

基于非线性规划的链路丢包率推理算法

顾 然* 邱雪松 乔 焰 李 娟 孟洛明

(北京邮电大学网络与交换技术国家重点实验室 北京 100876)

摘 要: 针对目前大多数链路丢包率推理算法不能兼顾精度和速度的问题, 该文提出一种新的覆盖网链路丢包率推理算法, 算法通过将链路丢包率推理问题转换为求解非线性规划的最优化问题, 从而较准确地计算出所有链路的丢包率。该算法仅需对每条端到端路径进行一次测量, 且没有额外部署成本。通过软件模拟和 Internet 实验对算法性能进行了验证, 实验结果证明了算法的可行性及优越性。

关键词: 网络测量; 网络层析成像; 单播; 链路丢包率; 非线性规划

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2012)06-1425-07

DOI: 10.3724/SP.J.1146.2011.01052

Link Loss Inference Algorithm with Nonlinear Programming

Gu Ran Qiu Xue-song Qiao Yan Li Juan Meng Luo-ming

(State Key Laboratory of Networking and Switching Technology,

Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: To address the issue that most link loss rate inference algorithm can not take into account the accuracy and speed, a novel link loss rate inference algorithm of overlay network is proposed. This algorithm converts the link loss rate inference issue into solution of nonlinear programming issue in order to calculate the packet loss rate of all links more accurately and only needs to probe once for every path, with no extra deployment costs. Through software simulations and Internet experiments, the feasibility and superiority of this algorithm is proved.

Key words: Network measurement; Network tomography; Unicast; Link packet loss rate; Nonlinear programming

1 引言

近年来, 随着计算机网络规模日益扩大, 受设备多样性与网络复杂性影响, 网络中的不稳定因素也在增加; 与此同时, 网络流量急剧增长, 各类新型业务, 如视频会议、视频点播、P2P 等, 也以极快的速度吞噬着网络资源, 网络性能的度量变得越来越重要。目前, 网络测量是获取网络状态参数最主要的方法。按照测量系统所处的位置, 网络测量可分为基于路由器的测量、端到端的测量以及路由器协助的测量。当前广泛应用的是路由器测量方法, 即在网络节点或者网络节点之间直接主动或被动测量网络性能, 但该方法存在着很多潜在的局限, 如成本、安全、性能、可行性等。以上这些原因使得我们需要引进新的测量方法, 能够在没有路由节点的配合下, 通过端到端的测量来诊断网络性能问题。

网络层析成像(network tomography)^[1], 是将医学上的计算机断层成像(Computerized Tomography, CT)思想引入到网络测量中, 在网络边界上进行端到端的测量, 根据网络边界的测量信息来分析和推断网络的内部性能, 是一种在没有网络节点协作条件下, 通过主动发送探测或被动收集网络内部有用信息的新技术, 结合统计学方法能够较准确地推理出网络所有链路的参数。

利用网络层析成像技术推理链路丢包率需要求解系统的线性方程组, 而此方程组在大多数情况下是没有唯一解的。为了解决这个问题, 需要获取额外信息或做出假设, 但是目前已有的方法都存在缺陷, 例如: 文献[2]提出的 LIA(Loss Inference Algorithm)算法对每条路径进行多次测量, 计算丢包率的方差, 进而唯一确定每条链路的丢包率。误差较小, 但是对网络造成负载较大, 所需的探测时间长, 运算复杂度高。文献[3]是 LIA 算法的改进, 准确度有所提高, 但是缺点和 LIA 算法相同。文献[4-6]对拥塞链路的分布做出假设, 并通过 RPI(Reference Path based lossy link Identification algorithm), SCFS(Smallest Consistent Failure Set)

2011-10-13 收到, 2012-04-06 改回

国家自然科学基金国家创新研究群体科学基金(61121061), 国家自然科学基金(60973108), 国家 863 计划项目(2011AA05A116)和国家 973 计划项目(2007CB310703)资助课题

*通信作者: 顾然 guran@bupt.edu.cn

或PBLI(Probability-based Bad Links Inference)算法求解先验概率,定位拥塞链路,速度很快。但是这3种算法无法确定丢包率大小。文献[7]通过引入最小可识别链路序列(MILS),可以唯一确定每个MILS(可能包含一个或多个链路)的丢包率。但是它不能唯一确定每条链路的丢包率。文献[8]使用多播探测包,文献[9]使用被动探测,由于绝大多数路由器不支持多播且无法采集数据,因而实用价值有限。

