BGP Replay Tool (BRT) v0.1

Bahaa Al-Musawi, Philip Branch, Grenville Armitage
Centre for Advanced Internet Architectures, Technical Report 160304A
Swinburne University of Technology
Melbourne, Australia
balmusawi@swin.edu.au, pbranch@swin.edu.au, garmitage@swin.edu.au

***Abstract*—This technical report describes the operation of BGP replay tool v0.1 (BRT v0.1), a tool to replay past BGP updates with time stamp. Comparing with other BGP replay and inject tools, BGP replay tool does not require kernel modification at host's OS and support different BGP attributes. The evaluation of this tool has been done using Virtual Internet Routing Lab (VIRL) as a controlled testbed.**

***Index Terms*—BGP, routing, emulation, Quagga, VIRL**

I. Introduction

The Border Gateway Protocol (BGP) is the Internet's default inter-domain routing protocol. BGP is a path vector protocol responsible for managing network reachability information between Autonomous Systems (ASes) with guarantees of avoiding routing loops. As with IP addresses, each AS has a unique identifier called the AS number, taken from either public or private AS number space. Original AS numbers were 2-bytes but due to growth in demand, 4-byte AS numbers were subsequently introduced [1]. BGP is an incremental protocol where BGP messages are exchanged to reflect changes in the topology and routing policies.

BGP messages can be obtained from local BGP log files or from public BGP repositories such as the RouteViews project and Réseaux IP Européens (RIPE) Network Coordination Centre [2], [3]. These two repositories provide data in Multi-Threaded Routing Toolkit (MRT) format described in [4]. The MRT format is not a human readable. Software such as bgdump [5] and pybgpdump [6] are used to convert it to a readable format. Since it was widely used, different types of BGP tools have been introduced which can be classified into extract significant information from a series of BGP updates such as BGP-Inspect [7], speed up processing such as in [8], inject a series of BGP updates such as in [9] and replay BGP updates such as in [10]. Our interest is in the latter. In particular, we are interested in replay past BGP updates.

This paper describes the operation of BGP replay tool v0.1 (BRT v0.1) [11], a tool for UNIX operating systems providing the ability to replay previously BGP updates downloaded from the public route repositories or local log files to test a variety of operations. It helps to classify BGP traffic, understand BGP behaviour at BGP speakers level and finally investigating BGP behaviour with different routers operating system (OS) such Cisco, Juniper, and Quagga.

The rest of this technical report is organised as follows: Section II includes basic background about BGP. Section III shows the operation of BGP replay tool. Section IV contains detailed information about configuration setup for the emulator. Section V is our evaluation for our tool using a controlled testbed. Section VI includes our conclusion and future work

II. BGP Background

BGP is an incremental protocol where after a complete exchange of routing table or Routing Information Base (RIB), only changes to the routing table information are exchanged through announcement messages, withdrawal messages or an update of existing route attributes. RIB for a BGP speaker consists of Adj-RIBs-In, Adj-RIBs-Out, and Loc-RIB. Adj-RIBs-In refers to routing information that is learned from neighbors but they are not processed while Adj-RIBs-Out refers to routing information that are ready for advertisement to specific peers. Loc-RIB refers to the routes that will be used by the local BGP speaker based on its local policies and Adj-RIBs-In received [12].

BGP uses Transmission Control Protocol (TCP) with TCP port number 179 [12]. Using TCP as a transport protocol avoids the need for BGP to manage message delivery and flow control between its peers and eliminates extra data used to confirm connection reliability. The size of BGP messages ranges from 19 octets, containing only a BGP header, to 4096 octets. Regardless of type, each message has a fixed size header as shown in Figure 1.

The first 16 octets are all ones to mark the start of a message. While the length field represents the total message length, the type field refers to one of four possibilities: OPEN, UPDATE, NOTIFICATION, and KEEPALIVE. OPEN message is the first message sent after establishing a TCP connection between two peers. When the other side accepts this message, KEEPALIVEs are periodically transmitted to confirm the connection.

