Features

- Ready to run VirtualBox image (see Download section)
- XML based configuration file
- Cisco like CLI with Tab completion, reverse search and history
- Short and detailed help
- Show configuration
- Set and show debugging options
- Start and stop configured BGP peers
- Establish connections to configured BGP peers
- Overview of configured BGP peers
- Detailed view of configured BGP peers (states, injected routes, etc)
- Show received routes (overview or detailed)
- Show sent routes (overview or detailed)
- Create routes with different BGP attributes
- Show created routes (overview or detailed)
- Remove routes
- Withdraw routes
- Generate random and pseudo-random routes
- Remove all generated routes
- Inject routes on one or more peers
- Simulate peer flapping
- Simulate route flapping
- Remove peer flapping
- Remove route flapping
- Simple test cases
- Tested on Linux, FreeBSD and MacOS X
- …

Edit

TODO list

- Installation script
- Add documentation
- Fix bugs
- Implement new features
- Cleanup code
- Add support for different AFI/SAFIs
- …

Edit

Bugs

- Tests seem to fail sometimes
- A lot of more…

Edit

Installation

Edit

Requirements

Please install the following Perl modules:

- Perl with thread support
- Net::BGP + Apply patch
- Term::ShellUI + Apply patch (only for Term::ShellUI versions < 0.91)
- Term::ReadLine::Perl or Term::ReadLine::GNU
- XML::Simple
- IO::Interface::Simple
- Data::Dumper

Edit

Installation

Download and unpack the current source code (see Download section):

```
tar xfvz inject-[version].tar.gz
```

```
cd inject-[version]/
```

```
cp -r Inject/ /usr/share/perl/<perl_version>
```

```
cp inject.pl /usr/local/bin/
```

```
chmod u+x /usr/local/bin/inject.pl
```

```
cd /usr/share/perl/<perl_version>/Net/BGP/
```

```
patch -p0 < ~/inject/patch/patch-net-bgp-update.diff
```

Only for Term::ShellUI versions < 0.91:

```
cd /usr/share/perl/<perl_version>/Term/
```

```
patch -p0 < ~/inject/patch/patch-term-shellui.diff
```

Create configuration directory and copy and edit the example configuration file:

Edit the configuration file and add at least one peer.

```
mkdir ~/.inject/
```

```
cp cfg/inject.rc ~/.inject/
```

Start program and have fun:

The program can only be started as root.

```
/usr/local/bin/inject.pl
```

Edit

Examples

Edit

Inject routes

Inject route 2.0.0.0/24 with nexthop set to 4.4.4.4 and a MED of 300 on Peer1:

```
Inject> route net 1 2.0.0.0/24
Apr 27 22:01:01: INFO: Route network attribute for RID 1 set
```

```
Apr 27 22:01:01: INFO: Route network attribute for RID 1 set.

Inject> route nexthop 1 4.4.4.4
Apr 27 22:01:27: INFO: Route nexthop attribute for RID 1 set.

Inject> route med 1 300
Apr 27 22:01:56: INFO: Route MED attribute for RID 1 set.

Inject> inject Peer1 1
Apr 27 22:02:13: INFO: Injecting the following route:

RID       : 1
Inject to: Peer1
Network  : 2.0.0.0/24
NextHop  : 4.4.4.4
ASPath   : 65123
MED      : 300

Apr 27 22:02:13: Injecting RID 1 on Peer1.
```

Edit

Generate routes

Generate 1000 random routes with nexthop 10.8.4.1, localpref 120 and MED 10 or 50 or 100:

```
Inject> generate routes Peer1 1000 nexthop(10.8.4.1) localpref(120) med(10|50|100)
Apr 27 22:13:24: INFO: Generating 1000 routes. One dot for each 1000 routes.
.
Inject>
Apr 27 22:13:24: INFO: Injecting RID __79 on Peer1
Apr 27 22:13:24: INFO: Injecting RID __13 on Peer1
etc.
```

Edit

Generate route with long AS path

Generate route 20.0.0.0/24 with a long AS path and inject it on peer Peer1:

```
Inject> route net 1 20.0.0.0/24
May 1 13:47:14: INFO: Route network attribute for RID 1 set.

Inject> route nexthop 1 10.0.0.1
May 1 13:47:50: INFO: Route nexthop attribute for RID 1 set.

Inject> route aspath 1 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
May 1 13:49:59: INFO: Route AS path attribute for RID 1 set.

Inject> inject Peer1 1
```

