Shrink: A Tool for Failure Diagnosis in IP Networks

Srikanth Kandula
MIT CSAIL
kandula@mit.edu

Dina Katabi
MIT CSAIL
dk@mit.edu

Jean-Philippe Vasseur
Cisco Systems
jvasseur@cisco.com

ABSTRACT

Faults in an IP network have various causes such as the failure of one or more routers at the IP layer, fiber-cuts, failure of physical elements at the optical layer, or extraneous causes like power outages. These faults are usually detected as failures of a set of dependent logical entities–the IP links affected by the failed components. We present Shrink, a tool for root cause analysis of network faults which, given a set of failed IP links, identifies the underlying cause of the faulty state. Shrink models the diagnosis problem as a Bayesian network. It has two main contributions. First, it effectively accounts for noisy measurement and inaccurate mapping between the IP and optical layers. Second, it has an efficient inference algorithm that finds the most likely failure causes in polynomial time and with bounded errors. We compare Shrink with two prior approaches and show that it substantially improves the performance.

Categories and Subject Descriptors

C.2.3 [**Computer Communication Networks**]: Network Operations

General Terms

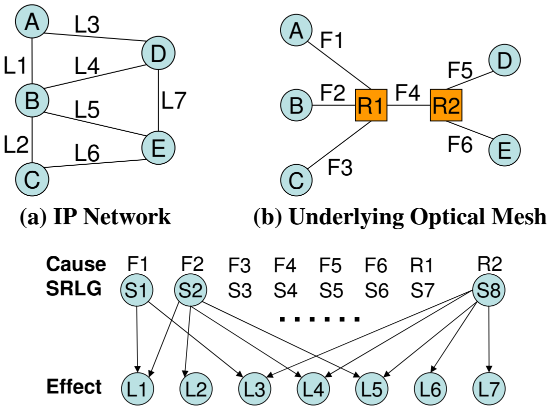
Algorithms, Design, Management, Reliability, Performance

Keywords

Shrink, Fault Diagnosis, IP networks, Optical, SRLG, Bayesian

1. INTRODUCTION

This paper addresses the problem of failure diagnosis using indirect and potentially noisy measurements. Modern ISP networks consist of thousands of routers, optical cross-connects, repeaters, several tens-of-thousands kilometers of optical fiber, and a variety of software modules. Such complex systems fail often [3, 5, 11], e.g., fiber-cuts, optical cross-connect failures, power outages, faulty amplifiers, bugs in the routing code. They also fail in complex ways [4, 12], e.g., a faulty amplifier increases loss rate of a link, TCP throughput of flows using that link suffers, this causes the BGP sessions traversing the link to timeout. A recent study at a major ISP [6] shows that failures are a part of the ISP's everyday operation and last between several minutes to several days before repair.



(a) IP Network **(b) Underlying Optical Mesh**

(c) SRLG – IP mapping

Figure 1: Problem Setup: IP links $L_1, L_2, \ldots L_7$ are realized using an underlying optical mesh consisting of fibers $F_1, \ldots F_6$ and optical cross-connects R_1, R_2. An SRLG is associated with each fiber/cross-connect that can fail independently, and consists of all the IP links affected by this failure. For example, if fiber $F1$ gets cut both IP-links $L1$ and $L3$ fail simultaneously and hence, the SRLG corresponding to $F1$ is $S1 = \{L1, L3\}$.

Faults in the IP and optical layers are correlated. When a physical object such as a fiber span, or a software module fails, it causes the simultaneous failure of a group of logical entities at the IP layer, e.g. all IP links that use this object. Shared Risk Link Groups (SRLGs) [11] are an easy way to represent this correlation. Formally, an SRLG is a group of logical entities that share risk due to their common dependence on a physical object. A network fault occurs when one or more SRLGs fail but is often detected as a failure of the set of dependent logical entities–the constituent IP links.

This paper presents Shrink, a network diagnosis system for failures in IP networks. Similarly to prior work [5, 10], Shrink adopts a *top-down* approach to the problem, i.e., given observations of IP-link status, Shrink pinpoints the SRLGs responsible for the fault. The alternate approach of diagnosing such failures in a bottom-up fashion is hard because precise status information for each entity that can fail is often unavailable (e.g. passive amplifiers, bugs in the routing code) [9]. Even if present, this information is spread across different administrative domains (as in the case of leased sub-networks), different logs (SONET logs, IP router logs) or different locations in the network [5]. Joining dynamic fault data from

SIGCOMM'05 Workshops, August 22–26, 2005, Philadelphia, PA, USA.
Copyright 2005 ACM 1-59593-026-4/05/0008 ...$5.00.

