Fable Of Contents

ISP TECH TALK

by Avi Freedman

ETHEL THE AARDVARK GOES BGP ROUTING

In this exciting column we'll actually walk through configuring a Cisco router for BGP. It's very important, however, that you look through June's column (which has 7 pages of background information on BGP) - and preferably May's column, which talks about multi-homing without BGP, before you think you're ready to configure a router to speak BGP.

A BASIC REVIEW

BGP4, or Border Gateway Protocol 4, is a routing protocol that is used by providers to announce routing information. Routes are promises to carry information (IP packets) to a given range of destination IP addresses. BGP4 as we're studying it, is spoken over peering sessions between routers in different networks, or Autonomous Systems. Each Autonomous System (AS) has a globally unique Autonomous System Number (ASN).

AGAIN, A WARNING

This is dangerous stuff. It's always best if you can test BGP configurations in a "lab" made up of a few Cisco 2501s before implementing them in a live network connected to the Internet, or, if you can, post your network topology and suggested

configuration to the inet-access mailing list and get feedback on it.

Making mistakes in BGP configuration can "blackhole" - or deny service - to remote parts of the Internet. It's very important that you understand basic IP routing, how to configure your router properly, and at least, the basics of BGP before you set out to configure your router.

Unfortunately, there's no good reference on using BGP to refer people to. Reading the RFCs (the Request For Comment documents that define the protocol at a low-to-mid-level), or even Cisco documentation (Cisco did not invent BGP, but Cisco's BGP implementation is definitely the most widely used) does not really tell you enough. Many of the "routing gurus" out there got started by looking at and working on running networks, where the architecture and implementation were already done. Most of the rest, however, started with the basics, and expanded their knowledge and experience as their networks grew.

Providers should aggressively filter their downstream BGP-speaking customers! The best way they can do this is to filter their announcements such that they will only hear certain specific routes from their customers. You may remember major network problems from late April that could have been avoided if any of a number of routers had strict filters installed.

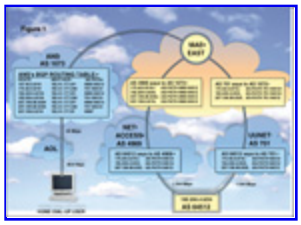
BEING "CONNECTED" TO THE INTERNET: YET ANOTHER REVIEW

Throughout this discussion it's critical to think about what it means to be "connected" to the Internet. To be connected to the Internet, for each host that is "on the Internet," you need to be able to:

- Send a packet out a path that will ultimately wind up at that host.
- That host has to have a path back to you. This means that whoever provides "Internet connectivity" to that host has to have a path to you - which, ultimately, means that they have to "hear a route" which covers the section of the IP space you're using, or you will not have connectivity to the

host in question.

Look at Figure 1. We'll explain more of the details below, but note the "Home Dial-up User." He's connected to AOL, which is served by ANS (AOL owns ANS). We're using 192.204.4.0/24 as an example.



In this example, the reason that an AOL dial-up user can send a packet to 192.204.4.0/24 (for example) is that the ISP (AS 64512) advertised that route to the two upstream providers (AS 4969 and AS 701), who in turn advertised that route to AS 1673 (ANS, which provides IP service for AOL).

Every IP address that you can get to on the Internet is reachable because someone, somewhere, has advertised a route that "covers" it. Similarly, if there is not a generally advertised route to cover an IP address, then no one on the Internet will be able to reach it.

AS-PATHS

Every time a route is advertised via BGP, it is "stamped" with the ASN of the router doing the advertising. As a route moves from Autonomous System to Autonomous System (network to network), it builds up an "AS-PATH." Each route starts out with a "null AS-PATH," represented by the regular expression ^$. See Figure 1- the blocks that show the routes as they move from hop to hop show you the AS-PATH accumulating as the route moves from network to network.

