

一种基于 IEEE 802.11s 的无线 Mesh 网管理方案

张佳星

(北京邮电大学信息与通信工程学院, 北京 100876)

摘要: 无线 Mesh (Wireless Mesh Network) 网络是一种与传统无线网络不同的新型无线网络技术, 它结构复杂, 目前关于它的管理方案很少。本文中, 一种新型的基于 OpenWrt 系统与 SNMP 协议的 IEEE 802.11s 无线 Mesh 网的管理方案被提出。它涵盖无线 Mesh 网络的节点发现, 拓扑显示, 节点信息配置, 控制 Mesh 终端和以太网设备的接入等。该方案基于 RB433AH 嵌入式开发板, 实现了以 Web 的形式管理无线 Mesh 网络的方案。并且通过了在不少于 15 节点的环境下的功能测试。

关键词: 网络管理; 无线 Mesh 网; IEEE 802.11s; OpenWrt; SNMP

中图分类号: TP393

A Management Solution for Wireless Mesh Networks based on IEEE 802.11s

ZHANG Jiaying

(School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876)

Abstract: Wireless Mesh Network is a new type of wireless network which is different from the traditional wireless technology. The management solution for this kind of network is fairly less due to its complex structure. In this paper, a new management solution is designed and developed based on the IEEE 802.11s protocol, the OpenWrt Linux system and the SNMP protocol. It covers Mesh node discovery, topology display, configuration of the node information, control of the Mesh access and other functions. The solution is developed on RB433AH embedded boards, presented in the form of Web, and passed the functional tests in more than 15 nodes' environment.

Key words: Network Management; Wireless Mesh Network; IEEE 802.11s; OpenWrt; SNMP

0 引言

无线 Mesh 网络^[1]是支持 TCP/IP 协议的无线接入技术, 它同时拥有 WLAN 和 Adhoc 网络的特点, 其架构是由很多点组成的网状结构, 具有组网灵活、自修复、自配置等特点, 是一种高带宽、大容量、网络可拓展性强、覆盖面积大的无线网络。IEEE 802.11s^[2] 是 IEEE 802.11 专为无线 Mesh 网络制定的附加草案, 定义了无线设备组建无线局域 Mesh 网络的方式。

基于 IEEE 802.11s 的无线 Mesh 网络具有分层的网络结构, 包括 Mesh 路由器和 Mesh 客户端。网管既要管理 Mesh 路由器, 也要对接入的移动 Mesh 客户端进行监视与控制, 还要对 Mesh 网关路由器管理, 最后形成分级管理的管理模型。这种分层的网络管理既包含整个网络的节点发现, 拓扑展示, 又包含单个 Mesh 节点相关参数的远程配置, 如 EssID、无线通信协议、信号功率、天线工作频率和信道的配置等。

作者简介: 张佳星(1992-), 男, 硕士研究生, 主要研究方向: 无线 Mesh 网络. E-mail: zhangjiaying@bupt.edu.cn

与传统的网管系统相比, 无线 Mesh 网管系统需要体现 Mesh 网络分层的结构, 而且在无线 Mesh 网络中, 虽然无线 Mesh 节点相对静止或者移动不多, 但无线 Mesh 客户端的移动很高, 从而导致部分拓扑结构的频繁的变化, 网管需要及时发现节点的离开与加入, 从而需要及时更新拓扑结构。

基于以上 Mesh 网管的特点和难点, 我们设计了一套通过 Web 实现 Mesh 网络管理的方式, 只要能够连接进 Mesh 网并且获得管理员权限, 就能方便的对 Mesh 网进行管理。

1 IEEE 802.11s 和 SNMP 简单网络管理协议

1.1 IEEE 802.11s 及其实现

IEEE 802.11s 作为第一个无线 Mesh 工业标准于 2011 年 9 月被提出^[3]。IEEE 802.11s 是 IEEE 802.11 标准的扩展, 它允许多个无线节点相互连接而无需通过使用 AP 接入。作为对现有 802.11 规范的修订, 所有的 Mesh 功能完全集成到现有的 MAC 层, 而底层的物理层则保持不变。

基于 IEEE 802.11s 的无线 Mesh 网络通常由 Mesh 接入节点 (MAP), Mesh 路由节点 (MP) 和 Mesh 网关节点 (MPP) 组成。其中, MAP 可以由笔记本电脑, 手机和其他嵌入式无线设备组成, 它既具有普通 AP 的接入功能, 又能够作为 Mesh 网络节点的一部分进行数据的转发^[4], 其结构如图 1 所示:

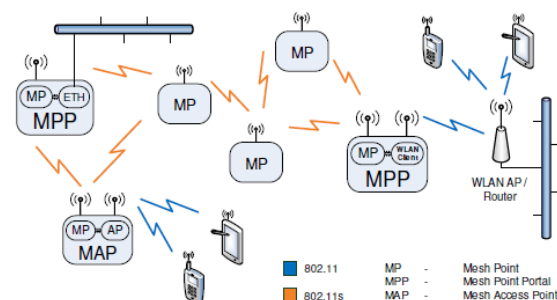


Fig. 1 The structure of wireless mesh network

OpenWrt 是适合于嵌入式设备的一个 Linux 发行版^[5]。相对底层 Linux 而言, OpenWrt 不是一个单一、静态的固件, 而是提供了一个可添加软件包的可写的文件系统。这使用户可以自由选择应用程序和配置, 而不必受设备提供商的限制, 并且可以使用一些适合某方面应用的软件包来定制设备。

Open802.11s 项目是一个 Linux 平台下的 802.11s 解决方案。它目前是 IEEE 802.11s 最新的解决方案, 且能够满足大部分 802.11s 标准的要求。OpenWrt 系统本身集成 Open802.11s 方案, 因此他能方便的实现 802.11s 协议。

OpenWrt 系统不仅能方便的实现 IEEE 802.11s 协议, 而且它还提供了配置集中化管理的解决方案 LuCI (Lua Configuration Interface), 这也为 Mesh 网管理提供了方便。

1.2 SNMP 简单网络管理协议及其实现

SNMP (Simple Network Management Protocol, 简单网络管理协议)^[6]提供了底层网络的管理框架, 网络管理软件不用管底层如何实现, 通过 SNMP 命令就可以对网络进行管理。SNMP 管理模型由三个基本部分构成: SNMP 管理端 (Manager)、SNMP 代理端 (agent) 以及管理信息库 (MIB)。SNMP 管理端通常存在于单个网络设备或共享网络设备中, 网络

管理员通过 SNMP 管理端发送相应命令监控被管设备，代理端接收到命令后，通过查询 MIB 中对应实体获得管理信息，然后将信息反馈到管理端。管理人员可直接通过 SNMP 相关操作同代理端之间进行数据通信，获取被管设备的实时信息，或者对网络中的设备进行配置等。其结构如图 2 所示：

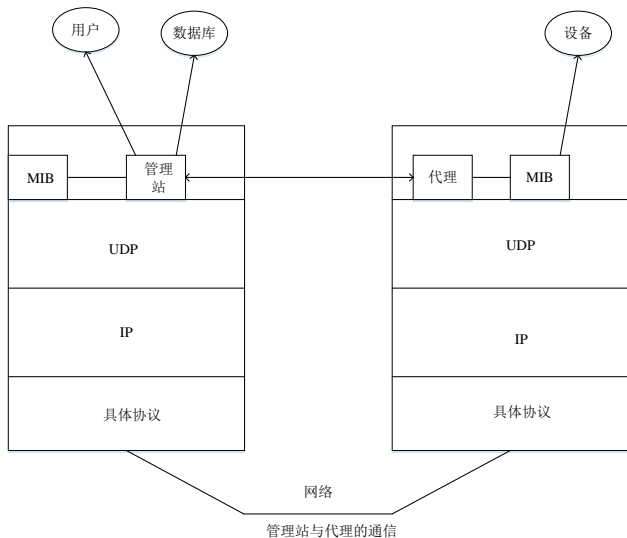


图 2 SNMP 简单网络管理协议结构
Fig. 2 The structure of SNMP

Net-SNMP 是一种开放源代码的简单网络管理协议软件。它是 SNMP 协议的实现方式之一，它支持许多 UNIX 发行版。OpenWrt 系统提供了基于 NET-SNMP 实现的软件包，可以方便运行 SNMP 协议。