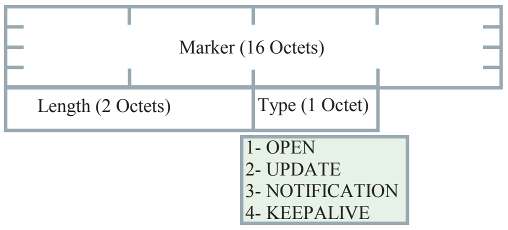


Figure 1: BGP Common Message Header Format

The most important message is the UPDATE message which is used to announce a new route, withdraw a route that was advertised previously, or update an existing route with new parameters. An AS can withdraw an announced route if and only if that AS previously advertised it. However, an AS can announce or withdraw multiple routes that have the same path attributes.

BGP uses a set of attributes to determine the best route among many possible routes. These attributes are mainly classified into four types: well-known mandatory (should be included in all BGP updates and all BGP speakers can recognise them), well-known discretionary (could be included in a BGP update and all BGP speakers can recognise them), optional transitive (can be recognised by some BGP speakers. They should be accepted and sent to peers even if it is not recognized by BGP peers) and optional non-transitive attributes (can be recognised by some BGP speakers. They can be ignored and not advertised to peers). However, the most well-known and widely used attributes are: Origin, AS-PATH, LOCAL-PREF, Multi Exit Discriminator (MED), and community attribute [12].

Origin is a well-known mandatory attribute created by the BGP speaker that generates the related routing information. It refers to the type of an originated update with three possibilities: 0 refers to an update originating from Interior Gateway Protocol (IGP) such as Open Shortest Path First (OSPF), 1 refers to an update originating from External Gateway Protocol (EGP) such as BGP, and 2 for INCOMPLETE, when a route originates from a route process instead of BGP.

AS-PATH is a well-known mandatory attribute. It identifies a list of ASes that have had an update message passing through their prefixes. BGP is a path vector protocol where each BGP speaker adds its own AS number in the path of a BGP update before passing it to an external BGP peer. This attribute prevents routing loops between BGP speakers.

LOCAL-PREF is a well-known discretionary attribute. It represents a degree of preference for a network operator for a route between many routes within an AS. A high value of this attribute shows a strong preference for a particular route. For example, in a business relationship Internet Service Providers (ISPs) prefer routes learned from their customers over routes learned from a peer; therefore, a high value of LOCAL-PREF in range 81-90 could be assigned for customers, 71-80 for peers, and 61-70 for providers [13]. This attribute, however, should not be used with external peers except for the BGP confederation case described in RFC5065 [14].

MED is an optional non-transitive attribute. MED attribute is used between external peers or within inter-domain to use a point among many exits or entry points to the same peered AS. The MED with the lower metric is preferred as an exit point.

Community is an optional transitive attribute which consists of set of four octet values. Each of these octet values refers to a specific community.

Among these attributes, a BGP router follows a sequence of comparisons to find its best route among various routes based on their attributes. Table I shows the sequence of comparisons.

Table I: BGP Path selection priority

Priority	Policy Attribute
1.	Highest LOCAL-PREF value
2.	Lowest AS-PATH length
3.	Lowest Origin Type
4.	Lowest MED value
5.	EBGP learned over IBGP learned
6.	Lowest IGP cost
7.	Lowest Router ID

BGP updates can be obtained using public repositories such as RIPE and RouteViews or from local BGP log files. BGP updates are commonly with the MRT binary format. MRT is not a human readable, tools such as bgdump [5] and pybgpdump [6] can be used to convert it to a readable format. These tools convert MRT format to different styles of readable format. For example, the bgpdump tool provides three options of conversion, this include [-H], [-m], and [-M] options. The [-H] option is the default option and used to convert MRT file to multi-line human readable. The [-m] option is used to produce one-line per entry with Unix time stamps while [-M] produce one-line per entry with human readable time stamps.

A typical example of the bgpdump tool with default option and with [-m] are shown in Figures 2 and 3 respectively.

```
TIME: 02/23/16 08:00:00
TYPE: BGP4MP/MESSAGE/Update
FROM: 213.144.128.203 AS13030
TO: 128.223.51.102 AS6447
ORIGIN: IGP
ASPATH: 13030 4230 263629
NEXT_HOP: 213.144.128.203
MULTI_EXIT_DISC: 1
COMMUNITY: 13030:1 13030:1013 13030:51904 13030:7184
ANNOUNCE
  179.125.45.0/24
  179.125.46.0/24

TIME: 02/23/16 08:00:44
TYPE: BGP4MP/MESSAGE/Update
FROM: 137.164.16.84 AS2152
TO: 128.223.51.102 AS6447
WITHDRAW
  95.85.96.0/19
  103.193.104.0/22
  205.71.208.0/20
```

