HTP V17 通讯协议介绍

修订记录

版本	日期	作者	内容
0.1	20150525	于明明	建档
0.2	20150601	于明明	加入自动发现协议
0.3	20150606	于明明	加入弱跳频协议
0.4	20150615	于明明	加入RF参数调整
0.5	20150619	于明明	修改弱调频为真实调频
0.6	20150623	于明明	第一次讨论后的修改,如下
			唤醒帧、ID帧、ACK帧格式修改
			恢复为弱跳频方式
			取消A7125方案
0.7	20150623	于明明	帧格式更新
0.8	20150629	于明明	加入开发计划
0.81	20150629	于明明	价签心跳和频率可调整、加入电量
0.82	20150701	于明明	定义基站上传数据格式
0.83	20150709	于明明	修改唤醒帧格式，精度从1ms细化到30us
			定义基站注册帧
0.84	20150716	于明明	添加价签注册功能测试需求
0.85	20150716	于明明	修改唤醒协议
0.86	20150730	于明明	动态调准WOR周期
0.87	20150801	于明明	1 去掉动态调准WOR周期功能
			2 slot值单位改为8ms
			3 注册时发射功率为0dBm
0.88	20150806	于明明	讨论价签注册机制，去掉功能开关、可配信道
0.90	20150810	于明明	加入协议设计章节，重点描述功能实现
0.91	20150812	于明明、耿文亮	整理协议设计章节，包含框图、输入输出、流程图
0.92	20150827	于明明	加入漫游章节
0.93	20150922	于明明	更新传输模式4的帧结构，更新漫游更新流程图
0.94	20150923	于明明	更新价签注册、动态绑定、更新漫游流程

协议特点

目前运行的HTP协议，包括上层系统与基站的传输协议，和基站与价签的通讯协议。此套协议最开始是服务于段码价签的，适合于短包收发。直至开发了点阵价签后，随着通讯的数据包越来越多，协议开始暴漏了许多不适应的地方，包括唤醒代价大、通讯速度慢。

于是设计了第二代HTP通讯协议，相比于第一代，它的优势主要如下：

- 极大减小唤醒代价，不通讯的价签被唤醒功耗减少10倍。

- 大数据量通讯时，速度提升近10倍。
- 加入价签注册协议，解决价签组网、漫游、绑定、丢失等问题。
- 加入弱跳频技术，避免信道干扰。

协议细节

第二代通讯协议重新设计了唤醒和数据交互的流程，尽最大化利用RF收发窗口，又充分考虑了价签的低功耗需求。

唤醒

WOR功能调整

WOR改动主要有两点：

- 将slot值从125ms改为5ms。
- 唤醒帧改为2帧（以下简述为帧0和帧1），用于精确休眠价签和指定数据通讯时的信道。

唤醒流程

价签收到唤醒帧后（前导码和ID Code匹配），通过解析帧0得知slot值，系统睡眠指定时间后，继续使用wakuepid和wakeup channel收听帧1。通过解析帧1得知自己是否需要进入数据通讯模式，以及数据通道使用的信道。

基站端

1. 基站负责填写帧0，并在唤醒时间内轮流发送帧0，slot值每5ms更新一次。
2. 帧1由上层负责填写，并在HTP报文里传给基站。唤醒时间窗口过后，基站在发送数据帧时，先发生30帧帧1，以后每隔30ms发送一次帧1。帧1使用唤醒信道和唤醒ID。

价签端

1. 唤醒后接收帧0，睡眠指定的slot时间。
2. 定时到后，价签继续使用wakeupid和wakeup channel接收帧1。如果100ms内都没有收到帧1。则转入WOR模式。
3. 如果收到帧1，则判断子网掩码里自己对应的比特位是否被置上。
    1. 如果置0，则表示不需要进行数据通讯。进入WOR模式。
    2. 如果置1，则表示需要进行数据通讯。使用帧1里的信道和自身ID进入数据接收模式。

调整后的功耗

静态功耗
因为帧0长度未发送变化，因此静态功耗和之前一样。

误唤醒功耗
误唤醒是指系统希望唤醒某个终端时，同组的其余价签也被唤醒。
之前如果1个价签被误唤醒，其功耗消耗为

启震	唤醒处理	睡眠slot	提前醒来	接收睡眠帧
900us*2mA	250us*16mA	4s*1mA	125ms*16mA	250ms*16mA

而在新协议中，此功耗降低为

启震	唤醒处理	睡眠slot	继续接收唤醒帧
900us*2mA	250us*16mA帧	4s*0.1mA	30*ms*16mA

直接降低了近10倍。

唤醒帧格式

唤醒帧由2帧不同的格式组成。

- 第一帧（下称帧0）长度和格式与现有的唤醒帧相同，只是slot的精度从原先125ms改为5ms。
- 第二帧（下称帧1）长度为26字节，与数据帧长度一致。

唤醒参数帧0，以下简称帧0

Preamble	ID Code	CTRL	SLOT	HW CRC
4B	4B	3b	13B	2B

唤醒参数帧1，以下简称帧1

Preamble	ID Code	CTRL	Unused	Channel	Netmask	Soft CRC	HW CRC
4B	4B	3b	5b	1B	22B	2B	2B

数据字段描述如下：

- **Preamble** RF前导码，RF硬件自动添加
- **ID Code** 价签的Wakeup ID，RF硬件自动添加
- **CTRL** 唤醒命令，占3bit，101b表示帧0，110b表示帧1
- **SLOT** 提示价签还需要多久才进入数据通讯模式，单位为5ms。用13比特表示，借用CTRL低5位。
- **Channel** 提示价签数据通信使用的信道
- **Soft CRC** 软件CRC值，从ID Code算起，到Soft CRC前一个字段止
- **HW CRC** RF层CRC值，从CTRL算起，到HW CRC前一个字段止
- **Netmask** 子网掩码，22字节共176个bit，即每个子网的容量为160个价签。

每个终端除了具有一个唯一的eslid外，还有一个1字节的组内id值，用于表示自己在本wakeup组的序号。终端根据这个序号得知自己的子网号。比如价签5A-11-22-99，假设其组内序号为22，则此价签的子网掩码为Bit 22（序号22%子网容量160）。

数据通讯

通过上面的唤醒流程，被唤醒的终端一定是真正需要通讯的。此时需要合理分配发送和接收窗口，避免出现空闲等待，尽量压缩交互次数。

第二代协议关于数据交互的改进有2个方面：

- 利用终端处理数据的时间，基站和其他终端进行通讯
- 终端被动集中返回ACK，大幅减少上行链路。

数据帧格式

数据帧

Preamble	ID Code	CTRL	SID	UNUSED	PKG NUM	DATA	Soft CRC	HW CRC
4B	4B	3b	5b	1B	2B	20B	2B	2B

终端收到此类数据帧后，只需处理。处理完后，继续接收下一帧，无需返回ACK。

- **CTRL** 010b表示数据帧
- **SID** SID值

数据查询帧

Preamble	ID Code	CTRL	SID	SLOT	TOTAL SEND	SALT	UNUSED	Soft CRC	HW CRC
4B	4B	3b	5b	1B	2B	6B	10B	2B	2B

- **CTRL** 011b 表示数据查询帧
- **SID** 此数据帧一致的SID值
- **SLOT** 每发一次数据包，此值加1
- **SALT** 第一个数据包的头6个字节
- **TOTAL SEND** 基站本次下发的包总数

终端收到这个查询帧后，将已经收到的包和总包数相对比，就知道丢失了哪些包，并通过ACK返回。

数据查询ACK帧

数据查询ACK帧由终端发回，因为回路数据长度受限，每次最多表示丢了10个包。
RF速率为100Kbps，payload加RF硬件层数据总共26字节，约需3ms

Preamble	ID Code	CTRL	Unused	SLOT	LOSS NUM	PKG NUM1~10	Soft CRC	HW CRC
4B	4B	3b	5b	1B	2B	20B	2B	2B

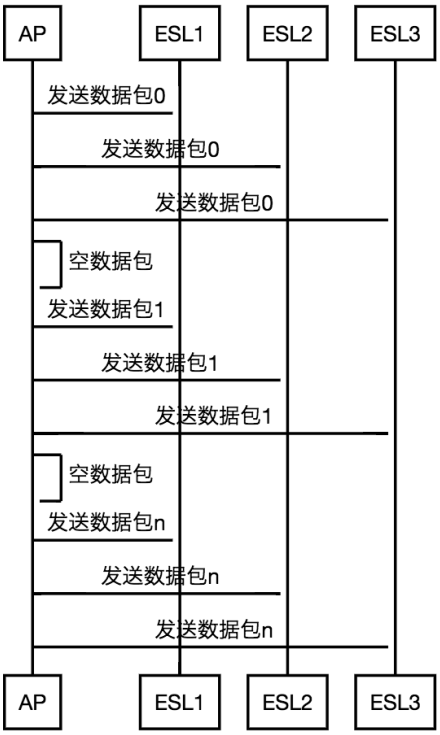
- **CTRL** 011b，价签发送时表示为数据查询ACK帧
- **SLOT** 必须是数据查询帧的slot值加1
- **LOSS NUM** 是否还有丢包。
    1. 如果为0，则表示有丢包
    2. 如果大于0，则表示至少丢了这么多包，丢失的包号前10个分为是PKG NUM1～10。
- **PKG NUM**1～10: 丢失的包号。

每个AKC帧最多能报告丢失的10个包号。基站只有再重新发送完这10个包后，才有可能知道终端是否还丢失了更多的包。

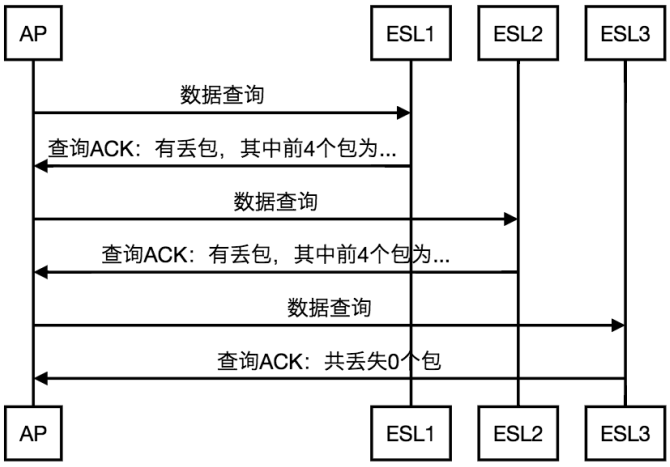
交互流程

基站给终端1～3发送共3n个数据包

- 第一步：发送数据



- 第二步：查询结果



- 第三步：如果超时或者发送完毕，则退出流程，否则回到第1步

更新速度

通讯速度
以通讯100个终端，每个终端16KB数据计算。

第一版协议的通讯耗时如下:

- 总共需要发送的数据包为81920（100*16KB/20B/每包字节)。
- 每次唤醒最多只能有3.5秒(扣除0.5秒的sleep帧)的发送窗口，每个数据包需要耗费8ms，每此唤醒最多发送437个

包
- 总共需要唤醒通讯187次，每次以20秒通讯时间计算，共需3740秒。

第二版协议的通讯耗时如下:

- 总共需要发送的数据包为81920（100*16KB/20B/每包字节)。
- 每次唤醒有4秒的发送窗口，每个数据包仅需耗费634us(34B*8bit/B*2us/b + 90us)，每此唤醒最多发送6309个包。扣除10%的ACK开销，每次发送数据5500包。
- 总共需要唤醒14次，第一次20秒，第2～5次每次13秒（此时终端唤醒周期缩短为1s），共需175秒。

唤醒次数减少了13倍，更新速度提升了21倍。

价签注册协议

现在的通讯协议非常依赖终端的ID信息，如果系统意外丢失了这些信息，将无法和终端通讯上。
新的协议中，将使终端具有主动发送数据的能力，可以较实时的广播自己的ID信息，基站不断捕获和汇总终端的ID信息，并上传给后台控制系统。
和以前的方式相比具有如下的特点:

1. 系统无需全局维护终端ID信息。
2. 终端可以自由漫游，只要能被基站覆盖到，就能精准通讯。
3. 缩短系统发现价签丢失的时间。
4. 实施时无须导入价签ID文件，即装即用。
5. 组网混乱后也能得终端的真实参数。

