基于最大公共路径匹配的拓扑推断算法

姜守达,尹文涛,杨京礼,魏长安
(哈尔滨工业大学自动化测试与控制系,黑龙江哈尔滨 150080)

摘 要: 针对存在节点动态加入和退出的网络,提出了一种基于最大公共路径匹配的拓扑推断算法. 该算法根据背景流量影响对"三明治"包中两个小包进行排序重组,利用重组后的"三明治"包对节点对相似度进行计算,以提高节点对相似度的估计精度;利用TTL跳数信息选择匹配路径,按照公共路径长度匹配搜索新加入节点的插入位置,减少测量过程中所需的探测次数,提高拓扑推断的效率. 仿真结果表明,该算法能提高网络拓扑结构推断的准确性和效率.

关键词: 网络测量; 网络层析成像; 拓扑推测; 最大公共路径匹配
中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2016)09-2189-08
电子学报URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2016.09.025

Topology Inference Based on Maximum Common Path Matching

JIANG Shou-da, YIN Wen-tao, YANG Jing-li, WEI Chang-an
(*Automatic Test and Control Institute, Harbin Institute of Technology, Harbin, Heilongjiang* 150080, *China*)

Abstract: For network with nodes joining and leaving dynamically, a topology inference algorithm based on maximum common path matching is proposed. In this algorithm, in order to improve the estimating precision of similarity metric, two small packets of sandwich probes are rearranged in accordance with cross-traffic effects, and the similarity metric is estimated according to the new rearranged sandwich probes. The new joined nodes are directly added into the existing topology by matching the length of common path. By using the information of TTL hop count to select match path, the efficiency of topology inference is improved. The simulating results show that this algorithm can effectively improve the accuracy and efficiency of topology inference.

Key words: network measurement; network tomography; topology inference; maximum common path matching

1 引言

随着计算机网络规模的不断扩大,网络拓扑信息在网络资源的管理和维护、网络协议的设计,以及网络结构的优化等方面具有越来越重要的意义. 传统网络拓扑测量方法需要网络内部节点之间的协作. 由于许多单位和组织基于安全或商业利益方面的考虑,不愿共享其内部网络信息,使得现有网络系统和设备具有非协作性的特点[1],传统的基于路由器协作的拓扑测量方法的可行性越来越低.

网络层析成像技术将医学上的计算机层析成像思想引入到网络测量中,根据端到端的观测数据采用统计方法来分析和推断网络拓扑和性能参数[2]. 基于网络层析成像技术的拓扑测量方法,可以在无需内部节点协作的条件下推断网络拓扑,克服了传统方法的不足. 文献[3]最早提出基于节点对融合的二叉树拓扑推断算法,文献[4,5]分别提出采用判决门限和双样本t检验对二叉树进行修剪,将节点对融合算法扩展到一般树拓扑模型. 文献[6~8]提出基于极大似然估计的网络拓扑推断算法. 上述算法主要针对网络节点集合相对稳定的情况,而在一些实际应用中,如基于P2P的应用,与一个源节点通信的目标节点是随时间不断变化的,即存在节点动态加入和退出的情况. 当新的网络节点加入后,该类算法需要重新推断网络拓扑,因而效率较低. 针对该问题,文献[9]首先提出一种序列化的拓扑推断算法(Sequential Topology Inference Algorithm, STI),当节点加入或退出网络后,只需在原有网络拓扑的基础上,推测更新后的拓扑,有效地提高了拓扑推断效率. 对于新加入节点,STI算法从根节点开始,自顶向下逐层搜索新加入节点在原有网络拓扑中的插入位

收稿日期:2014-12-02;修回日期:2015-01-23;责任编辑:梅志强
基金项目:国家自然科学基金(No.61501135)

置.由于该算法对于新加入节点都需要从根节点开始逐层搜索,影响了拓扑推断的效率.文献[10]提出 TSP(Traceroute with Sandwich Probe)算法,将 Traceroute 网络测量工具的测量原理引入到基于"三明治"探测包的探测方法中,通过设置"三明治"包中间大数据包的 TTL 值,获取从源节点到目的节点指定跳数的路径的加性特征量.该方法能获取更多的网络内部信息,因而具有更高的测量精度,但该算法通过穷举搜索新加入节点在网络中的插入位置,故效率较低.

针对上述问题,为提高网络层析成像框架下的网络拓扑推断准确性和效率,本文提出一种基于最大公共路径匹配的拓扑推断算法(Maximum Common Path Matching,MCPM).首先,对基于"三明治"包时延差的加性特征量的原理和背景流量对探测包时延的影响进行分析,提出了一种基于探测包重组的节点对相似度估计方法,对节点对相似度的估计精度进行提升,以提高拓扑推断的准确性.在此基础上,分析了节点对相似度与最近公共祖先节点位置的关系,提出了一种直接按公共路径长度进行匹配的节点插入位置搜索方法,提高拓扑推断的效率.