Figure 2: Example of bgpdump tool with [-H] option

```
BGP4MP|1456214400|A|213.144.128.203|13030|179.125.45.0/24|13030 4230
263629|IGP|213.144.128.203|0|1|13030:1 13030:1013 13030:51904 13030:7184|NAG||
BGP4MP|1456214400|A|213.144.128.203|13030|179.125.46.0/24|13030 4230
263629|IGP|213.144.128.203|0|1|13030:1 13030:1013 13030:51904 13030:7184|NAG||
BGP4MP|1456214444|W|137.164.16.84|2152|95.85.96.0/19
BGP4MP|1456214444|W|137.164.16.84|2152|103.193.104.0/22
BGP4MP|1456214444|W|137.164.16.84|2152|205.71.208.0/20
```

Figure 3: Example of bgpdump tool with [-m] option

III. BGP Replay Tool

BGP Replay Tool (BRT) is a Perl script that allows to setup a BGP adjacency with BGP peer. BRT enables users to send out BGP updates from a predefined BGP update file. BGP session and message handling are done by Net::BGP v0.16, a Perl module that implements BGP-4 inter-domain routing protocol. This tool helps researchers and operators to understand BGP behaviour with different circumstances.

The input of the BRT tool is a human readable BGP updates with Unix time stamps, bgpdump with [-m] can be used for this purpose. Before running the BRT tool to replay BGP updates into controlled testbed, we need to check that none of the AS numbers in the implemented topology are existing in AS-PATH of announced routes for the injected file. This is important to ensure that all injected BGP updates are forwarded between ASes as BGP guarantees of avoiding routing loops through preventing routes that contain its local AS number in the AS-PATH. The script named asn-checker-v0.1.sh, available in [11], is a bash script checks all ASes numbers in the implemented topology and notifies users if they need to change a specific AS number.

To run BRT script, some mandatory inputs are need. These include BRT AS number, BRT IP address, peer AS number, peer IP address, and the selected BGP update file. Users need sudo privileges to run BRT tool, this is necessary to enable Net::BGP module to binding to the well-known BGP port 179. For example, to run the BRT script in the network shown in Figure 4 we do:

```
#./BRT-0.1.pl -brtas 65001 -brtip 172.16.1.29
   -peeras 40 -peerip 172.16.1.129 -f in
```

BRT and other scripts are available in [11]. BRT creates a log file named brt.log that logs time stamps of user's OS with Unix format verses time stamp of the input BGP update file as shown in Figure 5. This log file shows that BRT sends past BGP updates with current time stamps of host's OS.

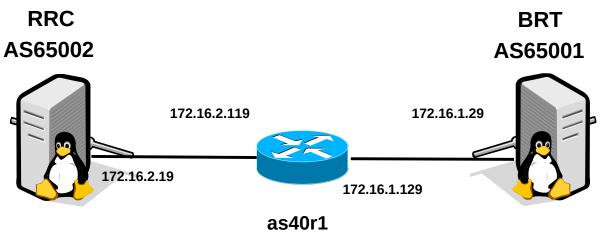


Figure 4: Simple Topology-1

```
System time at time stamp 1117217107 is -->1456881227
System time at time stamp 1117217114 is -->1456881234
System time at time stamp 1117217117 is -->1456881237
System time at time stamp 1117217124 is -->1456881244
System time at time stamp 1117217127 is -->1456881247
System time at time stamp 1117217134 is -->1456881254
System time at time stamp 1117217137 is -->1456881257
System time at time stamp 1117217144 is -->1456881264
System time at time stamp 1117217147 is -->1456881267
System time at time stamp 1117217154 is -->1456881274
```

Figure 5: Sample of BRT log file

BRT is highly experimental, and could be improved and extended in many ways. It has been proved to work quite well in specific context, but it might need some changes in the source code to make it work under different circumstances. BRT has been used and tested to peer with IPV4 and Cisco routers as peer BGP speakers but it may also work with other BGP speakers such as Juniper routers. BRT is based on using Net::BGP v0.16 so it has the limitation of Net::BGP such as terminating a connection when a collision appears.

