## 1、范围

本文为脉恩空调系统协议规范

## 2、心跳路由客户端和心跳服务器

#### 2.1、简要

心跳路由客户端与心跳服务器之间新增两条指令，分别为report与issue。report为设备上报数据到心跳服务器，issue为心跳服务器下发数据到心跳路由客户端。

#### 2.2、报文概述

###### REPORT指令（设备侧上报数据）

1.请求指令   
vendor 4字节   
options TLV格式   
data 不固定

2.响应指令   
SN 2字节，请求指令序列号

###### ISSUE 指令 （业务服务器请求设备侧）

1.请求指令   
vendor 4字节   
options TLV格式   
data 不固定

2.响应指令 与通知消息格式一致、消息类型不一致   
SN 2字节，请求指令序列号   
reponseCode 2字节

#### 2.3、report与issue报文格式

+-----+----------+-----------+---------+-----+-----+--------+---------+----------+

| flag | pktlen | version | pktTye | sn | Ext | vendor | option | data |

+-----+----------+-----------+---------+-----+-----+--------+---------+----------+

| <--------------- 包头消息 ------------------------>|<-- 消息字段 -->|

pktTye：消息字段类型，新增两条消息在此基础上扩展，分别为report与issue.   
pktlen : 消息长度，从vendor头开始到data尾部整个长度。   
option : 建议保留，上报和下发将来一定是经常用到消息，可用作为消息字段的扩展。

#### 2.4、report回应报文格式

+------+--------+---------+--------+----+-----+---------------+

| flag | pktlen | version | pktTye | sn | Ext | sn |

+------+--------+---------+--------+----+-----+---------------+

| <--------------- 包头消息 --------------->|<-- 消息字段 -->|

#### 2.5、issue回应报文格式

+------+--------+---------+--------+----+-----+----+--------------+

| flag | pktlen | version | pktTye | sn | Ext | sn | reponseCode |

+------+--------+---------+--------+----+-----+----+--------------+

| <--------------- 包头消息 --------------->|<---- 消息字段 ---->|

#### 2.6、实现建议：

1. report（issue）、datalen、vendor、option属于心跳协议的一部分，由心跳服务器与客户端来负责解析。
2. data段由厂商定义格式，可选择是否加密，心跳服务器与客户端对此段进行透明传输。
3. CPE侧，通过分析vendor字段，可将data数据会分发到MYE程序，由MYE程序进行解析。
4. 后台，心跳服务器将data数据放入消息队列后，由专门程序取出，分析出vendor字段并投递到相应程序中进行处理。

## 3、MYE、TR069客户端、心跳路由客户端之间IPC通信

#### 3.1、简要

* 这三者之间的ipc通信，通过使用UDP协议，json数据格式，restful风格，方便测试、阅读、开发与维护。
* MYE监听端口10300，心跳路由客户端监听端口10400，TR069客户端监听端口10500。

#### 3.2、心跳路由客户端与MYE

###### MYE上传业务数据

* MYE将传感器采集的数据封装成med格式，通过心跳路由客户端透传到后台算法服务器。value表示一个数组，可上传多条med指令。
* 上传数据无需返回消息。

{

"cmd\_url":"/heartbeatclient/business",

"cmd\_name":"report",

"vendor":"xxxx"

"value": [{

"med" : "xxxx"

}]

}

###### 发送指令到MYE

* 后台算法服务器将med控制指令，通过心跳客户客户端透传到MYE。value表示一个数组，可下发多条med指令。
* 下发指令无需返回消息。

{

"cmd\_url":"/zigbeeservice/business",

"cmd\_name":"issue",

"vendor":"xxxx"

"value": [{

"med" : "xxxx"

}]

}

#### 3.3、TR069客户端与MYE

###### 获取zigbee模块信息

* 查询zigbee模块信息，上报ACS。
* 外部程序向MYE发出请求

{"cmd\_url":"/zigbeeservice/module"，"cmd\_name":"query"}

* MYE返回模块信息表，result返回结果，0表示成功，失败为其他值。若成功，value为模块信息表内容。

{

"cmd\_url": "/zigbeeservice/module",

"cmd\_name": "query",

"result": 0,

"value": {

"MACAddress": "xxxx",

"NetworkPANID": "xxxx",

"Channels": "xxxx",

"numberofchildlimit": xx,

}

}

###### 设置zigbee模块信息

* ACS通过mye配置网关上模块信息表

{

"cmd\_url": "/zigbeeservice/module",

"cmd\_name": "setconf",

"node\_mac": "05030000000000F3",

"value":

