

IP网络坐标抖动感知与慢启动抑制

王 聪, 张凤荔, 刘梦娟, 王 勇

(电子科技大学计算机科学与工程学院 成都 611731)

【摘要】为了抑制Vivaldi网络坐标系统抖动现象并提高距离预测准确性,将Vivaldi归结为一类非线性方程组的迭代求解算法,并基于方程组的矛盾性提出了迭代因子的自适应估计问题。据此,定义单位化误差作为新的度量参数,通过对该参数的周期性测量实现坐标抖动的感知。进而对迭代过程进行逻辑分层,提出了一种基于慢启动策略的抖动抑制算法。实验证明,算法不依赖于迭代因子的人工选取,能够在不损失定位精度和收敛速度的前提下,将节点坐标抖动程度降低83.5%以上,并能与Vivaldi协议混合部署,实现坐标系统的平滑演进。

关键词 收敛性; 网络测量; 网络坐标系统; Vivaldi

中图分类号 TP393.4

文献标识码 A

doi:10.3969/j.issn.1001-0548.2012.06.020

IP-Based Network Coordinate Oscillation Awareness and Slow-Start Mitigation

WANG Cong, ZHANG Feng-li, LIU Meng-juan, and WANG Yong

(School of Computer Science & Engineering, University of Electronic Science and Technology of China Chengdu 611731)

Abstract To mitigate oscillation phenomenon in Vivaldi Network Coordinate System (NCS) and improve the accuracy of Internet distance prediction, Vivaldi is categorized as a kind of iterative method to solve non-linear equations and thus the adaptive estimation of iterative factor in Vivaldi is raised based on equations' contradictory. By defining a new metric unitization error and measuring it periodically, the awareness of network coordinate oscillation degree can be achieved in real time. Then we separate the iterative process of Vivaldi into 2 logical layers and propose a simple slow-start adaptive algorithm to estimate iterative factor automatically in order to mitigate network coordinate oscillation. The experiments and simulations show this algorithm can reduce the coordinate oscillation by no less than 83.5% without losing any velocity and accuracy, no matter what the initial value of iterative factor is. Furthermore, this algorithm can also be deployed in a hybrid environment together with raw Vivaldi to make the update of NCS smoothly.

Key words convergence; internet measurement; network coordinate system; Vivaldi

网络性能优化很大程度上依赖于网络节点间的距离信息获取。基于距离信息,内容分发网络可以优化拓扑结构以削减域间流量;流媒体网络可以感知距离最近的服务器降低交互成本;在线广告服务可以获知节点的地理位置以实施精准广告投放;云服务商也可以根据用户位置动态调度资源等等。真实网络应用通常采用往返时延(round trip time, RTT)作为网络距离度量的依据。然而RTT需要网络节点之间两两互发数据包,其测量计算复杂度为 $O(N^2)$,网络规模较大时不实用。于是考虑基于节点间的RTT信息将其嵌入到某个度量空间中,根据节点在空间中的虚拟坐标预测节点间的距离,从而产生了

网络坐标系统(network coordinate system, NCS)。

文献[1]提出了分布式网络距离感知算法Vivaldi,将网络坐标系统模拟为一个节点之间相互连接的弹簧系统,并将系统平衡状态作为最小误差状态求取。包括Vuze^[2], Xbox LIVE^[3]在内的许多互联网应用都直接或间接使用Vivaldi作为网络距离感知的工具,广域网环境下云资源的调度与定位同样依赖于Vivaldi^[4],部分DHT网络也基于Vivaldi算法实现了拓扑结构的优化^[5]。由于网络距离与地理距离存在近似的线性相关性^[6],文献[7]将网络节点嵌入到地球坐标系中,实现了街道级的网络节点定位精度。然而,NCS的一个重要前提是网络节点能够

收稿日期: 2012-08-14; 修回日期: 2012-09-12

基金项目: 国家科技重大专项(2011ZX03002-002-03); 中央高校基本科研业务费(ZYGX2010J077)

