分类号 _____ 密 级 _____

U D C _____ 编 号 10486

武漢大學

硕 士 专 业 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

摘　要

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。为了更好数据隐私保护，本文基于北斗卫星导航和北斗地基增强系统，利用软件定义网络（Software Defined Network, SDN）这一新型的网络架构搭建仿真实验环境，构建起高精度时空信息网络坐标系统，并对系统性能进行优化。

本文首先通过将北斗卫星定位和北斗地基增强系统和SDN相融合，实现实验环境下所有网络的节点的精准定位和授时时间同步，利用SDN架构，实现数据平面和控制平面的分离；通过对OpenFlow协议的修改，设计了一个Vlan的标签分配机制，实现了对网络内数据具有时空信息标注（“时空戳"），进而实现数据在网络坐标系统内的精准定位和路径追踪。

通过大量网络测量实验数据分析，网络中随机延迟污染现象以及违反三角不等式（TIV）现象是影响网络性能和准确性的重要原因，为了对网络性能进行优化，对于随机延迟污染现象，本文改进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。经过对近2000个节点的9000万条数据的仿真实验表明，改进后稳定抑制Vivaldi算法能够在保证准确性的同时，抑制坐标的抖动，其抑制能力提示了8.7%，提高了网络性能。

关键词： 大数据，时空戳，北斗导航系统，软件定义网络（SDN）

Abstract

Big data is profoundly affecting people's production methods, living habits, thinking patterns, and research methods. Big data is not only a front-of-the-range subject in the academia and industry, but has also become a national basic strategic resource. The leakage of privacy brought by big data brings serious losses and potential risks to users, which greatly undermines the social economic order and affects the industrialization and application of big data such as big government data, business big data, and healthy big data. At present, big data has risen as a national strategic resource, and protection of big data security and privacy is not only an international academic frontier, but also a major national strategic need. This project is devoted to the basic research of data protection, privacy protection, and digital watermark hiding in the network big data environment. It has important theoretical significance and application value.

In the big data environment, due to the more extensive correlation of data, the original isolated information becomes private. At the same time, the data processed by traditional anonymous technologies, etc., may still be analyzed after data mining and deep analysis through big data association. The privacy of users poses new challenges for privacy screening and privacy protection technologies in big data environments.
Construction of network big data environment with high-precision spatio-temporal information: Based on the data and privacy protection theories and methods proposed in this project, and for location big data, the data of the spatiotemporal information in the big data environment is constructed through Beidou's high precision positioning time synchronization technology. Verification environment with privacy protection theory.

Data analysis illustrates the prevalence of random delay pollution, introduces an existing random delay pollution suppression method MP-Filter, and proposes a TO-Filter random delay pollution suppression method. Two factors that degrade network performance are analyzed: random delay pollution phenomenon and TIV phenomenon.

At the same time, several attacks that affect the security of off-center network coordinate system are introduced. Three existing network coordinate distance prediction algorithms for suppressing TIV phenomenon are analyzed, and their advantages and disadvantages are analyzed. At the same time, a stable suppression Vivaldi algorithm is briefly introduced. A stable suppression Vivaldi algorithm for suppressing random delay pollution and TIV phenomena was proposed. The accuracy of simulation and jitter suppression capability were analyzed through simulation experiments.

Keywords: Big Data, Spatiotemporal Stamp, Beidou Navigation System, Software-Defined Network (SDN)

目 录

第一章 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。本文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。

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。

1.2 国内外研究现状和发展趋势

1.2.1 网络空间中的时间基准

移动互联网、信息-物理融合系统依靠时间同步技术建立并维持整网的统一时间基准，时间同步包含了时钟同步和频率同步两层含义。大规模网络高精度时间同步技术一直是网络通信等领域的研究热点和难点。通信传输速率和移动性的快速增长，都对时间基准提出了更高的要求[1-3]。通信技术从固定到移动、从低传输率到高传输率、从点对点到多点对多点、从单一网络到异构多网络等发展过程中（图 2-1），时间同步一直起着非常重要的作用：移动互联网中 FDD 无线系统（如 WCDMA）仅需节点间的频率同步即可，而 TDD 无线系统（如 LTE 等）则需要精确的时钟同步实现漫游和切换；在信息-物理融合系统中，由于物理世界的连续时间与信息世界的离散时间具有明确的鸿沟，因此通信需要更为精确的时间基准（如卫星导航系统中卫星之间的时钟同步要达到十纳秒级），且时间同步渗透到系统计算、网络和控制的各个层面[4,5]。

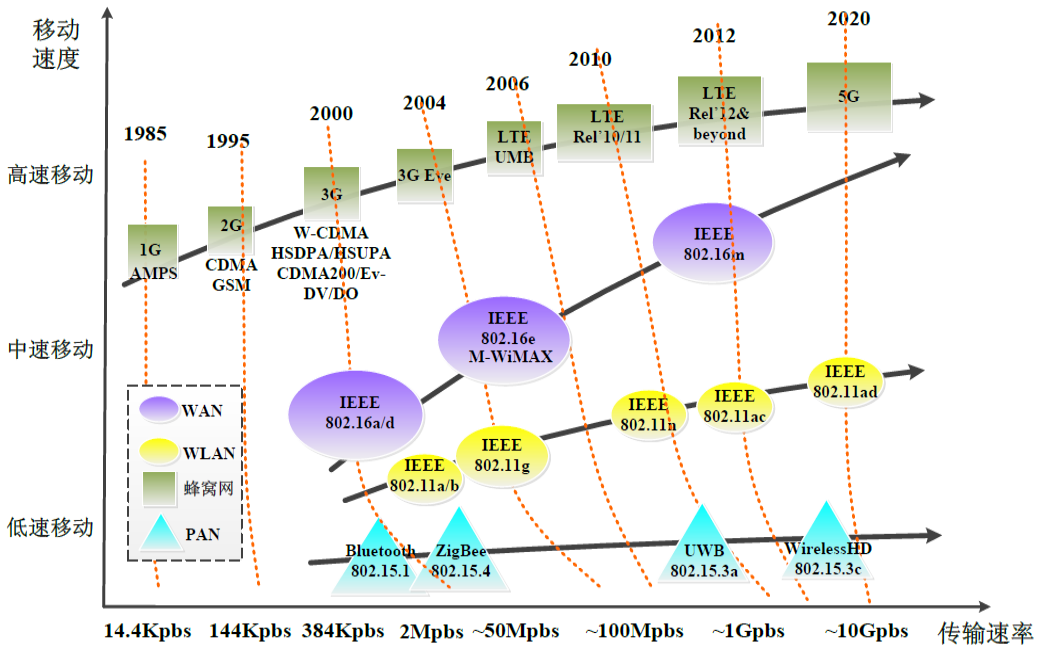


图 1-1 移动时代的通信系统速率发展

1.2.2 网络空间中的空间基准

在网络空间中建立空间基准，就是借助地理空间位置信息建立网络坐标系统。网络坐标系统研究是互联网领域中与地理位置有关的热点问题，但在移动互联网领域则刚刚开始。Michael 和Karthik 研究发现互联网主机网络距离和主机地理距离基本成正比关系[6]，通过建立网络距离空间与欧式空间的映射关系建立了网络坐标系统。网络坐标系统可用于刻画网络属性的时变分布规律与性质、性能状况、以及节点间相互关系等，在最优传输路径选择，增加传输速率， 增强网络稳定性等方面具有重要作用。在Google CDN 中采用了基于欧氏距离的中心式网络坐标系统GNP［7]，而分布式网络坐标计算在P2P 文件共享、分布式哈希和应用层路由中也有着广泛应用。项目成员李星等曾提出了几类网络坐标计算模型，具有较高的准确性以及对节点抖动的鲁棒性，在互联网广播系统中获得较好的应用效果[8]。

目前移动环境下的网络坐标系统的研究刚刚开始，尚未取得突破。与互联网的桌面环境不同，几乎所有移动终端和基站都具备实时定位能力，可获得不同精度的空间位置信息，可以借助这些信息构建移动环境中网络坐标系，研究网络拓扑、资源与业务分布等，关键是如何在大量广义精度谱的位置信息中提取移动终端的精确位置，本项目拟在此领域进行深入研究。

1.2.3 网络信息时空戳

网络空间中构建时空基准后，网络信息即可用对应的时空信息所标识，即加上时空戳，此概念类似于分布式系统中的时间戳。时间戳是计算机系统记录事件发生的时间信息，通常用来判定计算机系统中的事件发生顺序，在分布式系统、数据库事务处理、信息安全中具有极其重要的地位。

基于地理位置的路由是空间戳的研究热点。针对地理位置路由的研究大多集中于无线传感器网络，用于无线传感网与互联网的异构组网。其核心思路是：数据传输机制和路径由地理位置决定，网络节点不用时刻交换链接信息来维持网络。在大数据应用的背景下，数据和地理位置往往紧密关联，在对网络中的通信和计算资源进行统一调配和协同过程中，必须考虑地理位置信息。现有理论和技术不

能很好地满足实际需求，如何利用更加丰富的时空信息来优化网络路由效率成为一个重要研究方向。

1.2.4 北斗精准时空体系的建立

北斗卫星导航系统（BDS）是我国自主建设的全球卫星定位导航系统，我们国家拥有完全控制能力，与 GPS、GLONASS、GALILEO 并称世界卫星导航四大服务提供商。BDS 的坐标框架采用中国 2000 大地坐标系统（CGCS2000）的时间基准为北斗时（BDT），坐标系统和时间基准的定义与维持均采用国内独立自主研发的核心技术。截至到 2016 年 12 月 23 日，北斗卫星导航系统已成功发射 23 颗卫星，分析及大量实测数据表明：BDS 在中国及周边地区可提供平面 10m，高程 10m，单机授时 50ns 标准定位和授时，以及 120 个汉字/次的短报文通信服务[9, 10]。 北斗地基增强系统又称为北斗连续运行参考站网络，是建筑于地表的，由感知时空信息的基准站、通信网络、多级数据中心、移动互联网和带有北斗接收处理功能的用户终端构成的，提供定位定时服务的系统，也是北斗卫星导航系统重要的地面基础设施[11]，本质上属于信息-物理融合系统。通过采用实时载波相位差分、广域精密定位等技术，北斗地基增强系统可以向覆盖范围内的用户提供平面优于 10cm，时间同步精度优于 1ns 的时空信息[12, 13]。

2000 年至今我国已经建设接近 3500 个 GNSS 基准站，分属 100 个独立运行的系统，由于年代久远绝大部分不能接收并处理北斗信号。在《国家卫星导航中长期发展规划》提出：统筹建设国家统一的多模连续运行参考站网，为各类用户导航增强服务提供支撑；基于该系统建设基于位置服务的互联互通、门类齐全的基础平台，为国家、社会和行业共享应用提供支撑服务。因此改造升级现有资源组建“全国一张网”不仅是用户的实际需求，更是国家的重大工程与战略需求，最终是要实现向全国用户提供从毫米级到米级的广义精度谱的可靠安全泛在位置服务。 2013 年 3 月，由国务院中国卫星导航管理办公室批准的全国首个北斗地基增强系统示范项目—湖北省北斗精密定位服务系统在武汉通过验收，本项目的示范验证将基于该系统构建而成，为本项目研究提供精准的时空信息，同时又对研究成果进行实验验证。

1.3 研究问题的提出及意义

移动互联网、信息-物理融合系统依靠时间同步技术建立并维持整网的统一时间基准，时间同步包含了时钟同步和频率同步两层含义。大规模网络高精度时间同步技术一直是网络通信等领域的研究热点和难点。

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。本文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。

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。

1.4 主要工作内容及研究框架

首先，在北斗高精度时空信息感知理论与方法的指导下，探索北斗高精度定位时间同步技术，建立整网高精度时间同步数学模型，突破北斗高精度定位时间同步终端关键技术；研究与网络融合的终端集成技术，构建基于北斗卫星导航系统的网络空间高精度时空体系。研究系统监测评估理论，突破系统完备性监测技术，实现对时空信息精度的实时监测和评估，确保所构建精准时空体系的可靠性。

然后，基于高精度时空体系，从网络信息与网络空间两个层次研究精准时空信息的融合理论。一方面，研究网络信息单元中精准时空标识的嵌入，修改现有网络协议，改时间戳为“时空戳”，实现对信息的空间精准定位。另一方面，从几何与逻辑两个角度研究网络信息空间的表征与建模，实现基于精准时空信息的网络坐标系统以及信息空间的时空统一建模。设计分布式、地域邻近聚类的分层式网络坐标计算系统，采用矩阵分解模型的计算方法，降低因TIV因素带来的预测误差，探索利用移动设备实时定位能力提供精确度补偿的网络坐标计算方法。

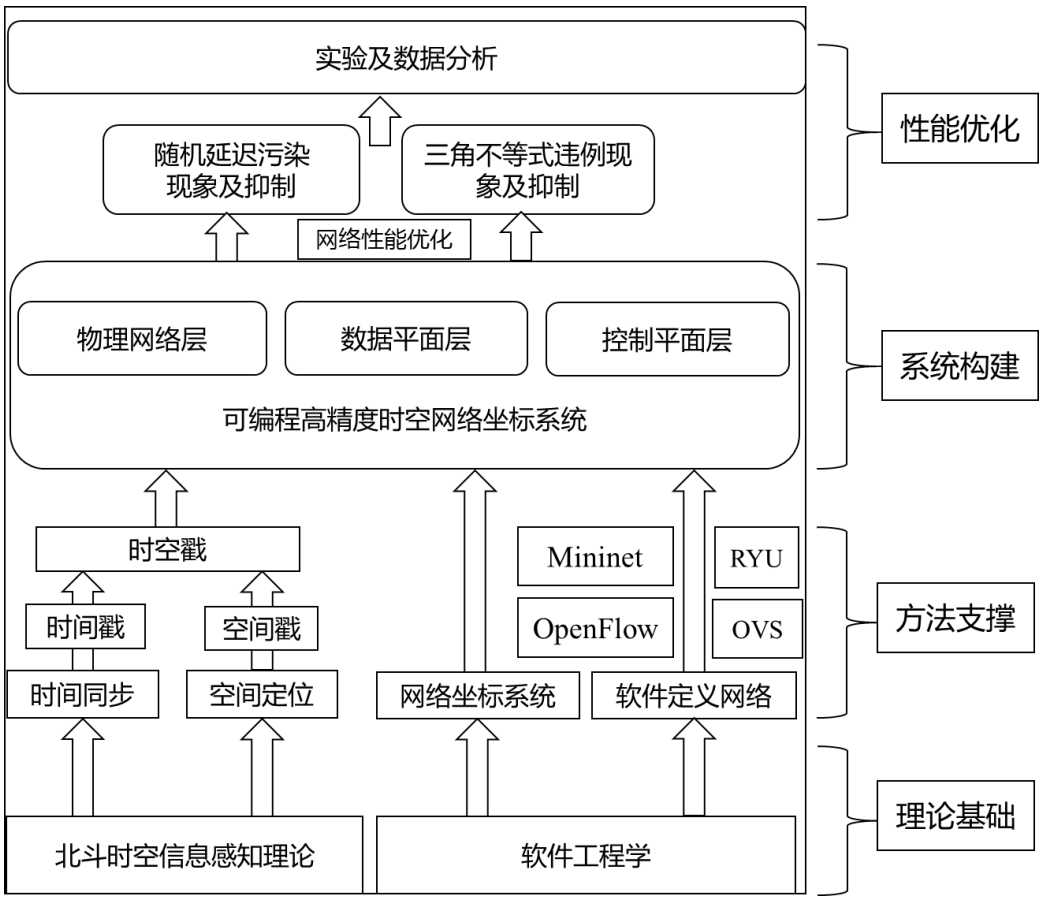


图 1-2 研究内容框架图

1.5 论文的组织结构

本文分为五个章节，主要内容如下：

第一章 绪论：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。

第二章 项目相关技术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、PIC、NPS和Vivaldi，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。

第三章 高精度时空网络坐标系统的设计与实现：基于实现了北斗系统实现了网络时间同步和定位，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。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。

第四章 时空网络坐标系统的性能分析及优化：网络中随机延迟污染现象以及违反三角不等式（TIV）现象是影响网络性能和准确性的重要原因，为了对网络性能进行优化，对于随机延迟污染现象，本文改进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。经过对近2000个节点的9000万条数据的仿真实验表明，改进后稳定抑制Vivaldi算法能够在保证准确性的同时，抑制坐标的抖动，其抑制能力提示了8.7%，提高了网络性能。

