BGPmon: Towards a More Robust, Fault-Tolerant Border Gateway Protocol (BGP) Monitoring System and Preliminary IPv6 Reachability Results

Miguel Cazares Jason Bartlett Cathie Olschanowsky Dan Massey
Texas State University *Colorado State University* *Colorado State University* *Colorado State University*
mc1512@txstate.edu bartletj@cs.colostate.edu cathie@cs.colostate.edu massey@cs.colostate.edu

***Abstract*—Society relies heavily on the Internet for a large portion of communication, business, and entertainment. However, there are issues with reachability within the Internet, e.g., when one Internet subnet suddenly cannot reach another subnet. Such issues can arise from malicious attacks or misconfigurations. Detecting these problems is the first step to combating large-scale unreachable Internet space.**

Thus, BGPmon, an Internet monitoring system developed at Colorado State University is introduced that addresses detection of these Internet issues. Further reachability analysis is conducted in a unique study spanning 20 days that addresses IPv6 reachability issues. This study utilizes data captured from a similar Internet monitoring system deployed at the University of Oregon (the Oregon RouteViews Project). Together, BGPmon and the 20-day IPv6 study seek to increase knowledge about Internet monitoring.

I. Introduction

The Internet and all of the functionality that depends on it requires a set of rules and policies in the form of protocols. One such protocol, the Border Gateway Protocol (BGP)[1], enables large independent networks within the Internet to connect to each other. However, BGP is susceptible to malicious attacks, such as prefix hijacking. Defending from these attacks requires that internet operators monitor BGP traffic and analyze the data. To fill this need, we introduce BGPmon - a real-time, scalable, free, and open-source BGP monitoring tool that enables operators and researchers to monitor and analyze BGP routing data[4].

II. Background and Related Work

An IP (Internet Protocol) address can be either 32 bits long (IP version 4) or 128 bits long (IP version 6). A prefix is a collection of IP addresses, also known as a subnet (subnetwork). A single prefix describes a network. Routing to a prefix involves maintaining a BGP routing table and announcing (as well as processing) changes to connections to neighboring networks.s A prefix hijack occurs when a route is falsely announced between Autonomous Systems (ASes) causing neighboring ASes to redirect traffic to the hijacker AS. On April 8th 2010, China Telecom announced 37,000 unique prefixes. This caused very large service outages across the globe because legitimate traffic to numerous ASes was re-routed to China Telecom. Another prefix hijack occurred in 2008 where Youtube traffic was re-routed to an AS in Pakistan. The high amount of traffic overwhelmed the Pakistani AS, causing a Denial-of-Service attack. As a result, Youtube became unreachable from most of the Internet. These hijacks illustrate the necessity of monitoring Internet routing data (BGP data).

Existing BGP data collectors such as RouteViews[3] and RIPE RIS[2] do not provide data in real-time. RouteViews uses only the monitoring subset of the Quagga software system. This software system's main design principle is to fully implement a BGP router with functionalities such as route selection, packet forwarding, sending BGP messages, and maintaining a BGP routing table. The latter is a significant issue for performance because a full BGP routing table at the time of writing contains more than 350,000 prefixes. Quagga is not designed to focus on BGP monitoring, thus using Quagga for this purpose yields low performance.

III. Scalable, Real-time Monitoring

Effective attack detection requires that a monitoring system scale to cover a large portion of the Internet. The coverage should include ASes that are both numerous and geographically distant. This enables the dataset to be much larger in volume and will be more useful to accurate analysis and mitigation of attack. The system must also be able to provide this data in real-time.