作者简介: 王聪(1981-),男,博士生,主要从事IP网络测量与增强技术方面的研究。

完美嵌入到目标度量空间, 考虑到互联网的高度非线性特征^[8], 该前提假设通常很难满足。文献[9]发现网络中三角违例(triangle inequality violations, TIVs)存在局部性高发现象并造成坐标抖动, 为了抑制TIVs对精度的影响, 随机选取部分邻居节点来检测坐标更新所依据的RTT值是否构成TIV, 并根据TIV严重程度修正更新的权重^[10]。而TIV检测是一个NP-Hard问题^[9], 仅抽取部分参考节点并不能保证所有TIV信息都被检出。RTT延迟污染的研究虽然可以降低TIVs发生率, 但该类方法或丢弃部分RTT信息^[11-12], 或需要增加网络基础设施以获取如RTT矩阵等全局信息^[13-14], 均未能从根本上解决TIVs问题。相反, 文献[15]的研究证明了TIVs来自于互联网路由算法的本质缺陷, 难以通过网络测量或数据处理方法的改进来根除。

本文为将坐标抖动归结为非线性矛盾方程组的迭代求解问题, 并在此基础上, 针对迭代步长的选取问题, 提出了一种基于单位化误差的抖动感知与慢启动步长选取策略, 在保证算法收敛速度的同时, 实现了坐标抖动的抑制。

1 坐标抖动问题

1.1 Vivaldi算法简介

Vivaldi算法将节点间空间距离与测量距离的误差模拟为弹簧作用力。当弹簧系统势能能达到最小时, 网络坐标系统达到稳定状态。

令节点*i*的坐标为向量*i*, Vivaldi计算过程如下:

1) 获取来自邻居节点*j*的坐标*j*, 误差*e_j*和传输时延RTT_{*i,j*};

2) 计算误差权值*w_{i,j}*, 令:

$$w_{i,j} = e_i / e_i + e_j$$

3) 计算距离相对误差*e_s*, 令:

$$e_s = \text{abs}(\text{RTT}_{i,j} - \|\mathbf{i} - \mathbf{j}\|) / \text{RTT}_{i,j}$$

4) 更新自身误差估计*e_i*, 令:

$$e_i = e_s c_e w_{i,j} + e_i (1 - c_e w_{i,j})$$

5) 更新自身坐标向量*i*, 令:

$$\mathbf{i} = \mathbf{i} + c_e w_{i,j} \frac{(\text{RTT}_{i,j} - \|\mathbf{i} - \mathbf{j}\|)(\mathbf{i} - \mathbf{j})}{\|\mathbf{i} - \mathbf{j}\|}$$

1.2 抖动问题的理论解释

Vivaldi实质上是对式(1)的迭代求解^[16]:

$$\text{RTT}_{i,j} - \|\mathbf{i} - \mathbf{j}\| = 0$$

$$i = 1, 2, \dots, N; \quad j \in \text{Neighbor}(i) \quad (1)$$

当节点完美嵌入到度量空间时, 式(1)成立。但

TIVs现象的广泛存在使得式(1)一般成为一个矛盾方程组。此时基于Vivaldi算法求解无论迭代步长大小, 都至少有一个方程不为0, 对该方程的迭代求解必然导致坐标的抖动。当迭代步长较大时, 算法能够以较快速度收敛, 出现剧烈的抖动现象; 而当迭代步长较小时, 抖动程度虽然相对较轻, 但网络收敛速度又不尽人意。虽然Vivaldi提出了一个自适应步长选取算法, 利用节点间的误差信息对迭代步长作自适应调节, 但由于 $c_e w_{i,j} + c_e w_{j,i} = c_e$, 使得Vivaldi的迭代步长依然严重依赖于 c_e 的取值, 该问题并未得到充分讨论, 文献[1]也仅在实验的基础上主观给出了一个经验值。

2 抖动感知与抑制算法

在TIVs不可避免的前提下, 既能加快收敛速度又能降低坐标抖动, 成为了NCS性能提升的重要问题。本文借鉴了TCP/IP的拥塞控制协议^[17], 建立了一个基于慢启动策略的变步长迭代算法。

2.1 算法工作原理

迭代步长与坐标抖动和收敛速度均具备强相关性。因此, 本文考虑将Vivaldi算法的迭代步作为子步(sub-step), 将多个子步聚合为一个超步(sup-step)。如图1所示, 本文算法在超步中感知节点当前状态, 收敛过程中超步会给定一个较大的迭代步长加快收敛; 收敛完成后, 超步会减小迭代步长以抑制坐标抖动。每当节点接收到一个RTT信息, 就启动一轮子步迭代, 基于超步确定的迭代因子执行标准Vivaldi算法, 并更新自己的坐标。当超步步长涵盖算法整个生命周期, 或禁止更新迭代步长时, 将退化为标准的Vivaldi算法。可认为节点将其邻居节点信息更新一遍即为一轮超步。于是, 算法须解决如何判断抖动程度和如何确定恰当的迭代因子两个关键问题。

