# 网络对讲机系统协议设计

王小敏

2013-11-23

## 系统结构图

用户2

用户n

…

用户4

用户3

用户1

如图所示,是一个以一个平台为中心的星形结构,所有用户都和中心平台进行连接.平台是一个无任何状态和数据记录的轻型服务器,方便部署.

## 通信流程

### 控制通道流程

平台

1.登录请求

用户

2.登录响应

3.数据通信(含保活)

5.关闭通道

1. 用户发起登录请求
2. 然后保持

### 数据通道流程

#### 基本流程

1. 在控制通道建立的时候,就分配一个数据通道的Token.
2. 获取联系人列表的时候,就获取好每个联系人的Token.
3. 用户进入通话状态后,使用UDP发送数据(根据网络决定是否重传,一般只做一次重传)
4. 无语音发送时,定时发送保活报文.

#### 单呼流程

1. 主叫方选择一个联系人发起呼叫请求.
2. 被叫方收到请求后,选择接受或者拒绝请求.
3. 连接建立后,直接使用UDP双向发送语音数据.
4. 任何一方都可以发挂断通话请求

#### 集群呼叫流程

1. 集群中的任何一人都可以发起集群呼叫.发起时,先向集群中的所有用户发邀请.
2. 其他用户收到邀请后选择接受或拒绝邀请.如果拒绝要求,以后也可以通过选择进入集群来参与通话.
3. 如果进入了集群通话,则向平台申请集群的Token.申请Token之后,就可以使用这个Token发送数据了.
4. 集群通话的时候,说话之前,必须按下按钮才能说话,按下按钮就开始申请Token,按钮按下去之后,Token成功之后,才能说话.(跟电脑的语音聊天室一样,只是在电脑上的意义远不如在手机上)

## 通信通信协议

### 控制通道协议

1. 控制通道使用TCP协议进行交互,平台作为TCP侦听端,用户作为TCP客户端.
2. 通信报文的格式采用类似HTTP头加XML的格式:

|  |
| --- |
| Version: 协议版本,目前为1.0  Verb: 动作类型,取值见后面定义.  Type: 报文的类型,取值和说明如下:  Request 请求报文  Response 响应报文  Notify 通知报文  From: 报文的来源UserID,和平台通信的报文,无此字段  To: 报文的目标UserID,和平台通信的报文,无此字段  Result: 响应报文的结果,响应报文才有这个字段.取值和含义参考附录中的定义.  Length: 后面的XML报文的长度.  <XML /> |

一个实例如下:

根据Verb和Type的不同,后面的XML报文格式也是不一样的,下面列出支持的Verb和Type及其XML报文的格式.描述模版如下:

1. 请求和响应模版

|  |  |  |
| --- | --- | --- |
| Verb |  | |
| 请求 | <Request>  </Request> | |
|  |  |
|  |  |
| 响应 | <Response>  </Response> | |
|  |  |
|  |  |
| 说明 |  | |

1. 通知模版

|  |  |  |
| --- | --- | --- |
| Verb |  | |
| 通知 | <Notify>  </Notify> | |
|  |  |
|  |  |
|  |  | |

#### 用户管理

##### "登录"请求和响应(用户<=>平台)

|  |  |  |
| --- | --- | --- |
| Verb | Register | |
| 请求 | <Request>  <User ID="" Path="" Client="" Network="" />  </Request> | |
| ID | 用户的名称,唯一标识这个联系人 |
| Path | 用户的分组路径,通过"/"进行分割,支持多级路径. |
| Client | 使用的客户端软件,目前取值如下:  Windows Windows客户端  Android Android客户端  iOS iOS客户端 |
| Network | 登录使用的网络,目前取值如下:  Line 有线网络  WiFi WiFi网络  Mobile 移动网络 |
| 响应 | <Response Token="">  <UserList>  <User ID="" Path="" Client="" Network="" Status="" />  ...  </UserList>  </Response> | |
| Token | 发送数据用的令牌,格式为一个0x打头的16进制字符串,转换成二进制后是4个字节. |
| UserList | 在线联系人列表信息,和请求中一样的就不说明了. |
| Status | 联系人的状态,目前取值如下:  Offline 离线  Online 在线,并且正常  Absence 离开  Busy 忙碌 |

##### "联系人信息改变"通知(平台==>用户)

|  |  |  |
| --- | --- | --- |
| Verb | UserChanged | |
| 通知 | <Notify>  <User ID="" Path="" Client="" Network="" Status="" />  </Notify> | |
| User | 改变后的用户信息,含义参考"登录请求和响应" |

#### 单点呼叫控制

##### "发起单点呼叫"请求和响应(用户<=>用户)

|  |  |  |
| --- | --- | --- |
| Verb | StartCall | |
| 请求 | <Request>  <Call ID="" Token="" Timeout="" />  </Request> | |
| ID | 标识这个呼叫的唯一ID,由发起方递增生成. |
| Token | 发起方登录时或得的Token,被叫方用这个Token发送数据. |
| Timeout | 呼叫的超时时间,单位秒 |
| 响应 | <Response /> | |