Term	Meaning
SRLG Description	The set of IP links that belongs to an SRLG, as reported by a database of such mappings. Descriptions may be erroneous i.e., contain IP links that are not in the SRLG or miss IP links that are actually in the SRLG.
True SRLG	The actual set of IP links present in the SRLG.
Marginal prob. of an SRLG's failure	The probability that the event represented by an SRLG, e.g. fiber cut, happens. Note, SRLGs capture independent events.
Network Fault F	State of the network when one or more SRLGs have failed. It is observed, however, as the failure of a set of dependent IP links.
Diagnosis F^d	A list of SRLG failures that explains the network fault (explains the IP link failures).

Table 1: Terminology and variables used in the paper.

multiple sources is complicated because the network elements usually support different standards (e.g. SONET, MPLS etc.) [5, 11]. At the same time, if some physical elements can report their failure status, Shrink uses this information to reduce the size of the problem and yield a more accurate diagnosis.

Identifying physical causes of failures from IP-link status is challenging for two reasons.

- **Under-Determinedness:** Mapping an IP link failure to an underlying physical cause is not easy because the failure could happen due to many causes, e.g. a particular link could fail because the underlying fiber got cut, the attached router port went down, or a cross-connect failed. Thus, fault localization of IP failures is an under-determined problem because the available information is insufficient for identifying the failed SRLGs.

- **Inaccurate Information:** ISPs maintain a database describing the mapping between SRLGs and their constituent IP links, but this is often inaccurate [5, 9]. The same IP-link might be mapped onto different optical layer paths at different times due to network upgrades, traffic engineering, or automatic recovery mechanisms such as SONET/MPLS Fast Reroute. Whenever the IP-link gets re-routed, the mapping needs to be updated and errors can creep in easily. As a result, the SRLG database often contains stale or incorrect descriptions of the SRLGs [5, 9, 11] (e.g., fiber $F1$, in Fig. 1b, carries the traffic of links $L1$ and $L3$, but the database may say that $F1$ carries the traffic of $L1$ and $L2$). Furthermore, when an IP link fails, the SNMP message reporting the failure may be dropped because of congestion or because the network is disconnected [5, 10]. We refer to the available, potentially inaccurate, SRLG-IP mapping as the *SRLG description* to distinguish it from the *true SRLG*.

A standard model of the fault diagnosis problem exists [5, 10]. Figs. 1a and 1b show an example IP network and its realization using an underlying optical mesh. Fig. 1c models the SRLG-IP mapping using a bipartite graph. For each SRLG and IP-link, there is a corresponding vertex in the graph. Each SRLG vertex is connected by directed edges to its constituent IP-link vertices. A failure in an SRLG results in failure of all IP-links whose vertices are connected to the SRLG vertex in the bipartite graph. Not all SRLGs are equally likely to fail; e.g., cuts are more likely in long fibers than in short fibers. Hence, each SRLG has an associated marginal probability of failure. Further, SRLGs correspond to physical entities that fail independently, e.g., whether or not the New York-Boston fiber got cut says nothing about the London-Paris fiber. Hence, SRLG failures are assumed independent.

Prior work on cross IP and optical layers fault localization follows two approaches. The first is the *MinSetCover* technique [5]. Consider the SRLG-IP bipartite graph in Fig. 1c. Note that each failed IP link has to be connected to some SRLG whose failure can explain its fault. Any set of SRLGs whose failure produces the faulty state forms a set-cover over the IP links that have failed. As explained earlier, the problem is under-determined and usually many SRLG sets can explain the IP failures. The MinSetCover technique guesses that the smallest set-cover over the failed links is likely to be the set of failed SRLGs. Since the minimum set cover problem is NP-complete, the authors of [5] obtain a *minimal* set-cover by greedily picking, at each step, the SRLG that explains the largest number of failed IP-links until all IP-links are explained. The MinSetCover technique has two limitations. First, it ignores the fact that SRLGs have different probabilities of failure. Second, though it tries to deal with inaccuracies in the SRLG description and the reported link status, it does not provide a systematic approach to address the issue.