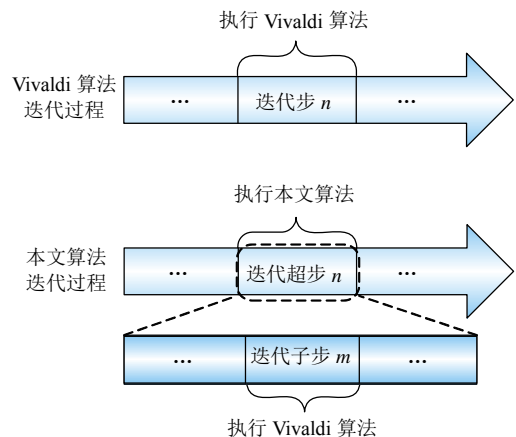


图1 算法逻辑层次

2.2 坐标抖动感知方法

首先给出单位化误差的定义与计算方法。单位化误差是在节点的某个时间切片中, 其邻居节点测量距离与坐标距离误差的均值。假定节点 i 在时刻 t 的坐标为 $\mathbf{i}(t)$, 则单位化误差可表示为:

$$\text{err}_i(t) = \frac{\sum_{j \in \text{Neighbor}(i)} \sqrt{(\text{RTT}_{i,j} - \|\mathbf{i}(t) - \mathbf{j}(t)\|)^2}}{|\text{Neighbor}(i)|} \quad (2)$$

式中, \square 运算为取集合的势。

随着迭代运算的不断进行, 节点的单位化误差呈逐渐下降的趋势, 并最终进入到一个往复振荡的阶段。单位化误差在不同阶段的动力学特性可作为抖动感知与评价依据。如果节点 i 在某轮超步 t 结束时生成的单位化误差增大, 即满足 $\text{err}_i(t) > \text{err}_i(t-1)$ 时, 算法认为此时发生了抖动, 从而在本轮迭代中减小迭代步长以抑制抖动; 否则, 认为算法仍在收敛当中, 在本轮迭代中增加迭代步长以加快收敛。

2.3 基于抖动感知的慢启动抑制算法

定义最小步长迭代因子 ε : 该因子是一个小于1的正数, 算法不得以小于 ε 的步长进行迭代。最小迭代因子的意义在于避免算法迭代步长过度退化, 以至于在网络发生突变时无法较快适应新的拓扑结构; 同时指定算法最大迭代因子为常数1。定义步长增长因子 l 作为放大器增加迭代步长; 定义步长衰减因子 h 作为衰减器减小迭代步长。基于抖动感知的慢启动抑制算法如下:

Sup-step1: 基于式(2)计算单位化误差 $\text{err}_i(t)$;

Sup-step2: 感知坐标抖动状态, 若满足 $\text{err}_i(t) > \text{err}_i(t-1)$ 则转Sup-step3, 否则转Sup-step4;

Sup-step3: 令 $c_c = \max(\varepsilon, c_c \times h)$, 并转

Sup-step5;

Sup-step4: 令 $c_c = \min(1, c_c \times l)$, 并转Sup-step5;

Sup-step5: 循环执行Sub-step至本轮超步结束, 并转Sup-step1;

Sub-step: 执行Vivaldi算法更新节点坐标。

由前文讨论易知, 当算法收敛时 $\text{err}_i(t)$ 总体呈下降趋势, 此时 c_c 更加倾向于正向增长; 而当算法收敛后会产生抖动, 此时应尽量减小 c_c 以抑制坐标抖动现象。因此, 在正常情况下步长增长幅度应小于衰减幅度, 即“慢速”启动、“快速”下降以保证算法的抖动抑制效率。算法虽名为“慢”启动, 但由于迭代步长会随着单位化误差的下降呈指数级增长, 因此, 算法通常可以在数步之内找到恰当的迭代因子。

本文仅在Vivaldi算法前端挂载了迭代步长预处理过程, 与原算法的计算复杂度同样为 $O(N)$ 。同时, 节点间RTT信息的获取可背负在正常通信任务之上, 并未改变测量负载的性质, 不会增加系统负担。