终端发送数据前需要检测空间信道是否空闲，如果空闲则立刻发送ID帧，如果忙，则delay几个时间片，如果还忙，则转入睡眠，等待下一个发送窗口。

数据结构

终端每3分钟进行一次广播心跳，RF频点为2，发射功率为0dBm，速率为100Kbps。
帧长24字节，其中数据部分16。发送一帧的时间为

> 24B * 8bit/B * 10us/bit + 130us RF Ready = 2.05ms

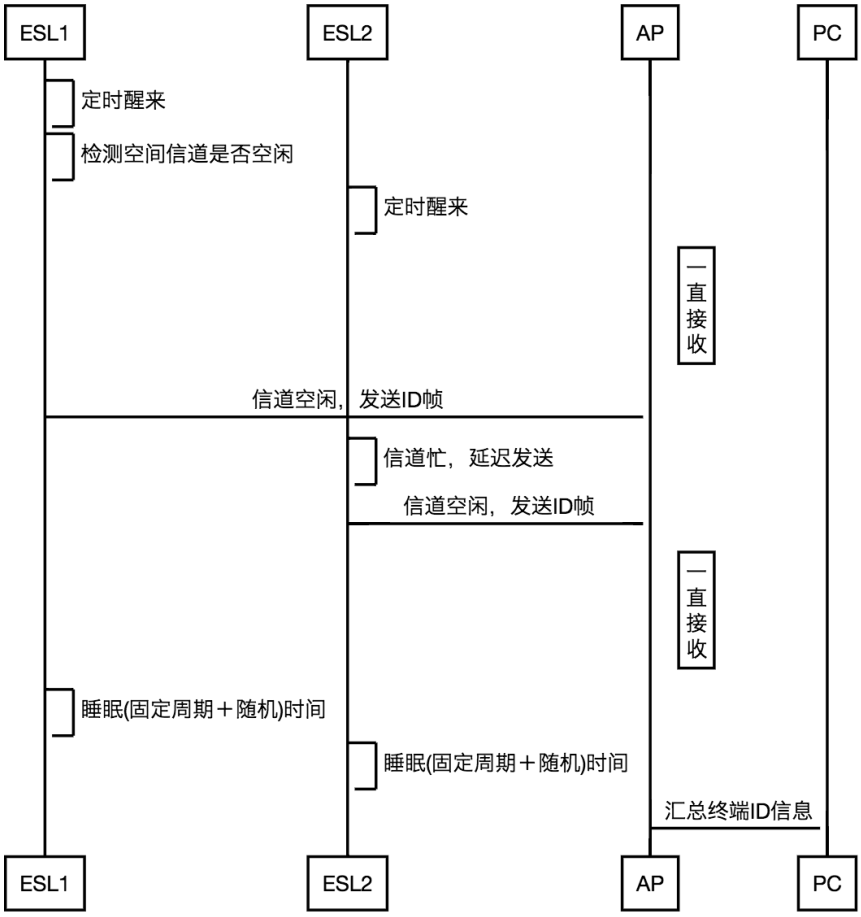
RF的工作时间比听帧多了约1倍，但频率为听帧的1/45。即相当于多了2%的动态功耗。

Preamble	ID Code	CTRL	Version	Wakeup ID	ESL ID	Channel	Subnet ID	Battery	Reserv	Soft CRC	HW CRC
4B	4B	3b	13b	4B	4B	1B	1B	1B	1B	2B	2B

- **CTRL** 110b，价签心跳帧
- **Version** 13比特，用于表示价签类型，决定屏幕分辨率、颜色、Flash大小、软件版本。
- **Wakeup ID** 终端的wakeup ID
- **ESL ID** 终端的通讯ID
- **Channel** 终端的通讯信道
- **Subnet ID** 组内ID号
- **Battery** 电压值
- **Reserv** 保留
- **Soft CRC** CRC16值（从ID Code到Subnet ID）

基站在接收终端的ID帧后，并附上RSSI信号能量检测值，上传给后台服务器。
服务器接收到多个基站反馈的终端ID帧后，根据RF Power能量值、价签分组、基站接收频率等因素决定将价签绑定到哪个基站上。

发现流程



基站注册协议

目前的协议里，基站定时1分钟向服务器发生1次心跳包，用于告知服务器自身的存在。

现在仿造价签注册机制，为基站也添加注册机制：基站定时使用RF广播自身，并和接收价签注册帧一样接收其他基站的注册帧，记录相关信息，再汇总给服务器。

此机制主要达到了如下目的，避免了实施时的人工参与

1. 系统可以知道基站A的信号都被哪些基站捕获到，以此推算出基站A的辅基站。
2. 系统可以知道基站A不在某些基站信号覆盖范围内，那些基站可以和基站A进行同频点更新。
3. 为后续的多门店管理系统简化相关逻辑

数据格式

基站每3分钟进行一次广播心跳。帧长24字节，其中数据部分16。

Preamble	ID Code	CTRL	Unused	APID	APIP	Status	Reserv	Soft CRC	HW CRC
4B	4B	3b	5b	4B	4B	1B	4B	2B	2B

:**CTRL** 111b，价签心跳帧

- **APID** 基站编号
- **基站IP** 基站IP地址
- **Status** 基站状态
- **Reserv** 保留
- **Soft CRC** CRC16值（从ID Code到Subnet ID）

弱跳频协议

现在的通讯协议里通讯信道都是预制的，基站和终端通讯的时候，如果空间信道受到干扰，已经没有机会更换信道，此时的通讯成功率将大大降低。

针对这种情况，我们在协议里增加了组网功能，可以更改终端的通讯参数，但存在如下的缺点：

1. 缺乏合理的触发机制，功能比较滞后
2. 组网可能不成功
3. 组完网后存在ID信息同步和丢失的风险

无线领域里解决此问题一般都是采用跳率的技术，要求发送者和接受者按照指定的序列，严格的时间间隔，一直变化信道进行通讯，避免信道干扰。

跳频的关键有三点：

- 起始时间一致
- 时间精度一致
- 跳频频点一致

后面将对此三点提供解决方法。
跳频涉及到信号发送者和信号接收者的时间同步，此同步通过在唤醒帧中加入相关处理机制实现。

- **起始时间一致** 通过唤醒帧中的slot值通知接收者在多少毫秒后开始进行数据通讯。
- **时间精度一致** 终端将读取若干个唤醒帧0，读取其中的时间差，从而调整自己的定时器精度。
- **跳频频点一致** 跳频频点通过后台服务器统一管理和下发。

信道质量扫描

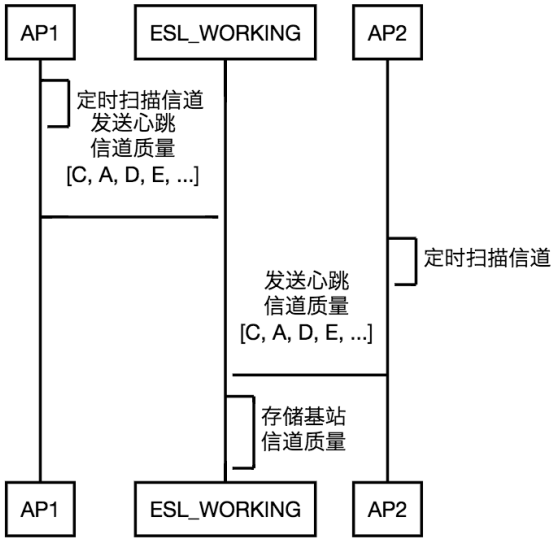
基站负责定时扫描自身周围的无线信道质量，每次心跳时向服务器上传信道质量值，服务器记录下每个基站的周围无线信道值。

唤醒时服务器将适合于基站环境的信道填入到唤醒帧中，指明了数据通信将采用的信道。

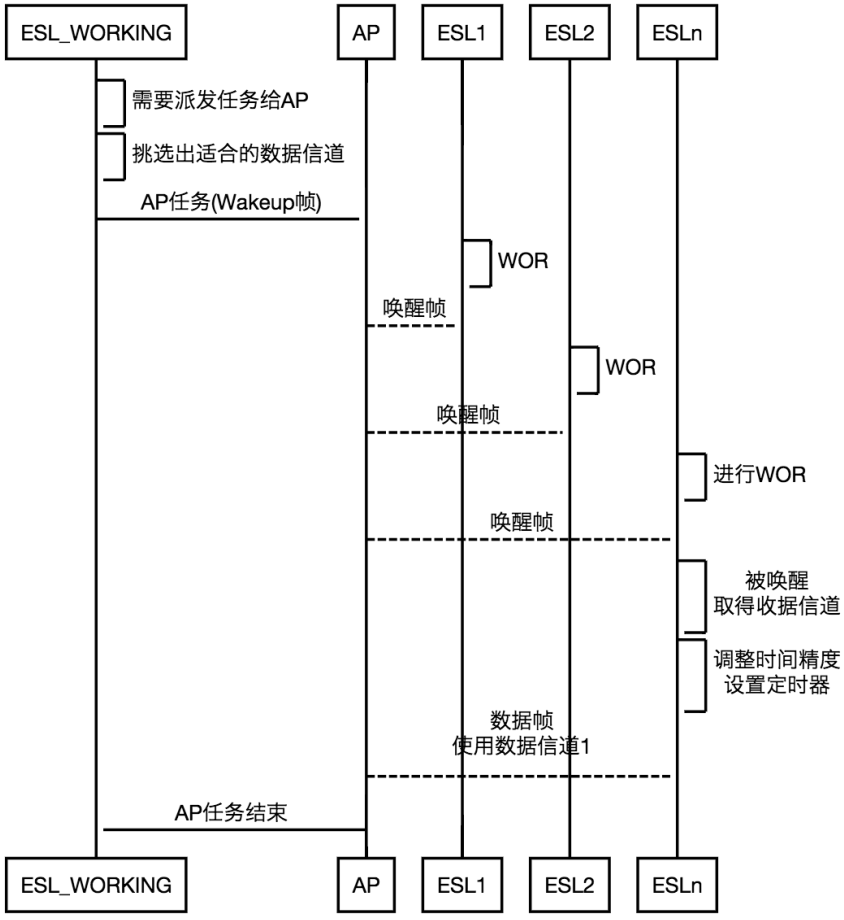
基站仍然使用终端的信道发送唤醒帧，由于基站的发送功率大发送时间长，因此即使存在信道干扰，也能比较成功的唤醒终端。但终端不能使用这个信道进行通讯，而是使用唤醒帧里面传入的推荐信道进行通讯。

跳频通讯流程

收集信道质量



通讯时指定数据信道



基站上传的数据格式

基站在空闲时收取1秒时长的广播帧，其中包括价签的注册帧，也包括基站的注册帧，二者的数据包可以汇总成一个UDP包发送给服务器，命令字都是51

价签注册

基站将价签发送的注册数据转成UDP数据包，发送给服务器，格式如下

> DATASTART**51**000[{“eslid”:”5A-15-97-99”,”nw1”:”5A-04-8E-66”,”nw3”:”75”,”rfpower”:”6”,”netid”:”151”,”apid”:”10”,”version”:”5”,”battery”:”32”,”reserve”:”0”}]0X0000DATAEND

基站注册

基站将收到的其他基站发送的注册数据转成UDP数据包，发送给服务器，格式如下

> DATASTART**51**000[{“apid”:”1234”,”apip”:”192.168.10.2”,”status”:”0”,”rfpower”:”6”,”reserve”:”0”}]0X0000DATAEND

- DATASTART**51**000: 命令字51表示价签心跳数据
- **eslid** 价签ID
- **nw1** 唤醒ID
- **nw3** 信道
- **rfpower** 基站接收到的价签或者基站RF能量值
- **netid** 价签子网号
- **apid** 基站编号
- **apip** 基站IP地址
- **version** 价签版本号
- **battery** 价签电量
- **reserve** 保留

信道扫描

基站将空间信道扫描值，转成UDP数据包，发送给服务器，格式如下

> DATASTART52000[{“4”:”20”, “50”:”55”,”90”:”75”,”150”:”45”}]0X0000DATAEND

结果由｛”**信道**“:”**能量值**“｝的健值对构成。
需要扫描的信道包括：2，4，7，46，50，54，90，94，98，102，142，146，150，154，158。
信道选择标准：

