A Comparative Analysis of BGP Anomaly Detection and Robustness Algorithms

Kotikapaludi Sriram Oliver Borchert Okhee Kim Patrick Gleichmann Doug Montgomery
National Institute of Standards and Technology
Gaithersburg, Maryland, MD 20899
{*ksriram, borchert, okim, patrick.gleichmann, dougm*}*@nist.gov*

Abstract

We present an evaluation methodology for comparison of existing and proposed new algorithms for Border Gateway Protocol (BGP) anomaly detection and robustness. A variety of algorithms and alert tools have been proposed and/or prototyped recently. They differ in the anomaly situations which they attempt to alert or mitigate, and also in the type(s) of data they use. Some are based on registry data from Regional Internet Registries (RIRs) and Internet Routing Registries (IRRs) - an example is the Nemecis tool. Others such as the Prefix Hijack Alert System (PHAS) and the Pretty Good BGP (PGBGP) are driven by BGP trace data. The trace data is obtained from Reseaux Internet Protocol Europeens - Routing Information Service (RIPE-RIS), Routeviews, or a BGP speaker where the algorithm operates. We propose a new algorithm that combines the use of both registry and trace data, and also makes some key improvements over existing algorithms. We have built an evaluation platform called TERRAIN (Testing and Evaluation of Routing Robustness in Assurable Inter-domain Networking) on which these algorithms can be tested and empirically compared based on real and/or synthetic anomalies in BGP messages. We will present a variety of results providing interesting insights into the comparative utility and performance of the various BGP robustness algorithms.

1. Introduction

There has been significant interest recently in detecting and mitigating routing anomalies in the operation of the Border Gateway Protocol (BGP) [1]-[35]. Major incidents have been reported in recent months and years that involved compromise of the routing infrastructure on the Internet [1]-[7]. These have resulted in misrouted traffic and denial of services. Prefix hijack attacks in which a BGP update with false origin information is propagated has been the subject of multiple recent studies. These attacks need to be detected early and accurately so that their propagation through the Internet can be stopped and damage can be mitigated quickly.

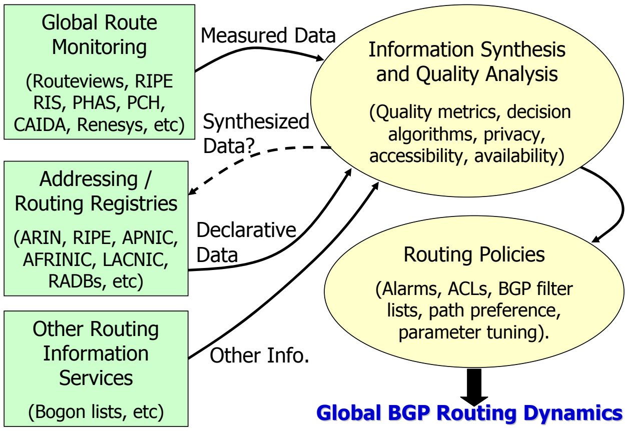

Global Route Monitoring
(Routeviews, RIPE RIS, PHAS, PCH, CAIDA, Renesys, etc)
Measured Data
Information Synthesis and Quality Analysis
(Quality metrics, decision algorithms, privacy, accessibility, availability)
Synthesized Data?
Addressing / Routing Registries
(ARIN, RIPE, APNIC, AFRINIC, LACNIC, RADBs, etc)
Declarative Data
Routing Policies
(Alarms, ACLs, BGP filter lists, path preference, parameter tuning).
Other Routing Information Services
(Bogon lists, etc)
Other Info.
Global BGP Routing Dynamics

Figure 1. A view of Internet routing data sources and their analysis and application for improving routing robustness.

Figure 1 depicts a conceptual view of Internet routing data sources and their analysis and application for improving routing robustness. In order to develop routing policies that can ensure global routing robustness, we need algorithms that take into consideration all types of available routing data. As shown in Figure 1, measured data is available from global BGP monitoring systems such as Routeviews, Reseaux Internet Protocol Europeens - Routing Information Service (RIPE-RIS), Cooperative Association for Internet Data Analysis (CAIDA), etc. Further, declarative data is available from addressing and routing registries such as RIPE, American Registry for Internet Numbers (ARIN), Routing Assets Database (RADB), etc. Throughout this paper, by RADB we mean the collective routing information of many organizations that is registered or mirrored at Merit Network's RADB[40]. There are other BGP information sources available such as bogon lists[36]. While there are a variety of data sources that can provide the ba-

978-0-7695-3568-5 2009

DOI 10.1109/CATCH.2009.20

sis for new BGP robustness techniques, it is fair to say that none of them were specifically designed for that purpose. As a result, the completeness, correctness, freshness, and consistency of the data derived from these sources must be taken into account by any decision algorithms based upon them. Routing robustness mechanisms or policies can include generation of alarms when anomalies are detected, Access Control Lists (ACLs) and BGP filter lists to prevent or allow acceptance and forwarding of specific addresses or prefixes, and adjusting path preference.

In this paper, we present an evaluation methodology for comparison of existing and proposed new algorithms for BGP anomaly detection and robustness. A variety of algorithms and alert tools have been proposed and/or prototyped recently. They differ in the anomaly situations which they attempt to alert or mitigate, and also in the type(s) of data they use. Some are based on registry data from Regional Internet Registries (RIRs) and Internet Routing Registries (IRRs) - an example is the Nemecis tool. Others such as the Prefix Hijack Alert System (PHAS) and the Pretty Good BGP (PGBGP) are driven by BGP trace data. The trace data is obtained from global BGP monitoring infrastructures (e.g., RIPE-RIS, Routeviews) or a BGP speaker where the algorithm operates. We propose a new algorithm that combines the use of both registry and trace data, and also makes some key improvements over existing algorithms. We have built an evaluation platform called TERRAIN (Testing and Evaluation of Routing Robustness in Assurable Inter-domain Networking) on which these algorithms can be tested and empirically compared based on real and/or synthetic anomalies in BGP messages. We will present a variety of results providing interesting insights into the comparative utility and performance of the various BGP robustness algorithms. Our objective is to share these early insights and invite feedback from the community to refine the TERRAIN evaluation tool to direct future work. Some preliminary results related to this work were presented at the NANOG-43 meeting[35].

The rest of the paper is organized as follows. In Section 2, we review the existing algorithms for BGP anomaly detection and robustness, and then we will describe the registry-based, history-based, and hybrid (i.e., combined registry and history data driven) algorithms used in this study. In Section 3, we present the results of a quality analysis for various regional registries (ARIN, RIPE etc.). Also, in Section 3, we present empirical analysis and comparisons of the registry-based, history-based, and hybrid algorithms. In Section 4, the conclusions are stated and the directions for future work are discussed.

2. Algorithms for BGP Anomaly Detection and Robustness

In this section, we first review the existing algorithms for BGP anomaly detection and robustness. Then we will present the algorithms used in this study which are enhancements and variants of the existing algorithms.

Nemecis system [14][15] uses a registry-based method driven by declarative data from the RIRs and IRRs. Data in the Routing Policy Specification Language (RPSL) [37][38] format as well as the Shared Whois Project (SWIP) format are considered. For a given {prefix, origin Autonomous System (AS)} pair from an update, it checks for existence of prefix registration (i.e, inetnum in RPSL-based RIRs or NetHandle in SWIP-based RIRs), AS registration (aut-num in RPSL and ASHandle in SWIP), and route objects in IRRs or RADB. It further checks for consistency between these declared objects by matching various attributes such as organization, maintainer, email handle, etc. The algorithm can in principle generate alerts if these checks fail, i.e., when there is a lack of full or partial consistency check. Full consistency check, for example, is when the route object is consistent with both the inetnum as well as the aut-num. Partial consistency check, for example, is when the route object is consistent with only the inetnum or only the aut-num.

The Prefix Hijack Alert System (PHAS) [16][17] provides alert messages if the update stream is detected to contain any of the following scenarios: (1) An origin AS in an update message that is new relative to the set of previously observed set of origin ASes for the same prefix, (2) A new more specific subprefix of an existing announced route is observed, (3) The last-hop AS (i.e., the AS that is one hop away from the origin AS) in an update message is new relative to the set of previously observed last-hop ASes for the same prefix.

