

无线嵌入式Web网络管理系统的设计与实现

作者姓名 路 皓 宇

学校教师姓名、职称 王 献 青 副教授

企业导师姓名、职称 武 志 强 高工

申请学位类别 工程硕士

学校代码 10701
分 类 号 TP311.5

学 号 1310122658
密 级 公开

西安电子科技大学

硕士学位论文

无线嵌入式 Web 网络管理系统的设计与实现

作者姓名：路皓宇

领 域：软件工程

学位类别：工程硕士

学校导师姓名、职称：王献青 副教授

企业导师姓名、职称：武志强 高工

学 院：软件学院

提交日期：2015 年 11 月

Design and Implementation of Wireless Embedded Web Network Management System

A thesis submitted to
XIDIAN UNIVERSITY
in partial fulfillment of the requirements
for the degree of Master
in Software Engineering

By

Lu Haoyu

Supervisor: Wang Xianqing Associate Professor Wu Zhiqiang

Senior Engineer

西安电子科技大学 学位论文独创性（或创新性）声明

秉承学校严谨的学风和优良的科学道德，本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同事对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文若有不实之处，本人承担一切法律责任。

本人签名：_____ 日 期：_____

西安电子科技大学 关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权属于西安电子科技大学。学校有权保留送交论文的复印件，允许查阅、借阅论文；学校可以公布论文的全部或部分内容，允许采用影印、缩印或其它复制手段保存论文。同时本人保证，结合学位论文研究成果完成的论文、发明专利等成果，署名为西安电子科技大学。

保密的学位论文在____年解密后适用本授权书。

本人签名：_____ 导师签名：_____

日 期：_____ 日 期：_____

摘要

面对如今无线网络日益飞快的发展，传统的简单网络管理协议 SNMP 已经不能满足当今复杂的网络结构，为此 IETF 提出了基于 XML 的下一代网络配置协议 NETCONF。NETCONF 协议在网络配置方面改进了很多 SNMP 存在的不足，而且可扩展性高，基于 NETCONF 协议的网络管理目前已经成为解决复杂网络管理配置的主流手段。

本文对比有线网络的管理和配置，提出了无线网络管理的主要内容。并针对这些管理的主要内容，以 NETCONF 协议为理论依据，WBM 方案的嵌入式模型为主要框架模型，构建出了无线嵌入式 Web 网管系统的基本原型。为了更加简单完成实现无线网管系统的基本原型，选用了成熟的 Web 服务器方案，所以在这个方面就没做过多的研究，把本论文的研究重点就放在了管理端的分析与设计实现上来。管理端既要给网管提供无线网络管理的接口，也要使得后台的服务器端能够与前台通信，基于这两个方面的考虑，以 NETCONF 协议为根据，按照层次设计的原则，将管理端设计为四层：业务配置层、控制接口层、报文表示层和通信会话层。业务配置层提供给网管业务配置的接口，该层次的设计从总体上分为五个部分，对应无线网络管理的五个主要内容；控制接口层是向业务配置层提供开发业务配置的能力，这种能力主要体现在两个方面，其一是页面控制能力，使得业务配置层的开发模块化，而且这样设计的好处是模块之间的耦合性更低；其二是数据处理能力，在报文表示层之上做一些数据的预处理，例如临时存储等，降低嵌入式 Web 服务器的负载，提高配置数据的效率；报文表示层完成了 NETCONF 报文的封装和解析功能，屏蔽了无线网络管理的各种报文的差异性，是通信会话层和控制接口层的数据流传递的枢纽；最低层是通信会话层，实现 NETCONF 会话连接的控制和报文数据传输功能。

由于该系统在业务配置层开发的相似性，所以在实现和设计上只是简单介绍了其中的一个模块，其他模块的开发的过程方法基本一样。在论文的最后，完成了该无线网管系统的整个功能测试，测试结果全部通过，完成了整个论文的工作。

关键词：网络管理， NETCONF， WBM， 无线局域网， 嵌入式

论文类型：应用软件开发

ABSTRACT

With the rapid development of wireless network, the traditional simple network management protocol has been unable to meet today's complex network structure, so IETF proposed the next generation network configuration protocol based on NETCONF XML. In the network configuration, the NETCONF protocol has been improved a lot of problems, and it can be expanded. The network management based on NETCONF protocol has become the main way to solve the complex network management.

This paper compares the management and configuration of the wired network, and puts forward the main content of the wireless network management. And according to the main content of the management, the NETCONF protocol is the theoretical basis, the embedded model of the WBM scheme is the main frame model, and the basic prototype of the wireless embedded Web network management system is constructed. In order to realize the basic prototype of the wireless network management system, the mature Web server program is selected, so the research of this thesis is focused on the analysis and design of the management. The management should provide the interface to the network management, and also make the server side to communicate with the foreground. Based on the aspects, the NETCONF protocol is designed according to the principle of hierarchy design. The management is designed as four layers: the business configuration layer, the control interface layer, the message presentation layer and the communication session layer. The business configuration layer provides the interface to the network management business, the design of this level is divided into five parts, which are the main parts of the wireless network management. The control interface layer is the ability to provide the development of the business configuration layer and the ability is mainly reflected in two aspects, on the one hand is the page control ability, makes the business configuration layer of the development of the module, and the benefits of such design is lower coupling between modules; on the other hand, the data processing capability, which is based on the message presentation layer, such as temporary storage, reduce the load of the embedded Web server, improve the efficiency of the configuration data, and the NETCONF packet is encapsulated and resolved. The communication session layer is shielded. The lowest layer is the communication session layer, which realizes the control of the NETCONF session and packet data transmission.

Because of the similarity of the development of the system in the business configuration layer, the implementation and design of the system is a simple introduction of the module, the other modules of the process method is basically the same. At the end of the paper, the whole function test of the wireless network management system is completed, and the whole thesis is accomplished through the whole test results.

Keywords: Network Management, NETCONF, WBM, WLAN, Embedd

Type of dissertation : Applied software development

插图索引

图 2.1	NETCONF 协议.....	5
图 2.2	现代网络管理模型	8
图 2.3	嵌入式 WBM 模型	9
图 3.1	无线网络管理功能分解图	12
图 3.2	无线网管系统的数据类型	15
图 3.3	无线网络连接拓扑图	16
图 3.4	无线网络管理系统结构图	16
图 4.1	嵌入式 Web 网管系统框架	19
图 4.2	管理端结构设计图	20
图 4.3	控制接口层数据流图	22
图 4.4	控制接口层流程图	23
图 4.5	NETCONF 报文封装模块流程图.....	25
图 4.6	NETCONF 报文解析模块流程图.....	26
图 4.7	通信会话层流程图	27
图 4.8	通信会话层类图	28
图 4.9	报文表示层类图	29
图 4.10	页面控制类图	31
图 4.11	数据操作管理类图	31
图 4.12	localStorage 对象和 Cookie 对象.....	32
图 4.13	本地存储数据库	33
图 4.14	工具类接口示例	34
图 4.15	业务配置层业务开发流程图	37
图 4.16	业务配置层业务开发类图	37
图 4.17	使能/去使能事件处理函数流程图	39
图 4.18	WM2U 全局参数配置事件处理函数.....	40
图 5.1	系统测试组网图	43
图 5.2	无线嵌入式 Web 网管系统.....	44
图 5.3	性能管理	48
图 5.4	设备接口信息	49
图 5.5	端口统计信息	49
图 5.6	漫游客户端	51

表格索引

表 2.1	NETCONF 操作层基本操作	6
表 5.1	无线配置测试用例	44
表 5.2	设备管理测试用例	47
表 5.3	告警和安全管理测试用例	49
表 5.4	漫游移动管理测试用例	51

略语对照表

缩略语	英文全称	中文对照
NETCONF	Network Configuration Protocol	网络配置协议
WBM	Web-Based Management	基于 Web 管理模式
SNMP	Simple Network Management Protocol	简单网络管理协议
AC	Wireless Access Point Controller	无线访问控制器
AP	Wireless Access Point	无线接入点
XML	Extensible Markup Language	可扩展标记语言
WM2U	Wireless Multicast to Unicast	无线组播转单播
CGI	Common Gateway Interface	通用网关接口
STA	Station	工作站

目录

摘要	I
ABSTRACT	III
插图索引	V
表格索引	VII
略语对照表	IX
第一章 绪论	1
1.1 选题背景及意义	1
1.2 国内外发展现状	2
1.3 本文研究的内容	2
1.4 论文章节安排	3
1.5 本章小结	4
第二章 相关技术概述	5
2.1 NETCONF 网络管理协议	5
2.1.1 NETCONF 协议概述	5
2.1.2 XML 技术的简单介绍	7
2.1.3 NETCONF XML 数据模型	7
2.2 嵌入式 Web 网络管理系统	8
2.2.1 网络管理模型	8
2.2.2 WBM 模型	9
2.3 本章小结	10
第三章 无线嵌入式网管系统的需求分析	11
3.1 可行性分析	11
3.1.1 投资必要性	11
3.1.2 技术可行性	11
3.2 无线网管管理的主要内容	12
3.2.1 无线网络配置管理	12
3.2.2 设备管理	13
3.2.3 性能管理	13
3.2.4 告警和安全管理	13
3.2.5 漫游移动管理	14
3.3 无线网管系统的功能模型	15

3.4	无线网管系统现有框架基础	16
3.5	本章小结	17
第四章	无线嵌入式网管系统的设计与实现	19
4.1	管理端整体功能分析	19
4.2	管理端结构设计	21
4.2.1	总体结构概述	21
4.2.2	控制接口层设计	22
4.2.3	报文表示层设计	24
4.2.4	通信会话层设计	26
4.3	管理端的具体实现	27
4.3.1	通信会话层	28
4.3.2	报文表示层	29
4.3.3	控制接口层	30
4.3.4	业务配置层	34
4.4	本章小结	42
第五章	无线嵌入式网管系统的功能测试	43
5.1	系统测试环境	43
5.1.1	软硬件测试环境	43
5.2	系统功能测试	44
5.2.1	无线网络配置管理	44
5.2.2	设备管理	47
5.2.3	性能管理	48
5.2.4	告警和安全管理	49
5.2.5	漫游移动管理	50
5.3	本章小结	51
第六章	总结与展望	53
6.1	总结	53
6.2	展望	53
参考文献	55
致谢	57
作者简介	59

第一章 绪论

1.1 选题背景及意义

伴随着互联网的飞速发展,网络规模的井喷式增长,网络结构的复杂性、异构性也愈来愈加明显。特别是,近年来,无线网络的发展尤为突出,3G 和 4G 网络的快速建设,如何对这些新型的复杂的网络进行配置和管理,已经成为当今互联网管理中的关键问题。

传统的网络管理协议简单网络管理协议(Simple Network Management Protocol, SNMP),设计简单,加入了符合 Internet 定义的管理信息结构(SMI, Struct of Management Imformation)和管理信息库(MIB, Management Imformation Base),采用 UDP 发送管理请求,提供了大型网络设备的管理和数据的获取等功能^[1],各个大型网络通讯设备公司在生产设备的时候都兼容了 SNMP 协议的内容。但是 SNMP 在面对当前这些新型的复杂网络时,就出现了传输数据效率低,难以进行网络配置,安全性低等缺点。

IETF(工程任务组)为了解决这种现状,提出来的基于 XML 的下一代网络配置协议(NETCONF),NETCONF 规定的空间和数据操作类型都用 XML 表示,XML 的优点是能够表达本身存在逻辑关系的、易于进行数据处理的、能够进行抽象化的管理模型,便于数据在系统的各个不同模块之间的交互,而且 NETCONF 是面向连接的,它规定通信端口之间必须保持持续性的连接,在安全性方面友好的解决了 SNMP 遗留的问题。

自从 2006 年正式通过 NETCONF 基本标准 RFC4741-4744 以来, IETF 又在接下来的几年的时间里接连提出了 RFC5277、RFC5539 等标准,很好的完善了 NETCONF 协议在故障处理和安全传输等方面的机制^[2]。使得 NETCONF 协议很快在各大网络设备生产商如爱立信、Cisco、朗讯和华为的产品中得到了推广和应用。而由于 Web 技术对于 XML 的良好支持,使得基于 Web 的网络管理渐渐成为主流,但是因为 NETCONF 协议没有明确定义管理信息模型的结构,导致数据量巨大的 XML 由于 DOM 技术的限制的而会变得不可控,针对目前这种情况,各大组织纷纷提出了各种数据模型的草案,如 Vlan 数据模型、ACL 数据模型、Monitor 数据模型等,这些草案正在不断完善之中,也许不久的将来就会为 NETCONF 协议提供一种良好的解决方案。

对于目前这种情况,基于 NETCONF 协议构建无线网络管理系统可以有效配置无线网络基础服务,方便管理数量日益增多的无线设备,为广大用户提供一个更加稳定

地、安全地无线网络访问环境，使得无线网络在生活和工作上的应用领域越来越广，推动了无线网络技术向一个更好的方向发展，两者之间完成了一个友好发展，相互促进的积极过程。

1.2 国内外发展现状

近几年，伴随着无线网络技术的日渐成熟，已经被广大运营商和开发者应用到了很多的领域，可以说无线网络技术正在走进家家户户。那么由于无线网络技术的大范围覆盖，过去的无线网络管理技术已经不能满足今天对无线网络的需求了。在无线网络还没有当今这种发展局面的时候，人们只是简单的将它作为有线网络的补充，在小范围面积小规模的应用和当作有线技术的辅助手段，但是现在不一样了，不能用过去管理小范围无线网络的方式来管理现在日益复杂、多元化的网络结构。

针对目前这种情况，很多通信设备制造商和研究机构纷纷推出了自己的无线网络管理解决方案。这些解决方案的整体思路基本都是在继承有线网络管理的基础之上，加入了管理无线网络特有属性的部分^[2]。由于无线网络利用无线电波作为传输媒介，所以相较于有线网，无线网络的传输更容易受到干扰，更加需要强有力的检控和管理，使无线网络能够安全可靠的运行。

由于无线网络还在发展的过程中，关于无线网络管理的内容和复杂性也随着指数级增长，可以想象到不久的将来现在这种无线网络管理的内容已经远远就不够满足网络管理的需求了。正是因为这种渐变的增长性，要求如果要构建无线网络管理系统必须要求有很好的扩展性、兼容性。所幸的是基于 XML 的 NETCONF 协议具有很好的延展性，在其上搭建无线网络管理系统应该是一个可行的办法。但是就算是利用了 NETCONF 协议，无线网络管理内容的增长仍然是阻挡网络管理系统健壮性的一座大山。

1.3 本文研究的内容

本文通过参考有线网络管理系统的建立过程，分析当前无线网管系统的设计与实现的原理，提出来基于 WBM 的嵌入式建立无线网管的解决方案。而嵌入式无线网管系统的设计与实现主要是基于 Web 技术的发展，当前 Web 技术的发展使得网管系统更多的功能可以放到浏览器端解决，有效解决了嵌入式系统 Web 服务器的负载资源吞吐量低的问题。

论文主要研究的内容如下：

1. 仔细分析和研究了 NETCONF 协议在无线网络管理上的应用。首先在详细研究了 NETCONF 协议的整体框架的基础之上，将无线网络管理的基本内容概括为

三种 XML 数据模型，即运行数据、配置数据和操作数据。针对这三种 XML 数据规定了无线网管系统的数据的元操作模型，也就是基本定义了本文要用到的 NETCONF 报文的基本格式。其次，对于常用网络系统的设计框架进行了分析，参考有线网络管理的经验，选用 WBM 嵌入式网络管理框架作为本文无线网络管理系统的应用结构模型，并对该模型的具体应用进行了仔细的学习。