- 远离WIFI信道1、6、11
- 和蓝牙信道不重复
- 融入现在价签系统已有的信道 50，90，150，和打算使用的4号信道。

RF参数调整

白化选项

Data Whitening 功能将减少RF芯片接收一连串bit0或者bit1时的解码失败率。
我们的数据帧里面会经常出现全0或者全1的数据，因此有必要开启RF的数据白化选项。

HTP V17 测试需求

价签注册功能测试需求

功能设计

见 `价签注册协议` 章节

接口设计

- esl-working系统在UDP 8800端口(和把枪端口一致，可通过配置文件修改)接收基站上传的价签注册信息，数据格式如下。一条注册信息最多支持30条记录。

  > DATASTART**51**000
  > [{“eslid”:”5A-15-D2-99”,”nw1”:”5A-05-30-66”,”nw3”:”50”,”rfpower”:”84”,”netid”:”210”,”apid”:”1”,”version”:”5”,”battery”:”30”,”reserve”:”0”},{“eslid”:”5A-15-D4-99”,”nw1”:”5A-05-0D-66”,”nw3”:”120”,”rfpower”:”8”,”netid”:”212”,”apid”:”1”,”version”:”5”,”battery”:”28”,”reserve”:”0”}]
  > 0X0000DATAEND

- esl-working在xml-server服务上提供命令 `"HEART_BEAT"` 命令，供价签注册使用。命令及数据格式如下。数据长度没有限制，但受限于操作系统的内存。

  > xmlserver.send_cmd(“**HEART_BEAT**“, [{“eslid”:”5A-15-D2-99”,”nw1”:”5A-05-30-66”,”nw3”:”50”,”rfpower”:”84”,”netid”:”210”,”apid”:”1”,”version”:”5”,”battery”:”30”,”reserve”:”0”},{“eslid”:”5A-15-D4-99”,”nw1”:”5A-05-0D-66”,”nw3”:”120”,”rfpower”:”8”,”netid”:”212”,”apid”:”1”,”version”:”5”,”battery”:”28”,”reserve”:”0”}])

功能需求

- 价签注册信息将保存到esl-working数据库的esl_list表中。
- 如果价签已经注册，则更新条目。
- 如果未注册，则插入条目。
- 其中 `version` 字段将翻译成op5、op6、nw4、op3的值。它们的映射关系在文件 `table_esl.py` 中写明。
- `battery` 字段是价签的实际电压四舍五入值。比如实际电压2.85v将表示为2.9v。要求从2.6v~3.0v各阶段的电压值正确。
- 注册模块只会初始化“op5”字段，其他任何情况下都不会更改op5的值。
- 注册功能对价签的5年功耗影响小于5%。

业务需求

价签注册是系统的一个基础功能，将影响如下业务

- 无论价签组网成败与否，系统都能够和价签通讯上。
- 系统丢失价签ID信息或者ID信息错误时，系统能够修复ID信息，并和价签通讯上。
- 系统根据基站捕获到的价签能量值(`rfpower` 字段)，将其绑定关系调准到与之最适配的基站上。逻辑如下：
  - 如果发现能量值比已绑定的基站高1个级别的情况，把绑定关系切换到能量值高的基站上。级别定义如下
    - 0~50 为同一个级别
    - 50以上，能量值增加10，则增加一个级别。
    - 能量值越低，表示价签和基站的通讯效果越好。

    - 如果发现能量值和已绑定的基站相当，且其他基站也绑定过本组的价签，且绑定的价签个数大于本基站，则将绑定关系调准到其他基站。如果本组价签被绑定到多个基站上，则挑选绑定个数最多的。
    - 如果能量值比已绑定的基站低一个级别，绑定关系不改变。
    - 本组价签还没有被其他基站绑定过，不切换绑定关系。

- 如果价签注册功能失效，通过手动或导入ID的方式，更新、绑定、解绑等业务能正常运作。
- 如果系统在1小时内都没有从价签所绑定的基站上收到过注册信息，将调整绑定关系到适配的基站上，即使新绑定的基站能量值比当前绑定的低1个级别。适配的逻辑如上所述。

性能需求

价签每3~4分钟在信道2上以100Kbps的速率发送1次注册信息。

- 空间环境理想情况下（信道2周围比较干净）
    - 价签的注册周期在3~4分钟左右。
    - 单个基站的情况下，价签必须在10分钟内注册成功。
    - 多个基站（3个或以上）的情况下，价签必须在4分钟内注册成功。
    - 1K个价签在多个基站的情况下，4分钟内全部注册成功。
    - 更新、绑定、解绑、查询等业务没有卡顿。

- 空间环境恶劣情况下（WIFI信道从1~11都一直在进行数据通信）
    - 单个基站的情况下单个价签必须在10分钟内注册成功。
    - 1K个价签在多个基站的情况下10分钟内全部注册成功。

- esl-working系统可以承受住10W个价签的注册信息（通过报文模拟测试）
    - 100个基站同时发送100条注册信息，每1秒发送一次，每10次改变报文的1个值。CPU占用在30%~40%左右。
    - 更新、绑定、解绑、查询等业务卡顿时间在2秒以内。

- 需要监测在1000个基站发送100W个价签注册信息时的性能情况。

HTP V17 涉及修改

esl-working

- 价签注册内容保存进esl_list表
- 价签绑定关系自动调整
- HTP支持传输模式4:
    - 单次发送5K数据包
    - 指定唤醒帧0，包含数据通道信道
    - 指定唤醒帧1，指定工作价签位图
    - RF数据白化选项

- 老式价签使用模式3传输，新式价签使用模式4传输
- 解绑时在价签左上角现实固定大小的OFF
- OSD支持4位小数

价签

- 支持传输模式4，不兼容之前模式，但查询组网不改

- 唤醒时，读取多个帧0，校准定时器，精准唤醒
- 唤醒时，读取数据通道信道
- 每3分钟广播一次心跳帧
- 心跳机制及心跳周期可由查询帧的头2个字节控制。值为0表示关闭，单位为分钟。
- OSD支持4位小数

基站

- 新增支持传输模式4，并兼容以前模式
    - 单次发送支持5K～2W个数据包
    - 更新唤醒帧0，slot值精确到ms
    - 唤醒帧1由服务器指定
    - RF数据白化选项
- 基站闲时接收价签的心跳帧，汇总上报给服务器
- 基站闲时扫描空间信道值，汇总上报给服务器

HTP V17 通讯协议设计

esl-working价签注册机制

模块框架

框架包含esl-working、基站、价签三个平台。

数据格式

基站发来的数据格式为

```
DATASTART51000[{"eslid":"5A-1B-1A-99","nw1":"51-04-04-
66","nw3":"35","rfpower":"119","netid":"26","apid":"10","version":"1","battery":"29","r
eserve":"0"},{"eslid":"5A-16-6B-99","nw1":"51-03-03-
66","nw3":"35","rfpower":"140","netid":"107","apid":"10","version":"5","battery":"30","
reserve":"0"}]0X0000DATAEND
```

体现在python log里为：(由于数据量大，只有debug模式才会出现)

```
DEBUG     ;2015-08-11 13:48:38;bind.py        [336][4481 ];
<<<:DATASTART51000[{"eslid":"5A-1B-1A-99","nw1":"51-04-04-
66","nw3":"35","rfpower":"119","netid":"26","apid":"10","version":"1","battery":"29","r
eserve":"0"},{"eslid":"5A-16-6B-99","nw1":"51-03-03-
66","nw3":"35","rfpower":"140","netid":"107","apid":"10","version":"5","battery":"30","
reserve":"0"}]0X0000DATAEND
```

该数据格式包含

- eslid

- nw1
- nw3
- rfpower : 能量值
- netid : 子网号（保存到OP2）
- apid : 由哪个基站反馈上来
- version : 下面详说该字段
- battery : 电压值（保存到OP1）
- reserve : 保留字段

version
version字段对用的是config/table_esl.ini文件的内容
该字段给出了某个价签op5， op4(可以没有)， op3， nw4，op6的默认值

```
{
    "1": {"op5": {"color":"154-black-right","version":1, "resolution_x":200,
"resolution_y":200, "flash_size":64, "max_package":900, "resolution_direction":0,
"screen_type":"HS_EL_5103"},"op4":"Demo20141205B","op3":"HS_EL_5103","nw4":"DOT20","op6"
:{"resolution_x":200,"resolution_y":200,"color":"black"}},
 ...
  "5": {"op5": {"color":"290-black-right", "version":1, "resolution_x":296,
"resolution_y":128, "flash_size":64, "max_package":900, "resolution_direction":0,
"screen_type":"HS_EL_5103"},
"op4":"Demo20141205B","op3":"HS_EL_5103","nw4":"DOT20","op6":
{"resolution_x":296,"resolution_y":128,"color":"black"}},
...
}
```

业务流程

- 系统受到价签注册信息后，如果信息发送变化（判断依据如下），则将数据更新进数据库
    - ’nw1’, ‘nw1’, ‘nw2’, ‘nw3’, ‘op1’, ‘op2’, ‘op3’, ‘op5’, ‘op6’, ‘op7’, ‘op8’ 值发生变化
    - ‘op4’, ‘op6’, ‘op9’ 在数据库中不存在，或其值为空
    - esl_list 其余字段不予考虑
- 如果通过esl_add方法修改了价签参数，系统将立即发觉，以上判断标准同样生效
- 如果通过数据库命令修改了价签参数，以上判断标准直至重启后才会生效

esl-working价签动态绑定

模块框架

框架包含esl-working、基站、价签三个平台。

数据格式

该数据的来源也是价签的心跳信息，故其格式和价签注册的数据同源。

业务流程

动态绑定流程分为2部分：

1. 根据心跳信息选择最佳基站
2. 在最佳基站挑选合适的组网参数

甄选基站模块

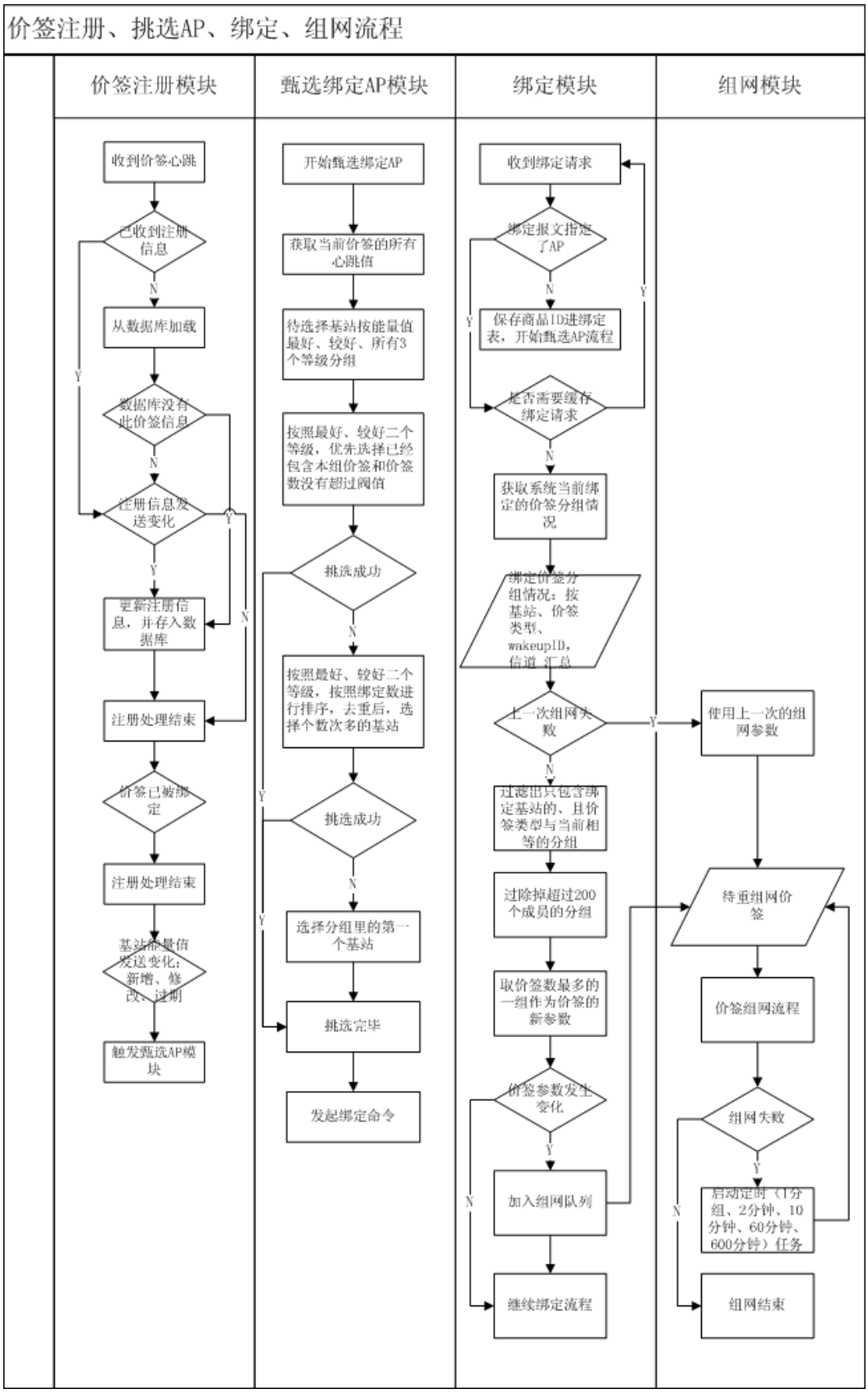
- heartbeat将收集每个价签的被每个基站捕获的最近一次信息。
- 根据ap的能量值和ap与同组价签的绑定关系选出最优的基站。选择原则如下：
    1. ap的能量值越高越好（等级越高越好）
    2. 如果某些ap的能量值是同一等级，那么某个ap下同组的esl数目越多越好。
    3. 同一组的esl尽量绑定到同一个基站
- 提交绑定请求（只更改基站）

