
和目视频云平台接入文档

(V1.0)

中国移动通信集团物联网公司

2017 年 12 月

目录

编制说明	1
1. 概述	2
1.1 协议介绍	2
1.2 术语、定义和缩略语	2
1.2.1 常用术语	2
1.2.2 定义	3
1.3 平台介绍	4
2 鉴权规则定义	5
2.1 WSSE 认证介绍	5
2.2 防重放攻击	6
3 接入流程	6
3.1 接入说明	6
3.2 二维码扫描	7
4 能力集定义	7
5 TD 与平台协议	7
5.1 OPENAPI 接入	8
5.1.1 TD 注册	8
5.1.2 TD 上报能力集	11
5.1.3 获取 HEDGW 服务	12
5.1.4 获取在线配置	13
5.1.5 更新在线配置	17
5.1.6 更新 TD 版本信息	18
5.2 HEDGW 信令通道接入方式	19
5.2.1 登录 HEDGW 平台	20
5.2.2 TD 心跳保活	21
5.2.3 TD 上传音视频数据	22
5.2.4 TD 上传 SD 卡缓存音视频数据	23
5.2.5 TD 接收客户端音频数据（语音对讲）	23
5.2.6 TD 事件上传	24
5.2.7 TD 上传设备缩略图	25
5.2.8 TD 接收 HEDGW 通知	26
5.2.9 客户端控制 TD 命令	27
5.2.10 客户端查询 TD 命令	48
5.2.11 TD 消息上报	50
5.2.12 维护性质接口	57
5.3 HEDGW 码流接入方式	58
5.3.1 码流接入方式	58
5.3.2 TD 编码要求	58

6	TD 端流程说明	59
6.1	TD 注册登录.....	59
6.2	TD 上传码流.....	60
6.3	在线配置	60
附录 A: TD 软件版本号要求		61
附录 B: 协议版本演进说明		61
附录 C: OPENAPI HTTPS 错误码		61
附录 D: PROTOBUF 文件		62
附录 E: 接入代码示例.....		64

编制说明

本规范规定了中移物联网监控云平台终端设备接入的技术要求，对终端设备接入平台协议、流程等方面做了硬性要求；规范在满足“开放性”和“安全性”的基础上，满足监控云平台业务的需要。

编号	版本	时间	说明
1	V0.0.1	2016.12.22	初稿建立，基于 V1.0 文档进行增删
2	V0.0.2	2016.12.23	完善了部分新增的内容等
3	V0.0.3	2016.12.27	新增了 Token 校验，序列号管理的内容等
4	V0.0.4	2016.12.28	新增和已有接口的修订
5	V0.0.5	2017.1.10	修订文档格式，完善 protobuf 文件
6	V0.0.6	2017.1.11	更新客户端直播请求命令
7	V0.0.7	2017.1.13	完善序列号管理章节内容
8	V0.0.8	2017.1.16	完善 WSSE 认证，包括 TD 注册的部分等
9	V0.0.9	2017.1.16	更新设置 wifi 及获取 wifi 列表的消息格式
10	V0.1.0	2017.1.22	完善了 WSSE 的说明以及部分接口新增 WSSE 内容
11	V0.1.1	2017.1.22	新增附录 E: TimeZone 列表等
12	V0.1.2	2017.1.23	完善维护性质接口参数，完善 OpenAPI 接口等
13	V0.1.3	2017.2.6	更新 TD 信息接口新增一个 deviceModelId 字段
14	V0.1.4	2017.7.31	增加客流统计相关接口
15	V0.1.5	2017.10.20	修改能力集上报方式，删除无用接口，增加接口说明，格式修订
16	V0.1.6	2018.07.02	添加事件录制套餐的说明
17	V0.1.7	2018.12.03	支持 H265 编码视频上传
18	V0.1.8	2018.12.25	添加 PTZ 控制命令：5.2.9.11，5.2.9.12，5.2.10.13，5.2.11.7
19	V0.1.9	2019.01.25	添加双目摄像头的支持：5.1.2 5.1.5
20	V0.1.10	2019.03.11	升级消息中 checksum 改为 checkSum
21	V0.1.11	2019.04.09	1. 增加有线绑定接口（5.1.2） 2. 增加运动跟踪功能开关和能力上报（5.2.9.21） 3. 增加全景拼接功能（5.2.9.23 5.2.11.7） 4. SD 卡补录数据增加补录事件的描述（5.2.4）

注：重大结构调整，如目录结构改变，主版本号增加；一般性目录调整，如变更目录条目，子版本号增加；不改变目录，内容变更时，修订版本号增加；高位版本号增加时，低位复位为 0。

1. 概述

本章分别从协议介绍、常用术语、平台介绍三个方面描述监控云平台接入终端设备的协议；在详细介绍之前，通过这三个方面增加对监控云平台的了解，方便终端设备开发人员接入开发。

1.1 协议介绍

本协议为终端设备接入监控云平台提供了规范。协议由两大类组成：**OpenAPI** 接口和媒体信令协议。

OpenAPI 接口采用基于 **HTTPS** 协议承载，内容采用 **JSON** 格式封装，主要包括如下功能：

媒体、信令协议采用 **TCP+SSL** 长连接传输，内容采用 **protobuf** 格式封装（使用 **protobuf 2.x** 版本），主要包括如下功能：

- 心跳保活
- 上传音视频数据
- 上传 **SD** 卡音视频数据
- 事件报警
- 在线配置
- 云台控制
- 远程升级

1.2 术语、定义和缩略语

1.2.1 常用术语

术语/缩写	含义
HPP	和目私有协议(Hemu Private Protocol)
OpenAPI	视频云平台开放性接入接口，提供基于 webservice 的接口，供第三方设备、平台与视频云平台进行交互。
HEDGW	视频云平台第三方设备接入网关，与第三方设备建立信令交互的长连接。接收第三方设备连接请求和平台控制指令下发，接收第三方设备发送视频数据，在系统内转发（Hemu Device GateWay）
CDS	内容解密、分发服务（Content Distribution Service）
UPNS	统一消息推送系统，将告警等信息推送给各个平台的客户端设备
NTP	Network Time Protocol （NTP）是用来使计算机时间同步化的一种协议
TD	第三方终端设备（Third Device），如含有视频采集能力的前端相机；终端设备后续采用 TD 简称

表 1.2.1 常用术语解释

1.2.2 定义

1.2.2.1 Json 规则

协议除媒体数据外，统一采用 Json 序列化和反序列化业务数据；遵照 Json 的格式定义，在本文档中的所有 Json 格式示例中，用一定的格式表示了 Json 值的类型，统一的规则如下：

示例	类型	取值
{ }	object 类型	以实际内容为准
[]	array 类型	以实际内容为准
""	string 类型	UTF-8 作为字符串的编码格式，本文档中的所有字符串也是如此，因此客户端应注意转换。 各种字符串值比如通道名称，网络地址等的长度默认最大为 1024 字节，除非特殊说明，字符串长度都包括\0 结束符
true,false	bool 类型	true 或者 false
整数	默认是 int 类型； 如果取值范围有明确标明 0 以上的，表示是 unsigned int 类型	默认取值范围是全范围，除非特殊说明。
小数	double 类型	默认取值范围是全范围，除非特殊说明

表 1.2.2.1 Json 格式统一规则

1.2.2.2 时间格式定义

若无特殊说明，协议中凡是携带时间的字段，需严格遵循 RFC3339 Date and Time on the Internet: Timestamps 标准规范。

举例：

本地时区为 Asia/Shanghai，当前时间为 2017-01-01 15:00:00，那么 RFC3339 格式输出为：2017-01-01T15:00:00+08:00

1.2.2.3 TD 厂家申请信息说明

TD 厂家的 TD 要接入到监控云平台，先向物联网提供申请厂家接入的 Key 和 Password。其中 Key 不含中划线。Password 要注意保密，如果侦测到被攻击，可能会屏蔽掉该厂家所有出货的设备。例如：

TD 厂商名称	厂商 Key(accesskey)	厂商 Password (access secret)
浙监	Zhejiang	XgijVJq7elp94Ax3cKvq

表 1.2.2.3TD 厂商信息

1.2.2.4 TD modelId 说明

TD 的 modelId 定义了某个厂商具体的某一类设备。主要用于 TD 的升级使用。TD 厂家需要根据研发和出货要求，提前向物联网申请。具体的规则是：厂商 accesskey+下划线+不超过 16 位随机值（字母，数字和下划线）。例如：Zhejiang_zj_cnmobile_001

1.3 平台介绍

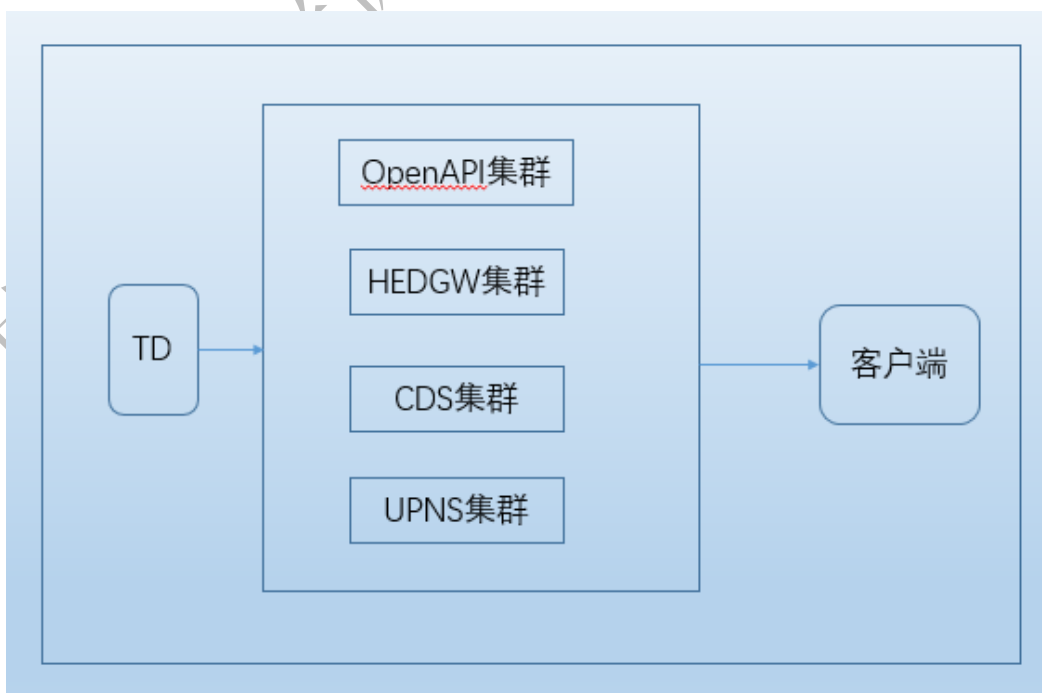


图 1.3 平台组成

2 鉴权规则定义

协议规定监控云平台需要对 TD 进行鉴权操作；否则，不允许 TD 注册、登录等操作。

遵循“开放、闭合”的设计原则和“安全性”的原则，监控云平台提供密钥推导生成库，供 TD 厂商接入监控云平台使用；本协议在此基础上，提供统一的规则描述定义，以此形成统一的行为约束。

2.1 WSSE 认证介绍

若无明确说明，协议规定 TD 的请求中必须携带鉴权信息，鉴权信息承载在 HTTP 报文中。鉴权规则遵从 WS-Security 规范中常用的用户名、密码方式中的格式要求实现，这里简称 WSSE 鉴权方式。

使用 WSSE 认证方式可以实现无状态的鉴权过程，简化交互流程，提升安全性。WSSE 认证流程如下：

1. TD 知悉用户名 username 和密码 password；
2. TD 创建随机字符串 nonce；协议规范 nonce 长度为 16 字节；
3. TD 创建产生时间戳信息 createTimeStamp，时间戳格式遵从 RFC3339 标准规范；
4. TD 创建密码摘要 passwordDigest，创建规则遵循如下公式：

$$\text{passwordDigest} = \text{Base64}(\text{SHA1}(\text{nonce} + \text{createTimestamp} + \text{password}))$$

其中：

nonce 随机字符串，长度 16 字节

createTimestamp 创建时间戳，遵从 RFC3339 标准

password 认证密码，见 TD 厂家申请信息说明

+ 字符串连接，无任何意义

鉴权信息承载在 HTTP 协议头部字段中，协议规定需遵循如下标准认证方式：

HTTP 方法 URI HTTP/1.1

Host:www.example.com

Content-Length:0

Authorization: WSSE profile="UsernamePwd"

X-WSSE:UsernamePwd Username="Zhejiang-110234567894562",

PasswordDigest="dGhpcyBpcyBhIGV4YW1wbGUyYmNiOTI4YWM4ZjlzOTMxYTdhOTVhNjM
yNGMwMmYzMWQxZWQwMjU5",Nonce="6FQHGU7M01ZMIA5G",Created="2017-01-01T
15:00:00+08:00"

其中：

Authorization 认证描述头部，使用 WSSE 用户名、密码方式

X-WSSE 携带认证信息头部：Username，生成方式是 accesskey+中划线+deviceId。PasswordDigest 密码摘要，生成方式见第四步公式。Nonce-随机字符串，见第二步要求。Created-创建时间戳，见第三步要求

若 TD 发起的请求中, 没有携带鉴权信息, 云平台返回 401 响应, 响应头部中需携带 Date 字段, Date 字段时间格式遵从 RFC3339 标准规范。返回响应格式如下:

```
HTTP/1.1 401 Unauthorized
Host:www.example.com
Date: 2017-01-01T15:00:00+08:00
Content-Length:0
WWW-Authenticate: WSSE realm="authcenter ", profile="UsernamePwd"
其中:
WWW-Authenticate 描述采用 WSSE 规范中的用户名、密码方式鉴权
Date              云平台当前时间, 遵从 RFC3339 标准
```

2.2 防重放攻击

为防止 TD 请求报文在互联网环境下, 被网络爬虫等复制, 从而对平台进行重放攻击。协议要求云平台能够过滤与平台时间差值超过 5 分钟请求的报文, 云平台在收到此类报文时, 应返回 401 鉴权失败响应, 响应头部携带 Date 字段, 时间格式遵从 RFC3339 标准。

