Longitudinal Study of BGP Monitor Session Failures *

Pei-chun Cheng †
pccheng@cs.ucla.edu

Xin Zhao ‡
zhaox@email.arizona.edu

Beichuan Zhang ‡
bzhang@cs.arizona.edu

Lixia Zhang †
lixia@cs.ucla.edu

ABSTRACT

BGP routing data collected by RouteViews and RIPE RIS have become an essential asset to both network research and operation communities. However, it has long been speculated that the BGP monitoring sessions between operational routers and data collectors fail from time to time. Such session failures can cause missing update messages as well as introduce extra duplicate updates, making any results derived from such data inaccurate at best. Since no complete record of the session failures is available, users either have to sanitize the data discretionarily with respect to their specific needs or, more commonly, assume that session failures are infrequent enough to cause significant problems. In this paper, we present the first systematic documentary and assessment on BGP session failures of RouteViews and RIPE data collectors over the past eight years. Our results show that monitoring session failures are rather frequent as more than 30% of BGP monitoring sessions experienced at least one failure every month. Furthermore, failures that happen to multiple peer sessions but around the same time suggest that a major factor to the session instability is the data collector's local problems. We have developed a web site as a community resource to publish all session failures detected for RouteViews and RIPE RIS data collectors. It will help users select and clean up BGP data before performing their analysis, and thus yield more accurate and credible results.

1. INTRODUCTION

RouteViews [3] and RIPE RIS [2] have been collecting BGP [12] routing data for over a decade. The original purpose was to provide network operators "looking glasses" from other networks point of view, for monitoring and diagnosis. Over time, these data have also become indispensable to the research community in understanding the global routing system, such as Internet topology [14], BGP convergence [11], ISP peering policies [8], and prefix hijack monitoring [10], to name just a few.

Unfortunately, the quality of collected BGP data is known to be far from perfect. BGP sessions between the data collector and peer routers can fail for various reasons, and when such a session failure occurs, the collector misses BGP routing updates during the session downtime and receives superfluous updates due to the table transfer right after each session re-establishment [12]. Such data deficiency must be taken into account when analyzing BGP data in order to get accurate results, which has been highlighted by several prior works, such as analyzing BGP update surge during worm attacks [16], comparing routing stability of different prefixes [13], and correlating routing events in a network [6]. Yet, there has been no systematic assessment on the quality of the collected BGP data or the stability of data collecting process. Users often assume that session failures are infrequent and simply carry out the analysis without accounting for potential data deficiency.

In this paper, we conduct the first longitudinal study of BGP monitoring session failures for six RouteViews and RIPE collectors over the last 8 years. Using the Enhanced Minimum Collection Time (eMCT) algorithm as the main tool [1], we identify BGP session resets between operational routers and the data collectors and measure their occurring frequency. We also analyze the impacts of collector instability and BGP timer on session failures. Our results confirm the speculation that the raw BGP data collected by RouteViews and RIPE contain noises caused by measurement artifacts. Our main findings can be summarized as follows:

- The monitoring session failures are relatively frequent, averaging a few times a month. Most failures have a session downtime within tens of minutes.
- A significant number of failures are caused by the collectors local problems, resulting in multiple session resets at the same time.
- Although disabling BGP Keepalive and Holddown timers, as RIPE did from 2002 to 2006, may make a BGP session more robust against packet losses, it can also lead to unnoticed session failures and extremely long session downtime.

As the main outcome of this study, we have developed a web site, `http://bgpreset.cs.arizona.edu`, to publish all the detected session failures with their times and durations for historical RouteViews and RIPE data. The web page is also updated periodically to include new data available every day. Given this information, users of RouteViews and RIPE data can choose which period of data to use and which part of the data to sanitize for accurate analysis. The impact of the data deficiency depends on the nature of the specific purpose. For example, missing updates during the session downtime may not affect the results of collecting Internet topology over a long period of time, but will affect

†Computer Science Department, University of California, Los Angeles.
‡Computer Science Department, University of Arizona.
*This work is partially supported by US National Science Foundation under Contract No CNS-0551736.

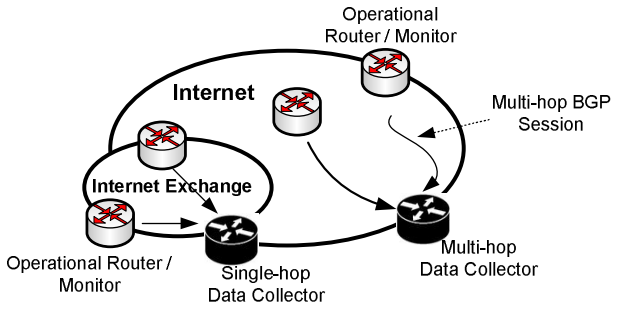


Figure 1: BGP Monitoring

the results of analyzing routing dynamics, and can even be critical to anomaly detection if the downtime is significant. Similarly, the extra updates from table transfers may also affect different work in different ways. The session failure information will help users make informed decisions about the data before putting valuable efforts into the analysis.

