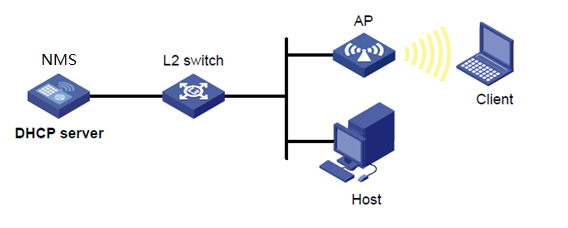
## 网络拓扑：



NMS（网络管理系统）通过二层交换机跟AP互通；

AP、Host通过DHCP获取IP地址，Client通过AP以NAT的方式或者桥接的方式获取IP地址；

NMS与DHCP服务器可以共用硬件平台和固定的IP地址，也可以分开。

## NMS网络管理系统需求：

### AP配置（支持批量修改/获取）

支持对SSID/password的批量修改/获取

支持对Client获取地址方式，即NAT/桥接的批量修改/获取

支持对AP登录名/密码的修改/获取

支持对AP NAT配置情况下地址池的修改/获取

支持对AP的网关地址和网管地址的修改/获取

### AP状态实时监控（支持导出到excel表格）

有方便的界面可以监控AP：在线/离线状态、用户数、型号（wifi4/5/6）、固件版本、天线数量、NAT\桥接配置、SSID、NAT地址池、网关地址等

### AP软件升级（支持批量升级）

有方便的界面可以对AP进行固件升级和降级，支持保留或者不保留原有配置

### AP日志管理（支持批量获取）

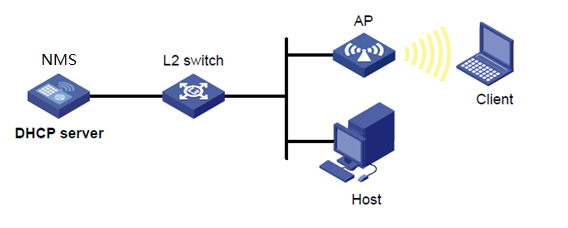
有方便的界面可以对AP的日志进行获取，存储等管理操作

### AP认证

拒绝不合法的AP接入NMS系统

## 软件系统架构：

### 网络拓扑



### 软件模块

AP

Web Server

X86

CentOS/Ubantu

AP Server

AP agent

OpenLinux

CentOS/Ubantu

ARM

X86

WEB SERVER APSERVER AP

#### Web Server：

##### Requirements：

WS0010：从AP Server获得AP的连接和状态信息并显示；

WS0020：根据用户的web指令通过AP Server对AP的配置信息执行下发、获取；

WS0030：根据用户的web指令传送固件到AP，接收并显示固件升级状态；

WS0040：获取AP日志等；

#### AP Server：

##### Requirements：

AS0010：监听从AP发来的连接请求，决定是否接受其连接；

AS0020：接收来自已连接AP的状态信息，发送heart beat到AP，维护跟AP的链接；

AS0030：根据Web Server的指令执行对AP的配置信息下发、获取；

#### AP Agent：

##### Requirements：

AA0010：运行在AP上的网管代理程序，负责发起到网管的连接请求，响应heart beat ，定时发聩AP状态信息，接受处理并反馈AP Server下发的命令等；

AA0020：接收固件，执行固件升级并反馈结果给Web Server；

AA0030：发送AP日志到Web Server；

## NMS网络管理系统开发计划：

### Feature Matric：

Work Load：1（最少）~5（最多）

Difficulty：1（最简单）~5（最难）

Phase：1（最先开发）~3（最后开发）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature | Work Load | Difficulty | Phase | Comments |
| WS0010：从AP Server获得AP的连接和状态信息并显示； | 4 | 3 | 1 |  |
| WS0020：根据用户的web指令通过AP Server对AP的配置信息执行下发、获取； | 4 | 3 | 1 |  |
| WS0030：根据用户的web指令传送固件到AP，接收并显示固件升级状态； | 3 | 3 | 2 |  |
| WS0040：获取AP日志； | 3 | 3 | 3 |  |
| AS0010：监听从AP发来的连接请求，决定是否接受其连接； | 4 | 4 | 1 |  |
| AS0020：接收来自已连接AP的状态信息，发送heart beat到AP，维护跟AP的链接； | 4 | 4 | 1 |  |
| AS0030：根据Web Server的指令执行对AP的配置信息下发、获取； | 4 | 3 | 1 |  |
| AA0010：运行在AP上的网管代理程序，负责发起到网管的连接请求，响应heart beat ，定时发聩AP状态信息，接受处理并反馈AP Server下发的命令等； | 4 | 4 | 1 |  |
| AA0020：接收固件，执行固件升级并反馈结果给Web Server； | 3 | 3 | 2 |  |
| AA0030：发送AP日志到Web Server； | 3 | 3 | 3 |  |

