack.

Edge Router Configuration

Alcatel-Lucent

```
router
    autonomous-system 64496
    bgp
        group "RR-CLIENT-FLOWSPEC"
            neighbor 198.51.100.1
                family ipv4 flow-ipv4
                peer-as 64496
            exit
        exit
        no shutdown
    exit
exit
```

Cisco [7]

```
router bgp 64496
  ! Initializes the global address family
  address-family ipv4 flowspec
!
  neighbor 198.51.100.1
    remote-as 64496
    ! Ties it to a neighbor configuration
    address-family ipv4 flowspec
```

Juniper

```
protocols {
    bgp {
        group RR-CLIENT-FLOWSPEC {
            type internal;
            neighbor 198.51.100.1 {
                family inet {
                    flow;
                }
            }
        }
    }
}
routing-options {
    flow {
        term-order standard;
    }
}
```

Route Server Configuration

Alcatel-Lucent

```
router
    autonomous-system 64496
    bgp
        group "RR-CLIENT-FLOWSPEC"
            neighbor 198.51.100.2
                family ipv4 flow-ipv4
                peer-as 64496
            exit
        exit
        no shutdown
    exit
exit
```

Cisco [7]

```
router bgp 64496
  ! Initializes the global address family
  address-family ipv4 flowspec
!
  neighbor 198.51.100.2
    remote-as 64496
    ! Ties it to a neighbor configuration
    address-family ipv4 flowspec
```

Juniper

```
protocols {
    bgp {
        group RR-CLIENT-FLOWSPEC {
            type internal;
            neighbor 198.51.100.2 {
                family inet {
                    flow;
                }
                export FLOWROUTES_OUT;
            }
        }
    }
}
```

Route Server Configuration

Cisco [7]

```
class-map type traffic match-all attack_fs
 match destination-address ipv4 203.0.113.1/32
 match protocol 17
 match destination-port 53
end-class-map
!
policy-map type pbr attack_pbr
 class type traffic attack_fs
  redirect nexthop 192.0.2.7
 class class-default
end-policy-map
!
flowspec
 address-family ipv4
  service-policy type pbr attack_pbr
exit
```

Juniper

```
routing-options {
    flow {
        term-order standard;
        route attack_fs {
            match {
                destination 203.0.113.1/32
                protocol udp;
                destination-port 53;
            }
            then discard;
        }
    }
}
policy-options {
    policy-statement FLOWROUTES_OUT {
        from {
            rib inetflow.0;
        }
        then {
            next-hop 192.0.2.7;
            accept;
        }
    }
}
```

How Do I Know It Is Working?

Alcatel-Lucent	Cisco [7]	Juniper
• show router bgp routes flow-ipv4 • show router bgp routes flow-ipv6 • show filter ip fSpec-0 • show filter ip fSpec-0 associations • show filter ip fSpec-0 counters • show filter ip fSpec-0 entry <entry-id>	• show processes flowspec_mgr location all • show flowspec summary • show flowspec vrf all • show bgp ipv4 flowspec	• show bgp neighbor <neighbor> | match inet-flow • show route table inetflow.0 extensive • show firewall filter __flowspec_default_inet__

Real World Example

"Where I think FlowSpec excels, is for protection of our mobile platform.

2 /24s are shared among a million mobile devices with NAT in a firewall.

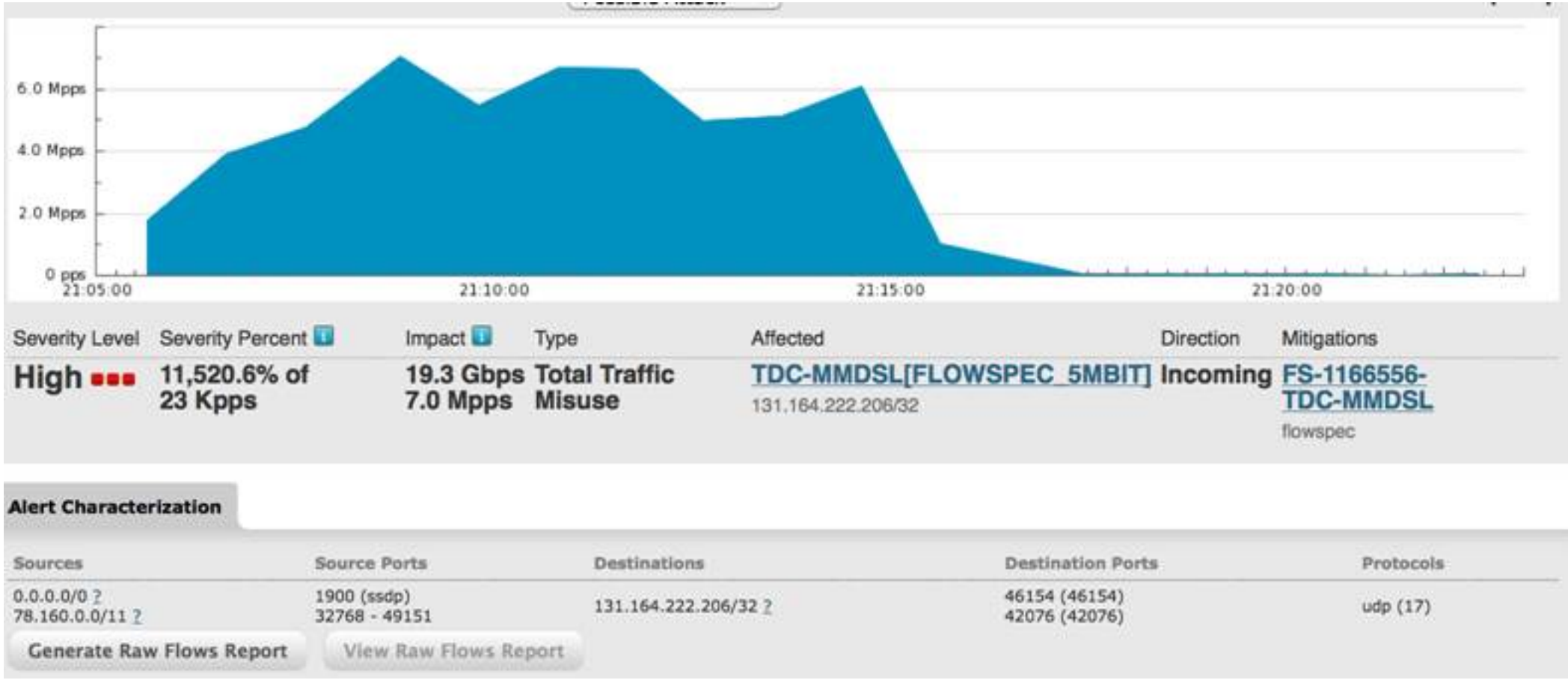
The link capacity (and in part the firewall itself) is overloaded by a simple DDoS attack against just one of these adresses.

The system detects a DoS attack against an address on the firewall.
It will identify total traffic, UDP, fragments, TCP SYN, ICMP, whatever, and depending on what kind of attack it is, a policer is added for the specific protocol/attack on individual peering routers. Protocols are policed with individual policers, so that for instance UDP and TCP SYN can be policed to different throughputs.

Basically, an attack against a single IP on UDP will not affect other customers being NAT'ed to the same address, using anything but UDP - and link capacity is protected."

Real World Example

- Attack on 1/13/16



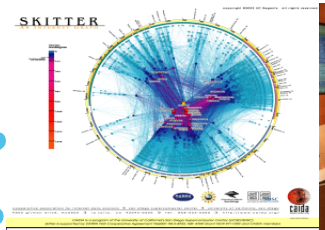
Where Are We Going?

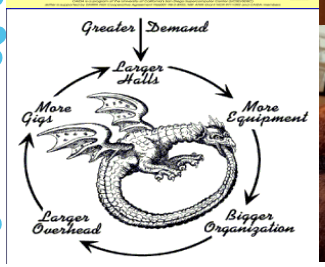
- IPv6 Support
  - http://tools.ietf.org/html/draft-ietf-idr-flow-spec-v6-06
- Relaxing Validation
  - http://tools.ietf.org/html/draft-ietf-idr-bgp-flowspec-oid-02
- Redirect to IP Action
  - https://tools.ietf.org/html/draft-ietf-idr-flowspec-redirect-ip-02

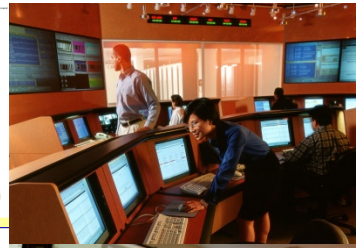
References

- [1] Kaspersky Lab – Every Third Public Facing Company Encounters DDoS Attacks **http://tinyurl.com/neu4zzr**
- [2] Verisign – 2014 DDoS Attack Trends **http://tinyurl.com/oujgx94**
- [3] NBC News – Internet Speeds are Rising Sharply, But So Are Hack Attacks **http://tinyurl.com/q4u2b7m**
- [4] Tech Times – DDoS Attack Cripples Sony PSN While Microsoft Deals with Xbox Live Woes **http://tinyurl.com/kkdczjx**
- [5] RFC 5575 - Dissemination of Flow Specification Rules **http://www.ietf.org/rfc/rfc5575.txt**
- [6] Cisco - Implementing BGP Flowspec **http://tinyurl.com/mm5w7mo**
- [7] Cisco – Understanding BGP Flowspec **http://tinyurl.com/l4kwb3b**

RTBH used on IXPs – Equinix Example



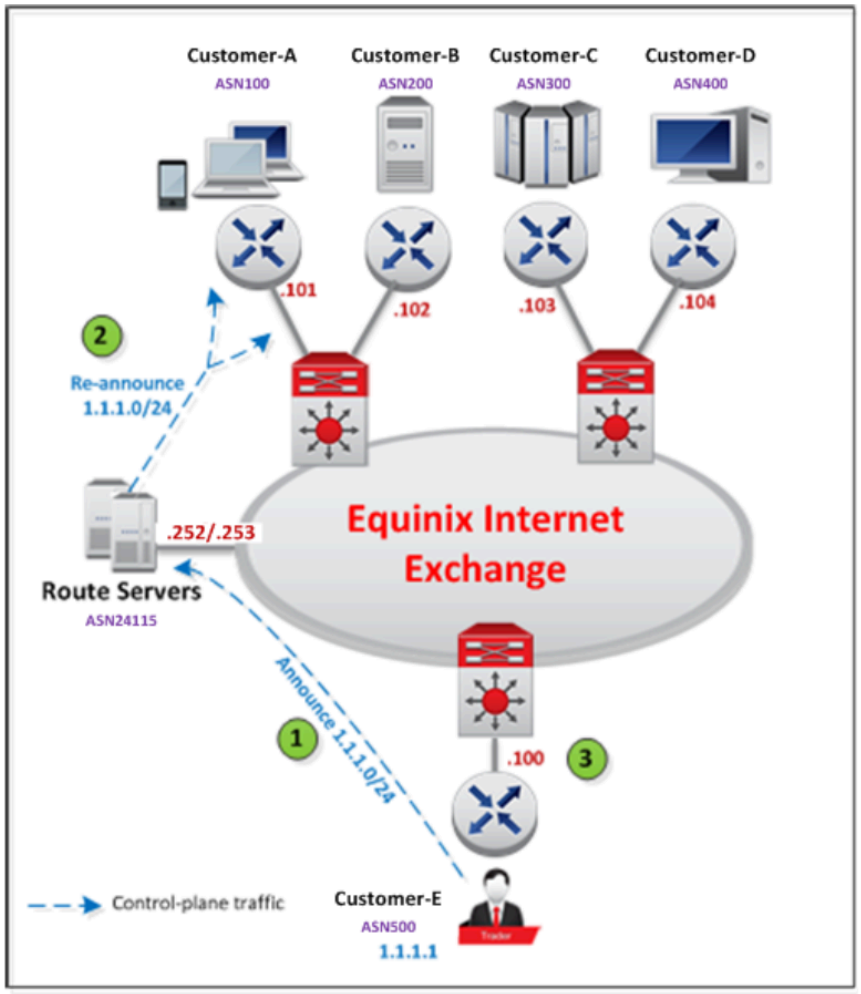




RTBH as Collective Action – Equinix Example

- Equinix has deployed RTBH capabilities on all their peering exchanges.
- “Remote Triggered Black Hole: Equinix Internet Exchange Platform” Presented at SANOG 28 in 2016

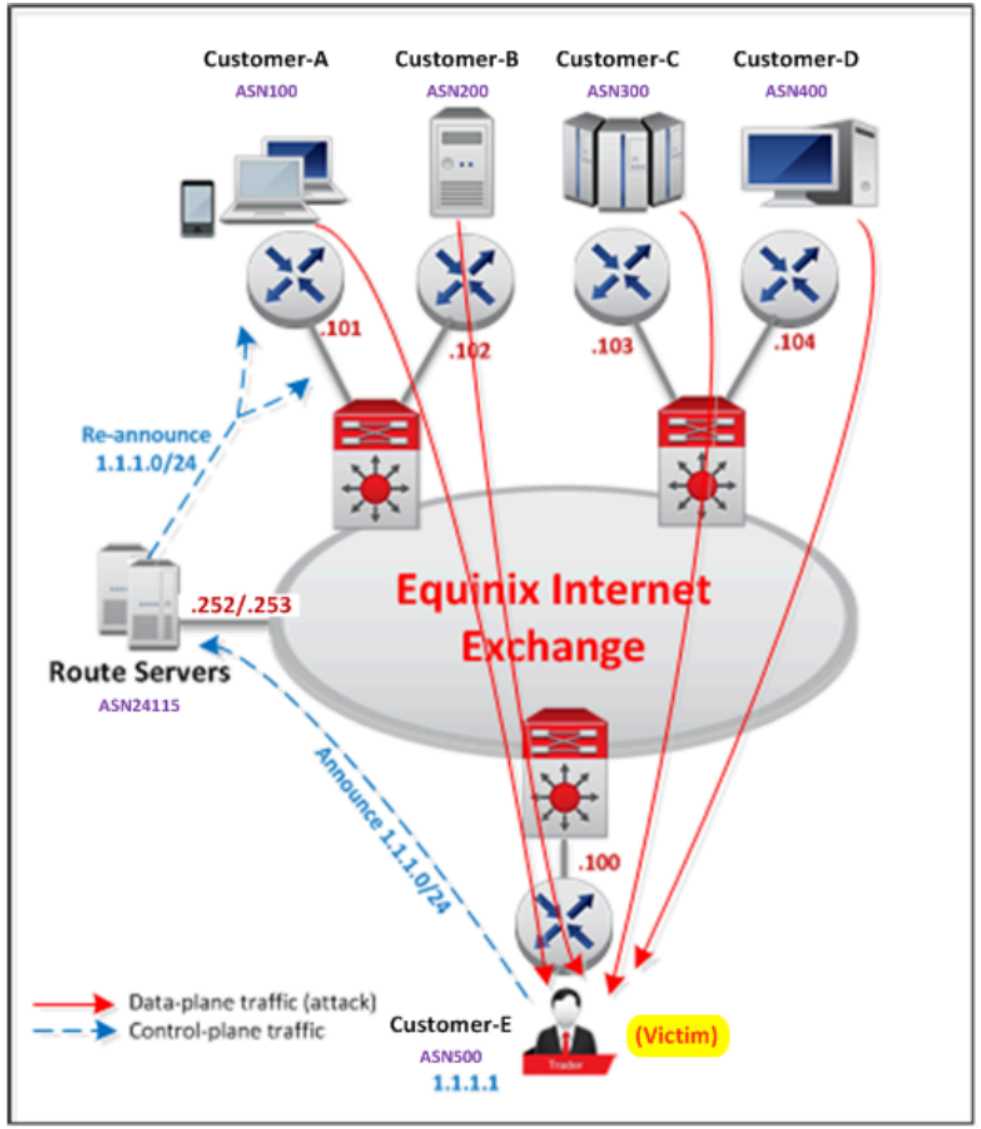
How it works …



Initially:

1. All customers established BGP peering to route servers via MLPE IX peering subnet. The victim customer announced their prefix **1.1.1.0/24** to route servers.
2. Route servers re-announce it back to other peering participants.
3. The next-hop to reach 1.1.1.0/24 prefix is **.100** which is the peering IP address of victim customer.

