Department of Information and Communication Engineering
Graduate School of Information Science and Technology
THE UNIVERSITY OF TOKYO

Master Thesis

Hunting BGP Zombies in the Wild

(実環境 BGP Zombies の検知手法およびその特徴に関する研究)

Porapat Ongkanchana
48-196414

Supervisor: Professor Hiroshi Esaki

January 2021

Abstract

As the key component of Internet's inter-domain routing, BGP is expected to work flawlessly. However, a recent study has revealed the presence of BGP zombies, that is withdrawn prefixes that are still active in routing tables and that can cause routing issues. While this past finding was enabled through the usage of experimental prefixes with scheduled withdrawals (BGP beacons), it did not accurately represent the status of the Internet. In this study, we aim at detecting BGP zombies for any prefixes announced on the Internet. To that end we study characteristics of withdrawn messages, and devise a method to differentiate withdraw messages corresponding to local topological changes to those standing for prefixes withdrawn by their origin AS. Based on this classification we study the occurrence of zombies in the wild across six years of BGP data. We found over 6.5 millions zombies, among those we have confirmed that 94% reported incoherent path and caused 468 potential routing loops. Our study also reveals that noisy prefixes, long AS paths, and ASes announcing a large number of prefixes are more prone to zombies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Border Gateway Protocol (BGP) is the inter-domain routing protocol of the Internet. Routers all across the world exchange reachability information using BGP, and it is of the utmost importance that these operations are correctly executed. However, BGP has no mechanisms to ensure the integrity of exchanged information which makes BGP vulnerable to different types of attack and mishaps [1, 2, 3, 4], and also discrepancies, such as BGP zombies [5].

Also known as stuck route or ghost route, BGP zombies can emerge when an AS stops announcing an IP prefix. Ideally, an AS sends withdrawal messages to all its peers and the messages propagate throughout the Internet triggering the removal of corresponding entries in all routers' Routing Information Base (RIB). However, a recent study [5] has shown that this basic BGP operation sometimes fails and causes BGP zombies, that is active RIB entries for withdrawn prefixes.

Despite this significant contribution to the research community, this past study is focusing solely on a few experimental prefixes that are withdrawn at scheduled time slots (BGP beacons [6, 7]) and omits all other prefixes that are in use on the Internet. As network operators have subsequently reported issues caused by BGP zombies [5], including a few outages and hijacks [8], an analysis of BGP zombies in the wild is needed to better understand the extent of BGP zombies on the Internet.

In this thesis we aim to quantify the impact of BGP zombies on the numerous prefixes in use on the Internet. First, we study characteristics of BGP withdrawal messages and devise a technique to differentiate messages caused by local topological changes to those representing prefixes withdrawn from the origin AS. Then, we use this technique to detect BGP zombies in the wild and quantify the extent of BGP zombies for any prefix in use from 2014.

BGP zombies are an ongoing trouble that does not properly receive attention. We hope that this work would help motivate others to pursue this issue in this field. We believe this study can assist network operators to improve the integrity of the Internet. Our main contributions are:

- We analyze 6 years of historical BGP data to characterize and classify different types of withdrawal messages.
- We devise a simple technique to detect zombies for any prefix (as opposed to beacon prefixes [5]) which can be easily implemented by network operators.
- Using this detection technique, we identify BGP zombies across 6 years of BGP

data and document their characteristics as well as their impact on popular Internet services.
- We report a total of 6.5M BGP zombies (i.e. pair <BGP peer, prefix>), 88% of which are IPv4 prefixes.
- We validate the detection method by checking for incoherence of AS path between route collector peers and comparing the result with past research.
- We show that zombies are common and also observed for popular content providers.
- We uncover how AS routing characteristics, such as number of announced prefixes, average path length, and number of update messages can contribute to the emergence of zombies.
- We observe 468 potential routing loops and 77k detours caused by BGP zombies.

The remainder of this thesis is organized as follows: First, we lay the basic knowledge of BGP in chapter 2, followed by the data and tool used in this research (chapter 3). In chapter 4, we define the terminology for BGP zombies and introduce related works on the field. We describe the zombies detection algorithm along with the reasons behind in chapter 5. We then apply the algorithm and analyze the detected BGP zombies' characteristics in chapter 6. Finally, we discuss and conclude this study in chapter 7 and 8.

Chapter 2

Border Gateway Protocol

The Border Gateway Protocol (BGP) is at the core of the Internet’ s interdomain communication. Any network wishing to join the Internet needs to know how to speak BGP. For readers to fully appreciate our work of BGP zombies, it is crucial that they have a deep understanding of how the protocol works. This chapter’ s purpose is to provide enough comprehensible information on interdomain routing and BGP, the key points include:

- Autonomous system
- Prefixes and AS Path
- BGP update message
- Pitfalls of BGP

2.1 Autonomous System

> “A global computer network providing a variety of information and communication facilities, **consisting of interconnected networks using standardized communication protocols**”
>
> The definition of internet from Oxford Languages

What makes the internet standout compared to other innovations, is that it is truly the product of collaborations. The Internet is not a single entity of one big giant network owned by one company, it is a connected network formed by various companies and organizations. We refer to these individual networks as the Autonomous System, also known as AS. ASN or AS number is an identifier used to specify a network. For example, the University of Tokyo has an AS number of 2501, its network would be called AS2501.

Within each AS, the owner can design and configure their network to suit their needs without having to care about the outside. For example, the owners can configure their routers to forward packets by hard coding the path into the routing table, or they can apply any intradomain routing protocols to figure out the path dynamically. However, when you wish to connect with the rest of the world you need to speak the international language of BGP.

Normally at the edges of each AS, there would be routers capable of communicating the interdomain routing protocol. These routers are called Peers. Packet forwarding to

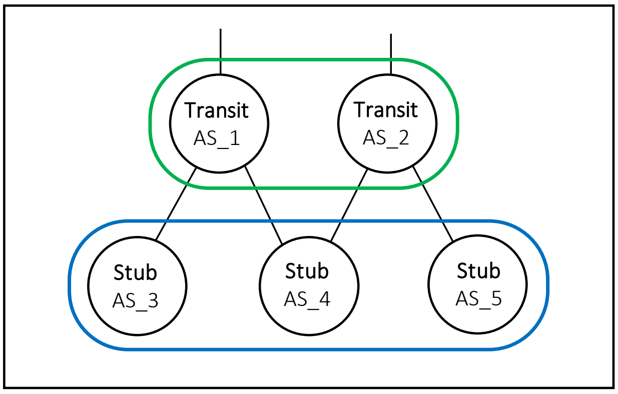


Fig. 2.1: Stub ASes have their traffic carried by transit ASes

this AS from outside or going out from this AS would pass through these routers.

Autonomous system can be categorized into 2 types: Stub AS and Transit AS [9] (figure 2.1).

- **Stub AS** is an AS which does not forward others’ packet. All packets going out originate within the AS, and all packets going in are destined to addresses in the AS. Networks of organizations, small companies or content providers are usually stub ASes.
- **Transit AS** is one which helps forwarding others’ packets. Internet service providers, such as NTT or Level3, are usually transit ASes.

Currently, most communications between ASes are conducted over Border Gateway Protocol version 4. In the next section, we discuss what information is propagated inside the BGP message.

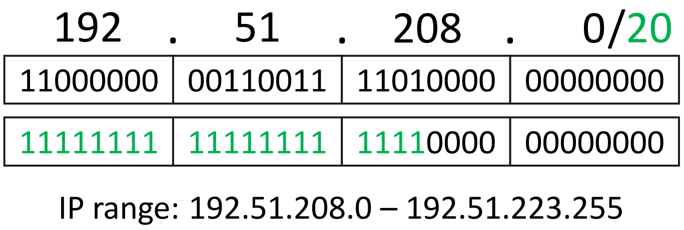
2.2 Prefixes and AS Path

BGP’ s main role is to exchange the network’s **reachability information** between ASes. The main components such information includes networks (Prefixes) which can be reached through AS and how the packets would be forwarded (AS Paths). BGP then utilizes the exchanged info to construct the routing table.

2.2.1 Prefixes

Prefixes is a batch of IP addresses. They are represented in the form of a network IP address followed by a netmask. Figure 2.2, shows an example of University of Tokyo’ s IPv4 prefix and its corresponding IP addresses. These prefixes are assigned to and owned by a company or an organization. An AS which owns the prefix is called **origin AS**. The owners of each prefix are responsible to manage and propagate it by themselves. If owners want to open their IPs to the rest of the world, it would be their responsibility to announce it using BGP. We would get back to how announcing and withdrawing a prefix works later in this chapter (§2.3).

Netmask is an integer between 0 and 32 ($0 \leq nm \leq 32$). A prefix would contain all IP



192 .	51 .	208 .	0/20
11000000	00110011	11010000	00000000
11111111	11111111	11110000	00000000

IP range: 192.51.208.0 – 192.51.223.255

Fig. 2.2: University of Tokyo’ s IPv4 prefix and its corresponding IP addresses

addresses with the same leading nm bits. Going back to the example of UTokyo’ s IPv4 prefixes, with netmask equal to 20, all IP addresses starting with `1100000.00110011.1101` (converted from `192.51.208`) are a member of this prefix. As expected, the number of addresses in a prefix can also be calculated using its netmask, $2^{(32-nm)}$.

2.2.2 AS_PATH

AS Paths is a path of autonomous systems. The path represents the series of AS numbers which packets would be passed along to reach the destination prefix. As mentioned before, BGP is a path vector protocol, when it propagates the routing information, it would forward both the prefixes (destination) and AS path (route) to its neighbor.

Figure 2.3, illustrates the networks of 5 ASes connected by BGP, vertices are ASes and edges are the connections between them. Prefix P_1 is located in AS1, assuming that all the ASes know how to forward packets to prefix P_1. Next, considering the scenario where AS5 wants to send packets to the P_1, this can achieve that by either passing the packets along `5-3-2-1` AS path or `5-4-1` AS path.

BGP utilized AS path for 2 main purposes: Finding the best path and Performing cycle detection.

- **Finding the best path** is rather rudimentary. When there are more than 2 AS paths to the destination prefix, BGP simply selects the path with the shortest length as the best path and installs it in the routing table.
- **Cycle detection** in undirected graphs has complexity of O(V+E), which is quite heavy considering that BGP have to keep track of all prefixes. Therefore, BGP uses a simplified version of detection method. When a router receives a BGP message with its own AS in the AS path, BGP would identify that looping has occurred and drop the message

In the next section, we would look into BGP’s actual operation.

2.3 BGP update message

The BGPv4 is defined in RFC 4271 [10]. Two BGP routers establish their connection over a TCP session and exchange routing info, known as BGP update message. BGP updates are mainly used for 2 purposes: announcing and withdrawing a prefix.

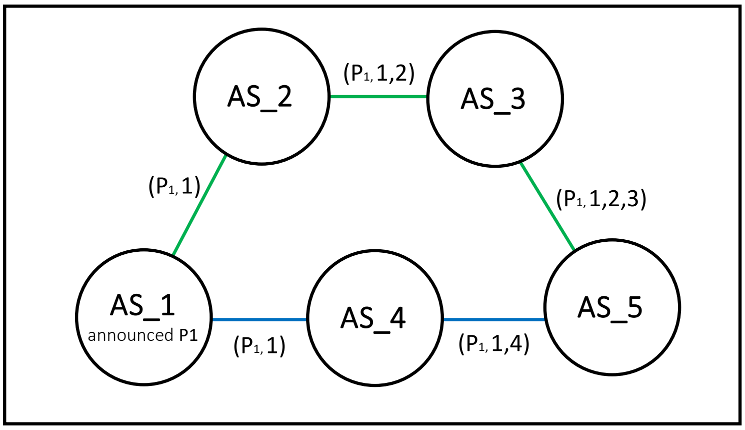


Fig. 2.3: An example of prefix P_1' s AS paths from AS5.

Prefix announcing message is used when a router wants to inform its neighbors about a path to prefixes it knows. In contrast, when prefix' s path has become unreachable, the router would notify its neighbours by sending out the withdrawing message. Next, we look into some examples of how BGP actually operates in practice.

1. BGP announcement
2. BGP withdrawal and path hunting
3. Longest prefix match
4. Path prepending

2.3.1 BGP Announcement

In the network topology of figure 2.3, the following example examines the propagation of BGP announcement messages for prefix P_1 first announced by AS1.

