分类号 密 级

U D C 编 号

武 汉 大 学

硕 士 学 位 论 文

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

研 究 生 姓 名：赵玉琦

学 号：2015202160008

指导老师姓名、职称：李兵 教授

专 业 名 称：软件工程

研 究 方 向：软件工程

二〇一八年五月

A Study on Network Coordinate System Construction and Performance Optimization Based on High-precision Spatiotemporal Information

**By**

***Yuqi Zhao***

**Supervised by**

**Prof. Bing Li**

**Wuhan University**

**Wuhan，430079 P.R.China**

**May，2018**

**郑 重 声 明**

本人的学位论文是在导师指导下独立撰写并完成的，学位论文没有剽窃、抄袭、造假等违反学术道德、学术规范和侵权行为，本人愿意承担由此而产生的法律后果和法律责任，特此郑重声明。

学位论文作者（签名）：

年 月 日

# 目录

[目录 4](#_Toc13254)

[第一章 绪论 8](#_Toc30565)

[1.1背景介绍 8](#_Toc8581)

[1.2 研究目标 10](#_Toc27420)

[1.3文章结构 10](#_Toc412)

[第二章 项目关键技术分析 11](#_Toc17086)

[2.1 时空大数据 11](#_Toc5669)

[2.2 网络坐标 13](#_Toc10620)

[2.2.1 概述 13](#_Toc4354)

[2.2.2 GNP简介 13](#_Toc22337)

[2.2.3 PIC简介 14](#_Toc10684)

[2.2.4 NPS简介 16](#_Toc12184)

[2.2.5 Vivaldi简介 17](#_Toc8015)

[2.3 北斗导航系统 19](#_Toc8378)

[2.4 软件定义网络（SDN） 19](#_Toc28021)

[2.5 OpenVSwitch 19](#_Toc3132)

[2.6 mininet 19](#_Toc31024)

[2.7 RYU控制器 19](#_Toc3564)

[2.8 本章小结 19](#_Toc22169)

[第三章 时空大数据平台建设方案 19](#_Toc23268)

[3.1 基于SOA的时空大数据平台体系 19](#_Toc6723)

[3.2 多层协同的架构体系 22](#_Toc31418)

[3.3 功能平台的服务体系 22](#_Toc24354)

[3.4 开放方向的数据体系 22](#_Toc27361)

[3.5 高效运行的管理体系 22](#_Toc27189)

[3.6 时空大数据标准体系 22](#_Toc23602)

[3.7 本章小结 22](#_Toc28480)

[第四章 系统原型设计与实现 23](#_Toc4155)

[4.1 SDN 23](#_Toc8010)

[4.2 Mininet和RYU安装 23](#_Toc23244)

[4.3 OpenVSwitch 协议解读 23](#_Toc29222)

[4.4 网络风暴解除，ARP回复 23](#_Toc680)

[4.5 原型设计与实现 23](#_Toc28893)

[4.6 本章小结 23](#_Toc7360)

[第五章 时空网络坐标系性能分析与优化 23](#_Toc29823)

[5.1 网络坐标系统 23](#_Toc24125)

[5.1.1 随机延迟污染现象 24](#_Toc1154)

[5.1.2 三角不等式违例现象 25](#_Toc997)

[5.2 随机延迟污染现象及抑制方法 26](#_Toc6424)

[5.2.1 随机延迟污染现象数据分析 26](#_Toc8607)

[5.3 抑制随机延迟污染的方法 27](#_Toc26907)

[5.3.1 MP-Filter抑制方法 27](#_Toc802)

[5.3.2 TO-Filter抑制方法 27](#_Toc26760)

[5.4 网络坐标中三角不等式现象级抑制方法 28](#_Toc20245)

[5.4.1 T-Vivaldi TIV感知的坐标系统 28](#_Toc926)

[5.4.2 抖动感知的慢启动抑制算法 29](#_Toc26661)

[5.4.3 能量更新抑制方法 31](#_Toc17430)

[5.4.4 稳定抑制Vivaldi算法 32](#_Toc9185)

[5.5 基于Vivaldi算法的抑制方法 32](#_Toc14617)

[5.5.1检测随机延迟污染 32](#_Toc27992)

[5.5.2坐标抖动感知方法 33](#_Toc32727)

[5.5.3抑制算法 34](#_Toc22993)

[5.5.4稳定抑制Vivaldi算法的执行步骤 34](#_Toc31219)

[5.5.5稳定抑制Vivaldi算法的性能分析 35](#_Toc10257)

[5.5.5.1准确性分析 35](#_Toc26647)

[5.5.5.2抑制抖动能力分析 37](#_Toc10001)

[5.6 本章小结 38](#_Toc4724)

[第六章 文章总结 40](#_Toc9994)

[6.1总结 40](#_Toc13818)

[6.2 机遇 41](#_Toc28739)

[6.3 挑战 41](#_Toc25735)

[致谢 42](#_Toc21776)

# 绪论

## 1.1背景介绍

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。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。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。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。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。

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，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。

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，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，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。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，从而形成一个个数据孤岛并导致大数据服务的不可用。

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。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。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。

## 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，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。

## 1.3 文章的创新点

（1）基于北斗卫星导航以及北斗地基增强系统

（2）构建高精度时空信息网络大数据环境

（3）对融合时空信息的数据与隐私保护理论进行分析验证

## 1.4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，为提供基于高精准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，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：

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。

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，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。

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TO—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，并对其优劣点进行了分析。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 网络坐标

### 2.1.1 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，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。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。

### 2.1.2 GNP简介

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。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。

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。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)，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。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。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。

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，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*P1,P2……PN*(*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*Li,j*，从而构造一个*N*×*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*X1*,*X2……XN*来表示N个锚节点坐标，从而使锚节点间坐标距离||*Xi*-*Xj*||与实测延迟*Li,j*的误差平方总值最小，即下面公式的值最小。

(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Lk，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。

(2.2)

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。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，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。

### 2.1.3 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。

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，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，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*Lk*，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相对误差平方总值最小，即下面公式的值最小。

(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*P1*,*P2……PN*，使用Ping方式获得锚节点间的实测时延*Li,j*，从而构造一个*N*×*N*的时延矩阵，并找到一组坐标*X1*,*X2……XN*来表示N个锚节点坐标，从而使锚节点间坐标距离|| *Xi*-*Xj*||与实测延迟*Li,j*的相对误差平方总值最小，即下面公式的值最小。

(2.4)

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：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：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，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：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：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：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；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。

### 2.1.4 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：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；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*Li*，则其选取的参考节点*j*所在层数*Lj*需要满足*Li*< *Lj*，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：

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。

2)测量获得普通节点与参考节点间的实测距离。

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。

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。

### 2.1.5 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。Vivaldi算法将网络节点通过将物理弹簧弹性定律，即“弹簧在拉伸和压缩时，其动能和势能会相互转化”，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。

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，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，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*RTTi,j*:

1)计算误差权值*ω*：

(2.5)

2)计算两点距离相对误差*es*：

(2.6)

3)更新自身误差估计*ei*：

(2.7)

4)更新本地坐标*xi*：

(2.8)

(2.9)

其中*xi*为节点*i*的坐标，*u*(*xi*-*xj*)为从节点j坐标到节点*i*坐标的单位向量，*ei*为节点*i*的误差因子，*ce*和*cc*为调节因子，其取值越大，则节点每次更新的程度越大。

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，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。

