**版本控制信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 文档名称： | 终端管理交互流程和接口设计 | | |
| 系统名称： | 云之翼云桌面管理平台 | | |
| 当前版本号： | 2.3 | 文档编号： |  |
| 撰写者： | 周里 | 日期： | 2020-06-03 |
| 密 级： |  | | |

**修改历史记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **修改** | **审批** |
| 2020-03-17 | 1.0 | 初稿 | 周里 |  |
| 2020-03-18 | 1.1 | 终端上传日志改为为一个压缩包 | 周里 |  |
| 2020-03-20 | 1.2 | 1、添加接口ClientLogin用于建立TCP链路和终端配置的关系 | 周里 |  |
| 2020-03-20 | 1.3 | 1、增加“终端配置更新”接口说明  2、修改对应thrift协议文件：添加ClientLogin、添加路由接口（兼容旧版本终端）  3、去掉“排序确认(order\_confirm)”的终端主动发起的命令，排序确定改用ManagerService的新接口order\_confirm(),这样终端可以收到确切的排序号码是否已经被采用  4、增加“排序取消”的命令功能，直接使用排序(order)命令，同一个“batch\_num”,序号“terminal\_id”= -1 | 周里 |  |
| 2020-03-20 | 1.4 | 1、修改thrift协议文件 | 周里 |  |
| 2020-03-21 | 1.5 | 1、添加备注：同一用户同时登录终端，先发“注销通知”再返回验证用户成功的返回 | 周里 |  |
| 2020-03-23 | 1.6 | 1、终端排序确认接口修改为command\_confirm增加接口函数的通用性 | 周里 |  |
| 2020-03-24 | 1.7 | 1、修改thrift协议文件，去掉多设计的密码字段  2、修改thrift协议文件，ScreenInfo修改为current\_scrren\_info,意思更加清晰  3、修改thrift协议文件，IPinfo修改为ip\_info  4、修改thrift协议文件，termial\_setup\_passwd修改为terminal\_setup\_passwd | 周里 |  |
| 2020-03-31 | 1.8 | 1、修改thrift协议文件，DesktopGropInfo修改为DesktopGroupInfo  2、修改thrift协议文件，string port修改为 i32 port  3、根据终端程序对于命令结构体的需求：  修改CommandMsg的说明 | 周里 |  |
| 2020-04-02 | 1.9 | 1、修改thrift协议文件，添加用户注销接口  2、修改thrift协议文件，RespDesktopInfo结构体修改返回错误码为string，添加错误描述字段 string description和已有的ResultInfo结构保持一致方便处理 | 周里 |  |
| 2020-04-09 | 2.0 | 1、用户登录后使用用户的会话ID来进行交互，无需每次接口传入用户名信息，涉及修改接口：user\_logout，get\_dskgrop\_info，get\_dskgrop\_info，desktop\_open  2、DesktopInfo结构体添加字段：dsk\_uuid用于存放桌面的唯一ID，用于关闭云桌面的接口使用  3、DesktopGroupInfo结构体添加字段：group\_uuid用于存放桌面组的唯一ID  4、添加接口all\_desktop\_close，用于终端发起关闭所有关联终端或者用户session相关的所有云桌面  5、关闭策略：修改字段close\_ strategy为两个字段：close\_desktop\_strategy、close\_terminal\_strategy  6、user\_login接口如果验证通过，返回的用户会话ID直接放在返回ResultInfo的Description字段里面 | 周里 |  |
| 2020-04-11 | 2.1 | 1、用户登录接口添加参数mac地址 | 周里 |  |
| 2020-05-03 | 2.2 | 1、排序查询接口返回值改为CommandMsg类型 | 周里 |  |
| 2020-06-03 | 2.3 | 1、DesktopGroupInfo结构体新增os\_type字段 | 周里 |  |

目录

[一、 整体介绍 4](#_Toc37516195)

[1. Thrift简介 4](#_Toc37516196)

[2. Thrift双向通信实现简介 5](#_Toc37516197)

[具体实现方案简介如下 5](#_Toc37516198)

[二、 功能点流程说明 6](#_Toc37516199)

[(一) 终端程序发起 6](#_Toc37516200)

[1. 终端用户认证、修改密码 6](#_Toc37516201)

[2. 云桌面相关接口 6](#_Toc37516202)

[3. 终端配置信息更新 7](#_Toc37516203)

