

计算机网络

实验五：HTTP服务器实验

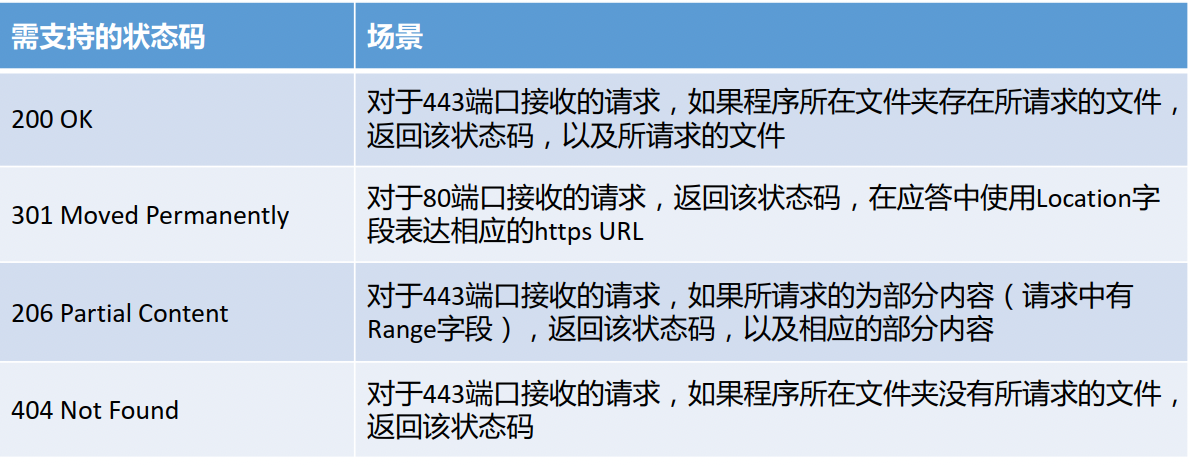
班 级：计算机学院702

实验成员：左天宇，

黄磊，2022E8013282156

日期：2022年11月5号

1. **实验要求**
2. 使用C语言实现简单的HTTP服务器，同时支持http 80端口和https 443端口，并使用两个线程分别监听；
3. 只需要支持GET方法，解析请求报文，返回应答和内容。需要支持的状态码如下：



1. 使用HTTP服务器分发视频，并使用vlc通过网络获取并播放。
2. **实验内容**
3. **HTTP消息结构**

HTTP基于TCP/IP协议进行数据传递，HTTP协议是建立在TCP协议上的一种应用。HTTPS是在HTTP上建立SSL加密层，并对传输数据进行加密，是HTTP协议的安全版。在服务器与客户端建立连接的过程中，会经历三次握手，成功后则开始传输实体内容。

HTTP的请求格式如下：



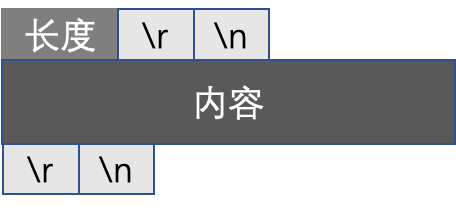
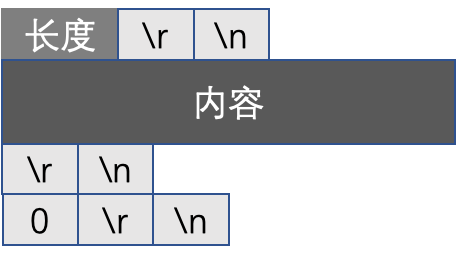
响应格式如下：



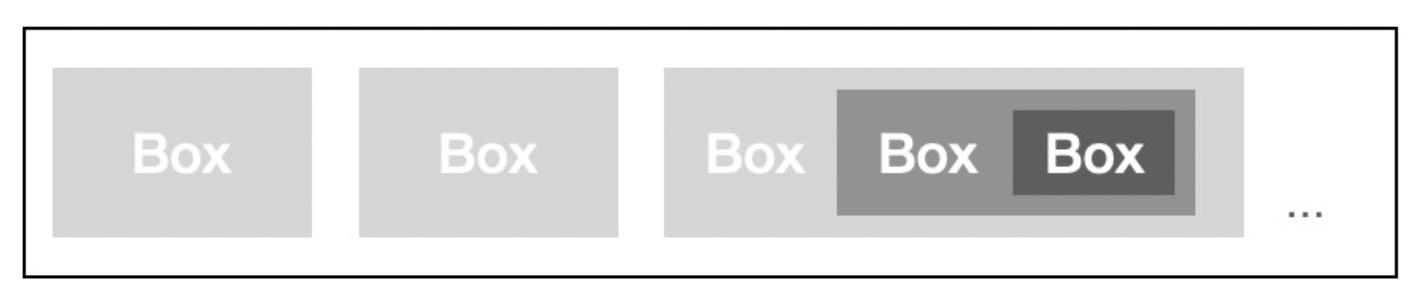
当实体较大，一次报文难以完全传输的时候，请求头部的中会有Connection:keep-alive以建立长连接，并采用设置响应头Content-Length或者Transfer-Encoding:chunked的方法来解析消息体。在具体实现中，即当实体较小的时候，没有Range信息，直接一次性传输，返回200 OK状态码；当实体较大的时候， Range显示为“n- ”，此时返回实体总长度，并进行分块传输，Range显示为“n-m”，过程中返回206 Partial Content。

1. **Chunked 编码**

Chunked编码一般使用若干个chunk串连而成，最后由一个标明长度为0的chunk标示结束。每个chunk分为头部和正文两部分，头部内容指定下一段正文的字符总数（非零开头的十六进制的数字）和数量单位（一般不写,表示字节）.正文部分就是指定长度的实际内容，两部分之间用回车换行(CRLF)隔开。在最后一个长度为0的chunk中的内容是称为footer的内容，是一些附加的[Header](https://so.csdn.net/so/search?q=Header&spm=1001.2101.3001.7020" \t "_blank)信息（通常可以直接忽略）。简而言之，chunked编码的基本方法是将大块数据分解成多块小数据，每块都可以自指定长度。