IV. Emulator Setup

To emulate past BGP updates with a controlled testbed network, we use BRT to inject past BGP updates and Remote Route Collector (RRC) to collect BGP updates. Figure 4 shows a simple topology to replay BGP updates and check the received data. RRC needs Quagga to be installed. Quagga is a routing software package that provides TCP/IP based routing services for different protocols such as OSPF, IS-IS, and BGP [15]. Quagga is made from several daemons that work together to build the routing table. These daemons include ospfd, ripd, bgpd, and zebra where zebra represents the kernel routing manager. Figure 6 shows Quagga system architecture.

RRC runs Ubuntu 14.04 LTS and Quagga version 0.99.22.4. The configuration files for Quagga installed in Ubuntu OS is under /etc/quagga where /etc/quagga/Quagga.conf is the configuration file of configuring routing. Table II shows an example of /etc/quagga/Quagga.conf to establish a peer connection between RRC and AS40 for the topology shown in Figure 4.

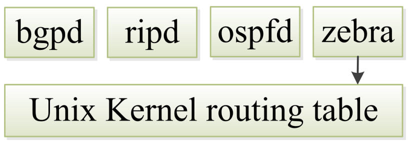


Figure 6: Quagga system Architecture

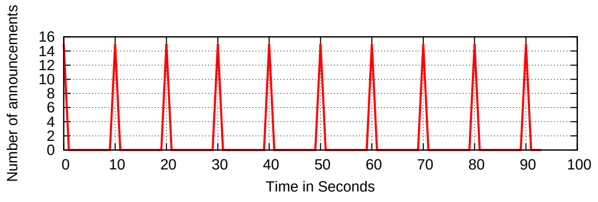
BRT requires using Net::BGP, a module of Perl software. This module can be installed as follows:

```
#Perl -MCPAN -e shell
cpan[1]> install Net::BGP
```

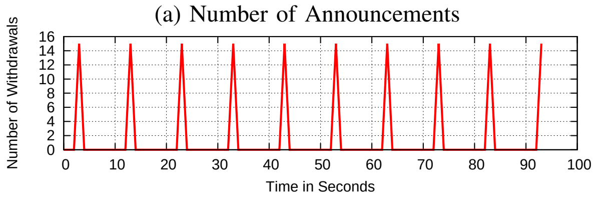
V. Evaluation

To evaluate the BRT tool, we inject BGP data into a controlled testbed. Our testbed is running on the Virtual Internet Routing Lab (VIRL), a network emulation system uses Linux KVM hypervisor, OpenStack, and a set of virtual machines running real Cisco network operating systems [16]. We do two experiments with two different topologies to evaluate BRT. In each experiment both BRT and RRC are running Ubuntu 14.04.2 LTS operating system. In addition, RRC is running Quagga version 0.99.22.4.

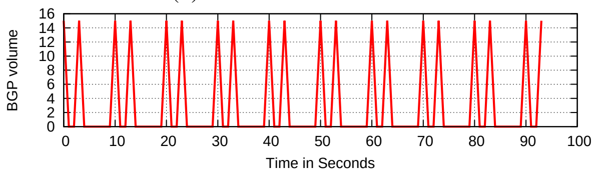
Experiment-1: In this experiment we use a simple topology shown in Figure 4. In this experiment, we inject a simple set of BGP updates that represents a series of announcements for15 prefixes then withdrawal them after 3 seconds for a period of 100 seconds. In this topology, as40r1 is running Cisco IOSv 15.2(2)T.



