

# The elusive Effect of Routing Dynamics on Traffic Anomalies

Ricardo Oliveira  
University of California, Los Angeles  
rveloso@cs.ucla.edu

Fernando Silveira  
Thomson & LIP6  
fernando.silveira@thomson.net

Renata Teixeira  
CNRS, France  
renata.teixeira@lip6.fr

Christophe Diot  
Thomson  
christophe.diot@thomson.net

## ABSTRACT

Network operators need to know the root cause of traffic anomalies to determine the appropriate action to mitigate their effect. This paper studies the one type of anomaly for which we can know the root cause: routing-induced anomalies. A major challenge in quantifying the effect of routing events on traffic is that the flows affected by routing are not easily distinguishable from the rest of the traffic. We study this problem using both logs of routing messages and traffic flow records of two research backbones. We define the notion of *routing impact* in order to quantify how much of the observed volume variation in the traffic matrix is caused by routing changes. Our solution involves a method to compute the traffic matrix that achieves up-to-second accuracy. Results indicate that routing-induced traffic anomalies correspond to a very small fraction ( $< 5\%$ ) of the total volume anomalies detected by a Kalman detector. We show that most of the significant routing shifts ( $> 80\%$ ) remain undetected because other volume changes hide their effect, or because they have a limited impact on traffic volume. However, the routing events detected by Kalman correspond to those events that have the biggest impact on the traffic. Running Kalman on destination address or prefixes instead of packets count can improve the detection of routing anomalies. These results represent the promise of a model of routing-induced anomalies, which could be used to accurately identify routing-induced traffic shifts even when no routing messages are available.

## 1. INTRODUCTION

Network operators need automatic techniques to detect abrupt changes in network traffic, so that they can take corrective action. For example, if a customer is experiencing a denial-of-service attack, the operator should quickly detect and block this attack. A common approach for network operators to detect large traffic changes is to (1) represent traffic into a matrix of load from each ingress point to each egress point in the network over a particular time interval - the *traffic matrix*; and (2) analyze the traffic matrix

with statistical techniques that pinpoint outliers or *traffic anomalies* as deviations from a statistical baseline defined by the aggregate behavior of the network. A trademark of previous studies on anomaly detection [9, 10, 17, 11, 16] is that they focus in detecting the *effects* of anomalous events, without complete knowledge about the *root cause* of these events. In fact, one of the major hurdles to evaluate anomaly detectors is the lack of comprehensive ground truth on the origin of anomalous events, and an approach often used is to manually infer the root cause of anomalies based on observed patterns.

In this paper, we use full knowledge of traffic and routing information of two research backbones (Abilene and Geant) to establish a causal relation between *routing dynamics* and the respective traffic variations that trigger traffic anomalies. It is important for network operators to understand such relation, since routing changes impact end-to-end performance, potentially causing delay variations, loss of reachability and link congestion. Furthermore, upon a routing anomaly, network operators may need to re-engineer their network to match the new traffic demands; whereas upon an anomaly such as an attack the action would be to block the attacker.

The main goal of our work is to design a technique to accurately distinguish routing anomalies from other traffic anomalies. As a first step, this paper quantifies how well a typical anomaly detector captures routing changes. Fundamentally, the problem stems from the fact that the flows affected by a routing change are not easily distinguishable from the rest of the traffic. Therefore, our main challenge lies in decoupling the variations of traffic associated with routing changes from variations due to traffic burstiness. Once we solve this problem, we will be able to automatically pinpoint the traffic anomalies that are solely originated by routing events.

There are several issues we need to address to identify routing-induced traffic anomalies. First, because of the transient nature of routing changes, the computed traffic matrix is sensitive to how often the routing information is updated. Section 4 describes a method to compute traffic matrices that accurately capture routing dynamics. Second, we need to be able to provide a metric that quantifies the effect of a given routing change. For instance, if traffic for a given destination is leaving the network through egress  $e_1$  and a routing event causes it to be routed to egress  $e_2$ , how to quantify the impact of the change  $e_1 \rightarrow e_2$ ? Considering only the flows that were active during the change might not be enough, since new flows are also affected by the change. At the same time, it does not seem reasonable to consider flows that start using  $e_2$  much later after the change, or flows that were using  $e_1$  much sooner

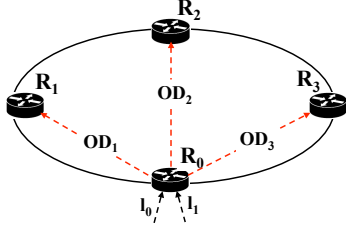


Figure 1: Traffic aggregation formalisms.

before the change. In Section 5, we define a metric called *routing impact* that quantifies the volume of traffic shifted because of routing changes under a certain time window. Third, given that a traffic anomaly was signaled by a statistical anomaly detector (the Kalman filter [17]), we need to determine if the anomaly was induced by a routing change. In Section 6, we describe a method to do this based on the routing impact metric. We find that routing anomalies represent a very small percentage ( $< 5\%$ ) of the total volume anomalies detected. Doing a reverse analysis, we find that only less than 20% of the significant routing shifts are captured by the Kalman detector, mainly because of other volume variations in the traffic aggregate that hide the effect of routing changes. Finally, we also explore two other metrics - destination IP address and destination prefixes - that are better correlated with routing changes than the volume metric. Our preliminary results indicate that these metrics can be used to infer routing changes just by looking at traffic. Such technique would allow network operators to accurately identify traffic anomalies induced by routing changes in neighboring networks.

## 2. TRAFFIC ANALYSIS BACKGROUND

This section introduces the formalisms needed for the rest of the paper. We start with defining the traffic matrix and how routing events can impact traffic, and then we describe the techniques we use to detect traffic anomalies.

### 2.1 Traffic matrix

In order to understand how traffic flows through their network, operators usually aggregate traffic in a *traffic matrix* or *OD flow* (origin-destination flow). Each element of the traffic matrix aggregates traffic from an origin to a destination router of the network, as shown in Figure 1 by flows  $OD_1$ ,  $OD_2$  and  $OD_3$ . For each OD flow, we also term the origin as *ingress router*, *i.e.* the router through which the flow enters the network (in this case  $R_0$ ). We term the destination as *egress router*, *i.e.* the router through which each flow leaves the network (*e.g.*  $R_1$  for flow  $OD_1$ ). The OD formalism is particularly useful when operators want to understand how their network is slicing the traffic to their neighbors, in other words, how traffic is *passing through and leaving* their network. Soule *et al.* [18] show that the traffic matrix formalism provides the best compromise between false positive and false negative rates when doing anomaly detection. Therefore, in this paper we focus on the traffic matrix formalism.

### 2.2 Impact of routing changes on traffic

Variations in the traffic that passes through a network  $M$  can happen among other factors because of routing changes. Figure 2

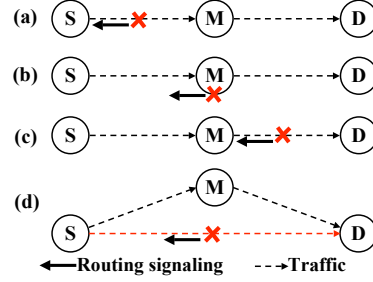


Figure 2: The impact of routing changes in traffic passes  $M$ : (a) a routing change between  $S$  and  $M$ , (b) a routing change inside of  $M$ , (c) a routing change between  $M$  and  $D$ , (d) a routing change between  $S$  and  $D$  after which  $M$  is used as backup.

depicts the possible cases where a routing event induces changes in the traffic that passes through  $M$ , where both  $S$ ,  $D$  and  $M$  represent different autonomous systems.

**Change upstream of  $M$ :** In Figure 2(a), traffic is flowing from a source  $S$  towards a destination  $D$ , passing through  $M$ , and there is a change in the path between  $S$  and  $M$ . This change will impact traffic that enters network  $M$ . For instance, the change may cause traffic from  $S$  to  $D$  to use a different ingress on  $M$ , or even to stop using  $M$  at all. However, since the change happened upstream, the routing event will not be signaled to  $M$  by the control plane.

**Change inside of  $M$ :** In Figure 2(b), a routing change happens inside of  $M$  which causes a hot potato change in BGP (*i.e.* an egress change). In this case the change appears as traffic moving from one OD flow to another. Further, the event will show up in BGP and IGP traces of  $M$ . Note that there can also be IGP changes inside of  $M$  that do not trigger any BGP change, but can still impact traffic *e.g.* by causing TCP connections to back off, however we believe these effects are negligible.

