文章编号:1001 - 9081(2006)04 - 0929 - 03

一种快速的报文丢失率推测方法

李勇军,蔡皖东,王 伟
(西北工业大学 计算机学院, 陕西 西安 710072)
(liyongjunxa@ hotmail. com)

摘 要:目前,在网络断层扫描的推测阶段主要采用的方法是似然估计,这些方法的计算量会随着网络规模的增加而急剧增长,从而影响在实际网络中的应用。为了克服似然估计引起的问题,提出了一种新的估计方法,该方法相对于似然估计,只需要简单的数值计算,计算量较小,且计算量不会随网络规模的增加而急剧增加。通过仿真比较可以看出估测的结果接近于真实值,能够真实反映网络报文丢失的趋势,在实际网络环境中有应用价值。

关键词:网络断层扫描;报文丢失率;性能参数推测
中图分类号: TP393 **文献标识码**:A

Fast approach to inferring loss performance

LI Yong-jun, CAI Wan-dong, WANG Wei
(*School of Computer Science*, *Northwestern Polytechnical University*, *Xi'an Shaanxi* 710072, *China*)

Abstract: Maximum likelihood estimates were often used in the network tomography to identify the loss rate. The time spent on the estimation increased sharply with the size of the network. To overcome the problems caused by MLE, a fast and simple approach was proposed to estimate loss rates. Compared with the previous methods, the proposed one only needed simple arithmetic calculation to determine loss rates, which saved more time than the MLE, and the time spent on the inference do not increased sharply with the size of the network. Through comparison and simulation, it is obtained that the loss inferences match the true results perfectly, and correctly show the loss trend. So the proposed method is very promising in the real network.

Key words: network tomography; loss performance; network performance inference

0 引言

网络性能参数是设计、管理和优化网络系统的必要条件,但是随着网络规模和复杂性的增长,获取网络内部性能参数变得越来越困难,尤其是获取某些自治系统内部的性能参数,因为网络安全和商业利益等原因,有些自治系统并不对外开放,难以实现内部节点的性能参数的共享。近年来,国际上提出一种新的网络测量技术,称为网络断层扫描(Network Tomography, NT),根据网络外部(即网络边界)的测量来分析和推断网络的内部性能[1]。这种测量方法的核心思想是多播或者类似多播的测量方法在网络内部的链路之间引入相关性,这种相关性可以用来推测网络内部性能,如报文丢失率和链路延时。

网络断层扫描方法主要分三个阶段:1)建立性能模型阶段;2)性能测量阶段;3)利用统计学方法分析和推测性能阶段。本文主要讨论在第三个阶段,目前网络断层扫描中采用的统计推测方法主要有极大似然方法、EM 方法和概似然方法等。这些方法一般要么通过迭代渐进真实值,要么通过求解高阶多项式得到结果,无论哪种方法在计算上花费的时间都会随着网络内部节点的增加会急剧增长,这就在一定程度上限制了这种方法在实际测量中的应用[2]。

为了克服似然估计带来的困难,本文提出了一种简单快速的网络内部报文丢失率的推测方法。和以前推测方法的不同之处在于计算方法上:首先利用在叶节点观测到报文丢失情况,推测网络内部节点的报文丢失情况,进而推测出网络内部链路的报文丢失率。该方法只需要简单的数值计算即可推测链路丢失率,计算结果可以很好地反映报文丢失的变化趋势。本文首先给出了报文丢失模型,在此基础上详细描述了报文丢失率推测算法,然后用仿真的方法证明了该算法的可行性。

1 报文丢失模型

1.1 树模型

用二叉树 $T = (V, L)$ 表示网络拓扑,文中所涉及到的测量和分析都以 T 为基础,其中 V 代表实际的网络设备,L 表示网络设备之间的连接。假设探测报文从根节点被多播到叶节点,根节点用 $0 \in V$ 表示,叶节点用 $R \subset V$ 表示。$\forall k \in V \backslash R \cup \{0\}$,称 k 为内部节点。任意一个内部节点 k 的孩子用 $c(k)$ 表示,任何一个非根节点 k 的父节点用 $p(k)$ 表示,链路 $(p(k), k) \in L$ 用 k 表示。用 $d(k)$ 表示节点 k 的子孙节点。