2. 对无线网络管理系统进行需求分析。对建立该无线网管系统的经济可行性和技术可用性进行调研，得出该无线网管系统从经济和技术上是可行的结论。既然建立无线网络管理系统是可行的，那么必须深入研究无线网络到底管理和配置哪些主要的内容。只有知道了无线网络管理配置的内容以后，才能在构建无线网络管理系统的时候有的放矢。研究了无线网络管理系统的主要内容以后，就需要构建基于 NETCONF 协议的 WBM 的嵌入式解决方案去管理这些内容，完成无线网络管理的主要功能。那么接下来就调研了构建该 NETCONF 协议的 WBM 的嵌入式解决方案现有的可以被有效利用的资源 and 框架，提出需要去突破和创新解决的问题，以便于去完成整个无线网络管理系统的设计与实现。

3. 设计与实现无线网管系统的管理端内容，完成系统的整体功能。对无线网管系统的管理端利用了分层设计的思想，由低到高细分为通信会话层、报文表示层、控制接口层和业务配置层四层。通信会话层完成 NETCONF 会话的控制，包括 NETCONF 会话的建立和释放等等内容；报文表示层主要负责 NETCONF 报文的封装和解析，控制接口层提供控制接口给业务配置层完成具体业务配置和管理。

4. 对当前实现的无线网管系统进行测试。主要还是对无线网络管理的具体内容进行功能性测试，从五个功能内容（无线网络配置管理、设备管理、性能管理、告警和安全管理、移动和位置管理）方面进行有效监控和配置下发的功能测试。

1.4 论文章节安排

论文全文共分为六章，具体章节安排如下：

第一章介绍基于 NETCONF 协议的无线网络管理系统的研究背景，介绍了关于无线网络管理系统在国内外的研究状况，并对论文整体需要研究的内容进行了简单的分析与调研。

第二章对构建无线网络管理系统所要用到的技术，例如 NETCONF 协议的详细内容、网络管理系统模型等进行了深入的学习和研究。

第三章对基于 WBM 嵌入式网络管理模型的系统进行了详细的需求分析。

第四章设计与实现了管理端的具体功能，构建完整的无线嵌入式 Web 网络管理系统。

第五章测试完成的无线嵌入式 Web 网络管理系统的功能是否满足需求分析所提出来的要求。

第六章总结了本文的研究工作，介绍了未来需要努力的方向。

1.5 本章小结

本章首先对无线网络管理系统的选题背景和意义进行了初步的分析与调研，然后查阅了该选题在国内外的发展历史和现状，并针对问题提出了本文需要研究的内容，最后对整个论文的章节进行了详细的安排与设计。

第二章 相关技术概述

构建无线网管系统，必须依靠相关技术原理。本章是对管理无线网络设备的通讯协议 NETCONF 协议进行了研究和分析，在此基础上探讨了网络管理模型的搭建，主要是对 WBM 嵌入式 Web 网络管理模型进行调研和学习，为设计实现无线网管系统作技术储备。

2.1 NETCONF 网络管理协议

2.1.1 NETCONF 协议概述

NETCONF 协议是一种基于 XML 的网络管理协议，它提供了一种对网络设备进行配置和管理的方法。该协议允许设备提供一系列完整、规范化的编程接口。网络管理系统可以方便的基于该接口设置、获取配置参数等。因其完整、规范化的编程接口及主流网络设备厂商的实现支持^[3]，使之成为未来网络管理系统与网络设备进行信息交互的优秀方案。

NETCONF 协议为了更加方便的进行网络配置管理，使得网络配置管理的方法抽象为四层结构，由下往上依次为：通信协议层、RPC 层、操作层和内容层^[4]，如图 2.1。

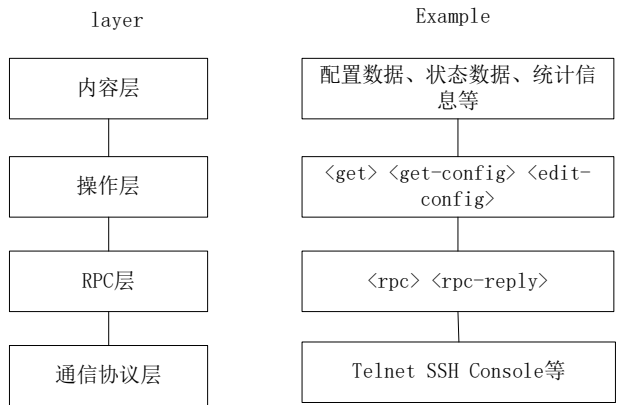


图2.1 NETCONF 协议

内容层(Content): 内容层抽象为被管理对象的最小集合，可以是统计数据、运行状态数据和配置数据等。各个生产厂商在内容层自定义产品的配置数据内容，并以 XSD(XML Schema Document, XML 架构文档)提供可具体读写的数据类型给用户，用户利用这些规定的数据类型进行设备系统的二次开发。

操作层(Operations): 操作层规定了在 RPC 中的原语操作集中的最小原语子集，

这个最小原语子集组成了 NETCONF 的操作中的基础能力部分。具体在操作层规范了十种基础操作原语，主要功能是完成配置操作、取值操作、会话操作和锁操作等，如表 2.1 所示。

表2.1 NETCONF 操作层基本操作

操作类型	作用描述
close-session	终止本次会话
commit	提交配置数据类型为运行数据类型
copy-config	复制配置数据
create-subscription	创建一个 NETCONF 通知订阅
delete-config	删除配置数据
edit-config	编辑修改配置数据
get	获取运行数据或者统计数据
get-config	获取配置数据
lock/unlock	对配置数据加锁/解锁控制写权限
kill-session	主动终止其他会话

RPC 层：RPC 层在 NETCONF 协议中利用简洁的编码机制屏蔽了众多传输协议的不同。通过使用<rpc>和<rpc-reply>标签元素分别来完成 NETCONF 请求数据功能和响应数据功能，使用<ok>和<rpc-error>元素来提供成功或失败的反馈信息。

通信协议层：和 NETCONF 协议本身无关，由网络设备和管理节点之间提供的会话载体。最新的 NETCONF 协议推荐标准 RFC5539 提供了对安全外壳协议(SSH, Secure Shell)、简单对象访问协议(SOAP, Simple Object Access Protocol)、块可扩展交换协议(BEEP, Blocks Extensible Exchange Protocol)、安全传输协议(TLS, Transport Layer Security Protocol)等通信交互协议的支持。

NETCONF 协议规定确立会话之前必须初始化连接，建立会话的过程中，服务器端和客户端相互发送 hello 报文给对方。服务器端等待客户端发送<rpc>请求，然后对每一个 rpc 请求反馈对应的包含<rpc-reply>的回复消息^[4]，客户端可以添加任意的自己所必需的 XML 属性在<rpc>请求中，服务器端将在回复消息中包含所有这些 XML 属性。

当 NETCONF 会话建立以后，客户端从 hello 报文中明确了服务器支持的操作类型。客户端发送<rpc>请求给服务器并且从服务器端接受请求的应答消息。客户端发送给服务器端的请求是必须按照队列进行排序处理，以便于保证 NETCONF 连接请求的可靠性。

2.1.2 XML 技术的简单介绍

XML (Extensible Markup Language), 意为可扩展标记语言, 是 W3C (万维网联盟) 提出的为了解决传输和保存数据的类似于 HTML 的可自行定义标签的推荐标准。XML 可以对数据和文档进行结构化处理, 有利于信息在 Internet 上的传输, 而且是跨平台的, 对于具体的操作系统环境没有依赖性要求。随着 Web 技术的发展, XML 已经逐渐成为各种信息系统之间传输和保存数据的最常用工具, 并且基于 XML, 发展创建了很多新的 Internet 语言, 其中 RDF 和 OWL(用于描述资源和本体)奠定了下一代 Web (语义 Web) 的基础。

早期对 XML 进行规范和定义的是 DTD (Document Type Definition, 文档类型定义), 因为 DTD 由于种种缺陷, 例如 DTD 提供的数据类型不够, 没有提供自定义的接口等, 后来被 XML Schema 取代。XML Schema 的出现, 可以让用户自己定制属于自己的解决方案的 XML 文档, 而且它最大的优势是不用学习新的语法规则, 本身就是用 XML 编写, 还提供了定制数据类型的方法。

XML 也有 DOM 方法, 可以通过接口来操作文档。如果在项目中有将 XML 的一种格式转换为另外一种格式的文档的需求, 可以通过使用 XSL (Extensible Stylesheet Language, 扩展样式表语言) 解决这个问题。XSL 有 3 个组件构成, XSLT (转换 XML 标准)、XPath (检索信息)、XSL-FO(格式化)。

2.1.3 NETCONF XML 数据模型

在 NETCONF 中, NETCONF XML 数据请求遵循 RF4741 规定, 数据请求被划分为三个基本类型: 运行数据、配置数据和操作数据^{[5][6]}。运行数据分为两个部分, 分为系统状态数据和配置数据, 这里的配置数据特指那些已经使用<commit>操作激活的配置数据。状态数据是指协议和用户配置在网管系统运行过程中状态变化在参数上的表现。配置数据表达的就是系统运行时候的配置参数, 比如系统运行时需要的接口速率、双工等参数的配置^[7]。不同于运行数据, 对配置数据所做的任何操作更改不会在网络设备中立即生效的。一旦配置数据配置好以后, 可以使用<commit>操作来激活配置数据的变化, 使得它们成为运行数据的一部分。操作数据通常描述一个时间点的动作节点, 例如清除统计接口信息。由于这部分的数据只是描述一个时间点的动作, 所以对该类数据不能做获取操作。

NETCONF 协议规定不同的数据类型支持操作的类型是不同的, 不同的数据类型在 XML 中体现为不同的命名空间, 即不同的命名空间需要做不同的操作类型。

1. 运行数据支持的操作类型

对系统的运行数据只能够做获取操作, 具体的体现在 NETCONF 协议使用"get"操作标签获取系统的运行数据。

2. 配置数据支持的操作类型

配置数据支持的操作从整体上分为两个部分，第一部分是配置数据的获取操作，第二部分是配置数据的设置操作。即系统配置数据，具体为"get"操作和"edit"操作。

NETCONF 协议使用"get-config"操作标签来下发系统配置数据获取操作^[8]。

配置数据设置操作在 NETCONF 协议中规定使用"edit-config"操作标签来下发。NETCONF 协议提供了四种配置数据的设置办法："merge"、"create"、"delete"和"replace"。

3. 操作类数据支持的操作

除了运行类数据和配置类数据，还存在一类特殊的操作类数据。这类数据更多倾向于描述的是一个动作上的节点，例如接口统计信息的清除动作。如果需要，这类型的支持返回的"action"结果，比如一个"ping"的响应数据。

NETCONF 协议扩展出了一个特殊的操作标签"action"，来下发操作数据。

2.2 嵌入式 Web 网络管理系统

2.2.1 网络管理模型

由于现代网络开放性、互联性的要求，国际标准化组织制定了开发式系统互连的网络模型，虽然开放式系统互连的网络模型没有直接对网络管理系统产生影响，但是在此模型的基础上，产生的基于远程监控的系统管理模型却是现代网络管理系统的标准。

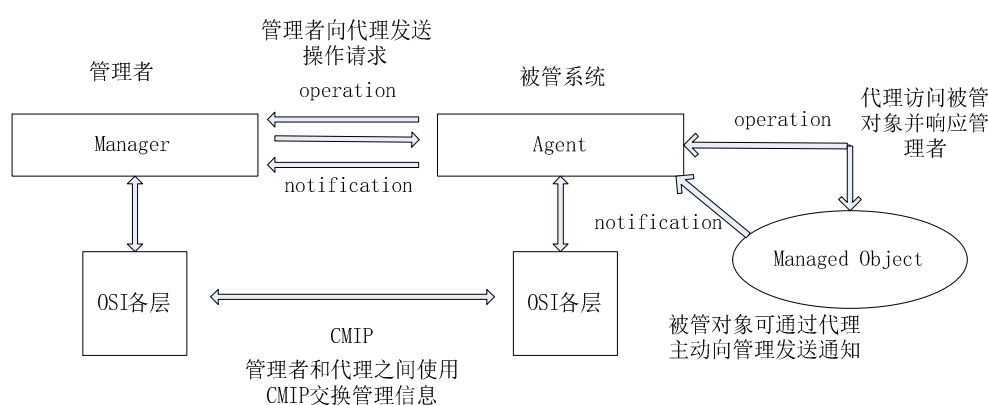


图2.2 现代网络管理模型

如图 2.2，系统管理模型建立在两个互联的能够通信的两个开放系统上，一个开放系统上的应用层配置一个 Manager 实体，而相应的在另一个开放系统的应用层之上需要配置一个 Agent 实体。将配置 Manager 实体的系统称之为管理系统，对应的配置 Agent 实体的系统称为被管系统。管理系统通过 Manager 实体发送访问被管系统中管

称之为管理系统,对应的配置 Agent 实体的系统称为被管系统。管理系统通过 Manager 实体发送访问被管系统中管理信息的操作请求,被管系统通过 Agent 实体接受操作请求,向被管对象(Managed Object)下发该操作,接受被管对象反馈,然后反馈给上一级的管理系统。当然被管系统中发生突发事件,也可以主动发送通知给 Agent 实体,然后 Agent 实体主动发送通知给上一级的管理系统,管理系统通过 Manager 实体接受此通知。值得注意的是管理系统和被管系统所扮演的角色不是固定的,它可以以 Manager 实体的作用作为管理系统,当然也可以以 Agent 实体的作用作为被管理系统。

其中配置 Agent 实体的被管理系统提供的基本服务有:事件报告功能、访问控制功能、日志登陆功能和时间安排功能。支持的被管对象有鉴别器、访问控制、日志控制和时间表等。

Manager 实体和 Agent 实体之间的通信协议,由位于应用层的 CMIP(公共管理信息协议)规定。

2.2.2 WBM 模型

WBM(Web-Based Management)是基于 Web 的网络管理模式,结合了现代网络管理技术,为网络管理员提供了一种更加方便友好的图像操作界面,降低了网络管理和维护对管理员的要求成本。WBM 允许网络管理员使用任何一种浏览器,在任何地点、任何时间配置和管理无线网络。WBM 具有平台独立性,支持任何操作系统和各种环境。当更换一种操作平台的时候,没有必要进行系统移植。

WBM 模型提供了两种不同的解决方案,可供选择:其中一种是基于代理的解决方案^[9],通常用于在大型网管系统的设计中,另外一种嵌入式方案。

基于代理的解决方案实际上是将 Web 服务器设计在网络管理平台的上层,使其成为网络管理员进行网络监控和配置的代理者,网络管理平台与被管理的各种不同的设备通常是通过 SNMP 或者通用管理信息协议(Common Management Information Protocol, CMIP)协议进行通信,管理和配置网络设备所需要的多种参数。

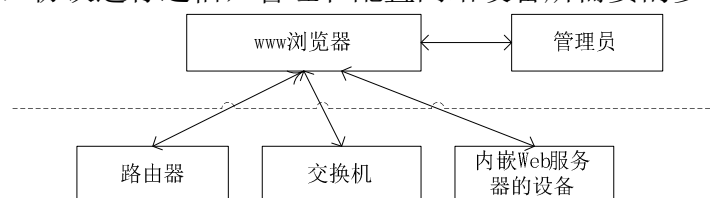


图2.3 嵌入式 WBM 模型

如图 2.3, 嵌入式方案是在被管理的设备中嵌入具有 HTTP 协议功能的 Web 服务器^[10]。每个设备需要为网管系统申请对应的 Web 地址,网络管理员只有通过浏览器和 HTTP 协议能够直接进行访问,才能完成后续的管理和配置操作。嵌入式方案的实

现的基本组成一般是一组文本格式的 Web 网页，用于提供操作接口，以便于网络管理员操作；一个在嵌入设备母体中运行的 Web 服务器，用于联系设备端信息；一个通信协议，用于规定数据交互的标准。网络管理员通过操作 Web 页面，将指令通过通信协议传递给 Web 服务器，Web 服务器调用相关设备服务程序处理，然后将处理结果又通过通信协议传递回 Web 浏览器，这时候网络管理员就可以在 Web 浏览器上看到反馈结果。