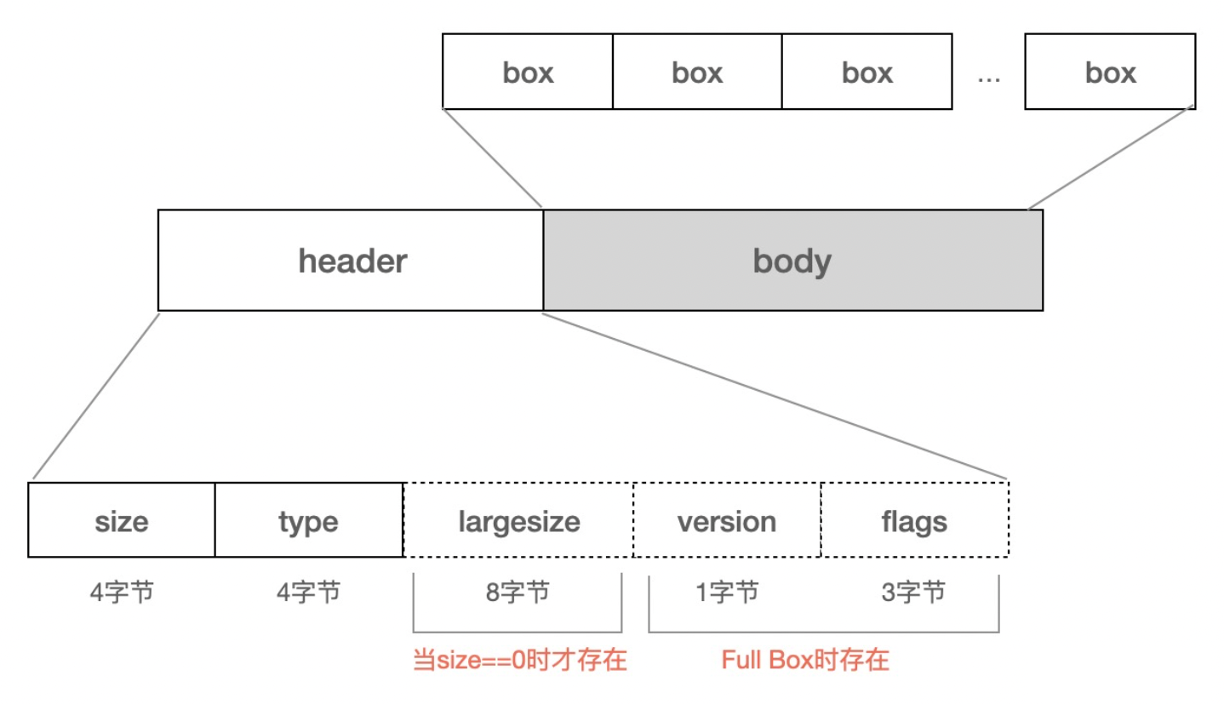
 对于实体分块部分，除最后一块的其他分块的数据格式如下图 (a) 所示；当最后一块传输的时候，会以来进行标识，数据块如下图 (b) 所示：

1. (b)
2. **MP4文件格式**

Mp4视频文件由多个box组成，每个box含有不同信息，且box之间是树状结构：

并且，box的类型有很多，常用的顶层box为：

1. ftyp：File Type Box，描述文件遵从的MP4规范与版本；
2. moov：Movie Box，媒体信息，仅有一个；
3. mdat：Media Data Box，存放的媒体数据实体，一般含有多个。

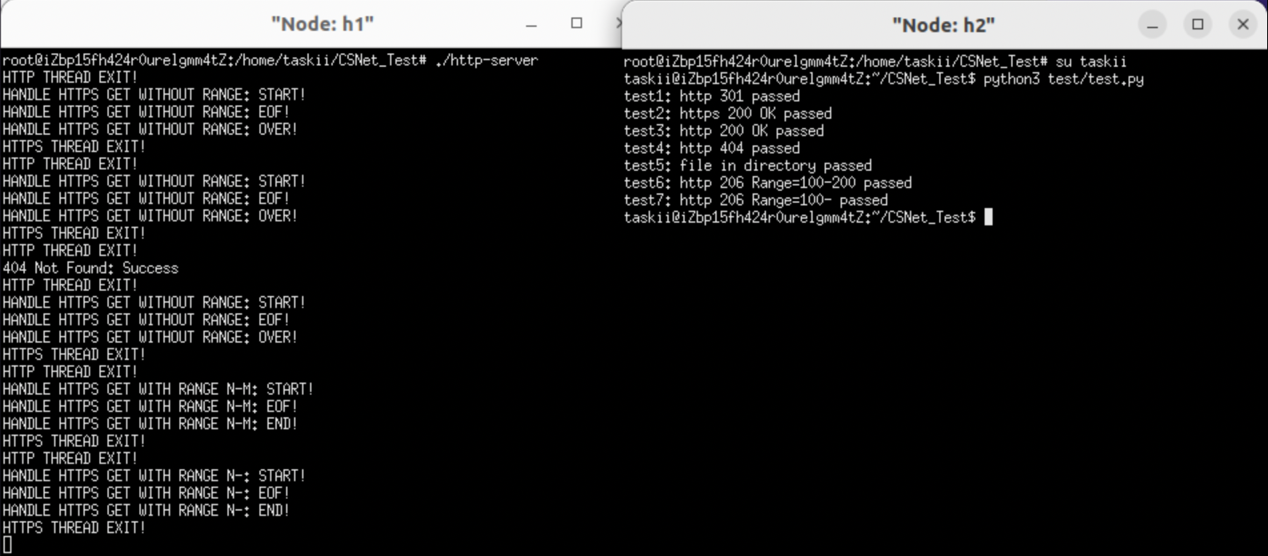
而每个box由头部（header）和实体（body）组成，头部只有size和type关键词是必须字段，其他字段在特殊情况下会被包含：

