An Experimental Design Approach for Link Loss Inference on Large Networks

Juan Li, Yan Qiao, Guanjue Wang, Xuesong Qiu
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing - 100876
Email: journeyy.lee@gmail.com

***Abstract*—An approach for link loss inference on large networks is proposed in this paper regarding with the cost of measurements. The measurements on large-scale networks usually cost much and the diagnosis of bottleneck on these networks are expensive and inefficient. We adapt Bayesian experimental design for measurement-path selecting with the total cost controlled in the budget to solve this problem. Through Bayesian experimental design with cost constrain, we can maximum the information getting from network measurements within the limited cost. Then the network inference method is used to infer the bottleneck link. The inference problem can be converted into a Linear Programming (LP) problem, which can be solved efficiently and accurately. We also carry out experiments to compare our approach with other approaches. The results show that with the same cost constrain, our approach is more accurate and has better performance.**

Keywords—link loss inference; experimental design ; Bayes optimal design

I. Introduction

The scale of network is increasing with the development of computer networks. There are many problems in large network performance measurement. One of the main problems is the large number of nodes, while the number of paths is square increasing of the increase of the number of network nodes, which makes it impossible to detect each path in the network. It requires efficiently selection of the paths with large amount of information rather than the paths with lower information.

The existing loss rate inference methods in large-scale networks are based on three ways: rank based, experimental design based and matrix decomposition based. Rand based methods [1][2] have the high degree of accuracy, but the number of paths need to detect equals to the rank of routing matrix, and this is still a very large number. So the probing overhead is too large; matrix decomposition based methods [3][4]need to detect a relatively small number of the path, but the downside is the error rate is high. Experimental design based methods [5] have a relatively smaller paths to detect. And in the case of the same detection paths, the accuracy is higher than matrix decomposition based methods, yet the disadvantage is the calculation process is time consuming.

To solve the above problems about inference loss rate in large-scale network, this paper designs an inference method which probes a smaller set of paths with shorter calculation time. The main contribution of this paper is as follows:

- Utilize Bayesian experimental design to select probing paths, which ensures optimal network path information.
- Utilize dynamic probing to significantly shorten the path selection time.
- Utilize linear programming method to calculate link loss rate, which is fast and accurate.

II. Problem Description

This paper defines the topology information as a tuple (L, A), where$\mathrm{L} = \{\mathrm{l}_1,\ \mathrm{l}_2, \cdots, \mathrm{l}_\mathrm{L}\}$ indicates the set of link. The definition of "path" is a set of links through which data stream travel from the input node to the output node. R represents the number of paths. The matrix $\mathrm{A} = [\mathrm{a_{ij}}]_{\mathrm{L}\times\mathrm{R}}$ is the routing matrix defined as follows:

$$\mathrm{A_{ij}} = \begin{cases} 1, & \text{if path i contains link j} \\ 0, & \text{otherwies} \end{cases} \tag{1}$$

$\widehat{\varphi_\mathrm{i}}$ represents the ratio that data can correctly reach the destination node on the path i, which is the pass rate of the path i. $\widehat{\varphi_{\mathrm{e_k}}}$ indicates the ratio that data passes through link $\mathrm{e_k}$ and has not been discarded, i.e. the pass rate of the link $\mathrm{e_k}$. Assuming the loss rate of links is independent of each other, and the loss rate of a path that goes through a link is approximately equal to the loss rate of that link. Let $\mathrm{Y_i} = \log\widehat{\varphi_\mathrm{i}}$, $\mathrm{X_k} = \log\widehat{\varphi_{\mathrm{e_k}}}$, so the relationship of routing matrix and vectors $\mathrm{Y} = (\mathrm{Y_1}, \mathrm{Y_2}, \cdots, \mathrm{Y_R})$, $\mathrm{X} = (\mathrm{X_1}, \mathrm{X_2}, \cdots, \mathrm{X_L})$ is as follows[5]:

$$\mathrm{Y} = \mathrm{AX} \tag{2}$$

The method can be divided into two parts: the selection of detection path and inference of the performance. The selection of detection path can be defined as follows. select subset $\mathrm{S} \subseteq \mathrm{P}$ from paths set $\mathrm{P}(|\mathrm{P}| = \mathrm{R})$. $\mathrm{A_S}$ represents the sub matrix of A with the size $\mathrm{s} \times \mathrm{L}$. Rows in matrix $\mathrm{A_S}$ correspond to s paths in the set S. $\mathrm{Y_S}$ is the sub vector of path performance Y, corresponding to the path in set s. So the selection of detection path is to select an subset S to probe. The performance inference part should infer the value of x through the probe data $\mathrm{Y_S}$ and detecting matrix $\mathrm{A_S}$ to determine the link loss rate. The linear relationship of these three is: $\mathrm{Y_S} = \mathrm{A_S X}$. The main

978-3-901882-50-0 ©2013 IFIP

problem in this part is that the probe data is not comprehensive, thus the linear equations $Y_S = A_S X$ is usually undetermined, and there is an infinite set of solutions.

III. Loss Rate Inference Algorithm

A. Path Selection

1) Description of Bayes Optimal Design

The Bayes optimal design is an important branch of the study of experimental design. The goal of design is to maximize the expected utility of certain test results. The utility are usually provided as a metric which can be accurately measured (such as the Shannon information function). The final optimal design is decided by the selected utility function.

The basic idea of experimental design is to improve the accuracy of statistical inference by suitable selection of control variable values, which was proposed by Lindley [7] in 1972. The content of framework includes [8]: assuming that the experimental system has unknown parameter x in the parameter space X, a design □ needs to be selected in set H before the test is carried out in this system, in the course of the experiment sample y will be observed from sample space Y, a final decision d can be obtained based on the data y in set D .The function $U(d, \boldsymbol{x}, \eta, \boldsymbol{y})$ is the generic utility function which reflects experimental purpose.

The Bayes optimal design indicates that a good experimental design should ensure that the expected utility of optimum decision is maximized. For a given design η, the expected utility formula of optimum decision is as follows:

$$U(\eta) = \int_{y} \max_{d \in D} \int_{X} U(d, x, \eta, y) p(x|y, \eta) p(y|\eta)\, dx\, dy \quad (3)$$

Where $p(\cdot)$ indicates probability density function.

Bayes Optimal Design's maximum design η^* under utility function $U(\eta)$ is:

$$U(\eta^*) = \max_{\eta \in H} \int_{y} max_{d \in D} \int_{X} U(d, x, \eta, y) p(x|y, \eta) p(y|\eta)\, dx\, dy \quad (4)$$

2) Design criterion

By choosing different utility functions, we can get different Bayes optimal designs. Through the previous experiments [4], it can be seen that Bayes A-optimal design performance is generally equivalent to or even better than Bayes D-optimal design in network performance inference. So we select Bayes A - optimal design as the method of probing paths.

For referring parameter x , squared error $||x - x_e||_2^2 = (x - x_e)^T (x - x_e)$ is generally used to assess the accuracy of the valuation x_e, the design η should be selected so expected utility is maximized:

$$U_A(\eta) = -\int (x - \hat{x})^T (x - \hat{x}) p(y, x|\eta)\, dx\, dy \quad (5)$$

Where, $\hat{x}$ is the estimation value corresponding to the best decision scheme d.

Assume that the system is a normal linear system $y_S|x, \sigma^2 \sim A_S x + N(0, \sigma^2 I)$, where σ^2 is square difference of a zero mean noise, I is the unit matrix. The priori information is as follows: $x|\sigma^2$ is randomly selected from a multivariate normal distribution, which the mean vector of this normal distribution is μ, variance matrix is $\Sigma = \sigma^2 R^{-1}$, where μ and matrix R are known information.

Let $D(\eta) = (A_S^T A_S + R)^{-1}$, and from Bayes optimal design formula we can get:

$$U_A(\eta) = -\sigma^2 \mathrm{tr}\{D(\eta)\} \quad (6)$$

Where tr{D(η)} (trace) is defined as the sum of the diagonal elements. The selection of expected design η shall maximize utility $U_A(\eta)$, that is to minimize the following function:

$$\phi_A(\eta) = \mathrm{tr}\{D(\eta)\} \quad (7)$$

The above equation is commonly known as Bayes A-optimal design criteria. We can get the design that maximizes the probing information. The Bayes A-optimal design criterion is a function of $D(\eta) = (A_S^T A_S + R)^{-1}$, with prior information $\boldsymbol{R}$, which is difficult to measure accurately. Therefore, we assume $R = \epsilon I$, where ϵ is a small constant and matrix I is the unit matrix. Through experiments we noticed that this value choose for R always results in a good experimental design. Since the optimal design is not sensitive to the value of ϵ, we set $\epsilon = 0.001$ in the following part.