2 网络模型与加性特征量

2.1 网络模型

与现有大多数基于网络层析成像技术的拓扑推断算法的文献[3~12]类似,本文仅考虑树状逻辑拓扑结构,一般网络拓扑可以通过多个树状拓扑融合得到[13,14].用 $T=(V,L)$ 表示树状逻辑网络拓扑模型,其中 V 为网络节点集合,代表网络中的路由器和主机,L 为连接节点的链路集合.节点 $O\in V$ 为 T 的根节点.从根节点 O 到节点 i 之间的路径用 p_i 表示.节点集合 $R\subset V$(无子节点的节点)表示所有的叶节点,即探测报文接收节点,$|R|$ 为叶节点个数.每一个非叶节点 k 都至少有一个子节点,用 $c(k)$ 表示节点 k 的子节点集合.每一个非根节点 k 有且仅有一个父节点,用 $f(k)$ 表示.链路 $(f(k),k)\in L$ 记为 e_k.定义 $f^1=f$ 且 $f^n(k)=f(f^{n-1}(k))$,其中 n 为正整数.用集合 $a(k)=\{i\in V|\ \exists n>0, i=f^n(k)\}$ 表示节点 k 的祖先节点.对任意两个叶节点 i 和 j,用 $a(i,j)$ 表示他们的最近公共祖先节点.用集合 $U=V\backslash\{O\}$ 表示非根节点集合.网络内部节点集合,即非叶节点非根节点集合,用 $I=U\backslash R$ 表示.以节点 k 为根节点,叶节点集合 D 为目的节点的子树用 $T(k,D)$ 表示.

2.2 基于"三明治"包时延差的加性特征量

定义 d 为树 $T=(V,L)$ 的加性特征量[9],当 d 满足:

(1) $0<d(e)<\infty,\ \forall e\in L$

(2) $d(i,j)=\sum_{e\in P_{a(i,j)}} d(e),\ \forall i,j\in R$

其中,$d(e)$ 为链路的长度,$d(i,j)$ 为叶节点对 (i,j) 的相似度,用该节点对的公共路径的长度表示.定义加性特征量 d 后,从根节点到叶节点 i 的路径 p_i 的长度用 $\rho(i)$ 表示.

现有算法中,常用的加性特征量有,基于丢包率的加性特征量[4]、基于时延协方差的加性特征[11]和基于"三明治"包时延差[6,8]的加性特征量.由于采用基于"三明治"包时延差的加性特征量的探测方案不需要时钟同步,且加性特征量由探测包排队时延自引入,本文采用该方案,其基本原理如图 1 所示.每个"三明治"包由三个探测包组成,其中第 1 个探测包 A_1 和第 3 个探测包 A_2 长度相同,第 2 个探测包 B 长度远大于第 1 个和第 3 个探测包.每次探测过程中,探测包 A_1 和 A_2 发往相同的目的地址(图中节点 j),探测包 B 发往另一目的地址(图中节点 i).三个探测包先经过一段共享路径后,到达节点 i 和 j 的最近公共祖先节点 k,然后分别发往各自目的节点.

假设网络中无背景流,在公共路径上,由于探测包 B 较大,发送时间较长,导致探测包 A_2 的排队时延较长.在每条共享链路上,探测包 A_1 和 A_2 之间的时间间隔都会增加.在非公共路径上,由于探测包 B 发往另一目的节点,不再影响探测包 A_2 的排队时延,故探测包 A_1 和 A_2 之间的时间间隔保持不变.设探测包 A_1 和 A_2 的发送时间间隔为 t_s,在节点 j 的接收时间间隔为 t_r,则其时延差为 $\Delta t=t_r-t_s$.对叶节点 i 和 j,其公共路径为从根节点到其最近公共祖先节点 $a(i,j)$ 之间的路径.定义叶节点对 (i,j) 的相似度 $d(i,j)$ 为探测包 A_1 和 A_2 的时延差,则 $d(e)$ 为探测包 A_1 和 A_2 在经过链路 e 时产生的时延差.叶节点对 (i,j) 的共享路径越长,即"三明治"包所经历的共享路径越长,则探测包 A_1 和 A_2 的时延差越大,叶节点对 (i,j) 的相似度 $d(i,j)$ 越大.

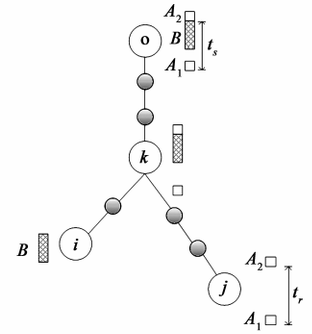

图1 "三明治"包测量示例

3 基于最大公共路径匹配的拓扑推断算法

3.1 节点对相似度估计

文献[6,8]将背景流量对测量结果的影响视为零

均值的随机过程,用探测包 A_1 和 A_2 的时延差的观测样本均值作为节点对相似度的估计值. 文献[12]通过选取受背景流量影响较小的探测包计算时延差观测样本均值作为节点对相似度的估计值. 本文通过分析背景流量对探测包的影响,提出一种基于探测包重组的节点对相似度估计方法,以提高节点对相似度的估计精度.

探测包在每条链路上的经历的时延由处理时延、传输时延、传播时延和排队时延四部分组成,其中前三部分时延主要由路由特征和探测包长度决定. 探测包 A_1 和 A_2 的长度相同,在经过同一链路时,经历的传输时延、传播时延和处理时延近似相等,故在该链路上引入的时延差为两个探测包的排队时延差.