The second technique, *BayesNet* [10], models the SRLG-IP mapping as a Bayesian network.[1] Failure diagnosis translates to the inference problem in Bayesian networks–given observations of certain random variables (IP link status), infer the most likely assignment of values for the remaining random variables (the SRLG status). In contrast to the MinSetCover approach, the BayesNet technique acknowledges that different SRLGs have different probabilities of failure. It uses these probabilities to address the under-determinedness of the problem; from all possible SRLG sets whose failure could have caused the observed IP links to fail, it returns the SRLG set that is most likely. Despite its appeal, the BayesNet approach has some significant practical limitations. First, it does not deal with inaccurate SRLG descriptions. Second, the general inference problem in Bayesian networks is NP hard [2]. Approximate polynomial-time belief propagation techniques exist for some special cases [8], but these techniques are fairly inaccurate [10] over realistic failure diagnosis models. Indeed, the designers of the BayesNet technique acknowledge that the scheme scales only to about 50 nodes [10], while practical networks consist of thousands of network elements.

Contributions: Shrink modifies and extends the Bayesian network technique in several ways to significantly improve diagnosis time and accuracy. It has these key contributions:

(1) Shrink is the first IP fault diagnosis system that efficiently and systematically accounts for inaccuracies in SRLG descriptions. In our simulations, Shrink yields predictions with fewer than 2% error even when SRLG descriptions are incorrect or SNMP reports get dropped. In similar situations, BayesNet has 40% error and MinSetCover has 20% error.

(2) In contrast to prior attempts to use Bayesian networks for network fault diagnosis [10], Shrink solves the problem in

[1] Bayesian networks are graphical models that succinctly represent a joint probability distribution over random variables using their independence information. The bipartite graph representing SRLG-IP mapping in Fig. 1c can be extended to a Bayesian network by annotating it with certain probabilities (See §2).

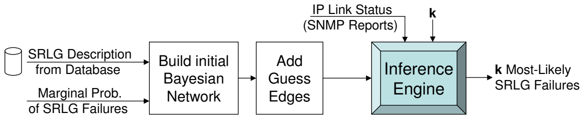

SRLG Description from Database
Marginal Prob. of SRLG Failures
Build initial Bayesian Network
Add Guess Edges
IP Link Status (SNMP Reports)
k
Inference Engine
k Most-Likely SRLG Failures

Figure 2: Shrink System Setup

polynomial time $\sim O(n^4)$, returns the correct answer with high probability and bounded error, and scales to thousands of network elements. In our simulations, Shrink takes fewer than $4s$, to diagnose a fault. In similar situations, BayesNet takes an average of $600s$ and MinSetCover takes about $.01s$.

(3) We are the first to present preliminary comparisons with the two main bodies of related work: MinSetCover [5] and BayesNet [10].

Finally, we note that although this paper focuses on failure diagnosis across IP and optical layers, Shrink is a fairly general tool for network fault diagnosis in environments with noisy measurements and incomplete information.

2. SHRINK

Shrink is a failure diagnosis tool. Table 1 describes our **terminology and variables**. Shrink takes as **input** (Fig. 2):

(1) A description of the SRLG-IP mapping. This is a potentially inaccurate list of SRLGs and their constituent IP links (e.g., $\{S_1 = \{L_1, L_2\}, S_2 = \{L_3\}\}$).

(2) If available, Shrink takes the marginal failure probability of each SRLG. Otherwise, it assumes all SRLGs are equally likely to fail.

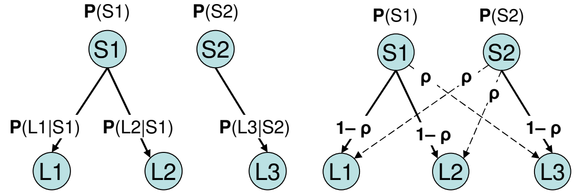
(3) The status of IP links in the network; a subset may *report* failure, some report liveness, and others have unknown state (e.g., $\{L_1, L_2\}$ are up, L_3 is down, no report from L_4).

Shrink **outputs** the most likely explanation for the network's faulty state–i.e., the collection of SRLGs whose failure is most likely given the IP link status. It can also output the k-most likely collections of SRLG failures, for integer $k > 1$.

