BGP Replay Tool (BRT) v0.2

Bahaa Al-Musawi, Rasool Al-Saadi, Philip Branch, Grenville Armitage
Internet for Things (I4T) Research Lab, Technical Report 170606A
Swinburne University of Technology
Melbourne, Australia
balmusawi@swin.edu.au, ralsaadi@swin.edu.au, pbranch@swin.edu.au, garmitage@swin.edu.au

***Abstract*—This technical report describes the operation of BGP replay tool v0.2 (BRT v02), a tool to replay past BGP updates with time stamps. Compared to other BGP replay and inject tools, BRT v0.2 does not require kernel modification at the host's OS, supports different BGP attributes, supports sending IPv6 BGP updates and peering over IPv6. The evaluation of this tool has been done using real Cisco routers, Quagga and Virtual Internet Routing Lab (VIRL) as controlled testbeds.**

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

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. BGP was developed at a time when information provided by an AS could be assumed to be accurate. Consequently, it includes few security mechanisms and so is vulnerable to different types of events such as hijacking, misconfiguration, and link or node failure. Considerable research has been carried out into BGP. Generally, research works can be classified as security improvements using cryptographic approaches, anomaly detection and mitigation, and BGP tools [1]. Our interest is in the latter. In particular, we are interested in presenting a tool that can help operators and researchers to improve security issues with BGP through replaying past BGP events into a controlled testbed.

BGP traffic has been characterised as complex, noisy, and voluminous [2]. BGP speakers, a router or a device that runs BGP, generate up to a gigabyte of control plane data a day. Unfortunately, as well as being large, there is no direct information to identify the network that triggered the BGP messages [1]. 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 [3], speed up processing such as in [4], inject a series of BGP updates such as in [5] and replay BGP updates such as in [6]. This report introduces BGP Replay Tool (BRT v0.2) [7], a tool for UNIX and Windows operating systems providing the ability to replay previously captured BGP updates downloaded from the public route repositories or local log files to test a variety of operations. Replying past BGP incidents into a control testbed helps to classify BGP traffic, understand BGP behaviour at BGP speaker level and investigating BGP behaviour with different routers operating systems (OSs) such as Cisco, Juniper, and Quagga. BRT v0.2 extends the ability of BRT v0.1 [8] to peer with different BGP speakers operating systems such as Quagga and real Cisco routers. It also supports IPV6 and connecting to multiple peers. The evaluation of the BRT v0.2 has been made using three different types of testbeds. These include real Cisco routers, Virtual Internet Routing Lab (VIRL), an emulation platform by Cisco, and Quagga using generated BGP updates and past BGP instability incidents.

BRT v0.2 uses Net::BGP [9], a module of Perl software, to implement BGP. Net::BGP provides the required functionality to establish BGP peering and exchanging BGP updates. However, Net::BGP does not support BGP updates for IPv6 nor BGP connection over IPv6. Therefore, we develop a patch that supports BGP updates for IPv6 based on using Multi-protocol Reachable NLRI (Network Layer Reachability Information) and Multi-protocol Unreachable NLRI, BGP attributes described in [10]. We also provide a script to calculate nine BGP features for comparing injected and collected BGP updates. These features are total number of IPV4 and IPv6 announcements, IPv4 and IPv6 withdrawals, maximum and average length of AS-PATH, total number of announcements, total number of withdrawals, and total number of announcements and withdrawals.

The rest of this technical report is organised as follows: Section II includes an overview of BGP, BGP messages, BGP attributes and BGP data. Section III

shows the operation of BGP replay tool. Section IV contains detailed information about configuration setup to emulate past BGP updates with a controlled testbed while section V represents our evaluation using different controlled testbeds. In section VI, we conclude our work and outline future directions.

II. BGP Background

A. BGP Overview

The Internet is a decentralized global network comprised of tens of thousands of Autonomous Systems (ASes). An AS is a set of routers under a single technical administration using an Interior Gateway Protocol (IGP) such as Open Shortest Path First (OSPF) to communicate with other routers within the AS and an Exterior Gateway Protocol (EGP) such as Border Gateway Protocol (BGP) to communicate with other ASes. Routing protocols are classified into three main types based on their algorithm: link state such as OSPF, distance vector such as Routing Information Protocol (RIP), and path vector such as BGP. BGP has two forms: Internal BGP (IBGP), running between BGP routers within an AS, and External BGP (EBGP), running between BGP routers within different ASes [1].

BGP is the Internet's default EGP. It maintains and exchanges NRI between ASes which are organized in a hierarchical fashion. As with IP addresses, each AS has a unique identifier called the AS number, taken from either public or private AS number space [11]. Original AS numbers were 2-bytes and ranged from 0 to 65535. Due to growth in demand, 4-byte AS numbers were subsequently introduced ranging from 0 to 4294967295 [12]. The Internet Assigned Number Authority (IANA) has reserved, for private use, the last 1023 numbers of 2-byte AS numbers, namely 64512-65534, and the last 94967295 numbers of 4-byte AS numbers, namely 4200000000-4294967294 [13]. Each AS has a range of IP addresses identified by a prefix. For example, the IPv4 address prefix 192.2.2.0/24 refers to all addresses in the range 192.2.2.0-192.2.2.255 while the IPv6 address prefix 2001::/19 refers to all addresses in the range 2001:: to 2001:1fff:ffff:ffff:ffff:ffff:ffff:ffff. BGP provides a set of mechanisms for supporting Classless Inter-Domain Routing (CIDR) described in RFC4632 [14]. These mechanisms include aggregation support of routes with their AS-PATH (a BGP's attribute described later in Section II-C) and advertising support for a set of destinations as a prefix. Aggregation is the process of combining the characteristics of several routes with common addresses into a single route. This helps reduce the number of routing messages as well as the number of advertised routes.

