附件1



**硕士学位论文开题报告**

才新

模式识别与智能系统

基于SDN网络的视频流媒体传输性能研究

2017年 12月 26日

二、论文选题的目的和意义（要求说明国内外现状研究状况）。

**2.1论文选题的目的和意义**

互联网时代，视频流媒体业务在人们的日常生活中不可或缺。据预计，2015～2020年互联网上出现的视频流量以复合年均增长率（CAGR）为３１％的速度增长[1]。视频直播是一种典型的实时视频流媒体应用，为保证用户能获得较好的服务质量，通常需要高带宽、低延时等比较苛刻的网络环境。事实上，现有互联网采用的是尽力而为的转发方式且是完全分布式的控制形式，这导致管理员无法对发生阻塞的链路进行精准定位，也不清楚业务报文在哪条路径上传输，是否有更优路径存在等情况，容易造成直播时视频流在传输过程中出现拥堵，数据不能在限定时间到达，就会导致直播出现中断，这将严重影响用户体验，甚至限制视频流媒体业务在互联网浪潮中的发展。

为了提高多媒体业务的服务要求，学术界和产业界提出了多种服务质量（Quality of Service，QoS）管理方案，如IETF提出的尽力而为服务模型、综合服务模型、区分服务模型和多协议标签转换，但都是在传统网络架构上添加新的协议，使原本就臃肿不堪的网络负担更重。软件定义网络**[2]**（Software Defined Network，SDN）作为新型网络架构，其核心思想是将网络设备控制面与数据面分离，从而实现了网络流量的灵活控制，使网络作为管道变得更加智能。引入新型SDN网络架构，利用其集中控制的优势，能够获得全网拓扑、流量变化等情况，建立全局网络视图，为全局优化决策提供基础支撑。通过路由算法动态的为直播视频流等非固定视频流选择合适的路径进行传输，减少传统路由的路径选择造成的延时与拥堵，提高视频流媒体的服务质量。

本课题研究了SDN架构下的动态视频流媒体传输的性能问题。主要是根据动态视频流媒体传输特性，选择其敏感的QoS度量参数，根据控制器实时收集网络状态信息，在网络中出现拥堵时根据路由算法及时调整视频流的路径，减少拥堵造成视频直播中出现卡顿、甚至是中断等不良现象，同时降低对其他流的影响，提高流媒体传输的性能。

**2.2国内外现状**

**2.2.1 SDN的国内外研究现状**

SDN技术并不是突然爆发成为网络技术前沿的，实际上一些早期的初步措施已存在多年，并最终为SDN技术的创新打下基础。在国内，虽然起步相对较晚，但是近些年，各大高校以及华为、中兴等公司也积极参与SDN相关项目的研究。

2003年，IETE提出的转发与控制单元分离**[3]** （Forwarding and Control Element SeparationFramework，ForCES），首次提出控制面与转发面分离的概念，以及一个网络框架，这个框架里已经有部署网络的规模和在SDN的雏形。虽然提交了多项RFC草案，但由于其重点在于理论创新和功能建模，未能在真实的网络中部署。2005年，4D建议书**[4]**《A Clean Slate 4D Approach to Network Control and Management》开创性地提出了网络技术从分布式网络元素向中央控制器的转变。

2006年，在Clean Slate项目中，斯坦福大学Nick McKeown教授带领的科研团队提出了控制转发分离的概念并应用在校园网络创新中，由此SDN的概念诞生，并于2008在ACM SIGCOMM 发表了题为《OpenFlow: Enabling Innovation in Campus Networks》的论文**[4]**，首次详细地介绍了OpenFlow 的概念。2011年3月，非营利组织ONF（Open Networking Foundation）正式宣布成立。其致力于SDN标准化、规范相关协议和产业化的推广。SDN概念从2006年被提出，十余年间得到深远的发展，被认为是最佳的解决传统网络弊端的网络架构形式。

