

文章编号:1007-5321(2012)02-0126-05

基于四元分组测量的网络拓扑推断算法

赵洪华, 胡谷雨, 倪桂强, 沙俊星

(解放军理工大学 指挥自动化学院, 南京 210007)

摘要: 针对在基于层析成像技术的拓扑推断中测量节点性能时大都需要节点间的时钟同步,并且受到网络负载的影响比较大的问题,提出了一种可以测量时延抖动和丢包率的四元分组序列测量方法,并设计了根据 2 个网络性能参数计算节点相关性的拓扑推断算法. 理论分析与仿真实验证明,该算法受到网络负载的影响较少.

关键词: 网络层析成像; 拓扑推断; 相关性; 四元分组序列

中图分类号: TP393

文献标志码: A

A Network Topology Inference Algorithm Based on 4-tuple Packets Measurement

ZHAO Hong-hua, HU Gu-yu, NI Gui-qiang, SHA Jun-xing

(Institute of Command Automation, People's Liberation Army University of Science and Technology, Nanjing 210007, China)

Abstract: When measuring network performance characteristics in network topology inference, the time synchronization is needed. Meanwhile, the time network topology inference suffers the limits of network load, so there are more limits when applying network topology inference. In order to conquer the limits, a 4-tuple packets measurement method is put forward that could measure the end to end loss and jitter. An algorithm of computing the correlation between nodes is proposed which combining jitter and end to end loss. The 4-tuple packets measurement and the algorithm of computing correlation are suited for the network of lightly loaded, moderately loaded and highly loaded. The measurement method and topology inference algorithm have less limits than other algorithms.

Key words: network tomography; topology inference; correlation; 4-tuple packets

获知网络拓扑是网络管理、网络维护的基础,网络拓扑发现技术是获取网络拓扑结构的重要途径,由于网络的异构性以及网络安全和商业利益的原因,获取网络的拓扑结构变得越来越困难. 基于层析成像技术的拓扑推断是一门较新的获得网络拓扑的技术,该技术不需要网络内部节点的合作,只需通过端到端的测量即可推断网络的拓扑结构,如可以通过端到端的时延或丢包率推断网络内部的拓扑结构^[1-2].

基于网络层析成像的拓扑推断主要分为下面 3

个步骤^[3].

1) 通过端到端的测量获得测量源节点到测量目标节点的端到端性能参数.

2) 根据端到端性能参数计算出测量端节点间的相关性.

3) 根据测量目标节点间的相关性推断网络的拓扑结构.

在基于层析成像技术的拓扑推断中,需要采用特殊的方式. 在单播网络的拓扑推断中,当前比较常用的测量方法主要有紧接分组对方法^[3-5]和“三明

治”分组列车方法^[6]. 紧接分组对测量的网络性能参数为端到端单向时延和丢包率,采用紧接分组对测量端到端时延需要节点间的时钟同步,在实际应用中受到较多的限制;而采用紧接分组对测量丢包率计算节点间的相关性只有在网络丢包较重时才能较好地反映节点共享链路的特性^[3]. 采用“三明治”分组列车测量的性能参数为节点对共享链路的排队时延,只有在网络负载较轻时才能较好地反映节点共享链路的特性^[3].

为了减少拓扑推断中网络负载的限制,根据网络性能的特点,设计了可以测量端到端时延抖动和端到端丢包率的四元分组序列测量方法,并设计了采用 2 种参数计算节点相关性的算法. 采用这种方法不需要节点间的时钟同步,并且可以适应不同负载的情况.

1 四元分组序列测量方法

四元分组序列^[7]的定义如下.

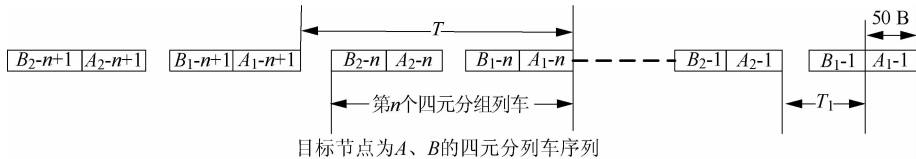


图 1 四元分组序列

通过四元分组序列可以测量的网络端到端性能参数,包括端到端丢包率和端到端的时延抖动. 测量时由测量源节点向所有测量目标节点对发送四元分组序列,图 2 所示的是采用四元分组序列测量的实例. 在四元分组序列中,分组 B_1 和分组 B_2 的目标地址都为节点 3,分组 A_1 和分组 A_2 的目标地址都为节点 2. 在目标节点 2 和目标节点 3 记录收到测量分