1.2 报文丢失模型

假设树 T 中链路 k 的报文丢失符合 Bernoulli 模型,报文在链路上丢失的概率为 α_k,则报文成功经过链路 k 的概率是 $p_k = 1 - \alpha_k$。用随机过程 $X = (X_k)_{k \in V}$ 描述节点接收探测报文

收稿日期:2005 - 10 - 10;**修订日期**:2005 - 12 - 13 **基金项目**:"航天科技创新基金"资助项目

作者简介:李勇军(1973 -),男,山东济宁人,博士研究生,主要研究方向:网络与信息安全; 蔡皖东(1955 -),男,山东威海人,教授,博士生导师,主要研究方向:网络与信息安全; 王伟(1969 -),男,陕西乾县人,工程师,博士研究生,主要研究方向:网络与信息安全.

的情况,如果报文到达了节点 k,则 $X_k = 1$;否则 $X_k = 0$。根据树的特性有下面关系:

If $X_k = 0, X_j = 0, \forall j \in c(k)$

If $X_j = 1, \forall j \in c(k), X_k = 1$

对 n 个探测报文来说,每个节点 k 就维护了一个大小为 n 的 0 - 1 序列,用 $\{X_k^{(i)}\}, k \in V$ 表示,其中 0 表示对应的探测报文没有到达该节点,1 表示到达了该节点,用 $n(X_k = 1)$ 表示节点 k 接收到的探测报文数量。定义链路 k 的 α_k 和 p_k 如下:

$$P_k = \frac{n(X_k = 1)}{n(X_{p(k)} = 1)}$$

$$\alpha_k = 1 - \frac{n(X_k = 1)}{n(X_{p(k)} = 1)}$$

2 推测方法

网络断层扫描测量方法是在网络边缘进行的,通过端到端的测量来推测网络内部的性能,不需要网络内部设备的协作。探测报文从节点 $0 \in V$,自顶向下被多播到 T 中的每个节点,只有在叶节点才可以观测到报文丢失情况。为了推测网络内部的报文丢失率,需要先推测网络内部各节点的报文丢失情况。

2.1 推测内部节点报文丢失情况

在推测网络内部节点报文丢失情况时,不失一般性采用如图 1 所示的报文传输模型,节点 k 有两个子节点 i 和 j。

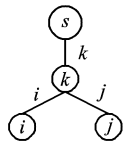

图 1 报文传输模型

假设节点 i 和 j 的链路丢失情况是已知的,分别为 $\{X_i^{(n)}\}$ 和 $\{X_j^{(n)}\}$,要推测的是节点 k 的报文丢失情况 $\{X_k^{(n)}\}$。根据节点 k 与其子节点之关系可知,节点 k 接收到的报文一部分成功到达了节点 i 或 j,还有一部分同时丢失在链路 i 和 j。对于前者来说可以从 $\{X_i^{(n)}\}$ 和 $\{X_j^{(n)}\}$ 得到,而对于后者这部分丢失的报文是不可推测的。因为节点 i 或 j 没有接收到报文,一种情况是丢失在链路 i 和 j 上,还有一种情况是丢失在链路 k 或者其上游链路上,这两种情况是难以区分的。用 Δ_{ij} 表示同时丢失在链路 i 和 j 上的报文数量,则节点 k 接收的报文数量为:

$$n(X_k = 1) = n(X_i = 1) + n(X_j = 1) - n(x_i = 1 \wedge X_j = 1) + \Delta_{ij} \quad (1)$$

Δ_{ij} 相对于其他部分相对较小,因此有:

$$\lim_{n\to\infty} n(X_k = 1) = n(X_i = 1) + n(X_j = 1) - n(X_i = 1 \wedge X_j = 1)$$