The rest of the paper is organized as follows. Section 2 gives brief background on BGP monitoring projects and BGP sessions, Section 3 describes the data source and the technique we use to detect session failures, Section 4 presents the overall statistic results and observations for RouteViews and RIPE monitoring session failures, Section 5 infers failures caused by collector's local problems by correlating session failures, Section 6 investigates the impact of historical decision on turning off BGP Keepalive/Holddown timers, Section 7 briefly reviews related work, and Section 8 concludes the paper.

2. BACKGROUND

RouteViews and RIPE RIS, the two best known BGP data collection projects, operate a number of *collectors* that establish BGP peering sessions with routers in many operational networks. We call each operational router connected to a collector a *monitor* or a *peer*, and the BGP session between the monitor and the collector a *monitoring session*. A monitoring session can be either *multi-hop* or *single-hop* depending on whether the session is across multiple router hops or just a single router hop. As shown in Figure 1, single-hop monitoring sessions are usually deployed at an Internet Exchange, while multi-hop monitoring sessions are established over wide-area networks. The data collectors receive BGP routing updates from its peers and write the collected BGP updates into files every 15 minutes (RouteViews) or every 5 minutes (RIPE) in the Multi-threaded Routing Toolkit (MRT) [5] format. These files are then made publicly available every 15 minutes for RouteViews and 5 minutes for RIPE in general. The collectors also dump a snapshot of BGP routing table, the RIB, for each of its peers every two hours in the MRT format.

BGP uses TCP for underlying reliable communication. After successfully setting up a TCP connection, the two BGP peers negotiate BGP timer settings and capabilities [12] before fully establishing the BGP session. Then they will exchange with each other their full routing tables, which are *table transfer* updates. After this initial table exchange, the peers send each other new updates when any of the routes changes, which are *incremental* updates.

A BGP session may fail due to a variety of causes, including (1) malformed updates which may in turn be caused by hardware or software defects, (2) TCP connection failures due to link or interface failures, (3) data traffic congestion that results in the loss of three consecutive BGP KeepAlive messages, or (4) either end (the host or its routing daemon) fails. BGP employs two timers, Keepalive and Holddown, which are 60 seconds and 180 seconds respectively by default, to maintain its session. BGP peers send to each other Keepalive messages at every Keepalive timer interval. If no Keepalive message is received before the Holddown timer expires, a BGP router will tear down the existing session and initiate a new one, which is a *session reset*.

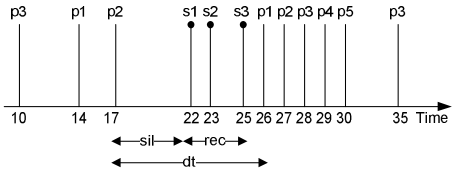


Figure 2: BGP Update Stream (sil: silence period; rec: session reconnection; dt: downtime)

Table 1: BGP Data Sources

Collector	Type	Start Date	Location
RRC00	Multi-hop	2001 Jan	Amsterdam
RRC01	Single-hop	2001 Jan	London
RRC02	Single-hop	2001 Mar	Paris
OREG	Multi-hop	2001 Oct	Oregon
LINX	Single-hop	2004 Mar	London
EQIX	Single-hop	2004 May	Ashburn

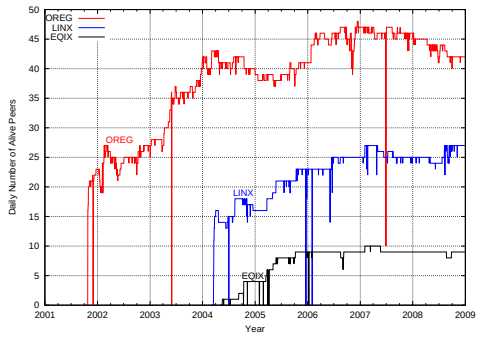
Assuming that a monitor has a routing table of 5 prefixes, Figure 2 shows a sample message stream abstracted from one session reset. First, three regular BGP updates (for prefixes $p1$, $p2$, $p3$) are received at time 10, 14 and 17 respectively. Then from time 17 to 22, the session fails and restarts at time 22. The session re-establishment takes time from 22 to 25, during which there are three BGP *state messages*. The state message $s1$ marks the time when a router initiates a BGP session, while $s3$ marks the time when the session is fully established. We only show three state messages here for illustration purpose. In general, establishing a BGP session may require more than state changes [12]. Following state messages are the table transfer updates from time 26 to 30, which consist of the entire routing table ($p1$ to $p5$), and then incremental updates after time 35.

From the above example, it is clear that BGP updates may be missing during session downtimes, and extra table transfer updates will be introduced when the session restarts. For researchers who use the collected BGP data, it is important to be able to accurately identify the periods of missing data and the extra updates due to table transfer.