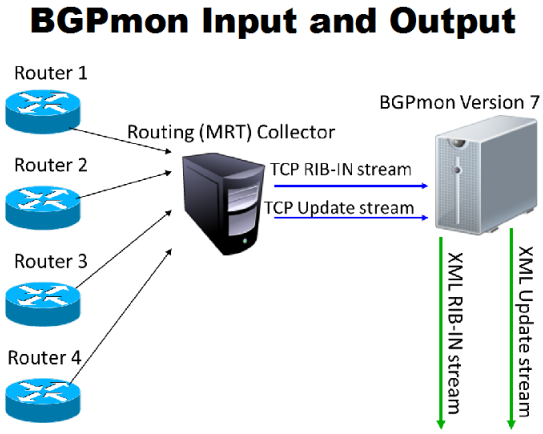

Fig. 1. An MRT Collector takes multiple peer inputs, then outputs messages and routing tables as TCP/IP streams to BGPmon.

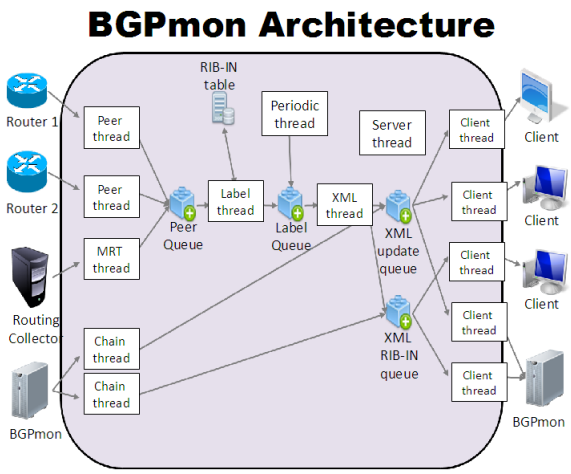

Fig. 2. Three supported input types go into BGPmon's queues and then are pushed out to clients.

Usefulness of data is directly related to how soon it can be accessed for handling and correction of attacks.

In addition, a monitoring system must be robust and fault-tolerant to handle corrupt data input. The approach described in the next sections focuses primarily on increasing monitoring system fault-tolerance and robustness (i.e. ability to handle corrupted and/or incomplete input at unknown points in time during execution).

IV. BGPmon Design

BGPmon utilizes a publish-subscribe model to achieve real-time data delivery[5]. In this model, there exist three entities: publishers, subscribers, and brokers. Publishers are the router peers (direct or MRT), subscribers are clients that connect to BGPmon to receive a live XML stream of data, and brokers are BGPmon systems that deliver this stream of data. The XML format contains representation of BGP data in human-readable format and also contains the original BGP data in machine-readable format. BGPmon outputs two streams of BGP data: XML RIB-IN Stream and XML Update Stream. The first stream's data describes the full BGP routing table of connected ASes. The second stream's data describes any changes to the BGP routing tables of connected ASes. The streams are independent and a subscriber can receive one or the other or both via a telnet connection[6].

BGPmon generates these streams by pushing data through queues so that clients can pull this data from the stream. The system also implements queuing and pacing algorithms to handle slow and fast readers of this data. Scalability is achieved mainly through chaining. Chaining of two BGPmons is defined as one BGPmon receiving XML output of another BGPmon. BGPmon systems can be arbitrarily chained together to distribute services and span a larger subset of BGP traffic. Chaining requires very little internal BGPmon processing[9] - XML data is not processed and is merely pushed out of the system as output. The system does, however, have to keep track of origination of XML data. This tracking is used to prevent XML loops. A loop occurs when a BGPmon sends out its XML data and another BGPmon (or sets of BGPmons) sends this data back through another chain to the originating BGPmon. This data is not new data and should be treated as previously sent data.

V. Implementation and Evaluation

The 7.2 release of BGPmon included the addition of corrupt message handling of Multi-threaded Routing Toolkit (MRT) collector input. The corruption could either occur due to faulty configuration of the collector or incorrect parsing and processing of MRT data. The former indicates actual corrupt data while the latter indicates merely possibly corrupt data. A 5-minute capture of MRT data was collected and stored in a file. The data was then sent to BGPmon to take as input so that possibly corrupt data could be identified. There were 5 possibly corrupt messages in this capture. The messages were then parsed manually and were subsequently identified as actually corrupt.

