

北京交通大学

硕士学位论文

视频点播缓存系统的优化研究

Optimization Research of Video on Demand Cache System

作者：王唯

导师：李纯喜

北京交通大学

2019年6月

学位论文版权使用授权书

本学位论文作者完全了解北京交通大学有关保留、使用学位论文的规定。特授权北京交通大学可以将学位论文的全部或部分内容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。学校可以为存在馆际合作关系的兄弟高校用户提供文献传递服务和交换服务。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：

导师签名：

签字日期： 年 月 日

签字日期： 年 月 日

学校代码：10004

密级：公开

北京交通大学

硕士学位论文

视频点播缓存系统的优化研究

Optimization Research of Video on Demand Cache System

作者姓名： 王唯 学 号： 16120191

导师姓名： 李纯喜 职 称： 副教授

学位类别： 工学 学位级别： 硕士

学科专业： 信息安全 研究方向： 信息网络

北京交通大学

2019年6月

致谢

在本论文的完成过程中，得到了李纯喜导师、赵永祥老师和实验室其他老师的悉心指导。老师们不仅为我指明了研究方向还提出了许多宝贵的意见，帮助我解决研究中所遇到的困难。老师们严谨的治学态度、实事求是的科研精神、渊博的学识都是我学习的楷模。在此我由衷地感谢李老师对我在学习和研究方面极大的帮助。

在实验室研究和撰写论文期间，还要感谢实验室的王璐师姐，黄楠栖师姐，赵红娜，张大富等同学以及我的舍友赵子涵和李媛媛在学习和生活期间对我的支持和帮助，感谢你们陪我度过这段难忘的学习生涯。

特别感谢我的家人和男朋友，家人的无私的付出和支持让我可以顺利的完成研究生阶段的学业，他们给予了我极大的帮助和勇气。

最后还要诚挚的感谢国家自然科学基金(No. 61872031)的资助。

摘要

人们越来越习惯使用移动终端在线观看视频，但用户体验受限于电池续航能力。在观看过程中，数据传输是主要能耗来源之一，研究高效的数据传输方法，以提高数据传输效率和节约传输能耗，进而延长电池续航时间，具有理论和实际意义。

现有针对传输节能的研究一般都假设信道带宽无限或是不可预测的变量。在此条件下，通过聚合数据流来延长移动设备无线接口休眠时间，或采取快速休眠机制以降低无线接口尾部能耗。然而，经研究证实，现实生活中人们通常会沿着固定路线行进（如通勤路线），沿途无线带宽是可预知的（可预先测量和预测）。在这个视角下，结合预知的无线信道带宽，有更大空间能够设计出更高效的视频传输方法，以便在满足用户视频观看质量的前提下，最小化传输能耗。

结合预知带宽高效传输视频面临以下关键挑战。首先，视频下载速度会受到移动终端缓存长度的限制，因此在有最好可用带宽时下载数据并不总是最好的选择。其次，由于视频启动后需要被连续播放，为避免缓存耗尽（引起播放卡顿），需要寻找一些好带宽以外的更合适的机会来下载数据。为此，本文提出了一种基于能效驱动的视频传输方法，该方法在视频播放连续性和传输能耗之间达到了巧妙的平衡，实现高效的数据传输。具体来说，本文的主要贡献如下。

(1) 建立了带宽预知的移动终端优化传输模型，以最小化视频传输能耗。该模型对连续视频时间进行离散化处理，进而将优化问题刻画为时间离散、以缓存限制为约束的整数线性规划问题，在保证视频传输播放流畅性的同时，最小化传输能耗。

(2) 提出了基于最大等效能效传输算法，以在保证视频播放连续性的同时节约传输能耗。基本思想是（按照贪婪传输）找到可优化段，对每个可优化段，以贪婪方式迭代选择具有最大等效能效（即单位能耗时间下载的数据量）的下载任务，直到数据全部下载完成。仿真结果表明，与贪婪传输和 On-off 传输相比，本文的算法分别节省了 83% 和 69% 的等效能耗，同时达到同贪婪传输相同的视频播放质量。

(3) 提出了重缓冲过程的动态缓存门限算法，对每个重缓冲（卡顿）段进行优化，在几乎不改变优化能耗的前提下，进一步减少视频卡顿时间。基本思想是，以现有播放连续性为底线，保持相邻重缓冲的后一个重缓冲过程的起始时刻不变，尝试改变前一个重缓冲过程的终止时刻，直到找到最小的重缓冲时刻，且不增加额外的卡顿。仿真结果表明，该算法可使总卡顿时间减少了原来的 82%。

关键词：在线视频；传输能耗；移动终端；数据传输

ABSTRACT

People are increasingly accustomed to using mobile terminals to watch online videos, but the user experience is limited by battery life. In process of viewing, data transmission is one of the main sources of energy consumption. Therefore, it is of theoretical and practical significance to study efficient data transmission methods to improve data transmission efficiency and save transmission energy consumption, thereby prolonging battery life.

Existing research on reducing transmission energy consumption generally assumes that the bandwidth of wireless interface is either unlimited or an unpredictable random variable. Under this assumption, some studies increase the sleep time of the wireless interface of the terminal device by aggregating data stream, or adopt a fast dormancy mechanism to reduce the tail energy consumption. However, some researches have confirmed that in real life, people usually move in fixed routes (such as commute routes), thus the changes in wireless bandwidth along the way can be treated as a known condition (pre-measured and predicts). In this perspective, combined with the known wireless channel bandwidth, there is more space to design an efficient video transmission method, which can minimize transmission energy consumption while satisfying the user's video viewing quality.

Efficiently transferring video in combination with known bandwidth faces the following challenges. First, the video download speed is limited by the length of the mobile terminal cache, so downloading data with the best bandwidth is not always the best option. Second, since the video needs to be played continuously after it is started, it is necessary to find some more suitable opportunities outside the good bandwidth in order to prevent running out cache. Therefore, this paper proposes an energy efficiency driven video downloading algorithm, which can achieve a delicate balance between video playback continuity and transmission energy consumption, and achieves efficient data transmission. Specifically, the main contributions of this article are as follows.

(1) This paper establishes a bandwidth-known mobile terminal optimized transmission model to minimize video transmission energy consumption. This model discretizes the continuous video time, and characterizes the optimal problem as an integer linear programming problem with discrete time. This problem is constrained by the cache

limit to ensure the best playback fluency and minimize the transmission energy consumption.

(2) This paper proposes a maximum equivalent energy efficiency based video downloading algorithm, which can reduce the transmission energy consumption while ensuring video playback continuity. The basic idea is to find optimizable segments (according to the Greedy algorithm). For each optimizable segment, it iteratively selects the download job with a maximum equivalent energy efficiency (i.e., the maximal amount of data that can be transmitted per unit energy consumption time), until the data is completely downloaded. Simulation results demonstrate that the proposed algorithm significantly outperforms the traditional Greedy algorithm and the On-off transmission algorithm with an equivalent energy efficiency decrease of up to 83% and 69%, respectively. At the same time, it achieves the same video playback quality as greedy transmission.

(3) This paper proposes a dynamic buffer threshold algorithm for the re-buffering process, it optimizes each re-buffering segment to further reduce the video re-buffering time while the energy consumption optimization is almost unchanged. The basic idea is to ensure the existing playback continuity, keep the starting moment of the next re-buffering process of the adjacent re-buffer unchanged, and try to change the termination time of the previous re-buffering process until the minimum re-buffering time is found without causing extra re-buffering time. The simulation results show that the proposed algorithm can reduce the total re-buffering time by 82%.

KEYWORDS: Online Video; Energy Saving Transmission; Mobile Terminal; Data Transmission

目录

摘要	v
ABSTRACT.....	vi
目录	viii
1 引言	1
1.1 研究背景和意义	1
1.2 研究现状	2
1.2.1 节能传输相关研究	2
1.2.2 基于未来带宽相关研究	3
1.3 研究内容	4
1.4 文章组织结构	5
2 相关研究	6
2.1 移动终端视频传输方式	6
2.2 移动终端视频服务衡量指标	7
2.3 移动设备传输能耗模型	8
2.3.1 4G-LTE 协议概述	8
2.3.2 4G-LTE 能耗模型	8
2.4 移动设备传输节能现状	9
2.5 利用未来带宽研究现状	11
2.6 本章小结	12
3 移动终端优化传输模型	13
3.1 应用场景	13
3.2 设计动机	14
3.3 面临的挑战	15
3.4 移动终端优化传输模型	17
3.4.1 参数化系统	17
3.4.2 优化模型	19
3.5 相关推导	22
3.6 本章小结	23
4 视频下载算法设计和实现	24

4.1	基于最大等效能效传输算法	24
4.1.1	问题提出和挑战	24
4.1.2	基本思想	24
4.1.3	算法概述	25
4.1.4	算法设计和实现	25
4.2	重缓冲过程的动态缓存门限算法	30
4.2.1	问题提出和挑战	30
4.2.2	基本思想	31
4.2.3	算法概述	32
4.2.4	算法设计和实现	32
4.3	相关推导	34
4.3	本章小结	35
5	性能评估	36
5.1	基于最大等效能效传输算法仿真分析结果	36
5.1.1	实验配置	36
5.1.2	下载行为分析	37
5.1.2	能效性能分析	38
5.2	重缓冲过程的动态缓存门限算法仿真分析结果	41
5.2.1	评估办法	41
5.2.2	卡顿性能分析	41
5.3	本章小结	43
6	结论	44
6.1	本文工作总结	44
6.2	后续研究计划	45
	参考文献	46

1 引言

1.1 研究背景和意义

视频已经成为互联网最流行的在线娱乐形式。据中国网络视听发展研究报告显示,截止到 2018 年 6 月份,中国网络视频用户规模为 6.09 亿,占网民总体的 76%,网络视频流量已经占据互联网流量的 80%以上^[1]。同年 11 月份,Sandvine 发布“全球互联网现象报告”称视频贡献了 58%的下行流量,占据网络主导地位,并且流媒体视频类程序一直是近年来互联网上的顶级应用程序类型^[2]。

随着信息技术的飞速发展,视频质量越来越高,可供用户观看的视频类型也越来越多。然而观看这些高质量、多选择的视频内容却极大的消耗了电池电量。与此同时,现有技术下电池的储电能力是有限的,由电池供电的移动设备例如手机、iPad 等无法长时间满足用户在线观看视频的需求。因此如何在有限的电池电量下降低移动设备的能耗以延长电池续航时间是一个值得研究的问题。

研究移动设备节能具有理论和现实意义。对环境而言,移动设备在消耗能源的同时造成二氧化碳排放,使全球变暖加重。为了实现能源和环境的可持续发展,越来越多的国家开始推广节能低碳的网络服务,移动设备的节能问题也受到广泛关注;对用户而言,节能能够使用户在观看同等质量和时长的视频下消耗更少的电量,从而为用户提供更持久的网络服务,使用户的生活更加便利。

目前,国内外已经有很多针对移动设备节能的研究。根据不同的耗能原因,现有的节能研究主要是从优化移动端硬件耗能组件、设计高效率的数据传输机制、减少无线接口尾部能耗、控制数据收发速率、利用云迁移将需要密集计算的任务委托给远程服务器等方式进行节能。

数据传输是移动设备能耗的主要来源之一^[3],降低传输能耗对移动设备有重要意义。现有的移动设备节能传输大部分都是假设未来无线接口的带宽变化是无限的或者不可预测的随机变量,然而在很多情况下人们通常会沿着固定的上下班路线或者使用导航路线,其固定路线上的带宽变化状况是类似的^[4-5],因此在实践中人们在路上所经历的未来的无线带宽是可预测的。这种可预测的带宽通常被用于减少视频卡顿时间以改善视频服务质量,而很少考虑到移动设备的能耗问题。因此,如何在提供给用户满意的视频流畅观看体验质量下降低传输能耗是一个值得研究的问题。

本文关注视频播放过程中的传输能耗问题和卡顿问题。针对传输能耗和视频

卡顿两个问题,在给定未来无线接口带宽信息的前提下,本文提出了基于最大等效能效传输算法和重缓冲过程的动态缓存门限算法。第一个算法在保证用户拥有同贪婪传输相同的流畅度播放质量下最小化传输能耗,第二个算法在第一个算法的基础上进一步的减少了视频卡顿时间。

1.2 研究现状

数据传输是移动设备的主要能耗来源之一。随着通信技术的飞速发展,越来越多的用户通过无线网络在移动设备上观看视频。但是移动设备可存储的电量是有限的,不能长时间支持视频播放,且播放质量越高的视频内容其耗费的电量越大。在电池技术尚未取得突破性发展之前,如何降低移动设备能耗的问题已经成为研究热点。在移动设备使用过程中,数据传输所消耗的能量最多。这是由于在传输数据时,移动设备上的无线接口需要保持在高耗能阶段以下载数据,从而需要大量功率。由此衍生了一系列以降低移动设备传输能耗为目标的研究。

现有研究采取了各种方法以减少移动数据传输能耗。这些方法普遍都是基于信道带宽无限或者信道是不可预测的变量的假设下,通过延长无线接口休眠时间或者减少无线接口尾部能耗降低传输能耗。但是,它们没有考虑到有限带宽的限制。实际生活中,无线网络经常会遭受到严重的带宽波动导致网络在短时间内完全失去响应,从而导致视频发生连接中断,卡断等现象,影响用户观看体验质量。下面,本文将根据无线接口带宽信息是否预知把现有的研究工作分为以下两类:第一类,假设无线接口的带宽是无限的或者不可预测的随机变量;第二类,假设无线接口的带宽不仅是有限的而且是可预测的。其中,第一类介绍现有的数据传输节能现状,第二类介绍利用未来带宽信息的研究现状。

1.2.1 节能传输相关研究

关于不利用未来带宽信息来降低移动设备传输能耗的研究主要有以下几种方法:

一些研究通过设计高效率的数据传输机制来优化数据传输能耗。例如,文献[6]测量出不同强度的网络信号下,数据传输所消耗的功率不同,提出了基于预测网络信号强度安排数据传输任务的动态下载算法,有效的节省了能耗。

一些研究通过减少无线接口尾部能耗来节省能量。当无线接口完成数据传输时并不会马上进入休眠状态,而是需要经过一段延迟时间。这段延迟时间被称为尾部时间,尾部时间消耗的能量称为尾部能耗。尾部能耗是电池被浪费的主要来源之

一^[7]，且当移动设备使用 4G-LTE、3G、WIFI 等通信网络时都会出现尾部时间^[8]。降低尾部能耗的解决方法主要可以分为两类，尾部分割机制和尾部共享机制。尾部分割机制通过预测未来的数据流到达信息来使无线接口在数据连续传输完成后快速进入休眠状态以减少尾部时间。例如文献[9]针对 3G 网络中的无线接口尾部能耗提出了 SmartCut 方案，该方案通过数据包到达时间的相关性来预测即将到来的数据并基于这些数据利用快速休眠机制有效的减少了不必要的尾部能耗。由于数据流的到达状况是没有规律的，因此在实际应用中，尾部分割机制的准确度有限。是没有尾部共享机制通过聚合数据流量，使无线接口大部分处于休眠状态以节省能耗。例如文献[10]通过重新调度来自上层应用程序的数据包并将数据包打包在一起传输以共享尾部。但是由于尾部共享机制会带来时延，因此不适应于流媒体等对实时性要求较高的应用。

此外，文献[11-14]通过计算迁移的方法将移动设备需要密集计算的任务委托给远程服务器上来降低能耗。计算迁移能够有效的降低能耗，但是只适用于需要密集计算的应用，像流媒体这类反而可能带来额外的能耗。

总之，上述的这些节能研究都假设无线接口的带宽是无限的或不可预测的随机变量。然而在现实生活中，由于当前技术的限制，无线网络经常会遭受到严重的带宽波动，使得网络在几秒甚至几分钟内完全失去响应，导致视频发生连接中断、卡顿等现象，影响用户观看体验质量。

1.2.2 基于未来带宽相关研究

现有的数据传输节能研究方案绝大部分是基于未来无线带宽未知的条件下。然而随着无线技术的发展，在实践过程中，未来带宽变化是可预测的。

已有研究表明，未来无线带宽变化可以由人类的移动行为推测得出^[15]，而人类 93%的移动行为是高度可预测的^[16]。文献[17]彻底研究了网络带宽的可预测性。文献[18]和文献[19]进一步发现人类的移动模式有助于提高预测带宽的可用性。文献[20]在以上的基础上提出了一个新的随机算法来预测移动网络用户经历的带宽变化。

人类移动行为的高预测性已经被很多研究用于推测未来的无线信道带宽变化，并进一步将未来带宽信息用于改善视频传输上。例如，文献[21]使用配有 GPS（全球定位系统）接收器的移动设备来构建数据库，该数据库通过地理位置和网络状况实现带宽预测，最后基于预测的带宽信息，向用户提供了播放质量更稳定，更流畅的视频。

目前，未来无线信道带宽信息已经被用于多方面的研究。例如，文献[22-25]用

于优化网络资源分配以提高系统利用率。文献[26]用于优化视频传输策略以减少视频卡顿时间。

综上所述,目前已经有研究将未来带宽信息用于改善视频传输上,这也证明了在实践过程中,未来带宽的确是可以被利用的。但是现有的利用未来带宽信息视频传输研究绝大多数都是以减少视频卡顿次数、提高播放流畅度为目标,而很少考虑到移动设备的能耗问题。因此,如何利用未来带宽信息研究视频传输方案,使得在提供给用户高流畅度的观看质量下尽可能的降低能耗是一个关键问题。

1.3 研究内容

用户通过移动设备观看在线视频时,有限的电池容量无法满足用户长时间的观看需求。而在移动设备工作过程中,传输能耗是主要的能耗来源,降低传输能耗以延长电池续航时间对移动设备具有重要意义。现有的针对传输节能的研究大都是假设无线接口带宽信息是未知的,由于带宽未知,这些方法难以抵御带宽波动带来的网络性能的不稳定性。已有观察发现,沿着相同路线上无线带宽的变化状况是类似的,可以被预先测量。基于这样的观察,有一部分研究利用未来带宽信息研究视频传输方案,但多以减少视频卡顿时间为目标,而很少考虑到耗能问题。本文利用未来带宽信息来研究视频下载方案,研究目标是在提供给用户令人满意的视频流畅度观看体验质量下,最小化传输能耗。本文的主要工作如下。

(1) 利用未来带宽信息建立视频传输模型,以最小化传输能耗。本文首先针对优化问题建立优化模型,将优化问题表示为一个时间离散的、以最小化传输能耗为目标、以缓存限制为约束的整数线性规划问题。

(2) 针对上一个模型,提出了基于最大等效能效传输的下载算法,目标是在保证用户拥有同贪婪传输相同的视频流畅性播放体验的同时减少传输能耗。该算法首先按照贪婪传输的方式下载视频以找到可优化段,然后将每个可优化段下载安排为一系列候选任务,这些任务以贪婪的方式在所有的候选任务中迭代选择,使得每个被选择的任务具有最大的等效能效(即单位能耗时间内下载的数据量),直到数据全部下载完毕。在本文的数值实验结果中,本文提出的算法在传输能耗上明显优于贪婪传输和 On-off 传输,分别降低了 83% 和 69% 的等效能耗。与此同时,本文的算法同贪婪传输具有相同的视频流畅性播放质量。