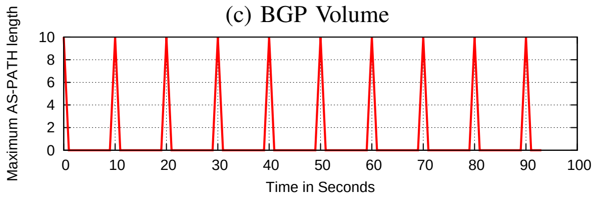
(a) Number of Announcements



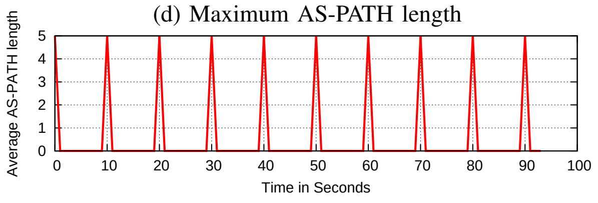
(b) Number of Withdrawals



(c) BGP Volume



(d) Maximum AS-PATH length



(e) Average AS-PATH length

Figure 7: BGP features for the injected data

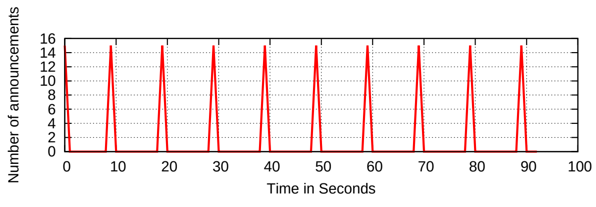
For a simple investigation and monitoring, we set the value of Minimum Route Advertisement Interval (MRAI) to zero. The MRAI refers to the minimum amount of time between two subsequent advertisements to a particular destination, the default value in Cisco routers is 30 seconds. This value can be set using the following commands:

```
en
conf t
router bgp 40
neighbor 172.16.2.19 advertisement-interval 0
neighbor 172.16.1.29 advertisement-interval 0
```

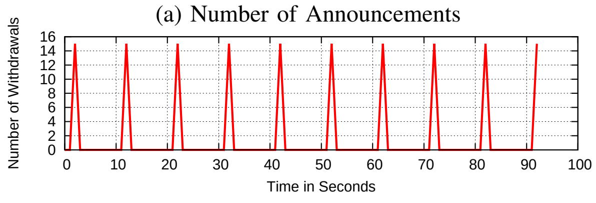
Figure 7 and 8 show five BGP features for the injected and collected BGP data, these include BGP volume (total number of announcements and withdrawal), number of

Table II: BGP Configuration at RRC

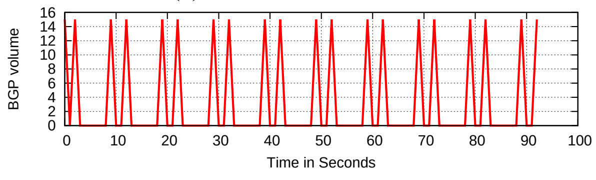
dump bgp updates updates.dump	Dump BGP updates to file updates.dump in the current directory. It is necessary that the file exists and is writable by Quagga.
debug bgp	Enable logging
debug bgp events	Enable logging of BGP events
debug bgp updates	Enable logging of BGP advertisements
router bgp 65002	Set AS number 65002 for the RRC
bgp router-id 172.16.2.19	Set router ID 172.16.2.19 to the RRC
bgp log-neighbor-changes	Enable logging of BGP neighbor status changes (up or down)
neighbor 172.16.2.119 remote-as 40	Set 172.16.2.119 as a peer AS 40
neighbor 172.16.2.119 filter-list 20 out	Do not send updates to AS40. This is necessary as RRC works only to collect data
ip as-path access-list 20 deny .*	Applies the filter-list 20 to all addresses



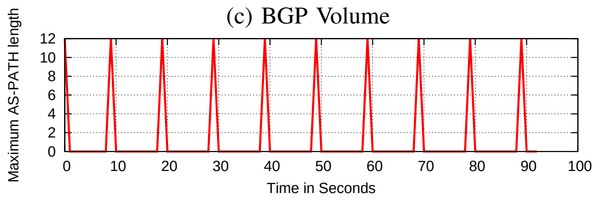
(a) Number of Announcements



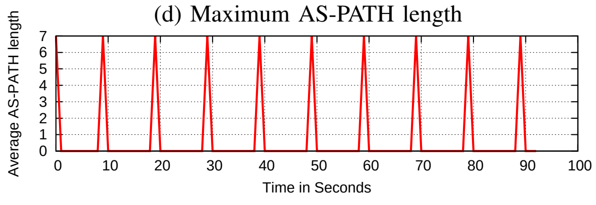
(b) Number of Withdrawals



(c) BGP Volume



(d) Maximum AS-PATH length



(e) Average AS-PATH length

Figure 8: BGP features for the collected data at RRC

announcements, number of withdrawals, maximum AS-PATH length, and average AS-PATH length calculated every second for the injected data. These BGP features are extracted using bgp-features-0.1.sh, a bash script available in [11]. As can be seen from these Figures that number of BGP announcements and withdrawals are identical per time stamp. However, there is a difference in length of AS-PATH which can be seen in Figures 7d, 7e, 8d and 8e. This due to increasing the number of hops as a result of adding AS65001 for BRT and AS40 for the Cisco router.

