

新一代互联网流媒体服务及路由关键技术

吴桦, 程光, 胡劲松, 徐健著

东南大学出版社



新一代互联网流媒体服务 及路由关键技术

吴桦 程光 胡劲松 徐健 著



东南大学出版社
SOUTHEAST UNIVERSITY PRESS

• 南京 •

内 容 提 要

流媒体应用已经成为因特网中的主要应用,随着技术的快速演变、网络结构的日益复杂,以及加密技术的应用,对流媒体应用的服务质量评估成为一个难题。本文首先对国内外主要流媒体服务商进行调研,针对流媒体数据分发的各个环节分析数据分发策略的优缺点和优化方案。对加密流媒体的分析,给出了一个通过加密视频流识别出视频关键属性的研究过程,最后给出针对流媒体应用的路由性能测量方法。本书可作为对流媒体进行架构和分发策略设计的参考书,也可为科研人员对加密流媒体进行深入研究提供参考。

图书在版编目(CIP)数据

新一代互联网流媒体服务及路由关键技术/吴桦
等著. —南京:东南大学出版社,2017. 11

ISBN 978-7-5641-7447-7

I. ①新… II. ①吴… III. ①计算机网络—
多媒体技术—研究 IV. ①TP37

中国版本图书馆 CIP 数据核字(2017)第 255045 号

新一代互联网流媒体服务及路由关键技术

著 者 吴桦 程光 胡劲松 徐健

出版发行 东南大学出版社

出 版 人 江建中

责任编辑 张 煦

社 址 南京市四牌楼 2 号 (邮编:210096)

经 销 全国各地新华书店

印 刷 兴化印刷有限责任公司

开 本 700 mm×1000 mm 1/16

印 张 18.75

字 数 347 千

版 次 2017 年 11 月第 1 版

印 次 2017 年 11 月第 1 次印刷

书 号 ISBN 978-7-5641-7447-7

定 价 58.00 元

(本社图书若有印装质量问题,请直接与营销部联系。电话(传真):025-83791830)

前 言

流媒体应用是当前因特网上最重要的应用之一,主要包括视频应用和音频应用,在媒体文件播放前并不下载整个文件,而只是开始部分数据到缓存,随后就可以一边播放一边下载。

随着移动通信网络和技术不断发展,移动流媒体业务相关技术日益走向成熟。基于超文本传输协议(Hyper Text Transfer Protocol, HTTP)的移动流媒体服务,由于使用 Web 服务器分发数据,部署简单,适用范围广泛,成为国内外流媒体服务商普遍采用的技术。但是 HTTP 协议的设计目标并不是针对流媒体应用的,这导致为了适应流媒体的应用特点在其上进行针对流媒体应用的优化是必须的。流媒体服务由于数据量大,需要的各种数据处理、存储、调度、带宽、版权资源都需要巨大的成本投入,国内外服务商必须进行各种技术优化以尽可能减少成本,提高用户的服务质量感受。各种技术的有机结合会和服务商节约大量的成本,本书针对其中的一些关键问题进行现状研究及分析讨论,并试图给出一些有益的建议。

本书主要关注的是流媒体应用中和网络相关的关键技术问题,对其中的一些问题进行了深入讨论和尝试。本书第 1 章是绪论,介绍了流媒体应用现状和流媒体应用的关键技术。

第 2 章介绍了因特网协议的基本概念和流媒体传输协议,着重介绍了基于 HTTP 协议的三种流媒体传输技术,并以优酷为例研究了流媒体传输效能。

第 3 章介绍了国外视频点播技术现状,以 YouTube 为例,分析了流媒体传输在文件格式、传输协议和流量控制方面的演变历史。同时也指出由于国外视频已逐步使用加密传输方法,在加密传输后,对流媒体服务流量模型的分析面临困难。

第 4 章对国内的主要视频服务:优酷、爱奇艺和腾讯视频进行了现状调查,结合国外视频服务技术进行了比较分析并指出了优化的方向。

第 5 章介绍了流媒体服务质量评价指标和常规的测量方法,并分别给出了智能手机点播视频和 IPTV 播放视频的 QoE 测量方法,以及 QoE 与相应 QoS 的相关性分析。第 5 章的分析都是在非加密场景下进行的,这种场景下视频质量评估可以基于 DPI 技术,获取到视频大小、可播放时长、码率等信息,可以较