(3) 提出了重缓冲过程的动态缓存门限算法,通过对每个重缓冲(卡顿)段进行优化来进一步减少视频卡顿时间。在上一个模型中,当视频发生卡顿时,总是将缓存区填满才能继续播放,卡顿停止的这段时间并没有在优化范围内。基于这个观察,本文通过动态改变缓存区的填充程度以进一步的对卡顿段进行优化。其基本思

想是在不增加现有视频卡顿时间的前提下，保持相邻卡顿段的后一个卡顿段的起始时刻不变，尝试改变前一个卡顿段的终止时刻，直到为前一个卡顿段找到最小的终止时刻。在本文的数值实验结果中，本文提出的算法能够进一步的将总卡顿时间减少了原来 82%，同时总优化能耗基本上不变。

1.4 文章组织结构

本文一共六章，每章节内容安排如下：

第一章介绍了论文的研究背景和研究意义，简要概述了移动设备的节能传输研究现状，并介绍了主要研究内容和文章组织结构。

第二章是相关背景研究，首先介绍了现有的移动终端视频传输方式，然后给出了基于 4G-LTE 网络的移动设备传输能耗模型，最后详细介绍了现有的移动设备节能传输研究。

第三章建立了移动终端优化传输模型。首先介绍了移动终端观看视频的应用场景，然后介绍了设计动机和面临的挑战，最后建立了节能传输模型并给出模型公式。

第四章提出了两个视频节能传输算法，分别是基于最大等效能效传输算法和重缓冲过程的动态缓存门限算法。并分别介绍了这两种算法的提出背景和面临的挑战、基本思想和实现方法、算法概述和具体实现过程。

第五章给出基于最大等效能耗传输算法和重缓冲过程的动态缓存门限算法的数值分析结果。

第六章对本文的工作进行了总结，并指出本文下一步的工作。

2 相关研究

本章主要介绍了现有的移动终端视频传输方式和移动设备耗能模型，以及移动设备节能传输研究现状。本章的结构安排如下：2.1 介绍移动终端视频传输方式；2.2 介绍移动终端服务面临的挑战；2.3 介绍基于 4G-LTE 无线网络的移动设备能耗模型；2.4 介绍现有的视频传输节能研究；2.5 介绍现有的利用未来带宽的研究；2.6 为本章小结。

2.1 移动终端视频传输方式

首先本节介绍现有的移动终端视频传输机制^[27]。如下图 2-1 所示，传输机制包括服务器端和移动终端两部分。其中服务器存放所有用户需要的视频文件，移动终端用来接收并播放从服务器下载的视频文件，在移动终端一侧配有一个缓存区，用于存放下载的数据。当一个用户在线观看一个视频时，首先由移动终端通过无线网络向视频服务器提出传输请求，然后由服务器响应请求。

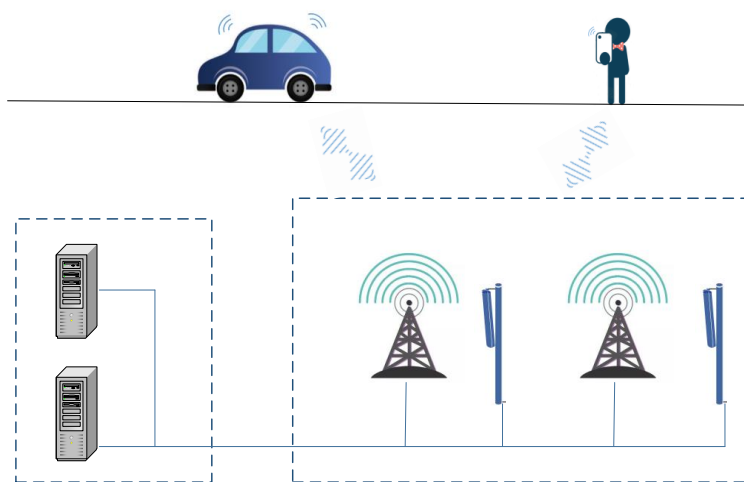


图 2-1 移动终端视频传输方式

Figure 2-1 Mobile terminal video transmission mechanism

服务器响应移动终端请求的方式大致上可以分为两种，完全下载方式和流式传输方式。完全下载是指用户端将视频数据全部下载完成后才开始播放。这种方式等待时间长，且很有可能因为用户中途放弃观看而造成资源浪费。流式传输是将要传输的视频数据分割成多个小的数据块，移动终端采用一边下载一边播放的方式，只要缓存区内存有一定的数据量移动终端就可以启动播放。与完全下载相比，流式传输具有低时延，高灵活性的优点，因此是普遍被用户采取的一种下载方式，这也是本文采用的视频下载方式。

当服务器采用流式传输技术传输视频数据块时，移动终端在收到数据块后先放在缓存区内，播放器再从缓存区内按照顺序取出数据进行播放。一旦视频开始播放，只要缓存区内有数据，播放器会按次序从缓存区取出数据并顺序进行播放。同时下载新的数据继续存放在缓冲区中。在整个视频播放过程中，移动终端需要不断的从视频服务器顺序下载新数据存入缓冲区中以保证视频能够持续流畅播放。

2.2 移动终端视频服务衡量指标

由于网络信号强度和移动设备处理能力等因素限制，流媒体视频服务面临着服务质量和能耗的挑战。如何在保证视频服务质量的情况下降低移动设备能耗是一个关键问题。

(1) 服务质量

用户体验质量是衡量流媒体视频服务质量一个重要指标。当用户通过无线网络观看在线视频时，网络信号强度、移动设备有限的缓存长度和计算能力等因素导致视频在播放过程中很可能会出现卡顿、甚至长时间停止播放等现象，从而明显的降低了用户的观看体验质量。

(2) 能耗

在视频传输过程中，移动设备一直处于高耗能阶段，需要消耗大量的电量。而现有技术下电池的电量是有限的，无法维持视频长时间播放。而且移动设备通常以便携性和轻巧性为设计目标，在该目标下不可能将电池做的特别大。因此，在电池技术尚未取得突破性进展的状况下，如何降低移动端能耗以尽可能延长续航时间是一个十分受人关注的问题。

在流媒体视频传输过程中，网络性能（如可靠性、可扩展性、用户体验质量）和能耗（包括网络设备能耗、计算能耗和应用程序能耗）之间是有直接关系的。根据文献[28]，在移动通信网络中，网络性能和能耗之间的关系如下图 2-2 所示。

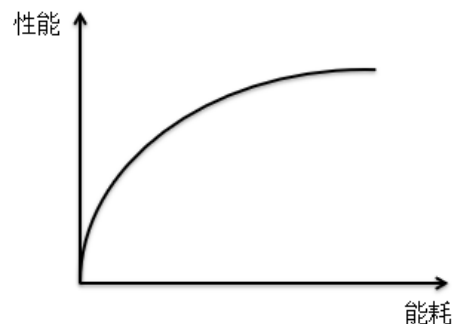


图 2-2 性能和能耗之间的关系^[28]

Figure 2-2 Relationship between energy consumption and performance^[28]

从图 2-2 中可以看出，更高的性能意味着需要消耗更多的能量。因此节能研究在降低移动设备能耗的同时，需要考虑到用户体验质量。如果一味的减少能耗而导致用户体验质量大幅度降低是得不偿失的。

2.3 移动设备传输能耗模型

2.3.1 4G-LTE 协议概述

4G-LTE (Long Term Evolution) 是 4G 移动数据网络的简称。4G 是第四代移动通信技术，包括 TD-LTE 和 FDD-LTE 两种制式，能够快速传输视频等数据，几乎能满足所有用户对无线服务的要求^[29-32]。

4G 以高速的通信速度、灵活的通信方式、高兼容性和提供增值服务等优良性能深受大众喜欢。目前，4G-LTE 协议已经被广泛应用于移动设备终端，它提供给用户更高质量的通信服务和更广泛丰富的多媒体服务。相比 3G 协议，4G 大幅度降低了网络延迟，使用户浏览网络的速度更快。但与此同时，4G-LTE 终端需要耗费更多的电量。

2.3.2 4G-LTE 能耗模型

针对 4G-LTE 通信网络，RRC 状态机模型是普遍被采用的移动设备能耗模型^[33]。在 4G-LTE 网络中，由无线资源协议(RRC)控制移动设备和基站之间的连接状况，它的工作机制直接影响了 4G-LTE 移动设备的能耗。4G-LTE 移动设备能耗模型如图 2-3 所示：

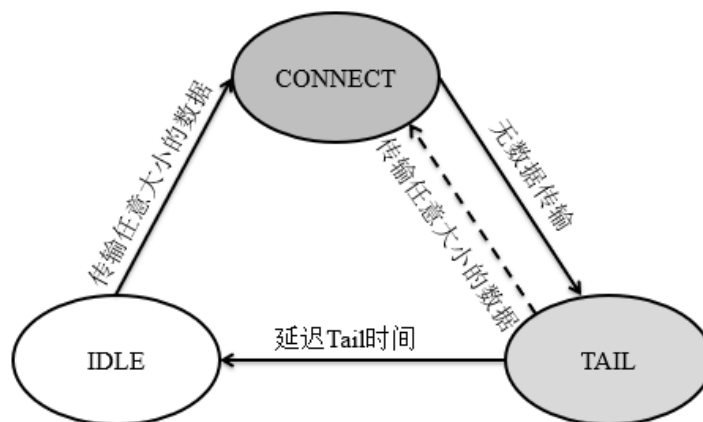


图 2-3 RRC 协议对应状态转换^[33]

Figure 2-3 RRC state transition^[33]

如图 2-3 所示, RRC 为 4G-LTE 网络中的无线接口设置了三个状态, 分别是 CONNECT 状态、IDLE 状态和 TAIL 状态。当无线接口需要传输数据时, 移动设备进入到 CONNECT 状态, 在此状态期间移动设备处于高能耗阶段, 需要消耗大量功率。当数据传输完成之后, 移动设备自动进入到具有较低能耗的 TAIL 状态。该状态下消耗的功率虽然比 CONNECT 状态低, 但是仍然不可忽视。在 TAIL 状态消耗的功率通常被称为尾部能耗。移动设备在 TAIL 状态会经历一个 Tail 时长的延迟, 如果 Tail 时长期间仍然没有数据需要传输, 移动设备会进入 IDLE 状态。在 IDLE 状态下, 移动设备被 RRC 设置为 DRX (不连续接收模式), 该模式消耗的功率极低, 可以忽略不计。一旦再次接受到数据, 无论要传输的数据有多小, RRC 将立刻触发移动设备使其自动切换到 CONNECT 高耗能阶段。

此外, 移动设备从 IDLE 状态提升到 CONNECT 状态需要经历一段时间延迟才能进行数据传输, 这段延迟时间通常被称为 Promotion 延迟, 本文将 Promotion 延迟期间移动设备所处的状态称为 Promotion 状态^[34], 该状态也会带来额外的功率消耗。

综上所述, 当移动设备处于 IDLE 状态下需要完成一次数据传输时, 无线接口需要经历以下的状态变化: Promotion (提升状态) → CONNECT (数据持续接受状态) → TAIL (尾部状态) → IDLE (休眠状态)。每个状态下无线接口消耗的功率值也是不同的。

据文献[35]测量结果显示, 通常手机在一次数据传输过程中, 在 TAIL 状态的延迟时长为 10.27 秒, 消耗的能量为 1266.62mW; 在 Promotion 状态的延迟时长为 0.67 秒, 消耗的能量为 1548.58mW; 在 CONNECT 状态消耗的功率为 1568.26mW。

3G 网络下无线接口的工作状态和 4E-LTE 类似^[36]: 3G 网络中基站为每个移动设备配置了两个计时器 Time1 和 Time2。当无线接口传输数据时移动设备保持在高功耗的传输状态, 如果在 T1 时间内无线接口仍然没有接受到任何数据, 移动设备将自动切换到尾部状态; 如果在 T2 时间内无线接口还是没有接受到任何数据, 则移动设备进入 IDLE 状态, 此状态几乎不消耗功率。接下来, 本文以 4G-LTE 网络为背景进行研究。

2.4 移动设备传输节能现状

流媒体视频一直是近年来深受用户欢迎的娱乐方式。随着通信技术的广泛普及, 越来越多的用户通过无线网络在移动设备上观看视频。由于移动设备的电池储量是有限的, 用户对视频质量要求越高电池将越快的被消耗, 无法长时间满足用户的观看需求。在移动设备的总能耗中, 数据传输占据了主要来源, 降低传输能耗对

于移动设备节能具有重要意义，由此激发了一系列致力于降低传输能耗的研究。

本文根据无线接口是否处于持续传输状态将现有的研究主要分为以下两类：休眠节能方案和减少尾部能耗方案。其中，休眠节能方案主要是通过改变数据传输策略来延长无线接口的休眠时间以减少传输能耗。减少尾部能耗方案主要是通过降低无线接口在尾部状态消耗的能量以达到节能的目的。

(1) 休眠节能方案

现有的休眠节能研究可以分为以下几方面：

一方面通过聚合来自上层应用程序不同的数据流以减少传输次数，从而使无线接口大部分处于休眠状态来达到省电的目的。例如文献[37]通过重新调度来自上层应用程序的数据包并将数据包打包在一起以共享尾部，从而有效的减少了传输次数以节省能耗，并进一步的预测了在高信号强度下传输打包数据的适合时间。但是这种解决方案通常会导致大量数据包延迟，影响用户观看体验，不适用于流媒体视频等对实时性要求较高的应用。

还有一方面通过设计高效率的传输机制来减少传输能耗。例如，文献[38]结合传输时间要求和无线信道环境提出优化传输算法，动态决定数据传输时间。文献[39]通过研究不同强度的网络信号下发现传输同样的数据所消耗的能量不同，提出了基于预测网络性能安排数据传输的动态算法。文献[40]提出了一种数据传输算法，该算法通过判断路由请求预测数据包的方法来降低移动设备在发送和接受消息时的能耗。文献[41]根据无线信道状况，将原有的数据包拆分成更小的数据单元，在不影响数据连续播放的前提下，重新安排数据单元的发送顺序，通过合并来自不同的数据包的数据单元进行传输，有效的合并了原有的发送空隙使无线网卡具有更长的时间进入休眠。

现有的移动端通常为用户提供多个无线接口，例如 4G、Wife、3G 等。因此还一些研究通过在移动端上分配不同的无线接口来降低能耗。文献[42]从移动端缓存区出发提出一个播放缓存感知算法。该算法将根据缓存区的剩余数据量决定是否开启 WIFI 接口。文献[43]提出通过预测网络性能来选择自动选择无线接口类型以减少传输能耗。

(2) 减少尾部能耗方案

当移动设备通过 4G-LTE、3G、WIFI 等通信网络进行数据传输时，传输完成后移动设备会经历一段时间的尾部状态后才转向休眠状态。尾部状态是无线网卡处于数据连续传输状态和休眠状态之间的过渡状态。需要传输数据时，无线网卡自动切换到连续数据传输状态。当没有数据需要传输时，无线网卡转向休眠状态以节省能量。尾部状态的存在能够使得无线网卡减少 Promotion 延迟，有效的保证了视频播放的流畅性。因此通常情况下会把 Tail 时长设置的较大，但同时，越大的 Tail

时长浪费越多的能量。因此在减少尾部时间时以降低移动设备能耗时需要考虑到用户体验质量。现有的减少尾部能耗的解决方案可以分为以下几个方面：

一方面提出根据网络负载状况和用户需求来自动设置尾部延迟的算法^[44-45]。但是这种方法会增加无线网卡状态转换次数，从而带来时延，降低用户的观看体验质量。

还有一方面通过利用未来数据包到达信息采用快速休眠方法减少尾部能耗。例如文献[46]基于链路层是否有数据到达提出一种 TOP 算法，当没有要传输的数据时，该算法启动快速休眠机制来缩短尾部时间，减少尾部能耗。文献[47]同样提出根据数据流传输状况利用快速休眠机制来降低尾部能耗的方法。但是这类方法都需要准确的预测数据流传输状况，然而实际生活中由于数据流的达到信息是很难有规律的，因此实现困难。

除了休眠节能方案和减少尾部能耗方案之外，还有一类研究通过调整缓存区大小来减少额外的传输能耗。由于用户在视频观看过程中可能会突然跳过或退出当前播放进度，使得缓存区内已经下载完成的数据被浪费，从而导致额外的电量消耗。一些研究针对这个问题基于预测用户观看行为提出改进方法。文献[48]设计了 GreenTube 系统，该系统根据用户当前观看视频的剩余时间动态的选择缓存区大小，以避免用户跳出或退出时下载太多的数据，从而达到优化视频下载能耗的目的。文献[49]通过统计用户历史观看记录来预测用户何时将退出视频播放，从而节省缓存区需要下载的数据量以减少能耗。但是没有考虑到跳过的情况。文献[50]提出一种在线感知的视频下载方案，它根据用户是否倾向于长时间观看视频或者倾向于跳出视频将用户分为稳定和不稳定两部分，并应用不同的方案来节省能耗。但是这种方法依赖于预测用户观看行为的准确度，而用户实际观看行为往往是难以准确预测的。

此外，还有一些研究通过改变视频编码方式来节省传输功率。文献[51]基于无线信道环境提出了一种数据传输节能算法，该算法通过动态的改变视频编码方式和传输算法来节省传输能耗。

总体来说，这些节能传输方法都是假设信道带宽是无限的或者是不可预测的变量。由于带宽未知，它们无法抵御带宽波动带来的网络性能的不稳定性影响，从而影响用户观看质量。

2.5 利用未来带宽研究现状

针对无线信道的未来带宽变化状况，已经有很多研究进行预测并将预测的信息用于改善视频传输上。例如，文献[52]针对快速行驶的车辆，首先通过重复测量

的方式将网络带宽映射到道路网络中，然后对映射的带宽结果的可行性进行了验证，最后将映射数据用于改进视频服务质量上，有效的减少了视频的中断次数。文献[53]提出了一种基于用户位置的自适应视频流传输算法，该算法通过获取用户移动的下一个位置的估计带宽来调整下一个视频块的质量，从而使用户更好的视频地理空间变化带来的带宽不稳定性影响。

目前，已经有很多无线信道带宽的数据集供人们研究使用。例如，文献[54]收集了来自奥地利的真实带宽跟踪数据。文献[55]收集了来自挪威的真实带宽跟踪数据。文献[56]展示并提供了移动设备使用 3G 网络时挪威一些热门通勤路线上的带宽测量数据集。这些数据集已经被各种利用未来带宽信息的研究所使用。

现有的利用未来带宽信息进行的研究有以下几个方面。例如，文献[57]提出了一种基于预测的数据速率来安排视频下载内容的算法，以减少中断次数。文献[58]针对移动端耗能问题优化了不连续接受 (DRX) 的周期长度以减少尾部能耗，但是这种 DRX 调整在实践中可能会引起额外的信令开销，降低用户观看体验。