HOW BGP PEERING SESSIONS WORK

When a peering session is established, each router sends all of its BGP routing information to the other router - unless "filters" are installed to restrict the information that gets passed. Then, once the initial routing information has been sent, "updates" (new routes being advertised and old, advertised routes, being withdrawn) flow back and forth until the session dies.

AS-PATH FILTERS

We'll shortly give a complete explanation of "AS-PATH filters." For now, we'll just go over the basics and give you the three fundamental AS-PATH filters for basic BGP.

The purpose of AS-PATH filters is to whack out huge chunks of routing information - so that you only send exactly the routes that you want to send.

THE FUNDAMENTAL AS-PATH FILTERS

First, the filter that "permits everything."

```
ip as-path access-list 1 permit .*


Second, the filter that "denies everything."


ip as-path access-list 2 deny .*
```

Third, the filter that "permits only OUR routes."

```
ip as-path access-list 3 permit ^$
ip as-path access-list 3 deny .*
```

A few comments:

- Don't play around with filters until you know more.
- All filters have an implicit deny .* at the end, but it doesn't hurt to put one in for safety.
- .* means "match any route."
- ^$ means "match every route with a NULL AS-PATH." The only routes with NULL (or "no") AS- PATHs are routes that are locally-generated.

BGP METRICS (ATTRIBUTES) AND ROUTE SELECTION: INTRODUCTION

Next month we'll go into BGP metrics and attributes, which are parameters associated with BGP routes that allow you to select and change the selection of "the best BGP route" for a certain destination.

For now, keep in mind that unless you do any tuning on your own:

- The most specific route always wins. Whether it's a BGP route or a static internal route, the most specific route always wins.
- If you have to choose between multiple BGP routes, the one with the shortest AS_PATH wins.

If you're multi-homed, then BGP will pick the route with the shortest AS_PATH if both providers offer you what is otherwise the same route to a given destination.

Once BGP picks the best route, it is then eligible to be installed in the IP Routing Table, which is the table that the router consults when making the actual packet-forwarding decisions.

WHAT TO KEEP IN MIND WHEN CONFIGURING BGP

When you're bringing up a new BGP session, or considering how to do BGP in general, the things to keep in mind for each peer are:

- What routes do you want them to hear? The most important thing is to ensure that you do not redistribute routes to which you are not providing "Internet connectivity."
- What do you want to do with the routes that you hear via the session? Do you want to "tune them"? Only take some? Take them all? Do you have the memory and CPU in your router to really do what you want?

For the example in this column we'll explicitly advertise only a few routes, and use AS-PATH filters to deny advertisement of any other BGP routes we may have heard.

For now we'll either deny all incoming BGP routes and use load-balanced default routes, or we'll take the incoming BGP routes, if the BGP-speaking router is capable of it. The latter requires a Cisco with at least 64 MB of RAM.

MULTI-HOMING AND LOAD-BALANCING

Generally, the goal of multi-homing is to use both connections in a sane manner and "load-balance" them somehow. Ideally, you'd like roughly half the traffic to go in and out of each connection. You'd also like "fail-over" routing, where if one connection goes down the other one keeps you connected to the Internet. In an ideal network, you'd be able to have any one of your connections to the Net go down and still maintain connectivity and speed.

We'll talk in the next few months about how you load-balance incoming and outgoing traffic to and from your network. Incoming traffic is controlled by how you announce your routes to the world (packets will flow into your network because someone heard of, and is using, a route announcement). Outgoing traffic is controlled by the routes that you allow to flow into your border router(s) - and is thus much easier to control and tune.

HOW TO ANNOUNCE YOUR NETWORKS: THE KEY TO BGP CONFIGURATION

Once you've decided what you want to do with BGP, it's time to translate those decisions into a router configuration.

The safest way to announce your routes with BGP is to configure everything statically. You can think of the process described below as turning internal routing statements into route announcements. To do this:

Identify every route that you "own" (or are "allowed to" announce).