当探测报文数量足够大的时候,可以用节点 i 和 j 的报文丢失情况来估测节点 k 的报文丢失情况。

$$\hat{X}_k = \{X_i^{(l)} \vee X_j^{(l)}, l = 1,2,\cdots,n\} \quad (2)$$

叶节点的报文丢失情况是可以在测量过程观察到的,可以用上述方法估测上一层节点的报文丢失情况,重复上述过程就可以得到所有内部节点的报文丢失情况。该推测方法很容易推广到一般的树形结构中,推测公式(2)变成如下形式:

$$\hat{X}_k^{(l)} = \bigvee_{i \in R \cap d(k)} X_i^{(l)}, l = 1,\cdots,n$$

2.2 推测算法

在推测出内部节点的报文丢失情况以后,根据报文丢失模型中的定义就很容易计算网络内部链路的报文丢失率,下面的算法就详细描述了报文丢失率的推测方法。

1) 输入:叶节点集合 R,探测报文数量 n,叶节点接收报文情况 $(X_k^{(i)})_{k\in R}^{i=1,\cdots,n}$。

2) $R' = R, P' = \Phi$

3) for $k \in R'$ do

if $b(k) \notin R'$ then

退出本次循环,继续下一个节点

if $|R'| = 1$ then

退出 for 循环

根据节点 k 和 $b(k)$ 的报文接收情况,计算节点 $p(k)$ 的报文接收情况

$X_{p(k)}^{(i)} = X_k^{(i)} \vee X_{b(k)}^{(i)}, i = 1,\cdots,n$

$$p_k = \frac{n(X_k = 1)}{n(X_{p(k)} = 1)}$$

$$p_{b(k)} = \frac{n(X_{b(k)} = 1)}{n(X_{p(k)} = 1)}$$

$P' \Leftarrow P' \cup \{p_k, p_{b(k)}\}$

$R' \Leftarrow R' \setminus \{k, b(k)\}$

$R' \Leftarrow R' \cup \{p(k)\}$

4) $$p_1 = \frac{n(X_1 = 1)}{n}$$

$P' \Leftarrow P' \cup \{p_1\}$

5) 输出 P'。

3 仿真结果分析

为了验证算法的有效性,用 NS2 进行了实验仿真。用 8 个节点的二叉树表示网络拓扑,如图 2 所示。

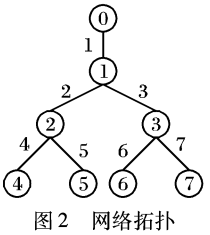

图 2 网络拓扑

图 2 中每条链路的带宽为 1M,传输延时 10ms,采用 DropTail 策略丢弃报文,队列大小为 10。另外为了在仿真过程观测到报文丢失,仿真过程中引入了 error model,设定链路 1 的报文丢失概率为 0.1,链路 2 和 3 的报文丢失的概率是 0.08,链路 4,5,6 和 7 的丢失概率是 0.05。

探测报文由 Exponential 产生,报文大小为 200 Byte,空闲时间(idle time)为 200ms,产生时间(burst time)为 300ms,产生速率为 32kbps。报文从节点 0,自顶向下被多播到网络中每个节点,测量从 0s 开始到 1000s 结束。在实验中,每隔 50s 在叶节点收集一次实验数据,作为一次采样。同时在网络内部节点上以相同的时间间隔记录报文丢失的真实状况。

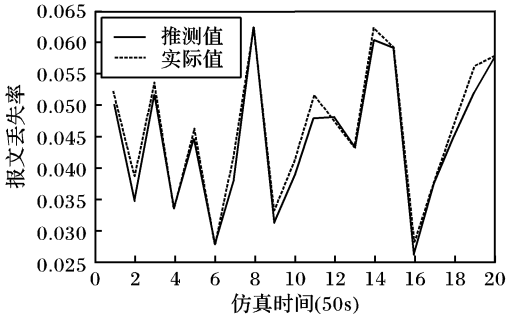

图 3 链路 4 上报文丢失率比较

通过实验可以得到 20 个不同时间段的实验数据,如图 3 和图 4 所示。图中同时显示了推测的链路丢失率和真实的链路丢失率。通过图 3 可以看出在叶节点依据本算法推测出的链路丢失率和真实的链路丢失率很接近,较好地反映了报文丢失的状况。