第五章 全文的总结：归纳和总结本论文所做的研究工作，分析存在的问题，并提出今后的工作设想。

第二章 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。为此，人们提出了网络坐标系统，来提高互联网距离测量效率。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之一，其核心思想是将网络中节点实测时延映关系射到数学的几何空间上，从而将获取网络节点间时延转化为获取节点坐标间距离。不难看出，其思想是将网络构建在一个几何空间上，如多维欧式空间，这样一来，网络中任何一台主机的位置就可以看作是该几何空间中的一个点。

在众多成熟的网络坐标系统中，GNP 是典型的的基于锚节点的时延预测机制来构建的网络坐标系统，主机在网络中分可以为两个部分：一个部分是事先将在网络中均匀分布的节点选取出来，作为锚节点(Landmarks)，它们会首计算并确定自己在几何坐标中的坐标，基于此，它们就可以用于作为其他普通节点坐标定位的参考。无论其他节点怎么变化，锚节点始终保持自己的坐标，同时散布给所有任何想要参与进来的主机节点。另一部分是网络环境中的普通主机节点，它们的坐标主要通过获得锚节点的坐标，来计算确定自己的坐标，因此该方法可扩展性很强。

在利用 GNP 构建网络坐标系统时，首先要做的是在网络中确定小部分主机节点作为网络中的特殊节点--即为锚节点，其作用是在坐标系统中为其它非锚节

点主机提供一组必要的坐标参考。

假设有 N 个锚节点 $P_1, P_2 \cdots\cdots P_N (N > d$，$d$ 为几何空间的维度)，使用 Ping 方式可以很容易的获得锚节点间的实测时延 $L_{i,j}$，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 $X_1, X_2 \cdots\cdots X_N$ 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。

$$Error_{N\times N} = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left(\|X_i - X_j\| - L_{i,j} \right)^2 \tag{2.1}$$

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。

$$Error_X = \sum_{k=1}^{N} (\|X - X_k\| - L_k)^2 \tag{2.2}$$

GNP 系统坐标定位系统是集中式的，可扩展性很强，网络距离预测准确性较高。但是也有不足之处，在 GNP 系统中，一旦有新的主机进入其中，系统就需要计算这个主机就需要测量到其他所有锚节点的距离，开销太大，因次可能会给节点带来过于沉重的负担，同时锚节点的分布情况、数量、位置等因素也会对系统的距离的预测准确度产生一定的影响。

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 系统极为相似的是，PIC 在设计中也是借鉴了将网络节点映射到几何空间的策略，当新的节点 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，与锚节点间的实测时延 L_k 同样可以使用 Ping 方式可以很容易的获得，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（2. 3）的值最小。

$$Error_X = \sum_{k=1}^{M} \left(\frac{\|X - X_k\| - L_k}{L_k} \right)^2 \tag{2.3}$$

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。

$$Error_{N \times N} = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left(\frac{\|X_i - X_j\| - L_{i,j}}{L_{i,j}} \right)^2 \tag{2.4}$$

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：首先选定一个暂时的目标节点，这个可以从已知坐标的节点中随机的选择，接着对该节点与其邻居节点的距离进行测量，假如通过测量，发现有距离自己更近一个邻居节点存在，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，普通节点和锚节点。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；其他的节点就可以看作是普通节点，与此同时这类节点也有可能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 L_i，则其选取的参考节点 j 所在层数 L_j 需要满足 $L_i < L_j$，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：

1)在普通节点加入系统时，通过计算可以从服务器节点获得参考节点的信息和更详细的系统参数。

2)通过测量得到普通节点与参考节点间的真实测量距离。

3)对普通节点坐标进行计算，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。

这样一来，每隔一段固定时间，节点自身的坐标将会被重新计算，使其适应不断变化的网络拓扑环境。

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 算法的核心思想是将网络节点利用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”，映射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。

可以看出，Vivaldi[3]是一种完全分布式的算法，节点仅需要获得少量的任何具有坐标的邻居节点的实测时延就可以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 $RTT_{i,j}$:

1)计算误差权值 ω：

$$\omega = \frac{e_i}{e_i+e_j} \tag{2.5}$$

2)计算两点距离相对误差 e_s：

$$e_s = \left|RTT_{i,j} - \|x_i - x_j\|\right| / RTT_{i,j} \tag{2.6}$$

3)更新自身误差估计 e_i：

$$e_i = e_s c_e \omega + e_i(1 - c_e\omega) \tag{2.7}$$

4)更新本地坐标 x_i：

$$\delta = c_c\omega \tag{2.8}$$

$$x_i = x_i + \delta\left(RTT - \left\|x_i - x_j\right\|\right)u\left(x_i - x_j\right) \tag{2.9}$$

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，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。

在众多网络坐标系统的构建算法中，Vivaldi[3]算法具有优秀性能是公认的，其优点得益于实现过程是全分布式的，无需额外基站设施的设立，在 Internet 拓扑变化的适应性上有较好的表现，网络坐标能够及时更新，通过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，以减少了网络计算过程中造成的不必要的开支。

2. 2 时间同步

2.2.1 授时

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。

一般来说，要成为一个授时系统必须具备两个条件：（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。（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。作为一个授时系统，不应限制用户的使用数量。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。

2.2.2 守时

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：

首先是用什么样的“钟”来产生并保持一个稳定的时间尺度。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。

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。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。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，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。

第三个方面是本地时间比对和远距离时间同步。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。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。

2.2.3 时间同步

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），通过某种方式使其时钟的时刻和时间间隔与 UTC 同步。

2.2.4 时间同步的必要性

2.2.4.1 网络管理系统

目前网络管理系统主要有各大通信网自建的集中系统组成，在他们各自的系统中，可以实现对网络数据的统计分析，各类性能指标也都能了如指掌，通过采集到的网络性能指标数据来排查网络错误和发出警告。一旦在网络中发现警号信号，就能够第一时间做出相应，并且在与之有联系的网络设备和资源上也挂出警告。由于网络设备中包含时间戳，这些时间戳都是在各个设备上的时间钟打上的，如果我们把全网的时间同步，那么我们就能根据时间信息得知最先出现问题的设备是哪一个，这样就可以从时间维度上对故障的源头进行定位，有利于网络资源故障定位和事故排查。

2.2.4.2 七号信令监测系统

针对这个问题，有人提出了可以采取在信令流量较大的节点上设置监测，通过检测系统，对信令流进行监控和分析，来准确定位通信信息网中各类警告，主要包括故障类型和故障点。具体的处理过程是，如果发现网络出现了异常警告，

与警告发起设备相关联的所有信令流量就会被系统甄别出来，进行数据分析。同理，信令流量的时间戳也是有各自网络基础设施设备产生的，所以如果各个设备采集点的时间同步，全网信令也能够实现同步。

2.2.4.3 安全认证系统

从网络安全的角度讲，通信网安全的优化离不开时间的同步，在安全认证系统的前提先，通信信息安全才能在传输过程中得到保障。然而，目前的通信网之间只能实现内部网路同步，无法实现全网时间同步，所以安全认证系统的构建也有一定的难度。时间同步对网络安全有至关重要的作用。

2.2.4.4 提高服务质量方面

通信网的服务质量也与时间同步密不可分，网络服务计费通常都是以时间计量维度的，一方面通信网内的时间同步与否会影响网内计费，另一方面还体现在不同的通信网间的时间同步基础上。目前的标准下，每个交换机在通话时会进行通信计时，记录下通话开始时间、结束时间和持续时间，但是因为不同交换机之间没有时间同步，时钟准确度低，不同的交换机一个月内的时间偏差可能会超过30 秒。

2. 3 北斗 GNSS 卫星精密授时

2.3.1 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 单向授时示意图如下图 2-1 所示：

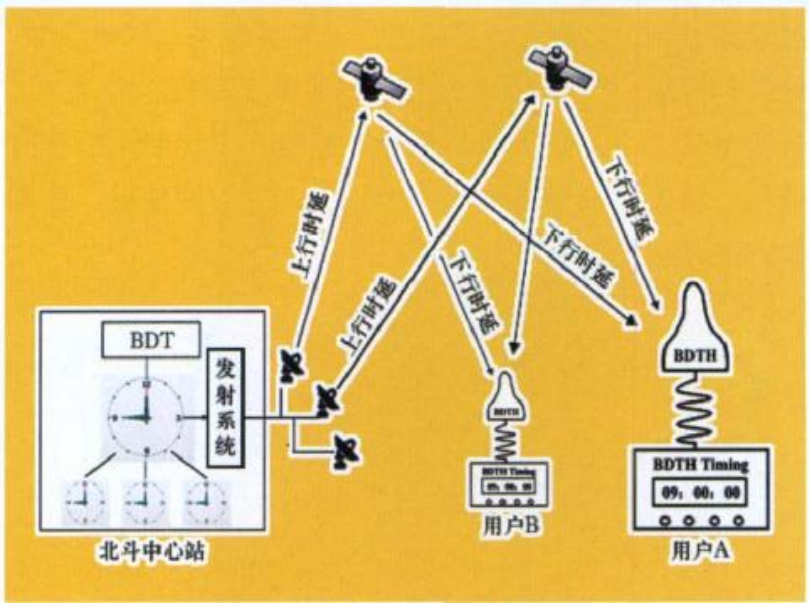


图 2-1 北斗卫星 RDSS 单项授时示意图

小于 1 秒的小数秒部分的 BDT 的时间传递过程如图 2-2 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：

$$\delta^{RD} = \xi^{RD} - \Delta t^{RD} - ({\tau_1}^{RD} + {\tau_2}^{RD} + {\tau_3}^{RD} + {\tau_R}^{RD}) \qquad (2.11)$$

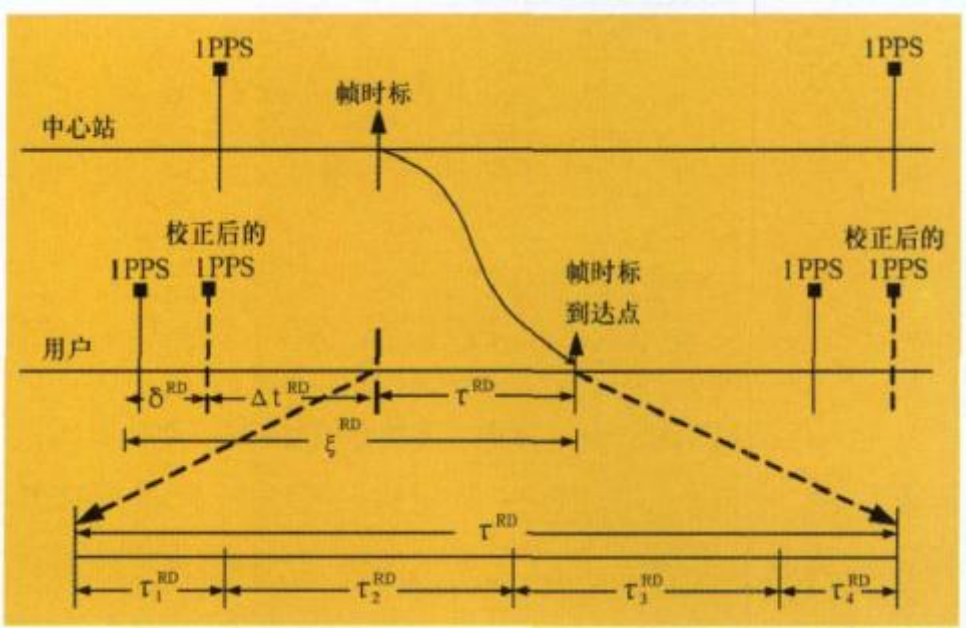


图 2-2 RDSS 单项授时原理图

对本地钟输出的 1PPS 进行移向处理调整，使得时间差δ^{RD}趋近于零，此时 BDT 的 1PPS 实现时间（相位秒部分）就与本地终端 1PPS 同步。单向授时精度优于 100ns 是北斗卫星 RDSS 的特点，并且不限制具体的用户量，从而使的绝大多数的时间内，用户的精度需求都能够得到满足。

2.3.2 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 双向授时的方法示意图如图 5 所示。

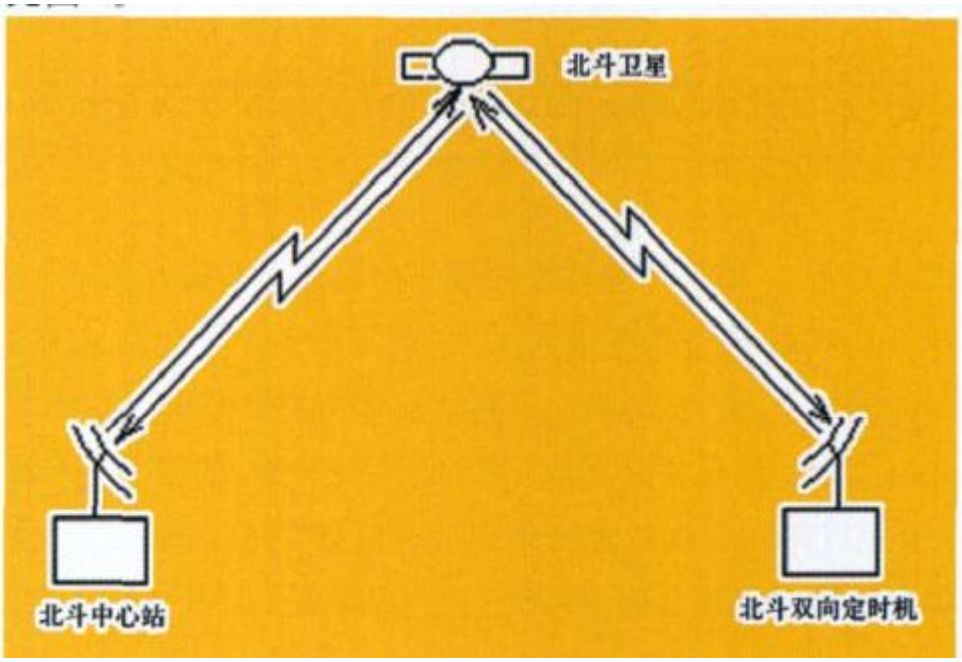


图 2-3 北斗卫星 RDSS 双向法授时示意图

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 ΔT 。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 $n\Delta T$ 。如前所述，RDSS 的帧周期为 31.25ms，即 $\Delta t = 31.25ms$；同时 $n\Delta T$ 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 $\Delta T'$；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 $\Delta 2$，并算出该信号由中心发出至用户机收到的正向传播时延 τ，再将 τ 发送给该用户作为双向定时时延改正值。由于 $\Delta T'$ 可以由用户直接测定，因此只要给出传播时延 τ，就可以得出用户终端时钟与 BDT 钟差 $\Delta T = 1 - \Delta T' - \tau - n\Delta t$，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。

2.3.3 北斗 RNSS 单向授时

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，从而实现授时是北斗 RNSS 单向授时的本质。只需 1 颗卫星即可在接收终端位置已知的情况下实现授时。授时过程与伪距单点定位/精密单点定位在测站坐标未知的情况下过程相同。

对于卫星 i 的伪距观测方程如下：

$$\mathrm{P} = \rho - dt^{S} + dt_{R} + dT + dI + \varepsilon \tag{2.12}$$

式中：P 为伪距观测值；ρ 为测站与卫星间距离；dt^{S} 为卫星钟差；dt_{R} 为接收机钟差；dT 为对流层延迟；dI 为电离层延迟；ε 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。如果不能确定接收终端坐标，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。为了提高授时精度，可以采用相位平滑伪距或载波相位观测值，这样授时精度可以达到 10ns 以内。

2.3.4 北斗/GNSS 共视法授时

北斗卫星共视法时间传递与之前所述方法略有不同，即利用位于不同地点的两个（或多个）观测站，对同一颗或多颗北斗卫星的同一时标信号使用北斗卫星信号接收终端同时观测，经观测结果传递交换和再处理，不同的两个观测站在异地就可以实现相互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，远距离卫星时间传递的精度得以提高。

