Observation and Analysis of BGP Behavior under Stress

Lan Wang[1], Xiaoliang Zhao[2], Dan Pei[1], Randy Bush[3], Daniel Massey[2], Allison Mankin[2], S. Felix Wu[4], Lixia Zhang[1]

Abstract— **Despite BGP's critical importance as the de-facto Internet inter-domain routing protocol, there is little understanding of how BGP actually performs under stressful conditions when dependable routing is most needed. In this paper, we examine BGP's behavior during one stressful period, the Code Red/Nimda attack on September 18, 2001. The attack was correlated with a 30-fold increase in the BGP update messages at a monitoring point which peers with a number of Internet service providers. Our examination of BGP's behavior during the event concludes that BGP exhibited no significant abnormality, and that over 40% of the observed updates can be attributed to the monitoring artifact in current BGP measurement settings. Our analysis, however, does reveal several weak points in both the protocol and its implementation, such as BGP's sensitivity to the transport session reliability, its inability to avoid the global propagation of small local changes, and its certain implementation features whose otherwise benign effects only get amplified under stressful conditions. We also identify areas for improvement in the current network measurement and monitoring effort.**

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No DABT63-00-C-1027. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the DARPA.

[1]Lan Wang, Dan Pei and Lixia Zhang are with UCLA. E-mail: {lanw, peidan, lixia}@cs.ucla.edu [2]Xiaoliang Zhao, Daniel Massey and Allison Mankin are with USC/Information Science Institute. E-mail: {xzhao, masseyd, mankin}@isi.edu [3]Randy Bush is with Internet Initiative Japan. E-mail:randy@psg.com [4]S Felix Wu is with UC Davis. E-mail: wu@cs.ucdavis.edu

I. Introduction

The Internet infrastructure relies on BGP[1] to provide essential routing information. Despite its critical importance, relatively little is known about how well, or poorly, BGP actually performs under stressful conditions. Infrastructure stress events can damage critical links and have a direct impact on BGP behavior. Moreover, recent work shows that even non-infrastructure events may also have an impact on BGP. In particular, [2] observed that the Code-Red/Nimda worm attack was closely correlated in time with a large spike in the number of BGP routing updates from multiple ISPs received at a monitoring point. Such impact on BGP is surprising since the worm was directed against web servers while BGP is running on routers which provide reachability to all networks.

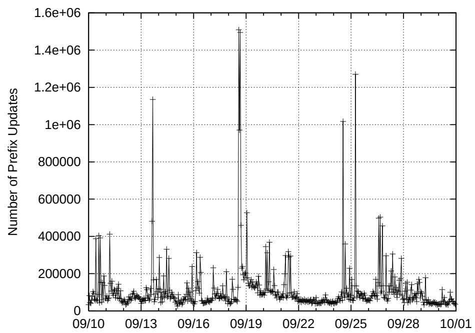


Fig. 1. Number of BGP Prefix Updates in Hourly Bins from Sept. 10, 2001 to Sept. 30, 2001

According to the SANS Institute, the scanning activity of the Nimda worm dramatically increased at approximately 1pm on September 18, 2001 [1], and abated in the following hours[3]. Figure 1 shows the number of BGP prefix updates received by the RRC00 monitoring point [4] at RIPE NCC.

[1]GMT time is used throughout this paper.

One can see a large spike of BGP updates received by the monitoring point around the worm attack time. More specifically, roughly 1.5 million BGP updates were received between 2pm and 3pm, which represents a 30 fold increase over the number of updates received between 12pm and 1pm on the same day. Although large spikes of BGP updates are observed on a few other days as well, the one on Sept. 18 rose much higher and lasted longer. Such behavior was taken as an indication that the worm attack caused global routing instability [2].

To gain a deeper understanding of BGP behavior under stress and to explain the observation during the Nimda worm attack in particular, we analyzed the BGP routing update data collected by RIPE NCC from Sept. 10, 2001 to Sept. 30, 2001 (the same data set is also used in [2]). We found that over 40% of the observed BGP updates during the attack were due to BGP session resets at the monitoring point (see Section III-B). These session resets resulted from the specific settings of the BGP monitoring infrastructure, but are not necessarily indicative of routing changes in the operational network.

We also found that a substantial fraction of the remaining updates did not lead to AS path changes (Section III-C and Section III-D). Rather, these updates are the results of either implementation choices or exposure of internal changes within an ISP; the stressful condition caused by the Code-Red/Nimda attack significantly amplified the impact of these otherwise benign protocol or implementation details.

Finally, we examined the BGP updates that did convey new AS path information during the attack period and found that the majority of these actual routing changes happened to only a small number of highly unstable networks (Section III-E). The BGP protocol allows these local changes to propagate globally and BGP's slow convergence [5] further exacerbated the problem. Slow convergence could be reduced or eliminated using techniques such as those proposed in [8]).

The paper is organized as follows. Section II describes our methodology and system for classifying BGP updates into meaningful groupings. Section III provides a more detailed look at the volume of updates received during our study period and presents our results. Section IV presents the related work and Section V summarizes our main findings.

A. *BGP Operations and Terminology*

BGP is a *path vector routing protocol* for inter-domain routing [1]. Its route computation is similar to Distance Vector protocols, but it prevents routing loops by attaching the complete path information (i.e. AS path) to every route advertisement.

To exchange routing information, two BGP routers first establish a peering session that operates on top of a TCP connection. The routers then exchange their full routing tables in a series of BGP messages. After the initial route exchanges, each router sends only incremental updates for new or modified routes. When a router discovers that it can no longer reach a destination (i.e. an IP address prefix) that it has announced before, it sends a message to its peer to withdraw the route. Note that each BGP router can have multiple peers.