虽然图 4 中显示的两种报文丢失率有一定的差异,但是推测的丢失率还是很接近真实值,反映了真实的报文丢失状况的变化趋势,差异增加是由于误差累积的结果。

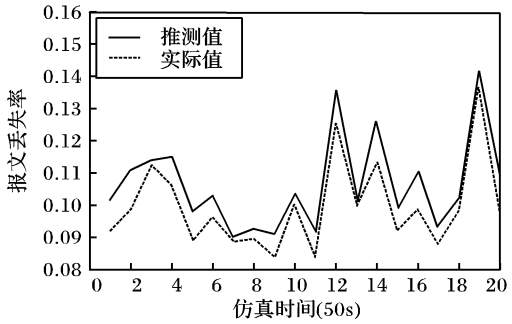

图 4　链路 1 上报文丢失率比较

以图 2 所示的网络拓扑为例进行误差分析,内部节点 2 依据本文提出的算法估测的接收到的报文数量与实际接收到的报文数量之差为 $n(X_2 = 1) - \hat{n}(X_2 = 1,) = \Delta_{45}$,同样节点 3 估测的误差是 Δ_{67}。而节点 1 的估测误差就成了同时丢失在其左右子树上的报文数量,即 $\Delta_{23} + \Delta_{4567} + \Delta_{267} + \Delta_{345}$。从上面误差分析不难看出,越靠近叶节点的节点上的估测误差越小,而越靠近根节点的节点估测误差相对越大。这种误差在实际网络测量中是不可测,因而在本方法中也就不可避免。

4　结语

网络断层扫描技术作为一种外部网络测量方法,引起了广泛的关注,主要根据发生在网络边缘的测量数据,推测网络内部的性能指标。目前在推测阶段主要采用的方法是似然估计,这种方法计算复杂,影响在实际网络中的应用。本文提出了一种简单快速的估计方法,通过仿真试验可以看出该方法能够有效地推测出报文丢失率。

参考文献:

[1] MARK C, HERO AO III, ROBERT N, *et al*. Internet Tomography [J]. IEEE Signal Processing Magazine, 2002, 19(3): 47 - 65.

[2] ZHU W, GENG Z. A bottom-up inference of loss rate[J]. Computer Communications, 2005, 28(4):351 - 365.

[3] RAMON C, DUFFIELD NG, JOSEPH H, *et al*. Multicast-Based Inference of Network-Internal Loss Characteristics[J]. IEEE Transactions on Information Theory, 1999, 45(7): 2462 - 2480.

[4] LIANG G, YU B. Maximum Pseudo Likelihood Estimation in Network Tomography[J]. IEEE Transactions on Signal Processing, 2003, 51(8): 2043 - 2053.

[5] Information Sciences Institute of University of Southern California. The Network Simulator 2[EB/OL]. www. isi. edu/nsnam/ns2, 2005.

(上接第 928 页)

[2] MILLER DC, THORPE JA. SIMNET: The advent of simulator networking[J]. Proceedings of the IEEE, 1995, 83(8): 1114 - 1123.

[3] ABRASH M. Quake's game engine: The big picture[J]. Dr. Dobb's Journal, Spring 1997.

[4] LEWIS M, JACBOSON J. Game Engines in Scientific Research[J]. Communication of the ACM, 2002, 45(1).

[5] NITTA T, CONE KF. An Application of Distributed Virtual Environment To Foreign Language[A]. IEEE Education Society[C]. Kansas City, Missouri, 2000.

[6] ZHANG J, LI FS, LI H. Multi-user Shared Virtual Reality in the Exhibition of Chinese Nationalities-Virtual Museum of Chinese Nationalities[A]. The Sixth International Conference on Computer Supported Cooperative Work in Design[C]. 2001. 83 - 88.