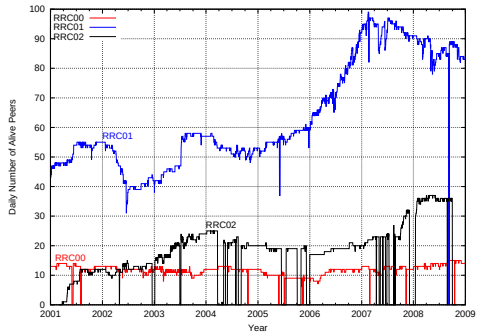
3. DATA SOURCE AND METHODOLOGY

This section describes the data sources and the basic approaches used to identify session resets in archived BGP data.

3.1 Data Sources



(a) RouteViews



(b) RIPE

Figure 3: Number of monitors over time.

RouteViews and RIPE started collecting BGP data in the late 1990's, but they went through a learning period in the first couple of years before the data collection process stabilized. Thus this paper uses the data from January 2001 onward, which is eight years worth of data. We take data from total 6 collectors, which include the earliest deployed collectors as well as collectors deployed in recent years. The summary information of the collectors is listed in Table 1. Figure 3 shows how the number of peers changes over time at these collectors. For each day we count how many unique peers have logged any BGP data. The downward spikes in the figures mean that a large number of peers did not log any data on those days, which could be caused by collector outage or maintenance, and we will investigate the collector's local problems in more details later.

3.2 Detecting BGP Session Resets

To detect session resets in the BGP data, we use both BGP session state messages and an enhanced version of the Minimum Collection Time (MCT) algorithm. As Figure 2 shows, session state messages ($s1, s2, s3$) mark when a new session is attempted and when it is fully established. With this information we can identify all session resets accurately. These state messages, though, are only logged by RIPE collectors, not RouteViews. They also do not mark the end of the table transfer. Zhang *et al.* [17] has developed an algorithm called Minimum Collection Time (MCT) that can identify the start and the duration of table transfers from BGP data without state messages. Based on the observation that all prefixes in the routing table are announced during a table transfer, MCT searches for the smallest time window during which the full table is announced. This method can detect over 94% of session resets using three month of data from 14 different monitored peers. We have developed an enhanced MCT (eMCT) [1], which further improves the detection accuracy. In this paper, we use eMCT as the main tool to detect BGP session failures, and also use state messages when dealing with RIPE data. Since eMCT assumes a reasonable large routing table size, in this study, we only consider monitors whose exported routing tables have more than 500 entries.

In [15] Wang *et al.* used syslog messages to detect failures of BGP sessions in a tier-1 ISP, however, such information is not available from RIPE or RouteViews. Currently, RIPE makes available the log files from Quagga [**?**], the routing software running on its collectors, but Quagga log does not explicitly record BGP session resets. RouteViews maintains logs from Rancid, a tool that monitors the changes of router configuration. However, Rancid log is only generated once every hour. We use these logs to cross check our results, but cannot rely on them as the main method to detect most session resets.

4. CHARACTERISTICS OF SESSION RESETS

In the following sections, we characterize session failures of RIPE and RouteViews monitoring sessions. We first present the overall statistics in this section, and then further investigate the stability of the collectors and the impact of disabling BGP keepalive and holddown timers in later sections.

Compared with operational BGP sessions, the monitoring sessions between data collectors and monitors are expected to be stable. Since data collectors only passively receive BGP updates from their peering monitors and are not involved in forwarding data traffic, the monitoring sessions should have simpler configuration, lower workload, and requires less maintenance. Thus, monitoring session resets are commonly assumed to be infrequent, and users of BGP data usually do not take session resets into consideration.

Our results, however, show that monitoring session resets are relatively frequent. Figure 7 shows the cumulative number of resets for two monitoring sessions, 66.185.128.1 and 217.75.96.60, over the past eight years. The session with 66.185.128.1 has 4.5 resets per month on average, a typical case among the sessions at OREG. The session with 217.75.96.60 is the worst case at OREG, averaging 15.8 resets per month. In all the cases we have seen, although some months have more resets than others, overall the resets occur persistently over time. In other words, it is the norm rather than exception at OREG.

Frequent session resets are also observed across all the collectors, regardless of the type of the session (single-hop or multi-hop), the age of the collector, or its location. Figure 4 shows the cumulative distribution of the number of resets per session per month. For the two multi-hop collectors, OREG and RRC00, 10-20% session-months do not have any reset, while the 50-percentile is 3 resets, and the 90-percentile is 12 to 15 resets per session-month. The worst case at OREG is a monitoring session that had 117 resets in one month, while one of the RRC00 peers had 4205 resets in one month. The single-hop collectors have fewer resets, but the numbers are still alarming. RRC01 and RRC02 also have some sessions that had thousands of resets in a month . These cases were likely caused by hardware problems or

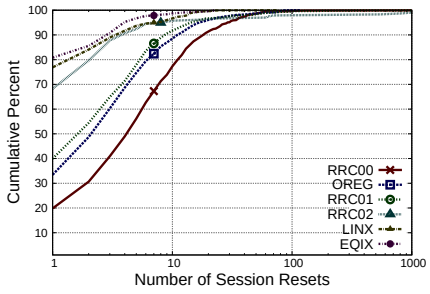


Figure 4: Number of Resets per session-month.

