CleanBGP: Verifying the Consistency of BGP Data

Ashley Flavel* Olaf Maennel† Belinda Chiera* Matthew Roughan* Nigel Bean*
*School of Mathematical Sciences, University of Adelaide †TU-Berlin/T-Labs

***Abstract*—BGP data contains artifacts introduced by the measurement infrastructure which can substantially affect analysis. This is especially important in operational systems where "crying wolf" will result in an operator ignoring alarms. In this paper, we investigate the causes of measurement artifacts in BGP data — cross-checking and using properties of the data to infer the presence of an artifact and minimize its impact.**

We have developed a prototype tool, CleanBGP, which detects and corrects the effects of artifacts in BGP data, which we believe should be used prior to the analysis of such data. CleanBGP provides the user with an understanding of the artifacts present, a mechanism to remove their effects, and consequently the limitations of results can be fully quantified.

I. Introduction

BGP data is collected to analyze the dynamic changes to a router's view of the Internet. However, the measurement infrastructure can introduce undesirable artifacts, for instance missing or re-ordered updates, partially recorded tables and monitoring-link failures. Automated systems which frequently report anomalies merely as a result of measurement artifacts are not acceptable, and will be ignored by network operators. In this paper, we cross-check BGP data to verify it is consistent, and in cases when it is not, use 'clues' contained in the data to discover the source of the inconsistency and ameliorate its impact on the data. BGP data has been used for many network management tasks such as debugging routing problems [1], anomaly detection [2] and policy inference [3] as well as router table analysis [4], [5]. However, all analyses require an inherent understanding of the input data together with its limitations. This understanding must be built-in to any tool usable by operators and forms the motivation for the design of the pre-processing tool described in this paper — CleanBGP.

It might be easy to argue "improve the measurement apparatus". Obviously, we would like equipment to be as accurate as possible, and improvements to such equipment are complementary to our work. However, regardless of such improvements, it is vital — particularly in operational systems — to calibrate the accuracy of all measurement devices [6]. The initial hypothesis of all measurement apparatus should be that it is flawed, and data taken from the apparatus can be considered accurate only when this hypothesis has been proved false. We have limited resources for such calibration, however we do have the capability to perform consistency checks on the data. It is this methodology that has allowed us to detect some artifacts that would never otherwise have been found through a "hunt and peck" approach.

The goal of collecting BGP data is to record the state of a single BGP router in the Internet. In-order to analyze this data, it must be recorded on disk where it can be processed offline. This can be done on the routers themselves, but this option is typically not chosen as it requires significant resources and can impact the operational stability of the router. Hence, collecting BGP data is undertaken with a more passive approach. A software route monitor establishes a BGP session with the operational router, possibly over multiple physical links. The operational router sends all its best path updates to the monitor as if it were part of the Internet's routing system. The monitor records this data to disk. Each component of the measurement infrastructure can fail. For instance, there can be bugs in the monitor implementation causing updates to not be recorded [7], or recorded non-chronologically — which is undesirable due to the hard-state nature of BGP. Further, the BGP session between the monitor and the router can fail causing missed updates, and during the re-establishment of the session a BGP update storm occurs as all routes are re-advertised. Including these updates in further analyses can result in starkly different conclusions [8].

Our methodology, which forms the basis for CleanBGP has several stages. First, we examine how the data is collected and explain how cross-checking the data can highlight the presence of measurement artifacts (Section II). Second, if a measurement artifact (described in Section III) is detected based on its characteristics (Section IV), we use techniques presented in Section V to estimate the interval affected and determine its source. Finally, we either exclude the data from further analysis or estimate the actual routing behavior using techniques based on the classification of the measurement artifact (Section VI).