挑选组网参数

系统受到绑定请求时，将挑选一组合适的参数对价签进行组网，挑选的原则如下：

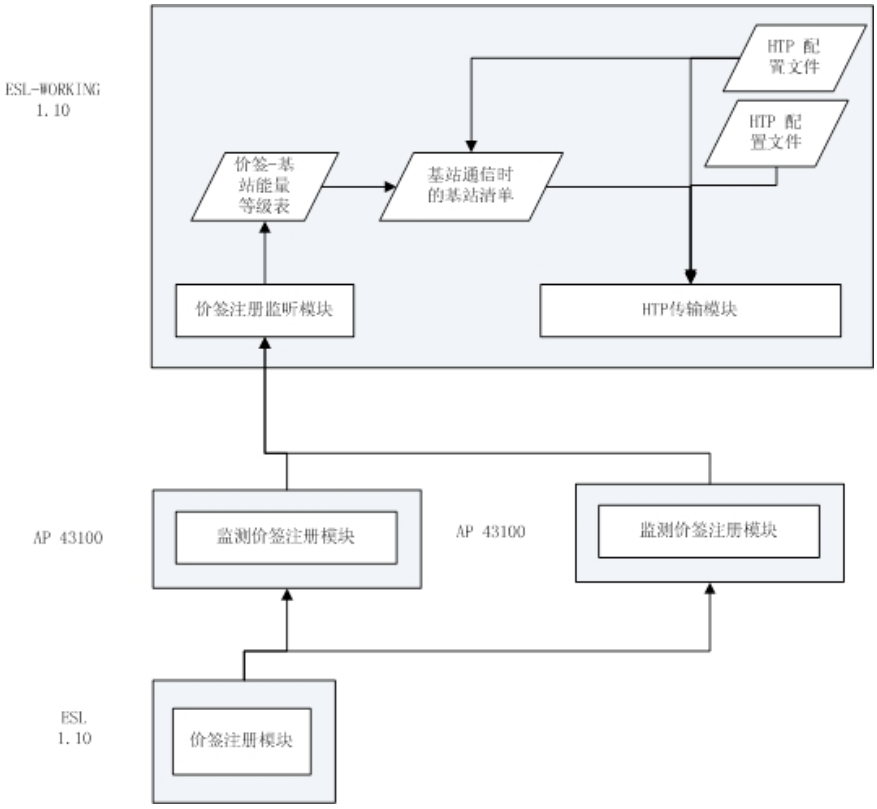
- 必须为同一种价签类型，由op3字段决定
- 某一个wakeupid、chn组，其成员数最多。如果组内成员数已满（等于200），则挑选次多的一组。
- 如果存在并列选择，优先选取和自身参数不相等的一组。
- 如果组网失败，会在1分钟，2分钟，10分钟，1小时，10小时之后重新进行组网，系统重启后，则丢弃此组网信息
- 如果组网失败，则将欲组网的新得参数保存在op5的new_nw1, new_nw3中。
- 如果上一次组网失败，又希望进行新的组网时，将仍然使用上一次失败的参数进行组网。不会使用新的参数。

流程图集


价签注册、挑选AP、绑定、组网流程
价签注册模块
甄选绑定AP模块
绑定模块
组网模块
收到价签心跳
已收到注册信息
N
从数据库加载
Y
数据库没有此价签信息
N
注册信息发送变化
Y
Y
更新注册信息，并存入数据库
N
注册处理结束
价签已被绑定
注册处理结束
基站能量值发送变化：新增、修改、过期
触发甄选AP模块
开始甄选绑定AP
获取当前价签的所有心跳值
待选择基站按能量值最好、较好、所有3个等级分组
按照最好、较好二个等级，优先选择已经包含本组价签和价签数没有超过阀值
挑选成功
N
按照最好、较好二个等级，按照绑定数进行排序，去重后，选择个数次多的基站
挑选成功
N
选择分组里的第一个基站
挑选完毕
发起绑定命令
收到绑定请求
绑定报文指定了AP
N
保存商品ID进绑定表，开始甄选AP流程
是否需要缓存绑定请求
N
获取系统当前绑定的价签分组情况
绑定价签分组情况：按基站、价签类型、wakeupID，信道 汇总
上一次组网失败
N
过滤出只包含绑定基站的、且价签类型与当前相等的分组
过除掉超过200个成员的分组
取价签数最多的一组作为价签的新参数
价签参数发生变化
Y
加入组网队列
继续绑定流程
使用上一次的组网参数
待重组网价签
价签组网流程
组网失败
Y
启动定时（1分组、2分钟、10分钟、60分钟、600分钟）任务
组网结束

esl-working 漫游更新

模块框架



数据格式

配置文件

- config/config.ini

```
[htp]
global_retry_time_max = 0
enable_roaming = yes
```

1. **global_retry_time_max** 为全局最大重试次数，但此值大于0时，表示系统所有价签的最大重试次数。单值为0时，表示通过数据库中esl_list的op8字段控制单个价签的最大重试次数。
2. **enable_roaming** 漫游功能开关。如果值为no，则系统只是价签绑定的基站进行更新。如果为yes，则在绑定的基站通讯失败后，再使用漫游基站继续进行更新。

- config/table_esl.ini

```
“1”: {“op5”: {“color”:”154-black-right”,”version”:1, “resolution_x”:200, “resolution_y”:200, >”flash_size”:64,
“max_package”:900, “resolution_direction”:0, “screen_type”:”HS_EL_5103”},
“op3”:”HS_EL_5103”,”nw4”:”DOT20”,”op6”:{“resolution_x”:200,”resolution_y”:200,”color”:”black”},
“op8”:”12”},
```

1. **op8** 字段定义了此种类型价签的最大重试次数。如果不填，则缺省值为6.
2. 对于老的点阵价签，需要手动将数据库的op8值填入合适的值。
3. 4.2寸及以上尺寸的点阵价签，重试次数为16；4.2寸以下的点阵价签，重试次数为12.
4. 所有断码价签，重试次数为6.

数据库

1. 新增op8字段表示价签的最大重试次数。
2. 新价签的op8字段通过价签注册自动填入，且每次都会被覆盖。
3. 老价签的op8字段如果不存在，需要手动设置值。否则系统会使用默认值6。

xmlserver接口

```
print server.send_cmd("GET_AP_LEVEL", [{"eslid":"5A-17-06-99",}])

['OK', [{'ap_level': [['5A-17-06-99', '1', 0, '6', '2015-08-26 21:06:01', '130 second before'], ['5A-17-06-99', '8', 0, '8', '2015-08-26 21:06:00', '130 second before']], 'eslid': '5A-17-06-99'}]]
```

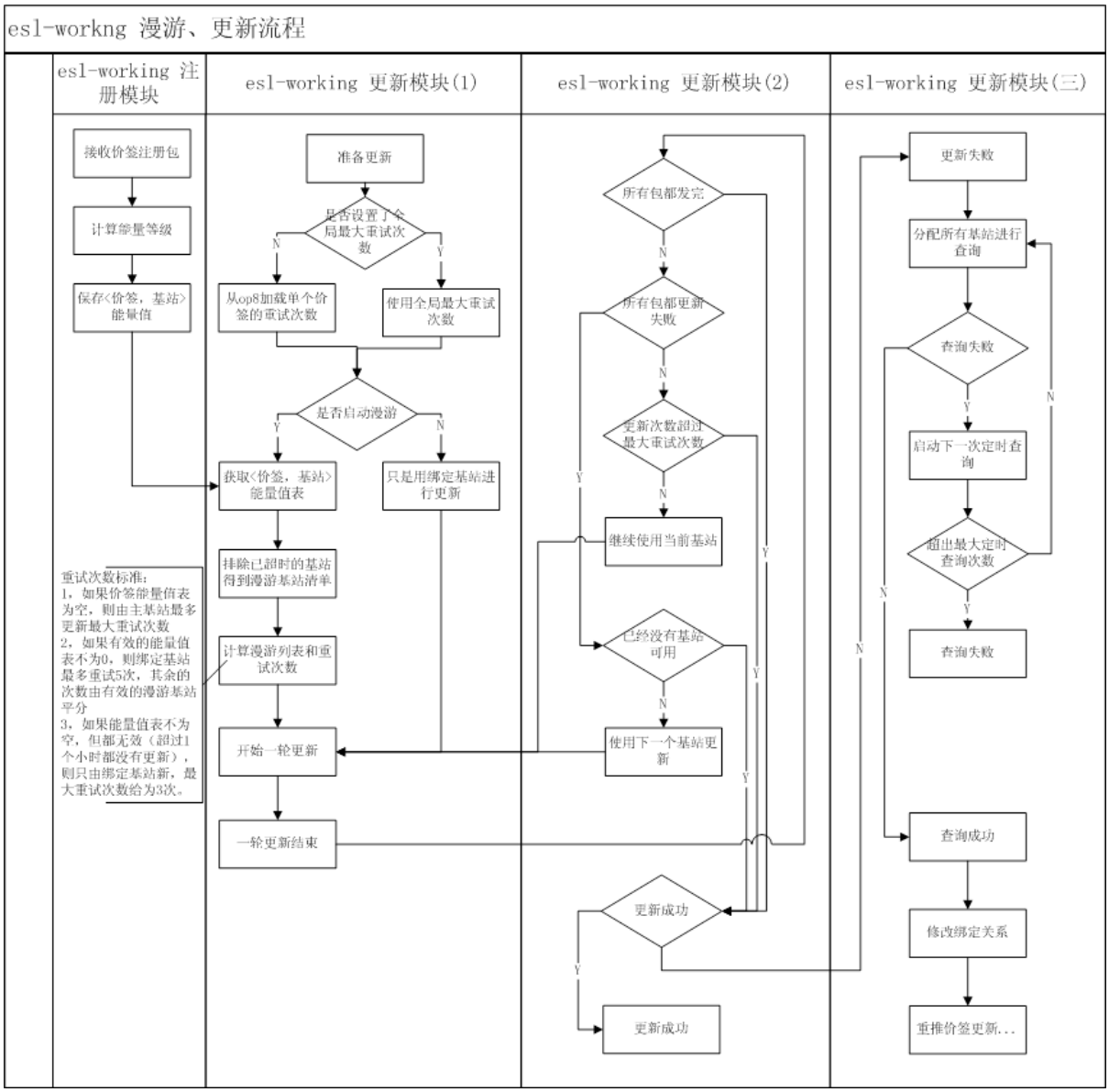
1. xmlserver 接口提供命令 GET_AP_LEVEL，供查看哪些基站捕获过价签的心跳信息。
2. **'5A-17-06-99', '1', 0, '6',** 分别是价签号，基站号，能量级别，能力值；
3. **'2015-08-26 21:06:01', '130 second before'** 分别表示最后捕获的时间，距离现在过了多少秒。