[4. 终端获取配置信息 7](#_Toc37516204)

[5. 终端获取配置版本号 8](#_Toc37516205)

[6. 终端排序查询 8](#_Toc37516206)

[7. 终端获取服务器时间 8](#_Toc37516207)

[(二) 服务器端发起 8](#_Toc37516208)

[1. 服务器发送终端控制命令 8](#_Toc37516209)

[2. 终端程序升级全流程(不考虑离线升级) 10](#_Toc37516210)

[3. 终端日志文件上传全流程 11](#_Toc37516211)

[4. 终端配置更新 11](#_Toc37516212)

[5. 终端序号排序 11](#_Toc37516213)

[6. 终端IP更新 12](#_Toc37516214)

[7. 云桌面关闭通知 12](#_Toc37516215)

[8. 用户注销通知 12](#_Toc37516216)

[(三) 管理平台自有功能 13](#_Toc37516217)

[1. 终端删除 13](#_Toc37516218)

[三、 功能实现的thrift协议接口 13](#_Toc37516219)

[四、 问题 24](#_Toc37516220)

# 整体介绍

## Thrift简介

Thrift是一种接口描述语言和二进制通讯协议，它被用来定义和创建跨语言的服务。它被当作一个远程过程调用（RPC）框架来使用，是由Facebook开发的远程服务调用框架[Apache Thrift](http://thrift.apache.org/)。

本方案采用最新[thrift](https://github.com/apache/thrift)稳定发布版本：Apache Thrift v0.13.0

## Thrift双向通信实现简介

默认thrift是只支持单向rpc的调用，终端和服务器采用thrift实现通信交互，再自己实现了rpc的双向调用，两种rpc的调用在本次方案中都有使用到。

Thrift实现双向rpc调用实现原理：thrift底层是采用TCP通信实现的，TCP本身就是支持双向通信的。thrift默认情况下只有client端调用server端而发送TCP消息的单向rpc的调用，如果server端保存client的TCP链路信息，同时client端在连接上server端后对该TCP链路进行监听写入的TCP消息，这样就可以实现server端主动rpc调用client端。

具体实现方案简介如下：

客户端和服务器端各自都实现一个thrift的service服务用于双向通信调用，名称为ConnectService，服务的所有函数返回值为void，并且是oneway修饰。

1. service ConnectService {
2. oneway **void** Command(1:CommandMsg cmd\_msg);
3. oneway **void** TokenId(1:string id);
4. oneway **void** Ping(1:string id, 2:i64 time);
5. }

由于实现双向通信是在TCP通信层的，所以会导致所有thrift层的service都会有影响，终端程序所有的连接必须都配套一个ConnectService服务。

目前设计只有ConnectService服务才需要实现双向调用的通信。终端每次连接到服务器端，服务器就会主动调用token\_id方法发送给终端一个表明通信连接唯一的md5值用于记录本次通信连接的key，终端收到这个key值后，如果该连接是需要实现双向通信的连接就立马回调服务器端，服务器端就对该连接进行保存用于双向通信。如果是不需要双向通信的连接就实现一个ConnectService的token\_id空方法。

# 功能点流程说明

## 终端程序发起

### 终端用户认证、修改密码

终端调用服务器只需要关键信息（用户名称、用户密码），服务器端返回认证结果。

对应接口：user\_login、user\_modify\_passwd

注意：如果收到同一用户请求同时登录不同终端，验证第二个终端请求的用户认证，

认证成功后，先发送前一个终端的“注销通知”，发送成功后才给第二个终端用户认证的响应。如果不先发送“注销通知”，那么会出现一个用户同时登录两个终端的情况。

### 云桌面相关接口

接口有：

**终端获取关联的资源枚举列表（get\_dskgrop\_info）**

就是获取桌面组信息，教学场景上传MAC地址，用户场景上传用户信息

**终端登录云桌面（desktop\_open）**

**一**个用户只能同时在一个终端上登录

同一(终端序号+终端IP）只能有一个终端占用同一个桌面，如果有同一(终端序号+终端IP）请求，则需要返回终端信息终端被其他终端占用

**终端退出云桌面（desktop\_close）**

**获取所有终端和云桌面关联列表信息（get\_desktop\_info）**

只有电子教学软件用,只取教学桌面, 电子教学软件那边会每隔3秒取一次数据(终端序号、Mac、IP,云桌面Ip)