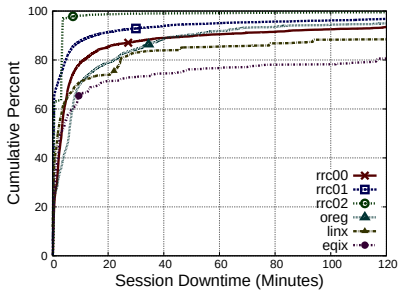


Figure 5: Session Downtime

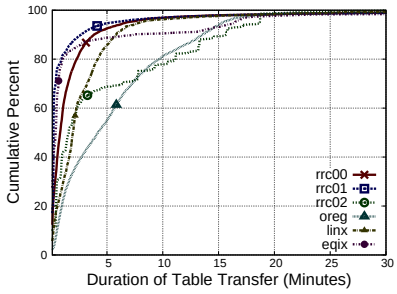


Figure 6: Duration of Table Transfers

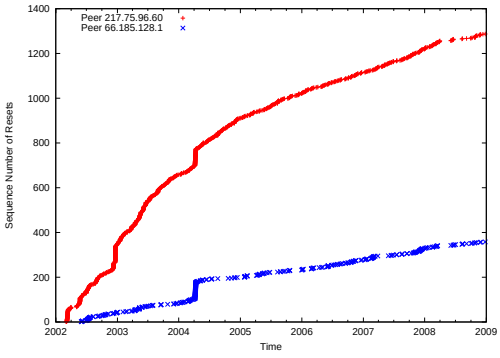


Figure 7: Resets of two example sessions

misconfigurations that made the sessions up and down frequently before they were fixed.

When a monitoring session fails, the observed session downtime is usually one or a few minutes to a few tens of minutes, during which routing updates will not be received from the peers. Figure 5 shows the cumulative distribution of session downtimes. Here, *session downtime* is defined as the time between when the failure occurs and when the BGP session is fully re-established. Since the failure itself is not logged in the BGP data, we measure session downtime from the last BGP update preceding a reset and the first BGP update after the session re-establishment, as illustrated in Figure 2 from time 17 to 26. In Figure 5, we observe that most session downtimes are in the order of tens of minutes, but some cases have very long session downtime. For example, OREG has 25-percentiles of 1 minute, 50-percentiles of 6 minutes, and 90-percentiles of 48 minutes in session downtime. All collectors have cases in which the session downtime is more than 10 days. Users of BGP data can easily spot very long session downtimes (e.g., days) and take precautions accordingly in their data processing. However, given that majority of the session downtimes are within tens of minutes, without knowing the existence of session resets, it is difficult for the BGP data users to identify these short durations of quiet periods as data deficiency and take corresponding measures.

Figure 6 shows the cumulative distribution of table transfer duration after each session reset. Over 90% of table transfers finish within around 5 minutes, while table transfers at OREG in general take longer time to finish, with 50-percentile of 4.5 minutes and 90-percentile of 14 minutes. We have calculated and found that the table transfer time is not significantly correlated with the routing table size, which indicates that the link bandwidth is not the limiting factor. As Houidi *et al.* [9] has discovered, slow table transfers are usually caused by router's timer-driven behavior in sending BGP updates.

The main point to take away from this section is that the BGP monitoring session resets occur relatively frequently, averaging a few times per month, and across all the 8 years and 6 collectors that we have examined. Most session downtimes last within tens of minutes and the ensuing table transfers usually complete within minutes, during which valuable BGP updates are missing and superfluous table transfer updates are introduced. There even exist cases that have thousands of resets a month, go down for days, or take tens of minutes or longer to finish the table transfer. It is imperative for users to be aware of these events and take them into account when using the BGP data.

5. COLLECTOR STABILITY

Maintaining stable data collecting service is critical to the quality of logged BGP data. Collecting service may be disrupted by hardware defects, software bugs, network problems, or planned maintenance. For example, RouteViews have reported sporadic collector outages owing to interface malfunction, memory problem, fiber cut, software upgrade, and other problems [4]. RIPE also occasionally announces degraded service for maintenance [2]. However, there has been little understanding of the impact of these events on the collecting service. Neither RIPE nor RouteViews maintains complete information about collector outrages.

5.1 Correlating Session Resets

From the session resets identified in the previous section, we find that session resets across different peers are sometimes clustered within a short time window. For example, Figure 8 shows the session resets for RRC00 during August, 2003. On August 19th, almost all peers had session resets, which implies that the collector might be having a problem.

We define *synchronized session resets* as a group of resets occurred within a time window w, *synchronized peers* as the peers appearing in synchronized resets, and *synchronization ratio* as the ratio of the number of synchronized peers to the number of total alive peers at that time. For example, if five out of ten peers have resets within w, then these five resets are synchronized resets with five synchronized peers and the synchronization ratio is 0.5.

Figure 9 shows the cumulative distribution the number of synchronized peers. For RRC00, about half of the session resets are standalone (*i.e.*, only one synchronized peer), while the other half are synchronized to some extent. For other collectors, synchronized resets are even more than 70% of all the resets. There is a significant increase near the tail of the curve, indicating that a significant number of session resets involves all or most peers.