The ability to send real-time data was evaluated by a client system that read and processed the output XML stream of BGPmon. For each peer, MRT collector, and chain, several attributes were stored that are contained

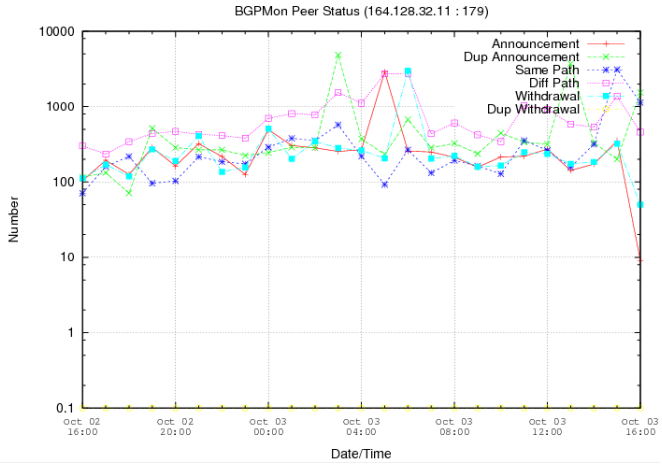

Fig. 3. Output of Statistics Web Client displays 6 BGP message types for a peer. Note the logarithmic scale for Y axis.

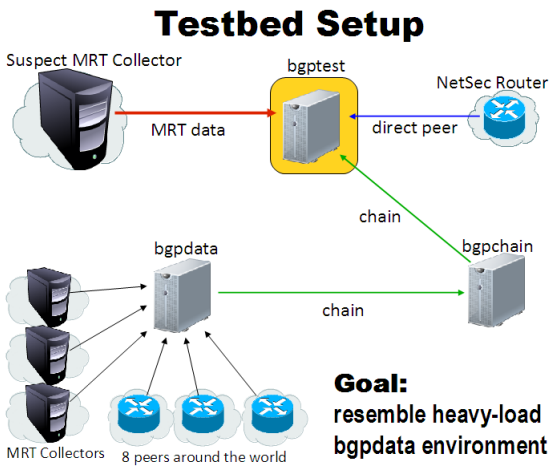

Fig. 4. BGPmon testbed allowed testing and modification of code while production servers remained online.

in a 24-hour sliding window; the collection of data was written as a web page for easy accessibility and visualization. A graph is generated every two minutes displaying magnitude of 6 message types for each peer within the past 24 hours. These message types are announcement, duplicate announcement, withdrawal, duplicate withdrawal, same path, and different path messages. In a typical day, BGPmon receives and processes more than 2 million update messages from direct BGP peers and more than 10 million update messages from MRT peers. This is done while consuming an average of 6.42 GB of memory.

The BGPmon team set up a live testbed. The goal of the testbed is first to allow testing on a non-production BGPmon. Second, the testbed should closely resemble the high data input of the main production BGPmon, bgpdata. The overall approach is to set the testbed up to receive all three types of BGPmon input: chain, direct peer, and MRT inputs. The chain in the testbed sent XML output from bgpchain to bgptest (the test BGPmon server). The direct peer was a router on the Colorado State University subnet (different subnet than bgptest's NetSec subnet) that announced a prefix periodically. The final input came from the suspect MRT collector physically located in Sao Paulo, Brazil. For the duration of testing, MRT data from this collector was disconnected from the production bgpdata and reconnected to the bgptest instance. This allowed BGPmon subscribers to continue to receive most of the BGP data output from bgpdata.

VI. Preliminary IPv6 Reachability Results

The work on IPv6 reachability extends previous work on IPv4 reachability[8] and static reachability analysis[7]. The main reason for exploring reachability differences is to explore and explain situations where the global (BGP) routing table of one Internet Service Provider (ISP) differs from the routing table of another ISP. Ideally, Internet service provided by one ISP should allow a subscriber to reach the same Internet space as a subscriber from another ISP. The underlying question is whether Internet coverage provided from one ISP differs from other ISPs. And if the coverage does differ, what are the reasons for this difference in reachability? When attempting to answer these questions, one must be careful to properly characterize differences between ISP BGP routing table contents. One potential pitfall is to prematurely conclude that a smaller number of entries in a routing table indicates less reachability.