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：

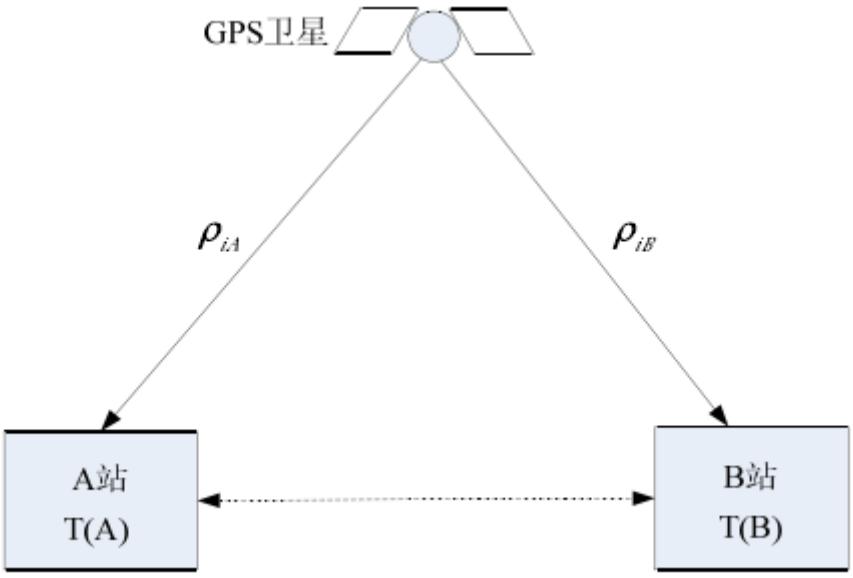


图 2-3 北斗卫星共视法原理图

若卫星 i 被 A、B 两站放置的接收机同时观测到的伪距为P_A和P_B，可得其接收机钟差分别如下如下：

$$dt_A = P_A^i - \rho_A^i + dt^i - dT_A^i - dI_A^i + \varepsilon_A^i \quad (2.13)$$

$$dt_B = P_B^i - \rho_B^i + dt^i - dT_B^i - dI_B^i + \varepsilon_B^i \quad (2.14)$$

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

作差可得：

$$dt_{AB} = dt_A - dt_B$$

$$= P_A^i - P_B^i - \rho_A^i + \rho_B^i - \left(dT_A^i - dT_B^i\right) - \left(dI_A^i - dI_B^i\right) + \left(\varepsilon_A^i + \varepsilon_B^i\right) \quad (2.14)$$

可以通过接收机位置和卫星星历计算得到ρ的值，卫星钟差可完全消去，同时考虑到测站之间的距离比较近，因此对流层延迟、卫星轨道误差和电离层延迟之间具有较高的延迟性，其大部分影响可以在做差后得到消除。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。

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。用户可以通过使用该偏差数据后，来得到与国家标准时间之间的偏差。

下图为其原理：

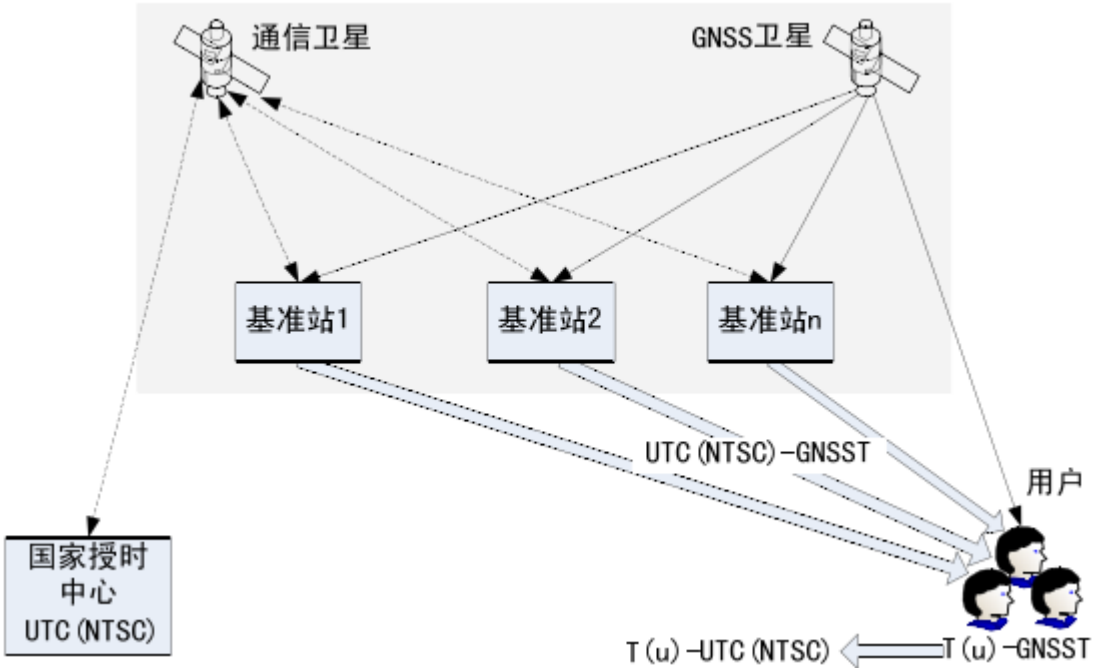


图 2-4　授时流程图[14]

假设基准站有个，每个基准站与标准时间中心的授时和同步都是通过卫星双向时间频率传递与标准时间 UTC(NTSC)来实现的。同步后的基准站接收 GNSS 卫星授时信号并进行检测，得到的时间差 UTC(NTSC)－GNSST 就是每颗卫星广播的系统时间与标准时间的差。此时经过系统的综合处理，用户就可以通过广播得到含有标准时间信息的时差数据。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 广播给用户，这时用户可以利用接收到的标准时间与 GNSST 的时差 UTC(NTSC)－GNSST 对本地的时间进行修正，以得到修正到标准时间，用户本地时间与标准时间就可以实现同步。

$$\mathrm{T}(u) - UTC(NTSC) = (T(u) - GNSST) - (UTC(NTSC) - GNSST) \quad (2.15)$$

该方法解决了传统的共同观看时间传输过程中存在的实时问题。在 GNSS 公共视图时间传输中，由于用户接收机时钟不稳定，因此有必要连续跟踪卫星一段时间。平滑跟踪数据可最大限度地减少随机误差的影响，如接收机时钟抖动和测量噪声。 共视授时法利用国家标准时间和 GNSS 系统时间的高度稳定性，实时向用户广播时间模型参数，不受跟踪时间段的限制，解决了实时性问题。其次，与共视授时法共享的数据交换方式不同，基于共视授时法原理的新方法以广播形式发布时间模型参数信息，所有用户都可以接收，所以用户数量不受限制。另外，

用户只需要配置一个单向设备即可获得央视通过的纳秒级定时精度。

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 Clean Slate 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整合后的发

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

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：

2.4.2 SDN 架构

根据 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 2.5 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。

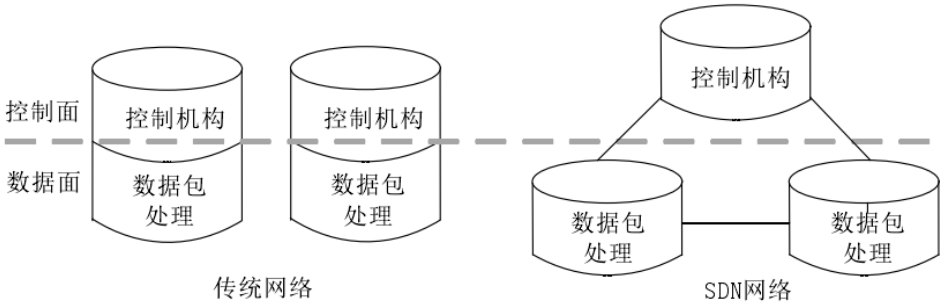


图 2.5 传统网络与 SDN 的结构对比图

图 2.6 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

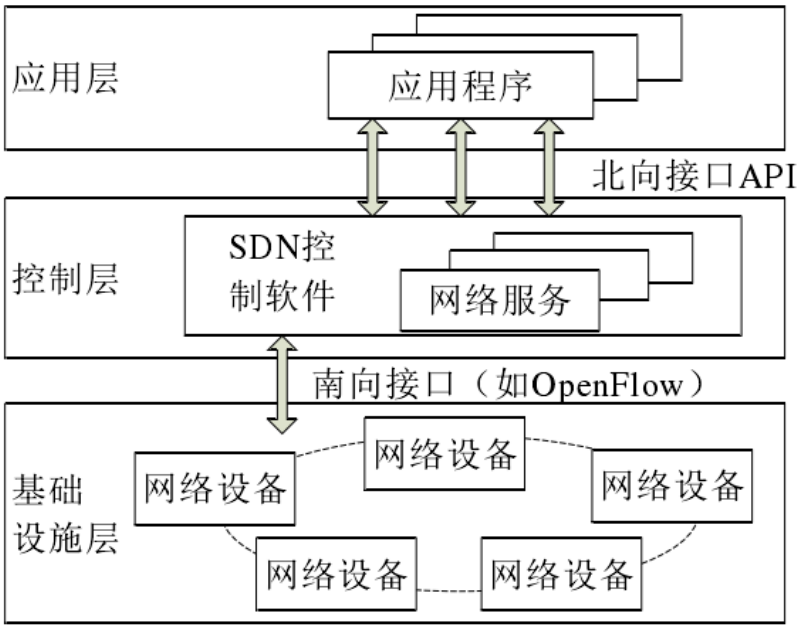


图 2.6　SDN 的结构图

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。

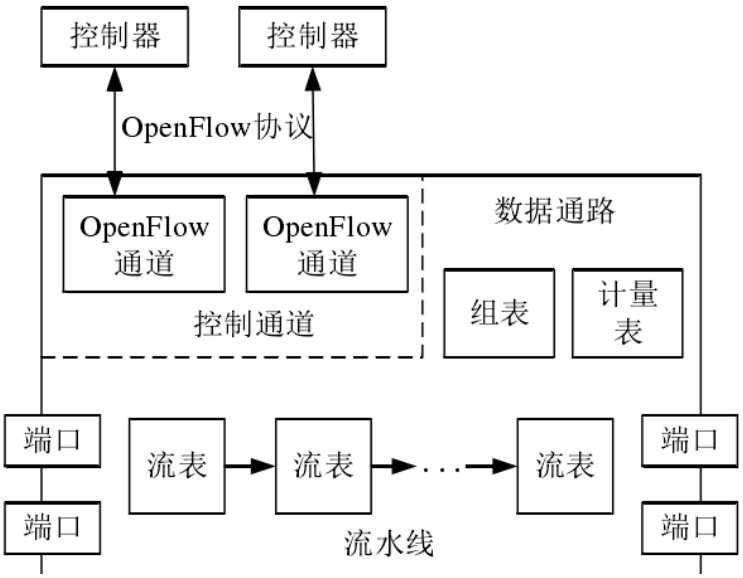


图 2-7 OpenFlow 交换机的结构

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。

2.5 本章小结

本章介绍了用到的相关技术和概念，他们分别是网络坐标、时间同步、北斗导航和精密授时、GNSS 卫星导航和精密授时，同时介绍了软件定义网络（SDN），为后续工作开展提供了必要条件。首先在网络范围内构建网络坐标系，然后对网络时间进行同步，通过北斗卫星来进行精准授时和定位，这样整个网络就是基于高精度时空信息的网络了，最后在 SDN 的网络环境下构建实验。

第三章 高精度时空网络坐标设计与实现

3.1 功能需求与模型构建

网络空间中精准时空体系的构建由于移动互联网、信息-物理融合系统等构成的网络空间不具备欧式空间的特性，因此如何在网络空间中构建精准时空体系将是面临的挑战性问题。基于北斗卫星导航系统的高精度时空信息感知技术为解决这一挑战提供了可行途径，但是如何利用部分、稀疏节点的高精度时空信息建立整网高精度时空基准是一个难点。而网络信息时空戳的制订、产生、撤销、跟踪、更新、维护，以及衍生出的测量、路由、信息表示等，则是另一个难点。精准时空体系下网络群体行为运动规律的认知融入精准时空信息的网络空间，包含移动互联网、信息-物理融合系统以及卫星导航系统等，是一个更加复杂的多元异构网络系统，具有典型的多重复杂性特征，如有限或不确定信息、非线性、时滞、网络博弈等。上述多重复杂性导致了精准时空体系下网络群体行为具有不同的运动规律及其干预调控机理。精准时空体系下网络群体行为的认知为实现资源的精准投放、精细管理以及按需服务提供强有力的理论支撑。因此，对精准时空体系下网络群体行为的运动规律认知也是本项目面临的基础性挑战。建立网络通信与计算协作模型，实现高效协同网络通信与计算资源均具有典型时空分布特性，在精准时空体系中上述时空特性会有怎样的作用以及具有时空属性的资源之间又如何相互作用，这些问题直接关系到网络通信与计算的协同效能。另外，不同应用中网络通信与计算资源的形态与运行机制不同，需求特点也不同，从而协同的内在机理也不同。 因此研究网络通信与计算的协作模型及其高效协同机制，实现质量保障的按需服务是本项目面临的重大挑战。

如前面所述，云计算的飞速发展为大数据的产生奠定了基础，大数据时代，对于数据隐私的保护显得尤为迫切，我们的系统原型主要基于北斗卫星导航以及北斗地基增强系统，构建高精度时空信息网络大数据环境，目标是对融合时空信息的数据与隐私保护理论进行分析验证。利用 SDN 构建网络系统，利用北斗和 GNSS 提供定位和授时基础，通过对数据打“时空戳”标签，可以在网络坐标系统内追踪和定位数据，同时实现纳秒（ns）级全网时间同步。

3.1.1 功能需求

网络虚拟化构建过程需要主要到很多功能方面的需求，首先要做的是虚拟网络在逻辑上的隔离性和连通性。隔离性是指虽然可能在一个物理网内传输，但是不同的虚拟网络流量是隔离的，而且流量无法访问其他的虚拟主机。虚拟网的连通性指的是流量在同一个虚拟网络内具有任意可达性，即可以访问内部虚拟网络的任意主机。网络内的主机都能够通过北斗卫星导航以及北斗地基增强系统实现全网同步授时。

3.1.2 原型环境要求

网络空间中精准时空体系的表征与建模，基于北斗卫星导航系统及其高精度时空信息感知技术，在网络空间中建立精准时空体系。针对融入精准时空信息的网络空间多样异构与动态时变的特点，建立适合移动环境下的网络坐标系统，实现准确高效的网络测量与计算，提供信息空间的时空统一建模方法。具体内容如下：

基于北斗卫星导航系统的高精度时间同步技术，研发高精度的定位时间同步终端及网络集成技术，北斗支持下单机时间同步精度优于 20ns，多机同步精度优于 5ns。研究基于北斗地基增强系统的整网高精度时间同步理论与方法，达到整网关键节点 1ns 的时间同步精度。

研究网络空间中时空基准完好性实时监测与精度评估技术，提升时空基准维持与评估能力，确保网络空间中精准时空体系的可靠性。

研究精准时空基准条件下，网络信息的空间定位方法与路由协议。利用卫星导航定位技术，在网络空间中构建高精度时空体系；研究对网络信息进行精准时空标识的“时空戳”方法以及相应路由协议，实现对网络信息的精准定位。探索无线传感网与互联网的异构组网中精准时空信息的作用机制与路由技术。

研究精准时空体系下的网络坐标系统构建理论。研究精准时空体系下网络空间与欧式空间的映射模型，提供准确、高效、鲁棒的网络坐标计算方法；探索移动环境中，基于移动设备实时定位能力的网络坐标获取与网络状态测量技术。

研究精准时空体系下信息空间的时空统一建模方法。根据时空关系的同一性

及相互作用，建立多维、多层次时空模型，分析模型中的映射关系、转换机制以及语义表达与推理，形成信息空间的统一时空模型。研究对时空对象进行表示和推理的方法，实现多模型协同。

如图所示，是原型设计的 SDN 环境模型原理图。从图中我们看到，SDN 技术在里得到了最大的应用，模型图的两侧是系统的计算资源，RYU 控制作为 SDN 控制器主要负责转发逻辑，OpenvSwitch 是交换机的主要构成。通过以上的实验设计，就可以通过再 RYU 控制器上对 APP 进行开发，以达到控制 Open vSwitch 交换机转发逻辑的目的，网络虚拟化功能得以实现，同时能够实现宽带保障功能。GNSS 全网授时和位置定位不断为虚拟网络提供精准的时空信息。

3.2 系统平台总体设计

基于 SDN 的网络环境搭建系统构成和管理架构，依次为租户管理层、控制层和物理网络层。租户管理层主要是对租户的网络实施来进行接收、分析、响应。控制层主要作用是接收物理网络层的通信和相关资源的注册，能够感知网络设备和链路资源的状态变化。具体到功能模块，主要有南向接口、租户管理

