Understanding BGP Misconfiguration

Ratul Mahajan David Wetherall Tom Anderson

{ratul,djw,tom}@cs.washington.edu
Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350

ABSTRACT

It is well-known that simple, accidental BGP configuration errors can disrupt Internet connectivity. Yet little is known about the frequency of misconfiguration or its causes, except for the few spectacular incidents of widespread outages. In this paper, we present the first quantitative study of BGP misconfiguration. Over a three week period, we analyzed routing table advertisements from 23 vantage points across the Internet backbone to detect incidents of misconfiguration. For each incident we polled the ISP operators involved to verify whether it was a misconfiguration, and to learn the cause of the incident. We also actively probed the Internet to determine the impact of misconfiguration on connectivity.

Surprisingly, we find that configuration errors are pervasive, with 200-1200 prefixes (0.2-1.0% of the BGP table size) suffering from misconfiguration each day. Close to 3 in 4 of all new prefix advertisements were results of misconfiguration. Fortunately, the connectivity seen by end users is surprisingly robust to misconfigurations. While misconfigurations can substantially increase the update load on routers, only one in twenty five affects connectivity. While the causes of misconfiguration are diverse, we argue that most could be prevented through better router design.

Categories and Subject Descriptors

C.2.3 [**Communication Networks**]: Operations—*management*; C.4 [**Computer Systems**]: Performance—*reliability, availability, and serviceability*

General Terms

Human Factors, Management, Reliability

1. INTRODUCTION

As the Internet's inter-domain routing protocol, the Border Gateway Protocol (BGP) [34] is crucial to the overall reliability of the Internet. Faults in BGP implementations or mistakes in the way it is used have been known to disrupt large regions of the Internet. Recent studies have examined several kinds of BGP problems, including excessive churn due to implementation deficiencies [26], delayed convergence [24], persistent oscillations due to policy interactions [18, 40], and instability caused by the propagation of worms [9].

In this paper we examine another source of unreliability: the misconfiguration of the routers that speak BGP. We know from numerous studies of highly reliable systems, such as aircraft, bank databases, and the telephone network, that human operator error can account for 20-70% of system failures [3, 6, 15, 21]. These studies have shown that as systems become more reliable, the human factor becomes increasingly important to overall reliability. We would expect the same to be true of the Internet. There is substantial anecdotal evidence that BGP configuration errors do occur, with serious consequences. The canonical example is the AS7007 incident [31], in which AS7007 accidentally announced routes to most of the Internet and disrupted connectivity for over two hours. Despite the publicity over this event, serious misconfigurations continue to occur. In April 2001, AS3561 propagated more than 5000 improper route announcements from one of its downstream customers [12], again leading to global connectivity problems.

In this paper, we complement the anecdotal study of infrequent large scale events with a microscopic study of more frequent "near misses" – globally visible BGP misconfigurations that occur many times per day but do not necessarily disrupt connectivity. As with the study of airplane near collisions, we hope that our study of microscopic events can help improve the design of systems to avoid future larger scale problems. To the best of our knowledge, this is the first systematic study of globally visible BGP misconfigurations; it complements other studies that have examined backbone failures in general [25]. Our goal is to answer four questions:

- *How frequently do these misconfigurations occur?*
- *What is their impact on global connectivity and routing load?*
- *Why do the misconfigurations occur?*
- *What can be done to reduce their frequency and impact?*

We consider two broad classes of faults that propagate across the backbone and hence are visible from our measurement points: i) the accidental injection of routes into global BGP tables, including address space hijacks; and ii) the accidental export of routes in violation of an ISP's policy. We focus on misconfigurations that are globally visible because they arguably have the potential to cause wider disruption than those that do not propagate across the Internet We analyze the entire stream of BGP updates taken from 23 different vantage points around the Internet for a period of 21 days. We show that it is possible to identify misconfigurations with simple heuristics. To validate our results, we surveyed the ISP operators involved in each incident via email. We asked them whether the in-

SIGCOMM'02, August 19-23, 2002, Pittsburgh, Pennsylvania, USA.

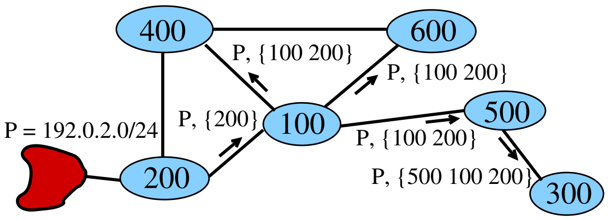

Figure 1: Border gateway protocol (BGP) network example.

cident was a misconfiguration and what caused it. We also actively probed the Internet to determine the impact of misconfiguration on connectivity. Our results should be considered a lower bound on the extent of BGP operator error because we do not observe misconfigurations with local effects and our heuristics for identifying global misconfigurations are conservative.

We find that 200-1200 prefixes, equivalent to 0.2-1.0% of the global BGP table size, suffer misconfiguration events each day and that the routing load of these incidents can be substantial. Misconfigurations increased the route update load by at least 10% for 2% of the time, and we observed one 15 minute period for which misconfigurations doubled the load across all of our vantage points. Close to 3 in 4 of the new prefix announcements seen each day are the result of misconfiguration. Despite this, we find connectivity to be surprisingly robust; only 4% of the bad announcements disrupt connectivity.

The causes of misconfiguration we uncover are diverse. In addition to involuntary slips by network operators, errors also result from router initialization bugs and a poor understanding of configuration semantics on the part of operators. Fortunately, as one would expect, a few BGP features are responsible for many of the misconfigurations. We uncovered a potential bug in the software of a major router vendor, now being investigated by the vendor, that is responsible for a large number of accidental insertions during router resets. And we identified a commonly used style for configuring routers that leads to unintended behavior during failures.

Based on our experience with this study, we propose several changes to router and protocol design that would eliminate most of the misconfigurations we observed. For instance, adding transactional semantics for configuration commands, by itself, would eliminate 22% of the accidental route insertions we observed.

The rest of the paper is organized as follows. Section 2 provides a brief background on BGP and router configuration. In Section 3 we specify what kinds of misconfiguration we study, and their impact on the Internet. Section 4 describes the methodology of our study, and Section 5 presents our results. We discuss the causes of misconfiguration in Section 6, and how the problems can be avoided in Section 7. Finally, we present related work and conclusions in Sections 8 and 9 respectively.

2. BACKGROUND

BGP is a path vector routing protocol run between autonomous systems (ASes) in the Internet. As shown in Figure 1, a BGP route announcement consists of a network prefix and a list of ASes: $(P, [AS_k\ AS_{k-1}\ \cdots\ AS_0])$. P is the block of IP addresses being announced; the list of ASes, called the *AS-Path*, is the ordered list of ASes traffic to P would traverse. The last AS in the list, AS_0, is the origin AS, or simply the origin, of the announcement. Each AS exports the routes to its neighbors after adding itself in the front of the received AS-path.

```
router bgp 200
network 192.0.2.0
neighbor 1.1.1.1 remote-as 100
neighbor 4.4.4.4 remote-as 400
neighbor 4.4.4.4 route-map RMAP out
!
route-map RMAP permit 10
match ip-address 1
!
access-list 1 deny 192.0.2.0 0.0.0.255
```

Figure 2: Sample BGP configuration for AS200 in Figure 1.

BGP is a policy based protocol; each AS chooses among the multiple routes it receives for the same prefix according to its own criteria. An AS can apply policy when exporting a route as well. An AS exports a route to a neighbor only if it is willing to accept and forward traffic to the prefix from that neighbor. For instance, in Figure 1 AS200 chose not to export its route to P to AS400.

BGP routers are individually configured using a configuration script. The script may be generated automatically, but is often written and edited by hand.

Figure 2 shows an example configuration in Cisco format [7] for a router in AS200. The first four lines configure the router to announce the network 192.0.2.0 from AS200, and to peer with two other routers, one in AS100 and another in AS400. In BGP, ASes may filter incoming or outgoing announcements to implement policies such as peering and transit. Filtering can be implemented using prefix filters, access lists, and route maps. Route maps are the most popular form of filtering. With a route map, an AS can specify both a matching filter and a list of actions to be taken upon a match. Actions can include setting various BGP parameters, such as local preference, MED (multi-exit discriminator), and community. In Figure 2, AS200 is configured to prevent announcements for the network 192.0.2.0 from being advertised to AS400.

Using these basic primitives and a few others, an AS can control the flow of announcements between its routers and their BGP peers.

3. MISCONFIGURATION

We define *misconfigurations* to be configuration errors that result in the unintended production or suppression of BGP routing announcements. Our definition includes both *slips* (inadvertent errors) and *mistakes* (design errors) in human factors terminology [33]. It is arguable whether intended but unconventional practices should be considered misconfigurations. For instance, for an extended period of time a major AS in the Internet was intentionally introducing loops in the AS-path to achieve a certain policy objective, a practice that most operators would consider undesirable. However, for the purposes of this paper we focus only on whether the AS intended the behavior. This requires us to verify each potential misconfiguration incident with the AS involved in it, a cumbersome but necessary step in the absence of universally accepted operational practices.