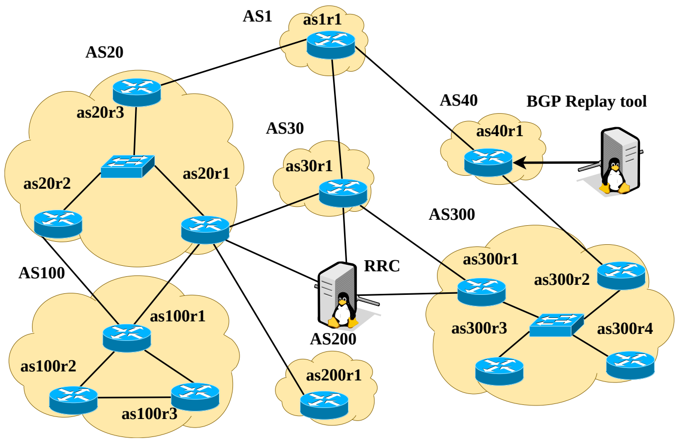
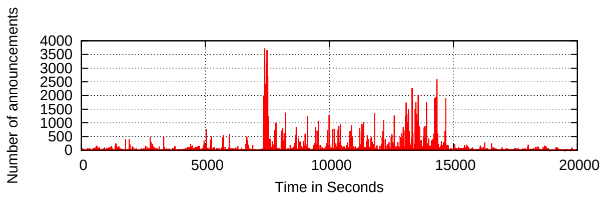


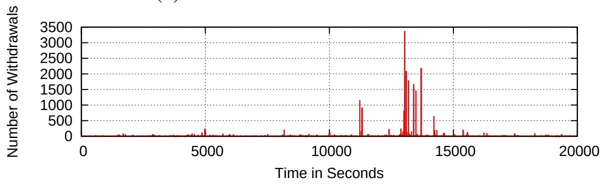
Figure 9: Testbed topology

Experiment-2: In this experiment we emulate TMnet event, one of the most recent incidents of BGP instability was observed on the 12th of July 2015 by Telekom Malaysia (TMnet) which caused significant network problems for the global routing system [17]. We use data downloaded from route-views4.routeviews.org during TMnet event. In particular, we replay 5.55 hours (20000 seconds) of BGP data at AS10102 during the event. In this experiment we use a complex topology shown in Figure 9, a topology that has been used in [18]. In this topology, we use the default value for MRAI, 30 seconds. In this case, each BGP speaker will forward best routes in BGP updates received from neighbors after MRAI timers expired.

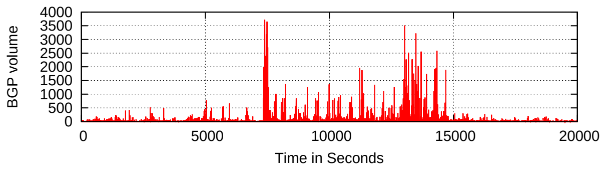
Figures 10 and 11 show BGP features for TMnet emulation event using controlled testbed. In Figures 10a and 11a, we can see notable differences between number of announcements for the injected and collected BGP updates. These differences due to the value of MRAI. All Cisco routers in the topology shown in Figure 9 do not forward all received BGP updates every second. They forward best routes for the last 30 seconds, the default value for MRAI. However, the difference in values of maximum AS-PATH lengths as shown in Figures 10d and 11d is reasonable due to 5 hops between the BRT and RRC.