3 实验结果与分析

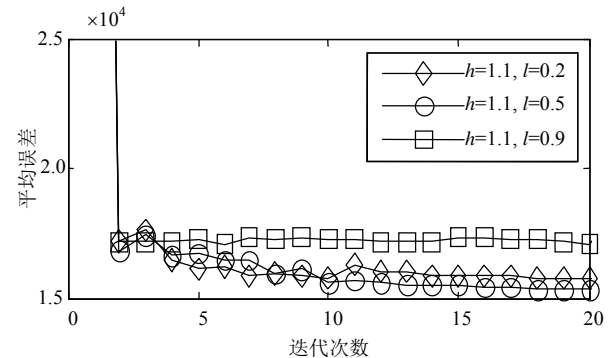
为了检验本文算法的速度、精度与鲁棒性, 本文以真实网络的数据集^[18]为基础进行仿真, 并以Vivaldi算法为基准算法进行比较。该数据集测量了2500个节点之间的RTT信息, 从中随机抽取200个节点作为实验样本集。为尽可能多地保留TIVs信息, 本文并未对样本集做删边操作。将度量空间定义为二维Euclid空间, 使其能够与节点在地理上的经纬度大致重合, 物理意义更加直观。

3.1 算法参数选取

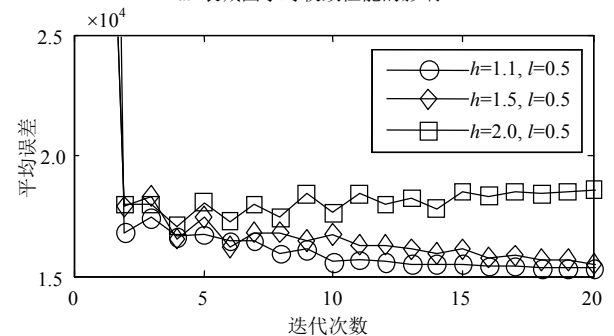
首先定义平均误差 ME (mean error)^[19]:

$$\text{ME} = \frac{\sum_{i=1}^N \sum_{j \in \text{Neighbor}(i)} \sqrt{(\text{RTT}_{i,j} - \|\mathbf{i} - \mathbf{j}\|)^2}}{\sum_{i=1}^N |\text{Neighbor}(i)|} \quad (4)$$

算法涉及增长因子 h 和衰减因子 l 两个重要参数, 如何选取参数取值以达到较优的性能, 是实现算法的关键问题。把 h 和 l 可能的取值区间分为高、中、低3个子区间, 分别选取较有代表性的取值进行讨论。



a. 衰减因子对收敛性能的影响



b. 增长因子对收敛性能的影响

图2 不同参数下算法迭代演进过程

由上节讨论可知,较小的 h 更有利于抖动抑制,因此首先取 $h=1.1$, $c_c=0.5$,讨论 l 的取值。取 $l=0.2$, 0.5 和 0.9 分别代表高中低3个子区间,可得算法演进过程,如图2a所示。可见,当 l 取值越大时,收敛速度会随之加快,但抖动也越剧烈,精度也会随之下降;而当 l 取值较小时,算法的收敛速度则不尽如人意。综合考虑,可认为 $l=0.5$ 是较优的取值。此时,取 $h=1.1,1.5$ 和 2.0 分别代表3个子区间进行讨论,可得到收敛过程如图2b。可见,当迭代步长增长过快时,坐标抖动也随之变得剧烈。考虑到网络拓扑结构的相对稳定性以及迭代因子指数级增长的特性,取 $l=1.1$ 较为合适。而收敛阶段的速度,则可通过取较大的 c_c 作为初始迭代因子补偿。

3.2 准确性分析

定义距离预测相对误差(relative error,RE)以度量算法的准确性^[19]:

$$RE_{i,j} = \text{abs} \left(\frac{RTT_{i,j} - \|i - j\|}{RTT_{i,j}} \right) \times 100\% \quad (3)$$