为准确的评估视频的初始缓冲时长、卡顿等体验指标,然而,由于越来越多的视频流量采用加密技术,加密流量的引入使得之前设计的视频质量评估方案无法获取到其需要的参数数据。

第6章~第10章针对视频流数据在加密场景下的行为分析方法展开研究。

第6章是对加密流分析方法的综述,介绍了加密流识别的对象、识别的类型、识别的方法,也给出了加密流量分析存在的问题和未来研究的方向。

第7章以 YouTube 加密视频流为研究对象,进行了加密流量的初步识别,首先识别 YouTube 流量,其次将观看同一个视频流的加密流量进行关联,最后给出对 YouTube 视频传输模式的识别方法。

第8章在第7章的基础上进一步识别出观看同一个视频流时的不同数据分段,给出了数据片段中音频片段的数据量分布范围、数据片段在数据流中的位置以及断线重连的数据片段的对应关系。这章的工作为后续的 YouTube 视频码率及分辨率识别提供了基础。

第9章给出了分别从 Android 平台的 YouTube App 的 HTTPS 加密流量和 iOS 平台的 YouTube App 的 HTTPS 加密流量中识别出 YouTube 视频的播放码率的方法,通过中间人攻击获得的真实数据,验证了对加密视频流码率识别的准确性。

第10章给出了对加密视频流进行视频分辨率的识别的方法,从加密流量中提取出用户所观看视频的所有视频片段以及这些视频片段的播放位置、播放码率、数据量和播放时长等特征。根据这些特征首先使用 C4.5 决策树算法对视频片段的分辨率进行分辨率识别,又引入了 k-means 聚类算法对 C4.5 决策树算法识别出来的分辨率结果进行辅助修正识别,同样通过中间人攻击获得真实数据对结果进行了准确性验证。

第11章介绍了影响 VoIP 服务质量的网络关键因素,对基于 E 模型的语音质量评估的测量方法做了介绍和改进,并设计实现了一个评估 VoIP 语音质量的系统。该系统可以完成对 VoIP 语音质量的评估。

第12章介绍了流媒体运营商在进行视频分发时如何评估视频流所经网络路径的拥塞状态。直接测量获得的网络延迟和丢包值并不能直接用来进行拥塞状态评估,本章分别定义了从网络路径的延迟测度和丢包测度给出的可以描述网络拥塞状态的派生测度,并给出了测量方法以及应用实例。这些测度可为流媒体运营商优化数据分发路由,提高用户的服务感受提供依据。

本书主要是作者在流媒体服务质量研究领域长期的研究成果的总结,也保留了作者指导学生参与的科研项目部分相关科研成果和论文。在本书的撰写过程中,潘吴斌、房敏、黄顺翔、李想、代甜甜、王玉翔、陈燕扬等给予了大力支

持,参与了本书部分章节的编写工作,全书由吴桦、程光统稿。

本书的研究成果受到国家重点研发计划[“地址驱动的网络安全管控体系结构及其机理研究(No. 2017YFB0801700)”中的“SDN/NFV 与 NDN 安全研究(No. 2017YFB0801703)」、国家 863 计划[IPv6 大规模编址与路由关键技术研究 and 验证(No. 2015AA010201)]等国家级项目的资助,在此表示感谢! 在本书的撰写过程中,得到东南大学计算机科学与工程学院、东南大学网络空间安全学院、计算机网络和信息集成教育部重点实验室(东南大学)、东南大学出版社等单位领导和专家的大力支持,在此深表谢意! 同时对本书中所引用的参考文献的作者及不慎疏漏的引文作者也一并致谢!

由于作者水平有限,编写过程中难免存在很多不足及顾此失彼之处,敬请读者给予批评指正!