Shrink has 3 main modules as shown in Fig. 2: (1) building the Bayesian model; (2) augmenting the model with guess edges; (3) inferring the most likely explanation (or k explanations). Below, we explain each of these modules in detail.

2.1 Building the Bayesian Network

Our Bayesian network is a bipartite graph, an example of which is shown in Fig. 3a. The graph consists of three parts. First, each vertex in the graph represents a random variable; an SRLG vertex S_i gets the value 1 if the corresponding SRLG fails and 0 otherwise; an IP-link vertex L_j gets the value 1 if the IP-link reports failure, and 0 if the link reports liveness. Second, the graph is annotated with the marginal probabilities that each SRLG fails $P(S_i = 1)$. An SRLG fails independently of other SRLGs and the marginal probabilities may vary by several orders of magnitudes. Third, the graph tries to capture with edges the dependencies between the random variables. An SRLG vertex is connected with directed edges to each of the IP-links in its description denoting that a failure of the SRLG would lead to failures of these IP-links. An edge from SRLG S_i to link L_j is weighted by the conditional probability of link L_j failing given that SRLG S_i has failed, i.e., $P(L_j = 1|S_i = 1)$. The larger this conditional probability, the stronger the dependency between the SRLG and the IP-link failure.



(a) Initial Bayesian Network **(b) With Guess Edges**

Figure 3: Shrink's Network Model: An SRLG S_i is annotated with its marginal probability of failure $P(S_i = 1)$ and the directed edge $S_i \rightarrow L_j$ is annotated with the conditional probability $P(L_j = 1|S_i = 1)$. Shrink adds *guess edges* to deal with inaccuracies in SRLG descriptions.

When the conditional probability is not available, Shrink sets it to 1, i.e. assumes certainty.[2]

2.2 Augmenting the Model to Deal with Inaccuracies

We augment the initial Bayesian network to deal with potential inaccuracies in the SRLG database description and the possibility of dropped SNMP link status reports. Both inaccuracies have the same effect; they add noise to the dependency between an observation (link failure) and its explanation (the failure of a parent SRLG). We model this noise by modifying the initial Bayesian network as follows: (1) to account for the possibility that a description might be missing some links, we add low-probability *guess* edges between an SRLG and links that are not part of its database description; (2) to deal with lost link status and potential incorrect links in an SRLG description, we make the probability that the parent SRLG's failure leads to the IP link failure being *reported* slightly smaller than 1. As a result of these modifications, the bipartite graph defined in §2.1 becomes a complete graph, with edge weights, i.e., conditional probabilities[3], as follows:

$$P(L_j = 1|S_i = 1) = \begin{cases} 1 - p, & \forall L_j \in S_i \\ p, & \text{otherwise,} \end{cases}$$

where $L_j \in S_i$ means that the database description states that link L_j is part of SRLG S_i. Fig. 3b shows the augmented model of the example Bayesian network in Fig. 3a.

The parameter p is a small noise probability whose default value is $p = 10^{-4}$. The system is insensitive to the exact value of p. Introducing a small amount of uncertainty/noise allows us to search over many more possible failure explanations, especially those with inaccurate descriptions, that might otherwise have been infeasible. On the other hand, the uncertainty is small enough that, in general, a diagnosis that has one or more *guess* edges would have a lower probability of happening than one that has all edges acknowledged in the description. The cost we pay for this augmentation is that solving the model becomes exponential, as explained in §2.3.

2.3 Inferring a Diagnosis

Given an observation of the state of the IP links $(L_1, L_2, ..., L_m)$, the inference algorithm finds the most likely assignment of values to the SRLG random variables $(S_1, S_2, ..., S_n)$, i.e.:

$$\arg \max_{S_1,\ldots,S_n} P(S_1, \ldots, S_n|L_1, \ldots, L_m),$$

where $L_j \in \{0, 1\}$ and $S_i \in \{0, 1\}$.

[2] To the interested user, we note that a complete description of the Bayesian network involves specifying the joint probability distribution of a node together with *all of its parents*. Shrink uses the popular *noisy-or* distribution [10].

[3] Have to be appropriately normalized if $P(L_j = 1|S_i = 1) \neq 1$.

Although a few *standard inference algorithms* exist, they have an exponential running time on our augmented graphs (see §2.2). In particular, the exact junction-tree algorithm [8] typically used for inference in Bayesian networks takes exponential time. Belief propagation [8] and Loopy Belief Propagation either do not apply or have exponential running time on complete bipartite graphs, such as our augmented graphs.

