

PaperPass旗舰版检测报告

简明打印版

比对结果(相似度):

总体 : 27% (总体相似度是指本地库、互联网的综合对比结果)
本地库 : 22% (本地库相似度是指论文与学术期刊、学位论文、会议论文、图书数据库的对比结果)
期刊库 : 14% (期刊库相似度是指论文与学术期刊库的对比结果)
学位库 : 15% (学位库相似度是指论文与学位论文库的对比结果)
会议库 : 3% (会议库相似度是指论文与会议论文库的对比结果)
图书库 : 3% (图书库相似度是指论文与图书库的对比结果)
互联网 : 15% (互联网相似度是指论文与互联网资源的对比结果)

编号 : 5AD4384FAE506EWK3

版本 : 旗舰版

标题 : 基于高精度时空信息的网络坐标系统构建与性能优化方法研究

作者 : xyz

长度 : 25704字符(不计空格)

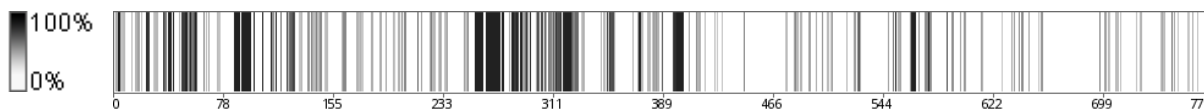
句子数 : 777 句

时间 : 2018-4-16 13:44:47

比对库 : 学术期刊、学位论文、会议论文、书籍数据、互联网资源

查真伪 : <http://www.paperpass.com/check>

句子相似度分布图:



本地库相似资源列表(学术期刊、学位论文、会议论文、书籍数据):

- 相似度 : 5% 篇名 : 《北斗卫星导航系统授时应用》
来源 : 学术期刊 《数字通信世界》 2011年6期
- 相似度 : 3% 篇名 : 《IP网络坐标抖动感知与慢启动抑制》
来源 : 学术期刊 《电子科技大学学报》 2012年6期
- 相似度 : 3% 篇名 : 《IP网络坐标系统的准确性增强技术研究》
来源 : 学位论文 电子科技大学 2011
- 相似度 : 2% 篇名 : 《网络空间嵌入模型与应用研究》
来源 : 学位论文 电子科技大学 2013
- 相似度 : 2% 篇名 : 《北斗卫星导航系统授时性能评估研究》
来源 : 学位论文 长安大学 2014
- 相似度 : 2% 篇名 : 《守时中的若干重要技术问题研究》
来源 : 学位论文 中国科学院国家授时中心 2007
- 相似度 : 1% 篇名 : 《基于虚拟坐标的IP网络定位理论与技术研究》
来源 : 学位论文 电子科技大学 2010
- 相似度 : 1% 篇名 : 《一种高精度GNSS单向授时方法实现研究》
来源 : 会议论文 2015-05-13
- 相似度 : 1% 篇名 : 《北斗系统时间(BDT)的认知与应用》
来源 : 学术期刊 《数字通信世界》 2013年8期

10. 相似度：1% 篇名：《基于虚拟坐标的IP网络定位技术研究》
来源：学位论文 重庆邮电大学 2008
11. 相似度：1% 篇名：《北斗/GPS授时技术的研究与实现》
来源：学位论文 桂林电子科技大学 2014
12. 相似度：1% 篇名：《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》
来源：学术期刊《通信学报》2011年12期
13. 相似度：1% 篇名：《一种高精度GNSS单向授时方法实现研究》
来源：学术期刊《时间频率学报》2016年4期
14. 相似度：1% 篇名：《TIV感知的IP网络坐标系统及其性能研究》
来源：学位论文 电子科技大学 2010
15. 相似度：1% 篇名：《基于改进的Vialdi算法的国内主机地理位置定位的研究》
来源：学位论文 浙江理工大学 2013
16. 相似度：1% 篇名：《新一代列车运行监控系统的研究与应用》
来源：学位论文 南京航空航天大学 2014
17. 相似度：1% 篇名：《基于大规模分布式网络的网络坐标系统研究》
来源：学位论文 哈尔滨工业大学 2015
18. 相似度：1% 篇名：《网络实体IP地理定位方法研究》
来源：学位论文 解放军信息工程大学 2015

互联网相似资源列表：

1. 相似度：4% 标题：《北斗卫星导航系统授时应用_百...》
<https://wenku.baidu.com/view/b016880801f69e314332948f.html>
2. 相似度：4% 标题：《北斗卫星导航系统授时应用》
<http://www.doc88.com/p-5867404544602.html>
3. 相似度：4% 标题：《培育数据文化，推动我国时空大数据发展.pdf》
<https://max.book118.com/html/2017/1029/138149127.shtm>
4. 相似度：2% 标题：《IP网络坐标抖动感知与慢启动抑制 - 电子科技大...》
<http://m.doc88.com/p-0394736711385.html>
5. 相似度：2% 标题：《IP网络坐标抖动感知与慢启动抑制》
<http://www.docin.com/p-741986060.html>
6. 相似度：2% 标题：《守时中的若干重要技术问题研究_百度文...》
<https://wenku.baidu.com/view/fe3a2f305a8102d276a22f65.html>
7. 相似度：2% 标题：《守时中的若干重要技术问题研究》
<http://d.wanfangdata.com.cn/Thesis/Y1626936>
8. 相似度：2% 标题：《守时中的若干重要技术问题研究》
<http://www.docin.com/p-911230199.html>
9. 相似度：2% 标题：《守时中的若干重要技术问题研究.pdf》
<http://www.docin.com/p-520921499.html?ref=360>
10. 相似度：2% 标题：《守时中的若干重要技术问题研究（PDF）》
<http://www.doc88.com/p-1972120044932.html>
11. 相似度：1% 标题：《卫星双向时间频率传递误差研究-天体测量与天体力学...》
<https://max.book118.com/html/2017/0712/121863931.shtm>
12. 相似度：1% 标题：《基于共视原理的卫星授时方法》
<http://www.docin.com/p-857740051.html>
13. 相似度：1% 标题：《一种高精度GNSS单向授时方法实现研究-会议论文...》
<http://www.doc88.com/p-7498229862413.html>
14. 相似度：1% 标题：《守时中的若干重要技术问题研究 - 中国论文下载...》
<http://www.shangxueba.com/lunwen/v95029.html>
15. 相似度：1% 标题：《大数据（巨量数据集）》
http://www.360doc.com/content/17/0528/18/16534268_658031523.shtml
16. 相似度：1% 标题：《太空新战场，游戏规则谁来主宰(图)_搜狐新闻_搜...》
http://www.sohu.com/a/125394333_313834
17. 相似度：1% 标题：《热点 美军改革重点为何是太空部队-来自微信公众号...》
<https://wx.abbao.cn/a/3630-951d275b0852c144.html>

- 18.相似度：1% 标题：《“互联网+”时空大数据与“GIS”的演进和发展...》
<https://wenku.baidu.com/view/9289b2b382d049649b6648d7c1c708a1284a0a9f.html>
- 19.相似度：1% 标题：《关于大数据最常见的10个问题 - 大数据分析_大...》
<http://www.raincent.com/content-10-6939-5.html>
- 20.相似度：1% 标题：《大数据：大数据的实际应用、发展趋势和面临的问题...》
http://www.sohu.com/a/191460264_99991132

全文简明报告:

绪论

1.1背景介绍

{52%：大数据正深刻影响着人们的生产方式、生活习惯、思维模式和研究方法。} {42%：大数据不仅是学界和业界的前沿课题，而且已上升为国家基础性战略资源。} {100%：大数据的开放、开发与利用已成为国家重大战略需求与重大工程需求。} {98%：同时，大数据应用中面临的数据安全威胁与隐私泄露也极大地破坏了正常的社会经济秩序，甚至危及国家网络空间安全。}

{41%：大数据是学界和业界的前沿课题，正深刻影响着人们的生产方式、生活习惯、思维模式及研究方法。} {43%：随着互联网、物联网、云计算等技术的迅猛发展，包括自媒体数据、日志数据和富媒体数据在内的网络大数据飞速发展。} {44%：然而，网络大数据在收集、存储和使用等全生命周期中面临着诸多安全挑战，} 大数据所带来的隐私泄露给用户带来了严重损失和潜在风险，极大破坏了社会经济秩序，影响了政务大数据、商务大数据、健康大数据等更多大数据的产业化应用。目前，大数据已上升为国家战略性资源，大数据安全与隐私保护不仅是国际学术前沿，也是国家重大战略需求。本项目致力于网络大数据环境中数据保护、隐私保护以及数字水印隐藏等方面的基础研究，具有重要理论意义和应用价值。

{44%：现有的数据隐私保护对象主要是用户的身份、社交关系以及用户属性等关键隐私信息。} 在大数据环境中，由于数据存在更广泛的关联性，使得原本孤立的信息成为隐私；同时，对经过传统匿名等技术处理后的数据，通过大数据关联挖掘和深度分析后，依然可能分析出用户的隐私， {46%：这些给大数据环境中的隐私甄别和隐私保护技术提出了新的挑战。}

{45%：基于大数据环境下的数据安全和隐私保护理论，面向时空大数据进行数据与隐私保护示范应用，} {52%：并对大数据环境下的数据与隐私保护性能与效率进行评价。} 针对位置大数据，突破北斗高精度定位时间同步技术，构建具有高精度时空信息的网络大数据环境， {42%：对大数据环境下融合时空信息的数据与隐私保护理论进行实验分析。} 课题的研究内容将本项目提出的数据与隐私保护理论机制落实到具体应用中，根据应用反馈修正数据与隐私保护理论，响应了指南的要求。

大数据的独特之处，除了规模巨大、类型多样、增长迅速等特性，最重要的是这些特性所导致的“全息”意义上的数据关联性，这种关联性将是实现未来商业模式、生产生活方式、管理流程等颠覆性变化的驱动力。 {92%：譬如国家电网智能电表的数据可用于估计房屋空置率，淘宝销售数据可用于预测经济走势，移动通讯基站定位数据可用于优化城市交通设计，} {83%：微博上的关注关系和内容信息可用于购物推荐和广告推送等。} 同时，数据关联性也是导致常规的数据保护与隐私保护方式失效的根本原因之一。例如，关联性挖掘分析使得仅通过匿名技术不能很好地保护用户隐私。但是，如果施加过强的数据保护策略，