最初的嵌入式方案中的 Web 服务器直接由若干的公共网管接口（CGI, Common Gateway Interface）组成，利用标准 HTTP 协议与前台的 HTML 页面和脚本交互。这种方案要求开发人员直接在设备上进行调试开发，网管程序和设备的耦合性太高，时间成本比较大，而且一旦网管程序出现异常，有可能影响到别的功能模块的正常运行。

后来，逐渐出现了 Web 服务器只负责提供基础的 Web 服务和传递设备 Shell 命令和执行结果的 CGI。浏览器将网络管理员的操作下发给 Web 服务器，Web 服务器调用相关的 CGI，通过 Shell 接口将对应的 Shell 命令下发给对应的 Shell 进程，该进程执行完后将结果传递给对应 CGI，然后由 Web 服务器返回给浏览器进行解析，显示。而且，这种方案最大的好处是设备端的开发变得统一而复用性强。开发人员只需要开发对应的 Web 网页和脚本，大部分工作都可以在个人 PC 上调试完成，开发周期大大缩短。Web 网页和脚本都属于文本文件，往往不需要占用设备太多的存储空间，方便于设备版本升级，Web 网页本身也可以根据需要缩减。

2.3 本章小结

本章概括了 NETCONF 协议的基本知识，分析了基于 NETCONF 的基本操作的 XML 报文，提出了 Web 无线网络管理系统的整体框架模型。通过本章的描述，可以明确无线网络管理系统的基本架构组成，为系统的设计提供理论依据。

第三章 无线嵌入式网管系统的需求分析

仔细分析研究了网络管理系统常用的框架原型，受限于 AC 设备硬件资源的限制，决定选用 WBM 模型的嵌入式方案作为建立本网管系统的基本结构原型。有了以上这些技术上的支持，下一步的工作就是分析本网管系统的需求。要完成这个任务，需要从两个方向去开展工作，调研无线网络管理的主要内容和现有网管系统可供利用的软硬件基础，本章就是基于这两点开展论文研究工作。

3.1 可行性分析

随着 NETCONF 协议的逐步完善、嵌入式软件的发展，开发一个无线嵌入式 Web 的网络管理系统的成本相对来说不会太大，但是能够极大的提高无线网络的可靠性和安全性，对于一个主要是通信业务的公司来说，是可以看得见商业回报并且能够提高公司的品牌知名度的。着眼于这一点考虑，无线嵌入式 Web 网络管理系统在投资上是非常有必要的，经济上是可行的。

3.1.1 投资必要性

当前建立无线网管系统是一个面向商业的应用，必须计算它的成本然后去分析它的商业价值，是否值得花费该成本去建立此系统。如果没有具体的商业价值，对于一个企业来说就是没有必要的，毕竟构建如此规模的网管系统是需要不小的人力和技术成本的。

随着 NETCONF 协议的逐步完善、嵌入式软件的发展，开发一个无线嵌入式 Web 的网络管理系统的成本相对来说不会太大，但是能够极大的提高无线网络的可靠性和安全性，对于一个主要是通信业务的公司来说，是可以看得见商业回报并且能够提高公司的品牌知名度的。着眼于这一点考虑，无线嵌入式 Web 网络管理系统在投资上是非常有必要的，经济上是可行的。

3.1.2 技术可行性

根据第二章的技术学习和分析，要建立一个基于 WBM 方案的嵌入式 Web 网管系统，首先需要有一个运行软件系统的设备硬件平台，由于是 H3C 公司的商业项目，本系统需要搭建在型号为 WX5004 的无线访问控制器。该型号的无线访问控制器本身的软件平台上是 Comware v7，该软件平台是基于 Unix 的，所以在该操作系统上应用 JavaScript、HTML5、Ajax 和 JQuery 技术和构建 Web 服务器从技术上都是可行的。

3.2 无线网管管理的主要内容

本论文是以无线网络连接设备的 AC(Wireless Access Point Controller, 无线访问控制器)为基础, 结合具体无线网络的情况进行研究。Web 无线网络管理主要内容有: 无线网络配置管理、设备管理、性能管理、告警和安全管理、漫游移动管理等^[11]。系统功能模型图如 3.1 所示。

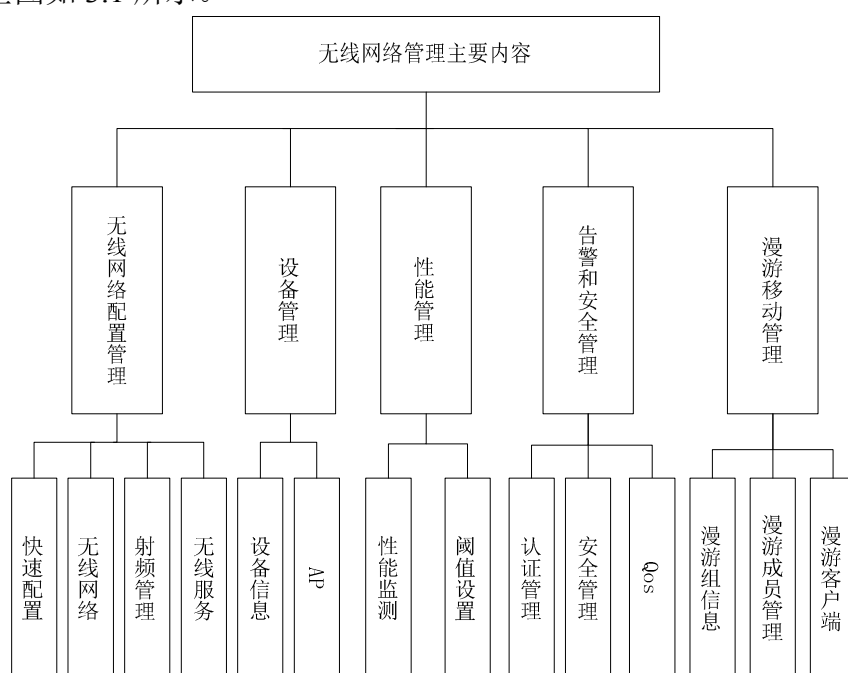


图3.1 无线网络管理功能分解图

3.2.1 无线网络配置管理

无线网络配置主要管理快速配置、无线网络基本参数、无线服务、射频等。快速配置的目的是为了给用户提供一个快速配置的向导, 可以让网络管理员迅速将网络设备配置到基本网络运行的要求。在快速配置过程中, 基本包括 IP 设置、无线网络和服务的配置、服务器的快速配置、加密和 AP 的快速设置等等内容。

无线网络的基本参数包括配置设备的 MAC 地址、虚拟局域网 (Virtual Local Area Network, VLAN) 相关参数、地址解析协议 (Address Resolution Protocol, ARP) 相关参数、无线组播转单播 (Wireless Multicast to Unicast, WM2U) 相关参数等。

无线服务是指无线提供的服务, 所有的基础参数的配置都是为了提供更加稳定、顺畅的无线服务。无线服务, 又称为无线接入服务, 通过配置无线服务所需要的参数来提供无线服务。无线接入服务, 是通过无线用户监测无线信号接入无线设备提供的无线服务, 大致上分为两个阶段, 第一个阶段是认证, 另一个阶段是关联阶段, 无线用户完成这个阶段就完成了无线用户接入。无线用户接入需要配置选无线服务名,

VLAN, 加密方式、绑定 radio 等参数。

射频参数配置顾名思义就是射频信号的具体配置,它不是为了让用户配置射频类型,而是让网络管理员更进一步的对射频速率的配置,管理员可以通过配置相关参数来实现功率信道优化的目的,甚至能够进行信道扫描、环境监测等等更加高级的功能。

3.2.2 设备管理

设备管理主要指管理无线网络的设备运行情况,实现上具体分为两个功能模块,第一个是当前设备的信息,第二个 AP 设备的管理。当前设备的信息分为基本信息、日期和时间、日志管理和配置管理。基本信息包括当前设备的名称设置、当前设备的闲置时间等。当前设备的闲置时间如果超过了阈值,就会被强制退出 NETCONF 会话,即强制退出 Web 网管配置页面。日期和时间是指当前设备的日期和时间的设置,以便于用户日志的记录等需要时间的地方,日志管理记录设备运行的时候的状态信息,以便于以后定位网络问题或者服务问题。这里的配置管理主要指设备的状态恢复和配置还原、备份等,方便设备的错误恢复,及时解决无线网络问题等。

AP 设备管理解决三个配置问题,解决的第一个问题是使 AC 和 AP 建立连接,AC 和 AP 之间的连接是通过 H3C 自己定义在应用层之上的协议完成的。第二个问题是自动 AP 配置,自动 AP 功能完成了 AC 上不需要配置 AP 的具体参数,就可以使得 AC 和 AP 连接在一起,这种连接方式不同于第一种正常配置的连接方式。第三个解决的问题是 AP 组管理,AP 组是为限制客户端只能在指定的 AP 组完成接入工作,不能接入不授权的 AP 组,为了保持无线服务个特定目的。

3.2.3 性能管理

性能管理主要目的是监控无线设备的各种功能,通过监控性能参数的变化,完成性能管理的任务,以便于网管优化网络的时候提供设备功能参数的依据。该功能分为两个主要的部分,其中一部分负责监控网络性能参数,另外一部分阈值设置,使得网络性能一直保持在阈值范围内。该部分的性能参数主要包括设备性能参数、端口流量统计等。

3.2.4 告警和安全管理

告警和安全管理主要分为三个小的子模块:认证管理、安全管理和 Qos。认证管理又区分为不同的认证方式,即 802.1x 认证、Radius 认证和 Portal 认证。三种不同的认证方式保证无线服务的接入的安全性,从不同的认证方式上防止非法用户接入无线网络服务,该部分的主要目的是从服务的接入时控制提高无线服务的安全性。

安全管理采取的手段主要两种,一种是安全配置,防止安全事故的发生,这主要

的就是通过制定各种安全规则：非法设备的监测、入侵设备的监测和黑白名单的配置。另外一种是用户隔离，如果没有用户隔离，处于同一个 VLAN 之间的接入客户端是可以互相访问的，当一个客户端接受到无线服务报文有可能被同一个 VLAN 内的其他客户端接受到，这将极大的增大系统的安全风险，所以系统提供用户隔离来预防此类风险发生。

Qos (Quality of Service, 服务质量)，该功能就是为了给网络要求质量的用户提供不同的网络质量的服务^[11]。分为了 ACL (Access Control List, 访问控制列表) 配置和 Qos 策略配置两个步骤。ACL 主要的目的就是实现过滤数据包的目的，通过配置各种访问控制规则和处理操作来完成该目的。具体的规则的配置根据 IP 地址不同的类型是具有不同的规则配置，但是大致上分为三种类型的，分别是基本规则、高级规则和链路层规则。三种规则的划分标准是通过配置报文的匹配规则区分的，在表现形式上通过 ACL 的标号范围来分别，具体是 2000 到 2999 为基本规则的标号范围，它通过限制报文的源地址来实现控制功能；3000 到 3999 为高级规则的标号范围，它的限制条件更加具体，不仅仅限制报文的源地址而且会限制目的 IP 地址、报文的协议类型等等；4000 到 4999 为链路层规则，它和前两者之间有明显的不同，链路层规则限制的是 MAC 地址，从源 MAC 地址、目的 MAC 地址等等来限制。ACL 配置完成以后，需要配置 Qos 策略匹配定义好的 ACL 规则，然后只需要应用 Qos 策略就行。

3.2.5 漫游移动管理

漫游移动管理是无线局域网相比于有线网络特殊的部分，由于无线局域网支持漫游特性，所以它的管理内容在不仅仅会多增加漫游移动管理功能，而且会增加无线网络管理的复杂性和异变性。这就要求无线网络在管理上更具有优势性，降低无线网络由于漫游过程中造成的各种不稳定。

用户信息包含了由于移动造成的用户的位置信息的改变，该用户是否选择 AC 间漫游，还是只是在 AC 内漫游。用户选择不同漫游类型是由具体设备的漫游组决定的，不同的漫游类型所形成的漫游组的记录以及参数配置都是不同的。但是不论是 AC 间漫游还是 AC 内漫游，用户都是在不同的 AP 间接入，所以需要记录该用户接入的 AP 位置。当用户发现此时接入的 AP 信息变弱的时候，自动开始向同一个漫游组的 AP 间漫游，此时用户漫游就会造成信息的改变，此时网管系统需要记录用户信息的变化，即漫游轨迹的变化。

该部分的详细信息包括漫游组信息、漫游成员参数和客户端具体信息的记录。漫游组信息记录了该漫游组绑定的服务类型、地址类型、源地址和认证信息等。漫游成员记录了成员的 IP 地址、VLAN 信息等，客户端信息记录客户端的 MAC 地址、网络服务标识、VLAN 标识和漫游方向等信息。

3.3 无线网管系统的功能模型

面对无线网络管理的五个主要的内容，本网管系统并不关注其怎么实现它们的功能的，它的实现是由无线控制器的守护进程来完成的。本网管系统实际上提供 Web 浏览器的接口来对该无线控制器进行配置操作以达到对整个无线局域网进行管理的目的。

NETCONF 协议规定了网管系统和它们对应的守护进程之间进行通信的方式，即通过 XML 数据报文。NETCONF 协议不仅仅规定了网管系统和守护进程之间通信的方式，还将其数据报文的类型进行了分类，这个在第二章就讨论过。在本系统中，继续遵循 NETCONF 协议规定，将 XML 数据报文包含的数据划分三大类型：运行数据、配置数据和操作数据。

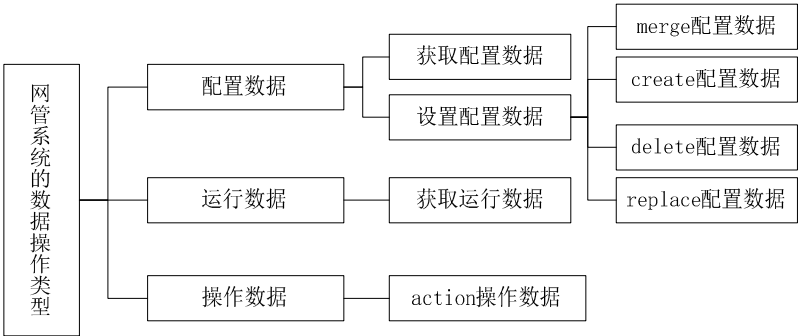


图3.2 无线网管系统的数据类型

如图 3.2 所示为将无线网络管理的主要内容抽象成三种数据类型，在本网管系统中将这三种数据类型定义为三种不同的 XML 数据命名空间，配置数据规定了两种操作类型：获取配置数据和设置配置数据，设置配置数据有包含了 merge、create、delete 和 replace 这四种操作。由于运行数据是指系统运行过程中的数据类型，它只能通过配置数据转化，而不能直接设置，所以只支持获取操作。操作数据比较特殊，不同与前两种数据，只表示一个动作，支持 action 操作。

无线网管系统所有设计与实现的内容，都是围绕该 XML 数据报文来展开的。既然这些数据报文是按照 NETCONF 协议的规定，所以它们有着明显的规律性，这种情况下，完全可以抛弃为每一条数据报文建立处理逻辑的这种效率低的想法，分离数据报文的共同点，为相同类型的数据报文建立共同的数据处理逻辑，提高了代码的可复用性，降低了硬件系统的负载，是比较符合嵌入式系统软件开发的原则的。

根据上述的分析，可以看出，建立无线网络管理系统实际上以无线网络管理的主要内容数据为依据，以 NETCONF 协议为处理规范，控制无线网络设备的守护进程之间交互的一种过程。

