

原创

Mango酱

于 2020-08-25 20:31:00 发布

2644

收藏 7

版权

分类专栏:

FFMPEG

文章标签:

C++

网络

FFMPEG 专栏收录该内容

9 订阅

27 篇文章

订阅专栏

FFMPEG之RTP协议(实时传输协议)01

1 RTP与RTCP控制协议

1) RTP协议功能：

用于网络上实时传输音视频数据的标准数据包格式，例如流媒体、视频会议、电视服务等。

2) RTP为何需要RTCP配合使用：

RTP可以提供低延时的数据传送服务，但无法保证数据包到达客户端时仍然保持着发送时的顺序，所以要依靠RTCP来完成流量控制和拥塞监控。

3) RTP为何需要与序列号？

在OSI七层模型中，RTP协议运行在传输层，其他底层协议也可以和RTP协同工作。RTP数据包被UDP通过封装头形成UDP包。RTP本身未包含任何服务质量保证机制，因此RTP无法保持接收到的数据域传送顺序相同，也不去验证底层网络所提供的服务是否可靠。RTP通过在数据包中添加序列号，使得接收方能够在收到数据包后通过检测序列号对数据包重新排序，以达到有序的目的。

4) RTP为何需要时间戳？

在流媒体数据传输时，人们通常面临这同一个严峻的问题，那就是数据传送到客户端的时间点时无法预测的，但是，流媒体的传送要保证数据能够在合适的时间传送到目的地以完成数据正确的回放。因此，RTP协议中为了控制实时数据流传输而包含时间戳、序列号等结构。时间戳在流媒体传送的过程中扮演了重要的角色，为客户提供重要的数据。在发送RTP报文时，发送端在数据包里放置了用于记录采样时间的数据，称为时间戳，数据包经网络到达接收端后，接收端则需要从数据包中提取出该数据端，依照它来恢复原始的数据次序。RTP只是传输层协议，并不负责同步。RTP将这部分功能留给应用层来完成，已达到简化数据处理、提高运行效率的目的。
RTP的数据单元是用UDP分组来承载的，当然并不是简单地将一个RTP数据单元封装在UDP分组中，而是把一帧数据分割后放入多个UDP分组中传送，本属于同一帧数据的UDP分组将具有完全一样的时间戳。

5) 为何要使用RTCP：

在RTP会话期间，RTCP协议的作用主要是传送监控数据传送正确率的交换控制信息。在会话期间，与会者们将会发送一些相关的统计信息给其他用户，例如已发送的数据包的数量和无法传送的数据包的数量，这些信息的发送通常每隔相同的时间进行一次，在会话进行过程中周期性完成。
服务器将解析这些数据，并据此提高或降低数据的发送速率，也有可能换用其它类型的有效载荷。这些改变将是动态进行的。RTP配合RTCP运行，通过返回控制信息和减小带宽消耗提升传输效率，从而保证数据传输的时延最小。

6) RTCP的主要4个功能：

RTCP主要有4个功能：通过反馈分配数据的传送质量来进行拥塞控制、监视网络和诊断网络中的问题；由于SSRC（同步源标识）并不会一成不变，当网络拥塞发生时或者与会者程序发生变化时SSRC多会随之更新，我们需要为RTP源提供额外的传送层标志；调整RTCP包的发送速度，以保证数据包能顺利到达接收端，因此需要依据参与者的数据来进行调整；传送会话控制信息。

2 RTP的文件头格式

RTP报文由两部分组成：报头和有效载荷，头部共12个字节+CSRC。其中头部结构如下：

0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
V	P	X		CC				M	载荷类型							序号															
时间戳																															
同步源标识符 SSRC																															
贡献源标识符 CSRC																															