1. AS1 announces (P_1, "AS1") to its neighbors AS2, AS4.
2. AS2, AS4 receive BGP updates from AS1 and now know how to reach P_1. AS2 and AS4 send out BGP updates (P_1, "AS2 AS1") and (P_1, "AS4 AS1") respectively to their neighbors.
3. AS3 receives an update from AS2, and conveys (P_1, "AS3 AS2 AS1") out to AS5.
4. (P_1, "AS4 AS1") and (P_1, "AS3 AS2 AS1") both arrive at AS5.
5. BGP installs the shortest path which is (P_1, "AS4 AS1") to its routing table. Keep in mind that BGP does not discard the longer path of (P_1, "AS3 AS2 AS1") despite not using it.
6. All ASes know how to forward packets to P_1.

The BGP announcement is fairly elementary; when there are many routes available, choose the shortest one.

2.3.2 BGP Withdrawal and Path Hunting

Prefix withdrawal message is sent when a prefix is no longer reachable or the origin prefix decides to stop announcing it. In this section, we look at both scenarios: a withdrawal caused by connection failure between AS1 and AS4 and a case where AS1, prefix P_1 origin AS, withdraws the prefix itself.

AS1-AS4 connection failure

1. A link between AS1 and AS4 has become unreachable.
2. AS4 confirms that its TCP session with AS1 has lost, and removes (P_1, "AS1") from its routing table.
3. AS4 starts sending BGP withdrawal messages to its neighbor.
4. AS5 receives the withdrawal message and removes (P_1, "AS4 AS1").
5. However, since AS5 still has paths left, it installs the next best path (P_1, "AS3 AS2 AS1").
6. AS5 notifies its neighbors about its new route.
7. AS4 receive new route and install P_1 new routes, (P_1, "AS5 AS3 AS2 AS1")
8. All ASes know how to forward packets to P_1.

As can observe from the above example, even when the network failure occurs, every AS still has a method of accessing the destination prefix. When the link AS1-AS4 has recovered, AS4 would receive a shorter AS path from AS1 and reinstall it, while propagating new paths to AS5, restoring to the stable environment.

P_1 prefix withdrawal

1. AS1 withdraws its prefix P_1 by sending withdrawing message to both AS2, AS4
2. Withdrawal message arrives at AS4. AS4 removes its (P_1, "AS1") path and forwards a withdrawal message to A5.
3. AS5 receives a withdrawal from AS4, removes its best route (P_1, "AS4 AS1") and installs the second best route of (P_1, "AS3 AS2 AS1").
4. Withdrawal message arrives at AS2. AS2 removes its (P_1, "AS1") path and forwards a withdrawal message to A33.
5. Similarly, AS3 uninstalls its route and forwards withdrawal to AS5.
6. AS5 receives a withdrawal from AS3, removes its route (P_1, "AS3 AS2 AS1"), it no longer has any routes left. AS5 continues sending withdrawal messages to its neighbor.
7. No AS has paths to P_1.

Prefix withdrawal becomes more complex when many paths are available. Even when the best path to the prefix was removed from the router, it won' t propagate the withdrawal message if there are still any paths left. A fault tolerant AS normally has more than 1 upstream provider, which means for this AS to remove its path, it has to wait until all of its providers declare so. The operation of removing all available paths is called "Path Hunting" . This operation takes time, it can be only a few milliseconds in single provider topology, or a matter of minutes in well defined topology.

2.3.3 Longest Prefix Match

When there are more than 2 different paths available which can both forward packets to the destination, routers would choose the route with the longest prefix match. In other words, the router would choose the more specific path. Considering the example where a router has 2 paths installed in its routing table: (`10.1.0.0/16`, "AS4 AS2 AS1") and (`10.1.1.0/24`, "AS5 AS3 AS1"). In this case, a packet destined for IP address `10.1.0.1.1` is complied with both rules. Meaning either path is taking the packet to the destination. However, since the router chooses the longest prefix match, or the more specific one, it would forward packets using (`10.1.1.0/24`, "AS5 AS3 AS1").

2.3.4 Path prepending

Path prepending is a announcing strategy used to manipulate the network traffic. Since BGP always chooses the shortest path, AS can append its AS number in the AS path multiple times to construct a longer AS path. Back to figure 2.3, considering a situation where AS4 has its network capacity fully utilized and does not wish to carry additional AS5 traffic. AS4 can announce (P_1, "AS4 AS4 AS4 AS1") to AS5, consequently AS5 would install a shorter path of (P_1, "AS3 AS2 AS1") in its routing table.

2.4 BGP Pitfall

Although BGPv4 is the most dominant inter domain protocol for over 20 years, few improvements to none have been applied. BGP is very susceptible to malicious activity [11], since it was designed in the era of peace in the internet history. It has no way of authenticating messages received, no method of protection against peer imposter, message modification, man-in-the-middle attack and denial-of-service attack.

In this section, we look into problems that the BGP community has encountered, and how network operators mitigate such problems.

- BGP hijack
- Route leak

2.4.1 BGP hijack

BGP hijack is an illegitimate attempt to take over a prefix. As mentioned before, BGP lacks methods of authentication the incoming routes. An imposter can hijack victim' s traffic by announcing a shorter route to the same prefix, or exploiting the longest prefix match and announcing the sub prefixes (example: imposter can hijack `10.10.0.0/24` by announcing `10.10.0.0/25` and `10.10.128.0/25`).

The aftermath of the BGP hijacking is quite severe. In Feb 2008, an unauthorised announcement of Youtube prefix by Pakistan Telecom (AS17557) rendered Youtube to become inaccessible for over 2 hours in the global scale [12]. A report from Military Cyber Affairs in 2018 [13], revealed that China telecom has performed various suspicious BGP activities manipulating other nations' government website' s traffic. Many attempts

[14, 15, 16, 4] are conducted to detect and mitigate the hijacking problems. Nevertheless without changing the BGP itself, it seems like network operators will continue facing the problems.

2.4.2 Route leak

RFC7908 [17] states that route leak is "**The propagation of routing announcement(s) beyond their intended scope**. That is, an announcement from an Autonomous System (AS) of a learned BGP route to another AS is in violation of the intended policies of the receiver, the sender, and/or one of the ASes along the preceding AS path"

In August 2017, Google, one of the world's largest service providers, made a configuration mistake and announced (leaked) its peers' route [18, 19]. This mishap affected NTT OCN the most, with Google announcing 25,000 of its prefixes, NTT became unreachable for over 40 minutes. While KDDI and IIJ' s route were not announced by Google, their traffic was detoured to Google' s network in Chicago, causing a significant delay.

In this section, various flaws of BGP have been revealed. Despite BGP being an old protocol with no real way of mitigating its problems, changing the Internet' s infrastructure protocol requires a tremendous effort. Considering most if not all interdomain routing is conducted over BGP, it is an undeniable truth that BGP will continue its ruling. It is our best effort to try understanding BGP better and find ways of working around its flaws.

Chapter 3

About BGP Data

In this chapter, we explore several aspects of conducting research with BGP data. The topic include: the **BGP datasource** used in modern research, **Analysis tools** mitigating nausea of working with BGP data and the **difficulties of conducting BGP research**.

3.1 Datasource

Route Collector (RC) [20] is a server connected with routers from multiple ASes at the Internet Exchange, collecting BGP updates and RIB. Internet Exchange Point (IXP) is a physical location where various Internet service provider’ s (ISP) and Content Delivery Network’ s (CDN) Autonomous systems are connected together.

Route Collector would periodically dump their collected BGP data to remote servers. The dumped data would then be processed, compressed and published online. Updates are normally uploaded at a refined interval (unit of minutes), while RIB’ s publish interval is much longer (hours).

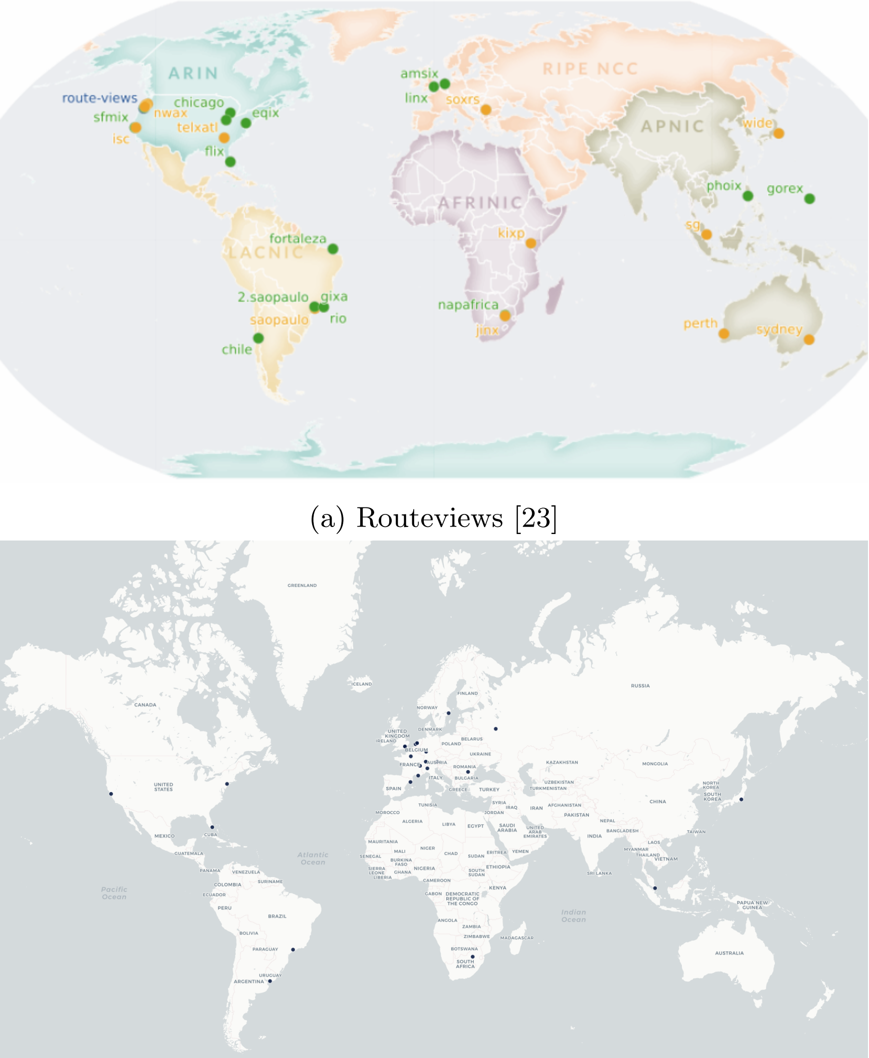
While there are over 20 BGP data providers at service, the 2 most used providers with the highest number of Route Collectors are Routeviews [21] and RIS RIPE [22]. Both BGP data archives have collected routes from over thousands peers spanned across IXP in 5 continents. A brief comparison between 2 main data is shown in table 3.1.

3.2 Analyzing Tools

This section introduces tools and API associated with conducting BGP research.

Table 3.1: The comparison between Routeviews and RIS RIPE.

	Routeviews	**RIS RIPE**
RIB Interval	2 hours	8 hours
Updates Interval	15 minutes	5 minutes
#RCs	30	20
#Peers	600	1,300
Oldest dump	October 2001	September 1999



(a) Routeviews [23]

(b) RIPE RIS [24]

Fig. 3.1: Maps demonstrate the location of route collectors

3.2.1 BGPStream

BGPstream [25] is a seamless tool for retrieving and analyzing routing data. It works with only Routeviews and RIS RIPE. Instead of directly downloading and managing BGP data manually, researchers can use BGPstream' s brokers to only select a particular set of BGP data they need. Filter options such as selecting BGP data from specific RC, selecting only for specific periods, selecting data of certain prefixes and more. Apart from that, BGPstream also has LIVE mode, where it will continually download new data whenever it becomes available.

3.2.2 RIPEstat

RIPEstat[26] is an API service provided by RIPE. While RIPEstat is calculated using only BGP data from RIPE RIS, it provides various meaningful statistical data on prefixes, ASes, country and host. For example, users can query API data for all updates of a prefix, a process which if calculated manually requires downloading all million prefixes data from all RIS RIPE route collectors, looking into each record and filtering out all unwanted data.

3.2.3 BGP Beacon

BGP Beacon [7] refers to a global visibility prefix with scheduled announcements (every 4 hours) and withdrawals (2 hour after it was announced). Even though beacon is designed to be used as an active measurement for prefix convergence delays and investigating the impact of route flap damping, with its clear data interface, beacon data can also be used for other studies as well ([5]).

