Detection and Localization of Network Black Holes

Ramana Rao Kompella, Jennifer Yates†, Albert Greenberg*, and Alex C. Snoeren
University of California, San Diego, †AT&T Labs – Research, *Micorsoft Research
{ramana,snoeren}@cs.ucsd.edu, jyates@research.att.com, albert@microsoft.com

***Abstract*— Internet backbone networks are under constant flux, struggling to keep up with increasing demand. The pace of technology change often outstrips the deployment of associated fault monitoring capabilities that are built into today's IP protocols and routers. Moreover, some of these new technologies cross networking layers, raising the potential for unanticipated interactions and service disruptions that the built-in monitoring systems cannot detect. In such instances, failures may cause data packets to be silently dropped inside the network without triggering any alarms or responses (e.g., the failure is not routed around). So-called "silent failures" or "black holes" represent a critical threat to today's rapidly evolving networks. In this paper, we present a simple and effective method to detect and diagnose such silent failures. Our method uses active measurement between edge routers to raise alarms whenever end-to-end connectivity is disrupted, regardless of the cause. These alarms feed localization agents that employ spatial correlation techniques to isolate the root-cause of failure. Using data from two real systems deployed on sections of a tier-I ISP network, we successfully detect and localize three known black holes. Further, we present simulation results demonstrating that our system accurately and precisely (both greater than 80% according to our metrics) localizes a variety of failures classes.**

I. Introduction

With the increasing reliance on Internet-based commerce and IP-based communication facilities, Internet outages are becoming ever more costly, both in terms of lost revenue for companies relying on the infrastructure and for those managing it. Not surprisingly, a great deal of recent work has focused on rapidly detecting, diagnosing, and recovering from faults in backbone networks. Happily, this work has resulted in an increase in the overall reliability of the Internet[1]. Unfortunately, a number of complex failure modes exist that current systems fail to detect, and, therefore, are unable to recover from. In cases where the network is unable to automatically react to failures, manual intervention is required. Such action occurs on human time scales, however, and a potentially large number of customers may be offline for extended periods of time. Such failures are known as *"silent failures"* or *"black holes"* because existing networking protocols and devices do not alert on or automatically compensate for the failed component.

Here, we focus on black holes and silent failures in the context of MPLS-over-IP backbone networks, an architecture deployed by a number of tier-one ISPs. In this setting, packets are switched over MPLS tunnels, which themselves are established using standard IP routing protocols such as OSPF or IS-IS. A black hole scenario can occur when the underlying IP infrastructure is operational, that is, each IP hop along the route is functioning properly, but the corresponding MPLS tunnel fails to deliver packets. Such a black hole can be silent in nature, with no router alarm indicating that the MPLS tunnel is actually broken.

Black holes have a variety of causes, ranging from delayed routing protocol convergence to mis-configurations to bugs in individual router implementations. While backbone networks are generally able to address each cause after an incident occurs (i.e., by asking the vendor to fix the bug), experience shows there is always another bug. Due to the ever-increasing complexity of router control software, it is highly unlikely they will ever disappear entirely. While such silent failures are rare, they can have a large and egregious impact; in many cases, a complete loss in connectivity results in a significant financial damage to both the ISP as well as the customer. These failures are extremely time-consuming to localize (order of hours to days) because there are no alerts/alarms to guide operators to the location of the failure. Hence, from an operational standpoint, it is imperative to design a mechanism that can quickly detect such failures, and, moreover localize the root-cause of the failure.

Luckily, most tier-one ISPs already conduct a significant amount of active probing within their networks to provide service-level agreements (SLAs). In this paper, we analyze the effectiveness of a fault localization methodology that leverages these existing end-to-end probing mechanisms to detect and localize silent failures. Using a spatial-correlation-based approach, we analyze three known silent failures in tier-one backbone network. We show that, had our localization system been in place at the time of the failures, they would have taken considerably less time to detect and localize. In order to validate our methodology further, we collected probe data resulting from a large number of (less serious) network events (e.g., routing changes, etc). By definition, such failures are not silent in nature, but this approach gives us a reasonably large—albeit non-random—set of cases to test our technique. Note that the majority of these events are very short-term, which makes them hard to detect and localize using active probing mechanisms. Even with this challenging data set, we find that our technique is highly successful in localizing the problems; we achieve greater than 80% accuracy and 80% precision across a wide range of failure scenarios.

The remainder of this paper is organized as follows. We begin by discussing the silent failure problem in Section II and our approach using fault detection and localization in Sections III and IV. Section V describes our system architecture followed by the evaluation methodology in Section VI. Our evaluation results are discussed in Section VII followed by related work in Section VIII and conclusions in Section IX.

[1]Measured in terms of packet loss rate [1], [17]

II. Silent network failures

Today's operational backbone networks are intrinsically layered. High speed IP networks are overlaid on optical networks; an IP link is implemented as a path through a set of optical components, some of which are shared across multiple IP links. Similarly, in many tier-one ISP networks, there may be an intermediate MPLS layer, wherein IP packets are transported via label-switched paths (LSPs), which in turn are established using IP routing protocols (such as OSPF). Moreover, multiple Virtual Private Networks (VPNs) may be overlaid on top of the MPLS topologies. Given the complex, cross-layer interactions between the different control planes, complex failure modes and fault scenarios arise. The first line of defense against such failures is provided by resiliency in the protocols themselves to heal around failed components, and SNMP traps providing alerts to network monitoring and management systems, which in turn trigger troubleshooting activities.

Unfortunately, there is no guarantee that this first line of defense succeeds. For example, in cases when LSPs are established following the shortest-path routing using OSPF, one failure scenario that has been observed in practice is when OSPF re-routes due to a problem, but MPLS does not follow it. While router work-arounds are being developed to deal with this problem, it is unlikely such problems are going to disappear completely. In other failure instances, MPLS control plane is working properly (hence no alarm), yet there is corruption in the forwarding plane due to poor implementations and/or configuration errors. Many other subtle failure scenarios in MPLS LDP can be found in [6].