另外, TD 存在时间与云平台不同步的异常情况, 协议要求 TD 在发起请求过程中, 若云平台返回 401 鉴权失败响应, TD 应根据 Date 字段时间更新 TD 设备本地时间。401 响应格式如下:

```
HTTP/1.1 401 Unauthorized
HOST:www.example.com
Date: 2017-01-01T15:00:01+08:00
Content-Length:0
WWW-Authenticate: WSSE realm="authcenter ", profile="UsernamePwd"
其中:
WWW-Authenticate 描述采用 WSSE 规范中的用户名、密码方式鉴权
Date              云平台当前时间, 遵从 RFC3339 标准
```

3 接入流程

3.1 接入说明

递交公司相关申请材料到接入平台, 申请获得 accessKey 和 accessSecret。每个设备接入厂家都有唯一的 accessKey 和 accessSecret, 没有合法的访问密钥就无法接入平台。密钥关联了接入厂家、允许接入设备及数量、设备服务订阅模式等信息。

TD 主要通过 OpenAPI 和 HEDGW 与视频平台进行交互。平台定义了详细的协议, 使 TD 能够接入平台并实现注册、实时视频点播、报警事件分发、设备信息查询、设备控制、录像

回放、配置查询功能等功能。

3.2 二维码扫描

TD 注册时，客户端 App 会生成二维码，用于初始化 TD 及进行 TD 注册。二维码数据解析方式见文档：《和目视频云平台补充说明》

4 能力集定义

不同的设备能力不同，如支持云台、支持无线连接等。协议对能力集作了区分，分为如下两种：

1. TD 设备注册时能够确定的能力集（固化能力集，需要重新注册设备才能修改能力集）：

编号	能力集	描述
1	PTZ	支持云台控制
2	WIFI	支持 WIFI 无线连接
3	无麦克风	不支持麦克风
4	鱼眼	鱼眼设备

2. TD 设备注册后可以更新的能力集（可变能力集）：

编号	能力集	描述
1	motion 区域设置	支持动态区域检测报警
2	双向语音通话	支持全双工双向语音通话
3	客流统计	支持客流统计功能
4	SD 卡	支持 SD 卡
5	人脸检测	支持人脸检测
6	事件录制	支持事件录制
7	双目摄像头	支持双目摄像头
8	运动跟踪	支持运动跟踪功能
9	人形检测	支持人形检测功能
10	全景拼接	支持云台转动进行全景拼接功能

表 4.1 能力集定义表

5 TD 与平台协议

TD 与平台之间信令协议主要通过 HTTPS 方式接入 OpenAPI 服务器。不支持以 HTTP 方式接入。针对 HTTPS 接入，只支持 TLS 1.2，不支持 1.2 以前的版本。访问每一个 OpenAPI 的接口请求必须经过 accessKey 和 accessSecret 生成签名后才能调用。

接入使用的域名见文档：《和目视频云平台补充说明》

本文中假设厂商接入的 accessKey 为 5e2abe86-6aa，accessSecret 为 ADtuchbM8TBPhQdSi8U4，

请求格式为:

[https://domain_name/andmu/v2/service/batch?accessKey=5e2abe86-6aa](https://domain_name/andmu/v2/service/batch?accessKey=5e2abe86-6aa&jsonObject={a:1,b:2,c:3}&signature=1661d0fa85d325c27030af5783f63998)
[&jsonObject={a:1,b:2,c:3}&signature=1661d0fa85d325c27030af5783f63998](https://domain_name/andmu/v2/service/batch?accessKey=5e2abe86-6aa&jsonObject={a:1,b:2,c:3}&signature=1661d0fa85d325c27030af5783f63998)

其中 jsonObject 即为接口的请求体，用于填充该请求需要的参数。

signature 生成规则如下：将参数和厂商 accessKey，如上：[accessKey,jsonObject]，对 accessSecret + accessKey+等号+参数值+jsonObject+等号+参数值进行 MD5 32bit 加密。

如上示例为：ADtuchbM8TBPhQdSi8U4accessKey=5e2abe86-6aajsonObject={a:1,b:2,c:3}。然后获取到签名 signature:1661d0fa85d325c27030af5783f63998

5.1 OpenAPI 接入

关于 OpenAPI 接口的通用参数说明

参数	说明
msgType	包括请求消息类型和响应消息类型，每个接口通过消息类型进行区分。 比如，TD 注册请求：MSG_P_THD_ADD_CAMERA_REQ， 响应：MSG_P_THD_ADD_CAMERA_RSP
errorCode	错误码，见附录 C：OpenAPI HTTPS 错误码
description	错误码对应的具体描述，见附录 C：OpenAPI HTTPS 错误码
deviceId	TD 的唯一标识，也就是每个设备对应的序列号。15 位数字串
deviceMacAddress	TD 的 Mac Address。要符合 Mac 的规则，且中间没有任何的连接符（冒号，点号，中划线）
deviceModelId	见 TD modelId 说明 TD modelId 说明，不区分大小写

表 5.1 通用参数说明

消息类型定义表如下：

消息类型	说明
MSG_P_THD_ADD_CAMERA_REQ	TD 注册
MSG_P_THD_SET_CAMERA_CAPACITY_REQ	TD 上报能力集
MSG_P_THD_GET_CAMERA_CHN_ADDR_REQ	获取 HEDGW 服务
MSG_P_THD_GET_CAMERA_SETTINGS_REQ	获取在线配置
MSG_P_THD_SET_CAMERA_SETTINGS_REQ	更新在线配置
MSG_P_THD_SET_CAMERA_INFO_REQ	更新 TD 版本信息

表 5.2 消息类型表

5.1.1 TD 注册

简介	TD 注册
----	-------

消息类型	HTTP1.1+JSON	
传输方向	TD->OpenAPI	
鉴权	是	
连接方式	短连接	
请求方法	POST	
请求格式	<p>HTTP Header:</p> <p>Authorization:WSSE profile="UsernamePwd"</p> <p>X-WSSE:UsernamePwd Username="Zhejiang-110234567894562",PasswordDigest="dGhpcyBpcyBhIGV4YW1wbGUyYmNiOTI4YWM4ZjZlOTMxYTdhOTVhNjMyNGMwMmYzMWQxZWQwMjU5",Nonce="6FQHGU7M01ZMIA5G",Created="2017-01-01T15:00:00+08:00"</p> <p>HTTP Body:</p> <p>accessKey:5e2abe86-6aa</p> <p>signature: yyyy</p> <p>jsonObject:</p> <pre>{ "msgType": "MSG_P_THD_ADD_CAMERA_REQ", "userMobile": "18868934737", "userToken": "xxxx", "deviceMacAddress": "2059a074e0bc", "deviceId": "110234567894562", "deviceModelId": "Zhejiang_zj_cnmobile_001", "comment": "'HW':3RD-X01','FEATURE':0", "registMode": "1",}</pre>	
请求示例	<p>1, POST 地址: https://openapi.h.reservehemu.com/andmu/v2/service/batch</p> <p>2, POST Header: Authorization:WSSE profile="UsernamePwd"</p> <p>X-WSSE:UsernamePwd Username="Zhejiang-110234567894562",PasswordDigest="dGhpcyBpcyBhIGV4YW1wbGUyYmNiOTI4YWM4ZjZlOTMxYTdhOTVhNjMyNGMwMmYzMWQxZWQwMjU5",Nonce="6FQHGU7M01ZMIA5G",Created="2017-01-01T15:00:00+08:00"</p> <p>3, POST Body:</p> <p>accessKey=5e2abe86-6aa&signature=0ba53e90b829efe05c6c0c420375f21a&jsonObject={"deviceMacAddress":"000000300002","userToken":"e10adc3949ba59abbe56e057f20f883e","msgType":"MSG_P_THD_ADD_CAMERA_REQ","userMobile":"13888888888","deviceId":"110234567894562","comment":["HW':3RD-X01','FEATURE':0"],"deviceModelId":"Zhejiang_zj_cnmobile_001"}</p>	
响应格式	<pre>{ "errorCode": "0", "description": "", "msgType": "MSG_P_THD_ADD_CAMERA_RSP", "deviceToken": "89Q832IF09JISDOFCJIOAWIOJ", "unifiedId": "KKOWAJFIWI32904493" }</pre>	
请求备注	userMobile	用户手机号码

	userToken	二维码里面包含的注册 Token
	deviceMacAddress	TD MAC 地址，不区分大小写，不带冒号
	registMode	如果二维码为加密方式，此值为"1"，否则可以不填或者为"0"
	comment	<p>厂商信息，TD 型号等。</p> <p>TD 需传入下面的值：（中间不要有空格）</p> <p><code>['HW':'3RD-X01','FEATURE':0]</code></p> <p>FEATURE 字段代表 TD 设备固化能力集，注册后不可修改，除非删除设备重新注册。值为十进制，按照二进制不同比特位代表不同含义，定义如下：</p> <p>0x00000001 支持 WIFI 无线连接</p> <p>0x00000010 支持 PTZ 控制</p> <p>0x00000200 无麦克风</p> <p>0x00000800 鱼眼设备</p> <p>例如：TD 同时支持 PTZ 控制和 WIFI 连接，'FEATURE'值则为 17</p>
响应备注	返回参数用于后续接口的鉴权	
备注	协议要求 TD 上电后，需向平台发起注册流程。TD 需要先通过扫描二维码获取到连接 Wi-Fi 信息及连接平台相关鉴权信息，连接 Wi-Fi 成功之后使用鉴权信息连接平台 OpenAPI。注册成功后，需要进行语音提示	

5.1.2 有线绑定

简介	对于有线设备，插上网线即可联网，不需要进行配网操作，除了二维码扫码方式绑定设备之外，还可以使用有线绑定方式。因此设备进入注册模式之后，除了启动二维码扫码模块之外，还要启动有线绑定模块，任何一个模块获取到注册信息之后，都需要发起注册请求，并将另一个模块停止。
消息类型	HTTP1.1+JSON
传输方向	TD->OpenAPI
鉴权	是
连接方式	短连接
请求方法	POST
请求格式	<p>HTTP Header:</p> <p>Authorization:WSSE profile="UsernamePwd"</p> <p>X-WSSE:UsernamePwd Username="Zhejiang-110234567894562",PasswordDigest="dGhpcyBpcyBhIGV4YW1wbGUyYmNiOTI4YWM4ZjIzOTMxYTdhOTVhNjMyNGMwMmYzMWQxZWQwMjU5",</p> <p>Nonce="6FQHG7M01ZMIA5G",Created="2017-01-01T15:00:00+08:00"</p> <p>HTTP Body:</p> <p>accessKey:5e2abe86-6aa</p> <p>signature: yyyy</p> <p>jsonObject:</p> <p>{</p>

	<pre>"msgType": "MSG_P_THD_GET_CAMERA_BIND_INFO_REQ", "deviceMacAddress": "2059a074e0bc", "deviceId": "110234567894562", }</pre>		
请求示例	参考 TD 注册请求示例		
响应格式	<pre>{ "errorCode": "0", "userMobile": "", "userToken": "", "msgType": "MSG_P_THD_GET_CAMERA_BIND_INFO_RSP" }</pre>		
响应备注	参数名称	参数说明	取值
	userMobile	用户名，注册时使用	
	userToken	用户 token，注册时使用	
备注	设备端进入注册模式之后，循环调用此接口（间隔不小于 10 秒），获取到用户信息之后，调用 TD 注册接口绑定设备		

5.1.3 TD 上报能力集

简介	当 TD 设备注册成功之后，需要主动上报可变能力集。 当能力集变化时，也需要主动上报（例如：通过在线升级更新 TD 固件版本支持了新的功能）
消息类型	HTTP1.1+JSON
传输方向	TD->OpenAPI
鉴权	是
连接方式	短连接
请求方法	POST
请求格式	<p>HTTP Header:</p> <pre>Authorization:WSSE profile="UsernamePwd" X-WSSE:UsernamePwd Username="Zhejiang-110234567894562",PasswordDigest=" dGhpcyBpcyBhIGV4YW1wbGUyYmNiOTI4YWM4ZjZlOTMxYTdh OTVhNjMyNGMwMmYzMWQxZWQwMjU5", Nonce="6FQHGU7M01ZMIA5G",Created="2017-01-01T15:00:00+08:00"</pre> <p>HTTP Body:</p> <pre>accessKey:5e2abe86-6aa signature: yyyy jsonObject: { "msgType": "MSG_P_THD_SET_CAMERA_CAPACITY_REQ", "deviceMacAddress": "2059a074e0bc",</pre>

	<pre> "deviceId": "110234567894562", "deviceToken": "89Q832IF09JISDOFCJIOAWIOJ", "supportKey" { "motionRegion": 0, "duplexAudioTalk":1, "faceDetect":0, "personStatistic":1, "sdcard":1, "eventRecord":1, "binoCam":1, "motionTrack":1, "peopleDetection":0, "ptzPano":1 } </pre>		
请求示例	参考 TD 注册请求示例		
响应格式	<pre> { "errorCode": "0", "description": "", "msgType": " MSG_P_THD_SET_CAMERA_CAPACITY_RSP" } </pre>		
请求备注	参数名称	参数说明	取值
	supportKey	支持的能力集属性，包含如下功能列表： <ul style="list-style-type: none"> ➤ motion 区域设置（motionRegion） ➤ 双向语音通话（duplexAudioTalk） ➤ 人脸检测（faceDetect） ➤ 客流统计（personStatistic） ➤ SD 卡（sdcard） ➤ 事件录制（eventRecord） ➤ 双目摄像头（binoCam） ➤ 运动跟踪(motionTrack) ➤ 人形检测(peopleDetection) ➤ 全景拼接(ptzPano) 	值为 1 代表支持，值为 0 代表不支持
备注	如果摄像头上报了双目摄像头的能力，则需要通过更新在线配置接口将双目摄像头参数上报到服务器		

