BGP Made Easy

John van Oppen

NANOG ON THE ROAD

September 14th 2017

What is BGP

- Snarky answer: RFC-4271
- BGP is an Exterior gateway protocol, the only one used on the public Internet and is used for inter-Autonomous System routing. (IE between discrete networks)
- BGP distributes (signals) the path to every destination on the Internet, the core of major providers typically don't contain a default route, they contain the paths to every prefix on the Internet.
- BGP learns multiple paths to a given route and selects the best path, only best path is sent between routers.

Typical reasons for running BGP

- Multihoming / Provider redundancy
- Equipment / Port redundancy
- Peering (privately or at an IX)
- Connectivity quality (better paths)
- Remote triggered black-holing

How does BGP work?

- Divides each network into Autonomous systems
- Exchanges routing information to build a global routing table for the Internet.
- Allows application of routing policy to implement business needs.

What is an Autonomous System?

- Typically:
  - A single network of one or more routers redundantly interconnected, but could be a single router.
  - Controlled by a single administrative domain (one company could have several ASNs but a given ASN is typically controlled by a specific group)
  - Common routing policy
  - Identified by a globally unique AS Number (ASN)

Exchanging routes

- Exchanging routes is done by “peering” but this does not mean its free (that is the other type of peering)
- Neighbors are setup on both sides, policy applied and executed.
- Static configuration of neighbors, unlike most other routing protocols

Enforcing and specifying (operational) policy

- Only accepting certain routes
  - Prefix lists / prefix sets etc
  - Community matching, etc (prepend to peers etc)
- Specific policy on a per route basis
  - RTBH or other custom features for special policies for single routes

Enforcing (business) policy

- Typical problem: I don't want to send routes received from my transit to someone who does not pay me.
  - Common solution: AS-path filtering, prefixes lists or a combination at **egress**.

Proper policy important = no leaks

– Route leaks are bad… Causing:

- Outages for others
- Outages for you
- Over-running links with traffic that you are not getting paid for (Yikes)
- Black holing something that the rest of nanog shames you for. ;)

Types of peering relationships

- Transit (routes that cost money to send traffic across)
- Peering (typically free, you see my customers, I see yours without charge) - peers across NWAX would be a good example of this type of relationship
- Customer (routes that are sourced from paying customers)
- Typically type of relationship dictates local-preference setting (50, 95 and 110 in this example)

Filtering tools for BGP routes (cisco)

- Prefix lists can be applied directly to BGP peer configuration
- Route maps can match various things, the most important for BGP are:
  - prefix lists
  - As-path access lists
  - Community lists
  - Metric
- IOS-XR replaces all of this with route-policies

Single policy solution: filter with communities

- Allows matching by relationship (IE BGP peer) and not just prefix
- Routes are tagged at ingress with to tell the rest of the AS what to do with them so that all egress filtering is automatic and shifts with policy applied at ingress.

Communities, tags for routes!

- Filtering only at ingress works for very small ASNs and very large ASNs alike, future proof the network!
- Allows for large ASes with lots of customer routes to scale by only filtering on customer sessions, no master prefix list, etc.
- Egress filter policy can be setup to deny by default (IE no community of the right type attached to route means the route is not exported). Typos less often result in leaks!
- Allows easy filtering to prevent internal routes from being sent to customers .

Communities, what else can they do?

- Blackhole routing, remotely (RTBH)
- Location indications to peering partners
- Trigger DDOS mitigation (scrubber) diversions
- Many more things!

Example community assignments

Communities used in examples (from AS11404):

11404:991 announce to customers
11404:992 announce to peers and customers
11404:993 announce to transit, peers and customers

11404:1000 All transit routes
11404:2000 All Peer routes

http://as11404.net has more of a list if you want a broader example. There are guides online for most major networks.