In this paper, we focus on two types of globally visible BGP misconfigurations:

1. **Origin Misconfiguration:** An AS accidentally injects a prefix into the global BGP tables. Examples of origin misconfiguration include i) failure to summarize an address space, leading to the injection of one or many more-specific prefixes into the global BGP tables; ii) *hijacks* – announcing part of

someone else's address space; and $iii)$ propagating prefixes meant to stay inside a private network.

2. **Export Misconfiguration:** The AS-path is in violation of the policies of one the ASes in the path, because the router exported a route it should have filtered. For example, in Figure 1 AS 400 could export the route to P to AS 600 against its policy.

We choose to focus on these global changes because they arguably have the greatest potential to widely disrupt Internet connectivity. Of course, there are other kinds of BGP misconfigurations that are not easily identifiable from observing changes to the global BGP tables. For instance, an AS may accidentally filter out routes it otherwise meant to announce; to a remote observer, this denial-of-service would be indistinguishable from a failure. Similarly, MED misconfigurations will typically be observable only between the participating ASes. A different methodology, such as examining events visible within each AS, would be needed to study the frequency and root causes of these kinds of errors.

The adverse impacts of misconfiguration include:

• **Routing load:** Misconfigurations increase routing load by generating unnecessary BGP updates. Many BGP speaking routers are already heavily loaded due to the rapid growth of the Internet [29]; any unnecessary load is a cause of concern in the operations community.

• **Connectivity disruption:** Misconfigurations can disrupt connectivity, either partially (from some parts of the Internet) or globally (from everywhere).

• **Policy violation:** By definition, misconfigurations violate the intended policy of the AS. For example, prefixes can be incorrectly leaked to the entire Internet, routes announced in error can be chosen over intended ones, and transit can be inadvertently provided to other ASes.

4. METHODOLOGY

We use data collected by the RouteViews' [30] new BGP listener, `route-views2.oregon-ix.net`[1]. At the time of the study, it had 23 peers in 19 different ASes.

Identifying misconfigurations in a stream of BGP updates is not an easy task because misconfigurations have signatures that are similar to those of events such as failures and policy changes. Our analysis is based on the assumption that policy changes typically operate on human time-scales, while changes due to misconfigurations and failures typically last for a much shorter time.

Figure 3 illustrates why the above approach makes sense. It plots the cumulative distribution function (CDF) of the length of time a new route lasted in the BGP table during the month of Nov., 2001. A *new route* is either a new prefix or a new origin for an existing prefix. We see that most new routes either last less than a day or last much longer; 45% of the changes last less than a day while 30% of them lasted more than 7 days. We hypothesize that the former include origin misconfigurations, while the latter are mainly due to policy changes or Internet growth.

Thus, to identify misconfigurations we focus on short-lived changes that last less than a day; the nature of the change depends on the type of misconfiguration, and is described in the following subsections. Some of the short-lived changes that we identify as potential misconfigurations can also be caused by legitimate events. To disambiguate, we use an email survey of operators involved in each incident. To determine the impact of misconfigurations on connectivity, we also actively probe the Internet.

[1]Even though it has fewer peers, we prefer it over the original listener as it archives all BGP updates and not just snapshots. This is important because most of the misconfigurations that we study last for less than the two hour snapshot interval of the older listener.

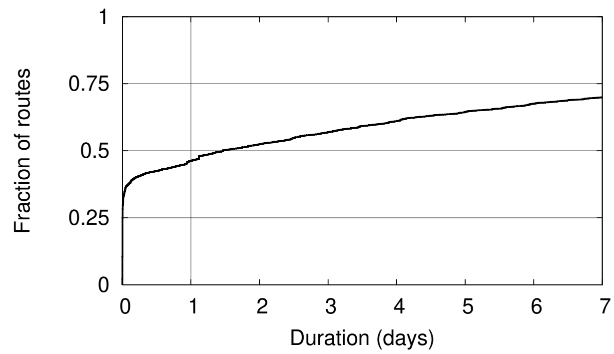

Figure 3: Fraction of new routes lasting $\leq x$ days.

	Old route	**New route**
Self deaggregation	a.b.0.0/16 X Y Z	a.b.c.0/24 X Y Z
Related origin	a.b.0.0/16 X Y Z	a.b.0.0/16 X Y a.b.0.0/16 X Y Z O a.b.c.0/24 X Y a.b.c.0/24 X Y Z O
Foreign origin	a.b.0.0/16 X Y Z	a.b.0.0/16 X Y O a.b.c.0/24 X Y O e.f.g.h/i X Y O

Table 1: Classification of origin misconfigurations.

We arrived at this simple strategy after discovering that other more obvious approaches do not work. For instance, comparing observed routes to those recorded in the Internet routing registries (IRRs) does not work well because of inaccurate or outdated registry information (Section 7.3).

Our study underestimates the extent of misconfiguration in several ways. First, we only consider misconfigurations that last less than a day, as a result of which we miss errors that persist for a longer time. However, Figure 3 shows that increasing this period beyond a day yields diminishing returns. Second, we do not consider all kinds of misconfigurations. Third, we can only observe misconfigurations that reach RouteViews despite filtering and best-path selection at each hop. For these reasons, our results should be viewed as a lower bound on the extent of BGP misconfiguration.

4.1 Origin Misconfiguration Analysis

An origin misconfiguration is the unintentional insertion of a route into the global BGP tables. Assuming the operator quickly caught the error, it would show up as a short-lived new route. Note that failures usually have the opposite signature – short-lived route disappearance. We use historical BGP data from the previous day to differentiate between new routes and old routes that reappear due to the end of a failure period. Future BGP updates are then used to determine how long a new route lasted. New routes appearing because of policy changes such as multi-homing, traffic engineering, and provider switch are likely to last longer than those due to misconfiguration. Occasionally, legitimate events such as failures and load balancing can show up as short-lived new routes, but we will see in Section 5.1 that this happens infrequently.

To shed light on the causes of origin misconfiguration, we clas-

Route export	Export policy
Customer → Provider	Only routes received from customers and siblings
Peer → Peer	Only routes received from customers and siblings
Provider → Customer	All routes
Sibling → Sibling	All routes

Table 2: Export policies for common commercial relationships. Providers provide transit to customers; peers exchange only traffic that is sourced and sinked by them, their customers or their siblings; and siblings provide mutual transit.

sify the new routes into various categories based on their relationship with the existing routes, as shown in Table 1. For each category, the table lists the old route and one of the possible new routes. In *self deaggregation*, an origin deaggregates its own prefix. In *related origin*, an existing prefix or its subset is announced by a new origin that appears related to the old origin in that one of the origins appears in the AS-path of the other. In *foreign origin*, a prefix or its subset is advertised by a different origin, and the two origins apparently have no relationship to one another.[2] New routes for prefixes that are neither present in the table nor have a less-specific prefix in the table are also classified as a foreign origin incident.

This classification is based on the likely underlying causes. For instance, it is likely that self-deaggregation incidents are the result of forgetting to aggregate at a router, while foreign origin incidents could be the result of an address space hijack. Similarly, a related origin is more likely to be actually connected to the network than a foreign origin is, though the latter can also be caused by a backup origin that only appears during failures.

4.2 Export Misconfiguration Analysis

An export misconfiguration (or a route leak, in operations parlance) is an inadvertent export of a route to a BGP peer in violation of the exporter's policy. Export policies arise from the commercial relationships between ASes. Table 2 shows common relationships between ASes and the export policies associated with them [14]. An example of an export misconfiguration is an AS exporting a route received from one provider to another, and as a result providing transit between its two providers.

If we knew the relationships between ASes, detecting export misconfigurations would be straightforward. But AS relationships are closely guarded secrets, which complicates identifying these errors. Gao infers these relationships from the BGP tables based on the following observations [14]. First, all valid AS-paths are *valley free*. Treating the provider to customer direction as downward, and siblings and peers at the same level, the *valley free* property means that a route that starts going downwards never goes up again. Second, an AS-path can have at most one peer-to-peer edge, which occurs at the highest point in the path. Third, ASes with more neighbors are more likely to be providers.

Based on the relationships inferred by Gao's algorithm[3] with historical BGP data as input, we identify AS-paths with short-lived subpaths that violate the valley free condition or contain multiple peering edges as probable export misconfigurations. Note that AS-paths observed during failures should still conform to the intended policy

Inferring AS relationships is not a perfect art, and as a result our analysis can miss misconfigurations or identify legitimate AS-paths as misconfigurations. The former leads us to underestimate real misconfigurations, and we will see in Section 5.2 that the latter happens infrequently.

Route export misconfigurations can be categorized as shown in Table 3. In each case, the AS is providing transit to traffic from its provider or peer to its provider or peer. The classification is based on the policy being violated. A route export misconfiguration can also contain siblings; for simplicity, a chain of siblings is considered to be one AS, resulting in the above classification.