必将割裂这些数据的关联性，从而形成一个数据孤岛并导致大数据服务的不可用。

{41%：大数据是学界和业界的前沿课题，正深刻影响着人们的生产方式、生活习惯、思维模式及研究方法。} {43%：随着互联网、物联网、云计算等技术的迅猛发展，包括自媒体数据、日志数据和富媒体数据在内的网络大数据飞速发展。} {44%：然而，网络大数据在收集、存储和使用等全生命周期中面临着诸多安全挑战，} 大数据所带来的隐私泄露给用户带来了严重损失和潜在风险，极大破坏了社会经济秩序，影响了政务大数据、商务大数据、健康大数据等更多大数据的产业化应用。目前，大数据已上升为国家战略性资源，大数据安全与隐私保护不仅是国际学术前沿，也是国家重大战略需求。本项目致力于网络大数据环境中数据保护、隐私保护以及数字水印隐藏等方面的基础研究，具有重要理论意义和应用价值。

{85%：“互联网+”、大数据、时空数据、时空大数据，是学界和业界讨论最多的问题。} {65%：时空大数据的提出有其重要的意义，从哲学层面看，空间与时间一起构成运动着的物质存在的两周基本形式。} 空间指物质存在的广延性； {100%：时间指物质运动过程的持续性和顺序性。} {91%：空间和实践具有客观性，同运动着的物质不可分割。} {100%：没有脱离物质运动的空间和时间，也没有不在空间和时间中运动的物质。} 空间和时间也是相互联系的。 {87%：现代物理学的发展，特别是相对论的提出，证明空间和时间同运动着的图纸的不可分割的联系。}

1.2 研究目标

高精度时空信息的网络大数据环境构建：以本项目提出的数据与隐私保护理论和方法为基础，针对位置大数据，通过北斗高精度定位时间同步技术， {46%：构建大数据环境中融合时空信息的数据与隐私保护理论验证环境。}

从运动着的万事万物表达的层面看，事务都可以分为空间维度（S-XYZ）、属性维度（D）和时间维度（T）。 {82%：时间维度（T）是指信息随时间的变化，具有时态性，需要有一个精确的时间基准；} {82%：空间维读（S-XYZ）是指信息具有精确的空间位置或者空间分布特征，具有可量测性，需要一个精准的空间基准；} {88%：属性维度（D）是指空间维度上可加载随时间变化的要素（现象）的各种相关信息（属性信息），} {96%：具有多为特征，需要有一个科学的分类体系和标准编码体系。}

{47%：从提升社会治理体系和治理能力现代化水平的层面看，随着全球化进程的加快，当今社会的一个重要特点是，} {66%：世界（区域、国家、城市）管理和治理对事件和空间的依赖程度越来越高，时空大数据} {92%：正日益成为全球（区域、国家、城市）治理体系和治理能力现代化的核心驱动力。} {63%：进入21世纪，太空成为继核武器之后的新型战略威慑力量，与核威慑、网络威慑交织融合，共同构成新的战略稳定架构；} {73%：太空实现全球作战力量一体化，不受传统的陆地、海上、空中飞越限制，具有天然的全球性和跨域性，} {95%：全球任何地点的作战力量和手段都能通过“天地一体网络”连接起来，形成一体化作战力量体系；} {73%：太空力量是战斗力的倍增器，可以在任何时间、任何地点、任何气象条件下打击地球上任何一个目标。} 而这必须有全球一体化的时空大数据平台保障。

1.3 文章结构

本论文以时空大数据为背景，结合北斗导航系统和网络协议构建时空坐标系， 为提供基于高精度时分信息的时空大数据研究提供研究基础，改进了 Vivaldi算法并提出了稳定抑制 Vivaldi算法， {61%：其目的是为了增加网络坐标系统的准确性。}

本文分为6章，主要内容如下：

{42%：第1章引言阐述了时空大数据和网络坐标系统的重要性，以及网络坐标系统的课题背景和意义，} 介绍了国内外时空大数据研究的进展和对网络坐标系统的部分研究成果和几种构建算法。

{50%：第2章介绍了什么是时空大数据，时空大数据的相关特征，} 四种现有的典型网络坐标系统构建算法 GNP[2]、PIC[10]、NPS[11]和 Vivaldi[3]，北斗导航系统（GNSS），网络实验环境和协议 OpenVSwitch和实验环境 MiniNet的介绍。

第3章介绍了时空大数据平台构建的方案，通过构建开放的体系架构为实现范围更广泛的信息资源共享与多层次多节点的协同工作提供崭新的运行环境。

第4章利用 Mininet和 OpenVSwitch协议，构建系统原型，实现了对时空数据的标签化处理和打标签操作， {43%：通过数据的时空戳属性，融合了空间坐标系和网络坐标系，构建了时空坐标系。}

{41%：第5章通过数据分析说明了随机延迟污染现象的普遍存在，介绍了一种现有的随机延迟污染抑制方法 MP-Filter[6]，} 并提出 T0-Filter的随机延迟污染抑制方法。并分析了恶化网络性能的其中两个因素：随机延迟污染现象以及TIV现象，同时介绍了对非中心式网络坐标系统安全性产生影响的几种攻击。三种现有的用于抑制TIV现象的网络坐标距离预测算法，并对其优缺点进行了分析。同时简单介绍了稳定抑制Vivaldi算法。提出了用于抑制随机延迟污染现象以及TIV现象的稳定抑制Vivaldi算法，并通过仿真实验分析其准确性和抑制抖动能力。

第6章是全文的总结。

项目关键技术分析

2.1 时空大数据

{100%：大数据（Big Data），指无法在一定时间范围内用常规软件工具进行捕捉、管理和处理的数据集合，} {100%：是需要新处理模式才能具有更强的决策力、洞察发现力和流程优化能力的海量、高增长率和多样化的信息资产。}

{95%：时空大数据，指基于统一的时空基准（空间参照系统、时间参照系统），活动（运动变化）} {100%：于时间和空间中与位置直接（定位）或间接（空间分布）相关联的大数据。} 之所以这样界定时空大数据是基于以下三个事实： {100%：一是，世界是物质的，物质是运动的，包括人类活动在内的万事万物的运动变化都是在一定的时间和空间中进行的，} {100%：而所有的大数据都是世界万事万物运动变化的产物；} {100%：二是，随着智能感知技术、物联网、云计算技术的发展，各个领域开始了“量化”的进程，} {97%：这种一切皆可“量化”（数字化）的趋势导致了大规模海量数据的产生，} {100%：而空间参照与时间参照是大数据的两个基本特征；} {97%：三是，从可视化角度讲，正是因为一切大数据都具有空间参照与时间参照特征，} {97%：所以才能直观地为人们提供大数据的空间位置，空间分布和时间标识。}

大数据具有以下特征： 容量（Volume）： {100%：数据的大小决定所考虑的数据的价值和潜在的信息；} 种类（Variety）： 数据类型的多样性； 速度（Velocity）： 指获得数据的速度； 可变性（Variability）： {100%：妨碍了处理和有效地管理数据的过程；} 真实性（Veracity）： 数据的质量； 复杂性（Complexity）： 数据量

巨大，来源多渠道； 价值 (value)： {100%：合理运用大数据，以低成本创造高价值。}

{56%：时空大数据除了具备一般大数据特征外，还具备其他特征：} 位置 (Location)：{90%：点、线、面、体的三维位置 (X、Y、Z) 拓扑、方向、度量很复杂；} 时间 (Time)：位置、属性等随时间的变化而变化； 属性 (Attribute)：{67%：每个位置点、线、面、体上都以自己的数量特征和说明信息；} 尺度 (Measure)： 空间尺度变化； 多维 (Dimension)：{79%：所有大数据与空间数据集成构成空间数据立方体，即“多维”可视化。}

{45%：时空大数据的产生、存储和管理需要一个时空大数据平台来完成。} {66%：当前，全球正在经历一场持久而深远的数据化（一切皆可“量化”）革命，跨界、融合、开放、共享是大数据时代的核心特征。} {71%：时空大数据平台，是把各种分散的（点数据）和分割的（条数据）大数据汇聚到一个特定的平台上一时空大数据平台。} {81%：这种聚合效应就是通过数据多维融合和关联分析与数据挖掘，揭示事物的本质规律，对事物做出更加快捷、更加全面、更加精确和更加有效地研判和预测。} {74%：从这个意义上讲，时空大数据平台是大数据的核心价值，是大数据发展的高级形态，是大数据时代的解决方案。}

2.2 网络坐标

2.2.1 概述

网络在服务人们、提供共享信息的同时，网络性能已成为了人们关注的重点，这是因为网络服务质量的提示有赖于网络性能的提高， {41%：因而如何更快的获取网络上的信息已经成为了一个研究热点。} {45%：网络坐标系统就是为了提高互联网距离测量效率而提出的。} 自2002年提出 GNP[2]算法以来， 现在已有基于中心式的如 GNP[2]、PIC[10]和基于非中心式的如 NPS[11]、 Vivaldi[3]等时延预测机制， 它们都以如何有效快速的获取网络节点间时延作为研究重点，同时都将网络节点放入 N维的坐标系统，通过计算节点坐标距离来作为网络时延的预测值。

2.2.2 GNP简介

{56%： GNP[2]是最早提出的网络坐标系统，将网络中节点实测时延映射到数学的几何空间上，} 将获取网络节点间时延转化为获取节点坐标间距离。 {61%：其思想是将网络构建在一个几何空间上，如三维欧式空间，并用该几何空间中的一个点作为网络中任何一台主机的位置。}

{46%：在众多网络坐标系统中，GNP是具有代表性的基于锚节点的时延预测机制，其将网络中的主机分为两个部分。} {41%：一个部分是事先选定好的在网络中均匀分布的节点，这些节点被称为锚节点 (landmarks)，} {44%：它们会首选在几何坐标中计算并确定自己的坐标，用于作为其他普通节点坐标定位的参考。} {60%：锚节点保持自己的坐标，同时散布给所有任何想要参与进来的主机节点。} {42%：另一部分是网络中的普通主机，它们通过获得锚节点的坐标，来计算确定自己的坐标，因此该方法具有很强的可扩展性。}

