# 第一章 绪论

1.1 研究背景与意义

随着计算机科学和多媒体技术的快速发展，VR（Virtual Reality, 虚拟现实）技术受到了人们越来越多的关注，虚拟现实技术构建的虚拟环境与真实环境真假难辨，用户从传统的以视觉为主的局限体验扩展到包含视觉、听觉、触觉甚至嗅觉的综合感官体验，在虚拟环境中体验到彷佛身处在真实世界的感觉。

虚拟现实主要有三大特性：沉浸式（Immersion）、交互性（Interaction）和构想性（Imagination）。沉浸式是指当用户处在虚拟现实模拟的世界中时，听到的看到的体验到的感受与真实世界完全一致，这是衡量虚拟现实技术优劣的关键指标；交互性指用户可以与虚拟环境进行互动。区别于传统的只能观看的用户体验，虚拟现实技术允许用户通过各种各样的传感器与虚拟环境进行交互，比如用户可以在射击游戏中拿起枪支对目标进行射击，可以蹲起躲避敌人，且整个交互过程遵循各种物理学定律，与真实世界一致；构想性也被称作想象性，用户在虚拟世界中除了可以体验到真实世界的感受外，还可以自由丰富地创建出真实世界不可能发生的事情，激发用户的创新性。

虚拟现实的概念可以追溯到20世纪80年代初，发展至今已经广泛地应用到影视娱乐、设计领域、医疗教育和军事等领域。2018年全球范围内虚拟现实的产业规模已近千亿元人民币[1]，中国的虚拟现实市场规模也在持续扩大，工业和信息化部预计到2021年，我国虚拟现实市场规模将达到544.5亿元，同时工信部表示将支持虚拟现实制造业创新中心筹备建设，推动虚拟现实技术在制造、教育、文化等领域应用[2]。虚拟现实巨大的发展市场，使得人们不断对其进行探索与研究。全景视频作为虚拟现实技术最为流行的应用方向之一，也逐渐成为人们争相研究的重点。

全景视频，又被称作360度视频、沉浸式视频，是一种用户可以随意转变观看角度的三维视频。传统的二维视频的缺点主要是缺乏互动性，用户只能被动观看视频内容，无法像在真实世界中那样可以随意转动头部观看不同的场景画面，全景视频的出现则打破了这种僵局。目前在各大网络平台看到的全景视频主要是利用由多个摄像机组成的全景摄像机拍摄多个角度的视频画面，然后通过特征点提取、特征匹配和融合等一系列步骤将各个角度的视频画面拼接成360度得到的。用户在观看全景视频时需要佩戴特殊的硬件设备VR眼镜或者HTM（Head Mounted Display, 头戴式显示仪），市面上常见的有Facebook的Oculus 和 HTC的Vive，然后用户就处在全景摄像机的位置观看360度的视频画面，用户可以自由地转动头部，视频画面也会随着变化，如同在真实世界中一样，获得一种身临其境的视觉体验。

由于相对于一般的平面视频而言，全景视频更加贴近用户的眼睛，对视频分辨率有着更高的要求，所以即使在相同的单眼分辨率情况下，由于全景视频需要保留360度的画面信息，全景视频的码率也要比平面视频的码率高很多，通常为前者的5倍以上。同时，全景视频对网络系统的传输时延有着较为严格的要求，业界公认的传输系统的MTP（Motion-to-Photos Latency, 头动到视野延迟），即转动头部和视频画面对应切换之间的响应时间必须控制在20ms以内[3]，否则会引起观看者的眩晕感，造成极差的观看体验。因此，全景视频“高码率、低延时”的传输特性对现有的无线传输网络提出了极大的挑战，如何在无线网络中为多用户提供高质量的全景视频服务逐渐成为亟待解决的问题。而在实际观看场景中，由于人眼角度的特性，在某一时刻，用户实际只能看到全景视频中的部分内容，称之为FOV（Field Of Viewport, 视口区域），如果以同等质量传输所有的画面信息势必会造成网络资源的巨大浪费，如何根据视口区域自适应传输全景视频内容也收到越来越多的关注。

另一方面，随着终端技术的发展，接入无线网络的智能设备愈来愈多，无线网络传输的业务类型也多种多样，从传统的二维视频、音频及图文到新兴的全景视频等等，不同类型的业务对网络的QoS（Quality of Service, 服务质量）的需求不一样，如全景视频这种实时业务对系统的传输时延有着较高的要求，而文本这种非实时业务则要求传输系统有着较低的丢包率来保证信息的完整度，如何合理利用有限的带宽资源来满足不同业务的需求也是极为关键的问题之一。在LTE（Long Term Evolution, 长期演进）和5G（5th-Generation, 第五代移动通信技术）中，基站侧的无线资源管理起着极为重要的作用，而其中涉及的资源调度算法则是关键。所谓资源调度，就是通过一定方式计算小区内等待调度的业务的优先等级，然后根据等级顺序将带宽资源依次分配给业务，直到所有待调度业务都满足了传输需求或者带宽资源全被占用。通常而言，资源调度算法需要考虑网络时延、吞吐量、时延以及用户间的公平性等多个方面，一个高效的调度算法则可以在各个方面取得均衡，使得效益最大化。因此，如何有效地利用有限的系统带宽资源，通过合理的调度与分配，为多用户提供高质量低时延的全景视频业务，同时也保证其他传统业务的服务质量，也是目前全景视频传输优化工作的重点与难点。

1.2 国内外研究现状

全景视频一直以来都是虚拟现实领域的重点研究方向，涉及到的技术包括全景视频的拼接、投影到基于视口的自适应传输再到客户端的渲染重建。本文主要聚焦于全景视频的视口预测算法和底层物理资源调度两方面。

1.2.1 全景视频视口预测研究现状

目前全景视频的传输方案主要分为全部传输和自适应传输两种[4]。全部传输是类似于传统视频传输方案，即将全景视频完整的传输给用户，目前主流的流媒体平台（例如YouTube）都是采用这种传输方式。这种传输方案可以保证用户看到全部的视频内容但是对网络负担极大，且会造成较多的带宽浪费。自适应传输方案则是针对视口区域进行差别传输，其中基于tile（区块）的传输方式被认为是传输全景视频的理想方案[5][6]。不同于传输整个画面，基于tile的方式以较高质量传输视口区域所在的tile，而以低质量传输非视口区域的tile甚至选择不传输。这种传输方式可以很好的降低带宽需求，但很依赖准确的视口预测算法，一旦视口区域估计错误，将会导致用户观看区域内出现明显的低质量画面甚至是黑块区域，这将会严重降低用户的观看体验，因此，如何准确有效地预测出用户的视口区域对全景视频的自适应传输有着极为重要的作用。

对视口进行预测，首先要明确的问题是视口是否可预测。文献[7]对视口中心的位置，即视点（viewpoint）数据进行了实验分析。文献主要研究表征视点位置的三个参数：偏航角（yaw）、俯仰角（pitch）和翻滚角（roll）的相关特性。首先分析了三个角度的累积分布图，实验结果表明偏航角的变化幅度最大，最难预测，其次是俯仰角；然后又分析了三个角度和各自差值的自相关函数，结果显示在500ms内有较强的自相关性，这表明视点数据是可以预测的。

目前全景视频视口预测的方法主要有两种：基于运动轨迹的预测和基于内容检测的预测。前者一直是研究的重点。

基于轨迹预测主要是根据该用户历史一段时间内的视口位置来预测接下来的视口位置，视口位置主要通过头部移动轨迹或者眼球移动轨迹来确定，而这些轨迹数据可以通过HMD的定位功能采集到。

基于运动轨迹的预测可归类为时间序列预测问题，因此很多时间预测的方法被应用进来，如滑动平均、线性回归和机器学习等。文献[8]对比了滑动平均、线性回归和加权线性回归三种预测算法分别预测yaw、pitch和roll三个角度的准确度。滑动平均复杂度最低，即使用历史窗口中所有视点数据的采样点的平均值作为下一时刻的预测值，当数据变化幅度较小时这种预测方法也可以获得不错的预测准确度。线性回归则是利用统计学中的回归模型，根据历史一段时间内的用户视点数据进行拟合，然后通过拟合出来的函数模型获取下一时刻的预测值，加权线性回归则是给予历史数据不同的权重值，如越近的数据拥有越高的权重值，以此来获得不同的影响程度，一定程度上提升预测准确度。[8]的结果表明，线性回归与加权线性回归预测结果要好于滑动平均，当使用历史1s的视点数据预测接下来1s的视点数据时，加权线性回归的预测准确度高达90%以上，但是随着预测时间的增加，预测准确度将会明显下降，当预测时间增加到2s时，预测准确度则会下降到70%。文献[7]则对比了朴素预测，即使用当前的角度值当作预测值，线性回归和神经网络预测未来0.1s~1s范围内的偏航角，实验结果证明神经网络的预测效果要好于前两个算法。

除了使用当前用户的视点数据外，一些研究还探索了跨用户的行为。文献[9]使用了结合KNN（K-Nearest Neighbor, K最邻近） 聚合的线性回归模型，基于当前用户的历史视点，使用线性回归模型预测出当前用户下一时刻的视点后，再结合观看同一视频的其他用户的历史数据，使用KNN来对预测结果进行纠正，进一步提升预测准确度。文献[10]对三个角度的轨迹进行建模，具有相似观看行为的轨迹被聚合在一起，并且为每个聚类计算出不同的轨迹函数。文献[11]则首次使用一种基于密度的聚类算法DBSCAN，在服务器端先对用户进行分类，然后在客户端，通过SVM（Support Vector Machine, 支持向量机）分类器来预测用户的类别，最后预测该类别所对应的观看轨迹。

另外一类是基于内容检测的视点预测。该方法主要从视频显著性的角度进行预测。视频显著性表征了用户对视频不同内容的关注程度。一般来说，视频内容的显著性越强，用户越关注，则内容被观看的概率也越高。近年来，基于深度CNN（Convolutional Neural Networks, CNN）的显著性检测方法相比于传统方法可以取得较优的结果[12][13]。文献[14]训练了一个浅层网络进行显着性检测，并应用了转移学习生成一个深层网络；同样的作者在文献[15]中介绍了SalGAN，这是一种通过对抗进行训练的深度网络；但是，这些网络模型都是应用在传统二维图像或者视频中，直接应用到全景视频时可能会造成中心偏差或者多目标混淆的问题，造成错误的检测。

近些年来出现了一些研究全景视频显著性检测的工作。由于用户观看全景视频是更倾向于观看赤道附近的内容，一些算法就将该特性应用到显著性检测中。文献[16]将全景图像划分为不同的区域，然后给每一部分区域设置不同的显著性权重。由于全景视频投影为二维视频的时候会存在边界失真的情况，文献[17]分别计算出几个投影平面的显著性图，然后取平均作为全景视频的显著性图。文献[18]则是将全景图像映射为6个视口子图像输入到CNN网络中，输出子图像的显著性图，最后将6个显著性图映射到矩形平面，得到全景图像的显著性图。

为了更准确地捕获未来视口与过去视口之间的非线性和长期依赖关系，文献[19]使用CNN开发了一个视口预测模型，模型删除了池化层，并添加了更多的卷积层以增强非线性拟合能力;文献[20]提出了两种深度增强学习模型：首先使用一个仅基于视觉特征来估计视频中每帧热图的离线模型，然后使用一个在线模型，根据过去观察到的头部移动位置以及离线模型得到的热图来预测头部运动。文献[21]提出了一个固定视点预测网络，该网络同时利用过去的视点位置和视频内容特征来预测接下来n帧中的视口轨迹或基于图块（tile）的观看概率。