3.3 Difficulty of analyzing BGP data

The difficulty of analyzing BGP data rises solely by the sheer amount of it. With the number of prefixes increasing every year the routing table size exploded. According to data published by APNIC [27], IPv4’ s routing entries have grown from 530,000 entries in 2015 to 814,000 entries in 2020. It is projected that IPv4 entries will increase to over 1M by 2024, despite IPv4 almost completely exhausted [28]. While IPv6 does not have the same drastic number as IPv4, 21,000 entries in 2015 and 79,000 in 2020, it is still concerning, considering its growth rate of almost 4 times in 5 years. Moreover, when IPv4 is completely depleted, all new networks would be all allocated with IPv6. This statistic demonstrates that if researchers want to do the full analysis on all IP addresses, they can expect to run almost twice the data they did 5 years ago.

Another difficulty faced when analyzing BGP data is in the nature of BGP data itself. Everytime researchers want to get any BGP attributes from a router, they need to download the closest RIB and replays all BGP updates up to the point of time. The process requires a significant amount of computation compared to the simplicity of the query. While BGPStat is an effective tool dealing with this problem, it fails when analysis on a huge amount of data is necessary.

Chapter 4

BGP Zombies

BGP zombies, also known as stuck routes or ghost routes, refer to an active Routing Information Base (RIB) entry for a prefix which is already withdrawn by its origin AS. In other words, it is a route to the non existing destination. The name "zombies" came from the fact that routers with BGP zombies exhibit the behavior of causing their peers to be infected as well.

This chapter is organized in the following way; we start the first section by introducing some terminologies, we then discuss prior research along with emphasizing the improvement and differences. Finally, we finish this chapter by describing the scope of this research.

4.1 Terminologies

- **BGP Zombies**, routing entry to the withdrawn prefix.
- **Zombie Path**, an AS path of the BGP zombie.
- **Zombie Peer**, a router which contains BGP zombies in its routing table.
- **Zombie AS**, an AS with zombie peers.
- **Zombie Outbreak**, a group of BGP zombies for the same prefix, occuring at approximately the same time.
- **Outbreak size**, the number of BGP zombies in a zombie outbreak.

4.2 Related Work

Fontugne et al. [5] have conducted the first thorough investigation on BGP zombies. They have confirmed the existence of BGP zombies, while providing in depth analysis on various characteristics. They showed that zombies are not uncommon and even large transit ASes can be affected by BGP zombies. However, their experiments are carried out using only RIPE' s RIS BGP beacon prefixes.

While conducting research on beacon prefixes is beneficial due to their controlled nature, RIS beacon prefixes do not represent the state of the Internet. The internet is a wild place, it is constantly changing and evolving. **The main purpose of our study is to extend previous research and investigate the characteristics of BGP zombies in the wild**. There is fundamental differences between the two studies:

Table 4.1: Comparison between previous research and current one

	This research	**PAM** [5]
#IP addresses	$\sim 1M$	27
Traffic to prefixes	Yes	No
	Wild Prefix	**Beacon Prefixes**
Prefix characteristic	Unknown withdrawal time	Predetermined withdrawal time

1. The study with BGP beacons only examined a small portion of prefixes. RIS are currently operating 18 IPv4 and 21 IPv6 beacon prefixes [6], which only occupy 0.0022% and 0.0266% of total IPv4 and IPv6 space respectively. Since the number of regular prefixes is several orders of magnitude higher than beacon prefixes, only scalable methods can be considered.
2. A beacon schedules its prefix’ s withdrawal periodically, however such a characteristic is not presented in regular prefixes. The withdrawal in the real environment is unpredictable, an owner of prefixes may decide to remove its prefix just to reduce a traffic load or a prefix may become unreachable because connection between 2 peers is lost. Since our study aims to generalize the idea of BGP zombies, we look into most prefixes announced on the Internet. This implies that we need to devise a technique to detect when a prefix is withdrawn by its origin AS (Chapter 5).
3. Beacon prefixes accommodate no host and no traffic, thus the impact of zombies for beacons is limited and harder to observe. Network operators have however reported that zombies for regular prefixes may trigger customer complaints [5] and outages [8].

The differences between two studies are summarized in table 4.1.

4.3 The Scope of this Study

This study answers two questions: How to detect wild BGP zombies (“Hunting Wild Zombies” , chapter 5) and What are the characteristics of zombies (“Zombies in the Wild” , chapter 6).

Chapter 5

Hunting Wild Zombies

This chapter discusses zombies detection algorithms for regular prefixes and the logic behind each design decision. Before driving into the detection itself, we first need to clarify about the BGP data used in this study and how it was processed to fit in with the experiment

5.1 Pre-experiment

5.1.1 BGP Data

In this study, we analyzed historical BGP data collected by RIPE RIS routing registry, starting from January 2014 to December 2019. Each month, we select 10 days, from 10th to 20th, of BGP data from route collectors which operated for at least half the measurement period. The route collectors are RRC00, RRC01, RRC03-07, RRC10-16, RRC18-21.

In order to ease data manipulation, we used only RIS data but based on the nature of route collector projects we do not expect the results to be significantly different, even if we use additional route collectors from other projects.

We have examined a total of 720 days of BGP data. Note that, we do not aggregate this data in any way, all 72 groups of 10 days data are analyzed separately. From this data set, we calculate the number of **active peers per prefix** ($A_p(t)$), the key metric for this study.

5.1.2 Preprocessing

Active peers $A_p(t)$ refers to the set of routers which were announcing the prefix p at time t. A router is added to set $A_p(t)$ if and only if the router' s most recent BGP update message, in prior to time t, was announcing a route to prefix p (i.e. not a withdrawal message).

The total number of active peers varies from one prefix to another. This is due to BGP peers being exposed to a different set of prefixes. In order to ease the study' s analysis, we computed the normalized number of active peers $n_p(t)$, such that $n_p(t) = \frac{|A_p(t)|}{max_i(|A_p(i)|)}$ and $max_i(|A_p(i)|)$ is the maximum number of active peers observed across a 10-day batch. This metric value ranges between 0 and 1 which respectively represents a prefix that is withdrawn (0) and a prefix globally reachable (1). We computed $n_p(t)$ for every prefix seen in the data set described above with a temporal granularity of 15 minutes.

For this study we filtered out locally reachable prefixes, that is prefixes consistently announced by a small number of peers, hence are not globally reachable. The definition of BGP zombies in this case is ill-defined because the propagation of these prefixes is intentionally limited. Moreover, from analysis point of view, the value of $n_p(t)$ can be greatly influenced with a slight change in number of active routers when $max_i(|A_p(i)|)$ is small, making $n_p(t)$ a non-uniform representation. We assumed that these locally reachable prefixes are never seen by a large number of peers thus we filtered out prefixes where $max_i(|A_p(i)|) < 100$. These local prefixes account for approximately 20% of all prefixes monitored in our data set.

5.2 Zombies Detection

BGP zombies emerge when a prefix is withdrawn but some routers fail to reflect the change. These are the two fundamental pieces of information we need in order to detect BGP zombies. While the routing table' s changes for RIPE RIS peers is directly available in our dataset, inferring prefix withdrawn in the wild is challenging for the four following reasons:

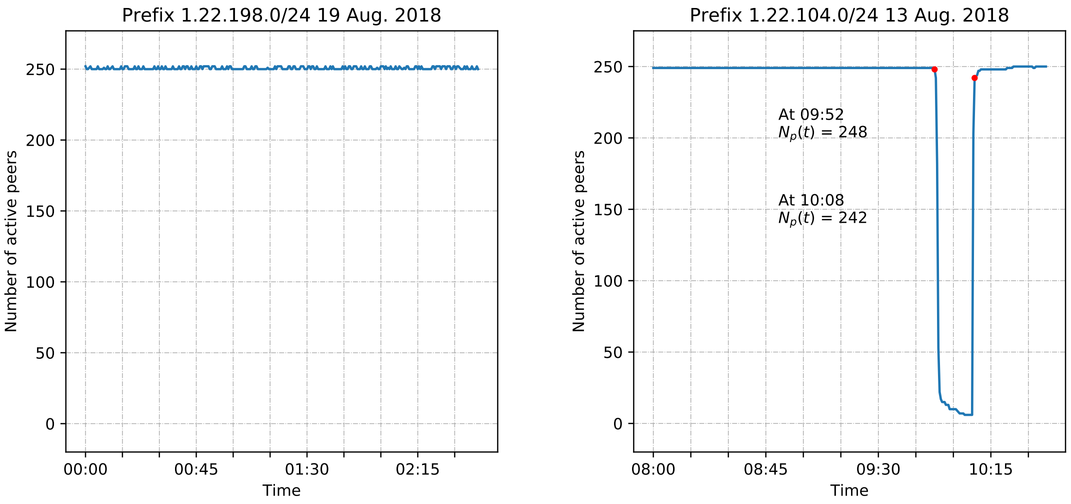
1. **Withdrawal can happen at any time.** Unlike beacon prefixes, regular prefixes are independently managed by their origin AS and can be withdrawn at any given points of time.
2. **Withdrawal propagation time is varying and unpredictable.** Past research [7, 29] has shown that the propagation time of withdrawal messages is significantly fluctuating. These variations are mainly due to path hunting and noise reduction techniques (e.g. MRAI and Route Flap Damping [30, 29, 31, 32, 33]) and are hardly predictable.
3. **Local topological changes.** Withdraw messages are observed in the case of local changes, although the origin AS has not withdrawn the prefix. This is, for example, due to reconfigurations of networks between RIS peers and the monitored prefix.
4. $n_p(t)$ **is not null when zombies emerge.** By definition, zombie peers have an active entry for a withdrawn prefix, thus the number of active peers $n_p(t)$ is not always dropping to 0 when the prefix is withdrawn by its originating AS.

5.2.1 Withdrawal Scope

Comprehending when a prefix is globally withdrawn and how long before BGP messages propagate throughout the Internet are the keys of our detection algorithm. The principles of the algorithm are simple, we make the assumption that a prefix p is withdrawn if its number of active peers, $n_p(t)$, has dropped below a certain value and stays low for an extended period of time.

Next, let' s consider a healthy prefix with most peers announcing its path. In other words, the value of $n_p(t)$ is closed to 1. Later, the value of $n_p(t)$ is dropping, the possible changing result are:

1. $n_p(t)$ **drops to zero:** This is the case of prefix withdrawal, the BGP messages have propagated throughout the entire network and all peers update to the latest change.
2. $n_p(t)$ **fluctuates then becomes stable again (~ 1):** This is the case of Topological change, as we have discussed briefly on the topic in §2.3.2 "AS1-AS4 connection failure" . The total number of active peers shifts, when routes to the prefix have changed. However, since the prefix is not withdrawn, it is expected that all peers would install a new route to prefix and the value of $n_p(t)$ would rise back to 1 again. The topological change can be further categorized into 2 types: global and local topological change. Local change refers to a case where only a small number of ASes and peers are affected, hence we are expected to find $n_p(t)$ fluctuates slightly. On the other hand, in the global cases, changes occurred in multiple ASes, therefore the value of $n_p(t)$ should shift up and down sharply. See the examples of topological change in figure 5.1
3. $n_p(t)$ **drops but stabilizes at low value:** This is BGP zombies, the prefix is withdrawn but some peers do not reflect new change.



(a) local topological change

(b) global topological change

Fig. 5.1: The examples of topological change

Considering these 3 patterns, illustrated in figure 5.2, when detecting zombies we thus face the trade-off of setting a threshold value *thres* that is low enough to avoid most local topological changes and high enough to detect all zombies.

In order to select a suitable threshold value, we investigate the typical $n_p(t)$ drops that happen between two stable states (excluding complete withdraws where $n_p(t)$ reaches 0).

Here we define a stable state as a constant $n_p(t)$ value for more than one hour and compute the maximum $n_p(t)$ drop as follows:

- At $t = t_a$, the number of active peers $n_p(t_a)$ was constant for more than one hour (stable) and the value was above 0.9.

