STRID: Scalable Trigger-based Route Incidence Diagnosis

Feng Wang[†], Lixin Gao[‡], Oliver Spatscheck[§], and Jia Wang[§]
[†]School of Engineering and Computational Sciences, Liberty University, Lynchburg, VA 24502
[‡]Department of ECE, University of Massachusetts, Amherst, MA 01002
[§]AT&T Labs-Research, Florham Park, NJ 07932

***Abstract*—As the Internet steadily increases in importance, it is still based on a quite fragile routing design. From network operators perspective it is therefore crucial to detect end-to-end path performance due to routing outages early to either mitigate them directly or contact other entities to mitigate them. In this work we demonstrate the feasibility of a real-time tool for detecting degraded forwarding performance due to routing problems. Our tool passively monitors the traffic within the network and actively probes paths for which the TCP traffic characteristics indicate a possible routing problem. More importantly, our tool focuses on detecting routing events that actually affect network traffic, which from the network operators' perspective is most relevant. The experimental results based on large-scale measurement in the Internet indicate that our tool effectively detects a significant number of routing outages and forwarding loops.**

I. Introduction

As the Internet becomes the critical information infrastructure for both personal and business applications, it is crucial that the end-to-end Internet paths have high availability, which are still based on a quite fragile routing design. Empirical studies have shown that degraded end-to-end path performance can be correlated with routing dynamics [7], [11], [3], [8]. Furthermore, end-to-end Internet path failures are widespread [3], [8], [9]. End-to-end path failures are typically attributed to either congestion or routing dynamics. Comparing with short-lived congestion, Internet path failures correlated with routing changes can last for a few seconds or tens of seconds. In addition, during route changes, end-to-end paths may experience packet losses, even though the destinations are not disconnected [14], [15]. Thus, it is critical to make fast and accurate detection for those end-to-end forwarding issues. This motivates us to develop a framework for online detection of forwarding performance issues due to routing outages along specific end-to-end Internet paths. This system can help network operators to detect failures early to either mitigate them directly or contact other entities to shorten the time to recovery from failures or to minimize performance degradation.

In this paper, we focus on the problem of forwarding performance issues due to routing outages along specific end-to-end Internet paths. We define *routing outages* as loss of routes at routers along end-to-end paths, or having invalid routes, such as routing loops. We use routing outages to generalize all events that will finally impact reachability at forwarding plane.

Detection of forwarding plane performance degradation due to routing outages is a challenging task because failure events appear without predictive patterns. Currently network operators rely on monitoring the control plane to detect unusual routing outages. First, even if the control plane is monitored, it is still unclear which changes in the control plane would cause actual problems in the forwading plane. Another challenge of using routing changes is that routing failures may not be reflected in routing updates [3], particularly if the vantage point of the network operator observation point is far from the source of the failure. Additionally, some routing failures, for example, forwarding loops, might not be observed from the control plane at all.

In this work, we develop a realtime failure diagnosis system called STRID, which helps network operators to detect Internet path failures more quickly to improve their response time, and therefore reduce the duration of outages on the Internet. STRID passively monitors the traffic on major links of an ISP. If the passive traffic indicates that there might be a problem, then STRID automatically and quickly initiates active probing to determine if the issues observed in the passive traffic is routing related and where the problem occurs. To avoid overloading target prefixes with probing, which might be considered as DoS attacks by remote hosts, STRID limits the amount of probing sent for each prefix. Similar to our system, PlanetSeer [17] is a system to locate Internet anomalies by using TTL and four retransmissions as indicators of Internet anomalies. The key difference between STRID and PlanetSeer is that STRID monitors only *unidirectional* traffic flows. This is an important feature since TCP flows in the data server can be asymmetric, which makes the capture of bidirectional traffic flow information challenging. Furthermore, the essence of our heuristic is that STRID triggers active probing if we observe a sufficiently long period of TCP retransmissions for all TCP flows that share a common path.

We deploy STRID in a major link of a tier-1 ISP. STRID is able to capture a significant number of routing outages. For example, STRID discovers that there are hundreds of forwarding loops per day. About 29% of forwarding loops are persistent loops with a duration of more than one hour. Note that it is difficult to discover these forwarding loops with the current control plane monitoring system. Our measurement shows STRID can detect all BGP events that occur during our measurement period and impact forward-plane traffic. In addition, STRID is also able to detect repeatedly occurring

978-1-4244-2390-3/08/$25.00 ©2008 IEEE

routing issues. About 66% of the prefixes experience routing outages on at least two different days during our measurement period. In the worst case, about 5% of destination prefixes experience repeatedly occurring routing outages every day. This implies that STRID can help network operators to discover and avoid those repeated events that are most likely related to route flapping.

