

# 基于丢包率的多播网络拓扑推断算法

吴文佳, 张建中, 张元鹏

(厦门大学信息科学与技术学院, 厦门 361005)

**摘要:** 在分析现有的网络拓扑推断算法的基础上, 提出一种改进的基于丢包率的多播网络拓扑推断算法。结合接收节点的层次信息、汉明距离及节点接收的探测包数量, 能够同时推断网络拓扑结构和链路丢包率, 并根据链路丢包率的估计值动态地调整拓扑推断的判决门限值, 提高了推断的准确性。仿真实验证明, 与现有算法相比, 该算法具有更好的性能。

**关键词:** 拓扑推断; 丢包率; 多播

## Algorithm of Multicast Network Topology Inference Based on Packet Loss Rate

WU Wen-jia, ZHANG Jian-zhong, ZHANG Yuan-peng

(School of Information Science and Technology, Xiamen University, Xiamen 361005)

**【Abstract】** By analyzing the existing network topology inference algorithms, this paper presents an improved multicast network topology inference algorithm based on packet loss rate. It combines the hop count of receivers, the Hamming distance and the number of received probes and can infer multicast network topology and loss performance simultaneously. It adapts dynamically the value of threshold according to the estimation of link packet loss rates, it improves the accuracy of the inference. Simulation results show that compared with the existing algorithms, this algorithm has better performance.

**【Key words】** topology inference; packet loss rate; multicast

### 1 概述

随着 Internet 的普及, 计算机网络的规模日益庞大, 人们对网络拓扑发现的需求越来越迫切。传统的一些测量方法需要网络内部路由器的协作, 但出于安全或商业利益的考虑, 不同组织不愿共享各自的网络状态, 而且将大量路由器统计的数据传递给中心网管系统, 消耗了大量带宽。为了解决上述问题, 网络层析技术(Network Tomography, NT)成为当前国内外研究的一个热点。

基于层析成像技术的网络拓扑推断使用端到端的性能测量值(如丢包率、链路延时等)<sup>[1-3]</sup>对网络内部结构及性能进行推断, 从而获得近似的网络拓扑。文献[2]描述了一种利用探测包丢包情况推断二叉树拓扑结构的 BLT 算法, 在判断兄弟节点的过程中引入了静态判决门限值  $\varepsilon$ , 将 BLT 算法从二叉树拓扑推广到一般树, 提出了通用树拓扑推断算法 GLT。但是 BLT 和 GLT 算法都不能够准确地识别出只有一个子节点的节点。针对该问题, 文献[3]对 GLT 算法加以改进, 提出 BHC 算法, 该算法根据测量的端到端探测包丢包情况, 结合节点的层次信息计算节点之间的汉明距离, 能够较为快速准确地推测出网络拓扑。但当网络中某些链路丢包较为严重时, BHC 算法的准确度明显下降。为了解决 BHC 算法存在的问题, 本文提出改进算法 HGLT(Hamming distance and hop count based classification algorithm for General Loss Tree), 通过比较和仿真分析, 证明了该算法能有效提高拓扑推断的准确率。

### 2 多播拓扑模型

$T=(V, L)$  表示一个逻辑多播树, 其中,  $V$  是节点集合;  $L$  是链路集合。根节点  $o \in V$  代表探测包的源节点, 叶节点集

$R \subset V$  表示接收节点。若  $\forall k \in V \setminus (R \cup \{o\})$ , 则称  $k$  为内部节点。每个内部节点  $k$  都至少有一个子节点  $d(k)$ 。除了根节点  $o$ , 每一个节点  $k$  都有一个父节点  $f(k)$ , 链路  $(f(k), k) \in L$  表示链路  $k$ 。节点集  $U \subset V$  中所有节点最近共同祖先节点记为  $a(U)$ , 如果集合  $U$  中所有节点拥有相同的父节点, 那么  $U$  中的节点就是兄弟节点。以节点  $k$  为根的子树表示为  $T(k)=(V(k), L(k))$ ,  $T(k)$  的接收节点集为  $R(k)=R \cap V(k)$ 。

假设树  $T=(V, L)$  的丢包率符合贝努力模型。用随机过程  $X=(X_k)_{k \in V}$  描述节点接收探测包的情况, 如果探测包到达了节点  $k$ , 则  $X_k=1$ ; 否则  $X_k=0$ 。当源节点发送了  $n$  个探测包时,  $X_k^{(i)}$  表示节点  $k$  接收第  $i$  个包的状态。每个探测包在链路  $k$  上成功传送的概率是  $p_k$ 。一般定义  $X_o=1$ ,  $p_o=1$ 。在多播网络中, 一个探测包从根节点发送, 当到达节点  $k$  时, 探测包将被复制并向  $k$  的所有子节点发送。对于  $\forall u \in d(k)$ , 有:  $p[X_u=0|X_k=0]=1$ ,  $p[X_u=1|X_k=0]=0$ ,  $p[X_u=1|X_k=1]=p_u$ ,  $p[X_u=0|X_k=1]=1-p_u=a_u$ 。若  $T(k)$  的叶节点  $u \in R(k)$  收到探测包, 节点  $k$  必收到探测包, 因此,  $k$  接收到第  $i$  个探测包的状态为  $X_k^{(i)}=\bigvee_{u \in R(k)} X_u^{(i)}$ 。

利用上述丢包模型, 假设源节点  $o$  发送  $n$  个探测包,  $n(u_i)$  表示节点  $u_i$  收到的探测包数,  $n_{u_1, u_2}$  表示节点  $u_1, u_2$  同时收到

**基金项目:** 福建省自然科学基金资助项目(2006J0044)

**作者简介:** 吴文佳(1984-), 女, 硕士研究生, 主研方向: 网络信息处理; 张建中, 教授、博士生导师; 张元鹏, 硕士研究生

**收稿日期:** 2009-06-27 **E-mail:** wuwenjia1984@126.com

的探测包数,同理,  $n_{u_1, u_2, \dots, u_m}$  表示节点  $u_1, u_2, \dots, u_m$  同时收到的探测包数。求链路丢包率有如下推导公式,若  $u_1, u_2, \dots, u_m$  是兄弟节点,则  $u_1$  对应的链路丢包率的估计值  $\hat{a}_{u_1}$  为

$$\hat{a}_{u_1} = 1 - \hat{p}_{u_1} = 1 - \frac{n_{u_1, u_2, \dots, u_m}}{n_{u_2, u_3, \dots, u_m}} \quad (1)$$

### 3 多播网络拓扑推断

#### 3.1 BHC 算法描述

BHC 算法提出可以通过读取探测包的 TTL 值来获取节点  $u$  的跳数值  $u.hop$ 。探测包经过的路由数越多,其跳数值越大,由此引入了节点的层次信息。已知  $o$  为发送节点,  $R$  为接收节点集合,根据跳数值对接收节点集进行分类,从跳数值最大为  $h = \max_{u \in R}(u.hop)$  的节点集  $W_m (m=h)$  开始,从  $W_m$  中选择汉明距离最小的 2 个节点  $u, v$  作为兄弟节点,节点  $u, v$  间的汉明距离表示为

$$H_d(u, v) = \sum_{m=1}^n X_u^{(m)} \oplus X_v^{(m)} \quad (2)$$

其中,  $n$  代表源节发送的探测包数;  $\oplus$  代表异或操作。

根据

$$H_d(u, v')(1 - \varepsilon_{BHC}) < H_d(u, v) \quad (3)$$

判断  $v'$  是否为  $u, v$  的兄弟节点,用  $r$  表示它们的父节点,且将  $r$  加入跳数值为  $r.hop = m-1$  的节点集。令  $m = m-1$ ,然后重复上述在节点集  $W_m$  中组织兄弟节点的过程。根据节点的层次信息自底向上组织兄弟节点,直到组织到根节点。

在多数情况下, BHC 算法推断出的拓扑能够较好地收敛于真实拓扑。但是当网络中某些链路丢包较为严重时, BHC 算法的准确度会明显下降。多播网络拓扑示例如图 1 所示。其中, 6, 7, 8 为兄弟节点; 5 为其非兄弟节点。若  $1 \rightarrow 5$  对应的路径与  $1 \rightarrow 6$  对应的路径丢包情况相似,而 8 对应的链路丢包特别严重时,兄弟节点间的汉明距离  $H_d(u_6, u_8)$  就很可能比非兄弟节点间的汉明距离  $H_d(u_5, u_6)$  大,此时 BHC 方法失效。