## 2.2 北斗导航系统授时

### 2.2.1 相关概念

#### 2.2.1.1 授时

所谓授时，顾名思义，就是指在确定某种时间尺度的前提下，把能够维持响应尺度的时间信息按照一定的方式，传递给使用者的过程。通常来讲，授时系统的构建需要两个必备的条：（1）根据国际电信联盟对授时系统上的规定，系统时间必须要跟国家的标准时间同步，就是要以国家时间标准作为基准。（2）授时系统在向用户提供授时服务的时候，必须要按照广播的方式来将授时系统的时间发播出去，以便于用户对时间进行接收和加以使用。在收拾系统内，用户的数量是不应该被限制的。不同的用户接到相同的授时信号时，都能够将本地时间与统一授时的时间尺度相匹配，实现时间上的同步。

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。

#### 2.2.1.2时间同步

时间同步指的则是，根据协调世界时（UTC）这个时间参考基准，各个应用系统和网络硬件设施的时钟，能够以特定方式与UTC进行时钟的时刻和时间间隔上的同步。

## 2.3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，卫星无线电导航业务）授时和共视法授时。

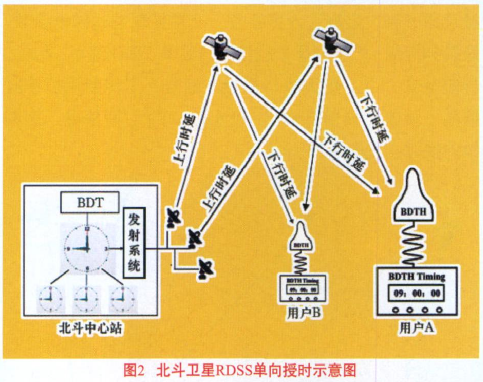
##### 2.3.2.1.1北斗RDSS单向授时

北斗卫星RDSS业务广播授时信号的载波频率为2491.75MHz，为连续帧信息结构，在时域上分为超帧和帧，固定帧长，1个超帧周期为1分钟，含1920帧，1秒包含32帧，帧周期为31.25ms。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。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；授时信息的内容每1超帧更新1次。

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：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。

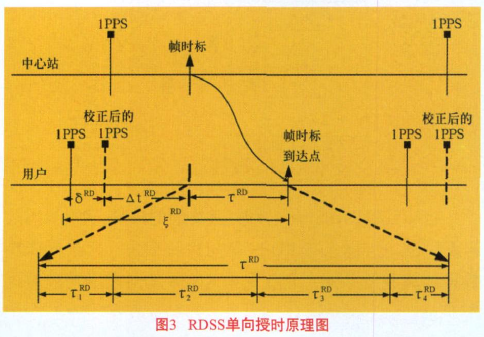
北斗卫星RDSS单向授时过程是在BDT控制中心主原子钟的监控下，发播工作原子钟产生卫星导航信号的频率、编码速率、相位、导航电文，由发射设备发送到北斗卫星，卫星转发器将授时信号下行传递到用户接收终端，终端解算输出1PPS（秒脉冲）和TOD（Time Of Day）时间信息，完成RDSS单向授时。

RDSS单向授时示意图见图2：



小于1秒的小数秒部分的BDT的时间传递过程如图3所示：在中心站出站信号某一“帧时标”与其前一个BDT整秒时刻（即1PPS）的时差为，“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。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：

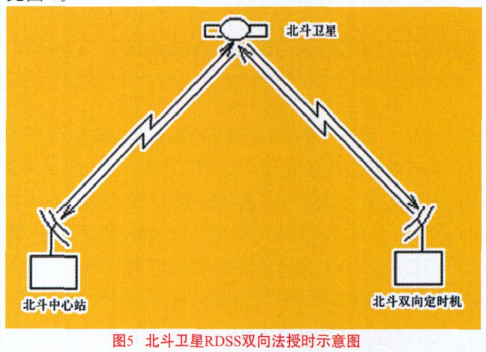


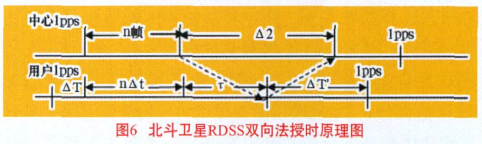


移相调整本地钟输出的1PPS，使时间差为零，调整后的本地1PPS就与BDT的1PPS实现时间（相位秒部分）同步。北斗卫星RDSS单向授时精度优于100ns，而且用户数量不受限制，完全可以满足绝大多数时间用户精度需求。

##### 2.3.2.1.2北斗RDSS双向授时

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。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，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。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。该方法采用双向比对测量确定信号单向传播时间延迟。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。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。





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。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；同时也即该帧号对应的北斗时间（小于1整分的部分）。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，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1PPS的时间间隔；同时，用户终端立即向中心控制系统回发相应信号，中心控制系统测出第n帧信号的往返时间值，并算出该信号由中心发出至用户机收到的正向传播时延，再将发送给该用户作为双向定时时延改正值。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，调整本机时钟，从而完成用户终端与中心站BDT的时间同步。

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，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20ns，但用户数量受到限制。

##### 2.3.2.1.3北斗RNSS单向授时

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。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。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：为伪距观测值；为测站与卫星间距离；为卫星钟差；为接收机钟差；为对流层延迟；为电离层延迟；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。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。

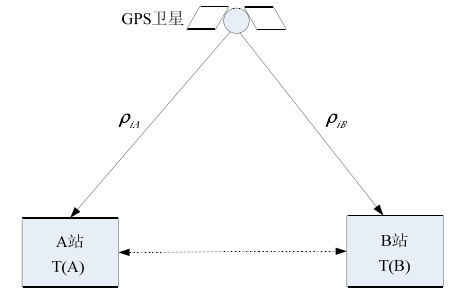
可以采用相位平滑伪距或载波相位观测值来提高授时精度，授时精度可以达到10ns以内。

##### 2.3.2.1.4北斗/GNSS共视法授时

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，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，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，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。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。

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的伪距为和，可得其接收机钟差分别如下如下：





其中：为伪距观测值；为测站与卫星间距离；为卫星钟差；、为接收机钟差；为对流层延迟；为电离层延迟；为观测噪声和其他误差。

作差可得：



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。

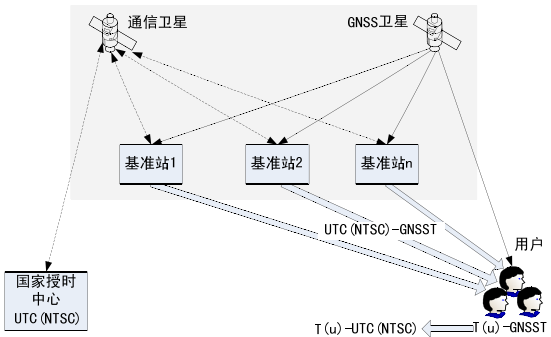
使用简单的单通道伪距接收机，即可保证实现两站优于10ns的时间同步精度。当两站的基线长度不大于100km 时，两站可以实现优于 2ns 的时间同步精度。目前，BIPM 已将共视作为全球70多个守时实验室之间时间比对的主要手段之一。

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。

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，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。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：



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。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。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。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，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)

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。

##### 2.3.2.1.5 GNSS全视法时间传递

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，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；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。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。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，时间比对的精度比上述两种方法有了明显的提高。从 2006 年开始全视法已经在计算 UTC/TAI方面替代了共视法。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）技术将网络应用、网络设备和网络服务紧密结合起来，设备间的交互。如消息传送、服务开通和安全预警等，实现了网元设备在逻辑上的集中控制，使得网络应用可以更加灵活的部署与运行。为了让运营商能够通过编程方式来控制网络，使网络管理和服务编排更加的自动化、智能化，跨服务器、交换机与路由器的配置应用策略，使得网络设备操作系统和应用程序进行更好的耦合。

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。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。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。

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。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。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术。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，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。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]：