为了解决现有算法探测困难、运算复杂度高、应用受限的缺陷,本文提出了一种新的链路丢包率推理算法。算法首先将丢包率测量问题建立成代数模型,并证明部分链路的丢包率是可以唯一确定的,将已确定链路从方程中移除,然后将求解线性方程组的问题转换为最优化问题,从而可以以较小代价得到较准确的结果。本文通过仿真实验和Internet实验与LIA算法做了对比,新算法的推理准确度比LIA算法提高了5%,并且推理时间至少缩短了40%。

2 代数模型

本小节以覆盖网中的链路丢包率推理问题为例,将丢包率推理问题建立成代数模型^[2]。

首先将该网络建立一个有向图模型 $G(V, E)$, V 为有向图中的节点集合,它代表网络中的路由器和主机, $E = \{e_1, e_2, \dots, e_{n_e}\}$ 为有向图中连接两个节点的有向边集合,代表网络中的链路集合,链路个数为 $n_e = |E|$ 。定义 $P = \{p_1, p_2, \dots, p_{n_p}\}$ 为网络中所有路径的集合,路径个数为 $n_p = |P|$ 。若路径 p_i 包含链路 e_k ,则 $D_{i,k} = 1$,否则 $D_{i,k} = 0$,这样可构建路由矩阵 D 。例如,图1所示的网络中, $n_p = 6$, $n_e = 6$,则路由矩阵可表示为

$$D = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \quad (1)$$

用 ϕ_i 表示路径 p_i 的通过率(通过率=1-丢包率),

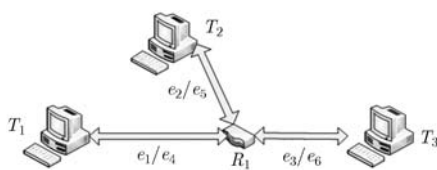


图1 网络示例

$\varphi_{i,k}$ 表示在路径 p_i 中,链路 e_k 的通过率。路径通过率与链路通过率的关系可表示为

$$\phi_i = \prod_{e_k \in p_i} \varphi_{i,k} \xrightarrow{\text{两边取对数}} \ln \phi_i = \sum_{k=1}^{n_e} D_{i,k} \ln \varphi_{i,k} \quad (2)$$

从源节点向目的节点发送 S 个单播数据包进行探测,获得被探测路径的拓扑和通过率。

定义 $\hat{\phi}_i$ 为路径 p_i 的 S 个探测包中正确到达目标的百分比; $\hat{\varphi}_{i,k}$ 为路径 p_i 中需要通过链路 e_k 的探测包中正确通过链路 e_k 的百分比; $\hat{\varphi}_k$ 为所有需要通过链路 e_k 的探测包中正确通过链路 e_k 的百分比。则路径 p_i 的通过率为 $\phi_i = E(\hat{\phi}_i)$,链路 e_k 的通过率为 $\varphi_{i,k} = E(\hat{\varphi}_{i,k})$ 。

对于端到端通过率与链路通过率的关系,本文做如下两个基本假设:

假设1 $\hat{\varphi}_{i,k} = \hat{\varphi}_k$,即一条路径经过一个链路的通过率近似等于该链路的通过率。

假设2 链路通过率 $\hat{\varphi}_k$ 是相互独立的。

这两个假设是任何端到端测量技术的基础假设^[2,3]。文献[2]指出,这两个假设建立了端到端通过率与链路通过率之间的关系,并给出了这两个假设在大多数情况下成立的证明。

令 $\mathbf{x} = [\ln \hat{\varphi}_1 \ \ln \hat{\varphi}_2 \ \dots \ \ln \hat{\varphi}_{n_e}]^T$, $\mathbf{y} = [\ln \hat{\phi}_1 \ \ln \hat{\phi}_2 \ \dots \ \ln \hat{\phi}_{n_p}]^T$,在假设1,假设2的基础上,式(2)可写为

$$\mathbf{y} = \mathbf{D}\mathbf{x} \quad (3)$$

一般情况下,路径个数 n_p 远比链路个数 n_e 大,但是矩阵 D 仍不满秩,即 $r_D = \text{rank}(D) < n_e$ 。因而在没有获得额外信息的情况下此线性方程没有唯一解。为了解决这个问题,下一小节中,本文将提出一种高效的丢包率推理算法,即使矩阵 D 不是满秩的,我们也可以得到一组较准确的链路丢包率值。

3 高效的链路丢包率推理算法

本文提出的链路丢包率推理算法主要分为3步:先将线性方程化简以减小式(3)的运算量;再求解线性方程组以求出部分链路丢包率值;最后使用非线性规划模型推算剩余的未知量。