对于图 1 中从根节点发送到叶节点对 i,j 的"三明治"探测包,设其公共路径由链路 $e_1,e_2,\cdots,e_k$ 组成. 设链路 e_i 的带宽为 B_{e_i},探测包 A_1、A_2 和 B 的长度分别为 L_{A_1}、L_{A_2}和 L_B. 当网络中无背景流时,探测包 A_1 的在链路 $e_l(1\leqslant l\leqslant k)$ 上的排队时延 $q_{A_1}^l$ 为 0. 在根链路 e_1 上,探测包 A_2 由于紧邻探测包 B 之后,其排队时延 $q_{A_2}^1$ 等于探测包 B 的传输时延,即

$$q_{A_2}^1=\frac{L_B}{B_{e_1}} \tag{1}$$

在链路 $e_l(2\leqslant l\leqslant k)$ 上,探测包 A_2 的排队时延 $q_{A_2}^l$ 由探测包 B 和探测包 A_2 到达该链路的时间,以及探测包 B 的在链路 e_l 上的传输时延决定. 由于在链路 e_{l-1} 上,探测包 B 离开队列后,探测包 A_2 需要经历 $L_{A_2}/B_{e_{l-1}}$ 的传输时延,故探测包 A_2 到达链路 e_l 的时刻较探测包 B 晚 $L_{A_2}/B_{e_{l-1}}$. 探测包 B 在链路 e_l 的传输时延为 L_B/B_{e_l},故当 $L_B/B_{e_l}>L_{A_2}/B_{e_{l-1}}$ 时,探测包 A_2 在链路 e_l 的排队时延 $q_{A_2}^l$ 为:

$$q_{A_2}^l=\frac{L_B}{B_{e_l}}-\frac{L_{A_2}}{B_{e_{l-1}}} \tag{2}$$

无背景流量时,在节点 j 观测到的探测包 A_1 和 A_2 的时延差 Δt_0 为公共路径上各链路的时延差之和,即探测包 A_1 和 A_2 的排队时延差之和:

$$\begin{aligned}\Delta t_0 &= \sum_{l=1}^{k}(q_{A_2}^l-q_{A_1}^l)\\ &=\frac{L_B}{B_{e_1}}-\sum_{l=2}^{k}\left(\frac{L_B}{B_{e_l}}-\frac{L_{A_2}}{B_{e_{l-1}}}\right)\end{aligned} \tag{3}$$

当网络中存在背景流时,由于背景流对探测包 A_1 和 A_2 的影响均为使其排队时延增大,背景流对探测包的影响越大,其排队时延增大的越多. 又由于探测包 A_1 和 A_2 在经过同一链路时,经历的传输时延、传播时延和处理时延近似相等,故可以根据探测包 A_1 和 A_2 在公共路径上经历的总的时延大小,判断其受背景流影响的程度大小.

从根节点向叶节点对 i,j 发送 N 个"三明治"包,设根节点和节点 j 分别记录的第 n 个"三明治"中探测包 A_1(记为 $A_1(n)$)发送和接受时刻分别为 $t_{A_1}^s(n)$ 和 $t_{A_1}^r(n)$. 设节点 j 相对于根节点的时钟偏差为 τ_j,则探测包 $A_1(n)$ 的时延 $T_{A_1}(n)$ 为:

$$T_{A_1}(n)=t_{A_1}^r(n)-t_{A_1}^s(n)+\tau_j \tag{4}$$

在无时钟同步的条件下,时钟偏差 τ_j 是未知的,故无法求出探测包 $A_1(n)$ 的时延值,用 $T_{A_1}'(n)$ 表示探测包 $A_1(n)$ 的时延相对值,则

$$T_{A_1}'(n)=T_{A_1}(n)-\tau_j=t_{A_1}^r(n)-t_{A_1}^s(n) \tag{5}$$

$T_{A_1}'(n)$ 可以用来表示探测包 $A_1(n)$ 时延值的相对大小,$T_{A_1}'(n)$ 值越大,即探测包 $A_1(n)$ 经历的时延越大,即受背景流量影响越大. 用式(5)计算出 N 个"三明治"包的 $T_{A_1}'(n)$ 值:

$$T_{A_1}'(1),T_{A_1}'(2),\cdots,T_{A_1}'(N)$$

即可得到 N 个"三明治"包中探测包 A_1 经历的时延受背景流量影响的相对大小.

设根节点和节点 j 分别记录的第 n 个"三明治"中探测包 A_2(记为 $A_2(n)$)发送和接受时刻分别为 $t_{A_2}^s(n)$ 和 $t_{A_2}^r(n)$,用 $T_{A_2}(n)$ 和 $T_{A_2}'(n)$ 分别表示探测包 $A_2(n)$ 的时延值和时延相对值,有

$$T_{A_2}'(n)=T_{A_2}(n)-\tau_j=t_{A_2}^r(n)-t_{A_2}^s(n) \tag{6}$$

同理,时延相对值 $T_{A_2}'(n)$ 越大,即探测包 $A_2(n)$ 经历的时延越大,受背景流量影响越大.

用式(5)和(6)分别计算 N 个"三明治"包的时延相对值,然后按时延相对值 $T_{A_1}'(n)$ 和 $T_{A_2}'(n)$ 从小到大的顺序,分别对 N 个"三明治"包中的探测包 A_1 和 A_2 进行排序,得到$\{A_1'(1),A_1'(2),\cdots,A_1'(N)\}$ 和$\{A_2'(1),A_2'(2),\cdots,A_2'(N)\}$.