1. **实验结果**

在运行之前，需要进行签名，运行 sudo sh ./installcert.sh 以完成签名。

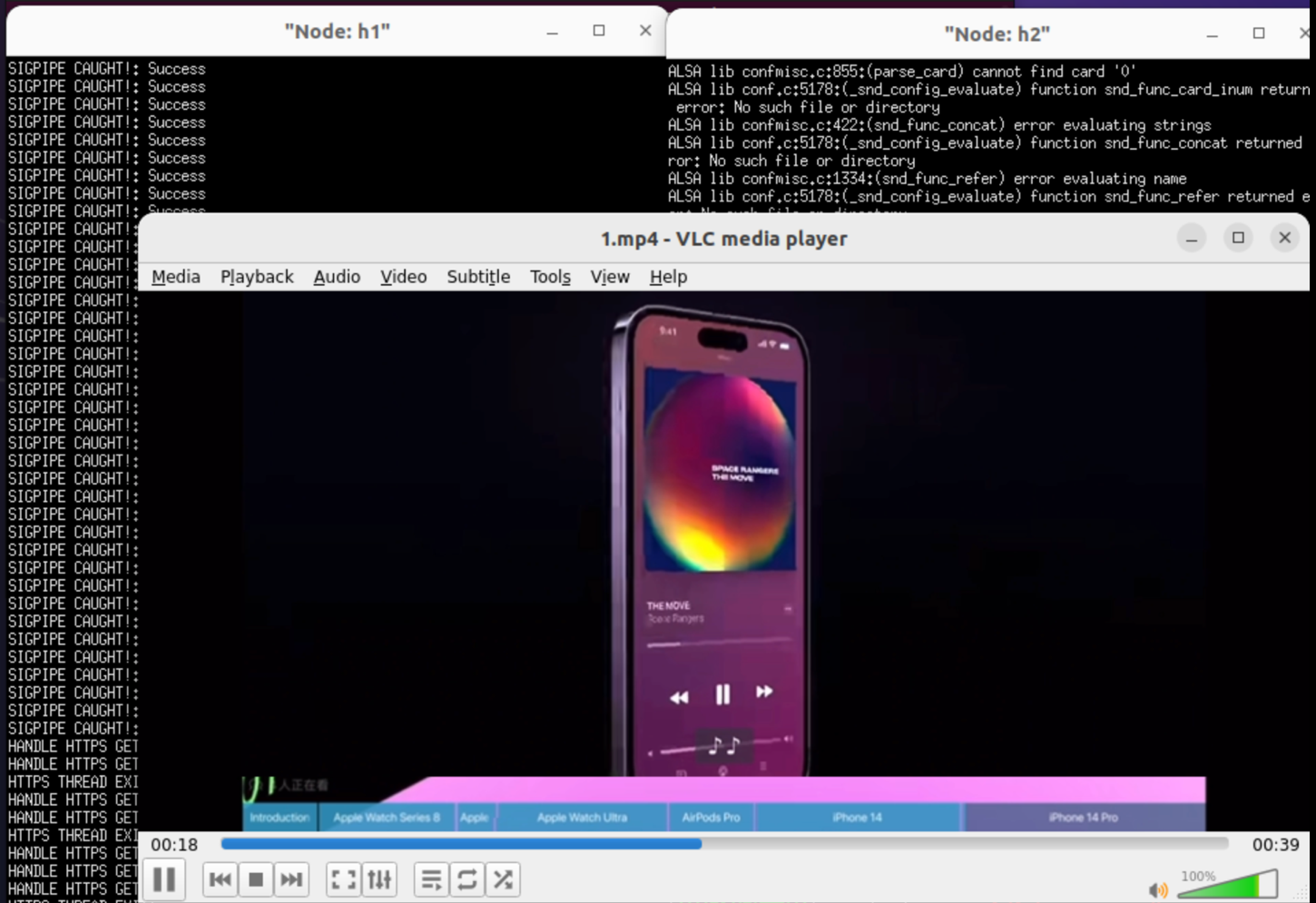
1. **测试文件test.py结果**

为了显示是否通过，在代码中加上了print以作显式打印，得到结果如下：



从上图可以看出，各个测试都通过。

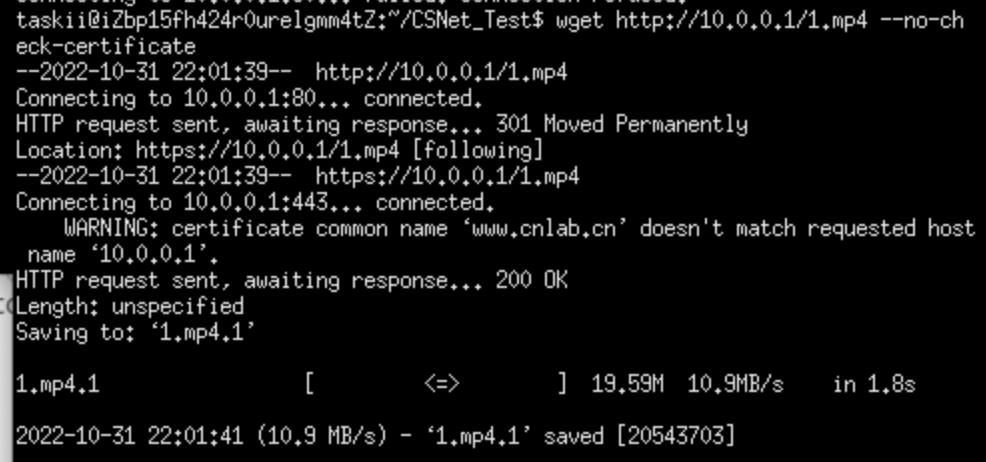
1. **VLC从网络获取视频文件并播放**

直接在VLC中使用Play Network Stream中输入<https://10.0.0.1/1.mp4>，点击播放，可见视频播放情况良好（视频采用苹果发布会iPhone14灵动岛片段，压缩后大小约为20M）：

另外，为了测试80端口对443端口的重定向，使用

wget <http://10.0.0.1/1.mp4> --no-check-certificate

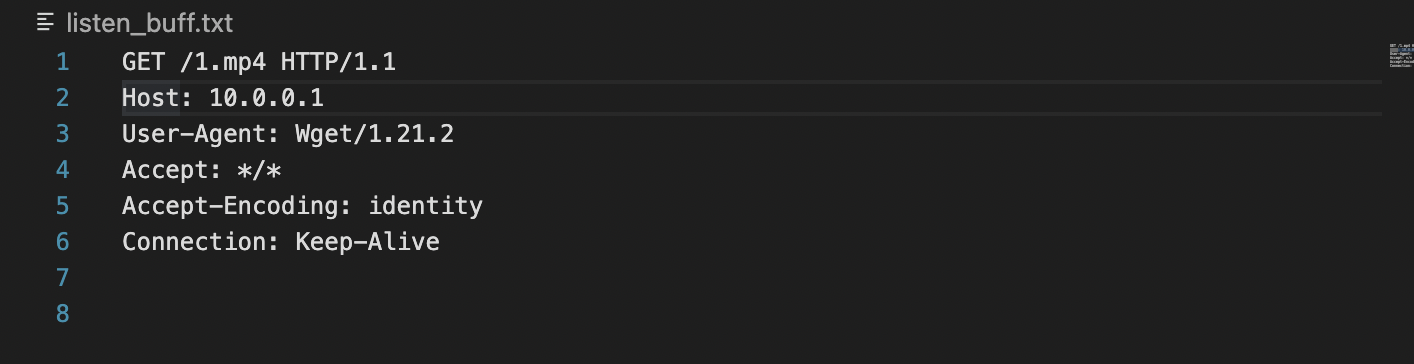
命令行测试得到：

可以看出，其连接到<http://10.0.0.1> 之后，得到301 Moved Permanently，之后重定向到 <https://10.0.0.1> 并下载视频文件，得到200 OK状态码。

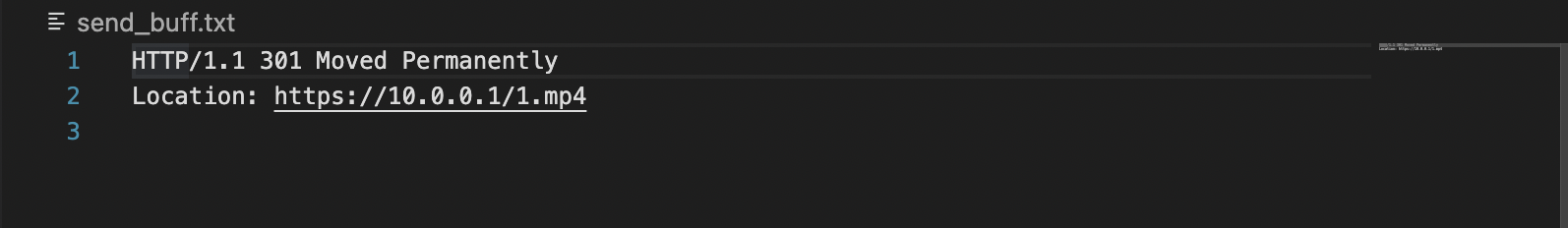
1. **SSL内容分析**

我们设置每次分块大小最大为4000字节，并为了显示报文是否正确，我们将除内容实体外的报文其他部分打印出来，包括首部、尾部和一些关键词，说明程序运行正确，符合预期。