3) Selection Algorithm

After the selection of design criteria, the next step is to find the optimal design η^* that minimizes design criteria.

To select s rows from routing matrix A making the given design criteria minimum is an NP-complete problem. So the accurate calculation of the optimal design is not feasible. Through sequence searching algorithm, we can get a better solution. The basic idea is: initialize the set of selected path to empty set. Each selection process traverses each row of routing matrix A. Calculate the design criterion $\phi_A(\eta)$ after adding probing path set to the design criterions. Select the smallest the path set which makes the $\phi_A(\eta)$ minimum as a new probe path. Put this path into the probing path set, and so forth until sufficient number of detection paths are selected.

But there is still a big problem about the above solution - the amount of calculation is too large and the computation time is too long. Therefore interactive active probing method is carried out to improve the greedy algorithm. This method maximizes the use of the probe results of each time. The information that already obtained can be used to narrow the size of the routing matrix to reduce the computing time, while avoiding duplicate information. The main idea is to use sequence searching algorithm to select a probe path, analyzing of the result of the detection. If the detection results are normal (if there is no loss), it shows there is no fault of the path. So the links belong to this path can be determined as normal state. And the corresponding network performance can be identified. So we can exclude links belong to the path from the routing matrix. Then we repeatedly do the same to the routing matrix.

Table.1 shows the detail of path selection algorithm. The proportion of abnormal links in the actual network is relatively small and most of the links are in a good state. So the method described above can effectively reduce the calculation time and optimize of the process of selecting probing path in the actual large-scale network.

TABLE I. Probing Path Selection Algorithm

```
S={ };   //Initializing the set of probing path S
Y={};//Initializing the vector of probing path performance
for i = 1:s  //there are totally s probing paths
    for j∈{1,2,3,…,p}-S
        calculate the design criterion $\phi_A(S\cup\{j\})$;
    end
    select path a which minimizes the design criterion;
    probe on path a, and get the loss rate y of a;
    if (y == 0)  //there is no loss on this path
    delete columns corresponding to path a's links in A;
    S=S∪{a} ;  //update the probing set
    Y[i]=y;        //record the probing path performance
end
return S;
```

B. *Performance Inference*

Since our method selected only a small amount of the paths in the probing phase, the number of observation parameter y is far less than the number of unknown variable x. The linear inverse problem of (2) is usually underdetermined.

The performances of most links are in good state, and loss rate is close to 0, $\log\{1-\text{loss rate}\}$ is close to 0, too. So the vector x is sparse. The ideal solution is to make nonzero elements in the vector x minimum, which is equivalent to minimize the following L_0 norm $||x||_0$:

$$\text{minimize } ||x||_0 \quad \text{subject to } y = Ax \tag{8}$$

However, L_0 norm is non-convex. L_1 norm is used to estimate L_0 norm, shown in [6]. The sparse solution of L_1 norm minimization and uncertain linear equations is usually the same.

L_1 norm minimization method is to replace L_0 norm in (8) with L_1 norm $||x||_1$, so we get the following problem:

$$\text{minimize } \left||y-Ax\right||_1 + \lambda||x||_1 \tag{9}$$

$\lambda \in [0,1]$ determines satisfaction degree of limiting conditions $y = Ax$. The calculation result is not very sensitive to value λ[10], therefore we choose λ = 0.001.

Problem (9) can be derived as an equivalent linear programming problem, as follows:

$$\begin{aligned} \text{minimize} \quad & \lambda\textstyle\sum_i u_i + \sum_j v_j \\ \text{subject to} \quad & y = Ax+z \\ & u \geq x,\ u \geq -x \\ & v \geq z,\ v \geq -z \end{aligned} \tag{10}$$

This linear programming problem can be solved efficiently by mathematical programming tools such as Matlab toolbox.