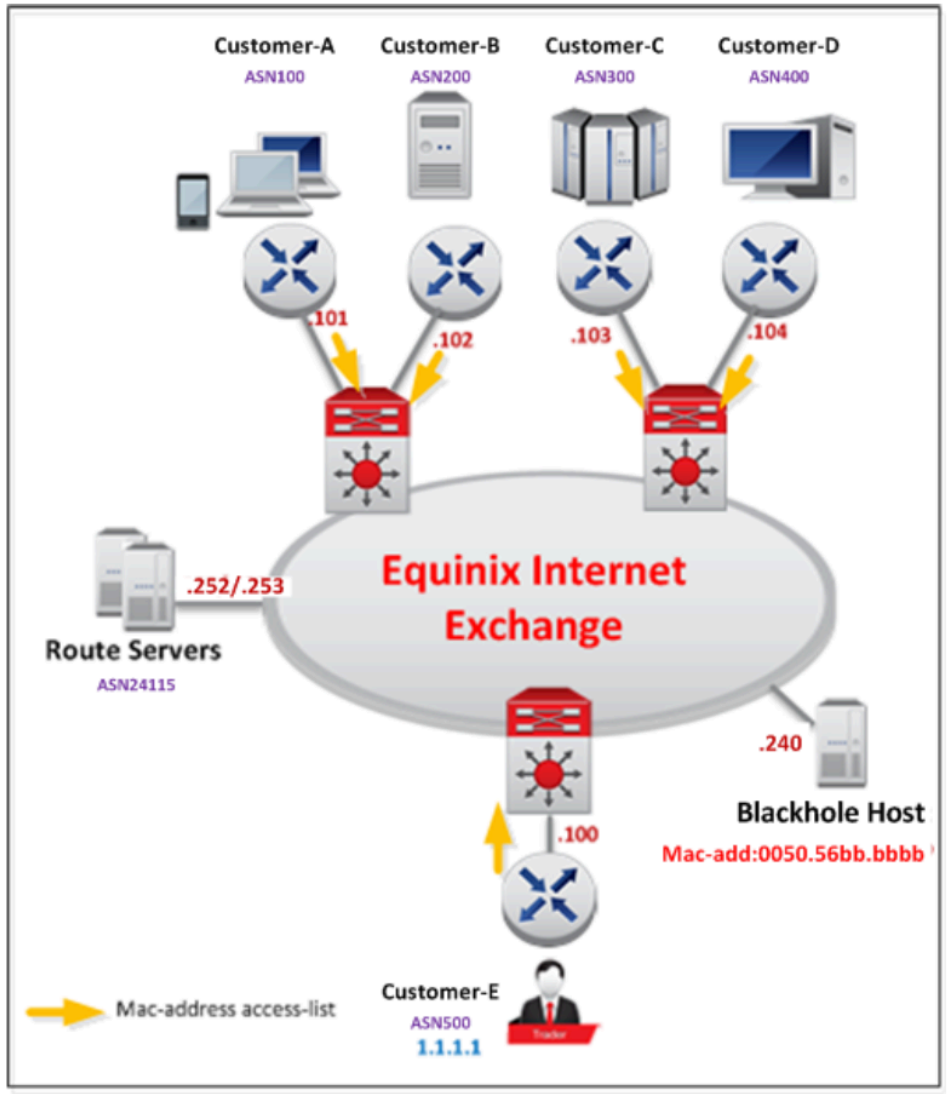
How it works: Under DOS Attack



Under DDoS Attack:

- There is an DDoS attack traffic towards 1.1.1.1 which is one of the web servers.
- The victim's EIE port has been flooded and inbound (towards customer) utilization hits to 100%.
- This flooding caused service disruption to all production services.
- The victim needs to free up the port utilization by stopping traffic to 1.1.1.1.
  - » by RTBH.

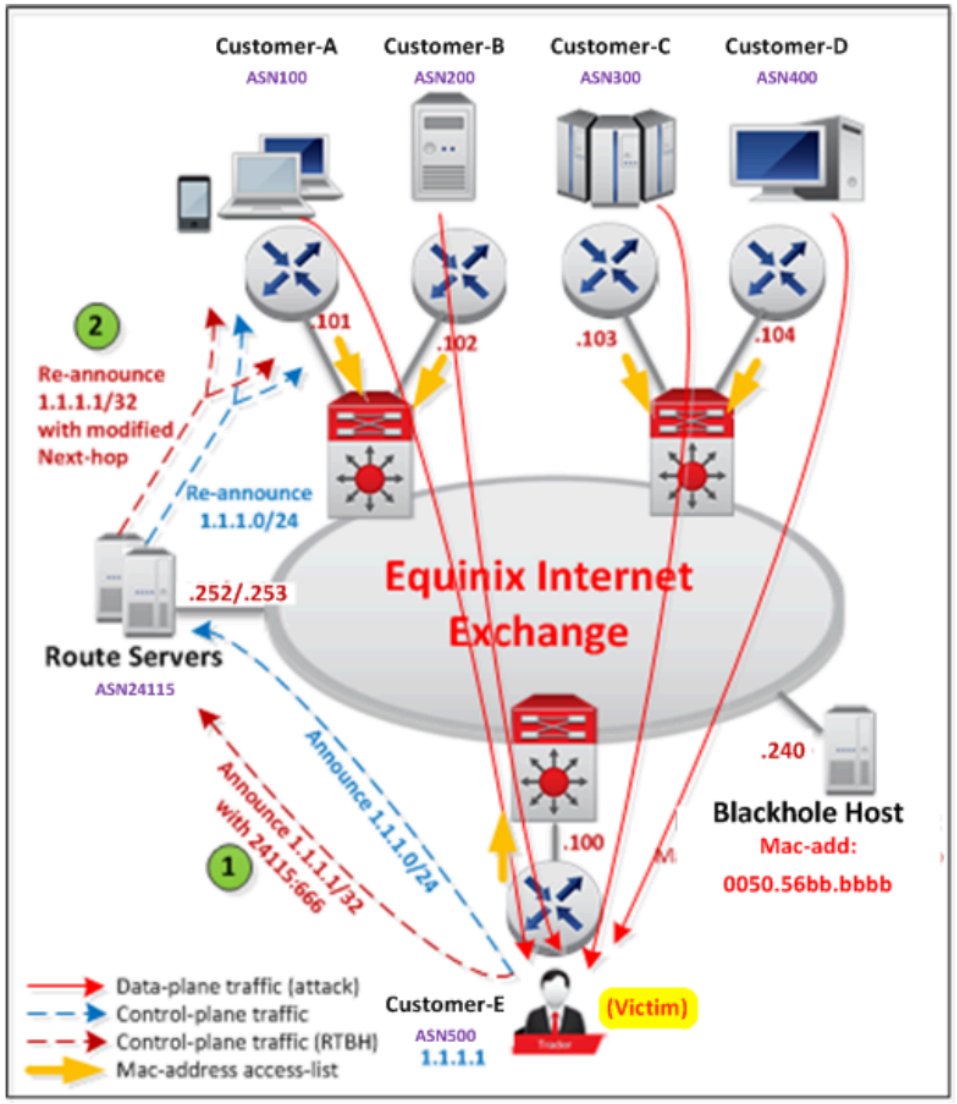
How it works: RTBH Setup



RTBH Setup:

- Equinix provides **Blackhole Host** with IP address **.240** whose mac address is **0050.56bb.bbbb.**
- All unicast traffic towards the Blackhole Host will be denied at customer facing ports (by mac-address ACL).

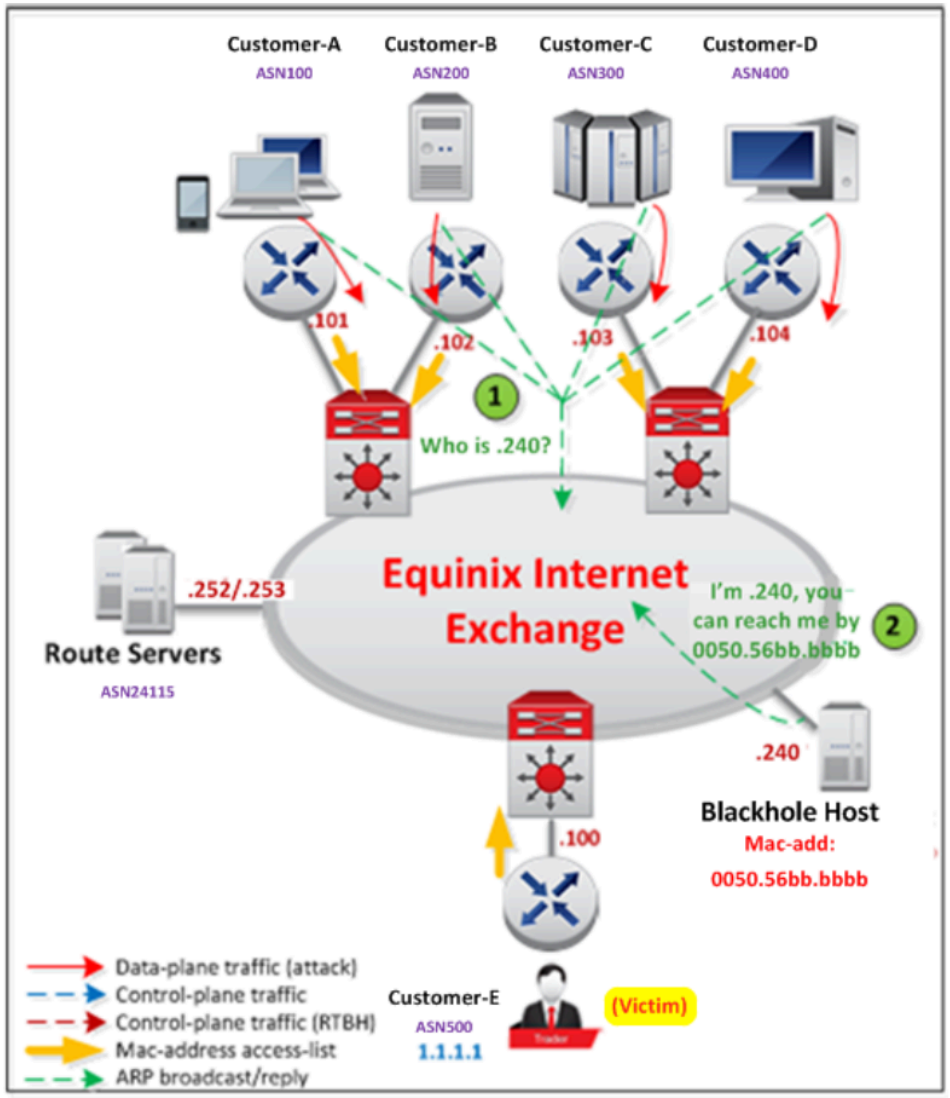
How it works: Mitigation



Mitigation:

1. The victim starts announce 1.1.1.1/32 with Blackhole BGP community **65535:666**.
2. Route servers will modified these prefixes (tagged with 65535:666) with it's next-hop to **.240** and re-announced back to other peering participants.

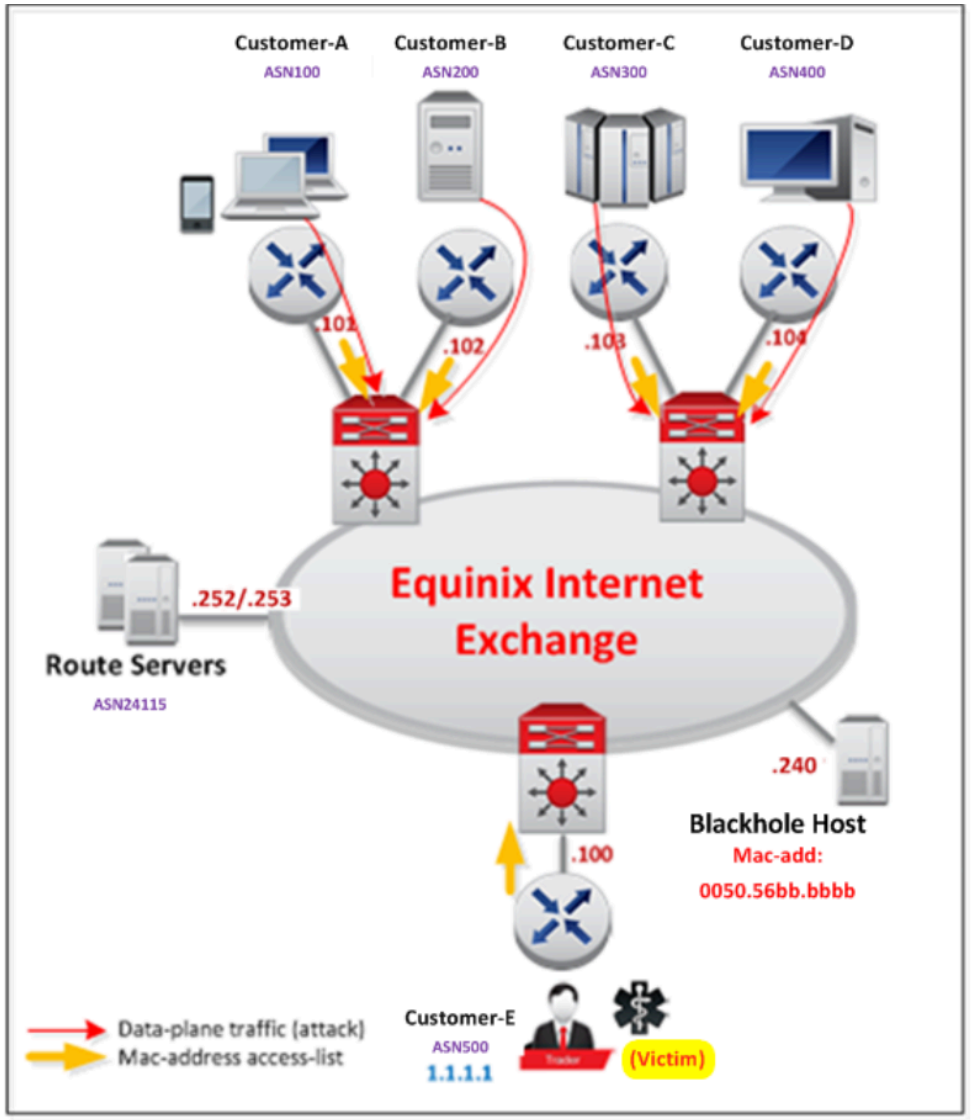
How it works: Mitigation (cont)



Mitigation (continued..):

1. Peering partners start to resolve next-hop IP address **.240** in order to reach 1.1.1.1.
2. Blackhole Host replied ARP with mac-address **0050.56bb.bbbb.**

How it works: Mitigation Complete



Mitigation (completed):

- The attack traffic with next-hop **.240** has been stopped by EIE switch **inbound access-list**.
- The victim's switch port has been mitigated and no more congestion.