5.1.4 获取 HEDGW 服务

简介	获取 HEDGW 服务地址
消息类型	HTTP1.1+JSON
传输方向	TD->OpenAPI
鉴权	是

连接方式	短连接
请求方法	POST
请求格式	<p>HTTP Header:</p> <p>Authorization:WSSE profile="UsernamePwd"</p> <p>X-WSSE:UsernamePwd Username="Zhejiang-110234567894562",PasswordDigest="dGhpcyBpcyBhIGV4YW1wbGUyYmNiOTI4YWM4ZjIzOTMxYTdhOTVhNjMyNGMwMmYzMWQxZWQwMjU5",Nonce="6FQHGU7M01ZMIA5G",Created="2017-01-01T15:00:00+08:00"</p> <p>HTTP Body:</p> <p>accessKey:5e2abe86-6aa</p> <p>signature: yyyy</p> <p>jsonObject:</p> <pre>{ "msgType": "MSG_P_THD_GET_CAMERA_CHN_ADDR_REQ", "deviceMacAddress ": "2059a074e0bc", "deviceId": "110234567894562", "deviceToken": "89Q832IF09JISDOFCJIOAWIOJ" }</pre>
请求示例	参考 TD 注册请求示例
响应格式	<pre>{ "errorCode": "0", "description": "", "msgType": "MSG_P_THD_GET_CAMERA_CHN_ADDR_RSP", "serverAddr": "220.1.1.10", "serverPort": "50922" }</pre>
请求备注	无
响应备注	<p>若响应成功，serverAddr 表示 HEDGW 服务器 IP 地址，ServerPort 表示 HEDGW 服务器端口</p> <p>若响应失败，返回 404 错误</p> <p>注意：serverAddr 有可能为域名方式，TD 需支持域名解析</p>
备注	<p>TD 登录 HEDGW 时，通过 OpenAPI 获取 HEDGW 服务地址</p> <p>TD 获取 HEDGW 服务地址之后，需要缓存地址信息。每次连接 HEDGW 时，应先使用之前缓存的 HEDGW IP 地址和端口进行连接，建议最多缓存 3 组 IP/端口，每组 IP/端口的有效期 12 小时，如果缓存的地址都无法连接成功，再使用该接口重新获取 HEDGW 地址，避免频繁调用 OPENAPI 接口，对服务器造成压力</p>

5.1.5 获取在线配置

简介	获取 TD 在线配置
消息类型	HTTP1.1+JSON
传输方向	TD->OpenAPI
鉴权	是

连接方式	短连接
请求方法	POST
请求格式	<p>HTTP Header:</p> <p>Authorization:WSSE profile="UsernamePwd"</p> <p>X-WSSE:UsernamePwd Username="Zhejiang-110234567894562",PasswordDigest="dGhpcyBpcyBhIGV4YW1wbGUyYmNiOTI4YWM4ZjIzOTMxYTdhOTVhNjMyNGMwMmYzMWQxZWQwMjU5",</p> <p>Nonce="6FQHG7M01ZMIA5G",Created="2017-01-01T15:00:00+08:00"</p> <p>HTTP Body:</p> <p>accessKey:5e2abe86-6aa</p> <p>signature: yyyy</p> <p>jsonObject:</p> <pre>{ "msgType": "MSG_P_THD_GET_CAMERA_SETTINGS_REQ", "deviceMacAddress": "2059a074e0bc", "deviceId": "110234567894562", "deviceToken": "89Q832IF09JISDOFCJIOAWIOJ" }</pre>
请求示例	参考 TD 注册请求示例
响应格式	<pre>{ "errorCode": "0", "description": "", "msgType": "MSG_P_THD_GET_CAMERA_SETTINGS_RSP", "camRW": { "HDVideo": "On", "antiFlicker": "50", "cameraImageRotate": "Off", "cameraMicrophone": "Off", "motionSensitivity": "80", "nightVision": "Auto", "soundSensitivity": "90", "status": "On", "statusLight": "On", "title": "my office", "motionRegions": "1;2;3;25", "motionNotification": "On", "soundNotification": "On", "motionTrack": "On", "motionTrackBackTime": "30", "cameraOffSchedule": { "valueList": [</pre>

	<pre> { "repeat": 127, "startTime ": "21:00", "endTime": "10:00", "status": true, "index": 0 }], "valueListNum": 1 }, "alarmAlertOffSchedule": { "valueList": [{ "repeat": 127, "startTime ": "21:00", "endTime": "10:00", "status": true, "index": 0 }], "valueListNum": 1 }, "personStatus": "On", "personRegion": { "valueListNum": 2, "valueList": [{ "x": 0, "y": 3, "width": 2, "height": 4, }, { "x": 3, "y": 7, "width": 2, "height": 4, }] }, " personSchedules": { "valueList": [</pre>
--	---

	<pre> { "repeat": 127, "startTime ": "21:00", "endTime": "10:00", "status": true, "index": 0 }], "valueListNum": 1 } } </pre>		
请求备注	无		
响应备注	参数名称	参数说明	取值
	title	设备名称	用户自定义
	status	设备开关	On/Off
	soundsensitivity	声音灵敏度	取值：80、85、90、95、100 默认是 90，值越大灵敏度越高
	motionSensitivity	运动灵敏度	取值：30、50、80、90、100 默认是 80，值越大灵敏度越高
	HDVideo	高清视频	On/Off
	cameraMicrophone	设备麦克风	On/Off
	statusLight	状态指示灯	On/Off
	nightVision	夜视状态	Auto/On/Off
	cameraImageRotate	旋转图像 180 度	On/Off
	antiFlicker	防闪烁	On/Off
	motionTrack	运动跟踪开关	On/Off
	motionTrackBackTime	运动跟踪回到原来位置的时间	秒数
	motionRegions	检测区域 5x5=25 个区块，标号分别是 1~25，从左到右 S 形排列，多个以英文分号分割。如果不传值则表示全选	示例： "1;2;3;25"
	cameraOffSchedule	设置 Camera 关闭的 schedule	具体参数含义可以参考：设置 TD 关闭 schedule
	personStatus	客流统计开关状态	On/Off
	personRegion	客流统计检测区域	具体参数含义可以参考：设置客流统计检测区域
	personSchedules	客流统计打开的 schedule	具体参数含义可以参考：设置客流统计 schedule
备注	TD 在注册完毕之后，或者在每次设备重启的时候需要从平台获取在线配置，确保和平台在线配置保持实时同步。此请求一般发生在连接 HEDGW 服务器之前		

5.1.6 更新在线配置

简介	当 TD 某些状态变化时,需要通知器更新 TD 在线配置		
消息类型	HTTP1.1+JSON		
传输方向	TD->OpenAPI		
鉴权	是		
连接方式	短连接		
请求方法	POST		
请求格式	HTTP Header: Authorization:WSSE profile="UsernamePwd" X-WSSE:UsernamePwd Username="Zhejiang-110234567894562",PasswordDigest="dGhpcyBpcyBhIGV4YW1wbGUyYmNiOTI4YWM4ZjIzOTMxYTdhOTVhNjMyNGMwMmYzMWQxZWQwMjU5", Nonce="6FQHGU7M01ZMIA5G",Created="2017-01-01T15:00:00+08:00" HTTP Body: accessKey:5e2abe86-6aa signature: yyyy jsonObject: <pre>{ "msgType": "MSG_P_THD_SET_CAMERA_SETTINGS_REQ", "deviceMacAddress": "2059a074e0bc", "deviceId": "110234567894562", "deviceToken": "89Q832IF09JISDOFCJIOAWIOJ", "camParam" { "status": "On", "personStatus": "On", "wifiSSID": "xxxx", "sdcardStatus": 1, "binoParam": "xxxxx", "deviceAngle": "355:96:69:65:10:30" } }</pre>		
请求示例	参考 TD 注册请求示例		
响应格式	<pre>{ "errorCode": "0", "description": "", "msgType": "MSG_P_THD_SET_CAMERA_SETTINGS_RSP" }</pre>		
请求备注	参数名称	参数说明	取值

	Status	设备开关: 当 TD 根据 App 设置的 schedule 主动打开/关闭 TD 时, 需要将开关状态主动通知服务器	On/Off
	personStatus	客流统计开关: 当客流统计功能根据 App 设置的 schedule 主动打开/关闭时, 需要将客流统计开关状态主动通知服务器	On/Off
	wifiSSID	Wifi 的 SSID: 当 TD 连接 wifi 成功之后, 或者 wifi 切换成功之后, 需要将新连接 wifi 的 SSID 上报到服务器	SSID 名称
	sdcardStatus	SD 卡状态: 当 SD 卡状态变化 (插入/弹出), 或 camera 上线时, camera 需要将 SD 卡状态上报到服务器	1: 无 SD 卡 2: SD 卡异常 3: SD 卡正常
	deviceAngle	用于 PTZ 摄像头全景拼接参数计算, 需要上报如下参数: - 设备水平方向最大旋转角度 - 镜头水平方向视场角 - 镜头垂直方向的视场角 - 对于水平方向的仰角 - 对于水平方向的俯角 - 设备默认的每次截图的旋转角度 上述参数使用冒号分隔。没有 PTZ 的设备没有该字段	
	BinoParam	双目摄像头相关参数	
备注			

5.1.7 更新 TD 版本信息

简介	更新 TD 信息
消息类型	HTTP1.1+JSON
传输方向	TD->OpenAPI
鉴权	是
连接方式	短连接
请求方法	POST
请求格式	HTTP Header: Authorization:WSSE profile="UsernamePwd" X-WSSE:UsernamePwd Username="Zhejiang-110234567894562",PasswordDigest="dGhpcyBpcyBhIGV4YW1wbGUyYmNiOTI4YWM4ZjZlOTMxYTdhOTVhNjMyNGMwMmYzMWQxZWQwMjU5",Nonce="6FQHGU7M01ZMIA5G",Created="2017-01-01T15:00:00+08:00" HTTP Body: accessKey:5e2abe86-6aa signature: yyyy

	jsonObject: <pre>{ "msgType": "MSG_P_THD_SET_CAMERA_INFO_REQ", "deviceMacAddress": "2059a074e0bc", "deviceId": "110234567894562", "deviceModelId": "Zhejiang_zj_cnmobile_001", "firmwareVersion": "1.2.300", "cameraAppVersion": "2.15.200", "deviceToken": "89Q832IF09JISDOFCJIOAWIOJ" }</pre>	
请求示例	参考 TD 注册请求示例	
响应格式	<pre>{ "errorCode": "0", "description": "", "msgType": "MSG_P_THD_SET_CAMERA_INFO_RSP" }</pre>	
请求备注	firmwareVersion	固件版本， 例如： 1.2.300
	cameraAppVersion	应用程序版本， 例如： 2.15.200
响应备注	无	
备注	TD 在注册完毕之后，或固件信息发生改变后需要向平台更新 TD 信息。确保后期能够正确的获取固件更新信息。	

5.2 HEDGW 信令通道接入方式

HEDGW 与 TD 之间交互包含两部分，第一是与 TD 之间的信令交互，第二是与 TD 之间的码流传输。两种都是通过长连接进行传输，数据包发送格式为：数据包长度（4 字节）+数据包内容（序列化后的 Protobuf 数据），Protobuf 消息定义，见

附录 D：Protobuf 文件。

TD 必须采用 TLS1.2 协议中 AEAD 模式与 HEDGW 建立加密的通信管道。并采用 Protobuf 作为数据交换格式。详细协议格式见附录 D。

信令消息主要包含：

```
message HEDGWMessage {
  enum MessageType { DevLoginReq = 1; DevLoginResp = 2; DevEventCmd = 4;
    DevServerCmd = 5; DevAppCmd = 6; DevIpCamCmd = 7; PING = 8; PONG = 9;
    TDMediaPackage = 10; }
  required MessageType message_type = 1;
  optional DevLoginReq dev_login_req = 2;
  optional DevLoginResp dev_login_resp = 3;
  optional DevEventCmd dev_event_cmd = 5;
  optional DevServerCmd dev_server_cmd = 6;
```

```

optional DevAppCmd dev_app_cmd = 7;
optional DevIpCamCmd dev_ipcam_cmd = 8;
optional Ping ping = 9;
optional Pong pong = 10;
optional TDMediaPackage td_media_package = 11;
}

```

5.2.1 登录 HEDGW 平台

简介	登录 HEDGW 平台	
消息类型	Protobuf	
传输方向	TD->HEDGW	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	<pre> message DevLoginReq { required string access_key = 1; required string access_secret = 2; required string user_name = 3; required string device_token = 4; required string unified_id = 5; required string device_mac = 6; required uint32 channel_id = 9; optional uint32 video_type = 10; required uint32 video_width = 11; required uint32 video_height = 12; required uint32 audio_type = 13; required uint32 audio_khz = 14; required uint32 audio_channel = 15; } </pre>	
响应格式	<pre> message DevLoginResp { required uint32 result = 1; optional string server_time = 4; optional uint32 device_status = 5; optional uint32 ping_pong_span = 6; optional uint32 sd_upload_mode = 7; optional uint32 person_count_span = 8; optional uint32 service_days = 9; } </pre>	
请求备注	access_key	TD 厂商 KEY
	access_secret	TD 厂商 KEY 对应的密钥
	user_name	用户的名称/手机号

	device_token	平台访问 token，可以通过 OpenAPI 获取
	unified_id	账号唯一标识符，可以通过 OpenAPI 获取
	device_mac	TD 的 MAC 地址，和注册时格式保持一致
	channel_id	0: 实时流，1: SD 卡数据流
	video_type	1: h264 2:h265，默认为 h264
	video_width	TD 分辨率宽
	video_height	TD 分辨率高
	audio_type	1: G726, 2: G711A
	audio_khz	音频采样率例如采样率为 8khz，此处值即为 8
	audio_channel	音频声道数
响应备注	result	0: 成功 2001: 请求认证失败或者 token 失效 2008: 服务器限制 TD 上线，TD 需要等待一段时间再尝试（90 秒） 2026: 设备已经被删除
	server_time	服务器当前 UTC 时间，单位：毫秒
	device_status	设备套餐状态： 0: 试用套餐 1: 带云套餐 2: 服务过期 3: 事件录制套餐（当事件发生时才进行录制，需要支持预录，事件前面预录至少 4~5 秒，后面补录 10 秒） 44: 设备已经删除
	ping_pong_span	ping pong 间隔，单位：秒
	person_count_span	客流统计上报间隔（单位：秒）
	sd_upload_mode	SD 卡缓存数据方式： 1: 缓存断网期间内所有音视频 2: 缓存断网期间内事件视频
	service_days	（只有当套餐为全天录制或事件录制时有效）
备注	登录失败后，需要重新请求 HEDGW 服务地址（参考获取 HEDGW 服务）	