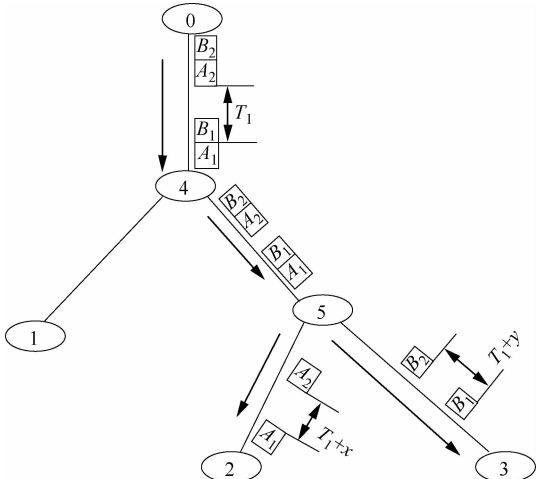


图 2 四元分组序列测量

1) 四元分组序列由 2 个相邻的紧接分组对组成,把 1 个四元分组序列表示为 $(x_1 - y_1, x_2 - y_2)$.

2) 紧接分组对 $x_1 - y_1$ 和紧接分组对 $x_2 - y_2$ 的目的地址都为 (x, y) . 其中,分组 x_1, x_2 具有相同的地址;分组 y_1, y_2 具有相同的地址.

3) 4 个分组的长度相同,并且为了减少测量流量,分组的长度都较小,例如可以设为 50 B.

4) 四元分组序列中 2 个紧接分组对之间具有固定的时间间隔,此值可以根据网络情况设定(通常为 10 ms),而紧接分组对内分组的间隔固定并且间隔较小(通常这个时间间隔非常小),以保证 2 个分组在相同链路所经历的网络状况相同.

5) 在四元分组序列中,每个分组携带的信息包括四元分组序列的编号、分组的发送时间和分组所属的紧接分组对序号(1 或 2). 四元分组序列的序号从 1 开始,依次递增,在四元分组序列内第 1 个紧接分组对序号为 1,第 2 个紧接分组对的序号为 2. 图 1 所示为四元分组序列.

组的时间,然后计算端到端的丢包率及端到端的时间抖动.

采用四元分组序列测量端到端时延抖动的同时,可以测量端到端的丢包率. 计算端到端的丢包率时把 1 个四元分组序列看作 2 个紧接分组对. 计算丢包率的方法与紧接分组对测量方法相同.

2 节点间相关性计算与拓扑推断算法

根据四元分组序列测量的端到端性能,可以计算节点间的相关性,在节点间相关性计算时有下面定义.

定义 1 设 $T = (V, L)$ 表示树状拓扑. 其中, V 为节点集; L 为链路集. 令 $a(i, j)$ 为叶节点 i 和叶节点 j 的父节点, $n_i < n_j$ 表示 n_i 是 n_j 的子孙,则 $i < a(i, j)$ 、 $j < a(i, j)$. 如果 $a(k, j) < a(i, j)$,则节点 k 和 j 的共享链路比节点 i 和 j 的共享链路多.

定义 2 节点间的相关性值是一种在树状结构的网络拓扑中,反映叶节点在网络中共享链路特性的值,叶节点的共享链路越多相关性值越大.

2.1 节点间相关性计算

在节点间能实现时钟同步的情况下,四元分组序列测量方法可以测量端到端单向时延,采用单向时延计算节点间相关性的方法与文献[8]中的计算相同,在节点间无法实现时钟同步的情况下,采用时延抖动和丢包率计算节点间相关性。

2.1.1 根据时延抖动计算相关性

时延抖动也称为时延变化,在数据流中选择1对数据分组,第1个分组和第2个分组时延的不同为时延变化,即时延抖动^[9]。

如图3所示,设四元分组序列的目标节点对为 (i, j) , $a(i, j)$ 为节点 i 和 j 的父节点, $Y_i^{(i,j)}$ 、 $Y_j^{(i,j)}$ 表示通过四元分组序列测量的到达节点 i 和 j 的端到端时延抖动向量, $J_a^{(i,j)}$ 、 $J_i^{(i,j)}$ 、 $J_j^{(i,j)}$ 分别为链路 $S \rightarrow a(i, j)$ 、链路 $a(i, j) \rightarrow i$ 和链路 $a(i, j) \rightarrow j$ 的时延抖动向量,计算目标节点 i 和目标节点 j 的时延抖动协方差。