IV. Experiment Analysis

The proposed optimization is based on NetQuest[4]. By dynamic detection, the calculation scale has been substantially reduced. The expected effect is to largely reduce the calculation time with same or better accuracy.

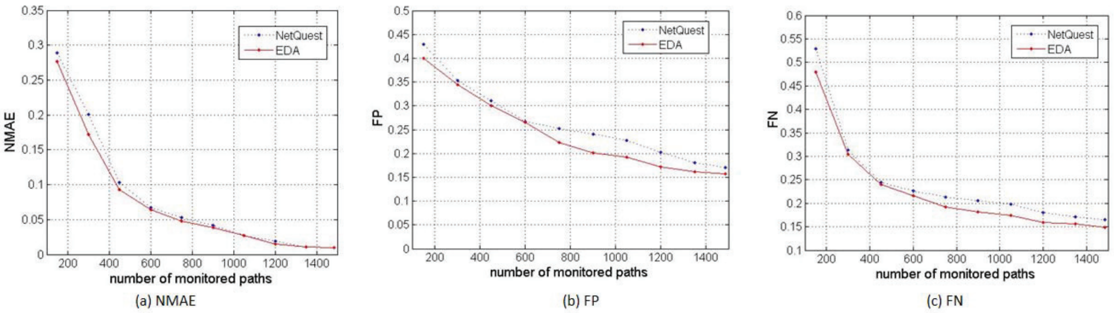
The experiment divided into two parts for analysis and comparison: First, the algorithm is implemented on different networks, and compared with NetQuest results. This is to test of the accuracy of the method proposed in this paper; Secondly, based on checked accuracy, count algorithm's running time, and compare with NetQuest's running time to verify the superiority of algorithm's time. We name our algorithm as "EDA" in short of experimental design algorithm.

A. *Experiment configuration*

1) *Experiment environment*

A large-scale simulation networks and a smaller PlanetLab real network are used as the experiment environment. Loss rate is set to any value between 0-1% in normal links and 5-10% in congested links.

The detail process of loss is determined by the loss model. This paper uses by Gilbert loss model to discard the data packet of each link. Using methods in [11], this paper takes path congestion probability as 35%. And the other state transition probability is depend on the default link loss rate, and thereby determines the average loss rate.



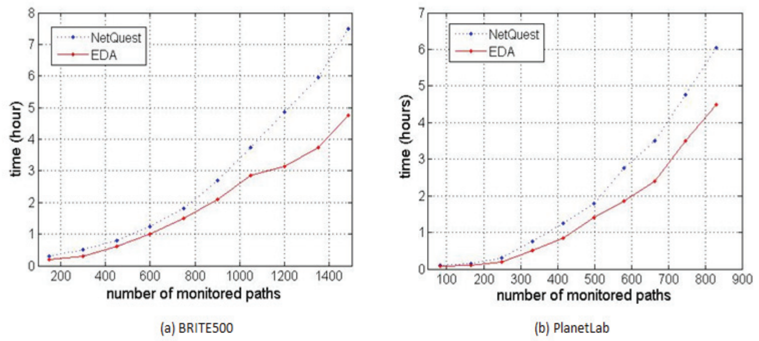
(a) NMAE

(b) FP

(c) FN

Fig. 1. Accuracy Comparison of different metrics on BRITE network

2) *Metrics*



(a) BRITE500

(b) PlanetLab

Fig. 3. Time Consumption Comparison on different networks

For accuracy assessment, we use three indicators to comprehensively evaluate.

Normalized mean absolute error (NMAE): this indicator reflects the error of performance inference.

$$\mathrm{NMAE} = \frac{\sum_i |\mathrm{inferred}_i - \mathrm{linkloss}_i|}{\sum_i \mathrm{linkloss}_i} \qquad (12)$$

where $\mathrm{inferred}_i$ indicates inferred value of the loss rate of link i, $\mathrm{linkloss}_i$ represents the actual value of the loss rate of the link i.