Shrink's Inference Algorithm: We develop an approximation algorithm that performs inference on our augmented graphs in small polynomial time, and yields exact results with high probability. Our algorithm is based on a simple observation. In realistic environments, the probability of an SRLG failure is significantly low; the probability that a router fails in any given hour is about 10^{-6}, the prob. that a 1000Km-long fiber gets cut in any given hour is about 10^{-4} [11]. Hence it is very likely that a network fault is caused by failures of *a small number* of SRLGs. In fact, if E_κ denotes the event that at most κ SRLGs failed within any hour, there are exactly $\binom{n}{\kappa}$ possible SRLG assignments that satisfy E_κ, where n is the total number of SRLGs. Our algorithm hypothesizes that at most κ SRLGs have failed. Given the observed link status, our algorithm greedily picks the most likely SRLG assignment in E_κ; i.e., it picks $(S_1, ..., S_n) \in \{0,1\}^n$ such that:

$$\arg\max_{S_1,\ldots,S_n} P(S_1,\ldots,S_n|L_1,\ldots,L_m),$$

subject to

$$number\ of\ \{S_i = 1\} \leq \kappa.$$

Running Time: There are fewer than n^κ value assignments in E_κ, and for each assignment $P(S_1,\ldots,S_n|L_1,\ldots,L_m)$ can be computed in $O(m+n)$ time, where m is the number of IP links and n is the number of SRLGs. So, Shrink's running time is $O(n^\kappa * (m+n))$. We use $\kappa = 3$ in all our results, as a nice trade-off between increasing the running time and lowering the probability of errors as discussed next.

Bounding the Errors: Compare our greedy inference algorithm described above with a brute force inference algorithm that computes $P(S_1,\ldots,S_n|L_1,\ldots,L_m)$ for all 2^n SRLG assignments, to find the most likely assignment given the link status. The brute force algorithm provides optimal inference but has exponential complexity. How worse are the errors in our greedy algorithm in comparison to the brute force algorithm? Recall that our algorithm hypothesizes that at most κ SRLGs have failed. Two cases exist: (1) either our hypothesis is true–i.e., the link failures are truly caused by κ or fewer SRLG failures; (2) or our hypothesis is false–i.e., the link failures are caused by strictly more than κ SRLG failures. If our hypothesis is true then our algorithm performs at least as well as the brute force algorithm as it searches for the most likely SRLG assignments in a space where it is guaranteed to exist. If our hypothesis is false, then our algorithm performs worse than the brute force algorithm. Thus, the probability that our algorithm yields worse than optimal inference is bounded by $1 - P(Hypothesis)$, i.e., the probability that more than κ SRLGs fail. This probability decreases exponentially with κ. The default value $\kappa = 3$ is valid for a wide-range of ISP network sizes and SRLG failure probabilities. In particular, for typical ISPs with up to $10,000$ SRLGs, and realistic SRLG failure probabilities of about 10^{-5} [11], the probability of Shrink being worse than the brute force algorithm is always less than 10^{-4} and often less than 10^{-8}.

3. EVALUATION

We evaluate Shrink in simulation and compare it with prior approaches, MinSetCover and BayesNet.

3.1 Experiment Setup

True SRLG-IP mapping: Similarly to prior work [5, 10], we create an SRLG-IP database, by first generating two network graphs with node degree distributions either Waxman or Albert-Barabasi, using the BRITE [7] generator. One of the graphs serves as the IP topology, with nodes representing IP routers joined by IP links. The other graph represents the underlying optical mesh, with "optical cross-connect" nodes joined by fiber. Second, we join each IP node to the closest optical cross-connect and map an IP link between two nodes in the IP graph onto the shortest path through the optical graph between these IP nodes. Finally, we assign an SRLG to each optical element (fiber/cross-connect), and associate the SRLG with all IP links whose paths traverse this element. This process is repeated multiple times to generate many graphs.

