A Probabilistic Approach for Failure Localization

Tania Panayiotou ¹, Sotirios P. Chatzis ², Georgios Ellinas ¹

¹KIOS Research and Innovation Center of Excellence, Department of Electrical and Computer Engineering, University of Cyprus, Cyprus

²Department of Electrical Engineering, Computer Engineering, and Informatics, Cyprus University of Technology, Cyprus

May 17, 2017
Introduction

- **General Objective:** Localize single-link failure in transparent optical networks

- **Specific Objective:**
  - Reduce the monitoring equipment (CAPEX)
  - Reduce the Mean-Time-To-Repair (OPEX)
• We focus in Transparent Optical networks where fault localization is not trivial.
Existing Fault Localization Methods

Generic Fault Localization Approach

- **Path Correlation**
- **Failure Detection**

**Link Identified?**

- **Yes** → **Repair** → **Link Identified**
- **No** → **Probing Lightpaths (Monitor Information)**

T. Panayiotou, S. P. Chatzis, G. Ellinas
Existing Fault Localization Methods

Path correlation procedures:

- May not unambiguously identify the faulty link
- Can effectively reduce the number of links being suspected of causing the failure

On-call engineers will have to resolve the problem (human effort, MTTR increases as the network grows)
Existing Fault Localization Methods

Probing Lightpaths (Monitor Information):

- Path correlation procedure complemented with monitoring information

![Diagram showing lightpaths and a failed link]

- Number of necessary monitoring equipment increases as the network grows (CAPEX increases)
- Bandwidth is required for fault localization (lightpaths established just for correlation purposes), affecting the network performance
Proposed Framework

- **Approach Overview:**
  - Path Correlation
  - Recover of the Traffic
  - Failure Detection
  - Link Identified?
  - Yes: Repair
  - No: Probabilistic Approach (Link Failure Prob.)

- **Advantages:**
  - Reduces the MTTR
  - Reduces the bandwidth required for fault localization purposes
  - Reduces the network cost (no monitors are assumed)
Proposed Framework: A. Path correlation

- **Graph Based Correlation Heuristic**

```
Algorithm 1 GBC heuristic

Input: The sets \( P \) and \( P' \), where \( P = \{P(i)|i = 1,....n\} \) and \( P' = \{P'(i)|i = 1,....n\} \).
Output: The set \( S \), where \( S = \{S(i)|i = 1,....n\} \).

1: for \( i = 1 \) to \( n \) do
2: \( A(i) = \bigcap_{m=1}^{t(i)} p_m(i) \)
3: \( A'(i) = \bigcup_{m=1}^{L(i)} p'_m(i) \)
4: if \( A(i) = \emptyset \) then
5: \( A(i) = p_1(i) \)
6: end if
7: \( S(i) = A(i) - (A(i) \cap A'(i)) \)
8: end for
9: return \( S \)
```

- Intersects the links utilized by the affected lightpaths
- Returns a set of suspect links
- Removes from the set of suspect links the links utilized by the unaffected lightpaths
- Returns the faulty link OR a set of suspect links
Proposed Framework: A. Path correlation

- **Example:** Graph Based Correlation Heuristic
  - Network properly working
  - Link B fails
  - GBC operation
    - \( \{A, B, C, D\} \cap \{B, C, D, E\} = \{B, C, D\} \)
    - \( \{B, C, D\} - (\{B, C, D\} \cap \{C, F\}) = \{B, C, D\} - \{C\} = \{B, D\} \)
    - Set of suspect links \( \{B, D\} \)
Proposed Framework: B. Probabilistic Approach

- **Approach Aim:** Generates a failure probability for each link suspected of causing the failure.