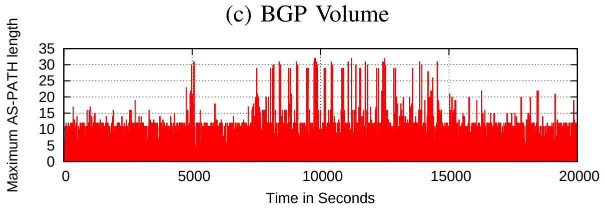
(a) Number of Announcements



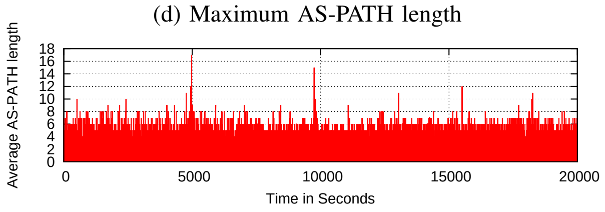
(b) Number of Withdrawals



(c) BGP Volume



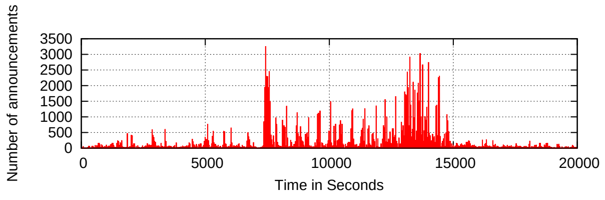
(d) Maximum AS-PATH length



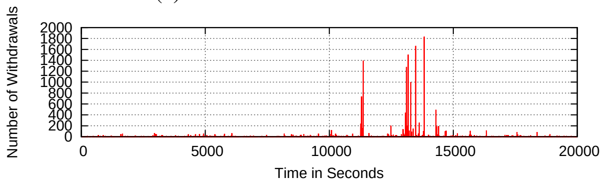
(e) Average AS-PATH length

Figure 10: BGP features for the injected data of TMnet

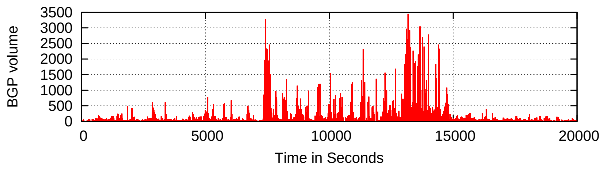
Table III shows a comparison of techniques described in [9], [10] as well as BRT tool. In [10] requires kernel modification at host's OS to replay past BGP events. This tool also does not check whether an AS number in the desired topology exist in the injected BGP updates nor



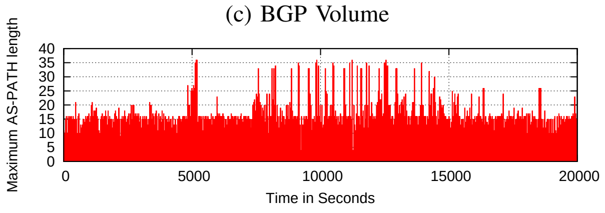
(a) Number of Announcements



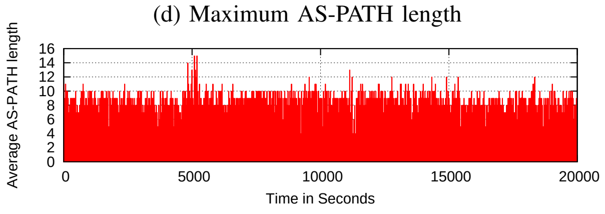
(b) Number of Withdrawals



(c) BGP Volume



(d) Maximum AS-PATH length



(e) Average AS-PATH length

Figure 11: BGP features for the collected data at RRC

Table III: Comparison Among BGP tools

Feature	MDFMT [10]	bgpsimple [9]	BRT
Replay BGP update with time stamp	Yes	No	Yes
Require modification in the Kernel	Yes	No	No
Support community attribute	No	No	Yes
Checking AS number with the implemented topology	No	No	Yes

support all BGP attributes such as community attribute. bgpsimple is a tool to inject BGP updates from a selected file. This tool again does not check whether an AS number in the desired topology exist in the injected

BGP updates nor support all BGP attributes such as community attribute. In contrast, the BRT tool does not require modification in the kernel of host's OS and support many attributes. In addition, this tool checks if an AS number in the intended topology exists in the selected BGP update file to provide a correct replay of past BGP events.