Filtering in action! (towards a customer)

Cisco example, showing basic portions of the BGP filtering configuration

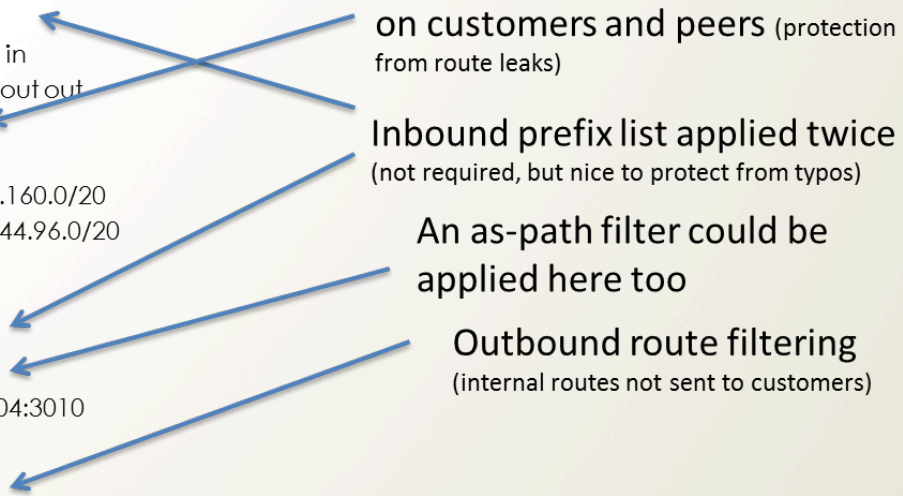
```
neighbor 192.0.2.2 remote-as 54858
neighbor 192.0.2.2 prefix-list as54858-in in
neighbor 192.0.2.2 route-map as54858-in in
neighbor 192.0.2.2 route-map full-tables-out out
neighbor 192.0.2.2 maximum-prefix 20

ip prefix-list as54858-in seq 5 permit 64.187.160.0/20
ip prefix-list as54858-in seq 10 permit 198.244.96.0/20

route-map as54858-in permit 500
 match ip address prefix-list as54858-in
 set local-preference 110
 set community 11404:993 11404:3000 11404:3010

route-map full-tables-out permit 1000
 match community full-tables-out

ip community-list standard full-tables-out permit 11404:993
ip community-list standard full-tables-out permit 11404:992
ip community-list standard full-tables-out permit 11404:991
ip community-list standard full-tables-out permit 11404:1000
ip community-list standard full-tables-out permit 11404:2000
```



Always place a max prefix limit on customers and peers (protection from route leaks)

Inbound prefix list applied twice (not required, but nice to protect from typos)

An as-path filter could be applied here too

Outbound route filtering (internal routes not sent to customers)

Example route-policy towards a customer

```
route-policy as54858--in
   if destination in as54858-in then
    set community customer-in
    set local-preference 115
  else
    drop
  endif
end-policy


route-policy full-tables-out
  if community matches-any full-tables-out then
    set med igp-cost
    pass
  else
    drop
  endif
end-policy
```

Notes: sets and matches communities from lists instead of directly like a route map.

Filtering in action! (towards a transit)

Cisco example, showing basic portions of the BGP filtering configuration

```
neighbor 207.8.14.109 remote-as 2828
neighbor 207.8.14.109 description XO Transit
 neighbor 207.8.14.109 route-map as2828-in in
 neighbor 207.8.14.109 route-map as2828-out out

route-map as2828-in permit 100
 set metric 0
 set local-preference 50
 set community 11404:1000 11404:1070 11404:1270 additive

route-map as2828-out permit 1000
 match community as2828-out
 set metric-type internal

ip community-list standard as2828-out permit 11404:993
ip community-list standard as2828-out permit 11404:9937
```