Assigning marginal probabilities to SRLGs: We use mean time to failure (MTTF) statistics from real ISPs [11]. An optical cross-connect has MTTF $[10^5, 10^6]$hrs, a 1000Km fiber has MTTF $[1200, 4800]$hrs. We assume that the time until first failure is exponentially distributed with $\lambda = \frac{1}{MTTF}$. We define the diagnosis interval, T, as a window of time during which the ISP collects SNMP link status. We approximate the probability that an SRLG fails in a given diagnosis interval by λT. T has to be at least a few RTTs. On the other hand, the larger T, the more likely multiple SRLGs fail in T, and the harder the diagnosis problem becomes. Since Shrink diagnoses faults in less than 4s, the ISP can continuously run Shrink with intervals $T \approx 4s$. To stress Shrink, however, we evaluate it over diagnosis intervals of about an hour.

Creating Network Faults: To generate a network fault, we allow each True SRLG to independently fail with its marginal probability causing all constituent IP links to fail. Our experiments report averages over 1000 such faults for each graph.

SRLG descriptions: We create realistic SRLG descriptions by probabilistically adding a few IP-links that are not in the true SRLG and removing a few IP-links that are in the SRLG. The probability of adding or removing i IP-links to an SRLG is q^i. We use $q \in [.05, .5]$. Note that our process of introducing description errors is conservative. First, a significant fraction of SRLGs remain unchanged for small q. Second, if an SRLG description is inaccurate, it is very likely to have only a few erroneous additions/deletions, as the probability decreases geometrically with the number of errors.

Alternative Approaches: We compare with two main classes of solutions for the failure diagnosis problem.

(a) MinSetCover: We implement the algorithm described in [5], making sure that all configurable parameters are adjusted to their default values.

(b) BayesNet: We implement the loopy belief propagation inference algorithm using the standard Bayesian Network Toolkit (BNT) for Matlab [1].

3.2 Metrics

We define the success rate of an inference algorithm, as the percentage of network faults that are correctly diagnosed, i.e.:

$$\textbf{Success Rate} = \frac{\#\text{Correct Diagnoses}}{\#\text{All Diagnoses}}.$$

A combination of SRLG failures creates a faulty network state. When the diagnosis is incorrect, there may be some SRLGs that haven't failed but are incorrectly diagnosed as failures, i.e., false-positives, and some SRLGs that did fail but weren't diagnosed as failures, i.e., false-negatives. Let F be the set of SRLGs that did fail, and $\overline{F}$ the set of SRLGs that did not fail. Also let F^d be the set of SRLGs that are diagnosed to have failed, and $\overline{F^d}$ the set of SRLGs diagnosed as up. Then:

$$\textbf{False-Positive Rate} = \frac{1}{\#\text{Diagnoses}} \sum_{\text{Diagnosis}} \frac{|F^d \cap \overline{F}|}{|F|},$$

$$\textbf{False-Negative Rate} = \frac{1}{\#\text{Diagnoses}} \sum_{\text{Diagnosis}} \frac{|F \cap \overline{F^d}|}{|F|}.$$

3.3 Evaluation Results

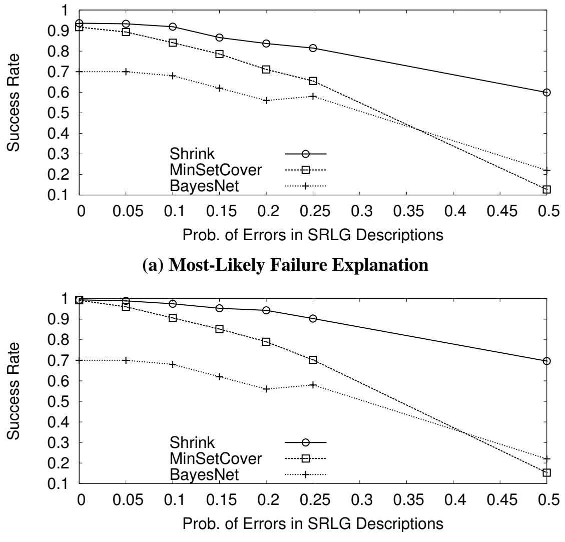
(a) Diagnosis Accuracy for Realistic Models: Fig. 4 compares the diagnosis accuracy of Shrink, MinSetCover and BayesNet as a function of the inaccuracy in SRLG-IP mapping. The network has 1000 nodes and 1611 SRLGs. The faults are generated based on the True SRLGs, but the algorithms use inaccurate SRLG descriptions.