著 者

2017 年 11 月

目 录

1	绪论	1
1.1	流媒体应用现状	1
1.2	流媒体应用中的关键技术	3
1.2.1	数据编码技术	3
1.2.2	数据分发策略技术	3
1.2.3	数据传输技术	4
1.2.4	数据加密技术	5
1.2.5	服务质量评价技术	5
1.2.6	流媒体服务评估相关测度的测量技术	6
1.3	本书的目的	6
2	流媒体传输协议	8
2.1	流媒体应用协议	8
2.1.1	协议	8
2.1.2	因特网协议	9
2.1.3	IP 层协议	10
2.1.4	传输层协议	12
2.1.5	应用层协议	13
2.1.6	流媒体应用协议	13
2.2	基于 HTTP 的流媒体传输技术	15
2.2.1	HPD 视频传输技术	15
2.2.2	DASH 视频传输技术	16
2.2.3	HLS 视频传输技术	18
2.2.4	技术比较	19
2.3	流媒体传输协议效率研究	20
2.4	优酷视频应用传输效能研究	21
2.4.1	数据采集	22
2.4.2	数据分析方法	22

2.4.3	不同限速情况下传输效能	23
2.4.4	未完成播放的数据下载效能的分析	26
2.4.5	不同限速情况下 TCP 流数的变化的分析	28
2.4.6	实验结论	30
2.5	本章小结	30
3	国外视频点播技术现状与分析	33
3.1	研究目的	33
3.2	YouTube 文件格式	34
3.2.1	MP4	34
3.2.2	WebM	37
3.2.3	技术演进分析	41
3.3	传输协议	42
3.3.1	TCP 协议的改进	42
3.3.2	QUIC 协议	43
3.3.3	QUIC 连接过程及其优势分析	44
3.4	流量控制	46
3.4.1	相关研究	46
3.4.2	流量模型	47
3.4.3	App 数据分析	49
3.5	小结	50
4	国内视频点播技术现状分析	54
4.1	研究目的	54
4.2	数据采集和分析方法	54
4.2.1	数据采集	54
4.2.2	分析方法	56
4.3	优酷视频	58
4.3.1	4G(联通)下使用 Android App	58
4.3.2	WiFi 下使用 Android App	61
4.3.3	优酷视频的“自动”策略分析	62
4.3.4	优酷视频的主要特征	64
4.4	爱奇艺	66
4.4.1	4G(联通)下使用 Android App	66
4.4.2	WiFi 下使用 Android App	67

4.4.3 爱奇艺视频的主要特征	68
4.5 腾讯视频	68
4.5.1 4G(联通)下使用 Android App	68
4.5.2 WiFi 下使用 Android App	70
4.5.3 腾讯视频的主要特征	73
4.6 WiFi 下使用 iOS App	74
4.6.1 优酷视频	74
4.6.2 爱奇艺	74
4.6.3 腾讯视频	75
4.7 在低速信道下使用 P2P 模式分发视频数据的特征分析	76
4.8 国内视频服务主要特征分析与优化方向	78
4.9 本章小结	79
5 视频应用服务质量体验评价方法	81
5.1 服务质量体验	81
5.2 影响流媒体应用 QoE 的主要因素	85
5.3 视频流媒体 QoE 参数测量方法	87
5.3.1 影响视频应用 QoE 的关键测度	87
5.3.2 终端测量	90
5.3.3 非终端测量方法	92
5.4 移动互连终端视频应用 QoE 研究	94
5.4.1 研究目的	94
5.4.2 QoS 性能参数的测量方法	94
5.4.3 优酷 QoE 监控系统的设计与实现	96
5.4.4 实验结果与分析	99
5.5 IPTV 视频应用中 QoE 与 QoS 关联分析	101
5.5.1 研究目的	101
5.5.2 IPTV QoE 评价模型	102
5.5.3 IPTV 的 MOS 值计算方法	103
5.5.4 测试系统	105
5.5.5 实验部署和结果分析	106
5.6 小结	107
6 加密流分析方法	110
6.1 引言	110