Before we present our methodology and findings, we would like to clarify the distinction between *a BGP message* and *a BGP prefix update*. A *BGP message* refers to the message used by BGP peers to announce/withdraw a route or to manage the BGP session. In the former case, it can carry one BGP route and *multiple* address prefixes that use the same route in order to minimize transmission overhead. To analyze the route changes for *individual* prefixes, we studied the sequence obtained by unpacking the BGP messages. These unpacked announcements or withdrawals are referred to as "BGP prefix update" (or "BGP update" for brevity).

II. DATA AND METHODOLOGY

We analyzed the BGP messages collected at RIPE NCC [4] from Sept. 10, 2001 to Sept. 30, 2001. RIPE NCC operates eight monitoring points (RRC00-RRC07) and each monitoring point peers with multiple operational BGP routers at various ISPs. Our analysis is based on data from 12 peers at the RRC00 monitoring point.

RRC00 is located at the RIPE NCC office in Amsterdam, Netherlands. It peers with 15 BGP routers through *multi-hop* eBGP sessions. We analyzed data from twelve sessions that were active during the observation period. Table I summarizes the locations of these twelve BGP peers. Three of the monitored ISPs are tier-1 ISPs in the US and the others are ISPs in Europe and Asia. The ISP names have been anonymized.

TABLE I
RRC00'S PEERING ASES THAT WE EXAMINED

Location	ASes that RRC00's peers belong to
US	ISP1, ISP2, ISP3
Netherlands	ISP4, ISP5, ISP6, ISP7
Switzerland	ISP8, ISP9
Britain	ISP3
Germany	ISP10
Japan	ISP11

A. *Update Classification Procedure*

To better understand the BGP behavior during the attack, we classify all the BGP updates into classes and then infer what may be the leading causes of each class. We are most interested in those that are indicative of actual route changes. Furthermore, we observe whether a behavior (e.g. an increase in a certain class) is specific to a subset of the peers, as such behavior is usually associated with specific BGP implementation features or the ISPs' network characteristics.

To classify the BGP updates, we note the timing of a BGP update and its relationship to the previous update. Useful clues include whether the update follows immediately after a session reset, whether the update follows a route withdrawal, and whether the update contains new route information or a duplicate of the previous information. Based on these clues, we categorize BGP updates into the classes shown in Figure 2.

At the top of the class hierarchy are two major classes: *announcements* and *withdrawals*. An announcement contains the sender's BGP route to an address prefix, while a withdrawal indicates that the sender wants to remove a previously announced route.

An announcement can be further classified into three sub-classes. If the sender announces a route to a previously unreachable address, this is a *new*

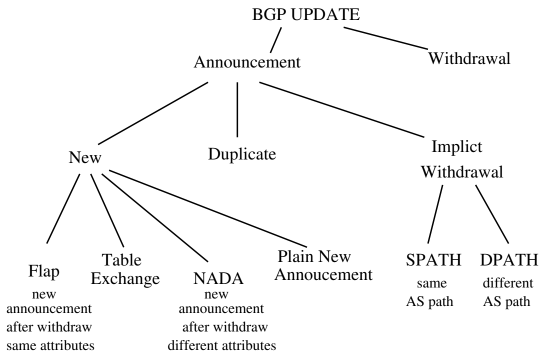


Fig. 2. BGP Update Class Hierarchy

announcement. If the sender announces a route to a currently reachable address and the new route is identical to the current route, this is a *duplicate announcement*. Otherwise, the sender is replacing the current route with a new route and this is an *implicit withdrawal*.

A *new announcement* can be further classified into four sub-classes. If the new announcement is sent during an initial BGP table exchange, it is labeled as "Table Exchange"; identifying such *Table Exchange* requires special care, as explained in the following section. If it follows a withdrawal and simply re-announces the withdrawn route, it is labeled as "Flap". If it follows a withdrawal and the new route differs from the previously withdrawn route, it is labeled as "NADA". If it fits none of the above three profiles, we call it a "Plain New Announcement".

Finally, an *Implicit Withdrawal* can be further classified into two sub-classes depending on new AS path information. If the new route contains the same AS path as the current route, it is labeled as "SPATH".[2] If the implicit withdrawal replace the current AS path with a new AS path, it is labeled as "DPATH".

For example, suppose we want to classify the BGP updates received from ISP1. First, we learn the routes that had already been announced by ISP1 at the beginning of our observation period. We can find this information by obtaining the monitoring point's routing table on Sept. 9, 2001 from RIPE NCC's archive. We then apply each BGP

[2] An SPATH implicit withdrawal is distinct from a duplicate announcement since the implicit withdrawal changes some attribute other than the AS path.

update (collected from Sept. 10 to 30) from ISP1 to this initial routing table. When a BGP update is received, we find out which class it belongs to by comparing the new route with the existing route (in addition to updating the routing table).

B. Identifying Table Exchange Updates

When a BGP session goes down, the routing table associated with that session is flushed. When the session is re-established, the entire table is re-advertised and the corresponding announcements are counted as *Table Exchange* updates. Although the log messages in the data set provide the timing information for the peering session state changes, one cannot take a naive approach of classifying all updates received immediately after a session reset as table exchange updates. Because the routing table contains about 100,000 routing entries which may take several minutes to be re-advertised, during this table exchange period other BGP changes may occur and can generate updates that are interleaved with the table exchange updates. Furthermore, there is no clear ending point for the table exchange since the routing table may have added or lost prefixes during the time the session was down.