本文主要关注移动设备在传输数据过程中的能耗问题。不同于已有的研究，本文提出的视频节能传输方案利用未来无线信道带宽信息并通过减少无线接口的总能耗消耗 (包括 CONNECT 能耗和尾部能耗) 来节省能量，同时无需在硬件设备上做出任何更改。

2.6 本章小结

本章主要介绍了移动终端视频传输机制和传输能耗模型，并总结了现有的视频节能传输研究。现有的节能传输研究很少利用到未来带宽信息，而将未来带宽信息用于视频传输策略的研究多以减少视频卡顿时间为目标而很少考虑到移动设备的能耗问题。不同于已有的研究，本文利用未来带宽信息不仅考虑了移动设备的能耗问题而且考虑了视频卡顿问题。

3 移动终端优化传输模型

本章利用未来带宽信息建立视频节能传输模型。本章的结构安排如下：3.1 介绍移动终端用户在线观看视频的应用场景；3.2 介绍设计动机；3.3 介绍面临的挑战；3.3 建立移动终端视频优化传输模型，并给出模型公式；3.4 为本章小结。

3.1 应用场景

本节首先介绍一个典型的移动终端用户观看在线视频的应用场景。假设一个用户正在通过无线信道下载数据以观看在线视频。无线信道的带宽随时间变化，且未来带宽信息是已知的。移动设备具有固定长度（设为 w ）的缓存区用于保存下载的视频数据，视频具有恒定的播放速率（设为 r ）。一旦视频开始播放，只要缓存区有足够的视频数据，移动终端会以播放速率 r 不断的从缓存区取出数据并按照顺序进行播放，空出的缓存区位置再次用来保存新下载的视频数据。在整个视频观看过程中，移动终端需要不断的从服务器下载视频数据填充空出来的缓存区位置以保证视频能够连续播放，同时在任意时刻已经播放的数据总量不能超过已经下载的数据总量。为了便于理解，本文使用图 3-1 来表示视频下载过程中需要满足的约束条件。

图 3-1 包含上下两个子图。下图是无线信道带宽图，图中横坐标表示时间，纵坐标表示信道容量，在这里假设信道容量足够好使得视频能够流畅播放。上图是视频下载图，图中横坐标表示时间，纵坐标表示下载的数据量。上图和下图中的时间是相同的，它们分别显示了在某一时刻的信道容量和下载的数据量。本文假设视频长度为有限长度 T (s)，视频从 0 时刻开始下载，此时缓存区的容量是满的，下载过程中需要满足的约束条件如上图所示。图中标注为“缓存区上限”的曲线表示下载上限，它表示在任意时刻 t ($t \in T$) 由于缓存区的限制用户在 t 时刻能够下载最大的总数据量（即 $y = r * t + w$ ）；标注为“缓存区下限”的曲线表示播放曲线，它表示在任意时刻 t ($t \in T$) 用户在 t 时刻已经观看的总数据量（即 $y = r * t$ ）。因此，在任意时刻 t ($t \in T$)，用户实际上能够下载的数据量被限制在这两条曲线之间。

如图所示，本文的目标是在缓存区上限曲线和缓存区下载曲线之间寻找一个合适的下载曲线使得移动设备在传输视频时具有尽可能低的传输能耗（暂时忽略图 3-1 中缓存区上限和缓存区下限之间的三条曲线，它们将在后文中介绍）。

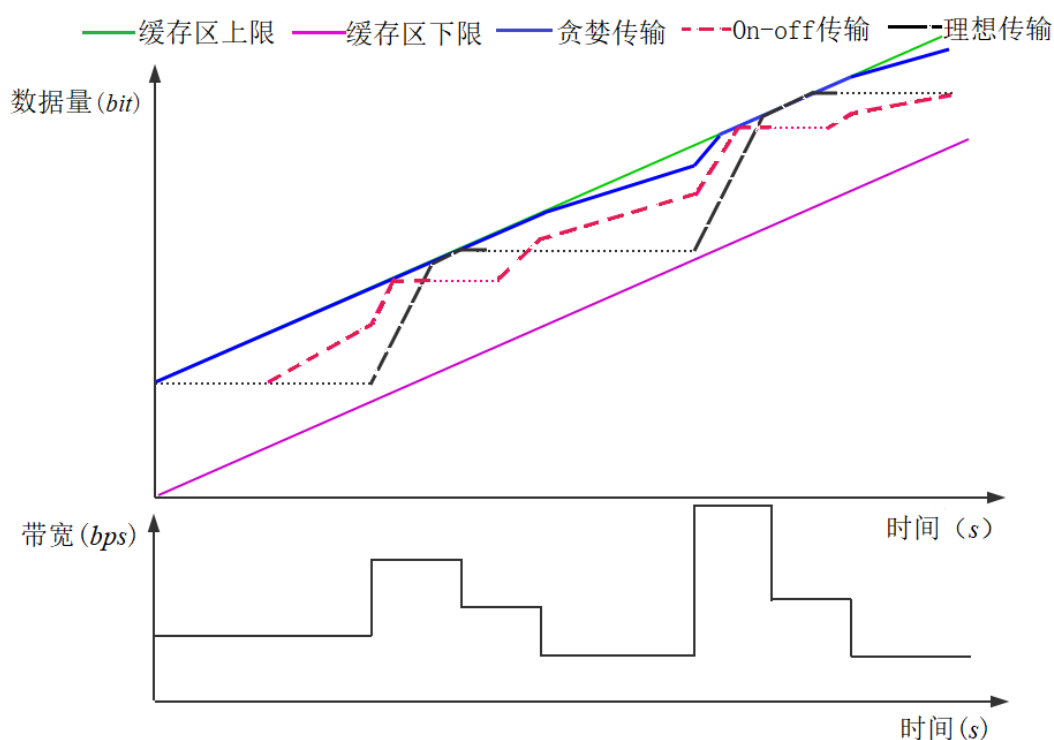


图 3-1 下载约束条件

Figure 3-1 The download constraint

3.2 设计动机

贪婪传输和 On-off 传输是两种典型的视频下载方案。本节首先观察这两种传输方案的下载过程以及它们在下载过程中无线接口的耗能状况。

如图 3-1 所示，这里假设视频是通过 4G-LTE 网络进行传输的，图中标注为“贪婪传输”和“On-off 传输”的曲线分别代表贪婪传输和 On-off 传输下载方式，标注为“理想传输”的曲线代表理想下载方式。图中实线部分表示无线接口处于 CONNECT 状态、TAIL 状态或 ROMOTION 状态对应的高耗能阶段，虚线部分表示无线接口处于 IDLE 状态对应的低能耗阶段。接下来本节分别介绍贪婪传输和 On-off 传输的具体下载方式。

贪婪传输：贪婪传输是一种尽可能的保持缓存区的数据量为满的视频传输方式。在整个贪婪传输期间，由于播放器持续的从缓存区取出数据进行播放，导致移动终端需要不断的从视频服务器处下载新的数据来填充缓存区，因此贪婪传输的下载曲线最接近于图 3-1 中的缓存区上限曲线。在整个贪婪传输期间无线网卡几乎一直保持在高耗能阶段对应的 CONNECT 状态，消耗大量能量。显然的，贪婪传输是一种耗能大的视频传输方案。但与此同时，由于贪婪传输期间无线网卡几乎一直处于开启状态下载数据，它也是卡顿时间最少，用户流畅度观看体验质量最好的

下载方案。

On-off 传输： On-off 传输被当前大多数在线视频播放系统所采用，它是一种由当前缓存区填充程度决定无线网卡状态是否开启的传输方式。On-off 传输为缓存区设置了两个阈值，分别为上限阈值“off”和下限阈值“On”。上限阈值“off”用来判断是否停止当前下载，下限阈值“On”用来判断是否需要开启下载。根据该传输方案，当缓存区内的数据量大于或等于上限阈值时，无线接口停止下载并在经历一段时间的 Tail 延迟后自动进入到 IDLE 状态，如图 3-1 中 On-off 传输曲线的虚线所示。当缓存区内的数据小于下限阈值时，无线接口提升到 CONNECT 状态并开始以带宽最高速率下载数据，直到缓存区内的数据量再次大于或等于上限阈值，如图 3-1 中的 On-off 传输曲线的实线所示。随着视频内容的不断播放，On-Off 传输以上述的工作方式不断的开启和关闭无线接口，直到所有的数据下载完毕。与贪婪传输相比，On-Off 传输在缓存区内的数据量大于（或等于）上限阈值时无线网卡会转向休眠状态，因此会节省一部分能耗。但是由于 On-off 传输完全根据缓存区内的数据量来决定无线网卡的开启和关闭而没有考虑到带宽信息，因此很有可能导致无线网卡频繁的在 CONNECT 和 IDLE 之间转换，从而额外的浪费了大量的功率，并没有达到省电的需求。与此同时，相比贪婪传输，On-Off 传输在播放过程中的卡顿时间更长，用户流畅度观看体验质量更差。

综上所述，以上的两种数据传输方式中，贪婪传输十分耗电但用户流畅度观看体验质量最好，On-off 相比贪婪传输节省一部分能量但仍然有浪费且用户流畅度观看体验质量较差。但是从本质上来说，这两种传输机制都是根据缓存区的填充程度来决定下载状态，而没有到考虑无线信道上的带宽信息，由此导致移动设备额外浪费了大量的能量。

因此，如何利用未来带宽信息来正确的安排视频下载任务使得下载期间能够充分利用无线信道上的最佳带宽以最小化传输能耗，同时能够达到同贪婪传输方案相同的用户体验质量是本文接下来要研究的问题。

3.3 面临的挑战

预知未来带宽变化信息使本研究有机会找到最佳视频下载时间以节省传输能耗。为了实现这个目标，一个直观的想法是通过贪婪的方式不断的寻找未来时间里最好的带宽容量对应的时间段，并在这段时间内启动无线接口进行下载，直到整个视频下载完毕。然而，在实际生活中它将面临以下两个挑战。为了便于理解，本文使用图 3-2 来显示上述想法在实际下载过程中遇到的问题。

挑战 1：并非所有的具有高速带宽的时间间隔都适合下载数据。现实生活中移

动终端的缓存区长度是固定有限的，它通常会存放可播放时间从几十秒到几分钟的数据量。因此在利用高速带宽对应的的时间范围内下载数据时，有限的缓存区长度会限制下载速度，导致在这段时间内实际下载的数据量十分有限。如图 3-2 所示，图中标注为“传输方式 1”的曲线即通过不断寻找最好的带宽对应的的时间间隔来下载数据的方式。从图中可以观察到，传输方式 1 总是优先选择好带宽进行下载，然而在很多情况下，缓存区的空余容量有限的且很快被填满，导致后续可下载的数据量有限。图中标注为“传输方式 2”的曲线是本文的理想下载方式，同传输方式 1 相比，传输方式 2 放弃了图中最好的带宽对应的的时间间隔，而是选择了其次的好带宽进行下载，反而在相同时间间隔内下载了更多的数据量。显然的，传输方式 2 的总能耗时间短于传输方式 1。因此，如何在移动端有限的缓存区长度限制下，安排视频下载任务以充分利用带宽来最小化传输能耗是一个问题

挑战 2：播放器一旦开始播放，视频必须以恒定的速率连续播放，播放的连贯性极大的影响了真实生活中移动终端用户的观看体验质量。因此，为了保持下载数据的连续性，除了在一些高速带宽对应的的时间间隔内下载数据之外，还需要在某些适当带宽上找到一些其它的时间间隔来进行下载。如图 3-2 所示，传输方式 1 只选择好带宽下载数据是明显不够的，为了使得缓存区内的数据不被耗尽，还需要像传输方式 2 一样选择一些其它的更合适的带宽进行下载。

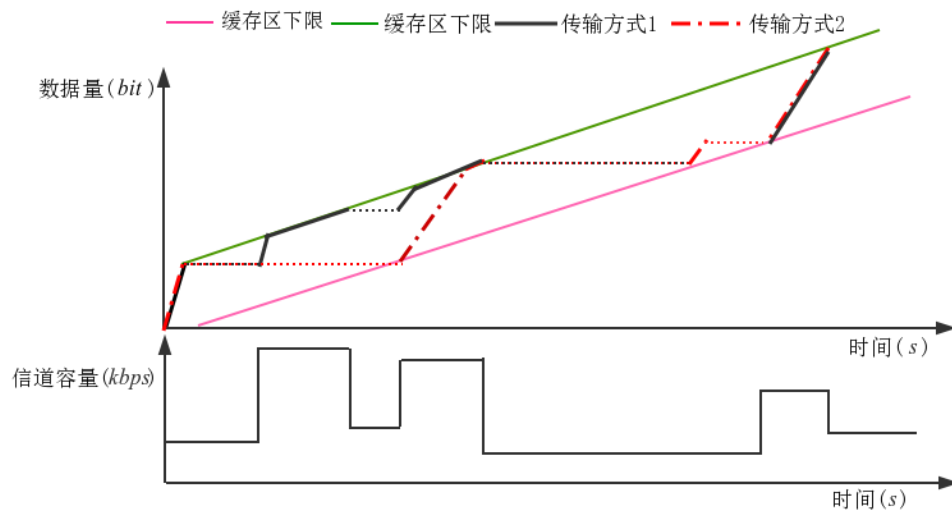


图 3-2 下载面临的挑战

Figure 3-2 Download challenges

为了解决以上挑战，本文首先针对优化目标建立了优化模型，将优化问题表示为一个时间离散的、以最小化传输能耗为目标、以缓存限制为约束的整数线性规划问题，使得在提供给用户最佳的视频播放流畅体验下最小化传输能耗。然后提出了基于最大等效能效传输算法来解决这个问题。为了应对以上两个挑战：首先，该方

法通过寻找实际可传输最大数据量对应的时间间隔来做最佳下载时间，而不是前文中提出的寻找最好的带宽容量对应的时间间隔，有效保证了被选择的下载任务能够最大程度的下载数据量。然后，该算法以贪婪的方式迭代的选择下载任务，使得每个下载任务按照顺序下载一定范围内的数据量，直到将视频所需要的数据全部下载完毕。具体解决思路将在下一章描述。

3.4 移动终端优化传输模型

本节首先对系统进行了参数化描述，然后建立优化模型公式。

3.4.1 参数化系统

首先介绍一个移动终端用户在线观看视频的应用场景。假设要观看的视频长度为有限长度 S byte，播放速率为恒定速率 r byte/s，移动终端具有有限长度为 w byte 的缓存区用以存放视频数据。假设用户从 0 时刻开始观看视频，视频采用贪婪传输的方式进行下载。视频一旦开始播放，移动终端需要不断的从缓存区中取出数据按照顺序进行播放。一旦在某一时刻缓存区内剩余的数据小于 r byte，移动终端停止播放，卡顿发生。直到缓存区重新被填满数据，移动终端才能继续播放。移动终端停止播放的这段时间通常被称为重缓冲段^{[59]1-2}，在整个视频播放期间，可能会出现一个或者多个重缓冲段。

接下来本节分别来介绍在上述的应用场景中无线接口的带宽变化和状态变化。本文假设在视频播放之前已经预先知道无线接口的未来带宽。无线接口的未来带宽可以被定义为一个分段常数函数^{[59]1-2}，如下图 3-3 所示，图中横轴代表时间，纵轴代表信道容量。给定一段时间，该段时间可以被分为 n 个时隙（时间间隔），每个时隙上对应着一个固定值为常数的信道容量。

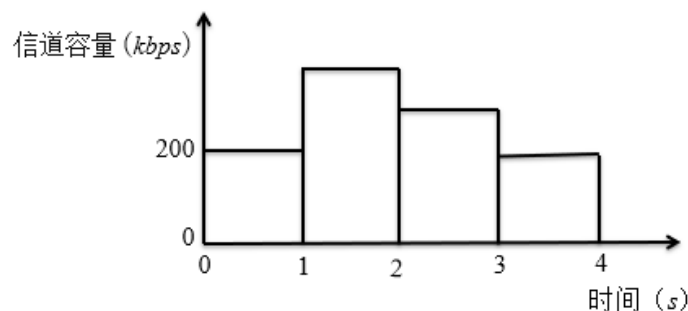


图 3-3 无线信道在四个时隙内的变化^{[59]1-2}

Figure 3-3 The channel variations in four time slots^{[59]1-2}

由此,本文对无线接口的未来带宽有以下假设。假设使用贪婪传输的方式下载整个视频,视频播放时长为 T , T 可以表示为由若干个等长的时隙组成的时间序列 \mathcal{T} , $\mathcal{T}=\{0, 1, 2, \dots, T\}$ 。其中每个时隙的长度为 1 秒。进一步的将每个时隙 i ($i \in \mathcal{T}$) 上对应的无线信道带宽值定义为一个恒定的常数值 c_i , 并使用离散序列 $\mathcal{C} = \{c_i | i \in \mathcal{T}\}$ 来表示视频播放期间无线接口的带宽变化。本文假设 \mathcal{C} 是已知的。

由 4G-LTE 网络下移动设备的耗能模型可知,在整个视频传输过程中移动设备的无线接口一共有四个状态,分别是 CONNECT 状态、IDLE 状态、Promotion 状态和 TAIL 状态。其中,CONNECT 状态、Promotion 状态和 TAIL 状态对应无线接口的高耗能阶段, IDLE 状态对应无线接口的低耗能状态。本文分别定义无线接口在 IDLE 状态(定义为状态 0)、Promotion 状态(定义为状态 1)、CONNECT 状态(定义为状态 2)和 TAIL 状态(定义为状态 3)下每个时隙内(即每秒钟内)消耗的功率分别为 \mathcal{P}_0 、 \mathcal{P}_1 、 \mathcal{P}_2 、 \mathcal{P}_3 。此外,本文假设每进行一次新的数据传输,无线接口在 Promotion 状态和 TAIL 状态分别经历的 Promotion 延迟时长和 Tail 延迟是一个固定值,并将 Promotion 延迟时长定义为 ϵ , Tail 延迟时长定义为 σ 。

综上所述,给定带宽序列 \mathcal{C} ,本章的研究目标是寻找一个二进制视频下载向量 \mathcal{X} , $\mathcal{X} = \{x_i | i \in \mathcal{T}\}$ 。其中每个 x_i 表示无线接口在时隙 i 内是处于开启 ($x_i = 1$) 还是关闭 ($x_i = 0$)。通过安排视频下载任务使得 \mathcal{X} 对应的传输能耗最小,同时能够达到同贪婪传输相同的用户最佳观看体验质量。当无线接口处于开启阶段时可能会处于 Promotion 状态或 CONNECT 状态并消耗与之对应的能量。当无线接口处于关闭阶段时,可能会处于 TAIL 状态或者 IDLE 状态,前者会消耗尾部能耗,后者不消耗能量。在下一小节中,本文将给出传输模型公式。为了查询方便,本节总结了公式中需要用到的符号和上面已经定义的符号到表 3-1 中,如下所示。