6.2	加密流量识别概述	111
6.3	识别对象	112
6.4	识别的类型	113
6.4.1	加密与未加密流量识别	113
6.4.2	加密协议识别	114
6.4.3	服务识别	116
6.4.4	异常流量识别	117
6.4.5	内容本质识别	117
6.5	加密流量识别方法	118
6.5.1	基于有效负载的识别方法	118
6.5.2	数据包负载随机性检测	118
6.5.3	基于机器学习的识别方法	119
6.5.4	基于行为的识别方法	119
6.5.5	基于数据包大小分布的识别方法	120
6.5.6	混合方法	120
6.5.7	加密流量识别方法综合对比	120
6.6	加密流量分析的问题	121
6.7	加密流量分析研究方向	123
6.8	小结	125
7	加密视频流量的识别、关联和传输模式识别方法	131
7.1	YouTube 移动端流量识别	131
7.1.1	问题分析	131
7.1.2	系统设计	132
7.1.3	系统实现	133
7.1.4	实验与结果分析	137
7.2	YouTube 移动端加密视频流关联	140
7.2.1	问题分析	140
7.2.2	TLS 会话恢复机制分析	142
7.2.3	系统设计	145
7.2.4	系统实现	146
7.2.5	算法测试实验	148
7.2.6	算法评价与应用	150
7.3	YouTube 视频传输模式识别方法	151
7.3.1	问题分析	151

7.3.2	系统设计	155
7.3.3	系统实现	156
7.3.4	实验与结果分析	163
7.6	小结	167
8	自适应流媒体中加密视频数据分段流量分析	169
8.1	自适应流媒体传输中的数据分段	169
8.2	研究加密视频流量分段的方法	170
8.2.1	中间人攻击方法获得明文	170
8.2.2	流量离线文件 PCAP 格式分析	172
8.3	YouTube 数据片段识别问题分析	174
8.4	YouTube 加密流量分段分析系统设计	174
8.4.1	系统总体设计	174
8.4.2	报文整合分段	175
8.4.3	音频片段分析	177
8.4.4	片段位置识别	178
8.4.5	断线重连识别	179
8.5	YouTube 加密流量分段识别应用实例	180
8.6	小结	183
9	加密视频流视频码率识别方法	184
9.1	加密视频流码率识别的基本问题	184
9.2	YouTube Android 平台码率识别	185
9.2.1	YouTube App DASH 视频传输特征分析	185
9.2.2	YouTube App DASH 视频码率识别问题分析	187
9.2.3	模块设计	188
9.2.4	音频片段等长性分析	188
9.2.5	片段类型识别	189
9.2.6	片段类型矫正	190
9.2.7	视频片段排序去重	191
9.2.8	播放码率计算	192
9.2.9	实验与结果分析	193
9.3	YouTube iOS 平台码率识别	196
9.3.1	YouTube App HLS 视频传输特征分析	196
9.3.2	YouTube App HLS 视频码率识别问题分析	198

9.3.3	模块设计	198
9.3.4	音频片段识别	199
9.3.5	视频片段处理	200
9.3.6	播放码率计算	206
9.3.7	实验与结果分析	207
9.4	小结	209
10	加密视频流视频分辨率识别方法	211
10.1	视频分辨率	211
10.2	视频分辨率和码率关系分析	212
10.3	C4.5 决策树算法识别 YouTube 视频分辨率	214
10.3.1	C4.5 决策树算法分析	214
10.3.2	实验与结果分析	216
10.4	k-means 算法辅助识别 YouTube 视频分辨率	217
10.4.1	k-means 聚类算法分析	217
10.4.2	实验与结果分析	218
10.5	小结	222
11	VoIP 服务质量体验评估方法	223
11.1	引言	223
11.2	VoIP 关键技术	225
11.2.1	VoIP 的网络性能要求	225
11.2.2	VoIP 协议架构	225
11.2.3	影响语音质量的因素	228
11.3	VoIP 的 QoE 评价方法及改进	230
11.3.1	VoIP 的 QoE 评价方法介绍	230
11.3.2	主观评价方法	231
11.3.3	客观评价方法	232
11.3.4	E 模型评价方法	233
11.4	E 模型优化	241
11.4.1	E 模型的不足之处	241
11.4.2	E 模型中抖动参数的加入	241
11.5	VoIP 的 QoE 评估系统	243
11.5.1	评估系统的实验环境	243
11.5.2	评估系统实施总体结构	244