In this paper, therefore, we specifically focus on detection and localization of silent faults arising from the interaction between MPLS and IP layers. Owing to the rate of change and innovation in the underlying technologies, these networks are potentially prone to silent failures and black holes. Our approach has two main components—fault detection and fault localization. Edge routers or special measurement servers connecting to them issue probes periodically that test connectivity to other edge routers, and report failures when probe packets are not acknowledged by their destination. These connectivity losses are then fed into a localization engine that spatially correlates probe data according to the underlying topology to identify a small set of likely locations of the failure. This localization step is the primary reconnaissance to a final—and often (necessarily) manual—step of actually diagnosing the root-cause of the failure and fixing the problem. Operational experience, however, shows that outage times from silent failures can be attributed primarily to detection and localization. Once the problem is localized, it can often be quickly recovered (e.g., traffic can be re-routed immediately); many times the actual repair can be completed off-line. In the next two sections, we discuss our fault detection and localization methodology in more detail.

III. Failure detection

In our detection mechanism, the monitoring server establishes connections with each edge router, injects periodic probes to every other destination edge router in the network and reports if any of the probes are lost. There are many efficient mechanisms suggested in literature (e.g., [14]) to design the probing methodology so that the number of probes injected into the network is minimal ($O(nlog\ n)$ instead of the obvious $O(n^2)$). However, our goal in this paper is not to optimize the probing methodology itself, but to devise an efficient mechanism to localize the failures once they are identified by the detection system. So, we assume that our detection mechanism operates in a rather brute force manner, issuing $O(n^2)$ bi-directional probes from every source to every destination at a certain frequency and according to a particular distribution.

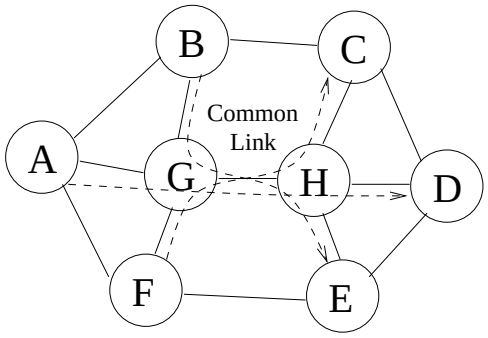


Fig. 1. Example topology with failure impacting a set of paths going through a given link (G-H).

During a persistent failure, probes that traverse the failed link (or the set of failed links) get dropped. In transient failures, only a subset of the probes that traverse the link during the failure period gets dropped while others succeed. In both these cases, the set of failed probes constitutes the *failure signature*. In our architecture, we divided time into fixed 15 minute intervals. The set of probes (out of the n^2 probes) that are lost in this 15 minute interval constitutes the failure signature. The monitoring server forms the failure signature by collecting information about lost probes from every edge router and passes it to the localization engine to localize the root-cause of the failure. Note that the 15 minute interval we have chosen is based on the granularity of the measurement data we had access to, and not a fundamental requirement of our system.

IV. Fault localization

The key idea we use for fault localization is *spatial correlation*. The set of od-pairs that belong to the failure signature is intersected to identify if there are any shared links along their respective paths. These shared links form the most likely explanation for the detected failure signature and hence the localization engine outputs this set of links as the *hypothesis*. For example, in Figure 1, if the set of paths A-G-H-D, F-G-H-C, B-G-H-E all fail in the same time interval (temporally correlated failures), spatial correlation leads to the only link that is common to all these paths – the link from G to H. The localization engine would, therefore, return the singleton set $\{$G-H$\}$ as the hypothesis for the black hole.

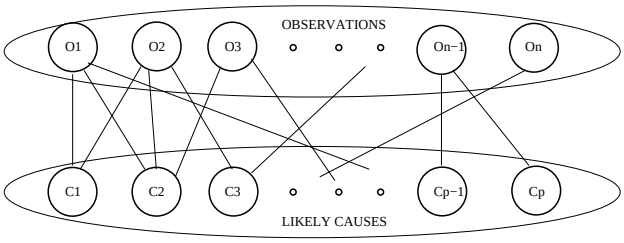


Fig. 2. Bipartite graph modeling of the failure localization problem as done in SCORE system.

In many situations, however, it is not as simple to localize the failure as in the above toy example. The localization engine does not have access to an oracle to determine whether a single failure, dual failure, or multiple failures occurred in the network; it has to determine the cardinality of the set of failures based on the end-to-end connectivity information alone. There are also additional spurious losses in the network due to noise that can complicate localization. Besides, for a given failure signature, there could be many different likely explanations. Exploring all the possibilities is often not feasible, and even when feasible, it is too overwhelming to a network operator. Such enumeration is also likely to contain many cases that are not typically observed in practice. We therefore attempt to identify hypotheses that can explain a given failure signature with the smallest set of failures. Our approach is in accordance with the principle of Occam's razor, that in a nut-shell, suggests that the simplest explanation is most likely.

A similar approach has been taken by a variety of localization algorithms studied in the literature (e.g., SCORE [10], SHRINK [9], [21], etc). Prior work in localization modeled the problem in a top-down fashion using a bipartite graph as shown in Figure 2. In the Figure, the observations (the set of od-pairs in the failure signature) are in the top partition and the likely root-causes (failed links) are shown in the bottom partition. An edge exists between an observation and a cause if the observation would have been made had the root-cause actually failed. In the case of SCORE and SHRINK, the observations are IP link failures and the likely root-causes are shared risk link groups (SRLGs) [19]—physical components shared across multiple IP links (e.g., routers, links, fiber spans, etc.). While our localization engine is similar in spirit to both SCORE and SHRINK, there are several key differences in our problem domain that prevent using either SCORE's greedy approximation to the set-cover problem or SHRINK's Bayesian approaches:

- **Lack of complete information.** In the problem domains considered by SCORE and SHRINK, the failure signature consists of all impacted IP links. Due to our active measurement methodology, we are unlikely to collect the full set of impaired links, leading to a partial failure signature in many transient black hole scenarios. Hence, algorithms which rely on the assumption that the lack of reported failure on a particular path implies that all the constituent links are good are not appropriate for our scenarios.