在SDN的应用上，国外的谷歌、微软、facebook等互联网公司均走在了前列，他们投入了大量的科研力量，谷歌建立了以SDN为建构的数据中心；与此同时，思科、IBM 等 IT 厂商也加紧了研制SDN控制器和交换设备的进度，取得了里程碑式的突破。

在国内，2012年，由清华大学主导，国内多个知名大学如东南大学、北京大学、北京邮电大学参与的国家“863“项目，该项目提出未来网络体系结构创新环境FINE(Future Internet innovation Environment)。基于FINE体系结构，将支撑各种新型网络体系结构和IPv6新协议的研究试验。由此拉开了国内各大高校、研究机构和主流公司对SDN研究的序幕。

浙大XFlow—基于SDN的可重构网络体系架构，浙大从网络操作系统和SDN架构这两个方向进行深入的研究。并于2014年4月在第二届全球开网络峰会上演示了XFlow，其基于应用功能模块化的设计思想并具有动态重构机制。

进入2015年，我国的电信行业的发展也促进了我国SDN研究的发展，中国联通浙江分公司联手华为部署了T-SDN，这一事件标志着我国在SDN研究领域取得了重大的突破，联通与华为在我国率先推动了SDN的研究与试点，这也是我国在推动SDN发展与商业部署上迈出的重要一步。通过这一举措，华为的SDN解决方案也被业界广泛的认可，解决方案在业界持续领先。

**2.2.1基于openflow的视频流媒体分发策略国内外现状**

在基于 Openflow的视频流媒体分发策略中，业界还没有实现一个统一的方案.大致可分为一下几种：

流媒体qos路由.文献[6]在现有网络 QoS 模式基础上提出了 Open Qo S 网络结构，采用动态路由的方式根据网络状况，动态改变路由路径，进而提高网络 Qo S。文献[7]采用带时延约束的拉格朗日松弛路由算法计算视频流转发路径。但该论文只单独研究了视频流的转发策略，并未考虑链路传输过程中数据流业务的多样性和复杂性

流媒体P2P路由。文献[8]针对P2P直播视频流媒体，提出一种基于OpenFlow的跨层 RASP（Rent-a-Super Peer）新方法，允许ISPs提供基于P2P实时视频流的网络层多播功能的服务。同时提高了用户、内容提供商和ISPs的P2P视频传输质量。文献[9]提出一种基于OpenFlow的Flash P2P流媒体传输架构的设计方案，通过在OpenFlow的多级流表前加上tag流表，弥补OpenFlow协议无法在应用层实现视频流的控制转发的功能，进而将OpenFlow的控制转发功能引入到传统的FlashP2P网络拓扑中，实现Flash P2P流媒体分片的高速转发，从而提高视频流媒体的服务质量。

流媒体SVC路由。文献[10]提出了一种针对流媒体的基于 Openflow 的可分级视频编码流媒体传输方案，通过构建基础层和增强层2个独立路由，实现了动态网络下SVC流媒体分级自适应高效传输。文献[11]提出一种基于OpenFlow网络的分级码流分发方法，给出分级视频流的分发模型、失真计算方法和具体分发步骤。采用率失真优化的方法恰当地选择可分级码流，保证传输和接收的可分级码流率失真性能较好。

流媒体RTSP/RTP路由。文献[12]通过对SDN网络状态信息和 RTSP/RTP 流媒体数据包的拦截分析，做出优化的 RTP 路由决策。实验表明此方案能够提高网络资源利用率。

本课题在SDN网络下对视频流媒体传输性能进行研究，视频采用动态视频即直播视频，保证视频流的不确定性，并充分考虑传输过程中数据流业务的多样性和复杂性，在模拟网络环境中加入其他数据流，在链路拥堵的情况下，及时调度视频流，减少拥塞，提高传输性能。