Export	Policy violation
Provider→AS→Provider	Route exported to provider was imported from a provider
Provider→AS→Peer	Route exported to peer was imported from a provider
Peer→AS→Provider	Route exported to provider was imported from a peer
Peer→AS→Peer	Route exported to peer was imported from a peer

Table 3: Classification of route export misconfigurations.

4.3 Email Survey

It is not necessary that all the short-lived changes we identify are misconfigurations, though we will see in Section 5 that most of them are in fact misconfigurations. To disambiguate the intentions of the network operator, we conducted an email survey of the network operators involved in each incident using email addresses from the Internet routing registries (IRRs). Each incident report contained the prefix(es), the AS-path(s), and the start time and the duration of the incident. We asked each operator whether the incident was a misconfiguration, whether it caused any connectivity problems, and what caused the short-lived change.

As an aside, inaccurate or stale information in the IRRs caused roughly 30% of the emails to bounce, and caused many of them to reach people who no longer worked at the organization or who were not related to the organization. However, the email survey worked often enough to provide useful data.

4.4 Testing Connectivity

Finally, we implemented a connectivity verifier to determine the extent of disruption due to misconfigurations. The verifier first downloaded the current BGP table from RouteViews. Then for all the suspect routes in the table, it checked if the prefix was reachable from various vantage points in the network. Multiple vantage points are required because BGP misconfigurations can lead to routes being available from some parts of the Internet while not from others, for example, when part of an address space is hijacked. We used public traceroute servers [23] to test for connectivity. Connectivity problems due to misconfiguration cause a prefix to be unreachable from the ASes that use the suspect route for the duration of the misconfiguration. If the prefix becomes reachable when the route disappears, it strongly suggests that the earlier lack of connectivity was due to the misconfiguration. A prefix is considered reachable if we can reach a live host within it. We seeded our databases of otherwise valid responsive IP addresses from Skitter [38], and performed a randomized scan to find a host within a

[2]Since we do not have a complete view of the Internet, some related origin instances may be classified as foreign origin; but misclassification should be infrequent given the rich view provided by RouteViews.

[3]We use the version of Gao's algorithm presented in Section 4.2 of the paper [14]; it infers peering relationships along with provider-customer and sibling relationships.

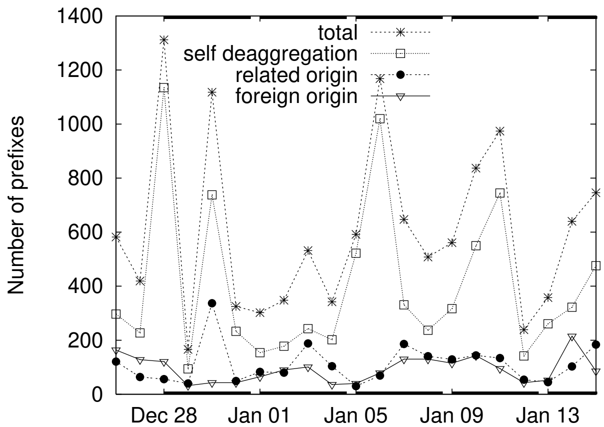

Figure 4: Probable origin misconfigurations per day.

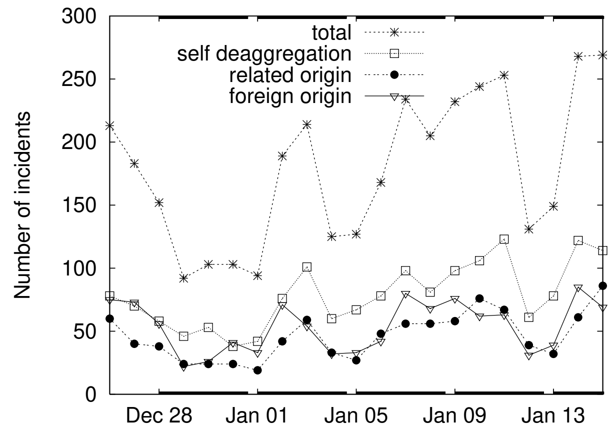

Figure 5: Probable origin misconfiguration incidents per day.

prefix we wanted to test.

The test was used as a coarse check to verify the results of the email survey. The verifier will sometimes report connectivity disruption when there is none, for example, due to end host failures, ICMP filtering or convergence delays. To determine the baseline failure rate, we used the results of the test on incidents that were known not to be misconfigurations from the email survey. The reported failure rate of test was adjusted for this baseline failure rate.

5. RESULTS

This section presents the results of our study, collected over a period of 21 days from 26 Dec., 2001 to 15 Jan., 2002. We report on origin misconfigurations in Section 5.1 and on export misconfigurations in Section 5.2. Section 5.3 discusses the impact on routing load.

5.1 Origin Misconfiguration

Figure 4 shows the probable origin misconfigurations (short-lived new routes) seen each day. On average, there are roughly 600 short-lived new routes advertised per day. The variability in the data is striking. From the breakdown, it is clear that most of the contribution comes from self deaggregation. Prefixes from related origin and foreign origin incidents contribute roughly the same, smaller amount.

To analyze the data further, we cluster the misconfigured prefixes into incidents. An *origin misconfiguration incident* is a set of prefixes originated by the same origin AS, whose route appearance and disappearance are closely spaced in time. Prefixes, with the same origin, that appeared within 15 minutes (the worst case BGP convergence time reported in [24]) of each other, and also disappeared within 15 minutes of each other, were counted as part of the same incident. The results are fairly independent of the exact time interval because BGP updates caused by the same event are usually very closely spaced.

Figure 5 shows the number of origin change incidents seen per day. Self deaggregation no longer dominates as it did in the previous graph. This means that while the number of self deaggregation incidents is only slightly higher than that of other types, they contain a much higher number of prefixes per incident.

Table 4 shows the overall results of the email survey. We received useful email responses for nearly 30% of the incidents representing more than half of the short-lived prefixes. While it is hard to accurately comment on the remaining incidents, we believe that they likely follow the same distribution as those for which we had responses. This is because the results from the email responses and active testing for connectivity disruption are in close agreement with each other (Section 5.1.1).

From the aggregate results in the total row of Table 4, we see that most of the short-lived origin changes we were able to classify are indeed the results of misconfiguration. The fraction is as high as 96% for prefixes, and 86% for incidents. If we assume that the fractions in the total column are valid for all the short-lived changes, not just those for which we received useful email responses, roughly 580 ($0.96 \times \frac{12716}{21}$) prefixes are affected by origin misconfiguration per day. To put this in perspective, we observed roughly 200 new long-lived routes each day, which means that at least 72% of new routes seen by a router during the day are the result of misconfiguration.

From the breakdown of the origin misconfigurations into various types, we see that the success rate for identifying misconfigurations is different for different types. The main reason for misidentifying related and foreign origin misconfiguration is failures. Often, there exist backup arrangements in which a related or foreign origin starts announcing the prefix or its subset when the primary announcement fails. Multihoming with backup, in which the secondary link is used only when the primary fails, is one such common setup. We did not attempt to identify coordinated primary failures with backup announcements, given the uncertainties in BGP update propagation and convergence time.

5.1.1 Connectivity

Table 4 also shows that connectivity is disrupted by origin misconfiguration in 13% of the incidents. We found that our two methods for checking connectivity problems – the connectivity test (Section 4.4) and the email responses – agreed with each other. While the former reported that connectivity was disrupted in 11% of the probable origin misconfiguration instances we tested, the email responses reported that connectivity was disrupted for 12% (0.86×0.13) of the temporary new route incidents.

In terms of prefixes, we see that only 4% of the misconfigurations disrupt connectivity. This is considerably lower than the 13% figure for incidents because most of the large incidents are not accompanied by connectivity problems. From the results in Table 4 it is also apparent that the different types of origin misconfigurations disrupt connectivity to different extents. Self deaggregation seldom

Type	Prefixes (top) Incidents (bottom)	Classified (% of total)	Misconfigurations *All* (% of classified)	*Connectivity* (% of misconfigs)	Non-misconfigurations (% of classified)
Self deaggregation	8424	5598 (66%)	5519 (99%)	74 (01%)	79 (01%)
	1648	616 (37%)	563 (91%)	13 (02%)	53 (09%)
Related Origin	2341	1153 (49%)	1068 (93%)	54 (05%)	85 (07%)
	969	229 (24%)	188 (82%)	25 (13%)	41 (18%)
Foreign origin	1951	642 (33%)	535 (83%)	142 (27%)	107 (17%)
	1131	257 (23%)	194 (75%)	85 (44%)	63 (25%)
Total	12716	7393 (58%)	7122 (96%)	270 (04%)	271 (04%)
	3748	1102 (29%)	945 (86%)	123 (13%)	157 (14%)

Table 4: Results for origin misconfiguration. The *classified* column lists the number classified using email replies. Out of those, the *all* subcolumn specifies the number of misconfigurations, the *connectivity* subcolumn lists misconfigurations that disrupted connectivity, and the *non-misconfigurations* column specifies the number that were not misconfigurations (i.e. false positives in our analysis).