3.4 无线网管系统现有框架基础

如图 3.3 为无线网络连接拓扑图。嵌入式网管系统运行在无线 AC(Wireless Access Point Controller, 无线访问控制器)中, 网络管理员通过个人计算机访问无线访问控制

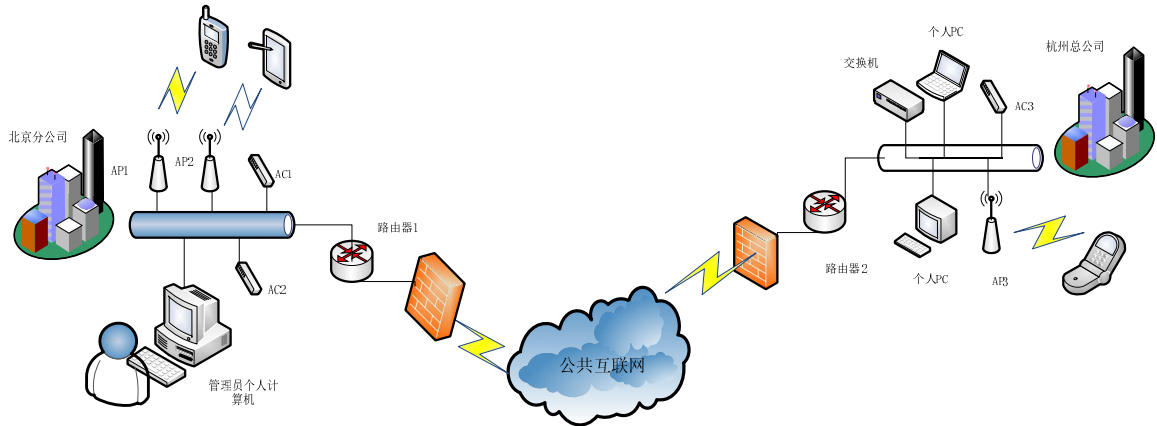


图3.3 无线网络连接拓扑图

器（即 AC 设备）去管理和配置网络设备和服务。例如在图 3.3 中可以通过访问无线访问控制器 AC1 去配置连接在同一个网段的 AP1 和 AP2(Wireless Access Point, 无线访问接入点), 通过在该无线访问控制器 AC1 上配置监控无线服务。

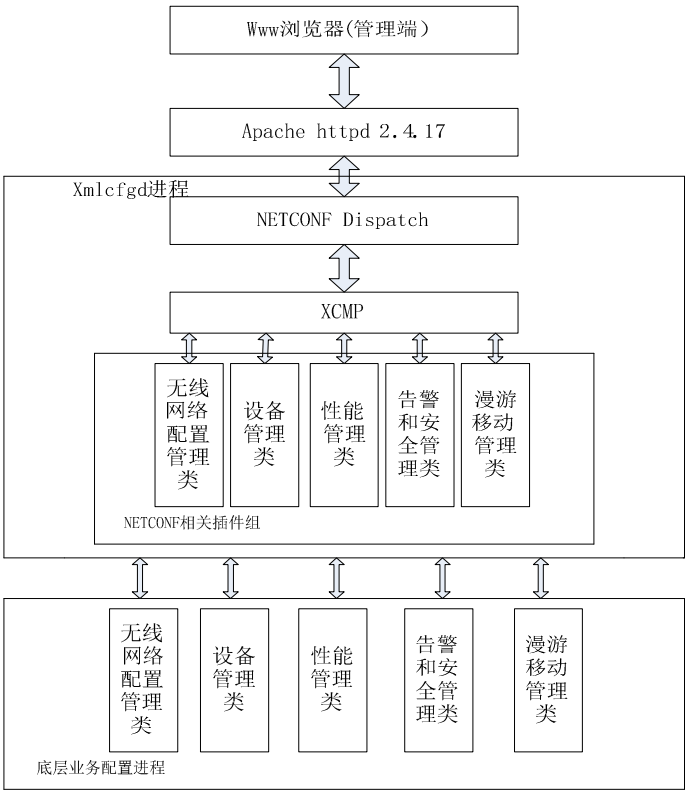


图3.4 无线网络管理系统结构图

图 3.4 所示为无线网管系统的总体部署图,网络管理员通过个人计算机浏览器访问无线访问控制器的地址,Web 服务器读取存储在文件系统中的网管文件,将该文件反馈给网络管理员的浏览器,浏览器解析文件并进行网络管理操作^[12]。该无线网管系统选用的 Web 服务器为 Apache 的 httpd 2.4.17 开源版本,浏览器下发操作的报文的内容为 NETCONF 请求操作,通过 Ajax 传递给 Web 服务器,服务器将该 NETCONF 请求转发给 Xmlcfdg 进程进行具体的配置操作。XMLcfdg 进程将配置请求的结果返回给 Web 服务器,Web 服务器通过 Ajax 传递给 Web 浏览器,浏览器解析结果在页面显示给用户,提供监控和后续配置操作。

Xmlcfdg 进程按照如下的过程处理 NETCONF 请求操作:首先由 NETCONF Dispatch 模块分发各种不同的 NETCONF 请求,业务请求进入 XCMP 模块(XML Configuration Management Process, XML 配置进程)。XCMP 将 NETCONF 请求转换成 XML 配置消息,下发到 XML 配置插件。各模块的 XML 配置插件通过 MOR(Managed Object Repository, 管理对象库)提供的 API 解析配置消息,然后和本模块的守护进程交互^[13],完成具体的配置功能,最后以配置消息的形式给 Web 服务器返回处理结果。

本网管系统的从大体上分为两个部分,管理端和后台服务器端。因为后台服务器端采用成熟的设计方案和开源的编程框架,所以在本论文的的研究过程中不做过多的分析和设计,将网管系统的重点放在了管理端的设计与实现上来,这项内容贯穿论文的整个设计与实现的部分。

3.5 本章小结

本章首先对无线网络管理系统进行了可行性的分析,证明了构建嵌入式无线网络管理系统的必要性。接着论述了无线网络配置管理的诸多内容,总结了该无线网络管理系统的功能需求为系统的设计与实现提出了明确的目标。

第四章 无线嵌入式网管系统的设计与实现

针对构建无线网络管理系统提出的各种需求,本章首先从管理端的结构上利用了分层设计的思想,进行了整体结构上的设计,然后将提出的整体结构层次上划分出多个功能模块,并对每一层每一个功能模块进行了详细的设计与实现。

4.1 管理端整体功能分析

面对无线局域网的这么多的管理内容,不同的管理参数的配置和检测,要建立一个完整的无线局域网管理系统是非常复杂的,而正是由于 NETCONF 协议的存在,让管理这么多的无线网络的内容变得可能。

无线网管系统从管理的内容上分为五个部分,所以从管理端的功能上来说,也要给网管即该无线网管系统的用户提供这五大类型的管理配置的内容,但是对于管理端来说,其输入输出分为两个方面,其一是向网管提供管理界面,其二是向后台服务器提供配置 XML 报文。

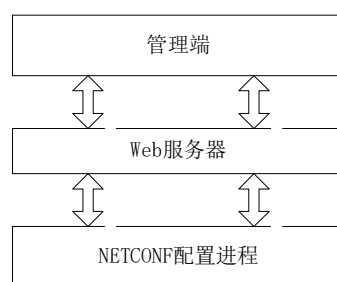


图4.1 嵌入式 Web 网管系统框架

由于管理端属于无线网管系统的一部分,符合 WBM 模型中的嵌入式 Web 网管设计。而该模型包括三个部分,如图 4.1: 管理端、Web 服务器以及 NETCONF 配置进程^[14]。管理端实际上就是一组交互文件和脚本,通过 Web 浏览器显示、下发网络管理信息和接受网络管理员操作。不仅用来提供管理员操作和监控的图形用户界面,而且负责将配置数据转化成 Web 服务器端能够下发给 NETCONF 配置进程的 XML 格式数据,当然也能够将 Web 服务器传过来的 XML 数据解析显示在用于监控的图形用户界面。实际上管理端在作用上等同于网络管理模型中的 Manager 端。

Web 服务器在设备中实现,提供标准的 HTTP 服务功能,且能够读取以压缩方式保存在文件系统中的管理端文件,并提供相关接口管理端文件进行单独升级。因为嵌入式系统的资源有限,Web 服务器不可能提供太过复杂的功能,它的架构必须精简,

资源使用率一定要低，不能影响整个嵌入式系统的运行，但是基本的 Web 服务功能一定要有，故而设计 Web 服务器必须要求可扩展性强、高精简等特点^[15]。Web 服务器和 Web 浏览器之间的通讯使用通过 Ajax 完成。这样有效的利用 Ajax 将前端和 Web 服务器分开，前端的开发就可以后端开发并发操作，而且前端只需要关注 Ajax 传递的数据进行操作，后台只需要将前端所需要的数据定义一致，这样前端和后台的开发耦合性更低，对开发程序员的要求就会降低，有效降低开发成本。

而嵌入式 Web 服务器必须能同时处理多个用户发送的请求，实际上该 Web 服务器可以实现同时响应 5 个 NETCONF 会话操作，所以要求该 Web 服务器的模型是并发服务器模型，采用多路复用 I/O 模型设计思想^[16]。不需要来一个请求就建立一个连接进程或线程，也不需必维护这个进程或线程，可以在单个进程中建立多个客户端的连接，实际上只有当 Socket 真正有 NETCONF 配置数据来的时候，才真正调用接收 NETCONF 配置数据的事件。

管理端文件管理模块主要是接受到请求以后，为系统查找提取管理端页面和脚本返回给浏览器，而管理端页面和脚本是压缩存储的，当需要的时候，会解压。在建立 NETCONF 连接以后，会自动进行解压缩，返回得到管理端地址。

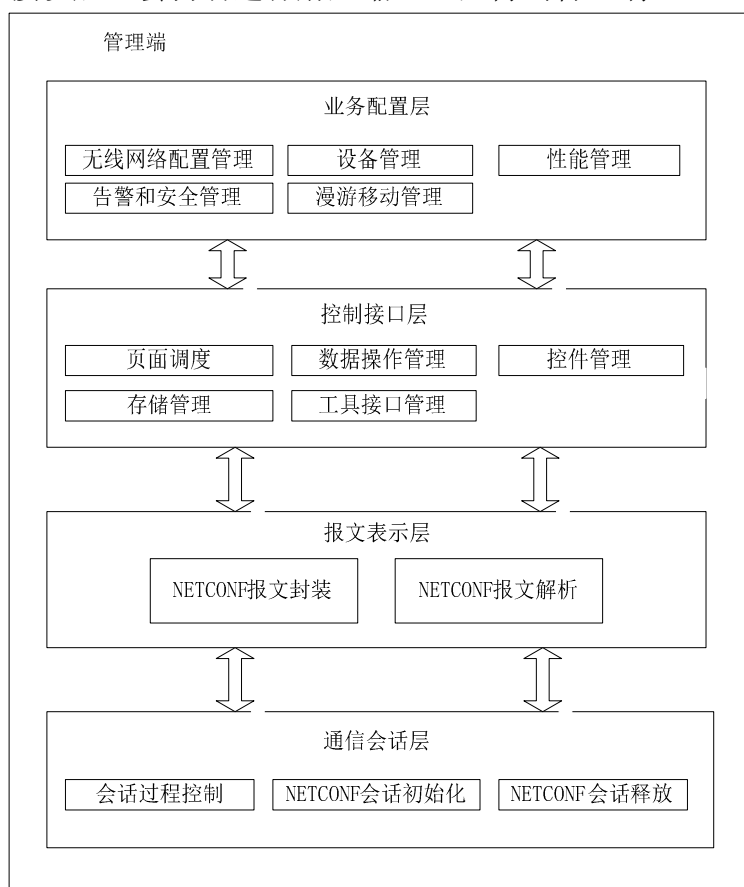


图4.2 管理端结构设计图

NETCONF 配置进程负责接收 Web 服务器发送来的 NETCONF 配置请求报文，然后将请求报文转化成配置消息，下发到 NETCONF 配置插件；各模块的 NETCONF 配置插件通过 MOR 库提供的 API 解析配置消息；然后和本模块的守护进程交互，完成具体的配置功能；最后以配置消息的形式返回处理结果^[17]。

4.2 管理端结构设计

4.2.1 总体结构概述

上面介绍了整个无线网络管理系统的整体设计流程，由于 Web 服务器端已经有成熟的设计方案，本论文重点研究管理端的结构设计，如图 4.2 所示。

管理端从总体结构设计上又划分为四层，分别为业务配置层、控制接口层、报文表示层和通信会话层。这样分层设计的好处是再降低各个模块耦合性的基础上，又提高了代码的复用性和整个管理端运行的效率，使得安全验证和异常处理在数据封装之前就可以进行完成，更加提高了系统的安全性和可靠性^[18]。

业务配置层对应于无线网络管理的具体业务内容，是整个管理端展现在网络管理员面前的最直观的一部分。网络管理员通过业务配置层的各种不同管理内容去管理无线网络，达到网管系统的目的。这部分是无线网管管理的业务内容，所示从功能模块上划分为五个方面，分别是无线网络配置管理、设备管理、性能管理、告警和安全管理以及漫游移动管理。业务配置层的开发从内容上说是分为五个方面，但是具体的模块却是跟后台的 NETCONF 配置插件模块一一对应的，这样做的好处是开发的时候可以进一步降低各个大的模块之间的耦合性，只需要定义它们之间的数据流向接口，而不用过多的关注各个小的模块之间的实现过程，有利于协同开发过程，便于定位开发过程中出现的错误与异常，如果是这样做的话，整个 Comware 版本升级的时候，网管系统只需要修改版本升级中某一个部分或者某几个模块就可以完成新的网管系统的移植，降低了工作量和开发成本。因为整个业务配置层的开发取决于后台 NETCONF 配置插件提供的 XML 报文参数，所以这部分的开发与后台的 NETCONF 配置插件模块是一一对应的。

控制接口层提供接口给业务配置层用于页面调度、数据操作管理、控件管理、存储管理以及一些工具接口类，而数据请求下发的时候，经过报文表示层，进行 NETCONF 报文封装，传递给通信会话层，会话层将消息发送给 Web 服务器，完成管理端操作的配置。Web 服务器反馈回的 NETCONF 报文，通信会话层接受到 NETCONF 报文传递给报文表示层，进行 NETCONF 报文解析，解析以后将参数传递给控制接口层，接口层提供接口给业务配置层，进行功能模块的数据处理。从这个数据流向可以看出，在业务配置层以下的控制接口层、报文表示层和通信会话层更加关

注的是 NETCONF 协议的实现,这部分的内容在 Web 技术没有今天如此进步的时候,都是由后台的服务器来完成的,但是由于本论文实现的嵌入式的网管系统的设备本身的硬件处理能力是十分有限的,当将 NETCONF 协议的报文的处理过程放在前台来处理以便于释放后台服务器的处理能力来做更多的其他业务,会大大提高系统的效率和吞吐量。

4.2.2 控制接口层设计

如图 4.2 所示,控制接口层完成的主要功能模块分为五个方面,页面调度负责动态页面的生成过程,由于嵌入式 Web 服务器本身的能力不足以在网管系统初始化的过程中,加载整个网管系统的所有页面,所以在这部分的内容是根据用户需求进行动态页面的生成,当你选择配置业务参数的时候,Web 服务器读取存储在文件系统中的文件进行加载,这样做更能进一步降低浏览器和 Web 服务器的负载^[20]。