**Change downstream of  $M$ :** This is the case in Figure 2(c), in which traffic from  $S$  to  $D$  is affected by a change between  $M$  and  $D$  that is potentially signaled to  $M$ . Such a change can cause just an egress change in  $M$ , or can make the flow stop/start using  $M$ . More precisely, the flow will disappear if  $D$  becomes unreachable, and appear as soon  $D$  becomes reachable again. Note that there can also be routing changes downstream that do not change the egress point in  $M$ .

**Change after which  $M$  is in backup path:** This is depicted in Figure 2(d), where  $M$  starts being used after a failure in the path  $S-D$  that does not contain  $M$ . In this case, no routing signaling is sent to  $M$ , hence we need external vantage points to detect this scenario. However, the effects should be visible in the traffic matrix.

In this paper we will focus on cases (b) and (c), since these are the cases where the changes are signaled to  $M$ , our monitored network, and therefore we can established a causal relation between the measured traffic and the routing changes. Our goal is to get a model of traffic anomalies caused by routing, so that we can detect cases (a) and (d) just by looking at the traffic matrix.

### 2.3 Statistical anomaly detection

Given a traffic aggregation model such as OD flows, statistical anomaly detectors identify outliers across time in the data series which correspond to *irregular* activity in the network, *e.g.* DoS attacks, flash crowds, routing disruptions. A number of anomaly detection methods [6, 8, 9, 22, 17] exist in the literature. The two

methods that have been applied to network-wide traffic analysis are Kalman [17] and PCA [9]. In this paper, we use the Kalman method because recent work has exposed some issues in calibrating the parameters for PCA approach [16]. We plan to explore the use of PCA and other detection techniques in the future. We briefly describe the Kalman below.

The Kalman detector is described in detail in [17] and fundamentally it works by modeling traffic as a multivariate linear model, exploiting both the spatial and time correlation available in the data. At any point in time, Kalman can be used to predict the next values of the time series and compares those predictions against the actual measurements. If the prediction error is too high compared to the expected variance in the data, then a statistical anomaly is signaled at the space-time point where that condition is true. More precisely, the Kalman filter is itself composed of three steps applied sequentially as soon as a new measurement is available for analysis. The first step predicts the value of the series at time  $t + 1$  based on previously observed values up to time  $t$ . The second step estimates the value of the series at time  $t + 1$  based on the measured value at time  $t + 1$ , and the predicted value of the first step. Finally, the third step signals all points where the prediction error is above a certain threshold:

1. *Prediction step*: The prediction of the next value of the time series for all the OD flows is accomplished by multiplying the current estimated values  $\hat{F}_t$  by the time update matrix  $C$  obtained in a calibration phase. In this matrix the diagonal elements capture the time evolution of each OD flow whereas the non-diagonal elements capture the correlation between the OD flows. The predicted values  $\tilde{F}_t$  are derived from the time update equation  $\tilde{F}_{t+1} = C\hat{F}_t$ .
2. *Estimation step*: The estimated value for the next bin is defined as the predicted value adjusted by a correction factor:  $\hat{F}_{t+1} = \tilde{F}_{t+1} + K(F_{t+1} - \tilde{F}_{t+1})$ , where  $K$  is the Kalman gain matrix, which accounts for the confidence in the prediction model. The matrix  $K$  and the variances of the time series are updated between each iteration of the filter. The time series of the difference between the predicted value and the estimated value  $\eta_t = \tilde{F}_t - \hat{F}_t$  represents the modeling error and is often called the *Kalman innovation*.
3. *Detection step*: Assuming the model is correct, a large modeling error indicates an unexpected change in values of the time series. Detecting anomalies consists in isolating these unexpected changes. We consider that an anomaly is detected on the  $i^{th}$  OD flow at time  $t$  whenever  $|\eta_t(i)| > k \cdot \sigma_i$  where  $\sigma_i$  is the estimate of the variance of the  $i^{th}$  OD flow,  $\eta_t(i)$  is the innovation of the  $i^{th}$  OD flow at time  $t$  and  $k$  is the detection threshold. Throughout this paper, we use a detection threshold of  $4\sigma$ , also used in previous work [17]. In order to give an idea, if the innovation follows a normal distribution, then the  $4\sigma$  threshold will encompass 99.994% of the observations, leaving 0.006% as outliers.

### 3. DATA SETS

In this section we describe the data sets that we use in this paper. Our data comes from two large research backbones: Geant and Abilene, and include both traffic (flow records) and routing (BGP and IS-IS) traces. Both data sets are publicly available for research purposes from their respective sources.

#### 3.1 Geant

Geant (AS20965) is the European research backbone connecting 30 national research and education networks representing 34 countries across Europe. Geant provides logs of BGP tables and updates, as well as IS-IS traces [3]. For our study, we used the entire month of August 2007, and during that period Geant had 20 Points-of-Presence in different countries, each hosting a backbone router. We also used Netflow data collected at the entry points of the network from input interfaces of these 20 routers. The Netflow data is exported in intervals of 15 minutes at a packet sampling rate of 1:1000. The BGP monitor in Geant is connected as part of the iBGP mesh, as depicted in Figure 5, where the monitor  $M$  receives all eBGP-learned routes. Geant enables connectivity between research and education networks (R&E) and the commercial Internet, and therefore its routers have full tables with  $\sim 225k$  prefixes.

#### 3.2 Abilene

Abilene (AS11537) is the network interconnecting universities and research institutes in the US. We use Abilene’s BGP tables and updates [1] for the entire month of January 2006, since this was the earliest month in which there was complete routing information from all the routers<sup>1</sup>. During that period, Abilene had 11 PoPs each hosting a backbone router. We use the Netflow data collected at the entry points of Abilene in these 11 routers<sup>2</sup>. The Netflow data is exported in 5-minute intervals at a packet sampling rate of 1:100. Furthermore, the last 11 bits of each IP address in the monitored traffic are anonymized (filled with zeros). The BGP monitoring in Abilene is slightly different than in the Geant case, since (1) there is a monitor per PoP and (2) each monitor is configured as a route reflector client, which means each monitor receives both iBGP-learned and eBGP-learned routes from the routers it connects to. Since Abilene generally does not provide transit to the global Internet, its routers have partial tables with only  $\sim 10k$  prefixes representing R&E networks.

### 4. ACCURATE TRAFFIC MATRICES

The accuracy of the traffic matrix depends on several factors, the most important being the accuracy of the routing tables used to reconstruct the flows and the correct alignment of these tables with the arrival times of the packets at each router. Since our goal is to assess the impact of routing changes on traffic, and since these changes may occur in very short time scales, it is important that we reconstruct routing tables as soon as the change in the table is signaled by routing updates. More formally, we are interested in obtaining a mapping function  $M(k_t^i)$  for each packet  $k_t^i$  arriving at ingress router  $i$  at time  $t$ , that will output the egress router through which the packet will leave the network. Previous approaches to obtain this mapping either used daily snapshots of routing tables [9], or took period snapshots spaced by 10 minutes or more [5, 19]. The main caveat of these approaches is that the impact of short duration routing changes on traffic remain undetected. Figure 3 shows an example of a set of egress changes for a router when reaching a given prefix  $P$ . The egress *null* means that the prefix is unreachable at the time. Since the routing table is only refreshed at the beginning and end of the interval, packets  $k_0, \dots, k_5$  will be all mapped to egress  $e_1$ , even though only half of these packets ( $k_0, k_1$  and  $k_5$ ) passed through it. Therefore, one

<sup>1</sup>The NY router stopped being monitored in February 2006, and the network suffered a major reconfiguration that started in November 2006.

<sup>2</sup>We actually had to remove the backbone links from the Netflow data, as we explain in Section 4.

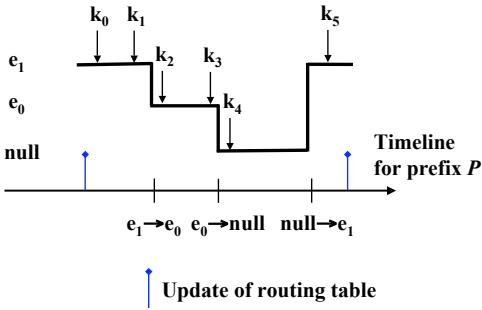


Figure 3: Example where the periodic update of routing table create errors in the traffic matrix.