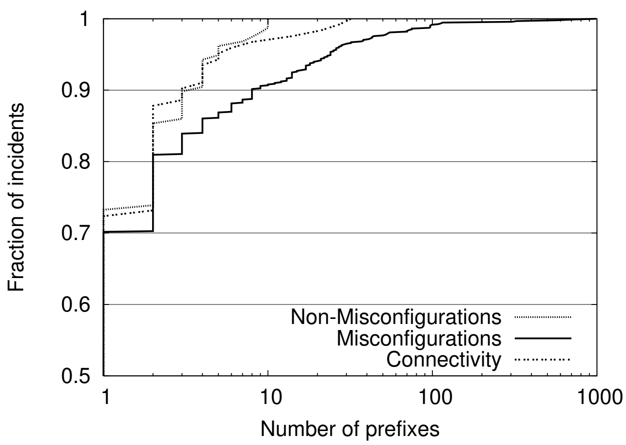

Figure 6: Fraction of incidents with $\leq x$ prefixes. Note that the y-axis does not start at zero.

causes connectivity problems, while foreign origin leads to connectivity disruption in almost half the misconfiguration incidents.

Interestingly, in some cases the impact on connectivity was not observed at the time by the network operators yet they confirmed that the incident reported would have caused connectivity problems. This failure to observe connectivity problems can happen for several reasons. First, some operators have no means of detecting connectivity problems in the absence of customer complaints. Second, most incidents that disrupt connectivity are short-lived, and users have come to expect intermittent failures on the Internet. Third, most connectivity incidents involved more-specifics of a stable prefix announcement. Because of sparse usage of Internet address space, they tend to affect only a small number of hosts.

To estimate the overall impact of misconfigurations on connectivity, we extrapolated the classified data by type to cover all of the incidents. The result is around 25 incidents that disrupt connectivity per day. To put this in perspective, we also counted failures as the short-lived disappearance of a prefix from all vantage points.[4] There were approximately 1000 failure incidents per day. Thus compared to failures, the globally visible misconfigurations we study do not play a large role in connectivity disruptions.

[4]While some short-lived BGP withdrawals can occur without failures, most of them are caused by failures [25].

5.1.2 *Prefixes and Incidents*

In Table 4, the statistics for prefixes are quite different from those of incidents. The reason can be seen in Figure 6, which plots the cumulative distribution function (CDF) of the number of prefixes in an incident for non-misconfigurations, all misconfigurations and misconfigurations that disrupt connectivity. It shows that while all the categories have a single prefix for roughly three quarters of all incidents, the misconfigurations have a heavier tail, with some of the incidents consisting of close to 1000 prefixes. The number of prefixes in non-misconfigurations is much smaller because they represent intentional events or configurations; usually, operators do not intentionally inject large number of prefixes into the global routing tables. The number of prefixes in misconfiguration incidents that impact connectivity is usually small, but goes as high as 30 during the period of study.

5.1.3 *Duration of Misconfigurations*

Figure 7 plots the CDFs of the duration of short-lived origin changes. Roughly half the misconfigurations last for less than 10 minutes, roughly 80% of them are corrected in less than an hour, and 95% of them are corrected in less than 10 hours. Short lived origin changes that are not misconfigurations last much longer; 55% last more than an hour, and 20% last more than 10 hours. The difference in the lifetime of misconfigurations and non-misconfigurations intuitively makes sense because, barring failures, non-misconfigurations represent intentional changes made to achieve policy objectives (Section 6.1), and thus work on timescales longer than those of accidental changes.

The graph in Figure 7 also shows that misconfigurations that disrupt connectivity get fixed sooner; only 50% of them last more than 10 minutes, and only 5% of them last more than an hour. We speculate that these misconfigurations are detected sooner than other misconfigurations, and are fixed with higher priority.

5.2 Export Misconfiguration

Figure 8 shows the number of probable export misconfigurations per day broken down by type. Each export misconfiguration is an AS-path in violation of some export policy. A *bad-export-sequence* is an ordered sequence of the three (or more with siblings) ASes that identifies the policy violation. The graph shows the number of AS-paths with a bad-export-sequence in them.

As we did for origin misconfigurations, we cluster export misconfigurations into incidents. A *export misconfiguration incident* is the set of paths with the same bad-export-sequence that appears and disappears at roughly the same time. Figure 9 shows the export

Type	Paths (top) Incidents (bottom)	Classified (% of total)	Misconfigurations (% of classified)	Non-Misconfigurations (% of classified)
Provider→AS→Provider	898	545 (61%)	512 (94%)	33 (06%)
	370	138 (37%)	114 (83%)	24 (17%)
Provider→AS→Peer	478	205 (43%)	201 (98%)	4 (02%)
	225	70 (31%)	66 (94%)	4 (06%)
Peer→AS→Provider	519	489 (94%)	486 (99%)	3 (01%)
	43	19 (44%)	17 (89%)	2 (11%)
Peer→AS→Peer	141	98 (70%)	91 (93%)	7 (07%)
	46	18 (39%)	14 (78%)	4 (22%)
Total	2036	1337 (66%)	1290 (96%)	47 (04%)
	684	245 (36%)	211 (86%)	34 (14%)

Table 5: Results for export misconfiguration. The *classified* column specifies the number classified using email replies. The next two columns list the number classified as misconfigurations and non-misconfigurations respectively.

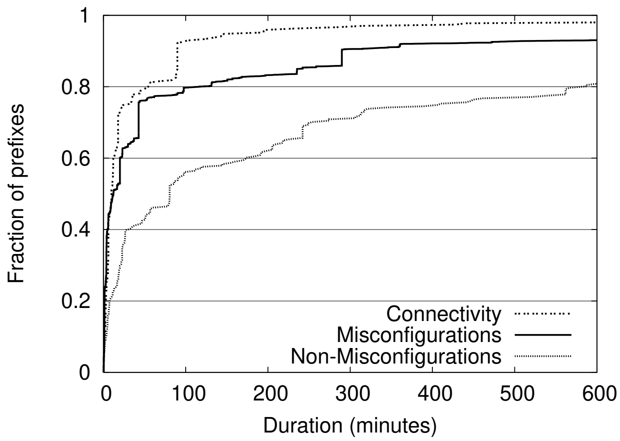

Figure 7: Fraction of short-lived routes lasting $\leq x$ mins.

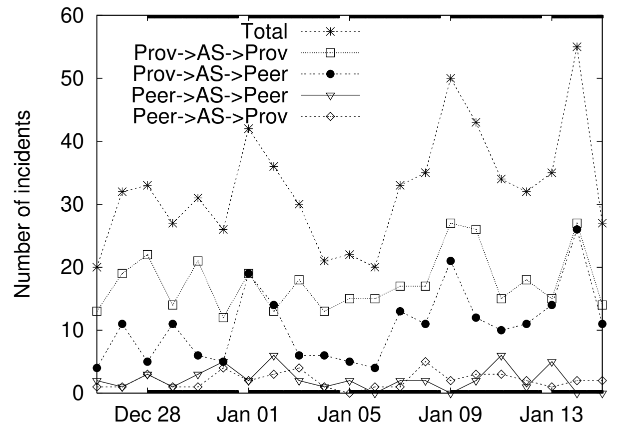

Figure 9: Probable export misconfiguration incidents per day.

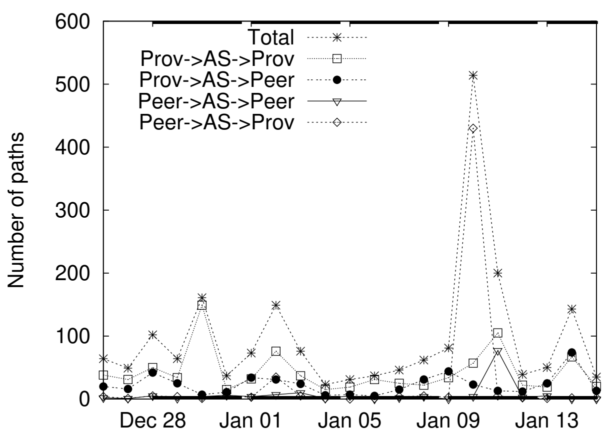

Figure 8: Probable export misconfigurations per day.

misconfiguration incidents per day. It is clear that the peak on Jan 11 in Figure 8 was a result of a single incident.

Table 5 shows the email survey results for export misconfiguration. Export misconfigurations do not cause connectivity problems directly, although they do bring extra traffic to the AS sourcing the problem. In some cases, our respondents observed link congestion and slowdowns, and in only a handful of incidents did they experience total outages. For simplicity, we ignore the connectivity outages due to export misconfigurations.

As in the case of origin misconfiguration, the misconfiguration detection accuracy is high. Most export misconfiguration incidents involve providers instead of peers, most likely because there are more AS-pairs with provider-customer relationship than peers.

