

# 一种高效的单播网络自适应拓扑推测算法

杨京礼, 姜守达, 魏长安, 孙 超

(哈尔滨工业大学自动化测试与控制系, 黑龙江哈尔滨 150001)

**摘 要:** 为提高单播网络拓扑推测的效率和准确性, 本文提出一种高效的单播网络自适应拓扑推测算法. 该算法利用探测包中的 TTL 信息作为二分深度优先搜索排序中参考节点的选择依据, 有效降低测量过程中所需的背靠背包对数量, 提高拓扑推测的效率; 通过对现有的深度优先搜索下网络拓扑推测算法原理的分析, 给出该算法能够正确推测出整个网络拓扑的充分条件, 在此基础上提出一种自适应判定阈值选择方法, 有效提高网络链路参数未知情况下的拓扑推测准确性. 仿真实验结果表明, 本文所提出的算法具备更高的拓扑推测效率和准确性.

**关键词:** 网络测量; 网络层析成像; 拓扑推测; 深度优先搜索

**中图分类号:** TP393 **文献标识码:** A **文章编号:** 0372-2112 (2013) 10-1888-07

**电子学报 URL:** <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2013.10.003

## Efficient and Adaptive Topology Inference Algorithm for Unicast Network Measurements

YANG Jing-li, JIANG Shou-da, WEI Chang-an, SUN Chao

(Automatic Test and Control Institute, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China)

**Abstract:** In order to improve the accuracy and efficiency of the topology inference algorithm for unicast network, an efficient and adaptive topology inference algorithm is proposed. With the information of TTL hop count, this algorithm reduces the number of the probe pairs needed in the process of bisection Depth-First Search Ordering, and improve the efficiency of the topology inference. On the other hand, through the analysis of the principle of the Depth-First Search topology inference algorithm, a sufficient condition for the algorithm to return the correct network topology is given. Based on this condition, an adapt threshold selection method is proposed, it can improve the accuracy of the topology inference when the network link parameters are unknown. Simulation results show this algorithm can obtain a higher accuracy and efficiency.

**Key words:** network measurement; network tomography; topology inference; depth-first search

### 1 引言

网络拓扑结构对于网络资源管理、网络协议设计和网络结构优化具有重要的意义<sup>[1]</sup>. 随着以太网技术的发展, 网络系统的异构性和复杂性不断提高, 尤其是出于网络安全和商业利益的考虑, 各服务商所提供的网络系统和设备往往具有非协作的特点, 使得直接测量的可行性大大降低<sup>[2]</sup>. 为了克服上述问题, 近年来国际上提出了网络层析成像技术(Network Tomography, NT)<sup>[3]</sup>, 利用可直接测量的端到端信息建立数学模型并通过统计方法进行求解, 从而获得网络内部链路级性能参数(如丢包率和时延)和拓扑结构等信息.

基于网络层析成像技术的拓扑推测算法最先应用

于多播网络中<sup>[4]</sup>, 由于以太网中许多路由设备不支持多播模式, 因此基于单播的拓扑推测算法逐渐成为研究的热点<sup>[5]</sup>. M Coates<sup>[6,7]</sup>和 M Shih<sup>[8]</sup>分别提出了最大似然树算法(Maximum Likelihood Tree, MLT)和基于图论的分层拓扑推测算法(Hierarchical Topology Estimation, HTE). Ni Jian<sup>[9]</sup>研究了单播/多播模式下的加性特征量, 在此基础上提出了邻节点合并算法(Rooted Neighbor-Joining, RNJ)进行拓扑结构的计算. 上述各种算法需要的探测包对数量极大, 造成拓扑推测的效率较低. 针对以太网拓扑结构动态变化的特点, Ni Jian<sup>[10]</sup>提出了一种序列拓扑推测算法(Sequential Topology Inference, STI), 能够有效减少拓扑推测所需要的探测包对数量, 提高拓扑推测的效率. B Eriksson<sup>[11,12]</sup>提出了基于深度优先搜索(Depth-First

Search, DFS)的拓扑推测算法,利用树型结构的深度优先搜索排序策略进一步提高拓扑推测的效率.但 DFS 算法在对目的节点进行深度优先排序时采用随机策略进行参考节点的选取,对于非对称网络可能造成大量的冗余探测,并且只能适用于解决网络内部链路性能参数已知条件下的拓扑推测问题.

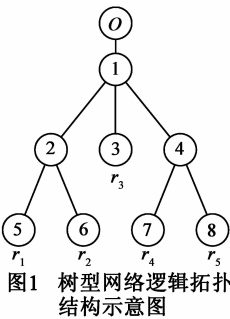
本文分析了 DFS 算法所需背靠背包对数量与二分深度优先搜索排序过程中所选参考节点之间的关系;通过理论推导出 DFS 算法能够正确推测网络拓扑结构的充分条件,在此基础上提出了一种高效的单播网络自适应拓扑推测算法(Efficient and Adaptive Topology Inference Algorithm for Unicast Network, EATI).该算法利用探测包中的 TTL 信息选取参考节点对目的节点进行深度优先搜索排序,减少单播测量中所需的背靠背包对数量,提高拓扑推测的效率;采用自适应的判定阈值选择方法,有效解决网络内部链路参数未知时的判定阈值选择问题.仿真实验结果表明,该算法可以显著提高拓扑推测的效率,并具备较高的拓扑推测准确性.