STRID has a number of important benefits. First, STRID only detects routing outages if traffic of the ISP is actually impacted. This is important in the sense that network operators have a limited amount of resources to track down routing issues. Second, STRID greatly reduces the amount of active probing. This is important because it allows us to monitor all locations for which we see traffic and probe only when there is likely to be a problem. As we do not claim that STRID alone will be sufficient to allow Internet network operators to achieve 99.999% uptime, we believe that it will be an important step toward this goal.

The remainder of the paper is structured as follows. Section II overviews the design of STRID. We describe methodology details and experimental setup in Sections III and IV. Sections V and VI demonstrate the performance of STRID through Internet deployment and measurement, and experimental results analysis. In Section VII, we discuss the related works. We conclude in Section VIII.

II. Design Overview of STRID

The major goals of STRID are to detect degraded end-to-end path performance on the data plane, and to launch active probing to measure the reachability of destinations when passive monitoring reflects the symptoms of routing outages. The probe results are then analyzed to infer the route state of the paths. In particular, STRID consists of two parts: (1) passive monitoring and (2) active probing. Passive monitoring is used to fire trigger of active probing. We will discuss those parts in turn.

A. Passive Monitoring

One of the challenges of identifying potential routing issues by passively monitoring is to choose the appropriate performance metric that can be used to distinguish routing failures from other factors, such as congestion. In this paper, we use TCP retransmission, which indicates packet loss, to monitor degraded performance.

After selecting TCP retransmission to monitor, we face the question "How can TCP retransmission distinguish packet loss due to routing outages from that caused by congestion?" To answer this question, we determine if retransmissions are caused by routing outages based on the following two key observations:

1) TCP typically recovers from congestion caused loss more quickly than from routing outages.
2) Routing outages should affect *all* TCP flows sharing a given path at the same time, while congestion often affects different TCP flows at different time.

STRID uses the two observations to filter out TCP retransmissions that are suspected due to congestion. In Section III, we will show how to monitor TCP retransmissions and trigger active probing. Note that TCP connections might synchronize their recovery, which might lead to synchronized packet loss in absence of routing failures [10], [4]. However, a routing failure will definitely impact all flows that share a failed path, while congestion does not necessarily affect all flows.

One interesting question is how to decide which TCP flows share a common path. In this regard, STRID utilizes the fact that Internet routing is performed at the granularity of BGP destination prefixes. A BGP destination prefix represents a group of TCP endpoints that are topologically close and under common administrative control. Measurement studies [5] have shown that aggregation on destination prefixes can identify paths with high accuracy. Therefore, STRID collects TCP flow information by using 4-tuple flow identification $(srcip, destip, srcport, destport)$ and uses the BGP routing prefixes to cluster TCP flows which likely share a common path.

B. Active Probing

After STRID identifies suspected symptoms of routing issues by monitoring a set of TCP flows to the same destination prefix, it triggers a set of probing packets to a selected destination. We select one destination address of the suspected flows destined to the prefix, and send probing to measure its reachability. STRID uses probing trains to detect a possible routing problem. We design a *probing train* as a sequence of UDP packets. For each probing train, a set of back-to-back UDP packets are sent to the destination with increasing TTL values until reaching the maximum TTL value. In our measurement, we select 30 hops as maximum TTL value. To probe the reachability of a destination, three probing trains are sent to facilitate diagnosis process in understanding if there is a routing problem as well as the location of the problem. To avoid overloading particular prefixes with probing packets, which might be considered as DoS attacks by remote hosts, we rate limit the amount of probing per prefix and set the interarrival time between a pair of probing trains as one second, in order to avoid congestion caused by probing trains.

C. Limitations

We emphasize several limitations of out diagnosis system:

1) STRID does not monitor traffic other than TCP. However, majority of today's Internet traffic uses TCP as its transport layer protocol. For example, 90% of traffic carried by in a large tier-1 ISP is over TCP in April 2006.
2) STRID cannot guarantee to capture short-lived routing problems. When a failure causes a TCP flow to enter retransmission state, STRID will probe the destination to examine the reachability. If the failure recovers quickly before active probing is triggered, STRID fails to detect the failure because probing packets can reach the destination. In this paper, we focus on *prolonged* failures that last long enough to impact traffic carried along the path.