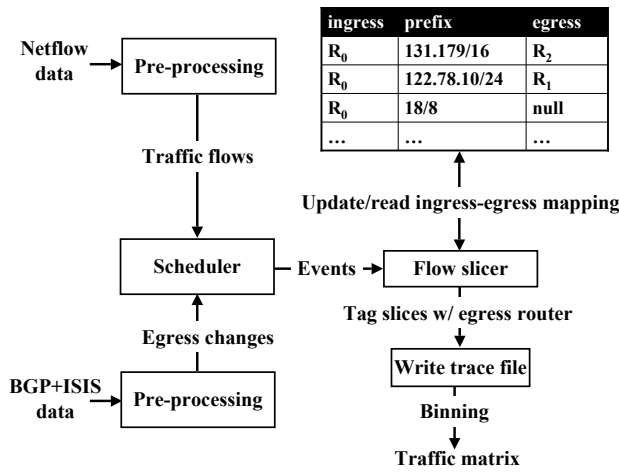


Figure 4: Traffic matrix computation.

would not notice a decrease in the traffic volume passing through  $e_1$ , even though it existed and corresponded to 50% of the observed traffic. Note that this would not happen had we refreshed the routing table as soon as the egress change was signaled in the form of routing updates, which is the approach we take in this paper.

Figure 4 summarizes our method of traffic matrix computation, which we outline here. Both routing and traffic data undergo a pre-processing stage, after which a list of events is extracted and passed to a scheduler. These events can be egress changes, flow arrivals or flow terminations. The scheduler sorts the events and places them into a queue, after which each event is processed in a first-in-first-out fashion. Traffic flows are sliced according to egress changes by the flow slicer, which also updates the ingress-egress mapping for each router and tags each slice with the respective egress, recording it to a trace file. The trace file is then analyzed and traffic volume is aggregated in bins of fixed size, corresponding to the traffic matrix. In the rest of this section we describe in detail each of these steps.

#### 4.1 Pre-processing routing data

**Detecting egress changes:** We describe here the steps we take to detect the egress changes from raw routing data available from Geant and Abilene. In Geant, we are able to reconstruct the routing

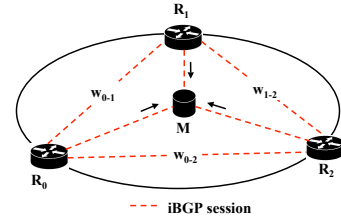


Figure 5: BGP data collection in Geant, the monitor  $M$  is configured as a router participating in the iBGP mesh. In contrast, in Abilene there is a monitor per PoP that is configured as a route reflector client.

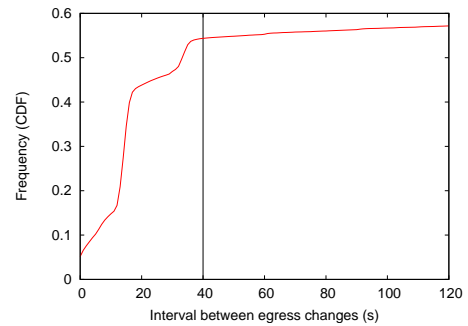


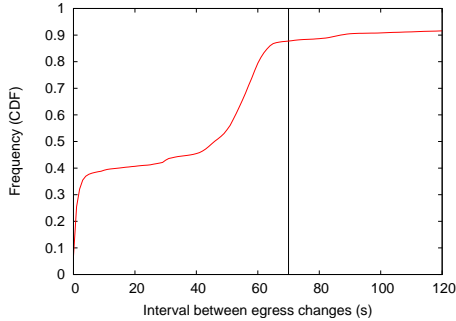
Figure 6: Interval between egress changes for all Geant routers, August 2007.

tables of all the routers by looking at the routes received by the monitor  $M$  in Figure 5 plus the IGP weights  $w_{i-j}$  extracted from IS-IS traces, following an approach similar to [21]. Since iBGP-learned routes are not propagated to the monitor  $M$ , we need the IGP weights to infer these routes. In a first phase, we parse the IS-IS traces and keep track of IGP weight changes. Every time there is a change in IGP weights (*e.g.* a link failure inside the network), we run the Floyd algorithm to recompute the shortest IGP distance between each pair of routers. In a second phase, we synchronize the IGP changes with the eBGP-learned route updates<sup>3</sup>, and for each router we simulate the BGP decision process up to the hot-potato step, *i.e.* we look at (1) local pref, (2) AS path length, (3) origin type, (4) MED, (5) eBGP over iBGP and (5) lowest IGP weight to egress. If all the previous attributes are the same for two routes, then we break ties by giving preference to the route that is currently in the table [7].

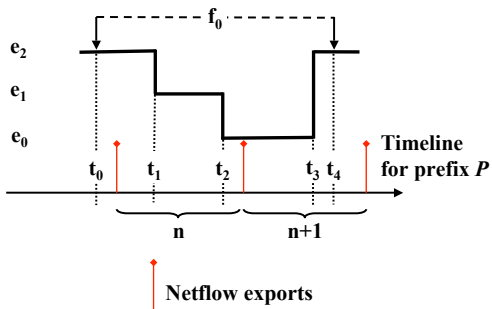
In Abilene this computation is simplified, since iBGP-learned routes are propagated to the monitor, and therefore at each router we use the BGP NEXT\_HOP attribute to find the egress point for each prefix (which does not require any IGP information).

**Filtering BGP path exploration:** BGP path exploration is triggered when an event such as a link failure makes a router explore in sequence alternative paths to reach a certain destination. During path exploration routing changes happen within very short time intervals[13], and therefore in such cases it is not possible to guar-

<sup>3</sup>Since in Geant the BGP and IS-IS collectors are in the same physical machine, there is no concern about timestamps being affected by clock offsets.



**Figure 7: Interval between egress changes for all Abilene routers, January 2006.**



**Figure 8: Example of flow slicing.**

antee an accurate synchronism between the control plane and the data plane on each router. Typically the delay between the time the router sends the BGP update for a prefix  $P$  and the time it updates the FIB for  $P$  is in the order of ms, but in some cases it can be on the order of several seconds. Therefore, in order to mitigate these inconsistencies, we decided to filter out the very fast routing changes. Similarly to [13], we use a relative timeout extracted from empirical data. Figure 6 shows the distribution of the interval between egress changes per prefix for all routers in Geant. We observe that there is a clear cutoff value at  $\sim 40$ s below which most transient egress changes happen. Therefore we select 40s as our timeout value, meaning any two changes spaced by less than 40s will be clustered in a single event, and only the last change of each event will be considered. Figure 7 shows the same distribution for Abilene. Here the sweet spot seems to be  $\sim 70$ s. We believe the difference between these values in the two networks stems from differences in the configuration of the MRAI timer (MinRouteAdvertisementInterval) [14] in the routers inside each network, as well as in the routers along the paths where the BGP routes are received. Once this step is done, all egress changes are passed to the scheduler as shown in Figure 4.

## 4.2 Pre-processing Netflow data

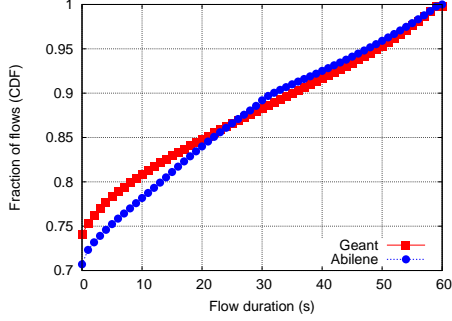
Abilene and Geant sample traffic data by having a collector receiving Netflow records from all routers in the network. As a first step, we checked if the clock of the BGP/IS-IS collectors was synchronized with the Netflow collector, and we verified all machines

were running NTP and synchronized up to the second accuracy. This is important to make sure the scheduler processes events in the correct order. Netflow [2] keeps a record per *flow*, which is defined as a set of packets having the same source/destination IP addresses, source/destination ports, protocol, ToS field and input interface. Netflow records contain important information about each flow such as the start/end times, the number of packets and the amount of bytes transmitted. The advantage of Netflow is that the traffic is collected and stored in a very compact way, allowing to easily collect day-long traces. The caveat is that the traces have the granularity of flows, hiding relevant characteristics such as traffic burstiness.

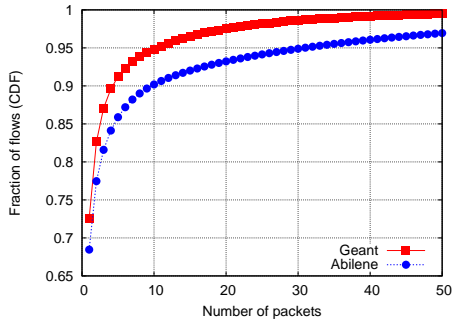
**Removing backbone interfaces:** To compute OD flows we are only interested in measuring the traffic that enters the ingress routers through external links, *i.e.* we are not interested in measuring the traffic that enters the routers through backbone links, since this is the same traffic we are trying to reconstruct in the OD flow formalism. Therefore, we need to exclude from Netflow data all flows that enter the network through backbone interfaces, and only consider those flows coming from external links. Geant only has Netflow running on routers’ external interfaces, but Abilene runs Netflow on a mix of both backbone and external interfaces. In order to filter the backbone interfaces from Abilene, we had to get the ID of each backbone interface through the command “show interfaces”. After matching these IDs with the ones in Netflow data, we were able to discard the backbone traffic. We estimate to have discarded more than 50% of flows per router belonging to backbone interfaces.