3.1 化简线性方程

通常情况下路由矩阵 D 的阶数很大,如果直接求解线性方程将消耗大量时间,因此需要先对其化简,以减少求解线性方程时的运算量,节省计算时间。

在路由矩阵 D 中,如果矩阵的第 k_1 列与矩阵的第 k_2 列相等,就表示所有包含链路 e_{k_1} 的路径,也都包含链路 e_{k_2} ,链路 e_{k_1} 与 e_{k_2} 的状态无法唯一确定。这种情况下,本文将链路 e_{k_1} 与 e_{k_2} 视为一条虚拟链路

(virtual link)。

由于路径通过率等于链路通过率之积，因此可以推断出，如果路径不拥塞，则这条路径通过的所有链路都不拥塞。本文定义路径拥塞判定阈值为 ϕ_i ，当 $\hat{\phi}_i \geq \phi_i$ 时，认定 p_i 不拥塞，并且 p_i 经过的所有链路均不拥塞。再将这些不拥塞的路径和链路从方程中去掉，可以进一步简化原方程组。

3.2 求解线性方程

研究发现，式(3)虽然没有唯一解，但是有一部分未知量是可以唯一确定的。本小节提出定理 1，将找出方程组中所有可唯一确定的未知量，并将它们移除，从而将待求问题化简。

待求方程为非齐次线性方程组，其解可以表示为非齐次线性方程组的导出组 $\mathbf{0} = \mathbf{D}\mathbf{x}$ 的基础解系 $\mathbf{Z}\mathbf{r}$ 与非齐次线性方程组的一个特解 \mathbf{x}_0 之和^[10]。令 $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_{n_e}]^T$ 为 $\mathbf{y} = \mathbf{D}\mathbf{x}$ 的通解，其表达式为

$$\mathbf{x} = \mathbf{Z}\mathbf{r} + \mathbf{x}_0 = \begin{bmatrix} Z_{1,1} & Z_{1,2} & \dots & Z_{1,n_e-r_D} \\ Z_{2,1} & Z_{2,2} & \dots & Z_{2,n_e-r_D} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{n_e,1} & Z_{n_e,2} & \dots & Z_{n_e,n_e-r_D} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_{n_e-r_D} \end{bmatrix} + \begin{bmatrix} x_{0_1} \\ x_{0_2} \\ \vdots \\ x_{0_{n_e}} \end{bmatrix} \quad (4)$$

其中 \mathbf{Z} 为矩阵 \mathbf{D} 的零空间中的一组基， \mathbf{r} 为任意实数向量， \mathbf{x}_0 为非齐次线性方程 $\mathbf{y} = \mathbf{D}\mathbf{x}$ 的一组特解， n_e 是待求链路数量， r_D 由矩阵 \mathbf{D} 的秩。在一般情况下，由于测量中不可避免的样本误差，式(3)不存在精确特解，因此本文采用最小范数解^[11]作为方程组特解。

定理 1 若对于所有 $1 \leq i \leq n_e - r_D$ ， $Z_{k,i} = 0$ ，则未知量 x_k 可唯一确定。

证明 由式(4)得出， $x_k = Z_{k,1}r_1 + Z_{k,2}r_2 + \dots + Z_{k,n_e-r_D}r_{n_e-r_D} + x_{0_k}$ 。因此，若对于所有 $1 \leq i \leq n_e - r_D$ ， $Z_{k,i} = 0$ ，则 $x_k = x_{0_k}$ 。显然， x_k 不包含 \mathbf{r} 中变量，可唯一确定。证毕

根据定理 1，可以将 \mathbf{x} 分解为 $\mathbf{x}^A = \mathbf{x}_0^A$ 与 $\mathbf{x}^B = \mathbf{Z}^B\mathbf{r} + \mathbf{x}_0^B$ ， \mathbf{x}^A 为方程中可以唯一确定的未知量， \mathbf{x}^B 无法唯一确定，需要进一步的推理。由于非线性规划需要在 n 维空间内搜索最优解，如果未知量 \mathbf{r} 中的元素个数较多，那么推理速度就可能很慢。通常来说，在对 \mathbf{x}^B 做行变换后，矩阵 \mathbf{Z}^B 的非 0 元素可以分成多块，因此我们再将 \mathbf{x}^B 分解为 $\mathbf{x}^{B^n} = \mathbf{Z}^{B^n}\mathbf{r}^n + \mathbf{x}_0^{B^n}$ ， \mathbf{Z}^{B^n} 为矩阵 \mathbf{Z}^B 中的一个子矩阵，共 n_n 个。这样一来，每一个 \mathbf{x}^{B^n} 中的元素就大幅减少，

可以节约运算时间。例如当对 \mathbf{x} 进行行变换后

$$\mathbf{Z} = \left[\begin{array}{cc|cc} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ \hline 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 \end{array} \right], \mathbf{x}_0 = \begin{bmatrix} -1 \\ -0.8 \\ -2 \\ -3 \\ -5 \\ -4 \\ -0.5 \\ -0.2 \end{bmatrix} \quad (5)$$