3) We assume that triggered active probing can always return the network state. If we observe losses in all three probe trains, we conclude that there is a routing failure. However, congestion might lead to the same observation. In this case, our method will incorrectly attribute packet loss as due to routing failures. In this paper, we set the interarrival time between a pair of probing trains as one second to avoid congestion caused by probing trains.
4) We use BGP routing prefixes to group TCP flows that may share a common path. However, when a prefix is split into several different subnetworks, which may not be shown in routing table because of prefix aggregation, STRID will fail to identify routing failures occurring within those subnetworks. One method to overcome this limitation, which is our future work, is to split prefixes with /8 or /16 length into several subnets with /24 length.

III. Active Probing Trigger

A TCP flow can experience retransmission timeout several times. The retransmission timeout is increased when a packet is retransmitted. We define the duration of the sequence of retransmissions as *retransmission duration*, and denote D_s as the retransmission duration to a destination s. As we described in previous section, STRID groups TCP connections based on BGP routing prefixes. We use C_p to represent all active TCP flows belonging to a prefix p.

STRID triggers active probes for a destination belonging to a prefix p if retransmission duration of *all active flows* (C_p)

$$D_{C_p} \geq T_{thre} = T_{ret} + T_{recover}$$

where T_{thre} is a threshold, which contains two elements: (1) T_{ret} is a threshold for retransmission duration, and (2) $T_{recover}$ is a threshold for recovery duration. T_{ret} is used to detect whether a packet retransmission might have one or multiple timeouts as soon as possible. Based on the threshold, we can detect if a suspected retransmission can last long enough to let us launch active probing to identify the failure. In the next section, we will show how we determine the threshold, T_{ret}. If a TCP flow-group already lasts longer than T_{ret}, STRID keeps monitoring the flow-group to ensure that the retransmitted flow is not recovering from failures when STRID sends active probes. Thus, the delayed probing waits for $T_{recover}$ seconds to make sure that the retransmitted packet is indeed lost. In the next section, we will discuss how to determine this delay.

In summary, we propose a mechanism to trigger active probing based on trigger steps described above. Fig. 1 summarizes the decision process. To detect packet retransmissions, STRID maintains several variables, $sendseq$, $retrandur$ for each TCP connection, and $flowcount$ for each flow-group. $sendseq$ is the sequence number of the most recently sent packet. $retrandur$ is the duration of a packet retransmission, and $flowcount$ is the number of flows entering retransmission state. We use $currentseq$ to represent the sequence number of the packet currently being sent. If $currentseq > sendseq$, the flow is making progress and we reset $flowcount$ and $retrandur$ to 0 and set $sendseq = currentseq$. Otherwise, the packet is retransmitted, and $flowcount$ is incremented by

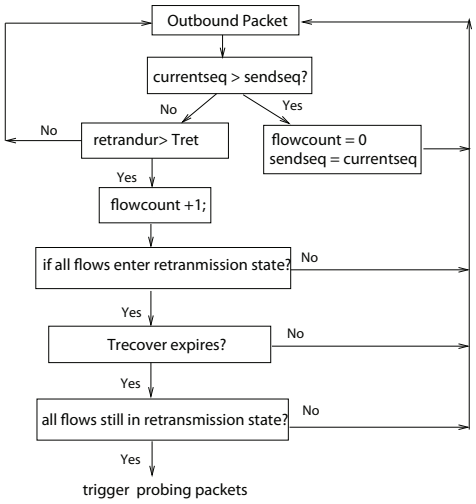

Fig. 1. Heuristic for triggering active probings for a prefix.

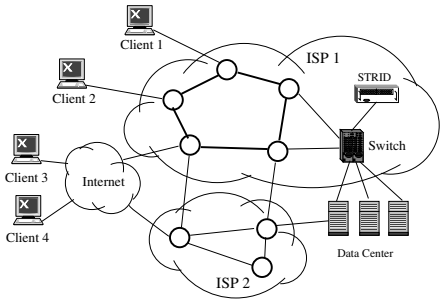

Fig. 2. Placement of STRID.