Verification

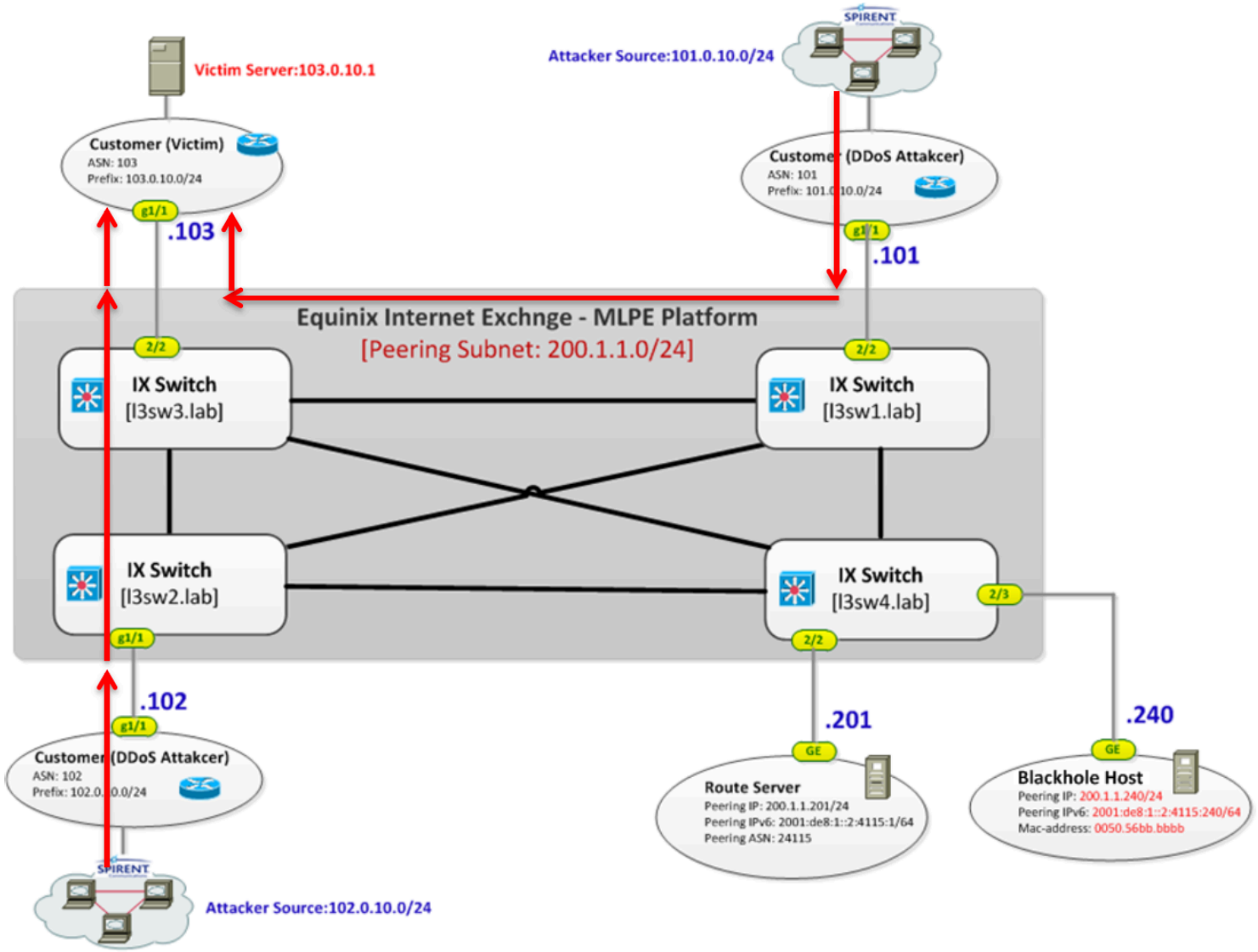
- **Verification**
  in the Lab

Initially, Customer (Victim) is announcing their prefix 103.10.0/24 to route server via MLPE.

Various Sources initiate DDoS attack to the destination IP, 103.0.10.1.

The attack traffic has been forwarded to the destination port

The victim port has been flooded with DDoS attack traffic.



Operations & Limitations

Operations & Limitation

- **RTBH Users must Announce its Own Prefixes**
  - If the customer wishes to announce RTBH 1.1.1.1/32
    - 1.1.1.1 prefix must be subset of it's own announced prefixes.

- **RTBH Prefix Announcement must be Accepted by Other Peering Participants**
  - In order RTBH to take it effect, the black hole announcement must be accepted by other peering partners.
  - Need to advise all peering participants to accept the prefixes with (prefix length > /24 and BGP community **65535:666**).
  - It's totally voluntary setup.

- **MLPA & BLPA**
  - In MLPA environment, EIE route servers manipulate Blackhole Host next-hop IP address.
  - In BLPA environment, the two peering partners need to engage for defining BGP community and next-hop IP address.
  - BLPA partners still can use Blackhole Host next-hop mac-address (0050.56bb.bbbb) in their implementation.

- **Support for IPv6**
  - It supports IPv6.
  - Route Server will change next-hop IPv6 address to Blackhole Host (example 2001::2:4115:240) if the customer prefixes are tagged with BGP community **65535:666**.
  - Blackhole Host will response to IPv6 ND.

Bi-Lateral Peering Notes

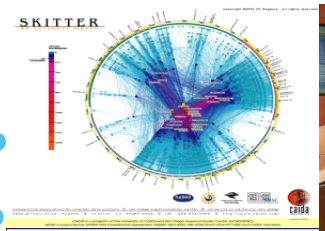
Bi-Lateral Peering Note

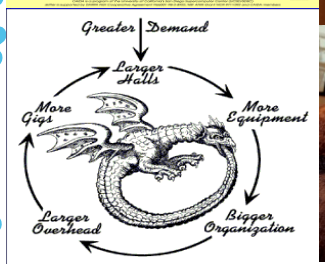
- Note
  - Some peering partners may not accept IX Well-known Blackhole Community (65535:666).
  - So BLPA peers must bilaterally agree on their own defined Blackhole Community + the router configuration to accept /32 announcement for that community.
  - Also customers are recommended to implement a route-map in which prefixes with blackhole community are changed its next-hop to local Null0 interface and re-advertise back to their peers.

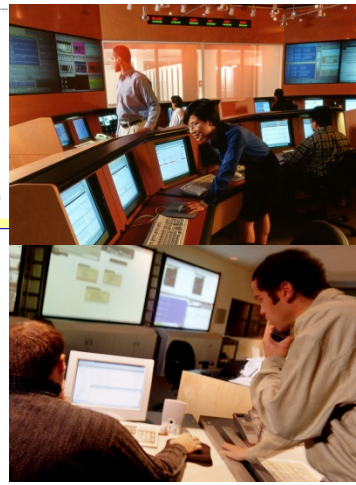
The Remote Triggered Black Hole (DDOS blocking) is based on the standard described in these evolving feature documents (use the last one of these, as it is most current):

- https://tools.ietf.org/id/draft-ymbk-grow-blackholing-00.txt
- https://tools.ietf.org/id/draft-ymbk-grow-blackholing-01.txt
- https://tools.ietf.org/id/draft-ietf-grow-blackholing-00.txt
- https://tools.ietf.org/id/draft-ietf-grow-blackholing-01.txt

RTBH Detailed Walk Through







Remotely Triggered Black Hole Filtering

- Remotely Triggered Black Hole (RTBH) filtering is an ISP network foundation tool for a series of reaction and traceback techniques of DoS/DDoS attacks
- Uses BGP to trigger network-wide attack flow responses
  - Simple static route allows ISP to trigger network-wide destination-based black holes at iBGP update speeds
  - Coupling with Unicast Reverse Path Forwarding (uRPF) Strict or Loose mode enables network-wide source-based triggered black holes, including at the ISP—ISP edge
  - Effective against spoofed and valid source addresses
- Preparation does not adversely impact ISP operations or performance

Reacting to an Attack with uRPF and RTBH

What Is Needed?

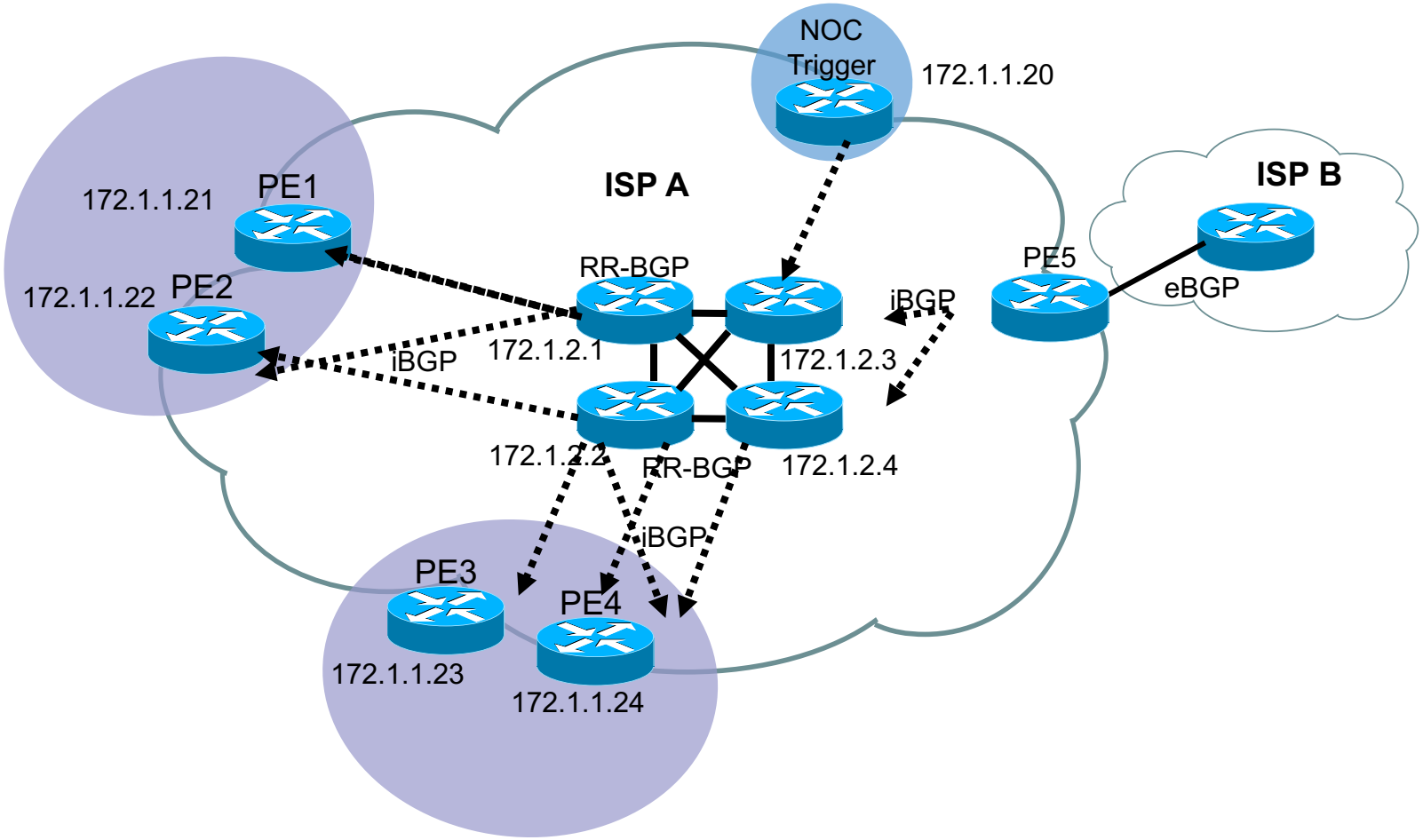
- Unicast Reverse Path Forwarding (uRPF) strict or loose mode on all border/edge routers
- Pre-configured static route to Null0 for a /32 address from TEST-NET (192.0.2.0/24) on all border/edge routers
- Method to inject a BGP advertisement into the network with a tag to trigger the drop
  - Should include the no-export, no-advertise communities, and good egress route-filters
- A way to quickly identify and classify the attack traffic (NetFlow/Arbor)

RTBH: Deployment Scenarios...

Drop Techniques…

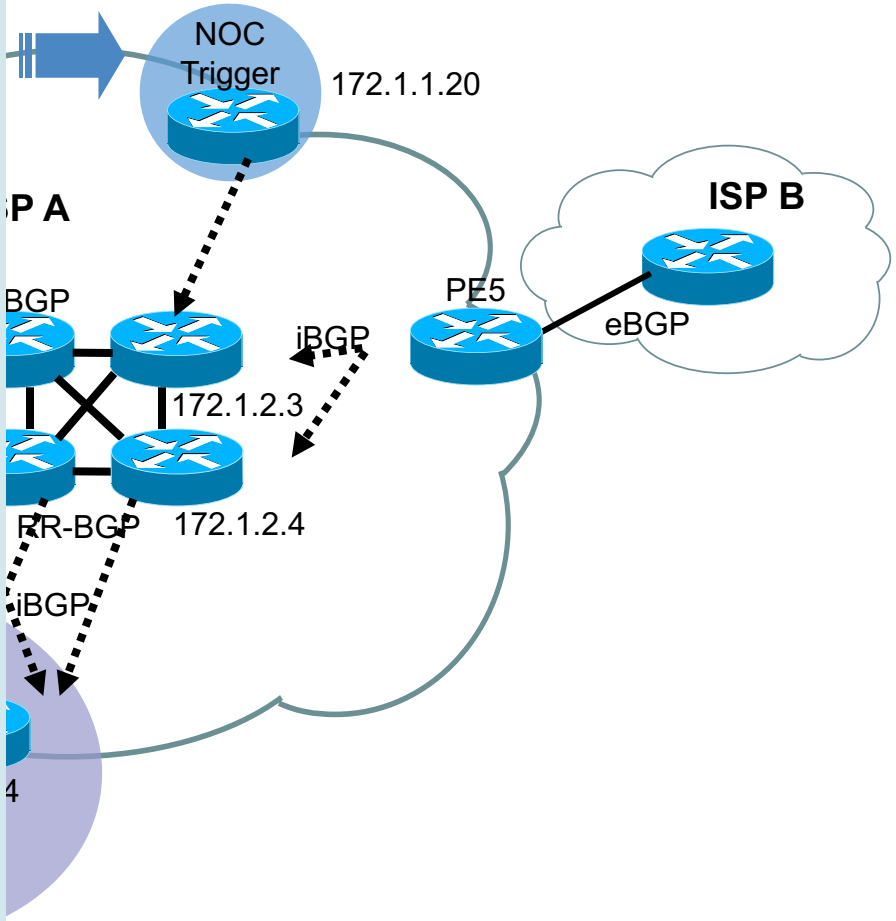
- Drops for different scenarios
  - Global Drops
  - Regional Drops
  - Different “tags” used to distinguish “type” of attack; e.g.,
    - Tag 25 for SPAM drops
    - Tag 666 for DoS drops
    - Tag 12345 for “customer initiated” drops
  - Drops only on certain routers (e.g., where ‘backscatter” can be collected)...
- Use tags for different outcomes…
  - Tags to “drop” on local router
  - Tags to “redirect” to sinkhole for analysis and then drop
  - Tags to “redirect” to services (cleaning) cluster

RTBH: Global and Regional Deployment Example: General Topology



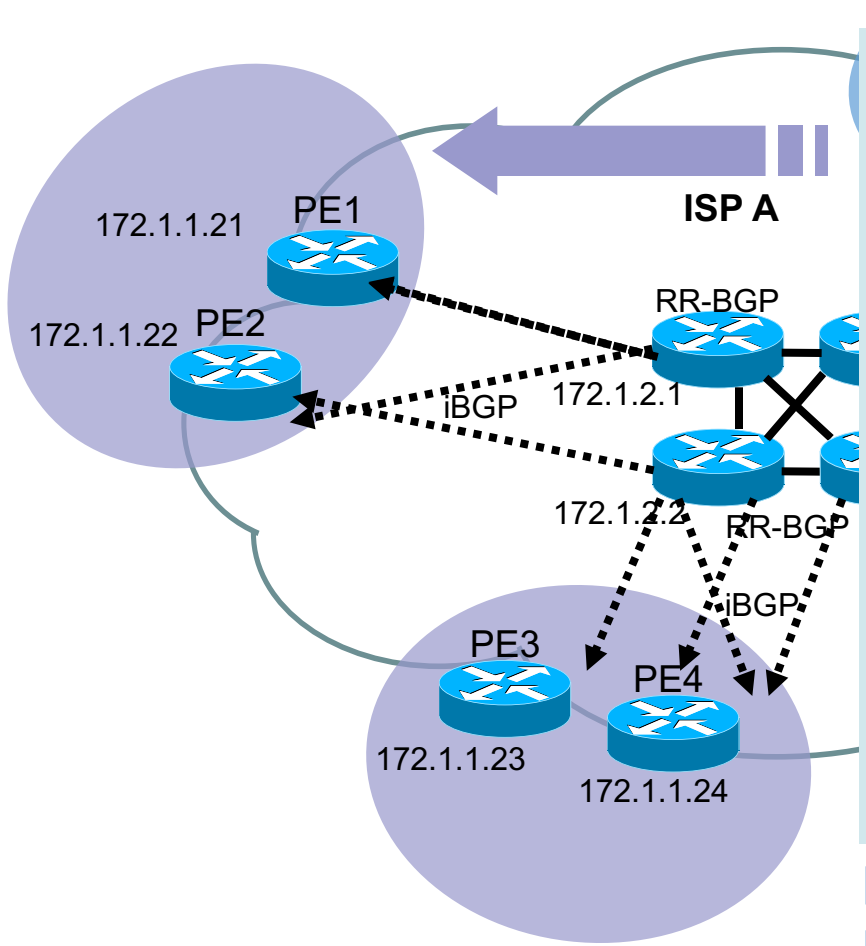
RTBH: Global and Regional Deployment Example: Basic Tag-Based Dst Drops

```
!
hostname trigger
ip bgp-community new-format
!
interface Loopback0
 ip address 172.1.1.20 255.255.255.255
!
router bgp 1
 no synchronization
 bgp router-id 172.1.1.20
 bgp log-neighbor-changes
 redistribute static route-map RTBH-trigger
 neighbor 172.1.2.3 remote-as 1
 neighbor 172.1.2.3 update-source Loopback0
 neighbor 172.1.2.3 version 4
 neighbor 172.1.2.3 send-community both
 neighbor 172.1.2.3 route-map no-prefix-in in
 no auto-summary
!
ip prefix-list no-in seq 5 deny 0.0.0.0/0
ip route 192.0.2.1 255.255.255.255 Null0
route-map no-prefix-in permit 10
 match ip address prefix-list no-in
!
route-map RTBH-trigger permit 10
 match tag 666
 set ip next-hop 192.0.2.1
 set local-preference 200
 set origin igp
 set community no-export
!
```