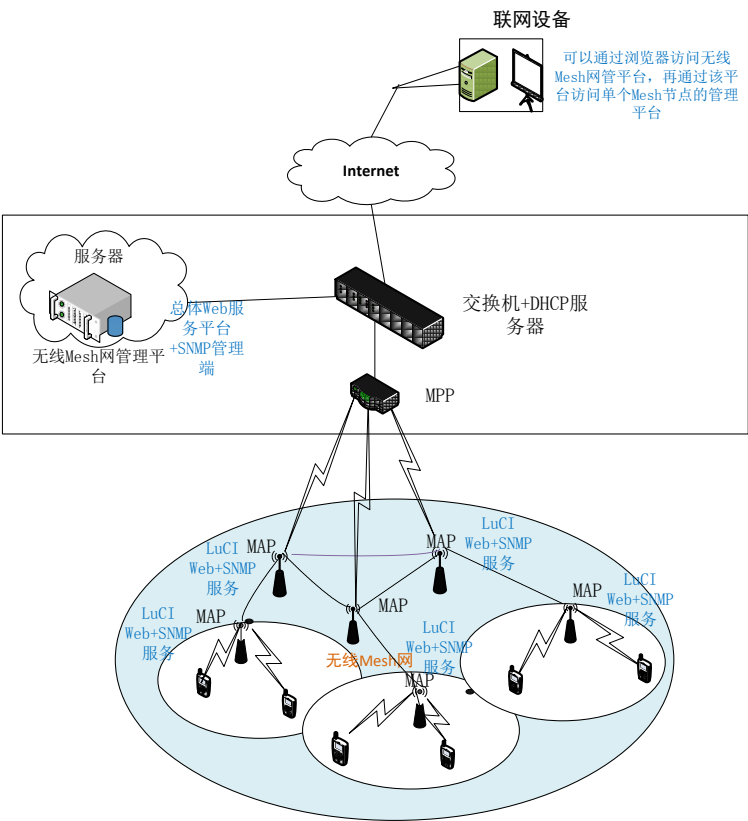
2 基于 IEEE 802.11s 的 Mesh 网管解决方案

针对以上 Mesh 网分层的特点，拓扑发现与管理难，远程配置 Mesh 节点信息难^[7]的特点，提出基于 OpenWrt 系统的分布式无线 Mesh 网络管理的解决方案。如果仅用 SNMP 协议去实现 Mesh 网的管理，在网络的整体管理方面比较适用，但是对于单个节点的庞大的细节配置管理项来说就显得非常复杂了。因此我们使用了 OpenWrt 基于 LuCI 的 web 交互界面实现单个节点的复杂配置。

2.1 分布式 Mesh 网络管理的整体架构设计

一个简单的网络管理系统管理实体部分主要包括被管设备和管理工作站。针对无线 Mesh 网络的管理，被管设备主要是 Mesh 节点，管理工作站主要是网管服务器。OpenWrt 提供了配置集中化管理的解决方案 LuCI，结合 SNMP 简单网络管理协议，设计了分布式的无线 Mesh 网的管理平台。无线 Mesh 网络常用于远程灾害救援中快速组网，我们依据这样的场景，来介绍该方案的整体架构，如图 3 所示。

其中每个 Mesh 节点安装 OpenWrt 系统和根据需求定制的 LuCI 平台。同时每个节点实现 SNMP 及编写的 SNMP 子代理程序，服务器端实现 PHP 后台加上前端的 Web 管理平台，并将每个子节点的访问路径嵌入该系统，这样就实现了分布式的 Web 管理平台，也就是说管理的任务并不是全部集中于管理服务器，而是分发到每一个节点，一个节点既是被管设备，同时也提供了自我管理的服务。服务器端提供了一个统一的管理接口。



105

图3 Mesh网管理整体架构

Fig. 3 The whole structure of the mesh management system

110

Mesh网管功能主要分为两个大的部分，类似节点发现，节点信息展示，节点拓扑发现和管理这些偏向于整体管理架构的服务被放在了管理服务器端；无线Mesh网络中的网关路由器和普通路由器的参数的远程配置如EssID、无线通信协议、信号功率、天线工作频率或信道的配置这些与每个节点关联的功能就分发到了每个Mesh节点自身的基于OpenWrt的LuCI web管理平台。如图4所示：