### 终端配置信息更新

对应接口：update\_config

终端调用服务器更新配置的接口、注意对配置版本号加一操作，否则配置会不统一

服务器端不去判断版本号，直接以终端上报数据为准来保存

默认情况下：终端开机就会立即发起终端信息同步（比对终端配置版本号，决定是否更新本地或者是服务器的终端配置），

i）、如果终端配置版本要高，则直接发起该请求上报更新配置

ii)、如果终端配置版本要低，则先获取服务器端的终端配置信息，再更新本地配置

iii)、如果终端配置版本和服务器端的配置是一致的，则无需进行更新操作

### 终端获取配置信息

对应接口：get\_config

终端上报mac地址，直接获取对应的配置数据信息

### 终端获取配置版本号

对应接口：get\_config\_version

终端上报mac地址，直接获取对应的配置版本（新终端无配置情况，返回-1）

### 终端排序查询

对应接口：order\_query

终端使用mac地址请求查询是否该终端处于排序中，服务器端返回一个int值，如果是返回-1则表示非排序中，大于等于0的表示目前正在排序的值。

终端收到返回值后：可以根据终端目前应该排序的号码，手动进行修改需要排序号码确认后回调Command的命令（命令名称为order\_confirm）通知服务器该号码得到排序、

并且最后还是由终端调用服务器的更新终端配置接口来更新终端序号的配置

### 终端获取服务器时间

对应接口：GetDateTime

此接口用于终端系统的时间更新为服务器时间，保证时间的一致

## 服务器端发起

### 服务器发送终端控制命令

命令列表：终端关机、终端重启、更新配置、升级终端程序、上传日志、终端IP更新、终端序号排序（需要回调服务器）、云桌面关闭通知、用户注销通知

1. **enum** CommandBodyType {
2. JSON = 1;
3. XML = 2;
4. TEXT = 3;
5. }
7. /\*\*
8. \*服务器发送给客户端的命令
9. \*/
10. **struct**  CommandMsg{
11. 1: i64 Num = 0
12. 2: string Body
13. 3: CommandBodyType BodyType
14. 4: string cmdstr
15. 5: string Version
16. 6: binary BodyBytes
17. 7: i64 TotalPackets
18. 8: string From//由谁发出
19. 9: string Tos//发送给谁
20. 10: **bool** IsNeedConfirm
21. 11: map<string,string> ArgsDic
22. 12: i64 batch\_num = 0
23. }

命令列表：命令名称不区分大小写

1. 终端关机(shutdown)
2. 终端重启(restart)
3. 更新配置(update\_config)
4. 升级终端程序(update\_soft)：

命令参数ArgsDic: “file\_name”= “x86\_windows\_2.2.2.0.zip”

1. 上传日志(upload\_log):

命令参数ArgsDic:

“start\_time”= “2015-05-01”

“end\_time”= “2015-05-02”

1. 桌面关闭推送(vm\_shutdown):

参数类型josn(utf-8编码), 推送list<DesktopInfo>数据

1. 终端IP更新(update\_ip)：

参数类型josn(utf-8编码), 推送IPInfo数据

1. 排序(order):

命令批次号：“batch\_num”= 11

命令参数ArgsDic：“terminal\_id”= “10”

当“terminal\_id”= “-1”的时候表示排序取消

1. 用户注销通知(user\_logout)

根据终端程序需求修改：发送给终端的命令格式如下

BodyType：固定为CommandBodyType.TEXT

Body:固定为 “Command:{cmd}|{json\_body}”

### 终端程序升级全流程(不考虑离线升级)

* 1. 服务器端直接发送更新指令(不对终端配置信息修改)
  2. 终端收到服务器调用的升级终端程序命令
  3. 终端强制升级、使用相应thrift接口来下载升级程序包文件
  4. 本地更新终端程序、重启终端

文件限制命名格式：终端CPU架构\_操作系统类型\_版本号.文件类型后缀，例如：” Windows端：x86\_windows\_2.2.2.2.zip

Linux端：x86\_linux\_2.2.3.zip

ARM端：arm\_linux\_2.2.3.zip

### 终端日志文件上传全流程

1、终端收到服务器调用的上传日志命令

2、终端使用相应thrift接口调用服务器来上传日志文件

文件限制命名格式： MAC \_YYYY-MM-DD.zip，例如：” 00-50-56-C0-00-08\_2030-12-21.zip”