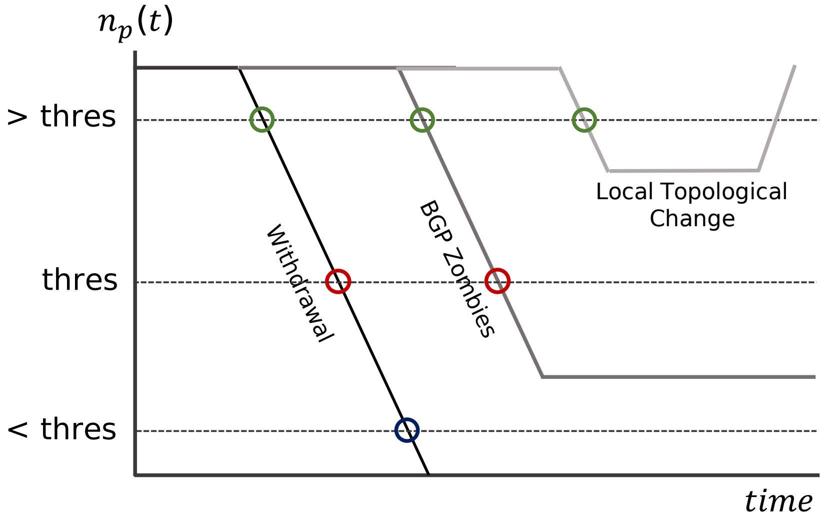


Fig. 5.2: The overview of how different threshold values would affect the withdrawal detection.

- At $t, t_a < t < t_b$, the number of active peers $n_p(t)$ has changed and continually shifted for more than one hour (unstable).
- At $t = t_b$, $n_p(t_b)$ becomes again stable.
- The maximum $n_p(t)$ drop is equal to $1 - min(n_p(t))$ where $t_a < t < t_b$ and $min(n_p(t)) \neq 0$

Figure 5.3 depicts the distribution of the maximum $n_p(t)$ drops observed in our data set. We can observe two typical ranges of values, (0,0.2) and (0.8,1). The smaller drops between 0 and 0.2 represent the local changes, while the larger ones between 0.8 and 1 represent potential BGP zombies. Based on these results, one should select a threshold value between 0.2 and 0.8. For this study, we arbitrarily set this threshold to 0.5, meaning that we ignore events that affect less than 50% of all observed peers for each prefix.

5.2.2 Withdrawal Propagation Time

Although the propagation time of prefix withdrawal is variable, we hypothesize that it is bounded by a certain duration T_w. We estimate T_w by looking at the time duration of $n_p(t)$ drops, that is $t_b - t_a$, necessary for prefix p to be completely withdrawn ($n_p(t_b) = 0$ and $n_p(t_a) > thres$).

Figure 5.4 shows that in our data set, for both IPv4 and IPv6, more than half of the prefixes are withdrawn within 15 minutes (our smallest time resolution). Only a small fraction of withdrawals completes in more than 90 minutes, hence for this study we discard withdrawals that last less than $T_w = 90min$ and look for zombies only in long lasting events.

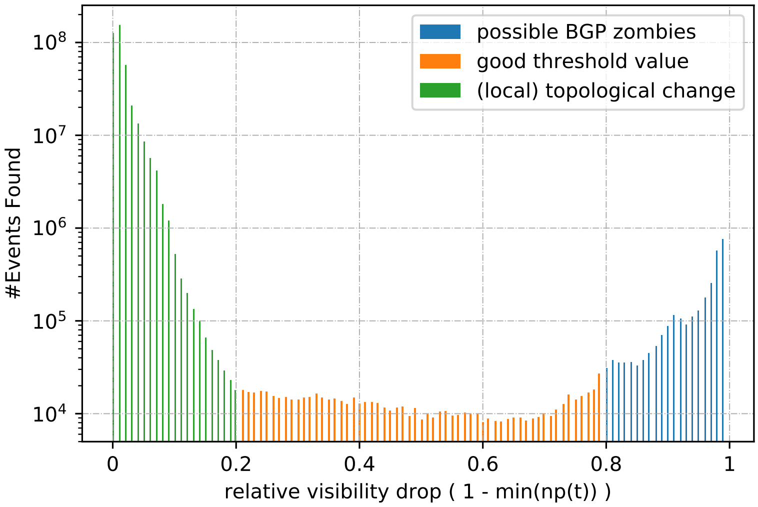


Fig. 5.3: Distribution of maximum $n_p(t)$ drop (i.e. $1 - min(n_p(t))$) during significant withdrawals.

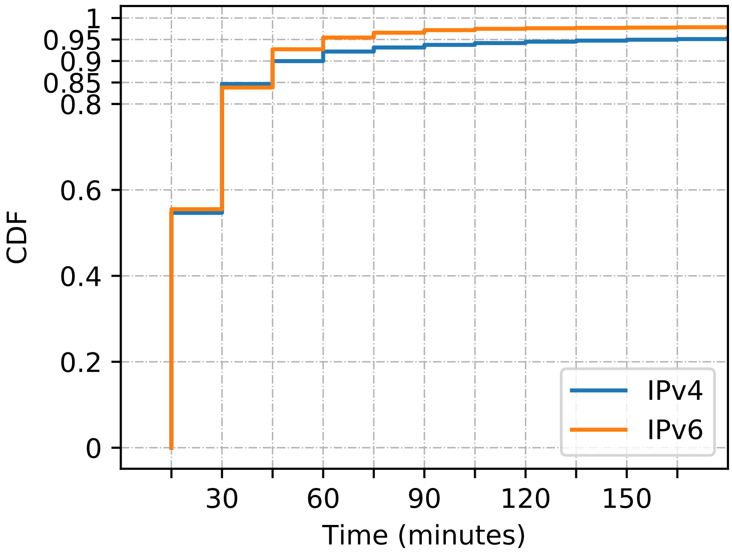


Fig. 5.4: Distribution of the time necessary for prefixes to be globally withdrawn.

5.2.3 Detection Algorithm

The observations above constitute the core of our BGP zombie detection algorithm. As illustrated in Figure 5.5, BGP zombies are reported at time t for prefixes that have an active number of peers $n_p(t-90)$ dropping below 0.5, but that is not reaching $n_p(t) = 0$ within the next 90 minutes.

BGP zombies are not reported if $n_p(t)$ is quickly going down to 0. In this case we

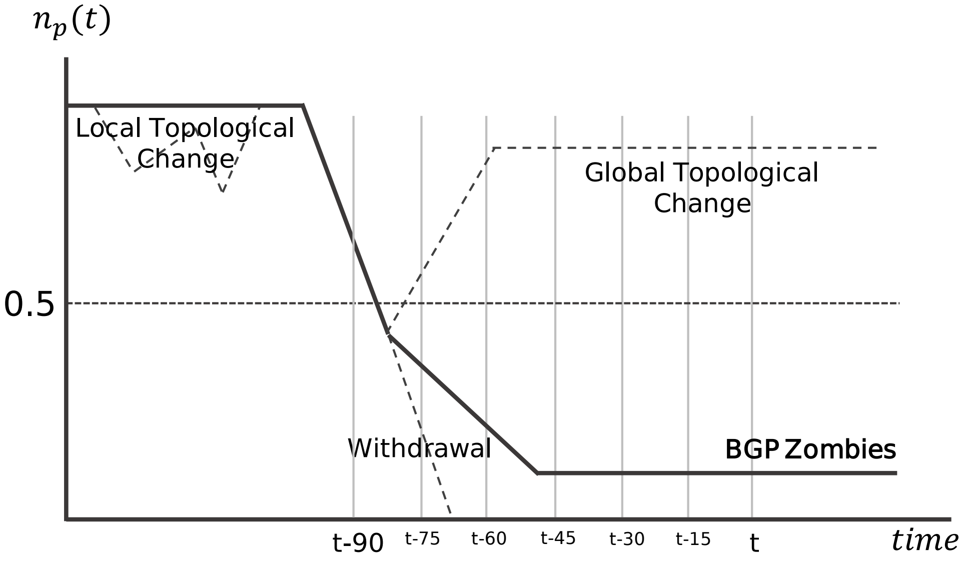


Fig. 5.5: Summary the zombie detection algorithm and corresponding thresholds.

infer that the prefix was successfully withdrawn by all RIS peers. Meanwhile, if $n_p(t)$ increases up to above 0.5, we classify this as a global topological change. This is a simple and practical method that has the advantages of being easily implementable by network operators and, as shown in the following, that provides efficient detection.

Chapter 6

Zombies in the Wild

By applying our zombie detection algorithm to the 6-years BGP data set, we found approximately a total of 6.5 millions BGP zombies (88% are for IPv4 prefixes) from 486k outbreaks.

In the following sections, we evaluate the proposed detection method by confirming the incoherence of AS paths between RIS peers during zombie outbreak (§6.1). We then examine zombies reported for BGP beacons (§6.2), and estimate the lifespan of BGP zombies (§6.3). After that, we compare zombie outbreaks for prefixes managed by popular content providers (§6.4) and reveal relationships between routing characteristics and zombie outbreaks (§6.5). We then investigate the detrimental effects of zombies on the routing infrastructure (§6.6). Finally, we conclude this study with the longitudinal analysis of BGP zombies (§6.7).

6.1 Path coherence between RIS peers

In BGP update messages the AS_PATH attribute indicates the preferred sequence of ASes for reaching a certain prefix. Routers implicitly trust the AS path received from their peers and utilize this information to select their preferred path and perform cyclic-detection. As routers prepend their own AS number to paths received from their peers, a prefix is expected to be known by all ASes along the AS path.

For BGP zombies, however, we expect that a zombie peer advertises an AS path including ASes that no longer have route to the corresponding prefix. These incoherent AS paths corroborate the presence of BGP zombies.

In order to check for the path coherence of a BGP zombie we need routing information from all ASes along the zombie path. In practice this comprehensive analysis is not possible with our data set, we can only investigate path coherence across RIS peers.

Investigating path coherence is straight forward:

1. We read all BGP updates of RIS peers and updated its status upto the time zombies were reported.
2. We then divided peer AS into 2 categories: withdrawal and announcement, based on peer' s routing status.
3. BGP zombies would then be labeled as either "Unknown" , "Coherence" or "Incoherence" . Zombie' s AS_PATH attribute excluding prefix origin AS and the peer AS are used to determine the category.

- **Unknown**: All ASes are not presented in either announcement nor withdrawal.
- **Incoherence**: Any ASes along the part are found in withdrawal group.
- **Coherence**: All ASes along the path are found in either announcement or unknown group.

31.3% of all detected BGP zombies are labeled as Unknown. However, for the remaining 68.7%, we observe over 94.7% of incoherent AS paths, thus verifying that these routing entries are indeed BGP zombies.

The rest 5.3% are coherent paths but the result does not dictate that these paths are valid, reachable paths. It could be due to several zombie RIS peers being presented along the paths.

For 99% of the coherent paths, the RIS peers are only one (80%) or two (19%) hops away, suggesting that they are likely to be part of the same zombie outbreak. More routing information from ASes closer to the origin AS is required to ensure the coherence of these paths.

Looking at zombies for prefixes originated by RIS peers (1.1% of all detected outbreaks) allows us to estimate the fraction of misclassified zombies. We found that 97.6% of these zombies are indeed withdrawn by their origin AS. The few cases (2.4%) where the origin AS has not withdrawn the prefix but the zombies were detected, illustrates our detection method’ s flaw that may rarely classify large topological changes as BGP zombies.

In summary, given that

- Path incoherence is never observed when prefixes are successfully withdrawn.
- 94.7% of detected zombie paths are provably incoherent
- Only a few detected paths are misclassified

We believe the proposed algorithm is effective for zombie detection.

6.2 Beacons and noisy prefixes

To validate our approach, we also compared our results with previous reports for RIS beacons and noticed interesting singularities for these prefixes. The 27 RIS beacon prefixes monitored in past research [5] accounts for 3.22% of all outbreaks detected in our dataset. This significant number of outbreaks for such a small number of prefixes, is clear evidence that a specific kind of prefixes are more susceptible to be affected by BGP zombies than others. While beacon prefixes possess various characteristics that are distinct from regular prefixes, one feature stands out. The amount of BGP updates produced by beacon is undeniably higher than others, considering its routine of withdrawal and announcement.

We thus investigate the relationship between the number of zombie outbreaks and the number of BGP update messages per prefix. To ease computation, we focus only on prefixes that have at least 10 outbreaks per 10-day measurement period in 2018 and 2019. Figure 6.1 shows that for each prefix, the number of outbreaks increases with the number of BGP update messages.