5.2.2 TD 心跳保活

简介	TD 和平台之间通过心跳包保活
消息类型	Protobuf
传输方向	TD->HEDGW
鉴权	是
连接方式	长连接;TLS1.2
请求格式	<pre> message Ping { required uint32 id = 1 [default = 0]; }</pre>

响应格式	<pre>message Pong { required uint32 id = 1 [default = 0]; }</pre>
请求备注	无
响应备注	无
备注	心跳间隔，在登录请求时返回，TD 登录之后应按照服务器返回的心跳间隔主动发送 Ping 消息

5.2.3 TD 上传音视频数据

简介	TD 设备向平台上传音视频数据包	
消息类型	Protobuf	
传输方向	TD->HEDGW	
鉴权	是	
连接方式	长连接; TLS1.2	
请求格式	<pre>message TDMediaPackage { required uint32 package_type = 1; required uint32 sync = 2; required bytes data = 3; required uint32 data_size = 4; required uint32 start_time = 5; required uint32 seq_num = 6; optional uint64 ipcam_time = 7; optional uint32 control_flag = 8; }</pre>	
响应格式	无	
请求备注	package_type	1:audio, 2:video, 3:sps/pps
	sync	0:非关键帧 1:关键帧
	data	音视频数据
	data_size	数据大小，单位：字节
	start_time	音视频时间戳，溢出之后重新从 0 开始，单位：毫秒 注意：音频打包的时间戳要求是 20ms 或者 40ms
	seq_num	帧序列号，音视频分别编号，溢出之后重新从 0 开始
	ipcam_time	TD 设备 UTC 时间，单位：毫秒
	control_flag	控制标记，只有事件录制套餐的时候需要
响应备注	无	
备注	<p>如果设备套餐状态为云存储状态，设备需要一直上传流媒体数据到服务器进行存储。如果设备套餐状态为过期，设备只有在收到服务器通知的时候才需要上传流媒体数据（DevServerCmd），上传关键帧数据之前，需要上传 SPS/PPS 数据。</p> <p>如果摄像头套餐是事件录制套餐，摄像头只需在有事件发生的时候将视频上传到云</p>	

	<p>端，摄像头需要缓存数据以支持视频预录功能，事件前面至少预录 4~5 秒，事件后面补录 10 秒。</p> <p>对于事件录制套餐的摄像头，摄像头上传视频有两种触发条件：一是有事件发生，二是有客户端在进行播放。因此在 TDMediaPackage 消息中增加字段 control_flag（所有包都需要带上此标记，包括 audio，video，sps/pps）：</p> <ol style="list-style-type: none"> 1. 事件触发的视频上传：control_flag=1，服务器会将视频录制到云端 2. 客户端播放触发的视频上传，control_flag=2，这时服务器不会将视频录制到云端 3. 客户端播放过程中，如果有事件发生，control_flag 从 2 变为 1，这时服务器会将 control_flag 为 1 缓存的视频补充到事件视频前面，作为预录视频
--	--

5.2.4 TD 上传 SD 卡缓存音视频数据和报警事件

简介	TD 向平台上传 SD 卡缓存的音视频数据包	
消息类型	Protobuf	
传输方向	TD->HEDGW	
鉴权	是	
连接方式	长连接; TLS1.2	
请求格式	<pre>message TDMediaPackage { required uint32 package_type = 1; required uint32 sync = 2; required bytes data = 3; required uint32 data_size = 4; required uint32 start_time = 5; required uint32 seq_num = 6; optional uint64 ipcam_time = 7; }</pre>	
响应格式	无	
请求备注	ipcam_time	SD 卡数据产生时的 UTC 时间，单位：毫秒
响应备注	其他参数解释参考：TD 上传音视频数据	
备注	<p>TD 如果支持 SD 卡，可以在网络状态异常时将数据和报警事件记录缓存到 SD 卡中，当网络恢复之后，TD 和 HEDGW 服务器可以建立一条新的连接(message DevLoginReq 中 channel_id=1)，利用此连接将缓存在 SD 卡的数据和报警事件记录上传到云端服务器。</p> <p>上传数据时需要填写音视频包产生时的 UTC 时间戳</p>	

5.2.5 TD 接收客户端音频数据（语音对讲）

简介	设备接收客户端音频数据（语音对讲）
消息类型	Protobuf
传输方向	HEDGW->TD

鉴权	是	
连接方式	长连接; TLS1.2	
请求格式	<pre>message TDMediaPackage { required uint32 package_type = 1; required uint32 sync = 2; required bytes data = 3; required uint32 data_size = 4; required uint32 start_time = 5; required uint32 seq_num = 6; optional uint64 ipcam_time = 7; }</pre>	
响应格式	无	
请求备注	package_type	1: audio
	sync	0
	data	音频数据
	data_size	音频数据大小, 单位: 字节
	start_time	音频时间戳, 单位: 毫秒
	seq_num	音频包序列号
	ipcam_time	UTC 时间, 单位: 毫秒
响应备注	无	
备注	该命令主要用于双向语音通话过程中, 设备端接收客户端采集的用户声音。音频格式为 G711 a-law, 采样率为: 8kHz, 通道数为 1	

5.2.6 TD 事件上传

简介	TD 事件上传	
消息类型	Protobuf	
传输方向	TD->HEDGW	
鉴权	是	
连接方式	长连接; TLS1.2	
请求格式	<pre>message DevEventCmd { required uint32 type = 1; required uint32 status = 2; required uint64 ipcam_time = 3; optional string msg = 4; }</pre>	
响应格式	无	
请求备注	type	1: 动态检测事件
		2: 声音检测事件
		3: 人脸检测事件

		4: 客流统计事件 5: 人形检测事件
	status	0: 结束, 1: 开始
	ipcam_time	时间戳, 单位: 毫秒
	msg	只有当 type 为 4 时需要填写
响应备注	无	
备注	<p>TD 有相应告警信息发生时, 需要告知到 HEDGW。</p> <p>对于客流统计事件: status 值始终为 1, 上报间隔需要根据 TD 登录 HEDGW 平台时服务器返回的间隔值来上报。上报时需要把客流信息填写到字段 msg 中, 使用 json 格式, 需要包含的字段有:</p> <p>"start": 开始时间 (单位: 秒, UTC 时间)</p> <p>"stop": 结束时间 (单位: 秒, UTC 时间)</p> <p>"count": 统计的客流人数 (int 类型, 高 16 位代表 TD 客流统计区域在统计时间内从下到上的客流人数, 低 16 位代表 TD 客流统计区域在统计时间内从上至下的客流人数)</p> <p>例如:</p> <pre>{ "start":1500631027, "stop":1500631627, "count":100 }</pre>	

5.2.7 TD 上传设备缩略图

简介	TD 设备向平台上传缩略图/定时截图	
消息类型	Protobuf	
传输方向	TD->HEDGW	
鉴权	是	
连接方式	长连接; TLS1.2	
请求格式	<pre>message TDMediaPackage { required uint32 package_type = 1; required uint32 sync = 2 ; required bytes data = 3; required uint32 data_size = 4; required uint32 start_time = 5; required uint32 seq_num = 6; optional uint64 ipcam_time = 7; }</pre>	
响应格式	无	
请求备注	package_type	5: 设备缩略图

	sync	0: non-key frame
	data	JPG 图片
	data_size	图片大小, 单位: 字节
	start_time	TD 启动相对时间, 单位: 毫秒
	ipcam_time	UTC 时间, 单位: 毫秒
响应备注	无	
备注	TD 需要根据服务器的控制命令 (DevServerCmd) 决定是否上传缩略图	

5.2.8 TD 接收 HEDGW 通知

简介	服务器发送命令通知 TD 状态变更, 或上传缩略图等操作	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	<pre>message DevServerCmd { required uint32 type = 1; optional uint32 device_status = 2; optional uint32 thumbnail_span = 3; optional uint32 thumbnail_width = 4; optional uint32 thumbnail_height = 5; optional uint32 service_days = 6; }</pre>	
响应格式	无	
请求备注	type	7: 设备套餐状态变更, 9: 上传缩略图
	device_status	type=7 时有效: 0: 试用套餐 1: 带云套餐 2: 服务过期 3: 事件录制套餐 44: 设备已经删除
	thumbnail_span	上传缩略图的时间间隔, type=9 时有效, 单位: 秒。 (值为 0 时停止上传缩略图)
	thumbnail_width	缩略图的分辨率: 宽度, type=9 时有效
	thumbnail_height	缩略图的分辨率: 高度, type=9 时有效
	service_days	套餐天数 (只有当套餐为全天录制或事件录制时有效)
响应备注	无	
备注	套餐变更和删除设备等 TD 状态变更, 平台会主动通知 TD。如果平台通知 TD 状态变更为 0 或 1 (带云套餐), 那么 TD 需要主动向平台上传音视频数据	

5.2.9 客户端控制 TD 命令

简介	APP 发送命令给 TD	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	<pre>message DevAppCmd { required uint32 type = 1; required string src_device_id = 2; optional string dst_device_id = 3; optional uint32 count = 4; optional uint32 id = 5; optional string msg = 6; optional uint32 data_size = 7; optional bytes data = 8; }</pre>	
响应格式	无	
请求备注	type	0: 在线客户端数量 8: 客户端直播请求 9: 客户端直播结束 33: APP 发送的设置命令, 当为此值时, TD 需要解析 msg 字段
	src_device_id	源设备标识符
	dst_device_id	目标设备标识符
	count	在线客户端数量, type=0 时有效
	id	命令编号, type=33 时有效
	msg	基本格式如下: 设置前端设备的时候 <pre>{ "msgSession": 0x12345678, "msgSequence": 0, "msgTimeStamp": 1381313133, "msgCategory": "camera", "msgContent": { "request": 0x071C, "requestParams": { "value": 1 } } }</pre>

		如果存在子请求，格式如下： <pre>{ "msgSession": 0x12345678, "msgSequence": 0, "msgTimeStamp": 1381313133, "msgCategory": "camera", "msgContent": { "request": 0x0700, "subRequest": 0x0010, "requestParams": { } } }</pre>
响应备注	无	
备注	HEDGW 会中转 App 的一些请求命令给 TD，例如：设置云台转动，设置分辨率等； TD 不需要关注消息中的 msgSession, msgSequence, msgTimeStamp, msgCategory 字段	

5.2.9.1 客户端开始/停止播放

简介	客户端 App 开始/停止播放	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	<pre>message DevAppCmd { required uint32 type = 1; required string src_device_id = 2; optional string dst_device_id = 3; optional uint32 count = 4; optional uint32 id = 5; optional string msg = 6; optional uint32 data_size = 7; optional bytes data = 8; }</pre>	
响应格式	无	
请求备注	type	8: 客户端直播请求 9: 客户端直播结束
	src_device_id	源设备标识符
	dst_device_id	目标设备标识符
响应备注	无	
备注	对于无套餐设备，TD 需要在有客户端直播请求的时候主动上传音视频数据，对于有	

	套餐设备，TD 无需关注此消息
--	-----------------

5.2.9.2 设置运动检测灵敏度

简介	设置 TD 的运动检测灵敏度	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	下面的内容，封装在 message DevAppCmd 中的 msg 中： <pre>{ "msgContent": { "requestParams": { "value": 30 }, "subRequest": 15, "request": 1793 }, "msgSession": 1732480381, "msgSequence": 0, "msgTimeStamp": 1403494411491, "msgCategory": "camera" }</pre>	
响应格式	无	
请求备注	request	1793：表示配置设备命令
	subRequest	15：表示配置设备异动检测灵敏度
	value	30、50、80、90、100，默认是 80，值越大灵敏度越高
响应备注	无	
备注	设置 TD 的异动检测灵敏度	

5.2.9.3 设置声音检测灵敏度

简介	设置 TD 的声音检测灵敏度	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	下面的内容，封装在 message DevAppCmd 中的 msg 中： <pre>{ "msgContent": { "requestParams": {</pre>	

	<pre> "value": 90 }, "subRequest": 29, "request": 1793 }, "msgSession": 1732480381, "msgSequence": 0, "msgTimeStamp": 1403494411491, "msgCategory": "camera" } </pre>	
响应格式	无	
请求备注	request	1793: 表示配置设备命令
	subRequest	29: 表示配置设备声音灵敏度检测
	value	80、85、90、95、100, 默认是 90, 值越大灵敏度越高
响应备注	无	
备注	设置 TD 的声音检测灵敏度	

5.2.9.4 设置麦克风开关

简介	设置 TD 的麦克风开关	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	下面的内容, 封装在 message DevAppCmd 中的 msg 中: <pre> { "msgContent": { "requestParams": { "value": 0 }, "subRequest": 37, "request": 1793 }, "msgSession": 1227620865, "msgSequence": 0, "msgTimeStamp": 1403494575788, "msgCategory": "camera" } </pre>	
响应格式	无	
请求备注	request	1793: 表示配置设备命令
	subRequest	37: 表示配置设备麦克风开关
	value	0 表示静音

		1 表示打开
响应备注	无	
备注	设置 TD 的麦克风开关，麦克风关闭的时候，TD 需要发送静音数据到服务器	

5.2.9.5 高清/低清切换

简介	设置 TD 的分辨率	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	下面的内容，封装在 message DevAppCmd 中的 msg 中： <pre>{ "msgContent": { "requestParams": { "value": 0 }, "subRequest": 30, "request": 1793 }, "msgSession": 1377903621, "msgSequence": 0, "msgTimeStamp": 1403494504362, "msgCategory": "camera" }</pre>	
响应格式	无	
请求备注	request	1793：表示配置设备命令
	subRequest	30：表示配置设备画面清晰度
	value	1 表示 HD（设备支持的最大分辨率） 0 表示 SD（VGA）
响应备注	无	
备注	设置 TD 的分辨率。切换分辨率后，TD 发送的音视频数据时间戳需要重新从 0 开始计算。切换的时候，要从关键帧开始切，并且切换之后，关键帧前面要上报正确的 sps pps 数据 音视频数据发送参考 TD 上传音视频数据	