### 终端配置更新

用于管理平台直接修改终端的配置信息（不包括：终端序号、终端IP），修改终端配置信后，把对应配置版本号加一，修改服务器端数据库记录，最后发送更新配置(update\_config)命令给终端更新配置。

### 终端序号排序

用于教学场景的教室的终端的序号重新更新： 服务器端对选定的所有终端群发“终端序号排序”的命令，并且带有发送的序号和命令的批次号，收到一个排序确认(ManageService.command\_confirm: cmdstr、batch\_num、terminal\_id填写和服务器发送的order参数一致)后，再把序号加一，群发没有回调的所有终端，重复操作直到所有终端都有回调确认。终端每次发起

排序取消：如果管理平台用户发起“终止排序”，则还有排序完终端都会收到order命令，但是排序号码是-1，表示取消排序，终端就会退出排序确认界面

异常情况：如果终端坏了的异常情况，就发起“终端排序查询”的接口来进行手动修改需要的终端序号确定后调用“终端配置信息更新”

### 终端IP更新

用于教学场景的整个教室的终端的IP重新更新

1、服务器端根据给定的起始IP地址和教室所有终端序号都按照从小到大的排序，在同一个排序序号的就形成对应关系来分配终端的IP地址。再对教室所有终端都发送更新IP的命令（命令参数就是IP地址）