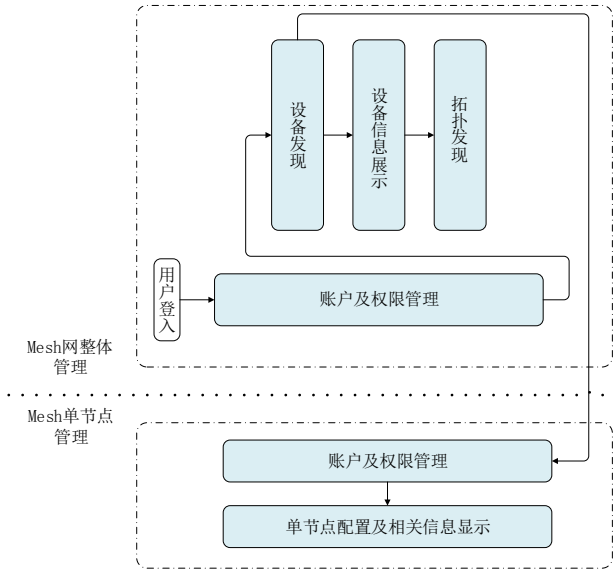


图4 Mesh网管功能模块

Fig. 4 The modules of mesh management system

2.2 基于 OpenWrt 的 LuCI 的 Mesh 网单节点管理

OpenWrt 默认使用 LuCI 作为 Web 交互界面^[8]。所谓的 LuCI 就是 Lua 和 UCI 的结合。Lua 是一个开源的可扩展的轻量级脚本语言。UCI 是"Unified Configuration Interface"(统一配置界面)的缩写，意在 OpenWrt 整个系统的配置集中化。OpenWrt 系统提供了统一的文件配置管理，也就是说所有的配置文件遵循相同的语法格式，只要按照这个格式修改相应的配置文件，重启相应的服务，那么更改的项目就会马上生效。

OpenWrt 的所有配置文件皆位于/etc/config/目录下，如表 1 所示。每个文件大致与它所配置的那部分系统相关。可用文本编辑器、UCI 命令行实用程序或各种编程 API(比如 Shell, Lua 和 C)来编辑和修改这些配置文件。

表 1 OpenWrt 的配置文件
Tab. 1 The configuration files of OpenWrt

基本配置	
/etc/config/dhcp	dnsmasq 和 DHCP 的配置
/etc/config/dropbear	SSH 服务端选项
/etc/config/firewall	中央防火墙配置
/etc/config/network	交换，接口和路由配置
/etc/config/system	系统配置
/etc/config/timeserver	rdate 的时间服务器列表
/etc/config/wireless	无线设置和无线网络的定义

LuCI 使用的是面向对象的库和模板，确保了高效的执行，轻量的安装体积，更快的执行速度以及最重要的是有更好的可维护性。与此同时，LuCI 从 MVC-Webframework 衍生出一个包含了很多库、程序以及 Lua 程序用户接口的集合，但是 LuCI 仍然专注于实现网页用户界面。uhttpd 是一个简单的 Web 服务器程序，主要就是 CGI (Common Gateway Interface, 通用网关接口) 的处理，OpenWrt 是利用 uhttpd 作为 Web 服务器，实现客户端 Web 页面配置功能。对于 request 处理方式，采用的是 CGI，而所用的 CGI 程序就是 LuCI。

根据以上原理，我们需要添加一个所需的管理模块的步骤为如图 5 所示：

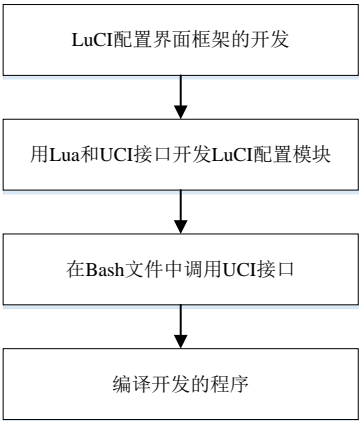


图 5 OpenWrt 的 Web 模块开发流程

Fig. 5 The procedures of developing OpenWrt Web module

实际上 OpenWrt 提供的 LuCI 管理平台原理上是提供了一个直接操作 UCI 统一配置接口文件的方法，利用这一手段我们可以方便的实现 Mesh 节点相关参数的远程配置，如 EssID、无线通信协议、信号功率、天线工作频率或信道的配置等。

2.3 基于 SNMP 和 Web 的 Mesh 网整体管理

基于 SNMP 和 Web 的 Mesh 网整体管理的实现依赖于被管理的节点上实现 SNMP 及其

子代理协议，以及管理端的 Web 功能模块设计。

2.3.1 基于 SNMP 的 Mesh 网管的被管端设计

OpenWrt 系统上使用 SNMP 协议可以获取无线 Mesh 网络设备的部分参数信息，比如节点设备的 CPU 利用率、进程、内存使用情况等。然而，涉及到无线 Mesh 网络的很多其他参数信息，传统的 SNMP MIB-II 并没有给予定义，例如基于 IEEE 802.11s 的无线 Mesh 协议的下一跳的信息。基于这种原因，我们要扩展 SNMP 的 MIB 库，我们使用了 Net-SNMP 的子代理协议去实现 MIB 库的扩展。其原理如下图 6：

