江苏省无锡市

震泽路18号

国家软件园双子座A座

电话：86-0510-81156666

传真：86-0510-81156688

感知网络

mac详细设计

**V1.1**

目录

[1 文档说明 3](#_Toc430184492)

[1.1 目的 3](#_Toc430184493)

[1.2 适用范围 3](#_Toc430184494)

[1.3 定义和缩略语 3](#_Toc430184495)

[1.4 参考 3](#_Toc430184496)

[2 系统概述 4](#_Toc430184497)

[2.1 系统说明 4](#_Toc430184498)

[2.2 系统任务 4](#_Toc430184499)

[2.3 系统目标 4](#_Toc430184500)

[2.4 需求规定 4](#_Toc430184501)

[2.4.1 功能需求 4](#_Toc430184502)

[2.4.2 性能需求 4](#_Toc430184503)

[3 总体设计 4](#_Toc430184504)

[3.1 文件结构 4](#_Toc430184505)

[3.2 接口设计 5](#_Toc430184506)

[3.2.1 感知网络mac外部接口 5](#_Toc430184507)

[3.2.2 感知网络mac内部接口 7](#_Toc430184508)

[3.2.3 感知网络核心协议接口 9](#_Toc430184509)

[3.3 运行结构设计 16](#_Toc430184510)

[3.3.1 gznet模块 16](#_Toc430184511)

[3.3.2 sync同步模块 17](#_Toc430184512)

# 文档说明

## 目的

本文档是感知网络概要设计的细化说明，详细阐述了接口功能、设计目的。

## 适用范围

## 定义和缩略语

## 参考

**表格1‑3：参考文档**

|  |  |
| --- | --- |
| **参考文档** | **描述** |
|  |  |
|  |  |

# 系统概述

## 系统说明

## 系统任务

## 系统目标

稳定可靠性、数据传输成功率

## 需求规定

### 功能需求

* 组建一个网络通讯框架，包含3种类型设备
* 同步、异步网络自由选择
* CAP发送关联请求、关联应答
* 网关不分配固定GTS，终端随机挑选GTS发送数据
* 负责同步功能的网关将只作为数据通道，取消关联表功能

### 性能需求

* 设备连接数（待定）
* 时隙占用率（待定）
* 数据传输成功率（>80%）

# 总体设计

## 文件结构

│ mac.c

│ mac.h│

├─core

│ ├─asyn

│ │ asyn.c

│ │ asyn.h

│ │ asyn\_frame.c

│ │ asyn\_frame.h

│ │

│ └─sync

│ sync\_ctrl.c

│ sync\_ctrl.h

│ sync\_define.h

│ sync\_gateway.c

│ sync\_gateway.h

│ sync\_general.c

│ sync\_general.h

│ sync\_mac\_package.c

│ sync\_mac\_package.h

│ sync\_router.c

│ sync\_router.h

│ sync\_terminal.c

│ sync\_terminal.h

│

└─general

assos\_table.c

assos\_table.h

general.c

general.h

mac\_frame.c

mac\_frame.h

mac\_package.c

mac\_package.h

neighbors.c

neighbors.h

## 接口设计

### 感知网络mac外部接口

|  |  |
| --- | --- |
| 文件名 | 功能 |
| mac.c | 1. 提供对外接口 2. 任务的调度 3. 网络协议的切换 |



mac启动流程图

#### 回调函数注册

typedef struct

{

void (\*recv\_cb)(sbuf\_t \*const sbuf);

void (\*send\_cb)(sbuf\_t \*const sbuf, bool\_t state);

void (\*mac\_assoc\_cb)(bool\_t state);

} mac\_dependent\_t;

|  |  |
| --- | --- |
| 函数名 | 功能 |
| recv\_cb | 注册一个接收数据的回调函数 |
| send\_cb | 注册一个发送数据完毕后的回调函数 |
| mac\_assoc\_cb | 注册一个mac关联成功后的回调函数 |
| mac\_dependent\_t | 通过mac\_dependent\_cfg完成注册 |

#### mac流程函数

bool\_t mac\_init(void)

bool\_t mac\_run(void)

bool\_t mac\_stop(void)

bool\_t mac\_send(sbuf\_t \*sbuf)

mac\_info\_t\* mac\_get(void)

|  |  |
| --- | --- |
| 函数名 | 功能 |
| mac\_init | 任务和消息、定时器的注册，只能被调用一次。 |
| mac\_run | 启动mac模块 |
| mac\_stop | 停止mac模块 |
| mac\_send | 像mac发送数据 |
| mac\_get | 获取mac信息，配置mac信息 |