Measurement artifacts are binary in nature. They are either present or not. However, no binary indicator is available to inform us of their presence. Hence, CleanBGP uses multiple characteristics of the data for detection. In Section VIII we investigate the frequency of detected artifacts and their effect on the measured data. We find our consistency check detects problems in 5% of cases with 81% of these caused by updates recorded in non-chronological order. A further 10% were caused by session resets, however, our approach also discovered resets occurred frequently even when no inconsistencies were present. Analysis of BGP data may or may not be substantially affected by some artifacts. However, knowledge of their existence is vital so a judgement on their effect can be made. CleanBGP may form a pre-processing step to any BGP analysis such as a monitoring system [9], or other network management tasks.

II. Data Consistency

BGP data is collected to represent the routing state of a router at a given time. This state is unique. Consequently, any data representing this state should be consistent and is the basis of the check outlined in this section.

	Policy Unchanged Session Reset	Policy Changed Session Reset	Incomplete Table	Missing Updates	Update Ordering
Additional prefixes in constructed table	+	+	✓	+	+
Different prefixes in constructed table				+	+
Missing prefixes in constructed table				+	+
Almost simultaneous updates for inconsistent prefixes					✓
Oldest prefix during inter-table interval	✓	✓			
No routing activity for extended period	+	+		+	
Burst of unique prefixes	✓	✓			
Burst of duplicate updates	✓				
State Information	+	+			

TABLE I

Data characteristics of measurement artifacts. Legend: ✓ Characteristic necessary for artifact presence. + Characteristic strongly indicative (but not necessary) for artifact presence.

Current generation routers support the real-time collection of routing changes. However the mechanism it uses (via the `debug` command) is resource intensive and requires administrator access to the router. Hence, it is often not enabled on operational routers. The current best-practice to determine a router's current view of the Internet is for a software router (for instance Quagga [10] or OpenBGPD [11]) to be used as a route monitor. The monitor establishes a BGP session with an operational router, which treats the monitor as any other router. Subsequently, all changes to the operational router's best route are announced along the session to the monitor, which records these changes to disk and periodically dumps an entire table. This BGP update collection procedure is undertaken by public route monitors [12], [13] as well as internally by ISPs [14].

The two types of BGP data collected — tables and updates — are views of the same system. Consequently, they should be consistent. BGP is a hard-state routing protocol, where updates are only sent once. Thus, constructing a table at time t_2 by combining the table at some time t_1, and all updates received in the interval $[t_1, t_2]$ should be consistent with the table recorded at time t_2[1]. This was not always the case in recorded BGP data. However, we were able to characterize the causes of inconsistencies into four main types of measurement artifacts and use characteristics of the data to detect them (detailed in Sections III and IV).

III. Measurement Artifacts

Measurement artifacts occur when the data stored by the monitor does not reflect the real state of the router under observation. In this section we consider the artifacts we unearthed and their characteristics, a summary of which is included in Table I.

A. Session Failures and Resets

The collection of updates relies on a BGP session between the route monitor and an operational router. If this session fails, no updates are recorded during the failure interval and the consistency check may fail. Monitor sessions are particularly vulnerable to failures as they often use multi-hop sessions (that cross multiple physical links) and are more vulnerable to timeouts as a result of link congestion. When the session is re-established, the entire table is re-advertised, resulting in a large influx of routing updates. These updates are not representative of changes in the observed network. Such updates can even cause a "BGP update storm" [8] that is not real!

B. Incomplete Tables

It is possible that a routing table is not fully written to disk. Consequently, the table is not representative of the operational router's view of the Internet.

C. Missing Updates

Kong [7] discovered some updates were not decoded by the monitor and consequently would be missing from both the update stream and recorded tables. This bug has since been corrected. However, if an entry is in a recorded table (indicating it has been decoded), but not in the update stream, then update files may be corrupted. This occurred for almost an hour on Jan 21, 2007 at RIPE route monitor RRC01.