There is more configuration than this, this is just the community specific part

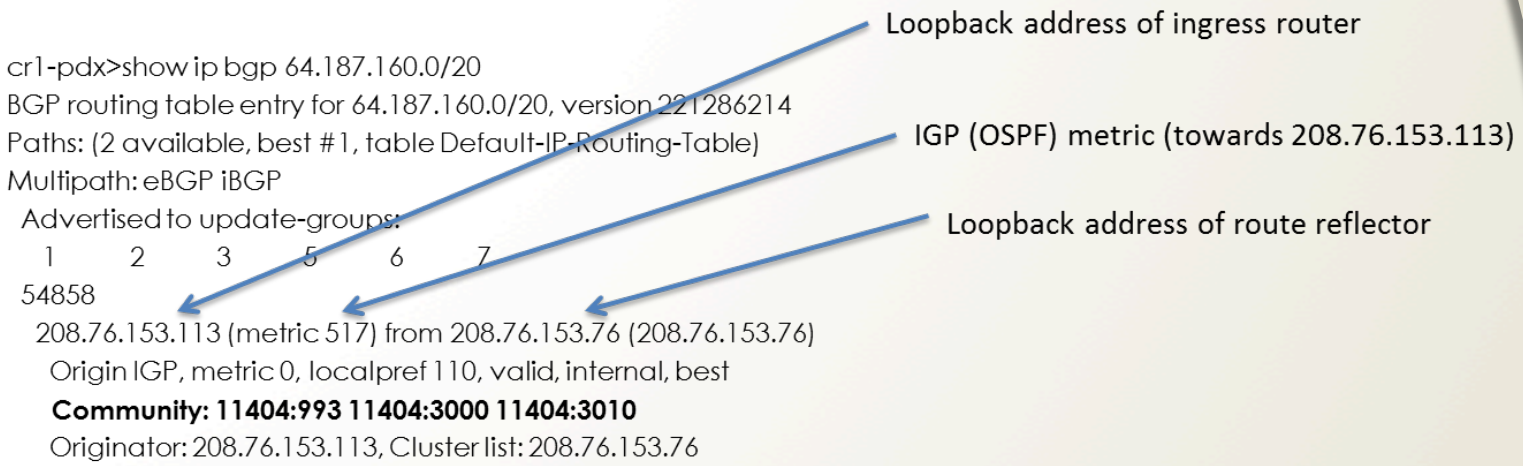
Ignore meds, force network to use nearest exit

Lower local-pref than default (we pay for this route)

Send MEDs based on IGP cost (make the carrier haul to nearest ingress point)

Outbound route filtering (match only routes tagged to announce to transit, validity of routes with this tag was assured at ingress)

Real world examples



- Higher than default localpref (110)
- Tagged as customer route (11404:3000) from Seattle (11404:3010)
- Tagged to announce to transit (11404:993)

Real world blackhole!

```
RP/0/RSP0/CPU0:cr2-pdx#show ip bgp 1.1.1.1
Tue Jun 14 15:58:12.729 UTC
BGP routing table entry for 1.1.1.1/32
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker             99390       99390
Last Modified: May 21 05:18:54.918 for 3w3d
Paths: (1 available, best #1)
  Advertised to update-groups (with more than one peer):
    0.9
  Path #1: Received by speaker 0
  Advertised to update-groups (with more than one peer):
    0.9
  65535
    192.0.2.0 (metric 673) from 208.76.153.3 (208.76.153.3)
      Origin IGP, metric 0, localpref 150, valid, internal, best, group-best
      Received Path ID 0, Local Path ID 1, version 99390
      Community: 11404:666 11404:800
RP/0/RSP0/CPU0:cr2-pdx#
```

!! Q&A !!

Questions?

More info?

Check a few of the relevant NANOG presentations:

Philip Smith NANOG 50:
http://www.nanog.org/meetings/nanog50/presentations/Sunday/NANOG50.Talk33.NANOG50-BGP-Techniques.pdf

Jason Schiller at NANOG 53:
http://www.nanog.org/meetings/nanog53/presentations/Sunday/bgp-101-NANOG53.pdf

Feel free to contact me:
John@vanoppen.com
206-437-1165