相关log

```
INFO ;2015-08-26 20:52:29;core.py [135][3248 ];ap_list 5A-17-06-99 of [u'8', u'8', u'8', u'8', u'8', '1', '1', '1', '1', '1', '1', '1']
```

1. 表示此次更新，价签5A-17-06-99通讯时，使用的基站清单为[8,8,8,8,8,1,1,1,1,1,1,1]。

业务流程



此中有几个子逻辑需要阐述下:

1. 计算能量等级
    - 0~50 为一档
    - 50~100 每间隔10个位一档
    - 能量数值越低一般代表价签离基站的通讯效果越好

2. 计算重试次数和漫游列表
    1. 重试次数标准
        1. 如果配置文件里全局重试次数被启用，则所有价签都是有全局的重试次数
        2. 否则使用价签数据库里op8的值，如果op8没有值，则默认为6

    2. 漫游列表标准
        1. 如果价签的能量值表位空，表示是新加入的价签，且还未被其余基站捕获到，此时没有漫游列表。只是用绑定基站进行更新
        2. 如果能量值表不为空，但全部都超期，则表示此价签被回收。此时也没有漫游列表，重试次数改为3.
        3. 如果能力值表不为空且日期较新，则使用绑定基站更新5次，其余的次数由漫游列表的基站按照等级

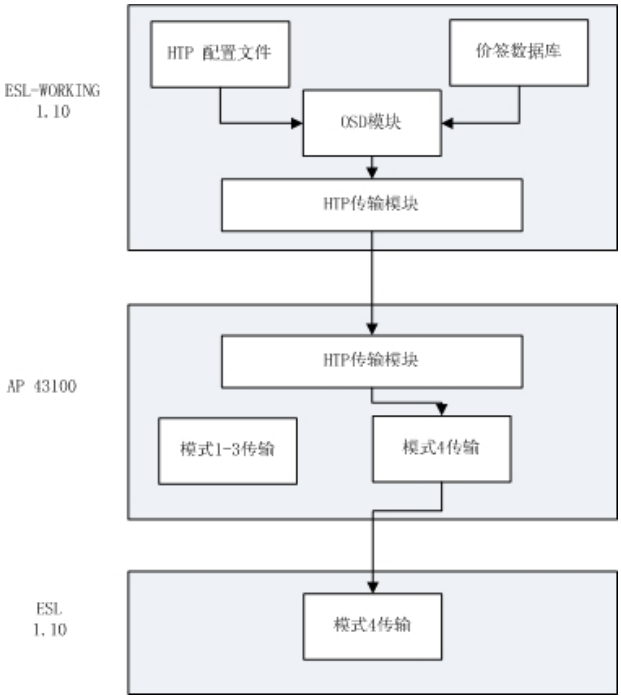
评分。比如绑定基站时1，漫游基站按照能力值排序为2，3，4，总共重试次数为12，则价签的通讯基站列表为[1,1,1,1,1, 2,2,2, 3,3, 4,4]

3. 价签更新失败后，将把价签加入定时查询任务中。
    1. 定时查询任务负责使用所有基站查询价签，如果所有基站都查询失败，则将任务周期延长。
    2. 任务周期为 1小时、5小时、12小时、6天。超过此次数后不再进行查找。
    3. 若查询成功价签，则切换绑定关系，并重推更新信息。

HTPV17 传输模式4

模块框架

框架包含esl-working、基站、价签三个平台。
其中esl-wokring版本 1.10 和基站版本43100兼容以前的传输模式。点阵价签只支持传输模式4。



数据格式

HTP配置文件

新的传输模式只适合于新的点阵价签(版本号SVN2698及以后)，系统需要为此类价签配置正确的通讯参数。

- config/htp_conf.ini
  在config/htp_conf.ini中增加了如下一个条目，将定义一种新的分组价签**HS_EL_5104**，此值对应价签ID文件中的op3值。HS_EL_5104和HS_EL_5103的价签不能分配或者组网到一起，否则会出现更新错误。

    1. 此条目中的**transfermode**值从之前的3为4，此类价签将采取传输模式4，
    2. 新增**query_timeout**字段，用于控制传输模式4中，发送链路层查询的等待超时时间，单位为ms。此值和**timeout**字段不同，后者指的是老的传输模式中的链路层等待超时时间，且单位为100us。

3. **register_on** 和 **register_chn** 控制价签注册功能的开关和价签的注册信道，目前无意义，字段保留。

```
"HS_EL_5104":
{
    "rf_para":{"wakeuptime":8,
    "ctrl":{"UPDATA":16, "NETLINK":17, "QUERY":18, "DOT_DIGITAL":16},
    "transfermode":4, "bps":100, "timeout":80, "register_on":1,"register_chn":2,
    "query_timeout":6,"framelen":26, "worktime":4, "sleepid_ctrl":2}
}
```

- config/esl_conf.ini
  这个文件主要定义了断码价签的显示模板，但是点阵价签也会将里面的字段用于查询和组网。
  文件中新增了字段**HS_EL_5104**，对应于价签数据库中的op5字段的**screen_type**字段。其值完全复制HS_EL_5103, 无需修改。

```
"HS_EL_5104":{
    "title" : ["Price", "Price1", "Price2", "Price3"],
    "Price" : {"format":"16s", "check_fun":"check_price", "maxlen":9},
    "Price1" : {"format":"16s", "check_fun":"check_price_int_str", "maxlen":5},
    "Price2" : {"format":"16s", "check_fun":"check_price_int_str", "maxlen":5},
    "Price3": {"format":"16s", "check_fun":"check_price_int_str", "maxlen":5},
    "check_key":["oldprice"]
},
```

- confing/config.ini
  config/config.ini文件中新增了**osd_args_v2**字段，里面的数字**4500**表示模式4时一次可以传输的最大包数。

```
[htp]
osd_args_v2 = LOG_OFF 4500 A8
```

价签数据库

凡是符合传输模式4的价签，其op3值都为**HS_EL_5104**，表示其是采用的是新的传输模式；op5中**screen_type**值也为**HS_EL_5104**，表示其采用了新的数据格式。

> 51-03-03-66,5A-2F-97-99,52-56-78-53,35,DOT20,30,151,**HS_EL_5104**,None,None,None,{‘max_package’: 900, ‘resolution_y’: 400, ‘resolution_x’: 300, ‘resolution_direction’: 1, ‘color’: ‘420-black-right’, ‘flash_size’: 32, ‘version’: 1, ‘screen_type’: ‘**HS_EL_5104**‘},{‘color’: ‘black’, ‘resolution_y’: 400, ‘resolution_x’: 300},0

HTP传输模块

htp 传输时主要的区别表现为参数**rf_transfer_mode**，其值为4.

> INFO ;2015-08-07 16:17:52;htp.py [543][4607];htp_task:slaveap num:0
> check_rcved_sec = 30, check_ack_sec = 60, socket_timeout = 6
> ftp_ip:127.0.0.1, ftp_port:21, ftp_user:hanshow, ftp_pass:hanshow
> apid:1, ap_addr:192.168.1.199, ap_port:5649, ap_data_len:156
> htp_version:16, htp_timestamp:20150807161737, htp_reserve:rf_ctrl:10, rf_para:00, rf_masterid:52-56-78-53, rf_wakeupid:5A-06-06-66
> rf_channel:133, rf_wakeup_time:8, rf_work_time:4, rf_frame_timeout:80, **rf_transfer_mode:4**
> rf_recv_bps:100, rf_fifo:26, sleep_times:0, sleepid_ctrl:2, rf_reserve:, data_num:1, sleep_num:0, crc=165A
> rf_query_timeout:6
> FOLLOWING IS ERROR MESSAGE:

基站输出log

传输模式4时的基站输出的log的基本格式和内容如下，在以后的版本可能会有一些微调。

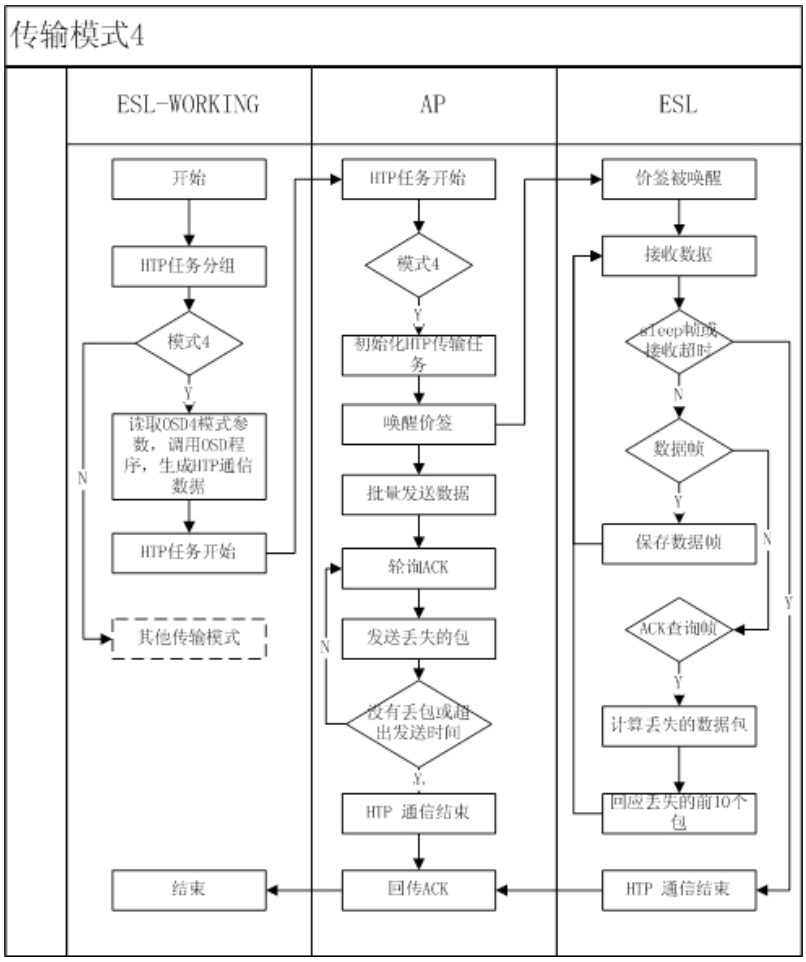
> (HP)RSF 192.168.1.104 (192.168.1.104)
> **packn** = 254, **esl num** = 1
> **0,5A-16-16-99, offset 0x117058, pkg 254, miss pkg 254, first miss pack 1**
> 0 send first package, left 1 tx frame 5303 slot.
> 1 send first package, left 1 tx frame 3827 slot.
> **5A-16-16-99 missing: 1,2,3,4,5,6,7,8,9,10,**
> 5A-16-16-99 missing: 11,12,13,14,15,16,17,18,19,20,
> 5A-16-16-99 missing: 21,22,23,24,25,26,27,28,29,30,
> 5A-16-16-99 missing: 31,32,33,34,35,36,37,38,39,40,
> 5A-16-16-99 missing: 41,42,43,44,45,46,47,48,49,50,
> 5A-16-16-99 missing: 51,52,53,54,55,56,57,58,59,60,
> 5A-16-16-99 missing: 61,62,63,64,65,66,67,68,69,70,
> 5A-16-16-99 missing: 71,72,73,74,75,76,77,78,79,80,
> 5A-16-16-99 missing: 81,82,83,84,247,248,249,250,251,252,
> 5A-16-16-99 missing: 253,254,
> **ack is 64 for 5A-16-16-99**
> **recv 1/1 success.**

- **packn** 本次HTP通讯的总数据包数
- **esl num** 本次HTP通讯的总价签数
- **0,5A-16-16-99, offset 0x117058, pkg 254, miss pkg 254, first miss pack 1**， 第0个价签，ID为5A-16-16-99，总共需要传254个包，默认丢失254个包，第1个丢失的包号为1。
- **5A-16-16-99 missing: 1,2,3,4,5,6,7,8,9,10,** 某次查询时，价签5A-16-16-99丢失了1~10个包。
- **ack is 64 for 5A-16-16-99** 价签最后通讯成功
- **recv 1/1 success.** 本次HTP通讯接收，成功价签数/总共价签数。