For IPv4, the Spearman correlation between these two quantities is $\rho = 0.6$ which confirms a non-negligible relationship between the number of update messages and the number of zombie outbreaks. In addition, we found that IPv4 beacons are quite outstanding in our results as they produce a lot more zombies than that of regular IPv4

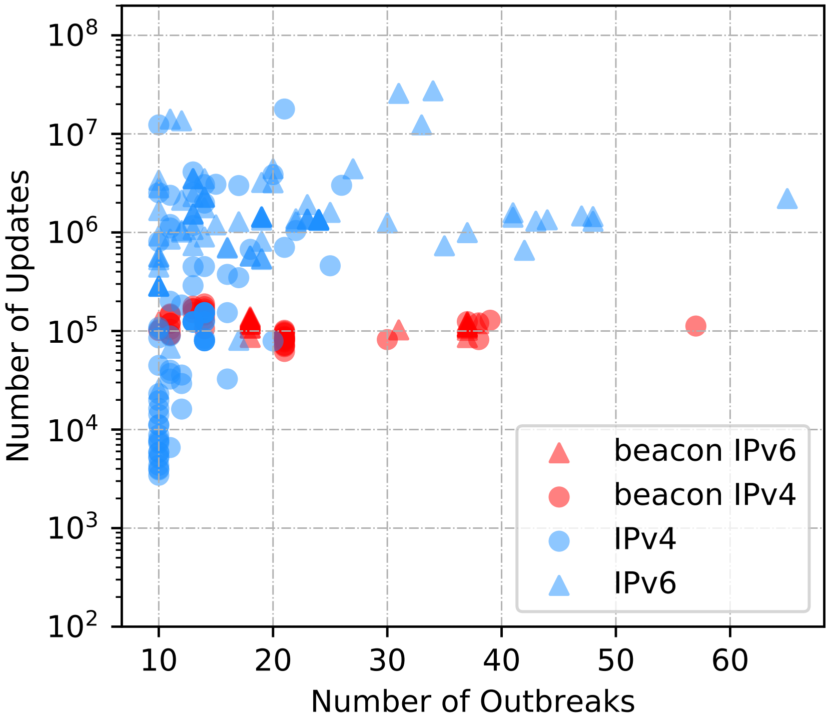


Fig. 6.1: Number of zombie outbreaks and number of update messages for prefixes with more than 10 outbreaks per 10-day measurement period (2018-2019).

prefixes.

In IPv6 case, while a lot of outbreaks have been observed in beacon prefixes, some regular prefixes significantly produce more updates with more zombie outbreaks. These kinds of overactive regular prefixes are occurring irregularly, only in some months, whereas the same beacon prefixes are seen in all measurement periods.

Given the small number of beacon prefixes and their frequent appearance in the most impacted prefixes, we argue that the recurring zombie outbreaks found for beacon prefixes is not representative of what we can expect for regular prefixes. This is an important point to keep in mind when interpreting results from past study [5].

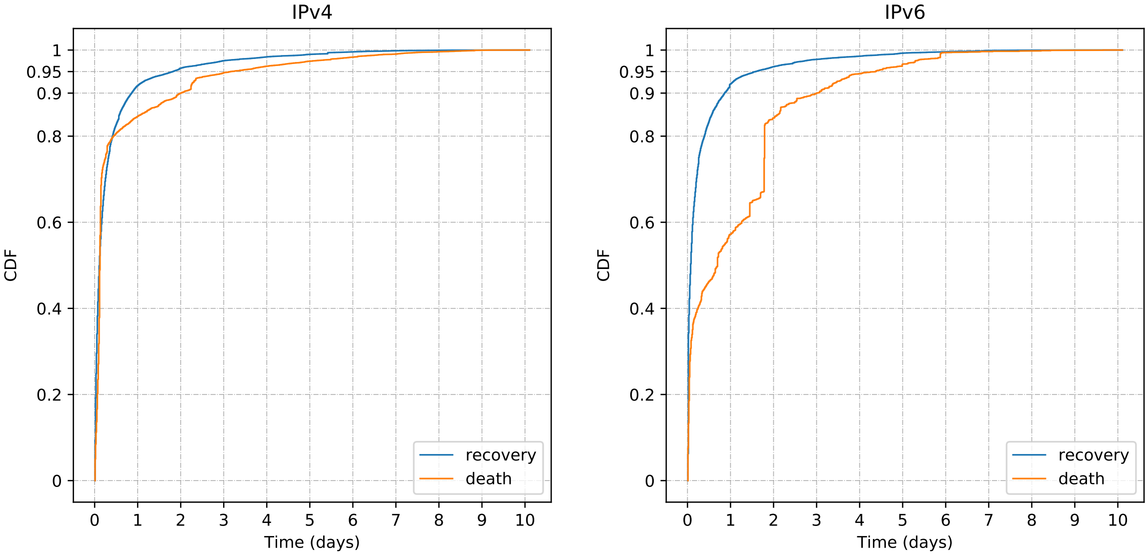
6.3 Zombie Lifespan

In this section, we investigate the lifespan of detected zombies, something that was not possible in past research because beacon prefixes are re-announced every 4 hours. We define the lifespan of BGP zombie outbreak as the period of time it takes before one of the following events happen:

- **Death:** The zombies disappear or no router is longer announcing the prefix ($n_p(t) = 0$).
- **Recovery:** The prefix is active again and over a half of the peers start announcing a route to the prefix ($n_p(t) >= 0.5$).

The recovery cases account for 89.91% of all observed BGP zombies, meaning that most BGP zombies are left unnoticed, until the prefixes are announced again. The death cases could be due to the late arrival of withdrawn messages, for example due to route flap dampening or BGP session resetting with zombie peers, either ways, the low percentage of death cases suggests that our T_w value is chosen appropriately.

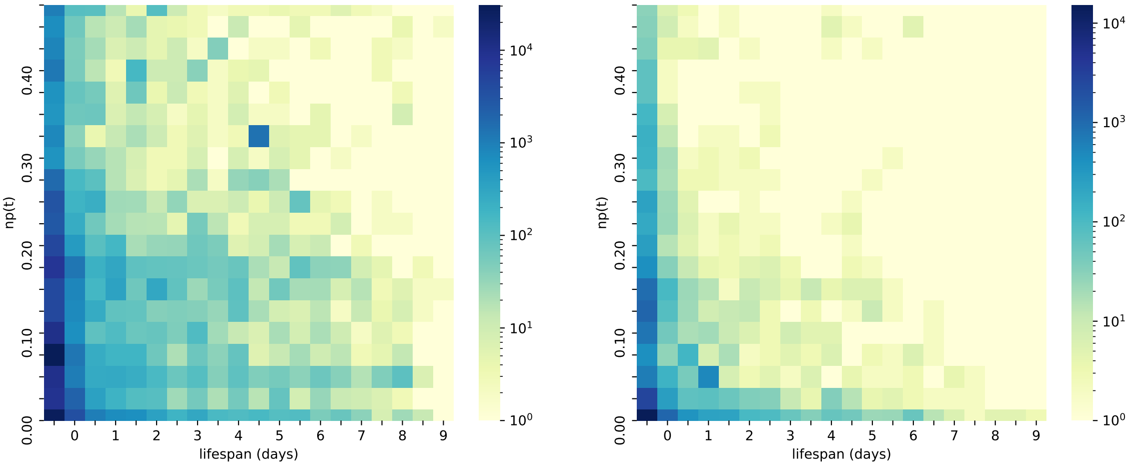
Figure 6.6 shows the distribution of zombie lifespan for both death and recovery cases. More than 90% of zombie outbreaks are resolved within 1 day. While some might take longer to recover, figure 6.3 shows that in such cases the outbreak size is usually smaller.



(a) IPv4

(b) IPv6

Fig. 6.2: CDF of zombies lifespan



(a) IPv4

(b) IPv6

Fig. 6.3: Heatmap of zombies, comparing lifespan and np(t)

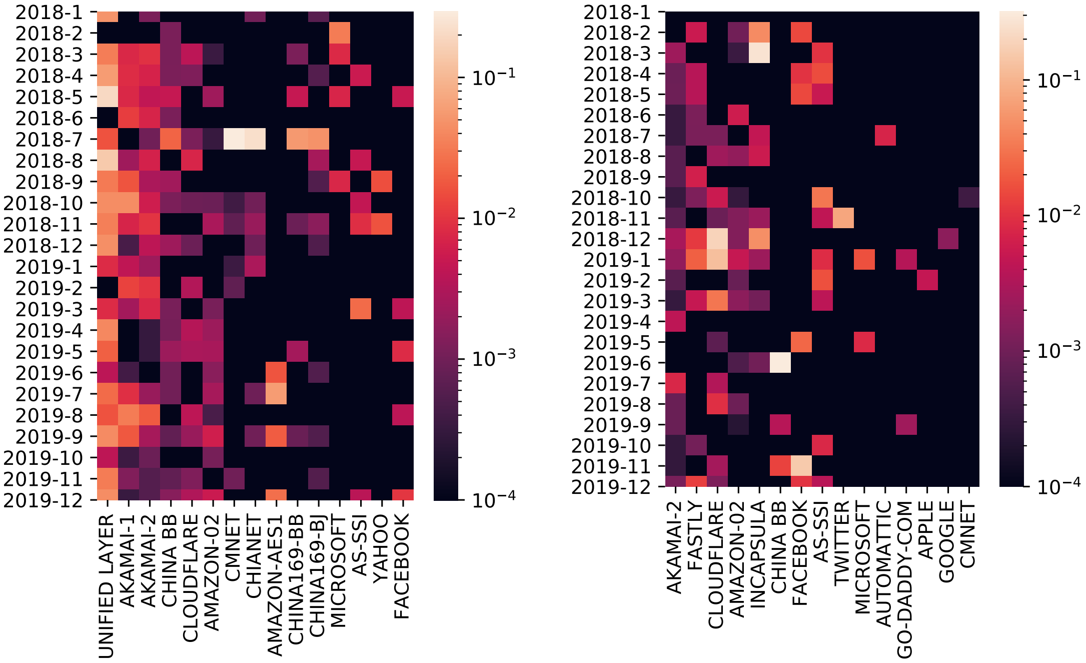


Fig. 6.4: Top 15 popular ASes affected by BGP zombies from Jan 2018 to Dec 2019 and for IPv4 (left) and IPv6 (right). Colors show the number of zombie outbreaks per prefix announced.

6.4 Zombies for Popular Content Networks

To illustrate the prevalence of BGP zombies in regular prefixes, we focused on popular content networks. We investigated the frequency of BGP zombies for 42 ASes that commonly appear in the top 25 of Alexa, Umbrella, and Majestic lists [34].

Figure 6.4 shows the results for the top 15 ASes whose prefixes are consistently appearing in our results for 2018-2019. These ASes are sorted by their median number of monthly #outbreaks per #prefix, this is hereafter referred to as the zombie ranking. For the full 42 ASes ranking, see appendix.

For both IPv4 and IPv6, we found that Akamai (AS16625 and AS20940) prefixes are generating the highest number of zombies. For AS20940, we observe IPV4 zombies for 22 (out of 24) measurement periods. This is an order of magnitude higher than what we record for some other large content providers, such as Google (AS15169, not even in the IPv4 top 15) which zombies are found only in 3 measurements period for IPv4 and 1 measurement period for IPv6. To understand these discrepancies we selected relevant routing characteristics for these ASes and cross-reference them with the emergence of zombies.

Table 6.1: Ranking of popular content networks according to prevalence of zombie outbreaks

AS	zombie rank	prefix rank	path rank
46606 Unified Layer	1	13	3
16625 Akamai	2	3	1
20940 Akamai	3	2	7
4134 China BB	4	7	15
13335 Cloudflare	5	6	12
16509 AMAZON-02	6	1	13
9808 CMNET	7	4	4
23724 CHIANET	8	9	5
14618 AMAZON-AES1	9	10	6
4837 CHINA169-BB	10	8	10
4808 CHINA169-BJ	11	5	2
8068 MICROSOFT	12	14	11
2906 AS-SSI	13	12	8
26101 YAHOO	14	15	9
32934 FACEBOOK	15	11	14

6.5 Routing characteristics and zombies

Due to the erratic emergence of zombies in routers, we expect the number of zombies to be proportional to the number of prefixes announced by an AS. That is an AS announcing more prefixes is more likely to have one of these prefixes turns into zombies. Similarly, we suppose that the probability of zombie emergence increases with the length of announced AS paths as these paths are likely involving more routers and zombies are mostly credited to routers' bugs [5, 8]. Based on these intuitions, we investigated the relation between the number of announced prefixes and AS path length to the occurrence of zombies.