5.2.9.6 设置 180° 旋转

简介	设置 TD 视频画面的旋转	
消息类型	Protobuf	
传输方向	HEDGW->TD	

鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	下面的内容，封装在 message DevAppCmd 中的 msg 中： <pre> { "msgContent": { "requestParams": { "value": 180 }, "subRequest": 23, "request": 1793 }, "msgSession": 1958459063, "msgSequence": 0, "msgTimeStamp": 1403494583696, "msgCategory": "camera" } </pre>	
响应格式	无	
请求备注	request	1793：表示配置设备命令
	subRequest	23：表示配置设备画面旋转
	value	0 表示不旋转 180 表示旋转 180°
响应备注	无	
备注	设置 TD 视频画面的旋转	

5.2.9.7 设置状态指示灯

简介	设置 TD 指示灯开关
消息类型	Protobuf
传输方向	HEDGW->TD
鉴权	是
连接方式	长连接;TLS1.2
请求格式	下面的内容，封装在 message DevAppCmd 中的 msg 中： <pre> { "msgContent": { "requestParams": { "value": 0 }, "subRequest": 19, "request": 1793 }, "msgSession": 1377903621, "msgSequence": 0, </pre>

	<pre>"msgTimeStamp": 1403494504362, "msgCategory": "camera" }</pre>	
响应格式	无	
请求备注	request	1793: 表示配置设备命令
	subRequest	19: 表示配置设备指示灯开关
	value	0 表示关闭 1 表示打开
响应备注	无	
备注	设置 TD 指示灯开关	

5.2.9.8 设置夜视

简介	设置 TD 夜视开关	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	下面的内容，封装在 message DevAppCmd 中的 msg 中： <pre>{ "msgContent": { "requestParams": { "value": 1 }, "subRequest": 21, "request": 1793 }, "msgSession": 1377903621, "msgSequence": 0, "msgTimeStamp": 1403494504362, "msgCategory": "camera" }</pre>	
响应格式	无	
请求备注	request	1793: 表示配置设备命令
	subRequest	21: 表示配置设备夜视开关
	value	0 表示关闭 1 表示打开 2 表示自动
响应备注	无	
备注	设置 TD 夜视开关	

5.2.9.9 设置防闪烁

简介	设置 TD 防闪烁 (Antiflicker)	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	下面的内容，封装在 message DevAppCmd 中的 msg 中： <pre> { "msgContent": { "requestParams": { "value": 60 }, "subRequest": 36, "request": 1793 }, "msgSession": 1377903621, "msgSequence": 0, "msgTimeStamp": 1403494504362, "msgCategory": "camera" } </pre>	
响应格式	无	
请求备注	request	1793: 表示配置设备命令
	subRequest	36: 表示配置设备防闪烁频率
	value	50 表示 50HZ 60 表示 60HZ
响应备注	无	
备注	设置 TD 防闪烁	

5.2.9.10 设置 TD 开关状态

简介	设置 TD 的打开和关闭	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	下面的内容，封装在 message DevAppCmd 中的 msg 中： <pre> { "msgContent": { "requestParams": { "value": 1 }, "subRequest": 36, "request": 1793 }, "msgSession": 1377903621, "msgSequence": 0, "msgTimeStamp": 1403494504362, "msgCategory": "camera" } </pre>	

	<pre> "subRequest": -1, "request": 1815 }, "msgSession": 469395714, "msgSequence": 0, "msgTimeStamp": 1403494354219, "msgCategory": "camera" } </pre>	
响应格式	无	
请求备注	request	1815: 表示设备开关命令
	subRequest	无
	value	1 表示打开 4 表示强制关闭
响应备注	无	
备注	设置 TD 开关状态。关闭的时候，请停止推流，并保持长连接心跳	

5.2.9.11 设置云台机转动

简介	设置 TD 云台转动	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	下面的内容，封装在 message DevAppCmd 中的 msg 中： <pre> { "msgTimeStamp": 1457353997, "msgSession": 6633732, "msgSequence": 0, "msgCategory": "camera", "msgContent": { "request": 1793, "subRequest": 5, "requestParams": { "value": 1 } } } </pre>	
响应格式	无	
请求备注	request	1793: 表示配置设备命令
	subRequest	5: 控制云台机步进转动（步进长度设备厂商控制） 82: 控制云台机连续转动

	value	1 表示向左转动 2 表示向右转动 3 表示向上转动 4 表示向下转动 0: subRequest=5 的时候表示回到原始位置 SubRequest=82 的时候表示停止转动
响应备注	无	
备注	设置 TD 云台转动, TD 如果支持云台转动, 需要在注册 TD 的时候, 通过注册参数中的“comment”字段将该信息通知服务器。(参考 TD 注册)	

5.2.9.12 设置云台机绝对位置

简介	设置云台机转动到指定位置 (转到预置点)	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	下面的内容, 封装在 message DevAppCmd 中的 msg 中: <pre> { "msgTimeStamp": 1457353997, "msgSession": 6633732, "msgSequence": 0, "msgCategory": "camera", "msgContent": { "request": 1793, "subRequest": 81, "requestParams": { "pan": 50, "tilt": 5 } } }</pre>	
响应格式	无	
请求备注	request	1793: 表示配置设备命令
	subRequest	81: 设置云台机转动到指定位置
	requestParams	pan:水平位置坐标 tilt:垂直位置坐标
响应备注	无	
备注	pan 和 tilt 的取值范围为-1000000 ~ 1000000, 最左端和最下端位置为-1000000, 最右端和最上端位置为 1000000。中心位置为 0 设置 TD 云台转动, TD 如果支持云台转动, 需要在注册 TD 的时候, 通过注册参数中的“comment”字段将该信息通知服务器。(参考 TD 注册)	

5.2.9.13 设置运动检测区域

简介	设置 TD 运动检测区域
消息类型	Protobuf
传输方向	HEDGW->TD
鉴权	是
连接方式	长连接;TLS1.2
请求格式	<p>下面的内容，封装在 message DevAppCmd 中的 msg 中：</p> <pre> { "msgSession": 1801275770, "msgSequence": 0, "msgTimeStamp": 1457355191145, "msgCategory": "camera", "msgContent": { "msgVersion": 1, "request": 1793, "subRequest": 56, "requestParams": { "valueListNum": 2, "valueList": [{ "id": 0, "left": 3, "top": 2, "right": 4, "bottom": 3, "sensitivity": 0, "enable": 1 }, { "id": 1, "left": 3, "top": 3, "right": 4, "bottom": 4, "sensitivity": 0, "enable": 1 }] } } } </pre>

	如果 motionregion 全选，valueListNum 为 0，消息如下： <pre>{ "msgCategory": "camera", "msgContent": { "msgVersion": 1, "request": 1793, "requestParams": { "valueList": [], "valueListNum": 0 }, "subRequest": 56 }, "msgSequence": 0, "msgSession": 559204845, "msgTimeStamp": 1474438872521 }</pre>	
响应格式	无	
请求备注	request	1793：表示配置设备命令
	subRequest	56：表示配置设备异动检测区域
	valueList	其中坐标值的范围为 0~5 valueListNum：表示选择区域的个数 left：所选区域左边框坐标（范围 0~4） right：所选区域右边框坐标（范围 1~5） top：所选区域上边框坐标（范围 0~4） bottom：所选区域下边框坐标（范围 1~5）
响应备注	无	
备注	设置 TD 异动检测区域，检测区域最大为 25 个区域（5*5 排列），区域内容用数组表示，包含每个区域的坐标信息，以及所选区域的个数，如果所有区域全选，数组为空，区域个数为 0。 TD 如果支持移动区域检测设备，需要在 TD 注册的时候，通过注册参数中的“comment”字段将该信息通知服务器。（参考 TD 注册）	

5.2.9.14 设置固件升级

简介	设置 TD 固件升级
消息类型	Protobuf
传输方向	HEDGW->TD
鉴权	是
连接方式	长连接,TLS1.2
请求格式	下面的内容，封装在 message DevAppCmd 中的 msg 中：

	<pre>{ "msgContent": { "requestParams": { "serviceType": 1, "url": "http://xxxx.com/Device/910fb531-4d0/chinamobilem2m_3rd_001/1/1.0.0.526/xxx.enc", "newServiceVersion": "1.0.0.526", "checkSum": "ec5fd1e4b7e6d3a31f821203a19ccf43" }, "request": 4097 }, "msgSession": 914570390, "msgSequence": 0, "msgTimeStamp": 1502171866698, "msgCategory": "camera" }</pre>	
响应格式	无	
请求备注	request	4097: 表示设备升级命令
	subRequest	无
	url	固件更新包链接, 可以是 Fw, 也可以是 CamApp
	checkSum	校验值, 校验文件的完整性, 这里是固件更新包的 MD5
	serviceType	1:FW 升级包 2: CameraApp 升级包
	newServiceVersion	升级包版本号
响应备注	无	
备注	设置 TD 固件升级, 服务器会将固件包的 url 地址以及文件校验信息发给终端设备, TD 下载后进行升级, 下载及升级过程中, TD 需要向服务器反馈下载及更新进度 (通过 DevIpCamCmd 消息)	

5.2.9.15 修改 WIFI

简介	设置 TD 连接的 WIFI 地址
消息类型	Protobuf
传输方向	HEDGW->TD
鉴权	是
连接方式	长连接;TLS1.2
请求格式	下面的内容, 封装在 message DevAppCmd 中的 msg 中: <pre>{ "msgContent": { "requestParams": { "encryption": "WPA/WPA2", "ssid": "20d28", "bssid": "14:E6:E4:80:08:D",</pre>

	<pre> "key": "12345678" }, "subRequest": 17, "request": 1793 }, "msgSession": 1377903621, "msgSequence": 0, "msgTimeStamp": 1403494504362, "msgCategory": "camera" } </pre>	
响应格式	下面的内容，封装在 message DevIpCamCmd 中的 msg 中：（type=2） <pre> { "msgCategory": "camera", "msgContent": { "response": 0, //0: 成功, -1: 失败 "responseRequest": 1793, "responseSubRequest": 17 }, "msgSequence": 1, "msgSession": 394728981, "msgTimeStamp": 1484541696 } </pre>	
请求备注	request	1793: 表示配置设备命令
	subRequest	17: 表示修改 WIFI 配置
	encryption	加密方式
	ssid	路由器 WIFI 的 SSID
	bssid	路由器 WIFI 的 BSSID
	key	WIFI 密码
响应备注	TD 收到该请求，需要发送响应命令到客户端	
备注		

5.2.9.16 设置 TD 关闭 schedule

简介	设置 TD 关闭的 schedule，在指定的时间段内，TD 应处于关闭状态（停止推流，长连接保持）
消息类型	Protobuf
传输方向	HEDGW->TD
鉴权	是
连接方式	长连接;TLS1.2

请求格式	下面的内容，封装在 message DevAppCmd 中的 msg 中： <pre> { "msgContent": { "requestParams": { "valueList": [{ "repeat": 127, "startTime": "21:00", "endTime": "10:00", "status": true, "index": 0 }], "valueListNum": 1 }, "subRequest": 26, "request": 1793 }, "msgSession": 1377903621, "msgSequence": 0, "msgTimeStamp": 1403494504362, "msgCategory": "camera" } </pre>	
响应格式	无	
请求备注	request	1793：表示配置设备命令
	subRequest	26：表示设置 TD 关闭的 schedule
	valueList	status：“true”表示生效，“false”表示不生效
		repeat：低 7 位有效，按二进制从低位到高位依次表示周日到周六： SUN:0x0001, MON:0x0002, TUE:0x0004, WED:0x0008, THU:0x0010, FRI:0x0020, SATU:0x0040
		index：schedule 序号，0~6
		startTime：关闭时间，格式为 HH:mm
		endTime：打开时间，格式为 HH:mm
响应备注	无	
备注	<p>valueList 定义为数组，可能会有多个 schedule 的情况。如果 startTime > endTime，则表示跨天，例如：repeat=1, startTime 为 21:00，endTime 为 10:00，则设置的时间段为每周日晚上 21:00 至周一上午 10:00</p> <p>TD 收到的时间为 UTC 时间，TD 需要将其转换为本地时间进行处理</p> <p>当用户添加或者删除 schedule 时，服务器会更新整个 schedule 列表给 TD 设备，TD 只需根据最新的 schedule 列表来处理即可</p> <p>当 TD 根据收到的 schedule 配置打开/关闭 TD 时，需要将 TD 状态通知服务器，通知</p>	

接口参考：更新在线配置

5.2.9.17 设置客流统计检测区域

简介	设置 TD 客流统计检测区域	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	<pre>{ "msgSession": 1801275770, "msgSequence": 0, "msgTimeStamp": 1457355191145, "msgCategory": "camera", "msgContent": { "msgVersion": 1, "request": 1793, "subRequest": 94, "requestParams": { "valueListNum": 2, "valueList": [{ "x": 0, "y": 3, "width": 2, "height": 4, }, { "x": 3, "y": 7, "width": 2, "height": 4, }] } } }</pre>	
响应格式	无	
请求备注	Request	1793：表示配置设备命令
	subRequest	94：表示设置客流统计检测区域
	valueList	检测区域坐标信息：

		x: 左上角 x 坐标 y: 左上角 y 坐标 width: 宽度 height: 高度
响应备注	无	
备注	坐标信息的取值范围 0~1000, TD 设备根据比例计算相应的检测区域	

5.2.9.18 设置客流统计开关

简介	设置 TD 客流统计检测开关	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	<pre>{ "msgContent": { "requestParams": { "value": 1 }, "msgVersion": 1, "subRequest": 92, "request": 1793 }, "msgSession": 1377903621, "msgSequence": 0, "msgTimeStamp": 1403494504362, "msgCategory": "camera" }</pre>	
响应格式	无	
请求备注	Request	1793: 表示配置设备命令
	subRequest	92: 表示设置客流统计开关
	value	1: 打开 4: 关闭
响应备注	无	
备注		