## 2 网络模型与单播拓扑测量

### 2.1 网络模型

网络层析成像技术下的网络结构<sup>[10]</sup>可以表示为一个有向图模型  $J = (\mathbf{v}, \boldsymbol{\varepsilon})$ ,  $\mathbf{v}$  表示有向图中的节点集合,代表网络中的路由器和终端;  $\boldsymbol{\varepsilon}$  表示有向图中的边集合,代表网络中的链路.对于网络中的节点  $i$  和  $j$ ,如果按照路由策略可以获得连接  $i$  和  $j$  的一系列链路,则称  $j$  对于  $i$  是可到达的.假定在测量过程中,两个节点之间的路径保持唯一不变,则从一个源端到若干目的节点的路由拓扑符合有向树结构.从网络路由拓扑结构可以导出逻辑拓扑结构(仅包含路由拓扑中的源节点、目的节点和子节点数量不小于 2 的内部节点),图 1 为树型网络逻辑拓扑结构示意图.

假设  $O$  表示网络中的源节点,  $\mathbf{R} = \{r_1, r_2, \dots, r_{|\mathbf{R}|}\}$  表示网络中的对源节点  $O$  所有可到达的目的节点集合,  $|\mathbf{R}|$  为目的节点个数.用  $T = (\mathbf{V}, \mathbf{E})$  表示由源节点  $O$  到目的节点集合  $\mathbf{R}$  组成的树型逻辑拓扑,其中  $\mathbf{V}$  表示所有节点的集合,  $\mathbf{E}$  表示所有链路的集合.每个非根节点  $k$  都有一个父节点,用  $f(k)$  表示.链路  $(f(k), k) \in \mathbf{E}$  记为链路  $e_k$ , 定义  $f^1 = f$  和  $f^n(k) = f(f^{n-1}(k))$ .如果  $k = f^n(j)$  成立,则  $j$  是  $k$  的子孙节点,记为  $j < k$ .集合  $U \subset \mathbf{V}$  中所有节点最近的共同祖先记为  $\alpha(U)$ , 如果集合  $U$  中所有节点的父节点相同,  $U$  中的节点为兄弟节



点.  $(\mathbf{V}(k), \mathbf{E}(k))$  表示以节点  $k$  为根的子树, 该子树目的节点集合为  $\mathbf{R}(k) = \mathbf{R} \cap \mathbf{V}(k)$ .

### 2.2 单播拓扑测量

在单播测量方式上,探测包对可以分为背靠背包、三明治包和四元列车包等类型.背靠背包是目前最常用的测量方式,每个包对由发送到不同目的节点的两个相邻探测包组成.由于探测包之间的时间间隔极小,可认为两个探测包在共享路径上经历的网络状态相同.Ni Jian<sup>[9]</sup>使用马尔科夫随机场模型(Markov Random Field, MRF)对加性特征量进行建模.对于两个目的节点  $r_m$  和  $r_n$ , 其共同祖先节点  $\alpha(r_m, r_n)$ , 用  $p_r$  和  $p_n$  表示从源节点  $O$  到上述两个节点的路径,  $p_{r_m}$  表示从源节点  $O$  到其共同祖先节点的路径.假设一个背靠背包对中的两个探测包从源节点分别发送到节点  $r_m$  (探测包  $b1$ ) 和  $r_n$  (探测包  $b2$ ), 则基于丢包率的链路加性特征量为:

$$\varphi(e) = -\lg(\alpha_e), \forall e \in \mathbf{E} \quad (1)$$

其中,  $\alpha_e$  为链路  $e$  的丢包率, 路径  $p_r$ 、 $p_n$  和  $p_{r_m r_n}$  的加性特征量分别为:

$$\Phi(r_m) = -\lg P(X_{r_m}^{b1} = 1) \quad (2)$$

$$\Phi(r_n) = -\lg P(X_{r_n}^{b2} = 1) \quad (3)$$

$$\Phi(r_m r_n) = -\lg \frac{P(X_{r_m}^{b1} = 1) P(X_{r_n}^{b2} = 1)}{P(X_{r_m}^{b1} = 1 \cap X_{r_n}^{b2} = 1)} \quad (4)$$

其中,若目的节点  $r_m$  接收到探测包  $b1$ , 则  $X_{r_m}^{b1} = 1$ , 否则  $X_{r_m}^{b1} = 0$ .若两个节点共享的路径越长, 则从源节点到其祖先节点路径的加性特征量的值也越大, 因此可以使用加性特征量作为拓扑推测的依据.

## 3 高效的单播自适应拓扑推测算法

本文提出的 EATI 算法主要流程与文献[11, 12]提出的 DFS 算法保持一致, 分为二分深度优先搜索排序和基于 DFS 序列的拓扑关系计算两个独立的过程, 前一个过程决定了所需的背靠背包对数量, 即拓扑推测的效率, 后一个过程决定了拓扑推测的准确性. EATI 算法在深度优先搜索排序过程中利用探测包中的 TTL 信息进行参考节点选择, 减少所需的背靠背包对数量, 提高拓扑推测的效率; 在拓扑关系计算过程中采用了自适应的判定阈值选择方法, 提高拓扑推测的准确性.