三、研究工作的主要内容和重点。

**3.1主要内容**

本课题结合SDN技术，对直播系统进行设计，提高直播系统中视频的传输性能。主要研究内容包括：

（1）研究软件定义网络的理论架构，对SDN的控制层、数据转发层、应用层结构及作用进行深入了解；学习南向OpenFlow接口协议以及其与控制器通信的原理；并对北向接口有所了解；了解转发设备（一般指路由器和交换机）与传统的网络设备工作原理的不同；对主流的SDN控制器架构及特点深入研究。SDN架构如图1所示：



图 1 SDN架构示意图

(2)熟悉视频流媒体传输的原理与技术，掌握QoS机制及直播视频在网络中传输需要的QoS多约束。

(3)掌握主机与服务器通信原理，了解摄像头挂载及服务器如何获取摄像头视频。

（4）熟悉并掌握dijkstra算法基本应用原理，掌握遗传算法的基本工作原理及其在路由寻路中的应用。

（5）熟悉Mininet使用原理并熟练搭建网络环境。Mininet [13]是一个轻量级的网络仿真器，其引入虚拟化技术，可以在短时间内在一台电脑上模拟出与真实网络相媲美的仿真网络：完整的网络主机、交换机和链接，便于实验测试和演示。另外，Mininet提供了对最新的OpenFlow协议提供完整支持，可以创建出交换机和主机以及它们相互连接而成的网络。本课题拟采用mininet作为实验的网络环境，研究SDN网络下的动态视视频流媒体传输性能。

（6）验证在SDN网络下算法的可行性。对视频流重路由对视频传输性能的提高。熟练使用视频传输性能评价方法。

**3.2研究重点**

基于以上研究内容本文的研究重点如下：

（1）Mininet中网络环境搭建

（2）客户端与服务器通信

（3）python语言编程

（4）智能路由算法的实现

（5）流媒体传输性能评价

1. 研究方案、技术路线及措施，所需的技术条件及试验条件，存在的主要问题和关键技术。

**4.1研究方案**

近年来，网络直播变得越来越流行，直播系统对网络的性能有着极高的要求，提高直播中视频传输的性能具有深刻意义。本文拟采用Mininet仿真器搭建一个基于SDN网络直播系统作为案例，使用Ryu控制器作为网络操作系统，使用OpenFlow 1.3作为控制器与底层交换设备的通信协议。基于SDN网络架构的优势，提高直播视频流传输的性能。总体架构如图2所示：



图 2 总体架构示意图

相应的数据流第一次进入边缘路由器后，路由器没有对应的流表，发送packet-in给控制器，控制器为所有流采取dijkstra路由算法。控制器周期的获得网络的状态信息，若发现网络拥堵，触发路径计算模块为视频流媒体重路由，并下发新的流表，同时将这个流表设为高优先级(防止视频流匹配到之前的流表)，减少由于拥堵造成客户端直播视频出现卡顿等问题。算法流程如图3所示：



图 3 算法流程示意图

基于以上内容，本文分为以下模块如下：

（1）搭建网络环境：应在mininet上搭建网络环境，创建网络拓扑，设定相关参数，并测试主机之间能否可以正常通信（使用ping命令）。

（2）主机与服务器通信：需要获得实时的视频，将摄像头挂载在mininet虚拟出来的服务器端;服务器将从摄像头中获得的流媒体视频传输给客户端;在进行视频传输之前，应保证服务器与客户端可以正常通信。

（3）控制器上扩展的模块包括：拓扑发现模块，流区分模块，信息统计模块，路径计算模块，流表下发模块。各个模块的作用相辅相成，共同协作完成对流媒体的控制。

（4）传输性能的检验：本文主要通过SDN实时获取网络状态信息，检测到网络中出现拥堵，根据路由算法实现直播视频流媒体的重路由，提高视频的传输性能，因此需要与传统路由算法进行对比，验证本文中的路由算法的流媒体传输性能。