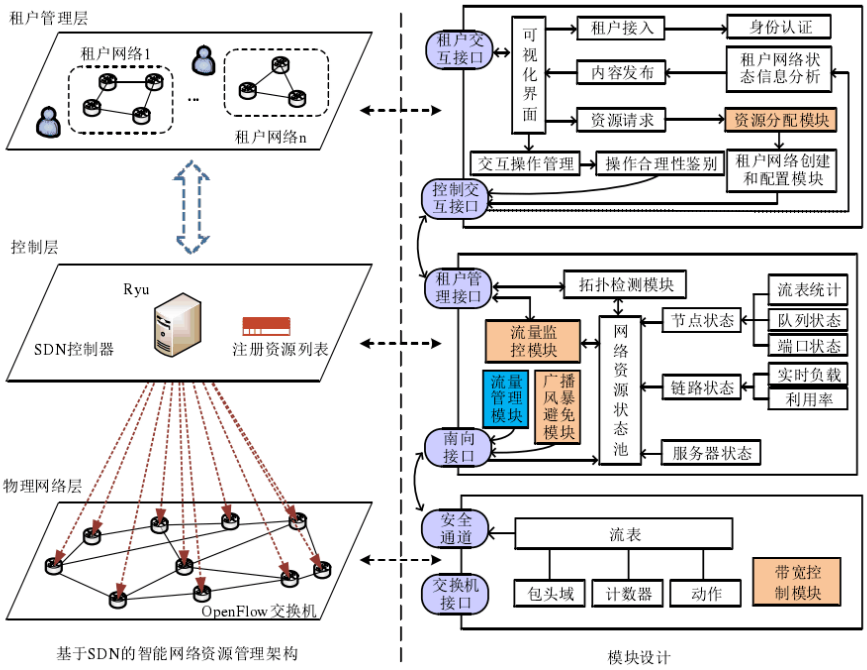


图 3-1 基于北斗授时定位的 SDN 网络系统架构

接口、拓扑检测、网络资源状态池、广播风暴避免，流量管理和监控模块等。基础架构层主要是指物理网络层和 OpenFlow 虚拟交换机，通过流表（Flow Table）、

OpenFlow 协议、OpenFlow 安全传输通道（Secure Channel）这三部分构成，OpenFlow 交换机与远程控制器的通信通过安全通道来实现。

3. 3 时间同步的具体实现

时间同步在互联网领域意义重大，互联网与各行各业联系紧密，随着不断地发展，对时间同步的要求也有了更高的要求。

3.3.1 NTP 协议

网络时间同步协议（Network Time Protocol，NTP），可以实现计算机时间与网络时间的同步。

D. L. Mills 教授于 1985 年提出，由美国德拉瓦大学的，初衷是为了维持计算机网络中不同计算机的时间一致性，通过将网络上的往返延迟封包和对计算机始终偏差进行估算，进而实现网络精准校准时间。

3.3.1.1 层状结构

时间同步的网络，在理论上可以从 0 取决于精度和重要性 16 个或更多阶段 15，其基本上不大于 6 个级别划分。准确性和重要的更高级别的代码低。时间分配 0 是具有高级代码级别的同步子网络。目前，全球卫星定位系统（GPS）被广泛使用，由 GPS 或 UTC 时间码广播。子网中的设备可以承担多种角色。例如，第二级装置是 NTP 协议第一客户端层，对于第三层中的服务器，NTP 协议扩展了对等网络系统，用于在相同的层是通过层与层的膨胀这种分层网络结构为互联网提供及时的服务。。

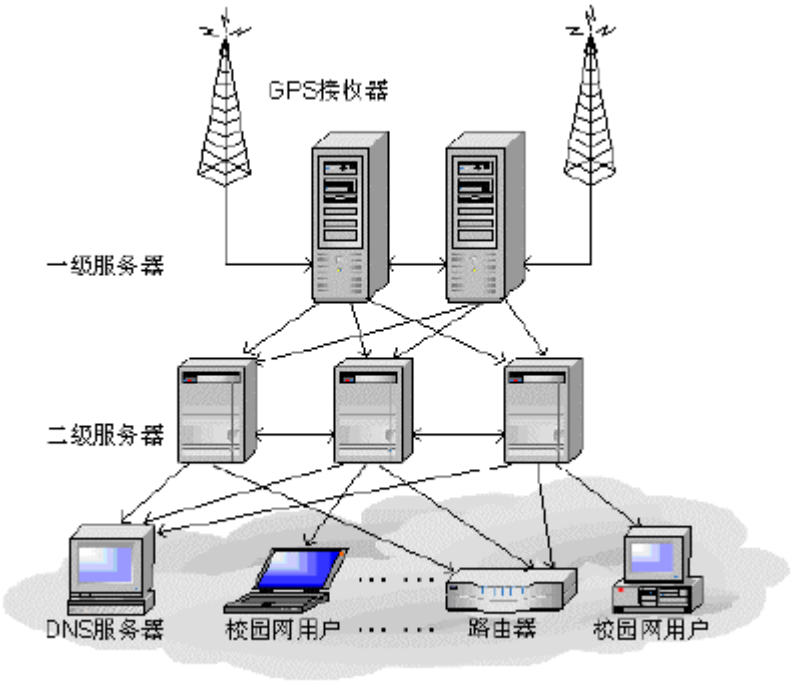


图 3-2 时间同步网络的层状结构

3.3.1.2 工作原理

NTP 算法首先基于服务器和客户端之间的往返分组来确定两个时钟之间的差异和分组传输中的延迟。θ表示客户端和时间服务器之间的时间偏差（offset）；δ表示对时过程中的网络路径延迟（delay）[59][65][66][67]。

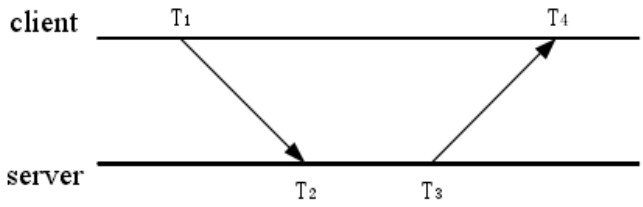


图 3-3 服务器和客户端对时过程

T_1表示客户端时钟记录的发送 NTP 报文的时间和T_4 是客户端时钟接收 NTP 报文的时间[59][62]，客户端接受和发送时间是T_2和 T_3。通过之前的知识，假设服务器时钟经过授时后时钟是准确的，而客户端和服务器之间的时间偏差用θ表示，从客户端发送报文到服务器端的路径延迟是δ_1，从服务器到客户端的路径延迟是δ_2，路径延迟总和是δ。那么可以列出三个方程式：

$$T_2 - T_1 = \theta + \delta_1 \tag{3.1}$$

$$T_4 - T_3 = \delta_2 - \theta \tag{3.2}$$

$$\delta_1 + \delta_2 = \delta \tag{3.3}$$

为了计算方便，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：

$$\delta_1 = \delta_2 = \frac{\delta}{2} \tag{3.4}$$

以上方程变为：

$$T_2 - T_1 = \theta + \frac{\delta}{2} \tag{3.5}$$

$$T_4 - T_3 = \frac{\delta}{2} - \theta \tag{3.6}$$

可解得服务器和客户端时钟的时间偏差：

$$\theta = \frac{(T_2 - T_1) + (T_3 - T_4)}{2} \tag{3.7}$$

客户端与服务器端总的网络路径延迟：

$$\delta = (T_4 - T_1) - (T_3 - T_2) \tag{3.8}$$

3.3.1.3 工作模式

NTP 协议的授时工作模式有三种：

（1）服务器/客户端模式：用户向一台或多台服务器发送服务请求。 基于交换的信息，计算两个位置的时间偏移和网络延迟，并且选择和调整最准确的时间偏移。 本地时钟。

（2）组播/广播模式：适用于高速 LAN。 局域网中的一台或多台服务器在一段时间内向组播地址广播时间戳。 客户端不计算时间偏移量和网络延迟。。

（3）对称模式：两台或两台以上的时间服务器互为主从，通过时间消息互相修改时间并保持整个同步子网的时间一致性。

3.3.1.4 报文格式

NTP 的时间戳用无符号定点数来表述，共有 64 位，前 32 位用来表示整数，后 32 位用来表示小数，以 1900 年 1 月 1 日零点零分零秒为记录起点，依次计算相对于 1900 年，到 2037 年时，此 64 位 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。UDP 传输协议是 NTP 信息在网内传输的协议，在 UDP 的标头里面 Source Port 和 Destination Port 用来记录号码和 123 端口号。

3.3.2 网络时间同步协议比较

表 3.1 GPS、NTP、PTP 协议比较

	GPS	NTP	1588
同步的网络范围	广域	广域	几个子网
使用的通信网络	卫星	互联网	以太网
授时准确度	半微秒	几毫秒	半微秒
工作方式	客户机/服务器	对等层	主时钟/从时钟
需要的资源	适中的计算	适中的网络和计算	小的网络和计算
时间校正	有	有	有
协议安全	不安全	安全	安全
管理	N/A	可配置	自组织
硬件	射频接收和处理	无	有，高准确度
更新间隔	约 1 毫秒	可变，通常在秒级	约 2 毫秒

通过以上 GPS、NTP 和 1588 时间同步协议的比较，可以发现，GPS 和 NTP 同步的网络范围是广域，而 1588 协议的同步范围是几个子网之间。三个协议是用的通信网络各不相同，GPS 依靠卫星来实现时间同步，NTP 可以通过互联网来完成，1588 主要依靠以太网实现时间同步。在授时准确度上，GPS 和 1588 都可以达到半微秒，而 NTP 的授时准确度是几毫秒。工作方式上三者各不相同，GPS 采用客户机和服务器相结合的方式，客户机通过同步服务器的时间来实现时间同步，NTP 协议中依靠对等层来实现同步，1588 协议主要依靠主时钟和从时钟的模式，先对主时钟进行授时，然后从时钟从主时钟处获得时间授时。GPS

和 NTP 需要的计算网络资源比较适中，1588 只适合在较小的网络资源下进行计算。三者都拥有时间校正功能。NTP 和 1588 采用更加安全的协议，而 GPS 的安全协议级别并不高。

3.4 基于北斗系统实现授时和定位

3.4.1 地面接收站设备架设

为了通过北斗系统实现授时，首先要在本地架设信号接收站，才能接收到北斗卫星信号，并进行授时和定位操作，具体操作步骤如下：

1) 首先选择相对空旷的不被遮挡的高层大楼楼顶架设好天线及相关仪器，在地面接收机上面连接固定好天线电缆，并将电缆接入室内；
2) 按图在接收机上连接线路（分别为左边的天线，最下面是网线，最右边为端口及电源线），然后开启接收机。

注：下图为网页连接的情况，串口是另外一种情况。

图 3-4 导航院 GPS 信号接收机

3) 重新设置接收机 IP 地址，观察电脑 IP，更改最后一位(接收机操作步骤：按键切换到 IP 界面，长按 OK 键，DHCP 已禁用，然后继续 ok，然后修改 IP)，重启接收机。
4) 观察接受机屏幕提示，看网络是否连接。
5) 打开网页，输入接收机 IP，账号 wnlbs，密码就是双击网页，显示可以复制的英文，然后复制。

6） 进网页具体操作。

3.4.2 网页连接

1） 首要任务是在服务器上上输入与接收机显示的同一 IP 地址，从而组件同一本地网。

2） 可以根据需要更改存储路径，使数据访问更加方便。

3） 在 I/O 的分类设置模块里，进行具体端口的设置，RTCM 设置和 TC18 设置（格式设置），传输协议采用 TCP 传输协议模式。

4） 基本设施搭建完毕后，完善基础设置，配置完成后，可以查看其他的基本信息

图 3-5 接收机基本配置信息

3.4.3 串口连接

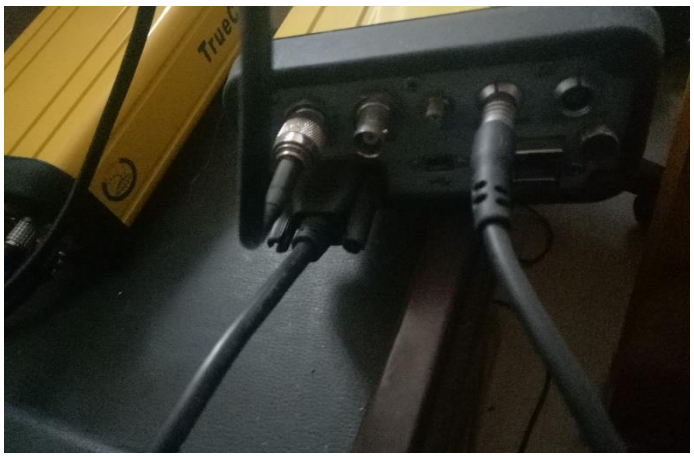


图 3-6 接收机串口连接示意图

1） 具体指令参照上次发的文件(包括：更改串口，收发数据等指令)

2） 查询串口信息，并且可以根据

3.4.4 访问数据方法设置

如果只拿接收机的数据的话，在接收机的页面，设置 NTRIP 将你想要让谁访问的目标 IP 和端口告诉接收机让接收机将数据推到那个端口上即可，将数据推送到导航院、导航中心。

3.4.5 连接并分析

运用 RTKNAVI 进行接收机状态活动情况的观察，从图中可以看到，接收机已经与卫星完成对接，并开始接受授时信息和卫星定位导航等地理位置信息。

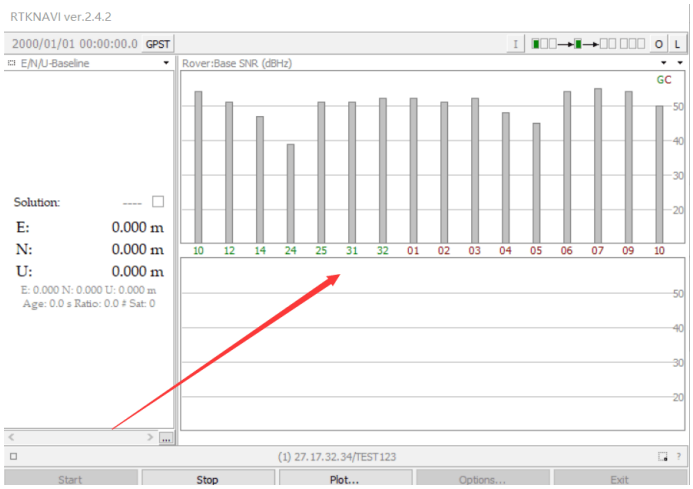


图 3-7 接收机与北斗卫星连通图

当前接收机如果位置发生变化，那么接收到的定位情况随时间变化如图 3-8 所示：

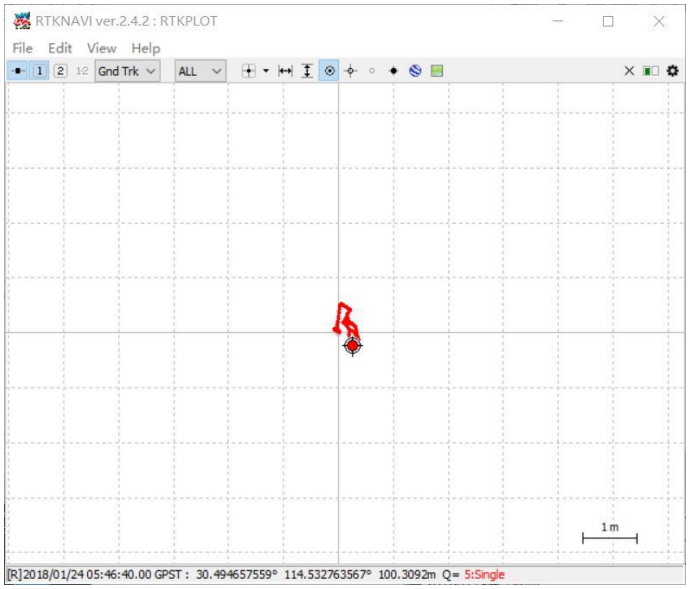


图 3-8 接收机位置变化图

3.5 网络坐标系实验平台搭建

本设计的主要部署环境是基于 Ubuntu16.04.4 LTS，Linux 内核版本是 4.13.0-38-generic 仿真实验平台是 Miniet2.3.0d1，SDN 的控制器选择的版本是 Ryu4.2.3，设计的核心是对网络虚拟化环境中的 Ryu 控制器进行开发，如图 3.9 所示，编程语言上采用 Python 语音对 Ryu 控制器进行开发，应用模块主要包括三个模块，交换机控制模块（通过 SSH 和 OVS 交换机进行通信），网络虚