Algorithm 1 `GREEDY(FailureSignature F)`

```
1: E = {};
2: U = FailureSignature F;
3: H = {}; // Hypothesis set
4: while (U ≠ {}) do
5:    for (observation o ∈ U) do
6:       //All links that belong to atleast one path
7:       //for the od − pair in observation o
8:       linkVector = getAllLinks(o);
9:       //Update stats of links in linkVector
10:      updateLinkStats(linkVector);
11:   end for
12:   linkSet = identifyCandidates();
13:   //Move events that contain links in linkSet
14:   //from U to E
15:   moveEvents(linkSet, E, U);
16:   addToHypothesis(H, linkSet);
17: end while
18: return H;
```

- **Noisy failure data.** A second major difference is the presence of noise that can bias localization towards large hypotheses with quite a few spurious candidates. We address this problem by subjecting the output of the localization algorithm to additional filtering described in Section IV-B.
- **Scale.** The topology is much larger (in terms of the number of end-to-end paths) and dynamic in nature; calculating paths on-the-fly between all od-pairs for localization of every fault is impractically time-consuming. This reality forces us to again deviate from algorithms that require considering all od-pairs during diagnosis. We restrict ourselves only to the set of od-pairs that indeed failed.
- **Redundant paths.** Finally, there can be multiple paths between a source and destination due to equal cost multi-path (ECMP)—a scenario that does not exist in the earlier IP fault localization problem studied in [10]. We address this problem by considering the union of all the paths between a given od-pair at the instant of the failure. This approximation, while conservative, works well in practice.

A. Core algorithm

We propose a localization algorithm called MAX-COVERAGE that employs a greedy approach for fault localization similar to SCORE algorithm in [10]. MAX-COVERAGE iteratively picks the link that explains the most number of observations in the failure signature, prunes this set of observations from the failure signature and repeats the process until no more observations remain in the failure signature. MAX-COVERAGE is therefore biased in favor of links that are present on a larger number of paths in the network. This iterative algorithm does not assume a priori any particular number of failures, and hence performs well even in multi-failure scenarios. Intuitively, MAX-COVERAGE partitions the given failure signature into multiple sets, each corresponding

to one among the multiple simultaneous failures.

The core algorithm GREEDY, for the MAX-COVERAGE algorithms is shown in Algorithm 1. The algorithm works as follows. GREEDY initializes two sets E and U to null set ϕ and the failure signature F respectively. The sets E and U correspond to the explained and unexplained set of observations in F. Then, for each observation $o \in F$ (od-pair in the failure signature), GREEDY first obtains a $linkVector$ containing all the links that correspond to o that are present on at least one path between a given od-pair as shown on line 8. It then maintains another data structure that updates the set of events o that are associated with a given link in the $linkVector$. In line 12 of Algorithm 1, GREEDY then picks a set of likely candidate links in each iteration out of all the possible links that were collected for events in U based on some metric. It then moves the events associated with these candidate links into the set E out of the set U and the iteration terminates when U becomes empty. MAX-COVERAGE picks a link with maximum coverage (the link that covers the largest number of observations) among all candidates in every iteration in the $identifyCandidates()$ function.

B. Additional issues

From the network operator stand-point, we need to address two additional issues. First, the core localization algorithm MAX-COVERAGE, discussed previously, outputs a hypothesis given a topology and a failure signature. However, routing changes during the time interval of interest can result in multiple topologies that must be accounted for. In order to localize failures in such scenarios, we need to use the route before and after each routing change. If multiple route-changes occur within a failure interval, applying different topologies results in different hypotheses for the same failure signature. Hence, we need a mechanism to combine all these hypotheses into one that the network operator uses for localization. Second, MAX-COVERAGE tries to generate an explanation for every observation in the failure signature, even those due to noise. This results in the hypothesis being unnecessarily large, making it cumbersome to the operator. We, therefore, need a mechanism to reduce the size of the hypothesis further so that the system is more effective and usable to the network operator.

To address these issues, we first generate multiple hypotheses for a given failure signature using all the available topology snapshots in the failure interval, and the following two algorithms in sequence to output the final hypothesis:

- *hypothesis selection algorithm* for selecting the hypothesis using different topology snapshots; and
- *candidate selection algorithm* for selecting candidate links within the hypothesis (based on the contribution of each failure in the hypothesis to the observation set).

An oracle hypothesis selection algorithm that has access to the ground truth (we call it ORACLE) can easily pick the best hypothesis that is closest to the truth. In real-life however, we do not have access to such an ORACLE, and hence we need an online approximation to this ORACLE that can select the best possible hypothesis among seemingly plausible competing hypotheses. One such algorithm we found to be a good approximation is UNION (shown later in Section VII-B), that computes the union of all the hypotheses with different topology snapshots.

