流转发通道协议

目录

[1：头文件 2](#_Toc519775305)

[2：tcp传输的数据头定义 2](#_Toc519775306)

[3 ：TCP设备与服务器交互 5](#_Toc519775307)

[3.1交互时序图 5](#_Toc519775308)

[3.2请求session： 5](#_Toc519775309)

[3.3登陆： 5](#_Toc519775310)

[3.4心跳 5](#_Toc519775311)

[3.5 发送流数据 6](#_Toc519775312)

[3.6停止发送流数据 6](#_Toc519775313)

[3.7代码参考 6](#_Toc519775314)

[4：TCP客户端与服务器交互 6](#_Toc519775315)

[4.1交互时序图 6](#_Toc519775316)

[4.2请求session： 7](#_Toc519775317)

[4.3登陆： 7](#_Toc519775318)

[4.4心跳 7](#_Toc519775319)

[4.5 请求流数据 7](#_Toc519775320)

[4.6代码参考 8](#_Toc519775321)

# 

# 1：头文件

XDefineMediaChannelStruct.h

经典服务器与物联网服务器在于端口号不同和设备id不同

经典服务器：

端口号：

设备id：纯数字

物联网服务器：

端口号：

设备id：数字与字母混合

# 2：tcp传输的数据头定义

#define MAGIC\_NUM 0XAA55

enum emMEDIA\_COMMAND\_DEFINE

{

//---------------------------------------------------------------------------------------------------------------

//用户登陆步骤 //1:NEED\_SESSIONID //2:LOGIN

XMEDIA\_COMMAND\_LOGIN = 0,

XMEDIA\_COMMAND\_LOGIN\_NEED\_SESSIONID,

XMEDIA\_COMMAND\_HEARTBEAT,

XMEDIA\_COMMAND\_LOGOUT,

//

XMEDIA\_COMMAND\_OPEN\_MAIN\_STREAM,

XMEDIA\_COMMAND\_CLOSE\_MAIN\_STREAM,

XMEDIA\_COMMAND\_OPEN\_SUB\_STREAM,

XMEDIA\_COMMAND\_CLOSE\_SUB\_STREAM,

XMEDIA\_COMMAND\_AV\_STREAM,

//以下是客户端与服务器交互

XMEDIA\_COMMAND\_OPEN\_STREAM = 20000,

XMEDIA\_COMMAND\_CLOSE\_STREAM,

XMEDIA\_COMMAND\_GET\_DEVICE\_LIST, //从中心服务器获取设备列表,请使用短链接的方式获取(包括了区域) (同时包括公共发布的视频) c->S

XMEDIA\_COMMAND\_GET\_STORE\_HISTORY\_LOG,//获取录像列表,请使用短链接的方式获取 client->media

XMEDIA\_COMMAND\_OPEN\_HIS\_STREAM,//下载 //服务器不返回该cmd，返回的是XMEDIA\_COMMAND\_AV\_STREAM，请注意判断result区分\_CMD\_OF\_SERVER\_RESULT\_DOWNLOAD\_XXXX

XMEDIA\_COMMAND\_CLOSE\_HIS\_STREAM,//关闭下载

};

enum emMEDIA\_RESULT\_DEFINE

{

XMEDIA\_RESULT\_OK = 0,

XMEDIA\_RESULT\_ERROR,

XMEDIA\_RESULT\_NOTREGISTER,//设备未注册，请注意，你可能偶尔收不到该回复

//

XMEDIA\_RESULT\_DOWNLOAD\_START, //下载开始标志

XMEDIA\_RESULT\_DOWNLOAD\_END,//下载结束标志

XMEDIA\_RESULT\_DOWNLOAD\_DATA,//数据标志

XMEDIA\_RESULT\_DOWNLOAD\_ERROR,//出错标志

};

enum emMEDIA\_FRAME\_TYPE\_DEFINE

{

//H264

XMEDIA\_FREAM\_TYPE\_H264\_IFRAME = 0,

XMEDIA\_FREAM\_TYPE\_H264\_PFRAME,

XMEDIA\_FREAM\_TYPE\_H264\_OTHERS,

//H265

XMEDIA\_FREAM\_TYPE\_H265\_IFRAME,

XMEDIA\_FREAM\_TYPE\_H265\_PFRAME,

XMEDIA\_FREAM\_TYPE\_H265\_OTHERS,

//MPEG4

XMEDIA\_FREAM\_TYPE\_MPEG4,

//MJPEG

XMEDIA\_FREAM\_TYPE\_MJPEG,

//Audio

XMEDIA\_FREAM\_TYPE\_PCM,

XMEDIA\_FREAM\_TYPE\_ADPCM,

XMEDIA\_FREAM\_TYPE\_G711A,

XMEDIA\_FREAM\_TYPE\_G711U,

XMEDIA\_FREAM\_TYPE\_AAC,

//JPG

XMEDIA\_FREAM\_TYPE\_JPEG

};

struct ST\_XMEDIA\_TIME {

uint16\_t wYear;

uint16\_t wMonth;

uint16\_t wDayOfWeek;

uint16\_t wDay;

uint16\_t wHour;

uint16\_t wMinute;

uint16\_t wSecond;

uint16\_t wMilliseconds;

};

struct ST\_XMEDIA\_HEAD

{

uint32\_t cmd;//emMEDIA\_COMMAND\_DEFINE

uint32\_t result;//emMEDIA\_RESULT\_DEFINE

uint32\_t session;

uint32\_t datalen;

uint64\_t tick;//相对时间-设备需要使用硬件时间保证该数值一直是递增的

uint32\_t reserve[2];