- Add a static route for it to the Interface Loopback0 with a weight higher than any other static route for that network. Higher numbers for static route weights mean that the routes are less preferred.
- Configure a router BGP clause like the one below, with static network statements to announce your routes, and "sanity filters" in place to make sure you only announce your routes and only take the routes you want.

For example, let's say you're routing the following networks (also called "netblocks" or "prefixes"):

```
170.40.0.0/16 (a /16 has a netmask of 255.255.0.0)
192.204.4.0/24 (a /24 has a netmask of 255.255.255.0)
207.106.96.0/20 (a /22 has a netmask of 255.255.252.0)
```

You'd first configure your router with:

```
int Loopback0
```

descr Loopback interface for routes to be nailed to. ip route 170.40.0.0 255.255.0.0 Loopback0 10 ip route 192.204.4.0 255.255.255.0 Loopback0 10 ip route 207.106.96.0 255.255.252.0 Loopback0 10

Then, put in your "as-path access-list filters".

```
ip as-path access-list 1 permit .*
ip as-path access-list 2 deny .*
ip as-path access-list 3 permit ^$
ip as-path access-list 3 deny .*
```

Then put in "router BGP" clause.

```
router bgp 64512

 network 170.40.0.0 mask 255.255.0.0
 network 192.204.4.0 mask 255.255.255.0
 network 207.106.96.0 mask 255.255.252.0
 neighbor  remote-as
 neighbor  next-hop-self
 neighbor  filter-list 3 out
 neighbor  filter-list 2 in
```

WHAT THIS DOES: ANTI-FLAPPING MEASURES

One of our goals is to prevent the route advertisements from "flapping" if parts of your network die temporarily. If you are the upstream provider for anyone who's multi-homed, you shouldn't statically announce any routes for them unless you really understand what you're doing. Anyway, to prevent the route advertisements from flapping, we put in backup routes to the Loopback0 pseudo-interface.

This method "statically nails down" the advertised BGP route announcements with the network statements. To nail them down, there must be: (1) underlying static routes with the same netmask as each route being advertised with a network statement; and (2) those underlying static routes must not go

away.

The purpose of the Loopback0 routes is to ensure that even if an existing primary route which matches the netmask of the route being announced (and this is often not the case) goes away, the Loopback0 route (with a weight of 10, which means it's only a backup route to any route without a weight at the end) will kick in and keep the BGP route advertisement stable. Loopback0 routes always stay installed since there's no physical interface to go down and cause the route to be withdrawn - the interface Loopback0 will always be up, so the routes pointed to them will always be installed. NOTE: If you are already using Loopback0, then pick another Interface (Loopback1, Loopback2, etc...)

WHAT THIS DOES: FILTERS

This example uses a "send only our local routes" outbound filter, so it won't accidentally re-advertise one of your upstream provider's routes to the other.

Here we also use a "deny everything" incoming filter, which will only announce routes and not accept any. If you want to accept all incoming routes, replace the filter-list 2 in with filter-list 1 in. Actually, you could just not specify an inbound as-path filter, and the effect would be the same, but it's better by far to be explicit about these things.

ADDING MORE PEERS

To add more peers, just create another similar neighbor statement. Cisco routers give you 30 seconds to finish typing the neighbor statement before trying to establish the session. It is critical that you get those neighbor somebody filter-li st xxx .. statements entered by then. The best way, by far, to do it is to either cut and paste or tftp in a complete neighbor statement to the router.

THE COMPLETED EXAMPLE

Here's an example of a completely filled-in BGP clause, based on Figure 1.

```
router bgp 64512
```

```
network 170.40.0.0 mask 255.255.0.0
network 192.204.4.0 mask 255.255.255.0
network 207.106.96.0 mask 255.255.252.0
neighbor 207.106.127.45 remote-as 4969
neighbor 207.106.127.45 next-hop-self
neighbor 207.106.127.45 filter-list 3 out
neighbor 207.106.127.45 filter-list 2 in
neighbor 137.10.10.121 remote-as 701
neighbor 137.10.10.121 next-hop-self
neighbor 137.10.10.121 filter-list 3 out
neighbor 137.10.10.121 filter-list 2 in
```

This says:

- Announce the networks 170.40.0.0/16, 192.204.4.0/24, and 207.126.0.0/18.
- Talk to Net Access (207.106.127.45) and give them only our routes (filter-list 3 out) and take no BGP routes in (filter-list 2 in).
- Talk to UUNET (137.10.10.121) and give them only our routes (filter-list 3 out) and take no BGP routes in (filter-list 2 in).