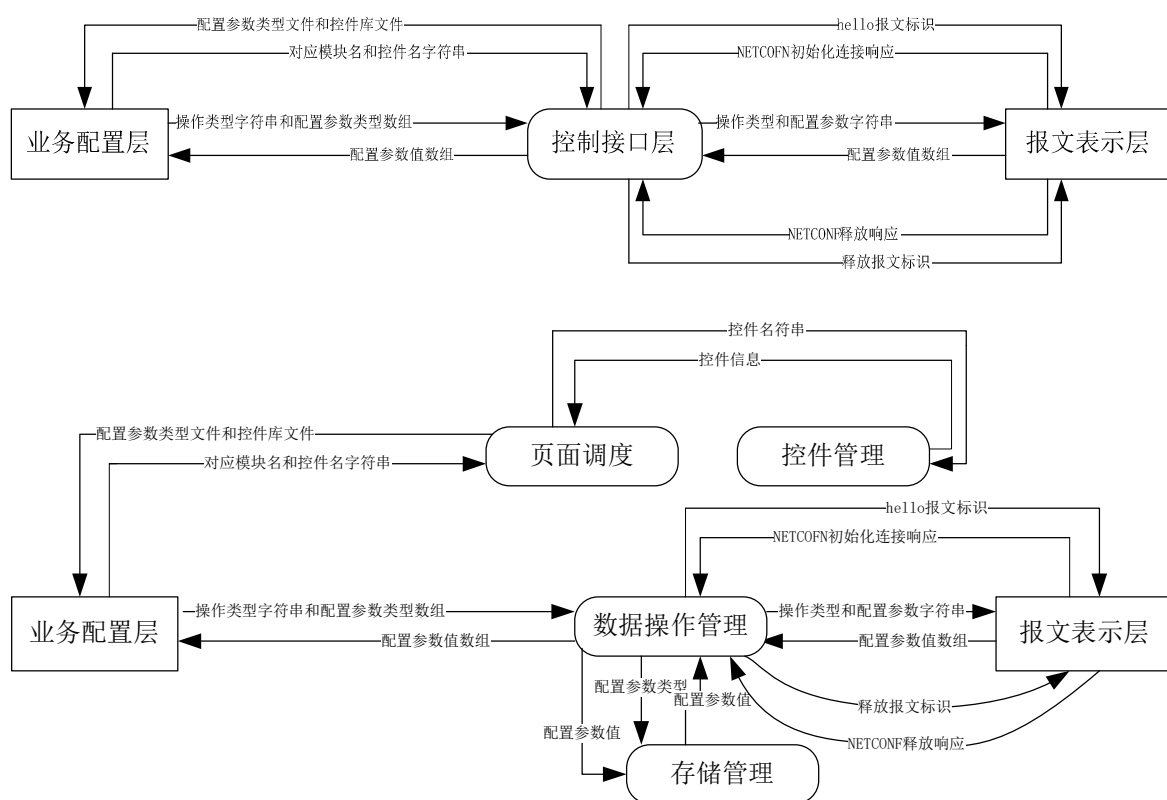


图4.3 控制接口层数据流图

如图 4.3 所示为控制接口层的数据流图,从总体的数据流向上看,控制接口层接口层从业务配置接收到的数据有两种:对应的模块名和控件名字符串、操作类型字符串和配置参数类型数组。这两类数据通过控制接口层的处理以后,变成三种类型的数据传递给报文表示层:NETCONF 连接初始化报文标识,即 hello 报文标识、NETCONF

连接释放报文标识、操作类型和配置参数字符串。

上述三种类型的数据经过报文表示层的下发处理以后，有三种响应报文反馈回来：hello 报文标识的响应报文、连接释放的请求的响应报文、配置参数值数组。控制接口层处理以后，反馈给业务配置层两种类型的响应：配置参数类型的声明文件和控件库文件、配置参数值数组。

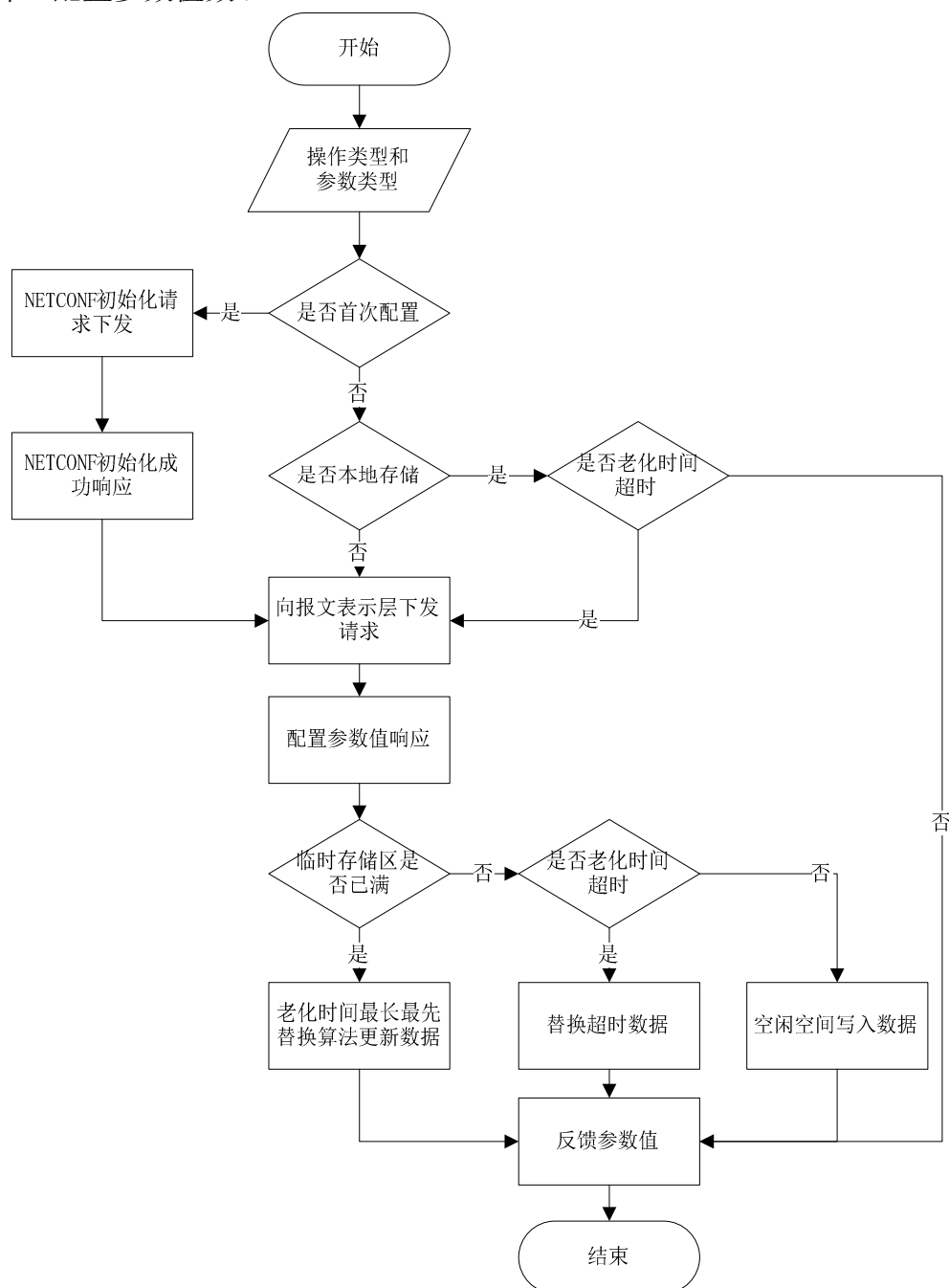


图4.4 控制接口层流程图

业务配置层将控件名和模块名字符串数据传递给页面调度模块请求调用控件和

响应模块的数据声明文件，页面调度模块从将以控件名字符串为入参，在控件模块查找对应的控件模块的全路径文件名，查找成功以后返回给页面调度模块进行相关控件库文件的引用工作，这是从业务配置层输入的第一条数据在控制接口层的处理。

如图 4.4 所示为控制接口层的流程图，主要分为以下步骤：

1. 获取操作类型字符串和参数类型数组。
2. 检查初始化标志是否置位，如果未置位，当前步骤跳转到第 5 步执行；如果已经置位，接着执行第 3 步。
3. 检查本地存储是否有该配置参数类型的数据值，如果没有当前需要的参数类型的数据值，当前步骤跳转到第 7 步执行，如果查询匹配到对应的参数类型的数据值，接着执行第 5 步。
4. 检查该配置参数类型的值对应的老化时间是否过期，如果已经过期，则当前步骤跳转到第 7 步执行，如果未过期，则当前步骤跳转到第 14 步执行。
5. 向报文表示层下发初始化连接请求
6. 初始化连接请求成功响应
7. 向报文表示层下发请求，获取该参数类型的数据值
8. 报文表示层成功返回该参数类型的数据值作为响应
9. 检查存储配置参数类型对应的数据值规定的存储空间是否已满，如果当前可供存储的空间已满，则执行第 10 步；如果可供存储的空间未满，则继续执行第 11 步。
10. 根据老化时间最长最先替换算法进行数据更新
11. 检查该参数类型的数据值的老化时间是否已经过期，如果该老化时间超出预定值，则说明该数据已经过期，有可能当前配置值已经发生改变，所以当前步骤跳转到第 13 步；如果老化时间未过期，则继续执行第 12 步。
12. 选择最先查询的空闲位置进行数据存储，并初始化老化时间。
13. 替换该过期数据值，并重置老化时间。
14. 向业务配置层反馈给配置参数类型的数据值，至此完成控制接口层的数据处理的全部过程。

4.2.3 报文表示层设计

报文表示层的设计分为两个主要的部分，即报文封装处理和报文解析处理。报文封装模块的处理由图 4.3 所示，其数据来源主要有三种。第一种是 hello 报文，也就是 NETCONF 连接初始化报文；第二种是 NETCONF 释放类报文；第三种是数据操作类型报文，此类报文是真正的配置参数类型报文，报文表示层处理的报文的主要类型就是该类型的报文。如图 4.5 所示为报文封装模块的处理流程图，当数据报文到来

的时候，首先进行逻辑判断，是否是数据类报文，如果是操作数据类报文，就开始构造数据请求报文体，对应 NETCONF 报文的操作层内容；如果判断不是数据类报文，需要再次判断是否是初始化连接请求，如果是初次请求 NETCONF 连接，就需要构造初始化连接报文，如果不是就需要构造释放连接报文体，最后再次封装 rpc 报文头，完成 NETCONF 报文的整个封装过程。

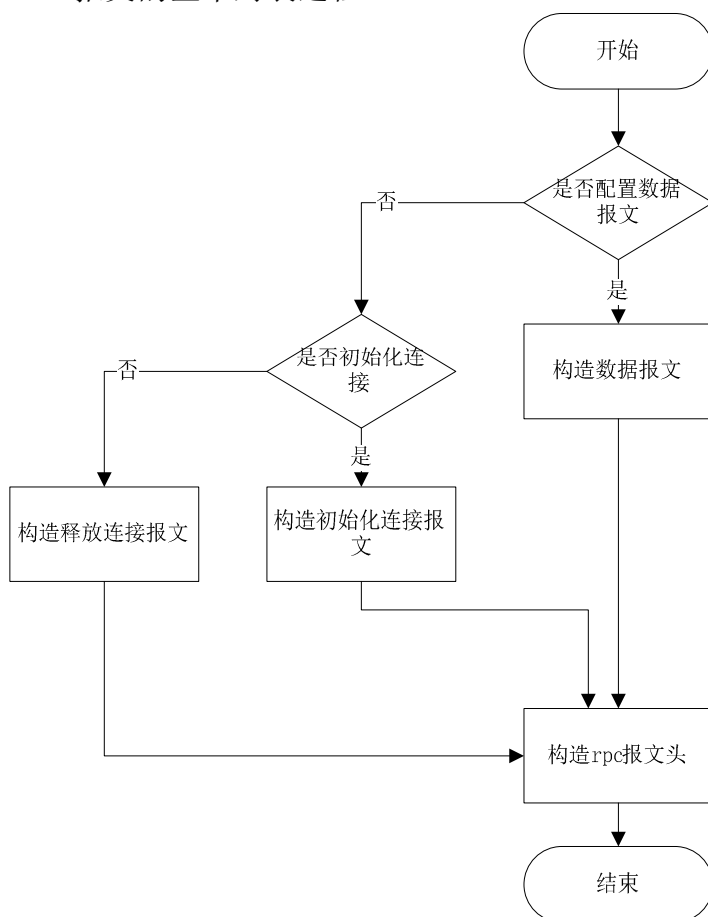


图4.5 NETCONF 报文封装模块流程图

如图 4.6 所示为 NETCONF 报文解析模块流程图，因为 NETCONF 报文解析模块的数据来源是来自通信会话层的报文，该报文是 NETCONF 报文的完整版。所以待该数据进入报文解析模块的时候，首先需要解析 NETCONF 报文头，即 rpc 报文头。解析完报文头部以后，进行逻辑判断，是否为配置参数响应类数据体报文内容，如果是，解析出数据的报文的内容，即配置参数值数据，传递给控制接口层以便于做后续的操作。如果不是数据响应类报文，需要再次作逻辑判断，该报文是否为初始化连接响应类报文，如果不是初始化连接响应，就意味着是释放连接响应，需要提供释放连接成功响应给控制接口层；如果是初始化连接响应类报文，即 hello 报文的响应类报文，

由于 hello 报文的响应类报文中包含后台 NETCONF 报文支持的操作类型能力，

需要解析该报文的内容,进行逻辑判断,是否支持基本操作,如果不支持该操作类型,主动释放本次连接,并提示连接失败响应给控制接口层,如果支持该基本操作类型,提示连接成功响应给控制接口层。

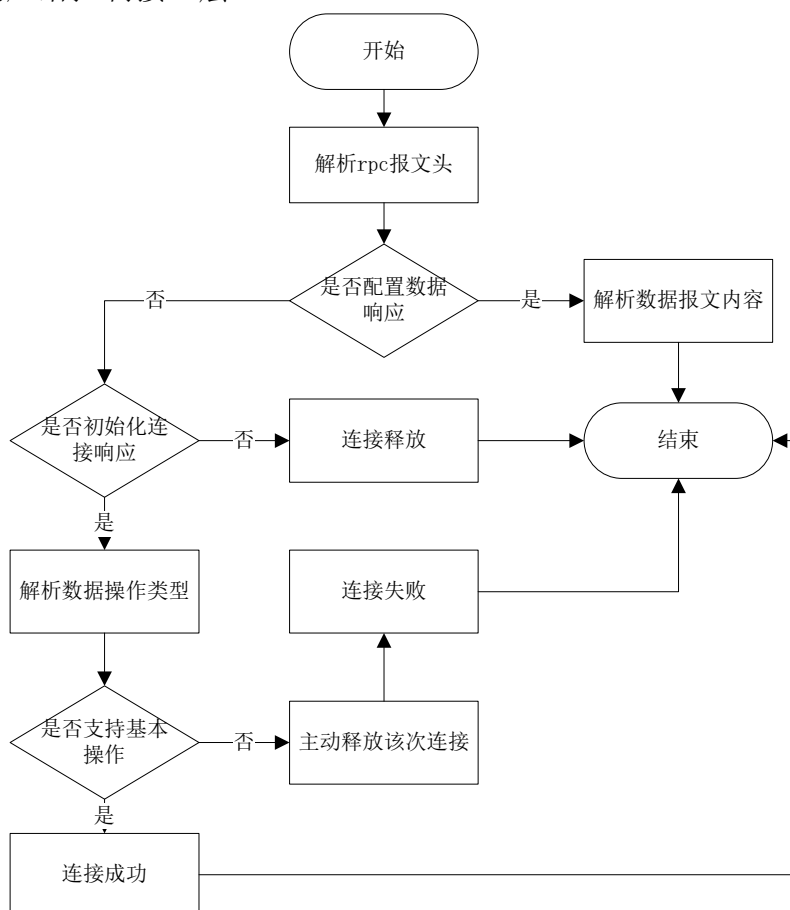


图4.6 NETCONF 报文解析模块流程图

4.2.4 通信会话层设计

通信会话层负责 NETCONF 会话的控制。由于 NETCONF 协议是建立在应用层之上的网络管理协议。为了保持通信会话的可靠性和安全性, NETCONF 会话是面向连接的,所以在通信会话层不仅要完成数据报文的传递,而且要维持通信会话的连接保持。所以通信会话层设计为三个部分:会话过程控制、NETCONF 连接的初始化与释放和报文的传递。