Figure 10 shows the cumulative distribution of the synchronization ratio. There is a sharp increase among all collectors between 0% to 10%, which is because we usually have 10 to 20 concurrently alive peers, leading to around 5% to 10% lower bounded synchronization ratio. After the synchronization ratio reaches 90%, there is another sharp increase, which contributes to around 10% to 30% of all session resets.

5.2 Identifying Collector Problems

We assume that if all or most of peers have session reset at the same time, it is very likely caused by the collector's local problems. We use 90% of synchronization ratio as the threshold, and require there must be at least 5 alive peers. Note that we call it "collector restart" even though there can be different local problems, such as rebooting the collector machine, restarting the BGP daemon, network problems, and so on.

As the result, we detect 72 collector restarts in RRC00 data from August 2002 to December 2008. August 2002 is used as the starting time because RIPE started to archive the process log of the collector daemon at that time. The process log records the termination and starting of the collector process, and thus can be used to verify our detection result. After matching the *observed* restarts against those *recorded* in collector process logs, we find 7 observed collector restarts that are detected by our method but not recorded in collector process logs. Further inspection finds that 5 cases are due to the error of collector log and 2 cases are due to a lot of BGP re-connections in a short time, which might be caused by network instability. There are also 22 collector restarts that we fail to detect but are recorded in collector process logs. Among these cases, 2 cases are due to two consecutive collector restart, so there is no BGP session successfully established in between. Other 20 cases are due to peer de-configuration or failures, with which a collector cannot successfully re-establish sessions to some peers after collector restart, so that the synchronization ratio is lower than our 90% threshold. Overall this simple algorithm yields over 95% of correctness and detects 80% of collector restarts.

Note that without using this inference algorithm, we may still directly identify collector restarts solely based on collector logs. However, as we show in the previous comparison, the collector log itself is not complete. In addition, collector logs are not even available for RouteViews. Detecting synchronized session resets provides a practical way to identify RouteViews collector problems.

Table 2 shows the number of collector restarts detected at each collector along with the number of session resets triggered by these restarts. We can see that the collector restarts contribute to 14% to 37% of session resets.[1] The problem is more pronounced for collectors that have many peers, such as OREG, whose 37% of session resets are due to collector's local problems. Since collectors' local problems is a major contributor to the session failures, improving the stability of the collector, including its network connections, software and hardware, is important in reducing session failures.

[1]Since RRC02 sessions are quite stable in general, the number of session resets is not large enough to conclude a collector restarts by using synchronization ratio.

Table 2: Session Resets on Collector Restarts

collector	no. restarts	no. session resets (%)
RRC00	105	1154 (14%)
RRC01	112	1999 (26%)
RRC02	-	-
OREG	178	6370 (37%)
LINX	29	673 (30%)
EQIX	9	69 (14%)

Table 3: RIPE BGP Timers Setting

Time Period	Keepalive	Holddown
Before 2002 Oct 17	60 sec	180 sec
After 2002 Oct 17 Before 2006 Nov 23	0 sec	0 sec
After 2006 Nov 23	60 sec	180 sec

6. KEEPALIVE AND HOLDDOWN TIMERS

At October 2002, RIPE decided to disable BGP Keepalive/Holdown timers. This was based on the observation that the old version of collectors got blocked during periodical RIB archiving, and thus most BGP sessions would timeout and trigger a surge of session resets. To alleviate this problem, RIPE configure collectors' local timers to zero to completely disable BGP timeouts. However, RIPE also noticed that after disabling Keepalive/Holddown timers, BGP lost the ability to detect connectivity problems such as link failures, and thus introduced unexpected long session downtime. At November 2006, since the newer collector version had fixed the RIB dumping problem, RIPE restored the BGP timers back to the previous values. Table 3 summarizes the timer setting for RIPE. We document and quantify the impacts of changing BGP Keepalive/Holddown timers on the stability of RIPE monitoring sessions.

One problem we observed is that while RIPE's plan was to disable timers for all its BGP sessions, , there were some peers that kept enabling Keepalive/ Holddown timers. This might be because the zero timers were not allowed in some Juniper routers back in 2002, or because of some misconfigurations which we would observe in later results.

To better understand the impact of changing BGP timers, we thus need to differentiate BGP sessions which enabled or disabled timers. We define *Keepalive-enabled(KAE)* as BGP sessions that enable BGP timers, and *Keepalive-disabled(KAD)* as sessions that disable BGP timers. The later represents the intended behavior for RIPE during Oct 2002 to Nov 2006.

6.1 Identifying KAE/KAD Sessions

Differentiating KAD and KAE sessions imposes a challenge since RIPE does not keep historical record for collector configurations. In this section, we proposed a heuristic method to sort out these two kinds of sessions.