First, note that Shrink outperforms both MinSetCover and BayesNet, and the performance gap increases with increased inaccuracy in the SRLG-IP mapping. Second, by comparing 4a against 4b, note that allowing the diagnosis tool to give the two most likely diagnoses increases the success rate substantially. This is a practical option because, in most cases, the objective of automated diagnosis is to pin down the failed SRLGs to a small set that can be checked and fixed manually. Third, the success rate of Shrink makes it a practical tool; Fig. 4b shows that for small inaccuracy probability $\sim$.1, Shrink diagnoses more than 99.5% of the faults correctly. When half the SRLGs are inaccurate, Shrink uses *guess edges* to diagnose up to 80% of the faults correctly. Finally, note that even when presented with the correct SRLG-IP mapping, BayesNet has about 30% error. This is because on several instances, BayesNet takes a very long time to converge, but we stop the runs at $2000s$ ($\sim$30min) (see Fig. 7).

(b) Understanding Performance: Fig. 5 compares the false-positive and false-negative rate for Shrink, MinSetCover, and BayesNet. Several interesting points are worth noting. First, BayesNet has a false-positive rate much larger than its false-negative rate. This is expected because this scheme ignores inaccuracies in the SRLG-IP mapping, and as a result, it prefers to include in its diagnosis SRLGs that explain a small subset of the failed IP links rather than leaving failed links un-explained. Second, Shrink & MinSetCover have similar false-positive and false-negative rates because they explicitly prefer answers with a small number of SRLGs, thereby having a roughly even probability of including SRLGs that failed and leaving out SRLGs that did fail. Finally, Shrink's false-positive and false-negative rates are 1-2 orders of magnitude lower than the other schemes, showing that even when Shrink makes an incorrect diagnosis, many SRLGs in the diagnosis are correctly diagnosed.

Shrink can provide multiple diagnoses, ordered by their likelihood. Fig. 6 attempts to understand Shrink's performance in this situation. This figure shows that Shrink's success rate improves considerably by considering just one extra diagnosis. Further, success rate quickly converges; considering more than the four most likely diagnoses yields little improvement in success rate.

(c) Running Time: Fig. 7 shows the time to diagnose a fault, averaged over several thousand failure instances, for topologies ranging from a 100 link, 118SRLG graph to a 3000 link, 3877 SRLG graph, on a 2GHz, 1GB RAM machine. (The probability of errors in the SRLG descriptions is 0.2 but the exact value of this parameter has



(a) Most-Likely Failure Explanation

(b) Two Most-Likely Failure Explanations

Figure 4: Diagnoses Accuracy. Figure compares how successful Shrink, MinSetCover, and BayesNet are in finding the correct failure diagnosis. Graphs are functions of increased inaccuracy in the SRLG-IP mapping. The network has 1000 nodes and 1611 SRLGs. Note Shrink has a much higher success rate than MinSetCover and BayesNet. Also, accuracy improves when each scheme provides the 2 most-likely diagnoses.

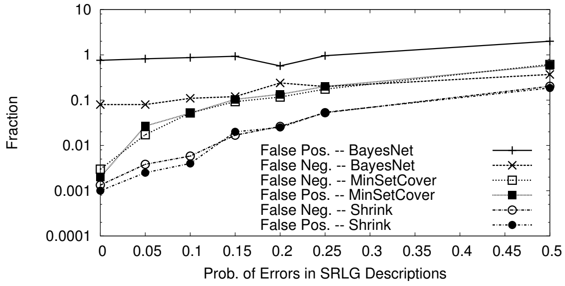


Figure 5: Nature of mistakes. Figure shows the false-positive and false negative errors as functions of increased inaccuracy in SRLG-IP mapping. The network has 1000 nodes and 1611 SRLGs. Shrink has 1-2 orders of magnitude fewer false-positives and false-negatives than MinSetCover and BayesNet.

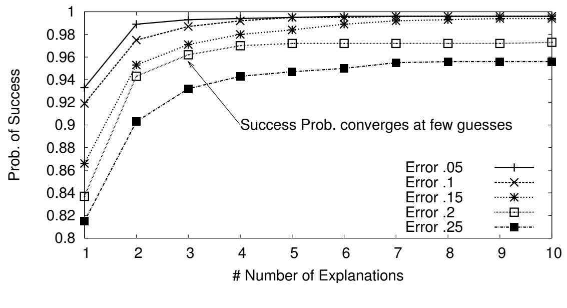


Figure 6: Multiple Diagnoses. Shrink provides multiple diagnoses ordered by their likelihood. This figure shows that success rate improves significantly by considering the two most likely diagnoses, and converges quickly at 4-most-likely diagnoses.