基本参数设为 $\varepsilon=0.001$, $c_e=0.2$, $l=1.1$, $h=0.5$,并将0.5作为本文算法的初始迭代因子进行计算。

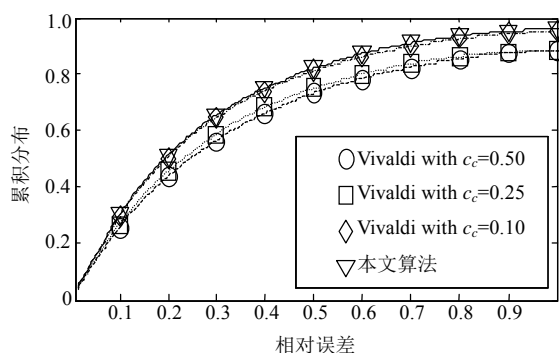


图3 相对误差累积分布

图3是经历200轮超步迭代后生成的相对误差累积分布,从图中可以分析算法距离预测的精确性。可以看出,当 $c_c=0.1$ 时,本文算法不具备明显的统计学意义,但当 c_c 增加到文献0推荐的较优值0.25时,显示出本文算法准确性。如本文算法有95.5%的距离预测相对误差在100%以内,而基准算法仅88.3%;当 c_c 进一步增大到0.5时,这一指标则下降为88.2%。这说明当网络趋于稳定时,本文算法的精确度高于基准算法。

3.3 收敛速度分析

以超步步长为间距进行采样,可得算法收敛过程如图4所示。

基准算法的收敛速度与误差受限于迭代因子 c_c

的取值,随着迭代因子的增加,在算法收敛速度逐渐加快的同时,误差也会随之增高。而本文算法则能较好地处理收敛速度与误差的关系,在保证算法平滑快速地趋近于收敛的同时,还能使误差始终保持在一个较低的水平。

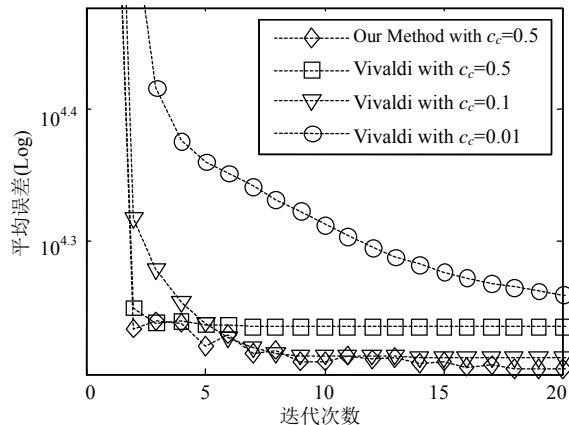


图4 算法收敛过程

3.4 抖动抑制能力分析

以上分析仅就超步迭代而言。事实上,算法在每个子步内都会更新自身坐标,其坐标抖动情况比超步层面上得到的观察结果要复杂,因此需在微观上对坐标抖动做进一步分析。假定节点 i 的 t 轮超步中包含 η_t 个子步,则节点 i 在 $m \sim n$ 轮超步中,节点坐标的数学期望 μ_i 和标准差 σ_i 为:

$$\left\{ \begin{aligned} \mu_i &= \sum_{t=m}^n \frac{\sum_{\eta_t \in t} i(\eta_t)}{t \times \eta_t} \\ \sigma_i &= \sqrt{\frac{\sum_{t=m}^n \sum_{\eta_t \in t} (\|i(\eta_t) - \mu_i\|)^2}{\sum_{t=m}^n \sum_{\eta_t \in t} \eta_t - 1}} \end{aligned} \right. \quad (5)$$

σ_i 可以反映节点坐标抖动程度。 σ_i 越大,坐标抖动相应地越剧烈。取不同的 c_c 值代入基准算法和本文算法,并取第101~200轮超步所蕴含全部子步迭代的时间切片分析节点坐标抖动的情况。

