A Probabilistic Approach for Failure Localization

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

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

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

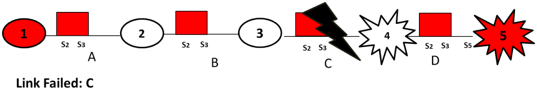
May 17, 2017

Introduction

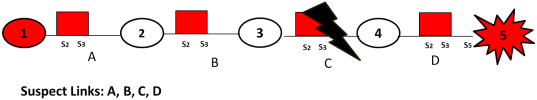
- **General Objective:** Localize single-link failure in transparent optical networks
- **Specific Objective:**
  - Reduce the monitoring equipment (CAPEX)
  - Reduce the Mean-Time-To-Repair (OPEX)

- We focus in Transparent Optical networks where fault localization is not trivial.

Link Failure in Opaque Network

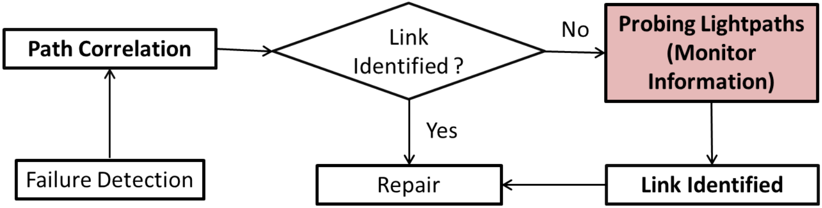


Link Failure in Transparent Network



Existing Fault Localization Methods

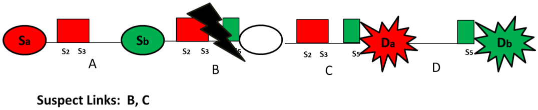
Generic Fault Localization Approach



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

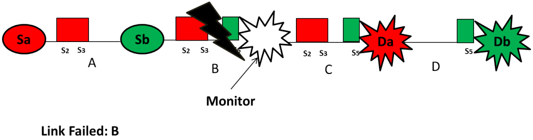


- On-call engineers will have to resolve the problem (human effort, MTTR increases as the network grows)

Existing Fault Localization Methods

Probing Lightpaths (Monitor Information):

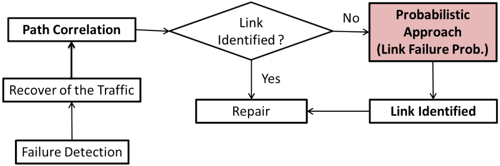
- Path correlation procedure complemented with monitoring information



- 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:**



- **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

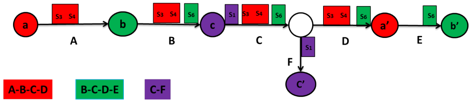
Algorithm 1 GBC heuristic

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 $S = \{S(i)|i = 1, ...n\}$.

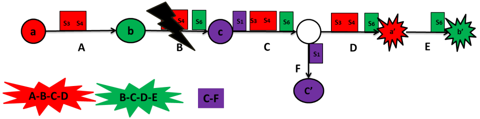
for $i = 1$ **to** n **do**
  $A(i) = \bigcap_{m=1}^{t(i)} p_m(i)$
  $A'(i) = \bigcup_{m=1}^{t'(i)} p'_m(i)$
  if $A(i) = \emptyset$ **then**
    $A(i) = p_1(i)$
  end if
  $S(i) = A(i) - (A(i) \bigcap A'(i))$
end for
return S

  - 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

- **Example:** Graph Based Correlation Heuristic
  - Network properly working



  - Link **B** fails



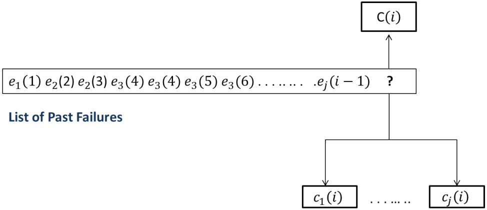
  - GBC operation
    - $\{A, B, C, D\} \cap \{B, C, D, E\} = \{B, C, D\}$
    - $\{B, C, D\} - (\{B, C, D\} \cap \{C, F\}) = \{B, C, D\} - \{C\} = \{B, D\}$
    - Set of suspect links $\{B, D\}$

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.

Proposed Framework: B. Probabilistic Approach

- **GP Classifier Formulation:** According to $c_j(*)$, $C(*)$, and according to the state of the network upon each failure incident.
  - Training/test Dataset: $\mathcal{D} = \{(\boldsymbol{x(i)}, \boldsymbol{y(i)})\} | i = 1, ..., n\}$

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

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

▷ 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.

$$\pi \triangleq p(y(*) = +1|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{x(*)})$$

$$= \int \sigma(f(*))p(f(*)|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{x(*)})df(*) \quad (1)$$

  - The failure probability is given by the posterior over the latent function $\sigma(f(*))$,
  - Latent function $f(*)$ 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 $\boldsymbol{X} = \{\boldsymbol{x(i)}\}_{i=1}^{n}$ and the test inputs $x(*)$ yields:

$$\begin{bmatrix} \boldsymbol{f(X)} \\ \boldsymbol{f(x(*))} \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \boldsymbol{K(X,X)} & \boldsymbol{k(x(*))} \\ \boldsymbol{k(x(*))} & k(\boldsymbol{x(*)}, \boldsymbol{x(*)}) \end{bmatrix}\right) \quad (2)$$