11.5.3 评估系统测试	245
11.6 小结	246
12 流媒体应用路由性能测量研究	249
12.1 研究目的	249
12.2 研究对象的定义	249
12.3 基于报文延迟测度估计网络路径拥塞	251
12.3.1 路径拥塞状态和报文 RTT 和关系	251
12.3.2 网络延迟特性测量	253
12.3.3 子网间延迟测度 Path-RTT 定义	255
12.3.4 基于 BDTRS 的派生测度定义	258
12.3.5 数据处理方法	260
12.3.6 分析实例	262
12.4 基于报文丢包测度估计网络路径拥塞	266
12.4.1 网络丢包特性测量	266
12.4.2 网络路径丢包测度定义	267
12.4.3 丢包平台基本性质分析	268
12.4.4 TCP 流丢包和丢包平台丢包的关系	269
12.4.5 基于 TCP 平行流的 Path_Loss 估计算法	271
12.4.6 算法误差分析	275
12.4.7 基于平行 TCP 流估计 Path_Loss 的算法验证	280
12.5 小结	282

1.1 流媒体应用现状

流媒体应用是当前因特网上最重要的应用之一,流媒体应用是指使用流式传输技术的多媒体应用。流媒体应用包括视频应用和音频应用,在媒体文件播放前并不下载整个文件,只是开始部分缓存一些数据到缓存,随后就可以一边播放一边下载。

流媒体应用使用网络协议分发音视频数据文件,流媒体分发技术主要有两大类,一是基于 RTP (Realtime Transport Protocol)/RTCP (Realtime Transport Control Protocol)/RTSP (Real Time Streaming Protocol, RTSP) 的实时流媒体传输技术,“实时”的概念是指在应用中数据的交付必须与数据的产生保持精确的时间关系,因此需要有相应的协议支持。另一类则是目前视频网站普遍采用的基于 HTTP 的渐进式下载技术。渐进式下载是指音视频数据顺序下载,但并不需要精确的同步,客户端具有一定的播放时间缓存能力。

RTP/RTCP/RTSP 协议族是最早被提出来实时流媒体协议的,用于数据传输的 RTP 协议是基于 UDP 的,这是考虑到 UDP 协议的传输效率高于 TCP,而且视频和音频数据本身对丢包可以有一定的容忍度。RTP/RTCP/RTSP 协议族的优点在于可以精确控制视频帧的传输,可以承载实时性很高的应用。如 H. 323 视频会议协议,底层一般采用 RTSP 协议。但是 UDP 协议在复杂网络下路由器的穿透会出现问题。RTP/RTCP/RTSP 协议族缺点是实现的复杂度高,对视频服务器的设备和信道带宽稳定性要求较高,不易实现,在信道共享的因特网上,如果可用带宽不能保证,会引起服务质量的大幅下降,目前主要用在一些专用的视频分发网络中。

随着移动通信网络和技术不断发展,移动流媒体业务相关技术日益走向成熟。基于超文本传输协议(Hyper Text Transfer Protocol, HTTP)的移动流媒体服务由于使用 Web 服务器网络环境,部署简单,适用范围广泛,成为广泛采用的技术。基于 HTTP 渐进式下载移动流媒体业务流程本质上类似于从 HTTP 服务器下载文件或图片。目前国外的 YouTube, Netflix,国内的腾讯视

频、优酷视频等播放器和媒体传输平台都支持 HTTP 渐进下载流媒体业务。在 HTTP 渐进下载的工作流程中,首先由服务提供商将原始的视频文件编码成为一定的格式,然后将编码后的文件放到普通的 Web 服务器中。客户端则通过 URL 地址访问到该视频文件,最后下载到本地进行播放。为了保证用户观看的连续性,音视频数据需要在客户端缓存一段时间后再播放,并且在播放期间不间断下载。这种方式的缺点是,当用户退出观看时如果缓存中还剩余很多未观看的视频,浪费了对应的带宽资源。一方面要保证用户的服务质量感受,需要缓存足够多的视频,另一方面,需要尽量节约网络的带宽资源,自适应流媒体传输技术被提出。