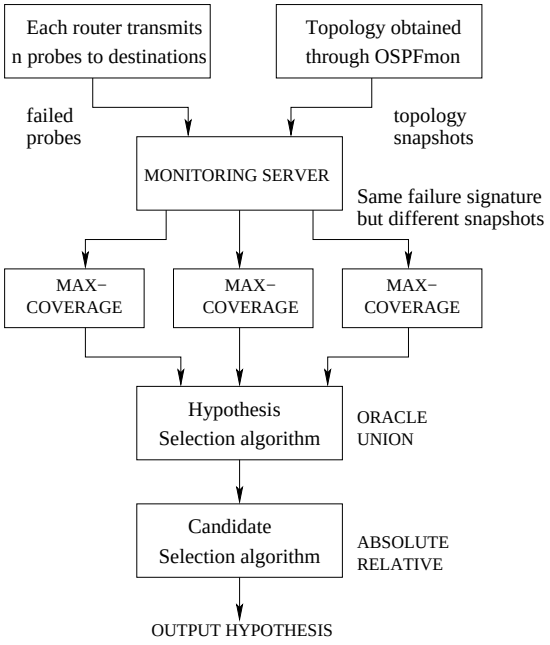


Fig. 3. System architecture

For candidate selection, we study two different algorithms based on absolute threshold (ABSOLUTE) and relative threshold (RELATIVE). As the name suggests, ABSOLUTE picks those candidate links that explain greater than a threshold number of observations, while RELATIVE picks those candidates that explain greater than a particular fraction of all the observations within the failure signature. Later, using results from real data in Section VII-A, we show how to pick the right algorithm for candidate selection.

V. System architecture

In Figure 3, we show the complete flow of information in the fault localization system. Each edge router issues n probes to other destinations and report the probes that get lost to the monitoring server. The monitoring server invokes the localization algorithm with the failure signature obtained from the detection system and obtains hypothesis corresponding to each topology snapshot for that failure interval obtained from the OSPF monitor. It then uses the hypothesis selection algorithm followed by the candidate selection algorithm to output the final hypothesis that the operator uses to perform further diagnosis. Next, we discuss the methodology we use to evaluate the localization system.

VI. Evaluation methodology

In order to evaluate the fault localization system, we need three ingredients – failure data, ground-truth of these failures, and metrics to compare the output of the fault localization

Feature	IPFM	MPFM
Topology	Core backbone	Core and outside backbone
Number of routers	10s	100s
Probe distribution	Poisson	Periodic
Probe frequency	1 per 3.3 seconds	1 per minute
Layer of the probes	IP	MPLS

TABLE I
COMPARISON OF FEATURES OF IP AND MPLS FAULT MONITORING SYSTEMS

system with the ground truth. In this section, we discuss these in more detail.

Failure data: We collected failure data from two different real monitoring systems, IP and MPLS fault monitoring systems (IPFM and MPFM) deployed on a tier-I ISP backbone network. Both systems operate very similar to each other, differing only in the topology, the frequency of probes, and the layer at which probes are issued. Table I outlines the features of these two systems. IPFM monitors the core IP backbone network, while MPFM monitors MPLS tunnels that originate from a subset of edge routers in the backbone, traversing the backbone and finally again terminating at other edge routers. Both systems share the same backbone network and hence any fault in the backbone network is typically detected by both systems. However, since IPFM issues probes at the IP layer, silent MPLS failures are typically not detected by IPFM but are detected by MPFM. So, in effect, MPFM detects a superset of failures that are detected by IPFM.

Ground truth: Since black holes are relatively infrequent, we have access to only three known silent failures that have already been diagnosed and fixed in the network. This small set of black holes is insufficient to thoroughly evaluate our system. Therefore, for majority of our experiments, we relied primarily on non-silent failures caused by routine routing and congestion incidents. We correlated our hypotheses with ground-truth extracted from three data sources – OSPF LSAs, syslogs and SNMP data. First, *OSPF LSAs* indicate topology updates in the network that can potentially affect forwarding for a short period, during which some probes can get dropped.

Second, in the core backbone network, many of the logical IP links are in fact a bundling of many interfaces, known as composite links [2]. The router typically load-balances the packets among these multiple interfaces. So, if one member of the composite link fails, the set of probes that traversed through that member link fails before the router load-balances again between other members. However, such failures do not appear as an OSPF LSA as the logical IP link is still in tact within the topology. We, therefore, used *syslogs* to obtain information regarding these failures.

Third, in conditions of high link utilization, such as during failures or during maintenance, links can experience heavy packet loss, and therefore, can cause end-to-end probes to get dropped along these links. In order to obtain congestion events based on link utilization, we relied on *SNMP data*.

Note that the ground-truth obtained through the above data sets is only approximate, as there can be instances when we do see a link failure in the ground-truth (using LSAs, syslogs and SNMP data) but the event does not affect real traffic. This is caused either because the duration of the failure was too small for active probes to detect or because of noisy probe losses that have no real root-cause in our ground-truth set.

Metrics for comparison: We define two metrics for comparison – accuracy and precision. Accuracy is the fraction of links in the ground-truth that are also present in the hypothesis. We define the ALL accuracy metric which makes its comparisons by assuming that all the candidate links in the ground-truth indeed affected traffic and hence are captured in the failure signature. However, as we mentioned previously, this might not be true in many cases, particularly in very short outages that are hard to capture with active probing methods. In order to take this into account, we define a more conservative accuracy metric (called ATLEAST_ONE) in which accuracy is 1 if at least one of the links in the ground-truth matches the hypothesis and 0 otherwise. We believe the real accuracy of our system lies between these two bounds.

Precision, on the other hand, is a metric that quantifies the size of the hypothesis in relation to the ground-truth. It is defined as the fraction of links in the hypothesis that are also present in the ground-truth. In effect, precision captures the amount of truth in the hypothesis. So, a precision of 80% would imply that the hypothesis has 80% links that are part of ground-truth.