2、终端收到更新IP命令后，就立即更新本地IP地址以及IP的相关配置（子网掩码、网关、DNS1、DNS2），然后就直接关机

3、运维人员根据终端是否关机来判断终端是否更新IP地址成功

### 云桌面关闭通知

当对应终端关联的云桌面关闭后，服务器端主动推送关闭状态，用于终端判断云桌面状态，应用于“关闭云桌面时关闭终端”的功能

### 用户注销通知

服务器端收到同一个用户验证登录请求，服务器端将对前一个登录进来的终端发送“用户注销通知”。

前一个登录的终端需要注销该用户的登录，并且终端需要主动断开已经连接的云桌面。

## 管理平台自有功能

### 终端删除

只有终端不在线的才可以进行删除操作

服务器端直接把终端配置信息数据表记录删除、终端桌面信息数据表记录也删除

应用场景：终端设备坏了，直接使用新的设置替换，新终端的终端序号、终端IP配置成坏了的设备一致，重启终端后，直接就可以连接上以前的云桌面

# 功能实现的thrift协议接口

实现服务有：ConnectService(双向通信)、ManageService、FileService ，接口文件内容如下

1. /\*\*
2. \* The first thing to know about are types. The available types in Thrift are:
3. \*
4. \*  bool    Boolean, one byte
5. \*  i8 (byte)   Signed 8-bit integer
6. \*  i16     Signed 16-bit integer
7. \*  i32     Signed 32-bit integer
8. \*  i64     Signed 64-bit integer
9. \*  double      64-bit floating point value
10. \*  string      String
11. \*  binary      Blob (byte array)
12. \*  map<t1,t2>  Map from one type to another
13. \*  list<t1>    Ordered list of one type
14. \*  set<t1>     Set of unique elements of one type
15. \*
16. \* Did you also notice that Thrift supports C style comments?
17. \*
18. \* version V2.0
19. \*/
21. **namespace** cpp Thrift
23. **struct** UserInfo {
24. 1:string user\_name;
25. 2:string user\_passwd;
26. }
28. /\*\*
29. \*IP信息
30. \*/
31. **struct** IPInfo{
32. 1: i32 IsDhcp
33. 2: string Ip
34. 3: string Subnet
35. 4: string Gateway
36. 5: string Mac
37. 6: string DNS1
38. 7: string DNS2
39. }
41. **struct** DesktopGroupInfo {
42. 1:string group\_name; // 桌面组名称
43. 2:i32    group\_id; // 桌面组序号
44. 3:string group\_desc; // 桌面组的描述
45. 4:string group\_uuid; // 桌面组UUID
46. 5:string os\_type; // windows\_7\_x64/windows\_7/windows\_10\_x64/Other
47. }
49. **struct** DesktopInfo {
50. 1:string ip;
51. 2:i32 port;
52. 3:string desktop\_name;
53. 4:string token; // 服务器开启虚拟机动态生成的md5值
54. 5:string dsk\_type; // 是指基于思捷、kvm等不同底层虚拟的桌面
55. 6:DesktopGroupInfo group;
56. 7:i32 status; // 0-关闭 1-运行
57. 8:string os\_type; // 桌面操作系统类型
58. 9:UserInfo dsk\_user; // 桌面操作系统用户名和密码
59. 10:string dsk\_uuid; // 桌面的唯一ID
60. }
62. **struct** RespDesktopInfo {
63. 1:string code; // "0000" 表示正常返回, "0001" 表示被其他终端占用
64. 2:string msg;
65. 3:string description;
66. 4:DesktopInfo dsk\_info;
67. }
69. // 教学桌面专用
70. **struct** TerminalDesktopInfo {
71. 1:i32 terminal\_id;
72. 2:string terminal\_mac;
73. 3:string terminal\_ip;
74. 4:string desktop\_ip;
75. }
77. /\*\*
78. \*服务器配置信息
80. \*/
81. **struct** YunServerInfo{
82. 1: string ServerUrl
83. 2: string ServerUrl1
84. 3: string UserName
85. 4: string UserPass
86. 5: string Domain
87. 6: i64 TimeOut
88. }
90. /\*\*
91. \*屏幕分辨率
92. \*/
93. **struct** ScreenInfo{
94. 1: i32 Width
95. 2: i32 Height
96. }
98. /\*\*
99. \*硬件信息
100. \*/
101. **struct** HardwareInfo{
102. 1: string CpuID
103. 2: string HardDiskID
104. 3: string MacAddress
105. 4: string YunId
106. }
108. /\*\*
109. \*教室信息
110. \*/
111. **struct** RoomInfo{