{58%：在 GNP网络坐标系统时，首先要确定网络中一小部分主机作为特殊节点，即为锚节点，} {59%：其作用是在坐标系统中为其它普通主机提供一组必要的参考坐标。}

假设有N个锚节点 P_1, P_2, \dots, P_N {41%： $P_N(N)$ d, d为几何空间维度)，使用 Ping方式可以很容易的获得锚节点间的实测时延 L_{ij} ， {43%： L_{ij} ，从而构造一个 $N \times N$ 的时延矩阵，该矩阵沿对角线对称，则 GNP需要找到一组坐标 X_1, \dots, X_N ，}

$X_2 \cdots \cdots X_N$ 来表示N个锚节点坐标,从而使锚节点间坐标距离 $\|X_i - X_j\|$ 与实测延迟 $L_{i,j}$ 的误差平方总值最小,即下面公式的值最小。

$$[Error]_X(N \times N) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N (\|X_i - X_j\| - L_{i,j})^2 \quad (2.1)$$

在建好了以锚节点组成的网络坐标系统后,在加入普通节点,同样使用 Ping方式可以很容易的获得与锚节点间的实测时延 L_k , 并通过该普通节点与各个锚节点实测时延来确定普通节点的坐标 X , 并使得其与各个锚节点的实测时延与预测时延误差平方总值最小,即下面公式的值最小。

$$[Error]_X = \sum_{k=1}^N (\|X - X_k\| - L_k)^2 \quad (2.2)$$

{45%: GNP系统是集中式的坐标定位系统,具有很强的可扩展性和较高的网络距离预测准确性。} {58%: 对于该系统,每有一个主机加入其中,这个主机就需要测量到所有锚节点的距离,因而可能导致节点负担过于沉重,} {59%: 开销太大,同时系统的距离预测准确度还受到锚节点的分布情况、数量、位置等影响。}

2.2.3 PIC简介

由于 GNP[2]过于依赖锚节点,普通节点需要测量到锚节点的距离,使得锚节点承担了相当的通讯开销,锚节点所能承受的通讯量则成为了网络系统性能的瓶颈,因而有研究者提出了 PIC[10]。PIC是在 GNP的基础之上,改进了锚节点的选择方案,由于PIC不需要固定的锚节点,其系统中任何已经有坐标的节点都可以被其他节点选作锚节点,从而使计算开销和通信开销能够均匀的分摊到各个节点上,同时也有效的避免了因为存在个别锚节点的坐标失效造成的误差。同时,PIC提出了一种锚节点选择机制,其有效性以及系统坐标准确性与GNP几乎一致。

{47%: 与 GNP系统一样,PIC也是采取了将网络节点映射到几何空间的策略,在节点 P加入 PIC系统中时,} {45%: 它从已存在坐标的 N个节点中选择 M(M ≤ d, d为几何空间维度)个节点作为其锚节点,} {41%: 与上述 GNP类似的,同样使用Ping方式可以很容易的获得与锚节点间的实测时延 L_k ,} {46%: 并通过该普通节点与各个锚节点实测时延来确定普通节点的坐标 X ,} {40%: 并使得其与各个锚节点的实测时延与预测时延相对误差平方总值最小,即下面公式的值最小。}

$$[Error]_X = \sum_{k=1}^M ((\|X - X_k\| - L_k) / L_k)^2 \quad (2.3)$$

而当系统中已存在的节点的数量N [M时,其节点的坐标计算方法不同: 此时会将该N个节点都作为锚节点。与上述GNP类似的,对N个锚节点 $P_1, P_2 \cdots \cdots P_N$, 使用Ping方式获得锚节点间的实测时延 $L_{i,j}$,从而构造一个 $N \times N$ 的时延矩阵,并找到一组坐标 $X_1, X_2 \cdots \cdots X_N$ 来表示N个锚节点坐标,从而使锚节点间坐标距离 $\|X_i - X_j\|$ 与实测延迟 $L_{i,j}$ 的相对误差平方总值最小,即下面公式的值最小。

$$[Error]_X(N \times N) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N (\|X_i - X_j\| - L_{i,j}) / L_{i,j})^2 \quad (2.4)$$

{48%: 关于节点P的锚节点选择有着以下三种不同的策略:}

1)随机策略: M个锚节点是从系统中已有坐标的N个节点中随机选择出来的。

2)最近策略： {52%：选择网络系统中离节点P最近的M个节点来作为锚节点。}

3)混合策略： M个锚节点中部分节点是通过随机策略选择出来的，另一部分则是通过最近策略选择出来的。

在最近策略中，PIC提出了两种方法来寻找离节点P最近的M个节点：

方法一： {44%：首先在已知坐标的节点中随机的选择一个节点暂作为目标节点，接着对该节点与其邻居节点的距离进行测量，} {41%：如果存在一个邻居节点离自己的距离更近，则这个邻居节点作为目标节点，然后反复执行这个过程，} 直到无法找到离自己更近的节点。

方法一对节点间的距离的反复测量会对网络带来更多的开销，因此PIC又提出了方法二： {46%：使用坐标计算节点间距离来取代直接测量。} 该方法可以有效的减少网络开销，但该方法只有在新加入的节点已有坐标的前提下才能实现，因此PIC提出： {59%：首先使用随机策略来求出该节点的坐标，接着再使用该坐标来寻找离该节点最近的锚节点。}

PIC通过数据实验表明，不同的锚节点选择策略对于节点坐标的准确度也会有着不同的影响： {57%：在使用随机策略选择锚节点时，对预测长距离的准确度更高；} {59%：在使用最近策略选择锚节点时，对预测短距离的准确度更高；} 而使用混合策略时，对长短距离预测的准确度整体最高，因而在PIC中，都是选择使用混合策略。

2.2.4 NPS简介

NPS[11]是一种分层的非中点式的网络坐标系统。它主要解决的是：在网络坐标分布式计算时遇到的适应性、一致性以及稳定性问题。NPS能够让节点在同一的坐标系上，适应网络环境变化，同时准确的反映网络拓扑结构的变化，并能减少不必要的节点坐标更新。

NPS将网络节点分为三种： {69%：服务器节点，锚节点，和普通节点。} 其中，服务器节点储存着系统参数以及其他节点的信息；锚节点与GNP中的锚节点类似，参与几何空间坐标系的构建，供普通节点作为参考； {51%：其他的节点则为普通节点，这类节点也可以作为其他节点的参考节点。}

NPS的网络坐标系统构建采用了分层次结构，将网络节点分布到0到L层。锚节点作为NPS坐标系统的基础，被放在第0层；而普通节点的放置时，记所在层数为 L_i ，则其选取的参考节点 j 所在层数 L_j 需要满足 $L_i \leq L_j$ ，该约束条件可以有效的避免节点间互相作为参考节点，使得坐标的计算保持了一致性。

NPS中，普通节点加入系统并完成计算坐标需有以下步骤：

{48%：1)在普通节点加入系统时，从服务器节点获得系统参数和参考节点的信息。}

{63%：2)测量获得普通节点与参考节点间的实测距离。}

{42%：3)计算普通节点坐标，使其满足普通节点与参考节点间预测距离与实测距离误差平分总值最小。}

{45%：同时，每隔一段固定时间，节点将会重新计算自身坐标，使其适应网络拓扑环境的变化。}

2.2.5 Vivaldi简介

在众多网络坐标系统中，Vivaldi[3]算法是具有代表性的基于模拟的时延预测机制，其使用物理模拟来预测时延。 {40%: Vivaldi算法将网络节点通过将物理弹簧弹性定律，即} {50%: “ 弹簧在拉伸和压缩时，其动能和势能会相互转化 ” } {47%: ，应用到网络节点坐标系统，将网络节点视为由弹簧相连的点，从而将节点间预测距离误差之和最小化问题模拟转化为} 求弹簧间系统势能最小值问题。

{42%: Vivaldi[3]算法是完全分布式的，节点仅需要获得任何具有坐标的邻居节点的实测时延并可完成自身坐标的更新，} {44%: 当实测时延与预测时延不一致时，则转化视为节点间弹簧发生了拉伸或者压缩，通过弹簧形变使得节点到达一个合适的位置，} {51%: 从而使整个弹簧系统势能最小，即网络坐标系统误差总值最小。} Vivaldi算法的具体实现为：对于节点*i*，获得来着邻居节点*j*的测试时延RTT_{*i, j*}：

1)计算误差权值 ω

$$\omega = e_i / (e_i + e_j) \quad (2.5)$$

2)计算两点距离相对误差 e_s ：

$$e_s = \frac{|\text{RTT}_{i,j} - \|x_i - x_j\||}{\text{RTT}_{i,j}} \quad (2.6)$$

3)更新自身误差估计 e_i ：

$$e_i = e_s \cdot c_e \cdot \omega + e_i \cdot (1 - c_e \cdot \omega) \quad (2.7)$$

4)更新本地坐标 x_i ：

$$\delta = c_c \cdot \omega \quad (2.8)$$

$$x_i = x_i + \delta (\text{RTT} - \|x_i - x_j\|) u(x_i - x_j) \quad (2.9)$$

{52%: 其中 x_i 为节点 i 的坐标， $u(x_i - x_j)$ 为从节点 j 坐标到节点 i 坐标的单位向量， e_i 为节点 i 的误差因子，} {41%: c_e 和 c_c 为调节因子，其取值越大，则节点每次更新的程度越大。}

{42%: Vivaldi[3]算法是公认的具有优秀性能的网络坐标系统构建算法，其优点在于其实现是全分布式的，} {62%: 无需额外设立基站设施，对 Internet 拓扑变化有较好的适应性，并能及时更新网络坐标，将网络坐标系统构建所需的时延探测任务交给应用级业务上，} 减少了网络开支。

