Shrink: A Tool for Failure Diagnosis in IP Networks

Srikanth Kandula

Dina Katabi and Jean-Philippe Vasseur
Failures in ISP Networks

- An ISP network:
  - Logically, the network is a set of IP links
  - Physically, the network consists of fiber, optical cross-connects, and amplifiers ...

- Failure at the IP layer are correlated with failures at the physical layer

- Failures are detected using SNMP messages that describe the state of IP links
Diagnosis Problem

- **Given,**
  - IP link status, a subset may have failed (logical failures), others are up
  - Database that maps IP links to underlying optical topology (physical components)

- Find the failed physical component(s)
Diagnosis Problem is Challenging (1)

- Mapping IP-link failures to underlying physical failures is an **under-determined** problem

Physical Components

- \( C_1 \)
- \( C_2 \)
- \( C_3 \)

IP Links

- \( L_1 \)
- \( L_2 \)
- \( L_3 \)

- When \( \{L_1, L_2\} \) fail, it is not clear which components have failed: either \( \{C_1\} \) or \( \{C_1, C_2\} \)
### Diagnosis Problem is Challenging (2)

- **Errors in database** that maps IP links to physical components
- **Measurement noise caused by lost SNMP reports**

![Diagram](Diagram_2.png)

- If `Edge(C1,L2)` is wrong, then `C1` and `C2` failed
- If report of L3 failure was lost $\Rightarrow \{C1,C3\}$ failed now valid
Prior Solutions

- **Min Set Cover**
  - Finds the smallest number of component failures that explain all IP link failures

- **Bayes Net Approach**
  - Takes into account that different components have different prob. of failure
  - Finds the most likely component failures given the IP link failures

**Our objective:**
Find a more accurate solution that deals better with database errors and measurement noise
This Talk

• Shrink
  - Explicitly deals with database errors and measurement noise
  - Uses rich probabilistic models
  - Fast Inference algorithm

• Simulation results show that Shrink is more accurate than MinSetCov and BayesNet
Shrink Setup

Inputs

- Possibly inaccurate IP-to-Optical database
- Marginal prob. of component failure
- SNMP reports of IP link status,
  - e.g. \{L1, L2\} are up, \{L3\} is down, no report from \{L4\}

Shrink

Output

Most likely subset of component failures given link status
Shrink Has 3 Modules

1. Building the Bayesian Network
2. Augmenting the model with guess edges to deal with database errors
3. Inferring a diagnosis
Module 1: Building a Probabilistic Model

- Two-level graph- components on top, links at bottom
- Connect component to all dependent IP links
- Assign prior probability of component failure (independent)
- For each edge, assign prob. of link failure given component failure if known (noisy-or model)
Module 2:
Sub-Problem: Errors in database \(\rightarrow\) Edge in model may not exist in reality (and vice-versa)

Solution: Augment the model with low-probability guess edges between un-connected components and IP links to deal with database errors

How does this help?
Why Augment with Guess Edges?

- Expands search to include explanations that were infeasible before, e.g. $P(C2|L2,L3)$
- Yet, prefer explanations that use few guess-edges, e.g., $P(C1|L1,L2) > P(C2|L1, L2)$

But, the augmented graph is complete $\rightarrow$ Standard Inference Algs. take exponential time
Module 3:

Shrink's Inference Algorithm

Likely that the correct explanation has only a small number of causes

$$\arg \max_{C_1, \ldots, C_n} P(C_1, \ldots, C_n \mid L_1, \ldots, L_m)$$

subject to

$$\text{number of } \{C_i = 1\} \leq q$$

Characteristics of Alg.:

• Polynomial time
• Bounded Error (for $q=3$, error is smaller than $10^{-4}$)
Putting it Together

Database Mapping
IP links to Physical Components

Marginal Prob. of Component Failures

IP Link Status (SNMP Reports)

Inference Engine

K Most-Likely Explanations

Build initial Bayesian Network
Add Guess Edges
Performance
Simulation Setup

- Both the physical and logical topologies are generated using the BRITE simulator
- Use known statistics of component failure probabilities
- Randomly pick the components that fail
- Insert errors in database by adding a small number of unrelated links or deleting related ones
Shrink is More Accurate than Prior Approaches
Despite exponential search space, Shrink’s inference algo. finds correct solution in a few seconds.
Shrink’s Contributions

- Augment Bayesian networks with guess-edges to model database errors
- Shrink’s Inference Alg identifies most likely failures within a few seconds
- Shrink is more accurate than prior work
- More general - replace components with SRLGs, mapping database with any other configuration database