在时间序列预测场景中常被使用的LSTM（Long-Short Term Memory, 长短期记忆）模型也被逐渐应用在视口预测中。LSTM是一种改进之后的RNN（Recurrent Neural Networks, 循环神经网络），可以解决RNN无法处理长距离依赖的问题;文献[22]使用LSTM编码视口区域扫描路径的历史记录，并将隐藏状态特征与视觉特征结合起来进行未来1秒的预测，结果表明基于LSTM模型的预测优于几种基准方案。

1.2.2 LTE资源调度算法研究现状

为了满足无线网络中多用户不同业务的服务质量需求，如何对有限的无线带宽资源进行有效的资源调度一直是学术人员的研究方向。本文主要聚焦于LTE系统中下行链路的资源调度问题，所以下文只介绍了下行调度的相关研究。

资源调度算法的研究可以追溯到上世纪80年代，最初学者们提出的算法较为简单，忽略无线信道的传输状况，最为常见的是RR算法（Round Robin, 轮询）[23][24]，该算法将资源轮流公平地分配给网络中的用户，而不考虑实时变化的信道条件，RR算法可以获得最好的用户公平性，但系统资源的利用率很差，导致系统吞吐量较小；Max C/I（Maximum Carrier to Interference, 最大载干比算法）[25][26]，该算法则是以系统吞吐量最大化为目标，只考虑用户信道条件的好坏，造成信道条件差的边缘用户一直得不到系统资源；为了同时考虑到无线信道条件和用户间的公平性，学者们提出了PF算法（Proportional Fairness, 比例公平），该算法结合了RR和Max C/I算法的特点，是目前LTE系统中中最常使用的调度算法。但是上述这些算法都没有考虑不同类型业务的QoS（Quality of Service, 服务质量）需求，在实时业务的调度场景中性能较差。鉴于视频和语音通话这类实时业务需求的增加，研究人员们又提出了一些面向实时业务的经典算法，比如EDF算法（Earliest Deadline First, 最早到期优先）[27]，该算法根据每个业务的QoS需求中的最大时延对所有任务队列进行排序，每次优先调度截止时间最小的业务，保证了时延敏感业务的调度，但是忽略了系统的整体吞吐量和用户之间的公平性；MLWDF算法（Modified Largest Weighted Delay, 改进的最大权重时延优先）[28][29]，该算法基于PF算法，综合考虑了分组数据的丢弃时延和信息质量条件，可以在系统吞吐量和实时业务丢包率之间取得较好的平衡。

在这些经典的调度算法的基础之上，大量改进的优良算法也被陆陆续续地提出。基于PF算法，文献[30]对非实时业务和实时业务进行了区分，对于实时业务，改进的算法考虑了其时延特性，保证了时延较为敏感的业务可以优先被分配资源得以调度；文献[31]对两种业务的调度算法均进行了研究，提出了一种改进的时延优先级函数，实验结果表明，改进的算法在系统吞吐量、丢包率和用户公平性等指标上均得到了提升。

除了从网络系统方面对算法进行优化外，研究人员也从用户侧出发，以QoE（Quality of Experience, 用户体验）为优化目标改进调度算法。文献[32]根据QoE常用指标MOS（Mean Opinion Score, 平均意见值得分），评估了LTE网络中三种流行的调度算法（PF、EXP-PF和MLWDF）。实验结果表明，最适合的下行调度算法是MLWDF，其端到端时延小于50ms，并且在可以接受的MOS得分（超过3.5分）下运行最大数量的用户接入（超过50个用户）。当用户超过30时，最为广泛使用的PF算法的端到端时延会增加到200ms以上，这表明PF算法并不适合调度实时业务；文献[33]针对VoIP业务提出一种QoE 驱动的资源调度算法。

为了更好地结合多方面的因素来达到高效调度的目的，跨层调度也得到了深入的研究。文献[34]提出了一种跨层方案，协同无线协议栈的应用层、MAC层和物理层参数共同优化，最大化利用网络资源并提升用户的感知服务质量。仿真结果显示提出的跨层框架可以最大程度的提高用户的感知质量并且在用户的公平性方面得到了显著的改进。文献[35]则在跨层优化的基础上，采取了多点协作的方式，进一步降低了实时业务的传输时延。

目前完全针对全景视频的资源调度研究十分有限。文献[36]提出了一种多路径协作路由方案，以促进5G小型小区网络中的全景视频的传输优化；文献[37]则设计了面向多用户的全景视频自适应传输框架，并通过部署移动边缘计算来增强基站的功能。

在最近的研究中，深度学习的相关方法也被应用在无线网络传输中。针对资源调度问题，文献[38][39]聚焦于基于深度Q学习（Q-Learning）的调度算法，建立MDP （Markov Decision Process, 马尔可夫决策过程）模型来代表系统中各个状态的改变，然后根据不同类别用户的CQI（Channel Quality Index, 信道质量指标）在每个传输时间间隔采用不同的调度规则。

1.3 主要研究内容

综上所述，全景视频的准确预测是传输优化问题的重要前提之一，然而，基于运动轨迹的预测没有考虑视频内容对用户观看行为的影响，在长期预测场景中存在较大问题；基于视频显著性检测的算法复杂度不利用对时延比较严格要求的全景视频，同时显著性检测也未能考虑观看者在年龄、性别、兴趣爱好等方面的区别，在内容提取方面也存在一些误差。因此，本文主要研究了视口预测的常见算法，并在现有研究的基础上，对基于运动轨迹的预测算法进行了改进优化。

随着全景视频等视频业务在无线网络中的增加，如何科学调度分配有限的带宽资源来满足不同用户不同业务的需求，也是亟待解决的问题之一。本文提出了一种新颖的调度算法，根据吞吐量、时延和公平性指标定义了若干种状态，然后基于Q-Learning算法来觉得在每种状态下应该选择何种调度策略来达到系统的性能最优化。

1.4 本文组织结构

本文共分为四个章节，各章节安排如下：

第一章：绪论。该章节主要介绍了全景视频传输的研究背景和研究意义，针对传输框架中的视口预测和资源调度两方面，简要阐述了相关的国内外研究现状和研究内容，最后给出本文的组织结构；

第二章：基于历史观看轨迹的视口预测算法。本章介绍了现有阶段视口预测的相关研究，分析了常见预测方案的不足之处，然后详细介绍了本章提出的基于历史观看轨迹的视口预测算法，首先根据当前用户的历史视口数据，使用长短期记忆模型初步预测未来的视口数据，然后再结合观看同一视频的其他用户的视口数据对预测数据进行矫正，得出最终的视口所在位置。最后阐述了相关实验过程与结果对比分析；

第三章：基于Q-Learning的LTE资源调度算法。本章首先分析了LTE网络的基础架构和关键技术，然后介绍了常见的资源调度算法与资源调度流程，在此基础上，本章提出一种基于Q-Learning算法的下线资源调度算法。该算法根据动态变化的网络状态，使用Q-Learning算法在每个传输间隔选择不同的调度算法，最后使得系统性能最优化。最后介绍了仿真平台与仿真环境，从系统吞吐量、丢包率和用户业务时延等方面对实验结果进行了分析与对比。

第四章：总结与展望。本章节对论文做了回顾总结与未来展望。

# 第二章 基于历史观看轨迹的视口预测算法

根据第一章的介绍，现阶段国内外视口预测主要分为两种方式：基于用户历史观看轨迹和基于视频内容。由于基于视频内容的视口预测算法时间复杂度较高，同时，不同年龄、性别的用户在观看视频时的感兴趣区域差异较大，通过视频敏感性来预测视口会造成较低的预测准确度，因此，本章只聚焦于基于历史观看轨迹的预测算法。首先介绍视口预测相关技术的理论基础，然后详细介绍基于神经网络的预测算法，并结合其他用户的视口数据辅助预测结果，最后，对比分析了其他算法与我们算法的预测准确度。

2.1 视口预测相关技术

视口预测作为全景视频传输优化问题的前提之一，已经出现了一些针对该问题的研究。本节主要介绍了视口的相关定义和现阶段的视口预测算法。

2.1.1 相关定义

（1）视口区域

在给出视口区域的定义前，首先介绍人眼和VR眼镜（或HMD）的特性。

人类眼睛单眼在水平方向最大可以观看到150度左右的画面，双眼在水平方向则可高达188度左右。但是双眼重合视区为120度左右，即在这个角度范围内，观看到的物体才会呈现出立体感。超过水平视角30度之外的视角称为诱导视角，即我们常说的余光，在这些区域的事物人眼是不敏感的。而在垂直方向上，单眼感光区域约为120度，视觉敏感区为60度，如图2-1所示。

### (a) 水平视角 (b) 垂直视角

### 图2-1 人眼视角区域

由于全景视频的特性，必须佩戴相关的观看设备才能观看，这些硬件设备在生产时都会有各自固定的视角区域。目前市面上入门级的VR眼镜的视角区域水平为90度，垂直区域为90度；像Facebook的Oculus和HTC Vive之类昂贵的头显设备的视角区域则为水平110度，垂直90度。

全景视频是一个横向呈360度，纵向成180度的三维视频，但由于人眼特性和全景视频观看设备的参数，在某个特定时刻，用户只能观看到全景视频的一部分内容，这部分就称为视口区域（Field Of Viewport, FOV），如图2-2所示：



### 图2-2 视口区域

为了方便后续的实验说明，本文中所指的视口区域特定为水平110度，垂直90度的矩形区域。

（2）视点

视口区域是一个矩形区域，我们定义该矩形的中心点所在位置为视点。当视口大小固定时，只要知道视点位置即可获取到视口区域。我们给出可以描述视点的几种常用参数如下：

a）x、y、z坐标

我们将全景视频构建成笛卡儿坐标系（Cartesian Coordinates）中的一个单位圆，视点位置即为圆上一点，可以使用空间三维坐标x、y、z来表示。

b）经纬度

全景视频可以通过多种投影方式展开为二维平面视频，常见的投影方式有EPR（Equirectangular Projection, 等距柱状投影）、PSP（Platonic Solid Projection, 多面体投影）和CMP（Cube Map Projection, 立方体投影）等。我们使用国内外主流内容平台都使用的EPR方式将全景视频投影为二维平面视频，如图2-3所示。ERP投影简单来说，就是经典地将三维地球仪打开成二维地图的方式。这种投影方式在赤道部分的误差较小，越往两级误差越大。



### 图2-3 EPR投影示意图

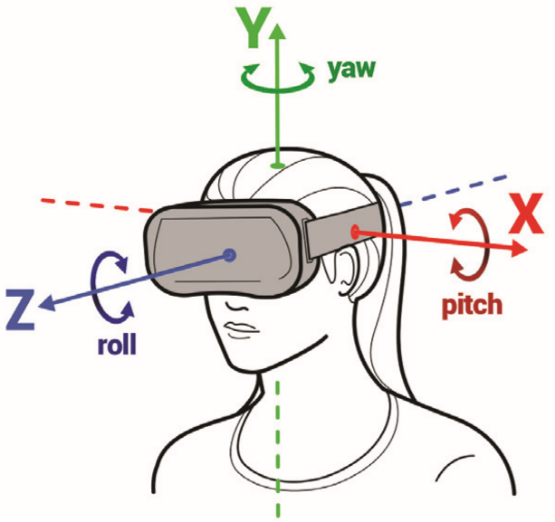
经投影展开后，三维球面上的视点就落在了平面上，我们使用经纬度来表示。经纬度与三维坐标可以通过下面的公式互相转化：

