The Temporal and Topological Characteristics of BGP Path Changes

Di-Fa Chang Ramesh Govindan John Heidemann
USC/Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, CA 90292, USA

Abstract

BGP has been deployed in Internet for more than a decade. However, the events that cause BGP topological changes are not well understood. Although large traces of routing updates seen in BGP operation are collected by RIPE RIS and University of Oregon RouteViews, previous work examines this data set as individual routing updates. This paper describes methods that group routing updates into events. *Since one event (a policy change or peering failure) results in many update messages, we cluster updates both temporally and topologically (based on the path vector information). We propose a new approach to analyzing the update traces, classifying the topological impact of routing events, and approximating the distance to the the Autonomous System originating the event. Our analysis provides some insight into routing behavior: First, at least 45% path changes are caused by events on transit peerings. Second, a significant number (23–37%) of path changes are transient, in that routing updates indicate temporary path changes, but they ultimately converge on a path that is identical from the previously stable path. These observations suggest that a content provider cannot guarantee end-to-end routing stability based solely on its relationship with its immediate ISP, and that better detection of transient changes may improve routing stability.*

1. Introduction

BGP [17] is a policy-based path-vector routing protocol deployed in Internet for inter-domain routing. The Internet is divided into tens of thousands of autonomous routing domains, of which over 15 thousand are currently associated with Autonomous System Numbers (ASNs) for the purpose of interdomain routing. BGP routers in each AS transmit routing messages to other BGP routers in the same AS and other ASes through internal and external BGP connections, respectively. Routing messages containing reachability information are called BGP updates. To facilitate the study on the operational use of BGP, there are public BGP routing message collection sites such as RIPE's RRCs [1] (Remote Route Collectors) and Oregon University's RouteViews [2] that collect BGP updates and routing tables from tens of BGP routers located in various ASes. These data sets provide researchers and operators a local perspective on the visible Internet BGP routing status. Researchers have been using the collected routing tables and routing messages to study the Internet topology at AS-level [19, 20, 6], monitor the Internet growth [8], examine the inter-domain routing stability [18], investigate BGP router misconfiguration [16, 21], and derive the model for BGP traffic [15]. In this paper we present a systematic approach to decompose the stream of BGP updates into small sequences of path advertisements with the purpose of distinguishing the *routing events* that cause the BGP routing changes. The goal of this study is to answer the following questions:

- When a BGP router observes a path (AS_PATH) change to some prefix, what do we know about which AS peering causes this path change?

- How many other path changes are caused by the same AS peering?

Labovitz *et al* [13, 14, 12] presented the first large-scale analysis of BGP dynamics. They investigated the pathological routing updates that are duplicated announcements and withdrawals due to sub-optimal implementations of routers; their primary focus was on the route convergence times. By contrast, our paper looks at the frequency and extent of route changes, and examines correlated route changes caused by events within the network. Feamster *et al* [7] [4] investigate the Internet path faults at router-level by monitoring the paths between 31 hosts and triggering traceroutes when paths become unavailable. They find that failures are more likely to appear within an AS than between ASes, and failures appear closer to the network core are more likely to coincide with the occurrences of BGP updates than failures near end hosts. Unlike that work, we analyze not only path failures but all routing events that cause AS-level topological changes by examining route changes on half million paths. Like their work, however, we are also interested in *where* routing events occur within the network.

In this paper, we present several interesting results:

- Most path advertisements (93%) in a stream of BGP updates have a topologically relation with a few others. To our knowledge, this is the first quantification on the topological relation among the path advertisements in a BGP update stream. It can help design a more realistic BGP traffic model for lab use in addition to those metrics considered in [15].
- At least 45% of path changes are caused by routing events in transit peerings, where transit peerings are the peerings that transit traffic to the destinations not inside the peering ASes.
- A significant number of path changes (23–37%) are transient which means that BGP AS_PATHs before and after the change are the same. This type of change is associated with 35–52% of path advertisements. Since BGP's main task is to hide information, we currently do not have the models nor the measurement infrastructure sufficient to determine the root causes.

2. Methodology

Fundamental to our analysis is the notion of a BGP *path change*. In this section, we first describe a notation for BGP path changes. Then, we carefully deconstruct the various kinds of path changes that can happen in BGP Finally, we describe algorithms that *cluster* path changes into "events," whose statistics we later analyze.

2.1. Notation for BGP Paths

A BGP path is a sequence of autonomous systems that packets travel to reach a range of IP addresses (called a prefix). It differs from an IP route which is a sequence of routers through which packets are forwarded from source to destination. Suppose there are V vantage points $v_1, v_2, ..., v_V$ which are BGP routers. The path for the vantage point v_i to reach the prefix f is denoted by $path(v_i, f) = (a_1, \ldots, a_m)$ where $m \geq 0$, v_i is located in AS a_1, a_m is called the origin AS, and $a_2, \ldots, a_{m-1}$ are intermediate ASes. The length of $path(v_i, f)$ is $m - 1$. The case of $m = 0$, *i.e.*, $path(v_i, f) = \emptyset$, represents that v_i has no path to reach the prefix f. The case of $m = 1$, *i.e.*, $path(v_i, f) = (a_1)$, means that the prefix is originated by the AS containing the vantage point.