Trigger router pre-configuration

RTBH: Global and Regional Deployment Example: Basic Tag-Based Dst Drops



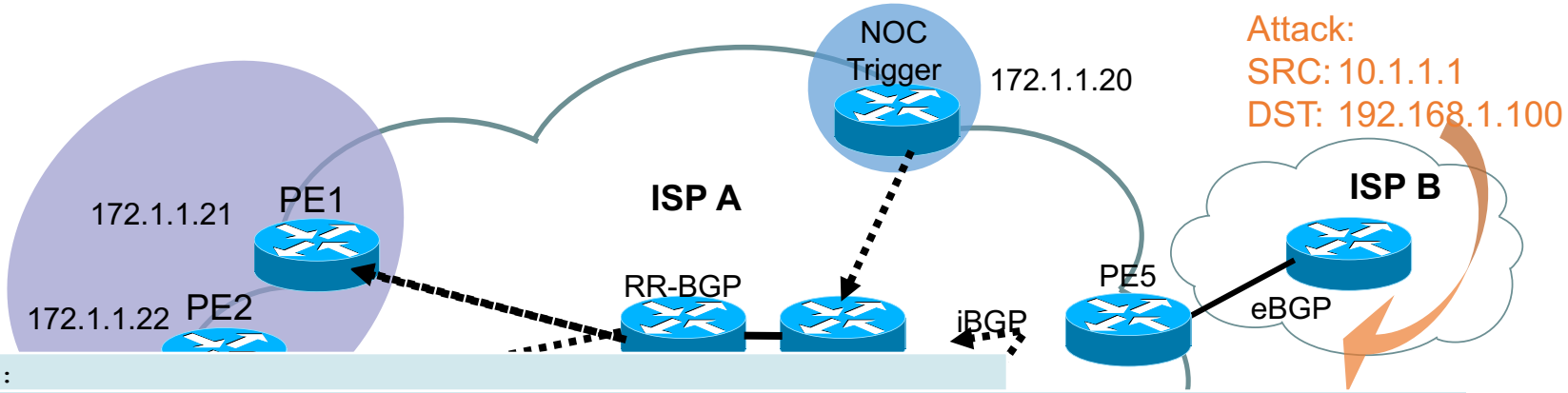

```
!
hostname PE1
ip bgp-community new-format
!
interface Loopback0
 ip address 172.1.1.21 255.255.255.255
!
interface Null0
 no ip unreachables
!
router bgp 1
 no synchronization
 bgp router-id 172.1.1.20
 bgp log-neighbor-changes
 neighbor RR-client peer-group
 neighbor RR-client remote-as 1
 neighbor RR-client update-source Loopback0
 neighbor RR-client version 4
 neighbor RR-client send-community both
 neighbor 172.1.2.1 peer-group RR-client
 neighbor 172.1.2.2 peer-group RR-client
 no auto-summary
!
ip route 192.0.2.1 255.255.255.255 Null0
!
```

PE1 Edge Router Pre-Configuration

(PE2, PE3, PE4, PE5 similar)

RTBH: Global and Regional Deployment Example: Basic Tag-Based Dst Drops

```
trigger(config)#ip route 192.168.1.100 255.255.255.255 null0 tag 666
```



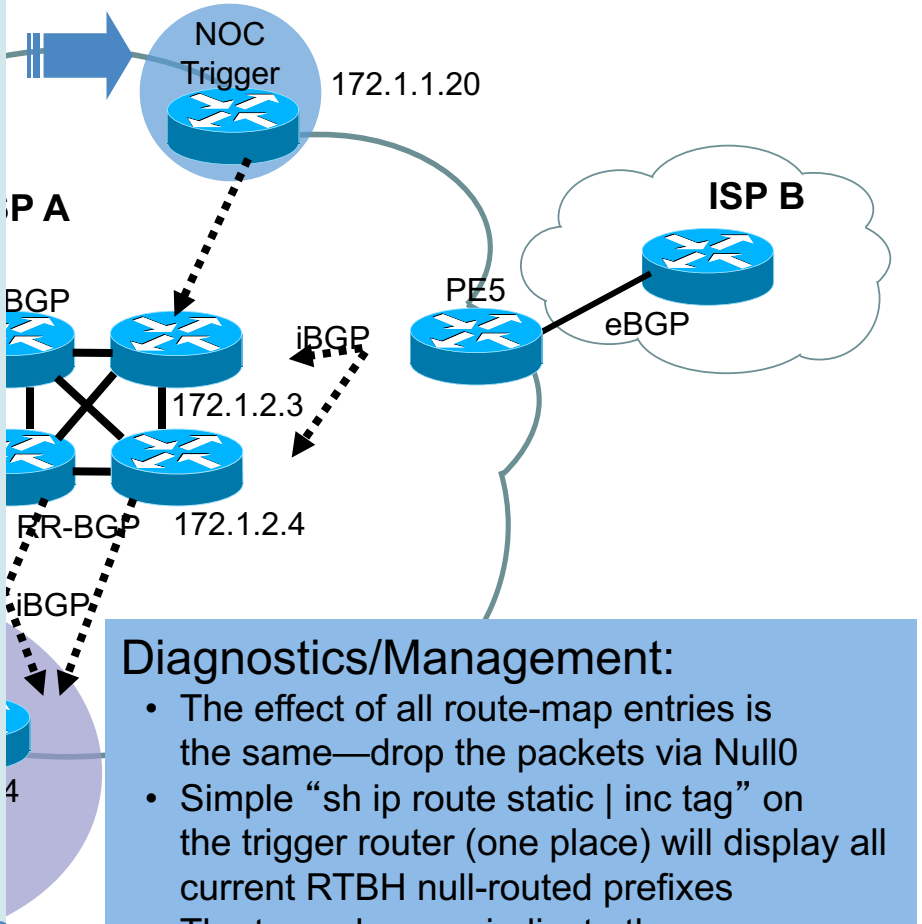

```
Before:
PE1#
192.
0 pa
  ta

  vi
```

```
After:
PE1#debug ip bgp update in
PE1#debug ip route
*Apr 29 08:38:15.825 UTC: RT: network 192.168.1.0 is now variably masked
*Apr 29 08:38:15.825 UTC: RT: add 192.168.1.100/32 via 192.0.2.1, bgp metric [200/0]
*Apr 29 08:38:16.921 UTC: RT: Nexthop for 192.168.1.100/32 updated
*Apr 29 08:23:21.613 UTC: BGP(0): 172.1.2.1 rcvd UPDATE w/ attr: nexthop x.x.x.x, origin i,
localpref 500, metric 0, originator x.x.x.x, community no-export
*Apr 29 08:23:21.613 UTC: BGP(0): 172.1.2.1 rcvd 192.168.1.100/32
*Apr 29 08:23:21.613 UTC: BGP(0): Revise route installing 1 of 1 route for 192.168.1.100/32 ->
192.0.2.1(main) to main IPtable

PE1#sh ip cef 192.168.1.100
192.168.1.100/32, version 18858, epoch 0
0 packets, 0 bytes
  tag information from 192.0.2.1/32, shared, all rewrites inherited
    local tag: implicit-null
  via 192.0.2.1, 0 dependencies, recursive
    next hop 192.0.2.1, Null0 via 192.0.2.1/32 (Default)
    valid null adjacency
```

RTBH: Global and Regional Deployment Example: Advanced Tag-Based Dst Drops

```
.
. <same>
.
ip route 192.0.2.1 255.255.255.255 Null0
.
route-map RTBH-trigger permit 10
 description - drops for SPAM
 match tag 25
 set ip next-hop 192.0.2.1
 set local-preference 200
 set origin igp
 set community no-export
!
route-map RTBH-trigger permit 20
 description - drops for ABUSE
 match tag 666
 set ip next-hop 192.0.2.1
 set local-preference 200
 set origin igp
 set community no-export
!
route-map RTBH-trigger permit 30
 description - drops for DOS
 match tag 12345
 set ip next-hop 192.0.2.1
 set local-preference 200
 set origin igp
 set community no-export
!
```

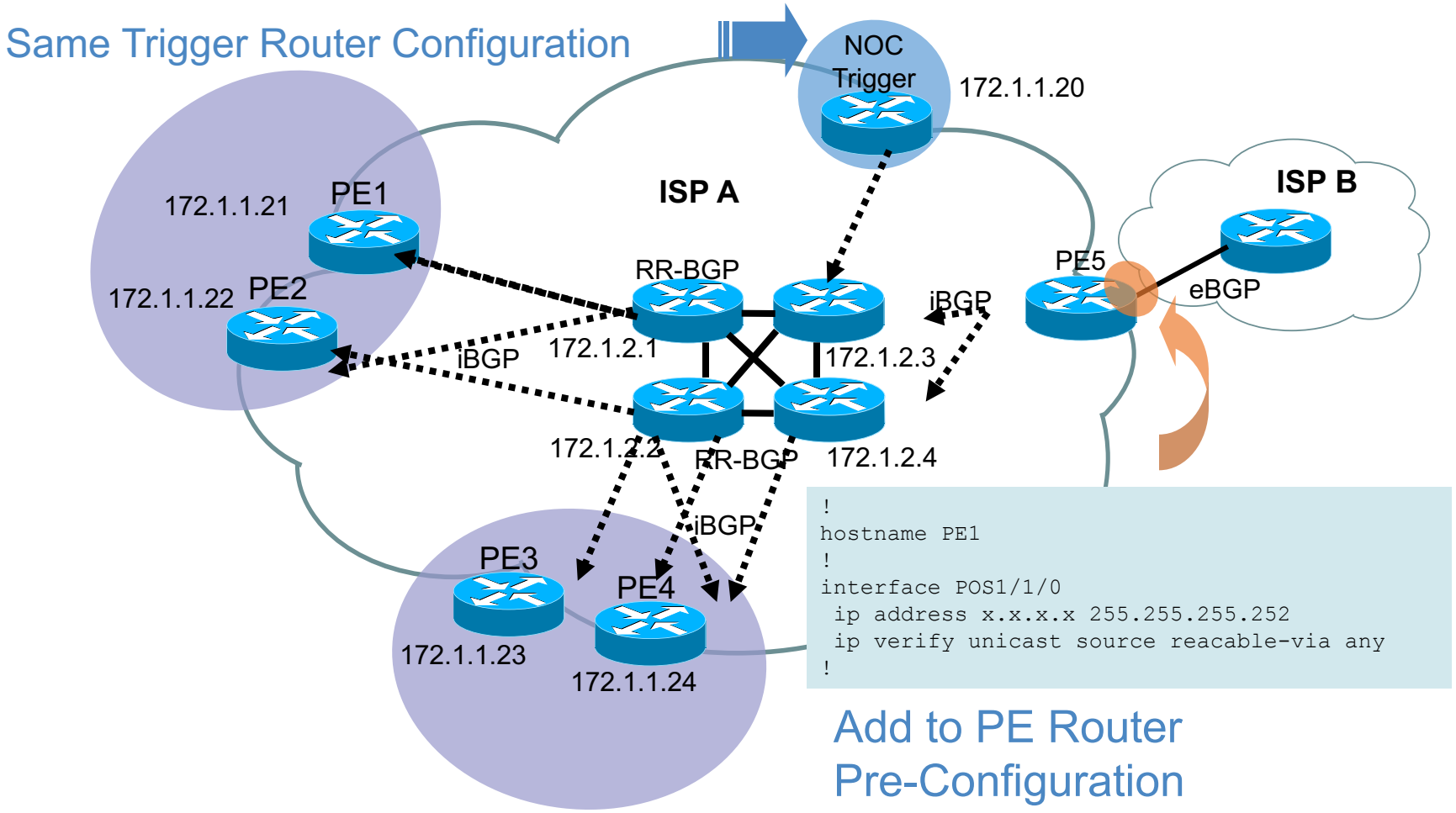
Trigger Router Pre-Configuration



Diagnostics/Management:

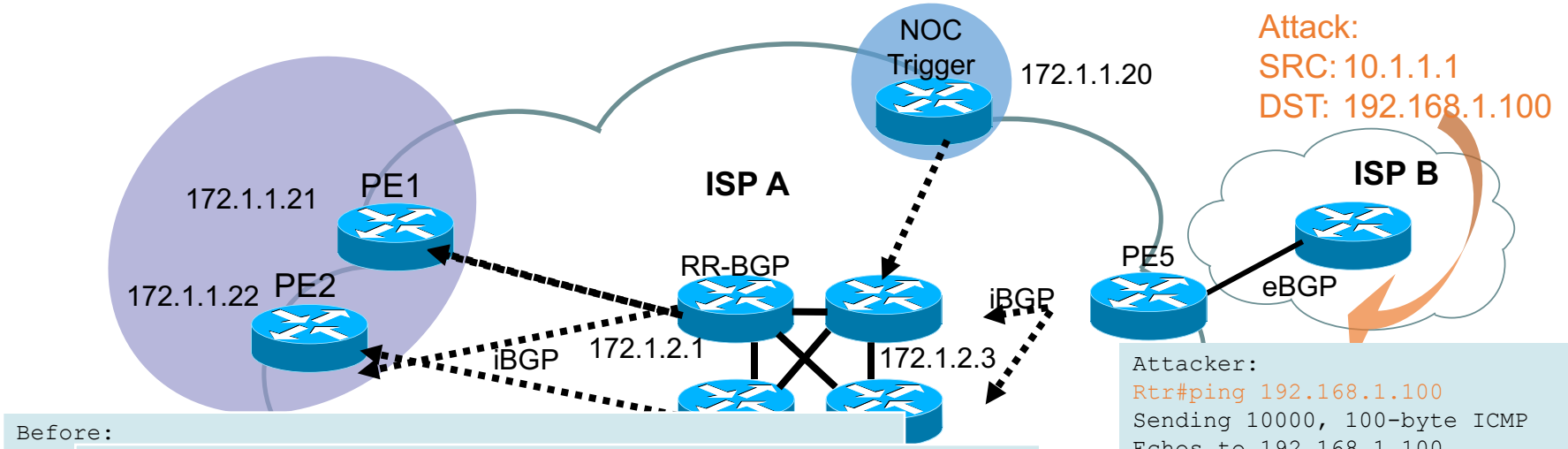
- The effect of all route-map entries is the same—drop the packets via Null0
- Simple “sh ip route static | inc tag” on the trigger router (one place) will display all current RTBH null-routed prefixes
- The tag value can indicate the reason why each prefix is null-routed

RTBH: Global and Regional Deployment Example: Basic Tag-Based Src Drops



RTBH: Global and Regional Deployment Example: Basic Tag-Based Src Drops

```
trigger(config)#ip route 10.1.1.1 255.255.255.255 null0 tag 666
```




```
Before:
PE5#
10.1
adja
0 pa
  vi
```