(2-1)

其中，R代表球体半径，为纬度，为经度。

c）偏航角、俯仰角和翻滚角

用户在佩戴VR设备观看全景视频时，头部从某个位置转到另外一个位置的旋转量可以使用三个欧拉角（Euler Angles）来表示，即偏航角（yaw）、俯仰角（pitch）和翻滚角（roll），如图2-4所示。俯仰角为头部旋转的水平夹角，向右转为正方向，范围为-180度到180度；俯仰角为头部旋转的垂直夹角，向上抬头为正方向，范围为-90度到90度；翻滚角则是视线看向正前方，头顶在视线的正交平面左右摆动，以向右摆为正方向，范围为-90度到90度。



### 图2-4 方向角示意图

d）四元数

旋转量使用欧拉角表示直观又简单，但是这种表示方式存在万向节死锁的问题。万向节死锁简单来说，就是如果使用欧拉角来表示旋转过程的话，在某些特定位置，三个互相正交的平面会出现其中两个平面重叠的现象，导致三维信息丢失。为了解决这个问题，研究人员提出了四元数的概念。四元数基于一个定理，即三维空间中的一个物体，绕着某个特定的旋转轴，旋转特定的角度，可以到达三维空间中的任意一个位置，如图 所示。

四元数本质来说是一个高阶负数，表示为，四元数与x、y、z可以通过如下公式进行转化：

公式 四元数转欧拉角

解释一下公式含义。

（3）视频区块

现阶段，全景视频的传输主要有两种方式，一种是传输全部内容，一种是传输部分内容。第二种传输方案中，需要从空间上将全景视频分割成不同的区域，称为区块（tile），如图2-5所示，我们将全景视频分成了16\*8个区块。假设视口区域为A，则基于tile的传输方式只需要传输视口区域所覆盖的6\*4块tile，如图中黄色区域所示：



### 图2-5 tile示意图

2.1.2 常用预测模型

全景视频视口预测主要有两种方法，基于历史运动轨迹和基于视频显著性检测。考虑到后一种方法的时间复杂度不太适合实时预测，本文只探讨第一种预测算法。

基于历史运动轨迹的预测，通常是使用历史的视点数据来预测接下来一段时间的视点数据，本质上可以划分为时间序列预测问题。时间序列的预测是数据挖掘领域热门的研究方向，研究人员们已经提出了许多优秀的预测模型，主要可以分为基于统计学的，基于机器学习算法的和基于神经网络的，本节介绍这三种类别中的具有代表性的预测模型。

（1）移动平均

移动平均（Moving Average, MA）是一种简单的时间预测模型，主要思想是根据历史一段时间（即窗口）内序列的平均值来预测接下来某一时刻的序列值，然后移动窗口继续预测，计算公式如下：

其中，表示t时刻的预测值，为历史n个序列值。

当数据变化范围较小，整体呈现平稳状态时，该方法可以取得较好的预测准确度。在移动平均预测的基础上，又提出了加权移动平均法，即给历史窗口中的每个序列不同的权重值，计算公式如下：

其中，为第n个权重值。

使用该方法进行预测时，权重值的选择尤为重要，最为简单的是经验法和试验法，通常情况下，最近的序列值最能影响未来预测值，所以权重值应该大一些。

（2）ARIMA

整合移动平均自回归模型（Auto-Regressive Integrated Moving Average , ARIMA）是基于统计学的十分流行的一种预测模型。该模型有着完善的理论依据，简单高效。

ARIMA模型是在移动平均自回归模型（Auto Regression and Moving Average, ARMA）的基础上进行d阶差分得到的。差分是将不平稳序列转为平稳序列的常用方法之一，所谓差分，就是使用后一个序列值减去前一个序列值后的结果，一阶差分即进行一次减法运算，二阶差分则在一阶差分的结果上再进行一次减法运算，d阶差分依此类推，计算公式如下：

ARMA模型则是由自回归模型（Auto Regression, AR）和移动平均模型（Moving Average, MA）模型组成，表达公式如下：

其中，是模型的输出，即t时刻的预测值，是AR模型的公式，是系数，代表时间序列值；AR模型主要是探索预测值与历史序列值之间的线性关系。而MA模型主要是探索预测值与预测误差的线性关系。由公式可以看出，ARIMA模型的关键是确定三个超算数p, q, d的值，差分中的d可以通过平稳性检验来确定，p和q一般是通过观察时间预测自相关图（Auto Correlation Function, ACF）和偏自相关图（Partial Auto Correlation Function, PACF）[39]的截尾和拖尾现象来确定。

ARIMA模型简单高效，但仅适用于线性规律，同时关键的p和q的值需要人工判断，且参数一单固定后就无法应对数据的变动，在实际的视口预测模型中，由于用户观看不同视频时的观看行为不同，需要对每一个视频都进行参数估计建立新的模型，工程量大效率低下，因此该模型并不适用于本文探讨的预测场景。

（3）线性回归

随着机器学习技术的发展，研究人员也开始使用一些机器学习的模型来进行预测。线性回归（Linear Regression, LR）[40]是机器学习中较为基础的一种算法。线性回归主要是找到数据之间的规律函数，对于单变量而言，线性回归拟合出一条直线，方程可表示为：，对于多个变量的话，拟合出来的表达式可表示为：

其中，是常数项，为对应的偏回归系数。

基于机器学习的预测算法主要是模型的训练，一般是根据历史窗口的大小将数据集划分为多个子序列段，每个子序列段后的h个值就是待预测的真实值，模型输出的即为对应的h个预测值。在这种方法中通常使用真实值与预测值的均方误差（Mean Squared Error, MSE）做为损失函数（Loss Function），计算公式如下：

其中，n为预测值个数，为系数，为真实值，为对应的预测值。

回归最后就转变成了优化问题，即找到使损失函数最小的解，划分好训练集与测试集之后训练模型即可。

除了线性回归模型，常用的机器学习模型还有支持向量机模型（Support Vector Machine, SVM）[41]和随机森林模型（Random Forest, RF）[42]。相比于基于统计学的预测方法，基于机器学习的方法可以处理非线性关系，同时训练出来的预测模型具有普适性。

（4）多层感知机

由于深度学习的崛起，一些神经网络的模型也被应用到时间序列的预测中来。与机器学习算法类似，我们也可以通过训练神经网络模型来获取到历史序列值与预测值之间的规律关系。

多层感知机（Multilayer Perception, MLP）[47]是最基本的人工神经网络模型，而最经典的MLP分为输入层、隐藏层和输出层三层网络，不同层之间使用全连接，如图所示。

### 图 MLP示意图

隐藏层的输出是由三部分构成的：权重、偏置和激活函数。其中，权重代表着神经元之间的连接强度，权重越大，可能性越大；偏置是模型中重要的参数，偏置的设置是为了正确分类样本，保证通过输入算出的输出值不能随便激活；激活函数起到非线性映射的作用，可将神经元的输出幅度限制在一定范围内，一般限制在-1～1或者0～1之间。最常用的激活函有Sigmoid函数、tanh函数和ReLU函数等，采用哪种激活函数得视具体情况而定。

MLP被应用到了许多预测场景中。文献[43][44]对比了ARIMA模型和MLP模型在时间序列预测方面的性能差异，实验结果表明，短时间内MLP可以获得更高的预测准确度。文献[45]使用MLP预测了河流流量，文献[46]则使用MLP预测了金融领域的数据。

（5）循环神经网络

循环神经网络（Recurrent Neural Network, RNN）[48]是以序列为输入，针对序列数据进行建模的深度神经网络。基础的神经网络只在层与层之间建立了全连接，RNN最大的不同之处在于层之间的神经元也建立了全连接。RNN神经网络的结构图如下：



图 RNN结构图

相比起基于统计和基于机器学习的模型，神经网络拥有强大的拟合能力，尤其是针对非线性关系的序列数据，文献[49]证明一般的神经网络可以任意逼近任何非线性函数。当前主流的神经网络都是基于梯度反向传播算法进行训练，可以实现端到端的训练过程，且可以增量式更新模型权重，有利于模型应对多变的环境。神经网络具有多种连接形式，以RNN为代表的模型可以针对序列数据的前后依赖关系建模，可以更好地处理时间序列相关性的问题。

2.2 基于历史观看轨迹的视口预测算法

根据上述介绍可以看出，针对视口数据这种呈现非线性关系的时间序列进行预测，使用深度神经网络更能取得较好的预测准确度。目前的研究文献中主要提出了基于线性回归和神经网络模型的预测算法预测准确度随着预测时间的增长下降明显，无法做到长期预测。

为解决上述问题，本文提出了一种基于历史观看轨迹的视口预测算法，首先使用LSTM模型（Long-Short Term Memory, 长短期记忆）分别预测当前用户视口中心位置的经度和纬度坐标，然后结合其他用户观看同一个视频时的观看轨迹，计算出最终视口区域所落在的tile位置。

2.2.1 数据集介绍

视点数据集的合理选择是预测算法成功的重要前提。由于视频内容、用户特性对用户的观看轨迹影响很大，因此合理的数据集应该包含各种类型的全景视频，观看视频的用户也应该分散到各个群体。

我们分析了现有公开的几个视点数据集[52][53][54]，最终使用wu等人采集的全景视频用户头部移动数据集[54]。该数据集为48个用户观看18个全景视频的头部移动数据，48名用户包括24名女性和24名男性，覆盖了各个年龄范围，18个全景视频则包含了演出、体育、电影、脱口秀和纪录片5种最流行的类别。

18个全景视频的时长从2分44秒到10分55秒不等，帧率为30fps，而头部移动采集设备每秒可以采集60~80个数据，且每两个采样点之间的时间间隔有轻微的随机性，为了后续数据分析的便捷性，本文使用线性插值法将原始数据下采样为每秒30个数据，与视频的帧率保持一致。最后我们将原始数据转换为经度和纬度值，其中经度的取值范围为0~360度，纬度的取值范围为0~90度。

2.2.2 LSTM介绍

由于后续的视口预测算法是基于LSTM模型的，本节先介绍一下LSTM的相关理论知识。

RNN在捕捉数据在长时间范围内的关系方面存在着明显的缺陷。使用递归结构，隐藏层到隐藏层到转换矩阵（公式 ，在介绍RNN时给出）被多次使用。例如，为了基于第一次输入来更新输入权重U，误差值需要通过整个路径向后传播到第一个时间步长。如果W的最大特征值大于1，则会造成梯度爆炸问题，而如果W梯度到最大特征值小于1，则会造成梯度消失问题，在训练中起不到任何实际作用。梯度爆炸的问题可以通过梯度截断来解决，但梯度消失的问题很难解决。

为了对更长时间的依赖关系进行建模以及解决RNN梯度消失的问题，长短期记忆（Long Short -Term Memory, LSTM）[50]被提出。LSTM引入了门函数来控制信息流。与传统的RNN相比，LSTM的关键特征是使用ct表示的存储单元，它用来连接时间序列并且充当累加器。该单元可以通过几个参数化的控制门（输入门、忘记门和输出门）来进行访问、写入和清除。输入门控制当前输入xt和最后一个隐藏状态ht1。如果忘记门ft开启，存储在单元中的信息也可以被丢