5.3 Routing Load

Misconfigurations increase the load on the routing system by inserting spurious prefixes in the global BGP tables, and more importantly, by generating unnecessary updates to be processed by other BGP speakers. We define *routing load* as the fraction of updates seen due to misconfigurations. Updates due to misconfiguration are announcements and withdrawals of the new route for origin misconfiguration, and of AS-paths containing the bad-export-sequence for export misconfiguration.

Figure 10 plots the routing load experienced due to misconfigurations calculated using 15 minute averages to smooth bursts. The *confirmed* line represents the load due to misconfigurations confirmed using the email survey. The second line represents an extrapolation of routing load if the unclassified probable misconfigurations were problems at the same rate as the classified ones. While the load was low most of the time, it was more than 5% for 5% of

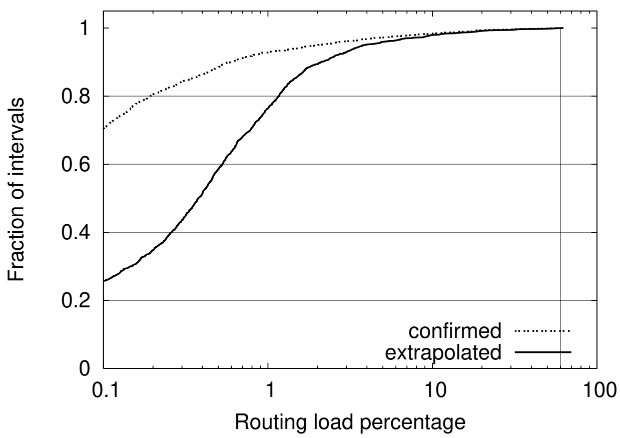

Figure 10: Fraction of 15 minute intervals with $\leq x\%$ routing load due to misconfigurations.

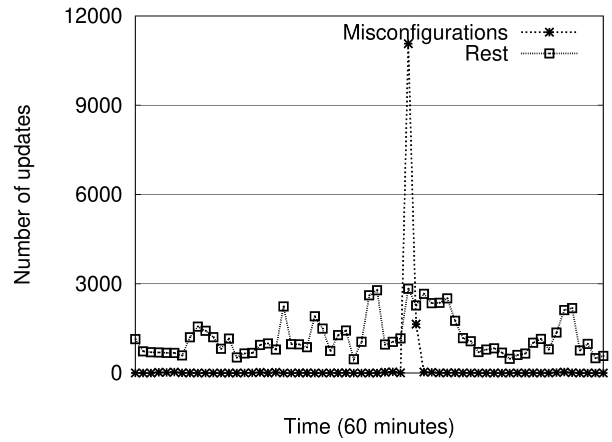

Figure 11: Microscopic view of BGP updates at RouteViews around the time of a misconfiguration incident.

the time, more than 10% for 2% of the time, and goes higher than 60% in the extreme even with 15 minute averaging (mainly because of deaggregation spikes). A routing load of 60% means that the update arrival rate more than doubled due to misconfigurations. This means that during certain periods of time, updates due to misconfiguration dominate in the same manner as those due to session resets [28].

As an example, Figure 11 shows the microscopic behavior of routing load around a misconfiguration incident that was confirmed by an operator. The graph plots the number of BGP updates received per minute by the RouteViews listener for a 60 minute period on 28 December when a major event occurred. The number of updates due to misconfigurations is small most of the time, but peaks astronomically for a very short time when a major misconfiguration happens.

6. CAUSES OF MISCONFIGURATIONS

We need to understand why misconfigurations occur before we can prevent them. Our aim is to be able to prevent all types of misconfigurations, despite the differing impact for the incidents we observed, because all have the potential to cause significant disruption in future incidents. In this section, we discuss the causes of the misconfigurations we observed based on information provided by the network operators who took part in our survey. The fidelity with which we can attribute causes to incidents is limited by the responses we received. Nevertheless, from the bulk of email responses, we were able to identify the common causes described below. Note that not all of the causes are mutually exclusive.

To categorize the causes, we use the standard classification of human errors into *slips* and *mistakes* [33]. *Slips* refer to errors in the execution of an otherwise correct plan. Examples include typos and forgetting a configuration command. *Mistakes* refer to errors in which the execution went as planned, but the plan itself was incorrect. Examples include implementation logic bugs and poor operational practices. It is not always possible to distinguish between a slip and a mistake; when in doubt we classify the cause as a slip assuming that the operators constructed the right mental plan for the configuration change.

6.1 Origin Misconfigurations

The causes of origin misconfiguration are shown in Table 6, which also lists the causes for short-lived new routes that were not misconfigurations. At the beginning of our study, we expected most of the misconfigurations to be caused by slips, but from the table it is clear that mistakes and slips are roughly equally responsible. We discuss mistakes first, and then slips.

Initialization bugs. During the course of this study we came across a frequent occurrence, in which an AS would announce several more-specific prefixes and withdraw them within 2-6 minutes. The number of more-specific prefixes ranged from 1-1000, with a median around 10. The operators who responded to such incidents claimed that they had made no configuration changes that could have caused this leak, but their router rebooted (intentionally or not), an interface went down or they ran their maintenance scripts during that time. We discovered that while a router is being rebooted or the filters are being updated, the more-specific prefixes present in the router's table can be leaked (pending new filters taking effect), only to be withdrawn when the reboot or update is complete. On deeper investigation we discovered that these occurrences are the result of a software bug in the routers of a major vendor, the interaction of configuration processing semantics with the way operators write their maintenance scripts,[5] or both. A bug report has been filed with the vendor, and is being investigated with the help of operators who observed this behavior.

Reliance on upstream filtering. Some ASes were found to announce routes based on the assumption that they would be filtered by their upstream provider, and thus would not be globally visible. In one such instance, 128 prefixes of length 32 were injected and withdrawn within half an hour. The operators involved in these incidents were surprised to hear that their announcements were being seen beyond their immediate upstream, and assured us that they would talk to their provider. Usually, this temporary route injection was done to fend off denial-of-service attacks or to load balance the incoming traffic between multiple links; when it was intended to be global, they were classified as non-misconfigurations.

Old configuration. A surprising cause for misconfiguration that

[5] A possible practice that causes this behavior when updating filters is to first remove the old filters, and then to rewrite the new filters using a script. Routes can be leaked before the new filters take effect.

Misconfigurations	Prefixes		Incidents		Type
	Total	Connectivity	Total	Connectivity	
Initialization bug	1580 (22%)	0 (00%)	43 (05%)	0 (00%)	mistake
Reliance on upstream filtering	977 (14%)	0 (00%)	431 (46%)	0 (00%)	mistake
Old configuration	72 (01%)	28 (39%)	36 (04%)	20 (56%)	mistake
Redistribution	2294 (32%)	1 (00%)	43 (05%)	1 (02%)	slip
Community	99 (01%)	2 (02%)	28 (03%)	2 (07%)	slip
Hijack	101 (01%)	101 (100%)	54 (06%)	54 (100%)	slip
Forgotten filter	53 (01%)	1 (02%)	13 (01%)	1 (08%)	slip
Incorrect summary	26 (00%)	0 (00%)	17 (02%)	0 (00%)	slip
Unknown configuration error	1053 (15%)	39 (04%)	90 (10%)	12 (13%)	slip
Miscellaneous	88 (01%)	16 (18%)	38 (04%)	10 (26%)	
Unclassified	779 (11%)	82 (11%)	152 (16%)	23 (15%)	
Non-misconfigurations	*Prefixes*		*Incidents*		
Failure		91 (34%)		50 (32%)	
Testing		66 (24%)		44 (28%)	
Migration		51 (19%)		26 (17%)	
Load balancing		22 (08%)		20 (13%)	
Miscellaneous		11 (04%)		7 (04%)	
Unclassified		30 (11%)		10 (06%)	

Table 6: Causes of origin misconfiguration. The *total* subcolumn lists the number and percentage of misconfigurations against each cause. The *connectivity* subcolumn lists the number and percentage of misconfigurations that led to connectivity disruption.

also led to connectivity problems came to light. In some instances of this problem, operators changed the configuration on their routers correctly, but did not commit changes to stable storage. This is a separate command on most routers, but unlike text editors, there is no warning about unsaved work. When the router rebooted the next time, the old configuration came back into effect. The router then started announcing old routes. In some other instances, operators plugged in a standby router without resetting its configuration from the previous use. In some other instances, operators were not able to ascertain how the short-lived prefix announced by their router was left in its configuration file (they had announced the prefix in the past), or why the router started announcing it suddenly.

Redistribution. Redistribution lets an operator specify which routes learned from other routing protocols, such as OSPF, should be advertised to BGP peers. There are multiple ways of achieving this; based on the responses we got back, it was not always possible to tell which configuration style or feature went wrong. We show two common mistakes below.

• *redistribute:*

```
  redistribute igrp 100 route-map igrp2bgp
```

This tells the router not to advertise everything in IGP tables that matches the route map *igrp2bgp*. If an operator fails to specify the route-map part or gets the route-map itself wrong, all the prefixes in the IGP tables would be announced via BGP.

