

昵称小猪
昵称0年

暂无认证

3633万+11万+30万+

原创周排名总排名访问等级

496011910355463

积分粉丝获赞评论收藏

头像

私信

关注

搜博文

文章

热门文章

linux中查看文件大小

QT编译报错: Cannot find file 问题解决

H264简介

软媒通信: 设备忙。请退出所有正在运行的应用程序。

抓包分析rtsp/rtp/h264

最新评论

Makefile生成多个可执行程序

CSDN-Ada助手: AI 写作助手上线啦! 限免4天, 快来创作试试功能吧 - https://edon...

QT编译报错: Cannot find file 问题解决

Wow.825. 谢谢

c调用c++接口, 报alice.hpp:31:19 fatal e...

zpb115. 你好请问具体是怎么解决的?

linux串口通信

题目好难做: ty84接收好像不行了 发了收...

软媒通信: 设备忙。请退出所有正在运行...

weixin_47118148. 还是不行啊

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

三角函数计算

Makefile生成多个可执行程序

Makefile编译静态库2

2023年1篇2022年131篇2021年89篇2020年42篇2019年38篇2018年80篇2017年12篇

RTP理解

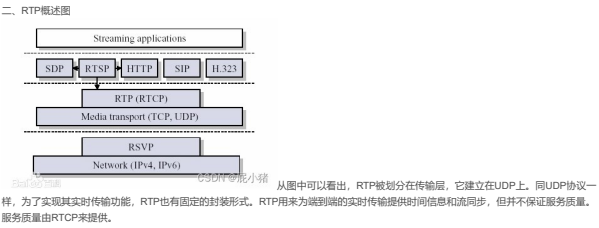
昵称小猪已于2022-06-15 10:51:01修改6643收藏35

分类专栏:音视频文章标签:网络linuxqt

音视频专栏收录文章内容7订阅16篇文章订阅专栏

一、RTP理解

RTP:实时传输协议 (Real-time Transport Protocol或简写RTP) 是一个网络传输协议。
RTP定义了两种报文: RTP报文和RTCP报文, RTP报文用于传送媒体数据 (如音频和视频) , 它由 RTP报头和数据两部分组成, RTP数据部分称为有效载荷(payload); RTCP报文用于传送控制信息, 以实现协议控制功能。RTP报文和RTCP 报文将作为下层协议的数据单元进行传输。如果使用UDP, 则RTP报文和RTCP报文分别使用两个相邻的UDP端口, RTP报文使用低端口, RTCP报文使用高端口。如果使用其它的下层协议, RTP报文和RTCP报文可以合并, 放在一个数据单元中一起传送, 控制信息在前, 媒体数据在后。通常, RTP是由应用程序实现的。



从图中可以看出, RTP被划分在传输层, 它建立在UOP上。同UDP协议一样, 为了实现其实时传输功能, RTP也有固定的封装形式。RTP用来为端到端的实时传输提供时间信息和流同步, 但并不保证服务质量。服务质量由RTCP来提供。

不少人也把RTP归为应用层的一部分, 这是从应用开发者的角度说的。操作系统中的TCP/IP等协议栈所提供的是我们最常用的服务, 而RTP的实现还是要靠开发者自己。因此从开发的角度来说, RTP的实现和应用层协议的实现没不同, 所以可将RTP看成应用层协议。RTP实现者在发送RTP数据时, 需先将数据封装成RTP包, 而在接收时RTP数据包, 需要将数据从RTP包中提取出来。

三、RTP报文格式

1.RTP报文头格式

RTP报文由两部分组成: 报头和有效载荷。

RTP报头格式其中:

IV: RTP协议的版本号, 占2位, 当前协议版本号为2。

IP: 填充标志, 占1位。如果P=1, 则在该报文的尾部填充一个或多个额外的八位组, 它们不是有效载荷的一部分。

X: 扩展标志, 占1位。如果X=1, 则在RTP报头后跟有一个扩展报头。

CC: CSRC计数器, 占4位, 指示CSRC标识符的个数。

IM: 标记, 占1位。不同的有效载荷有不同的含义, 对于视频, 标记一帧的结束; 对于音频, 标记会话的开始。

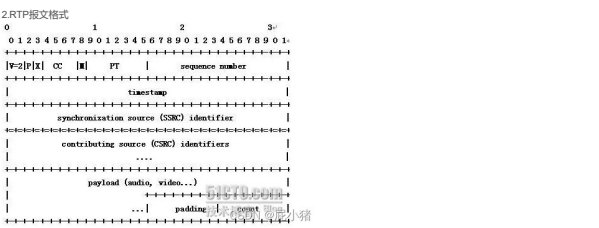
IP: 有效载荷类型, 占7位, 用于说明RTP报文中有效载荷的类型, 如GSM音频、JPEG图像等。