拟化控制逻辑模块（对数据平面的各种网络资源和物理设备进行维护）和虚拟网络信息库模块（利用 Ryu 提供的 API 实现整个网络进行虚拟化逻辑）。

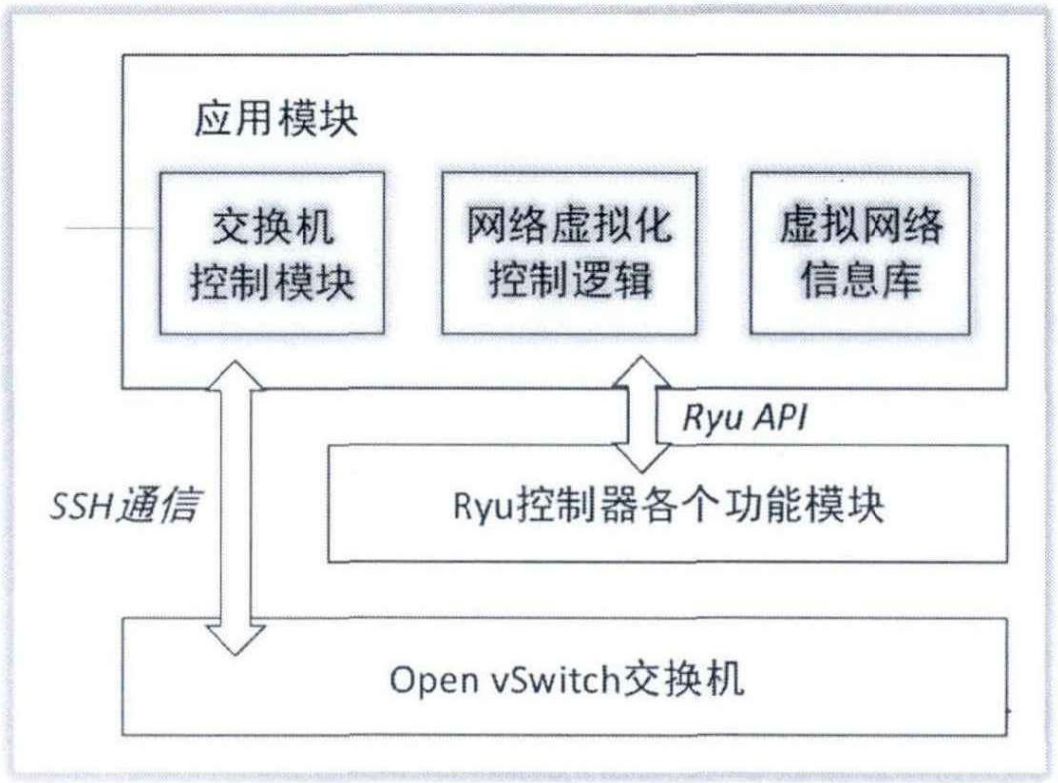


图 3-9 SDN 模块划分

3.5.1 基本环境

1．硬件环境：

处理器：Intel® Xeon™ CPU E3-1230 V5 @ 3.40 GHz 3.40 GHz

安装内存：64.00 GB

2．系统环境：

虚拟机：VMware® Workstation 14 Pro

虚拟机操作系统：Ubuntu 16.04.4 LTS

linux 内核版本：4.13.0-38-generic

3．实验环境

仿真平台：Mininet 2.3.0d1

SDN 控制器：Ryu 4.23

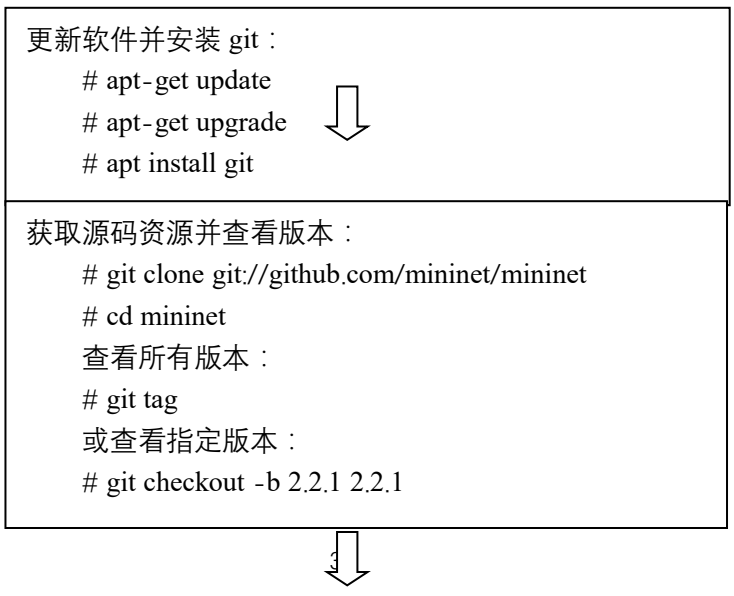
交换机：Open vSwitch 2.5.4

3.5.2 Mininet 部署

Mininet 是一个优秀的网络仿真工具，利用 Mininet 可以在单台机器上创建出逼真的虚拟网络，该虚拟网络运行真实的内核，交换机和应用程序代码，可以迁移到真实的网络中。同时，Mininet 为开展灵活的网络实验提供了强大的技术支持，它提供 CLI 命令和 API 接口，方便用户与网络交互并允许用户自定义网络。Mininet 的主要优点有：1. 可以创建虚拟网络，网络的三大组成部分为：控制器、交换机以及主机。2. 可以为 OpenFlow 应用开发提供有效的实验环境，Mininet 平台中的交换机支持 OpenFlow 协议。3. 可以通过 CLI（命令行接口）进行整个网络范围的测试及排错，。4. 可以运行代码（操作系统、OpenFlow 控制器、OVS、应用程序等代码）。5. 提供了 Python API，供创建和试验网络时使用，可以自定义拓扑结构。6. 兼容 Linux，硬件移植性好。

由此可见，Mininet 是一个研究 OpenFlow 协议和开发测试 SDN 的非常有效的工具。

关于 Mininet 的安装，在 VM 的 Ubuntu 中使用源码编译安装 Mininet 的方式如下：



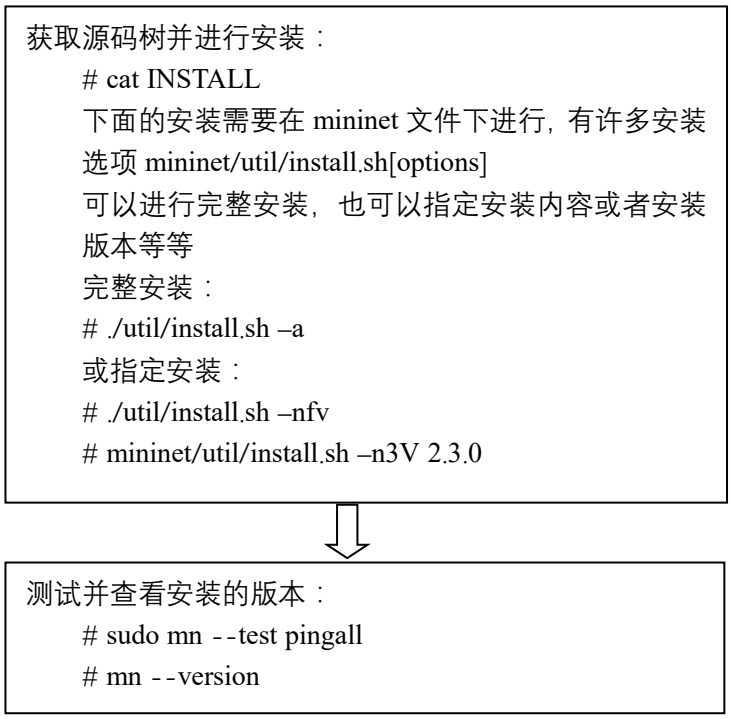


图 3-10 Mininet 源码编译安装步骤

表 3.3 Mininet 重要命令

mininet>node	查看节点
mininet>net	查看链路信息
mininet>help	获取帮助列表
mininet>xterm h1	打开 h1 的终端
mininet>exit	退出 mininet

3.5.3 Ryu 部署

Ryu 是一个支持多种网络管理协议的开源控制器，开发者可以通过修改 Ryu 源码来打造自定义控制器。Ryu 安装过程如下：

首先更新软件：
sudo apt-get upgrade

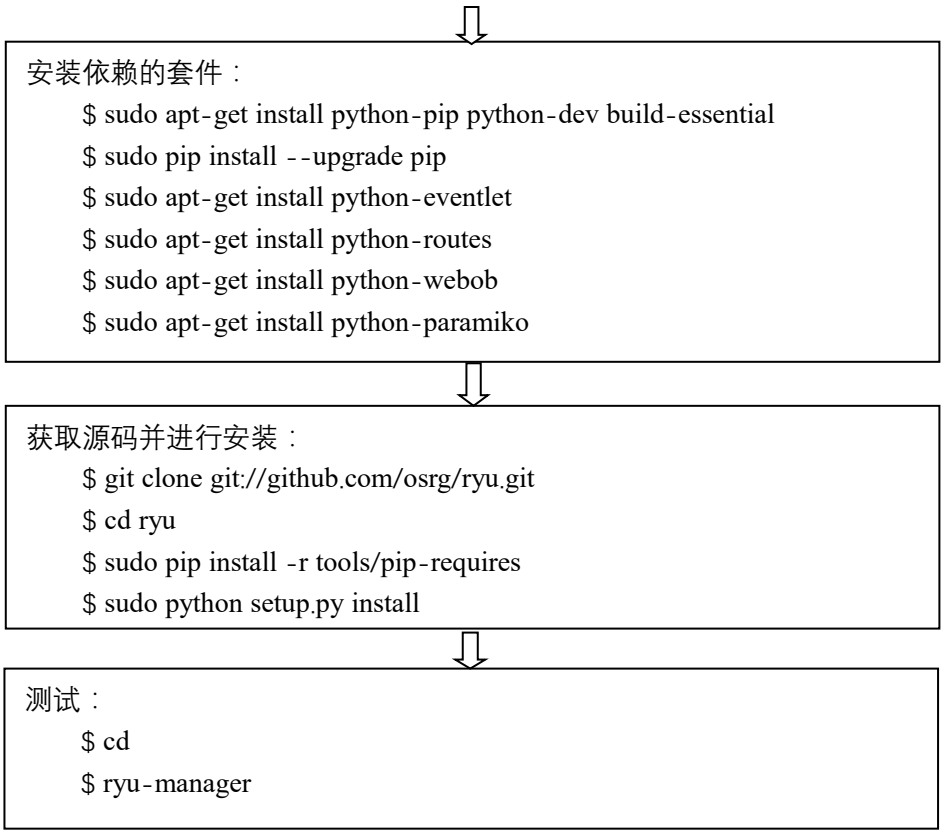


图 3-11 Ryu 源码安装步骤

3.6 模块实现

3.6.1 胖树拓扑实现

Mininet 提供了简单拓扑、链状拓扑、树形拓扑等拓扑类型，但胖树拓扑结构需要另外实现，实现方式可以是使用 mininet 内置的可视化工具 miniedit 搭建，也可以是编写 Python 代码进行定义。

一、使用可视化工具 miniedit 搭建拓扑结构的方法：

1. 执行 miniedit. py 脚本显示 mininet 的可视化界面

2. 拖拽控件搭建拓扑，并为控件设置信息

3. 可生成 Python 拓扑文件将拓扑结构进行保存

如图搭建拓扑修改设置：

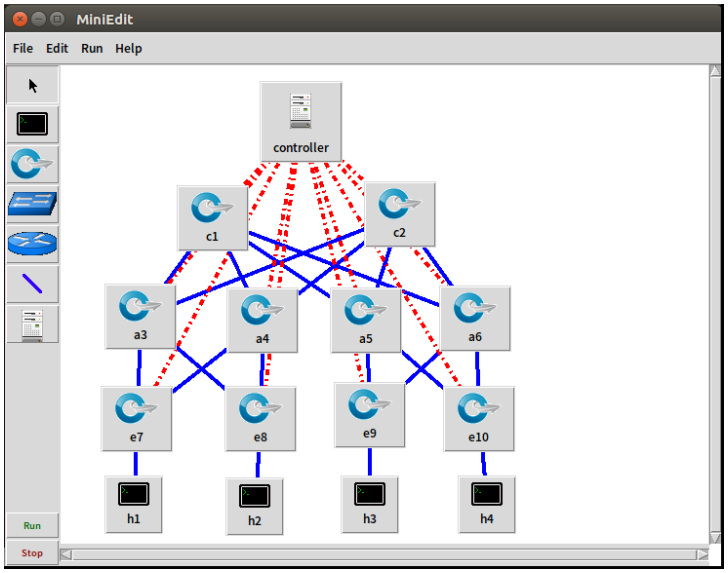
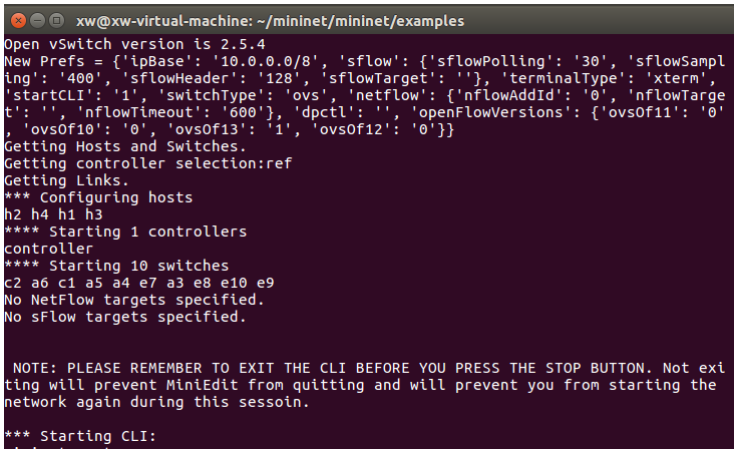


图 3-12 Mininet 网络拓扑图

点击运行，可以在终端看到：



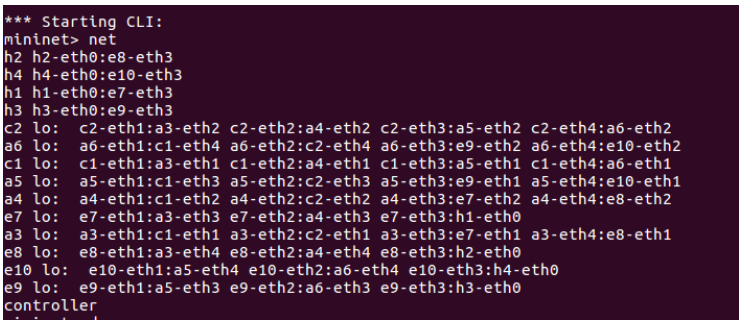
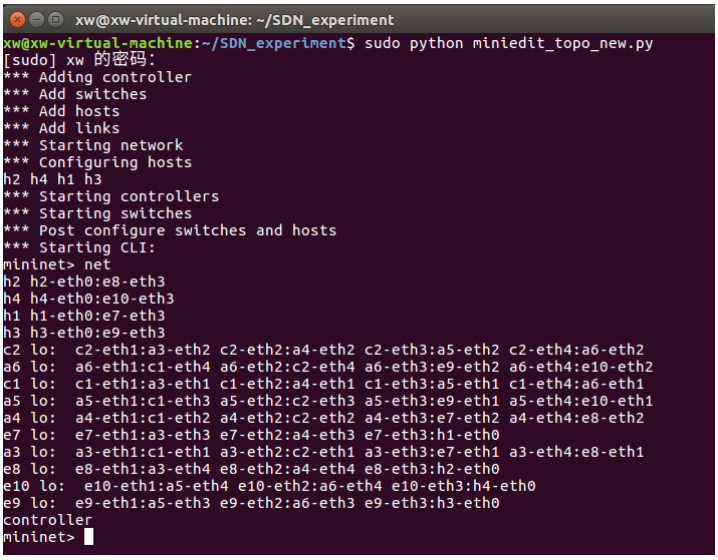


图 3-13 Mininet 下系统运行监测图

保存拓扑为 miniedit_yopo_new. py 文件，运行文件：



二、编写 Python 代码自定义拓扑结构的方法：

1. 编写了 fattree. py 文件，拓扑类都继承自 Topo 基类（位于 topo 模块下），该类提供了添加节点和链路的方法，定义拓扑结构时主要用到：addHost、addSwitch、addLink 等方法，本实验的拓扑构建代码逻辑遵循胖树拓扑结构的规则，构建了一个二元胖树拓扑结构。
2. 使用 sudo mn --custom fattree. py --topo mytopo 命令运行该文件实现拓扑结构，其中--custom 指定自定义拓扑的 python 文件，--topo 指定加载拓扑的名字。

以上过程实现了自定义拓扑脚本，另外，还可以自定义拓扑程序，主要方法是，在 Python 文件（假设为 create_topo. py）中创建拓扑并创建 net 对象，通过 net. start()启动 net 对象运行拓扑，对于自定义拓扑程序，使用 sudo python create_topo. py 命令即可运行拓扑。