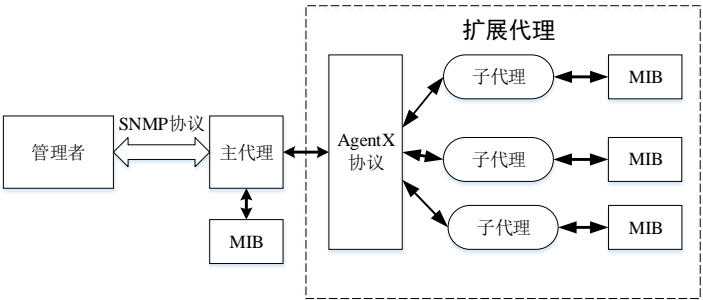


图 6 SNMP 的子代理架构

Fig. 6 The Structure of SNMP Subagent

Mesh 节点上运行着 SNMP 主代理和按需求编写的子代理，AgentX 为实现扩展代理的协议，子代理负责收集 Mesh 网络节点设备上自己负责 MIB 区域的信息，当主代理 SNMP 请求消息到来时，子代理解析该请求消息，查看请求消息是否在自己负责的 MIB 区域，如果是，生成 SNMP 应答报文，并将 SNMP 应答报文发送给主代理。当 Mesh 网络节点设备信息出现故障时，子代理及时向主代理发送告警信息。

子代理程序实际上是一段可与主代理通信的可执行程序，这个可执行程序获取所需要的信息。获取信息的方式可以通过读取包含所需信息的文件，也可以是执行 Linux 下的命令来获取。Net-snmp 软件提供的是依据 C 语言的子代理框架，根据该框架编写子代理编译成可执行文件即可。

每种信息可由管理信息库 MIB 中的对象标识符 OID（Object Identifier）来唯一指定，例如我们需要的关于节点下一跳信息的内容可以根据如下表 2 的方式指定：

表 2 私有 MIB 中的 OID 定义
Tab. 2 The definition of OID in private MIB

MIB 对象	OID	类型	含义
wlanMacAddr	.1.3.6.1.4.1.8072 .2.4.1.1.3	string	无线网卡物理地址
destMacAddr	.1.3.6.1.4.1.8072 .2.4.1.1.5	string	下一跳节点物理地址

2.3.2 基于 Web 的 Mesh 网管的管理端设计

基于 Web 的 Mesh 网管的管理端设计主要由软件架构设计与功能模块设计两个方面。从 Web 服务的角度看，软件架构分为三个主要模块：数据采集与存储模块、无线 Mesh 网络管理模块以及 Web 服务模块。其中数据存储与采集模块主要负责管理端利用 SNMP 协议去收集节点的相关信息，利用数据库中相应的数据表保存这些信息。Web 服务模块包括前端和后台模块，实现浏览器访问与交互。Mesh 网管理模块主要负责管理功能的实现，他负责实现所有的功能模块，是无线 Mesh 网络管理的核心点与主体，包括设备发现、节点信息