业务流程

esl-working处理流程

整体工作流程如下，包含esl-working、基站和价签。



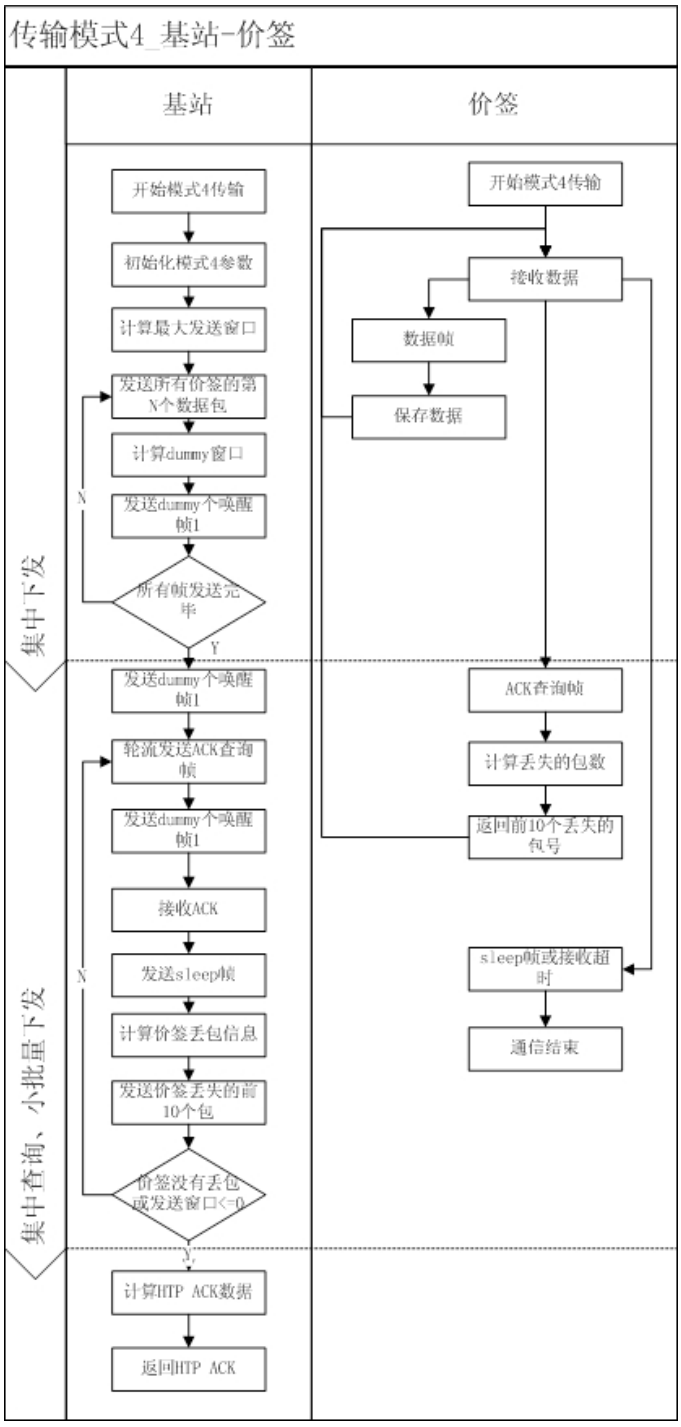
- 进入模式4的标准：
  - 价签的op3字段为HS_EL_5104
- 基站退出模式4的标准：
  - 每个价签都没有丢包
  - 或者超过最大发送窗口
- 价签退出模式4的标准
  - 接收时间超过4.2秒
  - 收到sleep帧

其中，**最大发送窗口**表示基站最多可以发送多少次数据包，计算公式为

> rf_work_time * 1000 * 1000 / 750（750：表示基站每发送26字帧长时间为750us）

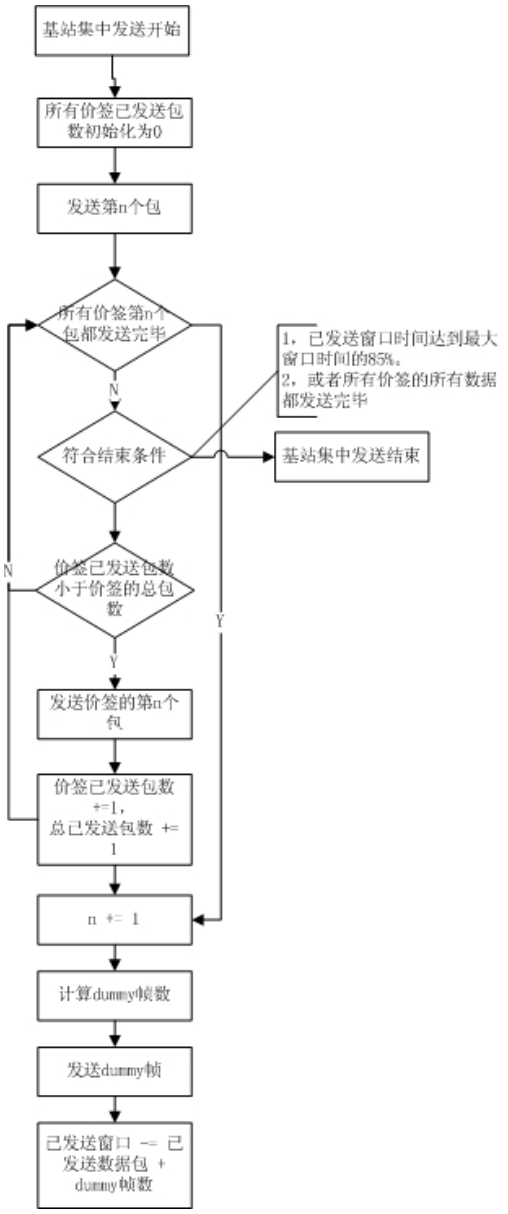
基站-价签处理流程

如下为基站和价签在模式4下的工作流程图



其中几个模块的工作流程介绍如下

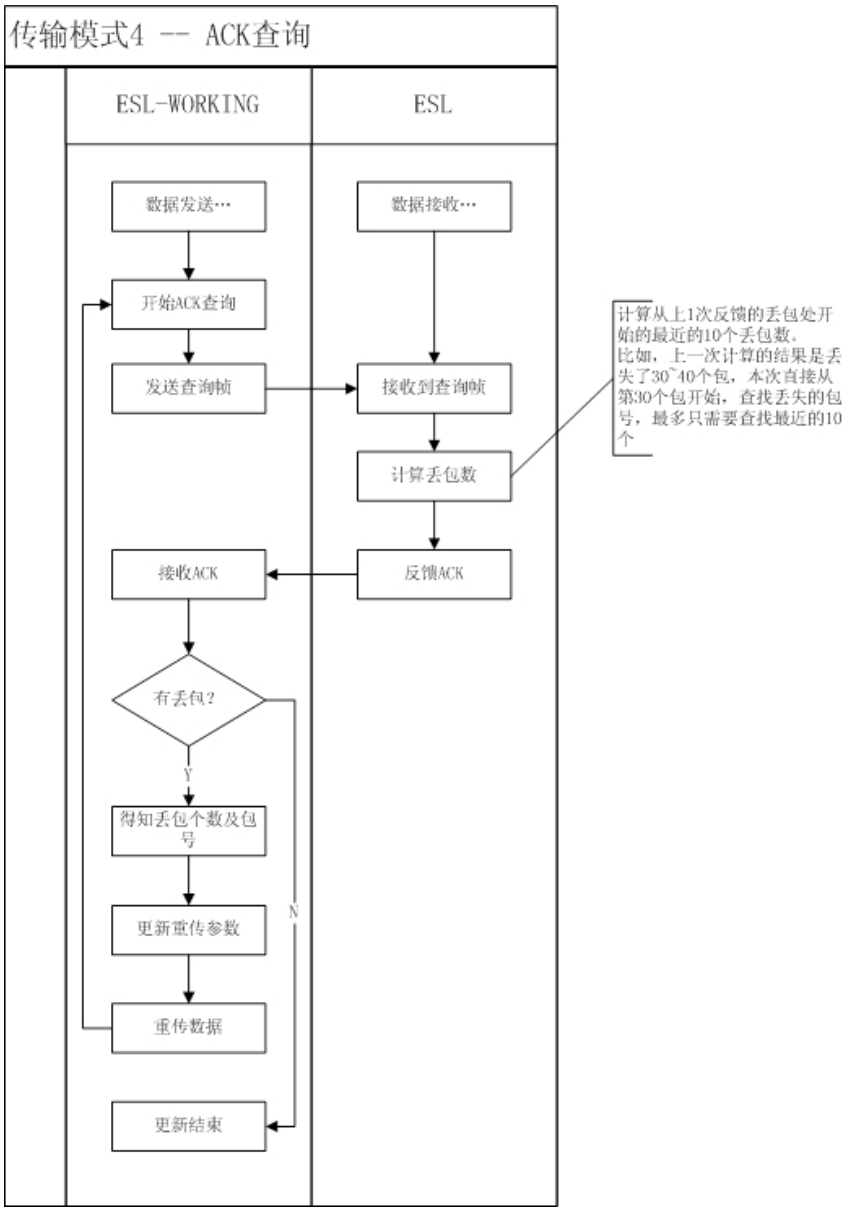
1. 集中下发流程



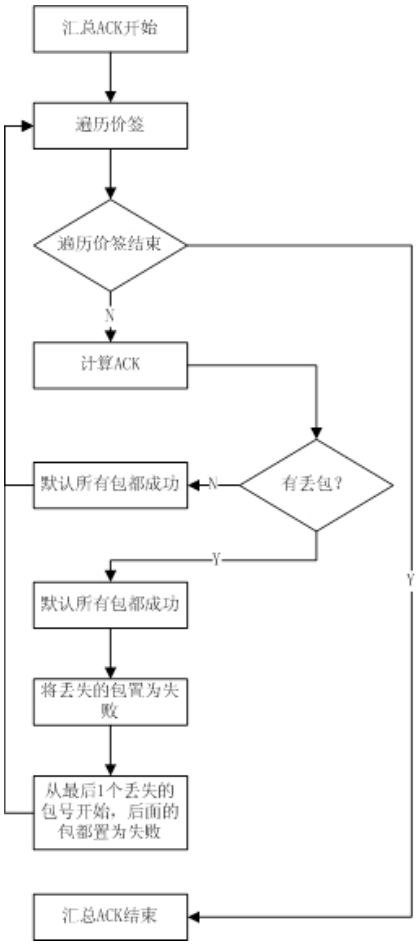
2. 发送dummy帧

    1. 价签每收到1帧数据，都需要进行处理（从RF芯片搬出数据、计算软件crc、保存到外部Flash），约耗时3ms。系统给予50%的富余，既为4.5ms。
    2. 基站每发送一帧的时间为720us，富余值为750us。
    3. 根据以上参数，基站认为1个价签在收到数据后，必须再过5（4.5ms/750us -1）个发送窗口，才能给同1个价签发送下1个数据。
    4. 基站在各个阶段（批量下发数据、发送查询、发送sleep）时都要考虑dummy帧算法。
    5. dummy帧的内容为唤醒帧帧1，主要作用是将误唤醒的价签休眠掉。
    6. 在发送sleep帧时，会强制发送1次dummy帧。