如图 4.7 所示为通信会话层的流程图,该通信会话层的数据来源来自报文表示层的 NETCONF 封装好的报文。当来自报文表示层的报文数据到来的时候,通信会话层首先会检测计时器是否超时,如果超时,需要调用 NETCONF 会话保活模块,进行 NETCONF 连接的重新连接包活,这个过程完成以后计数器清零,继续开始重新计数。如果没有超时,向后台服务器发送该 NETCONF 报文,然后检测响应,接受响应报文,

传送给报文表示层做进一步的处理，这就是整个 NETCONF 通信会话层的流程设计。

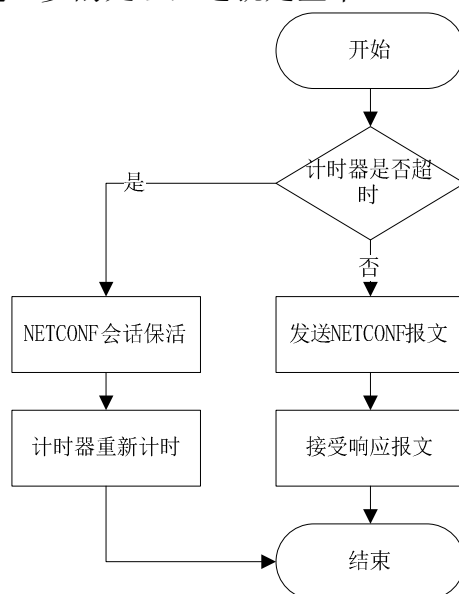


图4.7 通信会话层流程图

会话过程控制的目的主要是为了保持 NETCONF 会话的可持续性，而 NETCONF 连接的初始化与释放的作用显而易见，由于 NETCONF 协议的规定，在 NETCONF 连接建立之前必须有初始化过程，而 NETCONF 释放阶段，也要分为主动释放和被动释放两种情况。主动释放是当前连接向 Web 服务器请求主动释放本次的 NETCONF 连接，以便于服务器可以提供给其他用户连接请求。而被动释放是别的连接给 Web 服务器发送请求关闭其他的 NETCONF 连接，自己要独占服务器资源，这个时候服务器要发送通知报文给其他管理端以便于其他管理端资源的释放和请求的写入，防止因为连接的中断出现数据的丢失等情况。报文的传递主要负责报文向后台服务器的下发请求以及获取后台服务器的响应。

4.3 管理端的具体实现

4.2 节对管理端的结构进行了分析与设计，又将管理端细分为了四层，每一层负责不同的功能，降低了各个功能模块的耦合性，使得各个模块的开发可以并行操作，在开发上只要规定好接口规范，大大缩短了开发的时间，降低了开发成本，还能有效提高代码的重用性。由于不同层之间数据的隔离性，提高了代码的安全性，新的版本或者 NETCONF 协议有了新的规定的时候，只需要重新开发对应的关系层，方便了系统的移植和硬件版本的更新。

管理端的实现可以将网络管理配置的具体内容看成是数据，而把 NETCONF 协议的实现看成是逻辑，通过逻辑完成数据下发的过程就是完成管理端实现的主要内容。

4.3.1 通信会话层

通信会话层主要的功能是跟 Web 服务器进行通信,由于本系统是基于 NETCONF 协议开发的,所以在通信会话层要完成 NETCONF 连接的建立和释放,建立连接以后,封装好的报文,也是通过通信会话层发送给 Web 服务器,当然,Web 服务器反馈回来的 NETCONF 报文也是要通信会话层负责接收的。不仅如此,通信会话层还要管理 NETCONF 会话过程,包括 Web 浏览器需要定时保持连接持续性,用户长时间不操作超时退出登陆等。

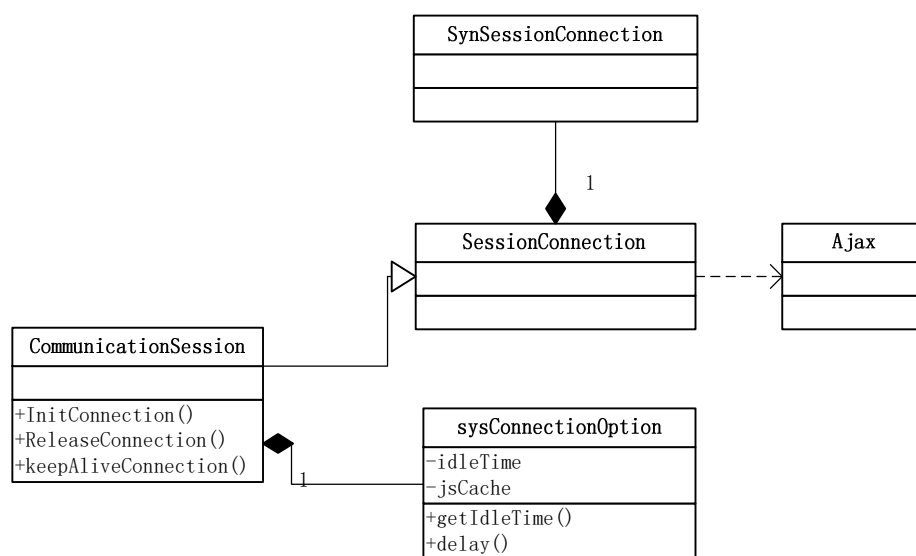


图4.8 通信会话层类图

如图 4.8 所示为通信会话层类图, `CommunicationSession` 对象实现了通信会话层的主要功能。它包含三个方法, `InitConnection`、`ReleaseConnection`、`keepAliveConnection` 分别为初始化 NETCONF 连接, 释放 NETCONF 连接和保活连接的持续性。将通信会话的共同的会话连接方法独立出来成为父对象, 隔离了通信会话的具体手段, 将具体的通信手段如 `Ajax` 独立成父对象, 在设计上降低了耦合性, 假如通信协议发生变化的, 只需要更改 `SessionConnection` 对象, 不用更改整个通信会话层结构设计, 就可以实现代码重用, 在设计上可以将通信会话管理内容, 放在独立的父对象中减少了代码的复杂性, 利于以后维护, 符合标准编程规范。`SessionConnection` 对象调用 `Ajax` 对象完成报文的发送和接受。而且 `SessionConnection` 对象的一部分 `SynSessionConnection` 对象完成通信会话的同步, 本来 `Ajax` 也提供了同步的手段, 但是 `Ajax` 同步的手段太过于低效, 而且有特殊的同步需求, 希望能保留 `Ajax` 异步的优点的同时, 有少许请求依靠前一个请求的回来的结果, 来发送下一次关联请求的报文的数据处理。在这里同步连接请求, 实现方法有很多种, 本系统采用信号量机制来完成连接请求的同步功能, 如果硬件系统资源充足的话, 也可以采用其他同步手段, 在

这里只需要重写 `SynSessionConnection` 对象即可。因为 `keepAliveConnection` 方法需要每隔一段时间，`getIdleTime` 获取系统空闲时间，即 `keepAliveConnection` 方法用户活动时候每隔 5 秒钟就请求一次连接，保持连接的持续性，当用户长期不操作时，调用 `getIdleTime` 全局变量 `idleTime` 查看系统空闲时间，如果超时，就调用 `ReleaseConnection` 方法释放连接，如果没有超时，调用 `sysConnectionOption` 对象的 `delay` 方法过一段时间检查继续是否超时。`InitConnection` 方法是连接初始化过程，由于 RFC4741 规定 NETCONF 会话利用 SSH 安全协议进行加密，脚本采用了开源工具 OpenSSH，生成了一对公钥密钥中的公钥发送给 Web 服务器，Web 服务器将公钥复制到 OpenSSH 所在目录下的 `authorized_key` 文件中，然后脚本文件和服务器建立 SSH 隧道。然后会在临时文件 `jsCache` 寻找 `sessionid`，如果找到发送给服务器端进行确认，如果没找到就发送请求 Hello 报文，获取到 `sessionid` 以后，保存到 `jsCache` 中，方便以后再确认处理。`ReleaseConnection` 方法实际上是发送 `<close-session>` 报文请求正常关闭连接，注 Hello 报文和关闭报文都是在本层完成处理，减少上层模块的报文处理，使得报文表示层的开发更加独立于 NETCONF 操作报文的封装和解析。

4.3.2 报文表示层

报文表示层，顾名思义，就是专注于报文的处理。NETCONF 操作在具体的实现上，就是 NETCONF 报文的不同。报文表示层实现的具体功能就是 NETCONF 报文的封装和解析。根据第二章对 NETCONF 的协议的概述可以知道 NETCONF 报文的具体格式，而 NETCONF 报文的封装和解析是相互的逆过程。

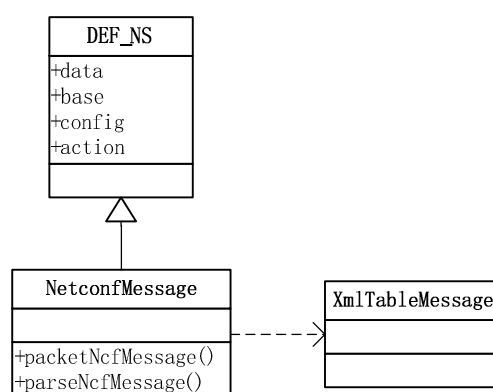


图4.9 报文表示层类图

如图 4.9 所示的报文表示层实现的类图。`NetconfMessage` 对象的方法主要有两个，`packetNcfMessage` 方法和 `parseNcfMessage` 方法。`packetNcfMessage` 方法主要负责封装 NETCONF 报文，封装的时候主要是调用了 `XmlTableMessage` 对象来完成封装 NETCONF 报文，`XmlTableMessage` 对象的主要数据结构如下：

```
var XmlTableMessage = {
    "get": makeGetXml,
    "edit-config": makeEditXml,
    "action": makeActionXml,
    "load": makeCfgFileXml,
    "rollback": makeCfgFileXml
};
```

packetNcfMessage 方法按照 XmlTableMessage["get"]("get",aTables,SType)这种方式调用来完成相关操作的 XML 报文的封装,然后再组成完整的 NETCONF 报文。DEF_NS 对象是 NETCONF 数据类型声明,是组成 NETCONF 报文不可获取的一部分,主要包括 base、data、config 和 action,表示基本类型字符串、运行数据类型字符串、配置数据类型字符串和运行数据类型字符串。parseNcfMessage 方法完成 NETCONF 报文的参数的解析,它将 NETCONF 报文参数值从字符串中解析出来返回结构体数组值,方法是利用正则表达式完成字符串解析。

4.3.3 控制接口层

控制接口层主要就是给业务配置层开发提供主要接口的。从业务配置层开发需求来分析,将控制接口层分为 5 个主要功能:页面调度、数据操作管理、控件管理、存储管理、安全验证和工具接口管理。页面负责整个业务配置层中每一个功能模块开发的 js 文件的执行流程、需要的 js 文件的加载、系统导航栏的读入等。数据操作管理提供操作业务配置层开发功能模块时定义的配置参数类型以及操作下层报文表示层报文的接口。控件管理是界面开发所需要的显示控件的接口,在这里设计了接口,可以用系统封装好的显示控件,也可以自定义控件或者利用开源控件来完成业务配置层各个模块的功能的开发,让页面更有表现力,显示更加丰富的内容。存储管理,提供本地临时存储区用于本地临时文件的存储,提高系统的使用效率,如果发送了两次相同的请求,会将前一次的请求数据暂存于本地临时存储区,当第二次查询的时候,就不用再向服务器下发相同的请求操作,只需要查询临时存储区就可以完成查询。安全验证,检测非法数据的请求让非法数据下发到设备之前就被检测出来,提高系统的安全性和可靠性。工具接口管理提供开发业务配置层功能所需要的常用接口,例如检查参数格式、配置提示对话框、页面复位向、获取 html 页面字符串等。

如图 4.10 所示为页面调度类图,页面控制主要完成三份工作,当前页面函数执行流程控制、相关 JS 文件的加载和导航栏的加载。TaskDispatch 对象的执行主要包括三个步骤,CurPagesFuncManagement 对象主要完成 init 函数和 destroy 函数的执行控制,在页面初始化的时候首先执行 init 参数指向的初始化过程,destroy 参数在页面

跳转向另一个页面的时候,或者关闭系统的时候,完成全局变量的释放或者控件显示的参数改变,比如 css 样式的还原等等。JSPagesManagement 对象负责加载相关 JS 文件,主要还是继承 FilesOperateManagement 对象完成找到文件目录的过程,这样设计的原因是当加载别的类型文件的时候,只需要改变 JSPagesManagement 对象的参数,用一个别的类型的对象去继承 FilesOperateManagement 对象就可以实现。具体加载的时候可以调用 Signal 对象完成文件加载过程控制。NavigationBar 是导航栏对象,完成导航的初始化,通过加载一个 XML 文件,这个 XML 文件是一个导航目录的页签内容,通过修改 XML 文件来完成不同产品不同功能的页面定制。

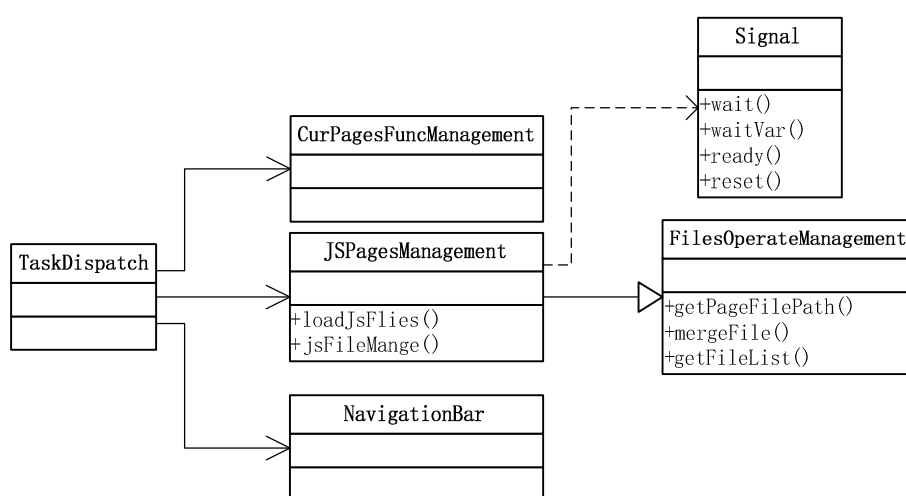


图4.10 页面控制类图

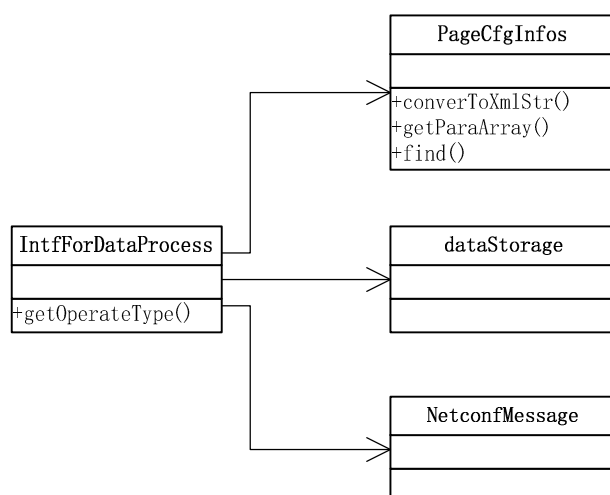


图4.11 数据操作管理类图

如图 4.11 所示为数据操作管理类图。IntfForDataProcess 对象提供了数据操作的

接口，该接口主要是获取用户的数据操作类型，读取对应功能模块的参数配置文件的变量数组，将操作和数组转化成字符串发送给报文表示层，来完成 NETONCF 报文的封装操作，另外该对象还提供报文表示层解析出来的参数数组，提供给业务配置层用于根据具体业务进行数据处理显示。