5.2.9.19 设置客流统计 schedule

简介	设置客流统计打开的 schedule, 在指定的时间段内, 客流统计功能应处于打开状态	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	

连接方式	长连接;TLS1.2	
请求格式	下面的内容，封装在 message DevAppCmd 中的 msg 中： <pre> { "msgContent": { "requestParams": { "valueList": [{ "repeat": 127, "startTime ": "21:00", "endTime": "10:00", "status": true, "index": 0 }], "valueListNum": 1, "status": 1 }, "msgVersion": 1, "subRequest": 93, "request": 1793 }, "msgSession": 1154471779, "msgSequence": 0, "msgTimeStamp": 1403494395054, "msgCategory": "camera" } </pre>	
响应格式	无	
请求备注	Request	1793：表示配置设备命令
	subRequest	93：表示设置客流统计打开的 schedule
	valueList	status：“true”表示生效，“false”表示不生效
		repeat：低 7 位有效，按二进制从低位到高位依次表示周日到周六： SUN:0x0001, MON:0x0002, TUE:0x0004, WED:0x0008, THU:0x0010, FRI:0x0020, SATU:0x0040
		index：schedule 序号，0~6
		startTime：关闭时间，格式为 HH:mm
响应备注	无	
	备注 valueList 定义为数组，可能会有多个 schedule 的情况。 最多只允许设置 7 个。如果 startTime > endTime，则表示跨天，例如：repeat=1, startTime 为 21:00，endTime 为 10:00，则设置的时间段为每周日晚上 21:00 至周一上午 10:00。TD 收到的时间为 UTC 时间，TD 需要将其转换为本地时间进行处理。	

	当用户添加或者删除 schedule 时，服务器会更新整个 schedule 列表给 TD 设备，TD 只需根据最新的 schedule 列表来处理即可。当 TD 根据收到的 schedule 配置打开/关闭 TD 时，需要将 TD 状态通知服务器，通知接口参考：更新在线配置。
--	---

5.2.9.20 格式化 SD 卡

简介	格式化 SD 卡	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	<pre>{ "msgContent": { "requestParams": {}, "msgVersion": 1, "subRequest": 87, "request": 1793 }, "msgSession": 1377903621, "msgSequence": 0, "msgTimeStamp": 1403494504362, "msgCategory": "camera" }</pre>	
响应格式	无	
请求备注	request	1793: 表示配置设备命令
	subRequest	87: 表示格式化 SD 卡命令
响应备注	TD 收到命令之后，需要将执行结果返回服务器，返回格式见：TD 响应 SD 卡格式化	
备注		

5.2.9.21 设置运动跟踪开关

简介	配置运动跟踪开关	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	<pre>{ "msgContent": { "requestParams": {"value": 1}, "msgVersion": 1, </pre>	

	<pre> "subRequest": 109, "request": 1793 }, "msgSession": 1377903621, "msgSequence": 0, "msgTimeStamp": 1403494504362, "msgCategory": "camera" } </pre>	
响应格式	无	
请求备注	request	1793: 表示配置设备命令
	subRequest	109: 配置运动跟踪开关
	value	1:打开运动跟踪 0: 关闭运动跟踪
响应备注		
备注	设备需要预先上报此能力	

5.2.9.22 设置运动跟踪归位时间

简介	配置运动跟踪归位时间	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	<pre> { "msgContent": { "requestParams": {"value": 30}, "msgVersion": 1, "subRequest": 116, "request": 1793 }, "msgSession": 1377903621, "msgSequence": 0, "msgTimeStamp": 1403494504362, "msgCategory": "camera" } </pre>	
响应格式	无	
请求备注	request	1793: 表示配置设备命令
	subRequest	116: 配置运动跟踪归位时长
	value	配置运动跟踪多少秒内无画面变化, 摄像机自动回到旋转前位置。

响应备注	
备注	

5.2.9.23 设置全景拼接

简介	启动或停止全景拼接动作，PTZ 摄像头收到启动命令时，需要做如下操作： <ol style="list-style-type: none"> 1. 旋转至最左边或最右边（根据命令参数判断） 2. 旋转一周，旋转过程中按照指定角度值截取缩略图（640*360），并发送至服务器，需要考虑摄像头倒装的情况 3. 设备回到原来的位置 注意：设备如果支持全景拼接操作，需要在启动时就上报 PTZ 角度相关信息，参考： 5.1.6 更新在线配置	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	<pre>{ "msgContent": { "requestParams": {"value": 1, "angle": 30, "stand": 1}, "msgVersion": 1, "subRequest": 153, "request": 1793 }, "msgSession": 1377903621, "msgSequence": 0, "msgTimeStamp": 1403494504362, "msgCategory": "camera" }</pre> 封装在 message DevIpCamCmd 中的 msg 中：（type=2）	
响应格式	<pre>{ "msgCategory": "camera", "msgContent": { "response": 0, //0: 成功, 1: 失败 "responseRequest": 1793, "responseSubRequest": 153 }, "msgSequence": 1, "msgSession": 394728981, "msgTimeStamp": 1484541696 }</pre>	
请求备注	Request	1793: 表示配置设备命令
	subRequest	153: 全景拼接操作

	value	1:打开全景拼接 0: 停止全景拼接
	angle	PTZ 全景拼接时每次转动的角度
	stand	1: 正装（从最左边旋转至最右边） 0: 顶装（从最右边旋转至最左边）
响应备注	设备发送 response 消息之后, 启动全景拼接流程, 将截取的每张图片上传到服务器, 参考: TD 发送全景拼接图片	
备注	设备需要预先上报此能力	

5.2.10 客户端查询 TD 命令

客户端通过发送 DevAppCmd 消息查询 TD 相关信息, 消息内容封装在"msg"字段, 主要包含查询 WIFI 列表, 查询 SD 卡容量等功能。

TD 不需要关注"msg"字段中的 msgSession, msgSequence, msgTimeStamp, msgCategory 字段

5.2.10.1 查询 WIFI 列表

简介	查询 TD 所处网络环境的 wifi 信息列表	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	下面的内容, 封装在 message DevAppCmd 中的 msg 中: <pre>{ "msgContent": { "requestParams": {}, "subRequest": 16, "request": 1792 }, "msgSession": 292822550, "msgSequence": 0, "msgTimeStamp": 1484532654015, "msgCategory": "camera" }</pre>	
响应格式	无	
请求备注	request	1792: 表示查询设备命令
	subRequest	16: 表示查询 WIFI 列表
响应备注	TD 收到该消息之后, 需要发送响应消息, 格式参考 TD 发送 WIFI 列表	
备注		

5.2.10.2 查询 SD 卡信息

简介	查询 SD 卡容量信息	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	<pre>{ "msgContent": { "requestParams": {}, "msgVersion": 1, "subRequest": 85, "request": 1792 }, "msgSession": 1377903621, "msgSequence": 0, "msgTimeStamp": 1403494504362, "msgCategory": "camera" }</pre>	
响应格式	无	
请求备注	Request	1792: 表示配置设备命令
	subRequest	85: 表示查询 SD 卡容量信息
响应备注	TD 收到命令之后, 需要发送 SD 卡信息到服务器, 参考: TD 发送 SD 卡信息	
备注		

5.2.10.3 获取云台机当前位置

简介	获取云台机当前位置坐标 (预置点设置)	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	<p>下面的内容, 封装在 message DevAppCmd 中的 msg 中:</p> <pre>{ "msgTimeStamp": 1457353997, "msgSession": 6633732, "msgSequence": 0, "msgCategory": "camera", "msgContent": { "request": 1792, "subRequest": 81, "requestParams": {</pre>	

	<pre> } } }</pre>	
响应格式	无	
请求备注	request	1792: 表示设备查询命令
	subRequest	81: 获取云台机当前坐标位置
响应备注	需响应，响应格式参考： TD 发送云台机当前位置	
备注		

5.2.11 TD 消息上报

简介	TD 发送命令给平台	
消息类型	Protobuf	
传输方向	HEDGW->TD	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	<pre> message DevIpCamCmd { required uint32 type = 1; required string src_device_id = 2; optional string dst_device_id = 3; optional string msg = 4; optional uint32 data_size = 5; optional bytes data = 6; }</pre>	
响应格式	无	
请求备注	type	1: 固件更新进度 2: 响应消息
	src_device_id	终端设备标识符
	dst_device_id	目标设备标识符（从 DevAppCmd 命令中的 src_device_id 获取）
	msg	消息体
响应备注	无	
备注	TD 需要发送命令到 App，例如：固件更新进度反馈等	

5.2.11.1 TD 发送 WIFI 列表

简介	TD 收到客户端发送的获取 WIFI 列表请求之后，使用此消息将 WIFI 列表发送给客户端
消息类型	Protobuf

传输方向	TD->HEDGW	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	下面的内容，封装在 message DevIpCamCmd 中的 msg 中：（type=2） <pre> { "msgCategory": "camera", "msgContent": { "response": 0, "responseParams": [{ "bConnected": 0, "bssid": "B0:D5:9D:33:26:2", "encryption": "WPA2", "signal": 1, "ssid": "FORTEST" }, { "bConnected": 1, "bssid": "8C:AB:8E:9B:EE:E", "encryption": "WPA/WPA2", "signal": 5, "ssid": "Atom_2.4Ghz" }, { "bConnected": 0, "bssid": "E4:3F:A2:01:95:9", "encryption": "WPA/WPA2", "signal": 4, "ssid": "CMCC_019594" }], "responseRequest": 1792, "responseSubRequest": 16 }, "msgSequence": 1, "msgSession": 394728981, "msgTimeStamp": 1484541696 } </pre>	
响应格式	无	
请求备注	response	0: 获取成功
	responseParams	Wifi 列表信息
	bConnected	0:未连接 1:已连接

	bssid	路由器的 BSSID
	encryption	加密方式
	signal	信号强度, 范围 1~5
	ssid	路由器的 SSID
	responseRequest	1792: 和查询 WIFI 列表请求里的 Request 值对应
	responseSubRequest	16: 和查询 WIFI 列表请求里的 subRequest 值对应
	msgSession	session id: 和查询列表请求里的 msgSession 值对应
响应备注	无	
备注		

5.2.11.2 固件更新进度反馈

简介	设置 TD 固件升级进度反馈	
消息类型	Protobuf	
传输方向	TD->HEDGW	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	下面的内容, 封装在 message DevIpCamCmd 中的 msg 中: (type=1) <pre>{ "msgTimeStamp": 1457353997, "msgSession": 6633732, "msgSequence": 0, "msgCategory": "camera", "msgContent": { "request": 2049, "requestParams": { "firmwareversion": "1.25.630", "cameraappversion": "2.15.301", "status": "1~8 ", "currentSize": "50", "totalSize": "100", "comment": "comment" } } }</pre>	
响应格式	无	
请求备注	request	2049: 代表固件更新进度反馈
	subRequest	无
	msgSession	会话 ID, 需要和固件升级命令里的 msgSession 一致
	firmwareversion	Fw 的版本号, 这个是和 url 对应的新 Fw 版本号
	cameraappversion	Camera app 的版本号, 这个是和 url 对应的新 camera app

		版本号
	status	1---开始下载, 2---正在下载, 3---下载成功, 4---下载失败, 5---开始安装, 6---正在安装, 7---安装成功, 8---安装失败
	currentSize	已下载的字节数
	totalSize	总的字节数
	comment	失败的时候传入相关便于排查问题的日志, 最长不超过 2048 个字符
响应备注	无	
备注	TD 在处理固件下载和安装的过程中都可以通知 App 当前的状态（进度），让用户知道当前 TD 固件升级的状况。并不是每个状态都需要的，比如：Fw 正在更新中，可能没有网络。 TD 更新完成，重启之后，需要上报状态 7：安装成功	

5.2.11.3 分辨率修改通知

简介	分辨率修改时, TD 通知服务器	
消息类型	Protobuf	
传输方向	TD->HEDGW	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	<pre>message DevIpCamCmd { required uint32 type = 1; required string src_device_id = 2; optional string dst_device_id = 3; optional string msg = 4; optional uint32 data_size = 5; optional bytes data = 6; }</pre>	
请求备注	type	3: 分辨率修改通知
	src_device_id	终端设备标识符
	msg	分辨率标识, 例如 1280x720
响应格式	无	
响应备注	无	
备注		

5.2.11.4 TD 发送 SD 卡信息

简介	TD 收到客户端发送的查询 SD 卡信息命令, 返回 SD 卡信息
消息类型	Protobuf
传输方向	TD->HEDGW

鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	下面的内容，封装在 message DevlpCamCmd 中的 msg 中：（type=2） <pre> { "msgCategory": "camera", "msgContent": { "response": 0, "responseParams": [{ "name": " /dev/sda0", "fs": "vfat", "total": 2147483648, "remain": 2147480000 }, { "name": " /dev/sda1", "fs": "vfat", "total": 2132234448, "remain": 2147480000 }], "responseRequest": 1792, "responseSubRequest": 85 }, "msgSequence": 1, "msgSession": 394728981, "msgTimeStamp": 1484541696 } </pre>	
请求备注	response	0: 成功
	responseParams	SD 卡分区容量信息
	name	分区名称
	fs	分区格式
	total	分区总容量，单位：字节，类型：64 位整型
	remain	分区剩余容量，单位：字节，类型：64 位整型
响应格式	无	
响应备注	无	
备注		

5.2.11.5 TD 响应 SD 卡格式化

简介	TD 收到客户端发送的格式化 SD 卡命令，返回执行结果
消息类型	Protobuf

传输方向	TD->HEDGW	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	下面的内容，封装在 message DevIpCamCmd 中的 msg 中：（type=2） <pre>{ "msgCategory": "camera", "msgContent": { "response": 0, "responseRequest": 1793, "responseSubRequest": 87 }, "msgSequence": 1, "msgSession": 394728981, "msgTimeStamp": 1484541696 }</pre>	
请求备注	response	0: 成功 -1:失败
响应格式	无	
响应备注	无	
备注		