2.3 北斗导航系统

2.3.1 相关概念

2.3.1.1 授时

{90%: 确定、保持某种时间尺度，通过一定方式把代表这种尺度的时间信息传送出去，供应用者使用，这一套工作称为授时。}

{100%: 一般来说，要成为一个授时系统必须具备两个条件：} {96%: (1) 授时系统的时间要与国家标准时间保持一致，即溯源到国家时间标准，这是国际电信联盟对授时系统的要求。} {94%: (2) 要求授时系统采用广播的方式发播授时系统的时间，供用户接收使用。}

{100%：作为一个授时系统，不应限制用户的使用数量。} {100%：接收相同授时信号的不同用户，均可以实现不同用户本地时间与同一时间尺度的同步。}

从国内参考文献来看，一般未对时间传递、时间频率传递、时间比对与授时概念进行严格区分。

1.2守时

{97%：标准时间的产生和保持（守时）是时间服务工作的核心，主要包含三个方面的内容：}

{100%：首先是使用什么样的“钟”来产生并保持一个稳定的时间尺度。} {96%：当前国际上大多数的守时实验室基本采用商品原子频率标准（包括氢脉泽、铯原子钟、铷原子钟等）来组成守时钟组。}

{100%：其次是时间尺度如何产生，以及将什么时间作为国际标准参考时间，如何得到标准参考时间。} {100%：关于地方原子时 $TA(k)$ 的产生，各个实验室根据自身的情况采用不同的方法。} {100%：而国际上，始于1972年的用加闰秒的新协调世界时（UTC）作为国际标准参考时间至今已30多年，} {100%：但是究竟用 UTC 还是直接用 TAI 或是重新定义新的时间尺度，是近年来国际时间频率领域正在激烈争论的问题。}

{100%：第三个方面是本地时间比对和远距离时间同步。} {100%：本地时间比对主要是本地守时钟时差的比对测量，当前采用的技术主要是时间间隔和相位比对测量方法。} {100%：远距离时间同步主要用于将全球的钟和不同的时间尺度同步到国际标准时间上来。}

1.3时间同步

{93%：时间同步是指各网络节点设备、应用系统的时钟使用同一时间参考基准——协调世界时（UTC），} {100%：通过某种方式使其时钟的时刻和时间间隔与 UTC 同步。}

2.3.2精密授时

2.3.2.1GNSS卫星授时

北斗授时方法主要包括 RDSS（Radio Determination Satellite Service，卫星无线电测定业务）单向授时、RDSS双向授时、RNSS（Radio Navigation Satellite Service，{50%：卫星无线电导航业务}授时和共视法授时。}

2.3.2.1.1北斗RDSS单向授时

{91%：北斗卫星 RDSS业务广播授时信号的载波频率为2491.75 MHz，为连续帧信息结构，} {93%：在时域上分为超帧和帧，固定帧长，1个超帧周期为1分钟，} {83%：含1920帧，1秒包含32帧，帧周期为31.25 ms。} {94%：RDSS业务的全部单向授时信息由各帧中4bits“广播”信息拼接完成。} {82%：每1超帧（1分钟）的第1帧~129帧就将该分钟的全部单向授时信息播发1遍；} 每1分钟播发授时信息8遍； {100%：授时信息的内容每1超帧更新1次。}

{92%：RDSS业务每超帧第1帧~第129帧的导航电文与授时相关的参数内容有：} {49%：表示本超帧第1帧所对应的 BDT时刻、表征 BDT与 UTC之差的 UTC闰秒、BDT与 UTC之} {64%：时差、本超帧授时数据所对应的地面 RDSS发射天线的编号、地面站天线至卫星的传播时延（含电离层和} {84%：对流层时延）、信号传输改正模型参数、系统单向设

备零值变化量、当前波束工作状态等。}

{51%：北斗卫星 RDSS单向授时过程是在 BDT控制中心主原子钟的监控下，发播工作原子钟产生卫星导航信号的频率、编码速率、相位、导航电文，} 由发射设备发送到北斗卫星， {95%：卫星转发器将授时信号下行传递到用户接收终端，} 终端解算输出1 PPS（秒脉冲）和 TOD（Time Of Day）时间信息， 完成 RDSS单向授时。

RDSS单向授时示意图见图2：

{50%：小于1秒的小数秒部分的BDT的时间传递过程如图3所示：} {80%：在中心站出站信号某一“帧时标”与其前一个 BDT整秒时刻（即1 PPS）的时差为，} {69%：“帧时标”经过总时延（包含系统设备单向零值延迟、上行延迟、下行延迟（包括大气层延迟）、用户设备单向零值延迟）之后，} 用户观测/提取“帧时标”信号的前沿。 {76%：用户以本地时钟1PPS作为时间测量计数器的开门信号，“帧时标”的前沿作为关门信号，可测得二者的时差 }。} 那么，用户本地时钟与BDT的时间差 即为：

{76%：移相调整本地钟输出的1PPS，使时间差 为零，调整后的本地1PPS就与BDT的1PPS实现时间（相位秒部分）同步。} {46%：北斗卫星RDSS单向授时精度优于100ns，而且用户数量不受限制，完全可以满足绝大多数时间用户精度需求。}

2.3.2.1.2北斗RDSS双向授时

{64%：北斗RDSS双向授时是一种建立在应答测距定位业务基础上的高精度授时方法。} {60%：由于北斗 RDSS单向授时精度受卫星星历误差、接收终端位置误差、大气层延迟误差、北斗授时信号发射时刻改正残差等诸多不确定性因素影响，} {48%：难以准确计算、改正卫星中心站到用户终端信号传播时间延迟，限制了北斗 RDSS单向授时精度。} {71%：为满足更高授时精度需求，在北斗RDSS应答测距定位业务基础上提出了RDSS双向授时方法。} {50%：该方法采用双向比对测量确定信号单向传播时间延迟。} {86%：双向法授时要求用户终端同时具备接收和应答发射的能力。} {74%：北斗RDSS双向法授时示意图见图5，时间延迟测定改正原理图见图6。}

{76%：中心1PPS代表中心站控制信号发射的BDT时刻 t ，用户1PPS代表用户终端机内时钟的某1整秒 $T(t)$ 时刻，二者的钟差为 }。} {85%：北斗第 n 帧询问信号参考时标与北斗时某1PPS（例如与整分时刻对应的秒脉冲）之间的时间间隔为 n 个帧周期 }。} 如前所述，RDSS的帧周期为31.25ms，即 ； {86%：同时 也即该帧号对应的北斗时间（小于1整分的部分）。} {89%：与此同时，用户终端接收到中心站控制系统播发的时间帧（第 n 帧）询问信号，} {92%：并测出收到的第 n 帧询问信号参考时标与本机钟整秒信号1 PPS的时间间隔；} {91%：同时，用户终端立即向中心控制系统回发相应信号，} {95%：中心控制系统测出第 n 帧信号的往返时间值，} {96%：并算出该信号由中心发出至用户机收到的正向传播时延，} {75%：再将发送给该用户作为双向定时时延改正值。} {60%：由于可以由用户直接测定，因此只要给出传播时延，就可以得出用户终端时钟与 BDT钟差，} {59%：调整本机时钟，从而完成用户终端与中心站 BDT的时间同步。}

{68%：双向法授时采用了往返路径相同，方向相反，影响单向授时的正向传播时延误差和其他各项误差就可以相互抵消，} {68%：残差可以忽略，大大削弱了各项误差的影响，因此，北斗卫星 RDSS双向授时精度可达20 ns，} 但用户数量受到限制。

2.3.2.1.3北斗RNSS单向授时

{40%：北斗RNSS单向授时的本质就是利用单站伪距/相位观测值以及卫星星历，来估计接

收机钟差，从而实现授时。} {46%：在接收终端位置已知的情况下，只需1颗卫星即可实现授时。} {47%：在测站坐标未知的情况下，授时过程与伪距单点定位/精密单点定位过程相同。}

对于卫星*i*的伪距观测方程如下：

式中： ρ_i 为伪距观测值； r_i 为测站与卫星间距离； δt_i 为卫星钟差； δt_r 为接收机钟差； δt_{at} 为对流层延迟； δt_{io} 为电离层延迟； δt_n 为观测噪声和其他误差。

由于卫星坐标和钟差可通过星历获得，电离层和对流层延迟均可通过模型改正或削弱。因此，在接收终端坐标已知的情况下，只需要1颗卫星，通过作差的方式可直接求出接收机钟差。 {46%：在接收终端坐标未知的情况下，至少需要4颗卫星，通过平差的方法实现授时。}

由于伪距观测值精度不高，通常在30cm~3m，因此采用伪距进行RNSS单向授时的精度约在50ns。

{44%：可以采用相位平滑伪距或载波相位观测值来提高授时精度，授时精度可以达到10ns以内。}

2.3.2.1.4北斗/GNSS共视法授时

{81%：北斗卫星共视法时间传递就是在位于不同地点的两个（或多个）观测站，} {42%：用北斗卫星信号接收终端同时观测同一颗或多颗北斗卫星的同一时标信号，} {84%：测量本地时钟信号与该时标信号的伪距/时间差，各自获得本地时钟与控制卫星时标信号的BDT差值，} {74%：经观测结果传递交换和再处理，实现异地两两观测站之间的高精度时间传递。} {76%：其技术关键在于可以消除或削弱北斗卫星（RDSS/RNSS）单向授时过程中若干共性误差，大大提高远距离卫星时间传递的精度。}

{45%：该方法类似于差分定位技术，通过相邻测站观测值间的相关性，通过作差的方法消除或削弱其中的共性误差，} 从而获得高精度的相对定位精度。

其原理如下图所示：

若 A、B 两站放置的接收机在同一时刻观测到卫星*i*的伪距为 ρ_{Ai} 和 ρ_{Bi} ，可得其接收机钟差分别如下如下：

其中： ρ_i 为伪距观测值； r_i 为测站与卫星间距离； δt_i 为卫星钟差； δt_{Ai} 、 δt_{Bi} 为接收机钟差； δt_{at} 为对流层延迟； δt_{io} 为电离层延迟； δt_n 为观测噪声和其他误差。