管理、拓扑发现等功能。其整体架构如图 7 所示：

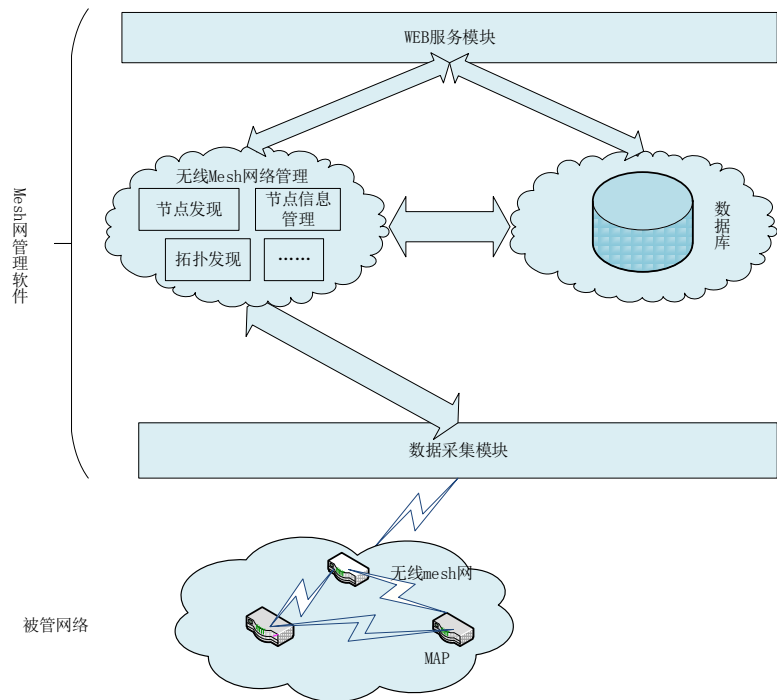


图 7 基于 Web 的无线 Mesh 网管管理端架构

175 Fig. 7 The structure of mesh management system based on Web

设备发现与节点信息管理主要是依赖于 PING 与 SNMP 相结合的方式，浏览器端指定一个 IP 地址段，服务器端接收客户端浏览器传递的起始 IP 地址，按照 IP 地址大小，PING 每一个 IP，若能 PING 通，判断该设备是否支持 SNMP，若支持，则发送 SNMP 命令，利用 SNMP 代理端获取信息，若不支持，则记录其 IP 地址信息。将获得的信息存入服务器的数据库的表中，直到所有的 IP 地址都轮询完成。处理过程如图 8 所示：

180 图 8 网管软件处理流程

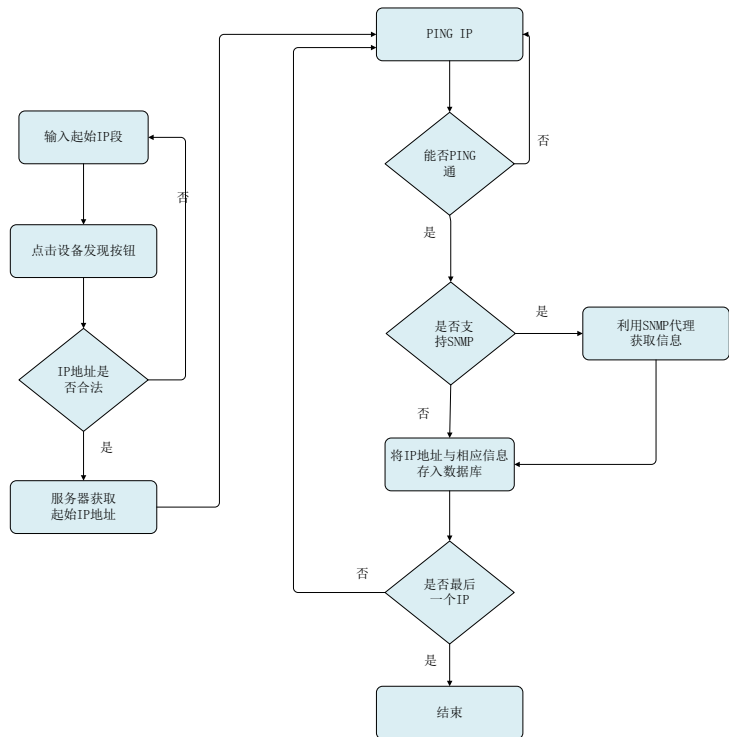


图 8 网管软件处理流程

Fig. 8 The working procedure of the mesh management system

对于拓扑发现模块，依赖于前面的节点发现模块的实现，也依赖于前面被管子代理的设计。在子代理的设计中，我们实现了关于 Mesh 网每个节点下一跳信息获取的扩展。在前面图中将 IP 地址与相应的信息存入数据库的过程中，我们可以存下该 IP 地址对应节点的类型和与他相连的所有节点的地址，其数据结构如图 9 所示：

Node
+ipAddress:节点IP地址 +macAddress:节点Mac地址 +processFlag:节点是否被处理 +type:节点类型是STA还是Mesh节点 +connectNode[]:节点的下一跳
+getNodeMacAddress():获取节点Mac地址 +getConnectNode():获取节点路由信息 +type():判断节点类型

图 9 节点的数据结构

Fig. 9 The data structure of the node

维护一个 Node 类型的数组，包含所有的节点，每个 Node 节点里面维护一个与之关联的下一跳节点，这种表示类似于图的邻接表表示。

依赖以上的数据结构设计，我们设计出拓扑发现的算法，具体步骤为：

① 利用 PING 工具获取所有的节点 IP 地址，利用这些 IP 地址实例化所有的 Node（包含 Mesh 和 STA），将所有的 Node 存储在一个 NodeArray 数组中。