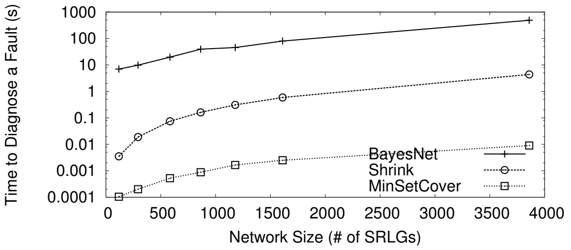


Figure 7: Time to diagnose a fault. Shrink is up to 3 orders of magnitude faster than BayesNet. Though Shrink is considerably slower than MinSetCover, Shrink diagnoses faults in $< 4s$ on reasonably large ISP networks and is much more accurate (Fig. 4).

no effect on the running time.) The graph shows that BayesNet takes up to 15 minutes to create a diagnosis. Though Shrink is slower than MinSetCover it diagnoses a fault in less than 4s even in large networks, and is remarkably more accurate (Fig. 4).

Note that the running time of both Shrink and MinSetCover approach straight lines asymptotically, confirming our analysis that their running times are polynomial in graph size. In general, the running time of BayesNet is exponential in the number of IP links that failed. However, we explicitly stop BayesNet, if it has been running for more than 30min. and return the best guess till then.

4. CONCLUSION & FUTURE WORK

We have presented Shrink, a tool for network fault diagnosis across the IP and optical layers, and showed that it has a better performance than previous methods. We applied Shrink to a specific problem, but Shrink is a general tool for fault diagnosis. Its main strength arises from its ability to deal with noisy measurements and inaccurate information. In the future, we would like to apply Shrink to other problem domains such as using the failure/success status of TCP connections to passively infer the failures status of various physical elements in an intra-net (routers, DNS/DHCP servers, etc.).

5. ACKNOWLEDGMENTS

We thank Arthur Berger, Allen Miu, Magdalena Balazinska, Rohit Singh, Sachin Katti, Eugene Shih and the anonymous reviewers for their constructive comments. We also acknowledge the support of Cisco Systems and the National Science Foundation under NSF Career Award CNS-0448287. The opinions and findings in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation or Cisco Systems.

6. REFERENCES

[1] Bayes Net Toolbox for Matlab. `http://www.cs.ubc.ca/~murphyk/Software/BNT`.

[2] G. F. Cooper. Probabilistic Inference using Belief Networks is NP-Hard. Technical Report KSL-87-27, Stanford, 1987.

[3] C. Ji and A. Elwalid. Measurement-Based Network Monitoring and Inference: Scalability and Missing Information. *IEEE JSAC*, 2002.

[4] I. Katzela and M. Schwartz. Schemes for Fault Identification in Communication Networks. *IEEE/ACM TON*, 1995.

[5] R. R. Kompella, J. Yates, A. Greenberg, and A. Snoeren. IP Fault Localization via Risk Modeling. In *NSDI*, 2005.

[6] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. N. Chuah, and C. Diot. Characterization of Failures in an IP Backbone Network. In *IEEE INFOCOM*, 2004.

[7] A. Medina, A. Lakhina, I. Matta, and J. Byers. BRITE: An Approach to Universal Topology Generation. In *MASCOTS*, 2001.

[8] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[9] P. Sebos, J. Yates, D. Rubenstein, and A. Greenberg. Effectiveness of Shared Risk Link Group Auto-Discovery in Optical Networks. In *Optical Fiber Commun. Conf.*, 2002.

[10] M. Steinder and A. Sethi. Increasing Robustness of Fault Localization through Analysis of Lost, Spurious and Positive Symptoms. In *INFOCOM*, 2002.

[11] J.-P. Vasseur, M. Pickavet, and P. Demeester. *Network Recovery: Protection and Restoration of Optical SONET-SDH, IP, and MPLS*. Morgan-Kauffmann, 2004.

[12] S. Yemini, S. Kliger, E. Mozes, Y. Yemini, and D. Ohsie. High Speed and Robust Event Correlation. *IEEE Communications Magazine*, 1996.