Web Server、AP Server和AP Agent各需一人负责开发。

## 流程设计：

### NMS本地数据库（读写操作由AP Server负责）：

配置模板（模板Index）；

AP（序列号标识）；

AP配置信息：SSID/password；NAT/Bridge；login name/password；IP pool；Gateway

AP状态信息：online/offline；version；users；model；

### Web Server流程：

#### 合法AP数据及 配置模板上报流程：

AP Server上报给Web Server本地数据库的合法AP及其配置状态和 配置模板，如果AP Server的IP地址存在于Web Server的数据库中，Web Server根据AP Server的上报数据更新本地的数据库（有冲突以Web Server的本地数据库为准，并回复AP Server，以Web Server的数据库更新AP Server的数据库），并指示AP Server在线并刷新AP的显示，然后启动Web Server跟AP Server之间的heart beat过程；

#### heart beat流程：

发送heart beat请求到AP Server，并启动定时器（5秒），连续超时次数置0，定时器超时后再次发送heart beat，连续超时次数置1，接收到heart beat后删除定时器，连续超时次数置0，连续超时达到3次AP Server置offline，停止此AP Server的heart beat流程，AP显示灰色，即不可配置状态；

#### 合法AP数据及 配置模板录入流程：

发送命令给AP Server增加/修改/删除AP和 配置模板，并检查AP Server的返回结果；

#### AP配置流程：

发送命令给AP Server配置AP的SSID/password；NAT/Bridge；login name/password；IP pool；Gateway，并检查AP Server的返回结果；

#### AP显示流程：

定期（5秒）发送命令给AP Server获取并显示AP配置（SSID/password；NAT/Bridge；login name/password；IP pool；Gateway）及状态信息（online/offline；version；users；model）；

### AP Server流程：

#### 合法AP数据及 配置模板上报流程：

AP Server上报给Web Server本地数据库的合法AP及其配置状态和 配置模板，等待来自Web Server的heart beat，如果定时器超时时没有得到Web Server的响应，再次上报给Web Server本地数据库的合法AP及其配置状态和 配置模板，启动定时器（时间根据请求次数double）；

#### heart beat流程（Web Server与AP Server之间）：

发送heart beat请求到AP，并启动定时器（3秒），连续超时次数置0，定时器超时后再次发送heart beat，连续超时次数置1，接收到heart beat后删除定时器，连续超时次数置0，连续超时达到3次AP置offline，停止此AP的heart beat流程；

#### AP 链接维护管理流程：

监听指定端口，接收来自AP的请求，通过跟本地数据库的合法AP序列号进行比对来确定AP是否合法，对于合法的AP给予肯定的回应（携带 配置模板信息），对于不合法的AP给予拒绝。对于合法AP，启动定时器（10秒）接收AP的对于肯定回应的Ack，收到Ack后启动heart beat流程，AP置online，否则AP置offline；

#### heart beat流程（AP Server与AP Agent之间）：

发送heart beat请求到AP，并启动定时器（3秒），连续超时次数置0，定时器超时后再次发送heart beat，连续超时次数置1，接收到heart beat后删除定时器，连续超时次数置0，连续超时达到3次AP置offline，停止此AP的heart beat流程；

#### 合法AP数据及 配置模板录入流程：

收到来自Web Server的命令，根据命令（增加/修改/删除AP）操作本地数据库，并返回结果给Web Server；

#### AP配置流程：

收到来自Web Server的命令，根据命令内容，发送配置信息到AP，并等待AP的返回结果，并把结果反馈给Web Server；

#### AP显示流程：

收到来自Web Server的命令，根据命令内容，读取本地数据库相应信息反馈给Web Server；

#### AP状态更新流程：

online状态的AP定期（5秒）发送状态更新给AP Server；【这个流程在实际实现中可以跟heart beat流程合并实现，在AP的heart beat ack中携带状态更新】