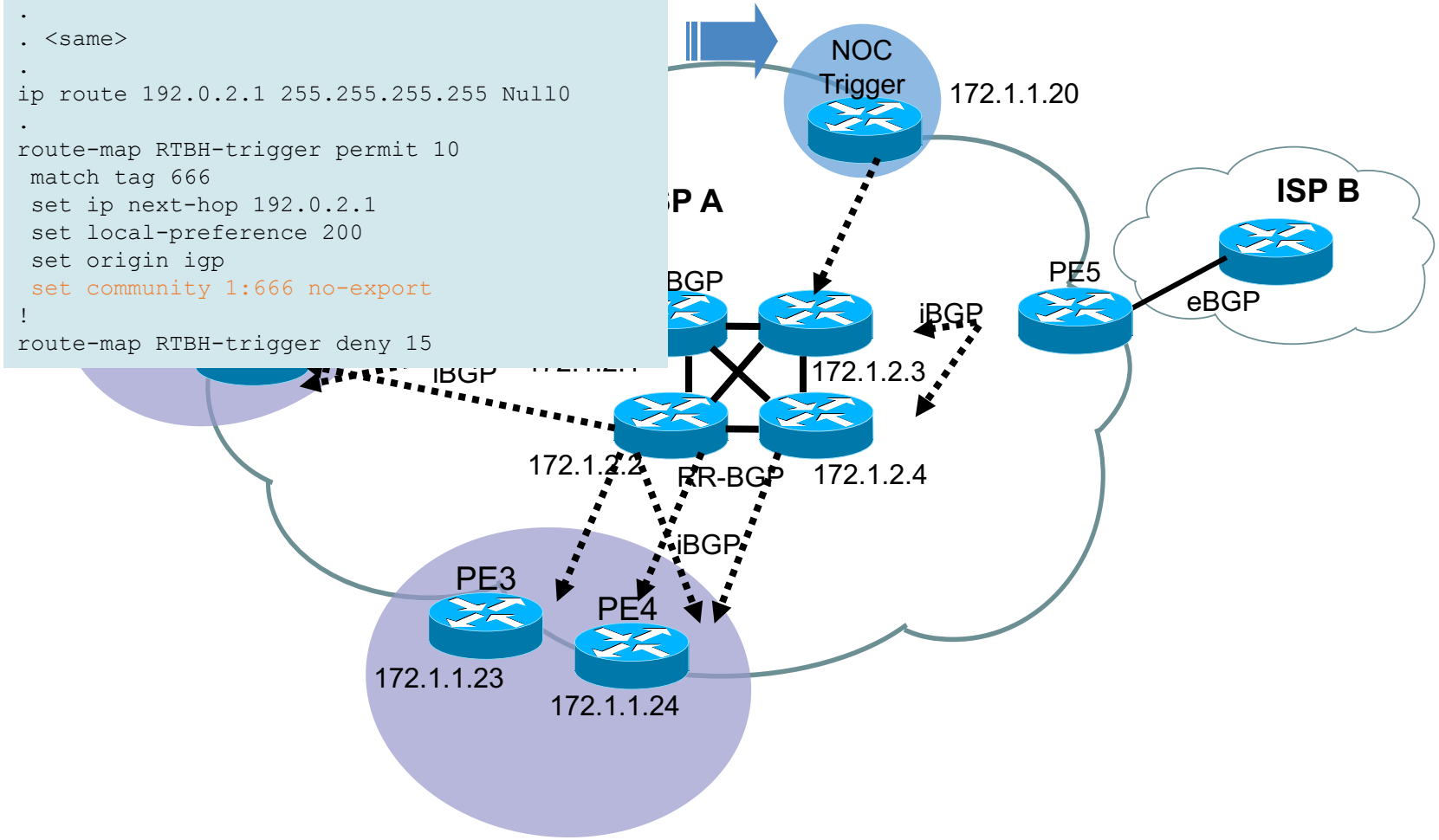
```
After:
PE5#sh ip int POS1/1/0 | incl verif
  IP verify source reachable-via ANY
   29 verification drops
    0 suppressed verification drops

PE5#sh ip cef 10.1.1.1
10.1.1.1/32, version 18896, epoch 0, cached adjacency to POS1/1/0
0 packets, 0 bytes
  tag information from x.x.x.x/32, shared, all rewrites owned
    local tag: 44
  via x.x.x.x, 0 dependencies, recursive
    next hop 9.0.1.2, POS1/1/0 via x.x.x.x/32 (Default)
    valid cached adjacency
    tag rewrite with PO1/1/0, point2point, tags imposed {}
```

```
Attacker:
Rtr#ping 192.168.1.100
Sending 10000, 100-byte ICMP
Echos to 192.168.1.100
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!.....................
....
```

Verification…

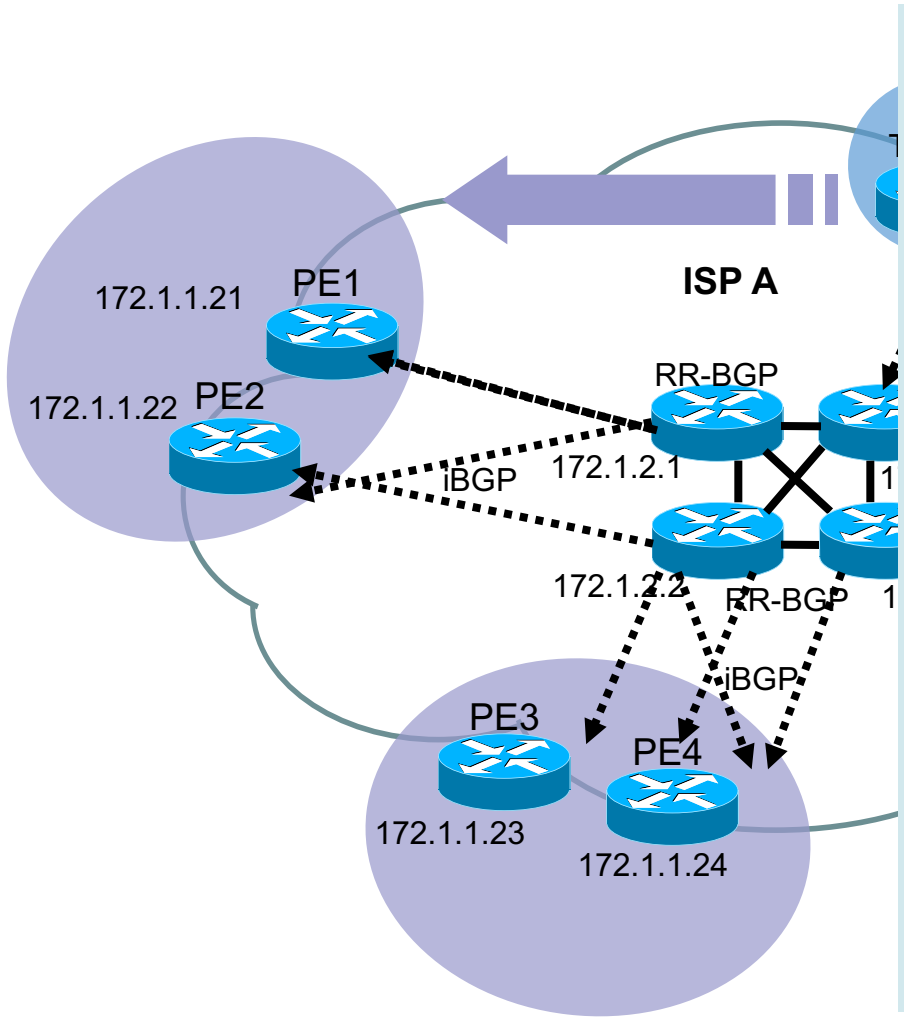
RTBH: Global and Regional Deployment Example: Basic Community-Based Drops

```
.
. <same>
.
ip route 192.0.2.1 255.255.255.255 Null0
.
route-map RTBH-trigger permit 10
 match tag 666
 set ip next-hop 192.0.2.1
 set local-preference 200
 set origin igp
 set community 1:666 no-export
!
route-map RTBH-trigger deny 15
```



Trigger Router Pre-Configuration

RTBH: Global and Regional Deployment Example: Basic Community-Based Drops



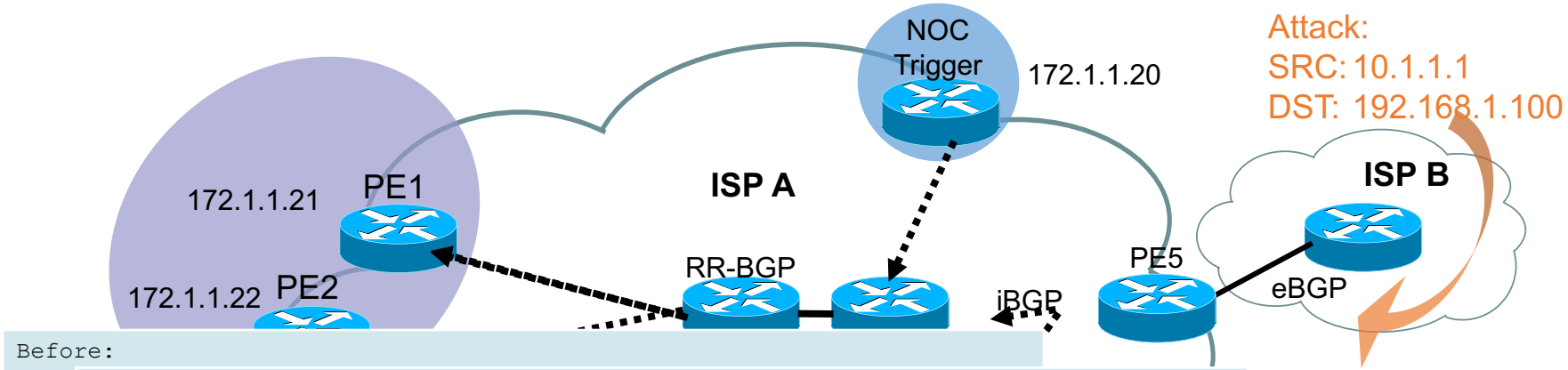

```
.
. <same>
.
router bgp 1
 no synchronization
 bgp router-id 172.1.1.20
 bgp log-neighbor-changes
 neighbor RR-client peer-group
 neighbor RR-client remote-as 1
 neighbor RR-client update-source Loopback0
 neighbor RR-client version 4
 neighbor RR-client send-community both
 neighbor RR-client route-map RTBH-drop in
 neighbor 172.1.2.1 peer-group RR-client
 neighbor 172.1.2.2 peer-group RR-client
 no auto-summary
!
ip route 192.0.2.1 255.255.255.255 Null0
!
ip community-list 1 permit 1:666
ip as-path access-list 1 permit ^$
!
route-map RTBH-drop permit 10
 match community 1
 match as-path 1
 set ip next-hop 192.0.2.1
 set local-preference 200
 set origin igp
 set community no-advertise additive
!
route-map RTBH-drop permit 15
```

PE1 Edge Router Pre-Configuration

(PE2, PE3, PE4, PE5 Similar)

RTBH: Global and Regional Deployment Example: Basic Community-Based Drops

```
trigger(config)#ip route 192.168.1.100 255.255.255.255 null0 tag 666
```



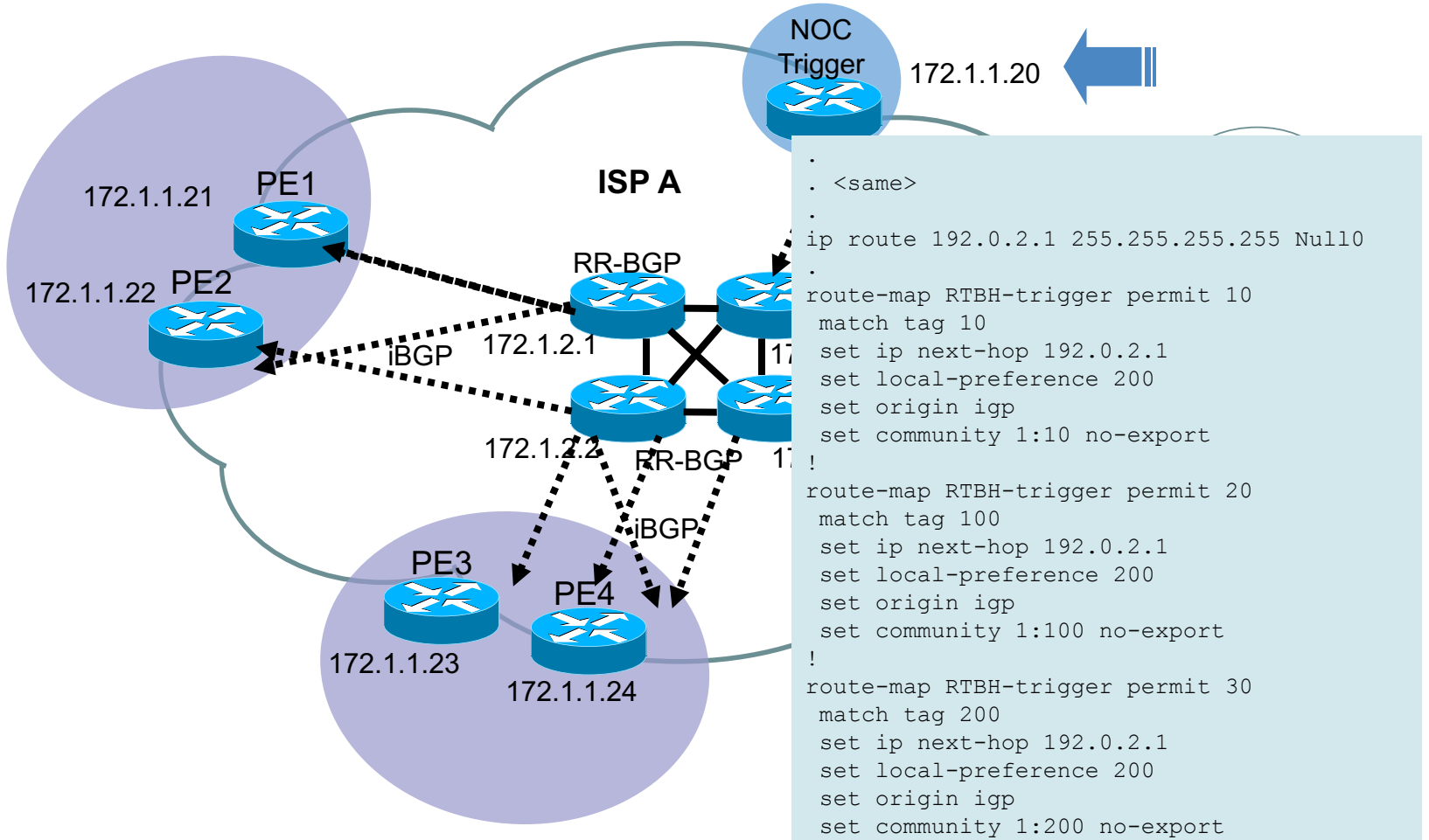

```
Before:
PE1#
192.
0 pa
  ta

  vi
```

```
After:
PE1#sh ip bgp community-list 1
BGP table version is 11, local router ID is 9.0.0.15
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i192.168.1.100/32 192.0.2.1                0    200      0 i
*>i                 192.0.2.1                0    200      0 i
PE1#sh ip cef 192.168.1.100
192.168.1.100/32, version 18858, epoch 0
0 packets, 0 bytes
  tag information from 192.0.2.1/32, shared, all rewrites inherited
    local tag: implicit-null
  via 192.0.2.1, 0 dependencies, recursive
    next hop 192.0.2.1, Null0 via 192.0.2.1/32 (Default)
    valid null adjacency
```

```
Attacker:
Rtr#ping 192.168.1.100
Sending 10000, 100-byte ICMP
Echos to 192.168.1.100
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!.....................
.......
```

Verification…

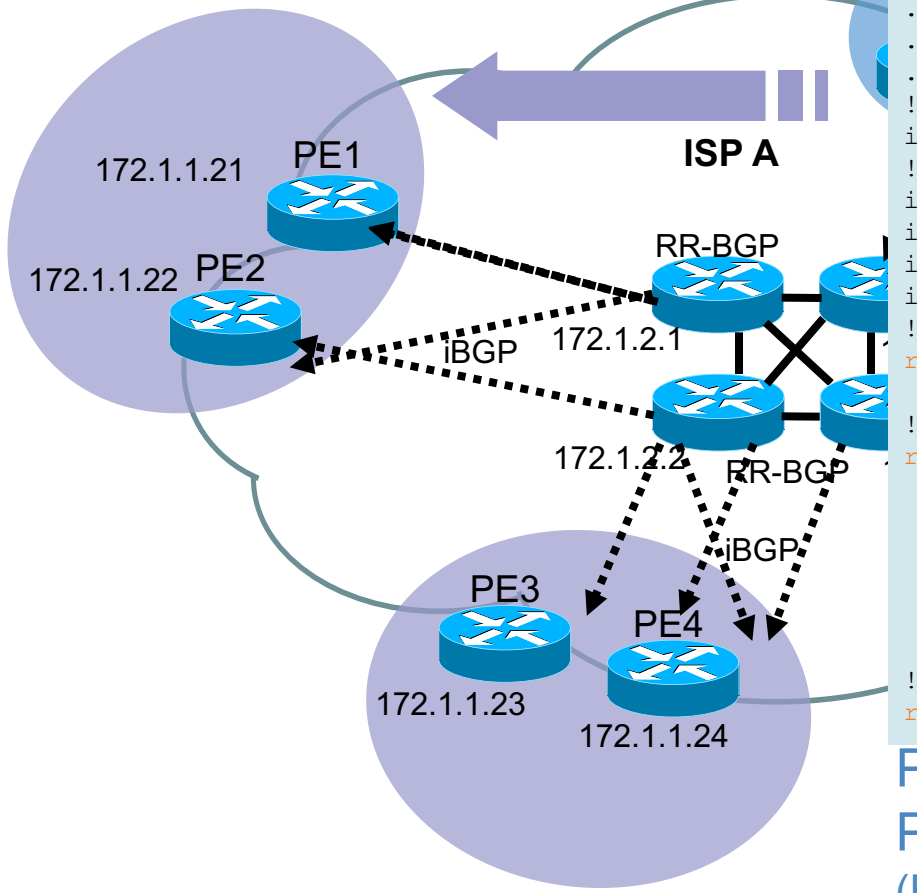
RTBH: Global and Regional Deployment Example: Regional Community-Based Drops




```
.
. <same>
.
ip route 192.0.2.1 255.255.255.255 Null0
.
route-map RTBH-trigger permit 10
 match tag 10
 set ip next-hop 192.0.2.1
 set local-preference 200
 set origin igp
 set community 1:10 no-export
!
route-map RTBH-trigger permit 20
 match tag 100
 set ip next-hop 192.0.2.1
 set local-preference 200
 set origin igp
 set community 1:100 no-export
!
route-map RTBH-trigger permit 30
 match tag 200
 set ip next-hop 192.0.2.1
 set local-preference 200
 set origin igp
 set community 1:200 no-export
!
route-map RTBH-trigger deny 40
```