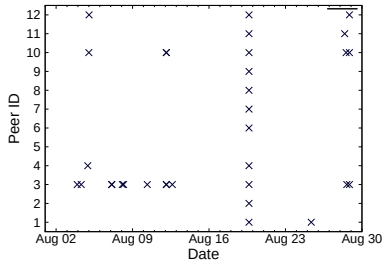


Figure 8: Session Resets in August 2003

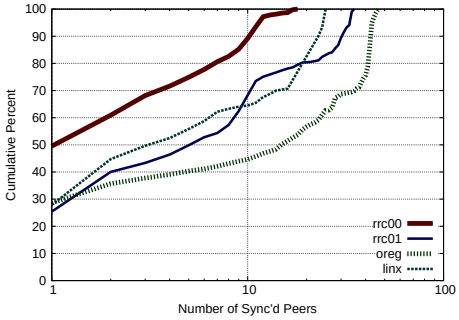


Figure 9: CDF of Sync'd Peers

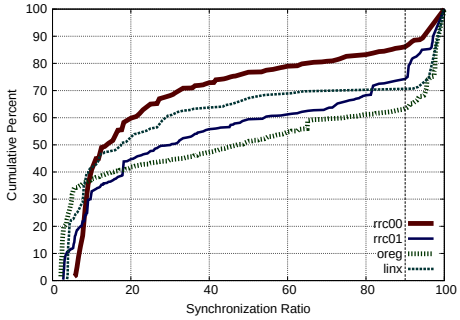


Figure 10: CDF of Sync Ratio

Table 4: KAE / KAD Peers

Collector	Total Peers	KAE	KAD
RRC00	42	9	33
RRC01	57	5	52
RRC02	15	2	13

The basic idea is to infer the BGP Holddown timer by the distribution of session downtime. More specifically, we divide a session downtime into a *silence period* followed by a *recovery period*. We define *silence period*, preceding a session re-establishment, as the duration when a failed session remains silent. Figure 2 shows an example silence period, *sil*, between time 17 and 22. In general, silence periods indicate how long does it take for a data collector to detect failures. For session resets triggered by the expiring of Holddown Timer, the silence period should be close to the length of Holddown Timer. Figure 11 shows the distribution of silence time of session resets for an example RRC00 session with 90 second Holddown Timer, which basically shows that a significant number of session resets are associated with 90 second silence period. We also define *recovery period* as the duration of BGP session re-establishment. figure 2 shows an example recovery period, *rec*, between time 22 and 25.

Based on these definitions, we identify KAE sessions as those have a single silence period which contributes to more than 10% of session resets. This 10% threshold is chosen conservatively based on the measurement result in [15], which observed that more than 20% of session resets are triggered by the expiration of the local Holddown Timers.

Applying this algorithm on RRC00 data, we identify 9 KAE sessions out of total 42 BGP sessions. Figure 12 and Figure 13 show the distribution of silence time for one identified KAE session and KAD session, respectively. The vertical lines mark the date when RIPE disabled and enabled BGP timers. These two figures verified that after RIPE disabled timers at Oct 17 2002, the identified KAE session still continued to trigger session resets after 90 seconds silent period, but not the KAD session. Table 4 summarizes the inference results for three RIPE collectors. In the following sections, we only consider the KAD session resets.

6.2 Number of Session Resets

We first measure the number of session resets before and after disabling timers.

Figure 14(a) shows the cumulative distribution of number of session resets for KAD sessions. We group session resets into three periods based on the date RIPE disabled and enabled timers: *Before 2002.11*, *2002.11 to 2006.11*, and *After 2006.11*. After disabling BGP timers at 2002.11, we can observe a left shift of the distribution, which indicates a drop in number of session resets. The median number of session resets of "*Before 2002.11*" is about 4 times of that of "*2002.11 to 2006.11*". This shows that disabling BGP timers did help reduce the number of session resets.

At 2006.11, when RIPE restored back the timers, the distribution shifts right but with a smaller scale. This is because the newer version of collector no longer had the blocking problem at dumping RIB files. Thus, even if we enabled Keepalive timers, we would not have as many resets as we had before Nov 2002. We also observed the similar distribution of number of session resets for other RIPE collectors.

6.3 Session Downtime

In this section, we measure the *silence period* and *recovery period* for the unnoticed side effect of disabling BGP timers.

Figure 14(b) shows the CDF of silence period for KAD sessions. Before disabling BGP timers, there are two consecutive sharp jumps at around 90 and 180 seconds silence time, which represent session resets trigger by 90 seconds and 180 seconds Holddown timers. However, after disabling Keepalive timers, these two jumps were basically eliminated and the CDF of silence period skewed to follow a long tail distribution. This is because after disabling BGP timers, BGP sessions could no longer detect failures such as connectivity problem at every timeout interval. These failures either went unnoticed or eventually detected by external signals such as TCP error, which negatively resulted in much longer silence time.

Figure 14(c) shows the cumulative percentage of recovery time for session resets. We observed that disabling BGP timers did change the distribution of recovery time. This seems counter-intuitive because Keepalive/ Holddown timers are expected to only affect the silence time but not the recovery time. One possible explanation is that though disabling timers does not change the recovery time for a given session failure, it could potentially change the *visibility* of some session failures.