### AP Agent 流程：

#### AP 链接维护管理流程：

根据配置的AP Server的地址和端口号信息发送链接请求到AP Server，携带序列号/ version/ model和本地配置，启动定时器（3秒），接收AP Server的反馈，得到肯定的回应后，根据肯定回应中的 配置模板启动配置AP，启动成功后反馈Ack（携带启动成功/失败和用户数信息）给AP Server并开始启动heart beat流程，更新本地信息（AP Server在线）；如果得到是拒绝的回应，更新本地信息（AP Server不在线，显示拒绝原因）；如果定时器超时时没有得到AP Server的响应，再次发送链接请求到AP Server，启动定时器（时间根据请求次数double）；

#### heart beat流程：

反馈Ack后启动此流程，监听指定端口接收来自AP Server的heart beat并即时反馈heart beat ack（对于跟状态更新流程合并的情形，这里需要携带状态信息）；长时间（3次heart beat）没有收到来自AP Server的heart beat请求，更新本地信息（NMS不在线，显示原因为heart beat超时）；

#### AP配置流程：

监听指定端口接收来自NMS的配置命令，根据命令配置AP，并反馈结果给NMS；

#### AP状态更新流程：

定期（5秒）发送状态信息给NMS【这个流程在实际实现中可以跟heart beat流程合并实现，在AP的heart beat ack中携带状态更新】

## 接口数据结构：

### Web Server 🡨-----------------🡪 AP Server消息

#### Web Server -----------------🡪 AP Server消息定义

|  |  |  |
| --- | --- | --- |
| command | size | content |

command：1字节长度，用于区别是哪种操作或其回应，目前定义有：配置模板增、删、改（0）、合法AP修改（1）、AP配置（2）、heart beat req（3）

size：1字节长度，用于标识有多少个AP或者 配置模板

content：size\*（ sizeof（合法AP）/ sizeof（ 模板））

#### AP Server -----------------🡪 Web Server消息定义

|  |  |  |
| --- | --- | --- |
| command | size | content |

command：1字节长度，用于区别是哪种操作或其回应，目前定义有： 配置模板上报（0）AP上报（1）、配置模板增、删、改回应（2）、合法AP修改回应（3）、AP配置回应（4）、heart beat ack（5）

size：1字节长度，用于标识有多少个AP或者 配置模板

content：size\*（sizeof（配置）/ sizeof（合法AP）/ sizeof（ 模板））

### AP Server 🡨-----------------🡪 AP Agent消息

#### AP Server -----------------🡪 AP Agent消息定义

|  |  |
| --- | --- |
| command | content |

command：1字节长度，用于区别是哪种操作，目前定义有：Register Ack【携带 配置模板】（0）、Register Rej【不在合法AP列表里】（1）、AP配置（2）、Heart beat Req（3）

content：sizeof（AP）

#### AP Agent -----------------🡪 AP Server消息定义

|  |  |
| --- | --- |
| command | content |

command：1字节长度，用于区别是哪种操作的回复，目前定义有： Register Req without configuration（0）、Register Req with configuration（1）、Register complete（2）、AP配置OK（3）、AP配置NOK（4）、Heart beat Ack（5）、Heart beat Ack with AP status update（6）

content：sizeof（AP）

### 公共数据结构

const int APSN\_length = 20;

const int ssid\_length = 20;

const int ssid\_psw\_length = 20;

const int login\_length = 20;

const int login\_psw\_length = 20;

const int version\_length = 20;

const int max\_web\_to\_ap = 256;

typedef struct

{

unsigned char c1;

unsigned char c2;

unsigned char c3;

unsigned char c4;

}ip\_address;

typedef struct

{

ip\_address start;

ip\_address end;

}ip\_pool;

typedef struct

{

unsigned char ssid[ssid\_length];

unsigned char ssid\_psw[ssid\_psw\_length];