We take a 2-step approach to address this problem. First, based on the observation that most routing table transfers were finished within 10 minutes, we set a relatively long time of 25 minutes as table exchange period. After a new BGP session comes up, our table is initially empty and we count any update that installs a *new routing entry* as a *Table Exchange* update during the next 25 minutes. Any further updates for this entry are not counted as table exchange updates, even if they occur during the 25-minute table exchange interval. Provided the routing table exchange is finished within 25 minutes, our approach will accurately count table exchange updates with one exception. If a prefix was not in the routing table before the session reset but is announced during the 25 minutes of the table exchange period, a perhaps rare event, it cannot be distinguished from a table exchange update. Thus our count of table exchange updates is equal to the actual number of table exchange updates plus some delta of prefixes that appeared for the first time within 25 minutes of the session reset. If the entire routing table actually took longer than 25 minutes to transfer, also a rare event[3], we would underestimate the total number of table exchange updates.

III. Results

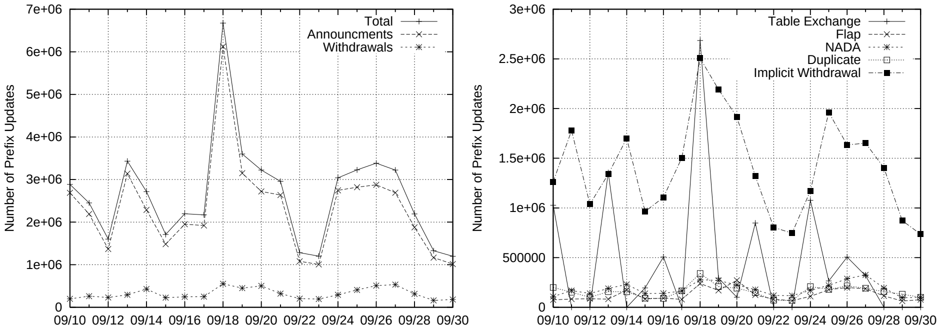
In this section, we first present an overview of the daily BGP update volume from September 10 to September 30, 2001. We then identify the major contributors to the dramatic increase of the number of BGP updates during the worm attack. In this process we identify the weakness in both the BGP protocol design and implementation and the weakness in the current monitoring settings that give rise to the observed high routing message volume.

A. Daily Variations

Figure 3(a) shows that the total number of BGP updates varied significantly during the 21-day observation period; the highest value, 6.67 million on Sept. 18 (the worm attack day) is more than 5 times the lowest value, 1.20 million on Sept. 30. The figure also shows clearly that the announcements are the main contributor to the sharp increase observed by RRC00 on Sept. 18. Over the 21-day period the announcements constitute 87.3% of the total number of updates per day on average, and they were even more prevalent on Sept. 18 (91.7%).

Further breakdown of the announcements (see Figure 3(b)) shows that, except on Sept. 18, implicit withdrawals are the largest component in the BGP announcements, accounting for 40.9% to 81.2% of the total daily announcements. The second largest component is the BGP table exchanges, although this class varies greatly from day to day. There were no BGP table exchanges on 9 of the 21 days, but on the other 12 days they contributed to 3.7% - 43.9% of the total announcements. The combination of these two components caused the other 3 update spikes in Figure 1 that occurred on Sept. 14, 24, and 26, respectively. The other three classes, Flap, NADA and Duplicate updates (see definitions in Figure 2), are relatively minor contributors to the total count.

[3] We did observe that, in a few cases, the number of prefixes exchanged within 25 minutes of the session reset was much lower than 100K.



(a) Announcements and Withdrawals

(b) Breakdown of Announcements

Fig. 3. Breakdown of Prefix Updates Received by RRC00

Sept. 18 saw both the largest number of BGP table exchange updates – 2.7 million, and the highest number of implicit withdrawals – 2.5 million (see Figure 3(b)). The other classes remained insignificant, although they also exhibited increases. More specifically, the composition of the BGP updates on Sept. 18 is as follows:

- BGP Table Exchanges: 40.2%;
- Implicit Withdrawals: 37.6%;
- New Announcements (excluding BGP Table Exchanges): 8.9%;
- Duplicate Announcements: 5%;
- Withdrawals: 8.3%.

In the following sections, we examine the causes of each BGP update class to see whether the increase in that class reflects inter-domain *routing instability*.

B. Session Resets

Approximately 2.7 million prefix updates (40.2% of the total count) received by the monitoring point on Sept. 18 were due to BGP table exchange. If we eliminate this category of BGP updates from the 21-day observation period, the total number of updates received on Sept. 18 is only 1.94 times of the average over the 21-day period (excluding Sept. 18). Below we examine the causes of the large number of BGP table exchanges.

We first found that, during the worm attack on Sept. 18, 2001, the monitoring point experienced a large number of BGP session resets. Figure 4 shows when the sessions went down on Sept. 18; all of them occurred between 2pm and 5pm. The X-axis is time and the Y-axis corresponds to the different peers of the monitoring point. Each mark indicates that the corresponding peer's session went down at that time. The figure shows that most of the twelve BGP sessions restarted multiple times during the worm attack period; one of the peering session restarted 6 times within the 3-hour period. Because each session reset means transferring the entire routing table (about 100K entries), even a small number of session resets can result in a large amount of BGP updates.

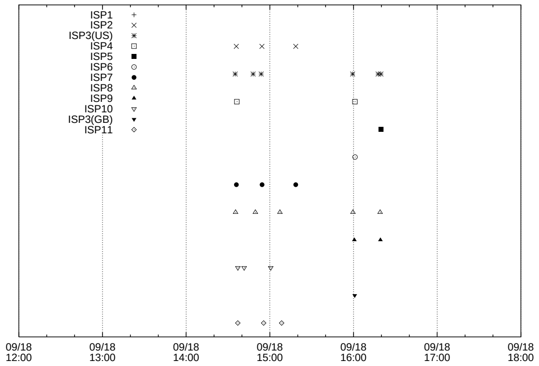


Fig. 4. Session Resets on Sept. 18, 2001

BGP session resets can be caused by physical connectivity failures (e.g. link failure or router crash), transient connectivity problems due to congestion, or even manual reboots. We observed, however, among the thirty session resets observed at the monitoring point on Sept. 18, twenty-seven sessions were re-established within 1 minute and the other three recovered within 7 minutes. Such fast session recoveries suggest that these resets were not caused by hardware failures that require human intervention, but most likely were due to transient link congestion or routing problems.

Furthermore, we observed that some of the session resets were highly synchronized, for example, 7 session resets occurred between 14:35 and 14:37 and 6 occurred between 15:59 and 16:03. Using tools provided by the RIPE RIS project to trace the routes from the monitoring point to the peers, we found that the routes to the twelve BGP peers all share the same first two hops, and the second-hop router is one of the BGP routers peering with RRC00. This router had two session resets on Sept. 18, one at 14:36:20 and the other at 16:00:58, an indication of serious congestion or routing problems within the first two hops of the monitoring point at those times.

The scan activity of the Nimda worm might have contributed to the congestion that led to the session resets. However, we would like to remind the reader that RRC00's monitoring peering sessions are multiple-hop eBGP sessions, while peering sessions between operational ISPs are usually set up across high speed LAN or switch interconnect or via direct links. The multi-hop BGP sessions that are commonly used in BGP monitoring projects are likely to suffer from overload at various points along the paths, therefore their *frequent* session resets may not be representative of what happened between the operational ISPs.

To verify the above conjecture, we analyzed the peering sessions at another monitoring point RRC04. RRC04 is located at the CERN Internet Exchange in Geneva and it has direct BGP sessions with 10 peers located at the *same* exchange point. Our data shows that none of these single-hop peering sessions had a reset on Sept. 18, although some session resets were observed on Sept. 21. Of particular interest is one BGP router that peers with both RRC00 (via multi-hop) and RRC04 (via direct link). On Sept. 18, this router had 5 session resets with RRC00 but none with RRC04. Discussions with people from a few other ISPs also suggested that those ISPs did not experience the frequent session reset behavior seen at the monitoring point on Sept. 18.

Overall, the 40.2% of updates due to session resets indicates that the *monitoring process* was affected by the worm attack. This is an interesting result, but it is not indicative of actual BGP instability in the Internet. It suggests that, in order to correctly infer the general behavior of the BGP infrastructure, data collected from multi-hop BGP sessions (such as RIPE and Oregon Route Views) should be sanitized to remove the side-effects caused by the multi-hop peering.

At the same time, we would also like to raise the question of whether the BGP protocol should be designed to work *only* under certain network conditions. An iBGP session may cross several network hops which could be subject to congestion as a multi-hop eBGP session. Moreover, a direct eBGP session could also break due to severe congestion or other types of failures (as you will see in Section III-D). BGP's sensitivity to the transport session reliability raises both a protocol design and an implementation issue that deserve further attention in order to improve the resilience and stability of inter-domain routing.

C. Duplicate Announcements

Duplicate announcements make up 4 to 10% of all the BGP updates over the observation period. Although all the peers exhibit this behavior to varying degrees, it is most serious for ISP1, whose duplicate announcements account for, on average, 31% of its total daily prefix updates.

The duplicate announcements may be due to a particular implementation issue [6]. Such an implementation will send out a BGP update message whenever there is a change to its BGP routes, regardless of whether the change is associated with a *non-transitive* attribute. Since all the non-transitive attributes will be stripped off before a route is announced, the receiver will see a duplicate announcement if all the transitive attributes remain the same. Discussion with ISP1 confirmed the existence of this problem in their router.

It has been argued in the past that tolerating such duplicate announcements leads to a simpler implementation. Our data show that the little saving in the implementation increases the overall sys-

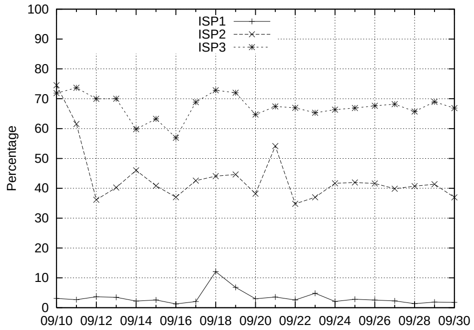


Fig. 5. SPATH Implicit Withdrawal Percentage (US)

tem overhead, as all peers have to process additional messages unnecessarily. Worse yet, because changes in non-transitive attributes, such as nexthop and local preference, are usually associated with changes in local network conditions, routers with this implementation feature tend to send more duplicate updates when the their internal network is under stress. Sept. 18 had 0.34 million duplicate updates, the highest over the 21-day period and more than twice the average, suggesting that seemingly harmless small overhead in protocol operation may get amplified under stressful conditions, and that protocol implementation decisions must take into account the potential global impact.

D. SPATH Implicit Withdrawal

The real piece of the puzzle is the implicit withdrawals, the second largest component in BGP update counts during the attack period. Further examination shows that on average 22% of all the implicit withdrawals did not contain new AS paths, but involved changes to other BGP attributes such as MED. To distinguish these types of updates, we call the ones without AS path changes *SPATH* and the rest *DPATH*. We examine the SPATH implicit withdrawals in this section and the DPATH implicit withdrawals in the next section.

First of all, we noticed that two US peers had an unexpectedly large portion of implicit withdrawals in this category. Figure 5 compares the prevalence of SPATH implicit withdrawals in the three US peers. The X-axis is time. The Y-axis is the percentage of a peer's implicit withdrawals that are SPATH on each day, and we can see that this statistic is about 40% for ISP2 and 70% for ISP3, much higher than the 22% value averaged across all the peers.

Because SPATH implicit withdrawals do not contain new AS paths, they do not reflect topology changes at the AS level, but more likely reflect local changes within an ISP. For example, an ISP may have a policy to dynamically compute the value of the MED attribute (or the community attribute) based on its internal network conditions, as these attributes influence how its neighbors may direct their traffic toward its network. Therefore, the large number of SPATH implicit withdrawals sent by ISP2 ad ISP3 may be explained by the fact that these two tier-1 ISPs have richer network topologies and more sophisticated policies than the other tier-1 or regional ISPs in the monitored set.

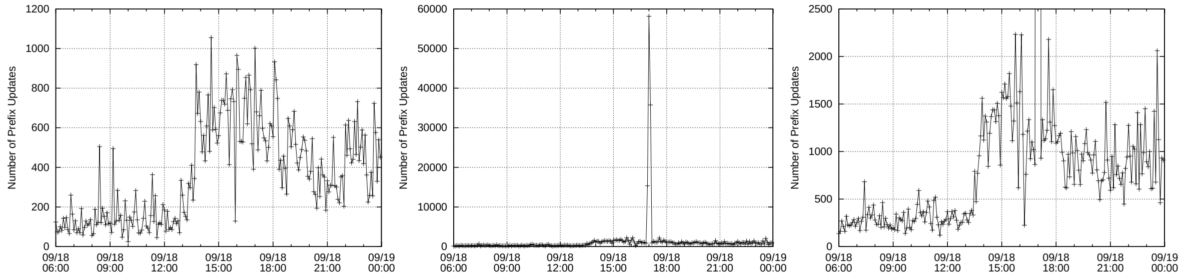
Similar to the duplicate updates, the SPATH implicit withdrawals do not represent inter-domain routing change. Moreover, increased internal network instability can lead to increased number of SPATH implicit withdrawals. This effect is evident from Figure 5, which shows that ISP1's curve rises from a normal level of 2% to more than 10% on Sept. 18 when its internal network was under stress.

E. DPATH Implicit Withdrawal

The DPATH implicit withdrawals represent 76.9% of all the implicit withdrawals received on Sept. 18, and they indicate real inter-domain routing changes. In this section we examine the causes in 3 categories: inferring the causes of observed large spikes in the updates, identifying the specific prefixes that had the most updates, and observing the impact of BGP slow convergence during the attack period.

E.1 Inferring the Cause of Spikes in DPATH Implicit Withdrawal

We plotted the updates due to DPATH implicit withdrawals on Sept. 18 for every peer in 5-minute bins. To illustrate the cases we typically see, we selected two peers (ISP1 and ISP5) and showed their updates in Figure 6. ISP1's curve started to climb around 1pm on Sept. 18 and reached its peak at 2:35pm (1,055 DPATH implicit withdrawals in 5 minutes). The curve remained relatively high from 2pm to 5pm and then started to decline. It slowly



(a) ISP1 (b) ISP5 (c) ISP5 (spike removed)

Fig. 6. DPATH Implicit Withdrawals

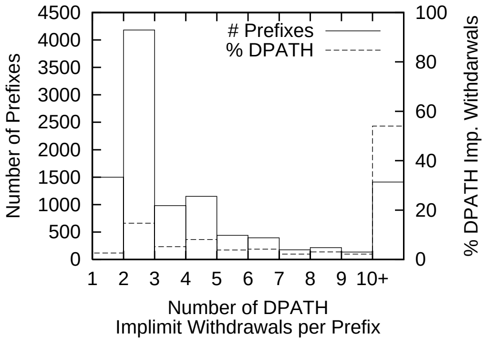
returned to a normal level after about three days (the figure does not show the days following Sept. 18). However, we can hardly see the same behavior in ISP5 (Figure 6(b)), because there is a huge spike reaching 58,150 around 5pm (most points are below 2500). If we remove the big spike by restricting the Y-axis to be between 0 and 2500, we see that ISP5's curve is very similar to ISP1's (Figure6(c)).

Based on empirical observations and additional information derived from the routing table, it is plausible to infer that the operational BGP session between the router at ISP5 and one of its peers, let's call it ISP-N, went down at this time. As a result, the ISP5 router had to replace all the routes that used ISP-N as the next-hop AS. Because the monitoring point peers with ISP5, we observed all the implicit withdrawals ISP5 sent out. When the session between ISP5 and ISP-N was re-established a few minutes later, all the affected routes had to be restored to their original state, which means another wave of implicit withdrawals from ISP5 to its peers. We conclude that ISP-N was a transit provider and ISP5 used it to reach a large number of prefixes, thus a single BGP session reset between the two could lead to large numbers of implicit withdrawals during a very short time period. The router at ISP1 could also have had BGP session resets with its clients or smaller ISPs but probably did not have any session resets with a major transit peer on Sept. 18, therefore it did not exhibit similar spikes in implicit withdrawals.