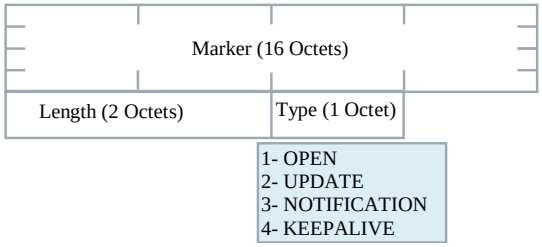


Figure 1: BGP common message header format

B. BGP Messages

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 (a router or a device that runs BGP) consists of Adj-RIBs-In, Adj-RIBs-Out, and Loc-RIB. Adj-RIBs-In refers to routing information that is learned from (adjacent) neighbours. Adj-RIBs-Out refers to routing information that is ready for advertisement to (adjacent) peers while 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 [15].

BGP uses the Transmission Control Protocol (TCP) with TCP port number 179 [15]. 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 2 shows BGP OPEN message format for a 2-byte AS number. A NOTIFICATION message supplies information regarding a terminated session.

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

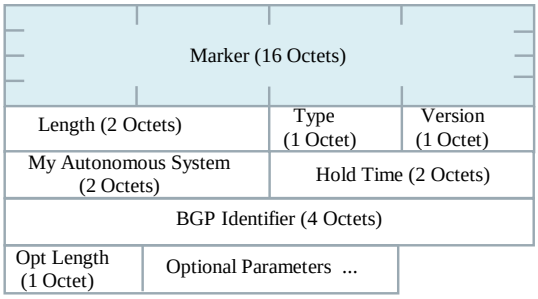


Figure 2: BGP open message format

route with new parameters. An AS can withdraw an announced route if and only if that AS previously advertised it. Also, an AS can announce or withdraw multiple routes that have the same path attributes.

Two identities for BGP speaker are represented in the OPEN message: "My Autonomous System" refers to AS number of the sender and BGP identifier described in [16], a unique identifier within an AS where its value is determined on startup and is the same for every local interface and BGP peer [1].

C. BGP Attributes

BGP attributes are a set of properties carried in a BGP update and used to determine the best route among many possible paths to a specific destination. 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). The most well-known and widely used attributes are: Origin, AS-PATH, LOCAL-PREF, AGGREGATOR, and Multi Exit Discriminator (MED) [15]. We also discuss Multiprotocol Reachable NLRI and Multiprotocol Unreachable NLRI that have been used in the BRT v0.2.

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 IGP, 1 refers to an update originating from EGP, and 2 for INCOMPLETE, when a route originates from another routing protocol instead of BGP such as static route.

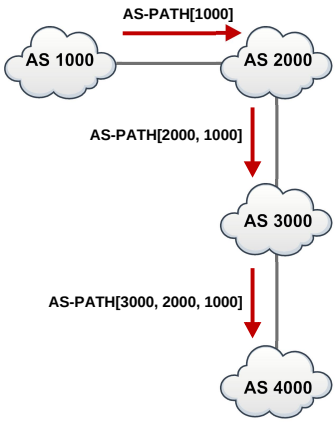


Figure 3: An example of BGP AS-PATH attribute

AS-PATH is a well-known mandatory attribute which identifies a list of ASes that have had an update message passing through their prefixes. The components of this list can be AS-SETs or AS-SEQUENCEs. AS-SET refers to an unordered set of ASes while AS-SEQUENCE refers to an ordered set of ASes. 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 EBGP peer. This attribute prevents routing loops between BGP speakers. Figure 3 shows an example of how the AS-PATH attribute works. Each AS inserts its own AS number before propagating a BGP update to its peers. When AS1000 sends a route to AS4000, it adds its own AS number to the beginning of the path. AS2000 receives the update and appends its AS number before passing it to AS3000. Finally, AS3000 receives the update, and inserts its own AS number to send it to AS4000. BGP is a path vector protocol where [3000,2000,1000] shows the full path for an update sent by AS1000 to AS4000.

LOCAL-PREF is a well-known discretionary attribute. LOCAL-PREF represents a degree of preference for a network operator for a route between multiple routes within an AS. A high value of this attribute shows a strong preference for a particular route. For example, in a business relationship ISPs will usually prefer routes learned from their customers over routes learned from a peer; therefore, a high value of LOCAL-PREF in range 99-90 could be assigned for customers, 89-80 for peers, and 79-70 for providers [17]. This attribute was used by PGBGP [18] to mitigate the propagation of suspicious routes through assigning them with low LOCAL-PREF. This attribute, however, should not be used with external

peers except for the BGP confederation case described in RFC5065 [19].

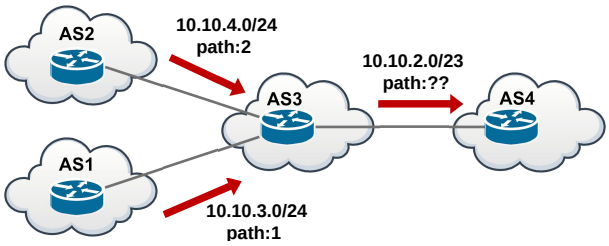


Figure 4: An example of BGP route aggregation

AGGREGATOR is an optional transitive attribute. It contains information about the BGP speaker that aggregate the route. Although the aggregation helps to reduce the number of advertising routes, it can hide AS-PATH and other attributes of the aggregated prefixes. Figure 4 shows an example for route aggregation. In this example, AS1 and AS2 advertise 10.10.3.0/24 and 10.10.4.0/24 respectively to AS3. AS3 aggregates these prefixes by sending the single prefix 10.10.2.0/23. The value of AS-PATH for the single prefix is based on the aggregation configuration at AS3. AS3 can hide the paths to AS1 and AS2 and send the prefix 10.10.2.0/23 with AS-PATH=[3], this can cause a blackhole if any of the advertised prefix by AS1 or AS2 withdrawal. AS3 can also configure the aggregation to include both of the originating ASes as AS-SET, in this case AS4 will receive the prefix 10.10.2.0/23 with AS-PATH=[3,{1,2}].

MED is an optional non-transitive attribute which provides a mechanism to influence external neighbours about the preferred path into an AS that has multiple entry points. The MED with the lower metric is preferred as an exit point.