The Pretty Good BGP (PGBGP) method [18][19] is based purely on observed BGP trace data. In the earlier version of the algorithm [18], observed {prefix, origin AS} pairs based on update history and Routing Information Base (RIB) entries over the last h days (h = 10 days) are recorded. The anomaly detector also eliminates old routes (older than 10 days) if they are no longer active. A new update is considered suspicious if the {prefix, origin AS} pair is not in the history record; the update is propagated with lower local-pref value. A subprefix announcement (with the same origin AS that its covering prefix has had or with a different origin AS) is always considered suspicious and quarantined. The quarantine lasts for a period of s hours (e.g., s = 24 hours); if the subprefix is not withdrawn during that time, then the update is propagated.

The earlier version PGBGP described above had some deficiencies in the following scenarios. When a prefix is

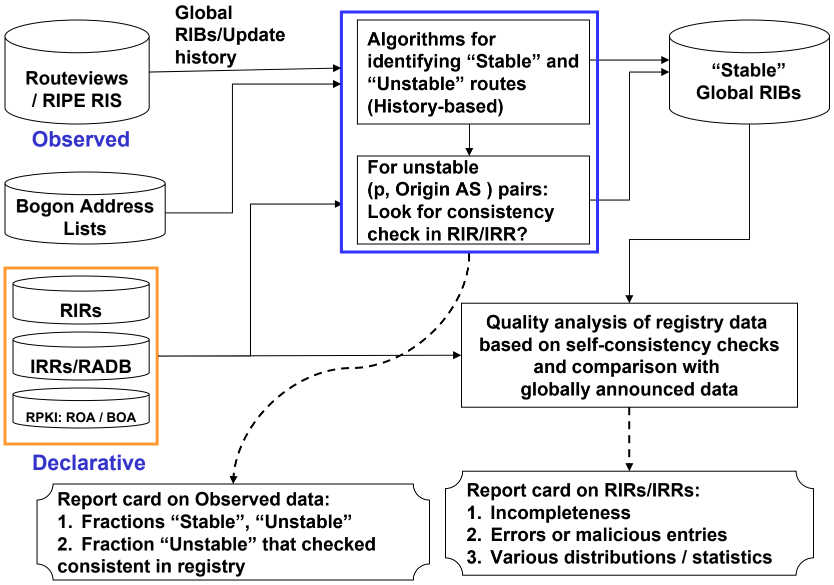
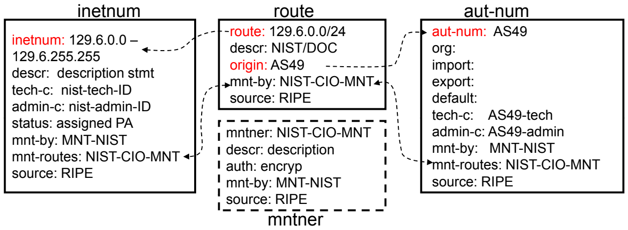

Routeviews / RIPE RIS
Observed
Global RIBs/Update history
Bogon Address Lists
RIRs
IRRs/RADB
RPKI: ROA / BOA
Declarative
Algorithms for identifying "Stable" and "Unstable" routes (History-based)
For unstable (p, Origin AS) pairs: Look for consistency check in RIR/IRR?
"Stable" Global RIBs
Quality analysis of registry data based on self-consistency checks and comparison with globally announced data
Report card on Observed data:
1. Fractions "Stable", "Unstable"
2. Fraction "Unstable" that checked consistent in registry
Report card on RIRs/IRRs:
1. Incompleteness
2. Errors or malicious entries
3. Various distributions / statistics

Figure 2. Integrated algorithm for evaluating quality and analyzing anomalies in registered and historical BGP data.



```
inetnum: 129.6.0.0 –
129.6.255.255
descr:  description stmt
tech-c:  nist-tech-ID
admin-c: nist-admin-ID
status: assigned PA
mnt-by: MNT-NIST
mnt-routes: NIST-CIO-MNT
source: RIPE

route: 129.6.0.0/24
descr: NIST/DOC
origin: AS49
mnt-by: NIST-CIO-MNT
source: RIPE

mntner: NIST-CIO-MNT
descr: description
auth: encryp
mnt-by: MNT-NIST
source: RIPE

aut-num:  AS49
org:
import:
export:
default:
tech-c:    AS49-tech
admin-c: AS49-admin
mnt-by:   MNT-NIST
mnt-routes: NIST-CIO-MNT
source: RIPE
```

Figure 3. Illustration of method for checking consistency of a registered route with corresponding inetnum and aut-num.

hijacked, the prefix owner's first obvious countermeasure aimed at restoring their connectivity would be to announce their own address space split into more specific subprefixes. In the PGBGP described thus far, the updates related to this countermeasure would also have been considered suspicious, since they gave new subprefixes of a known prefix. This weakness was acknowledged and the following modification was incorporated in a new version [19]: "PGBGP would not interfere if an AS announces sub-prefixes of its own prefixes in order to gain traffic back during a prefix hijack." However, this new version of PGBGP still has certain shortcomings. The short-span historical view (last ten days) has the following negative implications: (1) PGBGP will typically unnecessarily lower local-pref on path announcements due to multi-homing related AS origin change; (2) If a malicious user observes a prefix withdrawal by genuine origin AS and announces the prefix at that time, the malicious path propagates with a lower local-pref value and will be used (i.e., effectively a False Negative occurs); (3) If a prefix owner sometimes legitimately announces subprefixes in conjunction with multi-homing related AS origin change, PGBGP will quarantine the announcements.

The algorithms described above are either solely registry-data driven or solely trace-data (or history) driven. There can be benefits to an integrated approach which is based on the combined use of both types of data. Such an integrated approach is illustrated in Figure 2. In this approach, all available history data sources such as routeviews and RIPE-RIS are utilized, and simultaneously all available declarative data sources, namely, RIRs, IRRs, RADB, Resource-certificate Public Key Infrastructure (RPKI)[41], bogon lists are also utilized. As shown in Figure 2, the global RIBs and update history data can be analyzed to identify the historically stable and unstable routes or equivalently {prefix, origin AS} pairs. The definitions of stable and unstable routes will be discussed in Subsection 2.2. The registry data can be first analyzed in itself to determine the consistency of the declared routes. The definitions of consistency will be discussed in Subsection 2.1. Then, comprehensive quality checks can be performed by cross-checking the results of the analyses of history and declarative data stated above. This integrated approach is expected to provide more reliable results on registry quality as well as improve the performance of algorithms that seek to differenti-

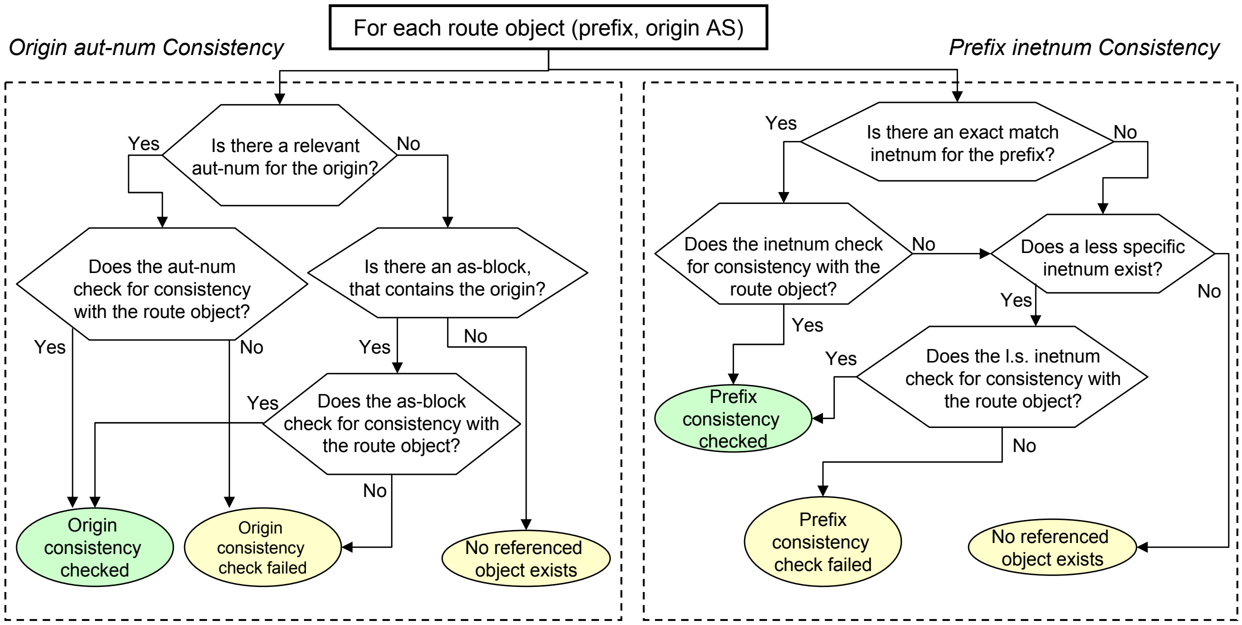

For each route object (prefix, origin AS)
Origin aut-num Consistency
Prefix inetnum Consistency
Is there a relevant aut-num for the origin?
Yes
No
Does the aut-num check for consistency with the route object?
Is there an as-block, that contains the origin?
Yes
No
Yes
No
Does the as-block check for consistency with the route object?
Yes
No
Origin consistency checked
Origin consistency check failed
No referenced object exists
Is there an exact match inetnum for the prefix?
Yes
No
Does the inetnum check for consistency with the route object?
No
Does a less specific inetnum exist?
Yes
No
Yes
Does the l.s. inetnum check for consistency with the route object?
Yes
No
Prefix consistency checked
Prefix consistency check failed
No referenced object exists

Figure 4. Detailed algorithm for checking registry consistency of prefix-origin pairs in registered route objects.

ate the good vs. suspicious (i.e., anomalies) in announced routes.