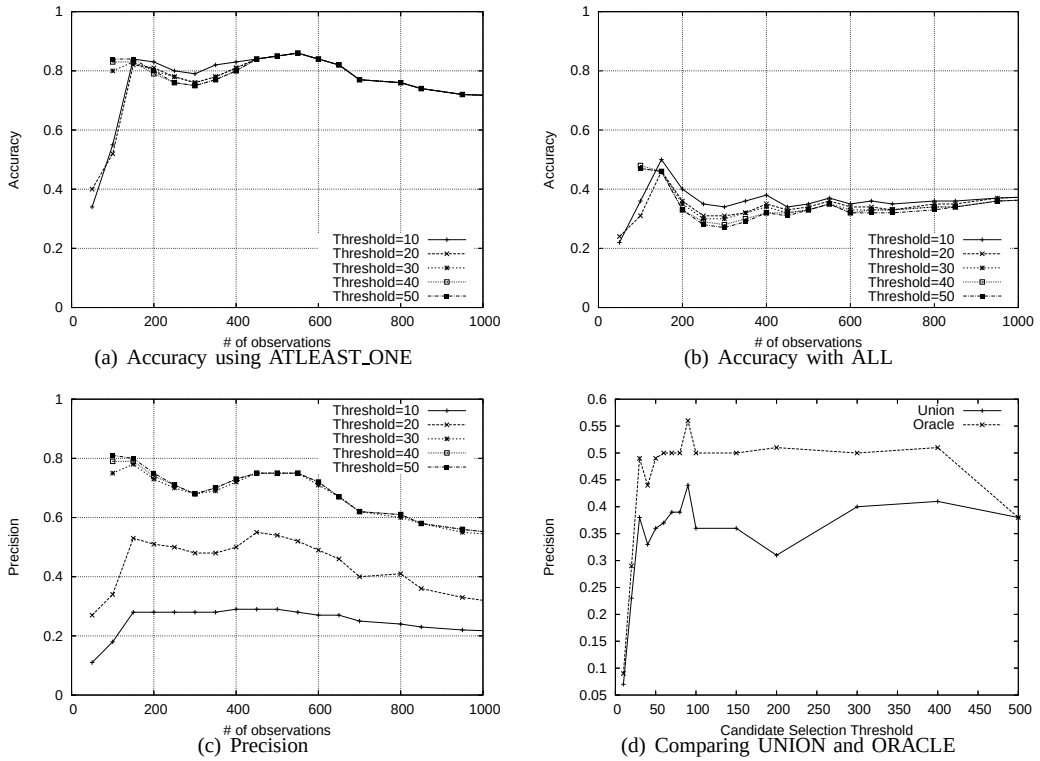
These two metrics together evaluate the efficiency and usability of the localization system to a given network operator. In other words, high accuracy of the localization system means low false negatives, and high precision means low false positives for the system. Obviously, an ideal localization system would have accuracy and precision be 1. In general, there exists a trade-off between accuracy and precision. For example, if we include every link in the network as part of our hypothesis, we would, in every case, be accurate 100% of the time. But such a hypothesis suffers from very low precision, making it completely useless.

VII. EVALUATION RESULTS

Now, we present the evaluation results using the methodology outlined in the previous section. Our results are in four phases. First, we evaluate the different candidate selection and hypothesis selection algorithms. Second, we divide the failures in several categories and compare the performance of our localization algorithm for these different failure scenarios. Third, we perform joint analysis on failures common to both IPFM and MPFM systems. Finally, we outline our experience in localizing real black hole scenarios in the network. As a guideline in many of these experiments, we do not seek to localize every failure in the network, but as many failures that the system can accurately and precisely localize.

A. Candidate selection algorithm

In Figure 4, we plot the accuracy using both ATLEAST_ONE and ALL metrics and precision obtained using the ABSOLUTE candidate selection algorithm for the MPLS fault localization system. We fixed the hypothesis selection algorithm to ORACLE. On the x-axis, we vary the cardinality of the failure signature (number of observations)



(a) Accuracy using ATLEAST_ONE (b) Accuracy with ALL

(c) Precision (d) Comparing UNION and ORACLE

Fig. 4. Accuracy (using both ATLEAST_ONE and ALL metrics) and precision in diagnosing failures when varying the threshold for the ABSOLUTE candidate selection algorithm. For the first three graphs, we used the ORACLE hypothesis selection algorithm. The final graph shows a comparison between the UNION and ORACLE algorithms.

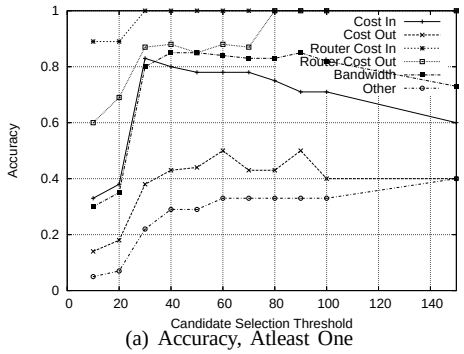
from 50 all the way up to 1000 observations in steps of 50. On the y-axis, the average accuracy/precision corresponding to all failure intervals that have at least x observations is shown. In effect, Figure 4 shows the trend in the accuracy/precision as the failures impact more and more od-pairs.

Several conclusions can be drawn. First, the number of failure intervals reduces exponentially from about 600 bins with more than 50 observations to about 20 bins with more than 1000 observations (not shown in Figure). This is expected, since the number of large failures is typically much smaller than the number of small failures. Second, the accuracy and precision of localization increase as the failure size increases initially from 50 to 150 observations due to a much stronger failure signature. However, it decreases slightly after that but is inconclusive as the number of failure intervals are extremely small for any statistical significance. Third, an ABSOLUTE threshold of 30 that selects candidate links in the hypothesis which cover at least 30 observations seems to represent a good trade-off between accuracy and precision. Below this threshold, the precision is significantly lower while accuracy is only slightly higher. Increasing the candidate selection threshold beyond 30 leads to marginal decrease in the average accuracy, while precision does not improve any further.