1. STRID keeps track of the first sending time of each packet. If the current packet is retransmitted, $retrandur$ is derived by the difference between the current retransmission time and the stored first transmitted time.

IV. Experimental Setup

In this section, we describe our measurement setup, and the data sets used in our experiments. We then determine the thresholds in our experiments.

A. Setup

We deployed STRID in a data center connected to a tier-1 ISP network containing hundreds of edge routers connecting to customer and peer networks, as shown in Fig. 2. In particular, we implemented STRID illustrated in Fig. 1 as a module within the Gigascope [1] high speed traffic monitor. The data center where STRID is deployed hosts more than 30,000 Web sites using a large number of servers. During our measurement period, there are around 200,000 distinct IP addresses that visit the content servers each day.

STRID monitors outgoing traffic from the data center using an Endace Dag4.3GE monitoring card, which is installed on a Dell 2650 server with dual 2.8Ghz Pentium IV processors and running FreeBSD 4.9. The traffic volume is over 800Mbit/sec. We monitored all outgoing traffic on one of multiple links leaving the datacenter.

TABLE I
PASSIVELY MONITORED TCP TRACES

Data set	Collecting time	Number of flows	Number of prefixes
D_1	Oct. 07 - 20, 2005	$\geq$ 60,000,000	68,914
D_2	Dec. 13, 2005	552,444	1,760
D_3	Mar. 5 - 17, 2006	672,168	671

B. Data Gathering

Using STRID and existing measurement infrastructure available within the tier-1 ISP, we collected three sets of data.

- **Passively monitored TCP traces.** We passively monitor outgoing TCP flows. The purpose of the passive data set is used to determine the thresholds used for active probing trigger. In particular, we collected three TCP traces as shown in Table I.
- **Active probing data.** STRID triggers active probing to targeted destination based on the algorithm shown in Fig. 1. We collected results of active probings triggered during the time period from April 19, 2006 to May 9, 2006. The trace consists of 979,629 probings to 105,872 destination prefixes.
- **BGP routing data.** We collected BGP routing updates from a router, which is at the same location where we deploy STRID.

C. Determining Thresholds

In STRID, two thresholds are used to control active probing. In this section, we show how the thresholds are selected. Even though the threshold is determined based on the observation from one vantage point, we believe that it is representative because our measurement covers over 100K distinct destination prefixes that visit the content servers.

1) T_{ret}: We first infer packet retransmissions for all prefixes in data traces D_1, D_2, and D_3. We consider packets with the same sequence number as the last packet of the connection as packet retransmissions. Second, for each retransmission, we correlate packet retransmissions with BGP data. We use BGP update messages, including BGP announcements and withdrawals, to correlate retransmissions within a 10 second window. Third, for those correlated packet retransmissions, we investigate the probability distribution of the first timeout given the duration of a retransmission duration. Due to page limitation, we present the result from data trace D_1 in this paper. The results on other data traces are available in the full version of the paper [13]. The results on other data sets also show a similar result. From Fig. 3, we observe that most retransmissions that have more than 0.7 second of the first timeout tends to continue to last more than 1 second after the first timeout. The figure implies that using the threshold (0.7 second), we can discard about 70% of retransmissions that are less likely due to prolonged routing problems. In this paper, we set the threshold of retransmission duration $T_{ret} = 0.7$ second.

2) $T_{recover}$: In order to ensure the current network is in routing failure status during the sending of active probes, STRID monitors retransmissions until the time $T_{recover}$ expires. Suppose that for each timeout, the latest retransmission occurs at time t_i, and at time t_x the retransmitted packet is ACKed. We define the duration $R = t_x - t_1$ as *recovery delay*, which is used to estimate the delay of triggering probing. Just like deriving T_{ret}, we first infer packet retransmissions and correlate packet retransmissions with BGP data. For those correlated packet retransmissions, we derive the recovery delay based on data traces D_1, D_2 and D_3. Similarly, we present the result on data set D_1 in this paper, and other results are available in the full version of the paper [13]. Note that all results on the three data traces show the same result.

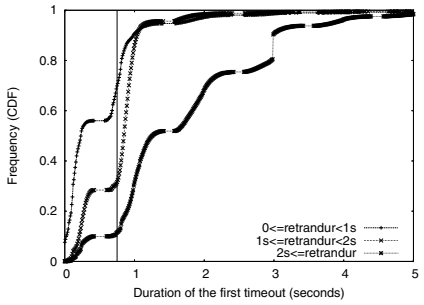

Fig. 3. Cumulative distribution of the first timeout for different retransmission duration. The vertical line indicates 0.7 second.