The community attribute is an optional transitive attribute which consists of set of four octet values. Each of these octet values refers to a specific community. It can be used to mark a set of routes or prefixes that share common characteristics.

Multiprotocol Reachable NLRI and Multiprotocol Unreachable NLRI are optional and non-transitive attributes described in [10] to enable BGP for supporting multiple network protocols such as IPv6. These attributes are used by BRT v0.2 to support injecting IPv6 of BGP updates and peering via interfaces assigned with IPv6.

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

D. BGP Data

BGP data can be obtained from local BGP log files or from public repositories such as RouteViews project [20] and Réseaux IP Européens (RIPE) Network Coordinate Centre (NCC) [21]. The RouteViews and RIPE NCC are the most well-known repositories that provide free download for BGP updates and RIB. RouteViews peers with many sites in north America and had provided BGP data since 2001, while RIPE peers with many sites in Europe and provides BGP data since 1999. The total numbers of collectors and peers change over time as a result of adding/removing some vantage points[1]. The RouteViews repository provides BGP updates every 15 minutes and BGP routing tables every 2 hours. Until June 2003, RIPE was providing offline BGP updates every 15 minutes with BGP routing tables every eight hours. From 2003 it offers BGP updates every 5 minutes. These two well-known repositories provide data in MRT (Multi-Threaded Routing Toolkit) format described in [22]. The MRT format is not a human readable. Software such as bgpdump [23] and pybgpdump [24] are 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 [-m] and [-H] options are shown in Figures 5 and 6 respectively.

[1]For example, as at the 18th of January 2016 there are 18 collectors for the RouteViews project with 588 peers in different locations around the world while RIPE peers with 14 collectors around the world with 566 peers.

```
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 5: Example of bgpdump tool with [-m] option

```
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 6: Example of bgpdump tool with [-H] option

III. BGP Replay Tool (BRT) v0.2

BGP Replay Tool (BRT) v0.2 is a Perl script that allows setting up a BGP adjacency with a BGP peer. BRT v0.2 enables users to send out BGP updates from a predefined BGP update file. This tool can help researchers and operators to understand BGP behaviour in different circumstances. BGP session and message handling are done by Net::BGP v0.16, a Perl module that implements BGP inter-domain routing protocol. Officially, Net::BGP v0.16 does not support IPv6 BGP updates neither IPv6 BGP peer connection. CAIDA [25] has developed a patch for Net::BGP that allows BGP speaker to send IPv6 announcements through Multi-protocol Reachable NLRI, an optional attribute supported as part of Multi-protocol Extensions for BGP described in [10]. However, this patch does not support IPv6 prefix withdrawn and required BGP speakers with ADD-PATH capability, an extension to BGP protocol described in [26] to allow advertisement of multiple paths for the same prefix. Therefore, we implemented IPv6 route withdrawn through Multi-protocol Unreachable NLRI optional attribute, and removed ADD-PATH BGP capability for compatibility purposes. The BRT v0.2 and the patch are tested on Perl 5.20.2 and Net::BGP 0.16, and it is publicly available at [7].

The input of the BRT v0.2 tool is a human readable BGP updates with Unix time stamps, bgpdump with [-m] can be used for this purpose. BRT V0.2 provides and option to check that none of the AS numbers in the implemented topology are existing in any AS-PATHs 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.

BRT v0.2 tool has optional and mandatory command line arguments as shown in Table II. It is worth noting that IPv6 options should be specified if bgpdump update files contains IPv6 prefixes or when the BGP connection is made over IPv6 protocol.

A simple example for using the BRT for a simple BGP topology shown in Figure 7 is:

```
$ perl brt-0.2.pl -brtas 65001 -brtip
    172.16.2.2 -peeras 65002 -peerip
    172.16.2.1 -f BGP_updates
```

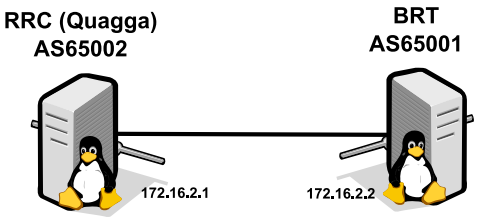


Figure 7: Simple topology with only RRC

BRT v0.2 also supports replay BGP updates to multiple BGP peers at once by storing BRT v0.2 tool arguments (command line arguments) for each peer as a line in a text file and specify that file to the tool after '-m' argument. In this case, we only need two arguments. That is, <-f> to specify BGP updates file and <-m> to

Table II: BRT v0.2 tool command line arguments

Argument	Value	Optional	Description
-brtas	<AS number>	No	BRT AS number
-brtip	<IP address>	No	BRT IPv4 address
-brtipv6	<IPv6 address>	Yes	BRT IPv6 address
-peeras	<AS number>	No	Peer AS number
-peerip	<IP address>	No	Peer IPv4 address
-peeripv6	<IPv6 address>	Yes	Peer IPv6 address
-ipv6		Yes	Connect to a peer using IPv6. This is necessary if the connection via IPV6 not IPV4; otherwise, it can be ignored
-f	<filename>	No	BGP update file in human readable with Unix format
-m	<filename>	Yes	Connect to multiple peers specified in <filename>
-s	<filename>	Yes	Check that none of the ASes in the implemented topology are existing in any AS-PATHs of announced routes for the injected file
-v		Yes	Verbose mode
-help		Yes	Display BRT tool help

specify all other mandatory and optional attributes. For example, the content of <-m> file for the topology shown in Figure 8 is:

```
-brtas 65001 -brtip 172.16.1.100 -brtipv6 fc00
    :3::1 -peeras 65002 -peerip 172.16.1.200 -
    peeripv6 fc00:3::2
-brtas 65001 -brtip 172.16.1.100 -brtipv6 fc00
    :3::1 -peeras 65003 -peerip 172.16.1.201 -
    peeripv6 fc00:3::3
```