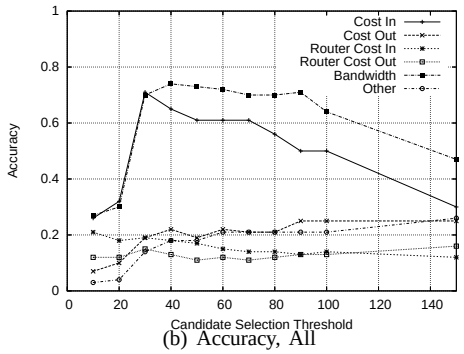
From Figure 4(a), we can also observe that for 20% of the failures, our hypothesis set contained no match with the ground-truth (using the ATLEAST_ONE metric). There are two reasons for this. First, as we mentioned before, our ground truth is only approximate. Due to the best-effort design of IP networks, in many cases, the loss of a probe (reported in the failure signature) does not automatically imply the existence of an associated alarm. Second, due to the coarse granularity, our measurement probes cannot possibly capture very short network events (e.g., during link-metric changes). In such cases, the ground-truth will have an event (since we use OSPF LSAs), while we may or may not have any associated probe loss (depending on the timing of the probe). Considering the approximate nature of our ground-truth necessitated by the lack of too many real blackhole situations, our results indicate the system is quite accurate.

For RELATIVE threshold (omitted for space reasons), we observed that even for a threshold of 50%, failures with fewer than 100 observations are included. Due to the fact that the signature is not too strong for such failures, the average accuracy is lower (around 60% for ATLEAST_ONE metric). Of course, if we only considered failures that have a minimum number of observations, the overall accuracy of the RELATIVE could be improved. The other option is to use the ABSOLUTE metric because placing a threshold on the individual candidates in the hypothesis automatically leads to considering only larger failures where the average accuracy and precision appears to be higher. We chose to use the ABSOLUTE metric and fixed the threshold to 30 for later experiments on the MPFM data.

For the IPFM system too, we performed a similar analysis and found that a threshold of 3 appeared to the best. In general, depending on the particular type of monitoring data, one needs to tune the thresholds to pick the one that represents a good trade-off between accuracy and precision. Of course, since



(a) Accuracy, Atleast One



(b) Accuracy, All

Fig. 5. Accuracy (using both ATLEAST_ONE and ALL metrics) and number of bins for different types of failures. For these graphs, we fixed the candidate selection algorithm to ABSOLUTE.

black holes themselves are rare, we have to rely on correlating other non-silent failures (using approximate ground-truth) to determine the right thresholds for the particular type of measurement data. We believe devising mechanisms to automatically adapt the threshold without human intervention is a challenging problem.

B. Hypothesis selection algorithm

In Figure 4(d), we plot the precision for the ORACLE and UNION algorithms that combine the multiple hypotheses obtained using different topology snapshots within a failure interval into one. The x-axis is the ABSOLUTE candidate selection threshold that we vary from 10 all the way up to 500. For each of these candidate selection thresholds, we identify all those failure intervals that had at least one candidate link remaining in the hypothesis after we apply the candidate selection thresholds and compute the average accuracy/precision for these failure intervals. The number of bins reduces with increasing candidate selection threshold due to the fact that we discard bins that do not have any candidates left in the hypothesis after we apply the threshold.

UNION performs similar to ORACLE and perhaps better than any other hypothesis selection algorithms as it includes candidate links from all the hypotheses obtained by applying different topology snapshots. However, there is a loss in precision when the UNION algorithm is applied compared to the ORACLE. This is because UNION includes all links to the hypotheses whenever there are two different hypotheses obtained using different topologies of the network, while ORACLE only includes the best possible match with the ground-truth. For majority of the bins, there is no change in topology. Therefore, when we compute the precision over all the failure bins, this loss in precision is insignificant. However, if we consider only the bins that had a change in topology (shown in Figure 4(d)), the difference in precision between UNION and ORACLE is around 15%, which we believe is acceptable. For accuracy however, we do not find any difference between UNION and ORACLE as expected (and hence not shown in the Figure).

C. Analysis by failure type

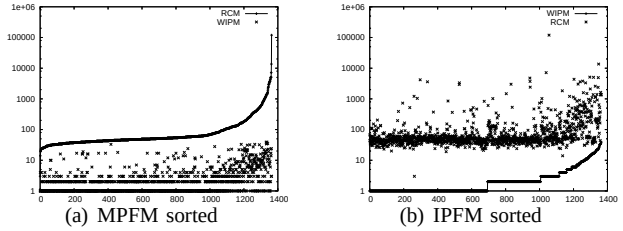
There is a variety of failure scenarios and the accuracy of localization is a function of the type of failure. We classify all the failures based on the OSPF LSAs into the following types:

- Router cost out: Traffic is removed from all links associated with an entire router by changing the routing protocol weight up to an excessively high value.
- Router cost in: Traffic is moved back on to a router.
- Link cost out: Traffic is removed from a particular link and not the entire router.
- Link cost in: Traffic is moved back on to a given link.
- Bandwidth events: A part of the composite links in the network has failed. This failure typically can affect certain probes in the network. These messages are usually reported as *bandwidth increased/decreased* for the composite links in the syslogs, and hence we call them bandwidth events.
- Others: Any other failure that does not fit the above categories is included here.

Since manual classification is not scalable, we applied simple hueristics to classify a failure event. For example, if all/most of the LSAs have one end-point in common, then it is a router-related incident. If the LSA indicates a change of metric from higher (lower) cost to lower (higher) cost, then it is a cost in (out) event. If there is only a few links (less than 5) experiencing this change of OSPF weight, then we deem it an individual link cost in/out event. Partial composite member link failures are identified through the corresponding notification in the syslogs.

In Figure 5, we compare the average accuracy on the y-axis varying the candidate selection threshold on the x-axis for both ATLEAST_ONE and ALL metrics. Recall that increasing the candidate selection threshold automatically considers only failure intervals that have a large number of observations. From Figure 5(a), we observe that localization was the most accurate for bandwidth-related events for the ATLEAST_ONE metric. This is because the number of simultaneous network events in the ground-truth for bandwidth-related failures is small (unlike router cost in where all the links of that router are part of ground-truth) and this leads to a more crisper and clearer signature to localize. Router cost out and cost in events ranked next in terms of accuracy according to the ATLEAST_ONE metric.