False positives (FP) and false negatives (FN): This group of indicators reflects links' recognition rate. The false positive rate shows the ratio of links being incorrectly inferred as congested, while false negative rate FN shows the ratio of links incorrectly inferred as normal.

$$\mathrm{FP} = p(\text{inferred to be congested} \mid \text{actually normal}$$
(13)

$$\mathrm{FN} = p(\text{inferred to be normal} \mid \text{actually congested}$$
(14)

The actual loss rate of links cannot be accurately accessed in PlanetLab platform. So we reference [9] using cross-validation methods. We take selected links as reference set, and unselected links as the validation set. Through inferring of probing data, we can get link loss rate x. And put x into the validation set for validation. If the difference between inferring loss rate and actual probing loss rate is within the tolerable error rate, this tells that inferring of all links on the path is correct.

For the assessment of the time-consuming, we mainly use CPU time consumed by the algorithm as an index for judging.

B. *Accuracy Comparison*

Fig. 1 shows the result on BRITE and Fig.2 shows the results on PlanetLab.

From the results, we can see that our method is equal or better than NetQuest algorithm in average absolute error, FP and FN in the same amount of probing paths. This means these two algorithms own the same accuracy. In large-scale network, the loss rate of most of links is approximately zero, so accuracy is increased to a lesser extent, which is not obvious.

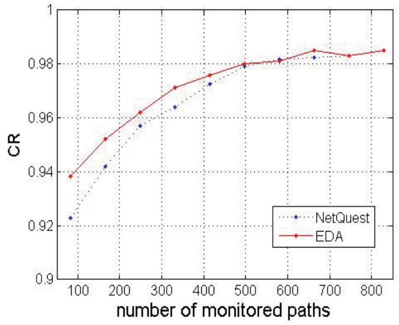


Fig. 2. Accuracy Comparison on PlanetLab network

C. *Time Consumption Comparison*

We got 10-group data of running time shown in Fig. 3. From the figure we can see that our method's running time is less than NetQuest's under the same conditions. And this time superiority is significant.

V. CONCLUSION

The balance between the overhead and accuracy of detection is always an important issue in the measurement of large-scale network. In this paper, the design of large-scale network loss rate inference method obtains a higher accuracy with a smaller probe overhead. Our method improves the time efficiency while shortens the computing time with lower cost and higher accuracy.

REFERENCES

[1] Y. Chen, D. Bindel, H. Song, and R. H. Katz, "An algebraic approach to practical and scalable overlay network monitoring," in Proc. ACM SIGCOMM, 2004, pp. 55–66.

[2] D.B. Chua, E.D. Kolaczyk, and M. Crovella, "Efficient monitoring of end-to-end network properties," in Proc. IEEE INFOCOM 2005, pp. 1701- 1711.

[3] D. B. Chua, E. D. Kolaczyk, and M. Crovella, "A statistical framework for efficient monitoring of end-to-end network properties," in Proc. ACM SIGMETRICS, 2005, pp. 390–391.

[4] H.H. Song, L. Qiu, and Y. Zhang, "Netquest: A flexible framework for large scale network measurement," IEEE/ACM Transactions on Networking, vol.17, no.1, pp.106-119, Feb. 2009

[5] H. X. Nguyen, P Thiran, "Internet Measurement Conference Network Loss Inference with Second Order Statistics of End-to-End Flows," in Proc. ACM SIGCOMM, 2007, pp. 227-240.

[6] D. L. Donoho. "For most large underdetermined systems of equations, the minimal L1-norm solution is also the sparsest solution," Sept. 2004. [Online]. Available: http://www-stat.stanford.edu/˜donoho/Reports/.

[7] D. V. Lindley, Bayesian Statistics—A Review. Philadelphia, PA: SIAM, 1972.

[8] K. Chaloner and I. Verdinelli, "Bayesian experimental design: A review," Statistical Science, 1995, vol. 10, pp. 273–304.

[9] V. N. Padmanabhan, L. Qiu, and H. J. Wang, "Server-based inference of internet link lossiness," in Proc. IEEE INFOCOM, Mar.2003, pp. 145-155.

[10] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan, "Network anomography," in Proc. Internet Measurement Conf., Oct. 2005, pp. 317-330

[11] V. N. Padmanabhan, L. Qiu, and H. Wang, "Server-based inference of Internet link lossiness," in Proc. IEEE INFOCOM, Mar. 2003, pp. 145-155