In Subsections 2.1 and 2.2 below, we will discuss further details of enhanced registry-based and history-based approaches. These approaches can be run individually or can be integrated into an enhanced hybrid algorithm as described in Subsection 2.3. The proposed algorithms were implemented and tested on the TERRAIN evaluation tool. In Section 3, we will present numerical results based on our analysis of the registry data and robustness algorithms.

2.1. Enhanced Registry-Based Algorithm

Although we describe the algorithms here in terms of RIPE RPSL object names such as route, inetnum, aut-num and their attributes such as *mntner*, *mnt-by*, etc., the description is also applicable to ARIN's SWIP format by appropriate substitution of corresponding object names (NetHandle, ASHandle) and attribute names (Org, TechHandle, etc.).

In the registry-based approaches (e.g., Nemecis), observed {prefix, origin AS} pairs are matched with registered route objects as well as the corresponding prefix and autonomous system registrations. Alerts are generated when the match fails. For this type of algorithm to perform satisfactorily, the registered routes must have self-consistency. In this section, we first describe the method used for checking self-consistency of routes within the registries. Here we take the cue from RIPE's Routing Policy System Security (RPSS) [39] as was done also in the Nemecis [14][15] tool. Figure 3 partially illustrates how the attribute values are matched to determine consistency between the different registered objects (route, intenum, aut-num) corresponding to a {prefix, origin AS} pair that appears in a registered route object. We look for the *mntner* attribute value match between route, inetnum, and aut-num. As shown in Figure 3, the actual attributes which store this information in different objects may vary, e.g., *mnt-by*, *mnt-lower*, *mnt-routes*, etc. In case the matching based on *mntner* attribute and its variants fails, then we look for a match based on values of other attributes such as organization ID (*org*), technical contact (*tech-c*), administrative contact (*admin-c*), etc.

In the above paragraph, we described the basic steps of consistency checks for registered route objects or equivalently the {prefix, origin AS} pairs in them. However, there are additional considerations that go into a careful methodology for consistency checking. These considerations are illustrated in the detailed route-object consistency algorithm of Figure 4. Here if a prefix in route object does not have a corresponding inetnum registered then we look for an inetnum for a less specific prefix. We still consider the route object as prefix-consistent if the checks succeed with this less specific inetnum. Sometimes the origin AS in a route object may not have a registered aut-num but there may be an as-block that contains the origin AS. In that case the con-

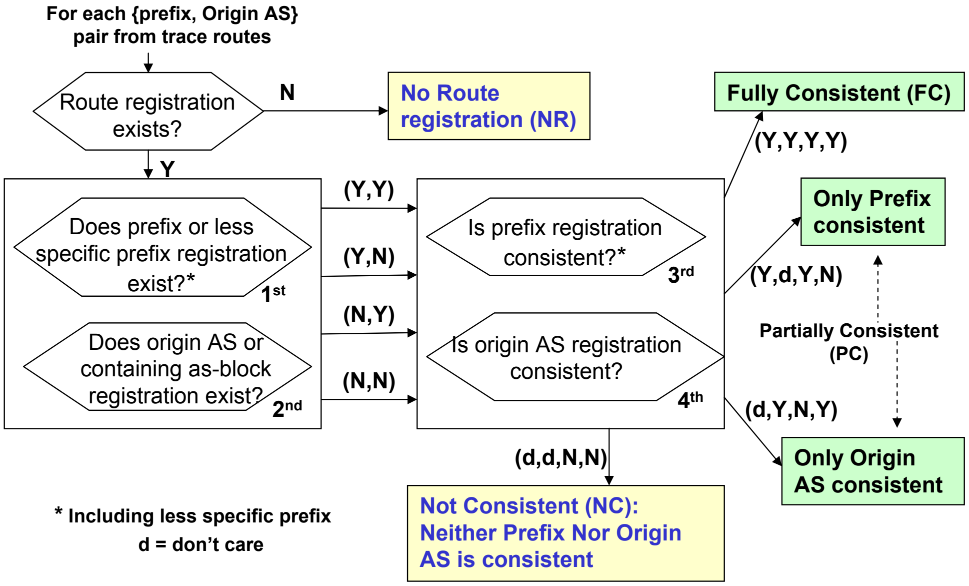

For each {prefix, Origin AS} pair from trace routes
Route registration exists?
N
No Route registration (NR)
Y
Does prefix or less specific prefix registration exist?*
1st
Does origin AS or containing as-block registration exist?
2nd
(Y,Y)
(Y,N)
(N,Y)
(N,N)
Is prefix registration consistent?*
3rd
Is origin AS registration consistent?
4th
Fully Consistent (FC)
(Y,Y,Y,Y)
Only Prefix consistent
(Y,d,Y,N)
Partially Consistent (PC)
(d,Y,N,Y)
Only Origin AS consistent
(d,d,N,N)
Not Consistent (NC): Neither Prefix Nor Origin AS is consistent
* Including less specific prefix
d = don't care

Figure 5. Registry-based algorithm: matching announced (prefix, origin AS) pairs with registered routes, prefixes and origin ASes.

sistency check is performed between the route and the as-block.

In the above, we discussed how we classify the consistency of route objects registered in IRRs with respect to other registry objects such as inetnums and aut-nums registered in RIRs. We will now discuss how historically observed (trace) routes are scored for their consistency checks with respect to their corresponding registry objects such as route objects, inetnums and aut-nums in the IRRs/RIRs. The algorithm for doing this is illustrated in Figure 5. This algorithm makes use of the results of the registry self-consistency checks already performed by the previous algorithm illustrated in Figures 3, 4. As shown in Figure 5, first we check if a each unique {prefix, origin AS} pair observed in a trace route has a corresponding route registration. If the prefix itself does not have a corresponding route registered, we look for a less specific prefix that may have a route registered with the same origin AS. The end result of the algorithm (see Figure 5) is that each unique {prefix, origin AS} pair observed in a trace route is scored in four ways: (1) Fully Consistent (FC) when both the prefix's inetnum (or that of its less specific prefix) as well as the aut-num of the origin AS (or that of a containing as-block) are consistent with the associated route object; (2) Partially Consistent (PC) if either the prefix registration or the origin AS registration is consistent but not both; (3) Not Consistent (NC) if neither is consistent; and (4) Not Registered (NR) if no corresponding route registration exists.

2.2. Enhanced History-Based Algorithm

Here we describe an enhanced history-based algorithm that incorporates several enhancements over the PGBGP algorithm that we reviewed earlier. For brevity, let us refer to {prefix, origin AS} pair as (p, OAS) pair. In the PGBGP algorithm, it is required that an observed (p, OAS) pair be seen in the RIB in the last ten days for it to be part of a *trusted* set of routes. If not seen in the RIB in the last ten days, a (p, OAS) pair is considered stale and discarded from the *trusted* list. This idea has a drawback as follows. In the case of multi-homing related switchovers from a primary route to a secondary route (i.e., switching from (p, OAS_1) to (p, OAS_2)), the switchover may last for a few days and the event may not recur for months. In this case, when the event does occur again after months, the secondary OAS for that prefix will be considered suspicious and the prefix will be assigned lower local pref value. Further, if the prefix owner happens to split the prefix between two alternate origin ASes after having used only one origin for months, then the subprefix that is originated from the secondary origin AS will be quarantined. Also, if a prefix owner splits a prefix across two origin ASes and varies the subprefix sizes from time to time for re-optimizing the traffic engineering, then again the subprefixes will be commonly quarantined in PGBGP.