VI. Conclusions

BGP Replay Tool (BRT) version 0.1 is a tool to replay BGP updates with time stamps. This tool can be used to inject a list of BGP updates and replay BGP updates based on time stamps. It helps operators and researchers to understand BGP behaviour at BGP speaker level, classify BGP updates, and investigate BGP behaviour at different routers OS such as Quagga, Cisco and Juniper IOS. Our future work will involve peering BRT with multiple BGP speakers with different routers OS such as Juniper's JUNOS.

Acknowledgements

We are grateful to the VIRL team at Cisco for providing free license and support. We also thank Simon Forsayeth from ITS/Swinburne University for setting up the physical rack of hosts on which we ran VIRL.

References

[1] Q. Vohra and E. Chen, "BGP Support for Four-Octet Autonomous System (AS) Number Space," RFC 6793 (Proposed Standard), Internet Engineering Task Force, December 2012. [Online]. Available: http://www.ietf.org/rfc/rfc6793.txt

[2] Reseaux IP Europeens Network Coordination Center. [Online]. Available: http://www.ripe.net/

[3] University of Oregon, "University of Oregon Route Views Project." [Online]. Available: http://www.routeviews.org/

[4] L. Blunk, M. Karir, and C. Labovitz, "RFC 6396: Multi-Threaded Routing Toolkit (MRT) Routing Information Export Format," RFC 4271 (Proposed Standard), Internet Engineering Task Force, October 2011. [Online]. Available: http://tools.ietf.org/html/rfc6396

[5] RIPE NCC RIS Projec, "bgpdump." [Online]. Available: https://bitbucket.org/ripencc/bgpdump/wiki/Home

[6] J. Oberheide, "pybgpdump." [Online]. Available: https://jon.oberheide.org/pybgpdump/

[7] D. Blazakis, M. Karir, and J. Baras, "BGP-Inspect - Extracting Information from Raw BGP Data," in *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP*, April 2006, pp. 174–185.

[8] M. Rossi, "Quagga-Accelerator: An Implementation for Accelerated Processing of Historical BGP Events using Quagga 0.99.13 - version 0.1 ," Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, Tech. Rep. 090730C, 30 July 2009. [Online]. Available: http://caia.swin.edu.au/reports/090730C/CAIA-TR-090730C.pdf

[9] Google Project Hosting, "Simple BGP Peering and Route Injection Script," February 2016. [Online]. Available: https://code.google.com/archive/p/bgpsimple/

[10] M. Rossi, "MRT dump file manipulation toolkit (MDFMT) - version 0.2," Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, Tech. Rep. 090730B, 30 July 2009. [Online]. Available: http://caia.swin.edu.au/reports/090730B/CAIA-TR-090730B.pdf

[11] B. Al-Musawi, "BGP Replay Tool (BRT)v0.1," March 2016. [Online]. Available: http://caia.swin.edu.au/tools/bgp/brt-0.1.tgz

[12] Y. Rekhter, T. Li, and S. Hares, "RFC 4271: A Border Gateway Protocol 4 (BGP-4)," RFC 4271 (Proposed Standard), Internet Engineering Task Force, January 2006. [Online]. Available: http://tools.ietf.org/html/rfc4271

[13] M. Caesar and J. Rexford, "BGP routing policies in ISP networks," *Network, IEEE*, vol. 19, no. 6, pp. 5–11, Nov 2005.

[14] P. Traina, D. McPherson, and J. Scudder, "Autonomous System Confederations for BGP," RFC 5065 (Standards Track), Internet Engineering Task Force, August 2007. [Online]. Available: http://tools.ietf.org/html/rfc5065

[15] K. Ishiguro, "Quagga Routing Suite." [Online]. Available: http://www.nongnu.org/quagga/

[16] J. Obstfeld, S. Knight, E. Kern, Q. S. Wang, T. Bryan, and D. Bourque, "VIRL: the virtual internet routing lab," in *Proceedings of the 2014 ACM conference on SIGCOMM*. ACM, 2014, pp. 577–578.

[17] A. Toonk, "Massive route leak causes Internet slowdown," BGPMON, June 2015. [Online]. Available: http://www.bgpmon.net/massive-route-leak-cause-internet-slowdown/

[18] H. Nguyen, M. Roughan, S. Knight, N. Falkner, O. Maennel, and R. Bush, "How to build complex, large-scale emulated networks," in *Testbeds and Research Infrastructures. Development of Networks and Communities*. Springer, 2010, pp. 3–18.