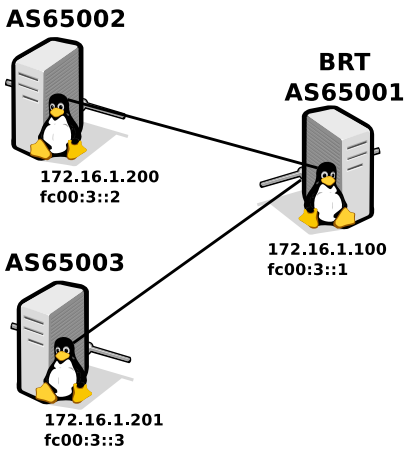


Figure 8: An example of peering BRT with two peers

BRT v0.2 is still experimental. BRT v0.2 has been used and tested on IPV4 and IPv6 peers with Quagga, real Cisco routers and VIRL as peer BGP speakers but not on other BGP speakers such as Juniper routers. It may work with these but has not been tested.

Table III shows a comparison of techniques described in [5], [6] as well as our tool. MRT Dump File Manipulation Toolkit (MDFMT) [6] is a pseudo BGP speaker.

Table III: Comparison Among BGP tools

Feature	MDFMT	bgpsimple	BRT v0.2
Replay BGP update with time stamp	Yes	No	Yes
Require modification in the Kernel	Yes	No	No
Supporting multiple attributes	No	No	Yes
Supporting IPV6	No	No	Yes
Supporting connection to multiple peers	No	No	Yes
Checking AS number with the implemented topology	No	No	Yes

It requires kernel modification at host's OS to replay past BGP updates. MDFMT does not support IPv6 peer connection neither IPv6 BGP updates. It also does not support many BGP attributes such as the community attribute. bgpsimple is a tool to inject BGP updates from a selected file. This tool does not send BGP updates based on time stamp. bgpsimple does not also support IPv6 for BGP updates and peering. In contrast, the BRT v0.2 tool does not require modification in the kernel of host's OS and support many attributes. Furthermore, BRT v0.2 supports sending IPv6 BGP updates and supports BGP peering over IPv6. It also supports many BGP attributes such as the community, aggregation, and MED.

IV. Emulator Setup

To emulate past BGP updates with a controlled testbed network, we use BRT v0.2 to inject past BGP updates and Remote Route Collector (RRC) to collect BGP updates. Figure 7 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 [27]. 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 9 shows Quagga system architecture.

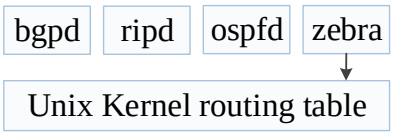


Figure 9: Quagga system Architecture

RRC runs Ubuntu 14.04 LTS and Quagga version 0.99.23.1. 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 IV shows an example of /etc/quagga/Quagga.conf to establish a peer connection between RRC and AS65001 for the topology shown in Figure 7.

BRT v0.2 requires using Net::BGP, a module of Perl software. Additionally, IO::Socket::INET6 module should be installed to add support for IPv6 BGP connection for the patched Net::BGP module. These modules can be installed as follows:

```
#perl -MCPAN -e shell
cpan[1]> install Net::BGP
cpan[1]> install IO::Socket::INET6
```

To apply IPv6 support patch to Net::BGP module, we provide a patch installation script that simplifies the process. This can be done using the following command:

```
# tar xzfv ipv6_bgpnet-0.1.tgz
# cd ipv6_bgpnet-0.1
# ./patch.sh
```

V. Evaluation

We evaluate the functionality of BRT v0.2 with three different types of testbeds. These include Quagga, Virtual Internet Routing Lab (VIRL), and real Cisco routers. VIRL is a powerful network emulation system uses Linux KVM hypervisor, OpenStack, and a set of virtual machines running real Cisco network operating systems to emulate complex network [28]. We conduct two experiments in the evaluation. In the first experiment we inject a simple series of generated BGP updates into the three types of testbed while in the second experiment we use one of the past well-know BGP incident.

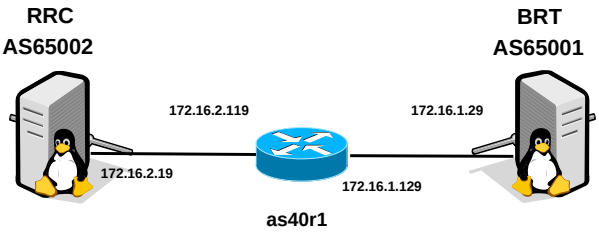


Figure 10: Simple topology with a Cisco router

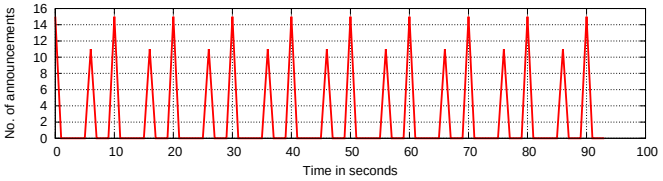
A. Replay a generated series of BGP updates

In this experiment, we inject a simple set of generated BGP updates that represents a series of announcements and withdrawals of IPv4 and IPv6 prefixes for a period of 100 seconds. In this experiment, we use three different types of testbed. That is, Quagga testbed for the topology shown in Figure 7, VIRL and real Cisco routers for the topology shown in Figure 10. Both BRT v0.2 and RRC are running Ubuntu 14.04.2 LTS operating system.