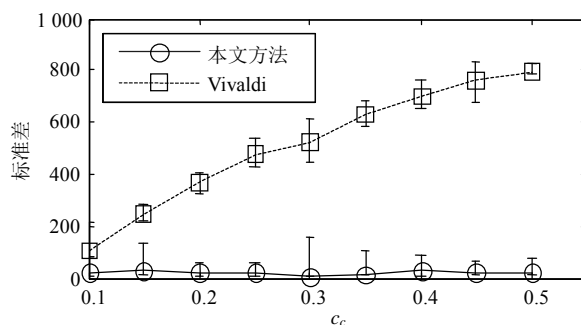


图5 坐标抖动与迭代因子相关性

对全部节点坐标更新序列的标准差进行统计, 以反映迭代因子 c_c 对坐标抖动程度的影响。统计结果如图5所示。当 $c_c=0.1$ 时, 执行基准算法的节点坐标抖动的标准差均在 $73\sim 210\ \mu\text{s}$ 之间, 平均约为 $103\ \mu\text{s}$; 而执行本文算法时标准差的范围则缩小到了 $7\sim 25\ \mu\text{s}$, 标准差平均仅为 $17\ \mu\text{s}$ 。并且, 无论初始迭代步长 c_c 的值如何变化, 坐标抖动的均值都可控制在 $30\ \mu\text{s}$ 之内, 而最大值也不超过 $160\ \mu\text{s}$ 。即使将迭代因子 c_c 取较小的值 0.1 进行迭代计算, 本文算法仍能在基准算法的基础上将坐标抖动程度进一步降低 83.5% 。如果为了兼顾收敛速度而将迭代因子 c_c 取较大的值, 如 $c_c=0.5$ 时, 本文算法的抖动抑制能力可达 98% 。可见, 本文算法在兼顾准确性和收敛速度的同时, 对抖动的抑制远优于基准算法。

3.5 协议演进能力分析

本文优化算法可与已被广泛部署的 Vivaldi 算法保持接口级兼容。将样本节点平均分为两部分, 一部分取 $c_c=0.25$ 执行基准 Vivaldi 算法, 另一部分则取 $c_c=0.5$ 作为初始迭代因子执行本文优化算法, 以检验算法兼容性与混合部署的能力。展开并统计第 $101\sim 200$ 轮超步迭代的节点坐标标准差累积分布, 可得两种节点的坐标抖动如图6。可见, 即使在混合部署的环境下, 本文的算法对抖动的抑制也明显优于 Vivaldi 算法。在执行本文算法的节点中, 坐标抖动标准差介于 $27\sim 121\ \mu\text{s}$ 之间, 相较执行基准算法的节点, 体现出了更加良好的抖动抑制能力。并且, 由于在部分节点中加入了慢启动抑制策略, 使得执行基准算法的节点坐标抖动程度也得到了明显的改善, 从单独部署时的 $428\sim 536\ \mu\text{s}$ 下降到了 $63\sim 144\ \mu\text{s}$ 。实验证明本文算法与基准算法的混合部署同样可以有效提高全网的坐标抖动抑制能力, 对应用软件的平滑升级是非常有利的。

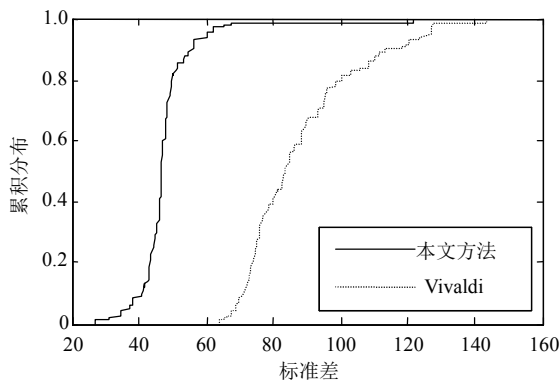


图6 混合部署性能

4 结论

本文重新解释了 Vivaldi 网络坐标系统, 将其归结为一类求解非线性方程组的迭代算法。据此, 首次定义了单位化误差作为坐标抖动感知系数, 提出了一种基于慢启动策略的快速迭代与抖动抑制算法, 通过对抖动的感知实现了迭代步长的自适应调整, 以抑制节点坐标抖动。本文算法可与已被广泛部署的基准算法无缝兼容, 实现坐标定位协议的平滑演进。实验结果显示, 本文提出的优化算法收敛性不依赖于迭代步长的选取, 能够在不丧失精度和收敛速度的前提下实现快速迭代, 有效抑制坐标抖动现象, 提高了网络距离感知的准确性与鲁棒性。