作差可得：

{46%：可以根据接收机位置和卫星星历计算得到，卫星钟差可完全消去，由于测站间距离较近，因此卫星轨道误差、对流层延迟和电离层延迟相关性较高，} 作差后可以消除其大部分影响。 因此，该方法可以获得高精度的相对钟差，从而实现时间传递。

使用简单的单通道伪距接收机，即可保证实现两站优于10ns的时间同步精度。 {45%：当两站的基线长度不大于100km 时，两站可以实现优于 2ns 的时间同步精度。} {100%：目前，BIPM 已将共视作为全球70多个守时实验室之间时间比对的主要手段之一。}

{43%：基于载波相位观测值的多通道共视接收机，由于其精度比伪距观测值更高，其时

间传递精度优于 1ns。}

{42%：标准共视的时间间隔为16分钟，其中13分钟用于数据采集，2分钟用于数据处理，1分钟等待下一个共视时刻的到来。} 因此，共视时间传递不具有实时性，严格意义上讲不属于授时。

虽然共视方法有很多局限性，但是不可否认的是其时间传递精度较高。 许龙霞提出了一种基于共视原理的授时新方法，该方法在多个坐标已知的基准站布设接收机，基准站本地时间与标准时间保持同步， {44%：监测卫星的系统时间与标准时间的偏差，并通过网络实时发送给用户使用。} {45%：用户使用该偏差数据后，即可获得与国家标准时间的偏差。}

其原理如下图所示：

{41%：假设有n个基准站，每个基准站与标准时间中心通过卫星双向时间频率传递与标准时间 UTC(NTSC)保持同步。} {55%：同步后的基准站监测 GNSS 卫星授时信号，进而获得每颗卫星广播的系统时间与标准时间的时差 $UTC(NTSC) - GNSST$ 。} {46%：系统经综合处理，将含有标准时间信息的时差数据广播给用户。} {55%：用户通过接收导航信号得到用户本地时间与导航系统的系统时间的时差 $T(u) - GNSST$ ，} {47%：然后利用接收到的标准时间与 $GNSST$ 的时差 $UTC(NTSC) - GNSST$ 将用户本地时间修正到标准时间，} 实现用户本地时间与标准时间的同步。

$$T(u) - UTC(NTSC) = (T(u) - GNSST) - (UTC(NTSC) - GNSST)$$

{73%：该方法解决了传统共视时间传递存在的实时性问题。} GNSS共视时间传递中，由于用户接收机时钟的不稳定需要对卫星连续跟踪一段时间，通过对跟踪数据进行平滑处理尽量减少接收机时钟抖动、测量噪声等随机误差的影响。而共视授时方法利用了国家标准时间和GNSS系统时间的高稳定性，不受跟踪时长的限制，向用户实时广播授时模型参数，解决了实时性问题。其次，与共视时间传递的数据交换方式不同，基于共视原理的授时新方法将授时模型参数信息以广播的形式发布，所有用户均可接收，因此用户数量不受限制。此外，用户只需配置单向授时的设备即可获得共视时间传递的纳秒级授时精度，是一种具有广泛应用前景的授时方法。

2.1.5 GNSS全视法时间传递

{78%：由于共视法时间传递时两地面站必须对同一颗导航卫星可见，并且随着两站基线的增加，} {86%：大气误差相关性差等因素，在某种意义上限制了该技术的推广应用；} {91%：随着IGS组织的发展，它可以提供 3~5厘米的卫星轨道精度以及0.1~0.2 ns的卫星钟差。} {100%：双频测地接收机的普及对于实时消除由于电离层引起的传播误差比利用模型改正有很大程度的提高。} {100%：不仅如此，其它误差修正方面也有很大程度的改进，以这些技术为基础，全视法以及精密单点定位方法有了很大的发展，} {100%：时间比对的精度比上述两种方法有了明显的提高。} {97%：从 2006 年开始全视法已经在计算UTC/TAI方面替代了共视法。} {100%：精密单点定位技术也已经有了广泛的应用。}

2.4 软件定义网络（SDN）

2.4.1 SDN起源与发展历史

软件定义网络（SDN）技术的

{51%：SDN起源于斯坦福大学的CleanSlate项目[18]，此项目初衷是改变传统网络架构。}

之后学生 Martin负责了一个关于网络安全的项目，设计了一个集中控制器，通过此集中控制器可以很便利地配置基于网络流的策略，可以把一些策略下放到其他网络设备中，进而实现了对整个网络的控制。2008年受此项目启发，Nick教授等人提出Open Flow协议，并发表《Open Flow: Enabling Innovation in Campus Networks》的论文，首次公开详细地描述了Open Flow 协议。{58%：2009年由于Open Flow协议给网络带来的灵活性，Nick教授及其团队进一步提出了SDN的概念和架构。}{42%：这一年SDN技术同时被选为年度十大前沿技术之一[19]，从此SDN技术取得了学术界和工业界越来越多的认同和大力支持。}

{48%：2009年Open Flow1.0版本被发布，它是具有划时代作用的首款可用于商业环境的南向协议。}{48%：2011年在Nick教授等人的推动下，成立了开放网络基金会，它的主要工作职能主要是大力发展和推动SDN架构及其技术的相关规范性工作。} 目前开放网络基金会有近100家成员。2012年开放网络基金会发布了SDN白皮书。{43%：在SDN白皮书中提出了SDN三层模型得到了很多同业人员的大力支持和认同。} 同年，谷歌公司广域网络被改造成使用 Open Flow协议的 SDN网络，此案例说明 Open Flow协议不仅是一个在学术层面的理论模型[20]，而是一个可以在商业实际环境中使用的技术方案。之后VMware收购Nicira公司，把网络相关的功能模块从硬件中分离开来实现软件化，这也是SDN迈向商业化的重要标志。2016年召开全球SDNFV技术大会，来自全球的企业和组织讨论企业网和云数据中心相关问题和技術。此时意味着SDN技术不是空洞的理念，而是正在迎合真正的需求，SDN产业化时代正在来临。

当下 SDN技术已经成为了网络领域中最热门的方向之一，很多互联网大公司都在 SDN技术研究中投入了大量的精力，{44%：思科等传统基础网络设备厂商也在研发 SDN控制器和 SDN交换机。} 随着SDN技术在不断研究发展中，SDN定义越来越宽泛，不再那么明确。{40%：当下普遍来说，那些可以对网络进行编程的网络架构都可以看作SDN范畴。} 而且在SDN发展中，为了实现上述理念而开发的技术和协议也是多样的，除了 Open Flow协议还涌现出很多类似技术例如 OF- CONFIG协议，所以 SDN也渐渐出现了不一样的发展路线[21]：

{41%：第一种是使用 OpenFlow协议，转发层和控制器层互相分离、采用集中控制方式，让网络设备逐渐标准化，} 所有控制空能模块集中到控制器，并以软件方式存在，这样突破过去比较封闭的网络状态。此中发展路线主要在创业公司和研究机构的比较多，从路线上看是狭义的SDN定义。

第二种是传统网络的设备制造商在新技术变革带来的市场压力下，他们为了能在市场中继续拥有绝对优势地位而发展 SDN技术来改造自身的产品，还有是很多运营商想采用新技术和新架构改造自身网络，但同时也想尽可能的保留当下的设备头再来尽可能减少损失，所有期望能对现有网络及相关设备进行剪辑师的发展和过度，所以演变出一种更为宽泛的SDN概念，这些都可以是广义 SDN定义。

最后，还有一种路线，此路线使用现有传统网络作为底层设备，然后在基础设备上采用虚拟化技术来衍生出逻辑网络层，实现网络虚拟化，从本质上说仍然是软件定义虚拟网络的范畴，此方案一般称为 Overlay方案。

不同的SDN演进发展形成了不同的技术风格和路线，但是集中控制、开放可编程接口是共有的特性，具体的异同如下表所示：

表2-1 SDN研究方案比较

发展路线优点缺点

狭义SDN方案网络管理比较简单难度高、技术不成熟

广义SDN方案有利于抱回现有网络设备投资只能使用支持的服务，快速扩展网络功能受限

Overlay方案网络只需要IP可达，不需要其他的大规模改变由于忽略物理网络，流量工程等无能为力

2.4.2 SDN架构

{61%：一句ONF给出的定义，软件定义网络（SDN）是一种新型网络架构，。} 软件定义网络在逻辑上采用集中控制方式，且把网络转发层和网络控制层分离，形成了一个课表成的网络架构并具有可编程开放接口，使得应用层能够不依赖于底层设备。这样能够使得网络架构打破底层物理硬件设备的限制，实现上层软件与底层硬件设备相互独立发展，底层设备标准化的发展得到提升，软件平台化进程也有了长足的发展。就可以像升级软件一样来对网络进行修改和配置，提升了网络功能的部署速度，加速了快速部署和敏捷开发的进程。

2.5 OpenVSwitch

2.6 mininet

2.7 RYU控制器

2.8 本章小结

时空大数据平台建设方案

3.1 基于SOA的时空大数据平台体系

3.2 多层协同的架构体系

3.3 功能平台的服务体系

3.4 开放方向的数据体系

3.5 高效运行的管理体系

3.6 时空大数据标准体系

3.7 本章小结

系统原型设计与实现

4.1 SDN

4.2 Mininet和RYU安装

4.3 OpenVSwitch 协议解读

4.4 网络风暴解除，ARP回复

4.5 原型设计与实现

4.6 本章小结

时空网络坐标系性能分析与优化

5.1 网络坐标系

网络在服务人们、提供共享信息的同时，网络性能已成为了人们关注的重点，这是因为网络服务质量的提示有赖于网络性能的提高，{41%：因而如何更快的获取网络上的信息已经成为了一个研究热点。}{45%：网络坐标系就是为了提高互联网距离测量效率而提出的。}自2002年提出 GNP[2]算法以来，现在已有基于中心式的如 GNP[2]、PIC[10]和基于非中心式的如 NPS[11]、Vivaldi[3]等时延预测机制，它们都以如何有效快速的获取网络节点间时延作为研究重点，同时都将网络节点放入 N 维的坐标系统，通过计算节点坐标距离来作为网络时延的预测值。