弃，即忘记。最后，LSTM的输出是由输出门ot控制并通过tanh函数驱动的的，输出值介于-1和1之间。LSTM的基本单元结构表达式如下：

公式 LSTM表达式

2.2.3 视口预测模型

由前面的分析可知，当视点位置确定时，视口区域也可以确定，因此，本节聚焦于视点的预测，根据历史的经纬度数据来预测未来一段时间内的经纬度数据。

假设和分别为t时刻的经度和维度值，和为用户初始的经纬度，和分别为到时刻之间的经纬度值，即和。在t时刻的时候，用户的头显设备记录下来从初始值到t时刻的全部的经纬度值（），视点预测即使用历史窗口中的若干经纬度值来预测接下来个时刻的经纬度值，记为（）,其中为预测窗口，然后移动历史窗口继续预测，直至视频观看结束。文献[]认为，视点的几个特征维度之间的自相关性大于互相关性，因此本文也采用独立预测经纬度的方式。考虑到经纬度序列的时间特性和非线性特性，我们引入LSTM模型进行长期时间的预测，历史窗口的数据即为模型的输入，而模型的输出则为我们想要的经纬度的预测值，模型结构图如下所示：



图 基于LSTM的预测模型结构图

从图中可以看出，在t时刻可以根据历史的一段数据预测出下一个时刻的数据，表示为，其中为训练好的LSTM模型的输入输出函数；为了继续预测接下来的数据值，我们将预测值做为额外的输入值，组成新的历史数据，则，依次类推，直至预测出需要长度的数据值。

可以看出，随着预测时间的增加，预测误差将会逐渐增大，为了进一步提高预测准确率，我们参考基于内容显著性检测的视点预测方法，这种方法根据视频内容的显著性来辅助预测，由于视频各个区域内容的显著性不同，人们在某个时刻观看某个区域的概率也不相同，当该区域视频内容较为显著性时，用户观看的可能性也较大，所以，多个用户的观看轨迹可以从侧面反映视频内容的显著性。为了验证这一假设，我们从数据集中随机抽取了2个视频文件，绘制了48名用户观看这两个视频的轨迹散点图，如图 所示，其中蓝色散点为用户的观看位置，红色散点为平均值。





(a) 观看视频1的轨迹图





（b）观看视频2的轨迹图

图 48名用户观看轨迹图

从图（a）中可以看出，视频开始时，由于用户对全景视频体验感到新奇，会随意转动头部以探索更多的画面，而当完全沉浸在视频内容中后，几乎所有的用户都保持着相同的观看轨迹；图（b）显示的用户观看轨迹相对来说杂乱些，说明视频2存在多个感兴趣区域，但多数用户也倾向观看同一位置。因此，我们认为，对于不同的全景视频，用户的观看行为都存在一定的相似性。

基于上述分析，使用LSTM模型初步预测出当前用户的视口数据后，我们在假设视频服务器存储了观看同一视频的其他用户的视口数据的前提下，结合其他用户的观看行为，对初步预测出来的数据进行调整，以提高预测准确度。

为了衡量其他用户历史数据的参考重要性，我们引入基于tile的传输机制，将视频划分为16\*8块tile，记第i行j列的tile的观看次数为，初始值设为0。首先，我们根据当前用户的历史数据分别预测出接下来的经纬度值，确定视口区域以及视口所覆盖的tile块，这些区域每块tile的观看次数加，然后根据其他用户在同一时刻的视口区域位置，将覆盖的tile的观看次数依次加，最后，每个tile的观看次数即为两者之和，计算公式如下：

其中，取值0或1，表示根据LSTM模型预测出来的区域是否覆盖该tile，n为视口区域覆盖该tile的其他用户的个数。和取值的比例关系代表了其他用户数据的参考性，例如，即我们认为最终预测的视口区域由LSTM模型预测出来的视口区域和其他用户的视口区域以相同的重要性共同确定。考虑到LSTM预测出来的视口数据的准确度会随着预测时间的增加而降低，的取值也应该随着的变化而变化，为了减小后续计算的复杂度，我们定义和的计算公式如下：

其中，N为其他用户的总个数，为系数，可以根据实验结果对的取值进行调整以提高预测准确度，为向下取整函数。



### 图 tile观看次数示意图

图 为一帧视频的观看次数示意图，假设视频帧划分为16\*8块tile，则视口区域占6\*4块tile，。计算出所有tile的观看次数后，我们可以获得一个二维数组，然后计算出6\*4块tile的和最大区域即为最终预测的视口区域所占tile，如图中红色区域所示。

2.3 实验与结果分析

本节介绍视口预测算法的实验过程和结果分析，并且介绍衡量预测算法准确度的两个评价指标，最后对比基于LSTM模型的预测算法与其他基准算法在不同评价指标下的预测准确度。

2.3.1 实验过程

根据之前的介绍，我们对经纬度坐标值分别预测，两者的预测过程一样，以纬度预测为例，主要包含以下几步：

（1）数据处理

我们首先对原始的纬度序列进行归一化处理，归一化可以将数据分布映射到一个确定的区间上，提高模型的收敛速度。我们使用线性函数转换方法将纬度特征映射到0-1的区间，线性函数转化方法可以在python的sklearn包中调用preproccessing库中的MinMaxScaler类实现。

归一化处理后，我们需要将时间序列转化为监督学习数据。为了验证我们的算法在长时间预测方面的准确度，使用历史1s的30个纬度数据来预测接下来1~6s的纬度，即我们将历史窗口pw的数据做为监督学习中的输入x，将预测窗口hw的数据作为监督学习中的输出y，并通过移动窗口的方式构造出n组样本数据，然后将数据格式转化成LSTM模型期望的3维格式[n组，时间步长30，特征纬度1]，为了有足够的数据进行学习，我们将观看同一视频的所有用户的数据拼接起来作为我们最终的数据集，采用交叉验证的方式，取钱80%的数据作为训练集，后20%的数据作为测试集。

（2）生成LSTM模型

我们调用Keras库定义了由2层LSTM共128个神经元构成的网络结构，然后使用全连接层Dense用于输出。采用均方误差MSE作为损失函数，ADAM[58]作为优化函数，比起传统的优化算法SGD,ADAM有着更快的收敛速度和更低的错误率，relu作为激活函数。设定好损失函数和优化函数后，调用compile()编译模型，编译好模型后调用fit()来训练模型，该网络模型采用参数分享的反向传播算法进行训练。

在模型的训练过程中，因为训练数据量巨大，没办法将全部数据一次性输入模型进行计算，因此需要设定不同的周期（epoch）和批次（batch）。当完整的数据集通过神经网络训练并输出一次，这个过程称为一个 epoch。当一个epoch的数据过于庞大时，就需要将数据分为多个小块，称为batch，然后将分块后的数据一块一块地传递给计算机进行拟合。而在神经网络的训练过程中，指传递一次完整的数据集是完全不够的，我们需要将有限的数据集多次传递给神经网络，多次更新权重值，使得拟合曲线从欠拟合到过拟合。本节最终设置epoch为50，batch为30。

（3）预测数据

当模型训练完毕后，便可调用predict()输入测试数据进行预测了，预测出来的数据区间也在0~1之间，使用MinMaxScaler类再将数据转换到原始区间，即可得到预测的纬度值。我们将预测的纬度值跟同一时刻真实的纬度值对应存储为csv文件，用于后续准确度的计算。

通过上述几步分别预测出经度和纬度值后，可以计算出视口所在位置以及对应的tile块。然后我们将同一视口其他用户的经纬度数据也按照历史窗口和预测窗口的大小划分成若干组，与预测数据的时间对应，并计算出每个用户视口所在tile，按照公式 更新tile数组。最终我们获得一个16\*8的数组，以6\*4的窗口去遍历该数组，计算出窗口中所有tile值的总和，总和最大的窗口即最终预测出来的视口区域。如果存在多个最大值，我们认为这种情况是无法区分哪个最大值更准确，因此就以最后一个最大值作为输出。

2.3.1 评价指标

为了对比我们提出的算法与其他基准算法的预测准确度，本节使用两种评价指标，经纬度绝对差值和tile重叠比。

（1）经纬度绝对差值

为了衡量LSTM模型的预测准确度，我们使用同一时刻真实的经纬度与预测的经纬度的绝对差值来代表预测误差，由于全景视频的特性，到在经度上，-150度和150度直接取差值为300度，但实际两者只差了60度，因此，我们使用文献[]中定义的角度差值的计算方式，公式如下：

其中，代表角度和的差值，为取模运算，基于此，我们定义预测时刻经纬度的绝对差值如下：

（2）tile重叠比

由于我们提出的预测算法最终预测的是视口区域所占tile的情况，所以我们使用同一时刻预测的视口所占tile在真实视口所在tile中的重叠比做为最终的预测准确度指标。

2.3.2 实验与结果分析

为了对比预测准确度，我们研究了两种回归方法，线性回归-LR和非线性回归-MLP，将时间和经纬度值视为独立变量，目标是根据历史窗口的数据拟合出回归函数，然后根据预测窗口得出预测数据，我们对经纬度分别建模，对于MLP，我们采用了一个简单的结构，隐藏层含有3个神经元，同样使用relu作为激活函数，ADME作为优化函数。最后，我们使用移动平均算法作为基准算法，即使用历史窗口中的数据的平均值作为整个预测窗口中的预测值。

2.4 本章小结

全景视频的视口预测是全景视频传输优化问题的重要前提，优秀的预测算法可以显著降低全景视频的传输数据量。本节针对现阶段视口预测算法长时间预测准确度明显下降的问题，提出了一种基于历史观看轨迹的预测算法。首先，该算法根据当前用户的历史观看轨迹，使用LSTM模型对轨迹的经纬度分别预测，得到初步预测结构。为了进一步提高预测准确度，在视频服务器存储了观看同一视频的其他用户的轨迹数据的情况下，我们计算其他用户在全景帧每块tile的观看次数，再结合LSTM模型的输出结果，综合计算出预测窗口所覆盖的tile块。实验结果表明，基于当前用户和其他用户历史观看轨迹的预测算法比起其他常用的预测算法有着较高的准确度。

# 第三章 基于Q-Learning的LTE资源调度算法

本章聚焦于LTE网络中下线链路的资源调度问题，首先介绍了LTE网络架构与关键技术，介绍了无线资源的定义与调度过程，然后详细介绍了经典的资源调度算法，最后针对全景视频的时延敏感性特征，提出了基于Q-Learning的资源调度算法，并通过仿真平台对比了提出的算法与经典算法的性能优劣。

3.1 LTE架构与资源调度

本节主要介绍了LTE网络架构与协议架构，详细阐述了LTE网络中的关键技术，然后介绍了无线资源的帧结构与资源块概念，最后介绍了资源调度过程。

3.1.1 LTE网络架构与协议架构

LTE（Long Term Evolution, 长期演进）是第三代通信技术（3G）的演进版本，2004年由3GPP组织（The 3rd Generation Partnership Project，第三代合作伙伴计划）制定。严格意义上来说，LTE并不是4G网络，而只是3G到4G技术之间的平稳过渡。相较于3G网络，LTE主要是为了实现如下几个目标[34]：

(1) 提高数据传输速率、频谱效率。在20MHz频率带宽下可使下行链路的峰值速率为100Mbps，上行50 Mbps；

(2) 在现有小区规模范围基本不变动的基础上，提高小区边缘的比特速率；

(3) 减小系统时延，保证用户单程传输时延小于5ms；

(4) 确保QoS要求，当小区范围在5km以内时，应当满足所有用户的QoS要求。

为了实现上述目标，LTE网络采用了一种扁平化的架构，即只由EPC（Evolved Packet Core, 演进核心网）、E-UTRAN（Evolved Universal Terrestrial Radio Access Network , 演进通用地面无线接入网）和UE（User Equipment, 用户设备）。这种架构一方面可以降低基础设备的数量，另一方面也可以降低系统的传输时延。LTE的网络架构图如下：