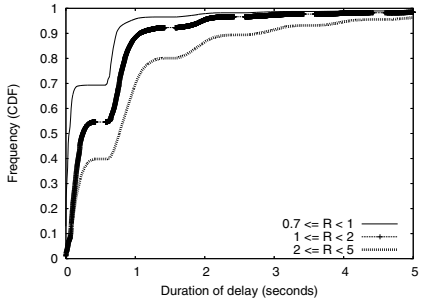

Fig. 4. Cumulative distribution of recovery delay R.

As shown in Fig. 4, we observe that if the latest timeout is less than 1 second, about 85% of retransmissions are recovered by 0.2 second. If the latest timeout is longer than 1 second and less than 2 seconds, about 80% of retransmissions are recovered by 0.8 second. If the timeout is longer than 2 seconds, about 80% of retransmissions are recovered by 1.2 seconds. Thus, we use the following threshold.

$$T_{recover} = \begin{cases} 0.2 \text{ second} & \text{if latest timeout is } < 1 \text{ second} \\ 0.8 \text{ second} & \text{else if latest timeout } < 2 \text{ seconds} \\ 1.2 \text{ second} & \text{otherwise} \end{cases}$$

V. STRID PERFORMANCE EVALUATION

We evaluate the performance of STRID in a real Internet experiment. In particular, we measure the following metrics based on data collected in our experiments.

1) **Routing failures covered by triggering mechanism**: This metric refers to the case that STRID detects a routing failure and triggers active probings to the corresponding destination, but the result indicates that the destination is reachable by active probings (i.e., the destination does not experience a routing failure).

TABLE II
CLASSIFICATION OF OVERALL REAL-TIME PROBINGS REPORTED BY STRID.

Failure Type	*# of probes (fraction)*	*# of prefixes (fraction)*
Unreachable	788389 (0.8048)	108196 (0.7445)
Reachable	157009 (0.1602)	33678 (0.2317)
Unknown	34191 (0.0350)	3457 (0.0234)
Total	979629 (1.0000)	145331 (1.0000)

TABLE III
CLASSIFICATION OF OVERALL UNREACHABLE PROBES.

Failure Type	*# of probes (fraction)*	*# of prefixes (fraction)*
Forwarding loop	3329 (0.0042)	536 (0.0050)
Forwarding failures	3469 (0.0044)	426 (0.0039)
IP-level path change	168981 (0.2143)	18134 (0.1676)
Multiple failure points	175500 (0.2226)	3469 (0.0321)
Total	351279 (0.4455)	22565 (0.2086)

2) **Routing failures missed by triggering mechanism**: This metric refers to the case that there is an observed BGP event that affects network traffic, but STRID cannot detect it in real-time.

A. Routing Failures Covered by Triggering Mechanism

During the time period from April 19, 2006 to April 30, 2006, STRID observes 9,889 destination prefixes experiencing packet retransmissions and probing packets are triggered to check their reachability. Table II shows the distribution of those probings during our measurement. We also count those probings on prefix level to understand the extent to which prefixes involve in routing outages. We observe that the number of unreachable probings is quite high (i.e., over 80% of all the probings, and over 74% of all prefixes). Only less than 5% of probings are filtered. They are excluded from our analysis in the rest of the paper.

B. Routing Outages Missed by Triggering Mechanism

We measure the routing outages missed by triggering mechanism in STRID through correlating BGP events with packet retransmissions. In particular, we collect BGP updates from a backbone router, one hop away from our monitoring point that is traversed by all outgoing traffic. There are two kinds of BGP events: announcements and withdrawals. We use withdrawal events in our correlation since a BGP withdrawal clearly represents the backbone router losing its routes to the corresponding destination. A routing failure can also appear to the observer as a set of BGP announcements (*implicit withdrawal*). However, we cannot distinguish those events from other announcement activity and therefore do not consider them. Note that, due to our BGP observation point, we are also not able to observe all the BGP events in the Internet. This leaves us with a subset of possible route failures impacting our traffic.