**4.2技术路线及措施**

本课题的技术路线主要由搭建网络环境、客户机与服务器通信、SDN控制器、QoS路由数学模型，流媒体选路算法设计，实验验证六部分构成，下面逐一进行分析：

4.2.1网络环境的搭建

本课题采用Mininet模拟整个测试网络环境，在VMware虚拟出的ubuntu系统中安装Mininet，Mininet支持自定义网络，可以搭建一套媲美真实硬件环境的复杂网络。考虑到真实网络环境的复杂性，需使用python语言完成复杂网络拓扑的编写。网络中往往是多种数据流并存，因此不能只是单独存在视频流，需要用Iperf软件生成了模拟真实网络环境的随机流量，模拟真实网络中的流量分布。

4.2.2主机与服务器通信

将摄像头挂载在服务器上（执行linux挂载命令），服务器调用摄像头（使用python调用camera的saveSnapshot方法获取视频），获取摄像头捕捉到的视频，通过socket建立的网络通信，将视频发送到客户端。

Socket是进程间通信的一种方式，与其他进程间通信的主要不同是：能够实现不同主机间的进程间通信。要解决网络上两台主机间的进程通信问题，首先要唯一标识该进程，在TCP/IP网络协议中，就是通过（IP地址、协议、端口号）三元组来标识进程的。解决了进程标识问题，就有了通信的基础了。Socket编程流程如图4所示：



图 4 socket 编程流程图

4.2.3 SDN控制器

(1)控制器选择

控制器是整个系统的核心，其类似于一个网络操作系统，同时它又是一种复杂程度较高的大型程序，为保证整个网络可靠而有效地运行，必须使用一个高质量的控制器避免控制器成为ＳＤＮ的瓶颈。目前市场上多种开源和商用的SDN控制器可用，如Floodlight、OpenDayLight、ONOS、Ryu、NOX/POX等。

OpenDayLight具有高度模块化、可扩展、可升级、支持多种协议的特点，基于java开发的，已是备受瞩目的开源SDN控制器；ONOS是开源的SDN网络操作系统，主要面向服务提供商和企业骨干网，不适合个人研究上使用；floodlight是基于java开发的，是早期流行的SDN控制器之一，由于其更新速度慢，已不在广泛使用；NOX是基于C++开发的，开发成本高，逐渐在控制器中没落，兄弟版的POX是完全基于Python的，适合初学者，但其在架构和性能上也存在缺陷，已不在广泛使用；Ryu 是由日本 NTT 公司贡献的基于组件的开源 SDN 框架，采用 Python 语言开发，支持openflow协议多种版本，提供 GUI 界面用于管理和监控网络拓扑和状态，模块清晰，可扩展性好。基于以上分析可知OpenDayLight和Ryu是非常合适的控制器，由于python语法简洁清晰、结构清晰，易于学习阅读，跨平台。提供大量的第三方库，善于做科学计算，是做爬虫、算法、机器学习等方面的研究的不二之选，所以本次课题选择Ryu作为控制器。

Ryｕ是基于组件框架设计的，组件是以一个或多个线程的形式存在，通过组件提供的接口可方便控制组件状态和事件产生。封装了具体消息的事件可以被多个组件共享，所以事件对象是只读的。Ryu架构图如图5所示：



图 5 Ryu 架构图



（2）控制器模块分析

控制器上扩展的模块包括：拓扑发现模块，流区分模块，信息统计模块，路径计算模块，流表下发模块。

（1拓扑发现模块：拓扑发现模块主要获取网络中所有的OpenFlow交换机结点及交换机之间的链路情况，获取的信息包括：一是所有交换机，二是链路（包括源交换机、目的交换机、源端口、目的端口）。在控制器上使用LLDP协议功能（链路层发现协议）能够实现拓扑发现，简单、扩展性强。

(2信息统计模块：获取网络状态的实时信息，通过对连接到控制器的所有交换机发送状态请求函数来获取所有端口状态信息，并对这些信息进行处理，将信息提供给路径计算模块。Openflow提供对每个流表、每个流、每个端口三种形式的数据进行收集。本文中链路利用率超过75%即为链路拥堵，需要为视频流重路由，提高视频的传输性能。链路利用率计算为：

(1)