则将其拆分为

$$\mathbf{x}^A = \mathbf{x}_0^A = [-0.2] \quad \left. \begin{aligned} \mathbf{x}^{B^1} &= \mathbf{Z}^{B^1}\mathbf{r}^1 + \mathbf{x}_0^{B^1} = \begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} + \begin{bmatrix} -1 \\ -0.8 \\ -2 \end{bmatrix} \\ \mathbf{x}^{B^2} &= \mathbf{Z}^{B^2}\mathbf{r}^2 + \mathbf{x}_0^{B^2} = \begin{bmatrix} -1 & -1 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_3 \\ r_4 \\ r_5 \end{bmatrix} + \begin{bmatrix} -3 \\ -5 \\ -4 \\ -0.5 \end{bmatrix} \end{aligned} \right\} \quad (6)$$

3.3 使用非线性规划推算未知量

由式(6)，此时 \mathbf{x}^A 中的元素都已经唯一确定，但是每一个 \mathbf{x}^{B^n} 中都包含 \mathbf{r}^n ，因而无法唯一确定，只能确定链路通过率间的线性关系。本小节利用已求出 \mathbf{x}^A 的丢包率分布，将求解 \mathbf{x}^{B^n} 的问题转换为非线性规划问题，从而较准确地得到剩余链路丢包率。

实验发现， \mathbf{x}^A 中已包含了大部分的未知量，因而可以从 \mathbf{x}^A 中求出所对应链路的通过率 $\hat{\varphi}_k^A$ ，并推算系统中所有链路的平均通过率 $\bar{\varphi}$ 。文献[12]证明网络中链路通过率服从高斯分布，因此对每个 n 应当找到一组 \mathbf{r}^{*n} 使得

$$\mathbf{r}^{*n} = \arg \min \sum (\hat{\varphi}_k^{B^n} - \bar{\varphi})^2 \quad (7)$$

其中 $0 < \hat{\varphi}_k^{B^n} \leq 1$ ， $\hat{\varphi}_k^{B^n} = e^{\sum_l Z_{k,l}^{B^n} r_l^n + x_{0_k}^{B^n}}$ 为 \mathbf{x}^{B^n} 中链路的通过率， r_l^n 为未知量。则问题可以化为非线性规划问题：

$$\begin{aligned} \min f(\mathbf{r}^{*n}), f(\mathbf{r}^{*n}) &= f(r_1, r_2, \dots) \\ &= \sum \left(e^{\sum_l Z_{k,l}^{B^n} r_l^n + x_{0_k}^{B^n}} - \bar{\varphi} \right)^2 \end{aligned} \quad (8)$$

$$\text{s.t.} \quad \sum_l Z_{k,l}^{B^n} r_l^n + x_{0_k}^{B^n} \leq 0 \quad (9)$$

式(8)为目标函数，式(9)为所求未知量需满足的约束条件。由于每一组 \mathbf{r}^{*n} 包含的元素不同，从而有 $\sum_n \min f(\mathbf{r}^{*n}) = \min \sum_n f(\mathbf{r}^{*n}) = \min f(\mathbf{r}^*)$ ，因此

对矩阵 \mathbf{Z}^B 的拆分不会对结果造成影响。求解该最优化问题就能够求出 \mathbf{r}^* , 从而唯一确定每一个 \mathbf{x}^{B^n} 。至此, 我们就可以计算出所有链路的通过率。