## 4.3 Slicing flows

In all of previous work that report studies using Netflow data, flows are processed according to the order in which they are exported. In this subsection we explain why this introduces errors in the traffic matrix computation. In Netflow, flows are ready to be exported to the collector if any of these conditions occur: (1) the flow is inactive for more than 15 seconds (default value); (2) the flow is active for more than 30 minutes (default value); (3) a TCP flag indicates the flow has terminated (*i.e.* FIN or RST flag). Note however that even though Netflow records can be ready to export at any time, they are only exported to the collector at fixed intervals. In Geant this interval is 15 minutes and in Abilene is 5 minutes. This means that the export time of a flow can have a large offset in relation to its starting time (at most  $30+15=45$  minutes), and we can easily envision cases where processing flows according to export times will create errors. Figure 8 shows an example of a flow  $f_0$  that starts at time  $t_0$  and ends at time  $t_4$ . The times in which Netflow cache is exported to the collector are marked in the timeline defining bins  $n$  and  $n+1$ . We can see that even though the flow is exported in bin  $n+1$ , most of its active time was in fact in bin  $n$ , *i.e.* flows are active across time and cannot be condensed to a single instant at the export time. This is specially important to ensure the accuracy of the traffic matrix, since the flow can be affected by multiple routing changes during its lifetime, *e.g.* in Figure 8 flow  $f_0$  starts passing through egress  $e_2$ , then  $e_1$ ,  $e_0$  and  $e_2$  again. Therefore, *each flow needs to be sliced at the points of egress changes, and the contribution of each slice accounted for the OD flow aggregate.* This is a major difference between our work and previous work that do not consider the division of each flow. In our traffic computation method in Figure 4, this is done by the *flow slicer*. The flow slicer starts by doing a longest prefix match of the destination IP address in the flow with a prefix in the ingress-egress table using a PATRICIA tree [12]. Once the prefix match is done, the current egress in use by the prefix is also ex-



**Figure 9: Distribution of flow duration for a single router-day in Abilene and Geant.**



**Figure 10: Distribution of number of packets per flow for a single router-day in Abilene and Geant.**

tracted from the table. The slice information is then written in disk into a trace file for further processing. For the example of Figure 8, the entries for flow  $f_0$  in the trace file would look like:

$$f_0|t_0 - t_1|P|e_2$$

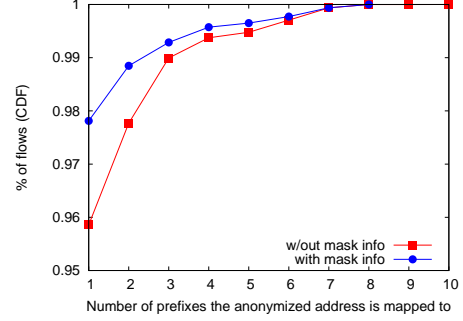
$$f_0|t_1 - t_2|P|e_1$$

$$f_0|t_2 - t_3|P|e_0$$

$$f_0|t_3 - t_4|P|e_2$$

where  $f_0$  represents all the information about the flow (IP addresses, ports, number of packets, number of bytes),  $P$  is the prefix the flow was mapped to, and  $e_0, \dots, e_2$  the egresses used by  $f_0$  during each slice.

The effect of flow slicing should be more significant for longer flows than for short flows, therefore it is important to have an idea of the duration of the flows captured by Netflow. In both Abilene and Geant, Netflow is configured slightly different than usual, and each flow is ready to be exported at fixed intervals of 1 minute. This basically means that the flow duration is mostly determined to the time remaining until the next 1-minute slot. Note that even though flows are marked as ready to export every 1 minute, they are only exported to the collector every 5 minutes in Abilene and 15 minutes in Geant, as already mentioned. Figure 9 shows the distribution of flow duration for a single router-day in Abilene and Geant. We note that between 70% and 75% of the flows start and end in the same second, and  $\sim 90\%$  of flows last less than 30s. In a similar way, looking at Figure 10 we observe that 70-75% of flows



**Figure 11: The impact of anonymization.**

only have 1 packet, and 90% of the flows have less than 10 packets. Note that the higher number of packets per flow in Abilene is due to its higher Netflow sampling rate (10 times higher than Geant). By looking at these values, we do not expect flow slicing to have a significant impact in the traffic matrix computation, however if the Netflow configuration is changed, for instance, the sampling rate is increased or the interval between exports is increased, the effect of slicing can be significant.

**Dealing with anonymized addresses:** In Abilene Netflow data, the last 11 bits of each IP address is anonymized, and therefore additional caution is needed when mapping addresses to prefixes in routing table. With the anonymization, the same address can potentially be mapped to more than 1 prefix, which will break the 1:1 mapping that happens without anonymization. For instance, the anonymized address 10.1.16.0 could either be mapped to the prefix 10.1.16/24 or 10.1.17/24. Sometimes this ambiguity can be resolved by looking at the mask information stored in Netflow data. Without solving this issue, flows can be mapped to the incorrect egress and introduce errors in the traffic matrix. We estimate how often these cases happen by looking at one router-day of Abilene. For each anonymized destination IP address in the Netflow data, we verified in the routing table how many prefixes could have been used to route the packet. Figure 11 shows the distribution of the number of prefixes each anonymized flow can potentially be mapped to. The curve “w/out mask info” shows the outcome without using the mask length info in Netflow, and almost 96% of the flows have a 1:1 mapping. The curve “w/ mask info” shows the result after using the mask length info, and the 1:1 percentage grows to almost 98%. To avoid ambiguous mappings and without sacrificing accuracy, we decided to ignore the 2% of flows that can be mapped to more than a single prefix, and map the remaining flows using both routing table and Netflow mask length information.

#### 4.4 Computing the traffic matrix

Once the trace file is available from the flow slicer, we can finally compute the traffic matrix. The traffic matrix is usually computed in time bins of fixed length, and in this paper we use bins of 1 minute, 5 minutes and 15 minutes. In fact, one of the advantages of flow slicing is that we can reduce the bin size to an arbitrary size. As a first step, time is divided in bins of fixed size, and for each bin we verify what flow slices fall inside of it. We then use the constant throughput assumption, which has been verified in [20] for these time scales. This assumption basically says that if a flow  $f$  of size  $N$  packets and length  $T$  takes a time interval  $t$  inside a bin, than the contribution of  $f$  to the volume aggregate in the bin is given by  $N \cdot \frac{t}{T}$ . Using this rule, we finalize the computation of

the traffic matrix by adding up the contributions of each flow slice inside each bin.

## 5. IMPACT OF ROUTING CHANGES

In this section we develop a metric to quantify the impact of a routing change on traffic. First, the impact of a routing event should be defined over a time window close to the event. If this window is too long, the cause-effect relation between the routing event and traffic becomes weaker, and the impact may be overestimated. If the window is too short, the impact may be underestimated. Since our goal is to ultimately assess the effect of routing changes in traffic matrix variations, the length of the window should be in the same order of the length of the bin used to compute the traffic matrix. This way the effect of a routing change encompasses at most a single variation between two consecutive bins in the matrix, *i.e.* if the bin size is  $\Delta t$ , then the window length should be  $2 \times \Delta t$ . This way the window will cover at least an entire time bin of the traffic matrix, *i.e.* if the event happens in the middle of bin  $n$ , then the window will cover entirely bin  $n + 1$ . Second, the routing impact metric should be measured in units that would help to diagnose anomalous scenarios in the network. For instance, in some cases the impact can be measured in volume (number of packets), in other cases number of flows (*e.g.* source-destination IP address pairs), and in some other cases by real-time traffic impacted (*e.g.* number of VoIP flows). Having these properties in mind, we propose the following definition:

**Routing impact:** the routing impact of an egress change from egress  $e_0$  to egress  $e_1$ ,  $R^{e_0 \rightarrow e_1}$ , is defined as a metric extracted from the set of packets that pass through  $e_1$  in a time window  $2 \times \Delta t$  after the change, where  $\Delta t$  is the bin size of the traffic matrix. For the special case where  $e_1 = \text{null}$ , since there is not any traffic passing through any egress after the change, the routing impact is extracted from the set of packets that were passing through  $e_0$  in a time window  $2 \times \Delta t$  before the change. Note that if there is an egress change at time  $t_0$  and a second change at time  $t_1$  s.t.  $t_0 < t_1 < t_0 + 2\Delta t$ , then the routing impact of the first change is only accounted in the interval  $[t_0, t_1]$ .