### 3.1 基于 TTL 的深度优先搜索排序

对目的节点集合进行深度优先搜索排序的原理是通过选定一个目的节点作为参考节点, 依次向该参考节点及剩余目的节点中的每个节点发送背靠背包对, 以共享路径加性特征量为依据, 通过最优二分递归的方法依次对目的节点进行聚类, 最终得到符合深度优先搜索的目的节点序列.

**定理 1** 设  $T$  为具有  $N$  个目的节点的树型逻辑拓扑, 每层拓扑具有  $M$  个目的节点, 则二分深度优先搜索排序过程所需的背靠背包对数量在  $[C\_low, C\_high]$  之间.

**证明** 在二分深度优先搜索排序算法中, 若选定的参考节点与源节点之间路径所包含的链路越多, 则在计算过程中能够获取的分割点数越多, 递归过程中的二分次数越少, 对应需要的背靠背包对数量也就越少; 反之需要的包对数量就越多.

(1) 若每次二分过程中所选的参考节点为最底层节点, 所需要的背靠背包对数量最少. 对于每层具有  $M$  个目的节点的树型拓扑, 其层数为  $N/M$ . 首先进行  $\log_2 N/M$  次二分过程, 将各节点按层次划分到相应的聚类中: 第 1 次二分所需要的背靠背包对数量为  $N-1$ , 之后第  $i$  ( $1 < i \leq \log_2 N/M$ ) 次二分所需要的背靠背包对数量为  $(N/2^{i-1} - 1) \times 2^{i-2}$ ; 最后对各层次节点聚类分别进行二分, 每个聚类所需要的背靠背包对数量为  $\sum_{j=3}^M (j-1)$ , 整个深度优先搜索排序过程所需要的背靠背包对总数量为:

$$C\_low = N - 1 + \sum_{i=2}^{\log_2 N/M} (N/2^{i-1} - 1) \times 2^{i-2} + \frac{N}{M} \times \sum_{j=3}^M (j-1) \quad (5)$$

(2) 若每次二分过程中所选的参考节点为最顶层节点, 所需要的背靠背包对数量最多. 由于每次二分过程只能确定一个节点的位置, 因此需要  $N-2$  次二分过程, 整个深度优先搜索排序过程所需要的背靠背包对总数量为:

$$C\_high = \sum_{i=3}^N (i-1) \quad (6)$$

图 2 给出了  $M=4$  时,  $C\_low$  与  $C\_high$  随目的节点数量的变化情况, 目的节点数量越多,  $C\_low$  与  $C\_high$  之间的差距越大.

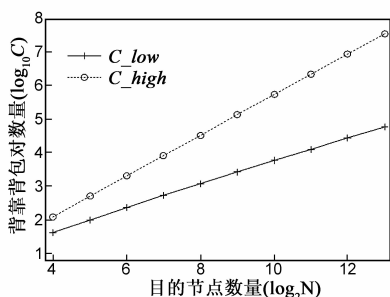


图2  $C\_low$ 与 $C\_high$ 随目的节点数量的变化情况

由于目前以太网中的数据包大多为含有 TTL (time-to-live) 域的 IP 包, 探测包从源节点到目的节点经过的路由个数 (包含的链路个数) 可由 IP 包中的 TTL 域得

到. 虽然路由拓扑与逻辑拓扑并不完全一致, 但将 TTL 信息作为选择参考节点的依据, 仍能一定程度上表征源节点与目的节点之间路径所包含链路的数量. 根据定理 1 的证明过程, 选择 TTL 跳数信息较大的目的节点作为参考节点, 能够减少测量过程所需要的背靠背包对数量, 提高拓扑推测的效率. 基于 TTL 的二分深度优先搜索排序过程如算法 1 所示.

#### 算法 1 基于 TTL 的二分深度优先搜索排序算法 ( $biset(R, \delta)$ )

输入: 随机目的节点序列  $R = \{r_1, r_2, \dots, r_{|R|}\}$ , 判定阈值  $\delta$   
 步骤 1: 从源节点向  $R$  中各目的节点发送单探测包, 收集 TTL 信息  
 步骤 2: 根据 TTL 信息, 选择参考节点, 向参考节点与剩余节点发送背靠背包对, 计算共享路径加性特征量  $\hat{\Phi}(r_j r_i), 1 \leq j \leq |R|, j \neq i$   
 步骤 3: 按照  $\hat{\Phi}(r_j r_i)$  对目的节点序列进行排序, 得到序列位置向量  $I$   
 步骤 4: 按  $I_\delta = |k, \hat{\Phi}(r_{I(k+1)} r_{I(k)}) - \hat{\Phi}(r_{I(k)} r_{I(k-1)})| > \delta/2$ , 寻找分割点  
 步骤 5: 通过  $k^* = \arg \min_{k \in I_\delta} |k - |R|/2|$ , 寻找最优二分节点, 将其分为两个目的序列  $R_1$  和  $R_2$   
 步骤 6: 若  $|R_1| > 2, I_1 = biset(R_1, \delta)$ ; 若  $|R_2| > 2, I_2 = biset(R_2, \delta)$   
 步骤 7: 分别使用  $I_1$  和  $I_2$  对序列  $R_1$  和  $R_2$  进行排序  
 输出: 产生符合 DFS 的目的节点序列  $R^{DFS} = \{r_1, r_2, \dots, r_{|R|}\}$

注: 其中  $\hat{\Phi}(r_j r_i)$  是  $\Phi(r_j r_i)$  的估计值.

### 3.2 基于 DFS 序列的拓扑关系计算

由于基于丢包率和时延的加性特征量在各链路上都取正值, 结合文献[12]中引理 1 的证明过程, 容易得到如下定理:

**定理 2** 对于一个符合深度优先搜索的序列  $R^{DFS} = \{r_1, r_2, \dots, r_{|R|}\}$ , 若  $r_m$  和  $r_n$  是  $R^{DFS}$  中的两个目的节点,  $\Phi(r_m r_n)$  具有以下性质:

$$\Phi(r_i r_{i+j}) \geq \Phi(r_i r_{i+k}), 1 \leq i \leq |R|, 1 \leq j \leq k \leq |R| \quad (7)$$

对于网络节点  $k$ , 父节点为  $f(k)$ , 从源节点  $O$  到节点  $k$  和  $f(k)$  的路径分别为  $p_k$  和  $p_{f(k)}$ , 对应的路径加性特征量为  $\Phi(k)$  和  $\Phi(f(k))$ . 链路  $e_k$  上的加性特征量可按下式进行计算:

$$\varphi(e_k) = \Phi(k) - \Phi(f(k)) \quad (8)$$

EATI 拓扑推测算法中基于 DFS 序列的拓扑关系计算过程如算法 2 所示.

#### 算法 2 基于 DFS 序列的拓扑关系计算过程

输入: 符合深度优先搜索的目的节点序列  $R^{DFS}$  及判定阈值  $\delta$   
 初始化: 初始化  $V = \{s_1, r_1, r_2\}, E = \{(s_1, r_1), (s_1, r_2)\}$   
 处理过程: for  $i = \{3, 4, \dots, |R|\}$   
 步骤 1: 向节点集合内增加节点  $r_i, V = V \cup \{r_i\}$   
 步骤 2: 如果  $|\hat{\Phi}(r_{i-1} r_{i-2}) - \hat{\Phi}(r_i r_{i-1})| < \delta/2, r_{i-1}$  父节点即  $r_i$  父节点,  $E = E \cup \{(f(r_{i-1}), r_i)\}$   
 步骤 3: 如果  $\hat{\Phi}(r_i r_{i-1}) \geq \hat{\Phi}(r_{i-1} r_{i-2}) + \delta/2$ , 产生节点  $s_i, V = V \cup \{s_i\}; s_i$  是  $r_{i-1}$  父节点的子节点,  $E = E \cup \{(s_i, f(r_{i-1}))\};$  去除  $r_{i-1}$  与

其原父节点链路,  $E = E \setminus \{(f(r_{i-1}), r_{i-1})\}$ ;  $s_i$  为  $r_i$  与  $r_{i-1}$  的父节点,

$E = E \cup \{(s_i, r_i), (s_i, r_{i-1})\}$ ;  $\hat{\Phi}(s_i) = \hat{\Phi}(r_{i-1})$

步骤 4: 如果  $\hat{\Phi}(r_{i-1}) \leq \hat{\Phi}(r_{i-1}r_{i-2}) - \delta/2$ , 在  $r_{i-1}$  祖先节点中寻找  $\hat{\Phi}(pa) \geq \hat{\Phi}(r_{i-1})$  的节点  $pa$

步骤 4.1: 如果  $\hat{\Phi}(pa) - \hat{\Phi}(r_{i-1}) < \delta/2$ , 令  $pa$  为  $r_i$  的父节点,  $E = E \cup \{(pa, r_i)\}$

步骤 4.2: 如果  $\hat{\Phi}(pa) - \hat{\Phi}(r_{i-1}) \geq \delta/2$ , 产生节点  $s_i$ ,  $V = V \cup \{s_i\}$ ;  $s_i$  为  $pa$  及  $r_i$  的父节点,  $f(pa)$  的子节点; 去除  $pa$  与  $f(pa)$  的链路,  $E = E \cup \{(f(pa), s_i), (s_i, pa), (s_i, r_i)\} \setminus \{f(pa), pa\}$

步骤 5: 网络内部链路参数未知时: 根据式(8)计算链路  $e_{i-1} = (f(r_{i-1}), r_{i-1})$  的加性特征量  $\hat{\varphi}(e_{i-1})$ , 更新  $\{\hat{\varphi}(e_1), \dots, \hat{\varphi}(e_n)\}$ , 更新  $\delta = \min\{\hat{\varphi}(e_1), \dots, \hat{\varphi}(e_n)\}$

输出: 拓扑结构  $\hat{T} = (V, E)$

与 DFS 算法不同, 当网络内部链路参数未知时, EATI 算法在进行拓扑推测时, 计算各链路的加性特征量, 更新判定阈值。

DFS 算法在拓扑关系计算过程中使用加性特征量的估计值进行节点关系的判断。虽然加性特征量的估计值是其真实值的无偏估计, 但实际测量中估计值与真实值存在一定的误差。下面给出当加性特征量估计值和真实值存在误差时, DFS 算法能够正确推测出网络拓扑结构的充分条件。

**定理 3** 对于符合深度优先搜索的序列  $R^{DFS} = \{r_1, r_2, \dots, r_{|R|}\}$ , 判定阈值  $\delta \leq \min_{e \in E} \varphi(e)$ , 则 DFS 算法正确推测出网络拓扑的充分条件是:

$$|\Phi(r_i r_{i-1}) - \hat{\Phi}(r_i r_{i-1})| < \delta/4, \forall i \in \{2, 3, \dots, |R|\} \quad (9)$$

**证明** 按照 DFS 算法的原理, 在推测  $r_i$  的位置时, 仅需要判断  $r_i$  与  $r_{i-1}, r_{i-2}$  的关系, 因此可以简化  $r_{i-1}$  与  $r_{i-2}$  的关系, 使其成为逻辑上的兄弟节点。在  $r_{i-1}$  与  $r_{i-2}$  确定后,  $r_i$  与  $r_{i-1}, r_{i-2}$  的关系有如图 3 所示三种可能。

(1) 如图 3(a) 所示,  $r_i, r_{i-1}$  与  $r_{i-2}$  互为兄弟节点, 故  $\Phi(r_i r_{i-1}) = \Phi(r_{i-1} r_{i-2})$ , 根据式(9)可知:

$$\begin{aligned} \Phi(r_i r_{i-1}) - \delta/4 < \hat{\Phi}(r_i r_{i-1}) < \Phi(r_i r_{i-1}) + \delta/4 \\ \Phi(r_{i-1} r_{i-2}) - \delta/4 < \hat{\Phi}(r_{i-1} r_{i-2}) < \Phi(r_{i-1} r_{i-2}) + \delta/4 \end{aligned} \quad (10)$$

因此  $-\delta/2 < \hat{\Phi}(r_i r_{i-1}) - \hat{\Phi}(r_{i-1} r_{i-2}) < \delta/2$ , 进入步骤 2 进行处理。

(2) 如图 3(b) 所示,  $r_i$  与  $r_{i-1}$  父节点是  $r_{i-1}$  与  $r_{i-2}$  父节点的子节点, 故  $\Phi(r_i r_{i-1}) \geq \Phi(r_{i-1} r_{i-2}) + \delta$ 。由式(9):  $\hat{\Phi}(r_i r_{i-1}) > \Phi(r_i r_{i-1}) - \delta/4 > \Phi(r_{i-1} r_{i-2}) + 3\delta/4 > \hat{\Phi}(r_{i-1} r_{i-2}) + \delta/2$ , 进入步骤 3 进行处理。

(3) 如图 3(c) 所示,  $r_i$  与  $r_{i-1}$  父节点是  $r_{i-1}$  与  $r_{i-2}$  父节点的祖先节点, 故  $\Phi(r_i r_{i-1}) \leq \Phi(r_{i-1} r_{i-2}) - \delta$ 。由式(9):  $\hat{\Phi}(r_i r_{i-1}) < \Phi(r_i r_{i-1}) + \delta/4 < \Phi(r_{i-1} r_{i-2}) - 3\delta/4 < \hat{\Phi}(r_{i-1} r_{i-2}) - \delta/2$ , 进入步骤 4 进行处理。

在步骤 4 中, 需要寻找  $r_{i-1}$  的父节点序列中加性特征量值小于  $a(r_i, r_{i-1})$  加性特征量值的节点  $pa$ 。按照 DFS 算法, 该情况可分为两种情况:

(1) 若节点  $pa$  是  $r_i$  与  $r_{i-1}$  的父节点, 因此  $\Phi(pa) = \Phi(r_i r_{i-1})$ , 由式(9):  $\hat{\Phi}(pa) - \hat{\Phi}(r_i r_{i-1}) < \Phi(pa) + \delta/4 - (\Phi(r_i r_{i-1}) - \delta/4) = \delta/2$ , 进入步骤 4.1 进行处理。

(2) 若  $r_i$  与  $r_{i-1}$  的父节点是节点  $pa$  的父节点, 故  $\Phi(pa) \geq \Phi(r_i r_{i-1}) + \delta$ 。由式(9):  $\hat{\Phi}(pa) > \Phi(pa) - \delta/4 > \Phi(r_i r_{i-1}) + 3\delta/4 > \hat{\Phi}(r_i r_{i-1}) + \delta/2$ , 进入步骤 4.2 进行处理。

综上所述, 式(9)是 DFS 算法正确推测出网络拓扑结构的充分条件。

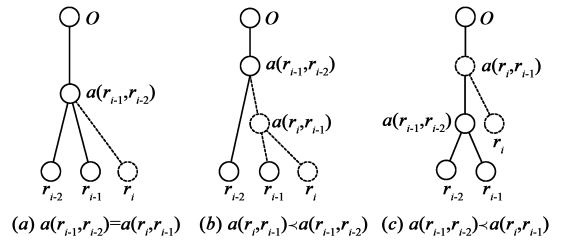


图 3  $r_i$  与  $r_{i-1}, r_{i-2}$  的关系

由定理 3 可知, DFS 拓扑推算算法能够正确推测出网络拓扑结构取决于两个因素: 一个是共享路径加性特征量的估计值与真实值的误差  $|\Phi(r_i r_{i-1}) - \hat{\Phi}(r_i r_{i-1})|$ , 另一个是判定阈值  $\delta$  的大小。假设网络中各链路丢包和时延相互独立, 对  $\Phi(r_i r_{i-1})$  进行  $n$  次采样得到采样值  $\{\tilde{\Phi}^1(r_i r_{i-1}), \tilde{\Phi}^2(r_i r_{i-1}), \dots, \tilde{\Phi}^n(r_i r_{i-1})\}$ ,  $\hat{\Phi}(r_i r_{i-1})$  为  $n$  次采样均值。根据中心极限定理: 若  $n \rightarrow \infty$ ,  $|\Phi(r_i r_{i-1}) - \hat{\Phi}(r_i r_{i-1})| \rightarrow 0$ , 因此通过增加采样数量来减小共享路径加性特征量的估计值与真实值的误差, 提高拓扑推测的准确性。

另一方面, 判定阈值  $\delta$  的大小对拓扑推测的准确性也有较大的影响。根据定理 3, 在采样数量一定的情况下,  $\delta$  取值较小时会使式(9)成立概率降低,  $\delta$  取值较大时会在拓扑推测过程中引入节点位置估计错误。文献[12]仅仅建议判定阈值  $\delta$  的大小应当不大于网络中的最小链路加性特征量, 但并未给出具体的取值方式。本文根据定理 3 及其证明过程, 建议判定阈值  $\delta$  的取值应为网络中最小链路加性特征量。

## 4 仿真实验与分析

为综合评价 EATI 算法的性能, 在拓扑推测的效率和准确性方面与目前性能较好的 STI 和 DFS 等算法进行比较。

### 4.1 拓扑推测效率

#### 4.1.1 仿真网络拓扑

近年来的相关研究结果表明<sup>[13]</sup>, 幂率分布更能够

描述当前以太网拓扑结构. 本文采用 BRUTE 拓扑生成器<sup>[14]</sup>产生仿真网络拓扑结构, 网络路由节点规模在 100 - 1000 之间, 逻辑拓扑中的节点个数在 50 - 800 之间. 在实验过程中, 选择度数为 1 的节点作为端节点, 并随机选择一个端节点作为背靠背包对发送的源节点, 其余端节点作为目的节点. 各种规模的拓扑结构均生成 100 次, 每种拓扑下的实验均进行 100 次, 取各实验过程中所需背靠背包对数量的平均值作为测量结果.

每次实验过程中, DFS 算法的参考节点随机选定, EATI 算法的参考节点按照与源节点之间路径最长的原则选定. 为保证能够正确推测出网络拓扑结构, 在实验过程中设定没有任何噪声的干扰 (Noise-Free), 各链路加性特征量 (加性特征量相等且为 0.25, 判定阈值大小为 0.1) 的估计值等于其真实值, 即  $|\Phi(r_i r_{i-1}) - \hat{\Phi}(r_i r_{i-1})| = 0$ , 式(9)恒成立, 即保证算法能够 100% 的推测出网络的拓扑结构, 因此本实验过程是在拓扑推测准确性一致的条件下, 比较各种算法所需背靠背包对数量. 图 4 为各种算法所需背靠背包对数量随网络节点数量变化情况.

从图 4 可以看出在网络规模相同的条件下, DFS 算法所需的背靠背包对数量是 STI 算法的 20% ~ 40% 左右; 与 DFS 算法相比, EATI 算法能够节省 15% ~ 20% 的背靠背包对数量, 有效提高拓扑推测的效率.

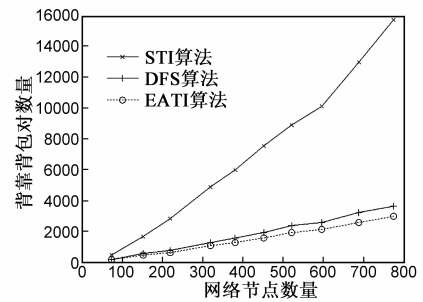


图 4 各种算法所需背靠背包对数量随网络节点数量的变化情况

#### 4.1.2 真实网络拓扑

Internet Topology Zoo 数据库<sup>[15]</sup>收集了许多实际网络拓扑结构, 从该数据库中选择若干网络拓扑结构验证 EATI 算法的性能. 各种网络拓扑下的实验均进行 100 次, 每次实验中随机选定源节点和目的节点来生成树型拓扑结构, 取各次实验过程中所需背靠背包对数量的平均值作为结果. 表 1 为不同网络拓扑结构下各种算法所需背靠背包对数量的对比.

从表 1 可以看出, 与仿真拓扑实验结果类似, 实际网络拓扑下 EATI 算法需要的背靠背包对数量最少, 且随着网络规模的不断增大, EATI 算法能够节省更多的背靠背包对数量. 与 DFS 算法相比, EATI 节省的背靠背包对比例在 8.6% (BT Asia) ~ 19.4% (GTS CE) 之间.

表 1 Internet Topology Zoo 网络拓扑下各种算法所需背靠背包对数量

网络名称	节点数	STI	DFS	EATI	网络名称	节点数	STI	DFS	EATI
BT Asia	20	101.3	54.3	49.6	China Tel	42	445.7	140.9	116.1
BT Europe	24	178.9	65.4	59.7	Renater	43	457.1	145.3	128.4
AGIS	25	185.2	70.1	63.2	Bell South	51	723.4	180.6	154.3
VinaREN	25	187.3	71.5	64.6	CESNET	52	736.1	192.4	160.9
WIDE	30	253.4	89.7	72.5	Tinet	53	762.6	198.1	168.7
BICS	33	281.1	107.8	88.3	DFN	58	862.9	218.6	189.2
ARNES	34	305.2	109.5	90.2	GARR	61	912.6	230.0	196.8
IJ	37	398.3	120.4	97.9	GTS CE	149	2811.8	645.8	520.3
GEANT	40	423.4	135.2	110.3	Cogent	197	3774.5	812.3	672.7

#### 4.2 拓扑推测准确性

为验证实际网络环境中各种算法的拓扑推测准确性, 本文在 NS2 工具中建立与真实网络环境类似的仿真环境. 如图 5 所示, 采用与文献[11]相同的网络拓扑

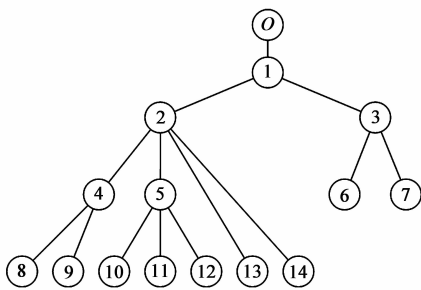


图 5 仿真网络拓扑结构

结构. 网络仿真环境的参数设置如下: 网络内部链路的带宽为 5Mb/s, 固定传输时延为 50ms; 边缘链路的带宽为 1Mb/s, 固定传输时延为 10ms. 链路的缓冲队列为 FIFO (First-In-First-Out) 队列, 缓存大小为 10 个数据包, 采用尾部丢弃 (Drop-tail) 拥塞避免算法. 源节点  $O$  以背靠背包对模式向各目的节点发送 UDP 探测包, 背靠背包对的发送速度为 20Kbit/s, 背靠背包对内部两个探测包紧邻发送, 每个探测包的大小为 40Bytes. 采用符合 Pareto ON-OFF 模型的背景流模拟互联网的真实网络状况, 链路中同时存在多条 TCP 与 UDP 链接, TCP 流量和 UDP 流量的比例为 9:1. 在整个仿真过程中, 整个网络的链路利用率在 30% ~ 70% 之间变化, 链路丢包率在 1

~ 11% 之间变化。

在实验过程中,采用基于丢包率的加性特征量进行网络拓扑结构的计算,上限判定阈值为  $\delta_{up} = \min_{e \in E} \varphi(e)$ 。由于基于网络层析成像技术的拓扑推测过程中,网络内部各链路的性能参数(如丢包率和时延等)通常情况下都是未知的。文献[8]认为如果判定阈值选择大于最小链路加性特征量,则可能会在拓扑推测过程中湮没那些链路加性特征量小于判定阈值的链路,因此该文建议在此时选取尽量小的判定阈值。

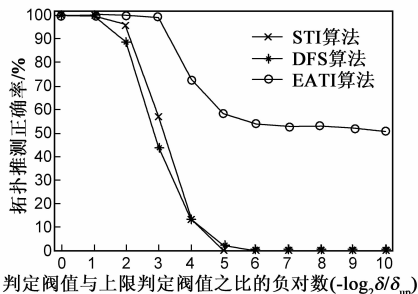


图6 拓扑推测正确率随判定阈值的变化情况

图6给出了当探测包数量为10000时,各种算法拓扑推测正确率随判定阈值与上限判定阈值之比( $\delta/\delta_{up}$ )的变化情况。在实验过程中假设STI算法与DFS算法的判定阈值及EATI算法的初始判定阈值从 $\delta_{up}$ 开始减小。从图6中可以看出,当判定阈值 $\delta$ 取值为上限判定阈值 $\delta_{up}$ 时,DFS算法的正确率最高;当判定阈值从 $\delta_{up}$ 减小时,式(9)成立的概率降低,拓扑推测正确率下降,上述过程与3.2节的分析过程一致。

随着判定阈值的减小,STI与DFS算法拓扑推测的正确率逐渐降低,当 $\delta/\delta_{up} < 2^{-6}$ 时,上述两种算法已经很难正确推测出网络拓扑结构。与之不同的是,EATI算法在判定阈值较小时,仍能保持一定的拓扑推测正确率,这是由于EATI算法在拓扑推测过程中根据计算出的链路加性特征量更新判定阈值,使其对初始判定阈值的选择具备自适应特点,可以显著改善由于判定阈值选择过小造成的拓扑推测准确性低的问题。

图7给出了初始判定阈值 $\delta = \delta_{up}/2$ 时,各种算法拓扑推测准确率随探测包数量的变化情况。当探测包数量为1500时,EATI算法拓扑推测的正确率为97%,

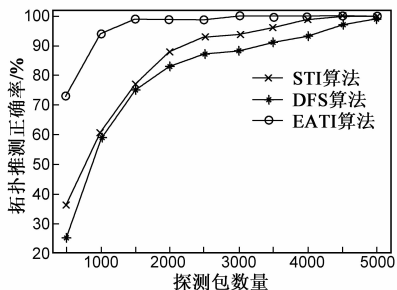


图7 拓扑推测正确率随探测包数量的变化情况( $\delta = \delta_{up}/2$ )