From the 15 ASes of Figure 6.4 IPv4 plot, we computed two other rankings based on the number of announced prefixes per AS and the average path length from RIS peers to these ASes. Table 6.1 shows these ranking values for the top 5 zombie rank ASes and reveals that these ASes either announce a large number of prefixes or have longer AS paths to RIS peers.

For example, Akamai (AS16625), has the longest average AS path length and announces a considerably high number of prefixes (ranked third in terms of the number of prefixes). On the other hand, Amazon (AS16509) ranked sixth, despite announcing the most prefixes. This possibly due to the fact that AS has a shorter path to RIS peers. (ranked 12 in terms of path length).

To better understand the contribution of both attributes to the emergence of zombies, we computed the Spearman correlation between these three quantities. Figure 6.5 shows the relation between the number of outbreaks and the product of average path length and number of prefixes per AS for all zombie outbreaks in 2018 and 2019.

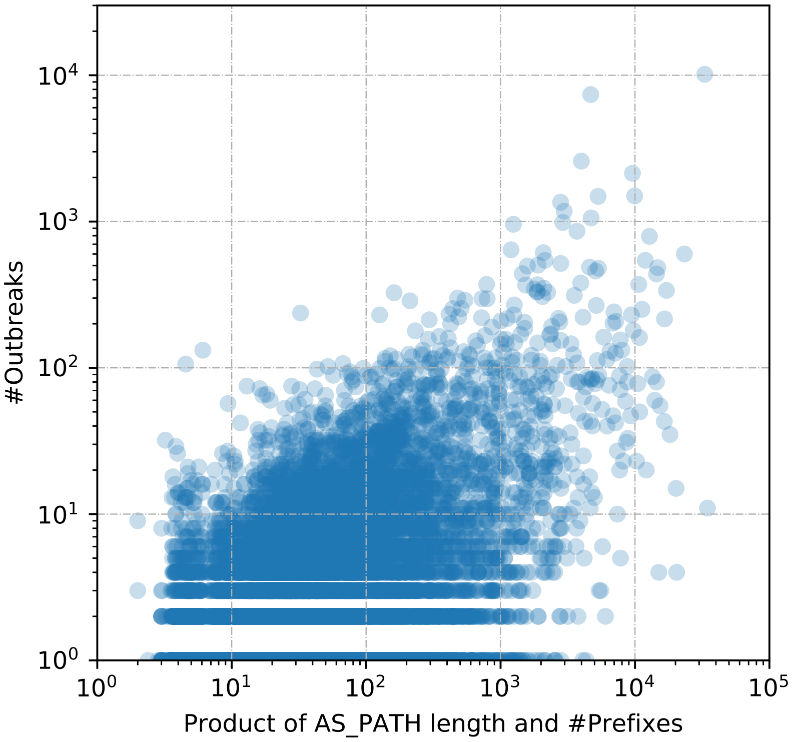


Fig. 6.5: Number of outbreaks against number of prefixes and average AS path length from January 2018 to December 2019 ($\rho = 0.40$).

The correlation coefficient between these two metrics is $\rho = 0.40$, which is higher than that of the number of outbreaks with only average path length $\rho = 0.03$ or only the number of prefixes $\rho = 0.39$.

This indicates that the emergence of zombies for an AS is mainly related to the number of announced prefixes and the path length may influence a bit that process.

6.6 Impact of BGP Zombies

It is possible that BGP zombies misdirect affected routers to peers that are sometimes undesirable. This may create detours that make routes longer than expected and that may resemble hijacking [8].

In our data set we found 77k zombies where the second hop in the AS path is different from the one found in the legitimate AS path of the covering prefix. Meaning that RIS peers, in these cases, are misdirecting traffic to an undesirable AS potentially inflating the route. For 51k zombies we found that the origin AS is different that the one found for the covering prefix. Hence, zombie AS may even misinterpret the origin of certain IP blocks. Our manual inspection of some of these results reveal that many of these zombies are prefixes delegated to customer ASes that have been withdrawn.

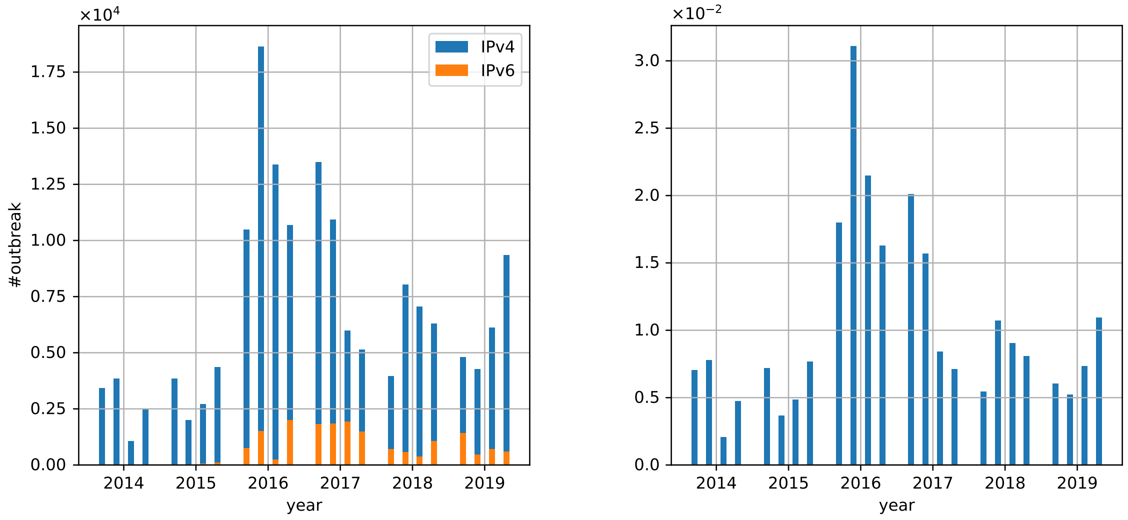
In certain cases detours caused by zombies can create routing loops. This happens, for example, when AS_A has a zombie for prefix (e.g 10.0.0.0/24) and its peer AS_B has no zombie, but a valid route for the covering prefix (e.g. 10.0.0.0/16). Then, if the zombie path contains the pair <AS_A AS_B> and the valid path from AS_B contains the pair <AS_B AS_A>, a routing loop will occur and traffic to the zombie prefix will not reach the destinations. Packets will go back and forth increasing the unproductive traffic.

To quantify the emergence of routing loops caused by detected BGP zombies, we retrieve for all RIS peers the AS paths corresponding to prefixes covering detected zombies and search for routing loops. As in Section 6.1, this analysis is limited by the number of RIS peers and their location. We found 468 potential routing loops where zombie paths contain a pair <AS_A AS_B> and other RIS peers report a pair <AS_B AS_A> in paths of covering prefixes. But we have not enough BGP vantage points to confirm that *AS_A* is indeed infected and *AS_B* is not. Inferring this information would require the use of machine learning or different types of measurement in near-real time (e.g. traceroute). We leave this task for future work, for examples of zombie-caused routing loops observed by network operators we recommend [8].

6.7 Zombie Pandemic

In this section, we discuss the trend of BGP zombies outbreak during the past 6 years. Figure 6.6a illustrates the average number of BGP zombies for every 3 months, while figure 6.6b depicts the same number normalized by the number of prefixes in the routing table. These result show that even though the number of outbreaks is increasing (2019 almost triple 2015), this is mainly due to the enlarging routing table, and the magnitude of #zombies per #prefixes did not shift much in the past years.

While there is a huge surge in the number of outbreaks found in 2016 and the first half of 2017, we believe that this is due to the erratic behavior of BGP, and we do not believe that this peak represents the current state of the Internet.



(a) #Outbreaks

(b) #Outbreaks/#Prefixes

Fig. 6.6: The average number of BGP zombies during 2014-2019

Chapter 7

Discussion

7.1 On BGP Zombies

The results presented in this study have several implications for the networking community. In regard to the increasing size of routing tables and the corresponding concerns about routers resources limitations, our results confirm that BGP zombies contribute to routing tables inflation.

In addition, as the number of zombies is increasing with the number of announced prefixes, an extensive use of prefix deaggregation [35] may be detrimental in terms of BGP zombies. Similarly as we showed that BGP noise is another important factor, the use of BGP optimizer that generates a lot of update messages may also be the cause.

Peering policies and IP space management have a certain impact on BGP zombies. We discovered that shorter paths are less susceptible to BGP zombies, this is evident for large-scale networks that provide complete connectivity at each peering (Google [36]) and anycasted networks (Fastly). Figure 7.1 shows the average path length of popular ASes to IXPs compared to Google.

One of the most important things this study reveals is the impact of BGP zombies on the integrity of the Internet. We discover that even though the incidents (routing loop, detour) caused by zombies can be seriously harmful, the amount of BGP zombies are much less than the past estimates from BGP beacons [5].

While this study provides various aspects of BGP zombies, **the root cause of BGP zombies and the first AS causing the outbreaks are two topics that we left out of scope**. The reason being that we could not acquire the ground truth necessary to confirm the prefix withdrawal. Even though path coherence demonstrates the anomaly in the detected routing entries, this is not sufficient to conclude that the route to the prefix is indeed withdrawn. Therefore, we intentionally left these topics out. We consider that such topics are not appropriate with our dataset and we encourage other researchers to pursue these challenges in a more controlled environment.

7.2 On Detection Algorithm

In this study, we have presented a simple yet effective method of detecting BGP zombies. We understand the trade off between the accuracy and the simplicity, but we truly believe that the problem of BGP zombies would benefit more by collecting multiple analyzed data

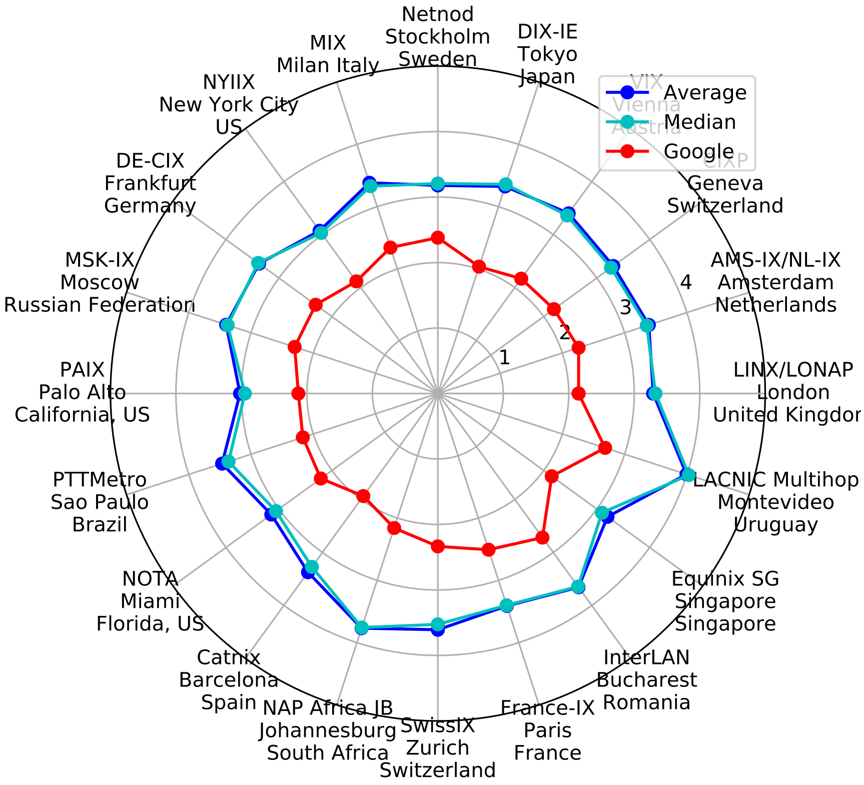


Fig. 7.1: Average path length of Popular ASes compared with Google’ s

from various points of the Internet. Our method is easy to understand and even easier to replicate and test, thus, we hope that the algorithm would attract more operators and researchers to try and conduct BGP zombies experiments, providing more insights of zombies and help improve our understanding on the problem.

We have observed some cases where the detected BGP zombies can possibly be an active routing entry to the locally visible prefix, and not to the withdrawn one. While it is impossible to identify the result without information from the origin AS, we believe that the proportion of such cases are only miniature and did not significantly affect the final result.

Finally, one of the most important results observed in this study is the fact that BGP zombies can cause the complete network outage by inducing the routing loop. While we did not conduct an extensive experiment on the routing loop detection itself, we believe that the zombies-loop detection system would really assist network operators’ community and we have already established a plan for the future work.