② 遍历所有数组中的 Node，针对每个 Node，利用 SNMP 协议获取节点的 Mac 信息，节点类型信息（Mesh 节点还是 STA），获取每个节点的下一跳信息，即执行每个 Node 方法里面的 getNodeMacAddress()方法，type()方法，和 getConnectNode()方法。这样就可以更新所有数组节点的相应信息。

③这里的 NodeArray 类似于图的邻接表，NodeArray 中的每个 Node 包含了该 Node 本身的信息和与之相关的所有 Node，根据图论算法，利用这些信息，可以恢复出图结构，方便的绘制出整个拓扑图。

3 功能实现与测试分析

我们根据以上的设计，搭建了无线 Mesh 网络平台^[9]和无线 Mesh 网络管理平台，并依据相应的方法，对该网络管理平台进行了测试与分析。

3.1 测试平台搭建

测试平台包括 Mesh 节点平台（被管端），和 Mesh 管理端平台。其中所有 Mesh 节点（包括 MPP，MP 和 MAP）硬件部分采用的是 RouterBoard 433AH 嵌入式开发板。Mesh 节点软件平台包括：所使用的嵌入式操作系统为 OpenWrt 系统，节点上安装 Net-SNMP 软件和自编译的子代理程序，而且安装自定义的 LuCI Web 管理平台。Mesh 管理端主要为安装 Ubuntu 12.04 的服务器，其中管理端软件系统后端采用 PHP 和 Mysql，前端采用 EasyUI 的前端框架。

我们搭建了若干节点的无线 Mesh 网络，我们一共部署了 3 个 Mesh 网关节点 MPP，接入一个服务器，该服务器提供 DHCP 功能，同时该服务器也提供 Mesh 网管的功能，Mesh

网的节点部署在 2.4Ghz 频段，为了减少同频干扰，节点分别采用了垂直的 1,6,11 信道。其中有一套 Mesh 设备部署在室内，两套部署在室外，每套包含 5 到 6 个 Mesh 节点，具体如图 10 所示：

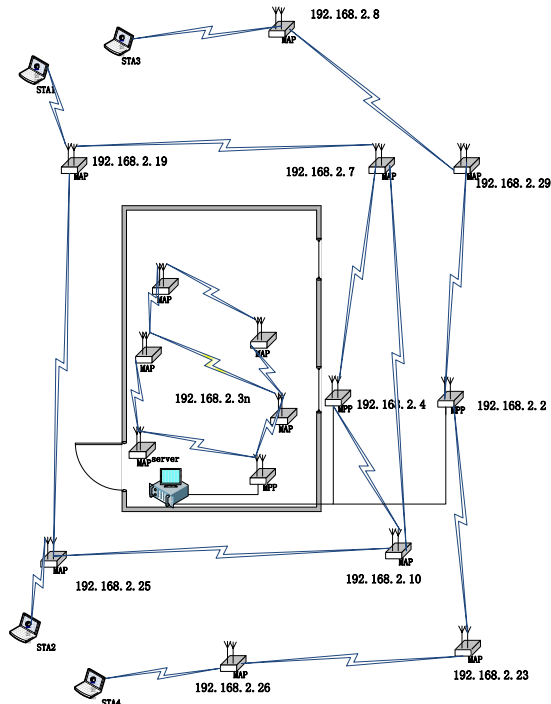


图 10 测试环境搭建

Fig. 10 The setting up of the testbed

3.2 测试方案与结果

为了保证软件的整体服务稳定，同时软件的各个功能模块能够正常的工作，并具备一定的容错性。测试中采用循环测试的方法，首先分别对每个子功能模块分别进行测试，若测试发现错误，则终止测试，对该模块代码进行修改，修改完之后，继续进行测试，直到该模块测试过程中没有错误。所有子模块测试成功后，会整合全部功能模块进行测试，若有错误，则定位到错误所在的子模块，继续修改，不断的测试，直至整个软件中的所有功能模块都通过测试。

分布式的无线 Mesh 管理平台包含整体管理部分与单节点管理部分，针对整体管理部分，按照前面提出的测试流程，我们分别对系统的节点发现，节点展示，拓扑发现等功能进行测试，测试结果符合预期要求，具体如图 11 所示。