To avoid the above stated problems, we suggest an enhanced history-based algorithm. In this algorithm, we consider trace data over a longer retention period (e.g., months).

Trace data is the entire update data at a collector or BGP speaker over a history period initialized with first few (typically three) days of RIB data. Referring to Figure 6, the elapsed time, $t_e(p, OAS)$, is the duration of persistence of a (p, OAS) pair in the RIB. It is defined as the time difference between the first announcement of that (p, OAS) pair from any peer and the last implicit/explicit withdrawal of that prefix by any peer resulting in the removal of that (p, OAS) pair from the RIB. In other words, during $t_e(p, OAS)$ the (p, OAS) pair remained as a part of a valid route in the RIB. We define a (p, OAS) pair as *historically stable* if $t_e(p, OAS)$ is greater than or equal to a certain time duration, say 48 hours. The rationale for 48 hours is that typically operators detect and eliminate malicious or mis-configured routes approximately within a day[19]. If $t_e(p, OAS)$ is less than that time duration (i.e., 48 hours), then the (p, OAS) pair is labeled as unstable. Stability sticks, which means if a (p, OAS) pair is stable once during the history period, then it is remains marked as stable at the end of the history (observation) period. That is how the algorithm works to create a template of stable and unstable observed (p, OAS) pairs in trace data. This template can be used to judge new updates immediately following the history period as stable (i.e, good) or as unstable (i.e., suspicious). Also, the algorithm is further enhanced by considering an otherwise unstable (p, OAS) pair to be stable if a route with the same origin AS and a less specific prefix (or superprefix) that covers the prefix p was found to be stable during the same observation period.

2.3. Enhanced Hybrid Algorithm

We feel that the registry and historical data can be complementary to each other in enhancing the performance of anomaly detection algorithms. So we propose here hybrid algorithms which make combined use of both type of routing data. A simple hybrid algorithm can be simply a combination of the registry-based algorithm described above in Subsection 2.1 and the key elements of the PGBGP algorithm. However, we feel that it would be more effective to integrate the registry-based algorithm described in Subsection 2.1 with the enhanced history-based algorithm described above in Subsection 2.2. We call this type of hybrid algorithm an enhanced hybrid algorithm. One additional consideration is to use the combined consistency results of the registry-based algorithm on two dates; one date immediately before and another date towards the end of or immediately after or the history period as illustrated in Figure 7.

To the extent the registry data is of good quality and reliable, the hybrid algorithm can accordingly provide improved performance over history-based algorithm alone. Here improved performance is in terms of more reliable detection of announcements with suspicious (p, OAS).

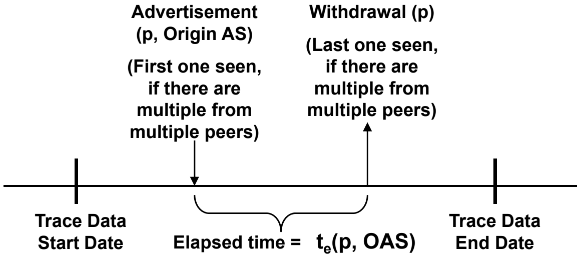

Advertisement
(p, Origin AS)
(First one seen, if there are multiple from multiple peers)
Withdrawal (p)
(Last one seen, if there are multiple from multiple peers)
Trace Data Start Date
Elapsed time = t_e(p, OAS)
Trace Data End Date

Figure 6. History-based algorithm for checking prefix-origin pairs observed in trace data.

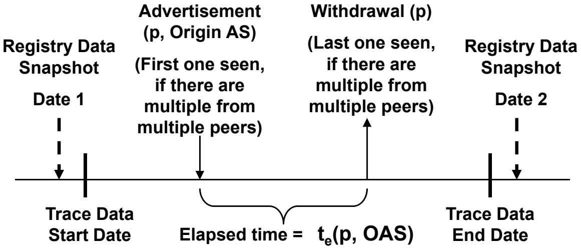

Registry Data Snapshot
Date 1
Advertisement
(p, Origin AS)
(First one seen, if there are multiple from multiple peers)
Withdrawal (p)
(Last one seen, if there are multiple from multiple peers)
Registry Data Snapshot
Date 2
Trace Data Start Date
Elapsed time = t_e(p, OAS)
Trace Data End Date

Figure 7. Hybrid algorithm for checking prefix-origin pairs observed in trace data.

One clear example is in the case of multi-homing related switchovers from a primary route (primary OAS) to a backup route (backup OAS). Such switchover may sometimes last only for hours and therefore the history-based part of the hybrid algorithm would score the prefix and backup OAS pair as unstable. But if there exists a registered route for the prefix and the backup OAS that checks well for consistency (please refer to Subsection 2.1), then the registry-based part of the hybrid algorithm will characterize the update with the backup OAS as good. Thus the hybrid algorithm can benefit from the potentially complementary nature of the registry and historical data. There would be a reduction in false positives for the hybrid algorithm as compared to that for the history-based or the registry-based algorithm individually.

2.4. Qualitative Comparison of the Algorithms

In Table 1, we list the key questions that should be asked or checks performed in anomaly detection algorithms, and show which of these checks are performed by several of the algorithms we have reviewed or proposed in this paper. These questions or checks were all discussed above while explaining the operations of the different algorithms. Based on the variety and quality of checks performed, we would expect the performance to vary significantly from one type of algorithm to another. In Table 2, several desirable fea-

Table 1. Comparison of checks included in various approaches

	Checks/Questions	Which checks are included in each approach?			
		Registry-based approach (e.g., Nemecis)	Trace-data-based (e.g. PGBGP)	Enhanced Trace-data-based	Enhanced Hybrid
Q1.	Is prefix registered (same or less specific)?	X			X
Q2.	Is there a route registered (with same or less specific prefix and origin AS)?	X			X
Q3.	Is announced (p, origin AS) fully consistent with corresponding registry objects in RIR/IRR?	X			X
Q4.	Is announced (p, origin AS) partially consistent with corresponding registry objects in RIR/IRR?	X			X
Q5.	Was (p, origin AS) seen in RIB in the last h (= 10) days? (Also, if it was suspicious, did it remain in RIB beyond the suspicious period of s (= 24) hours?)		X		
Q6.	Would a less specific prefix with the same origin AS pass the test in Q5?		X		
Q7.	Was prefix previously announced by the same origin AS and remained stably (48 hrs or more) in the RIB over the observation period (d months)?			X	X
Q8.	Would a less specific prefix with the same origin AS pass the test in Q7?			X	X

tures of anomaly detection algorithms are listed, and several of the algorithms are qualitatively compared based on which of these features they provide. The comments noted for different algorithms in terms of their features in Table 2 are directly related to their detailed descriptions provided in Subsections 2.1 through 2.4. The enhanced hybrid algorithm performs most checks and is expected to perform best in terms of detection and mitigation of anomalies. Results based on actual data for quantitatively comparing the various algorithms are presented in the next section.

3. Analysis of Data and Algorithms Using the TERRAIN Database

In this section, we will first discuss results based on an analysis of the quality of registry data, and then present and discuss several results pertaining to the analysis and comparisons of anomaly detection algorithms described in Section 2.

We have made extensive use of the TERRAIN database for the analysis presented in this section. The database based on MySQL is architected to facilitate access to both update and registry data in a highly structured manner to facilitate efficient search and analysis. From this database, we used the update data of RIPE-RIS rrc00 collector [42] over a six month period (January through June 2007). This represents close to 600 million updates. From the TERRAIN database, we also used complete registry data (RIRs, IRR, RADB) for two different dates (December 12, 2006 and June 18 2007).

3.1. Quality of Registry Data

Table 3 shows the numbers of different types registry objects and their growth in a six month period from December 2006 to June 2007. These objects are routes, inetnums, and aut-nums in the RPSL format from RIPE, Asia Pacific Network Information Center (APNIC), Latin America and the Caribbean NIC (LACNIC), African NIC (AFRINIC), ARIN and RADB, and additionally NetHandles and ASHandles in SWIP format from ARIN. In Table 3, APNIC includes Taiwan NIC (TWNIC), Japan IRR (JPIRR), Japan NIC (JPNIC), and of course APNIC. LACNIC is RIR only; it has no IRR. As mentioned earlier, by RADB we refer to the collective routing information that is registered or mirrored[40]. Further, RADB numbers shown in the table include all mirrored data but they exclude routes in ARIN, APNIC and JPIRR in order to avoid duplication. Also, it should be noted that routes can be registered in any IRR regardless of where their address spaces are allocated.