5.2.11.6 TD 发送云台机当前位置

简介	TD 收到客户端发送的获取云台机当前位置的请求后，响应消息	
消息类型	Protobuf	
传输方向	TD->HEDGW	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	下面的内容，封装在 message DevIpCamCmd 中的 msg 中：（type=2） <pre>{ "msgCategory": "camera", "msgContent": { "response": 0, "responseRequest": 1792, "responseSubRequest": 81, "responseParams": { "pan": 50, "tilt": 5 } }, "msgSequence": 1, "msgSession": 394728981, </pre>	

	"msgTimeStamp": 1484541696 }	
请求备注	response	0: 成功 -1:失败
	responseRequest	1792
	responseSubRequest	81
	responseParams	pan:水平位置坐标 tilt:垂直位置坐标
响应格式	无	
响应备注	无	
备注	pan 和 tilt 的参数范围参考: 5.2.9.12	

5.2.11.7 TD 发送全景拼接图片

简介	设备收到全景拼接请求之后, 启动拼接流程, 将截取的每张图片发送到服务器	
消息类型	Protobuf	
传输方向	TD->HEDGW	
鉴权	是	
连接方式	长连接;TLS1.2	
请求格式	图片内容和图片信息, 封装在 message DevIpCamCmd, 需要填写相关字段。	
请求备注	type	4: 透传数据请求
	dst_device_id	客户端 id, 从设置全景拼接请求字段中获取
	data	采用如下结构: cmdtype (4 字节) + sequence (4 字节) + count (4 字节) + size (4 字节) + pic_data 其中: cmdtype: 103: 全景拼接图片上传命令 sequence: 上传图片的序号, 从 1 开始 count: 本次拼接需要的总的图片张数, 设备端根据角度信息计算 size: 图片数据大小 pic_data: 二进制图片数据 注意: data 中的上述字段都采用网络字节序
响应格式	无	
响应备注	无	

5.2.12 维护性质接口

简介	前端设备发送命令给 HEDGW		
消息类型	Protobuf		
传输方向	TD->HEDGW		
鉴权	是		
连接方式	长连接;TLS1.2		
请求格式	<pre> message DevIpCamCmd { required uint32 type = 1; required string src_device_id = 2; optional string dst_device_id = 3; optional string msg = 4; optional uint32 data_size = 5; optional bytes data = 6; } </pre>		
响应格式	无		
请求备注	type	2: 前端设备统计信息	
	Msg	以 json 格式填写; <pre> { "statistics": { "startTime":1485011708000 "endTime": 1485098108000 "totalSize": 233312342, "totalVideoFrames": 200000, "dropVideoFrames": 23445, "totalAudioFrames": 200000, "dropAudioFrames": 23445, "onlineDuration": 36000, "disconnectTimes": 10, "rebootTimes":2, "delay":500, "openApiFailureTimes":0, "assignHedgwIpTimes":5, "selfDefine": xxxxxxxx }, "msgSession": 1377903621, "msgSequence": 0, "msgTimeStamp": 1403494504362, "msgCategory": "camera" } </pre>	
	startTime	开始时间, utc 时间, 单位: 毫秒	

		endTime	结束时间，utc 时间，单位：毫秒
		totalSize	总数据量大小，单位：MB
		totalVideoFrames	总的 video 帧数
		dropVideoFrames	丢弃的 video 帧数
		totalAudioFrames	总的 audio 帧数
		dropAudioFrames	丢弃的 audio 帧数
		onlineDuration	总在线时长，单位：秒
		disconnectTimes	掉线次数
		rebootTimes	重启次数
		delay	视频数据从采集到发送成功平均延迟时间，单位：毫秒
		openApiFailureTimes	调用 openapi 接口失败次数
		assignHedgwIpTimes	调用获取 HEDGW 服务的次数
		selfDefine	用户自定义参数，json 格式
响应备注	无		
备注	TD 设备定时上传丢帧率信息，码流统计信息，掉线次数，在线时长等信息，每 24 小时更新一次，服务器会存储统计信息。最好在夜间 00:00 到 04:00 之间上传。 TD 设备也可以上传一些自定义字段，用于后期的日志查询。		

5.3 HEDGW 码流接入方式

5.3.1 码流接入方式

视频云平台的 TD 接入网关 HEDGW 支持 protobuf 协议的推流方式，平台提供了基于 HEDGW 的 TCP 安全信令通道，上传 protobuf 格式的码流。由于是基于 TCP 通道进行传输，所以 QOS 有保障，但是会增加网络延时。需要 TD 端在上传阻塞的时候，根据自定义策略进行丢包，丢包以 GOP 为单位进行丢包

5.3.2 TD 编码要求

- 视频编码要求：对视频数据的压缩标准支持 H.264。
 - H.264 HD Quality (720P/1080P);
 - H.264 SD Quality (VGA);
 - H.264 Profile: Baseline/Main;
- 音频编码要求：视频监控 TD 所采用的音频编码器应支持以下标准： G.726/G.711A，音频采样率和声道数可以自定义(一般建议的采样率为 8K)。

6 TD 端流程说明

6.1 TD 注册登录

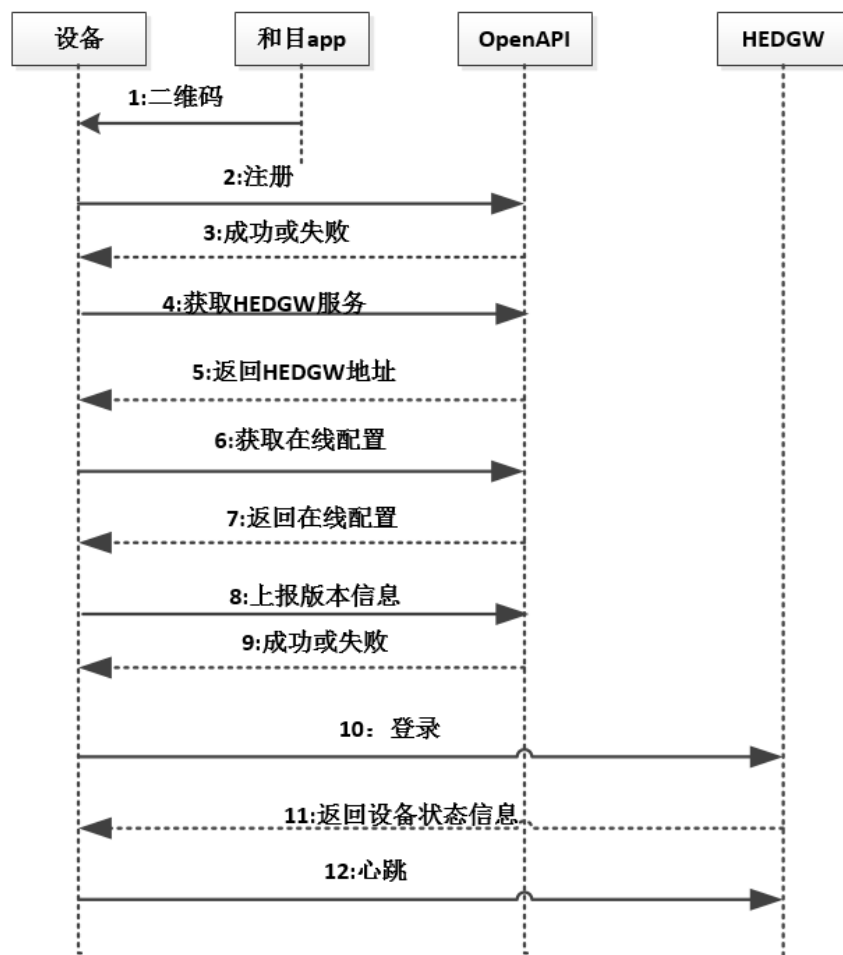


图 6.1 TD 注册登录时序图

- 扫描二维码: TD 通过扫描和目 App 的二维码获取用户账号信息、用户鉴权信息、Wifi 账号密码等。获取到的信息都通过前面申请的访问密钥进行加密。
- TD 注册: 通过 OpenAPI 设备注册接口将设备注册到平台。
- 获取 HEDGW 地址: TD 需要向 OpenAPI 请求获取信令通道地址, 以便后续接入 HEDGW, 进行 TD 与平台直接的信令交互, 及视频码流推送。
- 获取在线配置: 例如移动监测灵敏度、云存储套餐、麦克风开关等配置是保存在云平台上的, TD 在上线时需要向平台获取这些配置, 并对 TD 进行相关设置。
- TD 登录: 设备与 HEDGW 建立 TLS 长连接, 用于信令交互。登录成功平台会返回 TD 相关信息。
- TD 上报版本信息: TD 每次启动之后, 需要将版本信息上报到 OpenAPI 平台
- TD 心跳: TD 向 HEDGW 发送保活心跳包。

6.2 TD 上传码流

- TD 在下面几种情况下需要主动向平台发送音视频数据：
 - ✧ 在登录时，平台返回该设备具备云套餐。
 - ✧ 在连接过程中，平台主动通知 TD 状态变更为带套餐状态。
 - ✧ 没有套餐的情况下，收到客户端播放请求
- 然后 TD 通过 **protobuf** 协议向平台推送码流，**protobuf** 数据可以通过 **TLS** 方式进行推送。
- HEDGW 收到 TD 数据之后，进行录制存储、实时分发等。
- 所有 TD 在平台端存储的数据，都通过现有的和目 App 进行访问。

6.3 在线配置

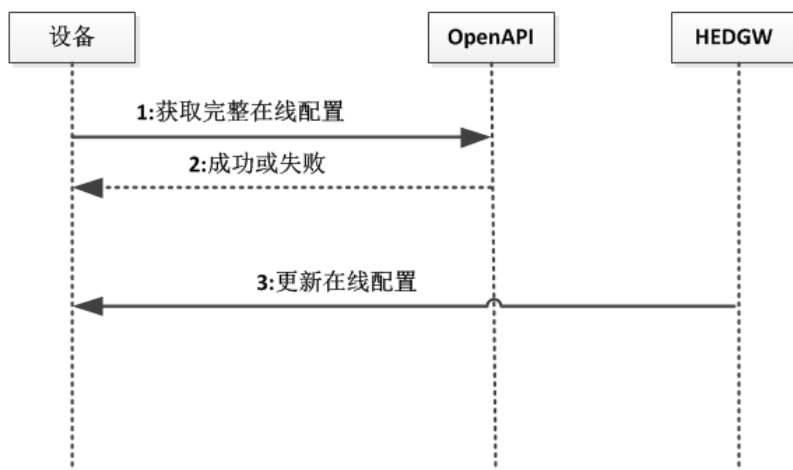


图 6.3 在线配置时序图

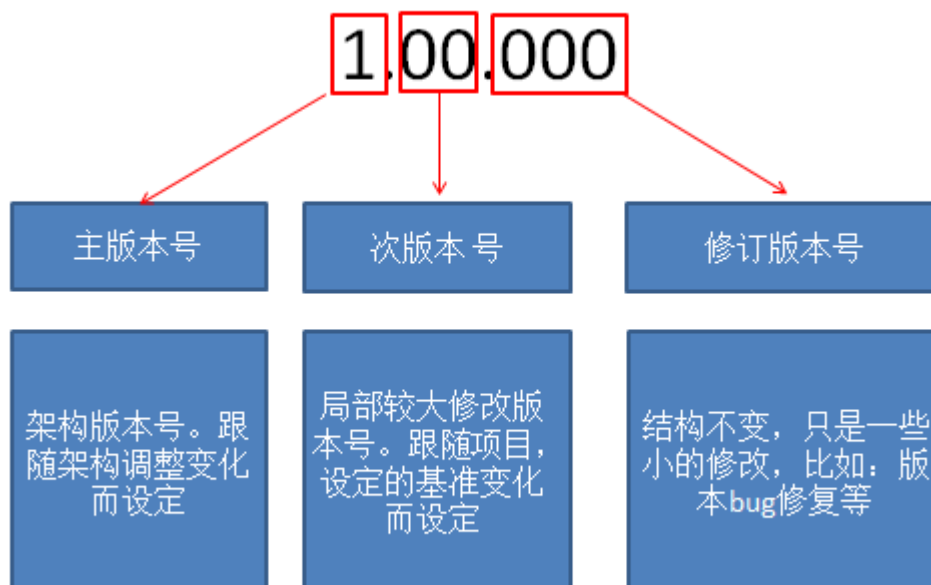
所谓的“在线配置”，是指保存在平台上的 TD 配置。TD 的在线配置必须随时与平台保持一致，因此，TD 注册后，在每次上线时 TD 需要向平台获取在线配置。

附录 A：TD 软件版本号要求

协议要求平台可以不对 TD 固件程序版本作出强制性的要求，这样 TD 需有自己独立、可持续演进的版本规划，平台遵循 TD 固件程序版本号来对设备进行升级等操作。

TD 厂商需向平台和甲方提供固件程序包规划说明文档和遵循自身规划演进硬件版本。

附录 B：协议版本演进说明



版本号说明

协议从 1.00.00 开始演进；协议要求厂商只能同时支持一个版本的协议，不能出现 A 接口属于版本 A1，B 接口属于版本 A2 的场景。

协议版本由主版本号、次版本号、修订版本号组成。

附录 C：OpenAPI HTTPS 错误码

错误码	描述
0	成功
400	请求错误
500	服务器内部错误
10005	密码错误
20000	服务器系统异常
20005	请求的 json 字符串为空
20008	请求的 json 字符串参数值为空
21001	json 解析错误