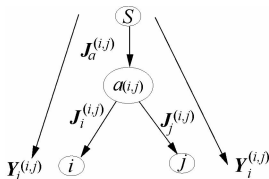


图3 时延抖动相关性

由于分组在不同链路是相互独立的,因此 $\text{Cov}(J_i^{(i,j)}, J_a^{(i,j)}) = 0$, $\text{Cov}(J_a^{(i,j)}, J_j^{(i,j)}) = 0$, $\text{Cov}(J_i^{(i,j)}, J_j^{(i,j)}) = 0$,可以得到下面结论:

$$\text{Cov}(Y_i^{(i,j)}, Y_j^{(i,j)}) =$$

$$\text{Cov}(J_a^{(i,j)} + J_i^{(i,j)}, J_a^{(i,j)} + J_j^{(i,j)}) = \text{Var}(J_a^{(i,j)}) \quad (1)$$

若路径 $S \rightarrow a(i, j)$ 由链路 L_1, L_2, \dots, L_n 组成, $J_1^{(i,j)}, J_2^{(i,j)}, \dots, J_n^{(i,j)}$ 分别为链路 L_1, L_2, \dots, L_n 的时延抖动,各链路相互独立,则

$$\text{Var}(J_a^{(i,j)}) = \text{Var}(J_1^{(i,j)}) + \text{Var}(J_2^{(i,j)}) + \dots + \text{Var}(J_n^{(i,j)}) \quad (2)$$

2.1.2 根据丢包率计算相关性

如图4所示,设四元分组序列的目标节点对为 (i, j) , $a(i, j)$ 为节点 i 和 j 的父节点。设 $P_a^{(i,j)}$ 、 $P_i^{(i,j)}$ 、 $P_j^{(i,j)}$ 分别表示链路 $S \rightarrow a(i, j)$ 、链路 $a(i, j) \rightarrow i$ 和链路 $a(i, j) \rightarrow j$ 的成功传输率,设 $S_i^{(i,j)}$ 为测量源节点 S 到叶节点 i 的端到端成功传输率, $S_j^{(i,j)}$ 为测量源节点 S 到叶节点 j 的端到端成功传输率, $S_{i,j}^{(i,j)}$ 表示紧接分组对中的分组都到达目标节点的概率。

率。令在测量源节点 S 发送的四元分组序列数量为 N_s ,当四元分组序列数量足够大时可以得到下面结论:

$$P_a^{(i,j)} = \frac{S_i^{(i,j)} S_j^{(i,j)}}{S_{i,j}^{(i,j)}}, N_s \rightarrow \infty \quad (3)$$

若路径 $S \rightarrow a(i, j)$ 由链路 L_1, L_2, \dots, L_n 组成, $P_1^{(i,j)}, P_2^{(i,j)}, \dots, P_n^{(i,j)}$ 分别为链路 L_1, L_2, \dots, L_n 的成功传输率,则

$$P_a^{(i,j)} = P_1^{(i,j)} P_2^{(i,j)} \dots P_n^{(i,j)} \quad (4)$$

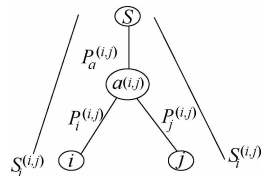


图4 丢包率相关性

2.1.3 2种参数计算相关性

2种参数计算节点间相关性的基本思想是通过四元分组序列测量方法,同时测量时延抖动和丢包率,然后计算时延抖动协方差和成功传输率。

由于时延抖动协方差和成功传输率具有不同的规律,为此采用下面的方法结合时延抖动协方差和成功传输率。对式(4)采用对数计算则可以把乘性关系转为加性关系,对经过对数计算的结果取反,则可以把 $P_a^{(i,j)}$ 转换成加性增的关系。

推论1 令 $J_{i,j}$ 表示叶节点对 (i, j) 的时延抖动协方差, $S_{i,j}$ 表示经过式(3)计算的叶节点对 (i, j) 的成功传输率,则 $J_{i,j}$ 和 $S_{i,j}$ 求和运算的值对节点对共享链路的特性起到放大作用。

由于不同参数的量纲不同,简单地把参数相加并不合适,为了解决不同参数的量纲问题采用下面的方法标准化,然后对时延抖动协方差和成功传输率加权求和。

令 $E(L)$ 为样本丢包率的均值,设 $D(J)$ 、 $D(S)$ 分别为 J 和 S 的方差, $\text{Min}(J)$ 和 $\text{Min}(S)$ 分别为 J 和 S 最小值,对 J 和 S 标准化计算,标准化公式和加权公式为

$$\left. \begin{aligned} \bar{J}_{i,j} &= \frac{\bar{J}_{i,j} - \text{Min}(J)}{\sqrt{D(J)}} \\ \bar{S}_{i,j} &= \frac{\bar{S}_{i,j} - \text{Min}(S)}{\sqrt{D(S)}} \end{aligned} \right\} \quad (5)$$

$$\left. \begin{aligned} \bar{M}_{i,j} &= (1 - \sqrt{E(L)}) \bar{J}_{i,j} + \sqrt{E(L)} \bar{Q}_{i,j} \\ 0 &\leq E(L) \leq 1 \end{aligned} \right\} \quad (6)$$