### 图3-1 LTE网络架构图

EPC由MME（Mobility Management Entity, 移动管理实体）、SGW（Serving Gateway, 服务网关）和PGW（Packet Data Network Gateway, 分组数据网关）。MME主要负责控制面功能，如非接入层信令的加密、完整性保护和安全控制，并对空闲状态下的移动台进行移动性管理；SGW负责用户面功能，如终止用户面数据包及用户面切换；PGW则是终结和外贸部数据网络的接口，是3GPP与非3GPP网络间的用户面数据链路层的锚点。相比于3G网络，E-UTRAN仅有基站（eNodeB）组成，取消了RNC（Radio Network Controller, 无线网络控制器），eNodeB具有3GPP的NodeB功能和大部分RNC在接入层的功能。

接口定义了网络中不同单元之间进行数据沟通的方式，接口协议的架构被称为协议栈。在LTE中真正实现了控制和承载相分离，控制信令通过MME进行交互，而业务则通过SGW与eNodeB进行交互。LTE中的协议栈根据功能可以分为控制平面协议栈和用户平面协议栈。控制平面主要负责对无线接口的管理和控制，用户平面主要完成数据头压缩、加密和调度等功能。下图为LTE系统协议栈示意图：



### (a) 用户平面 (b) 控制平面

### 图3-2 协议栈

用户平面由PHY层（Physical Layer, 物理层）、MAC层（Media Access Control, 媒体接入控制层）、RLC层（Radio Link Control, 无线链路控制）和PDCP层 （Packet Data Convergence Protocol, 分组数据汇聚协议层）组成，PHY负责处理编译码、调制解调、多天线映射以及其他电信物理层功能层，通过传输信道为MAC层提高数据传输功能；MAC层负责信道和传输信道之间的映射，调度信息的报告，基于HARQ（Hybrid Automatic Repeat Request，混合式自动重传请求）进行错误纠正，通过动态调度的方式，处理不同用户的优先级；RLC层主要负责分段重组数据，通过ARQ（Automatic Repeat Request，自动重传请求）进行纠错，并进行重复检测；而控制平面则在用户平面协议栈上增加了RRC层（Radio Resource Control, 无线资源控制层）和NAS（Non-Access Stratum, 非接入层）。RRC层主要负责对接入层的控制和管理，完成广播、寻呼、RRC连接管理、移动性管理、UE测量报告和控制功能；NAS层主要负责提供对非接入层部分的控制和管理，主要包括移动性管理、呼叫控制和安全控制等。

3.1.2 LTE架构关键技术

LTE架构中主要的关键技术如下：

（1）正交频分复用技术

OFDM（Orthogonal Frequency Division multiplexing, 正交频分复用）技术是高速率无线通信系统中有广阔应用前景的多载波数据通信技术，它的主要思想是：在频域内将信道分成若干互相正交的子信道，相当于把串行的高速率的数据信号分割成并行的低速率的子数据流，用这样的低比特率形成的多状态符号去调制互相正交的子载波，传输到接收端后，互相正交的信号可以通过技术手段分开，可以减少子信道间的干扰[35]，OFDM的频谱示意图如图3-3所示。



### (a) 传统的多载波技术



### (b) OFDM技术

### 图3-3 OFDM频谱图

在以往频分复用的系统中，为了消除子信道之间的干扰，需要在相邻的两个子信道之间增加更大的保护频带，而正交的技术允许没有保护频带，甚至相互覆盖仍不会造成干扰，比起传统的系统极大提高了带宽利用率；由于多个正交的子载波信号传输信息，比单载波信号的传输时间消耗更长，这使得在信道衰落和脉冲噪声的环境下 OFDM 可以更好地对抗衰落。由于 OFDM 在发送端用多个正交子载波实现信号发射，并且不同的子载波相互正交，拥有不一样的衰落概率。实现了频率分级，强化了系统抗频率选择性衰落的能力。另一方面OFDM使用快速傅立叶反变化（IFFT）和快速傅立叶变化（FFT）来实现调制和解调，运算量小，实现简单；OFDM也可以与多种接入方式组合使用，系统设计更加灵活高效。

虽然OFDM有着很多优点，也仍存在一些缺点，主要缺点有：

（a）对频偏和相位噪声比较敏感。OFDM技术区分各个子信道的方法是利用各个子载波之间严格的正交性。频偏和相位噪声会使各个子载波之间的正交特性恶化，仅仅1％的频偏就会使信噪比下降30dB。因此，OFDM系统对频偏和相位噪声比较敏感。

　　（b）功率峰值与均值比（PAPR）大，导致射频放大器的功率效率较低。与单载波系统相比，由于OFDM信号是由多个独立的经过调制的子载波信号相加而成的，这样的合成信号就有可能产生比较大的峰值功率，也就会带来较大的峰值均值功率比，简称峰均值比。对于包含N个子信道的OFDM系统来说，当N个子信道都以相同的相位求和时，所得到的峰值功率就是均值功率的N倍。当然这是一种非常极端的情况，通常OFDM系统内的峰均值不会达到这样高的程度。高峰均值比会增大对射频放大器的要求，导致射频信号放大器的功率效率降低。

（c）负载算法和自适应调制技术会增加系统复杂度。负载算法和自适应调制技术的使用会增加发射机和接收机的复杂度，并且当终端移动速度每小时高于30公里时，自适应调制技术就不是很适合了。

（2）多天线技术

多天线技术（Multiple Input Multiple Output, MIMO）是LTE大幅提升吞吐率的物理层关键技术，MIMO技术与OFDM技术一起并称为LTE两大最重要的物理层技术。MIMO是指在发射端和接收端分别使用多个发射天线和接收天线，它与传统的信号处理方式的不同之处在于其同时从时间和空间两个方面研究信号的处理问题，从而能够在不增加带宽与发射功率的前提下，提高系统的数据速率、减少误比特率、改善无线信号的传送质量。MIMO系统架构图如图3-4所示：



### 图3-4 MIMO系统框架图

可以看出，随着天线数量的增加，信道容量也呈线性增加，因此可以利用MIMO信道倍数增大无线信道的容量，在不增加带宽和天线发射功率的情况下，频谱利用率也可以倍数增大。MIMO有3类效果：a)空间分集，又称为发射分集或者传输分集，指利用较大间距的天线阵元之间或波束赋形之间的不相关性，发射或接手一个数据流，避免单个信道衰落对整个链路的影响；b)空间复用，指利用较大间距的天线阵元之间或波束赋形之间的不相关性，向一个终端或基站并行发射多个数据流，以提高链路容量，即峰值速率；c)波束赋形，指利用较小间距的天线阵元之间的相关性，通过阵元发射的波之间形成干涉，集中能量于某个或某些特定方向上，形成波束，从而实现更大的覆盖和干扰抑制的效果。

（3）高阶调制技术

调制是对信号源的编码信息进行处理，把要传输的模拟信号或数字信号变换成适合信道传输的信号的过程。高阶的调制方式可以提高带宽利用率，在信噪比比较好的情况下，比如基站附近的用户，或者小区中用户数量很少，可以通过增加调制星座图中的星座点密度使得每个调制符号传递更多的信息比特。LTE系统中制定了多种调制方式，主要采用QPSK、16QAM和64QAM三种调制方式。QPSK的调制星座图中有4个星座点，理论速率为2bit/符号，16QAM的调制星座图有16个星座点，理论速率是QPSK的2倍，而64QAM的星座图则扩展为64个点，理论速度是16QAM的1.5倍。

（4）链路自适应技术

链路自适应技术指随着信道质量的变化，动态地选择不同的调制方式、发射功率和编码速率，使得链路的传输速率最大化。链路自适应技术主要包含两方面：a)自适应调制和编码（Adaptive Modulation Coding, AMC）,指保证发射功率恒定的情况下，通过调整无线链路传输的调制方式和编码速率，确保链路的传输质量。小区内的用户测量信道质量的信息反馈（Channel Quality Indicator, CQI），每1ms或者更长的周期汇报给基站，基站根据CQI从预定义的CQI表格中选择具体的调制和编码方式；b)功率控制，即发射功率的自适应调整，维持接收端一定的信噪比，从而保证链路的传输质量，功率控制可以很好的避免小区内用户间的干扰；

3.1.3 无线资源调度

3.1.3.1 无线资源概述

无线资源分为时域和频域两个维度，在时域上为无线帧的形式。LTE支持两种双工模式，时分双工（Time Division Duplexing, TDD）和频分双工（Frequency Division Duplexing, FDD），因此定义了两种帧结构：TDD帧结构和FDD帧结构。

TDD帧结构如图 所示，每个无线帧也是10ms，等分为两个5ms时长的半帧，每个半帧由4个数据子帧和1个特殊帧组成。数据子帧有两个时长0.5ms的时隙组成，时长1ms的特殊帧则由三个特殊时隙构成。



### 图3-5 TDD帧结构

FDD帧结构如图 所示，每个无线帧时长为10ms，等分为10个子帧，每个子帧长1ms，一个子帧又可以分为2个时隙，每个时隙长0.5ms，因此一个无线帧由20个时隙组成，上、下行的传输会在不同的频率上同时进行



### 图3-6 FDD帧结构示意图

在频域上，无线资源则是由12个连续的子载波为一组构成的，每个子载波的带宽为15kHz。因此，我们定义在时域上占一个时隙（0.5ms），在频域上占12个子载波（180kHz）的资源为一个物理资源块（Physical Resource Block, PRB）， 如图所示。资源块是资源调度中可分配等最小单元。



### 图3-7 资源块结构图

系统带宽不同，物理资源块的数量也不同。在LTE系统中可以通过设置不同的子载波数量来实现对多种系统带宽的支持。系统带宽、RB数量和子载波之间的对应关系如表所示

表 系统带宽、子载波数量和RB数量关系

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 系统带宽（MHz） | 1.4 | 3 | 5 | 10 | 15 |
| 子载波数量 | 72 | 180 | 300 | 600 | 900 |
| RB数量 | 6 | 15 | 25 | 50 | 75 |

3.1.3.2资源调度流程

LTE系统中上下行链路的调度过程是分开的，本文重点探讨下行链路的调度过程。当小区内的多个用户请求不同业务时，基站处的缓冲池就保存着等待传输的不同应用的任务队列，调度器也存在于基站，可以获取到任务队列的长度和用户的信道质量参数（Channel Quality Index, CQI）。总结来说，资源调度就是决策当前时刻给哪些待调度的任务队列分配哪些RB，主要分为两个流程，即计算用户优先级和分配资源块。

（1）计算用户优先级

优先级计算定义的是第i个数据流在第j个资源块上的调度优先顺序。不同的调度算法计算优先级的公式不同，以此来达到各自的分配目标。例如轮询算法，每个数据流的优先级相同，资源就被依次分配。

（2）分配资源块

计算出优先级顺序后则要进行资源块的分配。假设系统带宽有4个资源块，记为RB1-RB4，有3个待调度的用户，记为user1-user3，根据调度算法计算出优先级结果如表 所示：

表 优先级矩阵

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | RB1 | RB2 | RB3 | RB4 |
| user1 | 15 | 8 | 10 | 8 |
| user2 | 11 | 12 | 14 | 13 |
| user3 | 9 | 4 | 7 | 11 |