112. 1: string RoomId
113. 2: string Name
114. 3: string Des
115. 4: **double** OrderId
116. 5: i32 SeatNum
117. 6: string Group
118. }
120. /\*\*
121. \*RDP远程桌面信息
122. \*/
123. **struct** RemoteDesktopInfo{
124. 1:string MachineName
125. 2:string Domain
126. 3:string Uername
127. 4:string Pwd
128. 5:string DesktopName,
129. 6:string SoftwareInfo
130. 7:string Domain0
131. 8:string Uername0
132. 9:string Pwd0
133. }
135. /\*
136. \*服务器信息
137. \*/
138. **struct** ServiceInfo
139. {
140. 1:string Ip
141. 2:i32 Port
142. }
144. /\*
145. \*屏幕广播参数
146. \*/
147. **struct** ScreenBroadCastInfo
148. {
149. 1:string CastServicename
150. 2:string ScrCastChannel
151. 3:string ScrCastIp
152. 4:i32 ScrCastPort
153. 5:i32 ScrCastQuality
154. 6:i32 ScrCastRecvWinStyle
155. 7:i32 ScrCastAllowRecord
156. 8:i32 ScrCastSvrScreenRecord
157. }
159. /\*
160. \*版本信息
161. \*/
162. **struct** VersionInfo
163. {
164. 1:string ConfVersion
165. 2:string Version
166. }
168. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
169. \*多系统切换 add by rock
170. \*/
171. **struct** SystemTableInfo {
172. 1:string DesktopId; ##桌面场景guid
173. 2:string SystemTabUid; ##系统切换的桌面UID
174. 3:string TabDeskName;##桌面名称
175. 4:i32 NeedNums; ##资源申请数量
176. 5:i32 ReadyNums; ##资源已就绪数量
177. 6:i32 LogonNums; ##已登录云桌面数量
178. 7:**bool** IsNeedSign; ##是否需点名
179. 8:**bool** AutoLogin; ##是否自动登录云桌面
180. 9:string TabDateTime; ##系统切换开始时间
181. 10:i64 AutoLoginDelaySecond; ##自动登录延时时间(从开始时间计)
182. 11:**bool** TabIsReady;##指定是否已分配已准备好的桌面给到当前客户端。非自动登录云桌面时生效
183. 12:string DeskDescription;//
184. 13:i32 RegisterCount;
185. 14:i32 UnRegisterCount;
186. 15:i32 OpenCount;//开机数量
187. 16:i32 CloseCount;//关机数量
188. 17:string Status;//桌面状态(1 开启 2 关闭 9 正在开机 10 正在关机)
189. 18:string StatusName;//状态名称
190. 19:**bool** Checked;
191. 20:**bool** IsClassing;
192. 21:**bool** IsReady;
193. }
195. **struct** EduClientInfo
196. {//教学客户端配置 暂未使用
197. 1: string RoomName
198. 2: string Group
199. 3: string ProtectPass
200. 4:ServiceInfo EduServiceInfo      //教学服务
201. 5:ServiceInfo EduTopServiceInfo       //教学顶层服务
202. 6:ServiceInfo ClientServiceInfo       //终端管理服务
203. 7:string SlientImagePath          //静默图片
204. 8:string FileReceivePath          //文件接收路径
205. 9:i32 DisEnableNet           //网络限制
206. 10:i32 Slient            //静默中
207. 11:i32 Casting               //广播中
208. 12:ScreenBroadCastInfo BroadCastInfo  //广播配置
209. 13:VersionInfo EduVersionInfo     //教学配置版本
210. 14:i32 UserType //3-教师终端-2-教师云桌面端 1-学生终端 0-学生云桌面端
211. 15:i32 PlatForm //0-x86  1-ARM
212. 16:i32 DemoModing               //演示中 add by rock
213. 17:i32 CutClassing              //切课中 add by rock
214. 18:SystemTableInfo SysTableInfo     //切换系统的状态
215. 19:i32 Montioring
216. }
218. /\*\*
219. \*座位信息
220. \*/
221. **struct** SitInfo {
222. 1:i32 Row
223. 2:i32 Col
224. 3:**double** RowId //row.(col\*0.01)  如5.03表示第5行第3列（最多到第99列）
225. 4:string Des
226. 5:string UserUuid
227. 6:string RoomId
228. }