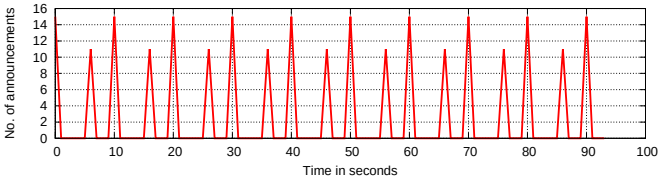
In all our experiments, we use Quagga version 0.99.23.1, VIRL BGP routers run Cisco IOSv 15.2(2)T and real Cisco BGP routers run Cisco IOSv 15.1(4)M10. For a simple investigation and monitoring, we set the value of Minimum Route Advertisement Interval (MRAI) to zero for VIRL and real Cisco routers for both IPv4 and IPv6. 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 while it is zero in Quagga. For example, setting the value of MRAI to zero in Cisco routers can be done as following:

```
en
conf t
router bgp 40
neighbor 172.16.2.19 advertisement-interval 0
neighbor 172.16.1.29 advertisement-interval 0
address-family ipv6
neighbor 172.16.2.19 advertisement-interval 0
neighbor 172.16.1.29 advertisement-interval 0
exit-address-family
```

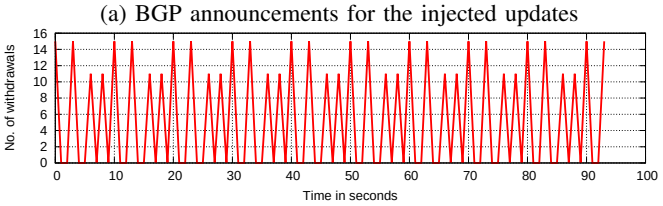
Figure 11 shows BGP features for the injected and collected BGP updates using real Cisco routers. These include BGP volume (total number of announcements and withdrawals), total number of announcements, total number of withdrawals, IPv4 announcements and withdrawals, IPv4 announcements and withdrawals, maximum and average length of AS-PATH. These BGP features are extracted using bgp-features-0.2.pl, a Perl script



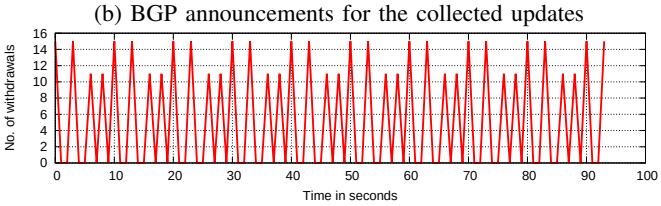
(a) BGP announcements for the injected updates



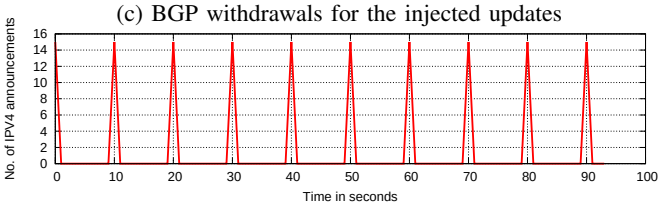
(b) BGP announcements for the collected updates



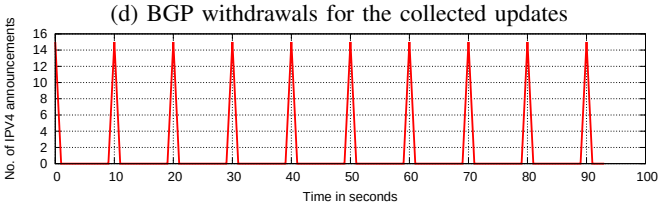
(c) BGP withdrawals for the injected updates



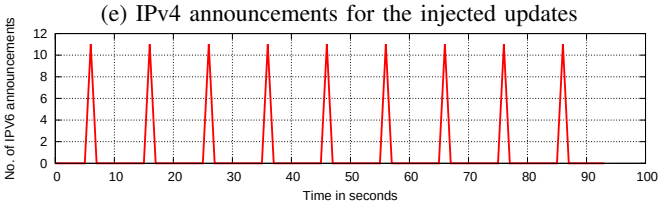
(d) BGP withdrawals for the collected updates



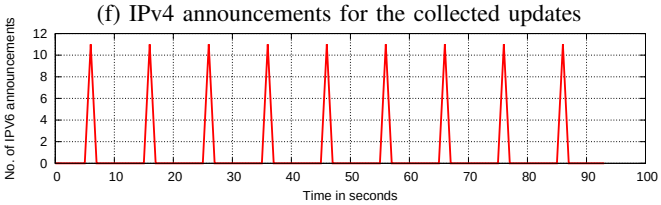
(e) IPv4 announcements for the injected updates



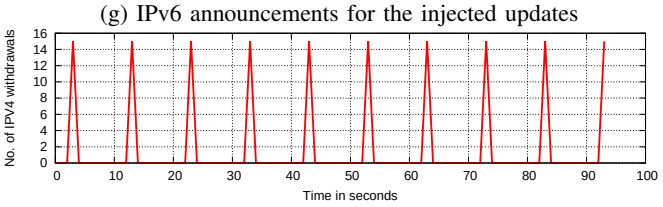
(f) IPv4 announcements for the collected updates



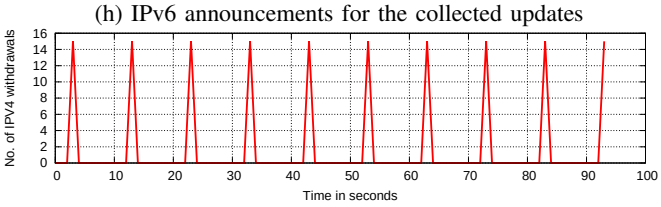
(g) IPv6 announcements for the injected updates



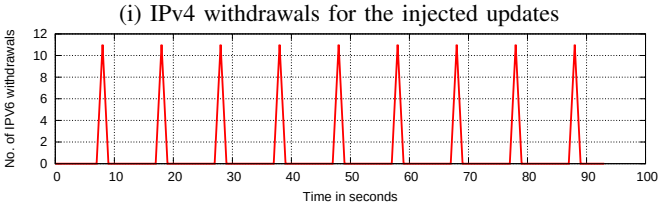
(h) IPv6 announcements for the collected updates