我们假设每个用户分配到两个资源块后即可达到传输需要。首先调度器开始遍历资源块RB1上的各个用户的优先级大小，发现user1的优先级最高，就将RB1分配给user1，然后在遍历RB2，发现user2的优先级最高，将RB2分配给user3，然后再遍历RB3，发现user2的优先级最高，将RB3分配给user2，最后遍历RB4，发现user2的优先级依旧最高，但是此时user2已经分配到两个资源块了，无需再分配，调度器就会比较剩下的用户user1和user3的优先级，发现user3的优先级较高，因此将RB4分配给user3。最后各个资源块的分配结果如下表所示：

表 资源块分配结果

|  |  |
| --- | --- |
| user1 | RB1 |
| user2 | RB2 RB3 |
| user3 | RB4 |

调度器就按照上述步骤依次遍历用户和资源块，直至所有资源块都被分配掉或者系统中等待调度的用户均已满足传输需求。

3.2 基于Q-Learning的资源调度算法

本节首先介绍了常见的四种资源调度算法的原理以及优缺点，在这些研究的基础上提出一种改进算法，详细介绍了改进算法的设计原理和框架。

3.2.1 经典的资源调度算法

当有限的带宽资源需要尽可能满足小区内不同用户的需求时，如何有效地采取措施进行分配调度至关重要。通常需要考虑用户公平性、系统吞吐量、系统丢包率和分组时延，这通常也是衡量一个调度算法好坏的指标，接下来从这几个角度来介绍四个经典的调度算法，这些算法对后续改进算法的提出奠定了基础。

（1）PF算法

最初研究人员提出的轮询算法RR和最大载干比Max C/I算法都只考虑单个方面的性能最优，应用到实际网络中存在着一定问题，因此，为了在系统吞吐量和用户公平性之间取得一种平衡，比例公平算法PF被研究人员提出。该算法根据用户的优先顺序进行调度，而在定义用户优先级上，则同时考虑了用户的历史吞吐量和信道质量两个因素，优先级公式定义如下：

其中，表示第个用户在第个资源块上的优先级；表示将第个资源块分配给用户时的即时传输速率，可以通过用户的CQI反馈获取到，的值越大，表明用户的信道质量越好；则表示用户的历史平均传输速率，在仿真中，该值的计算一般遵循以下公式：

从公式中可以清楚的看出，如果刚开始某个用户的信道质量比较良好，瞬时速率较高，优先级则较大，可以优先调度分配到资源，但随着资源的逐渐积累，该用户的历史平均速率也会升高，用户优先级随之下降，系统则会有可能将资源分配给边缘用户。

因为PF算法在系统吞吐量和用户公平性之间取得了平衡，一直是最为流行的调度算法。但是该算法没有考虑到各个业务对服务质量的需求不同，尤其是实时业务对时延的敏感性，所以需要更加适合实时业务的调度策略。

（2）EDF算法

为了解决实时业务的调度问题，最早截止先EDF算法被提出。该算法给予即将到达时延阈值被丢弃的业务最高优先级，给出最简单的计算公式如下：

为第个用户业务流的时延阈值，超过该阈值的数据包将会被丢弃，为用户的缓存队列首个数据包的等待时间。可以看出，越临近时延阈值的用户优先级越高。

（3）MLWDF算法

应对上述问题，改进的最大权重延迟优先算法MLWDF被提出。对于非实时业务，该算法计算优先级的公式等同于PF算法，而对于实时业务，该算法在PF算法计算优先级的公式的基础之上，引进了服务质量参数。优先级公式定义如下：

为队头时延超过时延阈值的最大概率，在仿真过程中一般设置为常数。

（4）EXP/PF算法

EXP/PF算法是对M-LWDF算法的一种改进，也同样适用于非实时业务和实时业务。该算法引入了缓冲时间来保证对实时业务对优先调度。对于实时业务，优先级公式定义如下：

其中的参数含义等同于公式 ，X计算公式如下：

其中，表示系统中等待调度的实时用户数。对于非实时业务，该算法的优先级公式与PF算法相同。

3.2.2 DSQL算法介绍

上述这些经典算法虽然被广泛使用，但在整个调度过程中始终使用单一的调度策略，这导致这些算法不能很好地适应不断变化的网络情况。我们希望可以根据不同的网络状态来选择最佳的调度策略以达到性能最优化。

近年来开始出现一些基于增强学习进行优化的研究。对调度而言，增强学习是一个很好的切入点，因其可以较好地学习动态变化的网络状态然后做出更高效的策略，Q-Learning算法就是这方面的翘楚，因此我们提出一种DSQL算法（Downlink Scheduler based on Q-Learning, 基于Q学习的下行调度算法）。

首先介绍Q学习的理论知识。

Q学习（Q-Learning）是增强学习领域最广为人知的算法之一。Q学习与模型状态无关，可以在环境未知的情况下，根据“状态-动作”获得环境的奖励，然后根据奖励不断调整动作使得奖励最大化，最后学习到最佳动作。可以看出，Q学习主要包含三个要素：（1）状态空间（2）动作空间（3）环境的奖励。

我们将学习的主体称为智能体（Agent），Agent在每一个的迭代中，根据当前的状态从动作空间中选择一个动作，然后获得环境的即刻奖励，然后环境的状态转移到，继续迭代，记为多次迭代后的累计回报，使用一个的二维表来记录该值，称为Q表（Q-table），在每次迭代中根据公式 来不断更新Q表的值。

其中，为学习率，范围在0~1之间，代表之前训练结果的影响权重，学习率越大，之前训练结果的影响越小；则为折扣因子，范围在0~1之间，代表了未来奖励的重要性，如果=0，只有当前的奖励会被考虑。代表了下一个状态时Agent可以获得的最大的累计奖励。经过多次更新后，Q值收敛到最优值。

从上述分析可以看出，将Q学习应用到资源调度场景中，负责调度过程的基站就充当Agent的角色，动态变化的无线网络为环境的状态，动作则是不同的调度算法，我们的目标旨在每一个TTI（Transmission Time Interval，传输时间间隔，LTE系统中为1ms）内选择最合适的调度算法来达到系统的性能最大化，系统吞吐量等性能参数即为奖励，基于Q学习的调度过程如下图所示：



### 图3-8 基于Q学习的调度过程

首先，基站从环境中获取到信息来判断当前所处状态，然后从定义好的若干个调度算法中选择一个，计算用户优先级完成资源分配后，环境状态从变化到，并且可以此次调度过程的即刻奖励，按照公式 更新Q表的值并开始下一次的迭代，直至调度完成。

根据上述分析，基于Q学习的调度算法重要的是需要定义状态空间、决策空间和奖励函数，接下来将详细介绍这几个关键点。

（1）状态空间

状态空间的划分是Agent可以学习到动作策略的基础，合理的状态应该满足以下两点要求：一是状态可以被Agent感知和处理，二是后一个状态只能依赖于前一个状态和选择的决策。一次资源调度后，系统中用户需要的数据量和等待时间都会发生变化，同时考虑到我们算法的目的主要是为了满足实时业务的QoS需求，例如系统吞吐量和数据包时延，定义状态空间为<*dataIndex, delayIndex*>，其中：

为等待调度的所有用户的数据量总和，为系统可传输的最大的数据量，表征了无线资源的短缺程度；为等待调度的用户数，表征了待调度应用流的时间紧迫度。当和都比较小时，说明更多的用户快要到了时间阈值且系统的资源较为充足，此时应该给时间紧迫的用户更高的优先级；而当和都比较大时，应该给信道质量较好的用户更高的优先级。

如果直接使用上述公式来定义状态的话，由于实际的网络情况比较随机，状态空间的数量将是巨大的。为了降低计算的复杂度提高收敛速度，我们首先统计了测试环境下和的数值范围和分布概率，然后将状态划分为16个区间，落在同一个区间的状态被赋予相同的值。

（2）动作空间

通过前面的介绍，显而易见的，决策空间应该被定义为不同的资源调度算法。为了保证QoS需求，计算方式需要综合考虑信道质量、用户公平性和实时业务的时延需求。因此，基于MLWDF算法，定义优先级的一般计算公式如下：

其中参数的相关含义与公式 相同。

考虑到系统吞吐量、用户公平性和业务时延的权重关系，我们选择其中特定4种情况构成动作空间，如下表所示。

表 决策空间

|  |  |  |  |
| --- | --- | --- | --- |
| 动作 | θ | β | ω |
| A1 | 1 | 1 | 1 |
| A2 | 1 | 1 | 2 |
| A3 | 1 | 0 | 1 |
| A4 | 2 | 1 | 1 |

每个调度动作可以得到不同的RB矩阵，根据RB矩阵带宽资源依次分配给用户完成一次调度过程。

（3）即刻奖励

即刻奖励是用来评估前一个动作是否合适的重要指标，奖励函数的设计需要基于系统性能参数。我们希望在每个TTI选择某个调度策略后，系统的QoS指标（吞吐量、公平性和时延）可以得到提升，因此，奖励与这三个指标紧密相连，由于这三个指标的数量级不同，需要将取值映射到同一区间，下面依次给出三个指标的计算公式，首先是系统吞吐量指标STI：

其中，N为当前调度流程时系统中待调度的用户数，为时刻用户实际获得的吞吐量，则为当前调度流程可以调度的资源总和，可以看出UTI的取值范围在0~1之间。

我们使用AJFI（Average Jain’s Fairness Index, 平均Jain公平指标）作为公平性指标，定义如下：

其中参数的含义与公式 相同。

我们定义系统时延指标SDI如下：

SDI的值越小，用户的等待时间越短。

由于学习目标是使得奖励最大化，所以即刻奖励应该跟吞吐量和公平性成正比，跟时延成反比，因此，我们定义即刻奖励的计算公式如下：

其中，η、λ和μ为权重值，可以通过设置不同的权重来达到所期望的权衡。因此，这个奖励函数期望系统吞吐量和公平性最大化，同时降低系统时延。

（4）探索与开发

除了上述三个要素外，Q学习算法还需要确定搜索策略。如果每次都是选择Q值最大的动作，这种搜索方式称为开发（exploit），开发模式会导致Q值小的动作一直不被执行，每次只选择当前阶段的最优动作而忽略了全局，探索（explore）则使得系统去尝试更多没有被选择过的策略。考虑到我们算法使用的仿真环境，我们使用一个简单的ε-贪婪算法，按照概率ε随机去选择动作，然后以1-ε的概率根据Q值来选择最优的策略。在仿真实验时，ε初始为1，即最开始完全随机的探索，然后逐渐降低到0.1后保持不变。

最后，我们给出DSQL算法的调度流程，如表 所示：

表 DSQL算法的调度流程

|  |
| --- |
| 1：初始化Q表：，设置权重值γ和α |
| 2：for t=0 to T-1 do |
| 3： 根据公式 从状态空间中选择当前的状态 |
| 4： 根据ε贪婪算法选择一种调度策略 |
| 5： 进行包调度和资源分配 |
| 6： 根据公式 计算奖励 |
| 7： 根据公式 更新Q表 |
| 8：end for |

3.3 实验与结果分析

本节介绍DSQL算法的实验过程与结果分析。

3.3.1 仿真工具介绍

由于LTE系统涉及的技术繁杂，在真实的网络环境中进行实验论证存在着诸多困难，而使用仿真工具可以显著提高实验的可行性，节约实验之间。仿真方法主要分为两种：一种是系统级仿真，一种是链路级仿真。系统级仿真实现了整个网络架构，而链路级仿真主要针对物理层。因此，为了验证DSQL算法的性能，我们需要进行系统级的仿真实验。