1序列号: 占16位, 用于标识发送者所发送的RTP报文的序列号, 每发送一个报文, 序列号增1。接收者通过序列号来检测报文丢失情况, 重新排序报文, 恢复数据。

1时戳(Timestamp): 占32位, 时戳反映了该RTP报文的第一个八位组的采样时刻。接收者使用时戳来计算延迟和延迟抖动, 并进行同步控制。



这里的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发送给接收者, 使接收者知道组成组合报文的各个SSRC



开始12个八进制数出现在每个RTP包中, 而CSRC标识列表仅出现在混合器插入时。各段含义如下:

①版本 (V)

2位, 标识RTP版本。

②填充标识 (P)

1位, 如设置置位, 在包尾将包含附加填充字, 它不属于有效载荷。填充的最后八个八进制包含应该忽略的八进制计数。某些加密算法需要固定大小的填充字, 或为在底层协议数据单元中携带几个RTP包。

③扩展 (X)

1位, 如设置扩展位, 固定头后跟一个头扩展。

④CSRC计数 (CC)

4位, CSRC计数包括紧接在固定头后CSRC标识符个数。

⑤标记 (M)

1位, 标记解释由设置定义, 目的在于允许重要事件在包中标记出来。设置可定义其他标示位, 或通过改变变量数量来指定没有标示位。

⑥载荷类型 (PT)

7位, 记录后面资料使用哪种 Codec , receiver 端找出相应的 decoder 解码出来。

常用 types:

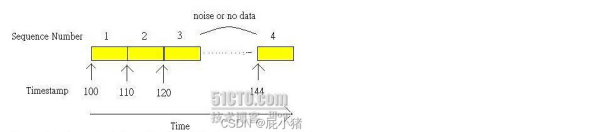
Payload Type	Codec
0	PCMU -Law
8	PCMA -Law
9	G.. 722 audio codec
4	G.. 723 audio codec
15	G.. 728 audio codec
18	G.. 729 audio codec
34	G.. 763 audio codec
31	G.. 761 audio codec

⑦系列号

16位, 系列号随每个RTP数据包而增加1, 由接收者用来探测包丢失。系列号初值是随机的, 使对加密的文本攻击更加困难。

⑧时标

32位, 时标反映RTP数据包中第一个八进制数的采样时刻, 采样时刻必须从单调、线性增加的时钟导出, 允许同步与抖动计算。时标可以让receiver知道在正确的时间将资料播放出来。



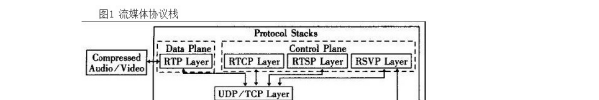
由上图可知, 如果只有系列号, 并不能完整按照顺序的将数据播放出来, 因为如果data中间有一段是没有资料的, 只有系列号的话会造成错误, 需搭配上它知道在哪个时间将data正确播放出来, 如此我们才能播放出正确无误的信息。

⑨SSRC

32位, SSRC段标识同步源。此标识不是随机选择的, 目的在于使同一RTP包连接中没有两个同步源有相同的SSRC标识。尽管多个源选择同一个标识的概率很低, 所有RTP实现都必须探测并解决冲突。如源改变源传输地址, 也必须选择一个新SSRC标识以避免插入成环行源。

⑩CSRC列表

0到15项, 每项32位。CSRC列表表示包内的对载荷起作用的源。标识数量由CC段给出, 如超出15个作用源, 也仅标识15个。CSRC标识由混合器插入, 采用作用源的SSRC标识。



XM

高效
极速执行力

获多重监督的经纪商

建立账户

投资交易具有较大风险并可能导致亏损的投资本金亏损, 请谨慎与慎重运用。

分类专栏

Qt (1篇)56篇

笔记 (1篇)40篇

工具36篇

网络1篇

git1篇

Makefile3篇

Linux32篇

音视频16篇

zlog1篇

小瓶源

linux c23篇

cmake1篇

小双光3篇

qml22篇

麒麟系统1篇

OGC1篇

python36篇

MFC3篇

C/c++ 笔试题32篇

设计模式5篇

android4篇

C++57篇

工作中问题24篇

C语言面试题18篇

四、流媒体协议栈

实时传输协议RTP（Real-Time Transport Protocol）：

RTP是针对Internet上多媒体数据流的一个传输协议。由IETF（Internet工程任务组）作为RFC1889发布，RTP被定义为在一对一或一对多的传输情况下工作，其目的是提供时间信息和实现流同步。RTP的典型应用建立在UDP上，但也可以在TCP或ATM等其他协议上工作。RTP本身只保证实时数据的传输，并不能为按顺序传送数据包提供可靠的传送机制，也不提供流量控制或拥塞控制，它依靠RTCP提供这些服务。