2.2 拓扑推断算法

采用2种网络性能参数计算节点间相关性的目的是扩大节点对共享链路的特征,并适用不同网络负载的情况,根据2.1.3小节分析,可知节点 i 和 j 的共享链路越多, $\bar{M}_{i,j}$ 越大. 拓扑推断算法是根据 $\bar{M}_{i,j}$ 的值来推断节点间共享链路情况,从而推断网络拓扑. 拓扑推断算法如下:

输入:根节点 $\{0\}$, 叶节点集 $V_r = \{\{1\}, \{2\}, \dots, \{M\}\}$,

边集 $L = \emptyset$, $M = \{\bar{M}_{1,2}, \bar{M}_{1,3}, \dots, \bar{M}_{i,j}, \dots,$

$\bar{M}_{m-1,m}\}$

输出: $T = (V, L)$

While $|V_r| > 1$ do

Begin

根据 J 对节点集划分 $V_r = A_1 \cup \dots \cup A_i \dots \cup A_n$, 对于每个集合 A_i , $\forall i \forall j (i \in A_i, j \in A_i) \rightarrow \bar{M}_{i,j} > \bar{M}_{i,m} (\forall_m m \notin A_i)$;

Foreach ($A_i \subset V_r$) do

Begin

If A_i 内节点数 > 1 then

Begin

合并 A_i 内的节点,生成新的节点 m ;

更新 J , 减去 J 中 A_i 的值, 加入新值 $\bar{M}_{m,x}$,

$\bar{M}_{m,x} (x \notin A_i) = \text{Max}(\bar{M}_{m,x} (x \in A_i))$;

$V = A_i \cup \{m\}$;

$L = \{(m, x), x \in A_i\}$;

$V_r = \{V_r - A_i\} \cup \{m\}$;

End;

End;

End;

拓扑推断算法的基本思想就是根据节点间的相关性值对节点做聚类,同类中的节点具有相同的父节点,采用递归聚类的方式直到把所有的节点聚为一类. 节点聚类的原则是节点类中,节点间的相关性大于不同类中节点间的相关性.

3 仿真实验

为了验证拓扑推断中采用四元分组序列测量方法和2种参数计算节点间相关性算法的正确性和有效性,在NS2环境下进行了仿真实验,在不同网络负载的情况下验证算法的正确性和健壮性,并与其他拓扑推断技术进行对比.

仿真实验采用较为普遍的树状结构网络拓扑,如图5所示. 图中与叶节点相连的链路带宽为1 Mbit/s,链路单向时延为10 ms,内部链路带宽为2 Mbit/s,链路时延为15 ms. 内部节点采用随机早期检测(RED)丢包策略,背景流量为自相似流.

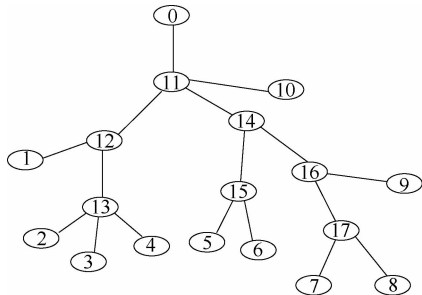


图5 NS2 仿真拓扑

在仿真实验中,对于每个叶节点对每组测量发送20个四元分组序列,测量的组数从1增加到25,根据多组测量的结果值计算节点对的相关性.

分别在网络负载较轻、网络负载适中和网络负载较重的情况下进行了多次仿真实验.

采用树编辑距离^[10]评价推断的拓扑树与原拓扑树的差别,树编辑距离是指把树 T_1 转为 T_2 所需要的基本操作步骤.

图6~8为仿真实验中不同推断技术获得的拓扑时,树编辑距离收敛的结果.

图6表明,在网络负载较轻的情况下,拓扑推断中采用时延抖动和综合参数推断拓扑的树编辑距离收敛速度较快,时延次之,采用丢包率的效果最差. 图7表明,在网络负载适中的情况下,拓扑推断中采用综合参数推断拓扑的树编辑距离收敛速度最快,时延抖动次之,丢包率最差. 图8表明,在网络负载较重时,拓扑推断中采用综合参数推断拓扑时树编辑距离收敛最快,丢包率次之,时延抖动最差.