Two mechanisms on path advertising make the structure of BGP paths different from IP routes. First, BGP4 adopts a path aggregation mechanism that allows an element of a path to be a set of ASes, called AS_SET. That is, a_j, $1 < j \leq m$, may denote a set of ASes. For example, the path (2914 3549 19548 [1239 3356 7843 19094]) means that a_4 can be *one* of AS1239, AS3356, AS7843, and AS19094. However, we observed that only a very small number (0.02%) of prefixes in Internet have paths including AS_SET. A more commonly adopted mechanism that affects AS_PATH attribute is AS prepending [17]. For example, the path (1103 3549 701 7474 7474 7474 7476 7570) means that AS7474 announces this path by prepending its AS number three times. AS prepending is used to increase the path length for traffic engineering purpose, since a path with longer length will be less likely selected by other ASes. As a consequence, it is possible that $a_j = a_{j+1} = \ldots = a_k$ for $1 < j < k \leq m$. We observe that the number $(k - j)$ of AS prepending varies from 1 to 14 in our data set. Our clustering algorithms can deal with the AS_PATH affected by these mechanisms.

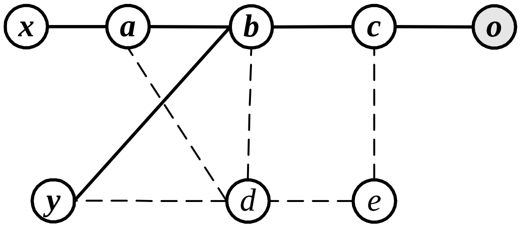

Figure 1. Example of AS topology. Two vantage points v_1 and v_2, in ASes x and y respectively, observe an event that changes their paths from old paths (solid lines) to new paths (dashed lines).

2.2. Defining BGP Path Changes

Before we can evaluate routing events, we need to carefully define what constitutes a BGP path change and how path changes relate to a routing event.

We say a *path change* is observed by vantage point v_i for prefix f if v_i has a new path $path(v_i, f) = (b_1, b_2, \ldots, b_n) \neq (a_1, \ldots, a_m)$, where $b_1 = a_1$. Denote the old path by $path_o$ and new path by $path_n$. For example, in the AS topology shown in Figure 1, the paths for vantage points v_1 and v_2 change from (x, a, b, c, o) to (x, a, d, e, c, o) and from (y, b, c, o) to (y, d, e, c, o), respectively. If the length of the new path is greater than that of old path, *i.e.*, $|path_n| > |path_o|$, we call the path change as a *long* path change. Similarly, if $|path_n| < |path_o|$ or $|path_n| = |path_o|$, we call it as a *short* or *equal* path change, respectively.

Because BGP is not a pure shortest-path-based routing protocol, there are *several possible events* (failures, policy changes) that could result in a path change. In the scenario of Figure 1 which consists of two *long* path changes, consider the peering link (b, c). The routing events that can cause this peering unable to carry the traffic destined to the prefix f include:

- Peering failures: such as a link failure, an exchange point failure, or a BGP session reset.

- Policy changes at b: AS b may add an input filter or decrease the local_pref for prefix f on this peering. As a consequence, b may use the path (b, d, e, c, o) or has no path to reach prefix f. This causes a to select the path (a, d, e, c, o) and y to use (y, d, e, c, o).
- Policy changes at c: AS c may add an output filter or increase the number of AS prepending for prefix f on this peering. The result is the same as the above.

In this case, the AS, called *event originator*, that makes (b, c) unavailable could be either b, c, or both. One of the challenges we face in this work is to estimate the location of event originators. Different originators can have different observed effects; in the extended version of this paper [5] we enumerate all the kinds of routing events that can occur in the example of Figure 1. As it turns out (and we discuss this in detail later), event originators can be identified by examining the difference between $path_o$ and $path_n$ (note that the event generators may not appear in $path_n$, *i.e.*, $b \notin (x, a, d, e, c, o)$). If we denote by $peerings_o$ the set of peerings in $path_o$ and by $peering_n$ the set of peerings in $path_n$, then the peerings where the routing events may take place include $(peerings_o \cup peering_n) - (peerings_o \cap peering_n)$ and some hidden peerings (*e.g.*, (b, d)).

In the above description, $path_o$ and $path_n$ refer to *converged paths*. When a path change happens, BGP updates are transmitted between routers to reflect the current routing status. Before this BGP update process has converged, vantage points may have some *transient paths*. In the trace of BGP updates collected from various vantage points, both converged and transient paths are included. A purpose of this study is to estimate the number of path changes and routing events by analyzing the trace of BGP updates. A routing event may result in many path changes, and each path change may trigger various amount of BGP updates. A simple formula can illustrate the problem: If there are E routing events occurred during the period T, and each event affects P prefixes, and, for each prefix change, there are V vantage points observing it, and each vantage point sends M BGP updates to the routing message collection site, then, in worst case, the site will receive $Q = O(E \times P \times V \times M)$ updates. What we can obtain from the trace of BGP updates is Q, while what we want to estimate is E.

2.3. Data Clustering

Clustering is a fundamental operation in data mining. There have been many clustering algorithms developed in last forty years [10, 9, 3]. It has been applied to many research fields like pattern recognition, image processing, information retrieval, DNA analysis, *etc*. There are two major approaches. Hierarchical clustering starts with each object in its own cluster, and continues to agglomerate the closest pair of clusters at each stage until all of the objects is in one cluster. At each stage, the algorithm groups the clusters produced at previous stage into a new set of clusters. The result is a nested series of partitions, in which the clusters in a partition are tighter than those in later partitions. Users can select the desired partition where the tightness of the clusters is under the desired threshold. On the other hand, partitional clustering starts with a pre-specified number of clusters and initial positions for the cluster centers, and continues to combine each object into closest cluster and update the position of the cluster until the overall clustering error is under a specified threshold.

Here we defines some terms and notation used to describe our clustering method.

Pattern: A pattern (or object) $\mathbf{x}$ is a single data item to be clustered. In this study, a BGP path advertisement is a pattern.