• *aggregate-address:*

```
  aggregate-address 192.168.0.0 \
255.255.0.0 summary-only
```

This tells the router not to advertise any subsets of the prefix. But if *summary-only* is forgotten, all the more-specific prefixes in the routing table would be advertised.

If redistribution is not done correctly, it can lead to a large number of faulty prefix advertisements (as is evident in the large prefix to incident ratio), such as in the AS7007 incident [31]. It also exposes BGP to variations in the IGP protocol.

Communities. Attaching the wrong community attribute to prefixes was another major cause of origin misconfigurations. BGP communities are used by ASes to color their routes to express policies such as "don't propagate further," or "export only to your immediate peers" [5]. When incorrect communities attributes are attached to routes, prefixes get propagated beyond where they were intended, or worse, do not get propagated at all.

Hijacks. Occasionally, an unrelated AS announces address space owned by another AS.[6] Although the potential to do this is a major security flaw, the more common cause of this is a typo, when the misconfigured AS owns prefixes that are similar (small edit distance [8]) to the hijacked prefix. Extrapolating from the results in Table 4, we get roughly 310 prefix hijacks incidents, or roughly 15 incidents per day.[7]

Forgotten filter. This represents the instances in which the operators responded with "yes, this is a misconfiguration; I forgot to filter these routes." No further details were provided.

Incorrect summary. By applying an incorrect summary mask an AS can announce an address block that is larger or smaller than the intended block. For instance, 255.255.0.0 is the right prefix mask for 192.168.0.0/16, but using 255.255.255.0 yields 192.168.0.0/24, and using 255.0.0.0 yields 192.0.0.0/8. Fortunately, none of these events caused connectivity problems, sometimes as a result of the longest matching prefix rule. There were many incidents of small ASes announcing /8s (mostly, 6[1234].0.0.0/8) that have not been allocated to any organization as a single block.

Unknown configuration error. This refers to cases when the operators responded with "yes, this is a misconfiguration; I made a mistake while making some configuration changes," with no accompanying detail on what aspect of the configuration was wrong.

Most of the miscellaneous category includes configuration errors not already listed. It contains six interesting incidents involving routing registries. The operators configured their routers correctly,

[6] In one incident, a portion of the address space of a major router vendor was hijacked, apparently because they used their real netblocks in product documentation.

[7] Operators have also been known to inject announcements for rival operators' machines (/32s) to cause connectivity problems.

Misconfigurations	*Paths*	*Incidents*	*Type*
Prefix based config	98 (08%)	46 (22%)	mistake
Old configuration	20 (02%)	9 (04%)	mistake
Initialization bug	18 (01%)	9 (04%)	mistake
Bad ACL or route map	445 (34%)	8 (04%)	slip
Typo	153 (12%)	13 (06%)	slip
Forgotten filter	109 (08%)	15 (07%)	slip
Community	69 (05%)	37 (18%)	slip
Unknown config error	193 (15%)	14 (07%)	slip
Miscellaneous	22 (02%)	5 (02%)	
Unclassified	162 (13%)	54 (26%)	
Non-misconfigurations	*Paths*	*Incidents*	
Backup Arrangement	22 (47%)	13 (38%)	
Special Arrangement	13 (28%)	11 (32%)	
Failure	9 (19%)	7 (21%)	
Unclassified	3 (06%)	3 (09%)	

Table 7: Causes of export misconfiguration. Parentheses contain the percentage of misconfigurations due to that cause.

but made mistakes in registering their route objects with RIPE. Filtering based on RIPE data is becoming increasingly common in Europe, and their correct routes were rejected by many ASes due to the misconfigured registry data. The operators then withdrew their announcements, corrected their registry information, and re-announced the routes. These incidents highlight the dangers of having a misconfigured trusted source; problems can arise even when the operators get their router configuration right.

6.1.1 Non-Misconfigurations

Table 6 also shows the reasons for non-misconfigurations. The causes behind them are interesting as they lend insight into valid (or intentional) reasons for short-lived routes and how operators use the knobs provided by BGP.

Failures. Some ASes use configurations in which an announcement with a different origin or a more-specific prefix takes place during failures. The most common instance of this is multi-homing with backup for an organization that does not have its own AS number, and is thus advertised by its second provider's AS number for the duration of the failure. In some instances, even organizations with their own AS numbers were found to have backup arrangements by which their backup provider would advertise their address space on failures using a (normally less preferred) static route. A small fraction of self deaggregation incidents are also the result of failures; when a primary link fails, announcements for more-specifics go out through the (low capacity) secondary link to maintain connectivity for the important subnets within the organization. Sometimes, failures also revealed a previously hidden origin that was announcing the address space persistently, but none of our BGP peers were choosing the route offered by it.

Testing. The second leading cause of intentional short-lived routes that were not misconfigurations was testing. Operators often test their present or future configuration on the Internet by directly applying the planned configuration changes or simulating a link failure, checking the results with route servers before returning to the base configuration. In several instances, testing was done to discover the filtering properties of the upstream providers (instead of asking them).

Migration. When moving physically to a new location, to a new provider, or to a new address space, operators migrate their networks in a way that ensures connectivity at least to the important

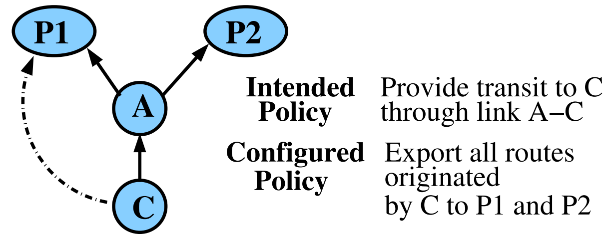

Figure 12: Prefix based configuration. The misconfiguration is uncovered when the link A-C fails.

hosts. As a result, there are windows of time during the migration phase when short-lived announcements are generated; for instance, when moving a subnet at a time.

Load balancing. Operators were also found to generate short-lived announcements in an attempt to load-balance their traffic between more than one link to the Internet. Often, this is done in an iterative fashion; by changing the size of the subset address space announced out via a particular link, observing the traffic ratios and so on. Also, when faced with a sudden shift in traffic patterns, BGP becomes a convenient tool to manage the incoming traffic.

The miscellaneous category mainly includes denial-of-service attack mitigation, and troubleshooting of routing problems.

6.2 Export Misconfiguration

We now discuss the causes behind export misconfiguration, which are listed in Table 7. The first three causes are mistakes, while the rest are slips. As before, the contribution of mistakes is significant, but less pronounced than that for origin misconfigurations. The causes that are common with those in Table 6 have already been explained in Section 6.1; we discuss the others below.

Prefix based configuration. This was found to be the biggest cause of export misconfiguration incidents. The problem is illustrated in Figure 12. $P1$ and $P2$ are providers of A, and C is a customer of A. C also has a path to $P1$ that does not go through A. Since A is the provider of C, it provides transit to C by exporting routes to both $P1$ and $P2$. This policy is expressed as "it is OK to export $C's$ prefixes to $P1$ and $P2$," which can be implemented by listing the prefixes explicitly or specifying that the origin AS should be C.

The above configuration works properly in absence of failures, since A always choses the direct path to C (routes from customers are preferred, so the path to C through $P1$ is not selected). Thus all the traffic that comes to A for C is sent via link A-C. Now assume that the link A-C fails. A would hear announcements for C's prefixes through $P1$. Based on its configuration it would announce this route to $P2$, thus becoming a transit between $P1$ and $P2$ for all traffic for C. In most cases this is an unexpected violation of policy (in some cases there is an intended arrangement between ASes to provide backup paths; such instances were not classified as misconfigurations); the intended policy being to provide transit to C's traffic when the incoming or outgoing link is A-C, but not otherwise. The misconfiguration can be avoided by configuring a different policy: export routes for C only when the AS-path is C (AS-path based filtering). Corrections to filters were made by the network operators when we told them about this error.

The problem with prefix based configuration holds even when either one or both of $P1$ and $P2$ are peers of A, though ISPs usually do a better job of filtering routes received from their peers.

Bad ACL or route map. This refers to instances in which operators said they made a mistake while programming route maps or access control lists (ACL) in their routers.

Table 7 also shows the reasons why some potential export misconfigurations were not misconfigurations. It lends insight into instances in which the relationship inference algorithm fails. Sometimes a *backup arrangement* exists between two or more ASes, by which they provide connectivity to each other in face of failures. Effectively, the relationship between them changes during failures, and we classify it as a probable misconfiguration based on the relationship during normal operation. *Special arrangement* refers to instances where the relationship between ASes are more complex than those listed in Table 2. The most common instance of this occurs when AS-X is a customer of AS-Y for all traffic but provides transit to AS-Y for a certain set of prefixes. The inference algorithm assumes that ASes have the same relationship for all prefixes, and hence flags these instances as probable misconfigurations. Sometimes, *failures* reveal new edges between ASes; the inference algorithm can conclude the wrong relationship between such ASes.