3. ACK查询流程

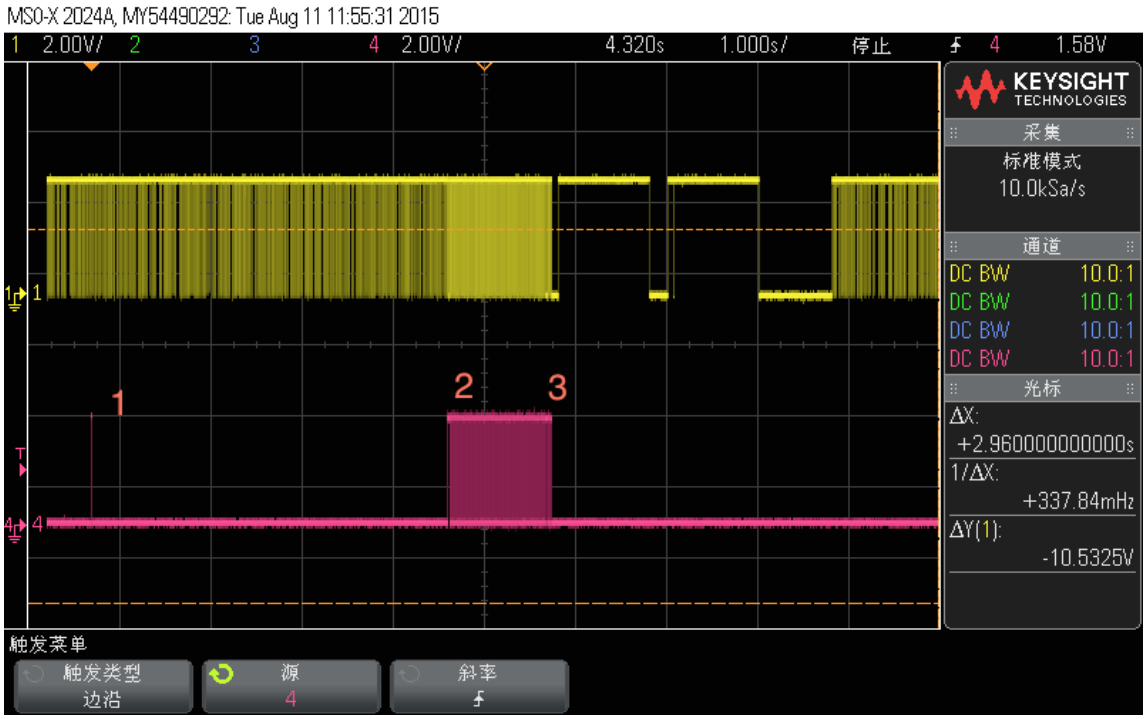

传输模式4 -- ACK查询
ESL-WORKING
ESL
数据发送…
数据接收…
开始ACK查询
发送查询帧
接收到查询帧
计算丢包数
计算从上1次反馈的丢包处开始的最近的10个丢包数。
比如，上一次计算的结果是丢失了30~40个包，本次直接从第30个包开始，查找丢失的包号，最多只需要查找最近的10个
接收ACK
反馈ACK
有丢包?
Y
N
得知丢包个数及包号
更新重传参数
重传数据
更新结束

4. 汇总HTP ACK


汇总ACK开始
遍历价签
遍历价签结束
N
计算ACK
有丢包？
N
默认所有包都成功
Y
默认所有包都成功
将丢失的包置为失败
从最后1个丢失的包号开始，后面的包都置为失败
Y
汇总ACK结束

时序图

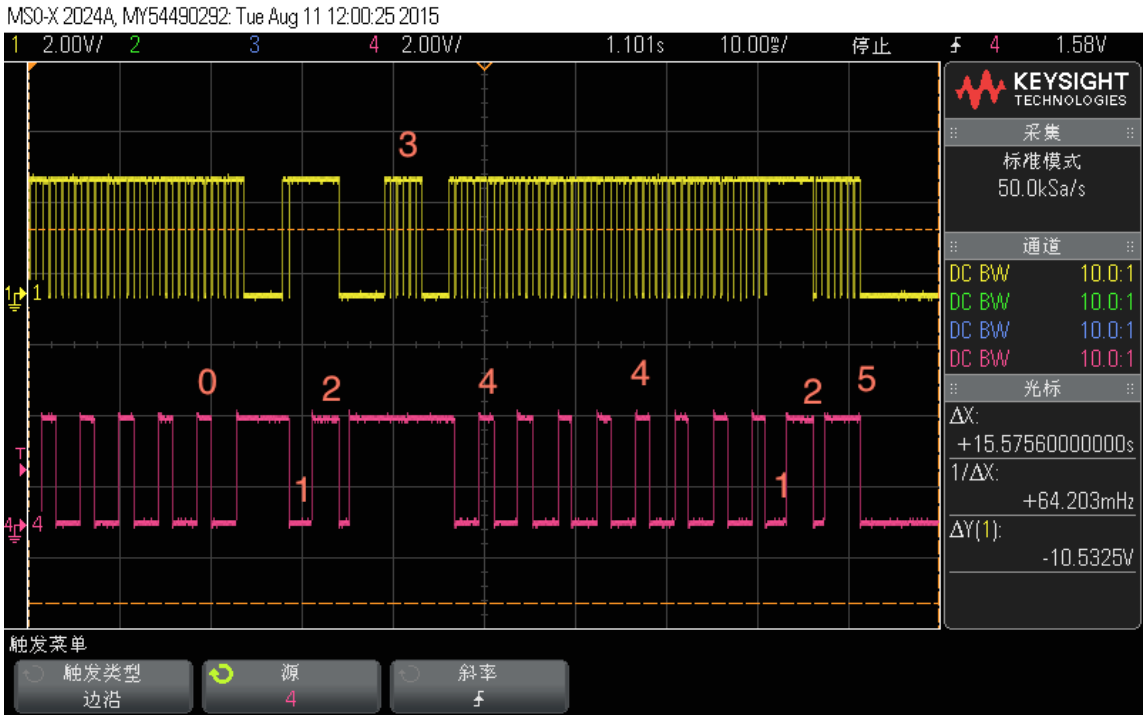
1. 整体时序图



- **1** 唤醒
- **2** 集中下发数据
- **3** 发送查询及sleep

2. 集中下发时序图

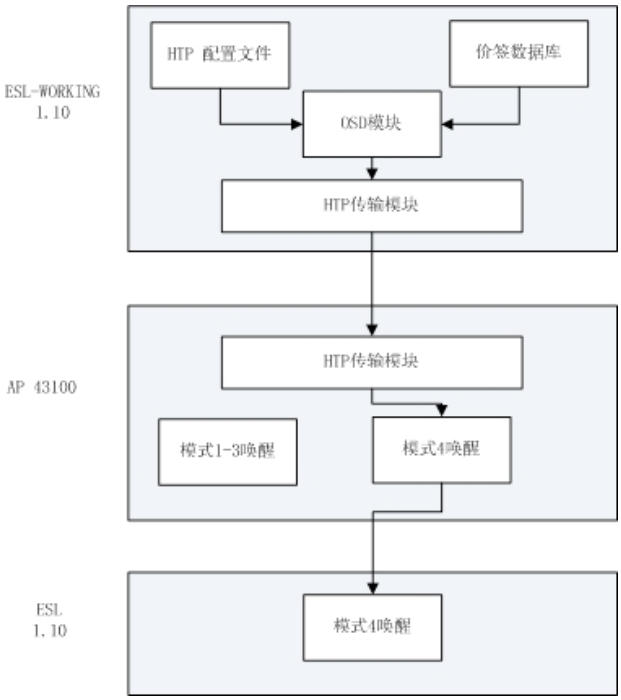
3. 查询时序图



- **0** 集中发送数据帧
- **1** 价签接收到查询帧
- **2** 价签返回查询结果
- **3** 基站发送dummy帧
- **4** 重收数据帧
- **5** 收到sleep帧

HTPV17 低功耗唤醒

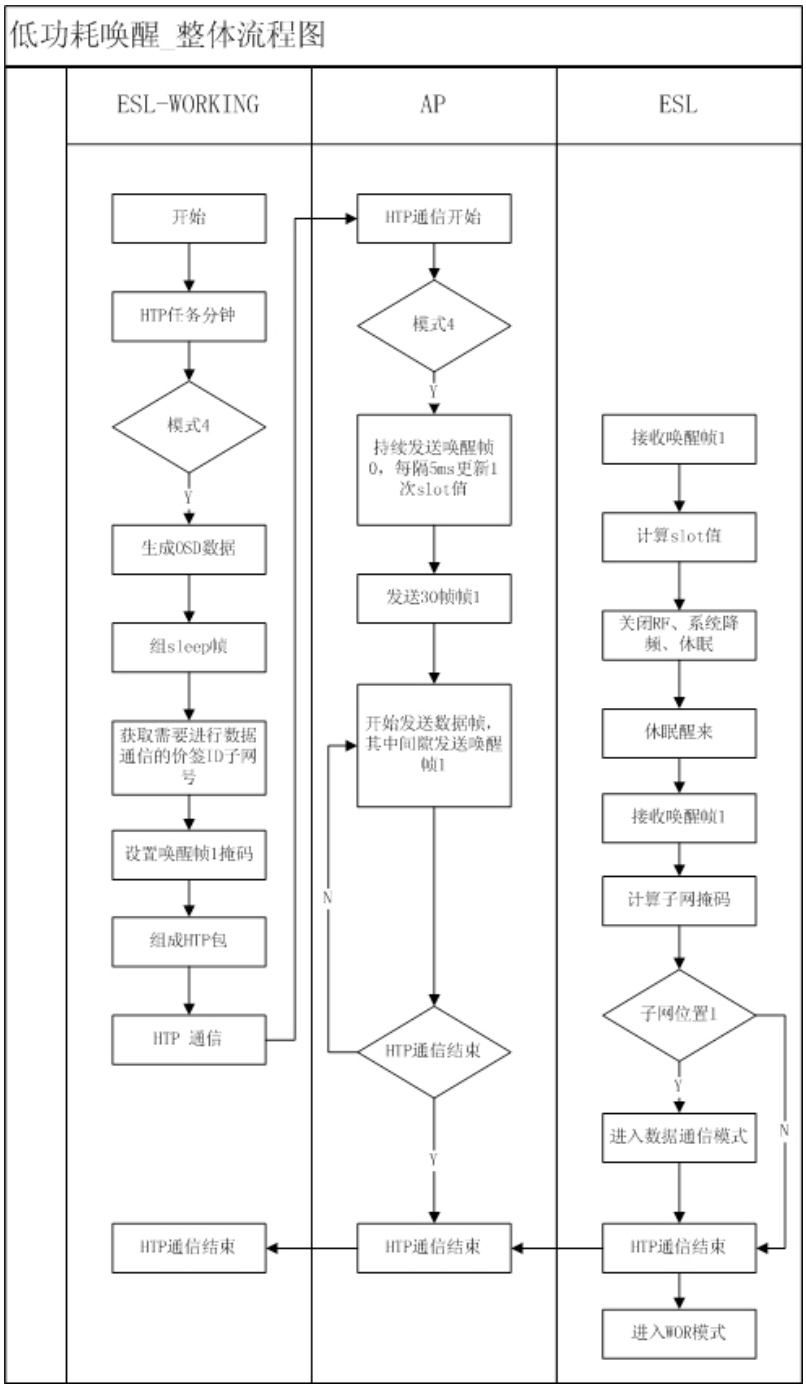
模块框架



数据格式

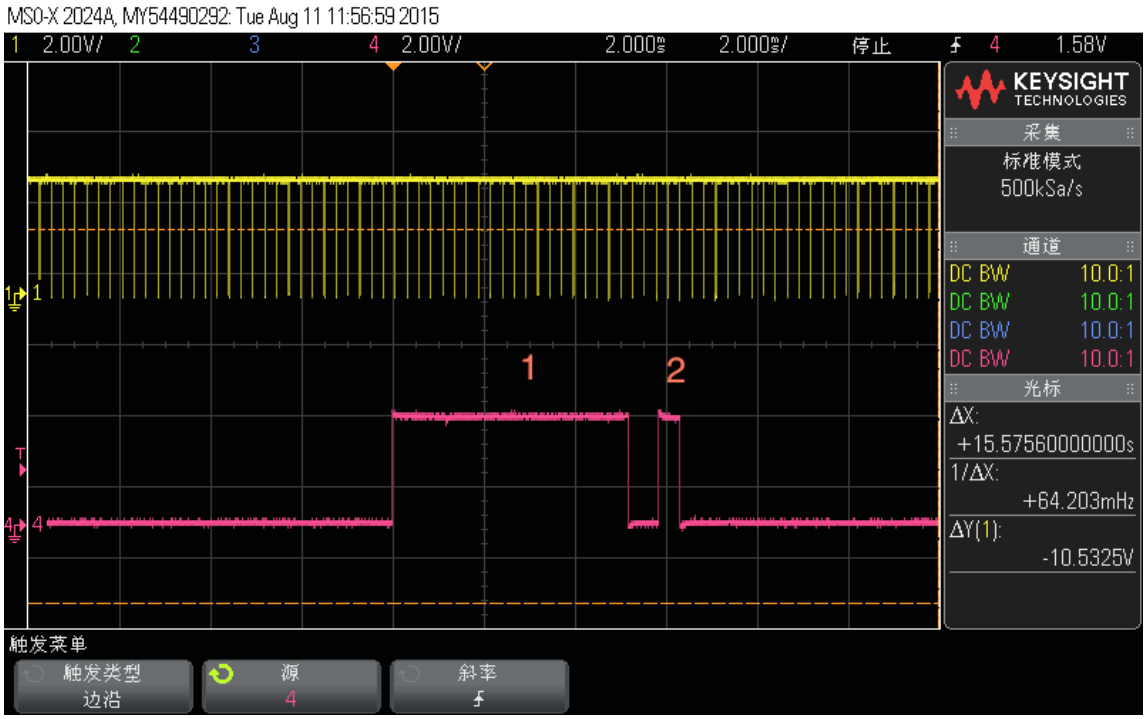
- config/htp_conf.ini
  需要存在键值**HS_EL_5104**， 且其传输模式为4。
- 数据库
  1. 对应op3以及op5中 **screen_type**值都是**HS_EL_5104**。
  2. op2值为价签的子网号。表示价签在所属的组中序号。