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

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。

网络设备的组成通常包括数据平面和控制平面，控制平面主要负责指定转发策略供数据平面使用，主要是各类协议，如路由协议、网关协议等。而这些策略的执行主要依靠数据平面来完成。在网络环境中，如图2.1所示，网络设备中封装着软件化的控制面，这些软件化的控制面是由各自的厂商独立研发的，所以设备的使用者就无法对其控制能力进行进一步的设计和改造。而SDN的出现则改变了这一现状，其思想的核心是，仅对硬件上的软件进行更新，这样一来就通过软件的升级和更新使得网络功能得到了扩展。从图中可以看出，传统的网络只是简单的区分控制面和数据面，具体的操作和处理是无法分开进行的，也没有办法实现控制层面上的交互。SDN可以真正意义上实现控制面和数据面的分隔，这样多个交换机就可以被一个控制机构同时控制，现有的网络架构的限制就被打破，提供了开放网络系统的基础上提供的新解决方案。

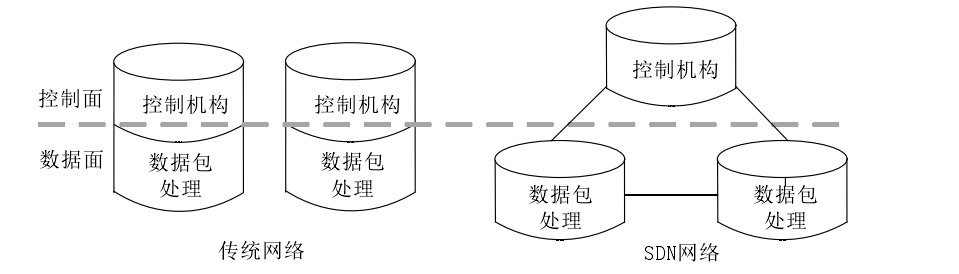


图2.1 传统网络与SDN的结构对比图

图2.2展示的是SDN的结构图，可以看出，主要有三层构成—应用层、控制层和基础设施层。基础设施层主要由网络系统中的OpenFlow交换价作为底层转发设备组成。控制层在中间，用于网络整体状态的统一维护，通过OpenFlow等南向接口，来实现对底层设备信息的获取，其北向接口可以为应用层提供扩展的

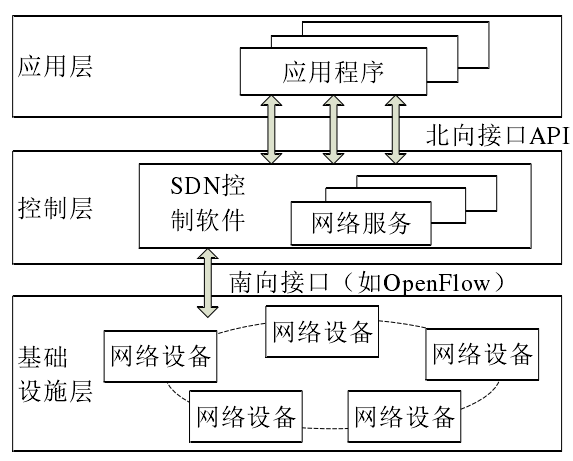


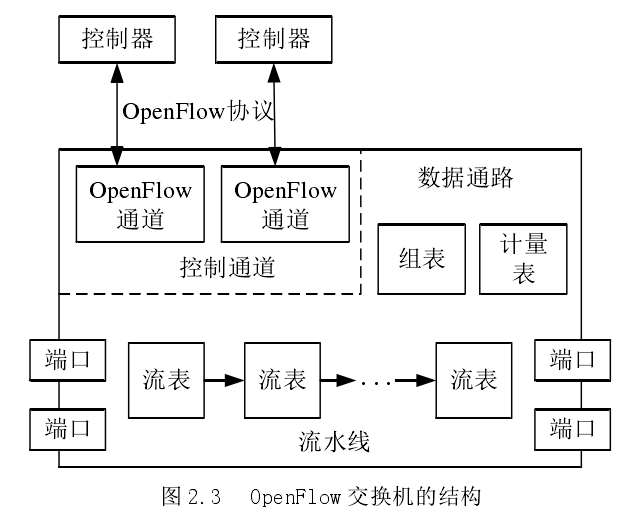
图2.2 SDN的结构图

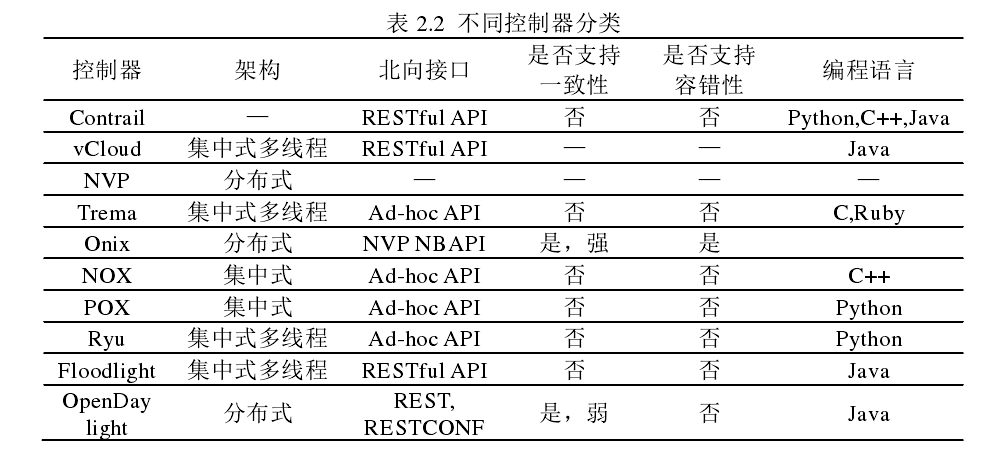
可能，使应用发布控制消息和得到当前网络状态更加方便。OpenFlow协议是比较标准的南向接口协议，同时也在不断的改善和升级过程中，北向接口仍然没有较为成熟的标准协议。这个模型体现了SDN网络的可编程性和灵活性，使得网络管理者可以更好地通过调度、管理并优化网络资源来实现高效可控的网络。

### 2.4.3 OpenFlow

OpenFlow协议集是一组API和协议，可以将传统网络的第二层和第三层协议进行整合和替代。OpenFlow协议集主要包括OpenFlow交换机规范（OpenFlow Switch Specification, OF-SWITCH）、OpenFlow管理与配置协议（OpenFlow Management and Configuration Protocol，OD-CONFIG）、OpenFlow光传输协议扩展（Optical Transport Protocol Extensions）以及其他的测试和服务协议。

在SDN中，OpenFlow是一个非常重要的组成部分，SDN指的是网络设计的一种理念，围绕这个设计理念有很多不同的解决方案，OpenFlow是实现SDN一个比较好的案例，相比较于其他的SDN解决方案，其内容成熟度、应用领域、产品兼容性等方面的优势突出。





### 2.4.4 mininet

## 2.5 本章小结

# 基于GNSS实现授时时间同步

## 3.1 时间同步的必要性

### 3.1.1网络优化方面

#### 3.1.1.1网络管理系统

目前，各种通信网大都建有各自的集中网管系统，能集中收集网络的性能数据进行统计分析，以及收集各网元设备的告警信息进行告警定位。当网上一台设备出现告警时，往往其它关联设备也出现各种告警。设备送出的告警信息中所含的时间标签是由各设备的时钟打上的，如果它们是同步的，网管设备只要把这些告警按时间排序，就可以分析出故障的源头及引起的后果。这对网上故障定位和查找故障原因是很有用的。

#### 3.1.1.2七号信令监测系统

为了准确定位通信网中出现的故障类型和故障点，需要在信令流量较大的信令转接点（Singnal Transfer Point，STP）设立信令采集点，建立一套完善的七号信令监测系统，用于监测和分析网上信令流的动向。当网上出现故障时，所有相应的信令流数据将被送往此监测系统进行分析处理，并进行快速故障定位。为避免因信令出现先后顺序的错误而产生虚假信息，必须要求所有信令流的时间信息是准确无误的，进而确保故障类型以及故障点快速定位的准确性。由于信令流的时间标签是由各信令采集点分别产生的，因此要求各个信令采集点必须保持时间同步。另外，若要利用七号信令监测系统对网络接通率和呼损率进行统计，信令采集点之间时间不同步有可能会使统计出现偏差，从而影响对整个通信网性能的评估。

#### 3.1.1.3安全认证系统

通信网进行网络安全方面的优化，建立安全认证系统，以确保传递的通信信息安全。但是由于目前通信网上大部分设备时间都不一致，从而根本无法对全国的网络进行全面的评估。时间不同步直接影响到安全认证系统的建设和顺利运行。

### 3.1.2提高服务质量方面

在通信网中，计费对时间同步的要求主要表现在两个方面：一是通信网自身计费对时间同步的要求；另一个方面是网间费用结算时对时间同步要求。目前在每个交换机都有通话记录产生模块，由此模块产生话单，打上时间标签，标示出通话的起始时刻、结束时间以及通话时长，并将其送到相应的计费中心进行处理。目前，计费模块的时钟准确度都比较低，差不多在 1E-5 到 1E-6 的量级，这意味着两个交换机在一个月内的时间偏差可能会达到半分钟。

各交换机之间大的时间偏差可能会导致出现有相互矛盾的话单，从而引起用户的申告。如：某用户在 10:01～10:20 通过 A 交换局打过一次长途电话，在10:22～10:30 又通过 B 交换局打了一个长途电话，如果 B 局交换机时钟比 A 局交换机时钟慢 5 分钟，以 A 局交换机时钟为准，则 B 局交换机的计费磁带上会记录这一用户的呼叫时间为 10:17～10:25，反映在最终话单上就是 10:01～10:20和 10:17～10:25 分别有一个长话。也就是说在 10:17～10:20 这一时间段里，用户同时在打两个长话。在最终的用户详细话单上出现了同一时间段某用户有多个呼叫等一些不正常现象。

如果具备准确的时间同步，可以大大提高网间结算的准确度，减少出现费用结算的纠纷，特别是随着计费单位变小（现在长途话费按 6 秒为单位进行计算），也要求计费要越来越准确，因此交换机等通信设备对时间准确性要求也越来越高。

### 3.1.3网络及新业务方面

#### 3.1.3.1IP网络及应用

在 IP 网中，为了对路由选择进行监视或控制，数据包每经过一个路由器都会打上该路由器的当前时间，通过分析这些时间戳，就可以计算出各段路径所引入的时延。因此，保证各个路由器的时间同步对于分析各段路由的流量，顺利完成路由选择是十分重要的。

对于基于 IP 的电子商务等，其网络安全性对时间同步也有严格的要求。网络安全是实现电子商务的基础，而一个通用性强，安全可靠的网络协议则是实现电子商务安全交易的关键技术之一，它也会对电子商务的整体性能产生很大的影响。安全套接层协议（Secure Sockets Layer，SSL）是目前安全电子商务交易中使用最多的协议之一。SSL 协议在“重传攻击”上，有它独到的解决办法。SSL协议为每一次安全连接产生了一个 128 位长的随机数——“连接序号”。理论上，攻击者提前无法预测此连接序号，因此不能对服务器的请求做出正确的应答。但是计算机产生的随机数是伪随机数，它的实际周期要远比 2128小，更为危险的是有规律性，所以说 SSL 协议并没有从根本上解决“信息重传”这种攻击方法，有效的解决方法是采用“时间戳”。但是这需要解决网络上所有节点的时间同步问题。

在各种政务和商务文件中，时间更是十分重要的信息。在书面合同中，文件签署的日期和签名一样均是十分重要的防止文件被伪造和篡改的关键性内容。在电子文件中，同样需对文件的日期和时间信息采取安全措施，即通过数字时间戳服务（digital time-stamp service，DTS）对文件进行安全保护，这也需要解决时间同步问题。IP 电话（voice over IP，VOIP）业务也对时间同步精度有一定要求。

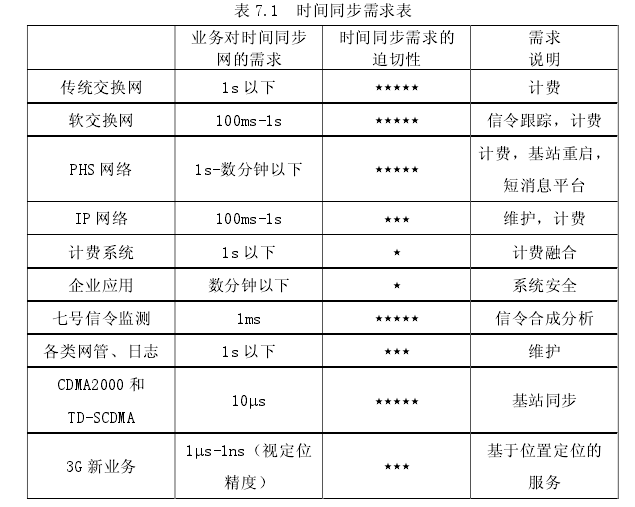
#### 3.1.3.23G网络及新业务

3G 网络有三种制式： WCDMA、 CDMA2000 和 TD-SCDMA，其中在CDMA2000 和 TD-SCDMA 系统中，基站系统之间需要准确的时间同步。在软切换中，如果 RNC 和 Node B 之间没有时间同步，可能导致在选择器中发生邮件指令不匹配，从而使通话连接不能建立起来。CDMA2000 和 TD-SCDMA 系统对时间同步的精度要求是 10μs。WCDMA 属于异步工作方式，其正常工作时不需要准确的时间同步。

在 3G 网络中，将会出现一些新的移动业务，例如基于位置定位的服务等。在移动通信网中实现基于位置定位的服务有几种技术选择，其中的一种实现手段是利用手机接收附近多个基站发送的无线信号进行定位。这要求基站的精确位置是预知的，并且都是时间同步的。根据定位精度要求的不同，基站时间同步的精度要求也是不同的。一般来说，若各基站之间时间同步的误差在 1μs，则位置定位的精度大约在 300 米左右。

除了通信网络外，时间资源作为一种重要的公共资源，在很多行业如银行、电视台、测绘、车站、调度、电力系统、导航、控制等都需要。只不过由于任务和需求的不同，对时间同步系统提供的时刻精度要求也不同，有的需要粗略的时刻，而有的则需要精确的时刻。

### 3.1.4时间同步需求



## 3.2 互联网时间同步

在互联网上进行时间同步具有重要意义。随着互联网发展和延伸到社会的各个方面，在网络的其他领域对时间同步也提出了多种要求，例如各种实时的网上交易、制造过程控制、通信网络的时间配置、网络安全性设计、分布性的网络计算和处理、交通航班航路管理以及数据库文件管理和呼叫记录等多种涉及时间戳的应用，都需要精确、可靠和公认的时间。

### 3.2.1 NTP协议

NTP（Network Time Protocol）是目前应用比较广泛的网络时间协议，它是为因特网内的计算机时钟能够与网络时间同步化而设计的的协议。

NTP是由美国德拉瓦大学的D.L.Mills教授于1985年提出，它是设计用来在Internet上使不同的机器能维持相同时间的一种通讯协定，除了可以估算封包在网络上的往返延迟外，还可独立地估算计算机时钟偏差，从而实现在网络上的高精准度计算机校时。

在通常的环境下，NTP 提供的时间精确度在 WAN上为数十毫秒，在 LAN上则为亚毫秒级或者更高，在专用的时间服务器上，则精确度更高。

#### 3.2.1.1发展现状

（1）网络时间协议（NTP）同步Internet中的主机和路由器的时间；

（2）目前Internet中分布着超过100,000台NTP服务器，它们为全世界提供时间服务；

（3）理论上，NTP为主机提供的准确性，在WAN中可以低于10ms，在LAN中可以可以低于1ms；在精确的时间源，如铯振荡器或者GPS接收器时，精确度可以达到微秒级；

（4）守护进程现在几乎可以运行在各种工作站和服务器平台上，如Unix、Windows、VMS和嵌入式系统。

#### 3.2.1.2应用领域

（1）网络故障的隔离、报告和恢复；

（2）网络监控、测量与控制；

（3）分布式多媒体流的同步；

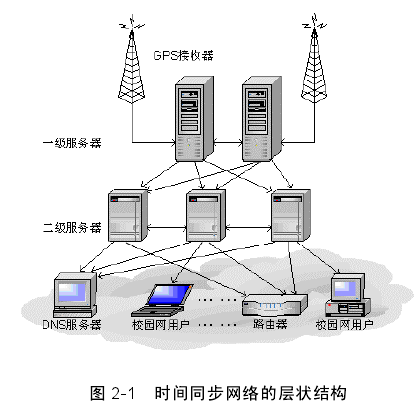
（4）确保系统之间的远程系统调用（RPC）能够正常进行，防止重播，维护序列号的唯一性；

（5）密钥管理和生存期控制；

（6）股票交易和购物订单等的确认时间戳。

#### 3.2.1.3层状结构

时间同步网络理论上根据其精确度和重要性一般分为从0-15的共16个级别或更多级，实际上不会大于6 级。级别编码越低，精确度和重要性越高。时间的分配自级别编码小的层次向较大的层次进行，即由第 0 级向第15级分配渗透。第0级设备是时间同步网络的基准时间参考源，它位于同步子网络的顶端，目前普遍采用全球卫星定位系统，即由 GPS 播发的 UTC 时间代码。子网络中的设备可以扮演多重角色。例如一个第二层的设备，对于第一层来说是客户机；对于第三层可能是服务器；对于同层的设备则可以是对等机。这里对等机的含义是相互用 NTP 进行同步的计算设备。NTP 协议就是通过这种网络层状结构一层一层延伸下去，为 Internet 提供对时服务的。



（1）一级服务器（Stratum 1）通过广播、卫星或者调制解调器与国家标准时间同步；

（2）二级服务器（Stratum 2）和客户通过层状结构与一级服务器实现时间同步；

（3）客户与服务器之间以主从、对称或者广播的模式进行通讯，可选择加密认证或不认证；

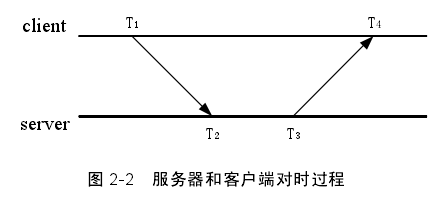
（4）可靠性有赖于服务器的冗余性和网络路径的多样性；

（5）工程算法可以减少抖动（Jitter），减轻多时间来源引起的误差并避免错误的服务器操作；

（6）通过针对网络时间的抖动（Jitter）和时间振荡器频率的wander而设计的适应性算法，使得系统时间和频率能够与权威时间保持同步。

#### 3.2.1.4工作原理

NTP 算法首先就要根据服务器和客户端的往返报文来确定两地时钟的差值和报文在网络中传输的延迟。这里先定义一下，客户端和时间服务器之间的时间偏差（offset）用希腊字母表示；对时过程中的网络路径延迟（delay）用希腊字母表示。



和 是客户端时钟记录的发送 NTP 报文和接收 NTP 报文的时间，和 是服务器端时钟记录的接收和发送 NTP 报文的时间。这里可以设定服务器的时钟是准确的，服务器和客户端时钟的时间偏差是，从客户端发送报文到服务器端的路径延迟是，从服务器到客户端的路径延迟是，路径延迟总和是。那么可以列出三个方程式：







如果假设从客户端到服务器的路径延迟和从服务器到客户端的路径延迟相等：



以上方程变为：





可解得服务器和客户端时钟的时间偏差：



客户端与服务器端总的网络路径延迟：



#### 3.2.1.5工作模式

NTP 协议支持三种对时工作方式：

（1）Server/Client mode（主从模式）：用户向一个或几个服务器提出服务请求，根据所交换的信息，计算两地时间偏差和网络延迟，从中选择认为最准确的时间偏差，并调整本地的时钟。

（2）Multicast/Broadcast mode（广播模式）：此种模式适用于高速的局域网中。局域网中一个或多个服务器以固定的时间周期向某个多播地址广播自己的时标，客户端不计算时间偏差和网络延迟，直接用接收到的时标修正自己的时钟，忽略各种误差。

（3）Symmetric mode（对称模式）：两个以上的时间服务器互为主从，进行时间消息

的通讯，相互校正对方的时间，以维持整个同步子网的时间一致性。

#### 3.2.1.6报文格式

NTP 时间戳以 64bit 无符号的定点数表示，整数部分以高 32bit 表示，相对于基准 1900.01.01 00:00:00 记录当前的秒累积数；小数部分以低 32bit 表示，其精确度可达 200 皮秒。相对于1900年，到2036 年时，此64bit 数将发生溢出，即每136 年，这64bit 的字段将归零，到时将再重新定义协议或废除使用此时间戳。NTP 信息在网络上传输都采用 UDP 协议，端口号是 123，此号码将被使用在 UDP标头里的 Source Port 和 Destination Port 两段中，剩余的 UDP 标头栏将被设定成规格的描述。

### 3.2.2 PTP协议

1992 年 NTP 版本的同步准确度可以达到 200μs，但是仍然不能满足测量仪器和工业控制所需的准确度。为了解决测量和控制应用的分布网络定时同步的需要，具有共同利益的信息技术、自动控制、人工智能、测试测量的工程技术人员在 2000 年底倡议成立网络精密时钟同步委员会， 2001 年在 IEEE 仪器和测量委员会美国标准技术研究所（NIST）的支持下，该委员会起草的规范在 2002 年底获得 IEEE 标准委员会通过并命名为 IEEE1588 标准。IEEE1588 的全称是“网络测量和控制系统的精密时钟同步协议标准”，通常称为精密时间协议（Precision Time Protocol，PTP）。PTP 协议可以达到 100ns 级的同步精度，适合于 Ethernet/IP，Profinet，PowerLink，EtherCat 等基于 LXI（LAN based eXtensions for Instruments）总线连接的以太网络的定时同步，目前主要应用于工业自动化系统。它的基本构思是通过硬件和软件将网络设备（客户机）的内时钟与主控机的主时钟实现同步，提供同步建立时间小于 10μs 的运用，与未执行 IEEE1588 协议的以太网延迟时间1000μs 相比，整个网络的定时同步指标有显著的改善。

### 3.2.3网络时间同步协议比较



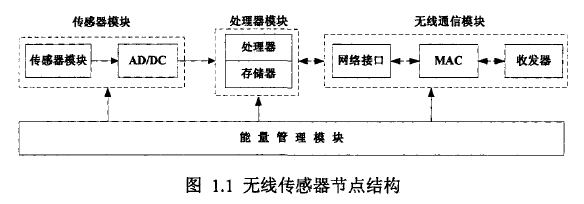
## 3.3 无线传感网时间同步

无线传感器网络(Wireless Sensor Networks，WSN)作为传感网的基袖和重要组成部分，它提供了一种新的感知世界和信息处理的平台，是实现智能化的关键一步。