However, when we compared the ALL metric for different failure types (shown in Figure 5(b), the accuracy of link cost in and out events was better than that of the router events. This is due to the fact that during router cost in/out events, a larger number of od-pairs are impacted, most of which are directly



(a) MPFM sorted (b) IPFM sorted

Fig. 6. Joint distribution of IPFM and MPFM data in terms of the number of observations sorted by both types of data.

passing through the router. So, the chances of finding at least one match between the ground-truth and the hypothesis is far stronger (hence higher accuracy according to ATLEAST_ONE metric), while the fraction of matches between ground-truth and the hypothesis is considerably lower for these events. Note that the lack of clear trend as we increase the threshold beyond 80 is due to a much small number of failure intervals (around 20-25).

For the IPFM system too, we observed a similar phenomenon, while the exact numbers are much smaller than that of the MPFM system. Due to lack of space, we omit these results.

D. Joint analysis

In this section, we consider the joint analysis of both the systems on failures that are commonly observed among the two. In Figure 6(a) and Figure 6(b), we plot the number of observations for both MPFM and IPFM system failures in a given failure bin sorted by MPFM and IPFM failure data respectively. From the figure, we can observe that there is a loose correlation between the two systems in terms of the number of observations. However, there are several anomalies too, where od-pairs are impacted in one system but not the other. We believe this is due to the detection system and the nature of the failure. For example, if the failure is too small, there could be cases where the probes in one system detected the failure while the other did not.

We also compared the accuracy and precision obtained by the two systems in diagnosing common failures using the same ground-truth we have obtained using OSPF LSAs and other syslogs. The correlation scatter plots are shown in Figure 7. In Figure 7, the x-axis of each point is the accuracy of the MPFM system and the y-axis is that of the IPFM system. Each point on the other hand corresponds to a set of failure intervals classified by the number of observations ranging from 50 to 500 in steps of 50 for the MPFM system. We did a similar analysis by varying the number of observations from 2 to 30 in steps of 2 for the IPFM system. The results were similar to the that for the MPFM system and hence in the interest of space, we omit these details.

In Figure 7(a), the accuracy of MPFM using the ATLEAST_ONE metric is higher in most of the situations (most points lie below the line), suggesting that the MPFM system is more accurate in localizing failures. On an average, we can observe at least 30% higher accuracy for MPFM system when compared to IPFM system. For the ALL metric however in Figure 7(b), while the MPFM still outperforms the IPFM system the difference is less significant than before. Even the precision results in Figure 7(c) show that the MPFM system has on average a 40% higher precision than that of the IPFM system.

The main conclusion that can be drawn from this correlation is that the MPFM system outperforms the IPFM system despite lower frequency of probes for many of the real failure scenarios. This is due to two reasons. First, the larger number of probes (because of a much larger number of od-pairs in the MPFM system) allows much stronger failure signatures as opposed to the IPFM system. Second, the link failure signatures are significantly different (in terms of the set of the impacted od-pairs), so the localization algorithm can crisply identify the failed link.

E. Real MPLS black holes

In this section, we briefly describe three known silent MPLS black hole scenarios that we analyzed using our system. In the first incident, misbehavior of a new device that was connected to the periphery of the network caused many routes to go through the device which were then subsequently black holed. This is a perfect example where we need to consider all the topology changes within a failure interval. In this case, our localization system on the IPFM data output two candidate links as the hypothesis – the link before and the link after the re-routing of traffic. For this incident, the localization accuracy therefore is 100% while precision is only 50%.

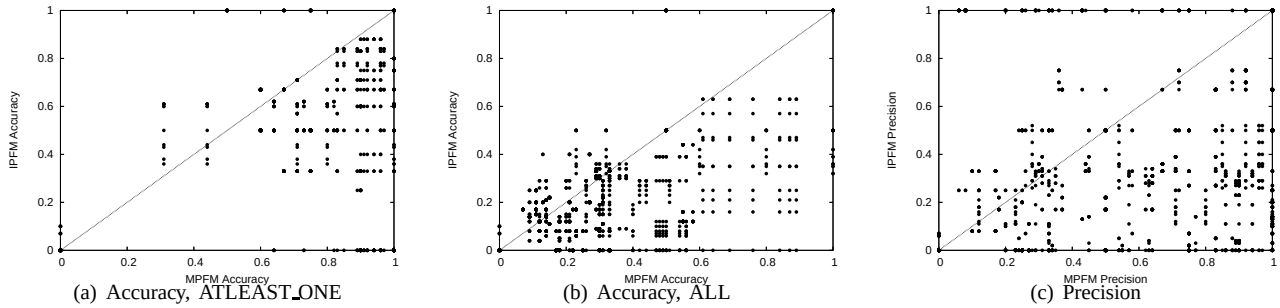
In another failure scenario, the forwarding component of a line card failed to dequeue packets until the card was reset. Our localization system output a hypothesis that had five candidate links, out of which, when we applied our ABSOLUTE threshold of 30 eliminated the four false positives out of the hypothesis and contained only the actual failed link. This hypothesis therefore has 100% accuracy and precision.

Another known black hole scenario happened due to a misconfiguration causing brief loss in connectivity to MPLS paths that traversed that link. Our localization algorithm output a hypothesis that contained four candidate links, two of which were eliminated after we applied our candidate selection algorithm. Out of the two, one of them was the actual black hole while the other was a false positive. However, the false positive could not be easily distinguished from the actual black hole since both these links appeared on all the paths corresponding to the impacted od-pairs (due to the mis-configuration).

VIII. Related work

Though there is a tremendous amount of literature in the area of network fault management and monitoring, there has been little discussion of silent failures or black holes of the types considered here. Systems and general techniques for network data correlation are widely used, and are under continuous refinement as new statistical methodologies for fault and anomaly detection are developed [18], [4], [11], [16], [5], [8].