Tr

RTBH: Global and Regional Deployment Example: Regional Community-Based Drops

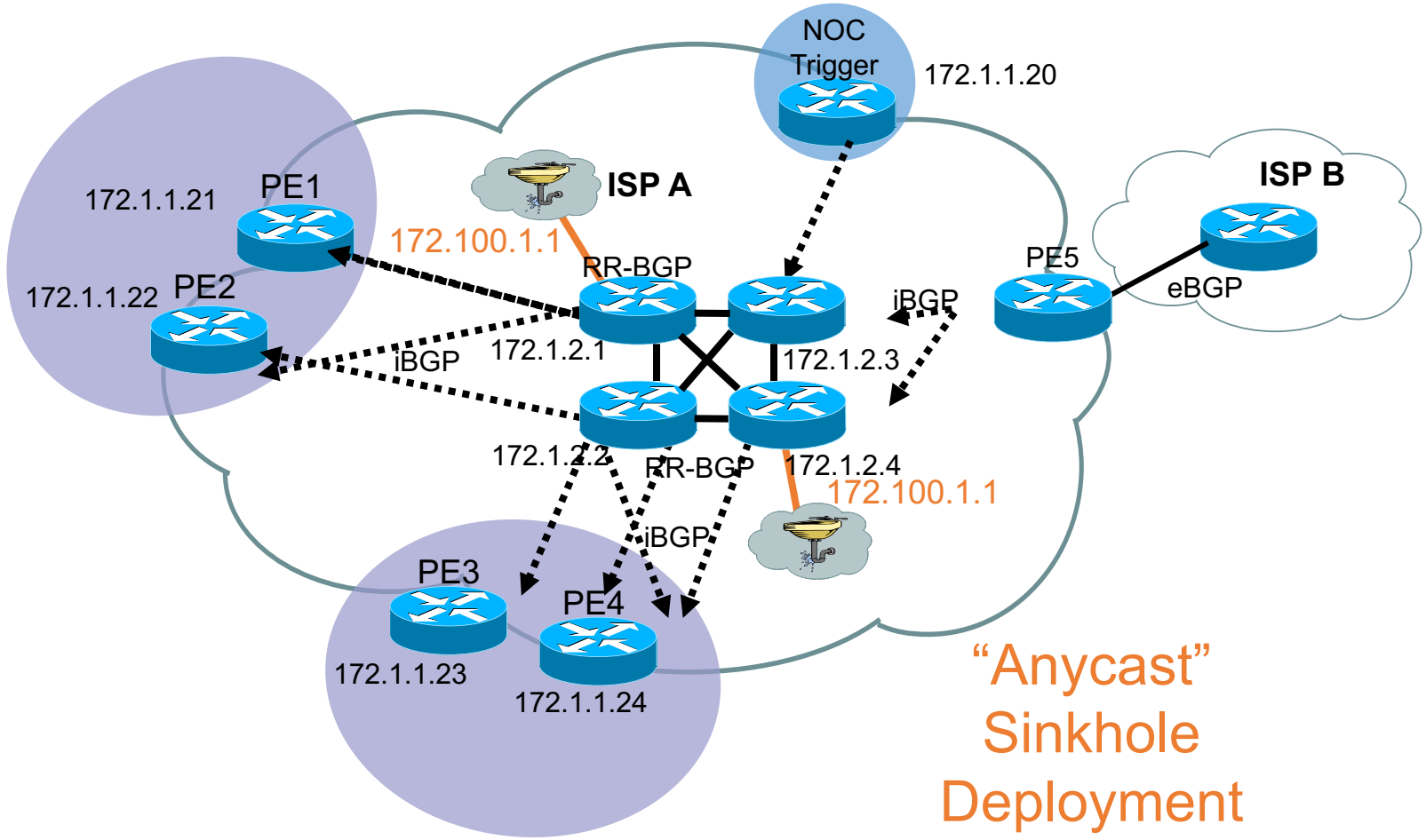



```
!
hostname PE1
.
. <same>
.
!
ip route 192.0.2.1 255.255.255.255 Null0
!
ip community-list 1 permit 1:10
ip community-list 1 permit 1:100
ip community-list 2 permit 1:200
ip as-path access-list 1 permit ^$
!
route-map RTBH-drop deny 10
 match community 2
!
route-map RTBH-drop permit 20
 match community 1
 match as-path 1
 set ip next-hop 192.0.2.1
 set local-preference 200
 set origin igp
 set community no-advertise additive
!
route-map RTBH-drop permit 15
```

PE1 Edge Router Pre-Configuration

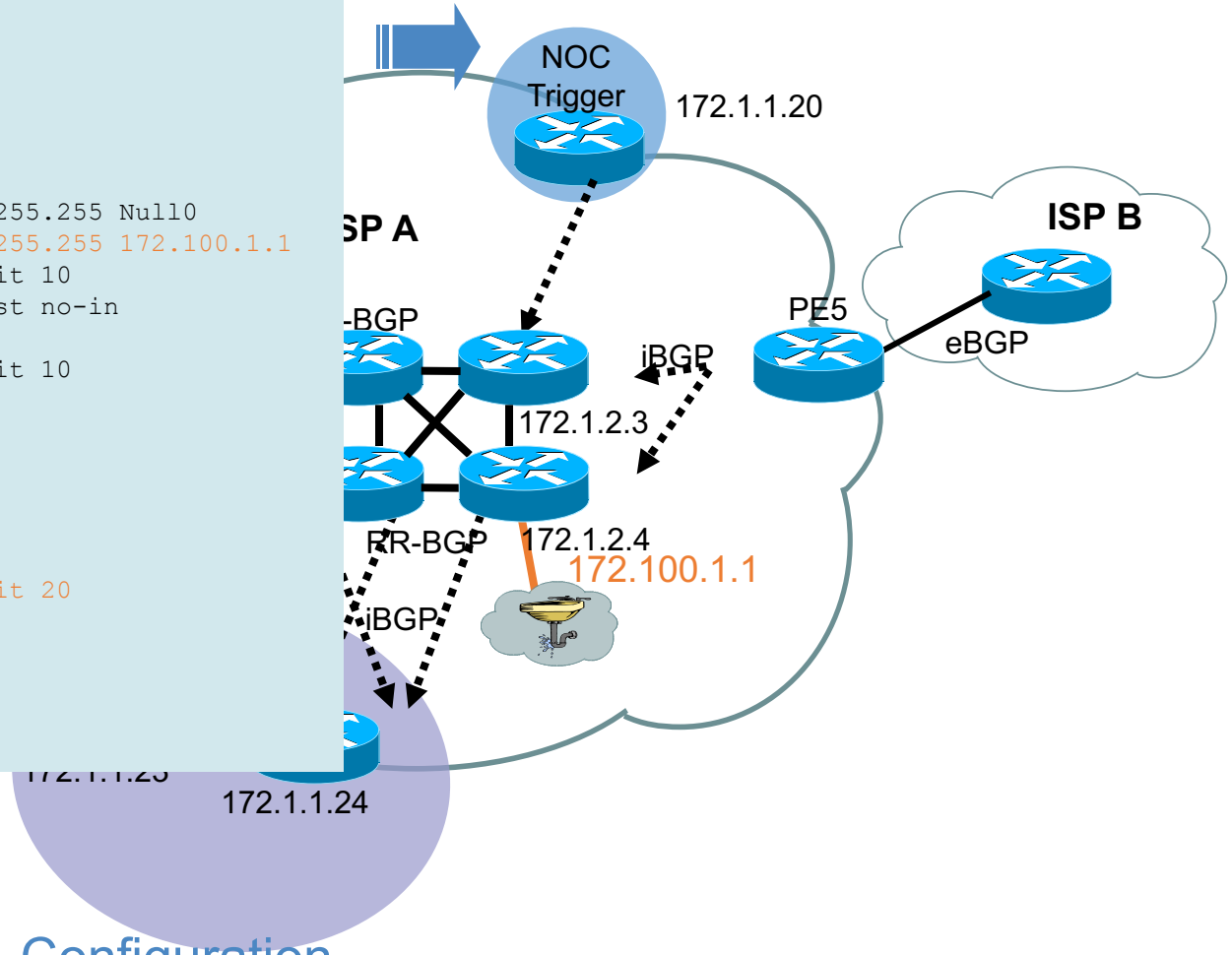
(PE2, PE3, PE4, PE5 Similar)

RTBH: Global and Regional Deployment Example: Basic Tag-Based Sinkhole



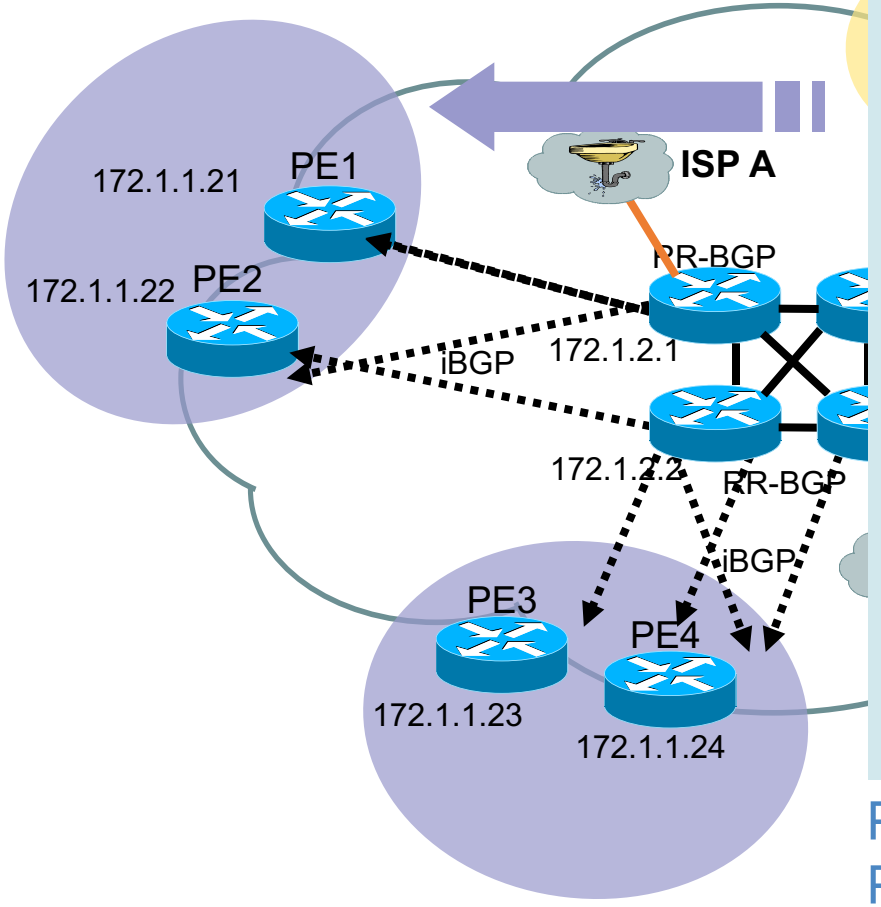
RTBH: Global and Regional Deployment Example: Basic Tag-Based Sinkhole

```
!
hostname trigger
!
.
. <same>
.
ip route 192.0.2.1 255.255.255.255 Null0
ip route 192.0.2.2 255.255.255.255 172.100.1.1
route-map no-prefix-in permit 10
 match ip address prefix-list no-in
!
route-map RTBH-trigger permit 10
 match tag 666
 set ip next-hop 192.0.2.1
 set local-preference 200
 set origin igp
 set community no-export
!
route-map RTBH-trigger permit 20
 matcch tag 411
 set ip next-hop 192.0.2.2
 set local-preference 200
 set origin igp
 set community no-export
```



Trigger Router Pre-Configuration

RTBH: Global and Regional Deployment Example: Basic Tag-Based Sinkhole

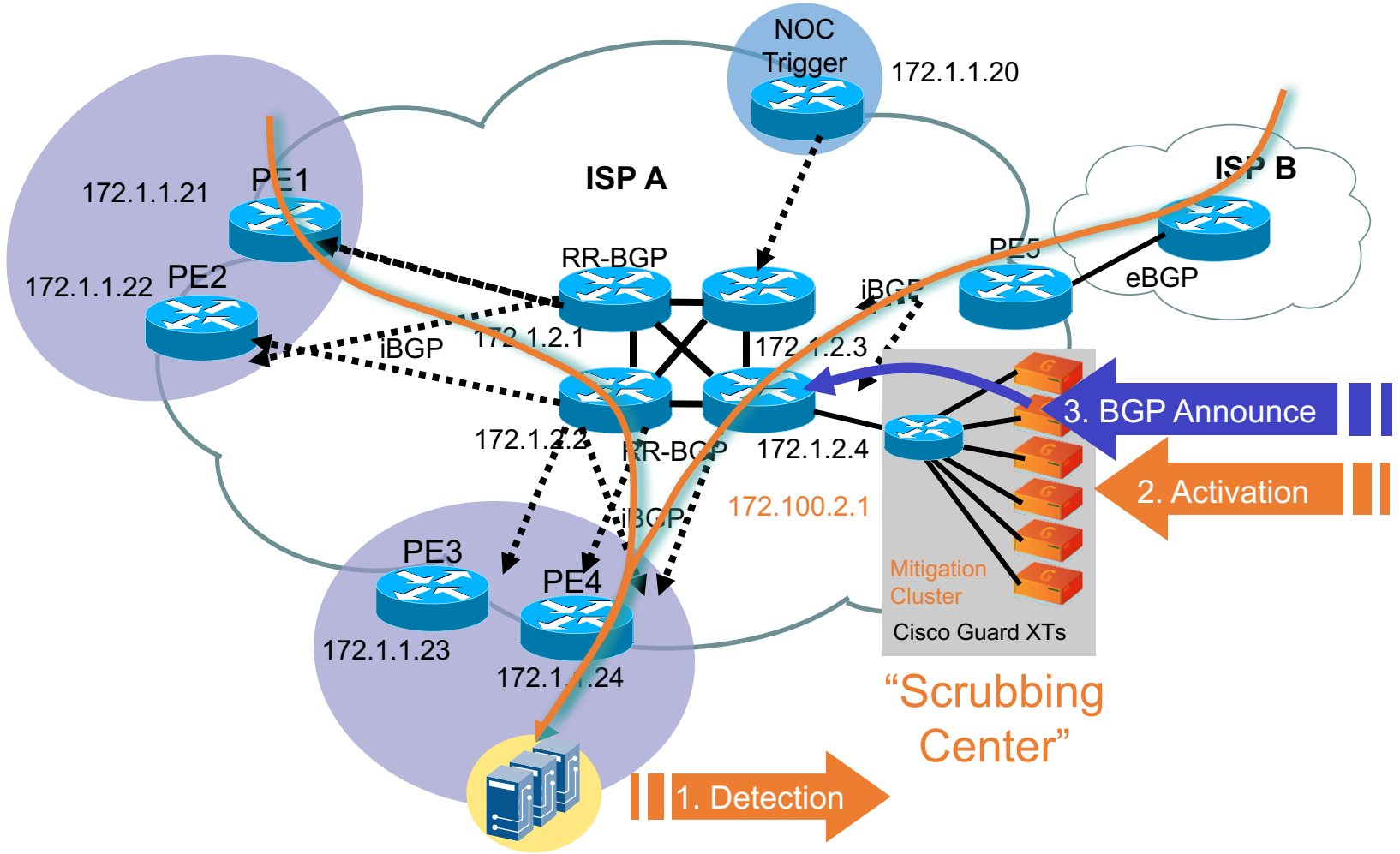



```
!
hostname PE1
ip bgp-community new-format
!
interface Loopback0
 ip address 172.1.1.21 255.255.255.255
!
interface Null0
 no ip unreachables
!
router bgp 1
 no synchronization
 bgp router-id 172.1.1.20
 bgp log-neighbor-changes
 neighbor RR-client peer-group
 neighbor RR-client remote-as 1
 neighbor RR-client update-source Loopback0
 neighbor RR-client version 4
 neighbor RR-client send-community both
 neighbor 172.1.2.1 peer-group RR-client
 neighbor 172.1.2.2 peer-group RR-client
 no auto-summary
!
ip route 192.0.2.1 255.255.255.255 Null0
ip route 192.0.2.2 255.255.255.255 172.100.1.1
!
```