231. /\*\*
232. \*客户端配置信息
233. \*/
234. **struct** ClientInfo {
235. 1: string Num//终端编号
236. 2: string HostName
237. 3: IPInfo IPinfo    //IP信息对应TerminalConf.IPinfo
238. 4: YunServerInfo Yuninfo
239. 6: ScreenInfo Screeninfo  //设置终端正在使用的屏幕信息
240. 7: i32 ShowLocalDesktop
241. 8: string  AutoOpen
242. 9: string  ConfigPass
243. 10: string MachineCode
244. 11: string ActivateCode
245. 12: string Activatestate
246. 13: string Ewfstate
247. 14: string version;
248. 15: string ConfVersion
249. 16: i64 ConfTime
250. 17: i64 OverTime
251. 18: HardwareInfo HardInfo
252. 19: i32 ShowLoginPage
253. 20: i32 ShowDesktopBootForm
254. 21: i32 ShowDesktopDesc
255. 22: RoomInfo roomoInfo
256. 23: string LocalPass
257. 24: i32 IdleTimeOut
258. 25: i32 CloseWhileExitYunDesk
259. 26: RemoteDesktopInfo remoteDesktopInfo
260. 27: i32 usertype //3-教师终端-2-教师云桌面端 1-学生终端 0-学生云桌面端
261. 28: i32 PlatForm //0-x86  1-ARM
262. 29: EduClientInfo EduInfo //教学客户端配置
263. 30: SitInfo ClientSitInfo
264. 31: TerminalConf TerminalConfInfo
265. }
267. /\*\*
268. \*KVM 终端配置
269. \*/
271. // 与服务器断开链接退回本地桌面设置
272. **struct** DisconnectSetup {
273. // -1表示不启用 >=0 表示启用，并且表示响应多少秒进入本地桌面
274. 1:i32 goto\_local\_desktop;
275. // true-进入本地桌面需要输入密码
276. 2:**bool** goto\_local\_auth;
277. }
279. **struct** DisplaySetup {
280. 1:**bool** show\_local\_button; // 显示本地桌面按钮 true-显示 false-隐藏
281. 2:string goto\_local\_passwd; // 进入本地桌面密码(注意:非本地操作系统密码)
282. }
284. **struct** TerminalConf {
285. 1:i32 terminal\_id;//终端编号
286. 2:string mac;//物理地址
287. 3:IPInfo ip\_info;// IP信息
288. 4:string terminal\_name;//终端名称
289. 5:string platform; // ARM x86
290. 6:string soft\_version;//终端程序版本
291. 7:i32 show\_desktop\_type; // 0-教学桌面、1-个人桌面、2-混合桌面
292. 8:i32 auto\_desktop; // 0-不自动进入、>1 表示进入第几个桌面
293. 9:**bool** close\_desktop\_strategy; // true-关闭桌面同时关闭终端
294. 10:**bool** close\_terminal\_strategy; // true-关闭终端同时关闭桌面
295. 11:**bool** open\_strategy; // true-通电启动终端
296. 12:ServiceInfo server\_info;//终端管理服务IP地址和端口
297. 13:list<ScreenInfo> screen\_info\_list; // 终端所有可用分辨率
298. 14:ScreenInfo current\_screen\_info;//设置终端正在使用的屏幕信息
299. 15:**bool** show\_modify\_user\_passwd; // true-显示修改密码(终端用户密码)
300. 16:string terminal\_setup\_passwd; // 终端设置项密码
301. 17:i64 conf\_version; // 用于对终端配置信息的服务器端和终端进行同步更新
302. 18:i32 window\_mode; // 1-全屏， 2-全屏可退出 3-全屏不可退出
303. 19:DisconnectSetup disconnect\_setup;//断开服务器连接设置
304. 20:DisplaySetup show;//是否显示本地桌面按钮图标
305. }
307. // 文件类型决定了文件上传或者下载对应服务器上面存储的路径的配置
308. **enum** FileType {
309. LOG = 1;
310. SOFT = 2;
311. PATCH = 3;
312. }
314. **struct** FileCtrlInfo {
315. 1:string file\_name;
316. 2:FileType file\_type;
317. 3:i64 total\_size;
318. 4:i64 operate\_offset;
319. 5:i32 operate\_length;
320. }
322. // 路由信息
323. **struct** RouteInfo {
324. 1: string RouteId
325. 2: string RouteAddress
326. 3: IPInfo IPinfo
327. }
329. /\*\*
330. \*结果信息
331. \*/
332. **struct** ResultInfo {
333. 1:string code
334. 2:string Message
335. 3:string Description
336. }
338. /\*\*
339. \*云桌面运行软件信息
340. \*/
341. **struct** SoftInfo{
342. 1:string ProcessName
343. 2:string ClassName
344. 3:string DefaultTitle
345. 4:string ProcessPath
346. 5:string Username
347. 6:binary ImgData
348. }
350. /\*\*
351. \* 云客户端显示信息
352. \*/
353. **struct** YunClientShowInfo{
354. 1:string ID,
355. 2:string ClientNum,
356. 3:i32 OnlineState,
357. 4:string ComputerName
358. 5:string ClientIPAddress
359. 6:string ClientMAC
360. 7:string LoginUser
361. 8:string LoginDomain
362. 9:string SafeState
363. 10:string ActivateCode
364. 11:**double** OrderID
365. 12:string Version
366. 13:string Note
367. 14:**bool** Checked
368. 15:string RoomID
369. 16:string Group
370. 17:string RemoteScreenConnectString
371. 18: i32 PlatForm //0-x86  1-ARM
372. 19:string PlatFormDesc //平台说明
373. 20:string ActivateState//激活状态
374. }
376. **enum** CommandBodyType {
377. JSON = 1;
378. XML = 2;
379. TEXT = 3;
380. }
382. /\*\*
383. \*服务器发送给客户端的命令
384. \*/
385. **struct**  CommandMsg {
386. 1: i64 Num = 0
387. 2: string Body
388. 3: CommandBodyType BodyType
389. 4: string cmdstr
390. 5: string Version
391. 6: binary BodyBytes
392. 7: i64 TotalPackets
393. 8: string From//由谁发出
394. 9: string Tos//发送给谁
395. 10: **bool** IsNeedConfirm
396. 11: map<string,string> ArgsDic
397. 12: i64 batch\_num = 0
398. }