};

struct ST\_XMEDIA\_AVHEAD

{

//emMEDIA\_FRAME\_TYPE\_DEFINE

uint16\_t nAVFrameType;

//video

uint16\_t nVideoFrameRate;

uint16\_t nVideoWidth;

uint16\_t nVideoHeight;

//audio

//nAudioBitRate，nAudioBitsPerSample在不同的第三方库中可能只用一种，但是是可以计算出另一个的

uint16\_t nAudioChannels;//通道数: 单声道为 1, 立体声为 2

uint16\_t nAudioSamplesRate; // nAudioSamplesPerSec //每秒的采样样本数，采样率//8000,44100等

uint16\_t nAudioBitRate; //数据传输的平均速率(每秒的字节数) 自定义的值

uint16\_t nAudioBitsPerSample;//采样大小(字节)比如16

//

ST\_XMEDIA\_TIME time;//设备不使用不填充该字段

};

struct stFrameX\_T

{

ST\_XMEDIA\_HEAD header;

ST\_XMEDIA\_AVHEAD avheader;

unsigned int data[0];//灵活数组结构

};

//ST\_XMEDIA\_LOGIN\_INFO

//设备：只填充pwd

struct ST\_XMEDIA\_LOGIN\_INFO

{

int reserve[2];

char user[32];

char despwd[128];

};

# 3 ：TCP设备与服务器交互

## 3.1交互时序图

端口:6504

## 3.2请求session：

发送：ST\_XMEDIA\_HEAD

cmd： XMEDIA\_COMMAND\_LOGIN\_NEED\_SESSIONID

服务器返回：

返回：ST\_XMEDIA\_HEAD

cmd：XMEDIA\_COMMAND\_LOGIN\_NEED\_SESSIONID

session：服务器填充了session

## 3.3登陆：

发送：ST\_XMEDIA\_HEAD + ST\_XMEDIA\_LOGIN\_INFO

ST\_XMEDIA\_HEAD .cmd：SINGNALING\_CHANNEL\_CMD\_LOGIN

ST\_XMEDIA\_HEAD .session: 服务器返回的session

ST\_XMEDIA\_HEAD . datalen: sizeof(ST\_XMEDIA\_LOGIN\_INFO)

ST\_XMEDIA\_LOGIN\_INFO. despwd:使用des混合加密设备ID与服务器返回的session

服务器返回：

返回：signaling\_channel\_head\_t

cmd：SINGNALING\_CHANNEL\_CMD\_LOGIN

result：XMEDIA\_RESULT\_OK

## 3.4心跳

发送：ST\_XMEDIA\_HEAD

cmd： XMEDIA\_COMMAND\_HEARTBEAT

服务器返回：

返回：ST\_XMEDIA\_HEAD

cmd：XMEDIA\_COMMAND\_HEARTBEAT

## 3.5 发送流数据

接收：ST\_XMEDIA\_HEAD

cmd：

XMEDIA\_COMMAND\_OPEN\_MAIN\_STREAM

或者

XMEDIA\_COMMAND\_OPEN\_SUB\_STREAM

发送：ST\_XMEDIA\_HEAD+ ST\_XMEDIA\_AVHEAD+数据

暂时不关心服务器的返回。

## 3.6停止发送流数据

接收：ST\_XMEDIA\_HEAD

cmd：

XMEDIA\_COMMAND\_CLOSE\_MAIN\_STREAM

或者

XMEDIA\_COMMAND\_CLOSE\_SUB\_STREAM

## 3.7代码参考

\_IncludeDevice 下的 XDeviceStream.cpp

# 4：TCP客户端与服务器交互

## 4.1交互时序图

## 4.2请求session：

发送：ST\_XMEDIA\_HEAD

cmd： XMEDIA\_COMMAND\_LOGIN\_NEED\_SESSIONID

服务器返回：

返回：ST\_XMEDIA\_HEAD

cmd：XMEDIA\_COMMAND\_LOGIN\_NEED\_SESSIONID

session：服务器填充了session

## 4.3登陆：

发送：ST\_XMEDIA\_HEAD + ST\_XMEDIA\_LOGIN\_INFO

ST\_XMEDIA\_HEAD .cmd：SINGNALING\_CHANNEL\_CMD\_LOGIN

ST\_XMEDIA\_HEAD .session: 服务器返回的session

ST\_XMEDIA\_HEAD . datalen: sizeof(ST\_XMEDIA\_LOGIN\_INFO)

ST\_XMEDIA\_LOGIN\_INFO. User:用户名

ST\_XMEDIA\_LOGIN\_INFO. despwd:使用des混合加密密码与服务器返回的session

服务器返回：

返回：signaling\_channel\_head\_t

cmd：SINGNALING\_CHANNEL\_CMD\_LOGIN

result：XMEDIA\_RESULT\_OK

## 4.4心跳

发送：ST\_XMEDIA\_HEAD

cmd： XMEDIA\_COMMAND\_HEARTBEAT

服务器返回：

返回：ST\_XMEDIA\_HEAD

cmd：XMEDIA\_COMMAND\_HEARTBEAT

## 4.5 请求流数据

发送：ST\_XMEDIA\_HEAD + ST\_SREVER\_OPEN\_STREAM

cmd：

XMEDIA\_COMMAND\_OPEN\_STREAM

接收：ST\_XMEDIA\_HEAD+ ST\_XMEDIA\_AVHEAD+数据

## 4.6代码参考

\_XInternetProjectClient\\_Include下的 XClientStream.cpp