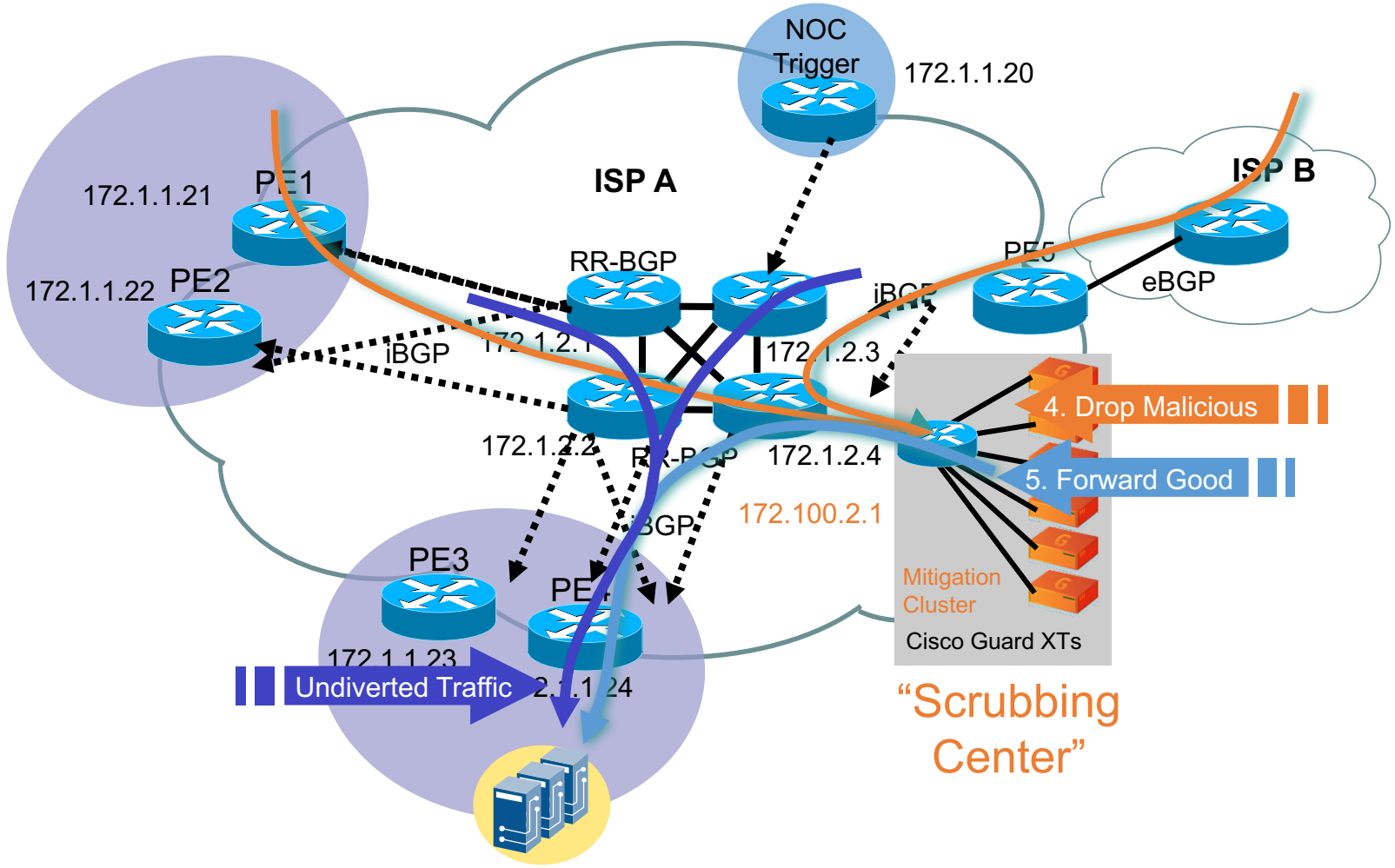
PE1 Edge Router Pre-Configuration

(PE2, PE3, PE4, PE5 Similar)

RTBH: Global and Regional Deployment Example: Basic Scrubbing



RTBH: Global and Regional Deployment Example: Basic Scrubbing



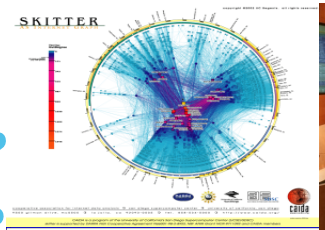
RTBH Considerations

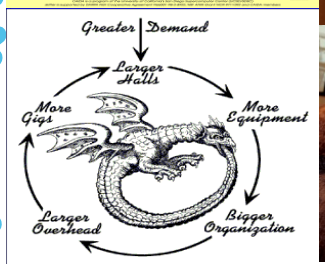
- Tag-based approach:
  - Concentrates configuration complexity on one “trigger” router
  - Edge devices require simple static route to Null0 (and uRPF deployments for source-based drops)
  - Monitoring (OpEx)—Prefixes which are being dropped (and why) best viewed on “trigger” router (e.g., “show run | include tag”)
- Community-based approach:
  - Configuration complexity spread equally to all devices
  - Allows greater flexibility for drop control (e.g., regional)
  - Provides ability to allow Customers to “self-drop” their own traffic through eBGP peering (see: MCI/UUnet NANOG 30 paper)
  - Monitoring (OpEx)—Prefixes which are being dropped on a particular device (and why) can be determined by reviewing the output of “sh ip bgp community” on that device

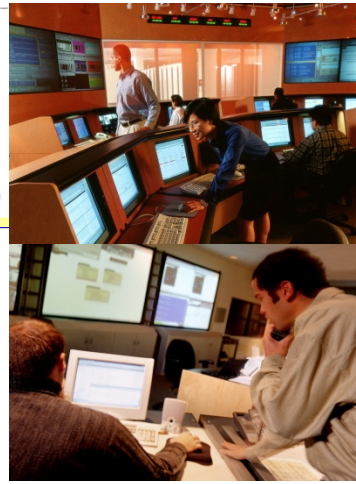
Reacting to an Attack with RTBH Techniques

- Key advantages:
  - No ACL updates in the heat of battle
  - No change to the router's configs
  - Drops happen in the forwarding path
  - Frequent changes occur when attacks are dynamic (or multiple attacks on multiple customers)—RTBH provides an excellent tool for dynamic changes
- Key disadvantages:
  - Granularity to L3 address information only
  - Unless a "scrubbing" device and diversion is used, RTBH drops all traffic with specified source and/or destination address

Extra Slides







Remote Triggered Black Hole Filtering

- How can we do that?
  - *Black Hole Filtering*: if the *destination* address equals Null0 we drop the packet.
  - *Remote Triggered*: trigger a prefix to equal Null0 on routers across the Network at iBGP speeds.
  - uRPF Loose Check: If the *source* address equals Null0, we drop the packet.
- Answer: Put them together and we have a tool to trigger drop for any packet coming into the network whose source or destination equals Null0!

Why Blackhole Traffic?

- Mitigate denial of service (DoS) attacks
  - Prevention is another matter all together
- Protect vital network resources from outside attack
- Provide protection services for customers
  - Customer can initiate it’s own protection
- Log and track DoS attempts/attacks

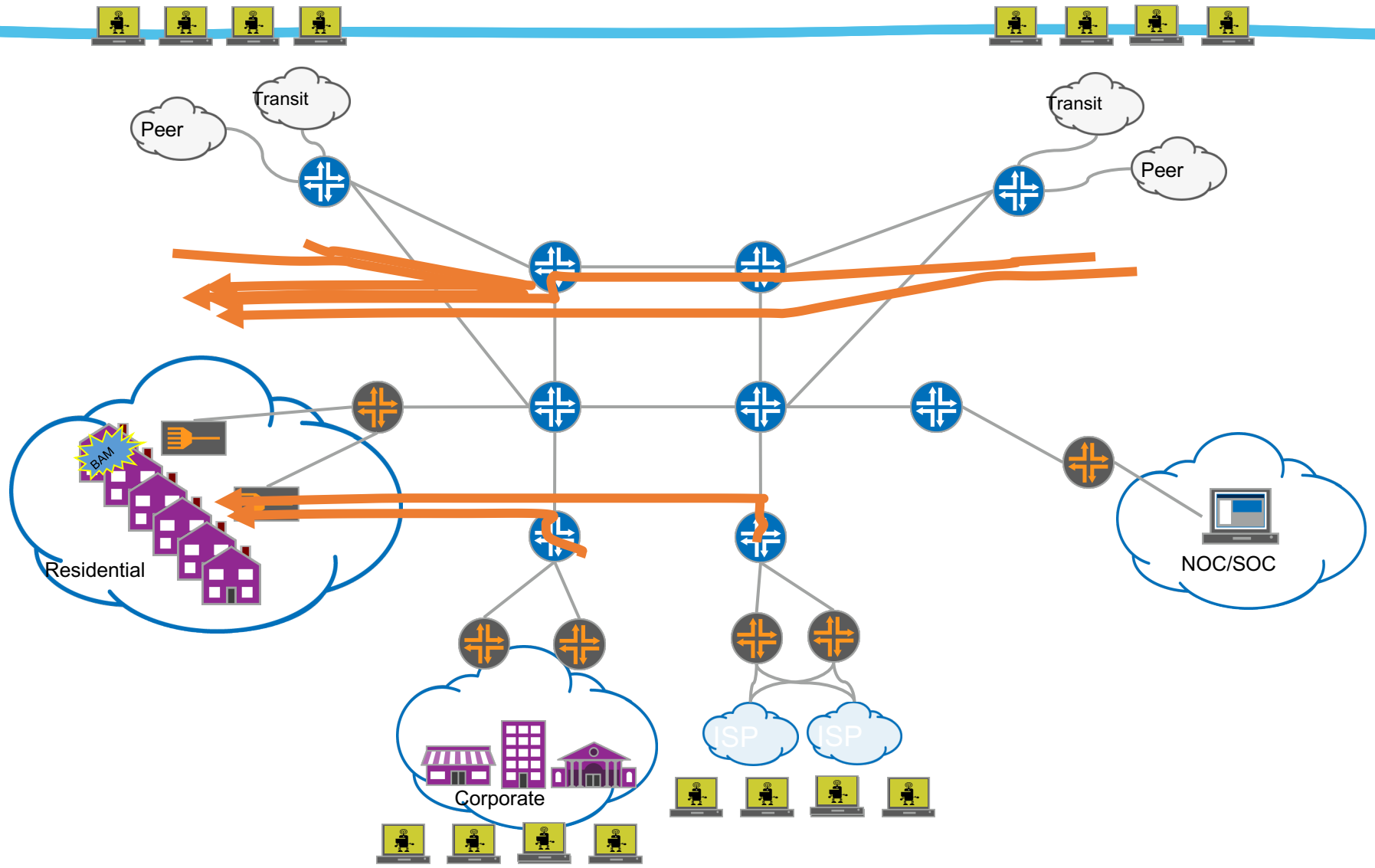
Requirements for Blackhole Routing

- Effective overall plan
  - Routing policies and route maps
  - Discard interfaces and null static routes
  - Good internal routing knowledge
- Willingness to install a potentially complex (dangerous?) system
  - Policies / route maps on all routers in the network
  - Potential for misuse
- Operational Guidelines
  - Strict access control and command logging
  - Audits to clean up stale blackhole routing
  - Integrate with existing NMS

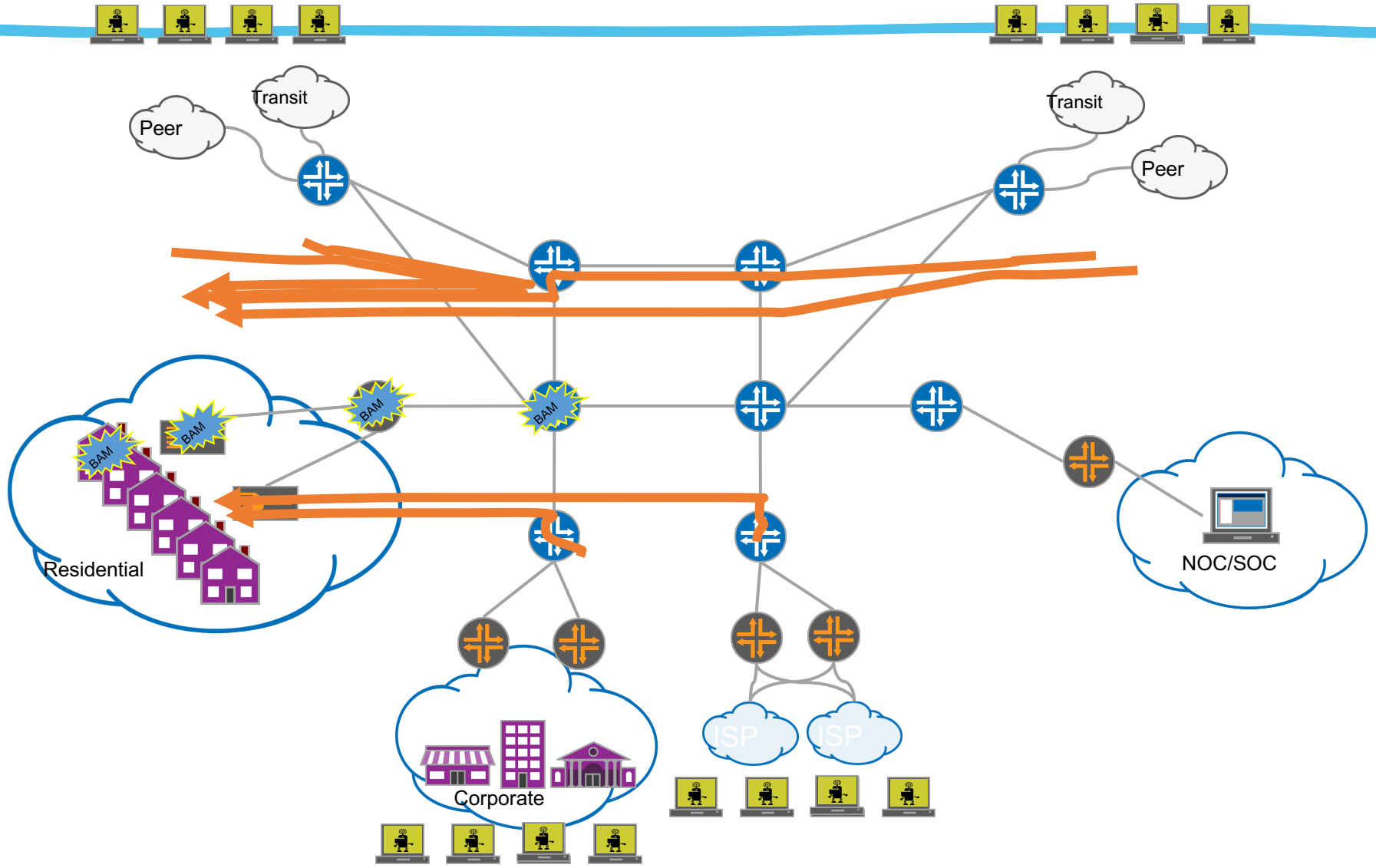
Who Should be Blackholed?