与STI算法(76%)相比提高了21%,与DFS算法(74%)相比提高了23%。

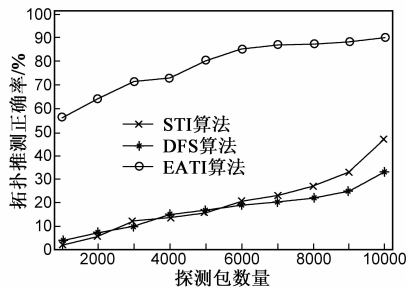


图8 拓扑推测正确率随探测包数量的变化情况( $\delta = \delta_{up}/10$ )

当初始判定阈值进一步减小时,STI算法与DFS算法拓扑推测正确率大幅降低。图8给出了初始判定阈值 $\delta = \delta_{up}/10$ 时,各种算法拓扑推测正确率随探测包数量的变化情况。当探测包数量为5000时,EATI算法拓扑推测的正确率(81%)是STI算法(16%)和DFS算法(17%)的5倍左右。

## 5 结论

本文分析了DFS拓扑推测算法所需背靠背包对数量与参考节点的关系,通过理论推导出DFS算法能够正确推测网络拓扑结构的充分条件,在此基础上提出一种高效的单播网络自适应拓扑推算算法。仿真实验结果表明,本文提出的EATI算法与DFS算法相比,能够节省15%~20%左右的背靠背包对,显著提高拓扑推测的效率;当判定阈值选择过小时,需要较少的探测包即能够保证较高的拓扑推测准确率,当探测包数量为5000( $\delta = \delta_{up}/10$ )时,EATI算法拓扑推测正确率是STI和DFS算法的5倍左右。

## 参考文献

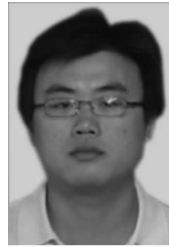
- [1] Coates M, Hero A, Nowak R, et al. Internet tomography[J]. IEEE Signal Processing Magazine, 2002, 19(3): 47 - 65.
- [2] Donnet D, Friedman T. Internet topology discovery: A survey [J]. IEEE Communications Surveys and Tutorials, 2007, 9(4): 2 - 15.
- [3] 赵洪华,陈鸣.基于网络层析成像技术的拓扑推断[J].软件学报, 2010, 21(1): 133 - 146. Zhao Hong-hua, Chen Ming. Topology inference based on network tomography[J]. Journal of Software, 2010, 21(1): 133 - 146.
- [4] Duffield N, Horowitz J, et al. Multicast topology inference from measured end-to-end loss[J]. IEEE Transactions on Information Theory, 2002, 48(1): 26 - 45.
- [5] Zhang X, Phillips C. A survey on selective routing topology inference through active probing[J]. IEEE Communications Sur-

- veys and Tutorials, 2011, 9(23): 1 – 13.
- [6] Coates M, Castro R, Nowak R. Maximum likelihood network topology identification from edge-based unicast measurements [A]. International Conference on Measurement and Modeling of Computer Systems [C]. Marina Del Rey: ACM, 2002. 11 – 20.
- [7] Fei G, Hu G. Improving maximum-likelihood-based topology inference by sequentially inserting leaf nodes [J]. IET Communications, 2011, 5(15): 2221 – 2230.
- [8] Shih Meng-Fu, Hero III, et al. Hierarchical inference of unicast network topologies based on end-to-end measurements [J]. IEEE Transactions on Signal Processing, 2007, 55(51): 1708 – 1718.
- [9] Ni Jian, Tatikonda S. A markov random field approach to multicast-based network inference problems [A]. International Symposium on Information Theory [C]. Seattle: IEEE, 2006. 2769 – 2773.
- [10] Ni Jian, Xie H H, Tatikonda S, et al. Efficient and dynamic routing topology inference from end-to-end measurements [J]. IEEE/ACM Transactions on Networking, 2010, 18(1): 123 – 135.
- [11] Eriksson B, Dasarathy G, Barford P, et al. Toward the practical use of network tomography for internet topology discovery [A]. Proceedings of IEEE INFO-COM 2010 [C]. San Diego: IEEE, 2010. 1 – 9.
- [12] Eriksson B, Dasarathy G, Barford P, Nowak R. Efficient network tomography for internet topology discovery [J]. IEEE/ACM Transactions on Networking, 2012, 20(3): 931 – 943.
- [13] Clegg R G, Cairano-Gilfedder C D, Zhou S. A critical look at

power law modeling of the Internet [J]. Computer Communications, 2010, 33(3): 259 – 268.

- [14] Alerto M, Ibrahim M, John B. On the origin of power laws in internet topologies [J]. ACM Computer Communication Review, 2000, 30(2): 18 – 28.
- [15] Knight K, Nguyen H X, et al. The internet topology zoo [OL]. <http://www.topology-zoo.org>.

#### 作者简介



**杨京礼** 男, 1984 年出生, 哈尔滨工业大学自动化测试与控制系博士研究生. 主要研究方向为网络测量与网络层析成像技术.

E-mail: icehit0615@163.com



**姜守达** 男, 1964 年出生, 哈尔滨工业大学自动化测试与控制系教授. 主要研究方向为虚拟试验技术、网络测量技术等.

**魏长安** 男, 1981 年出生, 哈尔滨工业大学自动化测试与控制系讲师. 主要研究方向为虚拟试验技术、自动测试技术等.

**孙超** 男, 1978 年出生, 哈尔滨工业大学自动化测试与控制系讲师. 主要研究方向为虚拟试验技术、图像处理等.