（2）

其中，表示0至t1时刻的路由端口接收的字节数，同理。

t1至t2时间段路由端口发送的字节数。

表示t1至t2时间段路由端口的流量速度。

表示路由端口的总带宽。

表示路由端口的链路利用率。

（3流区分模块：本模块主要在网络中含有多种流的情况下，区分出视频流和其他流。使用基于HMM的深度流检测( Deep Flow Inspection，DFI) 技术对数据流进行识别。利用数据包到达的时间间隔和数据包长度信息作为区分视频流和普通文本流的依据。

（4路径计算模块:路径计算模块从其他模块获得网络信息，发现链路拥堵，根据路由算法为视频流计算路由策略。Open Flow 交换机是使用流（Flow）的概念标识网络流量的，流表是由控制器生成和维护的，交换机本身只按照流表内容进行相应操作，并不生成流表。因此本文的视频流媒体路由策略主要针对 Open Flow 交换机网络中流的选路策略，通过在控制器上整体基础功能模块上进行扩展设计，并不针对某个特定的路由协议。

**4.2.4 QoS路由数学模型**

给定一个有权值的无向图G(V，E)，V表示网络中所有节点，E代表链路的集合，QOS路由就是找到满足QOS要求的源节点到目的节点的一条路径P(s，d).

实时视频业务对网络要求比较高，以丢包率、时延、时延抖动以及带宽约束的QoS路由模型如下：

（1）

（2）

（3）

（4）

（5）

（6）

公式（1）为带宽的约束，为剩余带宽，B为要求的最小带宽。公式（2）为时延的最小约束，为要求最大时延。公式（3）为丢包率的最小约束，为要求最大丢包率。公式（4）为时延抖动的约束，为要求最大的时延抖动。公式(5)为代价的计算，分别表示路径时延代价系数、路径丢包率路径时延抖动代价系数，通常根据约束的优先策略进行分配。

**4.2.5 流媒体路由计算-遗传算法**

常见的算法主要分为启发式算法和元启发式算法（也叫智能优化算法），本文的多qos约束文体，需用元启发式算法来解决。

智能算法中的蚁群算法在求解性能上具有较强的鲁棒性和搜索较好解的能力，但一般需要较长的搜索时间，而本文中对于视频流重路由需要快速的寻找可用的路径即需要快速的响应。粒子群算法适合求解实数问题，算法简单，计算方便，求解速度快，但是存在着陷入局部最优等问题。模拟退火算法的局部搜索能力较强，运行时间较短；但其全局搜索能力差，容易受参数的影响。遗传算法是一种全局优化[概率算法](http://www.so.com/s?q=%E6%A6%82%E7%8E%87%E7%AE%97%E6%B3%95&ie=utf-8&src=internal_wenda_recommend_textn)，搜索使用评价函数启发，过程简单，计算量小；搜索从群体出发，具有潜在的并行性，效率相对较高，速度快。综合以上几点，本文选择遗传算法作为路由计算的算法。

遗传算法是一种模拟生物在自然环境中的遗传和进化过程而形成的一种自适应全局优化概率搜索算法，已在组合优化、模式识别、经济预测等领域广泛应用。遗传算法的基本思想来源于遗传进化，主要是借助于生物进化机制与遗传学原理，按照自然选择和适者生存的原则，利用简单的编码技术和繁殖机制，模拟自然界生物群体优胜劣汰的进化过程，实现对复杂问题的求解。多QoS 约束路由遗传算法设计如下：

（1）个体初始化：带宽是凹性参数，在初始化种群前，对网络拓扑进行遍历，把链路带宽和给定限制的带宽对比，小于限制带宽的链路去掉，形成新的拓扑图。如若得到的拓扑图是非连通的，且源节点和目的节点不在同一连通子图里则无法提供服务。本文假设形成的新的拓扑图是连通的。之后不再考虑带宽这一条件。编码方法可以分为三大类：二进制编码法、浮点编码法、符号编码法。其中二进制编码是常用的编码方式。由二进制[字符集](https://baike.so.com/doc/201915-213495.html){0，1}产生通常的0，1[字符串](https://baike.so.com/doc/630752-667548.html)来表示问题空间的候选解，本文中编码时某个节点等于0表示未选择该节点，等于1表示选择该节点。