While this can be the case, it is not necessarily always the case. For example, ISP A can have 25 less prefixes in its routing table than ISP B. Reachability provided by these ISPs will be equivalent if ISP A has a number of prefixes that still span the space that its 25 less prefixes would have spanned. This example illustrates that number of prefixes does not indicate the span of those same prefixes. That is, a single prefix of arbitrary length will still be counted as a prefix. So a prefix that spans a large portion of the Internet will be counted the same as a prefix that spans a significantly smaller portion of the Internet. Thus, further investigation requires deeper analysis. The approach described in the next section furthers the analysis by characterizing types of prefixes and subsequently filtering out misleading information such as number of prefixes. The goal is

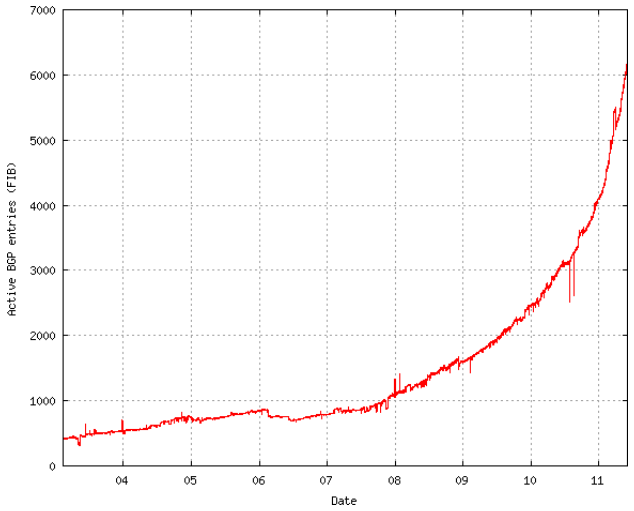

Fig. 5. The global IPv6 BGP routing table growth has accelerated significantly over the last two years.

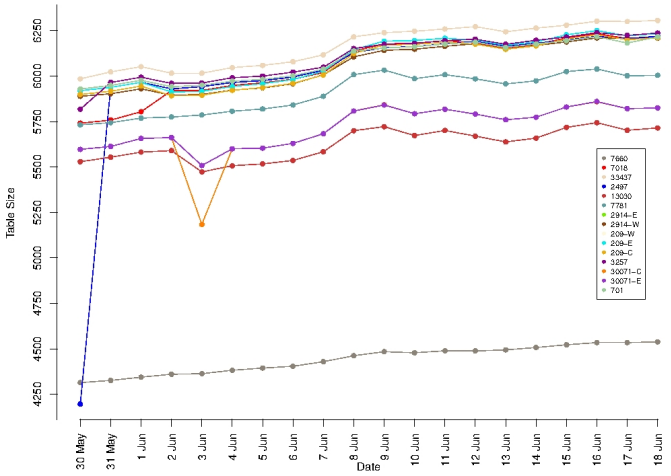

Fig. 6. Table sizes captured at noon every day during 20-day collection period.

to gain a more accurate understanding of reachability differences between ISPs.

VII. IPv6 Reachability Methodology and Analysis

To discuss IPv6 reachability, four result types were subsequently filtered and identitfied. The following analyzes IPv6 reachability through these four result types. The first type, labeled Type 1 data, describes prefixes that were both in Hurricane Electric's BGP routing table and a chosen AS. Type 1 prefixes represent no loss in reachability. The second type, Type 2 data, indicates prefixes that are covered by a larger(less precise) prefix in Hurricane Electric's routing table. These data, therefore, also show no loss in reachability.