3.4 算法伪代码

算法伪代码如表 1 所示。

表 1 算法伪代码

输入:	路由矩阵 \mathbf{D} , 路径通过率 $\hat{\phi}$
步骤 1 化简线性方程	
(1)	$\mathbf{D} = \text{unique}(\mathbf{D}, 1)$; //删除重复列与全 0 列
(2)	$\mathbf{y} = \ln \hat{\phi}$; $[\mathbf{D}, \mathbf{y}] = \text{deluncongested}(\mathbf{D}, \mathbf{y})$; //删除不拥塞的路径和链路
步骤 2 求解线性方程	
(1)	$\mathbf{Z} = \text{null}(\mathbf{D})$; //计算零空间
(2)	$\mathbf{x}_0 = \text{solve}(\mathbf{y}, \mathbf{D})$; //求出一组特解
(3)	$[\mathbf{0}, \mathbf{x}^A, \mathbf{Z}^{B^1}, \mathbf{x}_0^{B^1}, \dots, \mathbf{Z}^{B^{n_e}}, \mathbf{x}_0^{B^{n_e}}] = \text{split}(\mathbf{Z}, \mathbf{x}_0)$; //以 \mathbf{Z} 为基准按行拆分 \mathbf{Z}, \mathbf{x}_0
步骤 3 使用非线性规划推算未知量	
(1)	$\tilde{\varphi}^A = e^{\mathbf{x}^A}$; $\bar{\varphi} = \text{mean}(\tilde{\varphi}^A)$; //求均值
(2)	for($n=1; n_e$) {
(3)	$\mathbf{r}^* = \text{nlp} \left(f(r_1, r_2, \dots) = \sum \left(e^{r_i} - \bar{\varphi} \right)^2, \right.$ $\left. \sum_i \mathbf{Z}_{k,i}^{B^n} r_i^n + x_{0_k}^{B^n} \leq 0 \right)$; //求解 \mathbf{r}^*
(4)	$\mathbf{x}^{B^n} = \mathbf{Z}^{B^n} \mathbf{r}^* + \mathbf{x}_0^{B^n}$; $\tilde{\varphi}^{B^n} = e^{\mathbf{x}^{B^n}}$; }
(5)	$\tilde{\varphi} = \text{combine}(\tilde{\varphi}^A, \tilde{\varphi}^{B^1}, \dots, \tilde{\varphi}^{B^{n_e}})$; //将 $\tilde{\varphi}^A$, $\tilde{\varphi}^{B^n}$ 合并为 $\tilde{\varphi}$
输出:	链路通过率 $\tilde{\varphi}$

本算法最耗时的部分为求解线性方程的最小范数解, 根据文献[2], 其运算复杂度为 $O(n_p \times n_e^2)$, 相比 LIA 算法的 $O(n_p^2 \times n_e^2)$ 可知, 本算法在探测规模较大时有着明显的运算时间优势, 实验结果也证明了这一点。

4 仿真与 Internet 实验

本文使用模拟网络环境和 PlanetLab 实验验证本算法 (Non-Linear Programming Algorithm, NLPA) 的正确性, 并且与文献[2]提出的 LIA 算法、文献[5]提出的 SCFS 算法进行比较。

4.1 搭建仿真实验环境

网络拓扑 近年的研究表明, 大多数网络服从幂率分布^[13], 本文利用 BRITE 和 INET 拓扑仿真器生成 3 种类型的网络拓扑, 网络规模为 500~1000 个节点。

丢包模型 不同丢包模型对实验结果的影响很小^[2], 因此本实验中仅采用 LLRD1 模型^[14]与伯努利丢包模型。

探测 本算法及 SCFS 算法只需一次探测。LIA 算法需要多次探测以计算链路通过率的方差, 根据文献[2]的实验结果, 本文在对比时采用 $m = 50$ 次探测, LIA 算法的运算结果。我们令拓扑中度数最小的节点作为主机, 互相发送探测包。

拥塞链路 链路拥塞概率 p_{cr} , 在实验中可变。网络中的链路以 p_{cr} 为概率被选定为拥塞链路, 其余链路为不拥塞链路, 以 LLRD1 模型设置丢包率, 并以伯努利模型丢弃通过链路的数据包。每条路径发送探测包数量 $S = 1000$, 拥塞判定阈值 $\phi_t = 0.998$ 。

实验步骤

(1) 使用软件模拟探测, 获得路由矩阵 \mathbf{D} , 路径通过率 $\hat{\phi}$, 链路通过率 $\hat{\varphi}$ 。

(2) 执行算法: 输入路由矩阵 \mathbf{D} , 路径通过率 $\hat{\phi}$ (NLPA, SCFS 算法 1 组, LIA 算法 50 组)。输出链路通过率 $\tilde{\varphi}$ 。

(3) 将 $\tilde{\varphi}$ 与 $\hat{\varphi}$ 进行对比, 计算误差, 输出结果。

4.2 仿真实验结果

4.2.1 准确度评估准则 本文采用以下 3 个参数来衡量算法的准确度:

检测率 (Detection Rate, DR): 所有拥塞链路中, 被正确诊断为拥塞链路的百分比。误检率 (False Positive Rate, FPR): 所有求解出的拥塞链路中, 诊断错误的百分比。这两个参数用于评估算法的链路识别率。

$$DR = \frac{|F \cap X|}{|F|}, \quad FPR = \frac{|X \setminus F|}{|X|} \quad (10)$$

式(10)中 F 表示实际拥塞链路集合, X 表示算法推理的拥塞链路集合。根据 LLRD1 丢包模型, 我们设定链路拥塞判定阈值 $\varphi_t = 0.998$ 。

均方根差 (Error Root Mean Square, ERMS): 链路通过率误差的均方根, 这个参数用于评估算法的误差大小。因为 SCFS 算法不能推理出丢包率, 所以只有 NLPA 算法和 LIA 算法能够计算误差的均方根。

$$ERMS = \sqrt{\frac{\sum (\tilde{\varphi}_k - \hat{\varphi}_k)^2}{n_e}} \quad (11)$$