Edit

Show routes

```
Inject> show routes all

Prefix            NextHop      LPref  MED Peer              PeerID
```

```
91.185.246.0/24  193.47.73.1  0      0   via 193.47.73.1 / Peer1
88.81.192.0/24   193.47.73.1  0      0   via 193.47.73.1 / Peer1
195.66.126.0/24  193.47.73.1  0      0   via 193.47.73.1 / Peer1
```

Edit

Command overview

Command	Description
help (<arg1> (<arg2>))	Show help
help debug (<arg>)	Show “debug” help
help flap (<arg>)	Show “flap” help
help unflap (<arg>)	Show “unflap” help
help generate (<arg>)	Show “generate” help
help inject	Show “inject” help
help peer (<arg>)	Show “peer” help
help route (<arg>)	Show “route” help
help show (<arg>)	Show “show” help
help withdraw (<arg>)	Show “withdraw” help
help help	Show “help” help
help history	Show “history” help
help exit	Show “exit” help
debug all (<off>)	Activates all possible debugging options
debug error (<off>)	Activates error debugging
debug flap (<off>)	Activates flap debugging
debug inject (<off>)	Activates route inject debugging
debug keepalives (<off>)	Activates debugging of keepalive packets
debug notify (<off>)	Activates debugging of notification packets
debug open (<off>)	Activates debugging of peer openings
debug refresh (<off>)	Activates debugging of refresh packets
debug reset (<off>)	Activates debugging of session resets
debug update (<detail|off>)	Activates (detailed) debugging of update packets
debug withdraw (<off>)	Activates debugging of route withdrawns
exit	Stops all BGP sessions and exits the program
flap peer <peerid|all> <up_s> <down_s>	Flapps peer, up_s is the number of seconds a peer should stay up, down_s is the number of seconds a peer should stay down
generate remove	Removes all generated routes from all peers
generate routes <peerid|all> <num> (<args>)	Generate and inject a number <num> of routes (see “help generate routes”)
inject <peerid|all> <rid>	Inject route with specified RID on the specified peer
peer start <peerid|all>	Start peer

peer stop <peerid|all>	Stop peer
route aggregator <rid> <asn> <aggregator ip>	Set route aggregator
route aspath <rid> <as1>…<asN>	Set AS path
route atomic <rid> <0|1>	Set ATOMIC_AGGREGATE
route community <rid> <c1>…<cN>	Set community
route localpref <rid> <localpref>	Set localpref
route med <rid> <med>	Set MED
route net <rid> <network>	Set prefix
route nexthop <rid> <nexthop>	Set next-hop
route origin <rid> <1|2|3>	Set origin (0=IGP, 1=EGP, 2=INCOMPLETE)
route remove <rid|all>	Remove route (will be withdrawn if it is currently injected)
route show <rid|all>	Show route information
show config	Show config
show debug (<arg>)	Show debugging options
show peer <peerid|ip address|remote asn>	Show detailed peer information
show peers	Show peer overview
show route <peerid|all> <route>	Show detailed route information
show routes <peerid|all>	Show route overview
show sentroute <peerid|all> <route>	Show detailed information of sent routes
show sentroutes <peerid|all>	Show overview information of sent routes
unflap peer <peerid|all>	Stop peer flapping, peer will stay in last flap state
unflap route <peerid|all> <rid>	Stop route flapping, route will stay in last flap state
withdraw aggregator <peerid|all> <asn|ip>	Withdraw routes matching aggregator
withdraw all (<peerid>)	Withdraw all routes
withdraw aspath <peerid|all> <as1>… <asN>	Withdraw routes matching AS path
withdraw atomic <peerid|all> <0|1>	Withdraw routes matching ATOMIC_AGGREGATE
withdraw community <peerid|all> <c1>…	Withdraw routes matching community

<cN>	
withdraw localpref <peerid|all> <localpref>	Withdraw routes matching localpref
withdraw med <peerid|all> <med>	Withdraw routes matching MED
withdraw nexthop <peerid|all> <nexthop>	Withdraw routes matching nexthop
withdraw origin <peerid|all> <origin>	Withdraw routes matching origin
withdraw rid <peerid|all> <rid>	Withdraw a specific route
withdraw route <peerid|all> <route>	Withdraw route matching prefix
test start <testfile> <outputfile>	Start test
test sleep <seconds>	Wait some seconds
test waitfor <seconds> “<regexp>”	Wait for match