21113	设备已被其他账号注册
70002	Model ID 未初始化
70004	Deviceid 无效
80000	未定义错误，详情见返回的错误描述

附录 D: Protobuf 文件

将此附录 protobuf 文件保存为*.proto 文件，使用 protobuf 工具生成对应的*.pb.cc 和*.pb.h 文件，然后包含在代码工程中使用。

```

message HEDGWMessage {
  enum MessageType { DevLoginReq = 1; DevLoginResp = 2; DevEventCmd = 4;
  DevServerCmd = 5; DevAppCmd = 6; DevIpCamCmd = 7; PING = 8; PONG = 9;
  TDMediaPackage = 10; }
  required MessageType message_type = 1;
  optional DevLoginReq dev_login_req = 2;
  optional DevLoginResp dev_login_resp = 3;
  optional DevEventCmd dev_event_cmd = 5;
  optional DevServerCmd dev_server_cmd = 6;
  optional DevAppCmd dev_app_cmd = 7;
  optional DevIpCamCmd dev_ipcam_cmd = 8;
  optional Ping ping = 9;
  optional Pong pong = 10;
  optional TDMediaPackage td_media_package = 11;
}

message DevLoginReq {
  required string access_key = 1;           //TD 产品 KEY
  required string access_secret = 2;       //TD 产品 KEY 对应的密钥
  required string user_name = 3;           //用户的名称/手机号
  required string device_token = 4;        //平台访问 token，可以通过
OpenAPI 获取
  required string unified_id = 5;          //账号唯一标识符，可以通过 OpenAPI
获取
  required string device_mac = 6;          //TD 的唯一标识符，设备的 MAC 地
址
  required uint32 channel_id = 9;          //0:实时流, 1: SD 卡数据流
  required uint32 video_width = 11;        //TD 分辨率宽
  optional uint32 video_type = 10;         //视频编码格式
  required uint32 video_height = 12;       //TD 分辨率高
  required uint32 audio_type = 13;         //1:G726, 2:G711A
  required uint32 audio_khz = 14;         //音频采样率 8KHz, 16KHz

```

```
required uint32 audio_channel = 15;    //音频声道数
}

message DevLoginResp {
    required uint32 result = 1;          // 0:成功
    optional string server_time = 4;     // 服务器当前 UTC 时间, 单位: 毫
秒
    optional uint32 device_status = 5;   // 1:带云套餐, 2:服务过期, 44:
设备已经删除
    optional uint32 ping_pong_span = 6;  // TD 发送心跳包的间隔, 单位: 秒
    optional uint32 sd_upload_mode = 7;  // 1: 缓存断网期间内所有音视频,
2: 缓存断网期间内事件视频
    optional uint32 person_count_span = 8; // 客流统计上报间隔, 单位: 秒
}

message DevEventCmd {
    required uint32 type = 1;            // 1:动态检测 Motion, 2:Sound 声音,
3:人脸检测, 4: 客流统计
    required uint32 status = 2;          // 0:结束 1:开始
    required uint64 ipcam_time = 3;      // 时间戳
    optional string msg=4;
}

message DevServerCmd {
    required uint32 type = 1;            // 7:设备状态变更, 9:上传缩略图
    optional uint32 device_status = 2;   // 1:带云套餐, 2:服务过期, 44:
设备已经删除
    optional uint32 thumbnail_span = 3;  // 缩略图上传时间间隔, 单位 s, 0
表示停止上传
    optional uint32 thumbnail_width = 4;
    optional uint32 thumbnail_height = 5;
}

message DevAppCmd {
    required uint32 type = 1;            // 8: 客户端直播请求, 9: 客户端直播结束,
33:APP 信息
    required string src_device_id = 2;   // 来源设备标识符
    optional string dst_device_id = 3;   // 目标设备标识符
    optional uint32 count = 4;           // 在线客户端数量
    optional uint32 id = 5;              // 命令编号
    optional string msg = 6;             // 消息体
    optional uint32 data_size = 7;
```

```
optional bytes data = 8;
}

message DevIpCamCmd {
    required uint32 type = 1;          // 1: 固件更新进度, 2: 上传 TD 日志,
    3: 分辨率修改通知 4:透传数据
    required string src_device_id = 2; // 来源设备标识符
    optional string dst_device_id = 3; // 目标设备标识符
    optional string msg = 4;           // 消息体
    optional uint32 data_size = 5;
    optional bytes data = 6;
}

message Ping {
    required uint32 id = 1 [default = 0];
}

message Pong {
    required uint32 id = 1 [default = 0];
}

message TDMediaPackage {
    required uint32 package_type = 1; // 1:音频, 2:视频, 3:sps/pps, 5:
    缩略图
    required uint32 sync = 2 ;        // 0:非关键帧 1:关键帧
    required bytes data = 3;
    required uint32 data_size = 4;
    required uint32 start_time = 5; // 时间戳: 毫秒
    required uint32 seq_num = 6;
    optional uint64 ipcam_time = 7; // UTC 时间: 毫秒
    optional uint32 control_flag = 8;
}

option optimize_for = LITE_RUNTIME;
```



附录 E：接入代码示例

登录 HEDGW 服务器

登录流程分为 3 个步骤：

1. 建立 TCP 连接
2. 建立 SSL 连接，
3. 发送 camera 相关参数

1. 建立 TCP 连接

```
/* *****  
* param description:  
* @ip: HEDGW server ip, get it from open API  
* @port: HEDGW server port, get it from open API  
* *****/  
int SetupTcpConnection(char* ip, int port)  
{  
    struct sockaddr_in sin;  
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);  
    memset(&sin, 0, sizeof(struct sockaddr_in));  
    sin.sin_family = AF_INET;  
    sin.sin_addr.s_addr = inet_addr(ip);  
    sin.sin_port = htons(port);  
    if(connect(sockfd, (struct sockaddr *)&sin, sizeof(sin)) < 0)  
    {  
        printf("connect server failed.\n");  
        close(sockfd);  
        return -1;  
    }  
    return sockfd;  
}
```

2. 建立 SSL 连接

```
/* *****  
* param description:  
* @sockfd: socket fd, created from SetupTcpConnection()  
* *****/  
int SetupSslConnection(int sockfd)  
{  
    //创建 SSL context  
    const char* cipher_list =  
    "ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES1  
    28-SHA256:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AE  
    S256-GCM-SHA384";  
  
    SSL_METHOD *meth = NULL;  
    SSL_CTX* pssl_ctx = NULL;  
    int i = 0;  
    int seed_int[100]={0};  
  
    OpenSSL_add_ssl_algorithms();  
    SSL_load_error_strings();
```

```
meth = (SSL_METHOD *)TLSv1_2_client_method();
pssl_ctx = SSL_CTX_new(meth);
if (pssl_ctx == NULL)
{
return -1;
}
srand((unsigned)time(NULL));
for (i = 0; i < 100; i++)
{
seed_int[i] = rand();
}
RAND_seed(seed_int, sizeof(seed_int));
SSL_CTX_set_cipher_list(pssl_ctx, cipher_list);
SSL_CTX_set_mode(pssl_ctx, SSL_MODE_AUTO_RETRY);

//建立 SSL 连接
SSL* pssl = SSL_new(pssl_ctx);
if (SSL_set_fd(pssl, sockfd) <= 0)
{
return -1;
}
if(SSL_connect(pssl) <= 0)
{
return -1;
}
return 0;
}
```

3. 发送 camera 相关参数，建立 request 连接

```
/* *****
* param description:
* @sockfd: socket fd
* @pssl: ssl pointer
* @access_key: defined in document("27d7296f-101")
* @access_secret: defined in document("50r9q6mq4KNYYKYHJVJL")
* @username: user name of the account
* @cloud_token: generated when add camera
* @unified_id: generated when add camera
* @audio_type: audio type, defined in document
* @sample_rate: sample rate of audio
* @audio_chan_num: channel num of audio
* *****/
```

```
int SendRequest(int sockfd, SSL* pssl, char* access_key, char* access_secret, char* username,
```

```
char* cloud_token, char* unified_id, char* deviceid, int audio_type, int sample_rate,int
audio_chan_num)
{
    HEDGWMessage info;
    info.set_message_type(HEDGWMessage::DevLoginReq);
    DevLoginReq* req = info.mutable_dev_login_req();
    req->set_access_key(access_key);
    req->set_access_secret(access_secret);
    req->set_user_name(username);
    req->set_device_token(cloud_token);
    req->set_unified_id(unified_id);
    req->set_device_id(deviceid);
    req->set_channel_type(1);
    req->set_channel_id(0);
    req->set_video_width(1280);
    req->set_video_height(720);
    req->set_audio_type(audio_type);
    req->set_audio_khz(sample_rate/1000);
    req->set_audio_channel(audio_chan_num);

    int buflen=1024;
    char tcpbuffer[buflen];
    if (!info.SerializeToArray(tcpbuffer+4, buflen-4))
    {
        printf("serialize request info error\n");
        return -1;
    }
    int datasize_nl = htonl(info.ByteSize());
    memcpy(tcpbuffer,&datasize_nl,4);
    buflen = 4 + info.ByteSize();
    if (SSL_write(pssl, tcpbuffer, buflen) <= 0)
    {
        return -1;
    }

    int recvlen = 0;
    if (SSL_read(pssl, (void*)&recvlen,sizeof(int)) != sizeof(int)) //先读取长度
    {
        return-1;
    }
    recvlen = ntohl(recvlen);
    ::google::protobuf::uint8 recvbuf[recvlen];
    memset(recvbuf, 0, recvlen);
```

```

if(SSL_read(pssl, (void*)recvbuf,recvlen) != recvlen)
{
return -1;
}
HEDGWMessage res_info;
res_info.ParseFromArray(recvbuf, recvlen);
if (res_info.dev_login_resp().result() != 0)
{
printf("login error, result=%d\n",res_info.dev_login_resp().result());
return -1;
}
if (res_info.dev_login_resp().has_ping_pong_span())
{
printf("ping pong span is :%d(s).\n",res_info.dev_login_resp().ping_pong_span());
}
if (res_info.dev_login_resp().has_device_status() )
{
int status = res_info.dev_login_resp().device_status();
printf("camera service status is :%d.\n",status);  //// 1:带云套餐, 2:服务过期, 44:设备已经删除
}
return 0;
}

```

发送 Ping/pong 心跳包

连接建立之后，需要和 HEDGW 服务器之间发送 ping/pong 消息保持连接

1. 主动发送 ping 消息:

```

/* *****
* param description:
* @sockfd: socket fd
* @pssl: ssl pointer, created from SetupSslConnection()
* *****/
intSendPingMsg(int sockfd, SSL* pssl )
{
    HEDGWMessage info;
    info.set_message_type(HEDGWMessage::PING);
    Ping* ping = info.mutable_ping();
    ping->set_id(0);
    int buflen=100;
    char tcpbuffer[buflen];

```

```

    if (!info.SerializeToArray(tcpbuffer+4, buflen-4))
    {
        printf("serialize info error\n");
        return -1;
    }
    int datasize_nl = htonl(info.ByteSize());
    memcpy(tcpbuffer,&datasize_nl,4);
    buflen = 4+info.ByteSize();
    if (SSL_write(pssl, tcpbuffer, buflen) != buflen)
    {
        return -1;
    }
    return 0;
}

```

2. 收到 ping 消息之后，需要发送 pong 响应

```

/* *****
* param description:
* @sockfd: socket fd
* @pssl: ssl pointer, created from SetupSslConnection()
* *****/
int SendPongMsg(int sockfd, SSL* pssl )
{
    HEDGWMessage info;
    info.set_message_type(HEDGWMessage::PONG);
    Pong* pong = info.mutable_pong();
    pong->set_id(0);
    int buflen=100;
    char tcpbuffer[buflen];

    if (!info.SerializeToArray(tcpbuffer+4, buflen-4))
    {
        printf("serialize info error\n");
        return-1;
    }
    int datasize_nl = htonl(info.ByteSize());
    memcpy(tcpbuffer,&datasize_nl,4);
    buflen = 4+info.ByteSize();
    if(SSL_write(pssl, tcpbuffer, buflen) != buflen)
    {
        return-1;
    }
    return 0;
}

```



```
}
```

读取 HEDGW 服务器消息

连接过程中，需要一直读取 HEDGW 发送过来的消息

```
/* *****  
*   param description:  
*   @sockfd: socket fd  
*   @pssl: ssl pointer, created from SetupSslConnection()  
*   *****/  
int ReadHedgwMsg(int sockfd, SSL* pssl )  
{  
    HEDGWMessage info;  
    int recvlen = 0;  
  
    if(SSL_read(pssl, (void*)&recvlen,sizeof(int)) != sizeof(int)) //先读取长度  
    {  
        return-1;  
    }  
    recvlen = ntohl(recvlen);  
    ::google::protobuf::uint8 recvbuf[recvlen];  
    memset(recvbuf, 0, recvlen);  
    if(SSL_read(pssl, (void*)recvbuf,recvlen) != recvlen)  
    {  
        return -1;  
    }  
    info.ParseFromArray(recvbuf, recvlen);  
    if (info.message_type() == HEDGWMessage::PING)  
    {  
        return SendPongMsg(sockfd, pssl);  
    }  
    else if (info.message_type() == HEDGWMessage::PONG)  
    {  
        printf("pong message received\n");  
    }  
    else if (info.message_type() == HEDGWMessage::DevServerCmd)  
    {  
        if(info.dev_server_cmd().type() == 7)  
        {  
            printf("device status change to %d.\n",info.dev_server_cmd().device_status()); //1:带云套餐,  
            2:服务过期, 44:设备已经删除  
        }  
    }  
}
```

```
else if(info.message_type() == HEDGWMessage::DevAppCmd)
{
    if(info.dev_app_cmd().type() == 33 && info.dev_app_cmd().has_msg())
    {
        HandleClientCmd(info.dev_app_cmd().msg().c_str()); //参考下面的函数
    }
}
else if(info.message_type() == HEDGWMessage::MediaPackage
&& info.media_package().package_type() == 1)
{
    printf("this is audio talk data.\n"); //去掉 12 字节的 RTP 包头,剩余则是 G711 格式裸数据
}
return 0;
}
```

解析 HEDGW 转发过来的客户端 APP 设置 camera 的命令:

```
/* *****
*   used to handle request from client APP
*   param description:
*   @msg: json string, parsed from ReadHedgwMsg()
*   ***** */
void HandleClientCmd(const char* msg)
{
    Json::Reader reader;
    Json::Value j_value;
    Json::Value j_res;

    if (reader.parse(msg, j_value))
    {
        if (j_value.isMember("msgContent"))
        {
            int request = j_value["msgContent"]["request"].asInt();
            if(request == 1815) //设置 camera 关闭或打开
            {
                int value = j_value["msgContent"]["requestParams"]["value"].asInt();
                printf("client request to turn %s camera.\n",value==1?"on":"off");
            }
            if(request == 1793)
            {
                int sub_request = j_value["msgContent"]["subRequest"].asInt();
                printf("receive sub request:%d.\n", sub_request);
            }
        }
    }
}
```

}
}

中移物联网有限公司 (密)