5.1.1 随机延迟污染现象

{49%：在网络系统中，时延可以分为两种，一种是单向时延，另一种是往返时延(Round Trip Times, RTT)。}{60%：单向时延指的是一个报文或分组从一个网络的一端传送到另一个端所需要的时间，往返时延则比单向时延多了一个返回}

{41%：时间，即一个报文或分组从一个网络的一端传送到另一个端，另一个端接收后发生反馈的一个报文或分组回这一端所需要的时间。}{47%：单向时延和往返时延都由排队时延，传播时延，传输时延，本文主要考虑的是往返时延RTT。}

造成网络时延长的因素有很多：{45%：网络本身的物理性能下降，如线路老化，路由器处理能力下降；}节点发送的报文太长，包太大；网络环境拥塞，网络负载分担不均匀。当网络环境拥塞时，原先某个节点发送的包到另一个节点后，其反馈包由于路由器路径选择，可能选择了一条不一样的路径回来，这会导致实测时延急剧增大，而这种情况则称之为随机延迟污染[7]。

由于网络坐标系统的建立依赖于实测时延，当实测时延出现随机延迟污染现象时，该实测时延不具有可靠性，{52%：并严重会降低网络坐标系统的准确性，从而降低了其实用性。}

5.1.2 三角不等式违例现象

在网络中，由于网络自身物理因素，负载均衡，路由策略等因素，使得节点间时延经常出现三角不等式违例现象[5]。

图1三角不等式违例现象

如图所示，节点AB间时延与节点BC间时延总和为200，小于节点AC间时延300。{48%：而在网络坐标系统中，将这三节点放入几何坐标系(以欧式坐标为例)时，}欧式坐标中无法找到三个点，使他们间距离同时满足 ABC间的距离，而节点会努力调整自身坐标，使节点间预测时延与实测时延的总体误差最小，其结果就是 AB与 BC这两边会被拉长，AC会被缩短，同时节点 ABC会不停在一个区域内振荡。{53%：同时TIV现象在实际网络中广泛存在，这会降低网络坐标系统的准确性。}

在 Vivaldi算法中,将节点视为由弹簧相连的点,则在出现 TIV现象时, AB与BC这两边会被拉长, AC会被缩短,节点 ABC均会偏离实际位置,不停抖动,这是由于Vivaldi算法中对坐标的一次更新仅参考一个邻居节点的一次 RTT与邻居节点误差因子 e , {43%:缺少对 RTT可靠性的判断,从而使系统动荡,准确性下降。}

5.2 随机延迟污染现象及抑制方法

{41%:网络坐标系统的构建依赖于节点间的实测时延,但由于网络中存在的随机延迟污染现象降低了网络坐标系统的实测时延的准确性。} {54%:常用的解决办法是对节点间实测时延进行平均化处理,再由处理过的平均值的时延矩阵作为构建网络坐标系统的参考。} 然而经由平均化处理过的平均值,无法反映实际网络实测时延的变化,同时也使节点坐标偏离了正确的位置,扭曲了节点定位。

5.2.1 随机延迟污染现象数据分析

本文的时延数据来自文献[15]的数据集,该数据集有1953个节点,共97251194条记录,这些节点都有至少9个邻居节点,通过这些数据来进行随机延迟污染分析。节点延迟分布如下图2所示:

图2随机延迟污染数据分析

其中时延在0-100 ms之间的记录有约8830万条,约占总体90%,剩余约600万条记录在100-500 ms之间,有约90万条时延记录在500-1000 ms之间,1000-2000 ms之间的有约64万条记录,而3000以上也有近100万条记录,这说明了随机延迟污染是普遍存在的。{45%:而随机延迟污染现象会使以实测时延构建的网络坐标系统的准确性下降,节点坐标动荡,性能下降。} 因此,抑制随机延迟污染,减少实测时延不正常波动,对于提升网络性能有着重要意义。

5.3 抑制随机延迟污染的方法

5.3.1 MP-Filter抑制方法

Harvard 大学的 J.Ledlie 等人研究延迟污染现象并提出了MP-Filter[8]的抑制方法。该方法的主要思想是基于滑动窗口滤波,其主要步骤如下: {52%:根据实测时延顺序,放入一个滑动窗口,其滑动窗口长度 W ($W=4$),并将窗口 W 内的时延按进行升序排序,} 选择一个合适的百分位 $P=0.25$,然后选择第 N 位($N=P*W=1$, {62%:即最小时延值)时延用于更新坐标。} {78%:在进过仿真实验后, $P=0.25$ 和 $W=4$ 时, MP-Filter能有较好的抑制波动并且保持原始实测时延的统计特征,} {48%:其网络坐标系统的各方面性能指标是整体最优的。}

然而 MP-Filter的抑制方法在抑制随机污染方面有一定成效,但该方法舍弃了部分实测时延,没有保持原始的实测时延,这会使得其处理结果偏离实测时延,扭曲了节点坐标,使得准确度下降[9]。

5.3.2 TO-Filter抑制方法

TO-Filter抑制方法(Timeout Filter)是本文提出的一种抑制随机延迟污染的方法,其思想参照了 TCP超时重传机制,通过计算测量来获得当前 RTT的一个估计值,并以该 RTT估计值为基准来判定是否出现了随机延迟污染。原理是:对于节点 i ,获得来自节点 j 的每一个 $RTT_{i,j}$,计算 $MeanRTT_{i,j}$:

$$[\text{MeanRTT}]_-(i, j) = (1 - \alpha) [\text{MeanRTT}]_-(i, j) + \alpha [\text{RTT}]_-(i, j) \quad (3.1)$$

MeanRTT_{i, j}是RTT_{i, j}的指数加权移动平均(Exponentially Weighted Moving Average, EWMA), 这种平均能很好的反映网络的当前拥堵情况, 其中 α 的参考值是 $\alpha=0.125$ (即1/8)。同时除了计算RTT的估计值, 还要计算RTT的变化。定义DevRTT_{i, j}, 用于估计RTT_{i, j}与MeanRTT_{i, j}的偏离程度:

$$[\text{DevRTT}]_-(i, j) = (1 - \beta) [\text{DevRTT}]_-(i, j) + \beta |[\text{RTT}]_-(i, j) - [\text{MeanRTT}]_-(i, j)| \quad (3.2)$$

DevRTT_{i, j}是RTT_{i, j}与MeanRTT_{i, j}的差值的EWMA, 这里 β 的推荐值为1/4即 $\beta=0.25$

当节点i, 获得来自节点j的一个RTT_{i, j}时, 其估计值为MeanRTT_{i, j} + 4 DevRTT_{i, j}。首先进行随机延迟污染的检测, 如果RTT_{i, j} > MeanRTT_{i, j} + 4 DevRTT_{i, j}, j则视为出现了随机延迟污染, 此时的RTT_{i, j}可能是极大的, 不具有可靠性, 此时先计算DevRTT_{i, j}与MeanRTT_{i, j}的值, 然后让RTT_{i, j} = MeanRTT_{i, j}, j作为输出。

{50% : 5.4 网络坐标中三角不等式现象级抑制方法}

5.4.1 T-Vivaldi TIV感知的坐标系统

T-Vivaldi TIV感知的坐标系统是对三角不等式违例(TIV)现象[5]造成网络坐标系统振荡而提出的一种对TIV进行检测和抑制的方法。{64% : 其主要思想是用三角不等式条件, 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来检测违例边, } 并使用违例系数度量其违例程度。{98% : 根据该系数的值抑制违例边对坐标的更新, 从而达到了抑制TIV对坐标系统的影响的目的。}

具体步骤是: {47% : 对节点i获得来自节点j一个RTT_{i, j}, 如果在节点i的邻居节点中存在节点k与节点j也是邻居, } {50% : 则获取节点i, 节点j, 节点k间的时延, 并计算违例系数: }

$$\lambda = [\text{RTT}]_-(i, j) / ([\text{RTT}]_-(i, k) + [\text{RTT}]_-(k, j)) \quad (4.1)$$

如果 $\lambda > 1$ 时, 则认为RTT_{i, j}是违例边, 应减少其更新坐标的程度, 对坐标的更新系数要乘上1/ λ 否则当 $\lambda \leq 1$ 时, 则认为RTT_{i, j}不是违例边。

尽管T-Vivaldi算法对抑制坐标抖动有一定成效, 但由于TIV现象可以存在于任意三个节点之间, 因此仅仅选择部分的邻居节点并无法检测到所有TIV现象; 而若要检测所有邻居节点, 则开销会变得十分巨大。

5.4.2 抖动感知的慢启动抑制算法

{54% : 文献[6]则提出一种基于坐标抖动感知的慢启动抑制方法, 将Vivaldi算法归结成非线性方程组的迭代求解算法, } {92% : 并且基于方程组的矛盾性提出迭代因子的自适应估计问题。} {89% : 其原理是将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, } {100% : 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; } {100% : 收敛完成后, 超步会减小迭代步长以抑制坐标抖动。}

其感知方法[6]为： {58%：在某个时间切片中，在节点的某个时间切片中，计算该节点与邻居节点坐标距离与测量距离误差的均值为单位化误差，} 如节点 i 在时刻 t 的坐标为 $x_i(t)$ ， $Neighbor(i)$ 为节点 i 的邻居节点个数，则单位化误差 $err(t)$ 为下列公式(4.2)所示：

$$err(t) = \frac{1}{|Neighbor(i)|} \sum_{j \in Neighbor(i)} \sqrt{((RTT(i, j) - \|x_i(t) - x_j(t)\|)^2)} / |Neighbor(i)| \quad (4.2)$$

{57%：该方法随着算法的不断运行，节点的单位化误差会逐渐下降，直到进入一个反复振荡的抖动状态。} {58%：如果节点 i 在某个时刻的单位化误差增大，即 $err(t) > err(t-1)$ 时，则视为发生了抖动；} {100%：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。}

{57%：在执行前，定义了最小步长迭代因子 ϵ 以及最大迭代因子为 1，定义步长增长因子为 1 为放大器增加迭代步长，} {55%：定义步长衰减因子 h 为衰减器减小迭代步长， cc 为 Vivaldi 算法中步长调节因子。}