To identify BGP events that do affect TCP flows, we correlate BGP withdrawals and announcements with traffic flows by using a 10 seconds time window, and discard events that do not correlated with any packet loss. One reason why some flows might not experience packet loss even though we observed a BGP withdrawal event is that a supernet in the routing table might still offer a valid route. After these two steps we are left with a conservative set of BGP events that clearly impacted our traffic and therefore should have been detected by STRID. To compute the number of false negatives of STRID, we then correlate these BGP events with when STRID triggers probing using a 10 seconds time window.

In our experiments, we collect traffic trace for randomly selected 500 prefixes during time period from March 06, 2006 to March 17, 2006, and apply the above method to compute false negative. We observe that there are 13 BGP withdrawals and 274 announcements that impact traffic. STRID detects all withdrawals and about 70% of announcements in real time. Those undetected announcements may not cause any routing failures and packet loss.

VI. ANALYZING DETECTED ROUTING FAILURES

A. Diversity of Routing Outages

To further study the unreachable active probing, we analyze the diversity of routing outages by the actual path returned by the probes. We use the following definitions of observed routing paths that indicate routing issue:

- **Forwarding loops.** The probing indicates a forwarding loop.
- **Forwarding failures.** The probing returns an ICMP destination unreachable message, such as network unreachable (!N) and host unreachable (!H). This implies that a router does not have routing entries to the destination because of routing failures.
- **IP-level path change.** For each probing, we have three IP-level forwarding paths from the sender to a destination. We attribute changes in these three IP-level paths to routing outages because routing changes typically lead to IP-level path changes.
- **Multiple failure points.** For each probing, we define the last router in the probe that drops packets to a destination as the *failure point*. If packets are dropped at *multiple failure points*, and failure points get progressively closer to the source, we conclude this failure is caused by a routing failure. The reason is that the sequence of failure points, which has decreased distance to the source, corresponds to the propagation direction of BGP withdrawals.

Table III shows the results using the above method. We still count these four categories in term of probe and prefix. We confirm that more than 40% of unreachable probes and more than 20% of prefixes can be attributed to those classifications. We also observe that majority of those probes show IP-level path change or multiple failure points so that those packet retransmissions are most likely due to routing problems.

B. Repeatedly Occurring Routing Failures

To gain some insight in how frequently routing failures occur for given prefixes in our experiments, we measure the frequency of routing outages in term of number of days that a prefix or destination AS experiences at least one routing failure. Fig. 5 shows that among all prefixes that experience routing outages, about 66% of the prefixes or 72% of

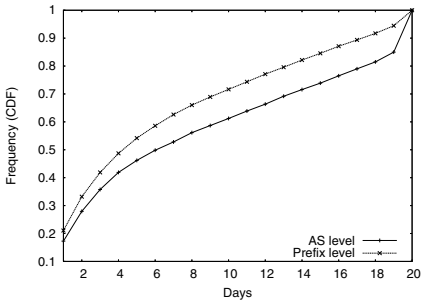

Fig. 5. Frequency of routing failures detected by STRID.

destination ASes experience routing outages on at least two different days during our measurement period (20 days). In the worst case, about 5% of prefixes or 15% of ASes experience reoccurred routing failures every day. Among those reoccurred failures, we examine the location of those failures, and find that more than half of them do not occur in destination ASes.

VII. Related Work

PlanetSeer [17], which is more related to our work, is a system to locate Internet anomalies by selectively invoking traceroutes from multiple vantage points. Similar to our system, PlanetSeer is a server-based system. The difference between STRID and PlanetSeer is that PlanetSeer simply uses TTL and four retransmissions as indicators of Internet anomalies.

Existing solutions in real-time detection of routing problems rely on detecting abnormal routing changes on the control plane, which address a different problem from our work. For example, [16], [2], [12], [6] used statistics-based detection to discover abnormal routing changes. In statistics-based detection, routing updates are first fed into a detection system. Then, the system is trained to learn the characteristics of normal routing behavior, and identifies anomaly if current behavior significantly differ from the normal one. Though a statistics based detection system is easy to deploy and can effectively detect large scale routing events (e.g., worm spreading) that either incur a lot of routing updates or affect a large number prefixes, it is not good at capturing routing problems which incur only a small number of routing updates, but affect certain prefix severely.