[7] FUNHOUSER T. RING: A Client-Server System for Multi-User Virtual Environments[A]. Proceedings of Symposium on Interactive 3D Graphics[C]. 1995. 85 - 92.

[8] DAS T, SINGH G, MITCHELL A. NetEffect: A Network Architecture for Large-scale multi-user Virtual World[A]. Proceedings of ACM VRST[C]. 1997. 157 - 163.

[9] HORI M, ISERI T, FUJIKAWA K, *et al*. Scalability Issues of Dynamic Space Management for Multiple-Server Networked Virtual Environments[A]. Proceedings of IEEE Pacific Rim Conference On Communications, Computers and signal Processing[C]. 2001. 200 - 203.

[10] LAU R, NG B, SI A, *et al*. Adaptive Partitioning for Multi-server Distributed Virtual Environments[A]. Proceedings of the ACM International Multimedia Conference and Exhibition[C]. 2002. 271 - 274.

[11] NGUYEN T, DUON B, ZHOU S. A Dynamic Load Sharing Algorithm for Massively Multiplayer Online Games[A]. The 11th IEEE International Conference on Networks[C]. 2003. 131 - 136.

[12] LUI JCS, CHAN MF. An Efficient Partitioning Algorithm for Distributed Virtual Environment Systems[J]. IEEE Transactions on Parallel and Distributed Systems, 2002, 13(2): 193 - 211.

[13] Ultima Online[EB/OL]. www. uo. com, 2005.

[14] Everquest[EB/OL]. www. everquest. station. sony. com, 2005.

[15] BASSIOUNI MA, CHIU MH. Performance and reliability analysis of relevance filtering for scalable distributed interactive simulation [J]. ACM Transactions on Modeling and Computer Simulation, 1997, 7(3):293 - 331.

[16] IEEE Std 1278.1 - 1995, Standard for Distributed Interactive Simulation-Application Protocols[S].

[17] SEOK-JONG YU, YOON-CHUL CHOY. A dynamic message filtering technique for 3D cyberspaces[J]. Computer Communications, 2001, 24(18):1745 - 1758.

[18] MOY J. Multicast routing extensions to OSPF, IETF RFC 1584[S].

[19] DEERING S. The PIM Architecture for Wide-Area Multicast Routing[J]. IEEE/ACM Transication on Network, 1996, 4:53 - 62.

[20] BALLARDIE T. Core Based Trees (CBT version 2) Multicast Routing, RFC 2189[S].

[21] DEERING SE. Host extensions for IP multicasting, RFC 1112[S].

[22] WAITZMAN D, PARTRIDGE C. Distance Vector Multicast Routing Protocol, RFC 1075[S].

[23] MOY J. OSPF Version 2, RFC 2328[S].

[24] ESTRIN D. Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol specification, RFC 2362[S].

[25] THALER D, ESTRIN D, MEYER D. Border Gateway Multicast Protocol (BGMP): Protocol specification[Z]. Internet draft, 1998.

[26] HOLBROOK H, CHERITON D. IP Multicast Channels: Express Support for large-Scale Single-Source Applications[A]. ACM sigcomm'99[C]. 1999.

[27] ERLMAN R. Simple Multicast: A Design for Simple, Low-Overhead Multicast[Z]. Internet draft, 1999.

[28] RAMAKRISHNA V, ROBINSOM M, EUSTICE K, *et al*. An Active Self-Optimizing Multiplayer Gaming Architecture[A]. Autonomic Computing Workshop[C]. 2003. 32 - 41.

[29] MACEDONIA MR, ZYDA MJ. Exploiting Reality with Multicast Groups: A Network Architecture for Large-Scale Virtual Environments[A]. Virtual Reality Annual International Symposium Proceedings[C]. 1995. 11 - 19.

[30] SAHASRABUDDHE LH, MUKHERJEE B. Multicast routing algorithms and protocols: A tutorial[J]. IEEE Network, 2000, 14(1): 90 - 102.