##### "单点呼叫被接受"通知(用户==>用户)

|  |  |  |
| --- | --- | --- |
| Verb | CallAccepted | |
| 通知 | <Request>  <Call ID="" />  </Request> | |
| ID | 同"发起单点呼叫"请求 |

##### "单点呼叫被拒绝"通知(用户==>用户)

|  |  |  |
| --- | --- | --- |
| Verb | CallRefused | |
| 通知 | <Request>  <Call ID="" />  </Request> | |
| ID | 同"发起单点呼叫"请求 |

##### "单点呼叫结束"通知(用户==>平台)

|  |  |  |
| --- | --- | --- |
| Verb | CallTerminated | |
| 通知 | <Request>  <Call Token="" Timeout="" />  </Request> | |
| ID | 标识这个呼叫的唯一ID,由发起方递增生成. |
| Token | 发起方登录时或得的Token,被叫方用这个Token发送数据. |
| Timeout | 呼叫的超时时间,单位秒 |
| 说明 | 呼叫没有建立成功的话,需要发送这个报文通知平台.  通话过程中,双方都可以发送这个通知来挂断通话. | |

#### 集群呼叫控制

##### "申请集群呼叫Token"请求和响应(用户<=>平台)

|  |  |  |
| --- | --- | --- |
| Verb | ApplyClusterCall | |
| 请求 | <Request /> | |
| 响应 | <Response Token="" /> | |
| Token | 申请到的集群呼叫的Token,格式和"登录"响应报文中的Token一样. |

##### "发起集群呼叫"请求和响应(用户<=>用户)

|  |  |  |
| --- | --- | --- |
| Verb | StartClusterCall | |
| 请求 | <Request>  <Call ID="" Token="" Timeout="" />  </Request> | |
| ID | 标识这个呼叫的唯一ID,由发起方递增生成. |
| Token | 发起方登录时或得的Token,被叫方用这个Token发送数据. |
| Timeout | 呼叫的超时时间,单位秒 |
| 响应 | <Response /> | |

##### "集群呼叫被接受"通知(用户==>用户)

|  |  |  |
| --- | --- | --- |
| Verb | ClusterCallAccepted | |
| 通知 | <Request>  <Call ID="" />  </Request> | |
| ID | 同"发起单点呼叫"请求 |

##### "集群呼叫被拒绝"通知(用户==>用户)

|  |  |  |
| --- | --- | --- |
| Verb | ClusterCallRefused | |
| 通知 | <Request>  <Call ID="" />  </Request> | |
| ID | 同"发起单点呼叫"请求 |

##### "集群呼叫结束"通知(用户==>平台)

|  |  |  |
| --- | --- | --- |
| Verb | ClusterCallTerminated | |
| 通知 | <Request>  <Call Token="" Timeout="" />  </Request> | |
| ID | 标识这个呼叫的唯一ID,由发起方递增生成. |
| Token | 发起方登录时或得的Token,被叫方用这个Token发送数据. |
| Timeout | 呼叫的超时时间,单位秒 |
| 说明 | 呼叫没有建立成功的话,需要发送这个报文通知平台.  通话过程中,双方都可以发送这个通知来挂断通话. | |

##### ApplyForSpeak

##### 草稿

1. **申请Token**,(并且带上集群中的用户列表)
2. 发送集群呼叫请求给其他用户(带上集群中的用户列表),具体的流程
   1. 用户发**集群通知给平台**
   2. 平台再**发集群呼叫通知给集群中的其他用户.**
3. 列表中的用户有三种状态
   1. 等待(发起的时候有用)
   2. 加入
   3. 掉线
   4. 拒绝
   5. 踢出
   6. 退出
4. 如果用户接受呼叫,
   1. 先向平台发送**接受的通知**.然后开始发数据(可能收到集群不存在的反馈).
   2. **平台发**通知给集群中的其他用户.
5. 如果用户拒绝呼叫
   1. 向平台发送**拒绝通知**
   2. **平台发**通知给集群中的其他用户.
6. 邀请其他人参加集群
   1. 先发**通知**给平台.
   2. **平台发通知给**被邀请的人(带上集群中的所有用户列表).
   3. **平台发通知**给集群中的其他用户(包括要求发起人)
   4. 被邀请人的接受和拒绝操作和刚刚开始的时候一样.
7. 踢出集群中的用户
   1. 这个只能由发起人干.
   2. 直接**发剔出通知**给平台就行了.
   3. 用户被踢出后,就不收这个用户的数据了.(每个用户需要一个集群的Token)
8. 如果其用户掉线了(包括发起人)
   1. 集群中的状态变成掉线,并且会**发掉线通知**给集群中的其他用户.
   2. 重新连接成功之后,只要Token还在,还可以继续加入集群呼叫.
      1. 先发**恢复通知**给平台.
      2. 然后开始发送数据(可能收到集群不存在的反馈).
9. 如果其用户要退出集群
   1. 给平台发**退出通知**就行了.
   2. 退出之后,只要集群还在,随时可以再加入进来.只要重新发**重新进入通知**就行了.
10. 如果平台中的用户都不在了
    1. 如果都退出(或拒绝,或踢出)了,则直接释放集群.
    2. 如果都在等待(或掉线),则等待一个超时时间再释放