{

"MACAddress": "05030000000000F3",

"NetworkPANID": "1234",

"Channels": "13,14,15,20,26",

"NumberOfChildLimit": "16",

"CurrentChannel": "13"

}

}

* myer回复此消息，result表示返回结果，0表示成功，失败为其他值。

{

"cmd\_url": "/zigbeeservice/module",

"cmd\_name": "setconf",

"node\_mac": "05030000000000F3",

"result": 0,

}

###### 获取zigbee设备信息

* 查询zigbee设备信息，上报ACS。
* 外部程序向MYE发出请求

{"cmd\_url":"/zigbeeservice/device", "cmd\_name":"query", "node\_mac":"xxx"}

* MYE返回zigbee设备信息，result返回结果，0表示成功，失败为其他值。若成功，value为设备信息表内容。

{

"cmd\_url": "/zigbeeservice/device",

"cmd\_name": "query",

"node\_mac": "xxx"

"result": 0,

"value": {

"xxxx" : "xxxx",

"xxxx": "xxxx",

"xxxx": "xxxx",

"xxxx": "xxxx",

}

}

###### 获取zigbee配置表

* 获取zigbee设备配置，上传ACS。
* 外部程序向MYE发出请求

{"cmd\_url":"/zigbeeservice/device", "cmd\_name":"getconf", "node\_mac":"xxx"}

* MYE返回zigbee设备配置信息，result返回结果，0表示成功，失败为其他值。若成功，value为设备配置表内容。

{

"cmd\_url": "/zigbeeservice/device",

"cmd\_name": "getconf",

"node\_mac":"xxx"

"result": 0,

"value": {

"xxxx" : "xxxx",

"xxxx": "xxxx",

"xxxx": "xxxx",

"xxxx": xx,

}

}

###### 下发zigbee配置表

* 设置zigbee设备配置。
* 外部程序向MYE发出请求，配置zigbee设备。
* value为配置表

{

"cmd\_url":"/zigbeeservice/device",

"cmd\_name":"setconf",

"node\_mac":"xxx",

"value": {

"xxxx" : "xxxx",

"xxxx": "xxxx",

"xxxx": "xxxx",

"xxxx": xx,

"white\_list": "null"或[mac1,mac2,mac3]

}

}

* MYE返回zigbee设备配置信息，result返回结果，0表示成功，失败为其他值。

{

"cmd\_url": "/zigbeeservice/device",

"cmd\_name": "setconf",

"node\_mac":"xxx",

"result": 0,

}

###### 配置上报

* zigbee设备配置发生变化，主动上报。
* MYE发出请求，上报改变配置
* value为配置表

{

"cmd\_url":"/tr069client/device",

"cmd\_name":"updateconf",

"node\_mac":"xxx",

"value": {

"xxxx" : "xxxx",

"xxxx": "xxxx",

"xxxx": "xxxx",

"xxxx": xx,

}

}

* MYE返回zigbee设备配置信息，result返回结果，0表示成功，失败为其他值。

{

"cmd\_url": "/tr069client/device",

"cmd\_name": "updateconf",

"node\_mac": "xxx",

"result": 0,

}

###### 获取当前完整拓扑信息表

* 获取当前zigbee网络完整topo信息，上传ACS。
* 外部程序向MYE发出请求

{"cmd\_url":"/zigbeeservice/topo", "cmd\_name":"enum"}

* MYE返回zigbee设备配置信息，result返回结果，0表示成功，失败为其他值。若成功，value表示数组，为当前所有zigbee设备拓扑信息表集合。

{

"cmd\_url": "/zigbeeservice/topo",

"cmd\_name": "enum",

"result": 0,

"value": [{

"xxxx" : "xxxx",

"xxxx": "xxx",

"xxxx": "xxx",

"xxxx": "xxx",

}]