（2）计算个体适应度：适应度函数是根据路径的代价来计算的，是评价路径是否满足要求同时路径是否更优的标准。同时适应度函数的设计也是遗传算法的关键部分，直接影响到算法收敛的速度，以及能否找到优化解。本文考虑传输所耗费的代价，因此个体适应度应是代价的函数。

适应度函数设计须满足的条件：适应度函数应是单值非负的，具有连续性，可最大化；能够反应对应解的的优劣程度；函数应尽可能简单，减少计算的复杂性。

（3）选择操作

从群体中选择优胜的个体，淘汰劣质个体的操作叫选择。选择的目的是把优化的个体直接遗传到下一代或通过配对交叉产生新的个体再遗传到下一代。选择过程是一种基于适应度的优胜劣汰的过程。目前常用的选择算子有以下几种:适应度比例方法、随机遍历抽样法、局部选择法。轮盘赌选择法作为适应度比例方法的表现形式是比较常用的遗传算法选择方法：每条路径被选中的概率和其适应度成正比关系，即适应度越高的个体被选中的概率就会越高，反之被选中的概率越低.某个个体被选取的概率Pi 表示为 。

(4)交叉：交叉指把两个父代个体的部分结构加以替换重组而生成新个体的操作。交叉算子根据交叉率将[种群](https://baike.so.com/doc/4794484-5010571.html)中的两个个体随机地交换某些基因（本文指的是交换机节点），能够产生新的组合。在路由寻路中的应用中体现了其较强的全局搜索能力，它扩展了求解空间，并为寻找到最优路径发挥着重要作用。最常用的交叉算子为单点交叉，其原理为选择某个点，然后以此分为左右部，两个[基因](http://www.so.com/s?q=%E5%9F%BA%E5%9B%A0&ie=utf-8&src=internal_wenda_recommend_textn)的左右部互相交换基因[序列](http://www.so.com/s?q=%E5%BA%8F%E5%88%97&ie=utf-8&src=internal_wenda_recommend_textn)，例个体*A*:1001111与*B*:0011001，经过交叉产生两个新个体*C*：1001000与*D*：0011111。

（5）变异：变异操作是按照一定的概率对个体的基因组进行改变，以增加基因的多样性。

（6）终止条件：当最优个体的适应度达到给定的阈值，或者最优个体的适应度和群体适应度不再上升时，或者迭代次数达到预设的代数T时，算法终止。算法流程图如图6所示：



图 6 多 QoS 约束路由的遗传算法流程图

遗传算法中需要选择的运行参数主要有群体大小*M*、交叉概率*pc*、变异概率*pm*、终止代数*T*等，这些参数的选择与遗传算法效率密切相关。*M*太小，则会降低种群多样性，难以求出最优解；*M*太大，又包含太多冗余信息，影响算法执行效率。变异概率太小时难以产生新的基因结构；太大使遗传算法成了单纯的随机搜索。交叉概率太小时难以向前搜索；太大则容易破坏高适应值的结构。通常情况下，如何选择合适的参数，需要在求解过程中通过一定的试验来确定。

**4.2.6实验验证**

验证分为以下两部分

（1）传统算法与文中算法选出的路径进行对比，是否满足多QoS约束限制。

（2）使用PSNR(峰值信噪比)作为评价视频画质方法。

**4.3所需技术条件及实验条件**

**4.3.1技术条件**

调阅收集有关信息和资料，包括计算机网络知识及SDN架构相关知识、OpenFlow协议有关知识、链路发现、拓扑相关知识、求解多QoS约束问题的多种智能算法、python编程。