表 3-1 符号和相应定义

Table 3-1 Symbols and their definition

符号	定义
r	视频播放速率
w	客户端缓存区长度
S	视频长度
\mathcal{T}	视频播放时间序列
\mathcal{C}	视频播放期间无线信道带宽序列
\mathcal{P}_i	单位时隙内无线接口对应状态所消耗的能量。 \mathcal{P}_1 表示 Promotion 状态消耗的能量, \mathcal{P}_2 表示 CONNECT 状态消耗的能量, \mathcal{P}_3 表示 TAIL 状态消耗的能量

表 3-1 (续表)

符号	定义
ϵ	Promotion 延迟时长
δ	Tail 延迟时长
\mathcal{X}	视频下载向量, $x_i = 1$ 表示无线接口在时隙 i 内开启, $x_i = 0$ 表示无线接口在时隙 i 内关闭
N	总的可优化段数
t_s^n	可优化段 n 的开始时刻
t_e^n	可优化段 n 的结束时刻
s_s^n	可优化段 n 在开始时刻已经下载的数据量
s_e^n	可优化段 n 在结束时刻已经下载的数据量
u_s^n	可优化段 n 在开始时刻已经被播放的数据量
L_i	任务 i 在下载期间的缓存区下限集合
U_i	任务 i 在下载期间的缓存区上限集合
d_i	任务 i 实际下载的总数据量
η_i	任务 i 的等效能效
k_i	任务 i 的等效能耗时间

3.4.2 优化模型

本章优化目标是在保证用户拥有最佳观看体验质量下寻找视频下载算法以最小化传输能耗。给定未来无线带宽序列,通过贪婪传输下载视频即可得到最佳的视频流畅度播放体验^{[59]2-3}。同时,在使用贪婪传输下载视频时,在视频传输期间可能会发生多个重缓冲段和可优化段^{[59]4-5},如图 3-4 所示。图 3-4 中横轴代表时间,纵轴代表数据量。图中标注为“缓存区下限”和“缓存区上限”的曲线分别表示下载上限和播放曲线,位于这两者之间的曲线表示实际下载曲线。实际下载曲线由三部分组成,其中标注为“重缓冲段”的曲线表示重缓冲段。从图中可以看到,这些重缓冲段将整个视频播放区间分成多个可连续播放的时间段。对于每一个连续播放的时间段 $[i, j]$,如果在 $[i + 1, j]$ 时间范围内存在带宽没有被完全利用的时间点,则定义距离 j 最近的一个带宽没有被完全利用的时间点 k ,那么 $[i, k]$ 这段时间被定义为可优化段。一个视频可能包括一个或多个可优化段,除了可优化段之外的其它时间段均属于不可优化段。在不可优化段期间,无线接口始终处于开启状态且带宽已经被充分利用以尽可能的减少视频卡顿现象,无法进一步优化。

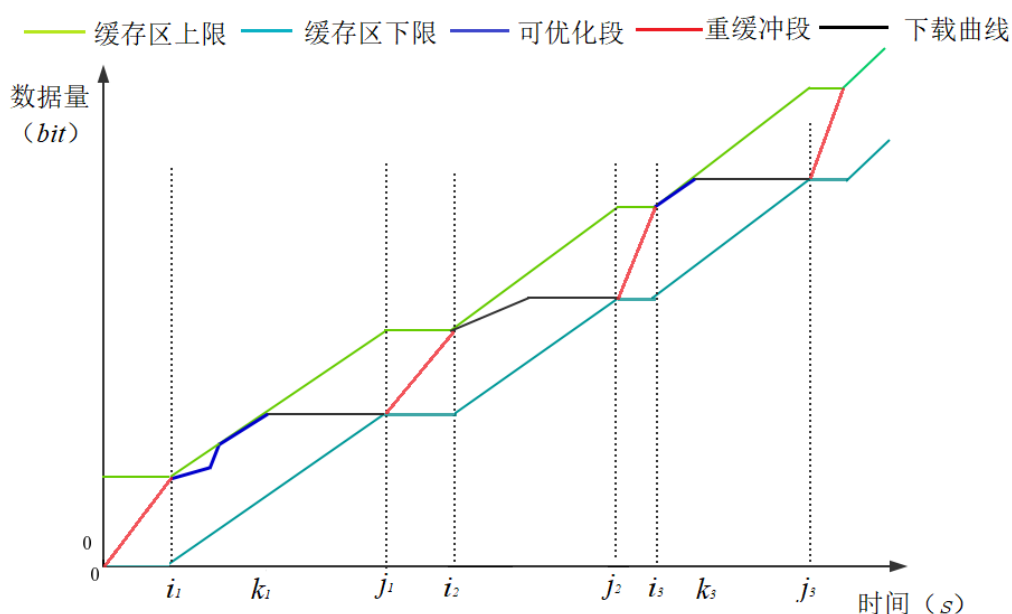


图 3-4 重缓冲段和可优化段

Figure 3-4 Rebuffering and optimizable segment

为了提供给用户同贪婪传输相同的最佳的视频流畅度观看体验，本文接下来只在可优化段内研究视频节能传输方案。研究目标是如何在可优化段内最小化传输能耗并保证视频在可优化段内能够流畅播放。因此，假设视频一共有 N 个可优化段，则本文将寻找整个视频下载向量 \mathcal{X} ($x_i = 1$, 无线接口开启; $x_i = 0$, 无线接口关闭) 的研究目标分解成对每个可优化段 n ($n \in [1, N]$) 寻找其下载向量 X^n 的研究目标，并在其余不可优化段期间内设定 $x_i = 1$ 。下面本文针对可优化段建立节能优化目标。

假设本研究使用贪婪传输的方式下载整个视频并找到 N 个可优化段。对每一个可优化段 n ($n \in [1, N]$)，定义其初始化参数为 $\{t_s^n, s_s^n, t_e^n, s_e^n, u_s^n\}$ 。其中， t_s^n 表示可优化段 n 的开始时间点， s_s^n 表示可优化段 n 在开始时刻已经下载的数据总量， t_e^n 表示可优化段 n 的结束时间点， s_e^n 表示可优化段 n 在结束时刻已经下载的数据总量， u_s^n 表示可优化段 n 在开始时刻已经被播放的数据总量，显然 $u_s^n < s_s^n$ 。对每个可优化段 n ，本章的研究目标是找到相应的下载向量 X^n ($X^n = \{x_i | i \in [t_s^n, t_e^n]\}$)，使得 X^n 具有最小的传输能耗，同时保证视频在该段时间内不会发生卡顿。

给定下载向量 X^n ，其传输能耗可以由下述的方法进行推导。

首先，本文定义 X^n 对应的无线接口的状态为 Y^n ， $Y^n = \{y_i | i \in [t_s^n, t_e^n - 1]\}$ ，其中 $y_i = 0, 1, 2, 3$ 分别表示无线接口在 i 时隙内处于 IDLE 状态、Promotion 状态、CONNECT 状态和 TAIL 状态。由 X^n 推导状态 Y^n 的方法如下： i 时刻无线接口的状态 y_i 由 x_i ($x_i = 1$, 无线接口开启; $x_i = 0$, 无线接口关闭) 和 y_{i-1} 决定，

由公式 3-1 可推导得出。

$$y_i = f(x_i, y_{i-1}) \quad (3-1)$$

具体推导过程见本章 3.5 节。

已知 Y^n 后，本文将 X^n 对应的传输能耗值定义为 $F(X^n)$ ，用来表示可优化段 n 的传输能耗。 $F(X^n)$ 可以用公式 3-2 表示：

$$F(X^n) = \sum_{\substack{\forall i \in [t_s^n, t_e^n - 1] \\ y_i \in \{0,1,2,3\}}} P_j I^{y_i=j} \quad (3-2)$$

其中， P_j 是无线接口处于状态 j 时单位时隙内消耗的能量， $I^{y_i=j}$ 是一个取值为 0 和 1 的二值函数，分别代表在无线接口当前状态是否耗能。

接下来，本节介绍在可优化段内下载数据需要满足的约束条件。由上文可知，在视频下载过程中，在任意时刻 $i (i \in [t_s^n, t_e^n])$ 无线接口实际能够下载的数据总量被限制在一定的范围内，分别是缓存区上限和缓存区下限。本文将 i 时刻的缓存区上限定义为 U_i ，缓存区下限定义为 L_i 。 U_i 和 L_i 分别可用公式 3-3 和公式 3-4 来表示：

$$U_i = \min(r * (i - t_s^n + 1) + u_s^n + w, s_e^n) \quad t_s^n \leq i \leq t_e^n \quad (3-3)$$

其中， i 时刻的缓存区上限受到缓存区大小 w 和可优化段内需要下载的总数据量 s_e^n 的限制。

$$L_i = \max(r * (i - t_s^n + 1) + u_s^n, s_s^n) \quad (3-4)$$

其中， i 时刻的缓存区下限受到 i 时刻已经被播放的数据量和初始时刻 t_s^n 已经下载的数据量 s_s^n 的限制。

因此本文将 i 时刻实际能够下载的数据总量定义为 d_i ， d_i 可以由公式 3-5 表示：

$$d_i = \min(d_{i-1} + c_i I^{y_i=2}, U_i) \quad (3-5)$$

其中初始值有 $d_{t_s^n} = s_s^n$ 。在下载过程中如果下载的数据量没有超过缓存区上限，则无线接口继续以带宽最高速率下载，否则取当前时刻上限值作为当前下载值。

综上所述，根据上述推导的目标函数和限制条件，本文为可优化段 n 建立如下的视频传输模型，如公式 3-6 所示，其中自变量是下载向量 X^n ，约束限制是当前时刻下载的数据总量不能超过缓存区上限且不能低于播放下限，优化目标是 minimized 传输能耗且保证用户在可优化段内能够流畅的观看视频。

$$\begin{aligned} & \text{Minimize}_{X^n} && F(X^n) && (3-6) \\ & \text{Subject to} && d_i \geq L_i && \forall i \in [t_s^n, t_e^n] \\ & && d_i \leq U_i && \forall i \in [t_s^n, t_e^n] \\ & && y_i = f(x_i, y_{i-1}) \\ & && x_i \in \{0,1\} \\ & && y_i \in \{0,1,2,3\} \end{aligned}$$

作为整数规划问题，公式 (3-6) 表示的优化问题难以解决，本文将在下一章

中提出一个启发式算法来解决它。

最后,本节评估上述优化问题的计算复杂度。假设使用贪婪传输下载视频一共找到 N 个可优化段,这 N 个可优化段一共有 M 个时隙,对于每个时隙,无线接口有两种状态,开启或者关闭,研究目标是通过寻找每个时隙内无线接口的状态,使得总传输能耗最小,因此很容易推导出该优化问题的计算复杂度为 2^M 。

3.5 相关推导

本节介绍由下载向量 X^n 推导无线接口状态 Y^n 具体方法。首先根据文献[60],给出 RRC 状态转换机制,如下图 3-5 所示。

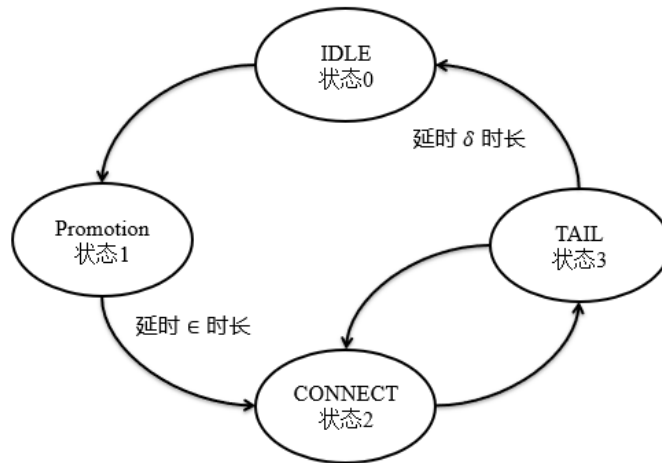


图 3-5 RRC 协议对应状态转换^[60]

Figure 3-5 The RRC state transition^[60]

具体来说,给定 x_i ($x_i = 1$, 无线接口开启; $x_i = 0$, 无线接口关闭) 的值,根据 y_{i-1} 的值 y_i ($y_i = 0, 1, 2, 3$ 分别表示无线接口在 i 时隙内处于 IDLE 状态、Promotion 状态、CONNECT 状态和 TAIL 状态) 有以下几种结果。

给定 $x_i = 0$, 根据 y_{i-1} 的值 y_i 结果如下:

(1) 如果 $y_{i-1} = 0$, 则 $y_i = 0$ 。即如果当前时刻无线接口关闭且前一时刻处于 IDLE 状态,则当前时刻继续处于 IDLE 状态。

(2) 如果 $y_{i-1} = 3$, 若当前时刻没有需要传输的数据,则根据 TAIL 状态已经持续的时长判断 y_i 的值,如果持续时长小于 δ 则 $y_{i-1} = 3$, 否则 $y_i = 0$ 。

(3) 如果 $y_{i-1} = 1$, 则根据 Promotion 状态已经持续的时长判断 y_i 的值,若持续时长小于 ϵ 则 $y_i = 1$, 否则 $y_i = 2$ 。

给定 $x_i = 1$, 根据 y_{i-1} 的值 y_i 结果如下:

(1) 如果 $y_{i-1} = 2$, 则 $y_i = 1$ 。即如果当前无线接口开启其前一时刻处于

CONNECT 状态，则当前时刻继续处于 CONNECT 状态。

(2) 如果 $y_{i-1} = 1$ ，则根据 Promotion 状态已经持续的时长判断 y_i 的值，若持续时长小于 ϵ 则 $y_i = 1$ ，否则 $y_i = 2$ 。

(3) 如果 $y_{i-1} = 3$ ，则无论需要传输多小的数据量，都有 $y_i = 2$ 。

(4) 如果 $y_{i-1} = 0$ ，则 $y_i = 1$ 。

3.6 本章小结

本章建立了移动终端视频优化传输模型，模型的优化目标是在保证具有和贪婪传输相同的用户观看体验质量下，优化视频传输方案以最小化传输能耗。首先本节介绍了应用场景和研究动机，然后建立了视频传输模型并给出了视频传输模型公式。

4 视频下载算法设计和实现

本章将详细介绍本文提出的两个视频下载传输算法。第一个是针对上一节中的优化目标提出的基于最大等效能效传输算法。第二个是在第一个算法的基础上考虑了缓存区的填充程度，提出的重缓冲过程的动态缓存门限算法。

本章的结构安排如下：4.1 介绍基于最大等效能效传输算法；4.2 对重缓冲过程的动态缓存门限算法进行介绍；4.3 为相关推导；4.4 为本章小结。

4.1 基于最大等效能效传输算法

本节介绍了基于最大等效能效传输的视频下载算法。首先，介绍算法的提出背景和面临的挑战，然后阐述了算法的基本思想并对算法进行了概述，最后介绍了算法的具体实现。

4.1.1 问题提出和挑战

针对上一章中的优化模型，本文提出了基于最大等效能效传输算法，目标在提供给用户同贪婪传输相同的最佳用户观看质量下，最小化传输能耗。具体来说，即对每个可优化段 n ($n \in [1, N]$) 寻找其下载向量 X^n ，通过决定可优化期间每个时隙的 x_i ($x_i=1$ ，无线接口开启； $x_i=0$ ，无线接口关闭) 的值来最小化传输能耗。在寻找过程中，该算法将面临以下挑战：由于可优化段内时隙数量较多，如果将每个时隙作为算法的基本调度单元则随着时隙数目的增加计算复杂度呈指数上升，将变得十分困难。

4.1.2 基本思想

为了解决上述挑战，本文通过合并时隙的方法来降低计算复杂度。具体来说，本文首先将多个相邻的时隙合并在一起形成一个下载任务，将一个下载任务而不是一个时隙作为算法的基本调度单元以减少调度量，降低算法复杂度。然后利用等效能效（即单位能耗时间内下载的数据量）作为评估每个下载任务的下载能力的指标，最后通过不断的选择具有最大等效能效的下载任务来完成整个可优化段的下载。具体思想如下：

- (1) 首先使用贪婪传输下载整个视频，识别可优化段，对可优化段进行优化。

(2) 将每个可优化段的下载调度为一系列下载任务，这些任务以贪婪的方式在所有候选任务中迭代选择，使每个选择的任务具有最大的等效能效，直到全部数据下载完成。

在上述方法中，第一步通过贪婪传输的方法寻找可优化段是现有研究中已有的方法。第二步在可优化段内实施基于最大等效能效传输的下载方法是本文提出的创新方法。

4.1.3 算法概述

基于最大等效能效传输的下载算法在可优化段内以贪婪的方式生成下载向量以节省传输能耗。具体来说，每个可优化段都有相应的下载时间范围和在该段时间内需要下载的数据量范围。给定一个可优化段，它将该可优化段的下载需求安排为一系列下载任务。这些任务以贪婪的方式在所有的候选任务中迭代选择。每个候选任务（即还没有被下载的时间点）以无线接口带宽的最高速率在需求的数据量范围内下载，直到满足一定的条件（例如达到缓存上限或者完成下载）停止下载。然后选择具有最大的等效能效（即每单位能耗可传输最大的数据量）的候选任务作为当前下载任务并同时更新候选任务集。这些被选择的任务将占用一定的时间下载一定范围内的连续视频数据。直到可优化段内所有数据下载完成。

4.1.4 算法设计和实现

本节将详细介绍基于最大等效能效传输算法的实现过程。首先介绍下载任务和等效能效的定义，然后介绍算法的具体实现。

(1) 下载任务

首先介绍算法实现的基本调度单元：下载任务（即任务）。每个可优化段都对应着一个初始的下载时间范围，以及在这段时间需要下载的数据量范围。对于每一个可优化段，算法通过调用一系列下载任务来满足其下载需求。下载任务的初始化过程如下：对于每个下载任务 i ，定义其初始化参数分别为 $(l_s^i, l_o^i, \zeta_s^i, \{u_t | t \in [l_s^i+1, l_o^i]\}, \{l_t | t \in [l_s^i+1, l_o^i]\})$ 。其中 l_s^i 表示任务 i 的开始时间点； l_o^i 表示任务 i 的结束时间点； ζ_s^i 表示任务 i 在开始时刻 l_s^i 已经下载的数据总量； u_t 和 l_t 分别是任务 i 在 $[l_s^i, l_o^i]$ 时间范围内的缓存区上限集合和缓存区下限集合。此外，本文使用符号 (l_e^i, ζ_e^i) 分别来表示下载任务 i 的实际下载停止时间点和在实际停止时间点下载的数据总量。

下载任务 i 是按照以下的方式进行下载的。任务 i 从 l_s^i 开始以无线接口带宽的

最高速率依次下载数据，一旦满足以下任意一个条件，下载停止，停止时间点为 l_e^i 。

- 1) 在某一时刻 t ($t \leq l_o^i$) 下载达到缓存区上限，此时有 $l_e^i = t$, $\zeta_e^i = u_t$ 。
- 2) 在某一时刻 t ($t \leq l_o^i$) 下载达到缓存区下限，此时有 $l_e^i = t$, $\zeta_e^i = l_t$ 。
- 3) 下载时间到达 l_o^i ，此时有 $l_e^i = l_o^i$ 。