版本号（V）：2bit，用来标志使用的RTP版本。
填充位（P）：1bit，如果该位置位，则该RTP包的尾部就包含附加的填充字节。
扩展位（X）：1bit，如果该位位置的话，RTP固定头部后面就跟有一个扩展头部。
CSRC计数器（CC）：4bit，SSRC后的CSRC的数目。
标记位（M）：1bit,该位的解释由配置文档（Profile）来负责。
载荷类型（PayloadType）：7bit，标识了RTP载荷的类型。详见下表。
序列号（SN）：16bit，每发送一个RTP数据包，序列号增加1。接收端可以据此检测丢包和重建包序列。
时间戳(timestamp)：4byte，记录了该包中数据的第一个字节的采样时刻。在一次会话开始时，时间戳初始化成一个初始值。即使在没有信号发送时，时间戳的数值也要随时间而不断地增加。时钟频率依赖于负载数据格式，并在描述文件（profile）中进行描述。

同步源标识符(SSRC)：4byte，同步源就是指RTP包流的来源。在同一个RTP会话中不能有两个相同的SSRC值。该标识符是随机选取的RFC1889推荐了MD5随机算法。

贡献源列表（CSRC）：0~15项，每项32比特，用来标志对一个RTP混合器产生的新包有贡献的所有RTP包的源。由混合器将这些有贡献的SSRC标识符插入表中。SSRC标识符都被列出来，以便接收端能正确指出交谈双方的身份。

注意：

- 同步信源是指产生媒体流的信源，它通过RTP报头中的一个32位数字SSRC标识符来标识，而不依赖于网络地址，接收者将根据SSRC标识符来区分不同的信源，进行RTP报文的分组。
- 特约信源是指当混合器接收到一个或多个同步信源的RTP报文后，经过混合处理产生一个新的组合RTP报文，并把混合器作为组合RTP报文的SSRC，而将原来所有的SSRC都作为CSRC传送给接收者，使接收者知道组成组合报文的各个SSRC。
- 前面的12个字节是固定存在于每个rtp中，而后面的CSRC只有当混频器存在时才会被插入。

7位的载荷类型如下：

RFC3551：

PT	encoding name	audio/video (A/V)	clock rate (Hz)	channels (audio)
0	PCMU	A	8000	1
1	1016	A	8000	1
2	G721	A	8000	1
3	GSM	A	8000	1
4	unassigned	A	8000	1
5	DV14	A	8000	1
6	DV14	A	16000	1
7	LPC	A	8000	1
8	PCMA	A	8000	1
9	G722	A	8000	1
10	L16	A	44100	2
11	L16	A	44100	1
12	unassigned	A		
13	unassigned	A		
14	MFA	A	90000	(see text)
15	G728	A	8000	1
16-23	unassigned	A		
24	unassigned	V		
25	CelB	V	90000	
26	JPEG	V	90000	
27	unassigned	V		
28	rv	V	90000	
29	unassigned	V		
30	unassigned	V		
31	H261	V	90000	
32	MPV	V	90000	
33	MP2T	AV	90000	
34-71	unassigned	?		
72-76	reserved	N/A	N/A	N/A
77-95	unassigned	?		
96-127	dynamic	?		

发布首篇原创文章，
原力分+10，点亮新秀勋章

C++数据结构	30篇
C/C++功能函数	31篇
python学习	11篇
#模板学习	8篇
面试相关	8篇
C++新特性相关	12篇
C++11多线程编程	13篇
音视频流媒体开发	18篇
ONVIF开发	8篇
软件库下载与安装	18篇
Linux网络编程	28篇
Windows编程	1篇
GDB调试，相关命令及环...	21篇
FFMPEG	27篇
代码管理工具及相关命令	13篇
C++STL专栏	9篇
Go	32篇
JavaScript学习笔记	121篇
docker	7篇
WebRTC	
redis专题	11篇
MYSQL	36篇
Qt遇到的BUG	6篇
C++设计模式	4篇
黑马Qt笔记	61篇
Linux系统编程	83篇
Tcpdump与Wireshark抓包	2篇
Nginx源码分析	9篇
MFC教程	30篇
rapidJson库	5篇
Linux_C++_Server	1篇



Mango酱

码龄5年

暂无认证

670	17万+	8879	53万+	
原创	周排名	总排名	访问	等级

7920	211	321	175	1669
积分	粉丝	获赞	评论	收藏

















FFMPEG之RTP协议(实时传输协议)01 解决了你的问题么?
可以写篇文章记录加深印象哦~


[写文章](#)


搜博主文章


热门文章


0xC0000005: 写入位置 0x00000000 时发生访问冲突的解决办法(内存对齐) 


190959

Linux下安装openssl  16811

04-----无法执行二进制文件: 可执行文件格式错误  16274

02-----音频通道数、采样频率、采样位数、采样个数(样本数)的概念及计算一帧音频的大小、每秒播放的音频字节大小、一帧的播放时长、音频重采样  9958

智能指针之unique_ptr概述，初始化，成员函数用法，以及转化成shared_ptr/07 

 9909

最新评论

04-----关于Qt下编译大文件的源码时报lo...
狮子取名字: 谢谢, 我是编译生成时, 报的这错, 看来不是同一个

04-----关于Qt下编译大文件的源码时报lo...
Mango嘴: qt的spro文件

MFC之列表控件24

姜老不恰塞: m_list.SetBackColor(RGB(67, 67, 67)); // 设置背景颜色 m_list.SetTextB...
狮子取名字: 关于Qt下编译大文件的源码时报lo...
狮子取名字: 我是cmake openvc时遇到lo...

04-----关于Qt下编译大文件的源码时报lo...
狮子取名字: -Wa,-mbig-obj, 请问这个在哪里添加呢

您愿意向朋友推荐“博客详情页”吗？

强烈不推荐 不推荐 一般般 推荐 强烈推荐

最新文章

python基础----11-----闭包、装饰器、单例、工厂、多线程、socket、正则表达式、递归

python基础----10-----python操作mysql

python基础----09-----类、对象、魔法方法、封装、继承、类型注解、多态、数据分析案例

2023年 11篇	2022年 54篇
2021年 356篇	2020年 256篇

3 代码实现RTP头格式

```

1
2 #pragma pack(1)/1位字节对齐
3
4 //变量: 数字表示位与操作, 结构体字节的长度按照平时的方法计算即可
5 typedef struct RTP_FIXED_HEADER{
6     /* byte 0 */
7     unsigned char csrc_len:4;          /* expect 0 */
8     unsigned char extension:1;         /* expect 1 */
9     unsigned char padding:1;           /* expect 0 */
10    unsigned char version:2;            /* expect 2 */
11    /* byte 1 */
12    unsigned char payload:7;
13    unsigned char marker:1;             /* expect 1 */
14    /* bytes 2, 3 */
15    unsigned short seq_no;
16    /* bytes 4-7 */
17    unsigned long timestamp;
18    /* bytes 8-11 */
19    unsigned long ssrc;                 /* stream number is used here. */
20 } RTP_FIXED_HEADER;

```

Mango酱 关注 1 7 0 专栏目录

08-09

如何发送和接收RTP包, 用FFmpeg分离、解码 qqwangfan的专栏 · 1730

RTP是网络上进行流媒体传输的一种常用协议, 现在有很多封装RTP协议的开源库, 比如: `ortp`, `rtplib`, 而其中最著名的要数`rtplib`, 本文给大家演示怎...

可以看到rtcp time和rtp time都是以9000000为基,而rtcp nto time是为基,所以在用rtcp nto time时要注意基数的转换。

ffmpeg 推出的rtsp流解析_ffmpeg_rtp_泰勒朗斯的博客 9-5
ffmpeg 推出的rtsp流解析 通过命令: ffmpeg -re -i /home/zhe/Videos/12.mp4 -an -c copy -f rtp rtp://127.0.0.1:10086 推出rtsp流,抓包如下: 首先第一个rtsp流...

ffmpeg和ffplay 推送和播放RTP流 走错路的程序员 882

下面这个是把abc.mp4文件推送到rtsp://127.0.0.1:5004/abc.mp4端口,同时输出sdp文件到test_rtp_h264.sdp。-an参数的作用是只推送视频,没推送音频。

ffmpeg rtp传输使用 qianbo042311 的博客 3321

1 ffmpeg rtp ffmpeg估计已经成为多媒体 流媒体开发的必备工具，编解码不可缺少的伙伴，rtp协议本身是非常优良的协议，刚进入协议编程的伙伴可以使...

FFmpeg `rtsp` 和 `rtsp_mpegts` 的区别_HUI的技术笔记的博客 8-29

`rtsp_mpegts` 是 `ffmpeg` 中支持的唯一一个 `rtsp` muxer, 通过 `rtsp_mpegts` 发送音视频数据, 可以解决 `rtsp` 支持一路流的问题. 默认支持的 audio codec 是 AAC, video cod...

使用FFmpeg命令行进行UDP、RTP推流(H264、TS),使用ffplay/VLC接收验证... 8-30

使用FFmpeg命令行进行UDP、RTP推流(H264、TS),ffplay接收我们在开发网络程序时经常用到UDP或RTP来发送和接收流媒体,而开发程序完毕需要搭建...

利用FFmpeg实现简单rtsp推送

java+FFmpeg+JavaCV实现无控件HTML页面视频实时预览, 录像等, RTSP协议 04-20

ffmpeg学习3-- ffmpeg解码rtp流_SingleOneMan的博客 9-4

ffmpeg学习3-- ffmpeg解码rtp流 ffmpeg学习3-- ffmpeg解码rtp流 遇到的问题: 1.发送端编码效率不够,解码时接受的rtp包丢包概率很大,导致解码的视频帧卡...

协议学习方法、协议是什么：双方约定好如何传输消息，比如视频传输协议，要告诉你这个包是h264包，还是aac音频包，这个信息一般放在协议头，对...

是用于Internet上针对多媒体数据流的一种传输层协议。RTP协议和RTCP一起使用，而且它是建立在UDP协议上的。RTP不像http和ftp可...

RTP基础之RTP流预判代码实战 梦想 913

RTP基础介绍 实时传送协议 (Real-time Transport Protocol或简写RTP) 是一个网络**传输协议**, 其主要用于在互联网上传送音频和视频的标准数据包。RT...

如何发送和接收RTP封包的H264, 用FFmpeg解码 jinking01的专栏 · 1651

这篇文章给大家介绍怎么打包H264分片到RTP包, 并且实现一个播放程序演示接收RTP包和用FFmpeg解码。为了让大家更了解H264打包成RTP的细节, 这里

android视频聊天源码(使用RTP协议,ffmpeg解码) 01-08

视频传输RTP协议解析 最新发布 08-19
模拟获取一段数据头进行解析RTP头

<https://blog.csdn.net/zhoubotong2012/article/details/86711097>
<https://blog.csdn.net/zhoubotong2012/article/details/86711097>
 正文 首...

FFmpeg4.3系列之24: RTP及rtpplib发送H.264码流 09-01

敬告: 该系列的课程在抓紧录制更新中, 敬请关注。敬告: 该系列的课程涉及: FFmpeg, WebRTC, SRS, Nginx, Darwin, Live555, OpenCV等。包括: ...

ffmpeg监听和接收rtp音视频流 xiaowei的专栏 5200

在网上找了一圈关于ffmpeg如何接收rtp流的Example，都是使用ffmpeg 命令行来实现的，还没看到用相关ffmpeg库和API来实现的，于是自己写了个小程序...

从事流媒体开发的朋友经常会用到rtsp协议，今天我就udp荷载rtsp编码过程中经常遇到的重复包以及丢包如何判断说一下自己的经验。总体思路如下：1...

ffmpeg5.0解码rtsp协议传输的h264码流 06-07

要使用FFmpeg 5.0解码RTP协议传输的H.264码流，需要使用以下命令：`'''' ffmpeg -i rtsp://address:port -vcodec copy -f h264 -''''` 其中，address是RTP...

“相关推荐”对你有帮助么?

☐ 非常没帮助
 ☐ 没帮助
 ☐ 一般
 ☐ 有帮助
 ☐ 非常有帮助

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kfu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 家长监护 网络110报警服务 中国互联网举报中心 Chrome下载 账号管理规范 版权与免费声明 版权申诉 出版许可证 营业性广告 ©1999-2023北京创新乐知网络技术有限公司