We performed registry consistency checks using the taxonomy and methodology described in Subsection 2.1. In Table 4 and Figuer 8, the results of the analysis for the June 2007 registry data are shown. We see that amongst the regional registries RIPE has the highest number of route ob-

Table 2. Comparison of algorithmic features of various approaches

	Algorithmic Features	Registry-based (e.g., Nemecis)	Trace-data-based (e.g. PGBGP)	Enhanced Trace-data-based	Enhanced Hybrid
1	Utilization of self-consistent registry objects	Yes	No	No	Yes
2	Utilization of update history	No	Yes	Yes	Yes
3	Utilization of historical RIB entries	No	Yes	Yes	Yes
4	Pass a subprefix announcement if a less specific prefix with same origin AS could be passed	Yes	Yes	Yes	Yes
5	False Positives: Alert raised when genuine prefix owner announces multi-homing related AS origin change	Moderate probability	High probability	Moderate probability	Low probability
6	Alert raised when attacker announces a prefix after sensing it has just been withdrawn	Yes	No (Path propagates with lower pref)	Yes	Yes
7	Pass a subprefix announcement in conjunction with multi-homing related AS origin change	Moderate probability	Low probability	Moderate probability	High probability

jects registered and it has an excellent score in terms of percentage (97%) that are fully or partially consistent. Both APNIC and ARIN have significantly fewer routes registered as compared to RIPE, but APNIC has route registration quality that is as good as that of RIPE. ARIN has a very low number of routes registered and about 80% of them are fully or partially consistent, while about 20% are not consistent. However, there is ongoing improvement taking place in the ARIN NetHandles due to a recent modification requiring inclusion of origin AS information[22]. This will be discussed in more detail in Subsection 3.2. The RADB data shown in Table 4 and Figuer 8 excludes the mirrored data of the regional registries, and hence reflects only the route registrations of Internet Service Providers (ISPs) in their IRRs. Although the numbers of routes registered in RADB (ISPs' and other organizations' IRRs) is very large, the consistency scores for these routes are poor. Only 27% of the RADB routes are fully or partially consistent while the other 73% are not consistent. The results of this registry consistency analysis play a direct role in influencing the outcome of analysis of the registry-based and hybrid algorithms as we will see in Subsection 3.2.

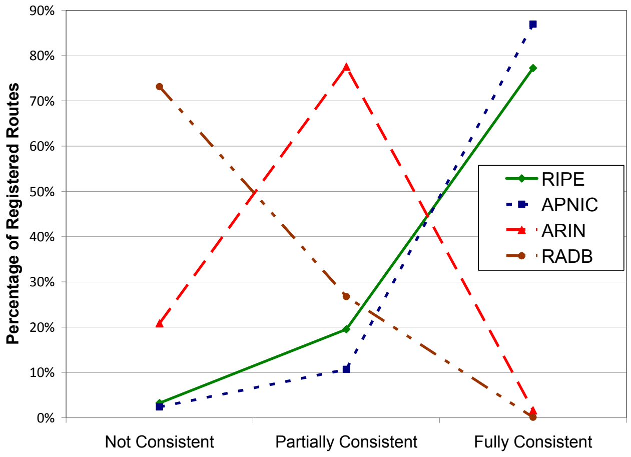


Figure 8. Comparison of registry consistency scores of registered routes for RIPE, APNIC and ARIN.

3.2. Analysis and Comparison of the Algorithms

As mentioned before, we consider trace data gathered over approximately six months (January 2007 through June 2007) from RIPE-RIS rrc00 collector. It included RIB data for the first three days of the period and update data for the whole period. We listed all the unique observed (p, OAS) pairs and analyzed them using registry-based, enhanced history-based and enhanced hybrid algorithms. We checked how many of these observed (p, OAS) pairs are stable and how many unstable using the algorithm described in Subsection 2.2. Then for the stable as well as unstable ones, we checked them for existence of registrations and consistency in the registries, and classified them accordingly to not registered (NR), not consistent (NC), partially consistent (PC), and fully consistent (FC) (see Subsection 2.1 for the definitions and algorithm). The results of these analyses are presented in Table 5 and Figure 9. The results are shown separately for RIPE, APNIC and ARIN registries based on determination of which of these registries the prefixes in the observed (p, OAS) pairs belong to. The Global entries in the table refer to routes seen globally regardless of the registries in which the prefixes belong. It is ideally desired that each observed (p, OAS) pair should be historically stable as well as consistent (FC or PC) in the registries. We note that observed (p, OAS) pairs that have prefixes belonging to RIPE

Table 3. Counts of various registered objects in RIRs, IRRs, and RADB

	12-12-2006	06-18-2007	12-12-2006	06-18-2007	12-12-2006	06-18-2007
RIR	Route		RPSL: inetnum (SWIP: NetHandle)		RPSL: aut-num (SWIP: ASHandle)	
ARIN	5749	7,330	318 (1,510,448)	338 (1,618,197)	697 (17,540)	758 (18,050)
RIPENCC	65,604	71,569	1,782,279	2,044,536	12,909	14,106
APNIC	13,075	23,616	698,546	822,891	4,229	4,559
AFRINIC	0	0	12,101	13,948	252	342
LACNIC	0	0	43,840	45,346	1,147	1,219
RADB	355,276	345,129	1	1	4,523	3,785
Total:	439,704	447,644	2,537,085 (1,510,448)	2,927,060 (1,618,197)	23,757 (17,540)	24,769 (18,050)

Table 4. Scores for registry consistency checks for route objects in RIPE, APNIC, ARIN, and RADB.

	RIPE		APNIC		ARIN		RADB	
Fully Consistent (FC)	55267	77%	20531	87%	119	2%	320	0%
Partiallty Consistent (PC)	13983	20%	2523	11%	5683	78%	92390	27%
Not Consistent (NC)	2319	3%	562	2%	1528	21%	252395	73%
Total # Routes Registered	71569		23616		7330		345105	

region perform best. APNIC and ARIN have a very high number of stable but unregistered (p, OAS) pairs. ARIN is undergoing a significant improvement since May 2007 due to inclusion of origin AS in the NetHandles [22]. The ARIN NetHandles with origin AS information in them can be thought of as equivalent to route registrations in RPSL. With that the scores for ARIN in Figure 9 have a potential to improve significantly. However, during the history data period of our analysis, ARIN did not have any significant number of NetHandles that contained origin AS information.

In Figures 10 and 11, for the sake of brevity the names Registry, History, and Hybrid refer to registry-based algorithm, enhanced history-based algorithm, and enhanced hybrid algorithm, respectively. In the analysis here, a (p, OAS) pair is considered *good* if it has full or partial consistency (FC or PC) in the registry, or if it is judged as stable in trace data. historically stable. Figure 10 compares the performance of the three algorithms. For the RIPE region, 84% of observed (p, OAS) pairs are declared *good* by the Registry algorithm while the other 16% are declared as *suspicious*. But still the list of *suspicious* (p, OAS) pairs would consist of many false positives due to imperfect quality of the RIPE registry. This problem is much worse when the Registry algorithm is applied to observed (p, OAS) pairs belonging to APNIC or ARIN regions. For the APNIC or ARIN regions, only small percentages of observed (p, OAS) pairs are declared *good* by the Registry algorithm while the majority (over 80%) are declared as *suspicious*. This is due to the poor registry data quality of APNIC and ARIN in terms of lack of sufficient route registrations and their consistency. As also shown in Figure 10, the History algorithm declares over 95% of the observed (p, OAS) pairs as stable and less than 5% as unstable for any of the registry regions. The nature of BGP routing updates is such that up to about 5% appear to covey announcements of instable (p, OAS) pairs. Some fraction of these can still be false positives. The Hybrid algorithm has the potential to further reduce

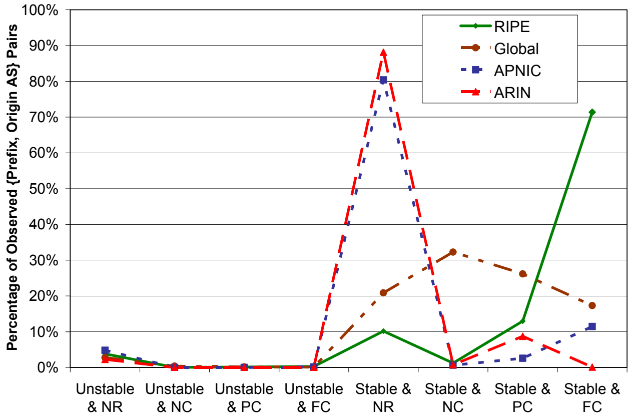


Figure 9. Comparison of performance of RIPE, APNIC and ARIN with detailed scores from historical stability analysis and registry checks.