More specifically, [15] observed that session failure can be categorized mainly into 4 groups: The first and second groups contains failures such as *admin resets*, *peer closed sessions*, which can recover very fast. The third group contains *local holdtimer expired* which spans middle range a

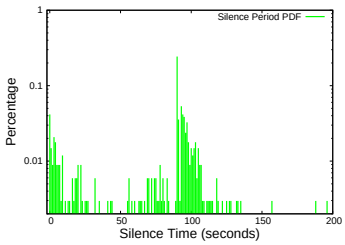


Figure 11: Sameple Silence Period Distribution

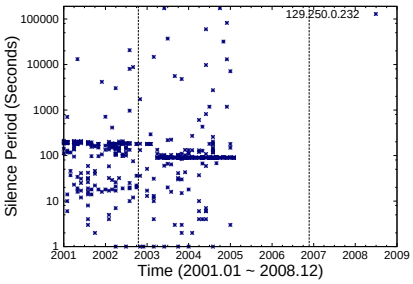


Figure 12: KAE Silence Period

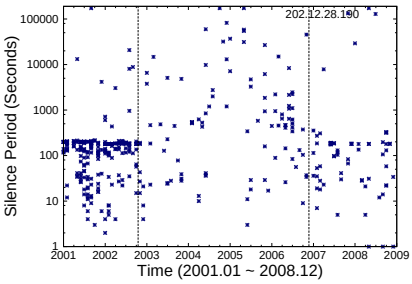
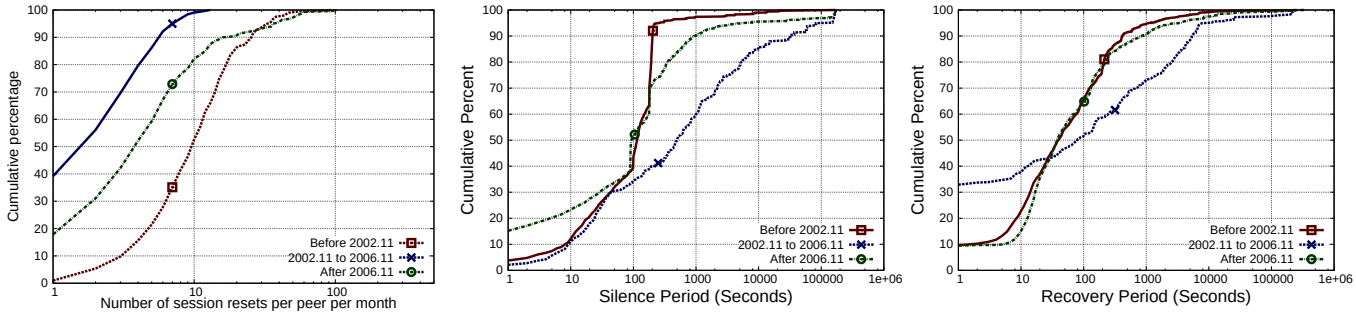


Figure 13: KAD Silence Period



(a) Number of Session Resets (b) Silence Period (c) Recovery Period

Figure 14: Impact of Disabling Keepalive Timer, RRC00.

downtime. And the fourth group contains *local router shutdown* and *peer de-configured* which requires very long recovery time. As the result, disabling Keepalive timers would make a BGP session *blind* to the third group of failures, and skew the distribution of recovery time to the other three groups, which have either much shorter or longer recover time. This explains the increase of percentage of both short recovery time and long recovery time in 14(c).

In this section, we have revealed, from RIPE session resets, that disabling Keepalive timers helped reduce the number of session resets. However, it also led to a long tail distribution of session silence time, during which failures went unnoticed and valuable BGP updates were missing. We thus recommend to avoid disabling Keepalive and Holddown timers, even it is allowed in the BGP specification[12]. In addition, researchers need to be aware of the long silence time when interpreting historical RIPE data, which might be caused by unnoticed BGP failures, but not that BGP becomes stable and quiet all of a sudden.

7. RELATED WORK

The RV/RIPE data quality is far from perfect because of measurement artifacts and missing data. Prior works have recognized the need to differentiate artifact table transfers from incremental updates. [16] uses BGP session state message to identify the start of a BGP session re-establishment. [17] uses MCT to detect the occurring and duration of table transfers from BGP update messages. [13] removes all duplicate announcement from the update stream, and [6] splits update stream into 30-second bins and discards any bin that contains more than 1000 prefixes. These approaches focus on cleaning up, but not quantifying and understanding the cause of measurement artifacts. Furthermore, as we showed in this paper, there were significant amount of session downtimes, during which valuable update messages are missing. Unlike artifact table transfers which could be filtered out, there is no way to recover missing historical data. It is critical to understand the cause of session failures to improve the monitor stability and thus the quality of BGP logged data.

[15] infers the root cause of session failures in one large ISP. By using syslog event, router configurations, and SNMP traffic data, their scheme provides a practical way to identify the direct cause of operational session failures. However, such information is unavailable from RV/RIPE to understand the failures between a data collector and its peering monitors. Also, [15] provides the normalized results for one ISP which might not fully represent the characteristics and the impact of session failures.