- Attacks to customers
  - From peers and/or other customers
- Attacks from customers
  - To peers and/or other customers
- Attack Controllers
  - Hosts providing attack instructions
- Unallocated address spaces?
  - BOGONS

Attacks Toward Customers



Co-lateral Damage in the Network



Discard Options

Discard/Reject Static Route

- On Juniper routers, create a static route for each next-hop used for blackholed routes
  - Select either discard or reject as the next-hop action
  - Be cautious of ICMP rate-limiting with reject action!

```
user@host> show configuration routing-options
static {
        route 192.0.2.101/32 discard;
        route 192.0.2.103/32 reject;
        route 192.0.2.105/32 discard;
}
user@host> show route protocol static terse

inet.0: 24 destinations, 24 routes (24 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

A Destination      P Prf     Metric 1     Metric 2    Next hop    AS path
*  192.0.2.101/32  S  5                               Discard
*  192.0.2.103/32  S  5                               Reject
*  192.0.2.105/32  S  5                               Discard
```

Null 0 Static Route

- On Cisco routers, create a separate static route to Null 0 for each next-hop you are assigning to the blackhole routes

```
ip route 192.0.2.101 255.255.255.255 Null0
ip route 192.0.2.103 255.255.255.255 Null0
ip route 192.0.2.105 255.255.255.255 Null0

ISP-BLKHOLE-RTR1#sh ip route 192.0.2.0
Routing entry for 192.0.2.0/32, 3 known subnets
         Attached (3 connections)

S 192.0.2.103 is directly connected, Null0
S 192.0.2.101 is directly connected, Null0
S 192.0.2.105 is directly connected, Null0

ISP-BLKHOLE-RTR1#
```

Discard Interface

- Alternatively on a Juniper router, create a discard interface which contains the next-hop you are assigning to the blackhole routes
  - Multiple IP addresses on a single logical unit
  - Use of the destination command works best
- This allows you to configure and assign filters to the interface for counting, logging, and sampling the traffic
  - Only a discard next-hop action is used

Discard Interface

```
user@host> show configuration interfaces dsc
unit 0 {
 family inet {
        address 192.0.2.102/32 {
                destination 192.0.2.101;
        }
        address 192.0.2.104/32 {
                destination 192.0.2.103;
        }
        address 192.0.2.106/32 {
                destination 192.0.2.105;
        }
      }
   }

user@host> show interfaces terse dsc

Interface               Admin Link    Proto    Local           Remote
dsc                      up    up
dsc.0                    up    up    inet     192.0.2.102     --> 192.0.2.101
                                             192.0.2.104     --> 192.0.2.103
                                             192.0.2.106     --> 192.0.2.105
```

Mapping Routes to Blackholes

Mapping Routes to Blackhole Services

- To actually blackhole packets requires either a routing policy or a route map attached to the BGP sessions
- Blackhole eligible packets must be located
  1. Use a route-filter or access list to locate individual routes (administratively hard)
  2. Use a BGP community value (quite scalable)

Locating Specific Routes w/ Route-Filter

```
protocols {
        bgp {
            import blackhole-by-route;
            }
        }

policy-options {
     policy-statement blackhole-by-route {
          term specific-routes {
              from {
                   route-filter 10.10.10.1/32 exact;
                   route-filter 10.20.20.2/32 exact;
                   route-filter 10.30.30.3/32 exact;
                   route-filter 10.40.40.4/32 exact;
                   }
              then {
                   next-hop 192.0.2.101
                   }
                 }
               }
              }
```

Locating Specific Routes w/ Access List

```
access-list 88 remark Bogon Filter List: v2.4 28 Apr 2004
access-list 88 permit 0.0.0.0 1.255.255.255
access-list 88 permit 2.0.0.0 0.255.255.255
access-list 88 permit 5.0.0.0 0.255.255.255
access-list 88 permit 7.0.0.0 0.255.255.255
access-list 88 permit 10.0.0.0 0.255.255.255
access-list 88 permit 23.0.0.0 0.255.255.255
access-list 88 permit 27.0.0.0 0.255.255.255
access-list 88 permit 31.0.0.0 0.255.255.255
access-list 88 permit 36.0.0.0 1.255.255.255
access-list 88 permit 39.0.0.0 0.255.255.255
.
.
etc (good source is BOGON route object (fltr-bogons) in RADB)

route-map BLKHOLE-ROUTES-IN permit 10
  match ip address 88
  set ip next-hop 192.0.2.101
!
neighbor 172.16.1.1 route-map BLKHOLE-ROUTES-IN in
```

Use Communities to Locate Routes

```
protocols {
        bgp {
            import blackhole-policy;
            }
            }

policy-options {
     policy-statement blackhole-policy {
          term blackhole-communities {
               from {
                   community blackhole-all-routers;
                   }
               then {
                   next-hop 192.0.2.101
                   }
                }
             }
       community blackhole-all-routers members 65200:55..$
    }
```

Use Communities to Locate Routes

- Different Community Regular Expression
  - JUNOS/IOS Implementations Differ
  - IOS: Regex applies to entire set of values
  - JUNOS: Regex applies to individual community

```
ip community-list expanded BLKHOLE-ALL-ROUTERS permit 65200:55.._

route-map BLKHOLE-ROUTES-IN permit 10
  match community BLKHOLE-ALL-ROUTERS
  set ip next-hop 192.0.2.101
!

neighbor 172.16.1.1 route-map BLKHOLE-ROUTES-IN in
```

Blackhole Communities

- A clear list of possible communities is needed Easier troubleshooting and operation
  - Better security
- Some hierarchy and stratification is good
  - Routes accepted from customers (5500-5509)
  - Injected customer routes (5520-5529)
  - Injected BOGON routes (5530-5539)
  - Injected provider routes (5540-5549)

Blackhole Communities

- CUST-ANNOUNCE-BLKHOLE-ALL
  - 65200:5501
  - Next-hop set to 192.0.2.101
  - Packets are dropped on all possible routers
- CUST-ANNOUNCE-BLKHOLE-PEER
  - 65200:5503
  - Next-hop set to 192.0.2.103
  - Packets are dropped on Peer routers only
  - Allows customer to stop attacks from off-net while continuing flows from other on-net connections

Blackhole Communities

- ISP-BLKHOLE-CUST-ALL
  - 65200:5521
  - Next-hop set to 192.0.2.101
  - Packets are dropped on all possible routers
- ISP-BLKHOLE-CUST-PEER
  - 65200:5523
  - Next-hop set to 192.0.2.103
  - Packets are dropped on Peer routers only
- ISP-BLKHOLE-CUST-CORE
  - 65200:5525
  - Next-hop set to 192.0.2.105
  - Packets are dropped on all but Edge routers

Blackhole Communities

- ISP-BLKHOLE-BOGON-MARTIAN
  - 65200:5530
  - Next-hop set to 192.0.2.101
  - Packets are dropped on all possible routers
  - Routes include things like 127/8, 128.0/16, and 192.0.0/24
- ISP-BLKHOLE-BOGON-RFC-1918
  - 65200:5531
  - Next-hop set to 192.0.2.101
  - Packets are dropped on all possible routers
  - Routes are 10/8, 172.16/12, and 192.168/16

Blackhole Communities

- ISP-BLKHOLE-BOGON-IANA-RSVD
  - 65200:5532
  - Next-hop set to 192.0.2.101
  - Packets are dropped on all possible routers
  - Routes match the list of current reserved addresses
- ISP-BLKHOLE-BOGON-PUBLIC-EXCHANGE
  - 65200:5533
  - Next-hop set to 192.0.2.101
  - Packets are dropped on all possible routers
  - Routes include subnet addresses from public peering points the ISP is not attached to

Blackhole Communities

- ISP-BLKHOLE-INFRA-PEERING-LINKS
  - 65200:5540
  - Next-hop set to 192.0.2.101
  - Packets are dropped on all possible routers
  - Routes include the peering connections of the ISP
- ISP-BLKHOLE-INFRA-LAN
  - 65200:5541
  - Next-hop set to 192.0.2.101
  - Packets are dropped on all possible routers
  - Routes include subnet addresses for protected internal services

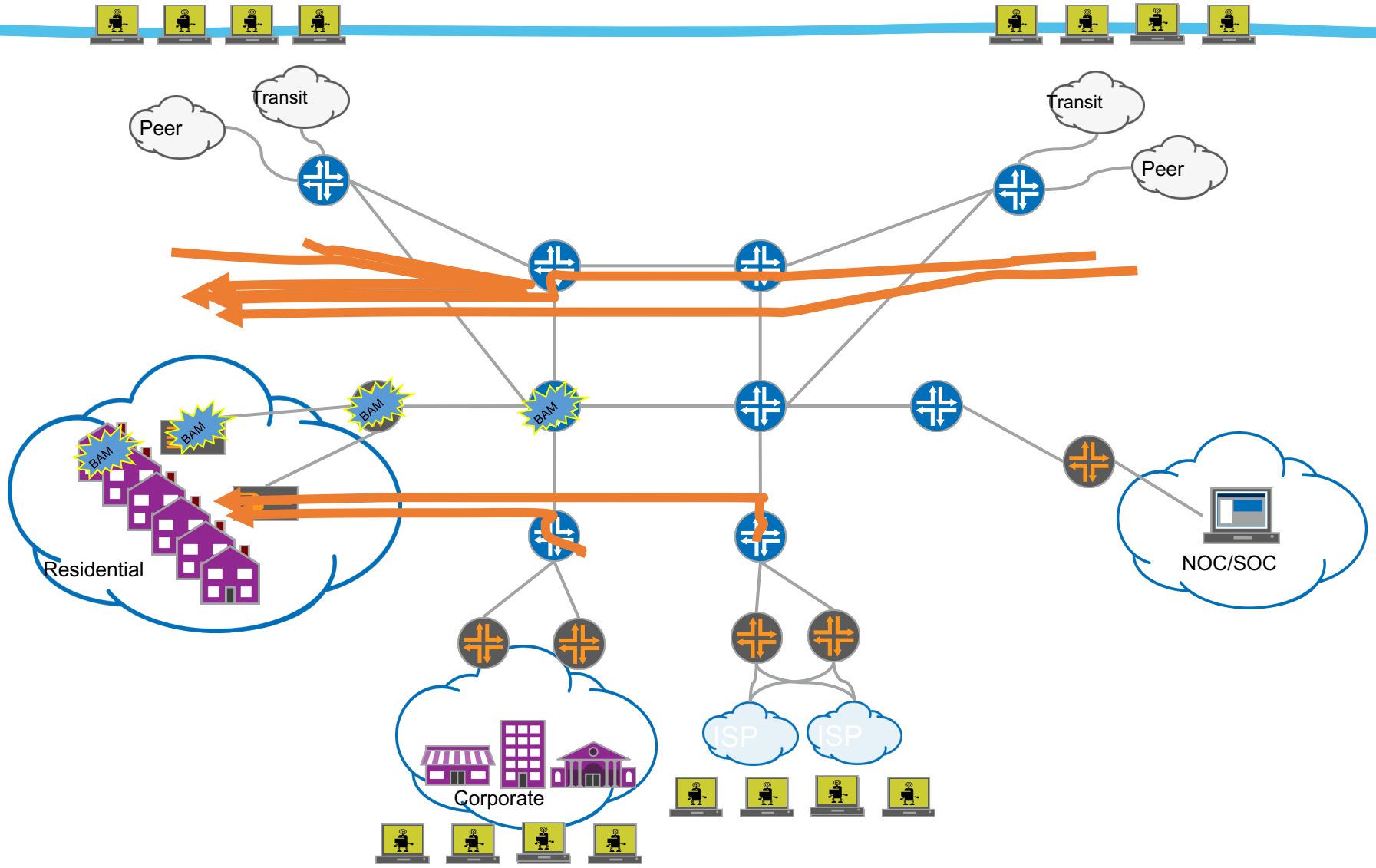
Using EBGP Multihop

- By default, next-hop must equal the EBGP peer address
- Altering the next-hop for blackhole services requires multihop on the EBGP sessions

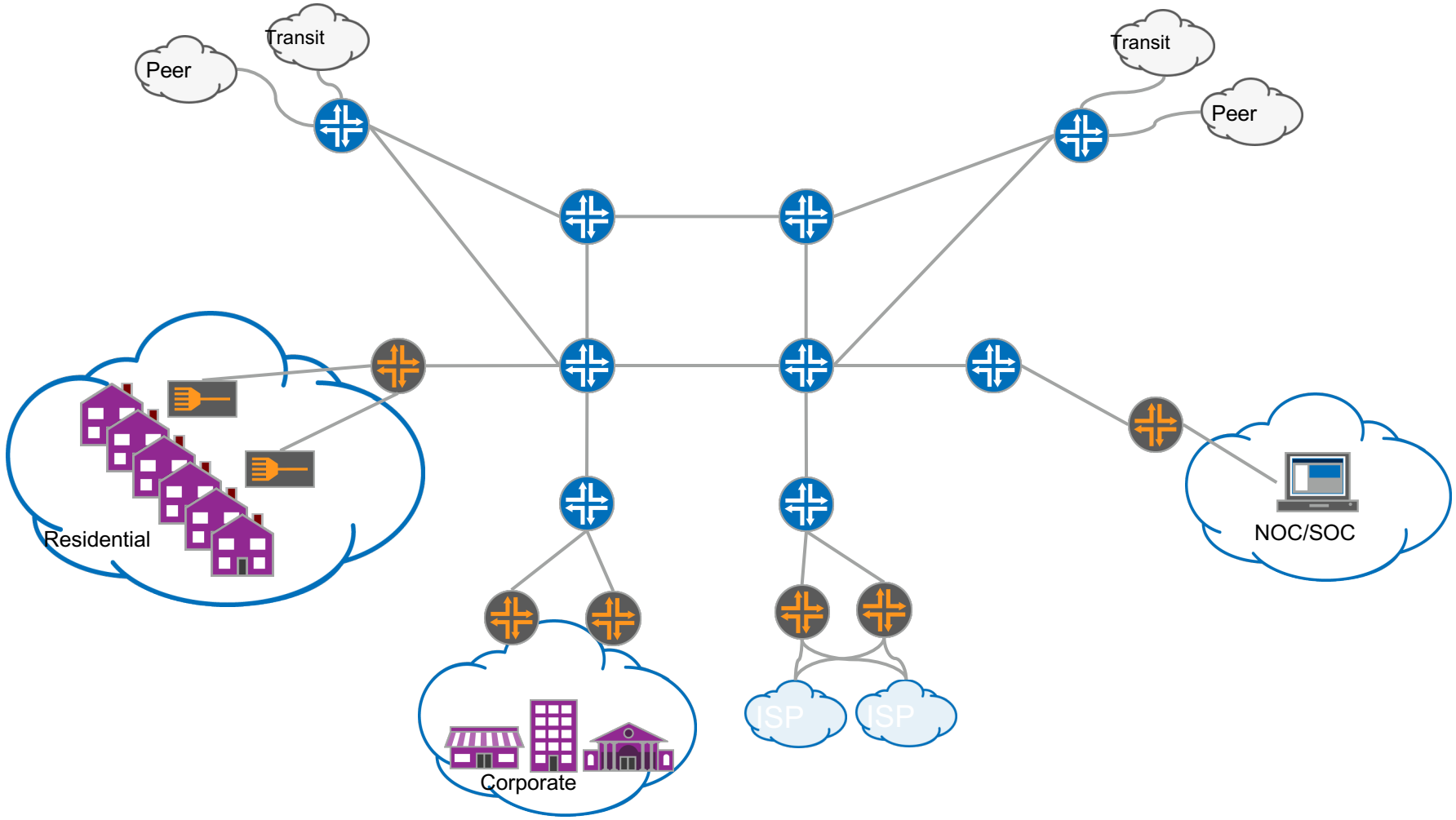
Multihop Configurations – Edge/Peer

```
protocols {
        bgp {
           group EBGP-Peers {
               neighbor 172.16.1.1;
               type external;
               peer-as 65432;
               multihop;
               local-address 172.16.254.254;
               import blackhole-policy-edge;
             }
           }
         }

router bgp 65200
   neighbor 172.16.1.1 remote-as 65432
   neighbor 172.16.1.1 description “ISP CORE-RTR1"
   neighbor 172.16.1.1 ebgp-multihop 2
   neighbor 172.16.1.1 send-community
   neighbor 172.16.1.1 route-map BLKHOLE-POLICY-EDGE in
```

Co-lateral Damage in the Network



Network Map


Peer
Transit
Transit
Peer
Residential
NOC/SOC
Corporate
ISP
ISP