7. DISCUSSION

In his study of computer system reliability in 1985 [15], Jim Gray notes:

The top priority for improving system availability is to reduce administrative mistakes by making self-configured systems with minimal maintenance and minimal operator interaction. Interfaces that ask the operator for information or ask him to perform some function must be simple, consistent and operator fault-tolerant.

The misconfigurations we have uncovered show that BGP router configuration is far from meeting this goal. In this section, we discuss how we might progress toward it, with suggestions that run the gamut from straightforward human factors design to implications for router and protocol design. Unfortunately, there is no magic bullet but rather a set of tasks that we believe can significantly reduce the Internet's vulnerability to accidental errors.

7.1 User Interface Design

There is a wealth of literature [27, 32, 33] that deals with designing the user interfaces of systems to minimize human errors. Relevant principles include: safe defaults; the more serious the consequences of the error, the less likely it should be that an operator will make it; consistency across versions; large edit distances between correct and incorrect settings; and minimization of dependence between multiple lines of configuration.

It is clear from the causes of misconfigurations that these principles have not been followed in the design of the router CLI (command line interfaces) of major vendors, and yet they could reduce errors. Applied systematically, we estimate by inspection that this could prevent around 10% of the incidents. For example, dangerous features such as redistribution could be changed so that operators explicitly list the prefixes to be announced through BGP rather than of leaving the door open to IGP variations. Indeed, the operator community is aware that some features are more error-prone than others and best practice documents do discourage their use [17]. However, this is clearly not sufficient to prevent misconfigurations; they will continue to occur until the router CLI is re-designed.

We were surprised to learn that a poor understanding of router command semantics (mistakes) were responsible for a large percentage of misconfigurations. Instead, we had expected improper execution of a sound plan (slips) to dominate. This suggests that the available configuration features are a poor fit to the tasks at hand, and that more apt features or semantics could reduce misconfigurations. For example, transactional semantics for configuration changes – allowing a router to apply all changes or none – would eliminate around 20% of the origin misconfiguration incidents.

7.2 High-Level Languages and Checking

It is apparent that router configuration is a low-level detailed task and therefore error-prone. Configuration tools that would let operators express policy directly in a high-level form (from which lower-level configuration can be generated) would potentially reduce many errors. For instance, the problem with prefix-based configuration, responsible for 22% of the export misconfiguration incidents, is caused by a simple high-level policy that is not obvious for operators to express at the CLI level.

We are not the first to suggest the use of high-level configuration specifications. For example, RPSL [1] is a high-level routing policy specification for use with the Internet Routing Registries (IRRs), and various network management systems (NMS) include high-level policy tools. However, these tools have generally not caught on with ISP operators. We asked operators why and discovered a variety of reasons, both technical and non-technical. First, the multiplicity of router vendors and versions makes any NMS non-universal; this weakens the case for using it as it must be used in addition to other tools. NMSs are of limited use unless they are flexible enough to support the full range of ISP policy, and of course vendor “lock-in” is a concern in the industry. Finally, the features provided by NMSs tend to lag new router features.

In light of these reasons, we observe that it seems compelling that any high-level interface be supported directly by the router vendors. A more viable short term approach may be to target configuration checking, rather than high-level configuration. A checker allows operators free use of low-level constructs. It looks for consistency within and between configurations, and warns the operators when a safety property has been violated, for example when a required routemap remains undefined. It may also prove effective to annotate configuration with optional, high-level, declarative expressions of the intended policy, as this would provide greater scope for consistency checking. One configuration checker has already been proposed for intra-domain traffic engineering in [13], and the authors describe it as an effective tool. Checking self-consistency has already proved to be highly effective for systems code [11], where high-level safe languages are often eschewed in favor of more flexible, low-level constructs.

7.3 Database Consistency and Registries

Many misconfigurations were the result of incorrect or inconsistent data being used for active configuration, due to typos and outdated router configuration. Part of the difficulty underlying these misconfigurations is database consistency. There are at least three different databases involved in configuration at various stages. First, each router contains its own version of configuration information. Second, the ISP maintains configuration-related data in its NMS. Third, allocations and policies may be maintained in industry-wide IRRs. If these databases are not derived from one another then inconsistency will lead to errors. One interesting kind of error we observed was correct filtering based on incorrect registry data. That is, if the ISP database and registry database become inconsistent, then the ISP's routers can be configured correctly but still suffer from misconfiguration as other ISP's routers filter their announcements based on incorrect registry entries.

Given these kind of errors, it seems compelling that consistency mechanisms be built directly into routers, e.g., the router update the registry directly, or at least check against it for consistency. Unfortunately, there are no such checks today, and the information in

	Total Prefixes	Registered Origins	Consistent Origin(s)	Inconsistent Origin(s)
Single Origin AS	115228	101952	70458 (69%)	31494 (31%)
Multiple Origin ASes	1720	1523	293 (19%)	1230 (81%)

Table 8: Effectiveness of the IRRs in detecting misconfigured origins.

the registries is widely believed to be inaccurate.[8] We looked at the correspondence between registries and actual announcements while we were trying to determine how to best identify misconfigurations, and indeed, frequently there is a mismatch. Table 8 shows the fraction of correct entries present in the IRRs for a BGP snapshot taken on 28 September, 2001. A prefix is considered to have consistent origin(s) if all the ASes it is originated by are registered as origin ASes in at least one (i.e., a best case scenario) of the regional registries (ARIN, APNIC, RIPE) or the RADB. The table shows that less than 70% of the single origin AS prefixes are advertised by their registered origin ASes, and more than 80% of the multiple origin ASes have at least one unregistered origin AS.

7.4 Protocol Extensions

Our work can also be used to help evaluate BGP protocol changes that aim to reduce misconfigurations. For example, S-BGP is one proposed extension to BGP in which routing announcements are authorized and authenticated as they travel along the AS-path to prevent improper announcements [22], whether accidental or malicious. We can readily determine from our data which kinds of misconfiguration S-BGP would prevent if it were deployed (assuming the registries on which it depends were up to date) and which it would not. S-BGP would prevent related and foreign origin misconfigurations, including hijacks, but would not prevent self-deaggregation or export errors. The latter comprise roughly half of the incidents and the majority of the incorrect prefix routes.

A different set of protocol changes would provide safe visibility to misconfigurations so that they get fixed. A problem with the way filtering is currently implemented is that it fails silently. A router silently drops bad announcements, and no information is returned to the source of misconfiguration. Indeed, we came across instances where an ISP's customer had a long-running misconfiguration but we were able to see it only when the ISP took off its filters while troubleshooting an unrelated problem. If instead of quietly dropping all buggy announcements, the ISP had informed its customer of the misconfiguration, it would have been fixed much sooner. This problem is analogous to the one observed in [39], where hosts and routers silently dropped packets with a bad checksum, which left no information for finding broken routers.

Currently, there is no systematic way in which misconfigurations are detected. In most instances, they are detected because of connectivity problems, BGP table blow-ups, some other operator observing the faulty announcement, or daily audits. The duration of the misconfigurations can be significantly reduced if there were tools that would verify the dynamic state of the system. A possible way of doing this is using the thousands of public looking glasses (some of which we use in our connectivity test). Currently, these looking glasses are used by humans to manually debug routing problems if they persist long enough for someone to notice, but there is no reason why a similar facility could not be used automatically by routers to check their advertisements.

[8]RIPE has recently started a routing registry consistency project which aims to improve the consistency of the registry information [36].

8. RELATED WORK

There have been numerous other studies of faults in computer systems. Notable among these are Gray's studies on failures with Tandem's computer system [15, 16], in which he discovered that the biggest causes of outages were software bugs (62%) followed by operations (15%). A decade ago, Danzig et al. analysed DNS traces and discovered that a large fraction of traffic was due to bugs and poor implementation choices [10], and a more recent study confirms that implementation bugs and configuration mistakes are still responsible for a significant fraction of the traffic [4]. Labovitz et al. studied wide area backbone failures and, among other results, concluded that misconfigurations could be responsible for 12% of the incidents [25]. Our work is most similar to the last study, but focuses specifically on the role of misconfigurations in BGP, and provides a more detailed treatment of this area.

There has also been much recent work that considers the various instabilities in BGP, beginning with the study by Labovitz et al. that analysed excessive churn due to implementation deficiencies [26]. In a continuation of that work they showed that convergence properties of path vector protocols such as BGP are much worse than previously assumed [24]. It has also been shown that independent and uncoordinated policies in BGP can lead to persistent route oscillations [18, 40]. Most recently, the impact of Internet worms on BGP stability has been studied [9]. Our study complements all of this work by analyzing the impact of another source of instability in inter-domain routing: the misconfiguration of BGP speaking routers.

Our study is preceded by various efforts that regularly provide statistics to monitor BGP instability, the presence of private identifiers [35] in the global routing table, and the growth behavior of BGP tables [2, 19, 20]. These projects are outgrowths of the operations community and have proved very useful at both helping to quantify problems and identifying the culprits (which may act as a deterrent to negligent configuration). Our work identifies different classes of misconfigurations and provides significantly more information on their occurrence, impact and causes. In other recent work, Zhao et al. have analyzed prefixes that are announced by multiple origin ASes (MOAS) [41]. They identify BGP misconfiguration as one of the potential causes along with multi-homing and failures. Our work is focused exclusively on misconfigurations and includes the MOAS incidents due to misconfigurations.