Note that we do not define routing impact in any specific units. However for this section and Sections 6.1 and 6.2 we measure it in number of packets for ease of understanding (in Section 6.3 we explore new metrics such as destination IP addresses and destination prefixes). An important characteristic of the routing impact is that for each bin right after the routing event, it gives the amount of traffic that appears or disappears from a certain egress during that bin because of the routing event. We introduce the notation  $R_n^{e_0 \rightarrow e_1}(e_0)$  or  $R_n^{e_0 \rightarrow e_1}(e_1)$  to denote the amount of traffic that disappears from egress  $e_0$  (appear in egress  $e_1$ ) during bin  $n$ , within the window of the event  $e_0 \rightarrow e_1$ . Note that traffic that disappears from an egress has a negative contribution. As an example, from Figure 12, we can extract the quantities described in Table 1. We further define  $R_n(e_0) = \sum_i R_n^{e_0 \rightarrow e_1}(e_0)$  as the total routing impact of bin  $n$  given by the contribution of all the events  $i$  involving egress  $e_0$ . For instance, from Figure 12 we would have  $R_1(e_1) = R_1^{\text{null} \rightarrow e_1}(e_1) + R_1^{e_1 \rightarrow e_0}(e_1) = +1 - 1 = 0$ .

Note that so far we have been referring to events affecting a single prefix  $P$ , but in order to get the total routing impact for a given bin, we need to add the contributions of the impact coming from all the prefixes, which will lead to the following definition:

**Routing impact matrix:** the routing impact matrix is defined

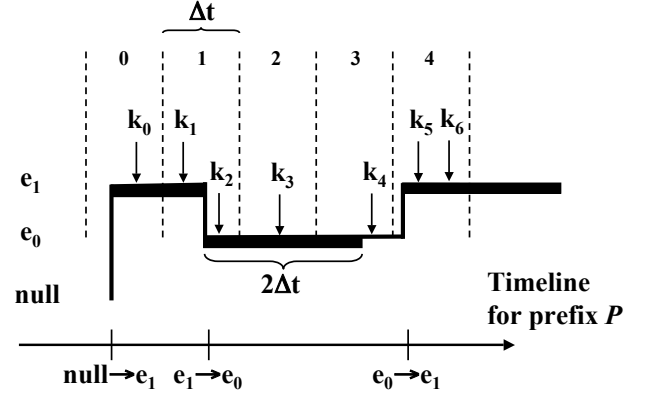


Figure 12: Arrival of packets destined to a prefix  $P$ , and their correlation with egress changes.

Event	Egress	Bin	Packets	Impact
$\text{null} \rightarrow e_1$	$e_1$	0	$k_0$	+1
	$e_1$	1	$k_1$	+1
$e_1 \rightarrow e_0$	$e_0$	1	$k_2$	+1
	$e_1$	1	$k_2$	-1
	$e_0$	2	$k_3$	+1
	$e_1$	2	$k_3$	-1
$e_0 \rightarrow e_1$	$e_0$	4	$k_5, k_6$	-2
	$e_1$	4	$k_5, k_6$	+2

Table 1: Routing impact example from Figure 12.

similarly as the traffic matrix, but instead of the traffic volume flowing between ingress and egress, it has for each bin  $n$  the total routing impact  $R_n$  from an ingress-egress pair, accounting the contributions of all the prefixes.

An important characteristic of the routing impact matrix is that for a given ingress-egress pair, it explains how much of the difference in volume between consecutive bins in the traffic matrix is caused by routing changes. Let  $\Delta V_n = V_n - V_{n-1}$  be the difference in volume between bins  $n$  and  $n - 1$  in the traffic matrix. We can decompose this variation in two components  $\Delta V_n = \Delta V_n^r + \Delta V_n^t$ , where  $\Delta V_n^r$  is the amount of traffic that shifted because of routing and  $\Delta V_n^t$  represents variations in the traffic aggregate that are not related to routing. Given our definition of routing impact, we have  $\Delta V_n^r \simeq R_n$ . Then we can write  $\frac{\Delta V_n^t}{\Delta V_n} \simeq 1 - \frac{R_n}{\Delta V_n}$ , and as the ratio  $\frac{R_n}{\Delta V_n}$  increases up to 1 it means  $\Delta V_n^t \simeq 0$  or if it is above 1, it means  $\Delta V_n^t$  and  $\Delta V_n$  have opposing signs. The natural conclusion is that the higher is the ratio  $\frac{R_n}{\Delta V_n}$ , the more of the variation in volume is explained by routing dynamics. Teixeira *et al.* [19] proposes a similar decomposition of volume variations into routing variations and non-routing variations. However their definition of routing variation is different from our definition of routing impact. We introduce the concept of time window to measure routing impact to make sure there is at least one bin after the routing change that contributes completely to the impact. This ensures that the effect of routing changes does not remain undetected

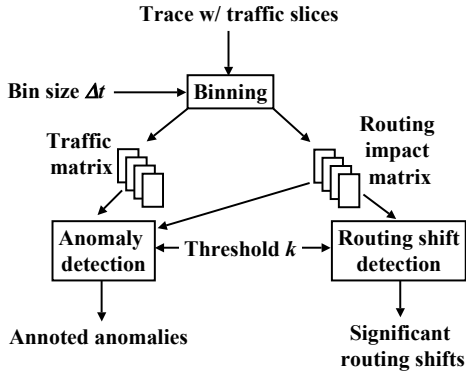


Figure 13: Mixed approach for anomaly detection.

if the change happens just before the end of a bin. Furthermore, differently from [19], we measure the impact only when the affected prefixes are using the egress, making our definition more accurate and more independent of the bin size.

## 6. ROUTING DYNAMICS AND TRAFFIC ANOMALIES

In this section, we start from a set of traffic anomalies given by the Kalman anomaly detection method described in Section 2, and use the routing impact metric to investigate the cause of the anomalies. Then we do the opposite, *i.e.* we start with a set of significant routing events as quantified by the routing impact metric, and we verify how well Kalman detects them. In Figure 13, we present an extension to Figure 4 that includes some of the functional blocks we are describing in this section.

### 6.1 Are traffic anomalies caused by routing?

For a given OD flow  $i$ , the Kalman anomaly detection is based on the value of the innovation at each bin  $n$ ,  $|\eta_n(i)|$ . If this value is above a certain threshold  $k \cdot \sigma_i$ , then an anomaly is signaled. The innovation is basically a low pass filter on the volume variations of each OD flow. An anomaly is triggered by a significant increase or decrease in the traffic volume that causes the innovation to cross the detection threshold. Once we find the root-cause of the sudden volume variation, we can also explain what caused the anomaly. As we saw in Section 5, the routing impact metric quantifies how much of the volume variations between consecutive bins in the traffic matrix is originated by routing dynamics. We propose the following definition:

**Routing-induced anomaly:** we define a bin  $n$  of an OD flow  $i$  having a routing-induced anomaly if (1)  $|\eta_n(i)| > 4\sigma_i$ , and (2)  $\frac{R_n}{\Delta V_n} > 0.5$ , where  $R_n$  is the routing impact of the flow in bin  $n$  and  $\Delta V_n$  the variation in volume between bin  $n - 1$  and  $n$  for the same flow. The first condition guarantees that Kalman signals an anomaly in the bin using detection threshold  $k = 4$  (as described in Section 2), while the second condition ensures that the majority of the volume variation is originated by routing dynamics. Note that even though the second condition does not guarantee that the anomaly could be caused solely by routing, the high percentage of routing impact in the volume variation is a strong indication that this could be the case.

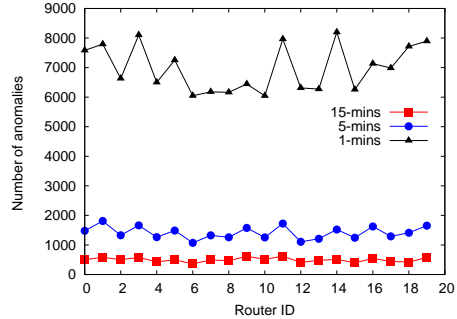


Figure 14: Number of Kalman anomalies in Geant, for different bin sizes.

