

Getting Familiar with the C-BGP Simulator

Anuj Sehgal, Jürgen Schönwälder
School of Engineering and Science
Jacobs University Bremen, Germany
{s.anuj, j.schoenwaelder}@jacobs-university.de

September 27, 2010

Abstract

The C-BGP simulator is a solver for the BGP protocol, which is a standard for exchanging routing information across domains on the Internet. C-BGP supports the complete BGP decision process, versatile import and export filters, route-reflection, and experimental attributes such as redistribution communities. The simulator uses a Cisco-like command line interface to implement the network structure. This tutorial is aimed at getting you familiar with installing and using the C-BGP simulator.

1 Introduction

C-BGP is aimed at computing the outcome of the BGP decision process in networks composed of several routers. It can be used as a research tool to experiment with modified decision processes and additional BGP route attributes. It can also be used by the operator of an ISP network to evaluate the impact of logical and topological changes on the routing tables computed in its routers.

This document will guide you through the installation process and introduce you to some features of the C-BGP simulator, which should help you get started with using its more complex features.

2 Installation

This document assumes that you are working on a recent Ubuntu release, however, the setup process should be similar on any other Linux platform as well.

2.1 Setting up the environment

Before the C-BGP simulator can be successfully compiled, you will need to install some libraries that it depends upon. On an Ubuntu system you may install most of these using `apt-get`, as shown below:

```
sudo apt-get install libpcre3 libpcre3-dev libbz2-dev
sudo apt-get install libreadline5 libreadline5-dev
```

In case you are on any other system, please install the following libraries before proceeding forward:

```
libpcre-dev
libreadline-dev
libbz2-dev
```

C-BGP also depends upon the `libgds` library. To setup this library proceed to issue the following commands in the terminal:

```
wget http://libgds.info.ucl.ac.be/downloads/libgds-1.4.5.tar.gz
tar xvfz libgds-1.4.5.tar.gz
cd libgds-1.4.5/
./configure
make
sudo make install
sudo ldconfig
```

Your environment is now setup with all dependencies to build C-BGP.

2.2 Compiling from source

Once the environment is setup with all the dependencies, you may now download C-BGP and compile it as shown below:

```
wget http://cbgp.info.ucl.ac.be/downloads/cbcp-1.4.4.tar.gz
tar xvfz cbcp-1.4.4.tar.gz
cd cbcp-1.4.4/
./configure --with-readline=/usr/lib/
make
sudo make install
sudo ldconfig
```

C-BGP is now installed on your system and can be started with the command `cbgp -i`.

3 Using C-BGP

In order to start C-BGP in interactive mode, so that you may manually enter commands or load scripts for simulation, start it with the command `cbgp -i`. Once your simulator starts you should see the following:

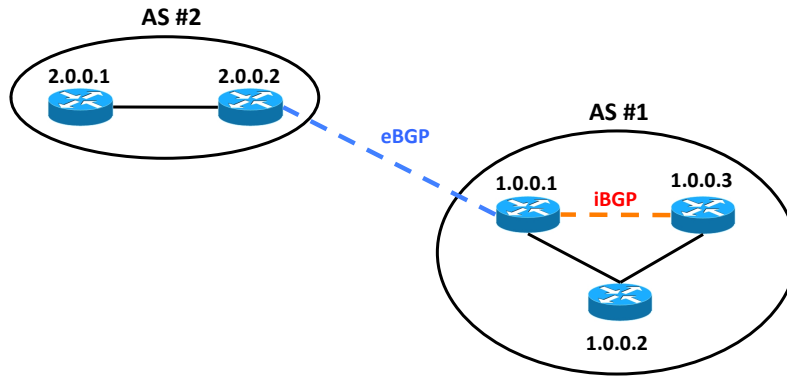


Figure 1: Sample Network with two Autonomous Systems

C-BGP routing solver 1.4.4
 Copyright (C) 2007 Bruno Quoitin
 IP Networking Lab, CSE Dept, UCL, Belgium

C-BGP comes with ABSOLUTELY NO WARRANTY.
 This is free software, and you are welcome to redistribute it
 under certain conditions; see file COPYING for details.

```
cbgp> init.  

help is bound to '?' key  

cbgp>
```

To simulate the simple network shown in Figure 1 simply enter the following commands in the cbgp simulator:

```
cbgp> net add node 1.0.0.1  

cbgp> net add node 1.0.0.2  

cbgp> net add node 1.0.0.3  

cbgp> net add node 2.0.0.1  

cbgp> net add node 2.0.0.2
```

The above commands create the actual nodes. Once the nodes are created they must be added to their own domains, i.e., the autonomous systems, using the following commands:

```
cbgp> net add domain 1 igp  

cbgp> net add domain 2 igp  

cbgp> net node 1.0.0.1 domain 1  

cbgp> net node 1.0.0.2 domain 1  

cbgp> net node 1.0.0.3 domain 1  

cbgp> net node 2.0.0.1 domain 2  

cbgp> net node 2.0.0.2 domain 2
```