}ssid;

typedef struct

{

unsigned char login[login\_length];

unsigned char login\_psw[login\_psw\_length];

}login;

typedef struct

{

ip\_address gateway;

ip\_address subnetmask;

}gateway;

typedef struct

{

unsigned char templateIndex;

ssid ap\_ssid;

unsigned char NatOrBridge;//0:bridge 1:nat

ip\_pool pool;

login ap\_login;

gateway ap\_gateway;

unsigned char options;

}Configuration\_template;

typedef struct

{

unsigned char AP\_SN[APSN\_length];

unsigned char templateIndex;

ssid ap\_ssid;

unsigned char NatOrBridge;//0:bridge 1:nat

ip\_pool pool;

login ap\_login;

gateway ap\_gateway;

unsigned char online\_state;//0:offline 1:online

unsigned char user\_number;

unsigned char model;//0:wifi4 1:wifi5 2:wifi6

unsigned char version[version\_length];

unsigned char antenna\_number;

unsigned char options;

}AP;

typedef struct

{

unsigned char commandOrAck;//0:add[ok] 1:del[nok] 2:modify

AP ap;

}legalAP\_operation;//合法AP

typedef struct

{

unsigned char modifyOrStateUpdate; //0:configure[ok] 1:state update[nok]

AP ap;

}AP\_configuration\_statusUpdate; //AP配置、状态更新

typedef struct

{

unsigned char commandOrAck;//0:add[ok] 1:del[nok] 2:modify

Configuration\_template ap\_template;

} Configuration\_template\_operation; // 配置模板

### Web界面

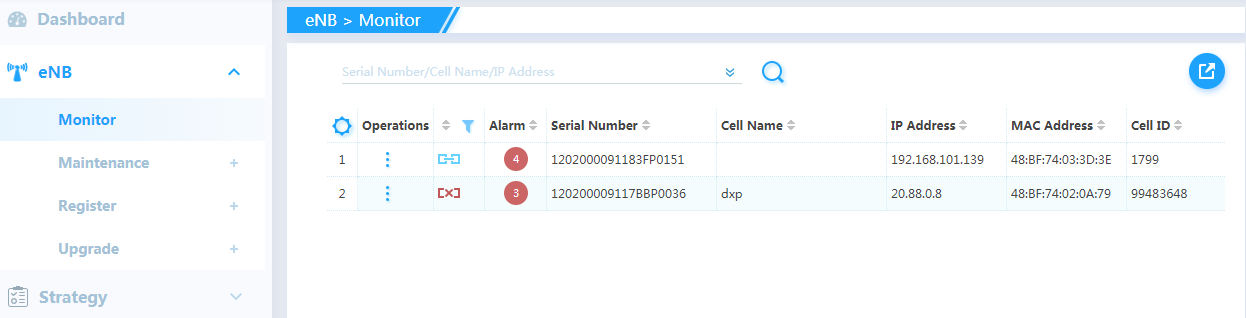
Login界面



首页（监控页面）

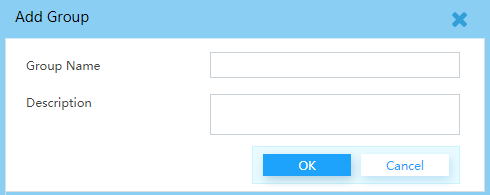
左面是导航栏（第一级:AP Server、Template、APs，第二级：Monitor（修改和删除在这里），增加）