Table 5. Details of stability and registry consistency scores for RIPE, APNIC, ARIN, and RADB

	RIPE		APNIC		ARIN		Global	
	Number	Percentage	Number	Percentage	Number	Percentage	Number	Percentage
Unstable & NR	2903	3.80%	4385	4.82%	3537	2.27%	9526	2.78%
Unstable & NC	31	0.04%	1	0.00%	5	0.00%	1295	0.38%
Unstable & PC	90	0.12%	10	0.01%	15	0.01%	509	0.15%
Unstable & FC	222	0.29%	109	0.12%	0	0.00%	300	0.09%
Stable & NR	7785	10.19%	73188	80.40%	137325	88.13%	71447	20.85%
Stable & NC	931	1.22%	563	0.62%	1192	0.76%	110529	32.26%
Stable & PC	9903	12.96%	2337	2.57%	13608	8.73%	89706	26.18%
Stable & FC	54541	71.38%	10434	11.46%	135	0.09%	59326	17.31%
Total Observed	76406		91027		155817		342638	

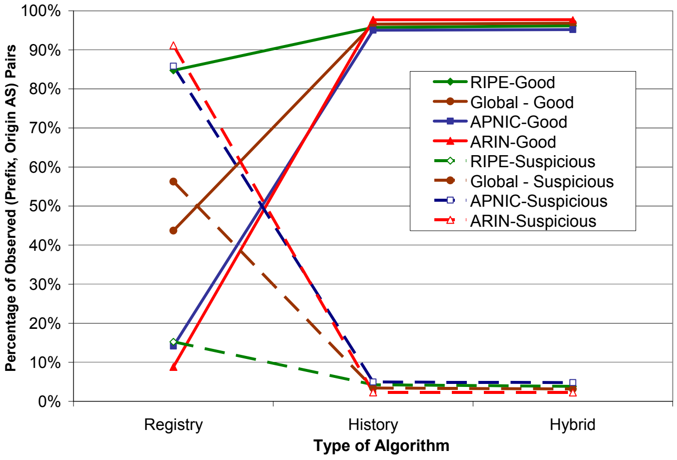


Figure 10. Comparison of performance of Enhanced Registry-based, Enhanced History-based, and Enhanced Hybrid algorithms for RIPE, APNIC, ARIN and Global prefixes.

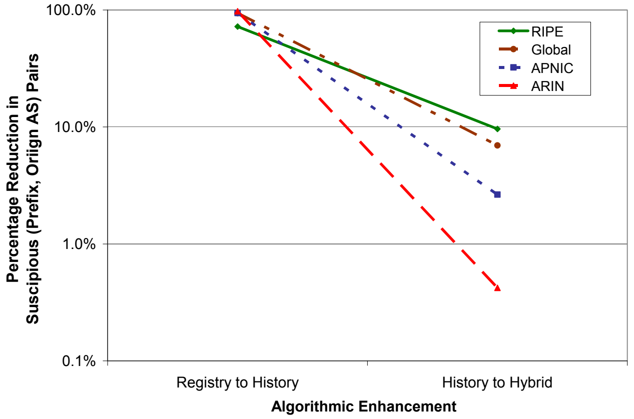


Figure 11. Progressive improvement in reduction of percentage of suspicious (p, OAS) pairs: Enhanced Registry to Enhanced History and Enhanced History to Enhanced Hybrid.

false positives because some unstable (p, OAS) pairs may check consistent in the registry and hence need not be considered *suspicious*. This reduction is not quite discernible in Figure 10. But in Figure 11, the percentage reductions in *suspicious* (p, OAS) pairs are shown illustrating the benefits of algorithmic enhancements from Registry to History and again from History to Hybrid. For example, for observed (p, OAS) pairs with their prefixes belonging to the RIPE region, the History algorithm reduces the number of *suspicious* (p, OAS) pairs by about 72% as compared to the same for the Registry algorithm. Further, the Hybrid algorithm reduces the number of *suspicious* (p, OAS) pairs by about 10% as compared to the same for the History algorithm. This 10% reduction is attributable to the superior registry data quality of RIPE RIR/IRR. However, it must be noted that this percentage would vary somewhat depending on the specific trace data being considered. Figure 11 also shows that the same algorithmic enhancement from History to Hybrid yields relatively much smaller benefit when the observed (p, OAS) pairs belong to APNIC or ARIN regions. It is clear that History algorithm performs significantly superior to Registry algorithm for any registry region. But there are benefits derivable by combining them into a Hybrid algorithm because they may act in a complementary way. Figure 11 also conveys the message that when the registry data is of better quality, the hybrid algorithm provides noticeable improvement over the history-based algorithm. This insight is further reinforced below via additional empirical observations in the context of multi-homed prefixes.

Another view of the results in Figure 10 can be presented using heatmaps [43] that are plotted using a Hilbert curve, and this representation is shown in Figure 12. Here the origin AS information is suppressed by suitably combining the *good* and *suspicious* scores of {prefix, origin AS}

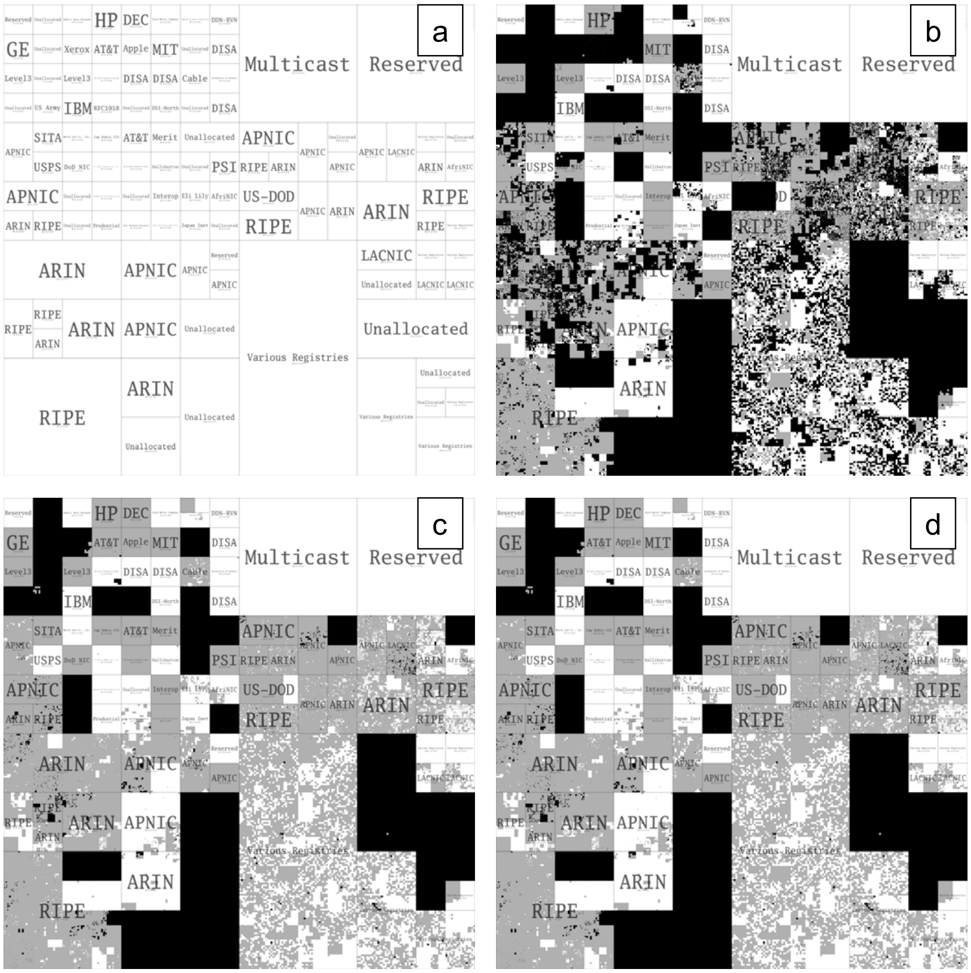


Figure 12. Comparison of performance of algorithms using heatmaps: (a) Hilbert curve of Internet prefix allocations; (b) Registry-based algorithm; (c) Enhanced history-based algorithm; (d) Enhanced hybrid algorithm.