Our idea includes a more active measuring method of calling ‘traceroute‘ to all possible loops, and reporting the positive result directly to the operators. However, unlike the analysis on historical BGP data which was done in this study, the zombies-loop detection system must be able to operate in near real-time in order to be proved useful. And with the gigantic amount of BGP data that needs to be downloaded and analyzed, a much faster data-infrastructure is crucial in the next study.

Chapter 8

Conclusion

In this thesis, we extended prior study by investigating BGP zombies in the wild. We have analyzed 720 days of BGP data from more than 200 peers to reveal common prefix withdrawal patterns and then implemented a BGP zombie detection algorithm based on our observations.

Using the detection algorithm we found that BGP zombies are not uncommon, more than a million BGP outbreak has been observed in our data set. We confirmed that detected BGP zombies report incoherent AS paths and discovered that BGP beacons are particularly prone to zombies. We also revealed that BGP zombies are also present for popular web services and especially ASes that announces a lot of prefixes and that are reached through long AS paths. Finally, we discussed the impact of zombies on the routing infrastructure, and provided some insights on the trend of BGP zombies.

References

[1] Shinyoung Cho, Romain Fontugne, Kenjiro Cho, Alberto Dainotti, and Phillipa Gill. Bgp hijacking classification. 2019.

[2] Pierre-Antoine Vervier, Olivier Thonnard, and Marc Dacier. Mind your blocks: On the stealthiness of malicious bgp hijacks. In *NDSS*, 2015.

[3] T. Kitabatake, R. Fontugne, and H. Esaki. BLT: A Taxonomy and Classification Tool for Mining BGP Update Messages. In *2018 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2018.

[4] Pavlos Sermpezis, Vasileios Kotronis, Petros Gigis, Xenofontas Dimitropoulos, Danilo Cicalese, Alistair King, and Alberto Dainotti. Artemis: Neutralizing bgp hijacking within a minute. *IEEE/ACM Transactions on Networking*, 26(6):2471–2486, 2018.

[5] Romain Fontugne, Esteban Bautista, Colin Petrie, Yutaro Nomura, Patrice Abry, Paulo Goncalves, Kensuke Fukuda, and Emile Aben. BGP Zombies: an analysis of beacons stuck routes. In *Passive and Active Measurement (PAM'20)*, pages 197–209, 2019.

[6] RIPE NCC. Current RIS Routing Beacons. `https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/current-ris-routing-beacons`, (Accessed June 2020).

[7] Z. Morley Mao, Randy Bush, Timothy G. Griffin, and Matthew Roughan. Bgp beacons. In *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, IMC ' 03, page 1–14, New York, NY, USA, 2003. Association for Computing Machinery.

[8] Paweł Małachowski. Zombie routes, PLNOG Q3. `https://www.slideshare.net/atendesoftware/bgp-zombie-routes`, 2020.

[9] Jeff Doyle. *Routing TCP/IP, volume II: CCIE professional development.* Cisco Press, 2017.

[10] Y. Rekhter, T. Li, and S. Hares. A border gateway protocol 4 (bgp-4). RFC 4271, RFC Editor, January 2006.

[11] Inc. S. Murphy Sparta. Bgp security vulnerabilities analysis. RFC 4272, RFC Editor, January 2006.

[12] `https://www.ripe.net/publications/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study`.

[13] Chris C. Demchak and Yuval Shavitt. China' s maxim – leave no access point unexploited: The hidden story of china telecom' s bgp hijacking. *Military Cyber Affairs*, 3, 2018.

[14] Zheng Zhang, Ying Zhang, Y. Charlie Hu, Z. Morley Mao, and Randy Bush. Ispy: Detecting ip prefix hijacking on my own. In *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*, SIGCOMM '08, pages 327–338, New York, NY,

USA, 2008. ACM.

[15] Xingang Shi, Yang Xiang, Zhiliang Wang, Xia Yin, and Jianping Wu. Detecting prefix hijackings in the internet with argus. In *Proceedings of the 2012 Internet Measurement Conference*, IMC '12, pages 15–28, New York, NY, USA, 2012. ACM.

[16] J. Schlamp, R. Holz, Q. Jacquemart, G. Carle, and E. W. Biersack. Heap: Reliable assessment of bgp hijacking attacks. *IEEE Journal on Selected Areas in Communications*, 34(6):1849–1861, June 2016.

[17] K. Sriram, D. Montgomery US NIST, D. McPherson, Inc E. Osterweil Verisign, and B. Dickson. Problem definition and classification of bgp route leaks. RFC 7908, RFC Editor, June 2016.

[18] https://blogs.oracle.com/internetintelligence/large-bgp-leak-by-google-disrupts-internet-in-japan.

[19] https://bgpmon.net/bgp-leak-causing-internet-outages-in-japan-and-beyond.

[20] W.A. Miltenburg and RIPE NCC and GII team. Research on RIS Route Collectors. https://labs.ripe.net/Members/wouter_miltenburg/Researchpaper.pdf/view, (July 2014).

[21] http://www.routeviews.org/routeviews/.

[22] https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/ris-raw-data.

[23] http://www.routeviews.org/routeviews/index.php/map/.

[24] https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/ris-peering-policy.

[25] C. Orsini, A. King, D. Giordano, V. Giotsas, and A. Dainotti. BGPStream: a software framework for live and historical BGP data analysis. In *ACM Internet Measurement Conference (IMC)*, Nov 2016.

[26] https://stat.ripe.net/.

[27] APNIC Geoff Huston. Bgp in 2019 – the bgp table. https://blog.apnic.net/2020/01/14/bgp-in-2019-the-bgp-table/, (Jan 2020).

[28] RIPE NCC. The RIPE NCC has run out of IPv4 Addresses. https://www.ripe.net/publications/news/about-ripe-ncc-and-ripe/the-ripe-ncc-has-run-out-of-ipv4-addresses, (Nov 2019).

[29] Alberto García-Martínez and Marcelo Bagnulo. Measuring bgp route propagation times. *IEEE Communications Letters*, 23(12):2432–2436, 2019.

[30] C. Villamizar, R. Chandra, and R. Govindan. Bgp route flap damping. RFC 2439, RFC Editor, November 1998.

[31] Zhuoqing Morley Mao, Ramesh Govindan, George Varghese, and Randy H Katz. Route flap damping exacerbates internet routing convergence. In *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 221–233, 2002.

[32] Alex Fabrikant, Umar Syed, and Jennifer Rexford. There's something about mrai: Timing diversity can exponentially worsen bgp convergence. In *2011 Proceedings IEEE INFOCOM*, pages 2975–2983. IEEE, 2011.

[33] Caitlin Gray, Clemens Mosig, Randy Bush, Cristel Pelsser, Matthew Roughan, Thomas C. Schmidt, and Matthias Wählisch. BGP Beacons, Network Tomography, and Bayesian Computation to Locate Route Flap Damping. In *Proc. of ACM Internet Measurement Conference (IMC)*, New York, 2020. ACM. Accepted for publication.

[34] Johannes Naab, Patrick Sattler, Jonas Jelten, Oliver Gasser, and Georg Carle. Prefix top lists: Gaining insights with prefixes from domain-based top lists on dns deployment. In *Proceedings of the Internet Measurement Conference*, IMC ’ 19, page 351–357, New York, NY, USA, 2019. Association for Computing Machinery.

[35] Luca Cittadini, Wolfgang Mühlbauer, Steve Uhlig, Randy Bush, Pierre Francois, and Olaf Maennel. Evolution of internet address space deaggregation: myths and reality. *IEEE Journal on Selected Areas in Communications*, 28(8):1238–1249, 2010.

[36] Google. Peering. `https://peering.google.com/#/options/peering`, (Accessed on June 2020).

Publications

- **Domestic (Presentation)**
  1. Romain Fontugne (IIJ), Porapat Ongkanchana (U-Tokyo)“COVID19 impact on Japanese Internet”, WIDE Meeting 2020, August 8th 2020
  2. Romain Fontugne (IIJ), Porapat Ongkanchana (U-Tokyo)“COVID19 impact on Japanese Internet”, JFLI seminar, October 28th 2020
  3. Porapat Ongkanchana, Romain Fontugne, Hiroshi Esaki, Emile Aben “Hunting BGP zombies in the Wild”, WIDE Camp 2021, March 2021 (to appear)

Acknowledgment

First, I would like to express my deep gratitude toward Professor Hiroshi Esaki, associate professor Hideya Ochiai and associate professor Manabu Tsukada, for letting me be a part of this wonderful laboratory. I will forever cherish the memory I made and the knowledge I have acquired here. Next, I am sincerely grateful to be able to work with one of the smartest and kindest people I have known, Dr. Romain Fontugne. Thank you for all your help, your guidance, your consultation, your kind and encouraging word and the tremendous patience for my ignorance, for these I truly appreciate. I would like to thank Yudai Aratsu, Sattoru Kobayashi, Tokubi Ayato and Hyuga Kobayashi for technical consultancy, my work could not be done without their help and for that I am grateful. I am thankful for all the time I have spent with my friends, Katsu, Hide, Mai, Mizu, Shin, my seniors and juniors. Lastly, I would like to thank my parents, my sister and brother, and everybody that I have met along the way, for filling my years with great memories that I will never forget.

January 2021
Porapat Ongkanchana

A

Routing Characteristics and Outbreaks

Following the fact that we found BGP zombies can possibly cause a serious problem, such as a routing loop, it is extremely important that we investigate the characteristics of networks which can withstand the zombies problem. While we have discussed briefly about the characteristics of Google's network, in this section, we provide the full detail on the rest of 34 Aes' routing characteristics. (Zombies are not found in 8 ASes, due to the low number of announced prefixes).

Figure A.1 illustrates the #outbreaks normalized by prefixes announced. Table A.1 and A.2, show each popular AS' s #announced prefixes (data taken from RIB file of RRC00), average AS path length (data taken from RIPEstat, queried on 31 Dec of its corresponding year) and number of zombies outbreaks.

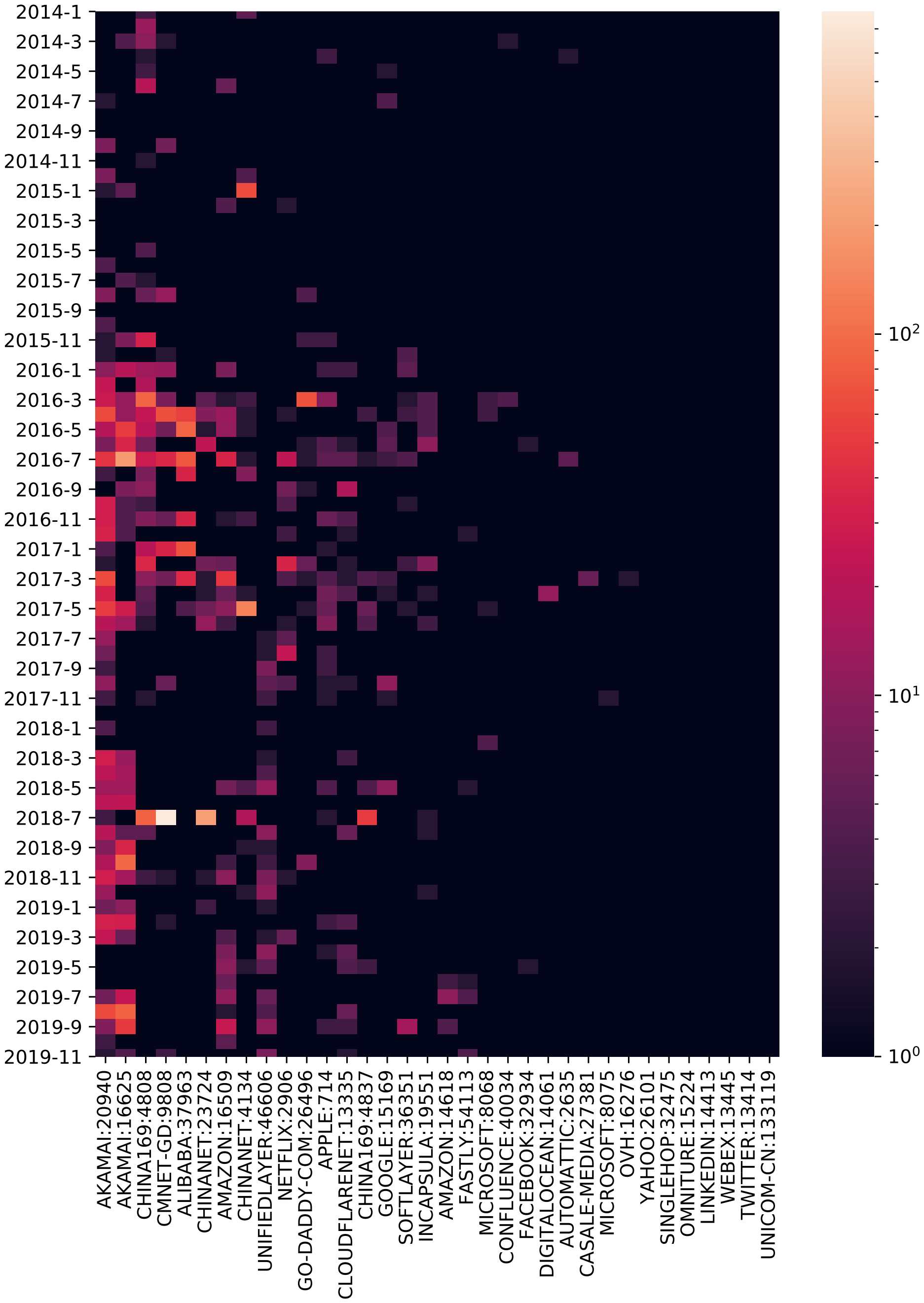


Fig. A.1: All popular ASes affected by BGP zombies from Jan 2018 to Dec 2019 (IPv4)