下载停止时，根据停止条件判定是否下载成功。当由于条件 1 缓存区发生上溢停止时，下载成功。当由于条件 2 缓存区发生下溢停止时，表明当前时刻视频发生卡顿，由于可优化段内不允许有卡顿现象，判定下载失败。当由于条件 3 达到结束时间停止时，同样判定下载失败。对于下载失败的任务，本文用 $\zeta_e^i = \zeta_s^i$ 来表示该任务无法下载任何数据。

给定一个可优化段 n ，其初始的时间范围为 $[t_s^n, t_e^n]$ ，对应的需要下载的数据量范围为 $[s_s^n, s_e^n]$ 。在对可优化段 n 进行优化时，算法将调用一系列的下载任务，每个下载任务的初始时间点分别是 t_s^n 到 $t_e^n - 1$ ，间隔为 1 秒。这一系列下载任务将全部作为候选任务按照上述下载方式进行下载，当所有候选任务下载完成后。算法将迭代的从下载成功的候选任务中选择具有最大等效能效的任务作为当前下载任务。当前下载任务占用了特定的时间下载了一定范围内的数据量，这部分时间和数据量会作为已经下载完成的部分从候选任务中去除。候选任务只需要在剩余的时间范围内完成剩余数据量的下载。候选任务集更新成功后，算法将会继续按照以上的方式迭代的选择具有最大等效能效的下载任务，直到所有数据全部下载完毕。

(2) 等效能效

接下来，本节将给出等效能效的定义。本文将等效能效定义为 η ，并用它作为衡量候选任务下载能力的指标。

给定一个候选任务 i ，其对应的下载的数据量的范围为 $[\zeta_s^i, \zeta_e^i]$ 。假设这个任务消耗的能耗为 k_i 。则该候选任务 i 的能效为：

$$\eta = (\zeta_e^i - \zeta_s^i) / (k_i) \quad (4-1)$$

其中， k_i 是无线接口执行下载任务 i 所消耗的功率， k_i 的值可以由公式 4-2 计算得出：

$$k_i = \mathcal{F}(\mathcal{X} + \{x_i | x_i = 1, i \in [l_s^i, l_e^i - 1]\}) - \mathcal{F}(\mathcal{X}) \quad (4-2)$$

由于此公式的计算方法太过于复杂，因此本文使用以下近似方法以简化计算。

由前文可知，无线接口的能量消耗是由 CONNECT 状态，TAIL 状态和 Promotion 状态所引起的。本研究忽略 Promotion 状态消耗的能量，并忽略 CONNECT 状态和 TAIL 状态消耗能量的差异，使用等效能耗时间来近似的代替上式中的等效能效。一个候选任务的等效能耗时间是由该候选任务所引起的无线接口额外的耗能时间。由此，本文重新定义 k_i 为：

$$k_i = \max(o(1), l_e^i - l_s^i + \min(l_s^i - t_{pre} - \sigma, 0) + \min(t_{pro} - l_e^i, \delta)) \quad (4-3)$$

其中, t_{pre} 表示无线接口在候选任务 i 开始之前最近的一个开启时间点, 即 $t_{pre} = \max(j | j < l_s^i, x_i = 1)$; t_{pro} 表示无线接口在候选任务 i 结束之后最早的一个开启时间点, 即 $t_{pro} = \min(j | j > l_e^i, x_i = 1)$; $o(1)$ 表示两个连续时间点之间的最小时间间隔。 k_i 的具体推导过程见本章 4.3 节。

最后, 本文将上述定义的符号和内容汇总到算法 1 中, 给出等效能效的计算过程。等效能效是用来评估候选任务下载能力的指标。评估方法是给定一个候选任务, 从候选任务的开始时刻以无线接口的最高带宽速率下载数据, 直到满足任意一个停止条件结束下载, 若下载成功则计算该候选任务的等效能效, 若下载失败则从候选任务集中删除该候选任务。

具体实现如表 4-1 所示。首先, 对每个候选任务 i , 第一行初始化变量, 第二行从 l_s^i 时刻开始每秒钟以无线接口最高带宽速率下载数据, 直到满足第 4 行的停止条件 1 达到缓存区上限, 由第 5 行更新参数 (l_e^i, ζ_e^i) 为 (j, u_j) 并在第 6 行停止下载退出循环; 或者满足第 7 行的停止条件 2 达到缓存区下限, 判定候选任务下载失败由第 8 行停止下载退出循环。当下载时间达到结束时间 l_o^i , 同样判断失败由第 10 行退出循环。候选任务 i 下载完成后, 第 11 行利用公式 4-1 和公式 4-3 计算该候选任务的等效能效。最后输出该候选任务的实际停止时刻 l_e^i 、在实际停止时刻下载的数据量 ζ_e^i 和等效能效, 评估完成。

表 4-1 下载任务生成算法

Table 4-1 The job computing algorithm

Algorithm 1	
Input: $l_s^i, l_o^i, \zeta_s^i, \{u_t\}, \{l_t\}, \mathcal{C}$	
1:	$v = \zeta_s^i, \zeta_e^i = \zeta_s^i, l_e^i = l_o^i$
2:	for all $j \in [l_s^i + 1, l_o^i]$ do
3:	$v = v + c_{j-1}$
4:	if $v \geq u_j$ then
5:	$l_e^i = j, \zeta_e^i = u_j$
6:	break
7:	else if $v < l_j$
8:	break
9:	end if
10:	end for
11:	利用公式 4-1 和公式 4-3 计算 η_i
Output: $\{l_e^i, \zeta_e^i, \eta_i\}$	

(3) 具体实现

最后介绍算法具体实现过程。给定一个可优化段 n ，算法将在该可优化段内调用一系列候选任务，并以迭代的方式选择具有最大等效能效的候选任务作为当前下载任务，同时更新候选任务集，直到所有数据全部下载完毕。具体过程如表 4-2 所示，首先，对每个可优化段 n ，第 1 行到第 5 行初始化可优化段的相关参数。其中，第 1 行初始化空集 J 作为任务集， J 作为当前候选任务集， J' 复制 J 以记录要计算的候选任务。第 2 行对候选任务集中的每一个候选任务设置其初始化参数，包括开始时刻 l_s^i 、开始时刻已经下载的数据总量 ζ_s^i 、实际下载停止时刻 l_e^i 、在实际下载停止时刻下载的数据量 ζ_e^i 、结束时刻 l_o^i 。第 3 行和第 4 行根据公式 3-3 和公式 3-4 初始化可优化段 n 在 $[t_s^n, t_e^n]$ 时间范围内的下载上限集和下载下限集。第 5 行初始化 \mathcal{X} 。第 6 行设定可优化段 n 的下载结束条件。接下来，第 7 行到第 24 行在给定的可优化段内以迭代的方式选择具有最大等效能效的候选任务作为当前下载任务，直到所有的数据下载完成。具体来说，第 8 行对所有的候选任务计算其等效能效。第 9 行选择具有最大等效能效的候选任务。如果由第 10 行判定该候选任务成功下载了一定的数据量，则第 11 行到第 20 行算法将该候选任务作为当前下载任务，并更新剩余的候选任务集。其中，第 11 行更新下载向量并在第 12 行将候选任务 θ 添加到任务集 J 中，并更新仍需下载的数据量 v ；第 13 行和第 14 行分别去除候选任务集中下载的时间范围和数据量范围与 θ 任务发生重合的候选任务。然后由第 15 行到第 18 行更新 J' 中被 θ 任务影响的候选任务。其中第 15 行和第 16 行将 θ 任务之后的候选任务添加到 J' 中并调整其起始数据量 ζ_s^i 为 ζ_e^θ ，第 17 行将 θ 任务之前的候选任务添加到 J' 中并调整其终止时间点为 $\min(l_s^\theta, l_o^i)$ 。之后由第 19 行到第 20 行更新下载上限和下载下限集合。其中，第 19 行调整 θ 任务之前的下载上限集合，第 20 行调整 θ 任务之后的下载下限集合。如果候选任务下载失败，由第 21 行到第 23 行退出循环。最后，下载完成后，输出下载向量 \mathcal{X} ，算法结束。

表 4-2 基于最大能效能耗传输算法

Table 4-2 The maximum equivalent energy efficiency downloading algorithm

Algorithm 2

Input: $t_s^n, t_e^n, s_s^n, s_e^n, u_s^n, \mathcal{X}, \mathcal{C}, r, w$

- 1: $J = \{\}, J' = \{i | i \in [0, t_e^n - t_s^n - 1]\}, J' = J$
- 2: $\{l_s^i, \zeta_s^i, l_e^i, \zeta_e^i, l_o^i\} = \{t_s^n + i, \max(L_{t_s^n+i}, s_s^n), 0, 0, t_e^n\}, \forall i \in J$
- 3: $\{l_t\} = \{L_t | t \in [t_s^n + 1, t_e^n]\}$
- 4: $\{u_t\} = \{U_t | t \in [t_s^n + 1, t_e^n]\}$
- 5: $x_i = 0, \forall i \in [t_s^n, t_e^n - 1]$

表 4-2 (续表)

Algorithm 2

```

6:  $v = s_e^n - s_s^n$ 
7: while  $v > 0$  do
8:    $\{l_e^i, \zeta_s^i, \eta_i\} = \text{Algorithm 1}(l_s^i, l_o^i, \zeta_s^i, \{u_t\}, \{l_t\}, \mathcal{C}), \forall i \in J'$ 
9:    $\theta = \max_{i \in J}(\eta_i)$ 
10:  if  $\theta \neq 0$  then
11:     $x_i = 1, \forall i \in [l_s^\theta, l_e^\theta - 1]$ 
12:     $J = J + \{\theta\}, v = v - \zeta_e^\theta + \zeta_s^\theta$ 
13:     $J = J \setminus \{i | l_s^i \in [l_s^\theta, l_e^\theta - 1]\}$ 
14:     $J = J \setminus \{i | \zeta_s^i \geq \zeta_s^\theta, u_{l_i} \leq \zeta_e^\theta\}$ 
15:     $J' = \{i | i \in J, i > \theta, \zeta_s^i < \zeta_e^\theta\}$ 
16:     $\zeta_s^i = \zeta_e^\theta, \forall i \in J'$ 
17:     $J' = J' + \{i | i \in J, \zeta_e^i > \zeta_s^\theta \text{ or } l_e^i > l_s^\theta\}$ 
18:     $l_o^i = \min(l_s^\theta, l_o^i), \forall i \in \{i | i \in J, i < \theta\}$ 
19:     $u_t = \min(\zeta_s^\theta, u_t), \forall t \in \{t | t_s^n + 1 \leq t \leq t_s^n + \theta\}$ 
20:     $l_t = \max(\zeta_e^\theta, l_t), \forall t \in \{t | t_s^n + \theta \leq t \leq t_e^n\}$ 
21:  else
22:     $\mathcal{X} = \{0, 0, 0, \dots, 0\}$ 
23:    break
24:  end while

```

Output: \mathcal{X}

进一步的, 本文将基于最大等效能效传输的下载算法应用于下载整个视频。具体实现如表 4-3 所示, 假设要观看的视频长度为 S , 第一行算法采用贪婪传输的方法下载整个视频并找到可优化段。具体来说, 假设使用贪婪传输下载整个视频需要 T 时间, 算法以 1 秒钟为时间间隔将时长 T 分成观看序列 $\mathcal{T}, \mathcal{T} = \{0, 1, \dots, T-1\}$ 。首先算法在观看序列 \mathcal{T} 中寻找可连续播放的时间段, 假设一共找到 \hat{N} 个这样的时间段, 并对其按照先后顺序进行编号。对每个连续播放的时间段 $i (i \in [1, \hat{N}])$ 初始化其起止时间为 $[\hat{t}_s^i, \hat{t}_e^i]$ 。接下来算法在时间段 i 中寻找可优化段。对每个连续播放的时间段 i , 如果存在一个时间点 $\hat{t}_e^i = \max(k | k \in [t_s^i + 1, t_e^i])$, 在这一点带宽没有完全被利用, 则时间段 $[\hat{t}_s^i, \hat{t}_e^i]$ 为可优化段。算法将 $[\hat{t}_s^i, \hat{t}_e^i]$ 添加到与初始化参数组 $\{t_s^n, t_e^n, s_s^n, s_e^n, u_s^n\}$ 相关联的可优化段集合中。其中, $t_s^n = \hat{t}_s^i, t_e^n = \hat{t}_e^i, s_s^n$ 和 s_e^n 分别是贪婪传输在 t_s^n 和 t_e^n 时刻已经下载的累积数据总量, u_s^n 是贪婪传输在 t_s^n 时刻已

经播放的累积数据总量。假设一个找到 N 个可优化段，编号分别为从 1 到 N 。然后，第二行将下载集合 \mathcal{X} 初始化为 $\{0,0,\dots,0\}$ 。第三行调用算法 2 基于最大等效能效传输的下载算法为每个可优化段更新下载向量 \mathcal{X} 并输出最终结果。由此，整个视频下载完毕。

表 4-3 基于最大等效能效传输的视频下载算法

Table 4-3 The video downloading algorithm of maximum equivalent energy efficiency

Algorithm 3
Input: $\mathcal{C}, \mathcal{S}, r, w$
1: 假设使用贪婪传输下载整个视频的时间为 T ，下载完成后一共找到 N 个可优化段，每个可优化段被记为 $\{t_s^n, t_e^n, s_s^n, s_e^n, u_s^n\}$, $\forall n \in [1, N]$
2: $\mathcal{X} = \{0,0,0, \dots, 0\}$
3: $\mathcal{X} = \mathbf{Algorithm\ 2}(t_s^n, t_e^n, s_s^n, s_e^n, u_s^n, \mathcal{X}, \mathcal{C}, r, w) \forall n \in [1, N]$
Output: \mathcal{X}

4.2 重缓冲过程的动态缓存门限算法

本节介绍了重缓冲过程的动态缓存门限算法。首先，介绍了算法的提出背景和面临的挑战，然后阐述了算法的基本思想并对算法进行了概述，最后介绍了算法的具体实现。

4.2.1 问题提出和挑战

在上一节中提出的基于最大等效能效传输的下载算法在使用贪婪传输寻找可优化段时，每个可优化段开始之前都会经历一段重缓冲时间。由于贪婪传输机制在出现重缓冲段时总是等到缓存区内的数据全部填满才能继续播放。因此可优化段在初始时刻缓存区内的数据量总是满的。这也意味着重缓冲段这段期间并不在上一节的优化范围内，如果能将这段时间考虑在内，通过改变重缓冲段的结束时刻来进一步优化缓存区填充程度，然后调用上一节中的视频下载算法来下载数据，应该能够获取更好的性能。基于这个想法，本文提出了重缓冲过程的动态缓存门限算法来进一步的优化每个重缓冲段

在对重缓冲段进行优化时算法面临以下挑战：一个视频可能包含一个或者多个可优化段，每个可优化段开始时刻的前一秒即一个重缓冲段的结束时刻。如图 4-1 所示。

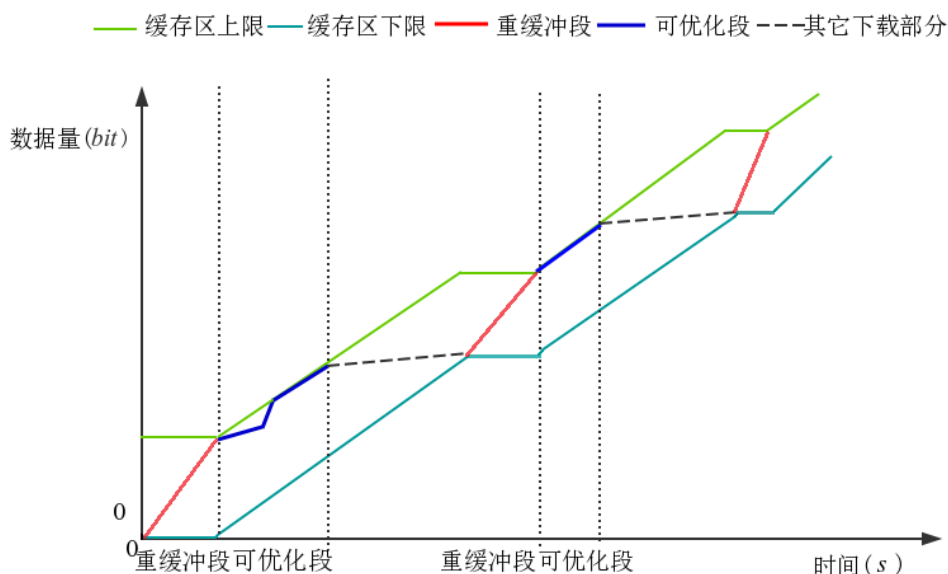


图 4-1 重缓冲段和可优化段

Figure 4-1 Rebuffering segment and optimizable segment

与此同时，重缓冲段之间是相互影响的。当尝试改变第一个重缓冲段的结束时刻以调整可优化段开始时刻缓存区填充程度时，可优化段结束时刻也随着改变，由此直接影响了下一个重缓冲段的开始时刻。因此，在对重缓冲段进行优化时，虽然能够在第一个重缓冲段内取得最好的优化结果。但是无法保证后续受到影响后的可优化段仍然能够取得之前的优化性能。

4.2.2 基本思想

为了解决上述难题，本文提出了重缓冲过程的动态缓存门限算法。基本思想是以现有播放连续性为底线，保持相邻重缓冲的后一个重缓冲过程的起始时刻不变，尝试改变前一个重缓冲过程的终止时刻，直到找到最小的重缓冲时刻，这样使得视频下载过程中所有重缓冲段之间仍然是相互独立不受干扰的，然后分别对每个重缓冲段进行优化。具体实现过程如下：首先使用贪婪传输寻找重缓冲段，对于每一个重缓冲段，算法以 1 秒钟为时间间隔依次提前结束重缓冲段的结束时刻，同时保持下一个重缓冲段的起始时刻不变，然后在紧邻的可优化段内调用上节中的基于最大等效能效算法来下载视频，如果不发生卡顿，则继续向前改变，直到找到最小的重缓冲结束时刻。

下一小节将介绍算法的具体实现，本文将实现过程中需要用到的一些符号定义总结到表 4-4 中，如下所示。

表 4-4 符号和相关定义

Table 4-4 Symbols and their definition

符号	定义
\widehat{t}_s^n	重缓冲段 n 的开始时间点
\widehat{t}_e^n	重缓冲段 n 的结束时间点
\widehat{s}_s^n	重缓冲段 n 在开始时刻 \widehat{t}_s^n 已经下载的数据总量
\widehat{s}_e^n	重缓冲段 n 在结束时刻 \widehat{t}_e^n 已经下载的数据总量
\widehat{u}_s^n	重缓冲段 n 在开始时刻 \widehat{t}_s^n 已经播放的数据总量

4.2.3 算法概述

重缓冲过程的动态缓存门限算法以现有播放连续性为底线，通过尝试改变重缓冲段结束时刻来进一步减少视频卡顿时间。具体来说，一个视频可能包括一个或者多个重缓冲段，每个重缓冲段都有相应的下载时间范围和在这段时间内下载的数据量范围。给定一个重缓冲段，它在保持后一个重缓冲段起始时刻不变的前提下，不断的以一秒钟为间隔提前结束卡顿，如果提前一秒钟结束卡顿不会带来后续额外的视频卡顿，则继续提前，直到找到该重缓冲段最小的结束时刻。

4.2.4 算法设计和实现

本节提出的算法应用场景和上一节中基于最大等效能效传输算法的应用场景相同。即假设一个用户正在通过无线信道观看移动设备上的在线视频，无线信道带宽 C 是已知的，用户从 0 时刻开始观看，此时缓存区为空，视频播放速率为 r byte/s，移动设备具有固定长度为 w byte 的缓存区。接下来介绍具体的算法实现过程。

具体实现如表 4-5 所示。第 1 行使用贪婪传输找到可优化段和重缓冲段并对其初始化。对于每对紧邻的重缓冲段和可优化段，由于重缓冲段期间视频停止播放，因此在 $[\widehat{t}_s^n, t_s^n]$ 时间范围内已经播放的视频总数据量是一个不变的值，即 $\widehat{u}_s^n = u_s^n$ 。第 2 行到第 11 行改变重缓冲段的结束时刻，即改变可优化段的开始时刻。给定一个可优化段 n ，其初始时间范围为 $[t_s^n, t_e^n]$ ，对应的数据量范围 $[s_s^n, s_e^n]$ ，算法逐步将可优化段 n 的下载时间范围扩展到 $[\widehat{t}_s^n, t_e^n]$ ，并改变其初始时间对应的下载数据量 s_s^n 。其中，第 2 行设置可优化段 n 的开始下载时间范围为 $[\widehat{t}_s^n + 1, t_s^n]$ ；第 3 行到 11 行设置可优化段 n 在 $[\widehat{t}_s^n + 1, t_s^n]$ 时间范围内对应的已经下载的总数据量范围。第 12 行到第 20 行算法以一秒钟为间隔不断的向前扩展可优化段的初始点。其中，第 12 行进入 for 循环；第 13 行扩展可优化段的开始时间点；第 14 行初始化下载向量

\mathcal{X} , 第 15 行在扩展后的可优化段内调用基于最大等效能效算法; 如果能够下载成功而不带来额外的卡顿则循环继续, 否则第 16 行到第 20 行退出循环, 并在第 17 行设置可优化段 n 的最佳开始时间点为发生卡顿之前的最近的一个时间点。最后, 输出 \mathcal{X} , 整个算法结束。

表 4-5 重缓冲过程的动态缓存门限的视频下载算法

Table 4-5 The video downloading algorithm of dynamic cache threshold algorithm for rebuffering process

Algorithm 4**Input:** $\mathcal{C}, \mathcal{S}, r, w$

- 1: 假设使用贪婪传输下载整个视频, 一共找到 N 个重缓冲段。对每个重缓冲段记为 $\{\widehat{t}_s^n, \widehat{t}_e^n, \widehat{s}_s^n, \widehat{s}_e^n, \widehat{u}_s^n\}$, 紧邻的可优化段记为 $\{t_s^n, t_e^n, s_s^n, s_e^n, u_s^n\}$, $\forall n \in [1, N]$ 。其中 $t_s^n = \widehat{t}_s^n + 1$, $u_s^n = u_s^n$
- 2: $B = \{k | k \in [0, t_s^n - \widehat{t}_s^n - 1]\}$
- 3: $g_t = \{\}$
- 4: **for each** k **in** B **do**
- 5: **if** $k == 0$ **then**
- 6: $g_{t_s^n - k} = s_s^n$
- 7: **else if** $k == 1$ **then**
- 8: $g_{t_s^n - k} = \widehat{s}_s^n$
- 9: **else**
- 10: $g_{t_s^n - k} = g_{t_s^n - k + 1} - c_{t_s^n - k - 1}$
- 11: **end for**
- 12: **for each** k **in** B **do**
- 13: $t_s^n = t_s^n - k$
- 14: $\mathcal{X} = \{0, 0, 0, \dots, 0\}$
- 15: $\mathcal{X} = \text{Algorithm 2}(t_s^n, t_e^n, g_{t_s^n - k}, s_e^n, u_s^n, \mathcal{X}, \mathcal{C}, r, w) \quad \forall n \in [1, N]$
- 16: **if** $\mathcal{X} \neq \{0, 0, 0, \dots, 0\}$ **then**
- 17: $t_s^n = t_s^n - k + 1$
- 18: $s_s^n = g_{t_s^n - k + 1}$
- 19: **break**
- 20: **end if**
- 21: **end for**

Output: \mathcal{X}

4.3 相关推导

本节介绍等效能耗时间 k_i 的具体推导过程。一个候选任务 i 的等效能耗时间定义为该候选任务引起的额外的无线接口的耗能时间。由上文可知，处于 IDLE 状态的无线接口进行一次连续的数据传输时，需要经历的状态变化为 IDLE \rightarrow Promotion \rightarrow CONNECT \rightarrow TAIL \rightarrow IDLE。本文将无线接口在每个状态消耗的时间分别定义为 t_{Idle} 、 $t_{Promotion}$ 、 $t_{Connect}$ 、 t_{Tail} ，并忽略无线接口在 IDLE 和 Promotion 状态消耗的能量，则传输一次数据需要的总耗能时间为 $t_{Connect} + t_{Tail}$ 。在这里，算法每调用一次候选任务 i 相当于传输一次数据。当调用候选任务 i 时，若无线接口在调用前处于 IDLE 状态且传输完毕后恢复到 IDLE 状态，则该候选任务的等效耗能时间同上述所示。但是当无线接口在调用前处于 TAIL 状态或者传输完毕后在 Tail 延迟未结束的状态下又转向 Promotion 状态，则由该候选任务引起的等效能效时间会发生变化。即每个候选任务 i 的等效能耗时间都和开始时刻无线接口的状态和结束时刻无线接口的状态有关系。进一步的，和前一个下载任务（如果有）的结束时间点和后一个下载任务（如果有）的开始时间点有关。

为了便于理解，本文用图 4-2 来表示候选任务 i 可能会遇到的无线接口在开始下载时刻和结束时刻的状态变化。

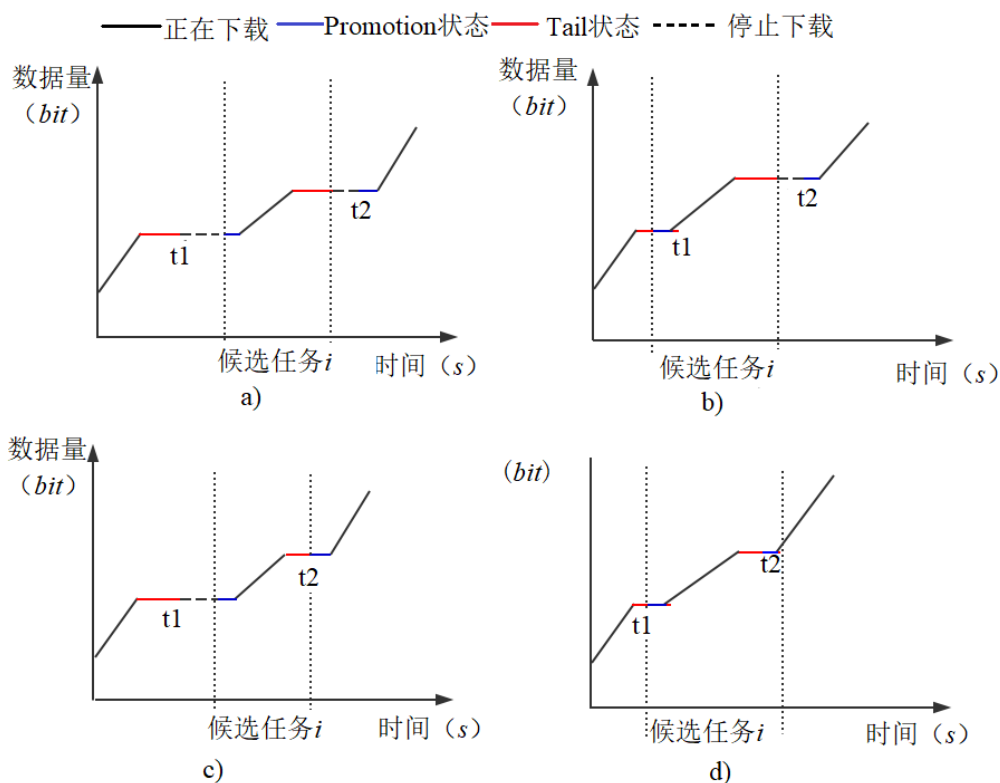


图 4-2 等效能耗时间

Figure 4-2 Equivalent energy consumption time

如图 4-2 所示, 图中一共有 4 个子图, 分别表示了候选任务 i 可能会遇到的四种状况。每个子图中横轴代表时间, 纵轴代表下载的数据总量, 标注为“正在下载”的黑色实线表示无线接口在当前时刻正处于下载过程中, 标注为“Tail 状态”的红色曲线表示无线接口在当前时刻处于 TAIL 状态, 标注为“Promotion 状态”的蓝色曲线表示无线接口在当前时刻处于 Promotion 状态。此外, 黑色虚线代表下载停止。本文定义两个和子图中 t_1, t_2 有关的参数。其中, t_1 表示无线接口在候选任务 i 开始之前最近的一个开启时间点, 定义为 t_{pre} , t_2 表示无线接口在候选任务 i 结束之后最早的一个开启时间点, 定义为 t_{pro} 。根据 $[t_{pre}, l_s^i]$, $[l_e^i, t_{pro}]$ 和 $[t_{pre}, t_{pro}]$ 时长值和 Tail 延迟时长 σ 值的不同, k_i 的计算方式也不同。

本文令 $d_1 = l_s^i - t_{pre}$ 表示当前下载任务距离上一个下载任务结束的时长, $d_2 = t_{pro} - l_e^i$ 表示当前下载任务距离下一个下载任务开始的时长, 则根据 d_1 和 d_2 的值不同 k_i 有以下几种结果:

1) 当 $d_1 > \delta$ 且 $d_2 > \delta$ 时, 表示候选任务 i 和当前已经下载完成的任务没有发生尾部冲突, 例如子图 a), 此时有 $k_i = l_e^i - l_s^i + \delta$ 。

2) 当 $d_1 \leq \delta$ 且 $d_2 > \delta$ 时, 表示候选任务 i 只和前一个已经下载完成的任务发生尾部冲突, 例如子图 b), 此时有 $k_i = l_e^i - l_s^i + \delta - (l_e^i + \delta - t_{pro})$, 即 $k_i = t_{pro} - l_s^i$ 。

3) 当 $d_1 > \delta$ 且 $d_2 \leq \delta$ 时, 表示候选任务 i 只和后一个已经下载完成的任务发生尾部冲突, 例如子图 c), 此时有 $k_i = l_e^i - l_s^i + \delta - (t_{pre} + \delta - l_s^i)$, 即 $k_i = l_e^i - t_{pre}$ 。

4) 当 $d_1 \leq \delta$ 且 $d_2 \leq \delta$ 时, 表示候选任务 i 和前后已经下载完成的任务都发生尾部冲突, 例如子图 d), 此时有 $k_i = \max(l_e^i - l_s^i + \delta - (t_{pre} + \delta - l_s^i) - (l_e^i + \delta - t_{pro}), 0(1))$, 即 $k_i = \max(0(1), t_{pro} - t_{pre} - \delta)$, 即当 $t_{pro} - t_{pre} \leq \sigma$ 时, 候选任务 i 几乎不消耗任何额外的能耗时间, 因此本文令 $k_i = o(1)$, $o(1)$ 是两个连续时间点的最小时间间隔。

综上所述, k_i 的定义如下:

$$k_i = \max(o(1), l_e^i - l_s^i + \min(l_s^i - t_{pre} - \sigma, 0) + \min(t_{pro} - l_e^i, \delta)) \quad (4-3)$$

4.3 本章小结

本章提出了两个视频节能传输算法。第一个是针对上一章的优化模型提出的基于最大等效能效传输算法, 旨在达到同贪婪传输相同的视频播放质量下最小化传输能耗。第二个是在第一个算法的基础上考虑了缓存区的填充程度, 旨在进一步减少视频卡顿时间。

5 性能评估

本章将给出本文提出的两种视频传输算法的数值分析结果。第一个是基于最大等效能耗传输算法，第二个是重缓冲过程的动态缓存门限算法。数值结果表明，第一个算法在拥有同贪婪传输相同的用户观看质量下有效的节省了移动端的传输能耗，第二个算法相比第一个算法进一步的减少了视频卡顿时间，提高了用户的观看体验质量，且总传输能耗基本不变。

本章的结构安排如下：5.1 给出基于最大等效能效传输算法的仿真分析结果；5.2 给出重缓冲过程的动态缓存门限算法的仿真分析结果；5.3 为本章小结。

5.1 基于最大等效能效传输算法仿真分析结果

本节将基于最大等效能效传输实验结果同传统的视频传输实验结果相比较。首先介绍实验配置，然后介绍在实验过程中本文提出的传输算法下载行为，最后将本文提出的传输算法同贪婪传输算法和 On-off 传输算法相对比。

5.1.1 实验配置

实验配置如下：本文在内存为 16GB，处理器为 Intel(R) Core(TM) i7-4590k，CPU 计算能力为 4.00GHZ，系统类型为 64 位 windows 的操作系统上模拟了三种不同的传输方式来下载视频。分别为：

(1) 贪婪传输，这种传输方式下无线接口几乎一直保持在开启状态直到视频传输完毕。

(2) On-Off 传输，无线接口设有两个上下限阈值，当缓存区的数据小于下限阈值时，无线接口开启下载数据，直到缓存区的数据高于上限阈值，无线接口关闭。在这里，本文将下限阈值设置为缓存区容量的 40%，上限阈值设置为缓存区容量的 100%。

(3) 基于最大等效能耗传输，即本文提出的传输算法，这种传输方式首先使用贪婪传输方案找到视频传输过程中的可优化段，然后在每个可优化段里利用最大等效能效的原理寻找最佳下载机会下载视频数据。

在实验过程中，本文使用的数据集如下：如上图 5-1 所示，这是一条来自爱尔兰运营商收集的真实的 LTE-BUS 的轨迹信息^[61-62]。它是由 2018 年 1 月 25 日收集的，带宽容量范围是从 0 Mbps 到 17Mbps，平均带宽是 3.46Mbps，一共有 2000 个

样本点，每个样本点之间都是以一秒为时间间隔。本文将这条轨迹信息作为实验中的带宽数据。

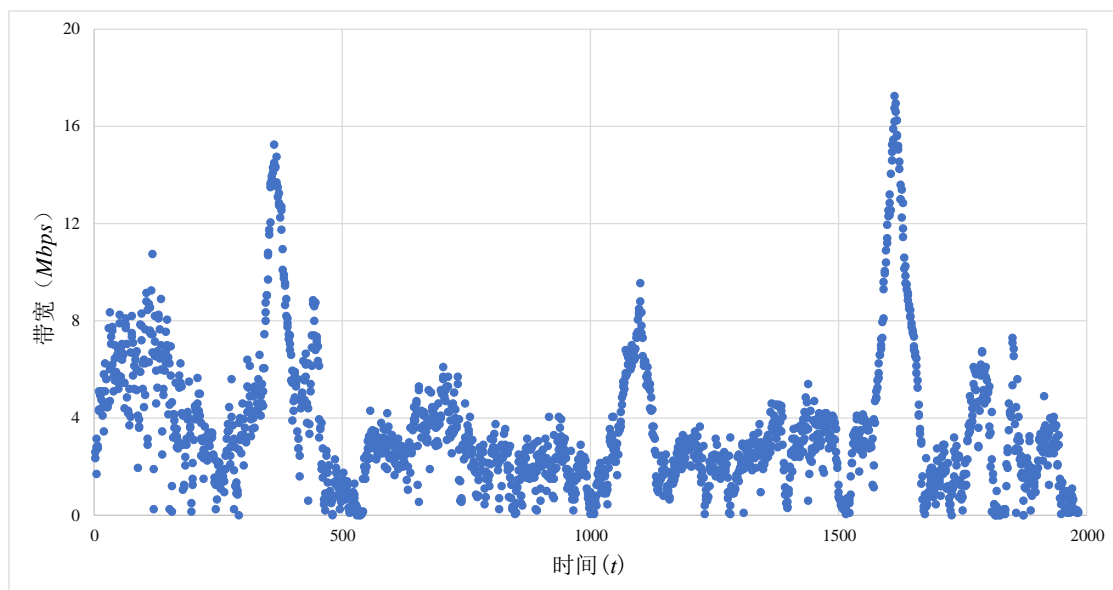


图 5-1 实验数据

Figure 5-1 The experimental data

在实验过程中，本文设定的参数如下：设定视频播放长度为 30 分钟，视频播放速率 $r=1Mbps$ ，本文将缓存区长度 w 设为变量，并分别模拟了缓存区 w 能够存储 1 至 5 分钟视频时长下三种不同的传输方式下的实验结果。

5.1.2 下载行为分析

首先介绍基于最大等效能效传输算法在视频下载过程中的典型下载行为，如图 5-2 所示。该图为设定变量 $r=1Mbit/s$ ， $w=4min$ 下的实验结果。图中横轴代表视频播放时间，纵轴代表下载的数据量。背景图是 4G-BUS 的轨迹信息图，图中每一个灰色的点都代表了在当前时刻的信道容量。图中一共有三条曲线，标注为“下载上限”的曲线表示播放过程中的下载上限，标注为“下载下限”的曲线表示播放过程中的下载下限，标注为“基于最大等效能效传输”的曲线表示基于最大等效能效传输的下载曲线，此时无线接口处于高耗能状态（包括 CONNECT 状态和 TAIL 状态）。此外，图中的青色线条表示停止下载，此时无线接口处于休眠状态。传输整个视频的总能耗时间即黑色部分占据的总时长。

由图 5-2，整个下载过程由 12 段黑色曲线组成，它们对应着视频传输过程中的 12 个下载任务。本文按照任务执行的先后顺序对这 12 段曲线进行了编号。例如，编号为 1 的任务是最早被执行的，编号为 12 的任务是最晚被执行的。显然可

以看到，除了任务 1 之外，任务 2, 3 的斜率明显大于任务 7, 8, 9, 10，实际上它们也大于任务 4 和任务 5，这是由于任务 2-12 中编号越靠前的任务具有越大的等效能效，即能够在单位时间内下载更多的数据。而任务 1 对应着视频从 0 时刻开始下载的重缓冲过程，并不包含在基于最大等效能效传输算法中。同时还可以看到，编号越靠前的下载任务往往对应了背景图中相对较好的带宽样本范围，这也验证了本文提出的算法的确按照等效能效的指标尽可能的利用了好的带宽，并每次优选选择下载能力强（即单位时间内尽可能的下载更多的数据量）的任务进行下载。此外，在利用好的带宽范围之外，任务 9、10、11、12 作为能够维持视频流畅播放的过渡期在一些相当不太好的带宽范围内进行下载，从而保证了视频能够流畅的播放。