Type 3 data indicate prefixes that are covered by smaller (more precise) prefixes in Hurricane Electric's table. Unfortunately, over the 20-day study period, we found no prefixes in any peer's table that was fully covered by smaller, more precise, prefixes in Hurricane Electric's table. Finally, Type 4 prefixes indicate unreachable prefixes. Any peer's prefix that did not into any of the previous 3 groups is considered to be unreachable by Hurricane Electric. Since this study merely showcases preliminary IPv6 reachability results, Type 4 is what rouses the most interest. We found that before IPv6 Day, the peers had around 200 prefixes that were not in Hurricane Electric's table. After IPv6 Data, the number of unreachable prefixes decreased slightly, near 160 unreachable prefixes. The four figures shown in this study demonstrate the overall trend discussed in this section.

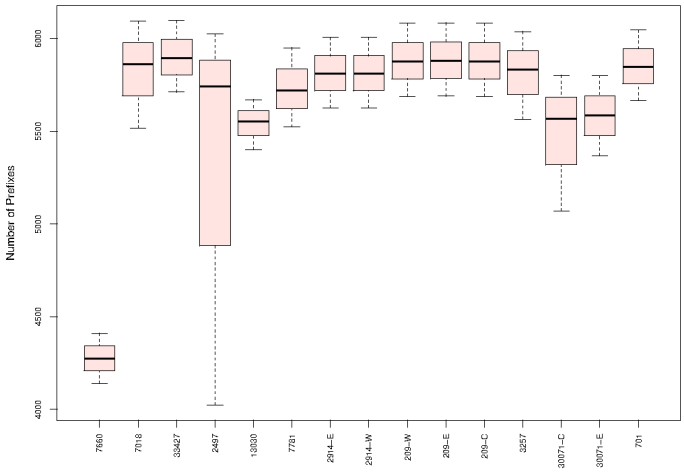

Fig. 7. Type 1 data are prefixes contained in both Hurricane Electric's routing table and the specified AS (by AS Number).

VIII. Conclusion and Future Work

A real-time and scalable BGP monitoring system was presented that enables continuous monitoring of worldwide Internet routing traffic. BGPmon achieves scalable and real-time data through chaining and publish-subscribe models. BGPmon successfully recovers from corrupt BGP messages and handles a high number of BGP update messages. This system can be used by Internet operators for real-time data analysis and by Internet researchers to better understand how Internet traffic is routed on a global scale. Thus, BGPmon enables

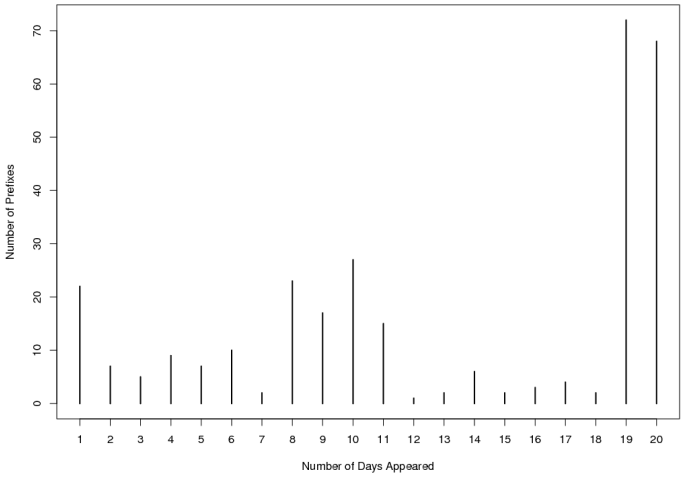

Fig. 8. Shows persistence of Type 4 data over the study, i.e., number of prefixes unreachable by Hurricane Electric.