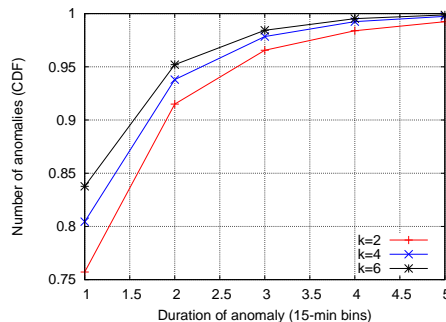
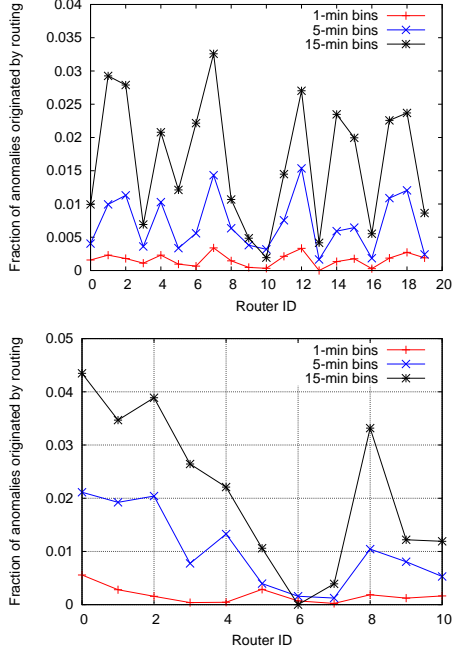


Figure 15: Number of bins per anomalous region in Geant using 15-minute bins.

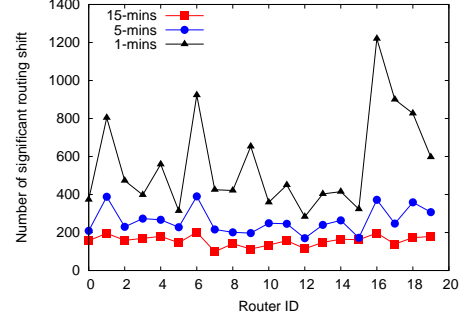
As a first step to nail down the routing-induced anomalies, we run Kalman in all the OD flows of Geant and Abilene with detection threshold  $k=4$ . Figure 14 shows the number of anomalous bins captured in Geant using bin sizes of 1 minute, 5 minutes and 15 minutes for each ingress router. We note that the number of anomalies roughly grows in the inverse proportion of the bin size, *i.e.* if the bin size doubles, the total number of anomalous bins is halved. We obtained a similar result for Abilene. This indicates that traffic anomalies persist in different time scales, *i.e.* as the bin size grows, multiple anomalies are condensed into single anomalous bins. Note that there can be cases in which an anomaly spans across multiple consecutive bins, defining an *anomalous region*. Since we measure the routing impact in a window that spans at most 2 bins, we need to be careful when analyzing the routing impact of anomalous regions that have more than 2 bins. For instance, we could have a scenario of 3 consecutive anomalous bins  $n, n + 1, n + 2$  that have a high innovation triggered by a routing event at bin  $n$ . In this case, only bins  $n$  and  $n + 1$  at most will be labeled as routing-induced anomalies, even though the anomaly in bin  $n + 2$  was also triggered by a routing change that happened in bin  $n$ . Figure 15 shows the distribution of the number of bins per anomalous region using 15-minute bins in Geant for different detection thresholds  $k = 2, 4, 6$ . We observe that for the detection threshold we are using ( $k = 4$ ), more than 80% of the anomalous regions only have a single bin, and almost 95% of the regions have at most 2 bins. Therefore, in order to simplify the analysis and without sacrificing too much the accuracy, we make a bin by bin processing instead of doing an analysis per anomalous region.



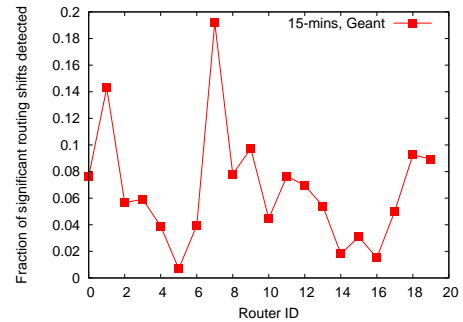
**Figure 16: Fraction of Kalman anomalies that are routing-induced in Geant (top) and Abilene (bottom).**

Once we have the set of anomalous bins given by Kalman, we signal the bins that have a ratio  $\frac{R_n}{\Delta V_n} > 0.5$  as per our definition of routing-induced anomalies. Figure 16 shows for Geant (top) and Abilene (bottom) the percentage of anomalous bins that are routing-induced, per ingress router. First, we observe that the percentage of anomalies that can be associated with routing is very small, less than 5% in both networks. This is in accordance with previous results [15, 5] that show that prefixes that carry more traffic tend to be more stable. Second, we note that as the bin size increases, this percentage also increases, and we verified that the set of routing-induced anomalies did not change significantly for different bin sizes. This is because (1) the routing dynamics originating the statistical anomalies are sufficiently spaced in time such that they are not clustered in single anomalous bins as the bin size increase, and (2) the routing impact in these bins stands out from the rest of the volume mix, even in larger time scales. On the other hand, we also find cases where by increasing the bin size, the variation in volume between bins is smoothed such that they are no longer detected by Kalman. This happens because as the bin size is increased, short-lived volume variations become hidden by the aggregate in the wider window. For instance, in Abilene, the OD flow from Kansas City to Denver was flagged with a single routing-induced anomaly using 5-minutes bins, which is indicated in Figure 16 (bottom) at Router ID=6 (Kansas ingress). Using 5-minute bins, the relative volume variation for this anomaly was  $\frac{\Delta V_n}{V_{n-1}} = 5$ , but using 15-minute bins, the corresponding bin had a relative variation of just 1.5, and thus was left undetected by Kalman. However, for the time scales we used, cases like this are more the exception rather than the norm.

**Summary:** Only a very small percentage (< 5%) of the statistical anomalies found in OD flows of Geant and Abilene are induced by routing. The same set of routing-anomalies persist across different bin sizes, indicating they occur spaced in time and are well



**Figure 17: Number of significant routing shifts in Geant for different bin sizes.**



**Figure 18: Fraction of significant routing shifts detected by Kalman in Geant using 15-minute bins.**

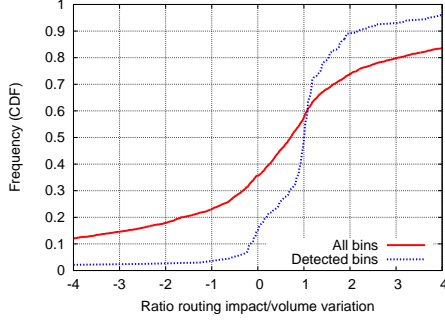
defined in terms of routing impact measured in traffic volume.

## 6.2 Does Kalman detect routing anomalies?

In the previous subsection, we analyzed to what extent routing dynamics induce statistical anomalies in the traffic volume of OD flows. In this subsection, we do the reverse analysis, and assess to what extent routing changes with significant impact are captured by statistical anomaly detectors such as Kalman. First we need to define what routing changes are significant, or where do we draw the line to determine what values of  $R_n$  stand out. Fundamentally, the value of  $R_n$  can be seen as a difference or a variation in traffic volume originated by routing. Therefore, it seems fair to use a similar approach as Kalman, and normalize it by its standard deviation. In this way, the highest variations within a same OD flow will stand out and will be captured as outliers. More exactly, we propose the following definition:

**Significant routing shift:** for a bin  $n$  of an OD flow  $i$ , we define *routing shift* as the routing impact of the bin, *i.e.*  $R_n(i)$ . A *significant* routing shift is defined s.t.  $|R_n(i)| > 4\sigma_i$ , where  $\sigma_i$  is the standard deviation of the routing impact for OD flow  $i$ . We pick  $k = 4$  here in order to be consistent with Kalman.

In order to find the significant routing shifts, we ignore those OD flows that have less than 30 bins with  $|R_n(i)| > 0$ , since in these cases the standard deviation does not have enough samples to be statistically meaningful. Figure 17 shows the number of significant routing shifts in Geant for different bin sizes. We observe

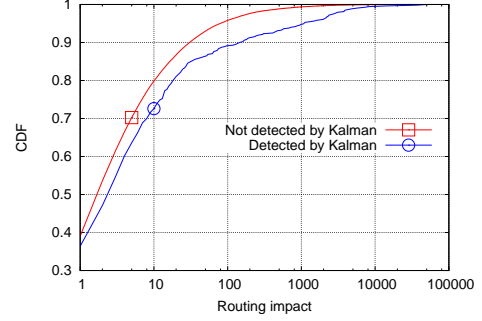


**Figure 19: Distribution of ratio  $\frac{R_n}{\Delta V_n}$  for bins with significant routing shifts in Geant using 15-minute bins.**

that even though for 5 and 15 minutes the number of routing shifts per router is similar, for 1-minute bins the numbers are higher. We obtained similar results for Abilene. This is because as the time scales increase, the values of the impact are smoothed out by aggregating traffic in wider windows. In a second step, we verified to what extent Kalman detected the significant routing shifts, using the same detection threshold  $k = 4$  used before. Figure 18 shows the fraction of significant routing shifts flagged by Kalman for each ingress of Geant using 15-minute time bins. We obtained similar results for Abilene and with other bin sizes. We observe that less than 20% of the shifts are detected by Kalman, and for most routers it is less than 10%.