例：



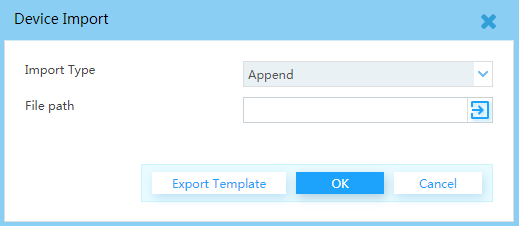
增加/删除/修改AP Server

例（Group改成AP Server，Description改成IP）：



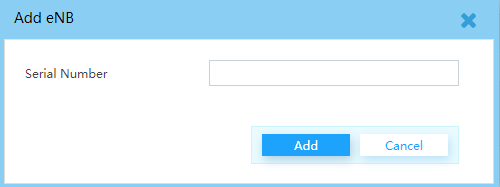
批量增加AP Server、 配置模板、合法AP（从文件导入）

例：



增加/删除/修改合法AP

例：（eNB改成AP）

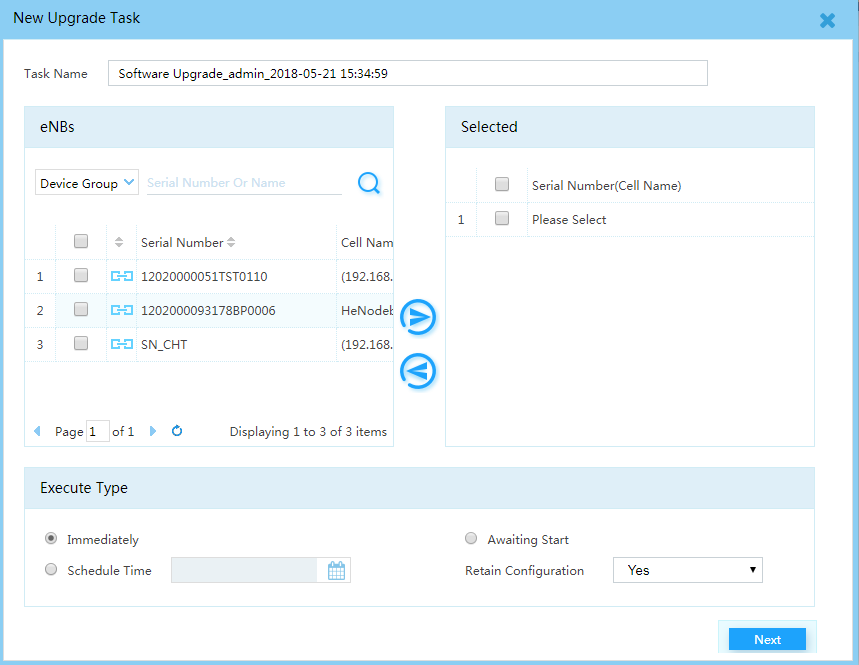


增加/删除/修改配置模板

包含： 配置Index、SSID/passwd、login name/passwd、BridgeOrNat、IP pool、IP gateway

批量配置

例：选择若干AP对应配置或者模板然后执行



单项配置可以直接在Monitor界面直接修改执行

## APServer-Web状态机：

APServer-Web状态：WebOff、WebConnecting、WeOn，其中WebOn有2个子状态：Idle、Configuring（将来可以增加Upgrading和Getting log等）

程序开始APServer-Web状态是WebOff状态，APServer程序根据本地数据库上报模板和合法AP数据给WebServer，进入WebConnecting状态，启动定时器，等待WebServer的heart beat req（3）消息，收到后回复heart beat ack（5），在此消息中携带AP状态，APServer-Web状态进入WebOn（Idle）状态，正常接收回复heart beat，如果定时器（目前设定是3倍heart beat时间，9秒）超时没有收到heart beat，APServer-Web状态进入WebOff状态，启动WebOff定时器；如果APServer在定时器超时没有收到来自WebServer的heart beat req（3）消息，启动WebOff定时器，目前设定是3600秒，定时器超时后继续此过程。

如果APServer在WebOn（Idle）状态收到配置模板增、删、改（0）或合法AP修改（1）消息，更新本地数据库并直接给予回应配置模板增、删、改回应（2）或合法AP修改回应（3）消息，不涉及状态机变化。

如果APServer在WebOn（Idle）状态收到AP配置（2）消息，APServer在检查了AP的合法性和状态后，给AP下发配置，等待AP回应，此时启动定时器，APServer-Web状态进入WebOn（Configuring）状态，等所有AP回应后，或者定时器超时后，APServer发送AP配置回应（4）给WebServer，APServer-Web状态回到WebOn（Idle）状态。

WebOn

## APServer-AP状态机：

AP状态：APOff、APConnecting、APOn，其中APOn有2个子状态：Idle、Configuring（将来可以增加Upgrading和Getting log等）