控件管理中的控件这里特指那些用来定义界面的显示类的部件。自定义了一部分的显示控件，用于业务配置层开发界面的时候统一显示界面，当然这里也可以用成熟的开源插件，在页面调度 TaskDispatch 对象成员 Widget 中加入该开源文件名称，就可以加载该开源文件，然后在业务配置层使用该控件作显示功能就可以了。自定义的控件要在编译的时候放入 Widget 默认的路径中，才能被正确找到加载。

当两次请求数据都一样的时候，可以将前一次的请求结果本地存储，第二次获取的时候，就可以先在浏览器本地查询，提高获取数据的效率，这里的数据不仅仅指配置数据，加载的控件库文件也可以暂时存在本地，但这样做也存在潜在的问题，就是当另外一个用户同时访问设备并且下发修改操作以后，本地存储的数据跟实际真实的数据将会出现不一致性。所以，设置本地存储的接口，开发业务配置层功能的时候，想要本地存储数据的时候，一定要定期检查数据生存时间，及时更新数据，防止因为数据不一致，而导致配置下发失败或者更严重的后果。

在本地存储模块的开发上，有两种选择方案：一个是利用 Cookie 进行存储临时数据，另一种是利用 HTML5 的特性的 localStorage 对象^{[28][29]}。由于 Cookie 受限于 4KB 的存储空间，而且使用 Cookie 技术在报文连接请求上会大大增加 Web 服务器的负担，这对于嵌入式 Web 服务器来说，明显影响了系统的性能，降低了工作的效率。所以说在本地存储的方案选择优先选择 HTML5 的 Web Storage 特性，在不支持 HTML5 的浏览器上将 Cookie 作为备选方案来使用^[30]。

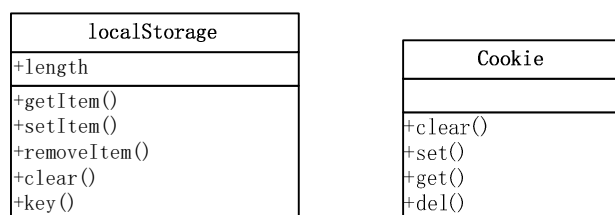


图4.12 localStorage 对象和 Cookie 对象

本系统实现采用 Web Storage 技术中的 localStorage 对象来实现本地存储方案，如图 4.12 所示为 HTML5 提供的 localStorage 对象的基本操作和属性。属性 length 是该对象唯一的属性，表示存储区的键值对数量；getItem(key)方法表示获取键值 key 对应的 Value 值；setItem(key,value)方法是为存储区添加新的键值对；removeItem(key)是删除对应的键值对，key(index)是根据 index 获取对应的键名。

如图 4.13 所示为数据管理对象 `dataStorage` 的实现，`dataStorage` 对象的操作的数据结构为 `DB_DATA`，而它主要的实现技术就是前面提到的 HTML5 技术和 Cookie 的备选方案，主要的属性 `length` 为当前规定的存储空间的大小，这个大小需要根据具体的访问时间定的，空间太大容易造成访问时间的延长，这样的设计反倒违反了当初设计的初衷。`LTPReplace` 为最长时间最先替换算法的实现方法，`getFirstFreeSpace` 方法实现了查找第一个空闲，`setData`、`delData` 方法实现数据插入和数据删除操作，`isEmpty` 方法实现判断当前规定的存储空间是否均未赋值。

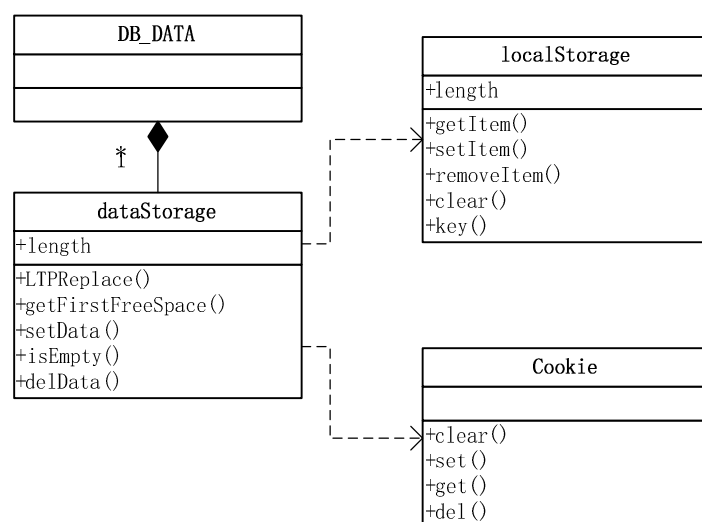


图4.13 本地存储数据库

DB_DATA:

```

{
    db_1: //数据库名称，每创建一个数据库就多一个该对象
    {
        attr: //本数据库的属性
        {
            flag: "read|write",
            retentionDuration: 100
        },
        keys: //本数据库的 keys
        {
            key_1: //具体的一个 key，调用 set 时会增加一个 key
            {

```

attr: /*本 key 的属性，包括事件通知数组，调用 `regNotify` 时会增加一个，调用 `deRegNotify` 时会删除一个*/

```

    {
        notifyCb: [pf1, pf2, pf3]
    },
    value: "abc" /*key 对应的值，调用 set 时会改变该值，该值的类型
由使用者决定*/
    },
    "cpu": //key 2 ...
    {
        //...
    }
}
},
db_2:
{
    //...
}
}
```

工具接口管理主要是为业务配置层开发功能模块提供工具接口，提高代码的复用性，降低由于代码的不规范造成的诸多 bug，为整个网络管理系统的开发降低不必要的风险。这个接口可以随时合入复用性高的工具代码，目前合入的代码有系统调试对象、输入参数合法性检测对象等，如图 4.14 所示。

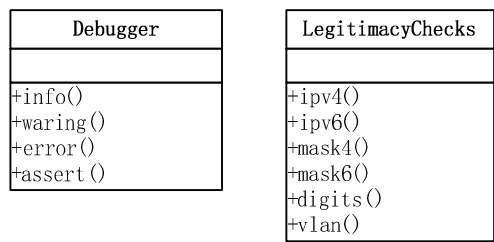


图4.14 工具类接口示例

4.3.4 业务配置层

业务配置层主要负责无线网络具体的业务配置，牵扯到具体无线网络管理的功能的内容，这个参数的确定是由后台 NETCONF 配置模块对具体无线网络功能的配置参数的设定，开发的过程是和后台 NETCONF 功能模块的配置参数关联起来的，所以后台提供的配置 NETCONF 属性表是前台开发业务配置层页面的依据。业务配置层的功

能开发都是类似的,后台有多少个NETCONF配置插件,即有多少个模块的NETCONF配置表,前台就需要开发多少个对应的功能模块,它们两者之间的关系是一一对应的。关于业务配置层的功能模块的开发,因为所要后台分配的NETCONF配置插件过多,前台也需要很多的功能模块的协同开发,但是它们的开发的步骤和原理都是类似的,所以在这里只用WM2U模块为例,介绍业务配置层各个业务功能模块的开发。

WM2U就是无线组播转单播技术,无线组播转单播的技术就是杭州华三通信技术有限公司提出就是为了解决组播技术的可靠性不足的,将组播帧在AP端变成单播帧的组播优化项目。组播优化表项中记录组播IP地址内包含哪些客户端的MAC地址,如果AP端收到的某组播IP地址在组播优化表项中没有记录,就不参与组播转单播的优化,直接以普通组播的方式传播。如果在组播优化表项查找到该组播IP地址,就将该组播IP地址下的所有客户端的组播报文转化成单播报文。

WM2U,即无线组播转单播,就功能上来说要提供给网络管理员组播流的监控,以便能够合理配置组播转单播。组播转单播的功能,不是对于每一个组播流,都要转换成单播,这样会让网络节点的负载过重,导致网络崩溃。所以有效监控每一条组播流,合理配置组播转单播阈值就是WM2U模块实现的两个主要目的。基于这一点,将WM2U模块的开发分为两个主要的功能:组播视频流流监控和WM2U功能配置。

WM2U模块由于在后台的NETCONF配置插件只有一个,所以在前台的虽然设计了两个小的模块,但是它们的数据配置参数类型却都属于同一个模块。功能模块的数据配置参数类型都是要通过在PageCfgInfos对象读入进行管理,所以在设计PageCfgInfos对象的时候规定了功能模块的数据配置参数类型的格式,这样PageCfgInfos对象就可以读入数据配置参数类型,进行管理封装NETCONF报文或者解析NETCONF报文,通过通信会话层跟后台的NETCONF配置模块进行交互。所以PageCfgInfos对象决定了参数的类型,为了方便数据的管理,所以在实现业务配置层各个功能的时候,首先定义NETCONF配置参数类型,并且将它们定义在单独的一个文件中,便于数据管理的模块化,参数配置定义文件的数据结构如下:

```
var PageInfo = {
    NC:
    {
        第一个配置的参数变量;
        第二个配置的参数变量;
        .....;
    }
};
```

根据WM2U功能模块需要的配置参数类型,需要配置的参数配置类型一共有10

个：记录了这个网络上所有访问网络的客户的加入了多少组播组的客户端参数变量、记录了组播组的参数变量、记录了发送组播报文的源地址的情况的参数变量、记录了服务模板下的 WM2U 功能的状态的服务模板变量、记录了该 AC 中建立的所有服务模板的基本属性的参数变量、记录了组播组老化时间的参数变量、记录了每一个客户端加入的组播组的数量限制的参数变量、记录了该网络系统中允许建立的组播组数量的参数变量、记录了每一个组播组允许加入的客户端数量的参数变量、记录了组播速率限制的参数变量。例如如下为客户端参数变量的数据结构：

```
Client: {
    nodes: ["WM2U","Clients"],
    row: "Client",
    index: ["MacAddress"],
    column: ["GroupNum4","GoupNum6"],
    menus: ["M_MonitoringMulticast"]
}
```

MacAddress 分量记录了该客户端的物理地址，GroupNum4 分量记录了客户端加入的 Ipv4 组播组的数量、GroupNum6 分量记录客户端加入的 Ipv6 组播组的数量。

1. 组播视频流监控

组播视频流监控，由于要给网络管理员提供组播流监控功能。系统要关注每一条组播流，然后分析在这个无线网络上那些组播流访问量大，这不仅对于监控网络状况是非常有意义的，而且可以提供服务商对自己服务的优化依据。在组播流监控的时候，根据需求分析，分为 Ipv4 和 Ipv6 两种组播类型，需要根据不同的组播类型提供不同的视图给网络管理员用于操作，但是在业务逻辑上来说，Ipv4 和 Ipv6 的组播流除了 AddressType（地址类型）属性不同以外，其他的处理都相同，所以在下面的设计与实现中只以 Ipv4 的为例，Ipv6 类似就不做说明了。

欲实现监控视频组播流，需要监控组播流的以下元素：客户端、组播组地址、模式、源地址、老化时间。每个客户端可以加入多个组播组，同一个组播组的源地址也有可能不同，它们代表不同的服务器发送的视频流。组播组老化时间、不同源地址发送的视频流的组播流的老化时间。

如图 4.15 所示为业务配置层功能业务模块开发流程图，大概分为 3 个部分，第一部分选用基础库文件加载，利用控制接口层提供的接口加载该业务模块需要用的库文件类型。控件数据初始化关键在于数据业务的处理逻辑，其实就是界面需要用到的数据和 Web 服务器传送回来的数据的关系转换。控件数据初始化，不仅包括数据的

初始化，还包括控件关联的事件函数的编写。最后一步是相关资源的释放，这里面比较典型的的就是全局变量的释放以及页面与页面之间数据交互的时候，暂存在

DBM 中的数据释放。

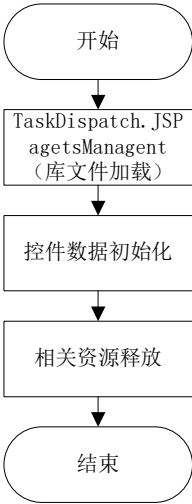


图4.15 业务配置层业务开发流程图

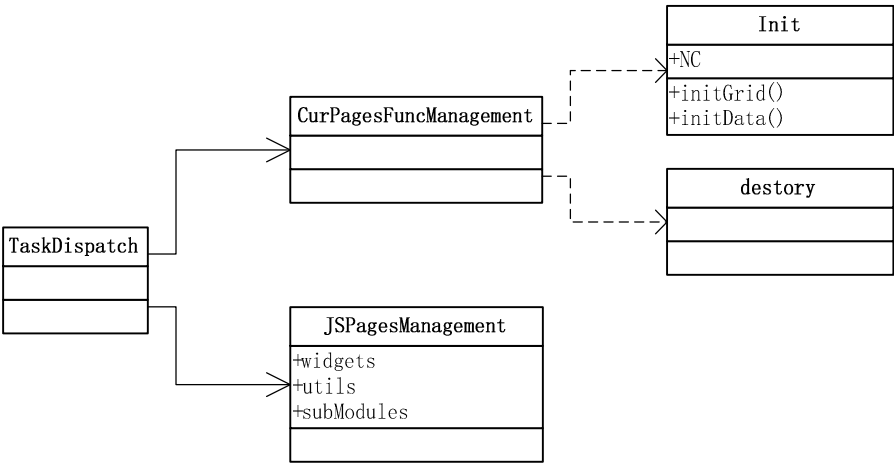


图4.16 业务配置层业务开发类图

如图 4.16 所示为业务配置层模块开发类图。页面调度对象 TaskDispatch 首先利用对象 JSPagesManagement 对象传入的参数进行页面的相关控件的加载，主要包括三个主要控件文件的加载，一个是显示控件，通过 widgets 参数加载显示控件，在 WM2U 模块的选用的显示控件为 Mlist；第二个是工具类控件，通过 utils 参数加载该类控件，这类控件文件通常会包括控制接口层提供的数据库操作对象、存储管理对象、安全验证对象和工具接口管理对象，在本模块中选用的为 IntfForDataProcess、DBM 和 Debugger；第三个是本模块的定义得配置参数类型文件，即在本模块开始之初定义到的本模块要用到的配置参数类型，包含 Group、Client 和 Source 变量的数据定义文件加载，通过 subMoudles 参数传递。

完成库文件的调度以后，接下来就是最重要的部分，初始化数据和控件内容的编

写了，即 Init 对象或者说过程的编写了。在 WM2U 模块的组播视频流监控这部分，主要包括 2 个部分，初始化控件和初始化业务数据。初始化控件相当于 MVC 框架中的 View 部分，而初始化业务数据相当于 Model 的部分。

初始化控件的主要目的是界面显示以及按钮事件函数的编写，在页面主要选用的显示控件是 Mlist。该控件在实际显示上是一个多列表控件，提供多个按钮接口。initGrid 方法主要完成对 Mlist 控件的初始化，主要包括定义列属性、关联按钮事件等。由于该部分的主要内容就是监控组播视频流，所以不牵扯到界面交互内容，所以不需要管理操作事件。在这里一共定义了 6 个列属性，分别是 MacAddress、GroupAddress、Mode、Version、SourceAddress 和 DurLastRefTime，表示客户端地址、组播组 Ip 地址、组播模式、组播协议版本、客户端地址和老化时间。

初始化业务数据，主要是请求后台传输过来的数据，将其转化成 Mlist 需要的数据。前台需要监控每一条视频组播流，所以对于每一个源地址发送来的组播流都需要处理。在初始化业务数据部分的操作，基本上分为 4 个步骤：