In order to understand why Kalman does not detect most of the routing shifts, we look at the ratio  $\frac{R_n}{\Delta V_n}$ , which gives an indication of the weight of routing in the volume variation. If  $0 < \frac{R_n}{\Delta V_n} < 1$ , then it means that  $|\Delta V_n| > |R_n| > 4\sigma$ , and therefore is very likely that Kalman triggers an anomaly. On the other hand, if  $\frac{R_n}{\Delta V_n} > 1$ , it means the routing shift was somehow compensated by other traffic that left or entered the egress, and less likely to be noticed by Kalman. Figure 18 shows the distribution of  $\frac{R_n}{\Delta V_n}$  for all bins with significant routing shifts (“All bins”) and for those bins detected by Kalman (“Detected bins”). We observe that when looking at all bins, the ratio  $\frac{R_n}{\Delta V_n}$  is quite spread out, and only about 20% of the bins have the ratio in the interval  $[0,1]$ . But when looking at the detected bins, we observed that the ratio is mostly within the range  $[0,1]$  or close to this interval, which explains why these bins were detected by Kalman. This is an important observation that we would like to emphasize, *i.e. conventional anomaly detectors such as Kalman are unable to detect most of the significant routing shifts because of other volume variations in the aggregate that hide the effect of routing changes.*

Since Kalman misses so many significant routing shifts, it is only fair to speculate which types of routing shifts it *does* find. We take all bins containing routing shifts, and compare the CDFs of the routing impact for (1) bins containing Kalman anomalies, and (2) bins where Kalman fails to trigger an alarm. Figure 20 shows that the routing shifts for which Kalman detects an anomaly are generally larger (measured by the routing impact) than the ones that Kalman misses. This is characterized by the CDF of detected routing shifts lying completely to the right of the CDF of missed routing shifts. This result holds qualitatively also for Abilene, and for different bin sizes and Kalman threshold. This is a positive result since the largest routing changes are likely the ones that an operator would care about. Note however, that some large routing



**Figure 20: Routing changes detected by Kalman have higher routing impact than those left undetected; Geant, 15-min bins,  $k=4$ .**

shifts are still missed by Kalman. This happens when these routing shifts get diluted in the aggregate volume change of the OD pair, as explained earlier. In the next section we explore the use of traffic metrics other than volume to detect those routing shifts. Recall that even though so far we have been talking solely of traffic in terms of volume, the scheme of Figure 13 is general enough to work with any other metric extracted from traffic, such as number of flows, number of prefixes or other special type of traffic such as VoIP.

**Summary:** From all the significant routing shifts present in the OD flows of Geant and Abilene, Kalman detects less than 20% of them. The inability of Kalman to detect most of these routing shifts is caused by other volume variations in the aggregate that hide the effect of the routing changes. However, the routing changes that it does find tend to be larger than the ones it misses.

### 6.3 Detecting routing anomalies using other traffic metrics

So far we have been measuring traffic in terms of volume and we have observed that only a small fraction ( $< 5\%$ ) of the volume anomalies are originated by routing events. Therefore, it is clear that the packet count metric is not aligned with routing disruptions, *i.e.* most of volume shifts are not caused by routing. In this section we explore new metrics other than volume to isolate those anomalies that are originated by routing events.

It is important to have a method that infers routing-induced anomalies just based on traffic information, because then the analysis of anomalies is independent of routing data and the processing is much simplified. Furthermore, since such method will not depend on routing data, we could use it to infer the cases a) and d) of Figure 2. We consider two new metrics: destination IPs and destination prefixes (or prefixes). Routing changes happen at prefix granularity, and cause prefixes to shift between different egresses (or to appear/disappear from egresses). Therefore, we expect to see a much better correlation to routing events using destination IPs and prefixes. In fact the correlation should increase as we go from packets  $\rightarrow$  destination IPs  $\rightarrow$  prefixes, *i.e.* as we approach the granularity level where routing changes happen. On the other hand, the absolute number of anomalies should increase by the order prefixes  $\rightarrow$  destination IPs  $\rightarrow$  packets, since it is easier to have larger variations when dealing with smaller aggregation levels. For example, a shift of 1,000 packets may correspond to a shift of a single prefix.

Figure 21 shows the total number of Kalman anomalies using the three different metrics for different routers of Geant. As ex-

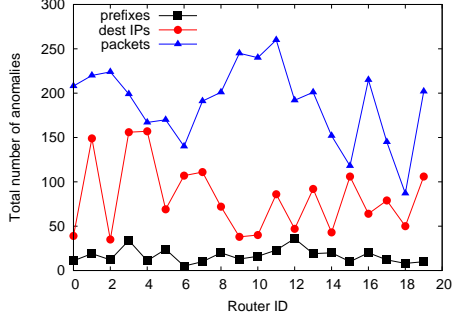


Figure 21: Total number of Kalman anomalies for different metrics, Geant,  $k = 6$ , 15-min bins.

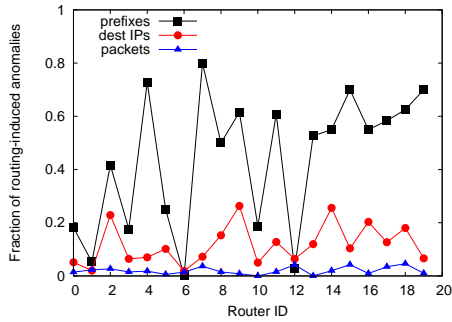


Figure 22: Fraction of routing-induced anomalies for different metrics, Geant,  $k = 6$ , 15-min bins.

pected, the total number of anomalies increases when going from prefixes to packets. Figure 22 shows the fraction of anomalies that are induced by routing in each metric. Clearly, when using prefixes, it is much more likely that the detected anomaly is caused by routing. An anomaly in the prefix metric means there was a significant number of prefixes  $\Delta p$  that switched to a new egress (or appear/disappear in the egress), which is precisely the fingerprint of a routing event. Note that because we are only correlating these anomalies with routing data sent from downstream, we are only capturing the cases b) and c) of Figure 2. Therefore, it is possible that the remaining anomalies in prefixes are caused by cases such as a) and d). This is something we plan to verify as part of our future work.

We now look at the intersection of the anomalies detected by the three metrics. Our goal is to find out whether a single detector can pinpoint most the routing changes, or else we need to understand which types of routing events are detectable by a metric but not by the others. Table 2 shows the chances of capturing an anomaly in metric  $y$ , knowing that there was an anomaly detected with metric  $x$ . We term this quantity  $P(y|x)$ . We note that most anomalies in volume (packets) do not have an associated anomaly in destination IPs or prefixes (first row of the table). This is because these anomalies generally do not involve a significant shift of destination IPs of prefixes. For example, in the OD pair between Athens and London, we were able to find a routing shift of almost 4,000 packets involving only four destination IPs in two prefixes. One possible explanation is that one of these IPs may host a popular server. On the other hand, most of prefix anomalies do not have associated a

$\downarrow x, \rightarrow y$	Packets	Dest. IPs	Prefixes
Packets	–	0.05	0.01
Dest IPs	0.11	–	0.15
Prefixes	0.17	0.72	–

Table 2: Values for  $P(y|x)$ , where  $x$  and  $y$  are Kalman anomalies; Geant, 15-min bins,  $k=6$ .

$\downarrow x, \rightarrow y$	Packets	Dest. IPs	Prefixes
Packets	–	0.35	0.27
Dest IPs	0.17	–	0.69
Prefixes	0.16	0.86	–

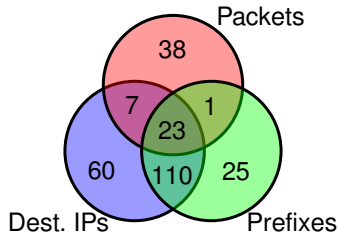
Table 3: Values for  $P(y|x)$ , where  $x$  are routing-induced anomalies and  $y$  are Kalman anomalies; Geant, 15-min bins,  $k=6$ .

volume anomaly, but they are associated with an anomaly in destination IPs (last row). This is because there are events where several prefixes are involved, but the shift in volume involved is not enough to trigger a Kalman anomaly. For example, in the OD pair between Paris and New York, we observed a shift involving over 600 prefixes and 1,100 destination IPs, but each of those IPs received on average little more than one packet. These observations are consistent with the results reported in [15, 5], where the authors show that routing changes happen more often to destinations that receive less traffic.