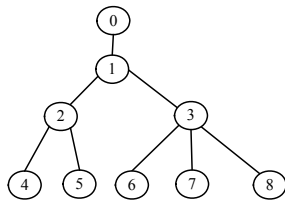


图 1 多播网络拓扑示例

#### 3.2 改进的算法 HGLT

为了克服 BHC 算法存在的问题,本文提出改进的 HGLT 算法,它采用公式

$$\hat{n}(a(U \cup \{u'\}))(1 - \varepsilon) < \hat{n}(a(U)) \quad (4)$$

代替了 BHC 算法中的式(3),并且根据链路丢包率的估计值动态地调整拓扑推断的判决门限值  $\varepsilon$ 。

如图 1 所示,首先从跳数值最大的节点集  $\{4, 5, 6, 7, 8\}$  开始,根据最小汉明距离选取 6, 7 为兄弟节点,节点集  $U = \{5, 6, 7\}$  最近共同祖先节点  $a(U)$  为节点 1,其接收探测包数的估计值为

$$\hat{n}(a(U)) = \hat{n}(1) = \sum_{i=1}^n X_{a(U)}^{(i)} = \sum_{i=1}^n (\bigvee_{u \in U} X_u^{(i)})$$

而节点集  $U' = \{6, 7, 8\}$  最近共同祖先节点  $a(U')$  为节点 3,其接收探测包数的估计值为

$$\hat{n}(a(U')) = \hat{n}(3) = \sum_{i=1}^n X_{a(U')}^{(i)} = \sum_{i=1}^n (\bigvee_{u \in U'} X_u^{(i)})$$

显然  $\hat{n}(1) > \hat{n}(3)$ ,即非兄弟节点的最近共同祖先节点接收探测包数的估计值要大于兄弟节点。因此,通过比较节点集的最近共同祖先节点接收探测包数来区分兄弟节点是简单有效的。

在 HGLT 算法中,判决门限值  $\varepsilon$  根据链路丢包率的估计值动态变化。在上述例子中,由于  $n(1)(1 - a_3) = n(3)$ ,当  $\varepsilon < \hat{a}_3$  时,  $\hat{n}(1)(1 - \varepsilon) > \hat{n}(3)$ ,不满足式(4),可推断出 5 不是 6, 7 的兄弟节点,所以当选取  $\varepsilon$  略小于拓扑结构中的任意一条链路的丢包率时,可用式(4)判断  $u'$  是否为节点集  $U$  的兄弟节点。本算法初始化时在区间  $[0, 0.06]$  内选择一个判决门限初值  $\varepsilon_0$ ,此时  $\varepsilon_0$  不一定小于所有内部链路的丢包率,然后在每一次判断出兄弟节点的同时,由式(1)估算出该节点对应的链路丢包率,再根据链路丢包率的估计值动态地调整  $\varepsilon$  值:若  $\varepsilon$  大于该链路丢包率则调整  $\varepsilon$  与其相等,否则保持不变。上述过程使  $\varepsilon$  一直处于较优的取值区间,从而提高了拓扑推断的准确率。

HGLT 算法伪代码如下:

```

Input The set of receivers R, number of probe packets n, observed
sequences at receivers  $(X_u^{(i)})_{u \in R}^{i=1, \dots, n}$ ;
 $R' = R, V' = \Phi, L' = \Phi, h = \max_{u \in R}(u.hop)$ ;
 $W_m = \Phi (m=1, \dots, h)$ , initialize  $\varepsilon = \varepsilon_0$ ;
foreach  $u \in R$ , do  $W_{u.hop} = W_{u.hop} \cup \{u\}$ ; enddo
 $m = h$ 
while  $m > 1$  do
  while  $W_m \neq \Phi$  do
    Select  $U = \{u, v\} \subset W_m$  that minimizes  $H_d(u, v)$ 
    for  $i = 1, \dots, n$  do  $X_{a(U)}^{(i)} = \bigvee_{u \in U} X_u^{(i)}$ ; enddo
    if there exists  $u' \in W_m \setminus U$ 
       $\hat{n}(a(U \cup \{u'\}))(1 - \varepsilon) < \hat{n}(a(U))$ , then do
         $U = U \cup \{u'\}$ 
      endif
    endif
    for  $i = 1, \dots, n$  do  $X_{a(U)}^{(i)} = \bigvee_{u \in U} X_u^{(i)}$ ; enddo
     $r = a(U); r.hop = m - 1$ ;
     $V' = V' \cup U; W_m = W_m \setminus U$ ;
     $W_{m-1} = W_{m-1} \cup \{r\}$ 
    foreach  $u \in U$ , do
       $L' = L' \cup \{r, u\}$ ;
       $\hat{a}_{u_1} = 1 - \hat{p}_{u_1} = 1 - \frac{n_{u_1, u_2, \dots, u_m}}{n_{u_2, u_3, \dots, u_m}}$ 
    enddo
    if  $\hat{a}_{u_1} < \varepsilon$ , do  $\varepsilon = \hat{a}_{u_1}$ ; endif
  enddo
   $m = m - 1$ 
enddo
 $V' = V' \cup \{o\}, L' = L' \cup \{o, r\}$ ;
Output Inferred topology and loss rate  $T' = \{V', L', \hat{a}\}$ 

```

#### 4 仿真实验与性能分析

为了验证算法的有效性,通过 NS2 进行仿真。待推测拓扑如图 2 所示。根据实际网络拓扑分布特点,内部链路的带宽和延迟大于边缘链路。边缘链路参数:带宽 1 Mb/s,延迟 10 ms;内部链路参数:带宽 1.5 Mb/s,延迟 20 ms。每个路由传输的缓冲区大小为 20 个数据包,支持多播路由,采用 Droptail 丢包策略。背景流量以 TCP 为主同时包含适当的 UDP,UDP 流量采用符合 Pareto 分布的开关型模型;探包产生过程符合 Poisson 分布。根据上面设置的仿真环境,采集

100 次不同时间段的观测数据进行统计。

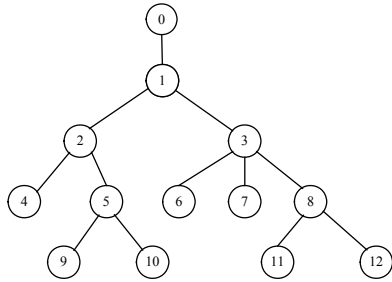


图 2 待推测拓扑

若节点  $u$  的父节点及其子节点都被正确推断出来，则称节点  $u$  推断正确。定义拓扑推断的准确率等于网络中被正确识别出的节点数与总节点数的比值。分别用 BHC 和 HGLT 算法推断网络拓扑，由上节分析可知，判决门限值  $\varepsilon$  的选择应略小于网络中的任一条链路的丢包率。

图 3 为链路丢包率相近时，拓扑推断准确率随门限初值  $\varepsilon_0$  的变化曲线，发送探测包数为 2 000， $\varepsilon_0 \in [0, 0.06]$ 。

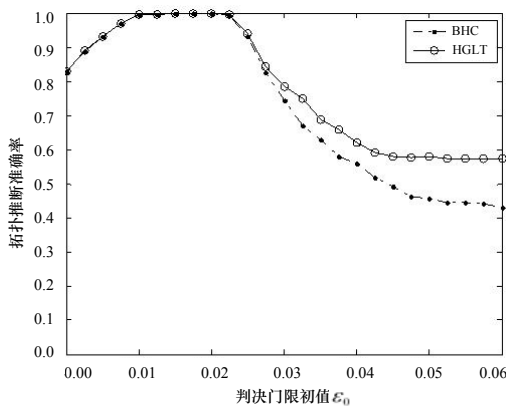


图 3 链路丢包率相近时拓扑推断准确率随门限初值的变化

当网络中各链路丢包率相似时，从图 3 可看出，BHC 算法和 HGLT 算法在门限初值  $\varepsilon_0$  较小时性能几乎一致，但随着  $\varepsilon_0$  的增大，BHC 算法的性能迅速下降，而 HGLT 算法根据链路丢包率的估计值动态调整  $\varepsilon$  值，能够始终保证 60% 左右的正确率。当  $\varepsilon_0 \in [0.01, 0.02]$  时，2 种算法的准确率都可达到 100%。选取  $\varepsilon_0 = 0.015$ ，如图 4 所示，当各链路丢包率相似时，随着发送探测包数的增加，2 种算法性能几乎一致。

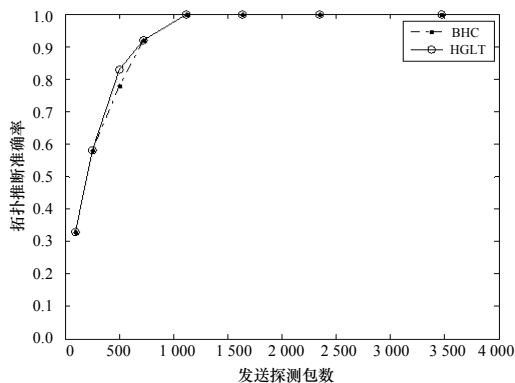


图 4 丢包率相似时拓扑推断准确率随探测包发送数量的变化

为了验证在某些链路丢包较为严重时，HGLT 算法比 BHC 算法性能有明显提升，特别设置链路的带宽为 0.5 Mb/s，

延迟为 10 ms。拓扑推断准确率随门限初值的变化如图 5 所示，发送探测包数为 2 000， $\varepsilon_0 \in [0, 0.06]$ 。

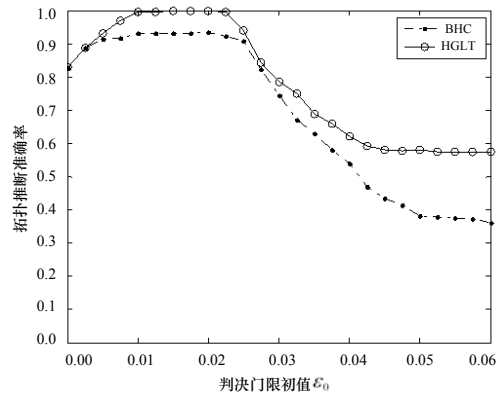


图 5 丢包较为严重时拓扑推断准确率随门限初值的变化

可以看出，无论  $\varepsilon_0$  如何选择，BHC 算法的准确率都无法达到 100%；而 HGLT 在  $\varepsilon_0 \in [0.01, 0.02]$  时，准确率仍可达到 100%，性能较为稳定。选取  $\varepsilon_0 = 0.015$ ，如图 6 所示，在链路丢包较为严重时，BHC 算法最高只能到达 93% 的准确率，即使探测包的发送数量不断增加，拓扑推断准确率也不会上升；而在 HGLT 算法中，当探测包数超过 1 700 时，推断拓扑与真实拓扑完全一致。

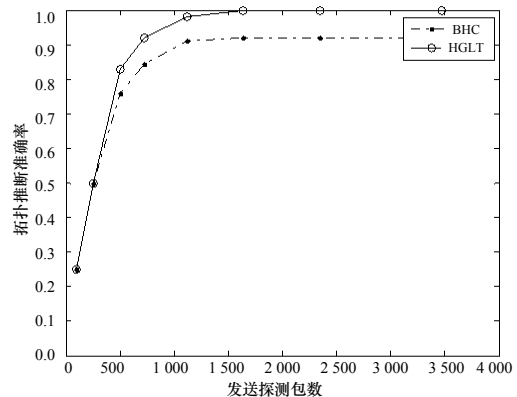


图 6 丢包较为严重时拓扑推断准确率随探测包发送数量的变化

综上所述，HGLT 算法的整体性能明显优于 BHC 算法，能够有效地改善拓扑推断的准确性。

## 5 结束语

为了克服 BHC 算法在某些链路丢包严重时拓扑推断误差较大的缺点，本文提出一种改进算法，根据推算出的链路丢包率动态地调整门限值  $\varepsilon$ ，有效地推断了网络的拓扑结构。NS 仿真结果表明本文提出的改进算法准确率更高，实用性更强。

## 参考文献

- [1] 李勇军, 蔡皖东, 王伟, 等. 基于端到端报文丢失的网络拓扑推断算法研究[J]. 通信学报, 2007, 28(10): 85-91.
- [2] Duffield N G, Horowitz J, Presti F L, et al. Multicast Topology Inference from Measured End-to-end Loss[J]. IEEE Transactions on Information Theory, 2002, 48(1): 26-45.
- [3] Tian Hui, Shen Hong. Multicast-based Inference for Topology and Network: Internal Loss Performance from End-to-end Measure[J]. Computer Communications, 2006, 29(11): 1936-1947.

编辑 顾姣健