根据上面的分析,"三明治"包中探测包 A_1 的作用仅为提供时间参考(使探测过程不需要时钟同步),故可将 N 个"三明治"包的探测包 A_1 和 A_2 视为独立的部分,按照重新排列后的顺序进行重组,得到:

$$\{(A_1'(1),A_2'(1)),(A_1'(2),A_2'(2)),\cdots,(A_1'(N),A_2'(N))\}$$

重组后得到的新"三明治"包按照受背景流干扰程度大小顺序排列,其中$(A_1'(1),A_2'(1))$受背景流干扰最小,$(A_1'(N),A_2'(N))$受背景流干扰最大. 由于受背景流干扰较大的探测包,对节点对相似度估计误差影响较大,为提高节点对相似度估计的精度,本文仅取前半部分受背景流干扰较小的重组后的"三明治"包,对其时延差按受背景流干扰程度不同进行加权平均作为节点对(i,j)相似度的估计值 $\hat{d}(i,j)$:

$$\hat{d}(i,j)=\sum_{n=1}^{[N/2]}\lambda(n)(T_{A_2}'(n)-T_{A_1}'(n)) \tag{7}$$

其中,$[N/2]$表示不大于 $N/2$ 的整数,$T_{A_1}'(n)$ 和 $T_{A_2}'(n)$

分别为$(A_1'(n),A_2'(n))$的相对时延值,$\lambda(n)$为第n个新"三明治"包时延差的加权系数,根据上述分析,$\lambda(n)$取值为:

$$\lambda(n)=\frac{[N/2]-n+1}{\sum_{i=1}^{[N/2]}([N/2]-i+1)}$$

当向每个节点对发送的"三明治"包数目较多时,探测包发送时间较长,时钟漂移会导致节点对相似度估计精度降低. 从本文仿真实验结果来看,本文算法仅需向每个节点对发送少量探测包即可达到较高精度,当发送探测包较少时,探测包发送时间较短,故时钟漂移对节点对相似度估计精度影响较小. 基于"三明治"探测包重组的节点对相似度估计过程详见算法 1.

算法 1 基于"三明治"探测包重组的节点对相似度估计算法

输入 N个"三明治"包的探测包A_1、A_2发送和接受时刻$\{t_{A_1}^s(n)\}$、$\{t_{A_1}^r(n)\}$、$\{t_{A_2}^s(n)\}$和$\{t_{A_2}^r(n)\}$,$1\leqslant n\leqslant N$

步骤 1 用式(5)和(6)分别计算N个"三明治"包的探测包A_1、A_2的时延相对值$T_{A_1}'(n)$和$T_{A_2}'(n)$;

步骤 2 按时延相对值$T_{A_1}'(n)$和$T_{A_2}'(n)$从小到大的顺序,分别对N个"三明治"包中的探测包A_1和A_2进行排序,得到$\{A_1'(1),A_1'(2),\cdots,A_1'(N)\}$和$\{A_2'(1),A_2'(2),\cdots,A_2'(N)\}$;

步骤 3 对排序后的探测包A_1和A_2进行重组,得到N个新的"三明治"包:

$$\{(A_1'(1),A_2'(1)),(A_1'(2),A_2'(2)),\cdots,(A_1'(N),A_2'(N))\}$$

步骤 4 取前$[N/2]$个新"三明治"包,用式(7)计算时延差加权平均值作为节点对(i,j)相似度的估计值$\hat{d}(i,j)$;

输出 节点对(i,j)相似度的估计值$\hat{d}(i,j)$

3.2 拓扑推断

由于"三明治"包在公共路径上每条链路的时延差均大于0,故基于"三明治"包时延差的加性特征量在每条链路的取值均大于0,即$d(e)>0,\forall e\in L$,结合 2.2 节加性特征量的定义,容易得到下面的定理:

定理 1 节点对(i,j)相似度$d(i,j)$与其最近公共祖先节点$a(i,j)$的位置有如下关系:

(1)当$\rho(k)<d(i,j)<\rho(l),k,l\in p_i$时,$a(i,j)$在节点$k$和节点$l$之间的路径上;

(2)当$d(i,j)=\rho(k),k\in p_i$时,$a(i,j)$为节点k.

假设当前拓扑为$T=(V,L)$,新加入节点为节点j. 本文通过节点j与当前拓扑叶节点进行公共路径长度匹配,搜索节点j的插入位置.

如图 2 所示,新加入节点j与叶节点为r进行公共路径长度匹配,有以下几种可能情况:

(1)当在路径p_r上存在节点i,使得节点对(r,j)的相似度$d(r,j)$满足

$$\rho(f(i))<d(r,j)<\rho(i) \tag{8}$$

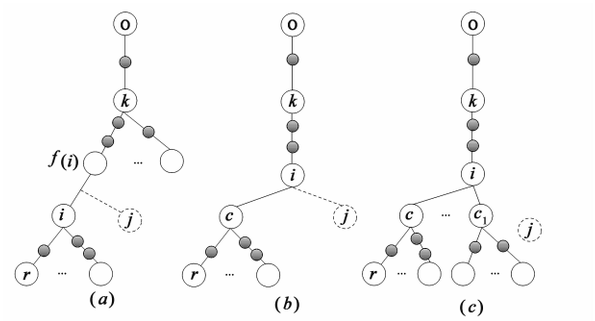

图2 公共路径长度匹配

根据定理 1,可得节点对(r,j)的最近公共祖先节点$a(r,j)$在节点$f(i)$与节点i之间的路径上,即链路$(f(i),i)$上. 故从根节点到节点对(r,j)的路径的分叉节点在链路$(f(i),i)$上,即插入位置为链路$(f(i),i)$上,如图 2(a)所示.