Table A.1: AS's routing characteristics and #outbreaks (2014-2016)

	2014			2015			2016		
AS	#O	#P	AS	#O	#P	AS	#O	#P	AS
AKAMAI (20940)	21	1400	4.017	23	1930	4.028	297	2636	3.992
AKAMAI (16625)	4	522	4.386	17	866	4.357	351	1122	4.359
CHINA169 (4808)	54	1393	4.376	48	1515	4.381	233	1725	4.217
CMNET-GD (9808)	9	1118	4.143	15	1590	4.074	144	2045	4.045
ALIBABA (37963)	0	59	4.795	0	103	4.905	287	177	4.898
CHINANET (23724)	0	506	3.878	2	887	3.981	43	951	3.979
AMAZON (16509)	7	375	3.253	7	506	3.263	73	1147	3.135
CHINANET (4134)	10	853	2.875	66	1116	3.042	24	1130	3.008
UNIFIEDLAYER (46606)	1	40	3.922	0	38	4.416	0	41	3.694
NETFLIX (2906)	1	69	3.309	6	113	3.394	42	135	3.782
GO-DADDY-COM (26496)	1	640	2.659	8	699	3.038	78	729	3.344
APPLE (714)	4	168	3.424	4	258	3.398	30	395	3.285
CLOUDFLARENET (13335)	2	251	3.120	0	360	3.176	36	504	3.216
CHINA169 (4837)	0	684	3.368	1	704	3.400	8	680	3.224
GOOGLE (15169)	7	227	2.254	2	290	2.351	14	298	2.292
SOFTLAYER (36351)	1	472	2.520	4	643	2.542	20	867	2.563
INCAPSULA (19551)	1	57	3.390	0	108	3.389	24	215	3.336
AMAZON (14618)	0	48	3.960	0	61	4.088	0	68	4.017
FASTLY (54113)	0	36	3.386	0	53	3.569	4	109	3.467
MICROSOFT (8068)	3	15	3.990	0	49	3.611	8	58	3.440
CONFLUENCE (40034)	5	34	4.020	2	55	3.931	6	46	3.867
FACEBOOK (32934)	0	42	2.852	0	47	2.967	5	55	3.021
DIGITALOCEAN (14061)	0	8	3.447	0	11	3.520	0	23	3.269
AUTOMATTIC (2635)	2	22	3.393	0	68	3.312	6	86	3.218
CASALE-MEDIA (27381)	0	6	3.787	0	10	3.604	0	13	3.513
MICROSOFT (8075)	1	136	2.496	0	128	2.145	0	135	2.116
OVH (16276)	0	70	2.391	0	65	2.449	0	69	2.440
YAHOO (26101)	0	33	3.495	0	35	3.655	0	37	3.393
SINGLEHOP (32475)	0	96	3.647	0	160	3.564	1	209	3.600
OMNITURE (15224)	0	45	3.443	0	48	3.504	1	54	3.509
LINKEDIN (14413)	0	2	3.331	0	10	3.433	0	18	3.349
WEBEX (13445)	0	21	3.236	0	26	3.288	0	28	3.257
TWITTER (13414)	0	5	3.076	0	10	2.466	0	37	2.426
UNICOM-CN (133119)	0	5	4.815	0	13	4.775	0	15	4.361

(#O: #Outbreaks, #P: #Announced prefixes (IPv4), AS: Average AS path length)

Table A.2: AS's routing characteristics and #outbreaks (2017-2019)

	2017			**2018**			**2019**		
AS	#O	#P	AS	#O	#P	AS	#O	#P	AS
AKAMAI (20940)	211	2948	4.049	187	2669	4.015	150	2736	3.912
AKAMAI (16625)	46	1102	4.582	218	1873	4.766	215	2574	4.714
CHINA169 (4808)	81	1622	4.169	98	1645	4.296	3	1650	4.318
CMNET-GD (9808)	49	2248	4.126	785	2245	4.026	6	2734	4.074
ALIBABA (37963)	114	213	4.935	0	259	4.955	0	315	4.954
CHINANET (23724)	32	962	4.005	216	961	4.001	5	991	3.980
AMAZON (16509)	73	1880	3.097	22	2334	3.101	99	3015	3.082
CHINANET (4134)	139	920	3.016	31	806	3.022	10	806	2.920
UNIFIEDLAYER (46606)	20	44	3.743	56	75	4.236	61	233	4.284
NETFLIX (2906)	76	119	3.640	5	102	3.596	7	128	3.579
GO-DADDY-COM (26496)	14	702	3.410	11	738	3.320	2	739	3.361
APPLE (714)	38	684	2.581	9	644	2.534	9	809	2.581
CLOUDFLARENET (13335)	11	540	3.035	13	607	3.228	29	722	3.251
CHINA169 (4837)	17	743	3.250	58	818	3.424	4	852	3.430
GOOGLE (15169)	21	310	2.197	12	427	2.193	0	493	2.231
SOFTLAYER (36351)	8	953	2.515	2	1018	2.476	17	991	2.501
INCAPSULA (19551)	15	357	3.301	7	497	3.310	1	547	3.388
AMAZON (14618)	0	74	4.022	0	83	3.938	24	168	3.926
FASTLY (54113)	0	115	3.527	5	148	3.576	14	213	3.590
MICROSOFT (8068)	3	75	3.239	7	89	3.291	0	86	3.352
CONFLUENCE (40034)	1	29	3.955	0	30	4.043	0	31	4.019
FACEBOOK (32934)	0	68	3.073	1	74	3.130	7	90	3.021
DIGITALOCEAN (14061)	12	188	3.232	0	347	3.035	0	494	3.116
AUTOMATTIC (2635)	0	93	3.184	1	99	3.194	0	113	3.256
CASALE-MEDIA (27381)	7	26	3.514	0	35	3.474	0	38	3.543
MICROSOFT (8075)	4	132	2.001	1	128	2.117	0	137	2.181
OVH (16276)	3	83	2.358	0	97	2.329	0	119	2.361
YAHOO (26101)	0	43	3.475	2	54	3.534	0	45	3.526
LINODE (63949)	0	359	3.007	1	364	3.139	0	370	3.236
SINGLEHOP (32475)	0	230	3.582	0	266	3.534	0	287	3.590
OMNITURE (15224)	0	60	3.510	0	70	3.523	0	76	3.640
LINKEDIN (14413)	0	20	3.402	0	23	3.350	1	23	3.459
WEBEX (13445)	0	30	3.310	1	32	3.243	0	34	3.435
TWITTER (13414)	0	54	2.604	0	92	2.592	1	87	2.732
UNICOM-CN (133119)	0	20	4.316	0	34	4.318	1	41	4.297

(#O: #Outbreaks, #P: #Announced prefixes (IPv4), AS: Average AS path length)

B

Calculating $n_p(t)$

In this appendix, we discuss how raw BGP data from IRR is transformed into $n_p(t)$. As mentioned before, this study is dealing with terabytes of BGP data and only possible with the scalable method. In the following sections, we present our data pipeline architecture and the logic behind each design.

B.1 Challenges

In order to calculate $n_p(t)$, we have to perform the following procedure.

1. Download RIB and Update BGP data from each RRC.
2. For each peer in RRC, starting from RIB, replays all BGP messages up to time t
3. Count the number of routers announcing prefix p.

While at first glance these steps might seem simple, the complexity lies in the details.

- **Tremendous amount of data:** System needs to track the status of over million prefixes for all connected peers. Given the limited resources, the system must be able to scale out when necessary.
- **Number of peers connected to RRC is different:** IRRs usually provide BGP data per RRC and not per peer. For example, while RIS has the same data interface for all its route collectors, RRC00 has over 140 connected peers while RRC14 has only around 20. This requires the data pipeline to have a proper method of dealing with uneven distributed data.
- **Connection lost:** Considering all BGP data must be downloaded from IRR, there is a high possibility that the connection error will occur. The implemented system must know how to handle this kind of network failure.

In the next section, we explain the architecture and operation flow of our system.

B.2 Data Analytics System

There are 3 main components in our proposed system: working queue, workers (analyzer and aggregator) and storage server.

- **Working queue** is used as the buffer between workers and workload. All workloads

are put in the queue before passing to the worker. Queue must also support message committing, only discard committed message while re-append the non committed one. In this experiment, we used Kafka as our working queue.
- **Worker**'s main responsibilities are performing the analysis based on the message it received, storing its computation result, and signalling the next inline workers when the job is done. There are 2 workers: analyzer for computing each RRC data and aggregator for combining analyzer' s result.
- **Storage server** stores metadata, partial results and the final aggregated result.

With all components introduced, next we describe the computational process.

1. Schedule the workload by inserting $(RRC, Year, Month)$ messages into the working queue.
2. The queue distributes messages to idle analyzer workers.
3. Analyzer receives the message and starts the computation: downloading BGP data, replaying BGP messages and dumping the partial result every 15 minutes.
4. Analyzer finishes its computing, stores the result on the storage server, then signals aggregator.
5. Aggregator receives the signal from all RRC analyzers, then combines all partial results from the corresponding year and month.

With this presented system, we can efficiently compute $n_p(t)$, while having the benefit of scaling out (increasing worker) and fault tolerance (message committing).

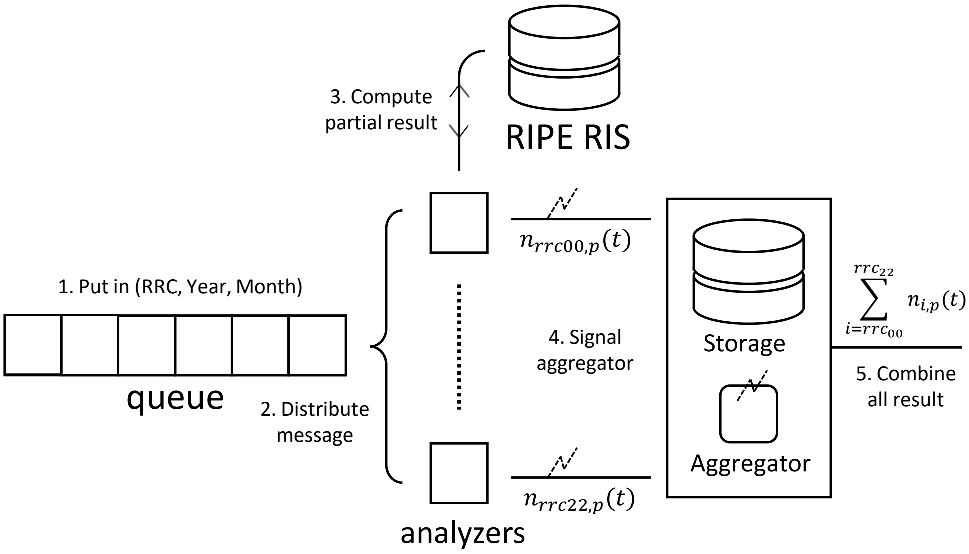


Fig. B.1: BGP data analytic system