自适应流媒体的传输机制如图 1.1 所示。

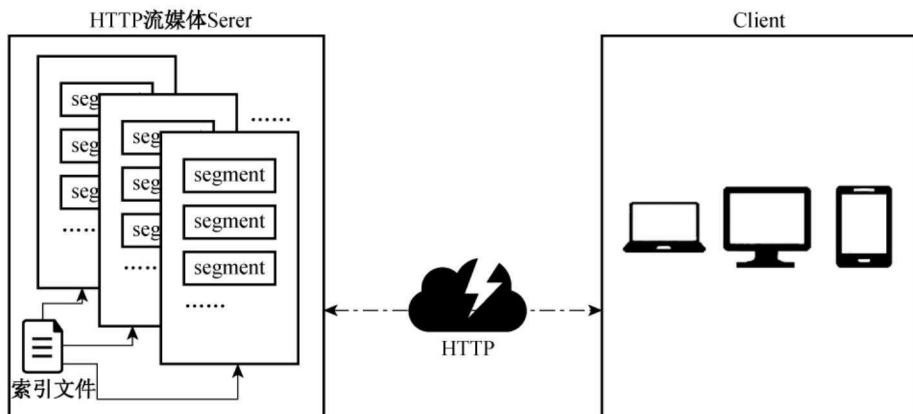


图 1.1 自适应流媒体传输方法

在自适应 HTTP 流媒体中媒体资源存储在 Web 服务器上并且被分割为很多小片段(segment),每一个小片段的解码独立于其他的片段。而索引文件则给出片段之间的联系。为了适应不同的带宽条件,同一个媒体文件通常会被编码为不同分辨率的片段。

自适应 HTTP 流媒体的工作流程是:客户端获取相应媒体资源的索引文件,然后根据索引文件下载相应的片段。如果在播放过程中发现视频卡顿的现象,客户端可以选择更低分辨率的片段下载。不同于传统的实时流媒体传输中流媒体服务器来控制数据包的发送,自适应 HTTP 流媒体允许客户端决定哪一个数据包应该被发送。同时,自适应 HTTP 流媒体业务比传统的渐进下载要更加接近实时的传送数据。由于 HTTP 使用 TCP 传输协议,只要能使用 HTTP 协议就能播放自适应流媒体,也避免了被防火墙阻隔的问题。

当前较为出名的自适应流媒体标准主要有 3GPP 和 MPEG 组织提出的

DASH (Dynamic Adaptive Streaming over HTTP)、苹果公司提出的 HLS (HTTP Live Streaming)、微软提出的 Microsoft Smooth Streaming 和 Adobe 提出的 HTTP Dynamic Streaming。

1.2 流媒体应用中的关键技术

1.2.1 数据编码技术

视频信息如果不压缩,其数据量是巨大的。如分辨率为 $1\,920 \times 1\,080$ P 的高清视频,每个彩色分量采用 8 bit 量化,帧率为每秒 25 帧,则数据量为每秒 414 Mbit,如果不加以压缩,在现有的网络接入条件下,根本不可能实时传输和播放。图像编码压缩技术的基本思想是去除图像数据中各种相关性所带来的冗余。第一代的编码技术仅考虑图像及图像序列中的空间冗余、时间冗余和信息的冗余。在第一代编码技术的基础上,进一步考虑视觉数据中的结构冗余、知识冗余和视觉冗余,产生了第二代编码技术。第二代图像编码方法建立在图像分析和合成、计算机图形学、计算机视觉等基础上,具有更高的压缩效率和更好的视觉效果。