(2)当在路径p_r上存在节点i,使得$d(r,j)$满足

$$d(r,j)=\rho(i) \tag{9}$$

根据定理 1,可得节点对(r,j)的最近公共祖先节点$a(r,j)$为节点i,即节点对(r,j)的公共路径为p_i,则从根节点到节点对(r,j)的分叉节点为节点i,可以确定节点j为节点i的子孙节点,节点j的插入位置在节点i上(如图 2(b)所示),或节点i的除去节点c所在分支的子孙节点或子孙链路上(如图 2(c)所示).

利用探测包的 TTL(Time-To-Live)域,可以获取从根节点到叶节点经过的路由跳数,即从根节点到叶节点包含的链路个数. 由于 TTL 跳数信息较大的叶节点与新加入节点具有更大的公共路径长度的可能性较大,故每次匹配选取目的节点中 TTL 跳数信息最大的节点,能提高算法效率. 节点j加入网络时,本文根据以上分析设计了如算法 2 所示的拓扑更新算法,其中判定阈值$\delta=1/2\min_{e\in L}d(e)$.

算法 2 拓扑更新算法 AddLeafNode(T,j,δ)

输入 当前拓扑$T=(V,L)$,新加入节点j,判定阈值δ

初始化 $k=O,D=R$

步骤 1 选择D中 TTL 跳数信息最大的叶节点r,向节点对估计节点对(r,j)发送"三明治"包,然后用算法 1 估计节点对相似度值$\hat{d}(r,j)$;

步骤 2 If 路径P_r上存在节点i,使得$\hat{\rho}(f(i))+\delta\leqslant\hat{d}(r,j)\leqslant\hat{\rho}(i)-\delta$,then

创建节点p,p为$f(i)$的子节点,节点i和j的父节点,更新节点集合V和链路集合L:$V=V\cup\{p,j\}$,$L=L\setminus\{(f(i),i)\}\cup\{(f(i),p),(p,i),(p,j)\}$,记录$\hat{\rho}(p)=\hat{d}(r,j)$;

步骤 3 *If* 路径P_r上存在节点i,使得$|\hat{\rho}(i)-\hat{d}(r,j)|$值最小,且$|\hat{\rho}(i)-\hat{d}(r,j)|<\delta$时,设节点$c$为路径$P_r$上节点$i$的子节点,then

①If $i==k$,then $D'=D\setminus R(c)$;Else $D'=R(i)\setminus R(c)$;

②If $|D'|=0$,then 节点 j 为节点 i 的子节点,更新节点集合 V、链路集合 L 和 $\hat{d}(i)$:$V=V\cup\{j\}$,$L=L\cup\{(i,j)\}$,$\hat{\rho}(i)=1/2(\hat{d}(r,j)+\hat{\rho}(i))$;

Else 节点 j 为节点 i 的子孙节点,其插入位置在子树 $T(i,D')$ 上. 更新 k 和 D:$k=i$,$D=D'$,跳转到步骤 1;

输出　加入节点 j 后新的拓扑 $T=(V,L)$

注:$\hat{d}(r,j)$ 和 $\hat{\rho}(p)$ 分别为 $d(r,j)$ 和 $\rho(p)$ 估计值.

定理 2　算法 2 将节点 j 插入到拓扑中正确位置的充分条件为节点对相似度的估计误差小于 $\delta/2$,即

$$|\hat{d}(i,j)-d(i,j)|<\frac{\delta}{2},\ \forall i\in R,j\in R \tag{10}$$

证明　当算法 2 每次进行最大公共路径匹配都能找出正确的分叉节点时,能将节点 j 插入到拓扑中正确位置.

(1)当匹配路径 P_r 上存在节点 i,使得节点对(r,j)相似度估计值 $\hat{d}(r,j)$ 满足

$$\hat{\rho}(f(i))+\delta\leqslant\hat{d}(r,j)\leqslant\hat{\rho}(i)-\delta \tag{11}$$

由式(11)右半部分可得

$$\hat{d}(r,j)+\frac{\delta}{2}\leqslant\hat{\rho}(i)-\frac{\delta}{2} \tag{12}$$

由节点对相似度估计误差小于 $\delta/2$,可得

$$|\hat{d}(r,j)-d(r,j)|<\frac{\delta}{2},|\hat{\rho}(i)-\rho(i)|<\frac{\delta}{2}$$

进一步可得到

$$d(r,j)<\hat{d}(r,j)+\frac{\delta}{2} \tag{13}$$

$$\hat{\rho}(i)-\frac{\delta}{2}<\rho(i) \tag{14}$$

由式(12)、(13)和(14)可到

$$d(r,j)<\rho(i) \tag{15}$$

同理,由式(11)左半部分可得

$$\rho(f(i))<d(r,j) \tag{16}$$

由式(15)和(16)可得式(8)成立,即从根节点到节点对(r,j)的路径的分叉节点在链路$(f(i),i)$上,即插入位置为链路$(f(i),i)$上,进入算法步骤 2 执行,如图 2(a)所示.

(2)当匹配路径 P_r 上存在节点 i 使得 $|\hat{\rho}(i)-\hat{d}(r,j)|$ 值最小,且 $|\hat{\rho}(i)-\hat{d}(r,j)|<\delta$ 时

$$\begin{aligned}&|\rho(i)-d(r,j)|\\&=|(\hat{\rho}(i)-\hat{d}(r,j))+(\rho(i)-\hat{\rho}(i))-(d(r,j)-\hat{d}(r,j))|\\&\leqslant|(\hat{\rho}(i)-\hat{d}(r,j))|+|(\rho(i)-\hat{\rho}(i))|+|(d(r,j)-\hat{d}(r,j))|\\&<\delta+\frac{\delta}{2}+\frac{\delta}{2}\\&=2\delta\end{aligned}$$

由 $\delta=1/2\min_{e\in L}d(e)$,可得 $2\delta\leqslant d(e)$,$\forall e\in L$,代入上式即得

$$|\rho(i)-d(r,j)|<d(e),\ \forall e\in L \tag{17}$$

即节点对(r,j)的公共路径长度与路径 p_i 长度的差值小于最小路径的长度,故 $d(r,j)=\rho(i)$. 即式(9)成立,节点对(r,j)的最大公共路径为 p_i,则从根节点到节点对(r,j)的分叉节点为节点 i,进入算法步骤 3 执行. 设节点 c 为路径 P_r 上节点 i 的子节点. 当节点 c 为当前搜索子树 $T(k,D)$ 上唯一子节点时,新插入节点 j 为节点 i 的子节点,如图 2(b)所示;当 $T(k,D)$ 上节点 i 存在多个子节点时,节点 j 的插入位置在节点 i 的除去节点 c 所在分支的子孙节点或子孙链路上,如图 2(c)所示.

综上可得,算法 2 每次进行最大公共路径匹配都能找出正确的分叉节点,能将节点 j 插入到拓扑中正确位置. 故式(10)为算法 2 将节点 j 插入到拓扑中正确位置的充分条件.

对于给定源节点和目的节点的网络,可以先在目的节点中选取 TTL 跳数最大的 2 个叶节点构建一个 4 个节点 3 条链路的简单二叉树,然后运用算法 2 将目的节点逐个插入到该二叉树中,具体详见算法 3. 当节点离开网络时,直接将相关节点和链路删除即可.

算法 3　拓扑推断算法

输入　源节点 O,目的节点集合 R,判断阈值 δ

步骤 1　按 TTL 跳数信息从大到小的顺序对叶节点进行排序,得到:$r_1,r_2,\cdots r_n$;

步骤 2　选取目的节点 r_1、r_2 与源节点构造一个节点数目为 4 的简单初始二叉树 $T=(V,L)$:$V=\{O,v_1,r_1,r_2\}$,$L=\{(O,v_1),(v_1,r_1),(v_1,r_2)\}$;

步骤 3　估计节点对(r_1,r_2)相似度值 $\hat{d}(r_1,r_2)$,记录 $\hat{\rho}(v_1)=\hat{d}(r_1,r_2)$;

步骤 4　for $i=1,2,\cdots,n$:

AddLeafNode(T,r_i,δ);

输出　拓扑 $T=(V,L)$

4　仿真

为对 MCPM 算法进行综合评价,本文采用 MATLAB 模型仿真和 NS 2 仿真,分别对该算法拓扑推断效率和准确性进行评估,并与目前算法中性能较好的 STI 算法[9]和 TSP 算法[10]进行比较.

4.1　MATLAB 模型仿真

为对算法拓扑推断效率进行评价,本文采用 BRITE 拓扑生成工具生成一系列节点个数分别为 100,200,…,1000 的网络拓扑,每种规模的拓扑均生成 100 个. 网络拓扑模型采用 Waxman 和 BA 两种经典模型,模型参数选择为:$\alpha=0.15$,$\beta=0.2$,$m=2$,MaxBw = 1024,MinBw = 2. 选取网络拓扑中节点度数较小的节点作为端节点,在端节点中任选一节点作为源节点,其余全部端节点作为目的节点,即可构成逻辑树型拓扑. 假设加

性特征量测量过程中无噪声干扰,即加性特征量的测量误差为0. 为每个链路随机分配一个取值服从0.1至0.4上均匀分布的加性特征量. 每次探测的节点对相似度为公共路径上各链路加性特征量之和.

测量一个节点对相似度值需要向节点对发送一组指定数目的"三明治"包. 将向一个叶节点对发送一组指定数目的探测包测量该节点对相似度视为一次探测,则推断拓扑所需测量的节点对相似度个数即为所需探测次数. 本文采用推断网络拓扑所需探测次数作为算法效率的评价指标. 对每个拓扑,先选取2个目的节点与源节点构成一个二叉树拓扑,然后分别采用MCPM、STI和TSP三种算法将其余目的节点逐个插入到二叉树拓扑中,比较各种算法所需的探测次数.

图3和图4分别给出了Waxman和BA拓扑模型下,三种算法所需探测次数随网络拓扑节点个数的变化情况,图中数据为仿真100次取平均的结果. 从图中可以看出,三种算法所需探测次数都随着网络节点个数的增加而增加,其中TSP算法增加的最快,本文算法增加的最慢. 在Waxman拓扑模型下,本文算法所需探测次数较STI算法减少51.35%至55.03%,较TSP算法减少85.46%至97.11%;在BA拓扑模型下,本文算法所需探测次数较STI算法减少50.41%至55.30%,较TSP算法减少78.56%至94.53%. 在Waxman和BA拓扑模型下,本文算法拓扑推断效率较STI和TSP算法都有明显提升.