A similar work to this paper is [7], which checks the consistency of BGP data. Though, we focus on a longitudinal study of one particular contributor, session resets, to the inconsistency.

Last, [9] found that, for three paricular router vendors, the table transfer takes significantly longer time because of router timers that regulate the sending of updates, which potentially explains why we didn't observe clear correlation between the routing table size and the transfer time. Further investigation is needed to verify the vendors of RIPE/RoutView monitors.

8. CONCLUSION

This paper reports the first systematic assessment on the BGP session failures of RouteViews and RIPE data collec-

tors over the last eight years. Our results show that failures of the BGP monitoring sessions are relatively frequent, averaging a few session resets per monitor per month. Our measurement also show that the data collectors fail from time to time and contributed between 14% to 37% of the total session resets. Although some cases might be the result of intended administrative maintenance, they nevertheless affect the quality of the data being collected. To help users avoid the negatively impact caused by the BGP monitoring session failures, we have developed a web site which reports all the detected session resets with their occurring time and duration for all historical and new data onward. Please refer to Appendix for detail information.

In the process of analyzing BGP session resets in the historical data, we also found that disabling BGP's Keepalive timer leads to negative consequence of unnoticed session failures. We proposed an efficient algorithm to detect ISP peers that turned off BGP timers. Users of historical RIPE BGP data should take into account the potential long downtime and missing update for the affected peers in order to achieve reliable results.

How to make BGP sessions robust against transient packet losses remains an open problem both in BGP monitoring projects and in operational networks. As we discovered that the collectors' local problems is a major contributor to the monitoring session downtime, one possible improvement can be setting up a backup collector, and smoothly handing off the sessions from one collector to the other when maintenance is needed. Considering the importance of BGP monitoring without interruptions, using a backup collector can be a worthwhile investment.

9. REFERENCES

[1] BGPReset website - eMCT algorithm. "http://bgpreset.cs.arizona.edu/#methodology".

[2] RIPE Routing Information Service. http://www.ripe.net/projects/ris/.

[3] The RouteViews project. http://www.routeviews.org/.

[4] The RouteViews project - Data Archives. http://www.routeviews.org/update.html.

[5] MRT routing information export format. http://www.ietf.org/internet-drafts/draft-ietf-grow-mrt-07.txt, 2007.

[6] D. G. Andersen, N. Feamster, S. Bauer, and H. Balakrishnan. Topology inference from bgp routing dynamics. In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 243–248, New York, NY, USA, 2002. ACM.

[7] A. Flavel, O. Maennel, B. Chiera, M. Roughan, and N. Bean. CleanBGP: Verifying the Consistency of BGP Data. In *Internet Network Management Workshop 2008*, Oct. 2008.

[8] L. Gao. On inferring autonomous system relationships in the Internet. *ACM/IEEE Transactions on Networking*, 9(6):733–745, 2001.

[9] Z. B. Houidi, M. Meulle, and R. Teixeira. Understanding Slow BGP Routing Table Transfers. In *IMC 2009*, Nov. 2009.

[10] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang. Phas: A prefix hijack alert system. In *USENIX Security Symposium*, 2006.

[11] R. Oliveira, B. Zhang, D. Pei, R. Izhak-Ratzin, and L. Zhang. Quantifying path exploration in the internet. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 269–282, New York, NY, USA, 2006. ACM.

[12] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271 (Draft Standard), Jan. 2006.

[13] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang. Bgp routing stability of popular destinations. In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 197–202, New York, NY, USA, 2002. ACM.

[14] G. Siganos, M. Faloutsos, P. Faloutsos, and C. Faloutsos. Power-laws and the as-level internet topology. In *ACM/IEEE Transactions on Networking*, August 2003.

[15] L. Wang, M. Saranu, J. Gottlieb, and D. Pei. Understanding bgp session failures in a large isp. pages 348–356, May 2007.

[16] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. Observation and analysis of bgp behavior under stress. In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 183–195, New York, NY, USA, 2002. ACM.

[17] B. Zhang, V. Kambhampati, M. Lad, D. Massey, and L. Zhang. Identifying bgp routing table transfers. In *MineNet '05: Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, pages 213–218, New York, NY, USA, 2005. ACM.

APPENDIX

A. BGPRESET WEBSITE

We have developed a website, *BGPReset*, which reports monitoring session failures for three RouteViews collectors (OREG, LINX, EQIX) and three RIPE collectors (RRC00, RRC01, RRC02). The website could be accessed at the url, `http://bgpreset.cs.arizona.edu/`.

Two types of failure information are reported:

- Session Resets

  The occuring time of session resets, together with the succeeding session downtime and duration of the table transfer when the session is re-established.

- Collector Restarts

  The occuring time of each collector's outage/restarts, identified by synchronized sesssion resets of all sessions on the same collector, including the number of monitor peers affected.

Users can either use the exported query interface to lookup session resets of particular collector, monitor, time period, etc, or download raw result files for offline processing.