- **Approach Motivation:**
  - Optical related link failures are reported to follow the Weibull distribution
    \[ L_j \sim Wei(\lambda_j, \beta_j) \]
  - Link failures are time dependent
  - The class of GPs is one of the most widely used families of stochastic processes for modeling dependent data observed over time.
Proposed Framework: B. Probabilistic Approach

- Assumption:
  - $c_j(i)$: the number of times link $e_j$ has failed up to incident $i - 1$.
  - $C(i)$: the total number of failures up to incident $i$. 

```
  C(i)  
     /  
   /   
 e_1 (1) e_2 (2) e_3 (3) e_3 (4) e_3 (5) e_3 (6) ....... e_j (i - 1) 
     /  
   /   
 c_1 (i) ....... c_j (i) 
```

List of Past Failures
Proposed Framework: B. Probabilistic Approach

- **GP Classifier Formulation:** According to $c_j(\ast)$, $C(\ast)$, and according to the state of the network upon each failure incident.

  ▶ Training/test Dataset: $\mathcal{D} = \{(x(i), y(i))\}|i = 1, \ldots, n\}$

  $$x_j(i) = \begin{cases} 
  -\frac{c_j(i)}{C(i)}, & \text{if } e_j \in S(i) \\
  0, & \text{otherwise.}
  \end{cases} \quad \forall e_j \in E$$

  $$y_j(i) = \begin{cases} 
  1, & \text{if } e_j \text{ has failed at } i \\
  -1, & \text{otherwise}
  \end{cases} \quad \forall e_j \in E$$

  ▶ $n$: the total number of known failure incidents.
Proposed Framework: B. Probabilistic Approach

- **Prediction Generation:** GP classifier produces a probabilistic prediction for each link in the network.

\[
\pi \triangleq p(y(\star) = +1|X, y, x(\star)) = \int \sigma(f(\star))p(f(\star)|X, y, x(\star))df(\star)
\]  

- The failure probability is given by the posterior over the latent function \(\sigma(f(\star))\).
- Latent function \(f(\star)\) constitutes the basic mechanism of the GP model.
Proposed Framework: B. Probabilistic Approach

- **Model Formulation:** The inferred latent function \( f(x) \) over all the training inputs \( X = \{ x(i) \}_{i=1}^{n} \) and the test inputs \( x(*) \) yields:

\[
\begin{bmatrix}
  f(X) \\
  f(x(*))
\end{bmatrix} \sim \mathcal{N}
\left(\mathbf{0},
\begin{bmatrix}
  K(X, X) & k(x(*)) \\
  k(x(*)) & k(x(*), x(*))
\end{bmatrix}
\right)
\]  

(2)

\( k(x(*)) \triangleq [k(x(1), x(*)), \ldots, k(x(n), x(*))]^T \)

- Matrix of the covariances between the \( n \) training data points (gram matrix \( K \)):

\[
K(X, X) \triangleq 
\begin{bmatrix}
  k(x(1), x(1)) & k(x(1), x(2)) & \cdots & k(x(1), x(n)) \\
  k(x(2), x(1)) & k(x(2), x(2)) & \cdots & k(x(2), x(n)) \\
  \vdots & \vdots & \ddots & \vdots \\
  k(x(n), x(1)) & k(x(n), x(2)) & \cdots & k(x(n), x(n))
\end{bmatrix}
\]  

(3)
Proposed Framework: B. Probabilistic Approach

- **Model Training:** Kernel function: \( k(x(z), x(m)) \) (expresses the similarity between two data points \( x(k) \) and \( x(l) \)).
  - ARD kernel: Determines how relevant each input component is, thereby omitting input components that are deemed irrelevant.
    \[
    k(x(z), x(m)) = \theta_0 \exp\left\{-\frac{1}{2} \sum_{j=1}^{n} \eta_j (x_j(z) - x_j(m))^2\right\} \tag{4}
    \]
  - \( \theta_0, \{\eta_j\}_{j=1}^D \) are hyperparameters of the kernel function
  - The hyperparameters are optimized as part of the training procedure of the GP classifier (maximization of the log-likelihood of the model).
Enabling the Proposed Framework