4.2.2 实验结果分析 表 2 比较了不同类型拓扑的运算结果, 链路拥塞概率 0.1, 网络节点个数固定在 1000 个。由于 BRITE 可调节参数较多, 而文献[2]

表 2 不同类型拓扑的运算结果

算法	Inet			Brite-BA2			Brite-Waxman		
	NLPA	LIA	SCFS	NLPA	LIA	SCFS	NLPA	LIA	SCFS
DR	0.9863	0.9317	0.9625	0.9805	0.5876	0.6693	0.9904	0.8446	0.9514
FPR	0.0120	0.0960	0.0631	0.3039	0.5829	0.4130	0.0136	0.0850	0.0945
ERMS	0.0061	0.0203	-	0.0132	0.0616	-	0.0061	0.0190	-

中并没有说明实验所采用的具体参数，因此本文采用 BRITE 在默认参数下生成的拓扑，具体参数为： $\alpha = 0.15$ ， $\beta = 0.2$ ， $m=2$ ，Max BW=1024，Min BW=10， $p=0.25$ ， $q=0.5$ 。虽然 LIA 算法的运算结果与文献[2]有较大差异，但是算法是在相同条件下进行运算，因而实验结果仍然具有参考价值。从表 2 可看出，本算法在不同种类拓扑下能够确保运算精度，可适用于不同类型网络。LIA 和 SCFS 在 Inet 与 Brite-Waxman 中表现较好，而在 Brite-BA2 中性能较差。在 Brite-BA2 拓扑的实验里， x^A 包含链路比例为 40%，而在 Inet 和 Brite-Waxman 的实验中，这一比例高达 80%，因而大多数链路都能够唯一确定且误差较小。

图 2 比较了不同节点数量下的运算结果，链路拥塞概率 0.1，网络节点个数从 500 到 1000。从图 2(a), 2(b) 可看出，本方法链路识别率明显高于 LIA, SCFS 算法。由图 2(c)，本算法推理出的链路丢包率偏离实际值的幅度较小，因此比 LIA 算法更为精确。由图 2(d)，随着网络个数的增加，3 种算法的计算时间都有所增加，而 LIA 算法时间增加的最快。由于 SCFS 算法仅推理拥塞链路，因此计算时间也最短。本算法运算时间介于 LIA 与 SCFS 算法之间，并且随节点个数的增加呈近似线性增长。各算法均不随网络大小改变而对精度产生较大影响，但是 LIA, SCFS 算法在不同的网络拓扑下波动较大，不能提供稳定的准确度保障。

图 3 比较了不同链路拥塞概率下算法的运算结果，网络节点数量为 1000 个。从图 3 可看出，当链路拥塞概率逐渐增大时，3 种算法的性能均有所下降，但相比其余两个算法，新算法的下降幅度较小，且表现较为稳定。

综上所述，本算法能够在发送较少探测，对网络造成较小干扰的前提下，以较短的计算时间，提供比 LIA 算法更高的链路识别率以及更低的误差，整体性能更优。

4.3 PlanetLab 实验

本文采用 PlanetLab 平台作为真实网络的实验环境。相对于仿真实验环境，Internet 环境中影响

探测的因素较多，对实验结果产生负面影响。首先，出于安全原因，部分路由器不响应 ICMP 请求，这些路由无法看到 IP 地址。其次，在测量过程中，路由可能发生变化。最后，在较长时间的测量过程中，链路状态可能发生变化。为尽量减少这些不稳定因素对算法的影响，本实验于测量前后各运行一次 traceroute 命令，丢弃包含匿名路由与路由发生变化路径，并且仿照文献[2]的实验，采用文献[15]提供的方法合并多址路由。

受网络环境所限，本文添加了 120 个 Planetlab 终端节点作为叶子节点，相互发送探测包，共进行 3 次实验。由于节点状态可能发生改变，无法保证每次实验都能连接上相同的节点，因此在 3 次实验中有不超过 20% 的节点改变。通过移除不稳定路径及终端节点，我们最终得到 90 个较稳定的 Planetlab 节点作为实验节点。

首先使用 traceroute 工具获得网络拓扑，包含 3015 个节点，7995 条物理链路。为减小网络负载，每条路径发送 $S = 500$ 个单播 UDP 探测包，在目的节点计算路径通过率。限制探测包在每个节点的发送速率为 50 kB/s，文献[12]表明，100 kB/s 以下的速率不会在 PlanetLab 的网络环境中产生拥塞。