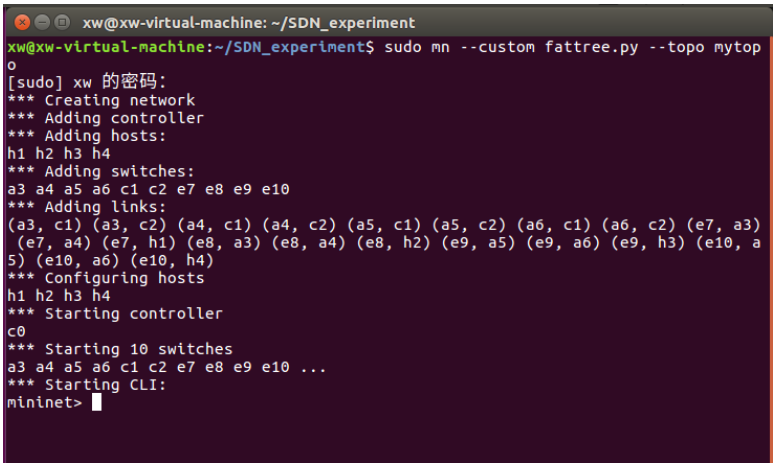


图 3-14 自定义拓扑脚本

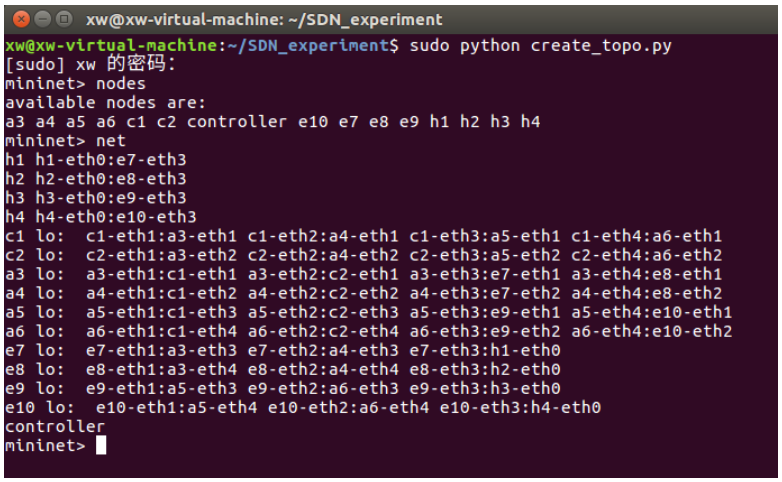


图 3-15 自定义拓扑程序

3.6.2 Ryu 拓扑获取改进

在 Ryu 中，/ryu/ryu/topology 目录下的 switches. py 文件是进行拓扑发现的关键模块，为了实现改进，需要对该模块进行修改。

修改思路为：

1. 修改控制器发送封装了 LLDP 数据包的 packet-out 消息的逻辑，将向每个端口发送的机制改为向每个交换机发送。另外，LLDP 数据包中，Chassis ID TLV 为交换机标志符，Port ID TLV 为端口号，由于在改进方案中不再通过 Port ID TLV 来标识端口，因此在拓扑发现中无需使用 Port ID TLV，可将其设置为 0

2. 控制器向交换机下发规则，当交换机收到的 LLDP 数据包来自控制器时，要求交换机先将 LLDP 数据包的源 MAC 地址设置为发送端口的 MAC 地址，再将数据包从端口发送出去，且交换机的所有端口都要发送相应的 LLDP 数据包。
3. 修改控制器处理 packet-in 消息的机制，让控制器不再通过 Port ID TLV 直接获得端口号，而是通过 LLDP 数据包的源 MAC 地址间接映射端口号。

改进完成后进行验证，确定控制器能够获得全局拓扑，使用 wireshark 抓取 OpenFlow 消息和 LLDP 数据包发现，从交换机发出的 LLDP 数据包的源 MAC 地址为发送端口的 MAC 地址，且在相同的拓扑结构下，改进后的网络中，packet-out 消息的数量减少。

3.6.3 添加标签与下发流表

针对第一种以数据包为主体的思路，本实验构建了一个小型网络拓扑，编写 add_tag_action 函数，该函数定义了给数据包增添标签的动作，由于本实验以链路号作为标签，并将标签保存在 vlan 中，因此该动作实际上就是负责对 vlan 进行操作，使得 vlan 的值为链路号的有序组合。该函数修改 vlan 的逻辑为，首先判断链路号是否需要作为标签存入 vlan 中，当该链路是唯一可达路径上的一部分时，该链路号不需要作为标签保存，然后根据前面是否已经存入了标签来定义保存该标签的不同动作。

在 SDN 架构中，下发流表是控制器控制交换机的手段。首先需要注意的一点是，应该在流表中安装一条默认的流表项，这条流表项实际上也是由控制器下发的，被称为 Table-Miss，它的目的是让交换机给控制器发送 Packet-In 消息。

交换机按照流表来进行转发，交换机收到数据包时，会查找流表来让数据包跟流表项进行匹配，如果匹配不到特定的流表项则会匹配为 Table-Miss，Table-Miss 命令交换机将信息上送给控制器，让控制器来制定对数据包的转发决策，控制器按照前面的功能模块制定决策，将决策封装到流表项再下发至交换机，交换机再按照控制器下发的内容处理数据包。

3.6.4 回溯路径

编写 Ryu 组件 getpath. py，功能是提取数据包的标签，回溯路径并记录当前时间。该组件的目的不是定义数据包转发规则，而是读取数据包内容，因此_packet_in_handler 函数中主要需要实现的是，按照保存标签的逻辑提取出标签并组成队列，对队列中相邻的元素进行树的广度优先遍历来找出两个元素之间的唯一路径，最后将这些路径连接起来组成一条完整的路径。

由于需要寻找两个元素之间的唯一路径，因此必须调用拓扑发现模块来对整网拓扑进行了解，本实验选择将获取的拓扑结构保存在 json 文件中，根据 json 文件来进行树的遍历操作。

3.6.5 Mininet 连接多控制器

本实验自定义了一个二元胖树网络拓扑程序，Python 文件为 create_fattree. py，在该文件中，使用 net. addController 函数（该函数有两个参数，第一个参数为控制器的名称，第二个参数为控制器的监听端口）添加了控制该拓扑的两个控制器 con0（监听的端口号为 6633）和 con1（监听的端口号为 6634），让 con0 与拓扑中的所有交换机相连，con1 则只与边缘交换机相连。

其中，控制器 con0 的任务是对所有交换机进行控制，实现数据包转发和给数据包添加路径标签的功能，运行的模块包括拓扑发现、负载均衡、添加标签以及下发流表等，本实验编写 Ryu 组件 addtag. py 实现上述目的。控制器 con1 的任务则是对边缘交换机进行监听，实现根据数据包携带的标签回溯路径并记录当前时间的功能，运行的模块包括拓扑发现和回溯路径模块，本实验编写 Ryu 组件 getpath. py 实现上述目的。

运行该网络的操作为：

打开终端 1，运行 Ryu 控制器 con0：

```
sudo ryu-manager addtag.py --ofp-tcp-listen-port=6633
```

打开终端 2，运行 Ryu 控制器 con1：

```
sudo ryu-manager getpath.py --ofp-tcp-listen-port=6634
```

打开终端 3，运行 mininet 构建网络拓扑并连接控制器：

```
sudo python create_fattree.py
```

终端 3 运行后命令后将进入 mininet 终端，在 mininet 的终端可进行数据包的 ping 操作，如果在实现组件的时候有将结果展示到终端，那么可以在终端看到相应的结果。

实验结果如下：

1. 在终端 1 运行控制器 con0

```
xw@xw-virtual-machine:~/SDN_experiment/multiple _Controllers$ sudo ryu-manager a
ddtag.py --ofp-tcp-listen-port=6633
[sudo] xw 的密码:
loading app addtag.py
loading app ryu.controller.ofp_handler
instantiating app None of Stp
creating context stplib
instantiating app ryu.controller.ofp_handler of OFPHandler
instantiating app addtag.py of AddTag
```

2. 在终端 2 运行控制器 con1

```
xw@xw-virtual-machine:~/SDN_experiment/multiple _Controllers$ sudo ryu-manager g
etpath.py --ofp-tcp-listen-port=6634
[sudo] xw 的密码:
loading app getpath.py
loading app ryu.controller.ofp_handler
instantiating app ryu.controller.ofp_handler of OFPHandler
instantiating app getpath.py of GetPath
```

3. 在终端 3 构建拓扑结构

```
xw@xw-virtual-machine:~/SDN_experiment/multiple _Controllers/mininet_topo$ sudo
python create_fattree.py
[sudo] xw 的密码:
mininet>
```

4. 在 mininet 终端进行 ping 操作

```
xw@xw-virtual-machine:~/SDN_experiment/multiple _Controllers/mininet_topo$ sudo
python create_fattree.py
[sudo] xw 的密码:
mininet> h1 ping -c 1 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=58.9 ms

--- 10.0.0.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 58.953/58.953/58.953/0.000 ms
mininet> h1 ping -c 1 h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=98.5 ms

--- 10.0.0.3 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 98.505/98.505/98.505/0.000 ms
mininet> h1 ping -c 1 h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=93.7 ms

--- 10.0.0.4 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 93.763/93.763/93.763/0.000 ms
mininet> h2 ping -c 1 h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=85.3 ms
```

5. 在控制器 con1 在终端 2 显示回溯路径的结果

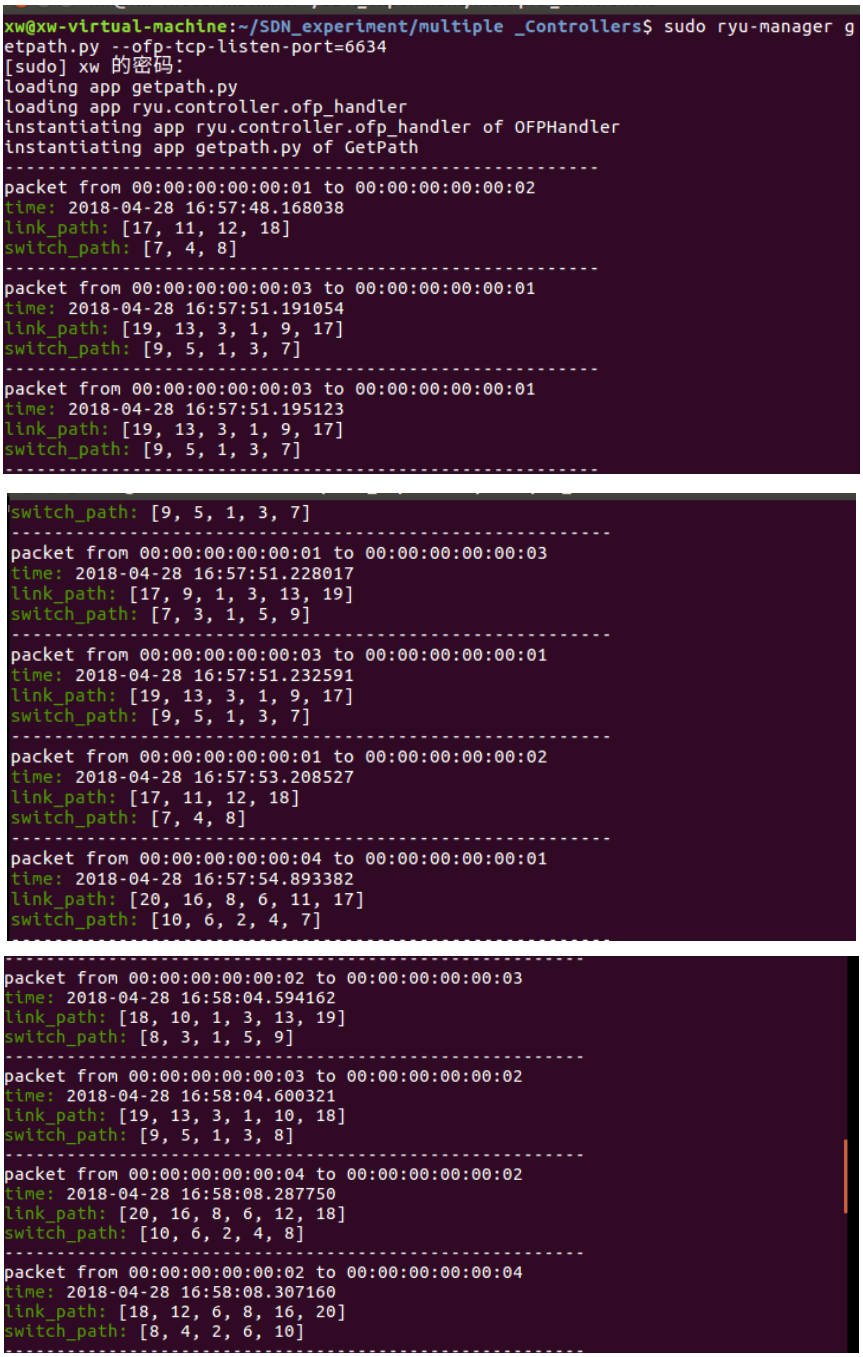


图 3-16 回溯路径模块实现

3.7 本章小结

本章主要系统的需求进行了进一步的分析，并且明确了技术研究路线和模型搭建，对平台进行了总体的设计，给出了应用框架设计的基本情况。同时介绍了搭建实验的环境，包括硬件环境和系统环境。

第四章 时空网络坐标系性能分析与优化

4.1 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[44]算法以来，现在已有基于中心式的如GNP[44]、PIC[52]和基于非中心式的如NPS[53]、Vivaldi[45]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，建立N维坐标系并将网络节点放入其中，通过计算节点坐标距离来作为网络时延的预测值。

4.1.1 随机延迟现象

在网络系统中，时延可以分为两种，一种是单向时延（One-way Delay，OWD），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。

4.1.2 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]。

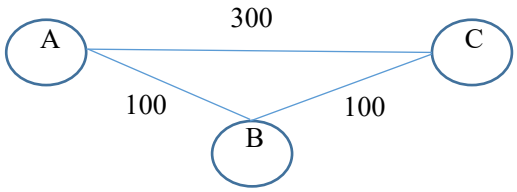


图 4-1 三角不等式违例现象

如图4-1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。

4.2 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。

4.2.1 随机延迟污染现象数据分析

本文的时延数据是通过仿真试验获得的数据集，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 4-2 所示：

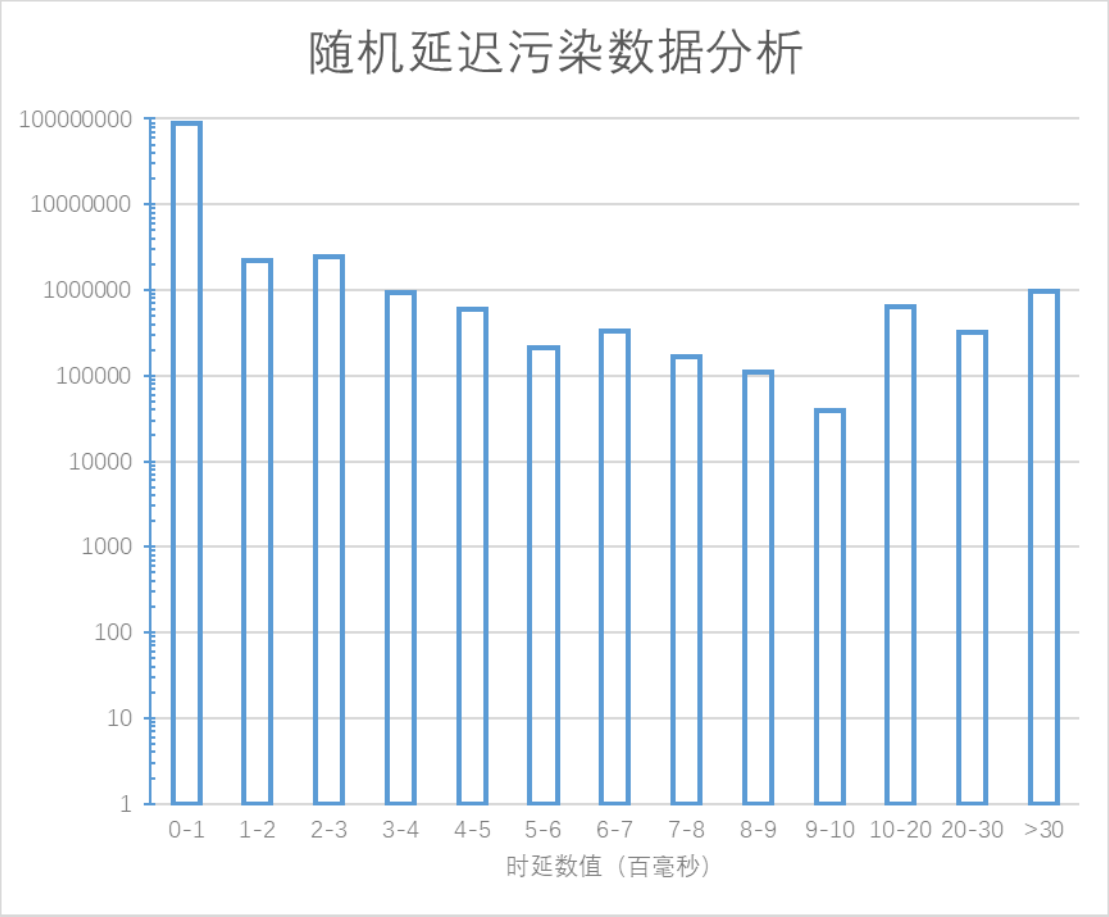


图 4-2 随机延迟污染数据分析

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。

4.3 抑制随机延迟污染的方法

4.3.1 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。经过仿真实验的验证发现，当 $P = 0.25$ 和 $W = 4$ 的时候，MP−Filter 对波动有较好的抑制作用并且能够将原始实测时延的统计特征保留下来，其系统的各方面的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]。