**4.3.2实验条件**

（1）PC机一台，Ubuntu系统虚拟机，mininet仿真器。

（2）服务器与客户端通信。

**4.4存在的主要问题和关键技术**

**4.4.1存在的主要问题**

（1）本文中所有的模块都运行在同一控制器上，一个控制器控制所有的交换机，一旦控制器出现问题，就会导致网络崩溃。可以采用多个控制控制一个交换机，分布式控制，缓解单一控制器的压力。OpenFlow1.3版本中是支持多控制器控制同一交换机的。

（2）如何设计适应度函数，在具体应用中，[适应度](https://baike.so.com/doc/5719794-5932522.html)函数的设计要结合求解问题本身的要求而定，本文是在路由寻路中采用遗传算法，[适应度](https://baike.so.com/doc/5719794-5932522.html)函数设计直接影响到遗传算法的性能，包括收敛速度以及能否找到最优解。

（3）如何对原有的选择算子进行改进，以提高结果的准确性以及群体大小*M*，交叉、变异概率的选择。

（4）本文是在sdn网络环境中，对直播视频进行性能上的提高，为了简化实验，只选择了一个客户端。事实上，存在同一时间多个客户端同时请求直播视频的情况，考虑现实情况，应能实现多个客户端同时请求并提高传输性能。

**4.4.2关键技术**

mininet搭建网络环境；主机与服务器通信；路由算法;python编程

五、预期达到的目标

（1）完成服务器与客户端之间的视频传输。

（2）在网络出现拥堵，多种数据流同时存在的情况下准确调度直播视频流。

（3）对比本文路由算法传输流媒体视频与传统路由算法传输的性能差异。

（4）在多客户端同时请求的情况下，准确调度视频流。

（5）完成小论文一篇。

六、研究计划及进度

2017年10月-2017年12月：选定课题，大量查阅文献，进行课题调研。选定最佳设计方案，确定要采取的技术手段和设计路线，完成开题答辩。

2018年1月-2018年3月：SDN、mininet相关知识学习、路由算法的深入理解、python语言的学习。

2018年4月-2018年7月：mininet搭建实验环境、完成主机与服务器通信模块、控制器相应模块的编写。撰写中期报告，完成中期答辩。

2018年8月-2018年9月：使用iperf对流媒体传输进行性能测试，根据测试结果调整算法，使传输性能达到最佳状态。

2018年10月-2018年12月：对系统进行优化，完成多客户端请求。

2019年1月-2019年3月：撰写论文，完成最终答辩 。

[1] [Cisco Visual Networking Index: Forecast and Methodology, 2015–2020](http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html/)[EB/OL]。[2016-07-15].http:∥www.cisco.com.

[2] 左青云, 陈鸣, 赵广松,等. 基于OpenFlow的SDN技术研究[J]. 软件学报, 2013(5):1078-1097.

[3] Yang L, Dantu R, Anderson T, et al. Forwarding and Control Element Separation (ForCES) Framework[J]. Heise Zeitschriften Verlag, 2004.

[4]张卫峰. 深度解析SDN:利益、战略、技术、实践[M]. 电子工业出版社, 2014.

[5] Mckeown N, Anderson T, Balakrishnan H, et al. OpenFlow:enabling innovation in campus networks[J]. Acm Sigcomm Computer Communication Review, 2008, 38(2):69-74.

[6] 陶智勇, 高潮, TAOZhiyong,等. 基于OpenFlow网络的流媒体传输QoS研究与设计[J]. 电视技术, 2015, 39(3):133-135.

[7] Egilmez H E, Civanlar S, Tekalp A M. An Optimization Framework for QoS-Enabled Adaptive Video Streaming Over OpenFlow Networks[J]. IEEE Transactions on Multimedia, 2013, 15(3):710-715.

[8] Ruckert J, Blendin J, Hausheer D. RASP: Using OpenFlow to push overlay streams into the Underlay[C]// IEEE P2P 2013 Proceedings. IEEE, 2013:1-2.