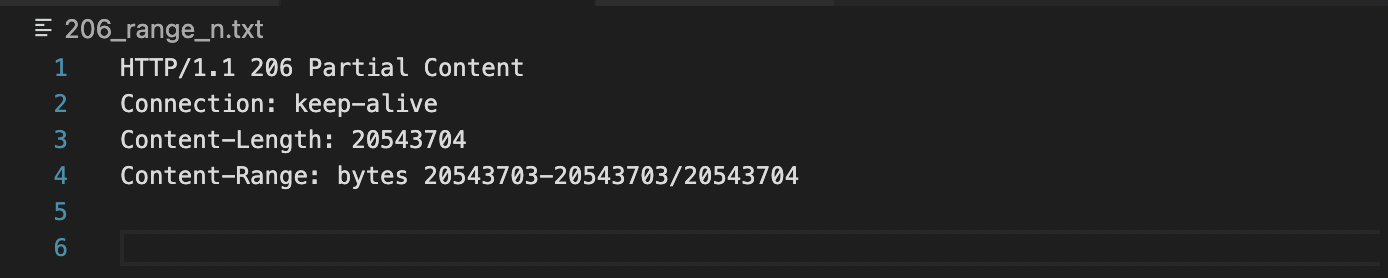
请求报文：



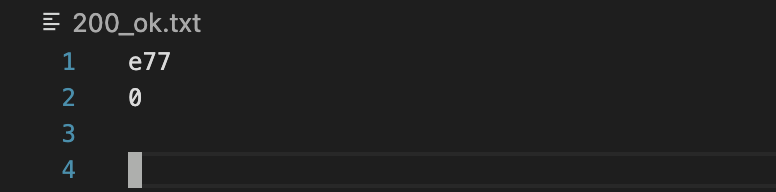
响应报文：



实体首部字段（带有Range）：



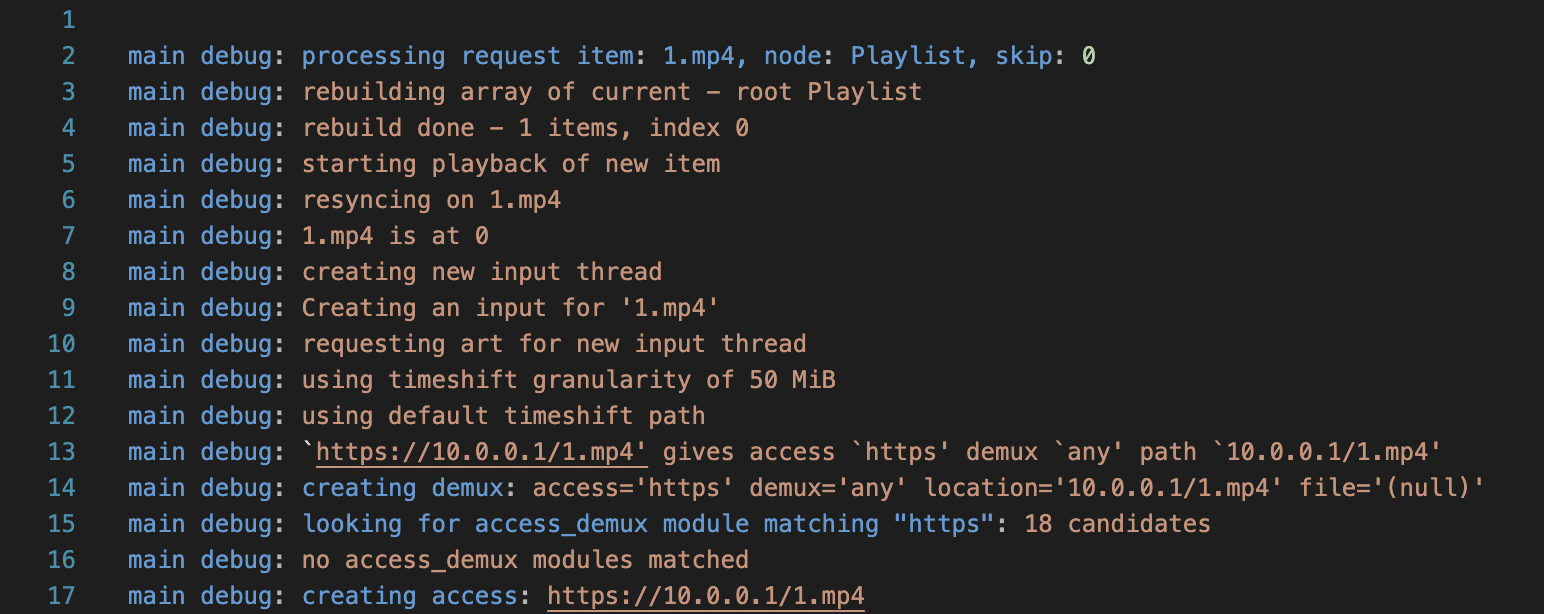
当最后一部分传输结束之后，会发送带有结束符的分块：