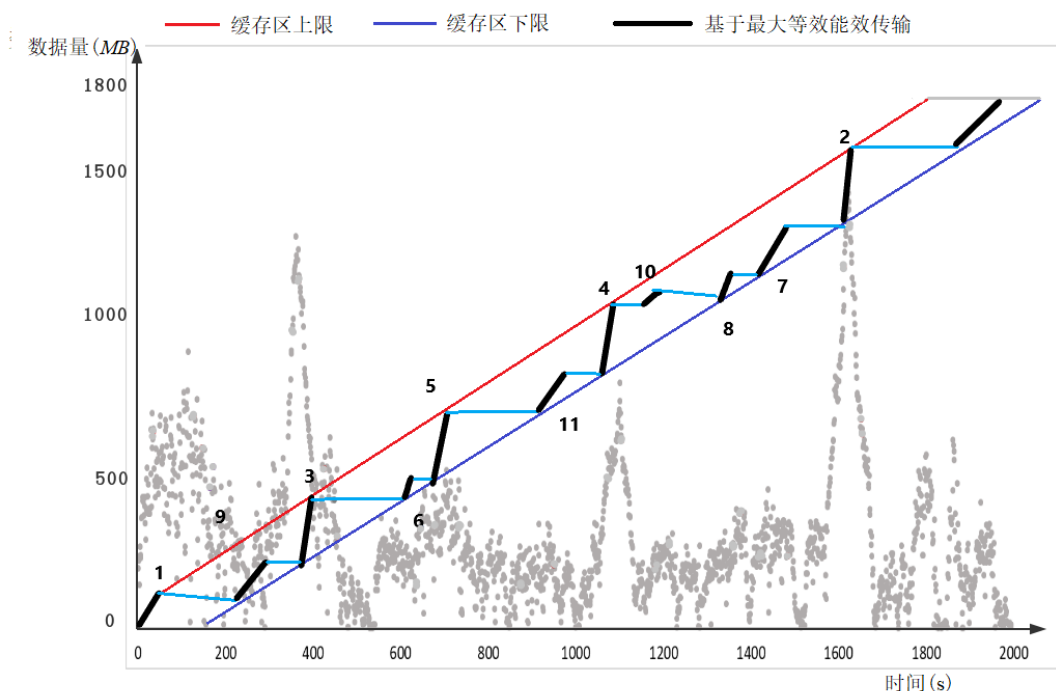


图 5-2 视频下载行为

Figure 5-2 Video downloading progress simulated

5.1.2 能效性能分析

首先本节介绍实验结果中用到的评估指标。本文的研究目标是尽可能的降低视频传输过程中消耗的电量，同时使用户在观看过程中保持与现有的下载方案相同的最佳流畅度观看体验。因此本节采用了两个评估指标用于评估实验性能。第一个指标是视频播放期间无线接口的总能耗时间，包括 CONNECT 时间和 TAIL 时间。第二个指标是视频播放期间的总重缓冲时间。第一个指标用来衡量视频传输方

案的耗能状况，第二个指标用来衡量视频传输方案的播放流畅性。接下来本节分别给出这两个指标对应的对比分析结果。

(1) 总能耗时间对比分析结果

三种传输方式的总能耗时间如图 5-3 所示。

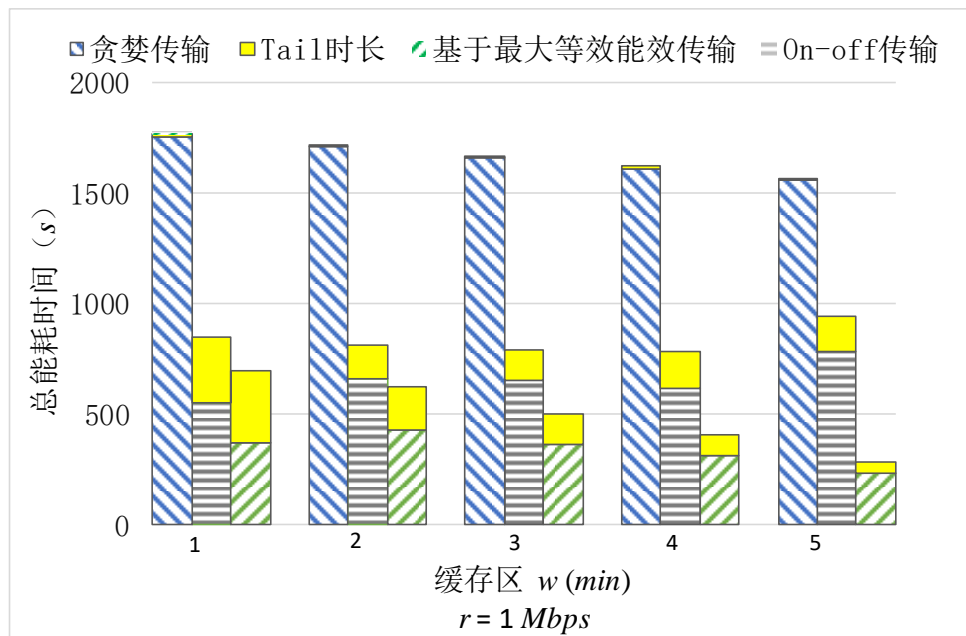


图 5-3 三种传输方式的总能耗时间对比

Figure 5-3 Comparison of total energy consumption time of three transmission modes

图中横轴代表缓存区长度，纵轴代表总能耗时间。本文模拟了在不同的缓存区长度下三种传输方式传输整个视频需要的总能耗时间。图中标记为“贪婪传输”、“On-off 传输”、“基于最大等效能效传输”的柱形高度分别代表了贪婪传输、On-off 传输和本文提出的基于最大等效能效传输的总能耗时间。标记为“Tail”时长的柱形高度表示这三种传输方式中无线接口尾部状态所浪费的总时间。

通过观察图 5-3 可以看出，本文提出的传输算法的总耗能时间明显优于其它两种传输算法。具体来说，在这三种传输方式中，贪婪传输在尾部状态浪费的时长是最少的，几乎可以忽略不计，而 On-off 传输和基于最大等效能效传输都浪费了相当一部分时间在尾部状态上。但与此同时，贪婪传输期间由于无线接口几乎一直处于 CONNECT 高耗能状态，因此总耗能时间也远远大于其他两种传输方式。On-off 传输虽然在 CONNECT 时间上相比贪婪传输节省了一部分，但是显然本文提出的传输方式在 CONNECT 消耗的时间上更少于 On-off 传输，随着缓存区的增大，这种优势更加明显。这是由于更大的缓存区使本文提出的传输算法能够更长时间的利用好的带宽范围下载更多的数据。而如果缓存区较小，即使遇到一段好的带宽样本范围，也只能下载一定的数据量从而错过剩余的一部分好带宽。同时，随着缓存

区的增大,本文提出的传输方式在 Tail 时间上浪费的时间有明显的减少,相比 On-off 传输浪费在 Tail 上的时间更少。这是由于缓存区能存放更多的数据量而导致本文提出的传输算法在下载过程中安排更少的任务,使得无线接口开启关闭的次数更少导致的。

通过观察图 5-3 还可以看出,随着缓存区的增大,贪婪传输和基于最大等效能效传输的总耗能时间都在减少,而 On-off 传输并没有这个规律。这是由于 On-off 传输虽然可能在开启阶段遇到好的带宽样本下载更多的数据量,但是在关闭阶段同样可能错过很多好的带宽所导致的。

数值结果表明,基于最大等效能效传输算法在无线接口的能耗时间方面明显优于传统的贪婪和 On-off 传输方案,分别降低了 83% 和 69%。

(2) 总缓冲时间对比分析结果

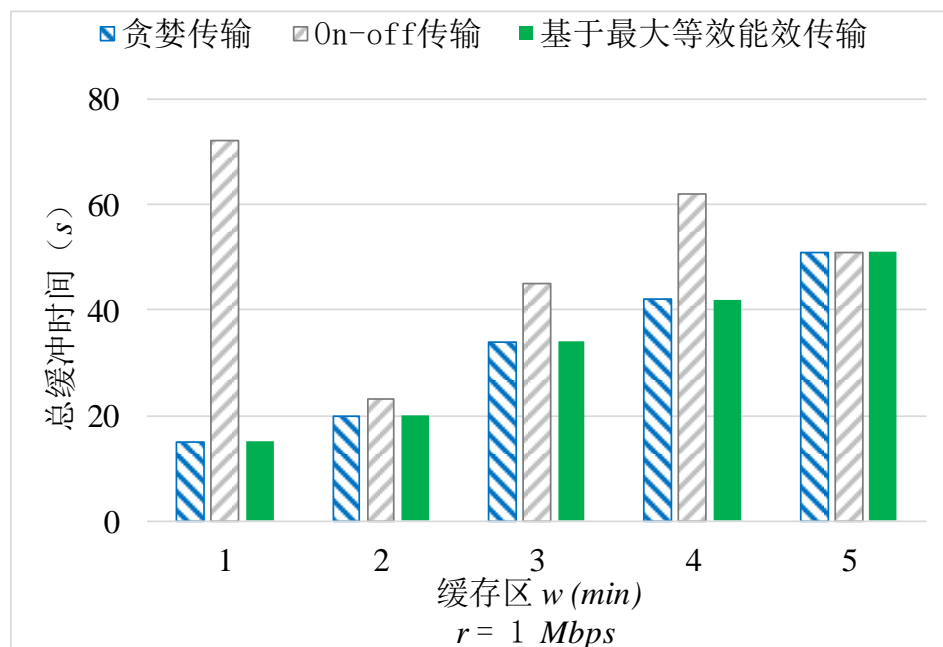


图 5-4 三种传输方式的总缓冲时间对比

Figure 5-4 Comparison of total rebuffering time of three transmission modes

如图 5-4 所示,横轴代表缓存区长度,纵轴代表重缓冲时间。本文模拟了在不同的缓存区长度下三种传输方式传输整个视频过程所需要的总重缓冲时间。图中标记为“贪婪传输”、“On-off 传输”、“基于最大等效能效传输”的柱形高度分别代表了贪婪传输、On-off 传输和本文提出的基于最大等效能效传输的总缓冲时间。

通过观察图 5-4 可以看出,本文提出的视频传输方案总是具有和贪婪传输方案相同的重缓冲时间。同时,这两种传输方式的重缓冲时间总是优于 On-off 传输。这是由于贪婪传输期间无线接口几乎一直处于开启状态,而 On-off 传输期间无线接口根据缓存区剩余数据量不断的进行开启和关闭,从而在开启期间可能会遇到

不好的带宽而缓存区又没有充足的数据导致播放卡顿所引起的。因此贪婪传输总是具有比 On-off 传输更短的重缓冲时间。而本文提出的传输算法本质上是在贪婪传输的基础上保证了具有和贪婪传输相同的重缓冲时间的前提下进行优化，因此具有和贪婪传输相同的最佳用户观看体验。

通过观察图 5-4 还可以看出，随着缓存区的增大，贪婪传输和基于最大等效能效传输的重缓冲时间也在增加。这是由于更大的缓存区意味着在视频在重缓冲期间需要更多的时间来下载更多的数据量以填满缓存区。而 On-Off 传输则没有这个规律。

综上所述，随着缓存区的增大，贪婪传输和本文提出的基于最大等效能效传输的总耗能时间更短，而总缓冲时间更长。On-off 传输的总耗能时间和总缓冲时间和缓存区大小没有直接关系。总体来说，贪婪传输具有最佳的用户观看体验，总耗能时间也最久；On-off 传输的用户观看体验最差，总耗能时间比贪婪传输短；本文提出的基于最大等效能效传输不仅具有最佳的用户观看体验，而且总耗能时间最短。

5.2 重缓冲过程的动态缓存门限算法仿真分析结果

本节介绍重缓冲过程的动态缓存门限算法的实验结果。重缓冲过程的动态缓存门限是在上述基于最大等效能效传输的基础上进一步的优化了缓存区的填充程度而形成的。本节将这两种传输方式进行比较。首先介绍算法评估方法，然后将给出实验对比分析结果。

5.2.1 评估办法

本节要对重缓冲过程的动态缓存门限方法和前文中提出的传输算法进行比较。比较方法如下：对给定长度的视频，本文采用上一节中的实验数据 4G-BUS 轨迹作为视频播放过程中的带宽样本信息，并使用基于最大等效能效的传输算法和重缓冲过程的动态缓存门限算法来下载视频数据。下载完成后，本文采用和前文相同两个指标来比较这两种传输方案的实验结果。

5.2.2 卡顿性能分析

基于最大等效能效传输和重缓冲过程的动态缓存门限的实验对比结果如下：

(1) 总能耗时间对比分析结果

对比分析结果如图 5-5 所示。图中横轴代表缓存区长度，纵轴代表能耗时间。

本文模拟了在不同的缓存区长度下这两种视频传输方式的总能耗时间。图中标注为“重缓冲段优化前”的柱体高度表示基于最大等效能效传输算法的总能耗时间，标注为“重缓冲段优化后”的柱体高度表示重缓冲过程的动态缓存门限算法的总能耗时间。

从图 5-5 中可以观察到，这两种传输方式的总能耗时间相差很小。当 $w=1、2$ (min) 时，重缓冲过程的动态缓存门限方式总能耗时间更短；当 $w=3$ 时，这两种传输方式的总能耗时间相同；当 $w=4、5$ (min) 时，基于最大等效能效传输方式总能耗时间更短。这是由于这两种传输方式在下载过程中都是基于最大能耗比的指标进行下载的，它们唯一的区别是下载的时间范围不同，重缓冲过程的动态缓存门限方式比基于最大等效能效传输方式下载开始的时间更早。当提前开始的这段时间对应的带宽较差时，优化后的算法能够在后续的时间找到更好的带宽对应的的时间间隔进行下载，因此能够取得总能耗时间上的优势。当提前开始的这段时间对应的带宽较好时，优化后的算法可能比优化前需要更多的时间来下载，因此在总能耗时间上花费更久。但是总体来说，它们的总能耗时间基本上相同。

从图 5-5 中还可以观察到，这两种传输方式随着缓存区的增加总能耗时间也在减小。这和上一节中的结论是相符合的。

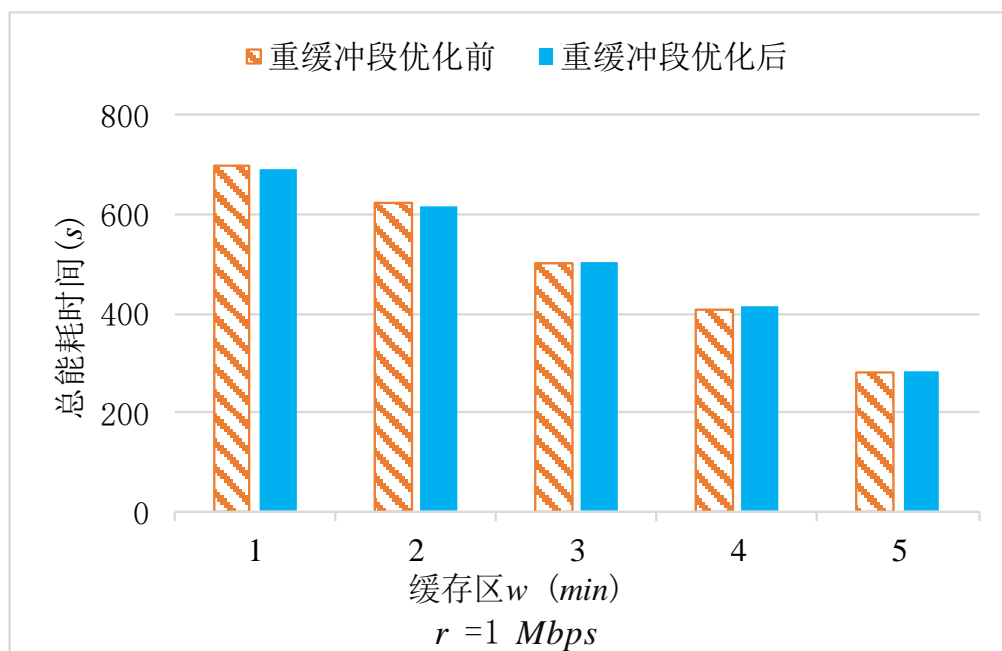


图 5-5 总能耗时间对比

Figure 5-5 Comparison of total energy consumption time

(2) 总缓冲时间对比分析结果

如图 5-6 所示，横轴代表缓存区长度，纵轴代表重缓冲时间。本文模拟了在不同的缓存区长度下基于最大等效能效传输算法和重缓冲过程的动态缓存门限算法

的总缓冲时间。图中标记为“重缓冲段优化前”的曲线代表基于最大等效能效传输算法，标记为“重缓冲段优化后”的曲线代表重缓冲过程的动态缓存门限算法。

从图 5-6 中可以观察到，优化后的视频传输方案在重缓冲时间上明显的优于优化前的视频传输方案。尤其是随着缓存区增大，这种优势更加明显。随着缓存区增大，基于最大等效能效传输方式所需要的重缓冲时间在不断的增大，而重缓冲过程的动态缓存门限方式的重缓冲时间仍然维持在一个较小的值。这是由于，随着缓存区的增大，基于最大等效能效传输在视频发生卡顿时，需要更多的时间来填满缓存区。而基于缓存区优化的等效能效传输则不需要在每次填满缓存区，而是在后续的时间里寻找更好的下载机会进行下载。

因此，重缓冲过程的动态缓存门限方式传输相比上一节中提出的传输算法虽然在总能耗时间上基本类似，但是有效的减少了重缓冲时间，使总的视频卡顿时间至少提高 20%，最高到 82%。