目前关于音视频压缩的国际标准化主要有两个系列。国际电联 (ITU-T) 制定的应用于网络通信行业的 H. 26x 系列标准和国际标准化组织 (ISO) 运动图像专家组 (MPEG) 制定的应用于媒体业务的 MPEGx 系列标准。实际上,许多标准是两大国际标准组织合作产生的。比如被广泛使用的 H. 264 标准,最初被称为 H. 26L。它是 ITU-T 在 1999 年开始开发的一种新极低码率视频编码标准,旨在代替 H. 263 标准。2001 年,ISO MPEG 与 ITU-T 的 VCEG 合并,组成 JVT (Joint Video Team) 联合开发 H. 26L。经过工业界和学术界 3 年多的努力,2003 年 5 月正式批准为国际标准。在谷歌的 WebM 出现之前,H. 264 是最高效的一个视频压缩标准。

使用 H. 264 编码的 MPEG4 是一个在编码效率和编码质量上都较为优秀的视频格式标准,也是目前应用非常广泛的标准,但是由于专利收费的问题,一直存在争论。谷歌推出的 WebM 更加偏向于开源,采用了 On2 Technologies 开发的 VP8 及其后续版本 VP9 视频编解码器,至今 YouTube 已经逐渐将其 80% 的视频格式都变为 WebM 格式。本书第三章对 MPEG4 和 WebM 进行了比较和分析。

1.2.2 数据分发策略技术

流媒体数据分发已从最初的服务器客户端模式走向了更为复杂的多种模

式并存的现状。如果直接从流媒体服务器为用户提供服务,远距离用户的服务质量难以得到保证,同时由于互联网用户接入互联网是从多个服务商接入,IP地址的位置与地理位置并非对应的,即使是同一个城市的用户,从不同接入商接入的终端用户有可能得到截然不同的视频观看感受。因此,服务商致力于将视频资源尽力推向用户接入网内,以提高用户的服务质量感受。

数据存储和信道带宽是流媒体数据分发中涉及的主要资源,以 YouTube 为代表的服务商在全世界使用多个 CDN(Content Delivery Network)网络,将视频资源分发到位于各地的视频服务器群组,构建其视频资源的自组网。当用户请求视频时,通过 DNS 的重定向机制为用户提供最近的服务器上的资源。这种模式要求服务商具有雄厚的资金实例,可以承担巨额的服务器群组和信道租用费用。

为了减轻服务商的压力,可以采用 P2P 的模式,用户终端的播放器除了下载和播放视频,还为附近的用户提供视频数据上传功能。但是这种模式需要消耗用户的资源,一方面需要当地的政策法规允许,另一方面,不能损害终端用户的利益,往往在用户终端通过 WiFi 和有线网络接入的时候才能使用。此外,用户的隐私在这种模式下如何得以保证也是一个难题。

数据分发策略的制定既要考虑到尽量减轻服务商的成本,也要考虑到尽可能提高用户的服务感受,同时要遵循服务区域的法律法规,是多种因素权衡的结果,当前国内外出现了多种模式并存的现状。

1.2.3 数据传输技术

流媒体传输中,最初人们关注的是使得视频帧或者音频在规定的播放时间按时传输到用户终端,也就是如何实现实时的流媒体传输。在因特网中,传输层协议只有 TCP 和 UDP 两种,TCP 提供可靠的传输但是需要有三握手建立连接,在传输过程中根据链路的拥塞状态进行发送速率的调整,传输速率无法控制,因此,实时流媒体传输不能直接使用 TCP 传输。

RTP/RTCP/RTSP 协议族中,考虑到数据传输的实时性,使用 UDP 承载了流媒体数据。这种架构下的流媒体应用实现的复杂度高,在信道共享的因特网上,如果不能保证信道的可用带宽,服务质量就无法保证。随着终端缓存的使用,只要使用缓存优化技术,就可以使用 HTTP 协议下载数据片段,在终端通过解码器按照媒体的播放节奏实时播放。这种架构部署简单,对信道的稳定性要求低。

但是,由于 HTTP 协议使用 TCP 协议传输,TCP 协议最初是基于低带宽高延时的信道设计的,在当前的信道条件和应用需求下,TCP 协议的拥塞控制机制存在若干缺陷,使得其不适应当前高吞吐量需求的流媒体传输。为了使得