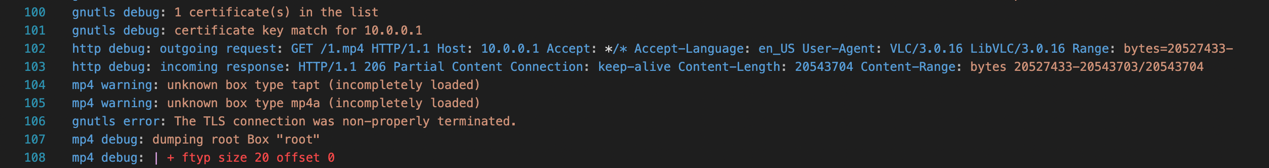
说明最后一个分块字节数为3703。

1. **VLC抓取内容分析**

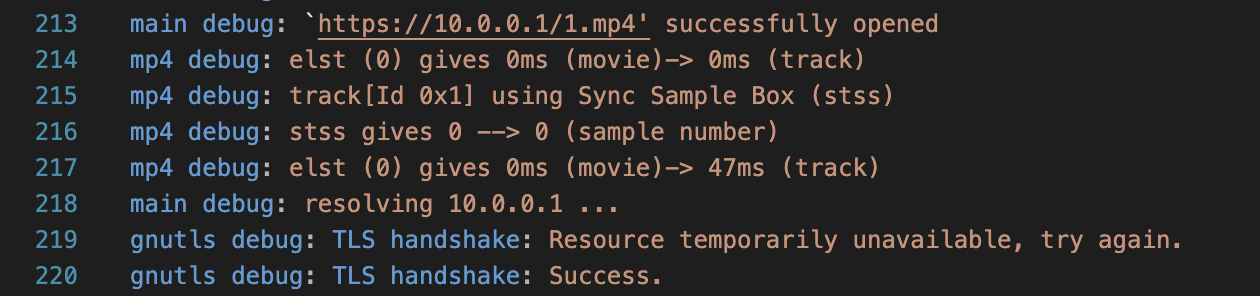
建立连接并定位文件：



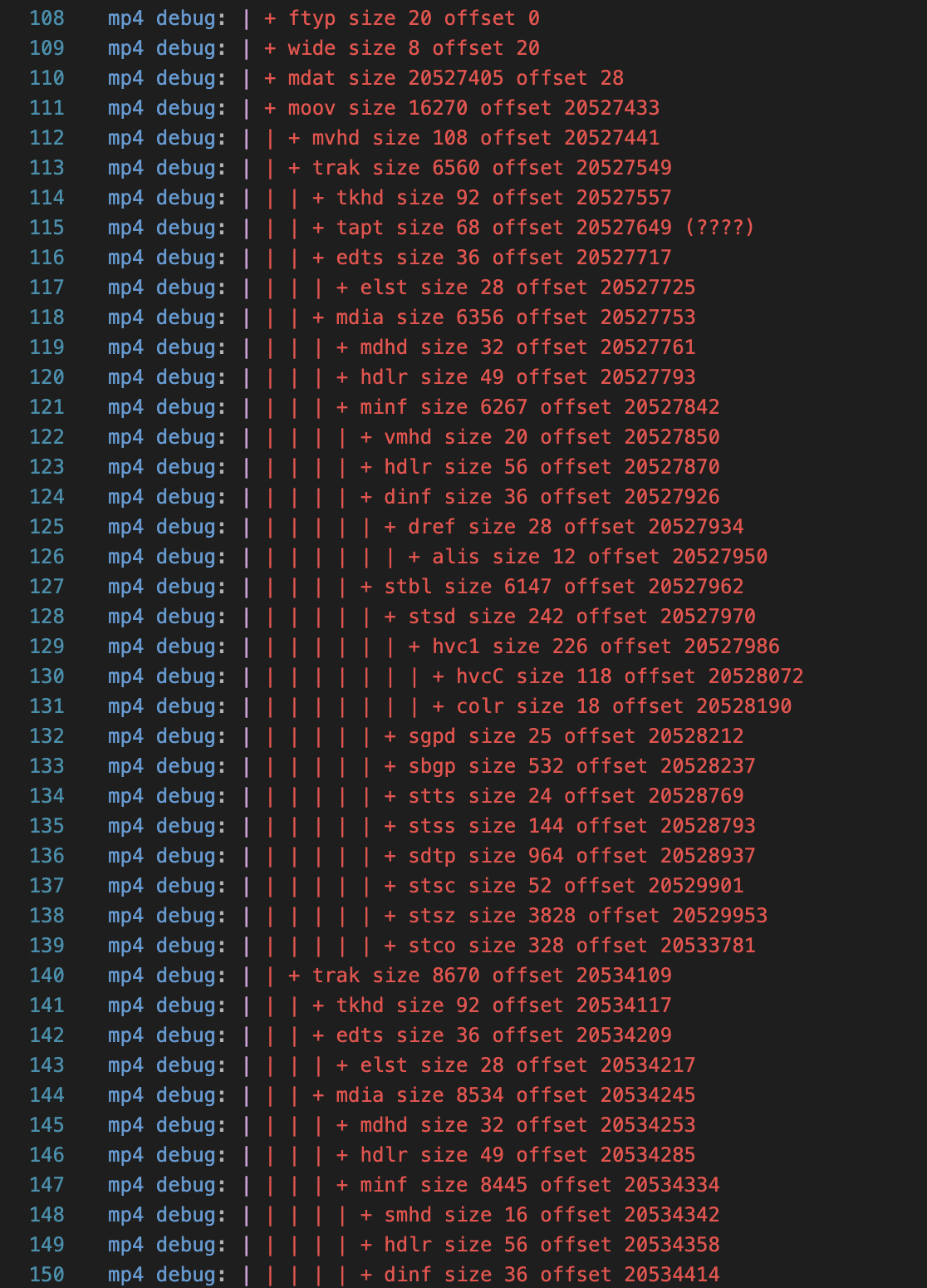
抓取文件并得到range，可见VLC会读取范围值，当未指定Range又一次发送不完的时候，Range只有一边：