### 3.3.1无线传感器网络结构

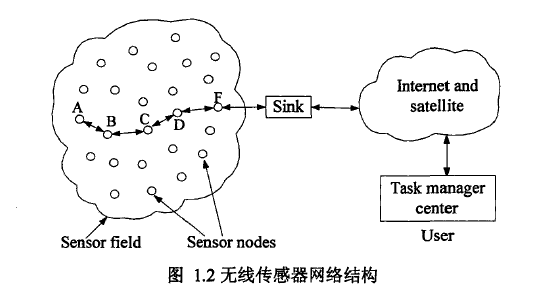
#### 3.3.1.1无线传感器节点结构

无线传感器节点由四部分组成:传感器模块、处理器模块、无线通信模块和能量管理模块。



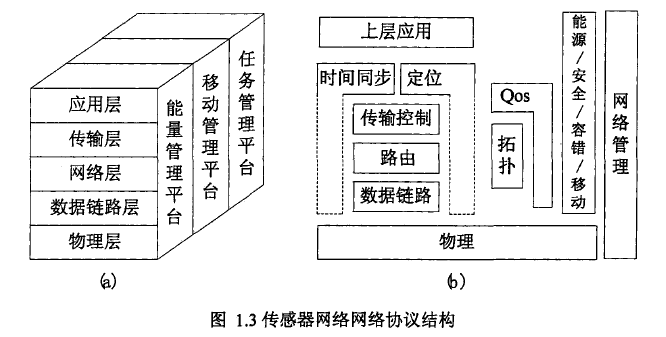
#### 3.3.1.2无线传感器网络系统结构

传感节点(sensornode)通常散布在检测区域,通过无线射频通信自组为无线网络。传感器节点收集到的数据经过其他传感器节点转发或直接发送给汇集节点(sinknode),再经由互联网、卫星网络或者检测专用网络到达任务管理中心(Task manage center)。监测数据在转发过程中可能被某些节点进行融合等处理(视不同的网络协议和应用而定)。用户在任务管理中心收集处理检测数据、发送检测任务,还可以对无线传感器网络进行管理和重新配置。



### 3.3.2无线传感器网络协议结构

无线传感器网络协议包含应用层、传输层、网络层、数据链路层和物理层的五层协议的体系结构，这是早期提出的一种协议栈。针对无线传感器网络资源受限的特点，为了节点能够高效地进行协同工作、拓扑管理和任务调度,在五层协议栈中融入了能量管理平台、移动管理平台和任务管理平台。下面简介各协议层和平台的功能：应用层包括针对各种应用的应用软件；传输层负责数据流的传输控制：网络层为来自传输层的数据提供路由选择以及路由维护；数据链路层负责数据成帧、媒体访问(MAC)和差错控制；物理层提供简单而健壮的调制技术和射频通信技术；能量、移动和任务管理平台管理传感器节点能量髙效使用，检测节点的移动、更新拓扑、更新维护路由，提供任务调度。



### 3.4.3无线传感器网络特点

#### 3.4.3.1自组织性

在环境监测、军事应用时，传感器节点可能是随机散布在检测区域，要求节点能够自组织为多跳网络，将感测到的数据传输到控制中心。

#### 3.4.3.2传感器节点资源受限

传感器节点一般有电池供电，更换困难，所以节点能量有限；传感器节点一般价格低廉、体积小，配备的控制芯片计算处理能力有限，存储容量也非常有限。

#### 3.4.3.3动态拓扑

由于传感器节点一般由电池供电，电池耗尽就会造成节点失效，工作在敌对环境下的节点可能被摧毁，无线链路受环境影响时断时续，用于水下检测的节点会有移动性，这些都会造成无线传感器网的拓扑结构动态变化。

#### 3.4.3.4大规模网络

无线传感器网络通常部署在很大的检测区域，并且节点密度很大，比如森林火灾检测、环境检测。

#### 3.4.3.5应用相关的网络

无线传感器网络是典型的应用型网络。对应不同的应用，需要配置不同的传感器、不同的处理器，利用不同的射频通信技术，还有不同的时间同步协议、路由协议、应用层协议等。

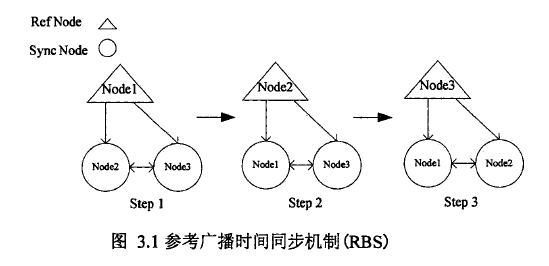
#### 3.4.3.6以数据为中心的网络

无线传感器网络是任务型网络，关注的是任务事件信息，而不是某个ID的节点。由于传感器节点随机部署，节点位置与节点ID没有必然联系。在网络中以数据本身作为查询而不是节点ID。

### 3.4.4无线传感器网络时间同步机制研究现状

#### 3.4.4.1RBS同步协议

RBS (Reference Broadcast Synchronization)是典型的基于接受者-接受者的同步机制，该协议通过第三方节点广播同步信令的方式来实现节点间的同步。如图3.1所示，要实现三节点同步需要三阶段，每一阶段由参考节点(Reference Node)发送同步参考包，各同步节点(Sync Node)接到同步参考包后利用本地时钟记录下接受到该包的时刻，然后各接收节点之间交换各自记录下的接受到同步参考包的时刻，计算出时间偏移后进而两两校准时钟，实现它们之间的时间同步通过节点轮流充当第三方节点可以实现全网同步。在同步消息传播路径上，发送时间和访问时间是同步消息延迟非确定因素的最主要部分。RBS同步协议利用了无线信道的广播特性，即对于同步参考包广播域内的所有同步节点来说，接收的同步参考包的访问时间和发送时间是相同的，从而消除了发送端延迟，广播域内节点之间的时间同步精度就得到了大大提高。通过节点轮流充当第三方节点的角色可以实现整个网络的时间同步。

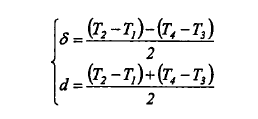


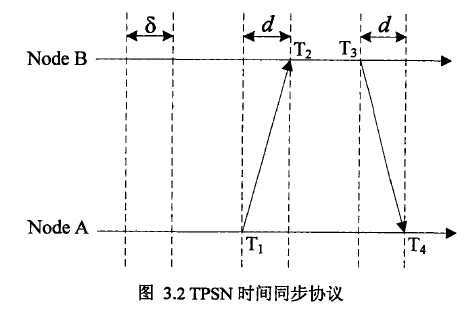
RBS机制釆用了 2中方法来提高同步精度:①最小方差线性拟合法。采用了线性拟合法估计同步节点间的相对频差；②统计技术。参考节点发送多个同步参考包，广播域内的同步节点求取接收到同步参考包的时间差的平均值，进一步消除接收处理时延的影响。接收节点间的时间漂移、接收方的非确定延迟和接受节点的数目等是影响RBS机制性能的主要因素。在多跳网络中执行RBS同步时，同步的误差随着跳数的增加而增大。

#### 3.4.4.2 TPSN同步协议

TPSN (Timing-sync Protocol for Sensor Networks)同步协议是典型的基于发送者-接受者的双向报文交换同步协议。TPSN协议的实现分为两个阶段:层次建立阶段和同步阶段。在网络部署后，将根节点的层次设定为0，一般选择网关节点作为根节点，根节点广播层次发现分组，层次发现分组包含根节点的ID和层次。根节点的广播域内节点接收到这个层次发现分组后，将自己的层次级别设置为0+1，然后广播包含自己ID和层次的层次发现分组。直到网络内所有节点获取自己的层次。为了减少洪泛广播引起的能量损耗和网络堵塞，节点只回应接收到的第一个层次发现分组。在层次结构建立完成后，根节点广播时间同步分组(sync\_pkt)。第1层节点收到sync\_pkt退避一随机时间后，同根节点执行比对同步。第i+1层节点侦听到第i层节点的交换消息后，等待一个随机时间保证第i层节点已经完成同步，然后与自己在层次发现阶段记录的第f层的节点执行比对同步。这样，每个节点同步至的上一层节点，最终网络中所有节点都达到同步。

假设节点A为第f层的节点，B为其第/心层的父节点，如图3.2所示，其同步过程为:节点A向父节点B发送同步请求包并利用本地时钟记录发送时刻T1，节点B收到该包后记录接收时刻的本地时间T2，并发送包含有T3和T2的同步回应包，节点A收到父节点B发送的回应包后利用本地时钟记录接收时刻T4。图中d表示同步消息的传播延时， 表示两个节点之间的时间偏差。假设两节点间为对称链路。则可得





为了消除发送时延、接入时延和接收处理时延的影响，TPSN协议采用链路层(MAC)打时间戳技术。TPSN协议实现了全网范围内节点的时间同步，但每个节点的执行同步需要传递3个同步消息，协议的同步开销较大，并且同步误差随跳数的增加而增大。TPSN协议没有对时钟的频差进行估计，节点的再同步阶段引入了由于频漂引起的时间误差，为了保持较高的时间同步精度节点就需要频繁同步。当网络拓扑发送变化(比如新节点加入)时，需要重新建立节点层次，因此协议的鲁棒性低。

#### 3.4.4.3FTSP时间同步协议

FTSP (Flooding Time Synchronization Protocol)协议是典型的基于发送端的单向广播同步机制。首先root节点作为时间基准点，发送同步报文，广播域内节点接收到该同步报文后，根据经验值估算报文传播时延等参数，再利用估计延迟调整本地时钟，从而同步至root节点。节点同步后作为新的时间基准点，一环环向外同步，直至达到全网同步至root节点。FTSP协议通过MAC层打时戳、估计补偿中断等待时间和编解码时间以及字节对齐时间等技术，消除了除了传播时间外其他不确定时延的影响，同时对时钟漂移和偏差进行了线性回归分析，补偿了时钟漂移带来的误差，同步精度达到了微秒级。FTSP协议还考虑了 root节点的选举机制、root节点失效问题。节点如果超过设定数目的同步周期仍然未收到同步报文就宣布自己是root节点。当多个节点同时宣布自己为root节点时，节点标识ID最小的节点当选为root节点。这样保证同步协议在网络拓扑结构变化和链路或节点的失效时的健壮性。

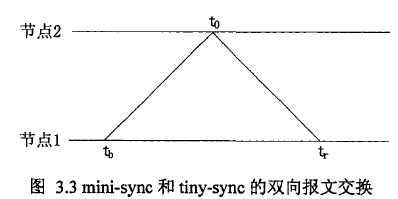
#### 3.4.4.4 mini-sync 和 tiny-sync 同步算法

mini-sync和tiny-sync是两个轻量级的时间同步算法，它们假设节点时钟晶振频率是稳定的，节点的时钟可以表示为理想时间t单调非递减线性函数。利用*ai*表示节点i的时钟频偏， *bi*表示节点/的相偏，则有以下时钟模型:

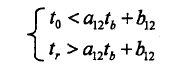


利用*a12*表示两个节点时钟的相对频偏， *b12*表示两个节点时钟的相对相偏，则可以推导出两个节点的时钟关系:





mini-sync和tiny-syne算法釆用双向报文交换同步机制。如上图所示，节点1向节点2发送一个包含发送时刻t0探测报文(probe\_pkt)， t0为利用节点1的本地时钟记录的发送时刻。节点2接收到probe\_pkt后，立即返回包含节点2时钟记录下接收probe\_pkt的时刻tb的消息给节点1，节点1收到消息后利用本地时钟打上一个时标tr这样就得到一个包含三个时标的数据点，显然三个时标满足以下关系:



由上式可以确定参数*a12*和*b12*的取值范围。经过多次探测报文交换可以获得一组数据点，通过求这些数据点约束下的线性规划最优解可以得出*a12*和*b12*的最精确值。

假设节点时钟漂移和偏移恒定，mini-sync和tiny-sync同步算法仅需要很少的存储容量和处理资源，非常适合资源受限的无线传感器网络，但低成本的无线节点一般都配置较低稳定度的晶振，很难保证时钟漂移和偏移恒定。

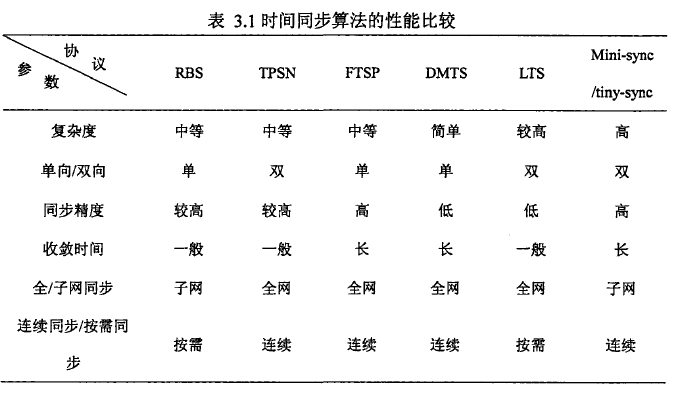
#### 3.4.4.5TPSN协议

传感器网络时间同步协议(Timing-sync protocol for Sensor Networks)类似于传统网络的NTP时间同步协议,可以提供传感器网络全网范围内节点间的时间同步"协议实现的前提条件是:在网络中有一个可以与外界时间精确同步的节点称为根节点,根节点可装配如GPS接收机的复杂硬件部件,并作为整个网络系统的时钟源节点。

#### 3.4.4.6LTS算法

有些传感器网络应用对于时间同步的精度要求并不是很高,也不需要整个网络的所有节点都达到时间同步,这样就可以使用简单的轻量的时间同步机制,通过减少时间同步的频率和参与同步的节点数目,在满足同步精度要求的同时降低节点的通信和计算开销,减少网络能量的消耗轻量级生成树同步算法（LTS，Lightweight Tree-based Synchronization)就适合这种传感器网络应用。LTS算法具有鲁棒性和自配置的特点,特别是在出现节点失效!动态调整信道和节点移动情况下,LTS算法仍能够正常工作"在分析单跳节点对之间基于发送一接收方式的时间同步机制基础上，J.Greunen和J.Rabaey提出了集中式和分布式两类LTS多跳时间同步算法。

#### 3.4.4.5时间同步算法性能比较



## 3.4 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 网络坐标系统

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，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。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。

### 5.1.1 随机延迟污染现象

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)。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

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。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：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，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]。

C

300

100

100

A

B

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。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。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，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

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。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0-100ms之间的记录有约8830万条，约占总体90%，剩余约600万条记录在100-500ms之间，有约90万条时延记录在500-1000ms之间，1000-2000ms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。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：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，即最小时延值)时延用于更新坐标。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，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*RTTi,j*，计算*MeanRTTi,j*：

(3.1)

*MeanRTTi,j*是*RTTi,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α=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*DevRTTi,j*，用于估计*RTTi,j*与*MeanRTTi,j*的偏离程度：

(3.2)

*DevRTTi,j*是*RTTi,j*与*MeanRTTi,j*的差值的EWMA，这里β的推荐值为1/4即β=0.25。

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

## 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。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。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：对节点*i*获得来自节点*j*一个*RTTi,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，则获取节点*i*，节点*j*，节点*k*间的时延，并计算违例系数：

(4.1)

如果*λ*>1时，则认为*RTTi,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*λ*≤1时，则认为*RTTi,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

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，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。其原理是将Vivaldi算法的迭代步作为子步(Sup-step)，将多个子步聚合为一个超步，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。

其感知方法[6]为：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*xi*(*t*)，|*Neightbor*(*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：

(4.2)

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。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；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。

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*l*为放大器增加迭代步长，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*(*ε*, *cc*×*h*)，跳转到子步Sup-step5

子步Sup-step4：让*cc* = max(1, *cc*×*l*)

子步Sup-step5：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，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，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。将WS和WC中储存的节点坐标统称为系统级坐标，则只有在系统级坐标的变化程度超过规定数值*Υ*时，才对节点更新其节点应用级坐标X。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。

(4.3)

其中a，b分别为WS和WC中储存的节点坐标，只有当*e*(*WS*，*WC*)的值超过规定数值*Υ*时，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。

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*Υ*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)。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*RTTi,j*时，首先进行随机延迟污染的检测，如果*RTTi,j* > *MeanRTTi,j* +4·*DevRTTi,j*则视为出现了随机延迟污染，此时的*RTTi,j*可能是极大的，不具有可靠性，此时先计算*DevRTTi,j*与*MeanRTTi,j*的值，然后让*RTTi,j* = *MeanRTTi,j*进入下部分，从而减少不可靠的延迟*RTTi,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*RTTi,j* ≤ *MeanRTTi,j* + 4·*DevRTTi,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。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，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。

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。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*RTTi,j*替换为*MeanRTTi,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*=|*Neightbor*(*i*)|，即节点*i*的邻居节点数)，让此时节点*i*的坐标为*xi*(t),本文的坐标抖动感知方法为：

(5.1)

当*errt*>*errt-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*RTTi,j*≤ *MeanRTTi,j* + 4·*DevRTTi,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

在三角不等式式违例现象[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，本文的抑制算是对于越长的时延，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：

(5.2)

(5.3)

*ln*(*x*)是以自然常数*e*为底的对数。而*d*的值在(0,+∞)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*errt* > *errt-1*之后，并且*RTTi,j* ≤ *MeanRTTi,j* + 4·*DevRTTi,j*时才执行；当*RTTi,j* > *MeanRTTi,j* + 4·*DevRTTi,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*RTTi,j*，稳定抑制算法的执行步骤为：

步骤1：进行初始化，让*d* = 1，计算器*n* = 0，让*errt-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*RTTi,j* > *MeanRTTi,j + 4·DevRTTi,j*则执行一次步骤1和步骤3，之后让*RTTi,j* = *MeanRTTi,j*后，跳转到步骤6；

步骤3：计算*DevRTTi,j*与*MeanRTTi,j*的值：

(5.4)

(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；否则计算器*n* = *n* + 1，如果*n* < |*Neightbor*(*i*)|，跳转到步骤6。

步骤5：让*n* = 0，计算*errt*：

(5.6)

若*errt* > *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*xi*。

### 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

定义相对误差RE用于度量算法的准确性：

(5.7)

其中*Xi*与*Xj*分别为节点*i*与邻居节点*j*的坐标，*RTTi,j*为节点间实测时延，||*Xi*-*Xj*||为节点间预测时延，MIN(a,b)表示a，b间的最小值。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。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，说明了稳定抑制Vivaldi算法的准确性比Vivaldi算法更高。

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。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：

(5.8)

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：

(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

对于节点*i*，其坐标*xi*在*m*次更新后的期望值*ui*，以及*xi*与*ui*的距离方差*si*，计算公式如下公式(5.10):

(5.10)

其距离方差*si*反映了节点*i*的抖动程度，方差*si*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*：

(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。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。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TO—Filter随机延迟污染抑制方法。

由于网络中普通存在的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。因而抑制TIV现象，减少节点坐标的抖动，对于提示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，并对其优劣点进行了分析。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，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。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，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，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，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：

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TO—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，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TO-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TO-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。

## 6.2 机遇

## 6.3 挑战

引文

致谢

漫漫人生路，求学十余载。转眼间，我在武汉大学国际软件学院已经度过了四个精彩而充实的春华秋实，回顾以往的学习和生活，感慨万千。短短几年间，我已经从一名不谙世事的珞珈少年，成长为自强弘毅求是拓新的武大青年。我有幸认识了很多学生渊博的老师和优秀的同学，你们是我本科生活中最大的收获与财富。正是你们的帮助与陪伴，我才能顺利完成学业，为大学阶段的学习画上一个句号。

“饮其流者怀其源，学其成时念吾师”。在本文完成之际，首先我要中心的感谢我的导师李兵教授。师从李兵老师虽不足一年，但我深为他那高深的学术造诣、严谨的治学态度、平易近人的师长风范所折服，也为他敏锐的学术眼光、儒雅的个人情怀、博大的胸襟抱负所惊叹。在与李老师的每次交流中我都能获益良多；感谢李老师在我开题、中期及答辩过程中为我指出问题，并严格要求，提出中肯的意见。在恩师的无私帮助和鼓励下，毕业论文才得以顺利完成。恩师言传身教，使我铭感五内。

作为空间信息与数字技术专业的一名学子，同时我要感谢培育了我四年的各位专业课老师，感谢边馥苓教授开创空间信息与数字技术专业，填补了我国空间信息方面软件人才培养的空白；其次我要感谢的是江聪世教授，犹记得当我本科二年级敲开江老师实验室门的时候，江老师和蔼可亲的笑容接纳了我，并给我提供了一个优异的学习与研究场所，引领我步入了科研的殿堂，作为创新创业项目的指导老师，为我指点迷津，江老师对我期望甚高，要求甚严，无奈本人愚钝，未能在学术上有所作为，有辱师望，心中万分惶恐，回首过往，在实验室的日子是我最难以忘怀的时光；熊庆文教授将我引入移动互联网的浪潮，同时为我提供勤工俭学的助教岗位；感谢谭喜成教授、王少华教授和李晓雷老师在课堂内外为我提供项目上的指导。感谢空间信息与数字技术专业的所有任课老师对我的悉心教导，你们渊博的学识和严谨的态度让我受益匪浅。同时我还要感谢蔡恒进教授在微软创新杯中的对我和我们团队的指导，您的意见总是具有前瞻性与启发性，让我不断的反思自我，超越自我，最终取得佳绩。

感谢国际软件学院的各位领导为我提供了一个国际化的平台，感谢黄治国书记与崔晓晖院长的辛勤工作；感谢于敏副书记的知遇之恩与谆谆教诲，感谢邢涛老师、金娟老师、程晓婷老师和周小萍老师对我的培养与教育，我在国际软件学院四年来的每一次成长都离不开你们在背后付出的汗水和努力。尤其是要感谢的是我的辅导员周小萍老师和程晓婷老师，你们不仅为我指引了人生的方向，还帮助我脚踏实地的走好脚下的每一步，受益终生。

感谢所有帮助过我的师长和课题组的熊伟、何鹏、张得光、刘晓师兄。

最后，我要深深地感谢我的父母。二十多年来，他们不辞劳苦，起早贪黑，给予了我求学的支持和舒适的生活环境。然而在外求学数载，聚少离多，每每念及，总是心怀愧疚，难以自已。谢谢你们的付出，我必将不负你们的殷切期望！

最后，附上朱熹的《劝学诗》一首，以期自勉。

少年易老学难成，一寸光阴不可轻。

味觉池塘春草梦，阶前梧叶已秋声。

2015年6月

赵玉琦 珞珈山下