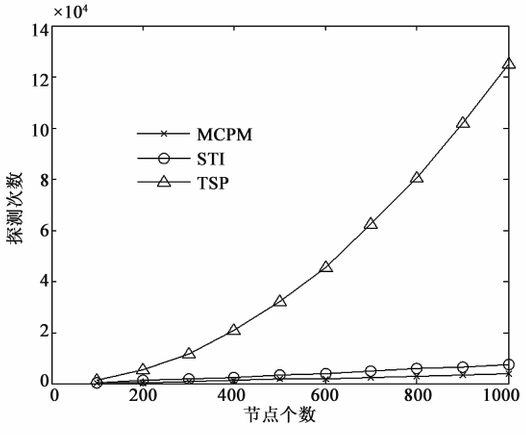

图3 Waxman拓扑模型下所需探测次数随网络节点个数的变化

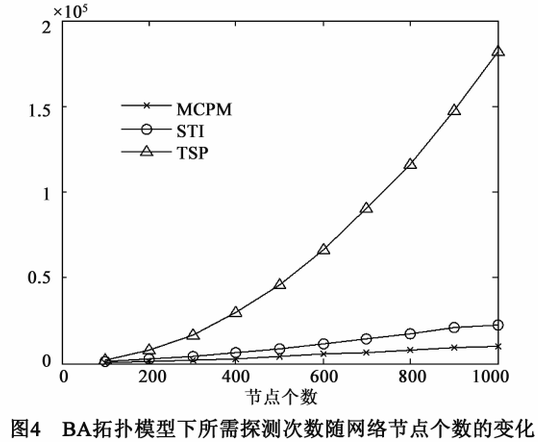

图4 BA拓扑模型下所需探测次数随网络节点个数的变化

4.2 NS 2 仿真

为对算法拓扑推断准确性进行评价,本文采用NS 2网络仿真工具构建如图5所示网络,该网络包含15个节点和14条链路. 所有边缘链路的带宽均为5Mbps,物理传播时延为5ms. 所有内部链路带宽均为2Mbps,物理传播时延为2ms. 所有链路队列长度均为50,排队模型为FIFO(First In First Out),拥塞避免算法采用尾部丢弃(Drop-tail). 探测包为根节点向叶节点对发送的"三明治"包. "三明治"包的大数据包长度为500Byte,小数据包长度为10Byte. 背景流量为服从Pareto分布的On/Off模型的UDP流和TCP流. 所有的UDP流和TCP流的发送节点和接收节点在网络节点中随机选择. UDP流和TCP流的速率分别为0.01Mbps和0.02Mbps. 为在不同网络负载情况下对算法性能进行比较,本文进行2组仿真实验,第1组实验网络负载较轻,背景UDP流和TCP流数目个数为100和200,第2组实验网络负载较重,背景UDP流和TCP流数目分别为150和300.

向每个叶节点对发送的"三明治"包数目分别取50,100,…,300,对每个给定的探测包数目均进行100次仿真. 每次仿真,先选取2个叶节点与根节点构成一个二叉树拓扑,然后分别采用MCPM、STI和TSP三种算法将剩余叶节点逐个插入到二叉树拓扑中,比较最终生成的拓扑准确率.

图6给出了第1组实验的结果,三种算法拓扑推断准确率都随"三明治"探测包个数的增加而增加. 本文提出的MCPM算法准确性最高,且在探测包较少时,这种优势更加明显. 当发送到每个节点对的探测包数目

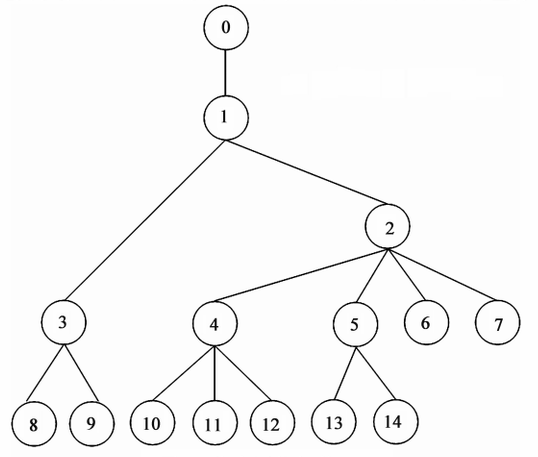

图5 仿真网络拓扑

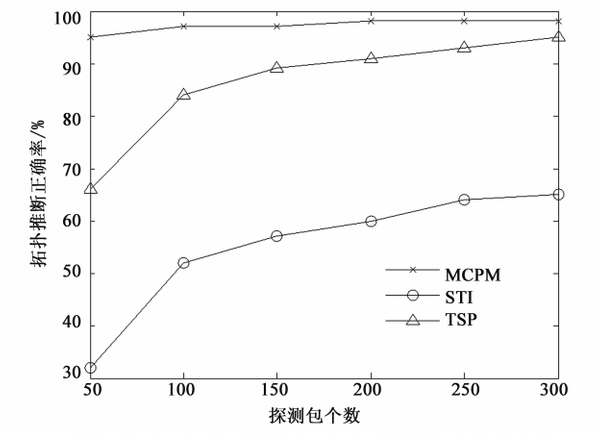

图6 第1组实验拓扑推断正确率随探测包个数的变化

为300时,MCPM算法拓扑推断准确率为98%,较STI算法(65%)提高约50.77%,较TSP算法(95%)提高约3.16%.当发送到每个节点对的探测包数目为50时,MCPM算法拓扑推断准确率为95%,较STI算法(32%)提高约1.97倍,较TSP算法(66%)提高约43.94%.