4.3.2 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 $MeanRTT_{i,j}$:

$$MeanRTT_{i,j} = (1-\alpha)\cdot MeanRTT_{i,j} + \alpha \cdot \ RTT_{i,j} \qquad (4.1)$$

$MeanRTT_{i,j}$ 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 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 $DevRTT_{i,j}$，用于估计 $RTT_{i,j}$ 与 $MeanRTT_{i,j}$ 的偏离程度：

$$DevRTT_{i,j} = (1-\beta)\cdot DevRTT_{i,j} + \beta \cdot \left|RTT_{i,j} - MeanRTT_{i,j}\right| \qquad (4.2)$$

$DevRTT_{i,j}$ 是 $RTT_{i,j}$ 与 $MeanRTT_{i,j}$ 的差值的 EWMA，这里 β 的推荐值为 1/4 即 β =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 $MeanRTT_{i,j} + 4 \cdot DevRTT_{i,j}$。首先进行随机延迟污染的检测，如果 $RTT_{i,j} > MeanRTT_{i,j} + 4 \cdot DevRTT_{i,j}$ 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 $DevRTT_{i,j}$ 与 $MeanRTT_{i,j}$ 的值，然后让 $RTT_{i,j}$ = $MeanRTT_{i,j}$ 作为输出。

4.4 网络坐标中三角不等式现象级抑制方法

4.4.1 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。为了达到了抑制 TIV 对坐标系统的影响的目的，可以通过根据该系数的值抑制违例边对坐标的更新来实现

具体步骤是：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，则获取节点 i，节点 j，节点 k 间的时延，并计算违例系数：

$$\lambda = \frac{RTT_{i,j}}{RTT_{i,k}+RTT_{k,j}} \tag{4.3}$$

如果 λ>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 $\lambda \leqslant 1$ 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。

4.4.2 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 算法的求解过程转化为非线性方程组的，然后利用迭代求解算法来对其进行求解，由于方程组具有一定的矛盾性，借此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。

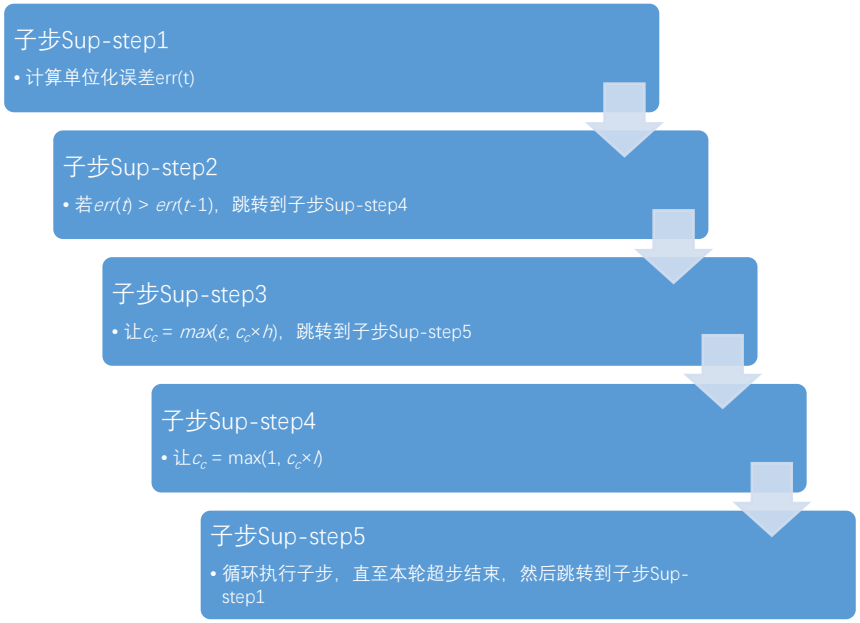
其感知方法[6]为：对于某个时间切片，在节点的某个时间切片中，计算邻居节点与该节点的坐标距离与测量距离误差的均值为单位化误差，如节点 i 在时刻 t 的坐标为 $x_i(t)$，$|Neightbor(i)|$ 为节点 i 的邻居节点个数，则单位化误差 err(t) 为下列公式(4.2)所示：

$$err_i(t) = \frac{\sum_{j \in Neightbor(i)} \sqrt{(RTT_{i,j} - \|X_i(t) - X_j(t)\|)^2}}{|Neightbor(i)|} \tag{4.4}$$

节点的单位化误差会随着算法的不断运行逐渐下降，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变大，即 $err(t) > err(t-1)$ 时，则视为发生了抖动；反正，则认为算法收敛过程仍在继续进行中，我们可以通过增加本轮迭代的迭代步长来加快收敛的速度。

在执行前，需要将最小步长迭代因子 ε 以及最大迭代因子定义为 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，c_c 为 Vivaldi 算法中步长调节因子。

对于一次具体超步步骤[6]是：



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。

4.4.3 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 W_S(Window Start)和一个当前窗口 W_C(Window Current)，分别用于保存节点坐标的历史记录，区别在于 W_S 只保存最开始的 n_1 个节点坐标记录，当 W_S 已经储存了 n_1 个节点坐标记录后将不再更新，而 W_C 是不断更新的，它储存着最新的 n_2 个节点坐标记录。将 W_S 和 W_C 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 γ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。

$$e(W_S, W_C) = \frac{n_1 n_2}{n_1+n_2}\left(\begin{array}{c}\frac{2}{n_1 n_2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\|a_i - b_j\| - \frac{1}{{n_1}^2}\sum_{i=1}^{n_1}\sum_{j=1}^{n_1}\|a_i - a_j\| \\ -\frac{1}{{n_2}^2}\sum_{i=1}^{n_2}\sum_{j=1}^{n_2}\|b_i - b_j\|\end{array}\right) \quad (4.5)$$

其中 a，b 分别为 W_S 和 W_C 中储存的节点坐标，只有当 $e(W_S, W_C)$ 的值超过规定数值 γ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 W_C 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 γ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]。

4.4.4 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。

4.5 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。

4.5.1 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 $RTT_{i,j}$ 时，首先进行随机延迟污染的检测，如果 $RTT_{i,j} > MeanRTT_{i,j} + 4 \cdot DevRTT_{i,j}$ 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 $DevRTT_{i,j}$ 与 $MeanRTT_{i,j}$ 的值，然后让 $RTT_{i,j} = MeanRTT_{i,j}$ 进入下部分，从而减少不可靠的延迟 $RTT_{i,j}$ 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 $RTT_{i,j} \leqslant MeanRTT_{i,j} + 4 \cdot DevRTT_{i,j}$ 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。对此，本文提出了一种坐标抖动感知方法。

4.5.2 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 $RTT_{i,j}$ 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($N=|Neightbor(i)|$，即节点 i 的邻居节点数)，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：

$$err_t = \frac{\sum_{j \in Neightbor(i)} \sqrt{(MeanRTT_{i,j} - \|X_i(t) - X_j(t)\|)^2}}{|Neightbor(i)|} \tag{4.6}$$

当 $err_t > err_{t-1}$ 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 $RTT_{i,j} \leqslant MeanRTT_{i,j} + 4 \cdot DevRTT_{i,j}$ 时才执行，否则需要重新计数。

4.5.3 抑制算法

4.5.3.1 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：

$$\delta = c_c \omega d \tag{4.7}$$

$$d = \frac{1}{\left(1+ln\left(1+\frac{RTT}{100}\right)\right)} \tag{4.8}$$

$ln(x)$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$(0, 1)$。

4.5.3.2 抑制算法的退出

值得注意的是，抑制算法只有在出现 $err_t > err_{t-1}$ 之后，并且 $RTT_{i,j} \leqslant MeanRTT_{i,j} + 4 \cdot DevRTT_{i,j}$ 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。

4. 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 $d = 1$，计算器 $n = 0$，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：

$$DevRTT_{i,j} = (1-\beta) \cdot DevRTT_{i,j} + \beta \cdot \left|RTT_{i,j} - MeanRTT_{i,j}\right| \tag{4.9}$$

$$MeanRTT_{i,j} = (1-\alpha) \cdot MeanRTT_{i,j} + \alpha \cdot \ RTT_{i,j} \tag{4.10}$$

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 err_t：

$$err_t = \frac{\sum_{j \in Neightbor(i)} \sqrt{(MeanRTT_{i,j} - \|X_i(t) - X_j(t)\|)^2}}{|Neightbor(i)|} \tag{4.11}$$

若 $err_t > err_{t-1}$ 则让 $d = 0$，否则让 $err_{t-1} = err_t$。

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 e_s。

4）用公式(2.7)更新自身误差估计 e_i。

5）用公式(5.2)计算 δ。

6）用公式(2.9)更新本地坐标 x_i。

4.5.5 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。

4.5.5.1 准确性分析

为了度量算法的准确性，利用下面公式来定义相对误差 RE：

$$RE_{i,j} = \frac{|RTT_{i,j} - \|X_i - X_j\||}{MIN(RTT_{i,j}, \|X_i - X_j\|)} \tag{4.12}$$

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。相对误差 RE 可以

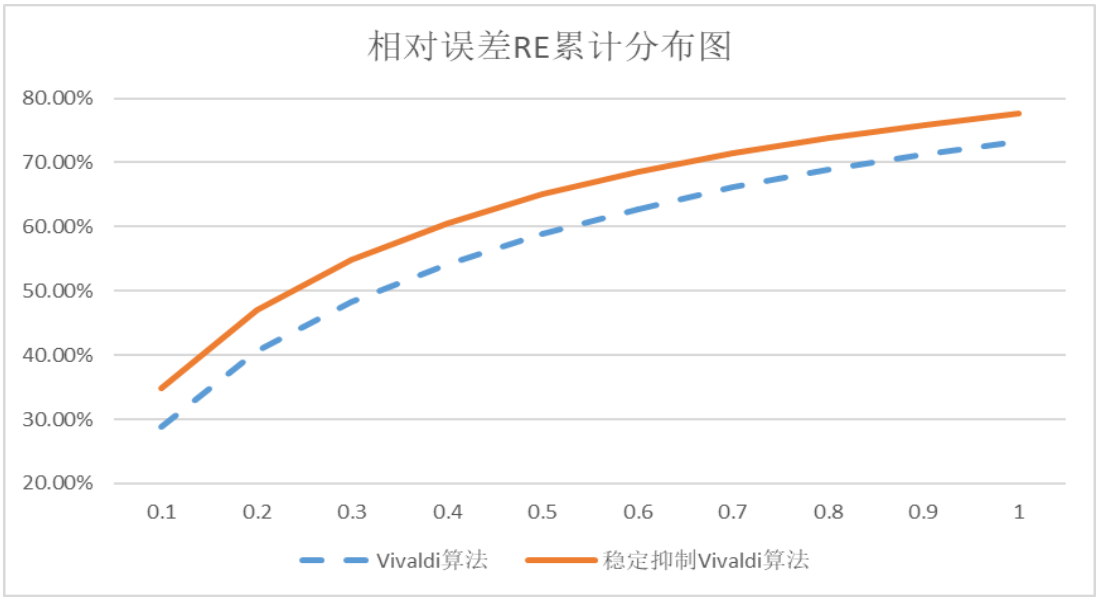


图 4-3 相对误差RE累计分布图

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 累计分布图如图 5-3：

从图 5-3 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：

$$e_{i,j} = \sqrt{\left(RTT_{i,j} - \|x_i - x_j\|\right)^2} \tag{4.13}$$

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：

$$e = \frac{\Sigma\sqrt{\left(RTT_{i,j} - \|x_i - x_j\|\right)^2}}{n} \tag{4.14}$$

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 4-4 所示：

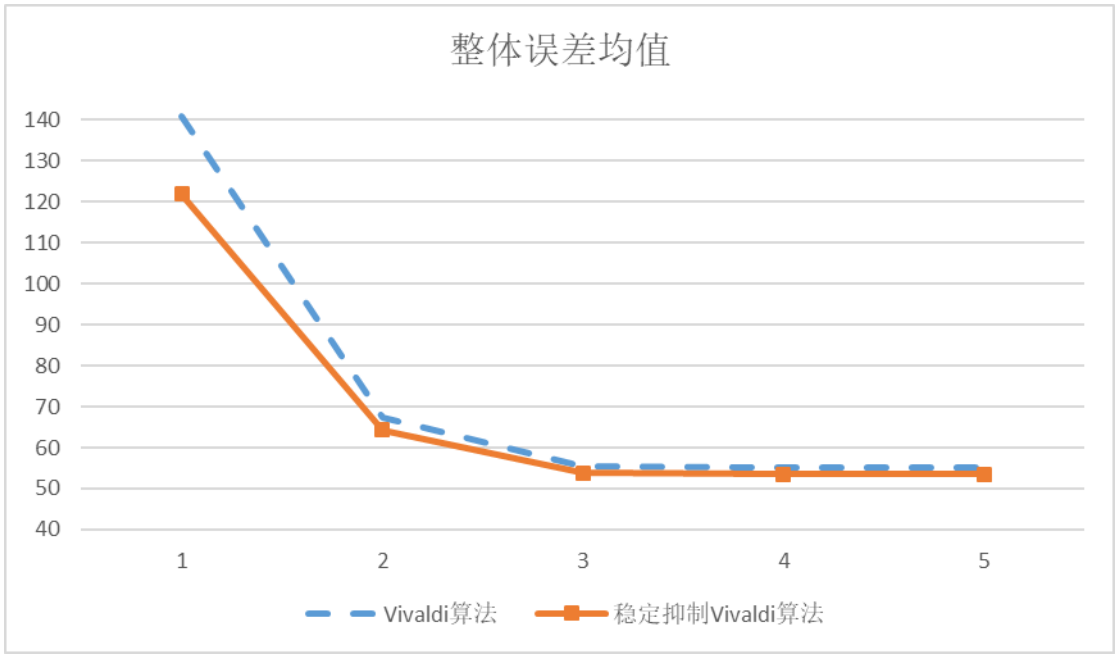


图 4-4 整体误差均值

从图 4-4 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。

4.5.5.2 抑制抖动能力分析

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. 15)：

$$\begin{cases} u_i = \dfrac{\sum x_i}{n} \\ s_i = \dfrac{\sum(\|x_i - u_i\|)^2}{m} \end{cases} \tag{4. 15}$$

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：

$$S = \frac{\sum_{i=1}^{n} s_i}{n} \tag{4. 16}$$

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 5-5 所示：

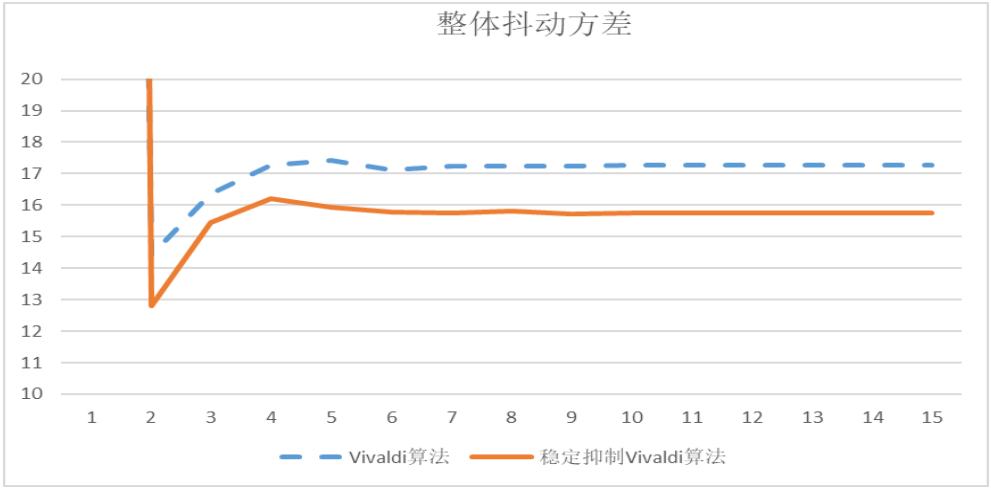


图 4-5 整体抖动方差

图 4-5 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%。

4.6 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%。

第五章 总结与展望

5.1 论文总结

首先，在北斗高精度时空信息感知理论与方法的指导下，探索北斗高精度定位时间同步技术，建立整网高精度时间同步数学模型，突破北斗高精度定位时间同步终端关键技术；研究与网络融合的终端集成技术，构建基于北斗卫星导航系统的网络空间高精度时空体系。研究系统监测评估理论，突破系统完备性监测技术，实现对时空信息精度的实时监测和评估，确保所构建精准时空体系的可靠性。

然后，基于高精度时空体系，从网络信息与网络空间两个层次研究精准时空信息的融合理论。一方面，研究网络信息单元中精准时空标识的嵌入，修改现有网络协议，改时间戳为“时空戳”，实现对信息的空间精准定位。另一方面，从几何与逻辑两个角度研究网络信息空间的表征与建模，实现基于精准时空信息的网络坐标系统以及信息空间的时空统一建模。设计分布式、地域邻近聚类的分层式网络坐标计算系统，采用矩阵分解模型的计算方法，降低因 TIV 因素带来的预测误差，探索利用移动设备实时定位能力提供精确度补偿的网络坐标计算方法。

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。

5.2 存在的问题和进一步工作

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。

参考文献

[1] D. Raychaudhuri, N. B. Mandayam, ―Frontiers of Wireless and Mobile Communications,‖ *Proceedings of the IEEE*, vol. 100, no. 4, pp. 824-840, 2012.