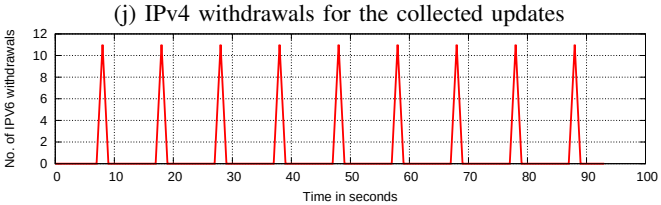
(i) IPv4 withdrawals for the injected updates



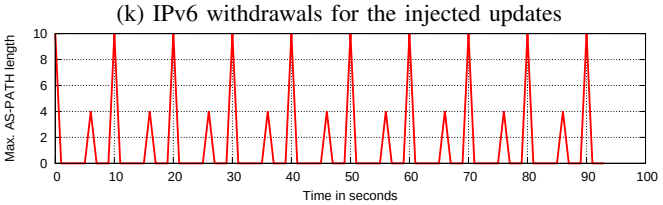
(j) IPv4 withdrawals for the collected updates



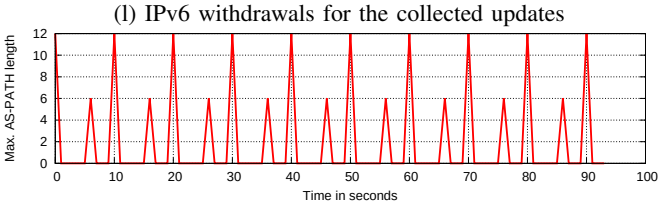
(k) IPv6 withdrawals for the injected updates



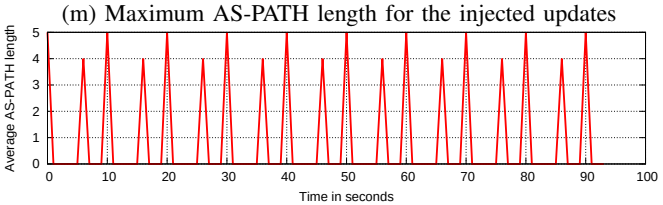
(l) IPv6 withdrawals for the collected updates



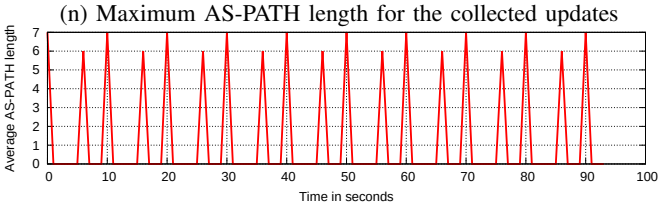
(m) Maximum AS-PATH length for the injected updates



(n) Maximum AS-PATH length for the collected updates



(o) Average AS-PATH length for the injected updates



(p) Average AS-PATH length for the collected updates

Figure 11: BGP features for the injected and collected data of experiment-1 using Cisco routers IOSv 15.1(4)M10

Table IV: Quagga 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 output directory 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 172.16.2.1	Set router ID 172.16.2.1 to the RRC
bgp log-neighbor-changes	Enable logging of BGP neighbor status changes (up or down)
neighbor 172.16.2.2 remote-as 65001	Set 172.16.2.2 as a peer AS65002
neighbor 172.16.2.2 filter-list 20 out	Do not send back BGP updates to BRT.
address-family ipv6	Configure IPv6 BGP
neighbor 172.16.2.2 activate	Activate172.16.2.2 peer to use IPv6 updates
exit-address-family	Finish IPv6 BGP configurations
ip as-path access-list 20 deny .*	Applies the filter-list 20 to all addresses

available within BRT v0.2 package. All the calculated features for the injected and collected BGP updates are identical except those related to AS-PATH as a result of increasing the number of hops (AS65001 and AS40).

B. Replay past BGP event

In this experiment we emulate the TMnet event, an example of BGP instability incident observed on the 12th of June 2015 by Telekom Malaysia (TMnet) which caused significant network problems for the global routing system [2]. We use BGP updates downloaded from route-views4 in the RouteViews during TMnet event. During the events, there were 31 peers connected to route-views4 in the RouteViews. We recall our simple topology shown in Figure 7 to replay 9001 seconds (around 2.5 hours) of BGP updates collected from the peer AS2914, one of the most peers that sent BGP updates during the event. Figure 12 shows BGP features for injected and collected BGP updates related to TMnet event. As shown in the figure, we can find a difference in the value of amplitudes for many BGP features. We do investigation to find if this difference as a result of unsynchronised clock, or time skew, between the two nodes [29] or a bug in the BRT. For that purpose, we enable a debugging message which notify users if BRT spends more than one second for a series of BGP updates with same time stamp. During the period of injected TMnet data (9001 seconds), BRT shows its ability to send all BGP updates with same time stamp within less than one second. Furthermore, RRC collected the same number of BGP updates which are by the BRT.

C. Known Issues

Although BRT v0.2 tool has been tested with Windows OS, we have occasionally experienced unknown errors during the implementation. However, BRT v0.2 shows very reliable and stable performance with Unix OS such as Debian, Ubuntu and FreeBSD.

BRT v0.2 has been tested to send different types and numbers of BGP updates using a desktop computer with 3 GHz Intel Core2 Duo CPU processor, 4GB memory, and 1Gpbs of network interface card. BRT v0.2 can send >15000 updates per second. However, this number may vary based on computer specification that uses BRT v0.2 and types of information in the BGP updates.

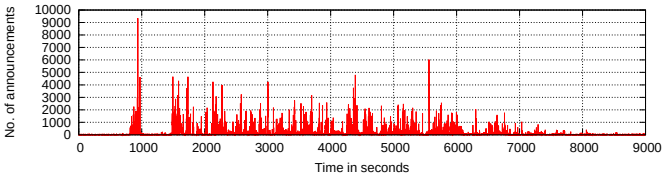
VI. Conclusions

BGP Replay Tool (BRT) version 0.2 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.