#### 提供给NKW的接口

注：以下接口以inline方式定义在mac.h

static inline void mac\_assoc\_again(uint16\_t addr)

static inline void mac\_dependent\_cfg(mac\_dependent\_t \*mac\_cfg)

static inline void mac\_beacon\_sent\_set(bool\_t state)

static inline uint8\_t mac\_drivce\_type(void)

static inline uint16\_t mac\_self\_saddr(void)

static inline uint16\_t mac\_pib\_coord\_short\_addr\_get(void)

static inline uint16\_t mac\_pib\_hops\_get(void)

static inline bool\_t mac\_assoc\_driver\_find (uint64\_t mac\_addr)

static inline void mac\_assoc\_driver\_remove(uint64\_t mac\_addr,mac\_addr\_mode\_e mode)

|  |  |  |
| --- | --- | --- |
| 函数名 | 功能 | 参数 |
| mac\_assoc\_again | 重新关联 | addr：该地址不再关联 |
| mac\_dependent\_cfg | 注册回调函数 | mac\_dependent\_t：回调函数的结构体定义 |
| mac\_beacon\_sent\_set | 启动mac发送beacon | state:TRUE |
| mac\_drivce\_type | 获取设备类型 | NULL |
| mac\_self\_saddr | 获取设备短地址 | NULL |
| mac\_pib\_coord\_short\_addr\_get | 获取上级设备短地址 | NULL |
| mac\_pib\_hops\_get | 获取网络跳数 | NULL |
| mac\_assoc\_driver\_find | 获取设备网络连接类型(直连OR中转) | mac\_addr：在关联表中能找到的地址为直连设备 |
| mac\_assoc\_driver\_remove | 从关联表中将设备移除 | mac\_addr：地址  mode：地址模式 |

### 感知网络mac内部接口

|  |  |
| --- | --- |
| 文件名 | 功能 |
| assos\_table.c | 关联表设计 |
| general.c | 通用函数设计 |
| mac\_frame.c | mac帧解析 |
| mac\_package.c | mac帧组包 |
| neighbors.c | 邻居表设计 |

#### assos\_table.c

|  |  |  |
| --- | --- | --- |
| 函数名 | 功能 | 参数 |
| assos\_table\_init | 关联表初始化 | NULL |
| assos\_table\_add | 增加关联表节点 | assoc\_req：节点带的关联信息  state：返回的关联状态  cluster\_index：如果是中继器，会返回簇序号  local\_beacon\_map：本地的邻居map  addr：mac地址 |
| assos\_table\_remove | 移除关联表节点 | mac\_addr：地址  mode：地址模式 |
| assoc\_driver\_find | 关联表设备查找 | mac\_addr：在关联表中能找到的地址为直连设备 |
| cluster\_node\_update | 簇节点更新 | cluster\_index:簇序号  src\_addr:要更新的地址 |

#### general.c

|  |  |  |
| --- | --- | --- |
| 函数名 | 功能 | 参数 |
| mac\_short\_addr\_get | 获取mac短地址 | mac\_addr：长地址 |
| get\_addr | 通过地址模式从pbuf中获取地址 | pbuf：数据载荷  mac\_addr\_mode\_e：地址模式  addr：拷贝到的目的地 |

#### mac\_frame.c

|  |  |  |
| --- | --- | --- |
| 函数名 | 功能 | 参数 |
| assoc\_request\_frame | 关联请求解析 | pbuf：载荷  assoc\_req：关联信息 |
| assoc\_response\_frame | 关联应答解析 | pbuf：载荷  mac\_pib：属性域 |

#### mac\_package.c

|  |  |  |
| --- | --- | --- |
| 函数名 | 功能 | 参数 |
| mac\_package\_init | 初始化 |  |
| mac\_ack\_fill\_package | 组ACK帧的mac头 |  |
| mac\_data\_fill\_package | 组数据帧的mac头 |  |
| mac\_assoc\_request\_package | 组关联请求帧 |  |
| mac\_assoc\_respond\_package | 组关联应答帧 |  |

#### neighbors.c

|  |  |  |
| --- | --- | --- |
| 函数名 | 功能 | 参数 |
| mac\_neighbors\_init | 邻居表的初始化，ID为无效值，生存周期为最大值 |  |
| mac\_neighbor\_node\_add | 邻居节点插入、更新 |  |
| mac\_get\_coord | 获取最优邻居节点 | node：邻居节点 |
| mac\_neighbors\_node\_set\_state | 设置邻居节点状态 | id：关联节点  assoc\_state：禁闭状态 |

### 感知网络核心协议接口

#### 同步协议

|  |  |
| --- | --- |
| 文件名 | 功能 |
| sync\_ctrl.c | 同步控制 |
| sync\_gateway.c | 网关 |
| sync\_router.c | 中继器 |
| sync\_terminal.c | 终端 |
| sync\_general.c | 通用接口 |
| sync\_mac\_package.c | 同步组包接口 |