- **Assumption:** A PCE element is present that is resource aware and is able to maintain a centralized TE database with detailed spectrum availability information.
Experimental Results

- Injected 10,000 failure incidents in a dynamic OFDM network (7,000 for training the GP, 3,000 kept for testing the GP).
- Requests follow the Poisson process with exponentially distributed holding times (a conventional RSA algorithm is utilized).
Experimental Results

Approach Accuracy vs Traffic Load

<table>
<thead>
<tr>
<th>Traffic load (Erlangs)</th>
<th>7</th>
<th>10</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td># Incidents in $D^{test}$</td>
<td>3000</td>
<td>3000</td>
<td>3000</td>
</tr>
<tr>
<td># Correctly Classified Incidents by GBC</td>
<td>1459</td>
<td>1816</td>
<td>2314</td>
</tr>
<tr>
<td># Incidents in $D^{test}_r$ (Passed to GP)</td>
<td>1541</td>
<td>1184</td>
<td>686</td>
</tr>
<tr>
<td># Correctly Classified Incidents by GP</td>
<td>1327</td>
<td>1068</td>
<td>655</td>
</tr>
<tr>
<td>GP Accuracy</td>
<td>0.86</td>
<td>0.90</td>
<td>0.95</td>
</tr>
<tr>
<td>Total Accuracy (GBC and GP)</td>
<td>0.93</td>
<td>0.96</td>
<td>0.99</td>
</tr>
</tbody>
</table>

- **Training Time:** Approximately 1 hour
- **Prediction Time:** Approximately 2 sec to classify a single incident
Experimental Results

- Examining how the $|\mathcal{D}^{train}|$ affects the GP accuracy

$|\mathcal{D}^{train}| = 5000$

<table>
<thead>
<tr>
<th>Traffic load (Erlangs)</th>
<th>7</th>
<th>10</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP Accuracy</td>
<td>0.84</td>
<td>0.89</td>
<td>0.96</td>
</tr>
<tr>
<td>Total Accuracy (GBC and GP)</td>
<td>0.93</td>
<td>0.95</td>
<td>0.99</td>
</tr>
</tbody>
</table>

$|\mathcal{D}^{train}| = 3000$

<table>
<thead>
<tr>
<th>Traffic load (Erlangs)</th>
<th>7</th>
<th>10</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP Accuracy</td>
<td>0.83</td>
<td>0.88</td>
<td>0.95</td>
</tr>
<tr>
<td>Total Accuracy (GBC and GP)</td>
<td>0.91</td>
<td>0.95</td>
<td>0.99</td>
</tr>
</tbody>
</table>

$|\mathcal{D}^{train}| = 1000$

<table>
<thead>
<tr>
<th>Traffic load (Erlangs)</th>
<th>7</th>
<th>10</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP Accuracy</td>
<td>0.84</td>
<td>0.88</td>
<td>0.94</td>
</tr>
<tr>
<td>Total Accuracy (GBC and GP)</td>
<td>0.92</td>
<td>0.95</td>
<td>0.98</td>
</tr>
</tbody>
</table>
• Examining how many monitors would be required for achieving the same accuracy as the one achieved by the proposed approach.
  ▶ The GBC heuristic is extended to the GBC heuristic with Monitors.

![Accuracy vs Number of Monitors Graph]

- 60% of the network nodes must be equipped with monitors.
Conclusions - Future Work

- Proposed fault localization scheme aims at reducing the MTTR (the human effort), and the CAPEX of the network.

- Two-step approach:
  - A. Path correlation procedure (GBC heuristic)
  - B. A probabilistic model is used (GP classifier)

- Achieved an overall high accuracy (93% – 99%) which is insignificantly affected by the number of training data.

- For achieving the same accuracy, as the one achieved by the proposed scheme (no monitors), it would require that 60% of the network nodes must be equipped with monitors.

- Future work: Scalability issues of the probabilistic approach as the network grows.