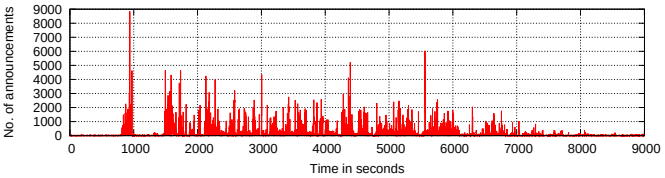
BRT v0.2 supports many BGP attributes such as community, AGGREGATOR, LOCAL-PREF, and MED. It also supports sending BGP updates of IPv6 and peering over IPv6. Furthermore, it supports connection to multiple peers. The evaluation of the BRT v0.2 has been implemented using Quagga, real Cisco routers, and VIRL as a testbed. Our future work will involve developing Net::BGP patch that supports IPv6 for listener mode. This help to avoid using RRC and enable real-time monitoring.

Acknowledgements

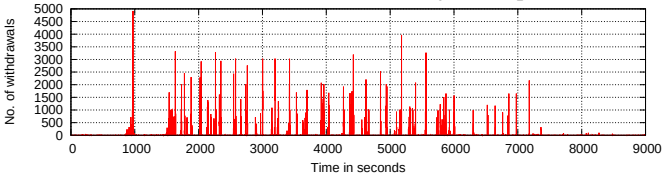
The development of BRT v0.2 has been made possible in part by "APNIC Internet Operations Research Grant" under the ISIF Asia 2016 grant scheme. We are also grateful to the VIRL team at Cisco for providing free license and support.



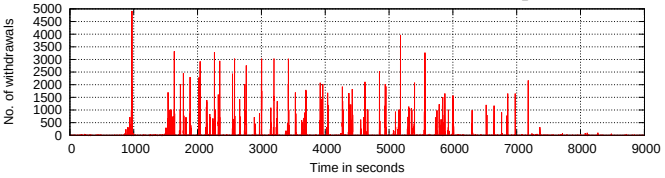
(a) BGP announcements for the injected updates



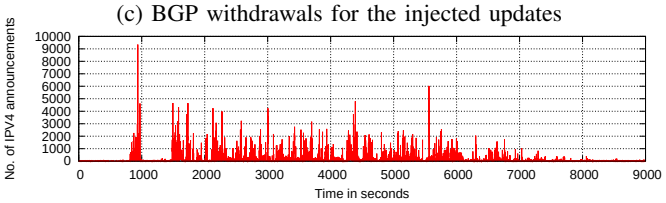
(b) BGP announcements for the collected updates



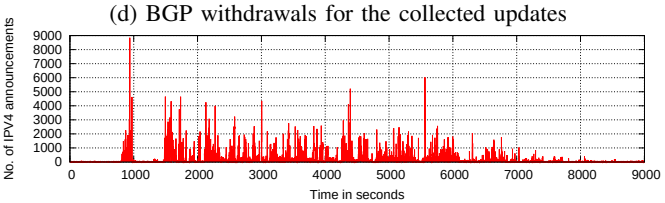
(c) BGP withdrawals for the injected updates



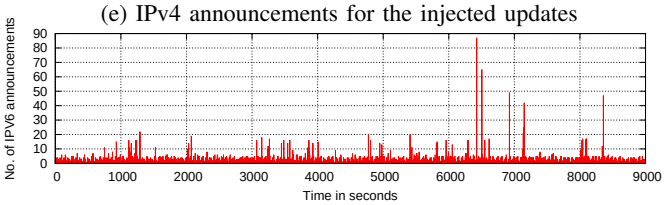
(d) BGP withdrawals for the collected updates



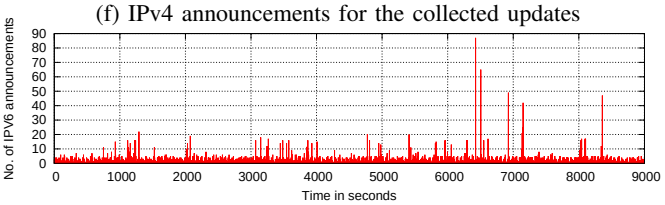
(e) IPv4 announcements for the injected updates



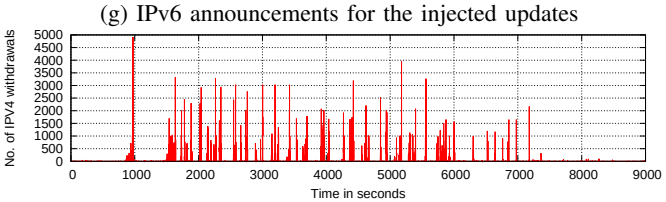
(f) IPv4 announcements for the collected updates



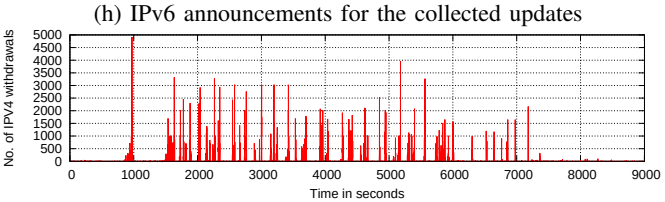
(g) IPv6 announcements for the injected updates



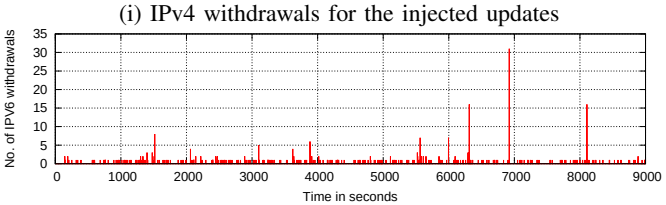
(h) IPv6 announcements for the collected updates



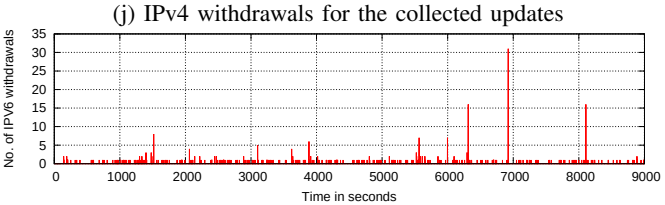
(i) IPv4 withdrawals for the injected updates



