**版本历史**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 描述（修改原因） | 作者（修改者） |
| 0.1 | 2015/08/10 | 初稿 | 程刚 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 引言

## 编写目的

本说明书在概要设计的基础上，对网络地址转换模块的逻辑、功能、接口、子模块等分别给出了实现层面上的要求和说明。

软件开发小组的实现成员应该阅读和参考本说明进行代码的编写、测试。

## 项目背景

软件名称：地址转换模块

提出部门：

开发部门：开发一部

完成后的模块可为协议栈网络层的网络地址模块提供NUI与网络短地址的转换以及网络设备的管理。

## 缩写与术语

## 参考资料

互联网上的相关必需资料。

# 总体设计

## 需求概述

网络地址转换模块是整个协议栈网络层路由协议模块的基础，地址转换模块需要提供一下操作：

NUI转换网络短地址：当某个设备入网的时候，需要把8字节唯一的NUI转换为网内唯一短地址，保证网络地址不冲突。

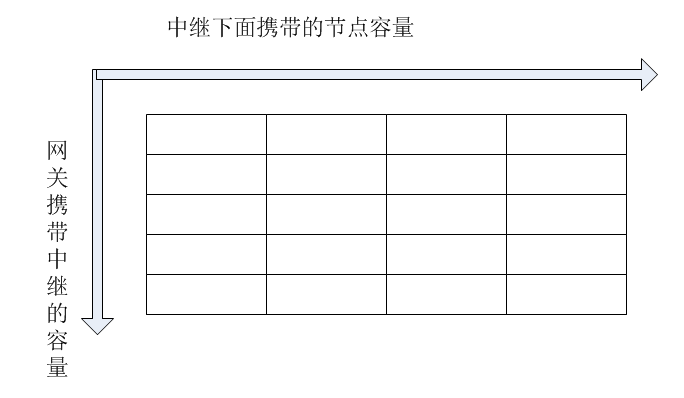
网络地址转换NUI：当网络设备复位以后，网络地址有可能发生变更，需要对发送的目的地址进行转换到NUI进行验证。

网络地址模块更新：当一定时间以后，网络地址模块需要进行自我维护，剔除长时间没有更新的节点，空出资源。

网络节点更新：在一定时间内，节点信息需要刷新存活时间。

## 软件结构

网络地址转换模块由一个二维数据进行维护。



二维数组的第一维用来表示中继的数量，第二维用来表示中继能携带节点的数量。

网络短地址是2字节的地址，高字节用来表示簇间地址，低字节用来表示簇内地址，所有的骨干设备都是簇内第一个地址，这样就可以把第一纬的数组下标来表示中继的簇间地址，所有的终端设备都是连接在骨干设备之下，可以把第二维的数组下标用来表示终端设备的簇内地址。

此网络地址转换模块的设计具有查找速度快、网内地址唯一的优点。

## 模块清单

本模块为单一模块。

# 程序描述

## 数据结构与函数

路由条目数据类型定义：

typedef struct route\_entry

{

uint64\_t nui;

} nwk\_addr\_node\_t;

算法

本模块没有采用自定义算法。

## 程序逻辑

## 接口

网络地址转换模块对外提供六个外部接口，分别为模块初始化、模块反初始化、NUI与网络地址转换、模块更新和网络节点更新。

### 初始化模块

/\*\*

\* @brief 网络地址管理模块的初始化，根据输入的骨干、簇内节点的数量来确定网络最大的容量

\* @note 定义一个二维数组指针用于保存申请到的网络地址资源

\* @param[in] backbone\_num 骨干的最大数量

\* @param[in] cluster\_node\_num 簇内节点最大数量

\* @param[in] live\_cnt 网络地址管理模块内节点的最大存活周期（一次心跳作为一次周期）

\* @return 申请网络地址管理模块初始化成功或失败

\* - 0 初始化成功

\* - -1 申请资源不够

\*/

int8\_t nwk\_addr\_module\_init(uint16\_t backbone\_num, uint16\_t cluster\_node\_num, uint8\_t live\_cnt);

### 反初始化模块

/\*\*

\* @brief 网络地址管理模块的反初始化，释放申请的资源

\* @return 网络地址管理模块反初始化成功或失败

\* - 0 反初始化成功

\* - -1 反初始化失败

\*/

int8\_t nwk\_addr\_module\_deinit(void);

### NUI转换成网络短地址

/\*\*

\* @brief 输入64位长地址，得出16位短地址

\* @note 转换后的地址高8位用于簇头号，低8位用于簇内号

\* @param[in] long\_addr 输入的64位长地址

\* @param[in] req 指向入网请求载荷的结构体指针

\* @param[out] short\_p 指向转换以后的16位短地址缓冲区的指针

\* @return 转换是否成功

\* - 0 转换完成

\* - -1 资源不够

\*/

int8\_t nwk\_addr\_long\_to\_short(uint64\_t long\_addr, nwk\_join\_req\_t \*req, uint16\_t \*short\_p);

### 网络短地址转换成NUI

/\*\*

\* @brief 输入16位短地址以后获取64位的长地址

\* @param[in] short\_addr 输入的网络16位短地址

\* @param[out] long\_p 指向转换以后的64位短地址缓冲区的指针

\* @return 转换是否成功

\* - 0 转换成功

\* - -1 转换失败，当前节点不存在网络中

\*/

int8\_t nwk\_addr\_short\_to\_long(uint16\_t short\_addr, uint64\_t \*long\_p);

### 网络地址转换模块更新

/\*\*

\* @brief 网络地址管理模块更新，把超过最大存活周期还没有刷新的节点剔除掉，把缓冲区资源释放出来

\*/

void nwk\_addr\_module\_update(void);

### 网络节点更新

/\*\*

\* @biref 输入网络的短地址更新网络地址管理模块内节点的存活周期

\* @param 节点是否记录于管理模块中

\* - TRUE 节点记录在管理模块中，并被更新

\* - FALSE 节点不存在管理模块中，无法更新

\*/

bool\_t nwk\_addr\_node\_update(uint16\_t short\_addr);

## 测试要点

无。