The links between the nodes must also be setup, the delays associated with them and igp weights assigned to them:

```
cbgp> net add link 1.0.0.1 1.0.0.2 0
cbgp> net link 1.0.0.1 1.0.0.2 igp-weight --bidir 10
cbgp> net add link 1.0.0.2 1.0.0.3 0
cbgp> net link 1.0.0.2 1.0.0.3 igp-weight --bidir 10
cbgp> net add link 2.0.0.1 2.0.0.2 0
cbgp> net link 2.0.0.1 2.0.0.2 igp-weight --bidir 10
```

We must instruct C-BGP to compute all the intra-domain node relationships by using the following:

```
cbgp> net domain 1 compute
cbgp> net domain 2 compute
```

The inter-domain links and traffic routes must now also be created:

```
cbgp> net add link 1.0.0.1 2.0.0.2 0
cbgp> net node 1.0.0.1 route add 2.0.0.2/32 2.0.0.2 2.0.0.2/32 0
cbgp> net node 2.0.0.2 route add 1.0.0.1/32 1.0.0.1 1.0.0.1/32 0
```

The last few steps are to enable BGP routing within the domains:

```
cbgp> bgp add router 1 1.0.0.1
cbgp> bgp add router 1 1.0.0.2
cbgp> bgp add router 1 1.0.0.3
cbgp> bgp domain 1 full-mesh
cbgp> bgp router 1.0.0.1 add peer 2 2.0.0.2
cbgp> bgp router 1.0.0.1 peer 2.0.0.2 next-hop-self
cbgp> bgp router 1.0.0.1 peer 2.0.0.2 up
cbgp> bgp router 1.0.0.1 add network 1/8
cbgp> bgp router 1.0.0.2 add network 1/8
cbgp> bgp router 1.0.0.3 add network 1/8
cbgp> bgp add router 2 2.0.0.1
cbgp> bgp add router 2 2.0.0.2
cbgp> bgp domain 2 full-mesh
cbgp> bgp router 2.0.0.2 add peer 1 1.0.0.1
cbgp> bgp router 2.0.0.2 peer 1.0.0.1 next-hop-self
cbgp> bgp router 2.0.0.2 peer 1.0.0.1 up
cbgp> bgp router 2.0.0.1 add network 2/8
cbgp> bgp router 2.0.0.2 add network 2/8
```

Lastly, we must run the simulation:

```
cbgp> sim run
```

The above command does not provide any output, but sets up everything in the simulator so that we may run tests on the network. Three very useful commands we have at our disposal are `record-route`, `show rt` and the common `traceroute` and `ping`. To show how these commands work, let us first select a particular node in the network to examine. For this purpose, let us use node 1.0.0.1. To select node 1.0.0.1 and test the `record-route` command on it, issue the following sequence of commands after the setup script/commands have been provided to C-BGP:

```
cbgp> net node 1.0.0.1
cbgp-node> record-route 1.0.0.3
1.0.0.1 1.0.0.3 SUCCESS 3      1.0.0.1 1.0.0.2 1.0.0.3
cbgp-node>
```

As you can see, `record-route` basically checks if the desirable node, in this case 1.0.0.3, is reachable from our current node, 1.0.0.3, and what path is taken to it. In our example, packets travel from 1.0.0.1 to 1.0.0.3 successfully via 1.0.0.2. Inter-domain routes can also be checked using this command, as is visible from below:

```
cbgp> net node 2.0.0.1
cbgp-node> record-route 1.0.0.3
2.0.0.1 1.0.0.3 SUCCESS 5 2.0.0.1 2.0.0.2 1.0.0.1 1.0.0.2 1.0.0.3
cbgp-node>
```

The `show rt` command is especially useful in checking if all routes are correctly setup and if the BGP routing is also working or not. To use this command, we must first select a node; the command may then be used as:

```
cbgp> net node 1.0.0.3
cbgp-node> show rt *
1.0.0.1/32      0.0.0.0 1.0.0.2 20      IGP
1.0.0.2/32      0.0.0.0 1.0.0.2 10      IGP
2.0.0.0/8       0.0.0.0 1.0.0.2 0       BGP
cbgp-node>
```

The `show rt *` command shows us the routing table for all prefixes and also outputs the method that was used to setup the particular routes. This becomes even more clear if we check the routing table of node 1.0.0.1:

```
cbgp> net node 1.0.0.1
cbgp-node> show rt *
1.0.0.2/32 0.0.0.0 1.0.0.2 10 IGP
1.0.0.3/32 0.0.0.0 1.0.0.2 20 IGP
2.0.0.2/32 2.0.0.2 2.0.0.2 0 STATIC
2.0.0.0/8  2.0.0.2 2.0.0.2 0 BGP
cbgp-node>
```

In this case, the static route between 1.0.0.1 and 2.0.0.2 that we defined is also visible, just as the route to the 2.0.0.0/8 network learnt from BGP is also visible besides the internal IGP routes. The last common network command set of `traceroute` and `ping` can be used just as in the real world, however, a source node must first be selected using the `net node <address>` command.

4 Conclusion

This tutorial has shown you how to setup C-BGP on your machines. We also covered setting up simple networks with multiple autonomous systems, as well as intra- and inter-domain routing using BGP. While this tutorial provides you a good starting point, you are encouraged to experiment with it on your own and also read through the manual provided on the C-BGP website.