401. // 实现双向通信，服务器和终端都需实现
402. service ConnectService {
403. // 支持命令：终端关机、终端重启、更新配置、升级终端程序、上传日志、桌面关闭通知、更新IP、终端排序
404. oneway **void** Command(1:CommandMsg msg);
405. // 服务器调用此方法时，如果终端是需要双向通信的连接必须有回调此方法
406. oneway **void** TokenId(1:string tokenId);
407. oneway **void** RouteAddress(1: string routeAddress);
408. oneway **void** Ping(1:string tokenId, 2:i64 Time); // 预留用于连接检测或者心跳
409. oneway **void** RouteAddressInfo(1:string tokenId, 2:RouteInfo routeInfo);
410. }
412. // 单向调用服务，用于管理终端
413. service ManageService {
414. //思杰
415. // 终端登录,终端MAC和通信链路建立绑定关系
416. **bool** ClientLogin(1:string tokenId,2:HardwareInfo info);//登录方法 判断是否激活
417. string GetServerConfVer(1:string tokenId);
418. **bool** UpdateInfo(1:string tokenId,2:ClientInfo info);
419. ClientInfo GetClientInfo(1:string tokenId);
420. oneway **void** UpdateCopyDiskProgress(1:string tokenId,2:string message);
421. string GetClientNum();
422. string GetClientUser();
423. string GetClientIp();
424. string GetDateTime();
425. **bool** ExistClientNum(1:string clientNum);
426. ClientInfo ChangeClient(1:ClientInfo clientInfo);
427. **bool** IsFirstLoginUpdatePwd(1:string username,2:string pwd);
428. **bool** UpdateUserPwd(1:string username,2:string oldPwd,3:string pwd);
429. ResultInfo OpenSoft(1:SoftInfo softInfo);
430. ResultInfo AddSoftInfoList(1:list<SoftInfo> listSoft);
431. list<SoftInfo> GetSoftInfoList();
432. list<YunClientShowInfo> GetList(1:i32 pageIndex,2:i32 pageSize,3:string keyword,4:string roomId);
433. /\*\*获取用户是否可以修改密码\*/
434. **bool** CanModifyPassword(1:string userName);
435. /\*\*验证用户密码是否正确\*/
436. **bool** ValidateUserPassword(1:string userName,2:string password);
438. //kvm
439. // 终端用户登录认证
440. ResultInfo user\_login(1:UserInfo user, 2:string mac);
441. // 终端用户登录注销
442. ResultInfo user\_logout(1:string user\_session\_id);
443. // 终端用户修改密码
444. ResultInfo user\_modify\_passwd(1:UserInfo old\_user, 2:UserInfo new\_user);
445. // 终端获取资源枚举列表信息(桌面组信息;教学桌面填写mac，个人桌面填写user\_session\_id)
446. list<DesktopGroupInfo> get\_dskgrop\_info(1:string mac, 2:string user\_session\_id);
447. // 获取所有终端和云桌面关联列表信息(电子教学软件用,只取教学桌面, 电子教学软件那边会每隔3秒取一次数据)
448. list<TerminalDesktopInfo> get\_desktop\_info();
449. // 终端请求打开云桌面系统
450. RespDesktopInfo desktop\_open(1:DesktopGroupInfo info, 2:string mac, 3:string user\_session\_id);
451. // 终端请求关闭单个云桌面
452. ResultInfo desktop\_close(1:DesktopInfo info);
453. // 终端请求关闭关联所有的云桌面(教学场景:只需要传参mac，user\_session\_id为空，个人场景: 两个参数都需要传)
454. ResultInfo all\_desktop\_close(1:string mac, 2:string user\_session\_id);
455. // 终端获取配置版本号, 如果没有终端配置，返回 -1
456. i64 get\_config\_version(1:string mac);
457. // 终端获取配置信息
458. TerminalConf get\_config(1:string mac);
459. // 终端请求更新配置信息
460. ResultInfo update\_config(1:ClientInfo conf);
461. // 终端排序查询
462. CommandMsg order\_query(1:string mac); // 返回值：terminal\_id -1表示非排序中 >=0 正在排序的号码
463. // 终端响应服务端发起的命令的确认
464. **bool** command\_confirm(1:CommandMsg cmd\_msg, 2:string mac);
465. }
467. // 单向调用服务，用于文件下载上传
468. service FileService {
469. //思杰
470. i64 GetFileSize(1:string fileName);
471. binary ReadByte(1:string fileName,2:i64 offset,3:i32 length);
473. //kvm
474. // 获取文件大小
475. i64 get\_file\_size(1:string file\_name, 2:FileType file\_type);
476. // 下载文件
477. binary read\_bytes(1:FileCtrlInfo read\_info);
478. // 上传文件
479. **bool** write\_bytes(1:FileCtrlInfo write\_info, 2:binary data);
480. }

# 问题

终端多网卡情况，mac地址不是唯一信息了

网络拓扑中架设路由器，可能导致终端ip重复情况，进而教学分组出现冲突（目前认为不会出现此情况、人为控制）