实时传输控制协议RTCP(Real-Time Transport Control Protocol)

RTCP负责管理传输质量在当前应用进程之间交换控制信息。在RTP会话期间，各参与者周期性地传送RTCP包，包中含有已发送的数据包的数量、丢失的数据包的数量等统计资料。因此，服务器可以利用这些信息动态地改变传输速率，甚至改变有效载荷类型。RTP和RTCP配合使用，能以有效的反馈和最小的开销使传输效率最佳化，故特别适合传送网上的实时数据。

五、RTCP报文格式

1.RTCP工作机制

当应用程序开始一个rtcp会话时将使用两个端口：一个给rp，一个给tcp。rtcp本身并不能为按顺序传送数据包提供可靠的传送机制，也不提供流量控制或拥塞控制，它依靠rtcp提供这些服务。在rtcp的会话之间周期的发送一些tcp包以用来传输服务质量并交换会话用户信息等功能。rtcp包中含有已发送的数据包的数量、丢失的数据包的数量等统计资料。因此，服务器可以利用这些信息动态地改变传输速率，甚至改变有效载荷类型。rtcp和rtcp配合使用，它们能以有效的反馈和最小的开销使传输效率最佳化，因而特别适合传送网上的实时数据。根据用户间的数据传输反馈信息，可以制定流量控制的策略，而会话用户信息的交互，可以制定会话控制的策略。

2.RTCP数据包

在RTCP通信控制中，RTCP协议的功能是通过不同的RTCP数据报来实现的，主要有如下几种类型：

①SR-发送端报告，所需发送端是指发出RTP数据报的应用程序或者终端，发送端同时也可以接收类型。

②RR-接收端报告，所需接收端是指仅接收但不发送RTP数据报的应用程序或者终端。

③SDS-源描述，主要功能是作为会话成员有关标识信息的载体，如用户名、邮件地址、电话号码等，此外还具有向会话成员传达会话控制信息的功能。

④BYE-通知离开，主要功能是指示某一个或者几个源不再有效。即通知会话中的其他成员自己将退出会话。

⑤APP-由应用程序自己定义，解决了RTCP的扩展性问题，并且为协议的实现者提供了很大的灵活性。

3.RTCP的封装

RTP需要RTCP为其服务质量提供保证，因此下面介绍一下RTCP的相关知识。

RTCP的主要功能是：服务质量的监视与反馈、媒体间的同步，以及多播组中成员的标识。在RTP会话期间，各参与者周期性地传送RTCP包。RTCP包中含有已发送的数据包的数量、丢失的数据包的数量等统计资料，因此，各参与者可以利用这些信息动态地改变传输速率，甚至改变有效载荷类型。RTP和RTCP配合使用，它们能以有效的反馈和最小的开销使传输效率最佳化，因而特别适合传送网上的实时数据。

RTCP也是用UDP来传送的，但RTCP封装的仅仅是一些控制信息，因而分组很短，所以可以将多个RTCP分组封装在一个UDP包中。

RTCP有如下五种分组类型

类型	缩写表示	用途
200	SR（Sender Report）	发送端报告
201	RR（Receiver Report）	接收端报告
202	SDS（Source Description Items）	源点描述
203	BYE	结束传输
204	APP	特定应用

表 1 RTCP的5种分组类型

下面只讲述SR类型

发送端报告分组SR（Sender Report）用来使发送端以多播方式向所有接收端报告发送情况。SR分组的主要内容有：相应的RTP流的SSRC，RTP流中最新产生的RTP分组的序列号和NTP，RTP流包含的分组数，RTP流包含的字节数。SR包的封装如图所示

0	8	16	24	32
V=2	P	RC	PT=SR=200	Length
SSRC of sender				
NTP timestamp, as it is significant				
NTP timestamp, least significant				
RTP timestamp				
Sender's packet count				
Sender's octet count				
SSRC_1 (SSRC of first source)				
Fraction lost	Cumulative number of packets lost			
Extended high-resolution sequence number received				
Interarrival jitter				
Last SR (LSR)				
Delay since last SR (DLSR)				
SSRC_2 (SSRC of second source)				
...				
Profile-specific extensions				