[2] N. Radia, Y. Zhang, M. Tatipamula, V. K. Madisetti, ―Next-Generation Applications on Cellular Networks: Trends, Challenges, and Solutions,‖ *Proceedings of the IEEE,* vol. 100, no. 4, pp. 841-854, 2012.

[3] C. X. Wang, F. Haider, X. Gao, X. H. You, et al., ―Cellular architecture and key technologies for 5G wireless communication networks,‖ *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 122-130, 2014.

[4] K. D. Kim, P. R. Kumar, ―Cyber-Physical Systems: A Perspective at the Centennial,‖ *Proceedings of the IEEE,* vol. 100, no. SI, pp. 1287-1308, 2012.

[5] N. M. Freris, H. Kowshik, P. R. Kumar, ―Fundamentals of Large Sensor Networks: Connectivity, Capacity, Clocks, and Computation,‖ *Proceedings of the IEEE,* vol. 98, no. 11, pp. 1828–1846, 2010.

[6] M. J. Freedman, K. Lakshminarayanan, D. Mazières, ―OASIS: Anycast for any service,‖ *In Proc. 3rd USENIX/ACM Symp. Networked Systems Design and Implementation (NSDI'06)*, San Jose, CA, USA, May, 2006, pp. 129-142.

[7] M. Szymaniak, D. Presotto, G. Pierre, V. S. Maarten, ―Practical large-scale latency estimation,‖ *Computer Networks*, vol. 52, no. 7, pp. 1343–1364, 2008.

[8] Y. Chen, Y. Xiong, X. Shi, J. Zhu, et al., ―Pharos: Accurate and decentralised network coordinate system,‖ *IET Commun.*, vol. 3, no. 4, pp. 539-548, 2009

[9] 北斗系统公开服务性能规范（1.0版）, http://www.beidou.gov.cn/

[10] Y. Yang, J. Li, A. Wang, J. Xu, et al., ―Preliminary assessment of the navigation and positioning performance of BeiDou regional navigation satellite system,‖ *Sci. China Ser. D*, vol. 57, no. 1, pp. 144-152, 2014.

[11] H. Liu, S. Guo, J. Liu, Z. Tian, et al., ―Present status analysis on the construction and application of CORS in China,‖ In *2012 Proc. China Satellite Navigation Conf. (CSNC)*, Springer-Verlag, Berlin: Heidelberg, pp. 401-409, 2012.

[12] H. He, J. Li, Y. Yang, J. Xu, et al., ―Performance assessment of single- and dual-frequency BeiDou/GPS single-epoch kinematic positioning,‖ *GPS Solutions*, Doi. 10.1007/s10291-013 - 0339-3, 2013.

[13] W. Meng, H. Zhang, P. Huang, J. Wang, et al., ―Design and experiment of onboard laser time transfer in Chinese Beidou navigation satellites,‖ Adv. *Space Res.*, vol. 51, no. 6, pp. 951-958,2013.

[14] A.Vespignani,―Predicting the behavior of techo-social systems,‖ *Science*, vol. 325, no. 5939, pp. 425-428, 2009.

[15] D. J. Watts, S. H. Strogatz, ―Collective dynamics of _small-world' networks,‖ *Nature*, vol.

393, no. 6684, pp. 440-442, 1998.

[16] A.L. Barabási, R. Albert, ―Emergence of scaling in random networks,‖ Science, vol. 286, no. 5439, pp. 509-512, 1999.

[17] R. Albert, R. Jeong, A. L. Barabasi, ―Error and attack tolerance of complex networks,‖ *Nature*, vol. 406, no. 6794, pp. 378-382, 2000.

[18] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, et al., ―Network motifs: Simple building blocks of complex networks,‖ *Science*, vol. 298, no. 5594, pp. 824-827, 2002.

[19] G. Palla, I. Derenyi, I, Farkas, T. Vicsek, ―Uncovering the overlapping community structure of complex networks in nature and society,‖ *Nature*, vol. 435, no. 7043, pp. 814-818, 2005.

[20] M. F. Mason, M. I. Norton, J. D. Van Horn, D. M. Wegner, et al., ―Wandering minds: The default network and stimulus-independent thought,‖ *Science*, vol. 315, no. 5810, pp. 393-395, 2007.

[21] H. Ohtsuki, C. Hauert, E. Lieberman, M. A. Nowak, ―A simple rule for the evolution of cooperation on graphs and social networks,‖ *Nature*, vol. 441, no. 7092, pp. 502-505, 2006.

[22] S. Eubank, H. Guclu, V. S. A. Kumar, M. V. Marathe, et al., ―Modelling disease outbreaks in realistic urban social networks,‖ *Nature*, vol. 429, no. 6988, pp. 180-184, 2004.

[23] R. Milo, S. Itzkovitz, N. Kashtan, L. Boitani, et al., ―Superfamilies of evolved and designed networks,‖ *Science*, vol. 303, no. 5663, pp. 1538-1542, 2004.

[24] A.S. L. Rodrigues, S. J. Andelman, M. I. Bakarr, et al., ―Effectiveness of the global protected area network in representing species diversity,‖ *Nature*, vol. 428, no. 6983, pp. 640-643, 2004.

[25] A.M. Song, S. Havlin, H. A. Makse, ―Self-similarity of complex networks,‖ Nature, vol. 433, no. 7024, pp. 392-395, 2005.

[26] J. Lü, G. Chen, ―A time-varying complex dynamical network models and its controlled synchronization criteria,‖ *IEEE Trans. Auto. Contr.*, vol. 50, no. 6, pp. 841-846, 2005.

[27] J. Lü, X. Yu, G. Chen, D. Cheng, ―Characterizing the synchronizability of small-world dynamical networks,‖ *IEEE Trans. Circuits Syst. I*, vol. 51, no. 4, pp. 787-796, 2004.

[28] R. O. Grigoriev, M. C. Cross, H. G. Schuster, ―Pinning Control of Spatiotemporal Chaos,‖ *Phys. Rev. Lett.*, vol. 79, no. 15, pp. 2795-2798, 1997.

[29] X. Wang, G. Chen, ―Pinning control of scale-free dynamical networks,‖ *Physica A*, vol. 310, no.3-4, pp. 521-531, 2002.

[30] J. Zhou, J. A. Lu, J. Lü, ―Pinning adaptive synchronization of a general complex dynamical nertwork,‖ *Automatica*, vol. 44, no. 4, pp. 996-1003, 2008.

[31] Y. Y. Liu, J. J. Slotine, A. L. Barabasi, L. Albert, ―Controllability of complex networks,‖ *Nature*, vol. 473, no. 7346, pp. 167-173, 2011.

[32] A.Clauset, C. Moore, M. E. J. Newman, ―Hierarchical structure and the prediction of missing links in networks,‖ *Nature*, vol. 453, no. 7191, pp. 98-101, 2008.

[33] G. Kossinets, D. J. Watts, ―Empirical analysis of an evolving social network,‖ *Science,* vol. 311, no. 5757, pp. 88-90, 2006.

[34] R. Guimera, B. Uzzi, J. Spiro, L. A. N. Amaral, ―Team assembly mechanisms determine collaboration network structure and team performance,‖ *Science,* vol. 308, no. 5722, pp. 697-702, 2005.

[35] Y. Y. Ahn, J. P. Bagrow, S. Lehmann, ―Link communities reveal multiscale complexity in networks,‖ *Nature,* vol. 466, no. 7307, pp. 761-764, 2010.

[36] D. Centola, ―The Spread of Behavior in an Online Social Network Experiment,‖ *Science,* vol. 329, no. 5996, pp. 1194-1197, 2010.

[37] D. Achlioptas, R. M. D'Souza, J. Spencer, ―Explosive Percolation in Random Networks,‖ *Science,* vol. 323, no. 5920, pp. 1453-1555, 2009.

[38] P. J. Mucha, T. Richardson, K. Macon, ―Community Structure in Time-Dependent, Multiscale, and Multiplex Networks,‖ *Science,* vol. 328, no. 5980, pp. 876-878, 2010.

[39] M. Ferdman, A. Adileh, O. Kocberber, S. Volos, et al. ―Clearing the clouds: a study of emerging scale-out workloads on modern hardware,‖ *ACM SIGARCH Comput. Architect. News,* vol. 40, no. 1, pp. 37-48, 2012.

[40]李德毅，林润华，李兵．中国电子学会云计算专家委员会．2013云计算发展报告，科学出版社，2013.

[41] R. Ahlswede, N. Cai, S.-Y. R. Li, R. W. Yeung. ―Network information flow,‖ *IEEE Trans. Inf. Theory,* vol. 46, no. 4, pp. 1204-1216, 2000.

[42] S.-Y. R. Li, R. W. Yeung, N. Cai, ―Linear network coding,‖ *IEEE Trans. Inf. Theory,* vol. 49, no. 2, pp. 371-381, 2003.

[43] 陈阳．网络坐标计算模型与应用研究[D]．清华大学, 2009.

[44] Ng T S E, Zhang H. Predicting Internet network distance with coordinates-based approaches[C]// Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. IEEE Xplore, 2002:170-179 vol.1.

[45] Dabek F, Cox R, Kaashoek F, et al. Vivaldi: a decentralized network coordinate system[J]. ACM SIGCOMM Computer Communication Review, 2004, 34(4):15-26.

[46] 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.

[47] Zheng H, Lua E K, Pias M, et al. Internet Routing Policies and Round-Trip-Times[M]// Passive and Active Network Measurement. Springer Berlin Heidelberg, 2005:236-250.

[48] 王聪，张凤荔，刘梦娟,等. IP网络坐标抖动感知与慢启动抑制[J]．电子科技大学学报, 2012(6):921-926.

[49] Dischinger M, Haeberlen A, Gummadi K P, et al. Characterizing residential broadband networks[C]// ACM SIGCOMM Conference on Internet Measurement 2007, San Diego, California, Usa, October. DBLP, 2007:43-56.

[50] Ledlie J, Pietzuch P, Seltzer M. Stable and Accurate Network Coordinates[C]// IEEE

International Conference on Distributed Computing Systems. IEEE, 2006:74-74.
[51] 周亮. IP网络坐标系统的准确性增强技术研究[D]. 电子科技大学, 2011.
[52] Costa M, Castro M, Rowstron A, et al. PIC: practical Internet coordinates for distance estimation[C]// International Conference on Distributed Computing Systems. IEEE Computer Society, 2004:178-187.
[53] Ng T S E, Zhang H. A network positioning system for the internet[C]// General Track: 2004 Usenix Technical Conference, June 27 - July 2, 2004, Boston Marriott Copley Place, Boston, Ma, Usa. DBLP, 2004:141-154.
[54] Kaafar M A, Mathy L, Turletti T, et al. Real attacks on virtual networks:Vivaldi out of tune[C]// Lsad 06: SIGCOMM Workshop on Large-Scale Attack Defense. 2006:139-146.
[55] Kaafar M A, Mathy L, Turletti T, et al. Virtual networks under attack:disrupting internet coordinate systems[C]// CONEXT Conference. 2006:1-12.
[56] 赵小菡. 非中心式网络坐标系统安全问题的研究[D]. 清华大学, 2009.
[57] https://pdos.csail.mit.edu/archive/p2psim/kingdata/
[58] Pietzuch P, Ledlie J, Seltzer M. Supporting network coordinates on PlanetLab[C]// Conference on Real, Large Distributed Systems. USENIX Association, 2005:19-24.
[59] 黄立. 基于轻量级J2EE框架的异构数据库专用中间件研究[D]. 河北工业大学, 2006.
[60] 许龙霞. 基于共视原理的卫星授时方法[D]. 中国科学院研究生院(国家授时中心), 2012.
[61] 陈洪卿, 陈向东. 北斗卫星导航系统授时应用[J]. 数字通信世界, 2011, No.78(6):54-58.
[62] 陈敏. 基于NTP协议的网络时间同步系统的研究与实现[D]. 华中科技大学, 2005.
[63] 林涛. 基于精确时钟协议的网络运动控制系统的研究[D]. 河北工业大学, 2007.
[64] 鲁美连. 嵌入式NTP网络时钟源的研究与开发[D]. 华中科技大学, 2007.
[65] 张延辉. 智能变电站同步对时网络优化方案研究[D]. 华北电力大学(北京), 2011.
[66] 徐丹丹. 小型服务器间时间同步软件的设计与实现[J]. 2014.
[67] 苏伟, 杨斌. 嵌入式分布系统中网络设备的时间同步[J]. 单片机与嵌入式系统应用, 2012, 12(2):12-14.
[68] 王玉洁. 数字化变电站中IEEE1588对时协议的实现[D]. 华北电力大学(北京), 2010.
[69] 张延辉. 智能变电站同步对时网络优化方案研究[D]. 华北电力大学(北京), 2011.

攻读学位期间发表的学术论文和主要工作

一、发表的论文

[1] **赵玉琦**，李兵，熊燚铭，刘晖，王静，等，基于 IP 坐标系统的 QoS 优化方法，小型微型计算机系统，（已录用）

[2] Cheng C, Li B, Li Z Y, **Zhao YQ(赵玉琦)** et al. Developer Role Evolution in Open Source Software Ecosystem: An Explanatory Study on GNOME[J]. 计算机科学技术学报(英文版), 2017, 32(2):396-414.

[3] Ding Y, Li B, **Zhao Y（赵玉琦）**, et al. A Novel Approach to Extracting Posts Qualification from Internet[M]// Geo-Spatial Knowledge and Intelligence. 2017.

二、参与的科研项目

[1] 国家重点研发计划,跨界服务融合理论与关键技术,2017YFB1400602

[2] 国家重点研发计划,网络大数据的数据保护与隐私保护,2016YFB0800401

三、获得的奖励

[1] 武汉大学 2014-2015 年度获得国家奖学金

[2] 武汉大学 2015 年度优秀共青团员

[3] 获得武汉大学 2015-2016 年度优秀研究生标兵。

[4] 武汉大学 2016 年度优秀共产党员

[5] 武汉大学 2016-2017 年国家奖学金

[6] 武汉大学 2017 年阮立平专项奖学金

[7] 武汉大学 2017 年度华为奖学金

致　谢

漫漫人生路，求学十余载。转眼间，我在武汉大学已经度过了七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，你们是我求学生涯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，为硕士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。师从李兵老师以来，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。

同时我还要感谢课题组的马于涛老师、王健老师和李增扬老师，感谢课题组何鹏师兄、潘伟丰师兄、熊伟师兄、杨荣师兄正是你们无私的奉献和专业的指导，使我能够顺利完成硕士期间的学习。

最后，我要深深地感谢我的父母和家人。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。

2018 年 4 月　珞珈山下