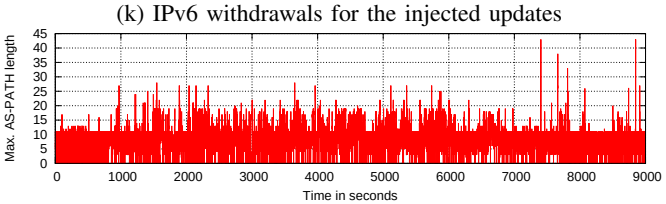
(j) IPv4 withdrawals for the collected updates



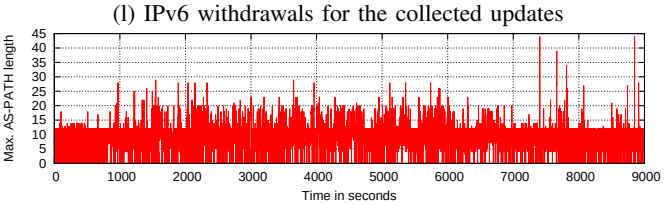
(k) IPv6 withdrawals for the injected updates



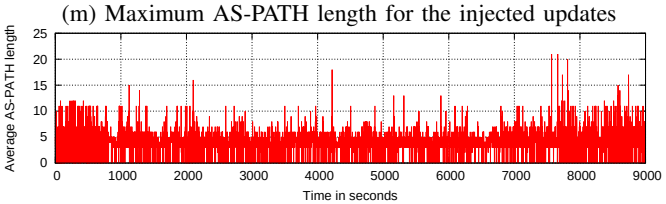
(l) IPv6 withdrawals for the collected updates



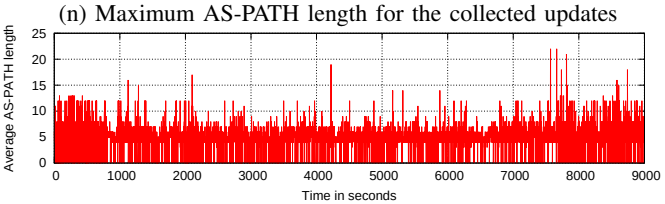
(m) Maximum AS-PATH length for the injected updates



(n) Maximum AS-PATH length for the collected updates



(o) Average AS-PATH length for the injected updates



(p) Average AS-PATH length for the collected updates

Figure 12: BGP features for the injected and collected for the TMnet incident

REFERENCES

[1] B. Al-Musawi, P. Branch, and G. Armitage, "BGP Anomaly Detection Techniques: A Survey," *IEEE Communications Surveys Tutorials*, vol. 19, no. 1, pp. 377–396, Firstquarter 2017.

[2] B. Al-Musawi, P. Branch, and G. Armitage, "Detecting BGP instability using Recurrence Quantification Analysis (RQA)," in *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*, Dec 2015, pp. 1–8.

[3] 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.

[4] 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

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

[6] 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

[7] R. Al-Saadi, "BGP Replay Tool (BRT) v0.2," May 2017. [Online]. Available: http://caia.swin.edu.au/tools/bgp/brt/brt-0.2.tgz

[8] B. Al-Musawi, P. Branch, and G. Armitage, "BGP Replay Tool (BRT) v0.1," Centre for Advanced Internet Architectures, Swinburne University of Technology, Melbourne, Australia, Tech. Rep. 160304A, 04 March 2016. [Online]. Available: http://caia.swin.edu.au/reports/160304A/CAIA-TR-160304A.pdf

[9] S. J. Scheck, "Border Gateway Protocol version 4 speaker/listener librar," September 2013. [Online]. Available: http://search.cpan.org/~sscheck/Net-BGP-0.16/lib/Net/BGP.pm

[10] T. Bates, R. Chandra, D. Katz, and Y. Rekhter, "Multiprotocol Extensions for BGP-4," RFC 4760 (Draft Standard), Internet Engineering Task Force, January 2007. [Online]. Available: https://tools.ietf.org/html/rfc4760

[11] J. Mitchell, "Autonomous System (AS) Reservation for Private Use," RFC 6996 (Best Current Practice), Internet Engineering Task Force, July 2013. [Online]. Available: http://tools.ietf.org/html/rfc6996

[12] 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

[13] Internet Assinged Number Authority (IANA), "Autonomous System (AS) Numbers," July 2014. [Online]. Available: http://www.iana.org/assignments/as-numbers/as-numbers.xhtml

[14] V. Fuller and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan," RFC 4632 (Best Current Practice), Internet Engineering Task Force, August 2006. [Online]. Available: http://tools.ietf.org/html/rfc4632

[15] Y. Rekhter, T. Li, and S. Hares, "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

[16] E. Chen and J. Yuan, "Autonomous-System-Wide Unique BGP Identifier for BGP-4," RFC 6286 (Proposed Standard), Internet Engineering Task Force, June 2011. [Online]. Available: http://tools.ietf.org/html/rfc6286

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

[18] J. Karlin, S. Forrest, and J. Rexford, "Pretty Good BGP: Improving BGP by Cautiously Adopting Routes," in *Network Protocols, 2006. ICNP '06. Proceedings of the 2006 14th IEEE International Conference on*, Nov 2006, pp. 290–299.

[19] 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

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

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

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

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

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

[25] CAIDA, "IPv6 announcement support patch for the Net::BGP perl modules," February 2016. [Online]. Available: https://github.com/CAIDA/bgp-hackathon/tree/master/bgpd-3

[26] D. Walton, E. Chen, and J. Scudder, "Advertisement of Multiple Paths in BGP," RFC 7911, Internet Engineering Task Force, July 2016. [Online]. Available: http://www.ietf.org/rfc/rfc7911.txt

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

[28] 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.

[29] S. B. Moon, P. Skelly, and D. Towsley, "Estimation and removal of clock skew from network delay measurements," in *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1, Mar 1999, pp. 227–234 vol.1.