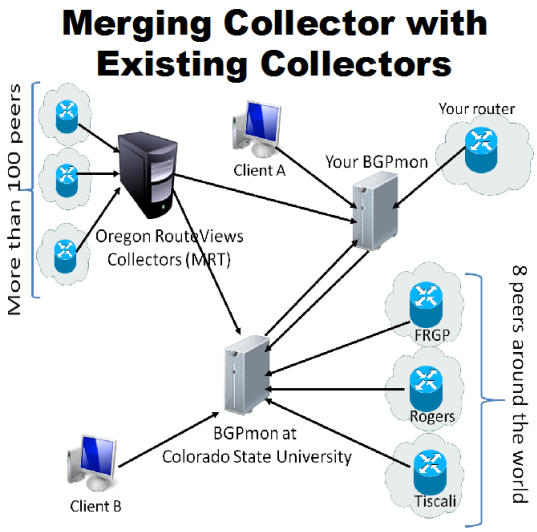

Fig. 9. A BGPmon instance can be easily deployed to add to the project's data set or to monitor own subnetwork.

a strong defense against prefix hijack attacks.

There are a three general areas of future work for BGPmon. These include data storage, performance analysis, and code maintenance. The first area, data storage, deals with deployment of a BGPmon archiver client that can read production servers' ouput and archive this XML data to permanent storage on the web for easy access. This will allow the same useful data as BGP data provided by the Oregon RouteViews project, but with the added benefit of data tagging provided by BGPmon that will allow easier traffic, data, and prefix hijack analysis. The second area, performance analysis, seeks to implement BGPmon performance metrics to measure general statistics on BGPmon performance, such as BGPmon uptime, percentage of Internet space coverage, and amount of memory and CPU used (on a production server). This will help to analyze how useful BGPmon is in its efforts to provide scalable, real-time data to clients interested in analyzing BGP events as they unfold. Finally, the third area, code maintenance, deals with BGPmon internal code and porting it to an object-oriented programming style for increased modularity and code reuse. Since BGPmon solves several problems while implementing real-time data delivery, such as queuing and pacing models, other projects will benefit from reusing this code for a separate or similar implementation.

Future work for IPv6 reachability analysis requires a better understanding of how exactly IPv6 to IPv4 and IPv4 to IPv6 transitionary protocols impact global IPv6 BGP routing tables. Protocols of concern are: 4-to-6 (IPv4 to IPv6), 6-to-4 (IPv6 to IPv4), and Teredo. The underlying question here that must be addressed is "how exactly are these protocols handled in the IPv6 global BGP routing table?". Further analysis must, therefore, make a step back in analysis of foundations before moving forward. Once this basic question is addressed, then we can proceed forward to start re-analyzing the data we captured during our 20-day period around World IPv6 Day.

References

[1] A Border Gateway Protocol 4 (BGP-4). http://www.ietf.org/rfc/rfc4271.txt.

[2] RIPE routing information service project. http://www.ripe.net/.

[3] RouteViews routing table archive. http://www.routeviews.org/.

[4] BGPmon: Using Real-Time Data in Research and Operations. http://bgpmon.netsec.colostate.edu/download/doc/BGPmon-deployment.pdf, 2010.

[5] D. Matthews, N. Parrish, H. Yan, and D. Massey. BGPmon: A real-time, scalable, extensible monitoring system. In *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC)*, 2008.

[6] D. Matthews, H. Yan, and D. Massey. BGPmon Administrator's Reference Manual. http://bgpmon.netsec.colostate.edu/download/doc/arm.pdf, 2008.

[7] G. G. Xie, J. Zhan, D. A. Maltz, H. Zhang, A. Greenberg, G. Hjalmtysson, and J. Rexford. On static reachability analysis of ip networks. In *in Proc. IEEE INFOCOM*, 2005.

[8] H. Yan, B. Say, B. Sheridan, D. Oko, C. Papadopoulos, D. Pei, and D. Massey. IP Reachability Differences: Myths and Realities.

[9] H. Yan, M. Strizhov, K. Burnett, D. Matthews, and D. Massey. BGPmon Version 7 Implementation and Technical Specification. http://bgpmon.netsec.colostate.edu/download/doc/techreport.pdf, 2010.