目前开源的LTE系统级仿真工具主要由三个：维也纳大学开发的Vienna LTE-A Simmulators，基于Matlab；意大利都灵理工大学开发的LTE-Sim，基于C++，运行在Linux环境下；NS-3，考虑到仿真工具的复杂度和对C++开发语言的熟悉度，我们使用LTE-Sim[39]作为仿真工具完成实验。

LTE-Sim是一个开源的LTE系统级仿真平台，使用C++作为开发语言，因其具有封装与继承的特点，易于扩展开发，近些年来被研究资源调度的学者广泛使用。该平台以事件驱动，定义了90个类函数，共有220个文件大约2万多行代码。

LTE-Sim平台再现了完整的LTE网络结构，包括E-UUTRAN和EPS，支持

支持单/多小区场景、单/多用户环境、QoS管理、多小区切换、频分复用技术、AMC调制等多个场景和技术。LTE-Sim主要包含四个模块：Simulator、NetworkManager、FlowsManager和FrameManager，主要负责开始、处理和结束事件；定义并处理LTE帧和子帧；管理应用流；创建小区节点、用户设备并设置用户位置以及实现小区切换和频分复用。LTE-Sim还实现了三个网络节点：UE、eNodeB和MME/GW，这三个节点是由NetworkManager创建和处理的。

LTE-Sim定义了三种用户移动状态：静止、步行和乘坐交通工具，设置对应的移动速度为0、3、30和120km/h，定义了两种移动模式：一种是用户随机选择一个方向一直走到小区边缘，然后再随机选择一个方向走动；另一种是用户选择一个方向走指定距离后再切换方向走指定距离。我们选择了前一种移动模式进行仿真。

LTE-Sim仿真了四种最常见的应用类：（1）Trace-based，即视频流，主要是根据真实的视频文件获得到的数据包；（2）VoIP，语音流，根据G.729标准构建的on/off马尔科夫链模型；（3）CBR，有固定比特率的数据包，可指定数据包的到达时间；（4）Infinite-buffer，模拟的持续发送的数据包，一般用户模拟背景业务。

针对资源调度算法的仿真问题，LTE-Sim基本实现了资源调用与分配的整个过程，DownlinkPacketScheduler类包含的函数实现了从设定用户流承载到计算优先级再到分配资源的全部功能，其中，与资源调度算法有着最直接关系的是ComputeSchedulingMetric()函数，在该函数里可以编写不同的计算用户优先级的公式，默认实现了PF、MLWDF和EXP/PF算法。由于CQI反馈是用户与基站之间重要的交互过程，LTE-Sim也实现了该功能。

在LTE-Sim中对资源调度过程进行仿真的主要步骤如下：

（1）初始化仿真器、网络管理器、帧管理器和流管理器

（2）创建小区、基站、网关和信道，创建用户并设置位置和移动模式

（3）为用户指定承载的应用流，定义QoS参数和持续时间

（4）指定资源调度算法，分配无线资源

（5）指定仿真运行时间，开始仿真

仿真开始后，用户开始请求应用，基站侧的调度器根据数据包的QoS参数和用户发过来的CQI反馈等信息选择调制编码方式，然后根据资源调度算法计算用户优先级，最后分配无线资源块后完成一次调度过程。在整个仿真运行时间内，基站在每一个TTI执行一次上述过程。

3.3.2 仿真环境与参数

我们模拟了单个小区多用户的环境。小区半径为1km，小区中央有一个基站，用户数量从10递增到40，随机分布在不同位置，同时用户在小区内以3km/h的移动速度选择随机的方向运动。为了防止仿真随机性对结果造成的影响，我们进行多次实验取平均值作为最终的实验结果，仿真环境具体参数见表

表 仿真环境参数设置

|  |  |
| --- | --- |
| 参数 | 参数值 |
| 带宽 | 10MHz |
| 小区半径 | 1Km |
| 频率 | 2GHz |
| 衰落模型 | Urban |
| 复用模式 | FDD |
| 帧长 | 1ms |
| 仿真时长 | 66s |
| 业务流时长 | 55s |
| 用户数 | 10-40 |
| 用户速度 | 3km/h |
| 应用 | Video, VoIP |
| 视频时延 | 20-100ms |
| VoIP时延 | 100ms |
| 学习速率 | 0.05 |
| 折扣因子 | 0.01 |

3.3.3 仿真结果与分析

首先介绍衡量调度算法性能的几种指标。

（1）系统吞吐量

系统吞吐量表征了单位时间内系统传输的数据量总和，在其他条件相同的情况下，系统吞吐量越大表明无线资源的利用率越高。本文实用如下公式来计算系统吞吐量大小：

其中，T为仿真时间内TTI的数量，N为用户数，为用户i在t时刻传输的数据包大小。

（2）用户公平性

追求系统吞吐量最大化可以充分利用带宽资源，但是会导致一些边缘用户一直分配不到资源，为了保证同一小区内的用户得到自由的机会平等，还需要定义用户公平性。公平性的评定方式有多种，本文采用使用AJFI作为公平性指标,计算公式如 所示。可以看出，系统吞吐量跟用户公平性是互相矛盾的。

（3）丢包率

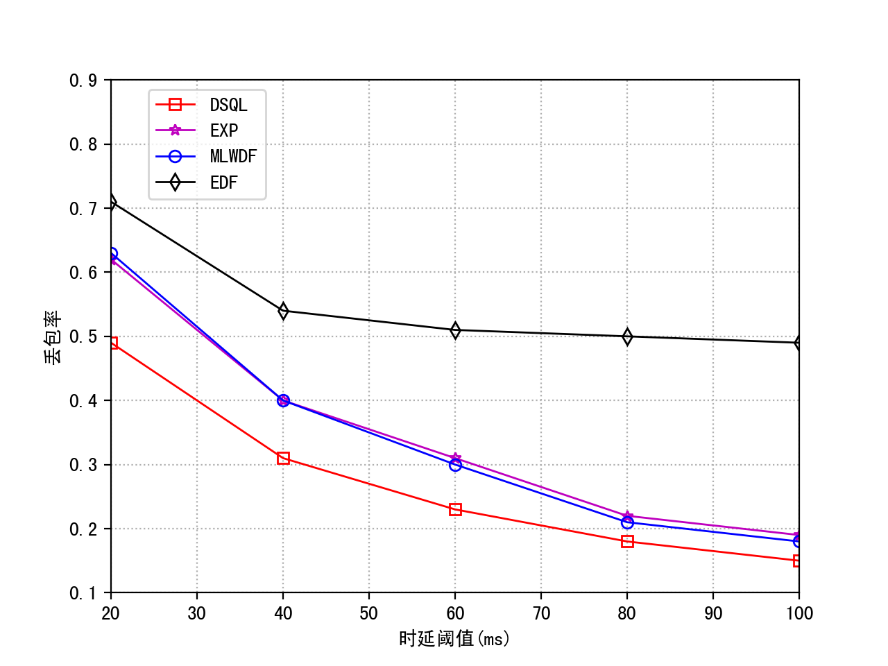
丢包率是丢失数据在总传输数据中的占比，丢包率小可以很好保证传输数据的完整度。丢包率主要取决于数据包的长度和排队时间。丢包率表示如下：

其中，为用户i在时刻t丢弃的数据包大小，则为用户i在t时刻成功到达基站缓存队列的数据包大小。

（4）系统时延

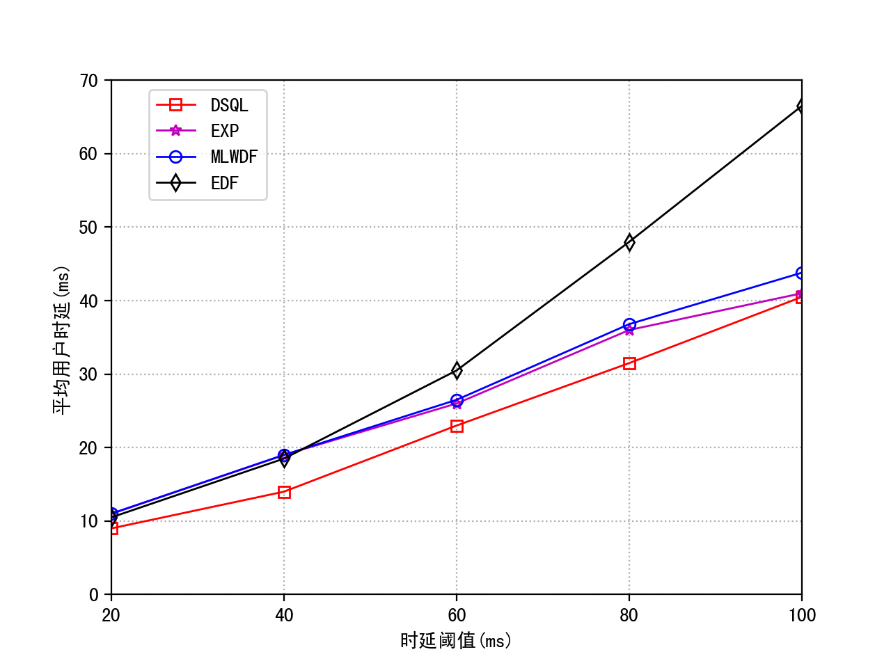
系统时延主要是衡量实时业务的参数，使数据端到端传输所消耗的时间，对于视频这类实时业务，时延超过时延阈值时数据包会被丢弃。定义系统的平均时延如下：

为了强调我们对视频业务时延性能的考虑，首先模拟了小区内有30个用户且每个用户都请求视频业务的场景，并给每个视频业务设置不同的时延阈值，我们选择经典的实时业务资源调度算法EXP、MLWDF和EXP/PF作为对比算法，仿真结果如图3-9~3-11所示。

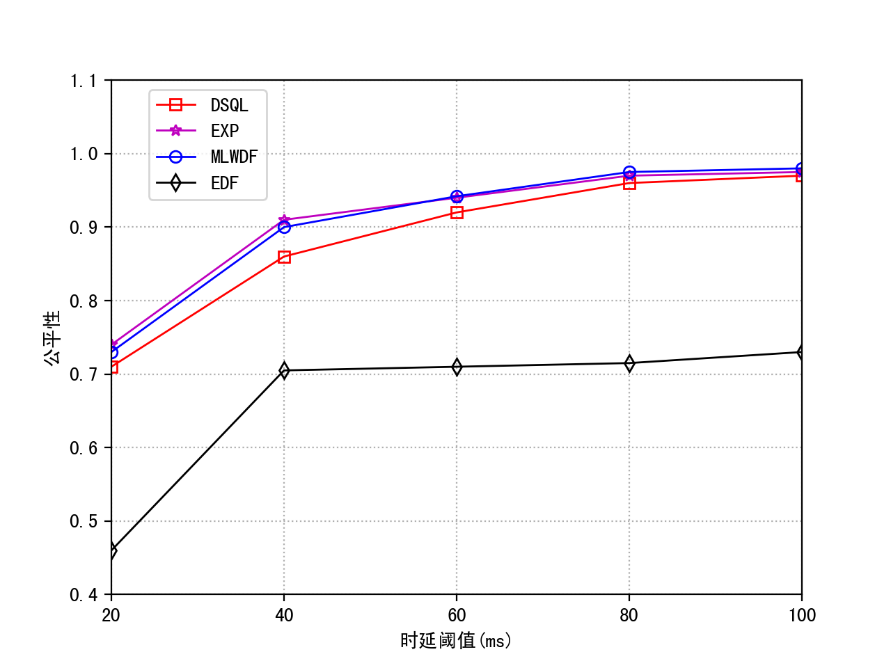


### 图3-9 不同时延阈值时系统丢包率

通常来说，丢包率是评估实时业务调度算法的重要指标之一。如图3-9所示， 显而易见的，随着用户业务时延阈值的增加，所有算法的丢包率都在逐渐降低。EDF算法只根据业务的时间紧迫度计算优先级，理论上会降低因超过时延阈值而被丢弃的数据包，但在实际调度过程中，可能由优先级最高的用户的信道质量很差，导致一次调度过程中用户只能传输少部分数据，只能继续等待下次调度，资源利用率较大导致后面资源急缺，造成大量用户得不到调度被丢弃。MLWDF和EXP均考虑了时延和信道质量参数，且计算优先级公式类似，因此两者的丢包率相似。DSQL算法可以自适应的选择最优的调度策略，因此丢包率最小，尤其在时延阈值效于40ms时显著优于其他三种算法。