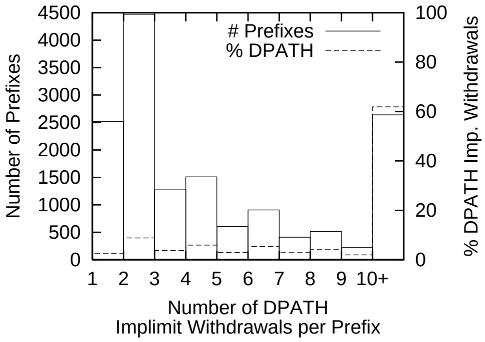
The above behavior indicates that a session reset between two BGP routers may lead to a cascading effect on other routers. Moreover, when BGP sessions resets involve major carriers and are caused by transient failures, routers adjacent to the involved carriers will typically see *rapid* route changes associated with a *large* number of prefixes over a *short* time period. In fact, we also observed a few spikes in five of the other peers (ISP4, ISP7, ISP8, ISP9, and ISP10) on Sept. 18. Thus we infer that, during the worm attack, some BGP session resets did occur in the operational network relatively close to these monitored routers, although the total number of the resets was probably much lower than that of the monitoring sessions. Note however that even those peers who show no spikes in the number of DPATH implicit withdrawals still have a noticeable increase in their curves. The following two sections examine the causes of these increases.

E.2 Prefix Distribution

We first show which prefixes contributed most to the routing changes, by plotting the distribution of prefixes based on the number of DPATH implicit withdrawals each prefix receives on a day. Figure 7 compares ISP1's distribution on Sept. 17 and Sept. 18. In both figures, the X-axis is the number of DPATH implicit withdrawals per prefix and we have divided it into ten bins, i.e. 1, 2, …, 9, 10 and up. The solid line represents how many prefixes fall into each bin and the dashed line represents what percentage of DPATH implicit withdrawals were generated by these prefixes. Their values should be read from the left and right Y-



(a) Sept. 17



(b) Sept. 18

Fig. 7. Distribution of the Number of DPATH Implicit Withdrawals per Prefix (ISP1)

axis respectively. For example, the first column of Figure 7(a) indicates that 1,499 prefixes had one DPATH implicit withdraw on Sept. 17 and they account for 2.6% of the total number of DPATH implicit withdrawals ISP1 sent on that day.

Comparing Figure 7(a) and (b), we can see that there are more prefixes in all the bins on Sept. 18 than on Sept. 17. This means there were more prefixes involved in routing changes on Sept. 18. But still only 14.4% of the prefixes in ISP1's routing table had at least one route change on Sept. 18.

Let's call the prefixes receiving 10 or more DPATH implicit withdrawals "highly unstable". Figure 7 shows that this group of highly unstable prefixes contributed disproportionately to the total count. On Sept. 17, 54% of the DPATH implicit withdrawals were sent for only 1,412 prefixes (1.4% of the routing table). The number in this bin increases to 2,649 on Sept. 18, which means more prefixes became highly unstable. However, they still constitute less than 3% of the routing table and contributed to 61.8% of the total DPATH implicit withdrawals. In fact, all the peers, except one that does not have proper rate limiting (see Section III-F), had less than 5,000 highly unstable prefixes on Sept. 18, yet this group of prefixes almost always contributed to more than 40% (and sometimes 80%) of all the DPATH implicit withdrawals a peer sent.

We examined 16 prefixes that appeared the most unstable in ISP1's distribution on Sept. 18 (each of them had at least 239 DPATH implicit withdrawals on that day), and found that 13 of them belong to a Cable Modem service provider in the US, 2 of them belong to a DSL and dialup service provider in the US, and 1 belongs to a small service provider in Argentina. Therefore, we suspect most of the highly unstable prefixes during the worm attack are located in edge networks. According to one major router vendor [7], the large number of probes that "Code Red" worms (and similarly Nimda worms) sent to random IP address caused routers in the edge networks to fill up their ARP cache and these routers would restart when their memory is exhausted. The high traffic load also caused some low-end routers to reboot. As a result, the BGP sessions between these networks and their providers would constantly break. These events could have triggered a large number of implicit withdrawals from those providers to their peers. And as we will explain in the next section, the BGP slow convergence problem further amplified the transient instability.

E.3 BGP Slow Convergence

We believe that a substantial amount of implicit withdrawals are likely due to the BGP slow convergence problem [5] and they are bogus routing changes. Figure 8 shows an example from the BGP updates during the worm attack.

This example shows that prefix 66.133.177.0/24 was withdrawn by its originator AS3549 at 14:05:10 (AS3549 sent a withdraw message to the monitoring point). We also observed, from the other peers' updates, that AS3549 withdrew this prefix from those peers at the same time, so it is

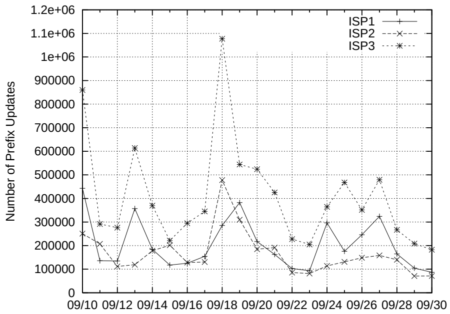
Time	AS	Action
09/18/2001 14:04:23	AS3549	originated prefix 66.133.177.0/24
09/18/2001 14:04:37	AS1103	announced aspath 1103 3549
09/18/2001 14:05:10	AS3549	withdrew 66.133.177.0/24
09/18/2001 14:05:36	AS1103	announced aspath 1103 8297 6453 3549
09/18/2001 14:06:34	AS1103	announced aspath 1103 8297 6453 1239 3549
09/18/2001 14:07:02	AS1103	withdrew 66.133.177.0/24

Fig. 8. BGP Slow Convergence Example

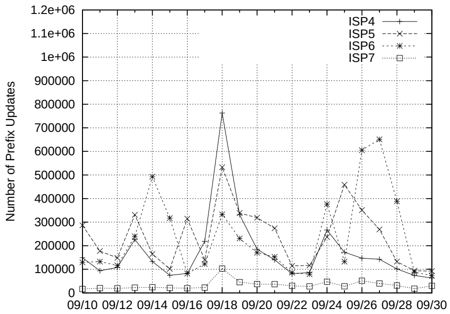
most likely that the connection from AS3549 to the 66.133.177.0/24 network was broken. In reaction to the withdraw message, AS1103 subsequently announced two different AS paths to the prefix. However, both paths were already obsolete because they went through the originator AS3549. The underlying problem is that BGP's path selection algorithm merely tries all the available paths when it receives a withdraw message, regardless of whether these paths have already been invalidated by the withdraw message[5]. Such exhaustive search not only results in long convergence time, but also produces a significant number of unnecessary BGP updates. In particular, this withdrawal triggered a total of 13 implicit withdrawals from six peers, all of which can be avoided if the mechanism proposed in [8] is deployed.