}

###### 节点更新

* 当zigbee网络中，某设备发生拓扑改变，主动上报tr069客户端。
* MYE发出请求，value表示数组，为所有改变拓扑信息集合。

{

"cmd\_url": "/tr069client/topo",

"cmd\_name": "update",

"value": [{

"xxxx" : "xxxx",

"xxxx": "xxx",

"xxxx": "xxx",

"xxxx": "xxx",

}]

}

* tr069客户端返回结果，result返回0表示成功，失败为其他值。

{

"cmd\_url": "/tr069client/topo",

"cmd\_name": "update",

"result": 0,

}

###### 获取单个节点拓扑表

* 获取请求指定mac设备拓扑信息表。

{

"cmd\_url": "/zigbeeservice/topo",

"cmd\_name": "query",

"node\_mac": "xxx"

}

* mye返回结果，result返回0表示成功，失败为其他值。value表示数组表示拓扑信息表。

{

"cmd\_url": "/zigbeeservice/topo",

"cmd\_name": "query",

"node\_mac": "xxx",

"result": 0,

"value": {

"xxxx" : "xxxx",

"xxxx": "xxx",

"xxxx": "xxx",

"xxxx": "xxx",

}

}

###### 节点上线

* 当zigbee网络中，添加了设备，发生拓扑改变，主动上报tr069客户端。
* MYE发出请求，value表示数组，为所有改变拓扑信息集合。

{

"cmd\_url": "/tr069client/topo",

"cmd\_name": "add",

"value": [{

"xxxx" : "xxxx",

"xxxx": "xxx",

"xxxx": "xxx",

"xxxx": "xxx",

}]

}

* tr069客户端返回结果，result返回0表示成功，失败为其他值。

{

"cmd\_url": "/tr069client/topo",

"cmd\_name": "add",

"result": 0,

}

###### 节点下线

* 当zigbee网络中，移除了设备，发生拓扑改变，主动上报tr069客户端。
* MYE发出请求，value表示数组，为所有改变拓扑信息集合。

{

"cmd\_url": "/tr069client/topo",

"cmd\_name": "delete",

"value": [{

"xxxx" : "xxxx",

"xxxx": "xxx",

"xxxx": "xxx",

"xxxx": "xxx",

}]

}

* tr069客户端返回结果，result返回0表示成功，失败为其他值。

{

"cmd\_url": "/tr069client/topo",

"cmd\_name": "delete",

"result": 0,

}

###### 升级zigbee模块

* 升级zigbee模块。
* 外部程序向MYE发出请求。
* seq为序列号，唯一标识该条消息，0表示不对该消息进行标识。

{

"cmd\_url":"/zigbeeservice/module",

"cmd\_name":"upgrade",

"node\_mac":"xxx",

"url":"xxx"，

"md5":"xxx"，

"seq":x

}

* MYE返回结果，result返回0表示成功，失败为其他值。

{

"cmd\_url":"/zigbeeservice/module",

"cmd\_name":"upgrade",

"node\_mac":"xxx",

"result":0,

}

###### 升级zigbee设备

* 升级zigbee设备。
* 外部程序向MYE发出请求。
* seq为序列号，唯一标识该条消息，0表示不对该消息进行标识。

{

"cmd\_url":"/zigbeeservice/device",

"cmd\_name":"upgrade",

"node\_mac":["xxx","xxx"],

"url":"xxx"，

"md5":"xxx"，

"seq":x

}

* MYE返回结果，result返回0表示成功，失败为其他值。

{

"cmd\_url":"/zigbeeservice/device",

"cmd\_name":"upgrade",

"node\_mac":["xxx","xxx"],

"result":0,

}

###### 重启zigbee设备

* 升级zigbee设备。
* 外部程序向MYE发出重启zigbee设备请求。

{

"cmd\_url":"/zigbeeservice/device",

"cmd\_name":"reboot",

"node\_mac":["xxx","xxx"]

}

* MYE返回结果，result返回0表示成功，失败为其他值。

{

"cmd\_url":"/zigbeeservice/device",

"cmd\_name":"reboot",

"node\_mac":["xxx","xxx"],

"result":0,

}

###### zigbee设备通知消息

* zigbee设备发送异步消息接口

{

"cmd\_url": "/tr069client/async",

"cmd\_name": "notify",

"type": x,

"value" "xxxxx"

}