### 图3-10 不同时延阈值下平均用户时延



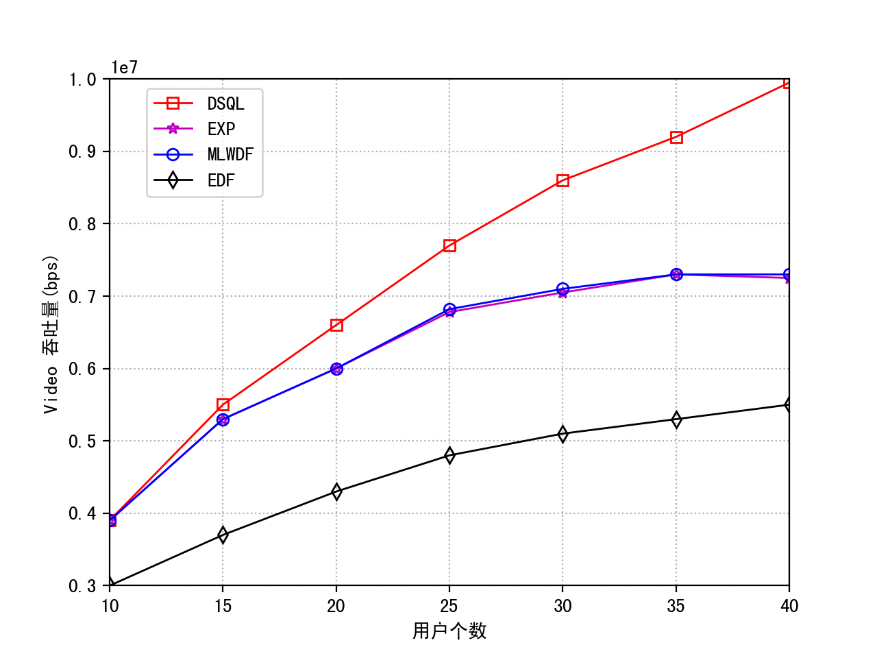
### 图3-11 不同时延阈值下用户公平性

图3-10展示了不同时延阈值下的平均用户时延。在时延阈值小于40ms的时候，可以看出EDF、EXP和MLWDF的平均用户时延基本一致，EDF略低于后两者，但这并不是说明EDF算法在时延方面优于后两者，结合之前的丢包率可以分析出，由于EDF的丢包率较大，许多时延较大的数据被丢弃了，导致参与计算的都是时延较小的用户。由于EXP算法中优先级与时延呈指数关系，对时延的反应更加灵敏，所以它在时延方面的性能要略优于MLWDF算法。与丢包率类似，在用户平均时延上DSQL算法仍保持着最优的性能，这说明变化的网络状态下灵活选择调度算法要优于只使用一种特定的算法。

图3-11展示了各算法在用户公平性方面的性能。理所当然，EDF算法有着较差的公平性，这也说明，通常情况下只考虑一个方面的调度算法的性能比较差。

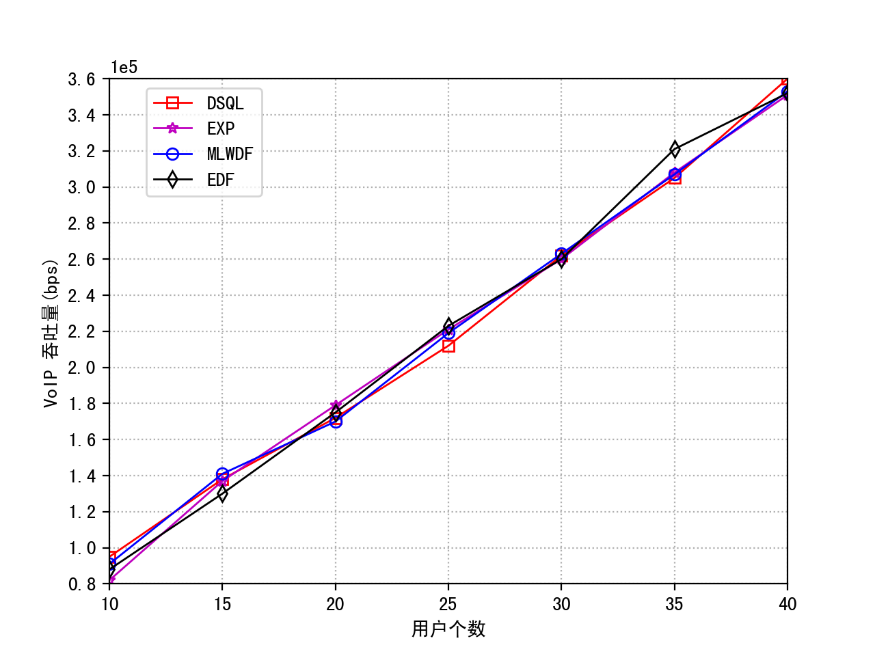
由于我们在设计决策空间和奖励函数的时候，重点考虑了系统吞吐量和时延参数，所以在公平性方面DSQL算法略差于EXP和MLWDF算法，而要做到可以在这几个指标上都得到优化是十分困难的。所以考虑到DSQL算法在系统丢包率和用户时延方面取得的优异表现，用户公平性方面的轻微牺牲是可以接受的。

为了进一步评估我们算法的性能，我们改变了仿真环境，用户数范围从10个以5个的间隔递增到40个，并且每个用户都请求视频业务和语音业务，两种业务的时延阈值分别设置为40ms和100ms。图3-12~3-13显示了混合实时业务下的算法性能。



### 图3-12 不同用户时video吞吐量

从图3-12可以看出，当系统中有10个用户时，DSQL、EXP和MLWDF的吞吐量一样，这是因为此时用户数少，系统资源充足，所有用户都分配到了资源，系统吞吐量即为所有用户所需的数据量总和。随着用户数的增加，系统吞吐量逐渐上升，但随之丢包率也开始上升，所以当用户数达到35时，EXP和MLWDF算法开始出现轻微下降的趋势，而DSQL算法仍在持续上升。EDF依旧表现最差。



### 图3-13 不同用户时VoIP吞吐量

图3-13则为混合业务中VoIP的吞吐量，可以看出几种算法有着相似的性能。这表明，相比于其他算法，DSQL可以在显著提高视频业务的性能的同时维持着语音业务的性能。

3.4 本章小结

本章首先介绍了LTE网络架构和协议栈架构，详细阐述了LTE系统中的关键技术；然后介绍了TDD和FDD两种无线帧结构，引出LTE系统中资源块的定义，描述了资源调度和资源分配的过程，并着重介绍了经典的资源调度算法。

本章针对视频这类实时业务，结合增强学习中的Q学习算法，提出一种基于M-LWDF改进的下行资源调度算法DSQL，该算法可以根据变化的网络状态和用户状态，动态地选择不同的调度策略，在系统吞吐量、系统时延和用户公平性之间取得平衡使得综合性能最大化。然后，本章介绍了使用的仿真平台LTE-Sim，描述了常用的仿真功能函数和仿真过程。最后，通过仿真实验从系统吞吐量、丢包率、时延和用户公平性几个方面对比了提出的算法与经典算法的性能优劣，最终验证了DSQL算法的性能优越性。

# 第四章 总结与展望

4.1 工作总结

随着计算机科学和多媒体技术的发展，全景视频作为一种全新的信息媒介以其沉浸式构想性的特点给人们带来了身临其境的观感，并逐渐应用到各个领域中。然而，全景视频数据量大、对时延要求严格的特性也给现有的传输网络带来了一些挑战，如何优化全景视频的传输一直是受人关注的研究领域。因此，本文从降低传输数据量和高效调度无线资源两个方面进行改进，主要工作研究与成果如下：

（1）为了降低全景视频的传输量，本文首先研究了视口预测算法。考虑到基于内容分析的预测算法的复杂性，本文只聚焦于基于历史轨迹的预测算法，该场景可以归于时间序列预测问题，在介绍了一些常见的时间序列预测算法之后，基于视口的二维特性，即经纬度，初步提出了基于序列到序列模型的预测算法，在数据集上初步验证了该算法的预测准确度高于其他一些算法，然后，为了进一步提高预测准确度，本文分析了观看同一个视频的其他用户的视口轨迹，发现轨迹之间的相似度极高，基于此前提，本文结合其他用户的视口数据对初步预测的结果进行矫正，最终，实验结果证明联合其他用户的预测算法在长时间预测时可以有效提高预测准确度。

（2）减少全景视频的传输量之后，本文又考虑了LTE网络中基站侧的无线资源调度。首先介绍了LTE的相关理论，包括网络架构和协议栈架构，详细阐述了LTE系统中的关键技术。然后介绍了无线资源块的概念，给出了资源调度和分配流程。针对全景视频这种对时延极为敏感的业务，本文研究了经典的调度算法的优缺点后，基于强化学习中的Q-Learning算法，提出了一种创新的下线资源调度算法DSQL，该算法根据动态变化的网络状态和用户状态，使用Q-Learning算法在每个传输间隔选择不同的调度策略，最终达到系统性能最大化目的。为了验证DSQL算法的性能优越性，本文介绍了仿真平台LTE-Sim的相关功能和仿真流程，通过仿真实验，从系统吞吐量、系统丢包率、时延和用户公平性几个方面综合对比了DSQL和经典算法的性能，最终验证了DSQL算法的优越性，尤其在时延敏感的实时业务调度方面。

4.2 未来展望

全景视频传输优化是一个庞大的研究领域，设计到计算机视觉、数据挖掘、无线网络传输优化等多个研究学科，本文只在现有研究基础上做了部分改进，仍有一些问题需要未来做进一步深入研究：

（1）本文只聚焦于基于历史轨迹的视口预测算法，没有充分考虑到影响视口的多个因素，如视频内容的多样性和用户的观看兴趣与习惯，没有建立充分完善的视口模型；在对视口预测时，分别预测了视口中心位置经纬度的规律，没有进一步探索其余预测模式，如基于区块的预测，这些因素都将会影响预测准确度，后续需要综合考虑深入研究。

（2）在资源调度方面，本文只是使用数据量较大这一个特征来代表全景视频业务进行调度，没有考虑全景视频的其他特性，如视口区域数据的重要性明显高于其他区域的数据，应当提高这些数据的优先级。同时在仿真过程中只考虑了单小区多用户的场景，而实际中更多的是多小区场景，存在小区间干扰等现象，之后应当更加紧密结合全景视频的特性，针对多小区的场景，进一步优化资源调度算法的性能。

（3）本文在研究资源调度算法时聚焦于LTE网络，而在研究后期，5G网络的发展逐渐加快，相关技术研究也不断完善，未来5G网络的覆盖将改善全景视频的传输问题，因此后续可以考虑结合新型网络架构与技术，进一步优化全景视频的传输。