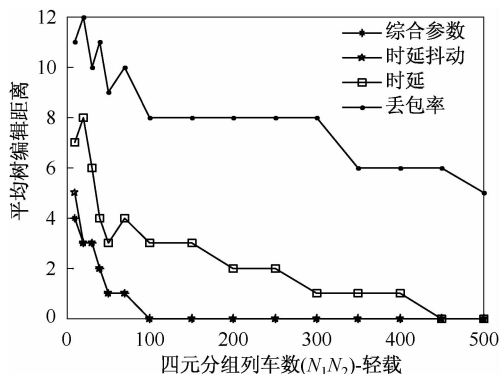


图6 网络负载较轻

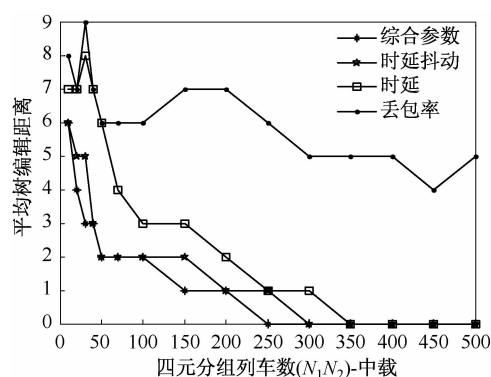


图7 网络负载适中

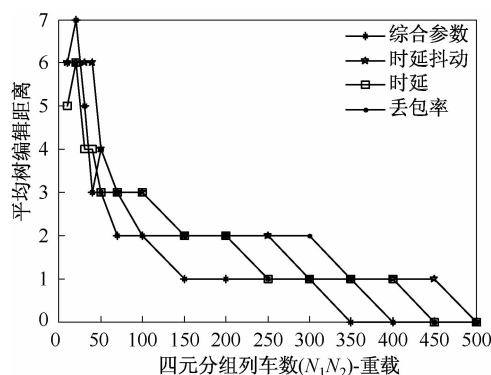


图8 网络负载较重

4 结束语

四元分组序列测量方法可以在各种网络负载环境下测量时延抖动和丢包率,根据时延抖动和丢包率计算的节点间相关性,在各种网络负载情况下都能正确反映节点间共享链路的情况.通过NS2仿真实验,在多种网络负载环境下验证了四元分组序列测量方法和多参数拓扑推断算法的优点.

参考文献:

[1] Ratnasamy S, McCanne S. Inference of multicast routing trees and bottleneck bandwidths using end-to-end measurements [C]//IEEE INFOCOM 1999. New York:

[s. n.], 1999;234-245.

- [2] Duffield D, Horowitz J, Presti F L, et al. Multicast topology inference from end-to-end measurements[C]//ITC Seminar on IP Traffic, Measurement and Modelling. Monterey:[s. n.], 2000;1321-1332.
- [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 Honghua, Chen Ming. Topology inference based on network tomography [J]. Journal of Software, 2010, 21(1): 133-146.
- [4] Coates M, Nowak R. Network loss inference using unicast end-to-end measurement[C]//ITC Conf on IP Traffic, Modelling and Management. Monterey:[s. n.], 2000;1120-1132.
- [5] Shih M F, Hero A O. Unicast-based inference of network link delay distributions with finite mixture models[J]. IEEE Trans on Signal Processing, 2003, 51(8): 2219-2228.
- [6] Coates M, Castro R, Nowak R. Maximum likelihood network topology identification from edge-based unicast measurements [C]//ACM Sigmetric. Marina Del Rey:[s. n.], 2002;1121-1130.
- [7] Zhao Honghua, Chen Ming. Network topology inference based on delay variation [C]//ICACC 2009. Singapore: Singapore, 2009;1401-1408.
- [8] Castro R, Coates M, Nowak R. Likelihood based hierarchical clustering[J]. IEEE Trans on Signal Processing, 2004, 52(8): 2308-2321.
- [9] Demichelis C, Chimento P. IP packet delay variation metric for IPPM[EB/OL]. (2002-04-02) [2011-10-16]. <http://www.ietf.org/internetdrafts/draft-ietf-ippm-ipdv-09.txt>.
- [10] Philip B. A survey on tree edit distance and related problems[EB/OL]. (2007-06-12) [2011-10-16]. http://grfia.dlsi.ua.es/ml/algorithms/references/editsurvey_bille.pdf.