[9] 季一木, 谈海宇, 孙延鹏,等. 基于Openflow的Flash P2P流媒体传输协议研究[J]. 计算机技术与发展, 2015, 25(11):82-86.

[10] 黄 韬，张 丽，张云勇，等． 基于 Open Flow 的 SVC 流媒体时延自适应分级传输方法［J］． 通信学报，2013，34( 11) :121－128．

[11] 卢冀, 张之义, 王俊芳,等. 基于Openflow网络的可分级视频流分发方法[J]. 无线电工程, 2014, 44(1):1-3.

[12] Karl M, Gruen J, Herfet T. Multimedia optimized routing in OpenFlow networks[C]// IEEE International Conference on Networks. IEEE, 2013:1-6.

[13] 张连成, 奚琪, 郭毅,等. 基于Mininet模拟环境的软件定义网络实验课程设计[J]. 计算机教育, 2015(6):104-107.

[14] Casado M, Freedman M J, Pettit J, et al. Ethane: taking control of the enterprise[M]. ACM, 2007.

[15] Jain S, Kumar A, Mandal S, et al. B4: experience with a globally-deployed software defined wan[M]// ACM SIGCOMM Computer Communication Review. ACM, 2013:3-14.

[16]Open networking fundation. http://www.opennetworking.org . 2014

[17] 许名广, 刘亚萍, 邓文平. 网络控制器OpenDaylight的研究与分析[J]. 计算机科学, 2015, 42(s1).

[18] 吴许俊, 王永利. 基于POX的软件定义网络的研究与实践[J]. 计算机测量与控制, 2013, 21(12):3414-3417.

[19] 林玉侠, 朱慧玲, 马正新,等. QoS路由度量参数的选择问题研究[J]. 电信科学, 2003, 19(7):22-27.

[20]IETF: Internet Engineering Task Force. <http://www.ietf.org/>.

[21]3GPP TS 36．211，Evolved universal terrestrial radio access( E-UTRA)，physical channels and modulation． v 9.0.0［S］． 2009．

[22] 赵钊. 基于OpenFlow的视频流媒体路由选择算法[D]. 深圳大学, 2017.

[23] 朱冠宇. 基于遗传算法的Qos路由选择策略研究[D]. 华中科技大学, 2004.

[24] 李勇, 刘学军. 基于OpenFlow的SDN网络中路由机制研究[J]. 移动通信, 2015, 39(7):51-56.

[25] 魏凯. 基于蚁群算法SDN负载均衡的研究[D]. 吉林大学, 2015.

[26]彭璐, 何加铭. 基于遗传算法的多约束QoS单播路由算法[J]. 移动通信, 2015(6):76-81.

[27] 文强. SDN网络业务量工程技术研究[D]. 电子科技大学, 2016.

[28] 何小燕, 费翔. Internet中一种基于遗伟传算法的QoS路由选择策略[J]. 计算机学报, 2000, 23(11):1171-1178.

[29] 崔逊学, 林闯. 一种带约束的多目标服务质量路由算法[J]. 计算机研究与发展, 2004, 41(8):1368-1375.

[30] 刘萍, 冯桂莲. 图的深度优先搜索遍历算法分析及其应用[J]. 青海师范大学学报(自科版), 2007(3):41-44.

[31] 姜立立, 曾国荪, 丁春玲. 数据流特征感知的交换机流表智能更新方法[J]. 计算机应用, 2016, 36(7):1772-1778.

[32] 梁宇宏, 张欣. 对遗传算法的轮盘赌选择方式的改进[J]. 信息技术, 2009, 33(12):127-129.

[33]靳洪兵. 基于OpenFlow的数据中心网络路由策略研究[D]. 重庆大学, 2014.

[34]李臻, 杨雅辉, 谢高岗,等. 一种基于数据流计数的概率衰落大业务流识别方法[J]. 计算机研究与发展, 2011, 48(6):1010-1017.

[35] 柯友运. 面向SDN的流量调度技术研究[D]. 西安电子科技大学, 2014.