pairs across multiple origin ASes when a prefix has been observed to have multiple origin ASes. If the {prefix, origin AS} is *suspicious* for a prefix for anyone of its possibly multiple origin ASes, then the prefix is scored as *suspicious*. A prefix is scored as *good* only if all its associated {prefix, origin AS} pairs have been scored as *good*. Figure 12(a) provides the basic map of Internet prefix allocations corresponding to various organizations and RIRs. To generate the heatmaps in Figures 12(b), (c) and (d), the prefix is colored gray if it is scored as *good* and black if scored as *suspicious*. Gray color is used in the case of the registry-based algorithm in Figure 12(b) to represent full consistency (FC) or partial consistency (PC). White color represents prefixes not unobserved in the trace data. Figures 12(b), (c) and (d) show heatmaps for registry-based algorithm, enhanced history-based algorithm, and enhanced hybrid algorithm, respectively. The registry-based algorithm has the most black and relatively less gray, while the enhanced history-based and enhanced hybrid algorithms have much more gray commensurate with the scores shown in Figure 10. Some of the large address blocks that are colored back in these figures are unallocated prefixes that have been accidentally announced and withdrawn quickly. It is also possible to see in Figure 12(b) that the RIPE allocated address blocks generally have much more gray (due to better consistency checks in the registry) than the ARIN and APNIC allocated address blocks. The hybrid algorithm (Figure 12(d)) does have slightly more gray than the history-based algorithm

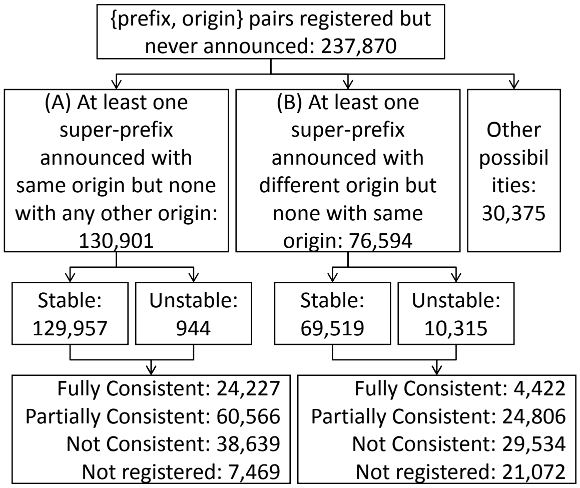

{prefix, origin} pairs registered but never announced: 237,870
(A) At least one super-prefix announced with same origin but none with any other origin: 130,901
(B) At least one super-prefix announced with different origin but none with same origin: 76,594
Other possibilities: 30,375
Stable: 129,957
Unstable: 944
Stable: 69,519
Unstable: 10,315
Fully Consistent: 24,227
Partially Consistent: 60,566
Not Consistent: 38,639
Not registered: 7,469
Fully Consistent: 4,422
Partially Consistent: 24,806
Not Consistent: 29,534
Not registered: 21,072

Figure 13. Analysis of large number of registered {prefix, origin AS} pairs that were never observed in RIBs or updates.

Table 6. Data on stability and registry consistency of multi-homed prefixes

(p, OAS1)	(p, OAS2)	# Prefixes
FC + Stable	FC/PC + Unstable	23
PC + Stable	FC/PC + Unstable	41
NC + Stable	FC/PC + Unstable	104
NR + Stable	FC/PC + Unstable	0
Total		168

(Figure 12(c)); the differences are hard to see but can be seen with magnification of the figure (see [44] to access an electronic copy of this paper).

The data in Table 6 answers a significant question about behavior of algorithms with regard to multi-homed prefixes. Assume that the primary route (p, OAS1) is observed to be stable and the secondary route (p, OAS2) is unstable (due to failovers which last less than 48 hours). Then the following questions arises: In how many of these cases, is the secondary path's (p, OAS) pair characterized as Fully Consistent (FC) or Partially Consistent (PC) in the registries so that the hybrid algorithm successfully avoids generating an alert? From the announced routes, we gathered all the prefixes that are dual-homed. Then we checked how many instances are there when a prefix is announced stably from one of its origin ASes and unstably from its other origin AS. We further investigate this data to count the instances of four of the possible combinations that are of interest to us as shown in Table 6. As the table indicates, there are 168 such instances with dual-homed prefixes. These instances facilitate a reduction in False Positives with the hybrid algorithms. Note that PGBGP would falsely depref the route and PHAS would falsely generate alerts in these situations, while the hybrid algorithms would successfully treat the updates in these situations as legitimate and thus avoid generating false alerts.

3.3. Analysis of Registered {Prefix, Origin AS} Pairs That Were Never Observed

One question that needs serious exploring is why in spite of large numbers of registered routes (e.g., 447,644 per June 2007 data shown in Table 3) there are very high percentages of stable routes that are not registered (see APNIC and ARIN plots in Figure 9). This in turn motivates us to examine how many (p, OAS) pairs are registered but not announced and if the number is large then why so? As shown in Figure 13, there is indeed a large number of (p, OAS) pairs that are registered but never observed in the trace (i.e., RIB and update) data. The analysis in Figure 13 shows that the majority of the prefixes in the unobserved (p, OAS) pairs in question are individually part of a superprefix that was announced. The superprefixes are observed in trace data with the original origin AS found in the (p, OAS) registration in about 2/3 of the cases, and with a different origin AS in the remaining approximately 1/3 of the cases. Figure 13 also shows the analysis of the superprefixes so identified in terms of their stability and registration status. The bulk of them are indeed stable, and a good fraction of them are also registered (with original ASes found in the prefix registration or with different ASes). At the same time, it should also be noted that a substantial fraction of the superprefixes in consideration are not registered or not consistent. What the various numbers in Figure 13 tell us is that if we examine the subprefixes (or deaggregates) of a prefix that is stably announced but not registered, then we would find that many of the subprefixes (or deaggregates) are indeed registered. But it would be rather hard to seek complete coverage in the registry of a prefix by trying to break it into subprefixes. Aside from that when an ISP aggregates certain prefixes into a superprefix, it should be expected to register the superprefix and a route for the same. That would naturally help enhance the quality of the registry data and also improve the performance of anomaly detection algorithms.

4. Conclusions and Future Work

We presented an overview and comparisons of BGP robustness and anomaly detection algorithms. We reviewed known algorithms and alert tools such as the Nemecis tool, PGBGP, PHAS, etc. We proposed a new algorithm that combines the use of both registry and trace data, and also makes some key improvements over existing algorithms.

We also presented an evaluation methodology and comparisons of existing algorithms and a proposed new algorithm. We have described an evaluation platform called TERRAIN on which these algorithms are being tested and empirically compared based on real and/or synthetically incorporated anomalies in BGP updates. We have presented a variety of results providing interesting insights into the comparative utility and performance of the various BGP robustness algorithms. We feel that the proposed enhanced history-only algorithm potentially offers good improvements over the PGBGP algorithm. Further, when the registry data quality is good as in the case of RIPE, the proposed enhanced hybrid algorithm provides about 10% improvement over the enhanced history-only algorithm in terms reduction in suspicious {prefix, origin AS} pairs. Our objective is to share these early insights and invite feedback from the community to refine the TERRAIN evaluation tool and direct future work.

We are presently extending this work to include consideration of multiple trace-data collectors. Further testing for robustness of the algorithms will be performed with additional real and synthetic trace data. The goal is to generate reliable empirical results for benchmarking the quality of registry data and the utility of BGP robustness algorithms. We hope to help the industry to understand implications of proposals emerging from various ongoing R&D projects in the area of BGP anomaly/attack detection and mitigation.

5. Acknowledgments

This research was supported by the Department of Homeland Security under the Secure Protocols for the Routing Infrastructure (SPRI) program and the NIST Information Technology Laboratory Computer and Network Security Program.

The authors would like to thank Josh Karlin and several anonymous reviewers for very helpful comments and suggestions.

References

[1] M. Brown, T. Underwood, and E. Zmijewski, “The Day the YouTube Died: What happened and what we might do about it,” MENOG 3, Kuwait April 2008. http://www.renesys.com/tech/presentations/pdf/menog3-youtube.pdf

[2] “YouTube Hijacking: A RIPE NCC RIS case study, RIPE News and Announcements (2008). http://www.ripe.net/news/study-youtube-hijacking.html

[3] A. Pilosov and T. Kapela, “Stealing The Internet: An Internet-Scale Man In The Middle Attack,” NANOG-44, Los Angeles, October 12-15, 2008. http://www.nanog.org/meetings/nanog44/presentations/Tuesday/Kapela_steal_internet_N44.pdf

[4] “Flaw Could Cripple Entire Net,” Associated Press, April 20, 2004. http://wired-vig.wired.com/news/technology/0,1282,63143,00.html

[5] “NISCC Vulnerability Advisory 236929: Vulnerability Issues in TCP,” April 20, 2004.