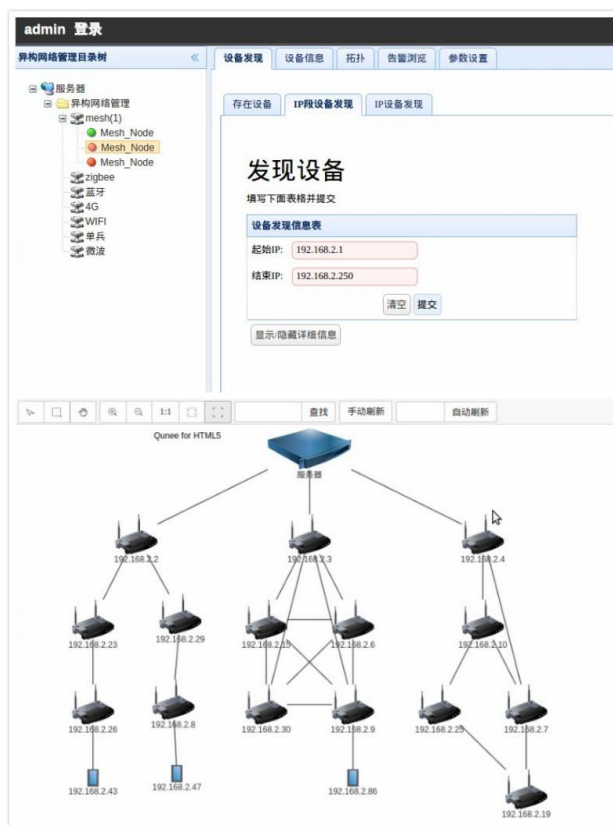


图 11 Mesh 网管整体功能测试结果

Fig. 11 The function tests of the whole mesh management system

每个 Mesh 节点还有一个自身基于 OpenWrt LuCI 的配置管理平台，我们依照同样的方法进行测试，结果符合预期，如下图 12 所示：



图 12 Mesh 网管单节点功能测试结果

Fig. 12 The function tests of single node management system

4 结论

240 本文依据 IEEE 802.11s 无线 Mesh 网络的特点, 结合开源的 OpenWrt 系统, 利用 SNMP
简单网络管理协议, 结合最新的 Web 开发技术, 设计了灵活, 方便的无线 Mesh 网络管理
245 方案。该方案创新的提出分布式管理的无线 Mesh 网管理体系, 将管理的任务不再集中于一个单一的服务器, 让节点自身也能提供管理功能, 让全局性的管理任务集中于统一的网管平台, 让涉及单个节点与细节部分的管理任务交给自身节点。实现管理功能的分层分离, 特别是解决了 Mesh 网管理中拓扑发现的难点, 提高了管理效率。同时, 本文根据该方案设计出实际的测试场景, 搭建出不少于 15 个节点的无线 Mesh 网络, 并通过了各个功能模块的测试。

[参考文献] (References)

- 250 [1] I. F. Akyildiz, Wang X. A survey on wireless mesh networks[J]. IEEE Communications Magazine, 2005, 43(9):S23-S30.
[2] Hiertz G R, Denteneer D, Max S, et al. IEEE 802.11 s: the WLAN mesh standard[J]. Wireless Communications, IEEE, 2010, 17(1): 104-111.
[3] 张牧, 严军荣. 802.11s 无线 mesh 网络研究进展与挑战[J]. 计算机工程与应用, 2010(22):75-79.
255 [4] Rethfeldt M, Danielis P, Moritz G, et al. Design and development of a management solution for wireless mesh networks based on IEEE 802.11s[C]// Ifip/ieee International Symposium on Integrated Network Management. IEEE, 2015.
[5] 许倩. 基于 OpenWrt 系统路由器的功能模块的开发[D]. 中国石油大学 (华东), 2013.
[6] 邵旭, 刘红, 丁炜. SNMP Agent 和嵌入式 Web 网管系统的研究与综合设计[J]. 北京邮电大学学报, 2001, 24(4):93-97.
260 [7] 代勇. 基于无线 Mesh 网络的网管软件的研究[D]. 北京交通大学, 2011.
[8] 曹为华, 凌强, 张雷, 等. 基于 OpenWrt 系统路由器的模式切换与网页设计[J]. 微型机与应用, 2015, 34(23): 91-94.
[9] Cheng J, Kang G, Gao Y, et al. From architecture to implementation for IEEE 802.11 s wireless MESH networks[C]//11th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2015). IET, 2015: 1-7.
265