

# BGP made easy

**John van Oppen**

Wave / Spectrum

AS11404

# 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.

# What is an Autonomous System?

Typical traits:

- Usually a network of one or more routers redundantly interconnected.
- 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)

# Typical rationales for running BGP

- Multihoming / Provider redundancy
- Equipment / Port redundancy
- Peering (typically larger ASes)
- Connectivity quality (better paths)

# 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)

## Enforcing relationships across peers

- 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.
  - Best solution: add in community strings as tags, filter on ingress.

## 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

# Communities, tags for routes!

- Community strings allow routes to be tagged at ingress with to tell the rest of the AS what to do with them.
- Filtering only at ingress works for very small ASNs and very large ASNs.
- Allows for large ASes with lots of customer routes to scale by only filtering on customer sessions.
- 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).
- Allows easy filtering to prevent internal routes from being sent to customers .



# Examples of community assignments from AS11404

Communities used in examples:

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.

# 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
```

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)

```
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
```

An as-path filter could be applied here too

```
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
```

Outbound route filtering (internal routes not sent to customers)

```
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
```

# 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
```

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

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

Ignore med's, force network to use nearest exit

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

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

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

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

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

# Real world examples of communities and local pref (AS11404)

```
cr1-pdx>show ip bgp 64.187.160.0/20
BGP routing table entry for 64.187.160.0/20, version 221286214
Paths: (2 available, best #1, table Default-IP-Routing-Table)
Multipath: eBGP iBGP
```

Advertised to update-groups:

1	2	3	5	6	7
54858					
208.76.153.113 (metric 517) from 208.76.153.76 (208.76.153.76)					
Origin IGP, metric 0, localpref 110, valid, internal, best					
<b>Community: 11404:993 11404:3000 11404:3010</b>					
Originator: 208.76.153.113, Cluster list: 208.76.153.76					

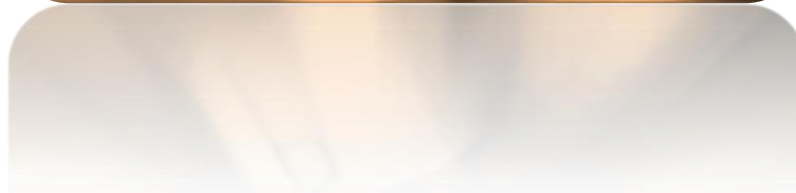
Loopback address of ingress router

IGP (OSPF) metric (towards 208.76.153.113)

Loopback address of route refelctor

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

# Q&A



# More info?

Check 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](mailto:John@vanoppen.com)

206-973-8302