Feature: A pattern consists of a vector of d measurements, *i.e.*, $\mathbf{x} = (x_1, \cdots, x_d)$, where x_i are called features and d is the *dimensionality*. Example features in a BGP path advertisement include prefix, origin AS, AS_PATH, timestamp, the router ID of the router that transmitted it.

Pattern Set: A pattern set is the set of patterns to be clustered and is denoted by $\mathbf{X} = \{\mathbf{x_1}, \cdots, \mathbf{x_n}\}$. For example, a stream of BGP updates is a pattern set.

Cluster: A cluster $C_i, 1 \leq i \leq k$, is a subset of the pattern set such that the patterns in a cluster are more *similar* to each other than patterns in different clusters.

Similarity: A measure of similarity between two patterns is calculated based on their features. The definition of similarity plays an essential role in the interpretation of the generated clusters. For example, the similarity defined by computing the difference of the origin ASes of two updates will result in clusters that consists of updates propagated from the same origin AS.

Since the partitional approach requires a pre-specified number of clusters which has no justifiable approximation, we take the hierarchical approach. Two hierarchical clustering algorithms are employed in this study: agglomerative single-link and complete-link clustering algorithms. The original algorithms require to process the pattern set many passes until all patterns are in one cluster, hence consume enormous computation power and storage. We modify the algorithms such that they terminates as soon as the desired partitions are generated.

The algorithms can be found in [9, 10], and are stated below for completeness.

Agglomerative Single-Link Clustering Algorithm: In this method, two clusters can be merged into one cluster if there exists a pair of patterns in the two clusters having a similarity measure above some threshold.

It has a tendency to generate clusters that are straggly or elongated.

1. Place each pattern in its own cluster. Calculate the measure of similarity between any two patterns and sort the list of similarity measures in ascending order.
2. Step through the sorted list of similarity measures, forming a graph where pairs of patterns closer than a pre-specified threshold of similarity are connected by a graph edge.
3. Each maximally connected subgraph forms a cluster.

Agglomerative Complete-Link Clustering Algorithm: In this method, two clusters can be merged into one cluster if *all* pairwise similarity measures between patterns in the two clusters are above the threshold. It produces more tightly bound or compact clusters than the single-link method.

1. Place each pattern in its own cluster. Calculate the measure of similarity between any two patterns and sort the list of similarity measures in ascending order.
2. Step through the sorted list of similarity measures, forming a graph where pairs of patterns closer than a pre-specified threshold of similarity are connected by a graph edge.
3. Each maximally *completely* connected subgraph forms a cluster.

2.4. Identifying Events from Routing Updates

In this section, we describe the methods to identify routing events in a stream of BGP updates. The method of identifying events is to decompose the stream into small sequences of path advertisements, *i.e.*, *clusters*. A BGP update consists of path advertisements for various prefixes. The path advertisement can be an announcement of new path, a withdrawal of the current path, or an update that changes attributes of the current path such as MED, LOCAL_PREF, *etc*. We are only interested in those path advertisements that changes path, since they represent a topological change in the inter-domain routing system. For simplicity, we will use the term “path advertisement” instead of “path advertisement that changes path.”

Routing events that result in path changes can be peering failures, peering repairs, peering resets (*i.e.*, a failure followed by a repair), policy changes, route oscillation, misconfigurations, *etc*. Unlike [16], which investigates two types of misconfigurations, namely, origin misconfigurations and export misconfigurations, we are interested in any event that causes path changes. Not all events in the BGP routing system are visible to each AS of Internet. Our goal is to identify the events visible to a set of vantage points across the Internet. We say a routing event is visible to a vantage point if the vantage point receives and sends path advertisements that reflect the path changes caused by the event. Thus, in a stream of path advertisements from all vantage points, we want to cluster those path advertisements triggered by the same routing event. This requires to do clustering in the three dimensions of P, V, and M as described in Sec. 2.2. This paper presents two types of clusters that provide upper bounds on the number of routing events:

Prefix-based cluster is a sequence of path advertisements for the same prefix that are sent by the same vantage point and closely spaced in time. It is meant to represent a *path change* to some set of IP addresses resulting from one event. This type of clusters represents the smallest sequence of path advertisements in a stream of BGP updates that are logically related.

Peering-based cluster is a sequence of prefix-based clusters that are closely spaced in time and contain a common set of peerings where a routing event was likely to take place. It is meant to represent a *path change* caused by a routing event occurring in a peering.

The prefix-based clusters are generated by the agglomerative single-link clustering algorithm, while the peering-based clusters are produced by the agglomerative complete-link clustering algorithm. The reason of using different algorithms will be discussed shortly. We first describe the similarity functions $s(p_i, p_j)$ that measures the similarity of two patterns p_i and p_j. In prefix-based clustering, a pattern is a path advertisement denoted by p = ($prefix$, $time$, vp, $path$), where $prefix$ is a tuple $(ip\_address, prefix\_length)$, $time$ is the time when the path advertisement was received by routing message collection site, vp is the vantage point that sends this path advertisement, and $path$ is the AS_PATH attribute that represents $path(vp, prefix)$ at the time $time$. The similarity function is defined as:

$$s_1(p_i, p_j) = \begin{cases} -1, & p_i.prefix \neq p_j.prefix \\ -1, & p_i.vp \neq p_j.vp \\ T_1 - |p_i.time - p_j.time|, & \text{otherwise} \end{cases}$$

T_1 is a parameter of the similarity function. When applying this similarity function to the single-link clustering algorithm, we set the threshold $s_1(p_i, p_j) \geq 0$ such that any pattern has a non-negative similarity measure with at least one pattern in the same cluster, while patterns in different clusters have negative similarity measure. In other words, path advertisements within a cluster are sent by the same vantage point for the same prefix and each of them has the receiving time within T_1 seconds of the receiving time of at least one path advertisement in the same cluster. The rationale behind this type of clustering is described as follows.

When a routing event happens, the BGP routing system can take minutes to converge to a stable routing state. During the converging period, many path advertisements are transmitted due to transient path changes. Since these path advertisements are all resulted from a single routing event, they should be grouped into a single cluster. *The parameter T_1 represents our assumption that no two routing events occur within T_1 seconds and cause path changes of the same prefix.* However, the routing convergence time may be longer than T_1. In this case, it is unable to determine whether two path advertisements that are for the same prefix, transmitted by the same vantage point, and within T_1 seconds are caused by different routing events. Thus, we just make the assumption that these two path advertisements are caused by the same routing event. This makes the single-link algorithm the appropriate one for prefix-based clustering.

After the clustering finished, we assign each cluster the path-change type and identify the possible peerings where the routing event may occur. The possible peerings are the peerings that are not shared by $path_o$ and $path_n$. The path-change type is one of *long*, *short*, and *equal*, and is determined by comparing the lengths of old path $path_o$ and the new path $path_n$ as described in Sec. 2.2. Several things are worth noting:

- $path_n$ is the path of the last path advertisement in the cluster, while $path_o$ is the path of the path advertisement *before* the first path advertisement in the cluster.
- If a path advertisement has an empty path, *i.e.*, it's a path withdrawal, then we say its path length is infinity. Thus, if $path_n = \emptyset$, then the cluster has a *long* path-change type. Conversely, if $path_o = \emptyset$, then the cluster has a *short* path-change type.
- Sometimes, $path_n$ is the same as $path_o$. This is because two routing events (*e.g.*, a peering failure followed by a peering repair, or a short BGP session reset) occur close in time such that the path advertisements they triggered are falsely grouped into one cluster. We call this path change as a *transient* path change. In this case, we want to select a transient path that can help determine the cause of path change. If the cluster consists of only two path advertisements, then the choice is obvious since there is only one transient path. If many path advertisements are contained in the cluster, the transient path is selected according the duration it remains unchanged — a similar heuristic as we select the converged paths. Specifically, we set $path_n$ to the first transient path that is different from $path_o$ and lives longest.

For peering-based clustering, a pattern is a prefix-based cluster denoted by $p = (time, type, peerings_o, peerings_n)$, where $time$ is the time of the first path advertisement in p and $type$ is -1, 1, or 0 corresponding to the *long*, *short*, or *equal* path-change types. In computing $peerings_o$ and $peerings_n$, we remove from them those shared peerings, *i.e.*,

$$peerings_o = peerings_o - (peerings_o \cap peering_n)$$
$$peerings_n = peerings_n - (peerings_o \cap peering_n)$$

The similarity function is defined as:

$$s_2(p_i, p_j) = \begin{cases} -1, & |p_i.type - p_j.type| > 1 \\ -1, & p_i.type, p_j.type \leq 0 \text{ and} \\ & p_i.peerings_o \cap p_j.peerings_o = \emptyset \\ -1, & p_i.type, p_j.type \geq 0 \text{ and} \\ & p_i.peerings_n \cap p_j.peerings_n = \emptyset \\ T_2 - |p_i.time - p_j.time|, & \text{otherwise} \end{cases}$$

T_2 is a parameter of the similarity function. When applying this similarity function to the complete-link algorithm, we set the threshold $s_2(p_i, p_j) \geq 0$ such that any two patterns in the same cluster have a non-negative similarity measure, while patterns in different clusters have negative similarity measures. The first condition in $s_2(p_i, p_j)$ states that if p_i (or p_j) is of type *long* and p_j (or p_i) is of type *short*, then they have a negative similarity measure, hence the two clusters containing them cannot be merged together. The intuition behind this type of clustering is described as follows. When a routing event causes *long* path changes, it may also cause *equal* path changes, but less likely the *short* path changes. Based on our observation, we also assume that this routing event is more likely to take place in $peerings_o$ than in $peerings_n$. Thus, we only consider the $peerings_o$ in the second condition. Similarly, when a routing event causes *short* path changes, it may also cause *equal* path changes, but less likely the *long* path changes. And we assume that this routing event is more likely to take place in $peerings_n$ than in $peerings_o$, which is stated in the third condition. The forth condition states that the first path advertisements in p_i and p_j must be transmitted within T_2 seconds.

We also assign each peering-based cluster a path-change type and identify the possible peerings where the routing event may occur. If the peering-based cluster contains at least one prefix-based cluster of type *long*, then it has a *long* path-change type. If the peering-based cluster contains at least one prefix-based cluster of type *short*, then it is of type *short*. Otherwise, it is of type *equal*. For peering-based clusters of type *long*, the possible peerings are the intersection set of $peerings_o$ of all prefix-based clusters in it. For peering-based clusters of type *short*, the possible peerings are the intersection set of $peerings_n$ of all prefix-based clusters in it. For peering-based clusters of type *equal*, the possible peerings include the intersection set of $peerings_o$ and the intersection set of $peerings_n$.

2.5. Identifying Where Events Happen

This paper provides analysis on the generated clusters to estimate the distances from the event originator to both ends of the path. We define d_o as the number of AS hops from the event originator to the origin AS, and d_v as the number of AS hops from the event originator to the vantage point. We use a conservative heuristic to determine their lower bound. The heuristic works as follows. Given a prefix-based cluster, find the set of peerings $(peerings_o \cup peering_n) - (peerings_o \cap peering_n)$. The ASes in these AS peerings are candidates of the event originator. Define d_o as the minimum number of AS hops between the candidates and the origin AS. Similarly, d_v is set to the minimum number of AS hops between the candidates and the AS containing the vantage point.

In the scenario of Figure 1, prefix-based clustering will generate two clusters: one for the path change between x and o, another for the path change between y and o. In the first cluster, the candidates of event originators include a, b, c, d, and e, hence the distance estimates are $d_o = d_v = 1$. On the other hand, in the second cluster the candidates of event originators include y, b, c, d, and e, hence the distance estimates are $d_o = 1$ and $d_v = 0$.

This heuristic does not work well for the cases that one of $path_o$ and $path_n$ is empty, which always results in $d_v = d_o = 0$. We use another heuristic for these cases. The idea is to use other vantage point's *stable* path to infer the d_o. For example, vantage point v_1 observes a path change at time t for prefix f: from $path_1(v_1, f)$ to $path_2(v_1, f)$ which is empty. Suppose that vantage point v_2 has a path $path(v_2, f) \neq \emptyset$ that remains unchanged during the period from $t - T_1$ to $t + T_1$, then $path(v_2, f)$ is a stable path. Then, we find the common peerings in both $path_1(v_1, f)$ and $path(v_2, f)$ to defined d_o. Distance d_v is always set to 0 for these cases. If there are no stable paths to this prefix or other vantage points don't even have paths to this prefix. we define $d_o = d_v = -1$ which means the heuristics are unable to determine the values due to lack of information.

For peering-based clusters, we set d_o (and d_v) to the mean value of the d_o's (and d_v's) of the prefix-clusters in them.

3. Trace of BGP Updates

Our analysis is based on a year-long trace of BGP updates collected from 31 vantage points from July 2002 to June 2003. This trace was collected by RouteViews [2], and contains about 1,680 million path advertisements. Of these, 62% (about 1,040M) change the AS_PATH, the remainder changes other path attributes. At the time of the study, RouteViews has 31 peers in 24 different ASes. These peers are regarded as vantage points. There are 205,408 unique prefixes appearing in this data set, but no vantage point observes all of these prefixes.

A well-known problem with the data sets of RouteViews and RIPE RIS is that the peering sessions between the data collection sites and the vantage points are not stable. As a consequence, when the sessions failover, vantage points may re-advertise the entire BGP routing table. However, this has no impact on our results since the paths readvertised are the same as those advertised before the BGP session reset. Our clustering algorithm only considers the path advertisements that change AS_PATH and will discard the duplicates (38% in our data set).

4. Analysis on Prefix-based Clusters

We conduct three analyses of the one-year trace by applying the clustering algorithms with parameter T_1 set to 60, 120, and 240 seconds, respectively. The numbers of prefix-based clusters generated are shown in Table 1(a).

4.1. Cluster Duration

The number of clusters generated by the clustering algorithm is a function of the value of T_1. The clustering with large T_1 may group the path advertisements caused by multiple related events if the events occur closely in time. For example, path advertisements caused by a peering failure followed by a repair can be grouped into one cluster if the failover time is smaller than $C_{fail} + T_1$, where C_{fail} is the convergence time of the failure. Convergence time of an event is the interval from the time when the event happens to the time when all ASes in Internet observe the new paths resulted from the event. Labovitz *et al* [11] demonstrated that most routing events converge within 180 seconds. Selecting a T_1 smaller than this value will therefore reduce the possibility of clustering multiple events as one.

We define *cluster duration* as the time interval between the first path advertisement and the last path advertisement in the same cluster. Figure 2 plots the complimentary cumulative distribution function (CCDF) of the duration of prefix-based clusters in log-log scale. As shown in Table 1(a), more than 97% of prefix-based clusters have duration ≤ 180 seconds as $T_1 < 180$, while the percentage drops to 89% as $T_1 > 180$. This justifies the suggestion of setting the parameter $T_1 \leq 180$.

Figure 3 shows the size of prefix-based clusters in numbers of path advertisements. About half of prefix-based clusters consist of only one path advertisement, hence have duration = 0. Part of the reason for a large number of singleton advertisements is the BGP rate limiting mechanism, where a router won't send more than one path announcement for the same prefix within MinRouteAdver time. As

(a)

T_1	**60 seconds**	**120 seconds**	**240 seconds**
Number of *Prefix-based Clusters*	690,755,435	599,011,590	516,080,835
with duration $=0$ sec	448,991,021 (65%)	323,466,254 (54%)	237,397,186 (46%)
with duration ≤ 180 sec	683,847,871 (99%)	581,041,243 (97%)	459,311,934 (89%)

(b)

T_2 (when $T_1 = 60$)	**30 seconds**	**60 seconds**	**90 seconds**
Number of *Peering-based Clusters*	39,312,537	32,476,735	29,499,547
with duration $=0$ sec	4,717,514 (12%)	3,245,563 (10%)	2,360,864 (8%)
with duration ≤ 180 sec	37,739,035 (96%)	30,201,353 (93%)	26,844,578 (91%)

Table 1. Results of clustering: (a) number of *prefix-based clusters*; (b) number of *peering-based clusters*.

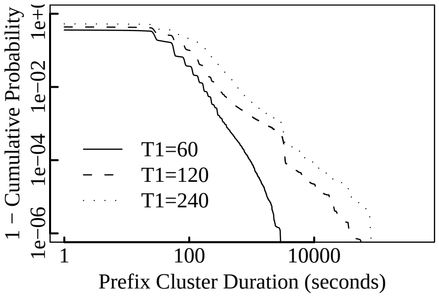

Figure 2. Distribution of the duration of prefix-based cluster.