VIII. Conclusions

Despite the fact that increasingly many Internet applications depend on high availability of end-to-end paths, existing monitoring systems cannot help network operators to detect failures in the control plane quickly. In this paper, we have presented a real-time system for diagnosing routing problems. In particular, our system STRID monitors TCP retransmission on data plane. Upon detecting a possible routing problem that could affect network traffic, which from network operators' perspective matters the most, the system will automatically and quickly initiates active probing to determine the problem. We implemented STRID and demonstrated its feasibility on a large tier-1 ISP network.

We show that our real-time detection can effectively identify routing problems that impact network traffic, and can detect routing problems with a low false identification rate. STRID only uncover routing problems if traffic of the ISP is actually impacted. Furthermore, our system detects more routing events than those by monitoring BGP. STRID is able to capture a significant number of routing outages that include routing failures and forwarding loops. These are the key features that differ our real-time system with existing approaches on detecting routing problems.

Acknowledgement

We would like to thank anonymous reviewers for their constructive comments. The work is partially supported by NSF grants CNS-0325868 and CNS-0325868.

References

[1] Cranor, C., Johnson, T., Spatscheck, O., and Shkapenyuk, V. Gigascope: A Stream Database for Network Applications. In *Proceedings of ACM SIGMOD* (June 2003).

[2] Das, S. R., and Das, S. K., Eds. *Distributed Computing - IWDC 2003, 5th International Workshop, Kolkata, India, December 27-30, 2003, Proceedings* (2003), vol. 2918 of *Lecture Notes in Computer Science*, Springer.

[3] Feamster, N., Andersen, D., Balakrishnan, H., and Kaashoek, M. Measuring the Effects of Internet Path Faults on Reactive Routing. In *Proceedings of ACM SIGMETRICS* (June 2003).

[4] Floyed, S., and Jacobson, V. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking* (1993), 397–43.

[5] Krishnamurthy, B., and Wang, J. On network-aware clustering of web clients. In *Proceedings of ACM SIGCOMM* (2000), pp. 97–110.

[6] Kruegel, C., Mutz, D., Robertson, W., and Valeur, F. Topology-based detection of anomalous bgp messages. In *RAID* (2003).

[7] Labovitz, C., Ahuja, A., Bose, A., and Jahanian, F. Delayed Internet routing convergence. *IEEE/ACM Transactions on Networking 9*, 3 (June 2001), 293–306.

[8] Labovitz, C., Ahuja, A., and Jahanian, F. Experimental Study of Internet Stability and Backbone Failures. In *Proceedings of FTCS* (1999), pp. 278–285.

[9] Paxson, V. End-to-end routing Behavior in the Internet. *IEEE/ACM Transactions on Network 5*, 5 (1997), 601–615.

[10] Qiu, L., Zhang, Y., and Keshav, S. Understanding the performance of many TCP flows. *Computer Networks (Amsterdam, Netherlands: 1999) 37*, 3–4 (2001), 277–306.

[11] Roughan, M., Griffin, T., Mao, Z. M., Greenberg, A., and Freeman, B. Combining Routing and Traffic Data for Detection of IP Forwarding Anomalies. In *Proceedings of ACM SIGCOMM NeTs Workshop* (2004).

[12] Teoh, S. T., Zhang, K., Tseng, S.-M., Ma, K.-L., and Wu, S. F. Combining visual and automated data mining for near-real-time anomaly detection and analysis in BGP. In *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security* (2004).

[13] Wang, F., Gao, L., Spatscheck, O., and Wang, J. STRID: Scalable Trigger-based Route Incidence Diagnosis. Technical Report, 2006.

[14] Wang, F., Gao, L., Wang, J., and Qiu, J. On Understanding of Transient Interdomain Routing Failures. In *Proceedings of IEEE ICNP* (2005).

[15] Wang, F., Mao, Z. M., Wang, J., Gao, L., and Bush, R. A measurement study on the impact of routing events on end-to-end internet path performance. In *Proceedings of ACM SIGCOMM* (2006).

[16] Zhang, J., Rexford, J., and Feigenbaum, J. Learning-based Anomaly Detection in BGP Updates. In *MineNet '05: Proceeding of the 2005 ACM SIGCOMM workshop on Mining network data* (2005).

[17] Zhang, M., Zhang, C., Pai, V., Peterson, L., and Wang, R. PlanetSeer: Internet Path Failure Monitoring and Characterization in Wide-Area Services. In *Proceedings of OSDI* (2004).