最初系统开始所有AP都是APOff状态，APServer绑定IP地址和端口等待AP的消息。

APServer收到command=Register Req without configuration（0）时，如果AP序列号在本地数据库内是合法的，给此AP发送消息Register Ack【携带 配置模板】（0），并置AP状态到APConnecting，启动定时器，收到来自此AP的Register complete（2）消息后，AP状态进入APOn（Idle），开始发送heart beat；如果定时器超时时没有收到Register complete（2）消息，AP回到APOff状态。

APServer收到AP的command=Register Req，检测到AP的序列号不在本地数据库时，直接发送Register Rej【不在合法AP列表里】（1）。

APServer收到command=Register Req without configuration（1）时，如果AP序列号在本地数据库内是合法的，说明AP有本地配置，所以直接置AP状态进入APOn（Idle），开始heart beat过程。

对于所有APOn状态下的AP，APServer都需要定期发送heart beat并检测heart beat回应，目前的机制是每3秒钟发送heart beat，连续3次heart beat超时，AP会从APOn（Idle）状态回到APOff状态。

APOn（Configuring）是在APServer在收到WebServer的配置命令，检测AP的合法和状态为APOn（Idle），发送AP配置（2）命令给AP后，启动定时器，等待AP回应的状态。

等到AP的AP配置OK（3）、AP配置NOK（4）或者定时器超时后，AP回到APOn（Idle）状态。

APOn

## WebServer状态机：

APServer状态：APServerOff、APServerConnecting、APServerOn，其中APServerOn有多个子状态：Idle、template modifying、AP modifying、Configuring。

最初系统开始所有APServer都是APServerOff状态，WebServer绑定IP地址和端口等待APServer的消息。

WebServer收到配置模板上报（0）时，存储此APServer的模板数据，并置APServer状态到APConnecting，启动定时器，收到来自此APServer的AP上报（1）消息后，APServer状态进入APServerOn（Idle），开始发送heart beat；如果定时器超时时没有收到APServer上报（1）消息，APServer回到APOff状态。

WebServer收到APServer的配置模板上报（0）和AP上报（1）消息，如果APServer的IP不在本地数据库内，WebServer将丢弃这些消息，记录日志。

对于所有APServerOn状态下的APServer，WebServer都需要定期发送heart beat并检测heart beat回应，目前的机制是每3秒钟发送heart beat，连续3次heart beat超时，APServer会从APServerOn（Idle）状态回到APServerOff状态。

APServerOn（Template modifying）是WebServer的配置命令，检测APServer的状态为APServerOn（Idle），发送配置模板增、删、改（0）命令给APServer后，启动定时器，等待APServer回应的状态。等到配置模板增、删、改回应（2）或者定时器超时后，APServer回到APServerOn（Idle）状态。

APServerOn（AP modifying）是WebServer的配置命令，检测APServer的状态为APServerOn（Idle），发送合法AP修改（1）命令给APServer后，启动定时器，等待APServer回应的状态。等到合法AP修改回应（3）或者定时器超时后，APServer回到APServerOn（Idle）状态。

APServerOn（Configuring）是WebServer的配置命令，检测APServer的状态为APServerOn（Idle），发送AP配置（2）命令给APServer后，启动定时器，等待APServer回应的状态。等到APServer的AP配置回应（4）或者定时器超时后，APServer回到APServer（Idle）状态。

APServerOn

## APAgent状态机：

APAgent状态：ServerOff、ServerConnecting、ServerOn

程序开始APAgent状态是ServerOff状态，APAgent程序根据本地配置的APServer地址数据和配置状态给APServer发送Register Req without configuration（0）【本地无配置数据】或Register Req with configuration（1）【本地有配置数据】，进入WebConnecting状态，启动定时器，等待APServer的Register Ack【携带 配置模板】（0）【如果初始注册无配置】或heart beat req（3）【如果初始注册有配置】消息，收到后等待或回复heart beat ack（5），在此消息中携带AP状态，APAgent状态进入ServerOn状态，正常接收回复heart beat，如果定时器（目前设定是3倍heart beat时间，9秒）超时没有收到heart beat，APAgent状态进入ServerOff状态，启动ServerOff定时器；如果APAgent在定时器超时没有收到来自APServer的heart beat req（3）消息，启动ServerOff定时器，目前设定是3600秒，定时器超时后继续此过程。