[6] “CERT Advisory CA-2001-09: Statistical Weaknesses in TCP/IP Initial Sequence Numbers,” http://www.cert.org/advisories/CA-2001-09.html, Original date May 2001, last revised February 2005.

[7] NISCC (UK Govt.) Best Practices Guidelines: Border Gateway Protocol (see note 5 on pg. 8), April 2004.

[8] O. Nordstrom and C. Dovrolis, “Beware of BGP attacks,” SIGCOMM Computer Communications Review (2004), Vol. 34 (2), pp. 1-8.

[9] K. Butler, T. Farley, P. McDaniel, and J. Rexford, “A Survey of BGP Security”, Technical Report TD-5UGJ33, ATT Labs - Research, April 2005. http://www.patrickmcdaniel.org/pubs/td-5ugj33.pdf

[10] Kim, J., S.Y. Ko, D.M. Nicol, X.A. Dimitropoulos, G.F. Riley, “A BGP attack against traffic engineering,” Proceedings of the 2004 Winter Simulation Conference, 2004.

[11] S.M. Bellovin and E.R Gansner, “Using Link Cuts to Attack Internet Routing”, AT&T Labs Research Technical Report, http://www.research.att.com/ smb/papers/reroute.pdf

[12] F. Gont, “ICMP Attacks against TCP,” IETF Internet Draft, draft-gont-tcpm-icmp-attacks-03.txt, December 2004.

[13] S. Convery, D. Cook, and M. Franz, “An Attack Tree for the Border Gateway Protocol”, IETF ID, February 2004. http://ietfreport.isoc.org/ids/draft-ietf-rpsec-bgpattack-00.txt,

[14] G. Siganos and M. Faloutsos, “A Blueprint for Improving the Robustness of Internet Routing,” 2005. http://www.cs.ucr.edu/

[15] G. Siganos and M. Faloutsos, “Analyzing BGP policies: methodology and tool,” IEEE Infocom, 2004.

[16] M. Lad, D. Massey, Y. Wu, B. Zhang and L. Zhang, “PHAS: A prefix hijack alert system,” North American Network Operators Group Meeting (NANOG-38), October, 2006. http://www.nanog.org/mtg-0610/presenter-pdfs/massey.pdf

[17] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang and L. Zhang, “PHAS: A prefix hijack alert system,” in Proceedings of 15th USENIX Security Symposium (USENIX Security 2006). http://www.cs.ucla.edu/ mohit/cameraReady/ladSecurity06.pdf

[18] J. Karlin, S. Forrest, and J. Rexford, “Pretty Good BGP and the Internet Alert Registry,” NANOG 37, June 5th 2006. http://www.nanog.org/mtg-0606/pdf/josh-karlin.pdf

[19] J. Karlin, S. Forrest, and J. Rexford, “Pretty Good BGP: Improving BGP by Cautiously Adopting Routes,” The 14th IEEE International Conference on Network Protocols, November 2006. http://www.cs.unm.edu/ treport/tr/06-06/pgbgp3.pdf

[20] R. Steenbergen, “What’s Wrong with IRR,” NANOG-44, Los Angeles, October 12-15, 2008. http://www.nanog.org/meetings/nanog44/presentations/ Tuesday/RAS_irrdata_N44.pdf

[21] M. Karir, L. Blunk, T. White, Kevin Chan, and Pat Pannuto, “Improving the Accuracy of Routing Registry Data,” NANOG-44, Los Angeles, October 12-15, 2008. http://www.nanog.org/meetings/nanog44/presentations/ Tuesday/Karir_lightning.pdf

[22] M. Kosters, “ARIN Engineering Report,” ARIN-22, Los Angeles, October 15-17, 2008. http://www.arin.net/meetings/minutes/ARIN_XXII/PDF/ friday/engineering_report.pdf

[23] 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,” Proceedings of the Second ACM SIGCOMM Workshop on Internet Measurement Workshop, 2002, Marseille, France, pp. 183-195.

[24] S. Kent, C. Lynn, and K. Seo, “Secure Border Gateway Protocol (S-BGP),” IEEEJSAC Special Issue on Network Security, April 2000.

[25] J. Ng, “Extensions to BGP to Support Secure Origin BGP (soBGP),” IETF ID draft-ng-sobgp-bgp-extensions-02.txt, April 2004.

[26] L. Subramanian et al., “Listen and whisper: Security mechanisms for BGP,” In First Symposium on Networked Systems Design and Implementation (NSDI’04), 2004.

[27] A. Heffernan, “Protection of BGP Sessions via the TCP MD5 Signature Option,” IETF RFC 2385, August 1998.

[28] J. Cowie, A. Ogielski, B. Premore, and Y. Yuan, “Global Routing Instabilities during Code Red II and Nimda Worm Propagation,” http://www.renesys.com/projects/bgp_instability, September 2001.

[29] T. Griffin, “BGP Impact of SQL Worm, 1-25-2003”, January 2003. http://www.research.att.com/ griffin/bgp-monitor/sql-worm.html.

[30] Z.M. Mao, R. Govindan, G. Varghese, and R.H. Katz, “Route Flap Damping Exacerbates Internet Routing Convergence,” Proceedings of ACM SIGCOMM, Pittsburg, PA, August 2002, pp. 221-233.

[31] G. Goth, “Fixing BGP Might Be Difficult–Or Not So Tough,” IEEE Internet Computing, vol. 07, no. 3, pp. 7-9, May/June 2003.

[32] K. Sriram, D. Montgomery, O. Borchert, O. Kim, and R. Kuhn, “Study of BGP Peering Session Attacks and Their Impacts on Routing Performance,” IEEE Journal on Selected Areas in Communications: Special issue on High-Speed Network Security, Vol. 24, No. 10, October 2006, pp. 1901-1915.

[33] K. Sriram, D. Montgomery, O. Borchert, O. Kim, and R. Kuhn, “Border Gateway Protocol (BGP): Investigation of Vulnerabilities and Simulation Studies of Attack Impacts,” NIST IIP project presentation slides, October 2006. http://www.antd.nist.gov/ ksriram/BGP_Security_Analysis_NIST_Study.pdf

[34] D.R. Kuhn, K. Sriram, and D. Montgomery, “Border Gateway Protocol Security,” NIST Special Publication 800-54 (Guidance Document for the Telecom Industry and US Government agencies), July 2007. http://csrc.nist.gov/publications/nistpubs/800-54/SP800-54.pdf

[35] K. Sriram, D. Montgomery, O. Borchert, O. Kim, and P. Gleichmann, “Evaluation of BGP Anomaly Detection and Robustness Algorithms,” Presented at the 43rd North American Network Operators Group (NANOG-43) Meeting, Brooklyn, New York, June 1-4, 2008. http://www.nanog.org/mtg-0806/presentations/tuesday/Sriram_algorithms_N43.pdf

[36] “Team Cymru: The Bogon Reference,” http://www.team-cymru.org/Services/Bogons/

[37] C. Alaettinoglu et al., “Routing Policy Specification Language (RPSL),” IETF RFC 2622, June 1999.

[38] L. Blunk et al., “Routing Policy Specification Language next generation (RPSLng),” IETF RFC 4012, March 2005.

[39] C. Villamizar et al., “Routing Policy System Security,” IETF RFC 2725. http://www.ietf.org/rfc/rfc2725.txt

[40] Merit Network’s Routing Assets Database, http://www.radb.net/about.html

[41] G. Huston and G. Michaelson, “Validation of Route Origination in BGP using the Resource Certificate PKI,” October 6, 2008, Work in progress, IETF SIDR Working Group, draft-ietf-sidr-roa-validation-01.txt.

[42] Reseaux Internet Protocol Europeens - Routing Information Service (RIPE-RIS) Project. http://www.ripe.net/ris/

[43] IPv4 Heatmaps Software, The Measurement Factory, http://maps.measurement-factory.com/software/ipv4-heatmap.1.html

[44] Online version of this paper with some additional details and color graphics, http://www.antd.nist.gov/ ksriram/NIST_BGP_Robustness.pdf