Table 3 shows similar results as Table 2, but this time  $x$  are routing-induced anomalies. Note that the numbers in the first row are significantly higher than in the previous table. This is expected, since now we are only looking at routing-induced anomalies in  $x$ , and it is much more likely that these types of anomalies create shifts in number of destination IPs and prefixes (in  $y$ ). Also note that routing-induced anomalies in destination IPs are very likely to be caught as anomalies in prefixes (second row) and vice-versa (third row).

To better understand these results, we show in Figure 23 a Venn diagram with the routing induced anomalies in the three metrics. Despite the intersecting regions, the figure shows that some anomalies are found only in specific metrics. For instance, 105 routing anomalies are not detected on prefixes. We analyzed the routing impact during these anomalies, and verified that they correspond to egress changes involving small numbers of large prefixes. Conversely, the 25 routing changes found only on prefixes involve several prefixes simultaneously advertised or withdrawn, each with a small number of packets and active destination IPs. While Figure 23 shows a diagram for routing anomalies in Geant with 15-minute bins and Kalman threshold 6, we have observed similar patterns in the diagrams for Abilene as well as for different parameter values.

Even though no single metric clearly outperforms the others, we can draw some recommendations based on our results. If one wishes to be conservative and avoid dealing with false positives (*i.e.*, anomalies unrelated to routing), the prefix metric is the best choice, according to Figure 22. However, if the operator is ready to cope with other sorts of anomalies, then taking the union of the three metrics increases also the total number of routing changes detected by Kalman. Part of our future work is to study ways of reducing both false positives and false negatives, so that we can exactly pinpoint the routing changes based only on traffic information (no routing messages).



**Figure 23: Venn diagram of routing induced anomalies found by the three metrics; Geant, 15-min bins,  $k=6$ .**

## 7. RELATED WORK

Traffic anomaly detection has recently received a great deal of attention in the research literature, and numerous techniques have been evaluated to detect outliers in traffic timeseries, including Wavelets [6], Fourier transforms [8, 22], Kalman filters [17], and PCA [9]. Even though the goal of these techniques is to improve the detection of the effect of the anomalous events, the lack of ground truth information poses a challenge for their thorough evaluation. In this paper, we contribute to close this gap by linking the occurrence of routing events to respective traffic variations. To best of our knowledge, our work is the first to propose a method to automatically infer the cause of those statistical anomalies that are induced by routing changes.

Several previous work focused on studying the effect of routing dynamics in traffic. One of the first studies to address this issue is the one of Rexford *et al.* [15], where they found that a small number of prefixes is associated with most of the BGP updates and that most traffic travels to a small number of stable prefixes (the popular destinations). Even though that study provides insights about how routing stability is related to the amount of traffic carried in the network, it does not attempt to quantify the impact of each routing event on traffic. A later work by Agarwal *et al.* [5] proposes a technique to measure the impact of routing changes by comparing traffic matrices constructed from updated routing tables to those reconstructed from stale routing tables. However, the prefix-egress mapping is only used at periodic intervals of 20 minutes, which may hide the effect of routing dynamics occurring between lookups. Teixeira *et al.* [19] proposes a method to compute the traffic matrix where the time between prefix-egress lookups is reduced to 10 minutes. They show that routing events are responsible for the largest traffic variations, but they limit their analysis to packet count. Even though their method improves over previous methods, it suffers from the binning effect of Netflow as described in Section 4. In contrast, we propose in this paper a method to compute the traffic matrix with up-to-second accuracy, by sorting/slicing flow records as egress changes happen. In addition, we also introduce a routing impact metric that quantifies exactly the amount of traffic shifted as a result of a routing change.

## 8. CONCLUSION

In this paper we describe a method to detect the traffic anomalies induced by routing, as well as a way of capturing routing shifts based on traffic information. Even though statistical anomaly detectors based on traffic volume do a good job in detecting anomalies associated with large changes in the aggregate, they are not particularly suited to detect routing changes. In one hand, they fail to trigger alarms for routing shifts when they are either too small or diluted in the midst of large traffic changes. On the other

hand, routing changes are but a small fraction of the total output of these detectors. Our long term goal is to develop a method to accurately detect the routing-induced anomalies just based on traffic information, *i.e.* without using routing messages. This will remove the burden of having to collect, process and store the routing data, and correlate it with traffic data. Furthermore this would enable to infer the traffic variations originated by the cases (a) and (d) in Figure 2. Towards this goal, Netflow v9[4] already includes a field `bgp-next-hop` in each record that specifies the next-hop BGP peer, which enables one to build the traffic matrix without the need of routing messages. Our results using alternative metrics such as destination IP addresses and destination prefixes are promising, and show that Kalman can be tweaked to achieve this goal. Our work is just a first step towards stimulating the search for anomaly detectors that can not only detect, but also automatically pinpoint the root cause of the anomaly with high confidence. Even though in this paper we focus on routing-induced anomalies, other type of anomalies which inference benefits from out-of-band information are yet to be explored.

## 9. REFERENCES

- [1] Abilene data. <http://noc.net.internet2.edu>.
- [2] Cisco ios netflow. [http://www.cisco.com/en/US/products/ps6601/products\\_ios\\_protocol\\_group.html](http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group.html).
- [3] Geant data. <http://rtmon.gen.ch.geant2.net>.
- [4] Netflow v9 export format. [http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123\\_1/nfv9expf.htm](http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123_1/nfv9expf.htm).
- [5] S. Agarwal, C. Chuah, S. Bhattacharyya, and C. Diot. Impact of bgp dynamics on intra-domain traffic. In *ACM SIGMETRICS*, 2004.
- [6] P. Barford, J. Kline, D. Plonka, and A. Ron. A signal analysis of network traffic anomalies. In *ACM IMW*, 2002.
- [7] E. Chen and S. Sangli. Avoid BGP Best Path Transitions from One External to Another. RFC 5004, Internet Engineering Task Force, September 2007.
- [8] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen. Sketch-based change detection: methods, evaluation, and applications. In *ACM IMC*, 2003.
- [9] A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. In *ACM SIGCOMM*, 2004.
- [10] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. *ACM SIGCOMM*, 2005.
- [11] J. Mai, C.-N. Chuah, A. Sridharan, T. Ye, and H. Zang. Is sampled data sufficient for anomaly detection? In *ACM IMC*, 2006.
- [12] D. R. Morrison. Patricia - practical algorithm to retrieve information coded in alphanumeric. *J. ACM*, 15(4):514-534, 1968.
- [13] R. Oliveira, B. Zhang, D. Pei, R. Izhak-Ratzin, and L. Zhang. Quantifying Path Exploration in the Internet. In *ACM IMC*, 2006.
- [14] Y. Rekhter, T. Li, and S. Hares. Border Gateway Protocol 4. RFC 4271, Internet Engineering Task Force, January 2006.
- [15] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang. Bgp routing stability of popular destinations. In *ACM IMW*, 2002.
- [16] H. Ringberg, A. Soule, J. Rexford, and C. Diot. Sensitivity of pca for traffic anomaly detection. *ACM SIGMETRICS*, 2007.
- [17] A. Soule, K. Salamatian, and N. Taft. Combining filtering and statistical methods for anomaly detection. In *ACM IMC*, 2005.
- [18] A. Soule, F. Silveira, H. Ringberg, and C. Diot. Challenging the supremacy of traffic matrices in anomaly detection. In *ACM IMC*, 2007.
- [19] R. Teixeira, N. Duffield, J. Rexford, and M. Roughan. Traffic matrix reloaded: impact of routing changes. In *Passive Active Measurement Conference*, 2005.
- [20] J. Wallerich, H. Dreger, A. Feldmann, B. Krishnamurthy, and W. Willinger. A methodology for studying persistency aspects of internet flows. *SIGCOMM Comput. Commun. Rev.*, 35(2), 2005.
- [21] J. Wu, Z. M. Mao, J. Rexford, and J. Wang. Finding a needle in a haystack: pinpointing significant bgp routing changes in an ip

network. In *USENIX NSDI*, 2005.

- [22] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan. Network anomography. In *ACM IMC*, 2005.