图 3 RTCP头部的格式

版本（V）：同RTP包头域。

填充（P）：同RTP包头域。

接收报告计数（RC）：5比特，该SR包中的接收报告块的数目，可以为零。

包类型（PT）：8比特，SR包是200。

长度域（Length）：16比特，其中存放的是该SR包以32比特为单位的总长度减一。

同步源（SSRC）：SR包发送者的同步源标识符，与对应RTP包中的SSRC一样。

NTP Timestamp（Network time protocol）SR包发送时的绝对时间值。NTP的作用是同步不同的RTP媒体流。

Sender's packet count：从开始发送包到产生这个SR包这段时间里，发送者发送的RTP数据包的总数。SSRC改变时，这个域清零。

Sender's octet count：从开始发送包到产生这个SR包这段时间里，发送者发送的净荷数据的总字节数（不包括头部和填充）。发送者改变其SSRC时，这个域要清零。

同步源的SSRC标识符：该报告块中包含的是从该源接收到的包的统计信息。

丢失率（Fraction Lost）：表明从上一个SR或RR包发出以来从同步源n(SSRC_n)来的RTP数据包的丢失率。

累计的包丢失数目：从开始接收到SSRC_n的包到发送SR从SSRC_n传过来的RTP数据包的丢失总数。

收到的扩展最大序列号：从SSRC_n收到的RTP数据包中最大的序列号。

接收抖动（Interarrival jitter）：RTP数据包接受时间的统计方差估计

上次SR时间戳（Last SR,LSR）：取最近从SSRC_n收到的SR包中的NTP时间戳的中间32比特。如果目前还没收到SR包，则该域清零。

上次SR以来的延时（Delay since last SR,DLSR）：上次从SSRC_n收到SR包到发送本报告的延时。

4.RTP的会话过程

当应用程序建立一个RTP会话时，应用程序将确定一对目的传输地址。目的传输地址由一个网络地址和一对端口组成。有两个端口：一个给RTP包，一个给RTCP包，使得RTP/RTCP数据能够正确发送。RTP数据发向偶数的UDP端口，而对应的控制信号RTCP数据发向相邻的奇数UDP端口（偶数的UDP端口+1），这样就构成一个UDP端口对。RTP的发送过程如下，接收过程则相反。

1. RTP协议从上层接收流媒体信息码流（如H.263），封装成RTP数据包；RTCP从上层接收控制信息，封装成RTCP控制包。
2. RTP将RTP数据包发往UDP端口对中偶数端口；RTCP将RTCP控制包发往UDP端口对中的接收端口。

六、相关协议

RTSP是一个应用层协议（TCP/IP网络体系中），它以C/S模式工作，它是一个多媒体播放控制协议，主要用来使用户在播放流媒体时可以像操作本地的影碟机一样进行控制，即可以对流媒体进行暂停/继续、后退和前进等控制。

资源预定协议RSVP工作在IP层之上传输层之下，是一个网络控制协议。RSVP通过在路由器上预留一定的带宽，能在一定程度上为流媒体的传输提供服务质量。在某些试验性的系统如网络视频会议工具vc中就集成了RSVP。

七、常见的疑问

1.怎样重组乱序的数据包

可以根据RTP包的序列号来排序。

2.怎样获得数据包的时序

可以根据RTP包的时间戳来获得数据包的时序。

3.声音和图像怎么同步

根据声音流和图像流的相对时间（即RTP包的时间戳），以及它们的绝对时间（即对应的RTCP包中的RTCP），可以实现声音和图像的同步。

4.接收缓冲和播放缓冲的作用

接收缓冲用来排序乱序了的数据包；播放缓冲用来消除播放的抖动，实现等时播放。

5.为什么RTP可以解决上述实时问题

RTCP协议是一种基于UDP的传输协议。RTP本身并不能为按顺序传送数据包提供可靠的传送机制，也不提供流量控制或拥塞控制，它依靠RTCP提供这些服务。这样，对于那些丢失的数据包，不存在由于超时检测而带来的延时，同时，对于那些丢弃的包，也可以由上层根据其重要性选择性的重传。比如，对于帧、P帧、B帧数据，由于其重要性依次降低，故在网络状况不好的情况下，可以考虑在B帧丢失甚至P帧丢失的情况下不进行重传，这样，在客户端方面，虽然可能会有短暂的不清画面，但却保证了实时性的体验和要求。

八、IP/TCP/UDP/RTP/RTCP包结构图

IP包头结构:

IP 包头结构:																																															
0								1								2								3																							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																
+-+																																															
Version								IHL								Type of Service								Total Length																							
+-+																																															
																Identification								Flags								Fragment Offset															
+-+																																															
																Time to Live								Protocol								Header Checksum															
+-+																																															
																Source Address																															
+-+																																															
																Destination Address																															
+-+																																															
																Options																Padding															
+-+																																															

CSDE 学院小站

TCP 包头结构:

