Detection System built upon BEAM

Our detection system built upon BEAM consists of three components, *i.e.,* the routing monitor, the BEAM engine, and the anomaly detector. At a high level, the routing monitor detects route changes from the BGP route update announcements. The BEAM engine utilizes a pre-trained BEAM model to compute the path difference scores of the route changes. The detailed path difference score computation process is presented in §4.2 in our paper. Here, we supplement its pseudo-code in Algorithm 1. Besides, the anomaly detector is responsible for identifying routing anomalies from the route changes and generating corresponding alarms. Here, we will introduce the details of the routing monitor and anomaly detector.

The Routing Monitor. This component collects the BGP route update announcements from global vantage points, to detect route changes. In particular, it maintains a routing table for each vantage point. To reduce the storage space and facilitate detection, the routing table is in a trie structure, where each node represents a unique prefix and its routing path. Also, the prefix of each parent node is the super prefix of its child nodes. When the routing monitor receives a new route update message that announces routing path l to prefix p from a vantage point, it searches p in the corresponding trie structure routing table. If a routing path to prefix p (denoted by l') already exists in the routing table, we compare it with l and find a route change when $l \neq l'$. If p is not in the routing table, we compare l with the routing path to the most specific super prefix of p, which can be obtained from the parent node of p. After detecting route changes, the routing monitor updates the routing table with the newly received update message.

The Anomaly Detector. This component aims to detect routing anomalies from the route changes and generate corresponding alarms. Its detection procedure includes three steps: detecting suspicious route changes, identifying anomalous prefixes, and locating responsible ASes.

Formally, we represent each detected route change by (t,r,p,p',l,l'), where t is the occurrence time of the route change, r is the AS number of the vantage point that captures the route change, p is the prefix announced in the update announcement that triggers the route change, p' is the prefix in the routing table that conflicts with p (*i.e.,* the same as p or the most specific super prefix of p), and l and l' are the routing path to p and p', respectively. The computed path difference score between l and l' is denoted as $d_{l,l'}$. If $d_{l,l'}$ is above a threshold th_d computed based on historical legitimate route changes (detailed in §5.2 in our paper), the anomaly detector flags the route change as *suspicious*.

It is necessary to prioritize the widespread routing anomalies captured by multiple vantage points, otherwise the transient route changes in the Internet will generate numerous insignificant alarms. Towards this end, the anomaly detector further groups suspicious route changes that share the same (p,p') and denotes each group as a *prefix event*, where each event is associated with a specific (p,p') and sorts the suspicious route changes by their occurrence time. To each event, the anomaly detector utilizes a sliding window (with time interval w) to count the number of individual vantage points that observe the suspicious route changes within the window. If the counted number (denoted by N_r) is above another threshold th_v, the anomaly detector considers the prefix event is caused by a widespread routing anomaly and regards it as anomalous. The formal definition of anomalous prefix event is:

Definition 1 (Anomalous Prefix Event)**.** *Given parameters* th_d, w, th_v*, a prefix event of* $(\hat{p},\hat{p}') \equiv P = \{(t,r,p,p',l,l') \mid p = \hat{p}, p' = \hat{p}', d_{l,l'} > th_d\}$ *is anomalous if and only if* $\exists A \subseteq P, \forall (t_1,r_1,p_1,p'_1,l_1,l'_1),(t_2,r_2,p_2,p'_2,l_2,l'_2) \in A, |t_1 - t_2| < w, N_r = |\{r \mid (t,r,p,p',l,l') \in A\}| > th_v$.

In realistic circumstances, a misbehaved ASes may impact multiple prefixes simultaneously, *e.g.,* AS 55410 hijacked more than 30,000 prefixes in April 2021, resulting in several different anomalous prefix events. To provide network administrators with comprehensive information about the impacted prefixes of each routing anomaly, the anomaly detector correlates all anomalous prefix events based on their responsible ASes. Intuitively, the ASes responsible for a suspicious route change (where $l = \{AS_1, AS_2, \ldots, AS_n\}, l' = \{AS'_1, AS'_2, \ldots, AS'_m\}$) are either within the ASes present in

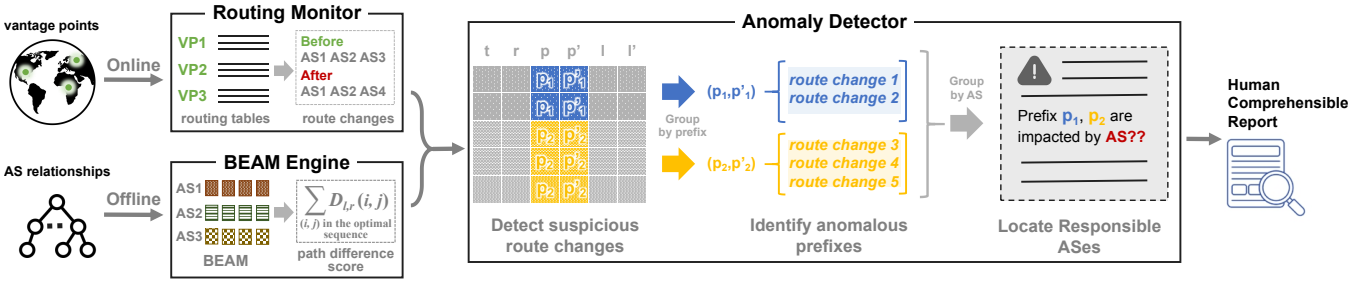

Figure 1: **The workflow of our routing anomaly detection system built upon BEAM.**

Algorithm 1 Measuring Path Difference Scores

```
1:  function PathDiffScore(S: Array [1...m], S': Array [1...n])
2:      external function D_{l,r}
3:      DIFF := Array [0...m, 0...n]
4:      for i := 0 to m do
5:          for j := 0 to n do
6:              DIFF[i, j] := +∞
7:      DIFF[0, 0] := 0
8:      for i := 1 to m do
9:          for j := 1 to n do
10:             diff := D_{l,r}(S[i], S'[j])
11:             DIFF[i, j] := diff + minimum(DIFF[i − 1, j],
                              DIFF[i, j − 1], DIFF[i − 1, j − 1])
12:     return DIFF[m, n]
```

l but absent in l' (*i.e.,* $l - l'$ in terms of set subtraction and we denote it as the *drop-out AS set*), or within the ASes absent in l but present in l' (*i.e.,* $l' - l$ in terms of set subtraction and we denote it as the *pop-up AS set*). For each anomalous prefix event, we compare its all suspicious route changes to find the intersection of their drop-out AS sets and the intersection of their pop-up AS sets, respectively. If any set is not empty, we denote the ASes in two intersection sets as the responsible ASes for the anomalous prefix event. Thus, the responsible ASes for an anomalous prefix event is defined as follows:

Definition 2 (Responsible AS)**.** *Given an anomalous prefix event A, let $L = \{(l, l') \mid (t, r, p, p', l, l') \in A\}$. If $I_{drop\text{-}out} = \bigcap_{(l,l') \in L}(l - l') \neq \emptyset$ or $I_{pop\text{-}up} = \bigcap_{(l,l') \in L}(l' - l) \neq \emptyset$, the responsible ASes for A are the ASes belonging to $I_{drop\text{-}out} \cup I_{pop\text{-}up}$. Otherwise, there are no responsible ASes for A.*

Then, the anomaly detector correlates the anomalous prefix events based on their responsible ASes. Given two different anomalous prefix events, if their time ranges are overlapped and they have common responsible ASes, we consider they are correlated. The formal definition of a correlated anomalous prefix event is:

Definition 3 (Correlated Anomalous Prefix Event)**.** *Given two different anomalous prefix events A and A', let $T = \{t \mid (t, r, p, p', l, l') \in A\}, T' = \{t \mid (t, r, p, p', l, l') \in A'\}$, denote the set of responsible ASes for A as R_A and that for A' as $R_{A'}$. A and A' are correlated if and only if $R_A \cap R_{A'} \neq \emptyset, \exists t_1, t_2 \in T, t_1 < t_2, t'_1, t'_2 \in T', t'_1 < t'_2, [t_1, t_2] \cap [t'_1, t'_2] \neq \emptyset$.*

Finally, we divide all anomalous prefix events into different sets, where the event in one set only correlates the other events in the same set. The anomaly detector treats each set as an individual routing anomaly and outputs a corresponding alarm that includes both the affected prefixes and the responsible ASes.