图7给出了第2组实验的结果,由于网络负载较重,"三明治"探测包受背景流干扰增大,三种算法准确率较第1组实验都有不同程度的降低.其中STI算法和TSP算法准确率显著降低,而MCPM算法仍能保持较高的准确率,这是由于MCPM算法通过基于探测包重组的节点对相似度估计方法提高节点对相似度估计精度,从而提高拓扑推断准确率.当发送到每个节点对的探测包数目为300时,MCPM算法拓扑推断准确率为93%,较STI算法(28%)提高约2.32倍,较TSP算法(80%)提高约16.25%.当发送到每个节点对的探测包数目为50时,MCPM算法拓扑推断准确率为74%,较STI算法(5%)提高约13.80倍,较TSP算法(34%)提高约1.18倍.

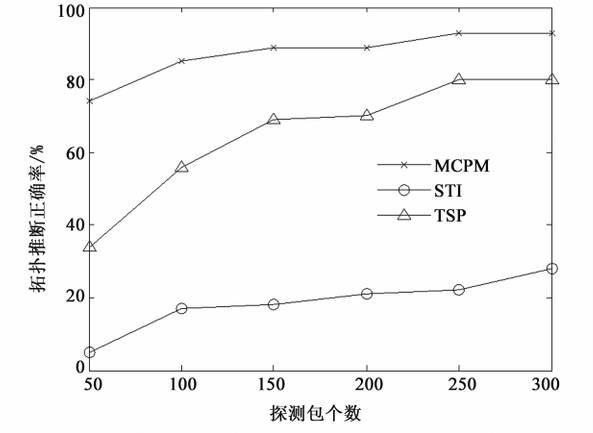

图7 第2组实验拓扑推断正确率随探测包个数的变化

5 结论

本文首先分析了基于"三明治"包时延差的加性特征量的原理和背景流量对探测包时延的影响,提出了一种基于探测包重组的节点对相似度估计方法.然后,在此基础上通过对节点对相似度与最近公共祖先节位置的关系进行分析,提出了一种基于最大公共路径长度匹配的拓扑推断算法.仿真结果表明,本文提出的MCPM算法较STI算法和TSP算法在拓扑推断准确性和效率方面都有明显提高.

参考文献

[1] Donnet B, Friedman T. Internet topology discovery: a survey[J]. IEEE Communications Surveys and Tutorials, 2007, 9(4): 56 – 69.

[2] 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. (in Chinese)

[3] Duffield N G, Horowitz J, Presti F L, et al. Multicast topology inference from end-to-end measurements[A]. ITC Seminar on IP Traffic, Measurement and Modelling[C]. Monterey, CA: ITC, 2000. 1 – 10.

[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 Runsheng, Li Yanbin, Li Xiaotian. Topology inference with network tomography based on t-test[J]. IEEE Communications Letters, 2014, 18(6): 921 – 924.

[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] Castro R M, Coates M J, Nowak R D. Likelihood based hierarchical clustering[J]. IEEE Transactions on Signal Processing, 2004, 52(8): 2308 – 2321.

[8] Shih M F, Hero A O. Hierarchical inference of unicast network topologies based on end-to-end measurements[J]. IEEE Transactions on Signal Processing, 2007, 55(5): 1708 – 1718.

[9] Ni J, Xie 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.

[10] Malekzadeh A, MacGregor M H. Network topology inference from end-to-end unicast measurements[A]. 27th In-

ternational Conference on Advanced Information Networking and Applications Workshops[C]. Barcelona:IEEE, 2013. 1101 – 1106.

[11] Duffield N G, Presti F L. Network tomography from measured end-to-end delay covariance[J]. IEEE/ACM Transactions on Networking, 2004, 12(6): 978 – 992.

[12] Di Pietro A, Ficara D, Giordano S, et al. Noise reduction techniques for network topology discovery[A]. IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications[C]. Athens: IEEE, 2007. 1 – 5.

[13] Coates M, Rabbat M, Nowak R. Merging logical topologies using end-to-end measurements[A]. Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement[C]. New York: ACM, 2003. 192 – 203.

[14] Di Pietro A, Ficara D, Giordano S, et al. Merging spanning trees in tomographic network topology discovery[A]. IEEE International Conference on Communications[C]. Dresden: IEEE, 2009. 1 – 5.

作者简介

姜守达 男,1964 年出生黑龙江伊春,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,数字信号处理等.
E-mail: jsd@hit.edu.cn

尹文涛 男,1983 年生于湖北孝感,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: huayichu@163.com

杨京礼(通信作者) 男,1984 年生于山东日照,哈尔滨工业大学自动化测试与控制系讲师. 主要研究方向为网络测量与网络层析成像技术. E-mail: icehit0615@163.com

魏长安 男,1981 年生于河北承德,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.