在 PlanetLab 平台中无法获得链路的真实丢包率，本文采用文献[14]中的交叉验证方法。SCFS 算法不能够推理出丢包率，因此不适用该方法，本文不再进行对比。将端到端探测路径随机分为等大小的两个集合，分别为推理集 P_{inf} 和验证集 P_{val} 。首先对推理集 P_{inf} 进行计算，求解出链路通过率 $\tilde{\varphi}$ ，再将求出的链路通过率代入验证集中计算。若 $\left| \hat{\varphi}_{\text{val}_i} - \prod_{e_k \in p_{\text{val}_i}} \tilde{\varphi}_k \right| \leq \varepsilon$ 成立，则路径 p_{val_i} 包含的所有链路都计算正确， ε 为可容忍误差，本文中取 $\varepsilon = 0.005$ 。

图 4 结果显示，采用本算法计算的链路丢包率，正确率 > 95%。由于 Internet 中大多数端到端路径为不拥塞路径，因此剩余链路较少，两算法性能也相差无几。受限于网络环境，本次实验得到的稳定 Planetlab 终端节点个数较少，我们会在以后的实验

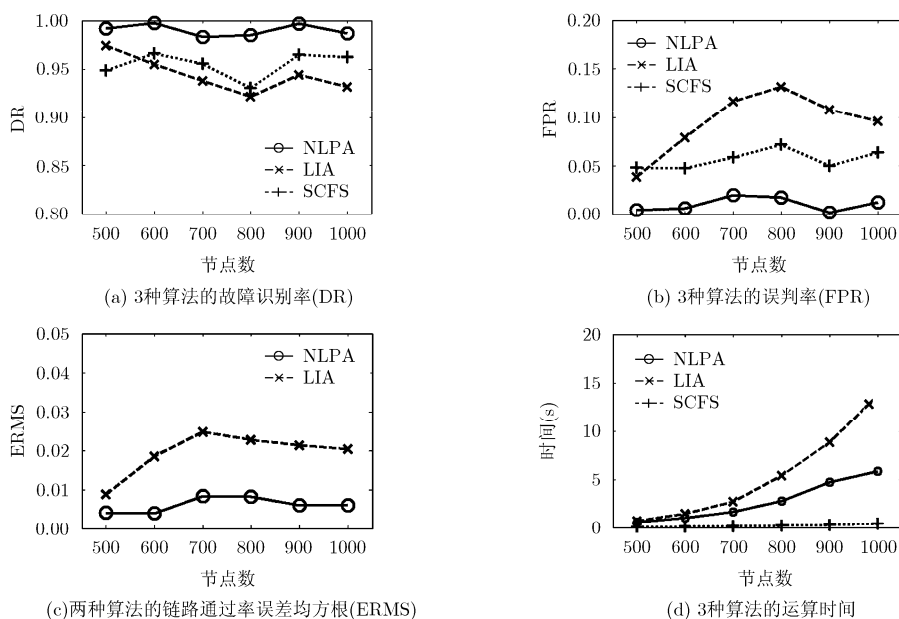


图2 500-1000节点Inet拓扑的运算结果

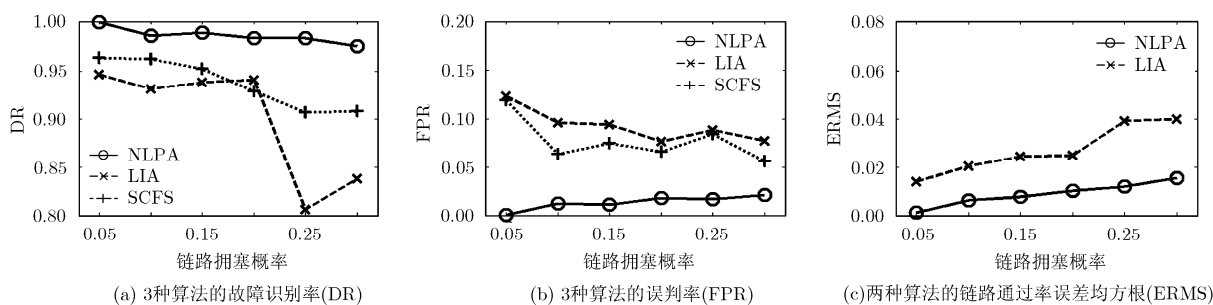


图3 1000节点Inet拓扑在不同拥塞概率下的运算结果

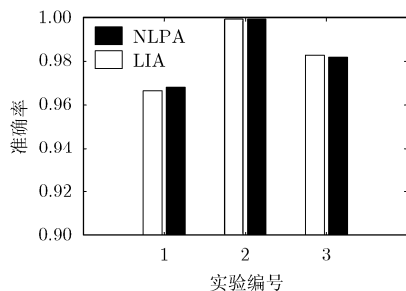


图4 PlanetLab实验结果

中使用更先进的实验工具及方法来获得较多的稳定节点进行实验,并证明本算法的优越性。

5 总结

网络层析成像是在网络边界上进行端到端的测量来分析和推断网络的内部性能,因此算法的好坏将直接影响测量精度。本文利用网络中丢包率的分布特性,通过将线性方程组的解转换为非线性方程在线性约束条件下的最优化问题,推理出所有链路