Please, even though it isn't required at all times, put inbound and outbound filters, of some sort, on every BGP neighbor session.

CONTROLLING OUTGOING DATA FLOW: "FULL ROUTING" AND OTHER OPTIONS

Next month we'll go into detail and give you examples of many different ways to use the routes you can hear via BGP to tune your outbound data flow.

Briefly, option one is "take everything." With a big enough router, you can take multiple views of the full routing table, and this should give you a somewhat better quality of Internet connectivity than just load- balancing default routes. For each route where there are multiple views, your router will select the best one to use at any time, which is based on AS_PATH length, unless you tune other parameters.

Option two is "take customer routes from each provider." Who can get to SprintLink customers better than SprintLink? No one, if SprintLink's built its network properly. You ask each provider to only send you routes for its customers. If your two

providers are not SprintLink and MCI, then you should be able to store those routes and use them even on a Cisco 2501. These routes are also called "peering routes" because the "routing load" that providers who have no customer-provider relationship (i.e. MCI to Sprint, UUNET to ANS) send to each other via BGP.

A SAMPLE ROUTER CONFIG

```
!
service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname jacks-router
!
enable secret 5 $1$h7jsdf$k23jMhJ.u5jads0.otE.
enable password 7 145C1B020D1726
!
interface Ethernet0

      ip address 207.106.96.0 255.255.255.0

!
interface Serial0
      description T1 to Net Access
              ip address 207.106.127.46 255.255.255.252
              encapsulation ppp

!
interface Serial1

      description T1 to UUNET
              ip address 137.10.10.122 255.255.255.252

!
router bgp 64512
network 170.40.0.0 mask 255.255.0.0
network 192.204.4.0 mask 255.255.255.0
network 207.106.96.0 mask 255.255.252.0
neighbor 207.106.127.45 remote-as 4969
neighbor 207.106.127.45 next-hop-self
neighbor 207.106.127.45 filter-list 3 out
neighbor 207.106.127.45 filter-list 2 in
neighbor 137.10.10.121 remote-as 701
neighbor 137.10.10.121 next-hop-self
neighbor 137.10.10.121 filter-list 3 out
neighbor 137.10.10.121 filter-list 2 in
!
ip name-server 207.8.186.1
```

```
ip name-server137.39.1.3
!
ip subnet-zero
ip classless
!
ip route 0.0.0.0 0.0.0.0 207.106.127.46
ip route 0.0.0.0 0.0.0.0 Serial1
ip route 170.40.0.0 255.255.0.0 207.106.96.10
ip route 170.40.0.0 255.255.0.0 Null0 10
ip route 192.204.4.0 255.255.255.0 207.106.96.10
ip route 192.204.4.0 255.255.255.0 Null0 10
ip route 207.106.96.0 255.255.252.0 Null0 10
ip route 207.106.96.128 255.255.255.192 207.106.96.7
ip route 207.106.97.0 255.255.255.0 207.106.96.11
ip route 207.106.98.0 255.255.254.0 207.106.96.11
!
ip as-path access-list 1 permit .*
ip as-path access-list 2 deny .*
ip as-path access-list 3 permit ^$
ip as-path access-list 3 deny .*
!
line vty 0 4

      password 7 0AB41A0C034907
      exec-timeout 0 0

!
```

Colorado Offices
13949 W Colfax Ave Suite 250, Golden, CO 80401
Voice: 303-235-9510; Fax: 303-235-9502

Fable Of Contents