  - $\boldsymbol{k(x(*))} \triangleq [k(\boldsymbol{x(1)}, \boldsymbol{x(*)}), ..., k(\boldsymbol{x(n)}, \boldsymbol{x(*)})]^T$
  - Matrix of the covariances between the n training data points (*gram* matrix $\boldsymbol{K}$):

$$\boldsymbol{K(X,X)} \triangleq \begin{bmatrix} k(\boldsymbol{x(1)}, \boldsymbol{x(1)}) & k(\boldsymbol{x(1)}, \boldsymbol{x(2)}) & \dots & k(\boldsymbol{x(1)}, \boldsymbol{x(n)}) \\ k(\boldsymbol{x(2)}, \boldsymbol{x(1)}) & k(\boldsymbol{x(2)}, \boldsymbol{x(2)}) & \dots & k(\boldsymbol{x(2)}, \boldsymbol{x(n)}) \\ \vdots & \vdots & & \vdots \\ k(\boldsymbol{x(n)}, \boldsymbol{x(1)}) & k(\boldsymbol{x(n)}, \boldsymbol{x(2)}) & \dots & k(\boldsymbol{x(n)}, \boldsymbol{x(n)}) \end{bmatrix} \quad (3)$$

Proposed Framework: B. Probabilistic Approach

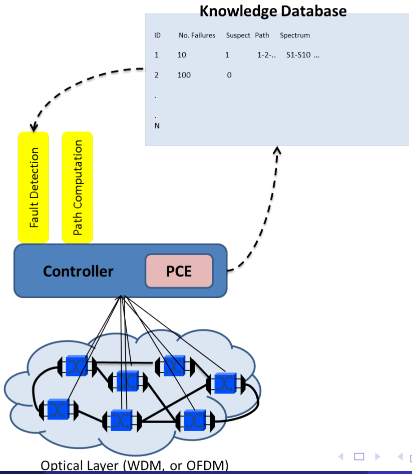
- **Model Training:** *Kernel* function: $k(\boldsymbol{x(z)}, \boldsymbol{x(m)})$ (expresses the similarity between two data points $\boldsymbol{x}(k)$ and $\boldsymbol{x}(l)$).
  - ARD kernel: Determines how relevant each input component is, thereby omitting input components that are deemed irrelevant.

$$k(\boldsymbol{x(z)}, \boldsymbol{x(m)}) = \theta_0 \exp\{-\frac{1}{2}\sum_{j=1}^{n} \eta_j (x_j(z) - x_j(m))^2\} \tag{4}$$

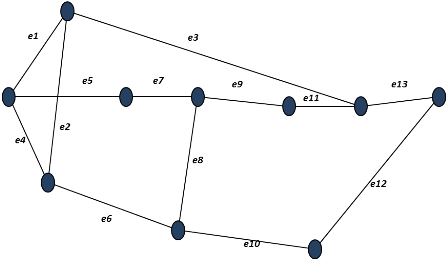
  - θ_0, $\{\eta_j\}_{j=1}^{D}$ are hyperparameters of the kernel function
  - The hyperparameters are optimized as part of the training procedure of the GP classifier (maximization of the log-likelihood of the model).

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



Link	Distance (km)	λ_j	β_j
e_1	1100	471	2.77
e_2	1600	202	1.55
e_3	2800	131	1.45
e_4	600	971	2.53
e_5	1100	321	1.62
e_6	2000	67	2.71
e_7	600	1454	2.1
e_8	800	229	1.77
e_9	800	629	2.22
e_{10}	1200	674	2.65
e_{11}	700	263	2.43
e_{12}	900	636	2.5
e_{13}	700	1094	2.1

- 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

Traffic load (Erlangs)	7	10	20
# Incidents in $\mathcal{D}^{test}$	3000	3000	3000
# Correctly Classified Incidents by GBC	1459	1816	2314
# Incidents in $\mathcal{D}_r^{test}$ (Passed to GP)	1541	1184	686
# Correctly Classified Incidents by GP	1327	1068	655
GP Accuracy	0.86	0.9	0.95
Total Accuracy (GBC and GP)	0.93	0.96	0.99

- **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

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

Traffic load (Erlangs)	7	10	20
GP Accuracy	0.84	0.89	0.96
Total Accuracy (GBC and GP)	0.93	0.95	0.99

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

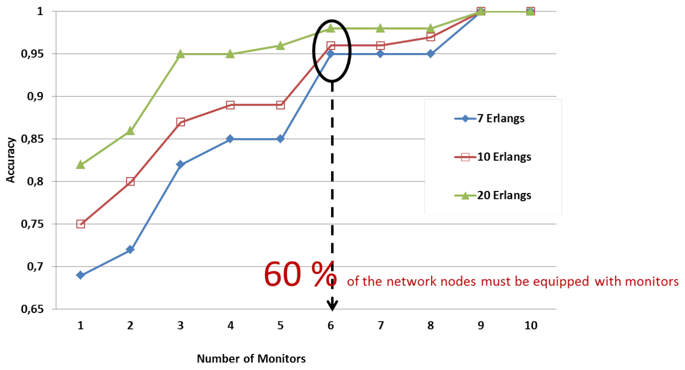
Traffic load (Erlangs)	7	10	20
GP Accuracy	0.83	0.88	0.95
Total Accuracy (GBC and GP)	0.91	0.95	0.99

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

Traffic load (Erlangs)	7	10	20
GP Accuracy	0.84	0.88	0.94
Total Accuracy (GBC and GP)	0.92	0.95	0.98

Experimental Results

- 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.



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.