的丢包率。对比LIA算法,本文的方法能够以1/50的探测数量,更低的运算复杂度,获得更高的推理精度,并且与LIA算法类似,本算法也具有普适性。因此,本算法整体性能相比LIA算法有较大提升。

参考文献

- [1] 赵洪华, 陈鸣. 基于网络层析成像技术的拓扑推断 [J]. 软件学报, 2010, 21(1): 133-146.
Zhao Hong-hua and Chen Ming. Topology inference based on network tomography [J]. *Journal of Software*, 2010, 21(1): 133-146.
- [2] Nguyen H X and Thiran P. Network loss inference with second order statistics of end-to-end flows [C]. Proceedings of ACM SIGCOMM'07, Kyoto, Japan, Aug. 2007: 227-240.
- [3] Ghita D, Nguyen H, Kurant M, et al. Netscope: practical network loss tomography [C]. Proceedings of IEEE INFOCOM'10, San Diego, CA, USA, Mar. 2010: 1262-1270.
- [4] 张志勇, 胡光岷. 一种新的故障链路识别算法 RPI[J]. 电子与信息学报, 2011, 33(8): 1924-1929.

- Zhang Zhi-yong and Hu Guang-min. A novel algorithm for lossy link identification[J]. *Journal of Electronics & Information Technology*, 2011, 33(8): 1924-1929.
- [5] Duffield N. Network tomography of binary network performance characteristics[J]. *IEEE Transactions on Information Theory*, 2006, 52(12): 5373-5388.
- [6] 赵佐, 蔡皖东. 基于先验概率的性能降级链路推断方法 [J]. 计算机工程, 2010, 36(3): 13-18.
Zhao Zuo and Cai Wan-dong. Inference method for lossy links based on prior probability [J]. *Computer Engineering*, 2010, 36(3): 13-18.
- [7] Zhao Y, Chen Y, and Bindel D. Towards unbiased end-to-end network diagnosis[C]. Proceedings of ACM SIGCOMM'06, Pisa, Italy, Sept. 2006: 219-230.
- [8] 吴文佳, 张建中, 张元鹏. 基于丢包率的多播网络拓扑推断算法 [J]. 计算机工程, 2010, 36(1): 124-126.
Wu Wen-jia, Zhang Jian-zhong, and Zhang Yuan-peng. Algorithm of multicast network topology inference based on packet loss rate[J]. *Computer Engineering*, 2010, 36(1): 124-126.
- [9] Mahajan R, Spring N, Wetherall D, *et al.* User-level internet path diagnosis [C]. Proceedings of SOSP03, Bolton Landing, New York, USA, Oct. 2003: 106-119.
- [10] Farhat C and Geradin M. On the general solution by a direct method of a large-scale singular system of linear equations: application to the analysis of floating structures[J]. *International Journal for Numerical Methods in Engineering*, 1998, 41(4): 675-696.
- [11] Golub G H and Reinsch C. Singular value decomposition and least squares solutions [J]. *Numerische Mathematik*, 1970, 14(5): 403-420.
- [12] Chen Y, Bindel D, Song H, *et al.* An algebraic approach to practical and scalable overlay network monitoring [C]. Proceedings of ACM SIGCOMM'04, Portland, Oregon, USA, Aug. 2004: 55-66.
- [13] Clegg R G, Cairano-Gilfedder C D, and Zhou S. A critical look at power law modelling of the Internet[J]. *Computer Communications*, 2010, 33(3): 259-268.
- [14] Padmanabhan V N, Qiu L, and Wang H J. Server-based inference of internet link lossiness [C]. Proceedings of IEEE INFOCOM'03, San Francisco, CA, USA, Mar. 2003: 145-155.
- [15] Spring N, Wetherall D, and Anderson T. Scriptroute: a public internet measurement facility [C]. Proceedings of the 4th Conference on USENIX Symposium on Internet Technologies and Systems (USITS), Seattle, WA, USA, Mar. 2003: 225-238.
- 顾 然: 男, 1987年生, 硕士生, 研究方向为计算机网络的故障管理与性能管理.
- 邱雪松: 男, 1973年生, 教授, 研究方向为网络管理与通信软件.
- 孟洛明: 男, 1955年生, 教授, 研究方向为网络管理与通信软件.