对于一次具体超步步骤[6]是：

子步Sup-step1: 计算单位化误差 $err(t)$

子步Sup-step2: 若 $err(t) > err(t-1)$ ，跳转到子步Sup-step4

子步Sup-step3: 让 $cc = \max(\epsilon, cc \times h)$ ，跳转到子步Sup-step5

子步Sup-step4: 让 $cc = \max(1, cc \times l)$

子步Sup-step5: {77%：循环执行子步Sup-step至本轮超步结束，并跳转到子步Sup-step1}

子步Sup-step: 执行Vivaldi算法来更新节点坐标。

经过仿真实验，在 $l=0.5$ ， $h=1.1$ 时，该方法有着良好抑制抖动能力， {57%：能将节点坐标抖动的程度降低83.5%以上，同时具有较快的收敛能力。} 但其算法仅考虑了TIV现象造成的抖动，对于其他如随机时延污染现象以及网络攻击造成的抖动没有防范措施，这可能使得节点坐标的更新过早的进入抑制状态，或者迟迟无法收敛，使网络坐标系统的准确性下降。

5.4.3 能量更新抑制方法

文献[16]则提出了一种能量更新抑制方法(Energy Method)，该方法的主要思想是：预先设置一个开始窗口 WS (Window Start)和一个当前窗口 WC (Window Current)，分别用于保存节点坐标的历史记录， {51%：区别在于 WS 只保存最开始的 $n1$ 个节点坐标记录，当 WS 已经储存了 $n1$ 个节点坐标记录后将不再更新，} 而 WC 是不断更新的，它储存着最新的 $n2$ 个节点坐标记录。 {55%：将 WS 和 WC 中储存的节点坐标系统称为系统级坐标，则只有在系统级坐标的变化程度超过规定数值 Y 时，才对节点更新其节点应用级坐标 X 。} {50%：其中系统级坐标的作用是抑制TIV现象造成的节点坐标抖动，而应用级坐标 X 才是用于预测时延的。} 该方法用于计算系统级坐标的变化程度的公式为能量模型 $e()$ ，具体如下图公式所示。

$$e(W_S, W_C) = \frac{(n_1 - n_2) / (n_1 + n_2) \left(\frac{2}{(n_1 - n_2) \sum_{i=1}^{n_1} (n_1 - 1) \sum_{j=1}^{n_2} \|a_i - b_j\| - 1 / [(n_1 - 1)^2 \sum_{i=1}^{n_1} (n_1 - 1) \sum_{j=1}^{n_2} \|a_i - a_j\| @ - 1 / [(n_2 - 1)^2 \sum_{i=1}^{n_2} (n_2 - 1) \sum_{j=1}^{n_2} \|b_i - b_j\|] \right)}{(4.3)}$$

其中 a_i , b_j 分别为 WS 和 WC 中储存的节点坐标, 只有当 $e(W_S, W_C)$ 的值超过规定数值 Y 时, {52%: 才对节点更新其节点应用级坐标 X , 其 X 的取值为当前窗口 WC 中储存的所有坐标的平均值。}

{46%: 能量更新抑制方法[16] (Energy Method) 通过计算最初和最近的节点坐标记录变化程度, 来对用于预测时延的应用级坐标的更新进行限制,} 从而减少了因 TIV 现象造成的坐标盲目更新。然而该方法存在着两大缺点: 一是对于不同的网络环境, 其变化程度阈值 Y 的值也需要随着进行调整来适应网络环境, 这增大了能量更新抑制方法适应不同的网络环境的难度; 二是其用于计算系统级坐标的变化程度的能量模型 $e()$ 计算过于复杂, 开销太大, 有较高的时间复杂度[9]。

5.4.4 稳定抑制Vivaldi算法

稳定抑制Vivaldi算法是本文提出的一种基于Vivaldi算法的改进算法, 其主要目标是抑制随机延迟污染现象与TIV现象。该算法主要分为三个部分, 第一部分是对随机延迟污染的检测, 第二部分是对网络坐标抖动的检测, 第三部分是对坐标抖动进行抑制。其思想参照了TCP超时重传机制, 通过计算并保存节点获得的每一个实测时延的估计值, 通过比较实测延时与估计值, 来判断是否出现了随机延迟污染, 同时通过计算预测延时与均值估计值的差值均值来判断是否出现了坐标抖动; 在抑制抖动方法中, 选择一种递减函数作为抑制函数来减少坐标更新程度。

5.5 基于Vivaldi算法的抑制方法

针对在实际网络坐标系统中, 由于各种随机延迟污染以及三角不等式违例现象造成网络坐标振荡, 本文提出了一种基于 Vivaldi 算法的稳定抑制 Vivaldi 算法 (Stable inhibition Vivaldi)。{40%: 该算法分为三部分, 第一部分是对随机延迟污染的抑制, 该部分使用了本文提出的 TO-Filter 抑制方法,} 第二部分是对网络坐标抖动的检测, 第三部分是对坐标抖动进行抑制。

5.5.1 检测随机延迟污染

该部分使用了本文提出的 TO-Filter 抑制方法: 当节点 i , 获得来自节点 j 的一个 $RTT_{i,j}$, j 时, 首先进行随机延迟污染的检测, 如果 $RTT_{i,j} \leq \text{MeanRTT}_{i,j} + 4 \text{DevRTT}_{i,j}$ 则视为出现了随机延迟污染, 此时的 $RTT_{i,j}$ 可能是极大的, 不具有可靠性, 此时先计算 $\text{DevRTT}_{i,j}$ 与 $\text{MeanRTT}_{i,j}$ 的值, 然后让 $RTT_{i,j} = \text{MeanRTT}_{i,j}$ 进入下部分, 从而减少不可靠的延迟 $RTT_{i,j}$, {50%: j 对节点 i 坐标更新的影响, 从而减轻网络坐标系统的抖动。}

当 $RTT_{i,j} \leq \text{MeanRTT}_{i,j} + 4 \text{DevRTT}_{i,j}$ 时, 则视为此时网络较为稳定, 此时三角不等式违例现象是造成网络坐标振荡的主要因素之一。{53%: 对此, 本文提出了一种坐标抖动感知方法。}

5.5.2 坐标抖动感知方法

该坐标抖动感知方法参考了文献[6]的坐标抖动感知方法，文献的坐标抖动感知方法使用了单位化误差的计算方法，{58%：在某个时间切片中，在节点的某个时间切片中，计算该节点与邻居节点坐标距离与测量距离误差的均值为单位化误差，} 单位化误差 $err(t)$ 为上文公式(4.2)所示。

{57%：该方法随着算法的不断运行，节点的单位化误差会逐渐下降，直到进入一个反复振荡的抖动状态。} {58%：如果节点*i*在某个时刻的单位化误差增大，即 $err(t) > err(t-1)$ 时，则视为发生了抖动。}

本文对该坐标抖动感知方法进行改进，将上述公式(4.2)中的 $RTT_{i,j}$ 替换为 $MeanRTT_{i,j}$ ，替换的目的是减少个别不可靠的 RTT 对感知方法的准确性的影响，同时不采用时间片的方法，而是使用计数的方法，当累计获得 N 个 RTT 后才进行 err 的计算($N = Neighbor(i)$ ，即节点 i 的邻居节点数)，{51%：让此时节点 i 的坐标为 $x_i(t)$ ，本文的坐标抖动感知方法为：}

$$\|err\|_t = \left(\sum_{j \in Neighbor(i)} \sqrt{((\|MeanRTT\|_t(i, j) - \|X_i(t) - X_j(t)\|)^2)} \right) / Neighbor(i) \quad (5.1)$$

当 $err_t \geq err_{t-1}$ 时，视为发生了抖动，则会开始执行第三部分的抑制算法。要注意的是，坐标抖动感知方法只有在 $RTT_{i,j} \leq MeanRTT_{i,j} + 4 \cdot DevRTT_{i,j}$ 时才执行，否则需要重新计数。

5.5.3抑制算法

5.5.3.1抑制算法的原理

{40%：在三角不等式违例现象[5]中，三个相连节点间的违例边是最长的，} 而在实测时延中，越长的时延，越有可能成为违例边，因此，本文的抑制算对于越长的时延，{54%：越是减少其对网络节点坐标更新的程度。} 具体是在Vivaldi算法中，对公式(2.8)中的 δ 乘上一个系数 d ，即：

$$\delta = c_c \cdot \omega d \quad (5.2)$$

$$d = 1 / ((1 + \ln(1 + RTT/100))) \quad (5.3)$$

$\ln(x)$ 是以自然常数 e 为底的对数。而 d 的值在 $(0, +\infty)$ 上单调递减，值域为 $(0, 1)$ 。

5.5.3.2抑制算法的退出

值得注意的是，抑制算法只有在出现 $err_t \geq err_{t-1}$ 之后，并且 $RTT_{i,j} \leq MeanRTT_{i,j} + 4 \cdot DevRTT_{i,j}$ 时才执行；当 $RTT_{i,j} > MeanRTT_{i,j} + 4 \cdot DevRTT_{i,j}$ 会退出抑制算法，即让 $d = 1$ ，退化为Vivaldi算法。

5.5.4稳定抑制Vivaldi算法的执行步骤

对于节点 i 获得来自节点 j 的一个 $RTT_{i,j}$ ，稳定抑制算法的执行步骤为：

步骤1：进行初始化，让 $d = 1$ ，计数器 $n = 0$ ，让 err_{t-1} 为一个极大数；

步骤2：对随机延迟污染现象进行检测，如果 $RTT_{i,j} > MeanRTT_{i,j} + 4 \cdot DevRTT_{i,j}$ 则执行一次步骤1和步骤3，之后让 $RTT_{i,j} = MeanRTT_{i,j}$ 后，

跳转到步骤6；

步骤3： 计算 $DevRTT_{i,j}$ 与 $MeanRTT_{i,j}$ 的值：

$$[DevRTT]_{i,j} = (1-\beta) [DevRTT]_{i,j} + \beta [RTT]_{i,j} - [MeanRTT]_{i,j} \quad (5.4)$$

$$[MeanRTT]_{i,j} = (1-\alpha) [MeanRTT]_{i,j} + \alpha [RTT]_{i,j} \quad (5.5)$$