Finally, to put our work in perspective, human errors have been studied by researchers in many different domains [27, 32, 33, 37]. Our study is a first step towards identifying the causes of errors in BGP, and in turn that is the first step towards fixing the problems that could cause both small and large-scale disruptions in the future.

9. CONCLUSIONS

We have presented the first systematic study of BGP configuration errors that propagate across the backbone of the Internet. Our study focused on two kinds of misconfigurations: the accidental insertion of routes into the global BGP tables (origin misconfiguration); and the accidental propagation of routes that should have been filtered (export misconfiguration). As input, we analyzed the BGP updates taken from RouteViews, which has views from 23 different vantage points across a diverse set of ISPs, over a period of 21 days. We showed how simple heuristics can be used to find misconfigurations in the stream of BGP updates. While our purpose was to study these incidents, we note that our heuristics could be used by other researchers to factor out misconfigurations from their experiments.

We were surprised by the extent of the misconfiguration that we observed. We found that 200-1200 prefixes, equivalent to 0.2-1% of the global table size, suffer from misconfiguration each day. To put this in perspective, close to 3 in 4 of the new route announcements per day are the result of misconfiguration. This is perhaps even more surprising in light of the fact that our results underestimate of the actual level of misconfiguration, probably significantly, because our methodology is conservative and considers only certain types of mistakes.

We also analyzed the impact of misconfigurations on Internet connectivity by actively probing paths that we suspected were faulty. We found that connectivity was surprisingly robust to most misconfigurations. It was affected in only 4% of the misconfigured announcements or 13% of the misconfiguration incidents. The routing load due to misconfigurations, however, was not insignificant. 2% of the time, it was more than 10% of the total update load, and it went higher than 60% of the total update load on at least on one occasion, even with 15 minute averaging.

To validate our results, we used email to survey the operators involved in the incidents. We further used the goodwill we built with operators (by alerting them of previously unknown problems) to compile a list of the causes of misconfiguration. We found these causes to be diverse, and not limited to human slips as we expected at the beginning of this study. Configuration features such as redistribution that control the fate of a large number of prefixes were the largest contributors to misconfigured announcements. We also uncovered a potential bug in the software of a major router vendor. For export misconfigurations, we found a common practice that led to undesired behavior during failures.

Our ultimate goal is to understand how human errors can be minimized in large distributed systems. We have argued for some simple changes in router and protocol design that would eliminate the potential for some of the errors we observed, reduce the likelihood of others, and minimize the impact when they did occur. These include high-level policy specification as part and parcel of routers, automated verification of configuration, and transactional semantics for configuration commands. We have become convinced that there is much that can be done to improve the operational reliability of Internet routing, and we see this paper as a step in that direction.

10. ACKNOWLEDGMENTS

This study would not have been possible without the cooperation of the operators who took time to respond to us. We thank David Meyer for RouteViews, and Lixin Gao for scripts from [14]. We also used public traceroute servers and data collected by CAIDA's Skitter project [38]; our thanks to the people who manage these resources.

Ben Black and Brad Volz helped us to understand the operational aspects of inter-domain routing. We also acknowledge the kind support of the nice folks who run the UW-CSE and UW campus networks, and their understanding towards the occasional complaints that our experiments generated.

This work was supported by DARPA under grant no. F30602-00-2-0565.

11. REFERENCES

[1] C. Alaettinoglu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T. Bates, D. Karrenberg, and M. Terpstra. Routing Policy Specification Language (RPSL). RFC 2622, IETF, June 1999.

[2] T. Bates. The CIDR Report. `http://www.employees.org/~tbates/cidr-report.html`.

[3] A. Brown and D. A. Patterson. Embracing Failure: A Case for Recovery-Oriented Computing (ROC). In *High Performance Transaction Processing Symposium*, Oct. 2001.

[4] N. Brownlee, kc Claffy, and E. Nemeth. DNS Measurements at a Root Server. In *Globecom*, Nov. 2001.

[5] E. Chen and T. Bates. An Application of the BGP Community Attribute in Multi-home Routing. RFC 1998, IETF, Aug. 1996.

[6] J. M. Christensen and J. M. Howard. Field Experience in Maintenance. In *NATO Symposium on Human Detection and Diagnosis of System Failures*, 1981.

[7] Cisco Documentation. `http://www.cisco.com/univercd/cc/td/doc/cisintwk/ics/icsbgp4.htm`.

[8] T. Cormen, C. Leiserson, and R. Rivest. *Introduction to Algorithms*. McGraw Hill, 1993.

[9] 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`.

[10] P. B. Danzig, K. Obraczka, and A. Kumar. An Analysis of Wide-area Name Server Traffic: A Study of the Domain Name System. In *ACM SIGCOMM*, Sep. 2000.

[11] D. Engler, D. Y. Chen, S. Hallem, A. Chou, and B. Chelf. Bugs as Deviant Behavior: A General Approach to Inferring Errors in Systems Code . In *SOSP*, Oct. 2001.

[12] J. Farrar. C&W Routing Instability. NANOG mail archives. `http://www.merit.edu/mail.archives/nanog/2001-04/msg00209.html`.

[13] A. Feldmann and J. Rexford. IP Network Configuration for Intradomain Traffic Engineering. *IEEE Network Magazine*, Sep. 2001.

[14] L. Gao. On Inferring Autonomous System Relationships in the Internet. In *IEEE Global Internet Symposium*, Nov. 2000.

[15] J. Gray. Why Do Computers Stop and What Can Be Done About It? Technical Report 85.7, Tandem Computers, June 1985.

[16] J. Gray. A Census of Tandem System Availability Between 1985 and 1990. Technical Report 90.1, Tandem Computers, Jan. 1990.

[17] B. Greene and P. Smith. Essential IOS Features Every ISP Should Consider (v 2.9), June 2001.

[18] T. Griffin and G. T. Wilfong. An Analysis of BGP Convergence Properties. In *ACM SIGCOMM*, pages 277–288, Aug. 1999.

[19] G. Huston. BGP Table Statistics. `http://www.telstra.net/ops/bgp/index.html`.

[20] IPMA Project. `http://www.merit.edu/ipma/`.

[21] B. H. Kantowitz and R. D. Sorkin. *Human Factors: Understanding People-System Relationships*. Wiley, 1983.

[22] S. Kent, C. Lynn, and K. Seo. Secure Border Gateway Protocol (Secure-BGP). *IEEE JSAC*, 18(4), Apr. 2000.

[23] T. Kernen. Traceroute.org. `http://www.traceroute.org/`.

[24] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet Routing Convergence. In *ACM SIGCOMM*, Sep. 2000.

[25] C. Labovitz, A. Ahuja, and F. Jahanian. Experimental Study of Internet Stability and Wide-Area Network Failures. In *Fault-Tolerant Computing Symposium (FTCS)*, June 1999.

[26] C. Labovitz, G. R. Malan, and F. Jahanian. Origins of Internet Routing Instability. In *IEEE INFOCOM*, June 1999.

[27] N. G. Leveson. *Safeware: System Safety and Computers*. Addison-Wesley, 1995.

[28] O. Maennel and A. Feldman. Realistic BGP traffic for test labs. In *ACM SIGCOMM*, Aug. 2002.

[29] C. D. Marsan. Faster 'Net growth rate raises fears about routers. `http://www.nwfusion.com/news/2001/0402routing.html`, Apr. 2001.

[30] D. Meyer. RouteViews Project. `http://www.routeviews.org`.

[31] S. A. Misel. Wow, AS7007! NANOG mail archives. `http://www.merit.edu/mail.archives/nanog/1997-04/msg00340.html`.

[32] D. A. Norman. Design Rules Based on Analyses of Human Error. *Communications of the ACM*, 1983.

[33] J. Reason. *Human Error*. Cambridge University Press, 1990.

[34] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). RFC 1771, IETF, Mar. 1995.

[35] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. Address Allocation for Private Internets. RFC 1918, IETF, February 1996.

[36] Routing registry consistency check. `http://www.ripe.net/ripe/docs/rr-consistencycheck.html`, Dec. 2001.

[37] J. Senders and N. Moray. *Human Error: Cause, Prediction and Reduction*. LEA Publishers, 1991.

[38] Skitter project. `http://www.caida.org/tools/measurement/skitter/`.

[39] J. Stone and C. Partridge. When the Checksum and the Data Disagree. In *ACM SIGCOMM*, Aug. 2000.

[40] K. Varadhan, R. Govindan, and D. Estrin. Persistent Route Oscillations in Inter-Domain Routing. *Computer Networks*, 32(1), 1999.

[41] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. An Analysis of BGP Multiple Origin AS (MOAS) Conflicts. In *ACM SIGCOMM Internet Measurement Workshop*, Nov. 2001.