



PLANETLAB

BGP Feed Configuration Memo

Akihiro Nakao
Princeton University

Larry Peterson
Princeton University

PDN-03-011
April 2003

Status: Ongoing Draft.

BGP Feed Configuration Memo

Akihiro Nakao and Larry Peterson
{nakao, llp}@cs.princeton.edu

Department of Computer Science
Princeton University

February 19, 2003

Overview

This memo describes how to configure a BGP router (R) to peer with a local PlanetLab node (P). The peering relationship is set up in a way that allows P to receive BGP updates from R , *without* R ever accepting updates from P .

Our goal is to build tools and services on PlanetLab in which P is able to answer queries about local Autonomous System (AS) paths. For example, a set of PlanetLab nodes that have accessed a local BGP feed could collectively construct a peering graph similar to that supported by the RouteViews project [1]. Note that this memo describes how to configure R , not P .

We illustrate two ways to configure R , the first based on the configuration used at Princeton, and the second based on the configuration used at Duke. In both cases, simply replace the IP addresses of R and P with those of your router and local PlanetLab node.

Note that although both Princeton and Duke use Cisco routers, since the command line interface for BGP is similar on most routers, this configuration should guide you in configuring your router. For more information about the commands used in these examples, as well as other commands, refer to the Cisco BGP command reference, [2] and regular expression reference [3, 4].

Scenario 1

Figure 1 shows a fragment of the actual configuration for the Princeton (AS 88) router R (128.112.128.114), specifying that it should feed BGP updates to the local node P (128.112.152.173). Princeton is connected to the Internet via three ISPs with AS numbers 1785, 7018, and 10466.

line 5: The BGP router id should be set to R 's IP address. This should already be in the configuration file, but we include this line for the sake of completeness.

line 7: Define P to be one of R 's peers, specifying P 's IP address and the local AS number.

line 9: This may not be strictly necessary but allows soft-reset.

line 11, 30, 31: Use the “`distribute-list in`” command to deny any updates from P . Although we will be

running code on P that *never* sends updates to R , you should make sure that R rejects any updates from P just to be safe.

line 13, 34, 35: Use the “`distribute-list out`” command to cause R to send updates to P .

line 15: Set up a filter for outgoing routes from R to P . We use access-list number 70, which is defined in the lines 21-27.

line 21-27: This AS access-list determines which AS paths will be delivered to P . It allows updates for only those AS paths that start with the ISPs' AS numbers (1785, 7018, 10466) to be sent. This means updates related to other private routes are not divulged to P .

Scenario 2

Figure 2 shows a fragment of the actual configuration for router R (128.109.210.45) at Duke (AS 81), specifying that it should feed BGP updates to the local node P (152.3.136.1).

line 8-9: The BGP router id should be set to R 's IP address, and the AS number should be defined. This should already be in the configuration file, but we include this line for the sake of completeness.

line 13-14: Define P to be one of R 's peers, specifying P 's IP address and the AS number. Configure P as a *route reflector client* of R .

line 15: This may not be strictly necessary but allows soft-reset.

line 19-22: Set up the inbound filter (`route-map`) “`planetlab-in`” for denying any routes from R to P with *priority* 10 (lines 19-20). The filter “`planetlab-in`” accepts a route if it matches IP access-list number 20 (line 21). The access-list number 20 rejects any IP addresses (line 22). As a result, router R rejects all the routes from P . Although we will be running code on P that *never* sends updates to R , you should make sure that R rejects any updates from P just to be safe.

Other Scenarios

In both the Princeton and Duke scenarios, both R and P are in the same AS and are configured to be iBGP peers of each other. Additional configuration scenarios are possible, but some do not meet our needs.

For example, P could be assigned a private AS number (different from R 's) with the two machines configured as eBGP peers of each other. This would allow P to see the routes that R knows about, but all the routes will specify R as the next hop, not the actual next hop that R uses. Configuring R and P to be iBGP peers in the same AS (as in Scenarios 1 and 2) allows P to see the same next hop as R .

As another example, should the AS have multiple BGP speakers, each connected to a different set of external ASes, P would have to peer with all of them as an iBGP peer in order to see the union of all the routes the set of speakers know about. This is because iBGP peers do not forward routes learned from the other iBGP peers.

References

- [1] Route Views Project. <http://antc.uoregon.edu/route-views/>.
- [2] Cisco Systems, Inc. BGP Command, 2002. http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_r/iprprt2/lrdbgp.htm.
- [3] Cisco Systems, Inc. Regular Expressions, 2002. <http://www.cisco.com/warp/public/459/26.html>.
- [4] Cisco Systems, Inc. Using Regular Expressions, 2002. http://www.cisco.com/univercd/cc/td/doc/product/atm/c8540/12_0/13_19/cmd_ref/appc.htm.

```

1:  ! This BGP configuration will send all of vgate1 BGP routes that have an
2:  ! AS path that originate from one of our ISPs (ATT-I1 FN-I1 Abilene-I2)
3:  ! to peer at Princeton CS 128.112.152.173.
4:  !
5:  bgp router-id 128.112.128.114
6:  !
7:  neighbor 128.112.152.173 remote-as 88
8:  ! Allow soft reset of connection if they support it.
9:  neighbor 128.112.152.173 soft-reconfiguration inbound
10: ! Control what BGP routes are received.
11: neighbor 128.112.152.173 distribute-list bgp-filter-Feed2CS-in in
12: ! Control what BGP routes are sent.
13: neighbor 128.112.152.173 distribute-list bgp-filter-Feed2CS-out out
14: ! Control what BGP AS origins are sent.
15: neighbor 128.112.152.173 filter-list 70 out
16: !
17: ! This is AS access list that determines which AS paths
18: ! will be sent to "Feed2CS" peer by only allowing AS paths
19: ! that start with the ISPs AS number to be sent.
20: !
21: no ip as-path access-list 70
22: ! Send FastNet I-1 BGP routes.
23: ip as-path access-list 70 permit _1785
24: ! Send Abilene I-2 BGP routes.
25: ip as-path access-list 70 permit _10466
26: ! Send ATT I-1 BGP routes.
27: ip as-path access-list 70 permit _7018
28: !
29: !
30: no ip access-list standard bgp-filter-Feed2CS-in
31: ip access-list standard bgp-filter-Feed2CS-in deny any
32: !
33: !
34: no ip access-list standard bgp-filter-Feed2CS-out
35: ip access-list standard bgp-filter-Feed2CS-out permit any
36: ! end

```

Figure 1: Example Configuration at Princeton (AS 88)

```
1:  ! This BGP configuration will send all the routes that
2:  ! the BGP speaker (R) (IP=128.109.210.45; AS 81) observes
3:  ! to PlanetLab node (P) (IP=152.3.136.1) at Duke.
4:  ! Note that we configure (P) as route-reflector-client of (R).
5:  !
6:  ! We are configuring BGP router (R) (IP=128.109.210.45; AS 81)
7:  !
8:  router bgp 81
9:  bgp router-id 128.109.210.45
10: !
11: ! Configure the peer (P) as route reflector client in AS 81
12: !
13: neighbor 152.3.136.1 remote-as 81
14: neighbor 152.3.136.1 route-reflector-client
15: neighbor 152.3.136.1 soft-reconfiguration inbound
16: !
17: ! Ensure that we will never accept routes from (P)
18: !
19: neighbor 152.3.136.1 route-map planetlab-in in
20: route-map planetlab-in permit 10
21: match ip address 20
22: access-list 20 deny any
23: !
24: ! End
25: !
```

Figure 2: Example Configuration at Duke (AS 81)