Inference problems generally are of wide interest in the operational and networking research communities. While we know of no other work that targets the silent failure detection problem we consider here, there is considerable research in



(a) Accuracy, ATLEAST_ONE (b) Accuracy, ALL (c) Precision

Fig. 7. Accuracy and precision correlation between MPFM and IPFM system failure data. Each point represents the accuracy of MPFM on the x-axis and IPFM on the y-axis for all bins with a certain number of MPFM observations.

use of partial or incomplete data to reconstruct unknown network internal and external topology, traffic and performance. We take a simple, greedy approach to inference, which we find works well. More complex approaches might also be of interest. In particular, there is the rich area of network tomography, an approach to (massively under-constrained) linear inference, which has been applied to inferring topology and link performance from end-to-end measurements, as well as to inferring Origin-Destination (OD) traffic demands from link traffic measurements [3], [7], [13], [15], [22], [23], [24], [25].

Our detection system uses standard mechanisms in route and topology monitoring and packet probing to identify reliability metrics such as packet loss, delay, etc., on a per-path basis. Such measurement is routinely performed by many ISP backbone operators using a variety of different tools such as ZING [12] and BADABING [20].

IX. Conclusions

In this paper, we developed and evaluated a simple yet effective methodology for the localization of black holes or silent failures in the network. One of our key contributions is the successful application of spatial correlation to localize failures in the presence of noisy data. Further, our methodology itself is quite general; hence, we expect it to be applicable for a variety of failures—not just those that are silent in nature. Using real failure data obtained from a tier-1 network's IPFM and MPFM systems, we demonstrated that both systems can effectively aid network operators in troubleshooting failures. Our results also indicate that MPFM failures can be localized more accurately than IPFM due to MPFM's larger topology, despite using a lower probe frequency. A key remaining challenge, however, is to enable automatic recovery from the silent failures detected and localized by our systems.

References

[1] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient overlay networks. In *Symposium on Operating Systems Principles, SOSP*, pages 131–145, 2001.

[2] AVICI Systems Inc. `http://www.avici.com`.

[3] J. Cao, D. Davis, S. V. Wiel, and B. Yu. Time-varying network tomography. *J. Amer. Statist. Assoc*, 95(452):1063–1075, 2000.

[4] C. S. Chao, D. L. Yang, and A. C. Liu. An automated fault diagnosis system using hierarchical reasoning and alarm correlation. volume 9, pages 183–202, 2001.

[5] R. H. Deng, A. A. Lazar, and W.Wang. A probabilistic approach to fault diagnosis in linear lightwave networks. In *In Integrated Network Management III*, pages 697–708, Apr. 1993.

[6] L. Fang, A. Atlas, F. Chiussi, K. Kompella, and G. Swallow. LDP failure detection and recovery. *IEEE Communications magazine*, 42(10):117–123, Oct. 2004.

[7] A. Gunnar, M. Johansson, and T. Telkamp. Traffic matrix estimation on a large ip backbone: A comparison on real data. In *Proc. ACM Internet Measurement Conference*, October 2004.

[8] P. Hong and P. Sen. Incorporating non-deterministic reasoning in managing heterogeneous network. In *In Integrated Network Management II*, pages 481–492, Apr. 1991.

[9] S. Kandula, D. Katabi, and J. P. Vasseur. Shrink: A tool for failure diagnosis in IP networks. In *Proc. ACM SIGCOMM MineNet Workshop*, Aug. 2005.

[10] R. Kompella, J. Yates, A. Greenberg, and A. C. Snoeren. IP fault localization via risk modeling. In *Proc. Networked Systems Design and Implementation*, May 2005.

[11] G. Liu, A. K. Mok, and E. J. Yang. Composite events for network event correlation. In *Integrated Network Management IV*.

[12] J. Mahdavi, V. Paxson, A. Adams, and M. Mathis. Creating a scalable architecture for internet measurement. In *Proceedings of INET'98*, 1998.

[13] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic matrix estimation: Existing techniques and new directions. In *ACM SIGCOMM*, Pittsburg, USA, August 2002.

[14] H. X. Nguyen and P. Thiran. Active measurement for multiple link failures diagnosis in IP networks. Juan Les Pins, France, April 2004.

[15] A. Nucci, R. Cruz, N. Taft, and C. Diot. Design of IGP link weights for estimation of traffic matrices. In *IEEE Infocom*, Hong Kong, March 2004.

[16] Y. A. Nygate. Event correlation using rule and object based techniques. In *Integrated Network Management*, pages 278–289.

[17] V. Paxson. End-to-end Internet packet dynamics. *IEEE/ACM Transactions on Networking*, 7(3):277–292, 1999.

[18] P.Wu, R. Bhatnagar, L. Epshtein, M. Bhandaru, and Z. Shi. Alarm correlation engine (ACE). In *In Proc. of Network Operation and Management Symposium*, pages 733–742, New Orleans, LA, 1998.

[19] P. Sebos, J. Yates, D. Rubenstein, and A. Greenberg. Effectiveness of shared risk link group auto-discovery in optical networks. In *Proc. Optical Fiber Communication Conference*, Mar. 2002.

[20] J. Sommers, P. Barford, N. Duffield, and A. Ron. Improving accuracy in end-to-end packet loss measurement. In *Proceedings of ACM SIGCOMM*, Aug. 2005.

[21] M. Steinder and A. Sethi. Increasing Robustness of Fault localization through Analysis of Lost, Spurious and Positive Symptoms. In *IEEE Infocom*, 2002.

[22] C. Tebaldi and M. West. Bayesian inference on network traffic using link count data. *J. Amer. Statist. Assoc*, 93(442):557–576, 1998.

[23] Y. Vardi. Network tomography: estimating source-destination traffic intensities from link data. *J. Am. Statist. Assoc.*, 91:365–377, 1996.

[24] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale ip traffic matrices from link loads. In *Proc. ACM SIGMETRICS 2003*, June 2003.

[25] Y. Zhang, M. Roughan, C. Lund, and D. Donoho. An information-theoretic approach to traffic matrix estimation. In *Proc. ACM SIGCOMM*, August 2003.