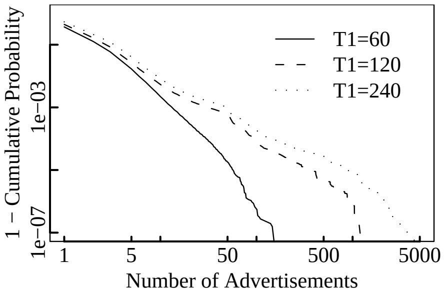

Figure 3. Distribution of the number of path advertisements in a prefix-based cluster.

explained in previous section, the convergence time of path changes is typically small due to the small number of alternate paths. Thus, if the path change converges before the vantage point's MinRouteAdver timer expires, then only one path advertisement is transmitted by the vantage point.

To understand know how many path changes are caused by origin AS changing, we calculate the number O of origin ASes appeared in a prefix-based cluster. Table 2 shows the results. $O = 2$ represents the path changes resulted from origin AS changing. The cases of $O \geq 3$ may indicate multihoming or misconfiguration in network [16].

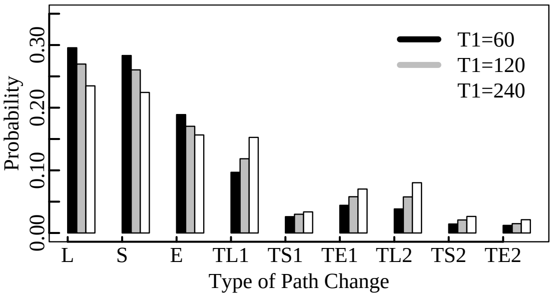

Figure 4. Type of path change for *prefix-based cluster*.

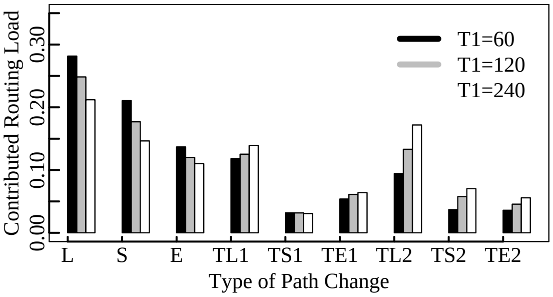

Figure 5. Routing load contributed by each type of path change.

4.2. Types of Path Changes

For each prefix-based cluster, we determine its type of path change using the heuristic described in Section 2.4. Figure 4 plots the probability of each type, where *long*, *short*, and *equal* are denoted by L, S, and E, respectively. The figure also separates the transient path changes from non-transient path changes. *Transient* path change means that the paths before the events are the same as the paths after the events. In the figure, Tx1 means there is only one transient path in the cluster, and Tx2 means the clusters have more than one transient paths. The number of *long* prefix-based cluster (L) is roughly the same as that of *short* prefix-based cluster (S). This complies to the intuition that a path failed will be repaired some time in the future.

It is interesting that 23–37% of prefix-based clusters are

T_1	$O = 1$	$O = 2$	$O = 3$	$O = 4$	$O = 5$	$O > 5$
60 seconds	678,390,873 (98.21%)	12,226,372 (1.77%)	138,151 (0.02%)	39	none	none
120 seconds	586,192,742 (97.86%)	11,860,429 (1.98%)	898,517 (0.15%)	59,801	101	none
240 seconds	503,385,246 (97.54%)	10,579,657 (2.05%)	2,064,323 (0.40%)	50,589	1020	none

Table 2. Number of origin ASes in a prefix-based cluster.

caused by transient path changes. These path changes represent the transient instability of the routing infrastructure. We measure the routing load by summing the sizes of clusters for each type of path change. The result is shown in Figure 5. The transient path changes cause significant amount (35–52%) of path advertisements in the one-year trace. This indicates a remarkable instability of BGP routing topology, which suggests a need for future work to reduce these routing overhead.

4.3. Distance to Event Originators

To understand *where* routing events happen we next approximate the distance to the event originator both from the vantage point and the origin AS. We apply the heuristics described in Section 2.5 and show the results in Figure 6.

Figure 6(a) shows that at least 45% of prefix-based clusters are caused by ASes other than the origin, *i.e.*, $d_o > 0$. On the other hand, when measuring distance from the vantage point (Figure 6(b)), this bias is not nearly as strong, with 65% of events happening in the first-hop AS and only 15% happening further away. This result is somewhat unexpected because traditional wisdom suggests that the core of the network is stable and most events happen at an edge's connection to the ISP. One reason that many path changes happen "in the middle" is that a single transit peering handles paths for many prefixes from many origin ASes, while a peering adjacent to an origin AS only handle the paths to reach the prefixes of this origin AS. Thus, a routing event on a transit peering results in more path changes than an event on a peering adjacent to an origin AS. This also suggests that prefix-based clustering may over-estimate the number of routing events actually occurred. As shown in later section, peering-based clustering is able to reduce this inflation effect on the estimation.

To characterize where an event happens, Figure 6(c) shows the *distance ratio* $(d_v + 1)/((d_v + 1) + (d_o + 1))$ which indicates whether the event originator is close to vantage point or origin AS. This computation considers several special cases: we ignore the 19% of events where we cannot determine the distance where either d_v or d_o are -1 or $d_v = d_o = 0$ since it would not make sense to apply this ratio to those events. Ratios close to zero implies the event originator is close to vantage point, while close to one implies events near the origin AS. The figure suggests that, where the distance can be estimated, most of path changes are closer to the vantage point.

5. Analysis on Peering-based Clusters

Although prefix-based clusters are useful to map individual routing updates to routing events, in practice a single event may change paths to many prefixes. Moreover, prefix-based clustering is per vantage point based, hence the more vantage points we have, the more clusters we obtain, regardless of the number of actual routing events. We therefore next consider peering-based clustering that can provide a better estimate on the number of routing events. We apply the peering-based clustering with T_2 = 30, 60, and 90 on the pattern set of prefix-based clusters generated with $T_1 = 60$. The results are shown in Table 1(b).