F. Different Behavior among ISPs

Due to differences in implementations and routing policies, the 12 routers peering with the monitoring point exhibited different behaviors, especially during the worm attack period. As we noted earlier, different ISPs generated different amount of duplicate updates and SPATH implicit withdrawals. Comparing the *total* volume of updates from different peers, we also found that one of the peer ISPs sent substantially more updates than others. Figure 9(a) shows the total numbers of updates by the 3 US peers. We note that ISP3 sent more updates than the other two in general, and more than two times during the attack. Closer inspection reveals that this peer did not exercise proper rate limiting on its updates. For example, the prefix 200.16.216.0/24 was first withdrawn at 00:07:15 on 9/10/2001 and then was announced four times in the following four seconds. This behavior violates the BGP4 specification [1] which prohibits the same prefix from being updated multiple times before the expiration of MinRouteAdverTimer (the recommended value for MinRouteAdverTimer is 30 seconds). After consulting the manual of the BGP implementation, we concluded that this behavior is the result of a combination of implementation defect and mis-configuration. The implementation seems benign in normal operations but leading to more pronounced impact under stressful conditions.



(a) US Peers



(b) Peers in Netherlands

Fig. 9. Comparison among the Peers

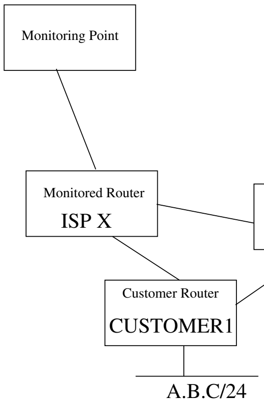


Fig. 10. Effect of Routing Policy on Monitoring

Figure 9(b) shows that one of the peers in Netherlands (ISP7) sent a much lower number of updates than all the others (note the curve near the bottom of the figure). We determined later that, contrary to the monitoring point's peering agreement, this router was not exchanging the full routing table with RRC00. Therefore our analysis is likely to have missed certain routing dynamics from ISP7, as explained in the next section.

G. Limitations in the Current BGP Monitoring Setting

BGP allows for a rich variety of routing policies that control the type of routes advertised (exported) and accepted by a BGP peer. Due to the constraints of export policies, a monitoring point may receive incomplete routing information from a peer and this may affect the accuracy of monitoring, as illustrated by the example in Figure 10.

In Figure 10, suppose the router at ISPX applies an export policy and sends only its customer routes to the monitoring point. Such policy can hide BGP route dynamics regarding prefix A.B.C/24. ISPX has two routes to prefix A.B.C/24: one route is learned directly from CUSTOMER1 and the other indirect route is learned from ISPY. The shorter route learned from CUSTOMER1 is preferred and advertised to the monitoring point. When the direct link to CUSTOMER1 fails, ISPX will select the back-up route via ISPY to reach CUSTOMER1. Because this back-up route is learned from ISPY and fails to match the export policy, ISPX simply sends a withdraw message to the monitoring point instead of informing the monitoring point the new route via ISPY. Due to this policy, the monitoring point cannot distinguish whether the reachability to CUSTOMER1 is lost or is replaced by another route in this case. Furthermore, the backup route via ISPY may experience BGP fluctuations while in use, unfortunately none of these fluctuations will be visible to the monitoring point.

Our effort in inferring the causes of spikes in DPATH implicit withdrawals also encountered challenges due to lack of necessary data. In Section III.E.1, we could only infer, based on indirect observation at the monitoring point, that the observed spike in Figure 6(b) was caused by the reset of operational peering session between ISP5 and ISP-N. Had we had monitoring data showing all the peering session state at ISP-N, we would have been able to confirm the inference.

As we have repeatedly pointed out in this paper, the current monitoring practice introduces certain measurement artifacts that could easily distort the results, if one is not careful in removing those artifacts. To overcome this and other limitations, we plan to collaborate with the University of Oregon in the design of a next generation Route Views project which should greatly alleviate measurement problems by putting measurement probes directly at a few critical peering points (to remove the eBGP multi-hop problem), and collecting full, as opposed to partial, views from their peers.

IV. Related Work

While BGP has been widely used in the Internet, its behavior in a real-world environment is yet to be fully understood. Labovitz, et. al. ([9] studied BGP routing messages exchanged between US service providers and reported that the majority of BGP messages consisted of redundant pathological announcements. [6] further identified the origins of certain pathological behavior. They also showed that routing instability had been significantly reduced in the core network by software improvements. However, our study shows that some of the previously identified problems, such as duplicate announcements and improper rate limiting, not only still exist in some of the today's operational networks, but they also produced far more pronounced impacts during the worm attack period.

Two studies have evaluated the effects of network congestion on BGP's behavior in experimental environments. Malan and Jahanian [10] observed in all of their experiments that, when the network was under peak utilization, TCP failed to deliver the keep-alive message in time and the peering session would break. Shaikh et. al. [11] also studied BGP's behavior under severe network congestion. Their analytical models and experimental results confirmed that, as the congestion level increases, the expected lifetime of a BGP session decreases. Our empirical observation seems to corroborate their findings. We showed that the multi-hop BGP peerings used by the monitoring point indeed suffered from severe network congestion and resulted in peering session failures.