步骤4： 感知坐标抖动，若此时 d 不等于1，说明已经开始抑制算法，跳转到步骤6；
否则计数器 $n = n + 1$ ，如果 $n \in Neighbor(i)$ ，跳转到步骤6。

步骤5： 让 $n = 0$ ，计算 $errt$ ：

$$[err]_t = (\sum_{j \in Neighbor(i)} \sqrt{([MeanRTT]_{i,j} - [X]_i(t) - [X]_j(t))^2}) / |Neighbor(i)| \quad (5.6)$$

若 $errt \leq errt-1$ 则让 $d = 0$ ，否则让 $errt-1 = errt$ 。

步骤6： 更新坐标

- 1) 如果 d 小于1，用公式(5.3)计算 d 。
- 2) 用公式(2.5)计算误差权值 ω
- 3) 用公式(2.6) 计算两点距离相对误差 es 。
- 4) 用公式(2.7)更新自身误差估计 ei 。
- 5) 用公式(5.2)计算 δ
- 6) 用公式(2.9)更新本地坐标 x_i 。

5.5.5稳定抑制Vivaldi算法的性能分析

为了检测稳定抑制 Vivaldi算法的性能，本文采用的时延数据来自文献[15]的实测时延数据集，并选取了其中231个节点，1907419条 RTT记录，这些节点都有至少9个邻居节点。

5.5.5.1准确性分析

{61%：定义相对误差RE用于度量算法的准确性：}

$$[RE]_{i,j} = ([RTT]_{i,j} - \|X_i - X_j\|) / \min([RTT]_{i,j}, \|X_i - X_j\|) \quad (5.7)$$

{52%：其中 X_i 与 X_j 分别为节点 i 与邻居节点 j 的坐标， $RTT_{i,j}$ 为节点间实测时延，} $X_i - X_j$ 为节点间预测时延， $\min(a, b)$ 表示 a, b 间的最小值。
{44%：相对误差RE可以很好的表示实测时延与预测时延的相对差异性。}
{41%：将1907419条RTT记录作为一次迭代，20次迭代后，其相对误差RE累计分布图如下图3：}

图3 相对误差RE累计分布图

从图3中可以看出，稳定抑制 Vivaldi算法中，相对误差小于1的节点占了总体77.60%，而 Vivaldi算法中，相对误差小于1的节点占了总体73.32%，并且稳定抑制 Vivaldi算法的相对误差 RE累计分布始终比 Vivaldi算法高，{45%：说明了稳定抑制 Vivaldi算法的准确性比 Vivaldi算法更高。}

{42%：除了使用相对误差RE来度量算法的准确性，本文还使用了另一种度量方法。}
{44%：对于节点i获得来着节点j的一个RTT，在更新完坐标后，此次更新误差为：}

$$e_{-}(i, j) = \sqrt{((\lfloor \text{RTT} \rfloor_{-}(i, j) - \|x_i - x_j\|)^2)} \quad (5.8)$$

{42%：则对于整个网络坐标系统中的n个测量RTT，定义整体误差均值为：}

$$e = (\sum_{i=1}^n \sqrt{((\lfloor \text{RTT} \rfloor_{-}(i, j) - \|x_i - x_j\|)^2)}) / n \quad (5.9)$$

则整体误差均值e反映了整个网络坐标系统的误差程度，整体误差均值e越大，网络坐标系统的误差越大。将1907419条RTT记录作为一次迭代，计算每一次迭代的整体误差均值，具体如下图4所示：

图4 整体误差均值

从图4可以明显看出 Vivaldi算法以及稳定抑制 Vivaldi算法都在较少随迭代次数内随着迭代次数增加而减少整体误差， Vivaldi算法在第3次迭代后，其整体误差均值在55.14上下波动，而稳定抑制 Vivaldi算法在第3次迭代后，其整体误差均值在53.29上下波动，说明了稳定抑制 Vivaldi算法比 Vivaldi算法有着更高的准确性，其准确性提高了3.35%。同时稳定抑制Vivaldi算法的整体误差均值一直比Vivaldi算法低，说明了稳定抑制Vivaldi算法具有比Vivaldi算法更快的收敛能力。

5.5.5.2抑制抖动能力分析

{41%：对于节点i，其坐标 x_i 在m次更新后的期望值 u_i ，以及 x_i 与 u_i 的距离方差 s_i ，计算公式如下公式(5.10):}

$$\{ \begin{aligned} u_i &= (\sum_{i=1}^m x_i) / m \\ s_i &= (\sum_{i=1}^m (\|x_i - u_i\|^2) / m) \end{aligned} \quad (5.10)$$

{45%：其距离方差 s_i 反映了节点i的抖动程度，方差 s_i 越大，该节点的抖动程度越大。}
则由n个节点组成的网络坐标系统的整体抖动方差S：

$$S = (\sum_{i=1}^n s_i) / n \quad (5.11)$$

则反映了网络坐标系统抖动程度。同样将1907419条RTT记录作为一次迭代，计算每一次迭代整体抖动方差，具体如下图5所示：

图5整体抖动方差

图5中反应了，算法刚开始执行时由于算法收敛造成坐标的剧烈变化， Vivaldi算法第一次迭代的整体抖动方差为141.35，而稳定抑制 Vivaldi算法第一次迭代的整体抖动方差为222.34，这在一定程度上反应了稳定抑制 Vivaldi算法有着比 Vivaldi算法更快的收敛速度。而当 Vivaldi算法在第7次迭代后，其整体抖动方差在17.25上下波动，而稳定抑制 Vivaldi算法在第6次迭代后，其整体抖动方差在15.75上下波动，说明稳定抑制 Vivaldi算法比 Vivaldi算法有着更好的抑制抖动能力，其抑制效果抖动提升了8.7%。

5.6 本章小结

随机延迟污染是由于网络拥塞，网络拓扑结构发生变化，不同的数据包及其响应可能沿不同路径转发而造成的。本章通过数据分析，阐述了随机延迟污染现象的普遍存在性。

{42%：因而抑制随机延迟污染现象对提升网络性能有着相当的重要性。} {50%：介绍了一种现有的随机延迟污染抑制方法MP-Filter，同时提出了T0-Filter随机延迟污染抑制方法。}

{42%：由于网络中普遍存在的TIV现象，使得在将网络节点置入几何空间的过程中，节点难以找到合适的位置来反映TIV现象，导致节点坐标的抖动。} {44%：因而抑制TIV现象，减少节点坐标的抖动，对于提升网络坐标系统准确性有着很大意义。} 本章介绍了三种现有的用于抑制TIV现象的网络坐标距离预测算法，并对其优缺点进行了分析。T- Vivaldi算法采用随机选取节点的邻居节点来测量是否造成 TIV现象，从而抑制 TIV现在造成的坐标抖动，然而由于 TIV现象，仅选取了少量邻居节点无法找到所有 TIV现象，对 TIV的抑制效果有限。抖动感知的慢启动抑制算法则采用迭代因子自适应的方法，通过抖动感知来决定增长或减少迭代因子， {66%：从而在保证收敛速度的同时，抑制节点坐标的抖动。} 然而该抖动感知方法仅考虑了TIV现象，对随机延迟污染现象与网络攻击没有进行考虑，这可能导致其准确度下降。 {48%：能量更新抑制方法通过限制应用级坐标更新来减少因TIV现象造成的坐标盲目更新，然而其适应不同网络环境的难度大，开销也高。}

本章提出了稳定抑制 Vivaldi算法，该算法能同时抑制随机延迟污染现象和 TIV现象， {41%：经过仿真实验表明，稳定抑制 Vivaldi算法能够在保证略高于 Vivaldi算法准确性的同时，} 抑制坐标的抖动，其抑制能力提示了8.7%。

文章总结

6.1总结

网络坐标系统是一种网络节点距离预测机制，能够有效快速的获取网络节点间的时延信息，对于提升网络性能，优化网络应用有着很大的帮助。但由于网络环境的复杂，网络坐标系统受到很多因素的影响，如由于网络拥塞，网络拓扑结构发生变化， {41%：不同的数据包及其响应可能沿不同路径转发而造成的随机延迟污染，以及在将网络节点放入几何坐标系时，} {40%：产生的三角不等式违例现象，都可能造成了网络坐标的动荡，降低了网络坐标系统的准确性。}

Vivaldi是基于模拟的时延预测机制，是全分布式的，无需额外设立基站设施，对Internet拓扑变化有较好的适应性。然而Vivaldi对于随机延迟污染现象与TIV现象并没有很好防范策略，本文则对这两个方面展开工作：

{48%：对于随机延迟污染现象，本文介绍了随机延迟污染抑制方法MP-Filter，同时提出了T0-Filter随机延迟污染抑制方法。}

其次对 TIV现象，提出了稳定抑制 Vivaldi算法，其特点是同时考虑随机延迟污染以及 TIV现象， {41%：其思想是通过计算并保存节点获得的每一个实测时延的估计值，} 通过比较实测延时与估计值，来判断是否出现了随机延迟污染，同时通过计算预测延时与均值估计值的差值均值来判断是否出现了坐标抖动；在抑制抖动方法中，选择一种递减函数作为抑制函数来减少坐标更新程度。

然而本文依然存在许多不足之处：本文目前的研究仅仅只在理论分析与仿真测试阶段，没

有在真实的网络平台环境中进行构建与测试。另外本文使用的网络节点间实测时延数据源于网站以前的数据集，时效性和真实性不够充足。但限于本人能力不足，难以获取现实网络节点间实测数据。对于本文提出的T0-Filter抑制方法与稳定抑制算法仍具有不足之处，缺少足够的实测时延数据进行测试。同时T0-Filter抑制方法对于随机延迟污染现象仅能起到有限的抑制效果，无法很好的避免随机延迟污染现象；而稳定抑制算法对TIV现象造成的坐标抖动抑制效果有限，受限于抑制函数的选取。

现在，时延测量对提升网络性能具有相当的意义，现阶段的网络坐标系统对于网络节点距离的预测值与实际值仍具有较大的差距，仍有待研究。

检测报告由PaperPass文献相似度检测系统生成

Copyright 2007-2018 PaperPass