##### sync\_ctrl.c



同步控制模块流程图

1. 提供给mac的回调函数注册

typedef struct

{

void (\*recv\_cb)(sbuf\_t \*const sbuf);

void (\*send\_cb)(sbuf\_t \*const sbuf, bool\_t state);

void (\*mac\_assoc\_cb)(bool\_t state);

void (\*mac\_restart)(net\_mode\_e mode);

} sync\_cb\_t;

2. 提供给mac的接口

struct sync\_t

{

bool\_t (\*run)(void);

bool\_t (\*stop)(void);

bool\_t (\*send)(sbuf\_t \*sbuf);

bool\_t (\*find\_sys\_coord)(void); /\*\*< 寻找上级设备 \*/

void (\*assoc\_switch)(void); /\*\*< 关联状态机 \*/

sync\_info\_t\* (\*get)(void);

};

##### sync\_gateway.c



同步时隙函数关系流程图

1. 提供给sync\_ctrl的回调函数注册

typedef struct

{

void (\*recv\_cb)(sbuf\_t \*const sbuf);

void (\*send\_cb)(sbuf\_t \*const sbuf, bool\_t state);

void (\*mac\_assoc\_cb)(bool\_t state);

} gateway\_cb\_t;

2. 提供给sync\_ctrl的接口

struct gateway\_t

{

bool\_t (\*run)(void);

bool\_t (\*stop)(void);

bool\_t (\*send)(sbuf\_t \*sbuf);

sync\_info\_t\* (\*get)(void);

};

##### sync\_router.c

同步时隙函数关系流程图：参考**sync\_gateway.c**

1. 提供给sync\_ctrl的回调函数注册

typedef struct

{

void (\*recv\_cb)(sbuf\_t \*const sbuf);

void (\*send\_cb)(sbuf\_t \*const sbuf, bool\_t state);

void (\*mac\_assoc\_cb)(bool\_t state);

void (\*mac\_restart)(net\_mode\_e mode);

} router\_cb\_t;

2. 提供给sync\_ctrl的接口

struct router\_t

{

bool\_t (\*run)(void);

bool\_t (\*stop)(void);

bool\_t (\*send)(sbuf\_t \*sbuf);

sync\_info\_t\* (\*get)(void);

};

##### sync\_terminal.c

同步时隙函数关系流程图：参考**sync\_gateway.c**

1. 提供给sync\_ctrl的回调函数注册

typedef struct

{

void (\*recv\_cb)(sbuf\_t \*const sbuf);

void (\*send\_cb)(sbuf\_t \*const sbuf, bool\_t state);

void (\*mac\_assoc\_cb)(bool\_t state);

void (\*mac\_restart)(net\_mode\_e mode);

} terminal\_cb\_t;

2. 提供给sync\_ctrl的接口

struct terminal\_t

{

bool\_t (\*run)(void);

bool\_t (\*stop)(void);

bool\_t (\*send)(sbuf\_t \*sbuf);

sync\_info\_t\* (\*get)(void);

};

##### sync\_general.c

|  |  |
| --- | --- |
| 函数名 | 功能 |
| sync\_config | 同步参数配置 |
| super\_frame\_cfg | 超帧配置 |
| mac\_frames\_cb\_init | 传输模块的回调函数注册 |
| frame\_switch\_init | mac帧类型回调注册 |
| beacon\_recv\_enable | 根据邻居表来判断当前是否要接收beacon |
| self\_coord\_beacon\_recv\_enable | 上级beacon接收时隙判断 |
| self\_beacon\_send\_enable | 自己的beacon时隙发送判断 |
| self\_intra\_gts\_recv\_enable | 自己的簇内时隙接收判断 |
| coord\_intra\_gts\_enable | 上级的簇内时隙判断 |
| self\_coord\_inter\_gts\_enable | 上级的簇间时隙判断 |
| self\_inter\_gts\_enable | 自己的簇间时隙判断 |
| intra\_gts\_range | 上级的簇内时隙范围 |
| inter\_gts\_range | 上级的簇间时隙范围 |
| mac\_beacon\_frame | beacon帧解析 |

##### sync\_mac\_package.c

|  |  |
| --- | --- |
| 函数名 | 功能 |
| mac\_beacon\_package | 组beacon帧 |
| gts\_list\_insert | 插入gts数据队列 |
| gts\_list\_node\_get | 获取上下行数据 |
| gts\_list\_sort | 分配slot序号 |
| gts\_list\_empty | 判断gts数据队列是否为空 |
| cap\_list\_insert | 插入cap 数据队列 |
| cap\_list\_node\_get | 获取cap上行数据 |
| sync\_mac\_package\_init | 初始化 |

#### 异步协议

## 运行结构设计

### gznet模块



### sync同步模块