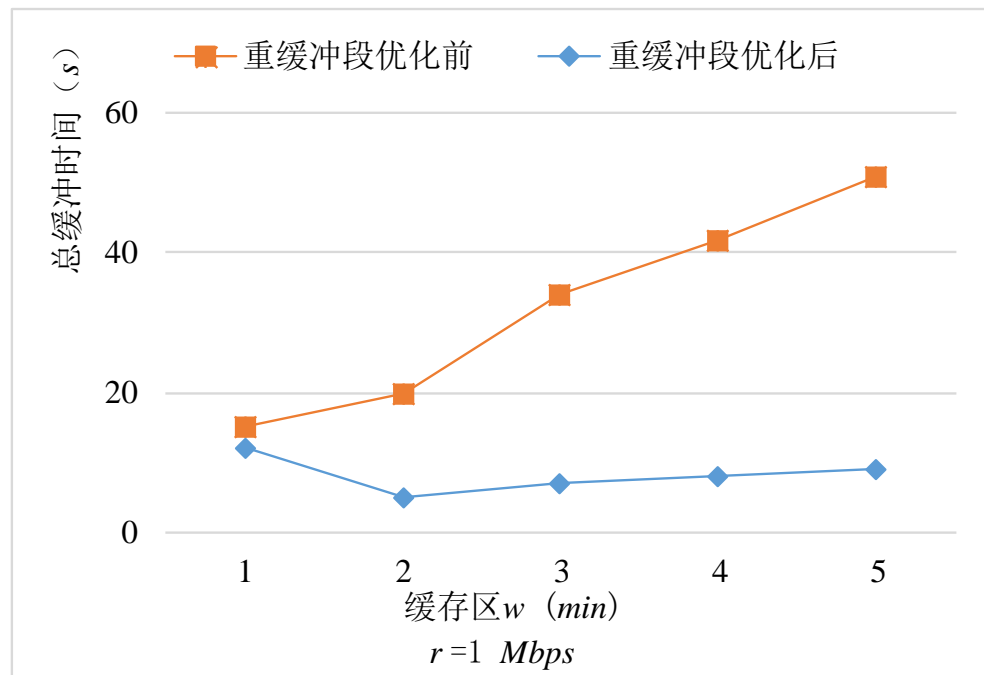


图 5-6 总缓冲时间对比

Figure 5-6 Comparison of total rebuffering time

5.3 本章小结

本章详细介绍了本文提出的两个视频传输优化算法，包括算法的提出背景、面临的挑战、基本思想、和具体实现方法。然后对这两个算法进行实验对比分析。实验结果表明，第一个算法实现了视频的节能传输，第二个算法在节能传输的基础上进一步提高了用户的观看体验质量。

6 结论

6.1 本文工作总结

随着通信技术的飞速发展,越来越多的用户通过移动设备观看在线视频。然而移动设备的电池电量是有限的,无法维持视频长时间播放。在视频播放过程中,数据传输是移动设备能耗的主要来源之一,降低传输能耗对移动设备有重要意义。

本文利用未来带宽信息来研究移动端视频节能传输算法,并完成了两个方面的研究。第一个方面,提出了基于最大等效能效传输算法,在保证用户拥有同贪婪传输相同的最佳的视频流畅性观看体验下最小化传输能耗。第二个方面,提出了重缓冲过程的动态缓存门限算法,该算法在第一个算法的基础上优化了缓存区的填充程度,在几乎不改变优化能耗的同时,进一步的减少了视频卡顿时间,提高用户的观看体验质量。

本文的主要工作可以分为以下三个部分:

(1) 利用未来带宽信息建立了视频节能传输模型。对连续视频时间做了离散化处理,从而将优化问题表示为以缓存限制为约束、以最小化传输能耗为目标的整数线性规划问题。

(2) 提出了一种基于最大等效能效传输的视频下载算法。该算法首先按照贪婪传输的方式将整个视频下载过程分为可优化段和不可优化区段,然后对可优化段进行优化。优化方式是在每个可优化段内安排一系列候选下载任务,采用等效能效(即单位能耗时间内下载的数据量)作为衡量每个候选下载任务下载能力的指标,以贪婪的方式迭代的在所有的候选下载任务中选择具有最大等效能效的候选任务进行下载,直到所有数据下载完毕。仿真结果表明,该算法有效的降低了传输能耗,同时达到同贪婪传输相同的最佳用户观看体验质量。

(3) 提出了一种重缓冲过程的动态缓存门限的视频算法。在上一个算法中,在使用贪婪传输寻找可优化段时,每个可优化段开始之前都会经历一段重缓冲时间。由于贪婪传输在出现重缓冲时总是填满缓存区才能继续播放。因此重缓冲期间并没有在节能传输范围内。基于这个观察,本文提出了重缓冲过程的动态缓存门限算法以进一步优化缓存区填充程度。该算法通过保持相邻重缓冲的后一个重缓冲过程的起始时刻不变,尝试改变前一个重缓冲过程的终止时刻,改变的同时以不会增加格外的卡顿时间为基准,直到找到最小的重缓冲时刻。仿真结果表明,该算法进一步减少了总视频卡顿时间,且总优化能耗基本不变。

6.2 后续研究计划

视频已经成为互联网最流行的在线娱乐形式，越来越多的用户通过移动设备观看在线视频。与此同时，现有技术下电池的储电能力有限，使移动设备无法长时间满足用户在线观看视频的需求。数据传输是移动设备能耗主要来源，因此开展移动设备的传输节能研究以延长电池续航时间具有现实意义。本文针对了传输能耗问题和视频卡顿的问题进行了研究。但是由于作者的能力和有限，论文中存在许多不足的地方，仍然需要进一步的探索，包括以下几个方面：

(1) 本文假设未来无线信道带宽变化是完全已知的，但在现实中，未来带宽信息还不能完全预测准确，因此下一步的工作是假设未来带宽变化是以一定概率知道的。

(2) 本文对算法进行了仿真实验以评估算法性能，进一步的工作应当在真实网络带宽下使用移动终端以测量算法性能。

参考文献

- [1] 中国互联网网络信息中心.第 42 次《中国互联网络发展状况统计报告》[J].中国经济报告, 2018(7).
- [2] Sandvine. 《2018 年全球互联网现象报告》[J]. 2018(11).
- [3] Vallina-Rodriguez N , Hui P , Crowcroft J , et al. Exhausting Battery Statistics Understanding the energy demands on mobile handsets[C]// Acm Sigcomm Workshop on Networking. ACM, 2010.
- [4] Bui N , Cesana M , Hosseini S A , et al. A Survey of Anticipatory Mobile Networking: Context-Based Classification, Prediction Methodologies, and Optimization Techniques[J]. IEEE Communications Surveys & Tutorials, 2017:1-1.
- [5] Barman N , Valentin S , Martini M G . Predicting link quality of wireless channel of vehicular users using street and coverage maps[C]// IEEE International Symposium on Personal. IEEE, 2016.
- [6] Schulman A , Navda V , Ramjee R , et al. Bartendr: A Practical Approach to Energy-aware Cellular Data Scheduling,” Mobicom[C]// International Conference on Mobile Computing & Networking. DBLP, 2010.
- [7] Perrucci G P , Fitzek F H P , Widmer J . Survey on energy consumption entities on the smartphone platform[C]// Vehicular Technology Conference. IEEE, 2011.
- [8] Wang L , Manner J . Energy Consumption Analysis of WLAN, 2G and 3G interfaces[C]// Green Computing & Communications. IEEE, 2011.
- [9] Yu F , Xue G , Zhu H , et al. Cutting without pain: Mitigating 3G radio tail effect on smartphones[C]// Infocom, IEEE. IEEE, 2013.
- [10] Balasubramanian N , Balasubramanian A , Venkataramani A . Energy consumption in mobile phones: a measurement study and implications for network applications[C]// Acm Sigcomm Conference on Internet Measurement. ACM, 2009.
- [11] Peng L . GScheduler: Reducing Mobile Device Energy Consumption[C]// Applied Computing & Information Technology/ Intl Conf on Computational Science/intelligence & Applied Informatics/ Intl Conf on Big Data, Cloud Computing, Data Science & Engineering. IEEE, 2017.
- [12] Qian H , Andresen D . Extending Mobile Device’s Battery Life by Offloading Computation to Cloud[C]// Acm International Conference on Mobile Software Engineering & Systems. IEEE, 2015.
- [13] Qian H , Andresen D . Jade: An efficient energy-aware computation offloading system with heterogeneous network interface bonding for ad-hoc networked mobile devices[C]// 15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). IEEE, 2014.
- [14] Qian H , Andresen D . An energy-saving task scheduler for mobile devices[J]. [IEEE 2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS). 2015:423-430.
- [15] Yao J , Kanhere S S , Hassan M . Improving QoS in High-Speed Mobility Using Bandwidth

- Maps[J]. IEEE Transactions on Mobile Computing, 2012, 11(4):603-617.
- [16] Song C , Qu Z , Blumm N , et al. Limits of Predictability in Human Mobility[C]// American Association for the Advancement of Science, 2010:1018-1021.
- [17] Paul U , Subramanian A P , Buddhikot M M , et al. Understanding traffic dynamics in cellular data networks[J]. Proceedings - IEEE INFOCOM, 2011, 8(1):882-890.
- [18] Gonzalez M C , Hidalgo C A , Barabasi A L . Understanding individual human mobility patterns[J]. Nature.
- [19] Song C , Qu Z , Blumm N , et al. Limits of Predictability in Human Mobility[C]// American Association for the Advancement of Science, 2010:1018-1021.
- [20] Bui N , Michelinakis F , Widmer J . A Model for Throughput Prediction for Mobile Users[C]// European Wireless ; European Wireless Conference. VDE, 2014.
- [21] Riiser H , Endestad T , Vigmostad P , et al. Video streaming using a location-based bandwidth-lookup service for bitrate planning[J]. ACM Transactions on Multimedia Computing, Communications, and Applications, 2012, 8(3):1-19.
- [22] Yao C , Yang C , Xiong Z . Energy-Saving Predictive Resource Planning and Allocation[J]. IEEE Transactions on Communications, 2016, PP(99).
- [23] Tao L , Gong Y , Jin S , et al. Energy efficient video QoE optimization for dynamic adaptive HTTP streaming over wireless networks[C]// IEEE International Conference on Communication Systems. IEEE, 2017.
- [24] Triki I , El-Azouzi R , Haddad M . NEWCAST: Anticipating resource management and QoE provisioning for mobile video streaming[C]// 2016 IEEE 17th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM). IEEE, 2016.
- [25] Draxler M , Blobel J , Dreimann P , et al. [IEEE 2015 International Conference and Workshops on Networked Systems (NetSys) - Cottbus, Germany (2015.3.9-2015.3.12)] 2015 International Conference and Workshops on Networked Systems (NetSys) - SmarterPhones: Anticipatory download scheduling for wireless video streaming[J]. 2015:1-8.
- [26] Draxler M , Karl H . Cross-layer scheduling for multi-quality video streaming in cellular wireless networks[C]// Wireless Communications & Mobile Computing Conference. IEEE, 2013.
- [27] 师媛, 温彬. 流媒体技术的分析及其系统平台的设计和比较研究[J]. 科技与企业, 2015(14):97-97.
- [28] Wang X , Vasilakos A V , Chen M , et al. A Survey of Green Mobile Networks: Opportunities and Challenges[J]. Mobile Networks and Applications, 2012, 17(1):4-20.
- [29] 张小娜. 浅谈 4G-LTE 无线通信技术[J]. 通讯世界, 2016(4):72-72.
- [30] Ghavimi F , Chen H H . M2M Communications in 3GPP LTE/LTE-A Networks: Architectures, Service Requirements, Challenges, and Applications[J]. IEEE Communications Surveys & Tutorials, 2015, PP(99):1-1.
- [31] 刘龙, 胡光宇. 4G-LTE 关键技术及其实际应用解析[J]. 企业技术开发: 中旬刊, 2016, 35(6):70-70.
- [32] 谢天龙. 浅谈 4G-LTE 网络优化关键技术及案例分析[J]. 数字通信世界, 2018, No.164(08):67.
- [33] Andres-Maldonado P , Ameigeiras P , Prados-Garzon J , et al. Optimized LTE Data Transmission Procedures for IoT: Device Side Energy Consumption Analysis[C]// IEEE International

- Conference on Communications Workshops. IEEE, 2017.
- [34] Gao S , Peng Z , Xiao B , et al. SCoP: Smartphone energy saving by merging push services in Fog computing[C]// 2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS). ACM, 2017.
- [35] Hu W , Cao G . Energy-aware video streaming on smartphones.[C]// Computer Communications. IEEE, 2015.
- [36] 陈博, 李曦, 周学海,等. 移动智能终端下 3G 网络低功耗延迟唤醒策略[J]. 中国科学院大学学报, 2015, 32(4):562-570.
- [37] Schulman A , Navda V , Ramjee R , et al. Bartendr: A Practical Approach to Energy-aware Cellular Data Scheduling,” Mobicom[C]// International Conference on Mobile Computing & Networking. DBLP, 2010.
- [38] Ra M R, Paek J , Sharma A B , et al. Energy-Delay Tradeoffs In Smartphone Applications[C]// International Conference on Mobile Systems. ACM, 2010.
- [39] Schulman A , Navda V , Ramjee R , et al. Bartendr: A Practical Approach to Energy-aware Cellular Data Scheduling,” Mobicom[C]// International Conference on Mobile Computing & Networking. DBLP, 2010.
- [40] Wu J , Chen Z . Reducing Energy Consumption and Overhead Based on Mobile Health in Big Data Opportunistic Networks[J]. Wireless Personal Communications, 2017, 92(4):1365-1385.
- [41] Dogar F R , Steenkiste P , Papagiannaki K . Catnap: Exploiting High Bandwidth Wireless Interfaces to Save Energy for Mobile Devices[C]// Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys 2010), San Francisco, California, USA, June 15-18, 2010. DBLP, 2010.
- [42] Bertozzi D , Benini L , Ricco B . Power aware network interface management for streaming multimedia[C]// Wireless Communications & Networking Conference. IEEE, 2002.
- [43] Rahmati A , Zhong L . Context-for-wireless: Context-sensitive energy-efficient wireless data transfer[C]// Proceedings of the 5th International Conference on Mobile Systems, Applications, and Services (MobiSys 2007), San Juan, Puerto Rico, June 11-13, 2007. DBLP, 2007.
- [44] Yeh J H , Chen J C , Lee C C . Comparative Analysis of Energy-Saving Techniques in 3GPP and 3GPP2 Systems[J]. IEEE Transactions on Vehicular Technology, 2009, 58(1):432-448.
- [45] Lee C C , Yeh J H , Chen J C . Impact of inactivity timer on energy consumption in WCDMA and cdma2000[C]// Wireless Telecommunications Symposium. IEEE, 2004.
- [46] Qian F , Wang Z , Gerber A , et al. TOP: Tail Optimization Protocol For Cellular Radio Resource Allocation[C]// IEEE International Conference on Network Protocols. IEEE, 2010.
- [47] Wang Y , Chen L , Lyv C . Cross-Layering Optimization for Low Energy Consumption in Wireless Body Area Networks[J]. IEEE Transactions on Communications, 2014, E97.B(9):1808-1816.
- [48] Li X , Dong M , Ma Z , et al. GreenTube: Power Optimization for Mobile Video Streaming via Dynamic Cache Management[C]// ACM Multimedia. ACM, 2012.
- [49] Hoque MA , Siekkinen M , Nurminen J K . [ACM Press the 19th annual international conference - Miami, Florida, USA (2013.09.30-2013.10.04)] Proceedings of the 19th annual international conference on Mobile computing & networking - MobiCom '13 - Using crowd-sourced viewing statistics to save energy in wireless video streaming[C]// Mobile Computing & Networking. 2013:377.

- [50] Ou Y F , Liu T , Zhao Z , et al. Modeling the impact of frame rate on perceptual quality of video[C]// Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on. IEEE, 2008.
- [51] Baset S A , Reich J , Janak J , et al. How green is IP-telephony?[C]// Acm Sigcomm Workshop on Green Networking. ACM, 2010.
- [52] Yao J , Kanhere S S , Hassan M . Improving QoS in High-Speed Mobility Using Bandwidth Maps[J]. IEEE Transactions on Mobile Computing, 2012, 11(4):603-617.
- [53] Fardous J , Kanhere S S . On the use of location window in geo-intelligent HTTP adaptive video streaming[C]// Networks (ICON), 2012 18th IEEE International Conference on. IEEE, 2012.
- [54] Draxler M , Karl H . Cross-layer scheduling for multi-quality video streaming in cellular wireless networks[C]// Wireless Communications & Mobile Computing Conference. IEEE, 2013.
- [55] &Lt R P . HTTP Live Streaming[J]. 2015.
- [56] Riiser H , Vigmstad P , Griwodz C , et al. Commute path bandwidth traces from 3G networks: analysis and applications[C]// Proceedings of the 4th ACM Multimedia Systems Conference. ACM, 2013.
- [57] Abou-Zeid H , Hassanein H S , Valentin S. Energy-Efficient Adaptive Video Transmission: Exploiting Rate Predictions in Wireless Networks[J]. IEEE Transactions on Vehicular Technology, 2014, 63(5):2013-2026.
- [58] Moradi F , Karaca M , Fitzgerald E , et al. Optimizing DRX for video delivery over LTE: Utilizing channel prediction and in-network caching[C]// International Symposium on Modeling & Optimization in Mobile. IEEE, 2017.
- [59] Lu Z , Veciana G D . Optimizing Stored Video Delivery For Mobile Networks: The Value of Knowing the Future[J]. IEEE Transactions on Multimedia, 2018, PP(99):1-1.
- [60] Huang J , Qian F , Gerber A , et al. A close examination of performance and power characteristics of 4G LTE networks[C]// International Conference on Mobile Systems. ACM, 2012.
- [61] D. Raca , J.J . Quinlan , A.H . Zahran and C.J. Sreenan , Beyond Throughput: a 4G LTE Dataset with Channel and Context Metrics[C]// ACM Multimedia Systems Conference (MMSys 2018), 2018.
- [62] 4G LTE Dataset, [http://www.cs.ucc.ie/~dr11/4G Dataset/index.html](http://www.cs.ucc.ie/~dr11/4G%20Dataset/index.html). Last accessed Jun. 4, 2018.

作者简历及攻读硕士学位期间取得的研究成果

一、作者简历

王唯 女 1993 年 3 月生

2012 年 9 月至 2016 年 6 月 青岛科技大学 信息工程 获学士学位

2016 年 9 月至 2019 年 6 月 北京交通大学 信息安全 获硕士学位

二、参与科研项目

[1] 基于熵理论的信息匹配网络测量与建模, 国家自然科学基金, NO. 6187203

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京交通大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：

签字日期：

年 月 日

学位论文数据集

表 1.1: 数据集页

关键词*	密级*	中图分类号	UDC	论文资助
在线视频, 传输 能耗, 移动终端, 数据传输	公开			国家自然科学基金 No. 61872031
学位授予单位名称*		学位授予单位代 码*	学位类别*	学位级别*
北京交通大学		10004	工学	硕士
论文题名*		并列题名		论文语种*
视频点播缓存系统的优化研究				中文
作者姓名*	王唯		学号*	16120191
培养单位名称*		培养单位代码*	培养单位地址	邮编
北京交通大学		10004	北京市海淀区西直 门外上园村 3 号	100044
学科专业*		研究方向*	学制*	学位授予年*
信息安全		信息网络	3	2019
论文提交日期*	2019 年 5 月 5 日			
导师姓名*	李纯喜		职称*	副教授
评阅人	答辩委员会主席*		答辩委员会成员	
	郭宇春		陈一帅, 郑红云, 赵永祥, 张立军	
电子版论文提交格式 文本 () 图像 () 视频 () 音频 () 多媒体 () 其他 () 推荐格式: application/msword; application/pdf				
电子版论文出版 (发布) 者		电子版论文出版 (发布) 地		权限声明
论文总页数*	45			
共 33 项, 其中带*为必填数据, 为 21 项。				