D. Update Ordering

We discovered that when updates occur almost simultaneously (on the order of seconds), the ordering of updates was not consistent with the recorded table. This may be a bug in the software. For this analysis we assume the updates were *recorded* in the incorrect order. The alternative is that updates were *applied* in the incorrect order to the table — more concerning as the table would then represent an *invalid* routing state. Software routers are used in some cases as replacements for operational routers to reduce costs. An operational router with an invalid state can seriously affect network performance. To determine which alternative is the cause, an external monitoring technique such as a wire-tap would be required and is beyond the scope of this work.

E. Other Artifacts

Several other artifacts were discovered whilst analyzing the consistency of data. On some occasions, such as in the table recorded at RRC00 on May 1, 2008 for the prefix 84.205.80.0/24, two routes learned from the neighbor 202.12.29.64 are recorded even though BGP does not allow this. One route appears to be valid (the second has a null AS number and no AS Path). Further, no update for the second route is recorded in the update stream. Interestingly, in a later table, this entry is replaced, without being explicitly withdrawn, by a route learned from neighbor 193.136.5.1 with a null AS and the same originating time. The new route is replaced later again by a route from 12.0.1.63 (with AS7018) and a new originating time — although the update is not present in the update stream and the consistency check fails.

Another artifact we discovered was when updates were not applied to the table. This occurred to several tables recorded at RRC00 on April 30, 2008. Here, we discovered several

[1] In addition, table dumps are not atomic. They take a period of time to write to disk – from several seconds to minutes depending on the size and number of monitored router tables. We do not consider any difference caused by an update arriving during this time as a failed table consistency check.

hundred updates occurring up to hours before the recorded table time and not applied to the table. This is confirmed by the originating time of the routes in the recorded table. This could be caused by the re-ordering of updates, however, the timestamp of the consecutive updates is much further apart than any other observed update re-ordering (minutes in contrast to seconds). Consequently, we believe some updates may not have been applied to the table. A table recorded on this day also took over 20 minutes to write to disk indicating a possible monitor failure. Other tables from the same monitor generally took less than 20 seconds to write to disk.

A third artifact we discovered was the alteration of originating AS. For example at RRC01 on Jan 25, 2007 the originating AS of the prefix 203.10.62.0/24 recorded in the table is AS23456. However, updates indicate it is AS2.2. This may be an issue with the software router or data recording process.

IV. Detecting Artifacts

The consistency check outlined in Section II is able to detect many inconsistencies. However, consistent data does not necessarily indicate an interval is free from measurement artifacts. Hence, we use multiple characteristics from the data to detect and classify measurement artifacts (see Table I).

A. Table Comparison

The consistency check outlined in Section II is also a characteristic we can use to detect artifacts. If a session reset occurs between two successive table dumps, additional prefixes may be in the constructed table when compared to the second recorded table. Consider the example in Fig. 1. We construct the table at time t_2 by applying the updates in chronological order to the table at t_1. However, during a period of downtime several updates on the operational router (W1, W5, A2) are not recorded on the route monitor. All prefixes in the table after the downtime are sent to the monitor. Hence, the withdrawals (W1, W5) are not recorded on the route monitor and thus if not re-advertised before a table is recorded, will result in differences between the constructed and recorded table at time t_2. Note that missing announcements (such as A2) will be delayed as they are announced during the re-establishment. If an announcement of a prefix withdrawn during the downtime occurs after the downtime (such as A5), the missing withdrawal W5 will not cause a data inconsistency.

Session resets are not the only cause of a table comparison failure. Any prefix without an equivalent route in both the constructed and recorded tables may be affected by non-chronological recording of updates. If an update occurs 'almost simultaneously' to the last update received, then we declare it as responsible for the inconsistency.

If there are missing or different routes in the constructed table, and the non-chronological ordering of routes cannot account for any discrepancies, it is likely that updates have not been recorded. If additional prefixes are in the constructed table, but other characteristics discount the possibility of a session reset, then it is likely that the recorded table is incomplete.