业务流程


低功耗唤醒_整体流程图
ESL-WORKING
AP
ESL
开始
HTP任务分钟
模式4
Y
生成OSD数据
组sleep帧
获取需要进行数据通信的价签ID子网号
设置唤醒帧1掩码
组成HTP包
HTP 通信
HTP通信结束
HTP通信开始
模式4
Y
持续发送唤醒帧0，每隔5ms更新1次slot值
发送30帧帧1
开始发送数据帧，其中间隙发送唤醒帧1
N
HTP通信结束
Y
HTP通信结束
接收唤醒帧1
计算slot值
关闭RF、系统降频、休眠
休眠醒来
接收唤醒帧1
计算子网掩码
子网位置1
Y
N
进入数据通信模式
HTP通信结束
进入WOR模式

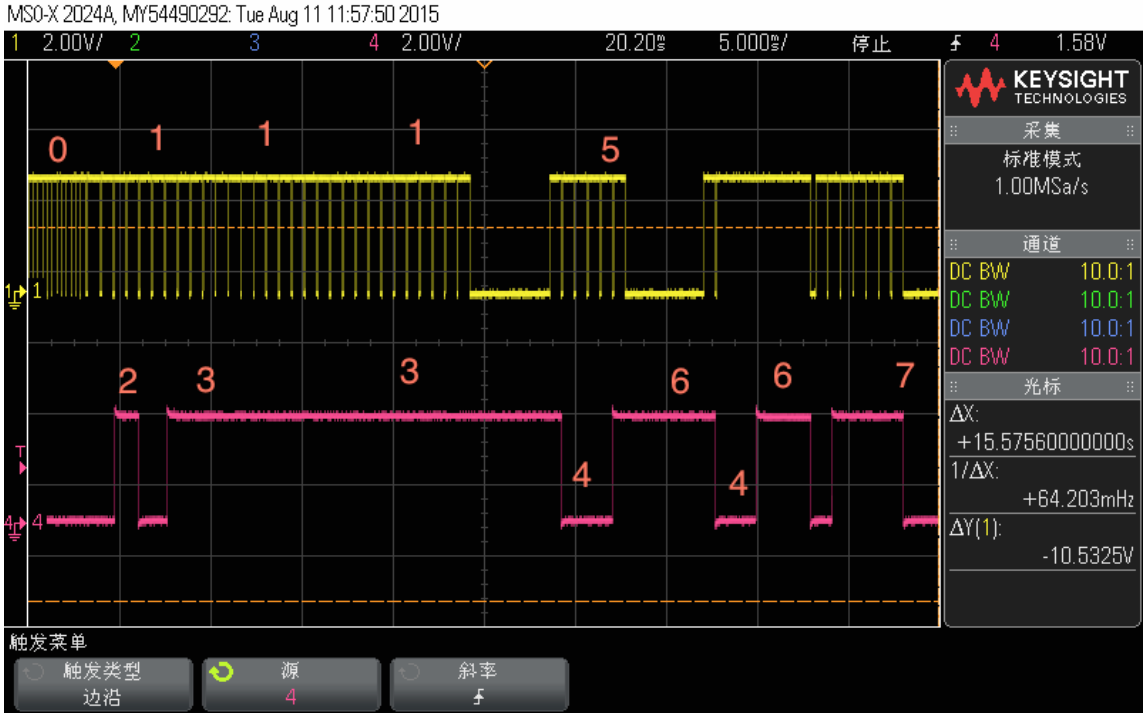
时序图

1. 帧0时序图



- **1** 价签被唤醒
- **2** 价签收到1个唤醒帧

2. 帧1时序图



- 0 基站仍然在发送帧0
- 1 基站开始发送帧1
- 2 价签醒来收到帧1
- 3 价签使用slaveid接收数据包

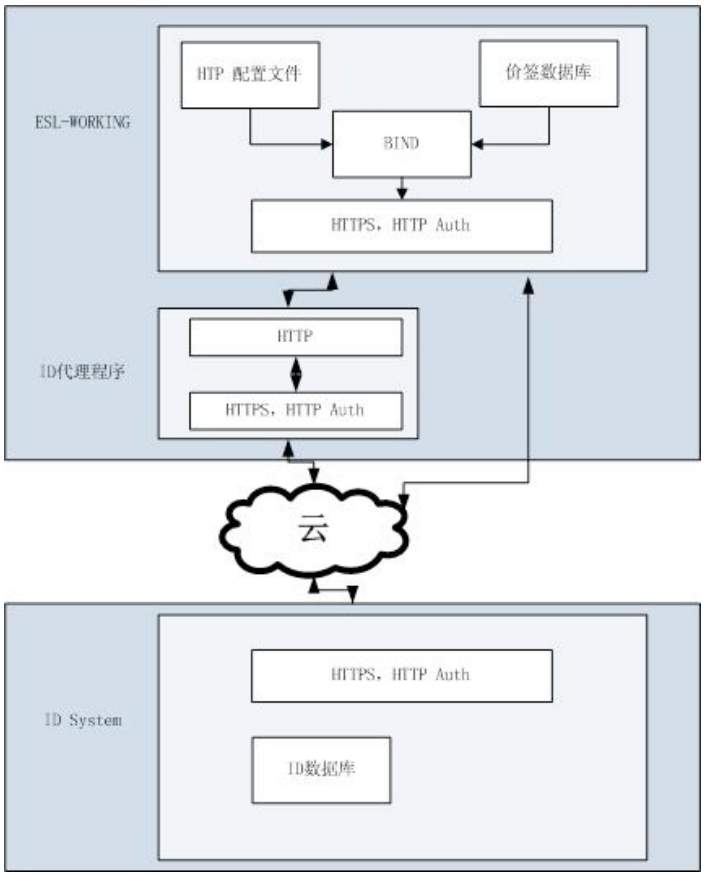
- 4 价签收到数据包（第1个标为4的地方）/查询包（第2个标为4的地方）
- 5 基站发送dummy包
- 6 价签等待下一个数据包(第1个标为6的地方）/回ACK(第2个标为6的地方）
- 7 价签收到sleep帧

价签ID信息查询系统

系统框架

价签ID信息查询系统分为2部分

- 嵌入到esl-working系统内的查询模块。此查询功能将由bind请求激活。当系统收到绑定请求，且发现绑定的价签不在本地数据库中，将激活ID查询功能。
- 位于公网的ID信息查询系统。此系统只对外提供ID查询功能，并使用https和http auth访问机制。
- 但内网的esl-working系统访问不了公网时，可以通过idproxy工具转发请求。此时需要运行idproxy工具的电脑能够同时访问esl-working网络和公网。



数据格式

config/config.ini

```
[idsystem]
```

```
enable = yes
url = https://123.57.225.81:8082
#url = http://192.168.1.100:8082
timeout = 3
username = hanshowidsystem
password = hanshowidsystem
```

在配置文件里新建一个配置项为idsystem，其选项如下

- **enable** 表示是否开启此功能
- **url** ID查询网址。网址分为**https**和**http**两种请求类型，https用于直连公网服务器，需要走https和http登陆验证。http为通过内网代理工具(**test/idproxy.py**)间接访问外网。
- **timeout** 访问服务器超时时间，单位为秒
- **username、password** 访问公网ID服务器的密码

test/idproxy.py

```
......
top_level_url = "https://123.57.225.81:8082"
username = 'hanshowidsystem'
password = 'hanshowidsystem'
......
```

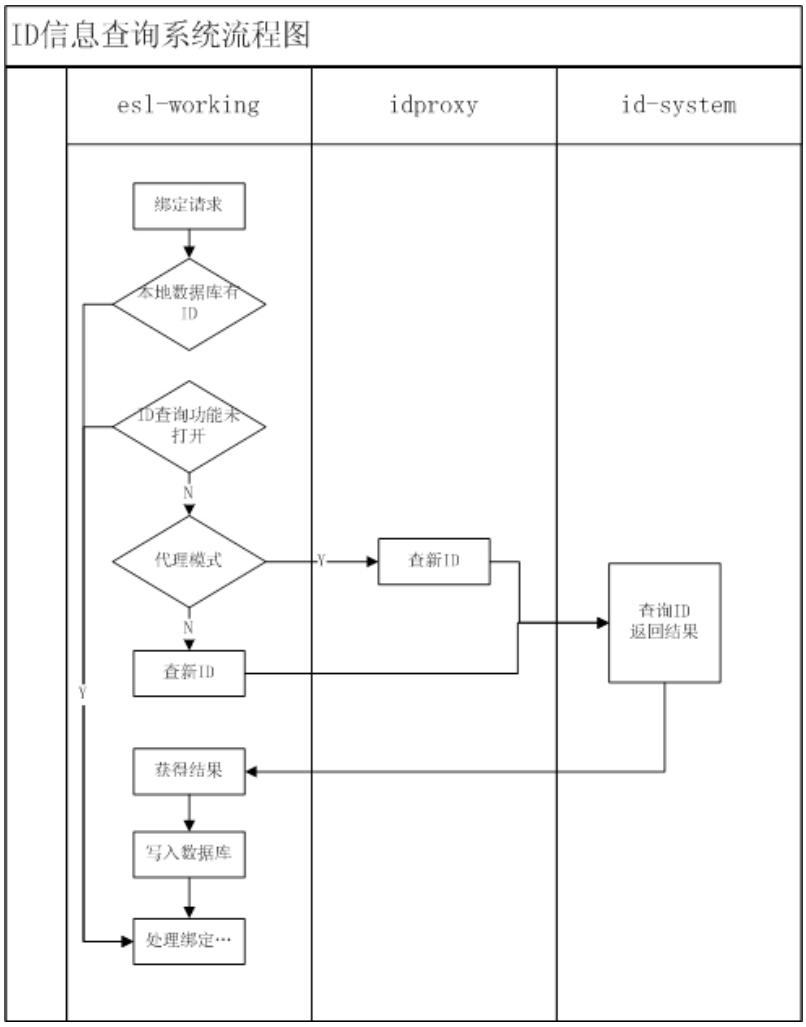
这个文件是ID查询代理脚本，将内网的ID查询请求转发到公网上。
以上三行为公网ID服务器的访问IP和账号。

HTTP API 请求示范

```
wget -q -O - --no-check-certificate
https://hanshowidsystem:hanshowidsystem@123.57.225.81:8082/5A-11-1B-99

 {"nw4": "DOT20", "op4": "ESLID_SN", "eslid": "5A-11-1B-99", "op1": 1, "nw2": "52-56-
78-53", "nw3": 90, "lastworktime": "Fri Jan 09 13:10:06 2015", "op2": 1, "op3":
"HS_EL_5103", "op6": "", "status": "online", "op5": "
{'max_package':512,'screen_type':'HS_EL_5103','version':1,'flash_size':16,'temperature':
'0-10','resolution_x':296,'resolution_y':128,'resolution_direction':0}", "nw1": "58-02-
04-66"}
```

ID信息查询流程图


ID信息查询系统流程图
esl-working
idproxy
id-system
绑定请求
本地数据库有ID
ID查询功能未打开
N
代理模式
Y
查新ID
N
查新ID
Y
查询ID
返回结果
获得结果
写入数据库
处理绑定…