Chang, et al [12] studied the effects of large BGP routing tables on commercial routers. They showed that some routers would reset one or all of the BGP peering sessions when they run out of memory and then repeat this behavior after they re-establish the BGP sessions. As a result, routing table size would oscillate. When such routers form a chain, the routing table oscillation would propagate. They also studied whether various existing mechanisms can prevent the BGP sessions from failing under large routing table load.

The above three experimental studies show how BGP may behave abnormally under certain extreme conditions. Our study provides the first in-depth analysis of BGP's behavior under stressful conditions in real operational environment.

One way to mask the negative effect of session resets is to retain stale BGP routes across resets, as proposed in [13]. This so called "Graceful Restart" mechanism enables a restarting BGP speaker and its peer to continue using the routes learned from each other in the previous session until the restart process is finished. Although this mechanism can minimize the negative effect of BGP resets, it can potentially lead to routing loops or blackholes if there are topology changes during the restart process.

V. Conclusion

Through in-depth BGP log data analysis, we conclude that BGP stood up well during the Nimda worm attack; the majority of the network prefixes exhibited no significant routing instability. Our analysis, however, does reveal several weak points both in the protocol and in its implementation. We suggest that BGP be further improved in order to be better prepared for unforeseen future network faults or attacks. In particular, we would like to point out the following three issues:

First, although the excessive BGP session resets at the monitoring point are a monitoring artifact, it is an evidence that BGP peering does not work well over "rocky" network connectivities. Even though BGP peering sessions seem relatively stable over good connectivity of short distance, the common setting in today's operational Internet, we believe that a global routing protocol must be truly robust and perform well even under adverse conditions.

Secondly, although Code Red and Nimda attacks mainly affected connectivity at certain edges, whose intermittent reachability rippled through to the rest of the Internet as rapid BGP update exchanges, it is an evidence that, with the current BGP design, a local change can indeed cause a global effect. A truly resilient *global* routing protocol must keep local changes *local* in order to scale.

Finally, although BGP's slow convergence after a failure or route change has not been shown to significantly impact the Internet performance, during the worm attack, the slow convergence's "amplifier" effect made the superfluous BGP updates, such as those due to local connectivity changes, multiple-fold worse. Our analysis suggests the need to quickly deploy a BGP fast convergence solution, such as the one proposed in [8], in order to get BGP well prepared to run gracefully under stressful conditions.

Last but not the least, we also identified areas to improve the current network measurement and monitoring effort. To understand BGP *in action* requires collecting data from the *operational* Internet. Although the current effort, such as [4] and [14], aims at that goal, the results seem to suffer from two shortcomings. The first issue concerns how well the monitoring setting matches the operational setting. The second, and perhaps more fundamental issue concerns the *indirection* of the measurement. One cannot collect all the information needed to understand the operation of a router through a BGP peering session between a monitoring point and the router; a complete understanding requires direct monitoring of the router's peering sessions with all its other *operational* neighbors.

VI. ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers whose constructive and insightful comments helped improve the qualify of this final version.

REFERENCES

[1] Y. Rekhter and T. Li, "Border Gateway Protocol 4," *RFC 1771*, July 1995.

[2] J. Cowie, A. Ogielski, B. J. Premore, and Y. Yuan, "Global routing instabilities triggered by Code Red II and Nimda worm attacks," Tech. Rep., Renesys Corporation, Dec 2001.

[3] Networking System Adminisration and Security Institute (SANS), "Nimda worm/virus report," http://www.incidents.org/react/nimda.pdf.

[4] RIPE, "Routing Information Service Project," http://www.ripe.net/ripencc/pub-services/np/ris-index.html.

[5] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet routing convergence," in *Proceedings of the ACM SIGCOMM*, August/September 2000.

[6] C. Labovitz, G. R. Malan, and F. Jahanian, "Origins of internet routing instability," in *Proceedings of the IEEE INFOCOM '99*, New York, NY, March 1999, pp. 218–26.

[7] Cisco Systems, "Dealing with mallocfail and high cpu utilization resulting from the "code red" worm," http://www.cisco.com/warp/public/63/ts_codred_worm.shtml.

[8] D. Pei, X. Zhao, L. Wang, D. Massey, A. Mankin, S. Wu, and L. Zhang, "Improving BGP convergence through consistency assertions," in *Proceedings of the IEEE INFOCOM*, June 2002.

[9] C. Labovitz, G. R. Malan, and F. Jahanian, "Internet routing instability," in *Proceedings of the ACM SIGCOMM '97*, Cannes, France, September 1997, pp. 115–26.

[10] G. R. Malan and F. Jahanian, "An extensible probe architecture for network protocol performance mea surement," in *Proceedings of the ACM SIGCOMM '98*, Vancouver, BC, Canada, September 1998.

[11] A. Shaikh, A. Varma, L. Kalampoukas, and R. Dube, "Routing stability in congested networks: Experimentation and analysis," in *Proceedings of the ACM SIGCOMM 2000*, Stockholm, Sweden, September 2000, pp. 163–74.

[12] D.-F. Chang, R. Govindan, and J. Heidemann, "An empirical study of router response to large BGP routing table load," Tech. Rep. ISI-TR-2001-552, USC/Information Sciences Institute, December 2001.

[13] S. Ramachandra, Y. Rekhter, R. Fernando, J. Scudder, and E. Chen, "Graceful restart mechanism for BGP," *Internet Draft*, October 2000.

[14] Univeristy of Oregon, "The Route Views Project," http://www.antc.uoregon.edu/route-views/.