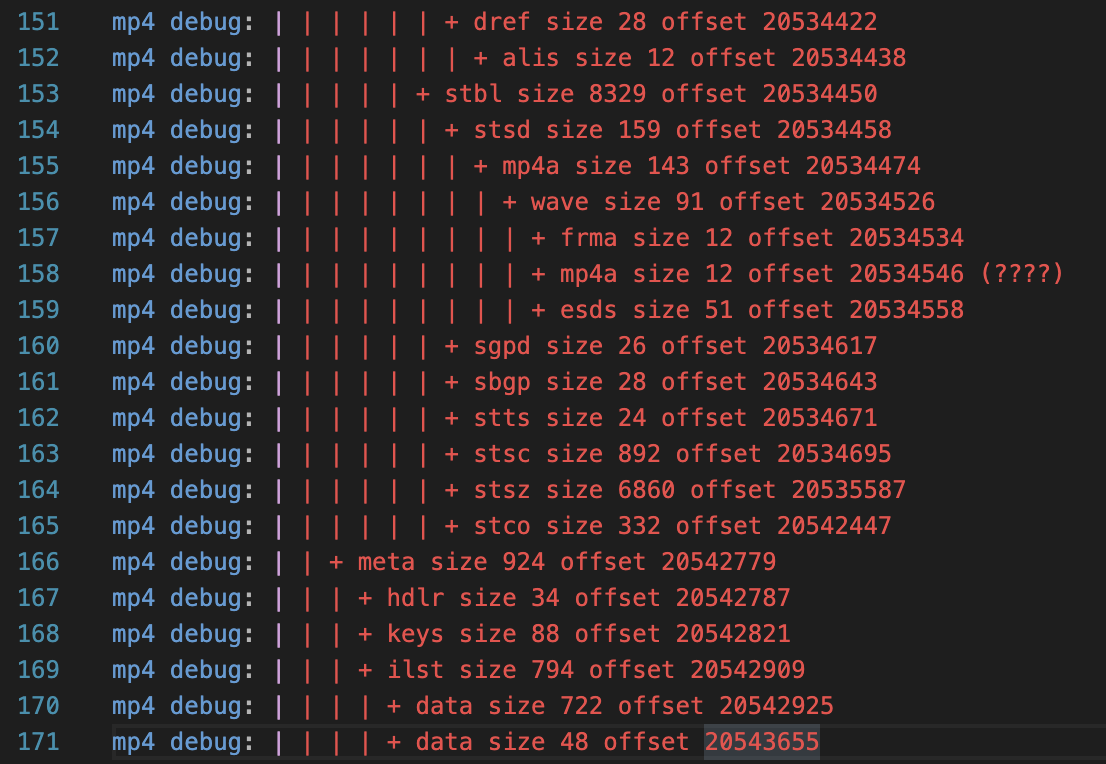


之后分块传输，在VLC上打开视频文件：

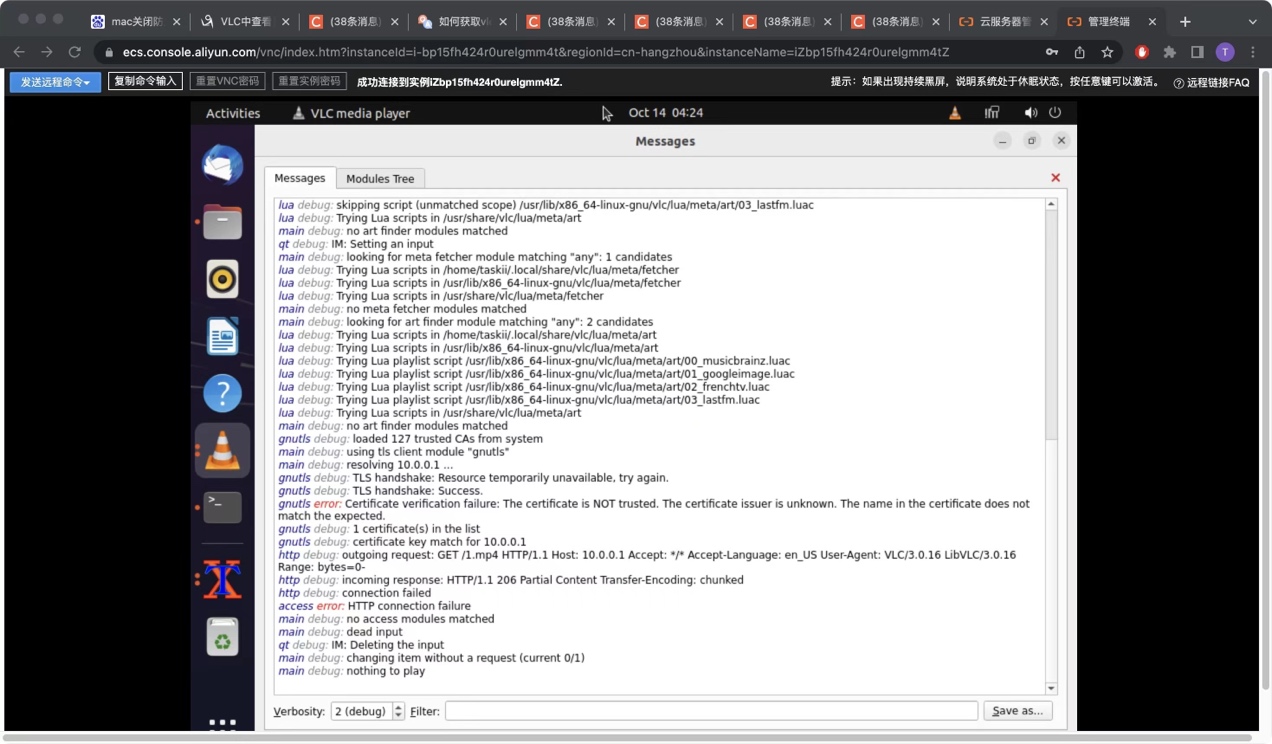


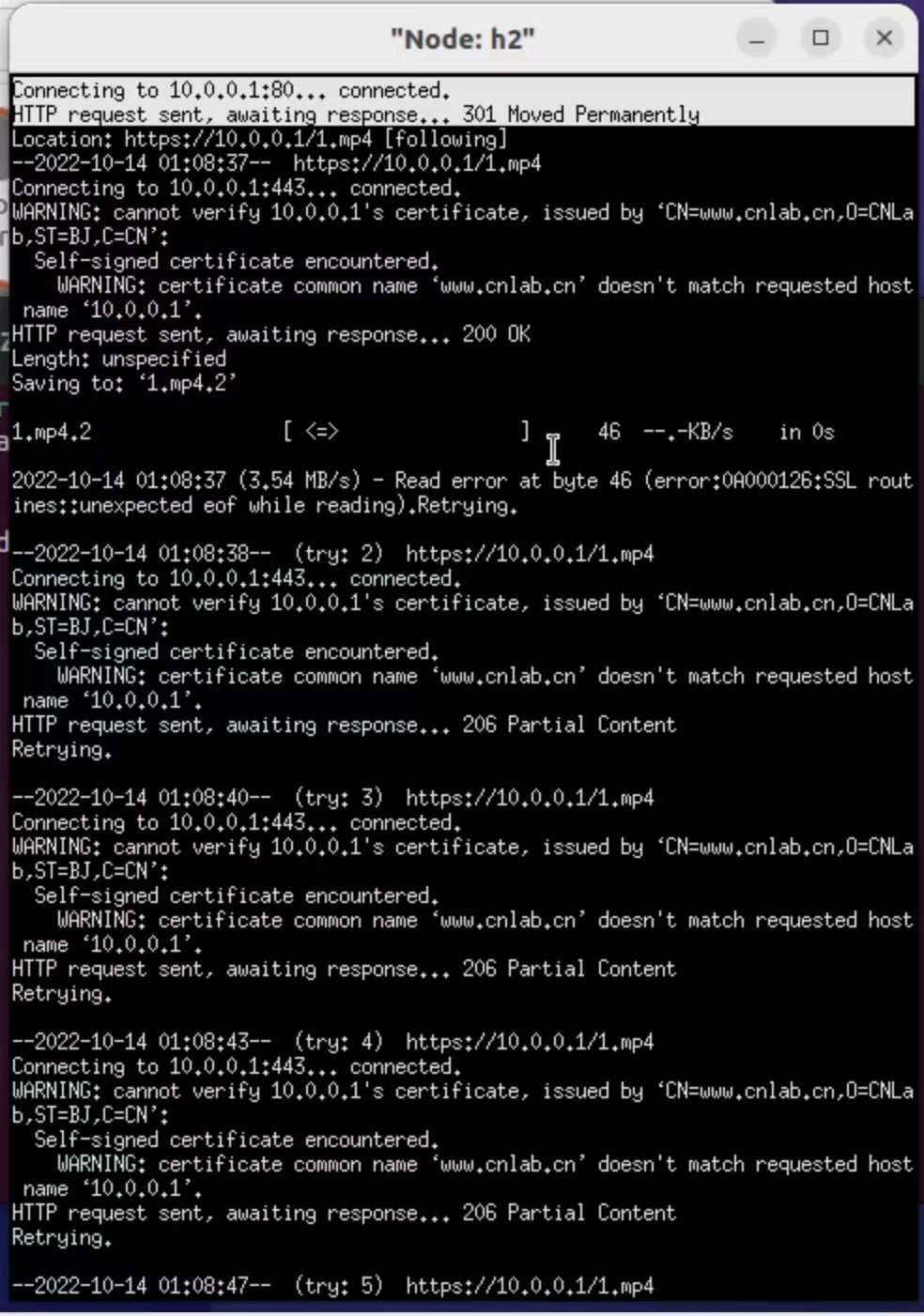
另外可见视频文件格式如下图，可见与上述视频文件格式相符合。



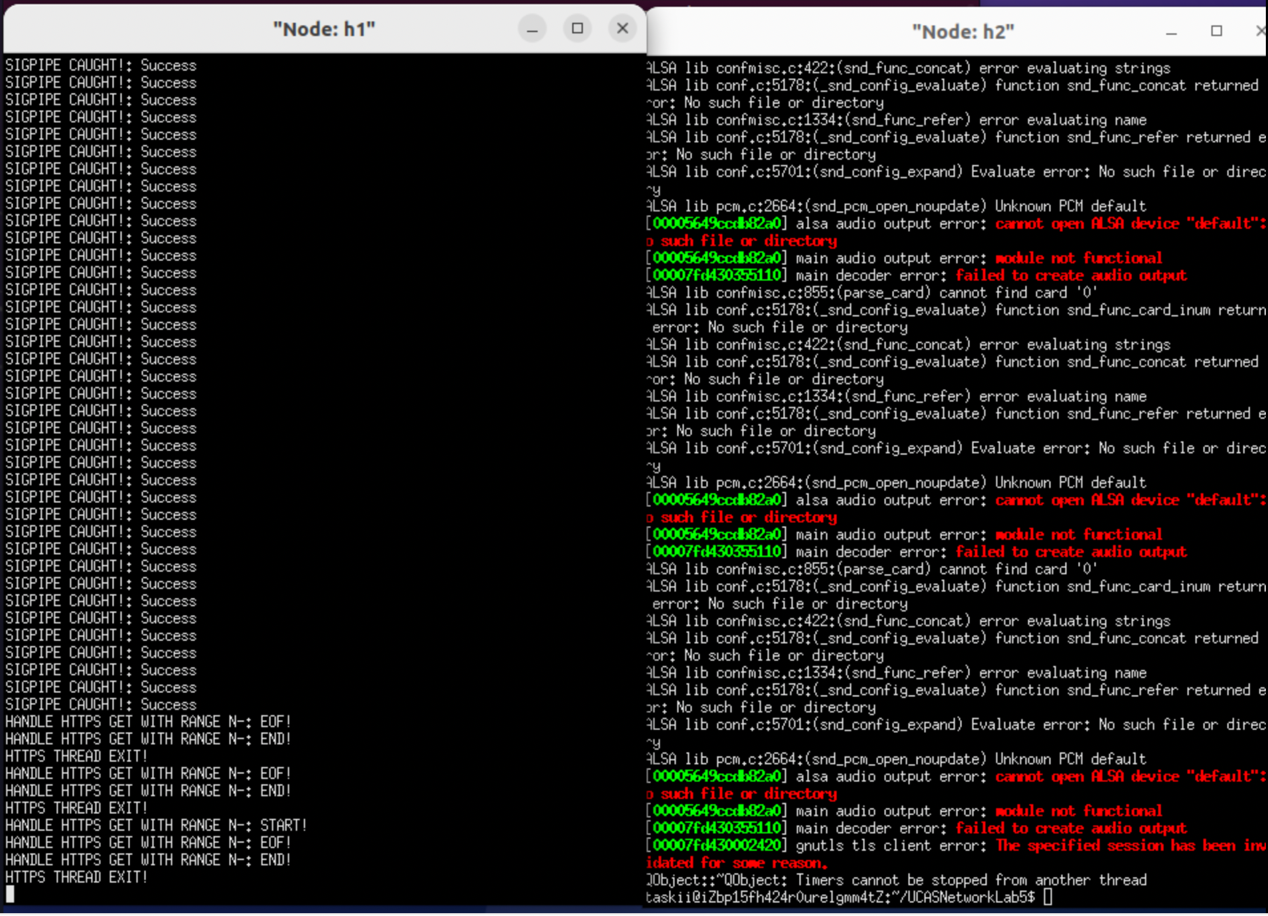


1. **抓包分析**
2. **Bug分析**
3. VLC在传输了第一个包之后出现fail的情况：

之前发现VLC能够接受一个分块，但之后就会出现error：HTTP connection failure：

使用wget也能获取一部分内容，但之后一直Retrying：

分析后发现VLC在接收完第一个分块之后，主动关闭了连接，但是http服务器检测不到，因此还在写内容，因此需要处理SIGPIPE信号，对该信号进行阻塞。并在h1的输出中显示。在读取视频的过程中，可以看到h1中有如下显示：



因此可以看出，对SIGPIPE信号进行阻塞可以解决该bug。