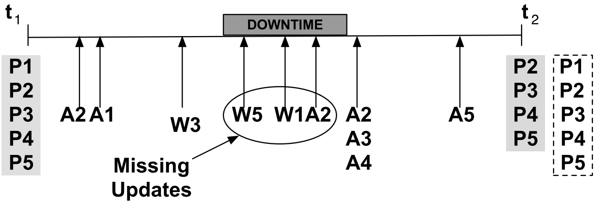

Fig. 1. The recorded table (shaded) at time t_2 is compared to a table constructed (dashed) from the table at t_1 and all updates *recorded* in the interval $[t_1, t_2]$. The announcement of prefix 1 is annotated by A1. The withdrawal of prefix 1 is annotated by W1 and so forth.

B. Oldest Prefix

Session resets cause the entire table of the operational router to be re-advertised. Consequently, if a session reset occurs at a time t_1, the oldest prefix in the recorded table at time t_2 can not be prior to t_1. However, we cannot assume that a session reset occurred at the timestamp of the oldest-prefix. Normal routing operation will result in all prefixes being re-announced at some point and we cannot say a session reset definitely occurred at the time of the oldest-prefix. However, a majority of the prefixes in the table are stable [5], [15]. Consequently, with regular snapshots (RIPE records them at 8 hour intervals) if the oldest prefix lies between the current and previous table snapshots, it is indicative that a session reset has actually occurred.

C. State Information

State information is included in some data sources to indicate the up and down times of a BGP session. However, state information can be missing and does not identify other measurement artifacts. Further, in monitors such as Route-Views state information is not even recorded. This information is used in CleanBGP by default, but for the purposes of this analysis, it is used purely as validation of session resets.

D. Downtime

BGP undergoes constant changes, so a long period where no updates are received can be indicative of a session reset or a failure to write to disk. However, receiving no announcements for a period of time may be part of normal routing behavior, especially in BGP tables with a relatively small number of prefixes. The recording of keep-alives — messages sent specifically to keep a BGP session alive during low routing activity — allows a low downtime threshold to be set, enabling the detection of measurement artifacts as early as possible.

E. Session Re-establishment

When a BGP session is re-established after a failure, all routes in the table are re-announced as fast as possible. The session re-establishment phase has two main characteristics:- a large number of prefixes announced in a short interval and a large number of non-table altering (or duplicate) updates.

BGP changes occur frequently but only to a subset of prefixes [15]. Hence, the table after a session failure is likely to be similar — however not necessarily identical — to the table prior to a session failure. If a large number of unique prefixes[2] are announced in a short interval it is indicative of a session re-establishment. In addition, as a large number of

[2] Defined by counting only one of the collection of announcements for each prefix in the given interval; this removes the effect of highly active prefixes.

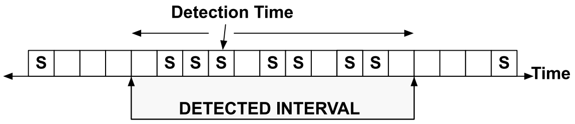

Fig. 2. 'S' indicates a suspicious interval. The detected interval is the largest contiguous collection of suspicious bins and isolated non-suspicious bins that include the detection time.

routes do not change, a substantial number of updates will not affect the routes in the constructed table (policy-unchanged reset). We define such updates as *duplicate announcements*. If an administrator of the operational router changes policy during the downtime, the number of duplicate announcements may be low (policy-changed reset).

A re-establishment after a session failure may not be complete due to a persistent failure resulting in multiple partial re-establishments. However, in this case, we are likely to see a substantial number of duplicate announcements, or a long downtime. The technique of Zhang *et al.* [16] does not account for such cases. Unlike other characteristics, thresholds are required for the session re-establishment characteristic. This threshold must be high enough not to classify normal routing operation as a measurement artifact, but must be low enough to ensure we do not miss any artifacts. We outline how we obtain these thresholds in Section VII.

We use all of the above characteristics to detect measurement artifacts in BGP data sources. CleanBGP uses a sliding window on the update stream to initially detect a measurement artifact — an extended downtime or a burst of unique prefixes/duplicate announcements is indicative that a measurement artifact may be occurring. State information can also be used to detect session failures. When a table is available for comparison, we compare the constructed table with the recorded table and examine the oldest prefix in the table for further evidence as to whether a measurement artifact (and what type) occurred during the interval between the previous and current table. When a measurement artifact is detected, CleanBGP enters the localization phase.

V. Localization in Time of Measurement Artifacts

Session resets and missing updates are artifacts that span a period of time. The detection of these measurement artifacts simply indicates that a measurement artifact is occurring. The second phase of CleanBGP is to localize the affected interval. Our desire is to precisely identify all data that is representative of the operational router's behavior and all data which is not. However, this is difficult when there are no markers in the data to indicate which data is representative. State information recorded in some data sources can provide a starting point, however state information can be delayed, missing and does not indicate the conclusion of the re-establishment phase[3]. For other measurement artifacts, no meta-data is available and cross-referencing the data is our only feasible option.

If the sliding window detects a possible measurement artifact, we localize it by considering small disjoint bins surrounding the detected time. If a bin contains no updates/keep-alives or a large number of unique prefixes/duplicate announcements, we declare the bin as suspicious. We localize the measurement artifact as the largest contiguous collection of suspicious bins and isolated non-suspicious bins that include the detection time. Over-estimation of the affected interval is preferred to under-estimation as we want to provide a guaranteed level of accuracy in the data.

The above technique for localizing the interval affected by session resets and missing updates is ideal for applications requiring real-time data (such as network monitoring tools). However, some measurement artifacts may not be detected by such a threshold based technique. When a table is available for comparison it may have characteristics of a session reset or missing updates. If this is the case, and the sliding window was unable to precisely identify the affected interval, we conservatively declare the entire interval between two successive tables part of a measurement artifact. Other artifacts such as re-ordered updates are only able to be detected by examining a recorded table. If such an artifact is detected, no localization is required and the process moves directly to the cleaning phase.

VI. Cleaning Data

We must be very cautious when cleaning data to avoid unnecessarily altering data. To this end, we 'mark' updates and table entries which we alter and clearly identify the interval cleaned. Consequently, applications using the BGP data can determine what data to include or exclude.

The most obvious form of data cleaning is *exclusion* — removing any interval affected by measurement artifacts from further analysis. This would be ideal for applications sensitive to large numbers of updates or long periods of downtime and is the approach we recommend if contiguous routing updates are not required. However, as many applications require a continuous stream of data this is not always an attractive solution. Thus, we now introduce a new technique for the *estimation* of routing behavior during measurement artifacts.

A. Session Resets

Removing all duplicate announcements has been used to minimize the effect of session resets [15], [18]–[20]. Duplicates reflect *no* change in the routing state, however they can be caused by internal AS routing changes [8], [21]. We mark all duplicate announcements as part of a measurement artifact *only during a detected session reset interval*. Thus, we alter the minimal amount of data and ensure all updates which reflect routing changes during the downtime are still present in the data although they may be delayed. We mark all other updates during this interval as being possibly delayed.

Prefixes may be withdrawn on the operational router during the downtime of a session. These 'ghost' withdrawals are only noticed when comparing a constructed table to a recorded table as the recorded table will not include the prefixes withdrawn during the downtime. Hence, we can assume these prefixes were withdrawn during a session reset and consequently estimate the time the withdrawals occurred (at the conclusion of the session reset interval). If multiple session resets occur during a single inter-table interval, withdrawals are placed where they are consistent with multiple session failures.

B. Incomplete Tables

A table at any time is able to be constructed from updates and a previous table. Any table not completely recorded to disk can hence be ignored.

[3] An End-of-RIB marker is proposed in [17].

C. Missing Updates

If missing updates are detected, it is possible to estimate the actual routing behavior during this time by adding announcements of routes at the originating time recorded in the table. In addition, any prefixes not in the recorded table can be withdrawn during the detected interval. If updates are not able to be added during this time to ensure consistency, for example when a later announcement obscures an added withdrawal, the entire interval between consecutive tables is declared an interval affected by a measurement artifact.

D. Update Ordering

If a non-chronological ordering of updates is detected, the order of these updates can be permuted such that the constructed table is consistent with the recorded table.

When a measurement artifact is discovered within the measurement infrastructure of a single AS, the correlation between router decisions may be used to predict routing behavior during a measurement artifact [14].

VII. Parameter Selection

In this section we outline our default parameter selections for CleanBGP which we summarize in Table II.

A. Sliding Window Length

The sliding window is used to detect session resets using the downtime and re-establishment phases. It must be long enough to identify a session reset from the unique prefixes and duplicate announcements (including the possibility of multiple partial session resets), whilst being short enough such that normal routing behavior is differentiated from session reset behavior. For a full-feed operational router, we found a session generally re-established in less than 10 minutes. Routers with partial-feeds re-establish more quickly as they have fewer prefixes. We found after experimenting with several sliding window lengths that detecting measurement artifacts was quite insensitive. We use a conservative sliding window of 1 hour to ensure multiple partial session resets can be captured.

B. Re-establishment Phase Thresholds

The unique prefixes threshold must be large enough to not classify normal routing behavior as an artifact, whilst small enough to detect all resets. By default we choose 50% of the previous recorded table size as the threshold.

The duplicate announcements threshold can be lower than the unique prefixes threshold as duplicates are less common. We used 25% of the previous recorded table size to identify faults causing a session to persistently fail whilst re-establishing. The low threshold is useful for detecting problems as early as possible — required for real-time applications.

We have developed an automated technique to tune these parameters and refer the reader to [22] for further details.

C. Downtime Threshold

An update or keep-alive message must be received within the hold-time interval for a BGP session to remain alive. Consequently, a bin of hold-time duration which has no activity is an indicator that the session is down. If no keep-alives are recorded as with RouteViews, the downtime threshold would need to be configured based on the previous non-suspicious routing activity. When an operational router has a full feed, it is likely the inter-arrival time of updates will be low and consequently a low downtime threshold can be set.

Description	Default Value
Sliding Window Length	1 hour
Bin Length	Hold time (180 seconds for RIPE)
Unique Prefixes	50% of table size
Duplicate Prefixes	25% of table size
Downtime Threshold	1 Bin
Suspicious Bin Unique Prefixes	10% of table size or 300 updates
Suspicious Bin Duplicates	10% of table size or 300 updates

TABLE II
Default Parameter Settings

D. Bin Length

Disjoint bins are used to determine suspicious intervals when localizing measurement artifacts. We use the hold-time as the default bin length. If no keep-alive or update is received during this time, the BGP specification states the session is down.

E. Suspicious Bin Thresholds

An interval is declared suspicious if it may be part of a session reset or missing update interval. These thresholds are only used in the localization phase of CleanBGP when an artifact has been detected by the sliding window. In addition the characteristics are considered over a shorter interval. Consequently, we set more aggressive parameters than for the sliding window — 10% of the table size for both duplicate announcements and unique prefixes. Further, we have an absolute value for the number of updates received. We found this is needed when several monitoring sessions fail simultaneously, likely due to the failure of a shared physical link or monitor failure, and all re-establish in tandem. The monitor is physically unable to write all updates to disk instantaneously — full feed BGP neighbors currently have approximately $250,000$ prefixes and for instance the route monitor RRC00 at RIPE has 13 of these neighbors. Accordingly, the burst of updates appears spread out in comparison to a single session failure. We use a low absolute threshold together with a proportion of the table to mark an interval as suspicious. We found 300 updates per bin was a good default parameter, although it can be tuned based on the limitations of the route monitor. In addition, if no updates or keep-alives are received in the interval, we assert the session is down and the bin is also suspicious.

VIII. Results

We analyzed 260 (BGP neighbor, month) pairs of RIPE [12], finding inconsistencies in 4.7% of the $23,099$ table comparisons. A summary of the results is shown in Table III. Of the 4.7% of problems during the consistency check, 81% can be attributed to non-chronological recording of updates. Although generally less than 10 prefixes were affected, we found cases of up to 712 prefixes affected by this artifact. We found several instances where updates were applied in a permuted order with timestamps up to 16 seconds apart. However, 76% of these prefixes had updates with an equivalent timestamp but were written to file in the incorrect order. All such instances of non-chronological updates we discovered were caused by a withdrawal being written to disk prior to an announcement, but applied to the table after.

Monitor, Month	BGP Neighbors	Table Comparisons	Consistency Check Failures	Missing Updates	Unknown	Re-ordered Updates	Reset - Policy Unchanged/Changed	State Info Detected/Missed	Reset No State
RRC01, Jan-07	95	8274	150 (2%)	14	67	18	694 / 31	1158 / 27	45
RRC01, Apr-08	91	8102	722 (9%)	0	15	668	1121 / 102	852 / 17	827
RRC02, Apr-08	37	3282	112 (3%)	0	0	107	133 / 21	252 / 9	26
RRC02, May-08	37	3441	107 (3%)	0	0	98	121 / 20	235 / 17	22

TABLE III
ARTIFACTS AND CONSISTENCY CHECK FAILURES IN RIPE DATA.

State information indicated 2567 state changes (up and down) in the data analyzed. CleanBGP groups numerous session failures occurring close in time into a single interval. In addition, a session reset may not cause an inconsistency in the recorded table. Hence, although we detect 2243 of such intervals, they contribute only 10% to the causes of a consistency check failure. If a session continually fails, identifying individual resets is difficult and the technique in [16] is inadequate for this purpose. We claim CleanBGP to be successful in identifying session resets if state information is inside a localized interval. We found 97% of state information was included within localized artifact intervals. We investigated the cause of the 3% of state information outside a localized interval. Part of the re-establishment process of a session is to send a keep-alive. CleanBGP believes this is a non-suspicious interval. Multiple session re-establishments during a failure interval result in a number of keep-alives appearing as normal operation. In practice, CleanBGP would use state information to assist in the detection and localization of measurement artifacts and hence be more accurate.

We also detected intervals which had all the characteristics of session resets, but no state information. This may indicate state information is missing, outside of a detected interval (i.e. localization of the reset was inadequate) or parameters used were overly aggressive on occasions. For the intervals we examined, most reset intervals contained state information. In April, a single highly active BGP neighbor resulted in 737 of the 827 detected reset intervals without state information.

In the interval 17:33:17 and 18:25:02 UTC on January 21, 2007 no updates were recorded for any BGP neighbors of RRC01. We did not discover any partially recorded tables.

The ‘Unknown’ category represents the consistency check failures for which the oldest prefix discounted the possibility of a session reset and no period of downtime was detected indicating missing updates. We investigated these cases individually finding many were caused by updates occurring several seconds prior to a table dump but not being recorded in the table dump, that is the non-atomic nature of the table spanned outside the timestamps of the recorded interval of the table. The other artifacts described in Section III-E were also found in this category.

IX. Discussion

We have seen the benefit throughout this paper of state information and keep-alive messages for determining measurement artifacts. In addition, more frequent table dumps would ensure even greater identification of measurement artifacts as it would provide a greater ability to cross validate. If this technique was undertaken automatically during the collection process, a recorded table consistent with a constructed table could be discarded as it provides no additional information. This process would increase the accuracy of data, whilst not increasing storage requirements.

We envisage CleanBGP as the first step to all BGP data analysis. Currently the tool is in prototype stage, although we aim to release a public version in the near future.

Acknowledgement

The authors acknowledge the support of the Australian Research Council grant DP0557066 and the ARC Communications Research Network. This work was undertaken while Olaf Maennel was employed by the University of Adelaide.

References

[1] A. Feldmann, O. Maennel, M. Mao, A. Berger, and B. Maggs, “Locating Internet Routing Instabilities,” in *ACM SIGCOMM*, 2004.

[2] M. Roughan, T. Griffin, Z. M. Mao, A. Greenberg, and B. Freeman, “IP Forwarding Anomalies and Improving their Detection Using Multiple Data Sources,” in *ACM SIGCOMM Workshop on Network Troubleshooting*, 2004.

[3] W. Mühlbauer, O. Maennel, S. Uhlig, A. Feldmann, and M. Roughan, “Building an AS-Topology Model that Captures Route Diversity,” in *ACM SIGCOMM*, 2006.

[4] G. Huston, “Analyzing the Internet BGP Routing Table,” *The Internet Protocol Journal*, vol. 4, no. 1, March 2001.

[5] A. Flavel, M. Roughan, N. Bean, and O. Maennel, “Modeling BGP Table Fluctuations,” in *20th International Teletraffic Congress*, 2007.

[6] V. Paxson, “Strategies for Sound Internet Measurement,” in *ACM Internet Measurement Conference*, 2004.

[7] H. Kong, “The Consistency Verification of Zebra BGP Data Collection,” Agilent Labs, Tech. Rep.

[8] 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 *ACM Internet Measurement Workshop*, 2002.

[9] D. Matthews, Y. Chen, H. Yan, and D. Massey, “BGP Monitoring System,” NANOG 40, 2006.

[10] K. Ishiguro, “Quagga Routing Suite,” `www.quagga.net`.

[11] H. Brauer and C. Jeker, “OpenBGPD,” `www.openbgpd.org`.

[12] RIPE NCC, `www.ripe.net`.

[13] University of Oregon RouteViews project, `www.routeviews.org`.

[14] A. Flavel, J. McMahon, A. Shaikh, M. Roughan, and N. Bean, “Humpty Dumpty: Putting iBGP Back Together Again,” Technical Report, 2008, `http://internal.maths.adelaide.edu.au/people/aflavel/humpty.pdf`.

[15] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang, “BGP Routing Stability of Popular Destinations,” in *ACM Internet Measurement Workshop*, 2002.

[16] B. Zhang, V. Kambhampati, M. Lad, D. Massey, and L. Zhang, “Identifying BGP Routing Table Transfers,” in *ACM SIGCOMM Workshop on Mining Network Data*, 2005.

[17] S. Ramachandra, Y. Rekhter, R. Fernando, J. Scudder, and E. Chen, “Graceful Restart Mechanism for BGP,” 2007, Internet Draft.

[18] O. Maennel and A. Feldmann, “Realistic BGP Traffic for Test Labs,” in *ACM SIGCOMM*, 2002.

[19] M. Caesar, L. Subramanian, and R. Katz, “Towards Localizing Root Causes of BGP Dynamics,” UCB/CSD-03-1292, Tech. Rep., 2003.

[20] D.-F. Chang, R. Govindan, and J. Heidemann, “The Temporal and Toplogical Characteristics of BGP Path Changes,” in *Proc. ICNP*, 2003.

[21] C. Labovitz, R. Malan, and F. Jahanian, “Origins of Internet Routing Instability,” in *IEEE INFOCOM*, 1999.

[22] A. Flavel, O. Maennel, W. Mühlbauer, M. Roughan, B. Chiera, and N. Bean, “CleanBGP: The First Step Towards an Internet Alarm System,” Technical Report, 2007.