5.1. Topological Relations Among Path Advertisements

First, we look at the composition of a peering-based cluster. Figure 7(a) plots the CCDF of the size of a peering-based cluster which consists of prefix-based clusters. The median size is 4–5, and mean 16–24, indicating a non-Gaussian distribution of cluster size. Since peering-based clustering puts into the same cluster those path advertisements that have topological relation with each other, it is interesting to know to what extent the path advertisements in a stream of BGP updates are topologically correlated. Figure 7(b) shows that 92–94% of peering-based clusters contains more than one path advertisements, while the median and mean numbers of path advertisements in a cluster are 8–9 and 39–53, respectively. In other words, most path advertisements are topologically related to 7–8 other path advertisements. This characterization of topological relations among path advertisement can help design a more realistic BGP traffic for simulation use in addition to those metrics consider in [15], *e.g.*, prefix length distribution, fanout, nesting structure of prefixes, *etc*. Figure 7(c) shows the distribution of cluster duration. Compared with Figure 2 for $T_1 = 60$, it shows long duration prefix-based clusters are more likely to be merged into peering-based clusters than short duration ones.

5.2. The Impact of Path-Change Events

A peering-based cluster represents the set of path changes that are caused by a single routing event and observed by some vantage points in the Internet. The clustering results indicate that there are around four thousand

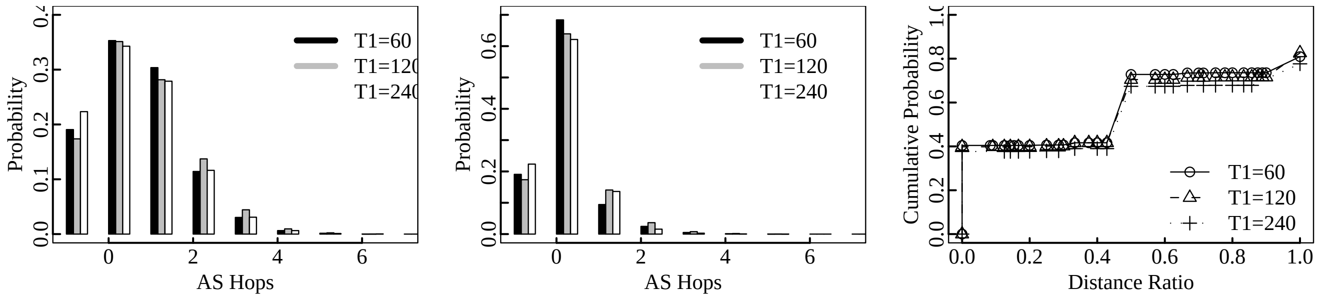

Figure 6. (a) Lower bound d_o of the distance from event originator to origin AS. (b) Lower bound d_v of the distance from event originator to vantage point. (c) Distance ratio $(d_v + 1)/(d_v + d_o + 2)$.

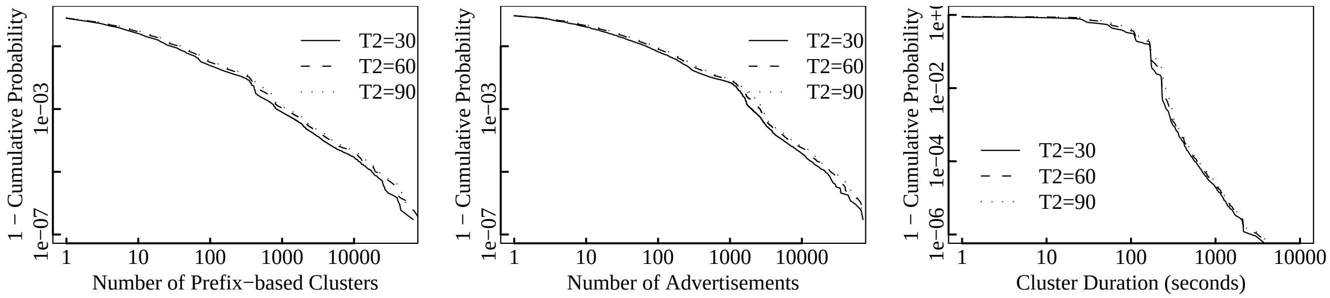

Figure 7. CCDF of (a) the number of prefix-based clusters, (b) the number of the path advertisements, and (c) the duration in a peering-based cluster.

Type	$T_2 = 30$	$T_2 = 60$	$T_2 = 90$
Long	22M (54.87%)	18M (55.35%)	16M (55.57%)
Short	15M (39.11%)	13M (39.42%)	12M (40.01%)
Equal	2.4M (6.02%)	1.7M (5.23%)	1.3M (4.42%)

Table 3. Type of path change for a peering-based cluster.

path-change events occurring per hour (Table 1(b)). It is interesting to know to what extent the path-change events impact the inter-domain routing.

Figure 8(a) shows the CCDF of the number of unique prefixes appearing in a peering-based cluster. The median number is 2 indicating that most events changes the paths to a few prefixes. On the other hand, there are 88-91 events affecting the routing of tens of thousands of prefixes, which may correspond to peering resets occurring near to vantage points. Considering the effect on the path length, we find that more than half of events result in longer paths to at least one prefix, as shown in Table 3.

5.3. Where The Events Happen

This section addresses the problem of where the routing events take place. First, we compute the distance from an event to the origin (d_o). Figure 8(b) shows that at least 47% of routing events taking place in transit peerings ($d_o > 0$). This observation implies that a content provider cannot guarantee end-to-end routing stability based solely on its relationship with its immediate ISP.