参考文献

- [1] DABEK F, COX R, KAASHOEK F, et al. Vivaldi: a decentralized network coordinate system[C]//ACM SIGCOMM. Portland, OR, USA: ACM, 2004: 15-26.
- [2] LEDLIE J, GARDNER P, SELTZER M. Network coordinates in the wild[C]//The 4th USENIX Symposium on Networked Systems Design & Implementation. Cambridge, MA, USA: USENIX, 2007.
- [3] AGARWAL S, LORCH R. Matchmaking for online games and other latency-sensitive P2P systems[C]//ACM SIGCOMM. Barcelona, Spain: [s.n.], 2009.
- [4] AGARWAL S, DUNAGAN J, JAIN N, et al. Volley: automated data placement for geo-distributed cloud services[C]//The 7th USENIX Symposium on Networked Systems Design & Implementation. San Jose, CA, USA: USENIX, 2010.
- [5] KOJIMA T, ASAHARA M, KONO K, et al. Embedding network coordinates into the heart of distributed hash tables[C]//IEEE Ninth International Conference on Peer-to-Peer Computing. Seattle, WA, USA: IEEE, 2009.
- [6] PATRIK M, DAN K, RADIM B, et al. Study and performance of localization methods in IP based networks: Vivaldi algorithm[J]. Journal of Network and Computer Applications, 2011, 34(1): 351-367.
- [7] WANG Y, BURGNER D, FLORES M, et al. Towards Street Level Client Independent IP Geolocation[C]//The 8th USENIX Symposium on Networked Systems Design & Implementation. Boston, MA, USA: USENIX, 2011.
- [8] LUMEZANU C, BADEN R, SPRING N, et al. Triangle inequality variations in the internet[C]//Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference. New York, NY, USA: ACM, 2009.
- [9] LEE S, ZHANG Z L, SAHU S, et al. On Suitability of euclidean embedding for host-based network coordinate systems[J]. IEEE/ACM Transactions on Networking, 2010, 18(1):27-40.

- [10] 黄琼, 刘熙, 阳小龙, 等. T-Vivaldi: TIV感知的IP网络坐标系统[J]. 电子科技大学学报, 2012, 41(1): 147-151.
HUANG Qiong, LIU Xi, YANG Xiao-long, et al. T-Vivaldi: TIV-aware coordinates system for IP networks[J]. Journal of University of Electronic Science and Technology of China, 2012, 41(1):147-151.
- [11] LEDLIE J, PIETZUCH P, SELTZER M. Stable and accurate network coordinates[C]//International Conference on Distributed Computing Systems. Lisbon, Portugal: IEEE, 2006.
- [12] 张敏, 周亮, 黄琼, 等. 随机延迟污染对IP网络坐标系统的影响及其抑制方法研究[J]. 通信学报, 2011, 32(1): 29-35.
ZHANG Min, ZHOU Liang, HUANG Qiong, et al. Influence of stochastic delay contaminations on IP network coordinate system and contamination mitigation methods[J]. Journal on Communications, 2011, 32(1): 29-35.
- [13] CHAN-TIN E, HOPPER N. Accurate and Provably Secure Latency Estimation with Treeple[C]//Proceedings of the Network and Distributed System Security Symposium. San Diego, CA, USA: ISOC, 2011.
- [14] 王占丰, 陈鸣, 邢长友, 等. S-Vivaldi: 一种基于空间修复的因特网时延空间嵌入算法[J]. 通信学报, 2012, 33(3): 163-170.
WANG Zhan-feng, CHEN Ming, XING Chang-you, et al. S-Vivaldi: a space repairing based Internet delay space embedding algorithm[J]. Journal on Communications, 2012, 33(3): 163-170.
- [15] ZHANG B, NG T, NANDI A, et al. Measurement Based Analysis, Modeling, and Synthesis of the Internet Delay Space[J]. IEEE/ACM Transactions on Networking, 2006, 18(1): 246-254.
- [16] WANG C, ZHANG F, YANG X, et al. A differential dynamic model of distributed network coordinate system[C]//The 10th International Conference on Communications, Circuits and Systems. Taichung, Taiwan, China: IEEE, 2012.
- [17] JACOBSON V, KARELS J. Congestion avoidance and control[J]. ACM Computer Communication Review, 1988, 18(4): 314-329.
- [18] WONG B, SLIVKINS A, SIRER, E. Meridian: a lightweight network location service without virtual coordinates[C]//ACM SIGCOMM. Philadelphia, PA, USA: ACM, 2005: 85-96.
- [19] LUA K, GRIFFIN T, PIAS M, et al. On the Accuracy of Embeddings for Internet Coordinate Systems[C]//The 5th ACM SIGCOMM conference on Internet Measurement. Philadelphia, PA, USA: ACM, 2005.

编辑 税红