Previous researches suggest that different paths have different instability characteristics, although they only examine a limited set of paths [7]. Here we provide a global estimate on the occurrence rate of routing events for each peering. Figure 8(c) plots the distribution of the estimated number of events occurring in a peering in the month of Mar. 2003. The median are mean number are 9–13 and 138–191, respectively. This skew distribution conforms to the expectation that the peering instability is significantly varied over the Internet.

6. Conclusion and Future Directions

To our knowledge, this paper represents the first study identifying inter-domain path-change events from a stream of BGP updates. To characterize the events, this paper develops:

- A clustering method to partition the stream of BGP updates into small clusters that approximate path-change events in the inter-domain routing system.

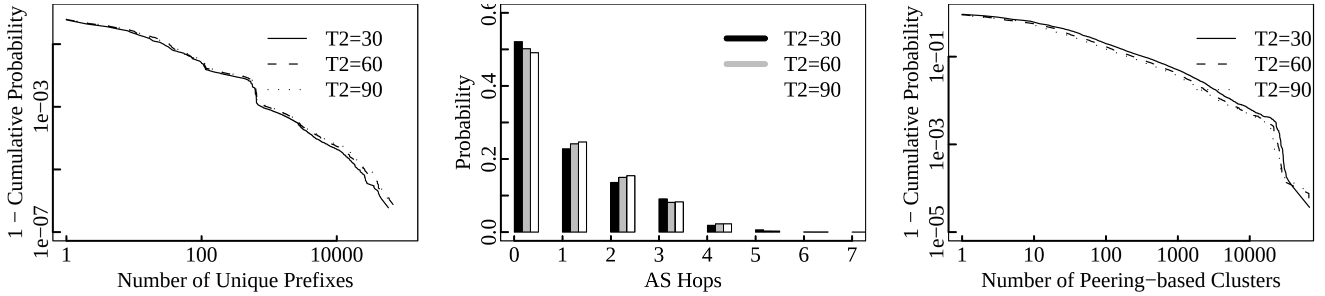

Figure 8. Peering-based cluster: (a) CCDF of the number of unique prefixes. (b) Approximate distance (d_o) from an event to the origin. (c) CCDF of the number of peering-based clusters caused by events in a single peering.

- An approach to approximating the distance between the events and the observer and originator of the prefix. This analysis suggests that at least 45% of path changes occur outside the origin AS.

Our analysis results suggest several directions for future work:

- We provide an upper bound on the number of BGP path-change events. However, the large number of candidates for event originator prevent us to do further clustering. Thus, estimating the exact number of events remains an open question.
- We observed that many path advertisements (35–52%) result from transient path changes. This result suggests that the number of routing updates can be noticeably reduced by reducing these transient events.

References

[1] RIPE. *Routing Information Service*. `http://data.ris.ripe.net/`.

[2] University of Oregon. *Route Views Project*. `http://www.antc.uoregon.edu/route-views/`.

[3] M. R. Anderberg. *Cluster analysis for applications*. Academic Press, 1973.

[4] D. Andersen, N. Feamster, S. Bauer, and H. Balakrishnan. Topology inference from BGP routing dynamics. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, 2002.

[5] D.-F. Chang, R. Govindan, and J. Heidemann. The temporal and topological characteristics of BGP path changes. Technical report, USC/Information Sciences Institute, Aug. 2003. http://www.isi.edu/˜difac/isi-tr-2003.ps.gz.

[6] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. In *Proceedings of the ACM SIGCOMM*, pages 251–262, 1999.

[7] N. Feamster, D. Andersen, H. Balakrishnan, and M. F. Kaashoek. Measuring the effects of Internet path faults on reactive routing. In *Proceedings of the ACM SIGMETRICS*, 2003.

[8] G. Huston. *BGP Table Statistics*. `http://www.telstra.net/ops/bgp`.

[9] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice Hall, 1988.

[10] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, Sept. 1999.

[11] C. Labovitz, A. Ahuja, A. Abose, and F. Jahanian. An experimental study of delayed Internet routing convergence. In *Proceedings of the ACM SIGCOMM*, pages 175–187, 2000.

[12] C. Labovitz, A. Ahuja, and F. Jahanian. Experimental study of internet stability and wide-area network failures. In *Proceedings of the FTCS*, 1999.

[13] C. Labovitz, G. R. Malan, and F. Jahanian. Internet routing instability. *IEEE/ACM Transactions on Networking*, 6(5):515–528, Oct. 1998.

[14] C. Labovitz, G. R. Malan, and F. Jahanian. Origins of Internet routing instability. In *Proceedings of the IEEE INFOCOM*, pages 218–226, 1999.

[15] O. Maennel and A. Feldmann. Realistic BGP traffic for test labs. In *Proceedings of the ACM SIGCOMM*, 2002.

[16] R. Mahajan, D. Wetherall, and T. Anderson. Understanding BGP misconfiguration. In *Proceedings of the ACM SIGCOMM*, 2002.

[17] Y. Rekhter, T. Li, and S. Hares. *A Border Gateway Protocol 4 (BGP-4)*, draft-ietf-idr-bgp4-20.txt edition, Apr. 2003.

[18] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang. BGP routing stability of popular destinations. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, 2002.

[19] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the Internet hierarchy from multiple vantage points. In *Proceedings of the IEEE INFOCOM*, June 2002.

[20] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Network topology generators: Degree-based vs. structural. In *Proceedings of the ACM SIGCOMM*, 2002.

[21] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. Wu, and L. Zhang. An analysis of BGP multiple origin as (MOAS) conflicts. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, pages 31–35, 2001.