第一步，获取配置参数类型；通过控制接口层提供的 PageCfgInfos 对象的接口获取数据参数类型声明文件中的配置参数类型。在该部分主要获取 Client、Source 和 Group 参数，并调用 converToXmlStr 方法将其转化成 NETCONF 请求所需要的字符串。

第二步，请求配置参数类型对应的值；调用控制接口层提供的 NetconfMessage 对象的接口下发配置数据获取操作即"get"操作对应的报文，向 Web 服务器请求数据响应，并提供成功回调函数接口 myCallback 给第三步骤操作。

第三步，在回调函数中处理已经反馈回来的数据。在 myCallback 函数中调用 NetconfMessage 对象中报文解析方法 parseNetconfMessage 方法进行报文解析，获取 Client、Source 和 Group 对应的数组值。Client 对应的数组中的每一个元素包括 3 个属性：物理地址、加入的 Ipv4 组播组的数量和加入的 Ipv6 组播组的数量。Source 对应的数组值中的每一个元素主要包括 5 个属性：发送组播流的源地址、地址类型、组播组地址、客户端物理地址和该源地址发送的组播流的老化时间。

Group 对应的数组中的每一个元素包括 7 个属性：组播组地址、地址类型、组播组地址、组播协议类型、组播模式、发送该组播流的源地址数量和该组播组的老化时间。从这客户端数组中获取客户端地址，通过检查该客户端地址加入了多少个组播组，找到 Group 对应的数组中的物理地址（即 aClient[i]. MacAddress == aGroup[j]. MacAddress）相同的组播组属性，获取该客户端加入的组播组地址、当前组播组的老化时间、组播协议类型、组播模式，最后嵌套查找（即 aSource[k].GoupAddress == aGroup[j].Address&& aSource[k].MacAddress == aClient[i].MacAddress）该发送该组播组的组播流的源地址和该源地址发送的组播流的老化时间。通过三层循环查找，找

到对应的每一条组播流的数据，完成数据初始化操作。

第四步，刷新控件数据；通过 Mlist 控件的刷新操作，将第三步得到的结构体数组反馈到界面进行显示。

由于本块业务没有用到共享资源和全局变量，所以 destroy 对象赋值为空，不需要做资源释放操作。

2. WM2U 功能模块配置

WM2U 模块功能配置需要完成两项任务：第一项任务提供所有服务模板的使能/去使能 WM2U 功能，第二项任务提供 WM2U 功能全局参数的配置。这部分还是分为 Ipv4 和 Ipv6 两个方面，由于 Ipv6 和 Ipv4 的配置内容基本相同，所以在这里还是以 Ipv4 的配置为例说明。

要提供所有服务模板的使能/去使能 WM2U 功能，必须获取所有服务模板的服务模板名字，因为在后台 WM2U 使能/去使能的 NETCONF 配置插件只能获取使能 WM2U 状态的服务模板，所以需要从别的模块获取所有服务模板的模板名字。所以在配置参数类型文件中需要定义了无线用户接入模块 ServiceTemplates 变量。

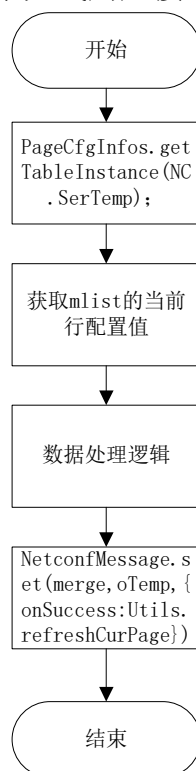


图4.17 使能/去使能事件处理函数流程图

全局参数的配置主要涉及到 WM2U 功能的老化时间(即该使能 WM2U 的服务模板的作用时间)、每一个客户端加入的组播组的数量限制、允许建立的组播组数量上限值、每一个组播组允许加入的客户端数量上限值以及达到上限的处理动作、组播速

率限制。

由图 4.15 所示，WM2U 功能模块配置的实现主要也分为两个方面，一个是控件初始化，另一个是业务数据初始化。

控件初始化，服务模板的使能/去使能选用的控件还是 Mlist 控件，一共有 2 个列属性，一个是服务模板名字号，另一个是 WM2U 状态。因为要支持服务模板修改 WM2U 状态，所以要定义按钮事件函数。如图 4.17 所示为按钮事件处理函数的流程图，首先通过 PageCfgInfos 对象的 getTableInstance 方法获取当前配置参数类型的对象，包含三个分量：服务模板号、WM2U 状态、地址类型。第二步获取 mlist 的当前行配置值。第三步实现数据处理逻辑，将 mlist 的界面获取的配置值转化成配置参数类型值。最后一步实现数据向服务器的下发；实现的时候是调用 NetconfMessage 对象的 set 方法实现配置数据的 merge 报文下发给 Web 服务器，成功以后刷新页面，显示配置后的数据。

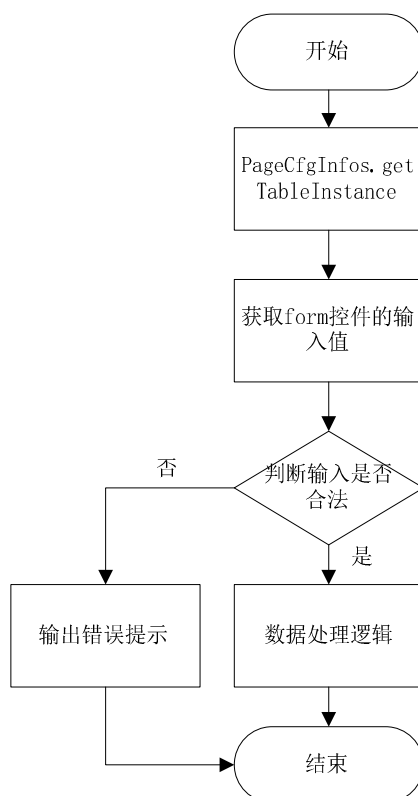


图4.18 WM2U 全局参数配置事件处理函数

实现 WM2U 全局参数的配置选用的是 form 控件，form 控件的包含 4 个子配置项：Aging Time（老化时间）、Optimization Group Limit（组播组优化限制）、Optimiztaion Client Limit（客户端优化限制）、Rate Limit（速率限制）。form 控件的提交处理按钮的事件处理函数的流程如图 4.18 所示。首先通过 PageCfgInfos 对象的 getTableInstance 方法获取全局配置参数类型的对象。第二步获取 form 控件的界面的当前配置值。第

三步,首先判断当前界面上每一个组播组的客户端数量上限值和达到上限值得处理是否都有输入值,如果两个输入值其中一个配置值没有输入,需要提示配置参数错误,以便于这两者配置下发时候,缺少必要的 xml 报文参数。在业务配置层实现报文的合法性、安全性验证。当提交按钮被触发的时候,需要判断是否输入框都有输入值(除去以上所述的组播组的客户端数量上限值),如果其中有输入框没有输入值,那么数据的时候都按默认值处理,例如老化时间限制为 60 到 3600 秒之间,默认值为 260 秒,如果没有默认值且没有输入值的情况下,需要利用 NetconfMessage 对象的 set 方法实现配置数据的 remove 报文下发给后台 Web 服务器,相当于下发给后 NETCONF 模块删除该全局配置,如果有输入值的话,首先要判断该输入值是否合法,例如组播组上限值在 8 到 8192 之间、每个客户端加入的组播组在 8 到 1024 之间。如果输入值合法,调用 NetconfMessage 对象的 set 方法实现配置数据的 merge 报文下发给后台 Web 服务器,并设置成功回调函数:如果配置下发成功,在输入框显示当前 WM2U 模块全局配置参数的值,以便于调整配置当前 WM2U 模块全局配置参数。

业务数据初始化,同样分为两个部分,第一个部分是使能/去使能 WM2U 状态的数据初始化,第二部分是 WMU2 全局参数配置初始化。这两部分数据的初始化过程同时处理,分为 4 个步骤:

第一步,获取配置参数类型;通过控制接口层提供的 PageCfgInfos 对象的接口获取据参数类型声明文件中的配置参数类型。在该部分主要获取全局服务模板、老化时间、客户端优化限制、组播组优化限制和速率限制参数对象,并调用 converToXmlStr 方法将其转化成 NETCONF 请求所需要的字符串。

第二步,请求配置参数类型对应的值;调用控制接口层提供的 NetconfMessage 对象的接口下发配置数据获取操作即"get"操作对应的报文,向 Web 服务器请求数据响应,并提供成功回调函数接口 myCallback 给第三步骤操作。

第三步,在回调函数中处理已经反馈回来的数据;在 myCallback 函数中调用 NetconfMessage 对象中报文解析方法 parseNetconfMessage 方法进行报文解析,获取服务模板、老化时间、客户端优化限制、组播组优化限制和速率限制的数组值。服务模板对应的数组中的每一个元素包括 3 个属性:服务模板名字、WM2U 的状态和 Ip 地址类型。由于获取到的 WM2U 的状态为 0 或者 1,而 Mlist 控件所需要的数据类型为 bool 类型,所以必须将其转换(aSerTemp[i].State = aSerTemp[i].State == 0 ? false : true)。

老化时间对应的数组值中的每一个元素主要包括 2 个属性:老化时间值、地址类型。客户端优化限制对应的数组中的每一个元素包括 3 个属性:每一个组播组加入的客户端数量上限值、达到上限值时触发动作和地址类型。组播组优化限制对应的数组中的每一个元素包括 3 个属性:组播组数量上限值、每个客户端加入的组播组上限

值和地址类型。速率限制对应的数组中的每一个元素包括 3 个属性：组播阈值、组播区间和地址类型。获取到的数据只需要提取数组中配置的具体值赋值给变量即可，不需要做过多转化。

第四步，刷新控件数据：通过 Mlist、form 控件的刷新操作，将第三步得到的数据反馈到界面进行显示。在 destroy 对象中释放全局变量，即可完成 WM2U 功能模块配置开发。

业务配置层的其他功能配置模块均跟 WM2U 模块的开发流程相似，主要是完成 Init 对象的开发，分为两个部分：控件初始化和业务数据初始化。总体来说，就是每个功能模块根据具体功能选择合适的界面显示控件，获取后台对应的 NETCONF 配置模块的配置数据，根据显示控件的数据要求处理后台反馈回来的配置数据，使得两者之间可以正常交互，就可以完成业务配置层所有的功能模块的开发。

4.4 本章小结

本章首先通过对无线网络管理系统的整体功能上的分析，提出了整体结构上的设计。重点对管理端进行了分析设计，利用分层设计的思想，将整个管理端的开发分为四个层次：通信会话层、报文表示层、控制接口层，并对每一个层次需要完成的功能进行了分析设计实现。最后完成整个无线网络管理系统的设计实现。

第五章 无线嵌入式网管系统的功能测试

前面各章完成了对无线网络管理系统的设计与实现，本章主要对该系统进行功能上的测试。首先搭建了测试该嵌入式 Web 无线网络管理系统的软硬件测试环境，环境搭建好了以后在该环境上针对无线网络管理系统的五个主要管理内容，分别设计了测试用例，以便于能够满足该网管系统的功能需求。

5.1 系统测试环境

5.1.1 软硬件测试环境

本无线网络管理系统所使用的硬件设备主要是型号为 WX5004 的 AC 设备，辅助测试设备有型号为 WA2620 的 AP 设备、交换机，计算机 PC 等。具体的组网图如 5.1 所示。AC1、AC2 设备通过交换机和 AP1、AP2 设备相连，服务器（该服务可充当 Radius/Portal 服务器）计算机连接在交换机上。将 AC1 在 VLAN1、VLAN2 上的 IP 地址设置为 192.168.21.100、192.168.21.99，AP1 在 VLAN1 上的 IP 地址设置为 192.168.21.101，AP2 在 VLAN1 上的 IP 地址设置为 192.168.21.102，PC 配置地址为 192.168.21.103，这样使得 AC、AP1、AP2 和 PC 之间能够相互 ping 通。

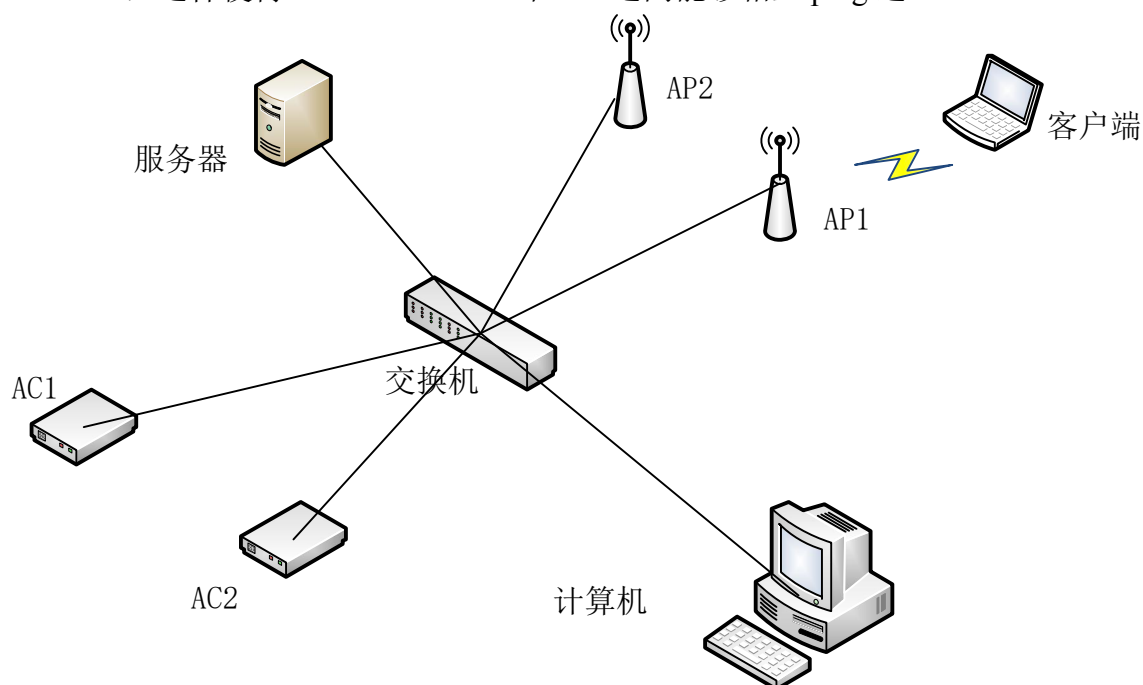


图5.1 系统测试组网图

AC 设备的运行系统是 Comware V7 系统，PC 通过在浏览器地址栏输入 "http://192.168.21.100"访问管理端进行网管系统的配置管理操作。浏览器选用 IE8、42.0.0.5780 版本的 Firefox 浏览器、46.0.2490.86 版本的 Chrome 浏览器分别进行测试。

5.2 系统功能测试

由于本论文的重点在于建立管理端的数据处理能力，所以只需对无线网管做功能测试，如图 5.2 所示为无线嵌入式 Web 网管系统的初始化页面视图，页面视图分为两个主要的部分，图示红色 1 区域为导航栏，该导航支持动态页面的定制；2 区域为网管参数配置和监控区。如图所示该网管系统导航栏分为十个主要的部分：快速配置、系统概览、设备管理、无线网络、无线服务、Radio 管理、认证管理、安全管理、Qos 和高级设置，该部分主要针对该十个部分设计测试用例，进行无线网络管理系统的功能测试。



图5.2 无线嵌入式 Web 网管系统

5.2.1 无线网络配置管理

无线网络配置管理包括四个模块：快速配置、无线网络、无线服务和 Radio 管理。针对每一个无线网络配置管理的子模块进行，进行分测试点测试每一项功能，如表 5.1 所示。

表5.1 无线配置测试用例

功能模块	测试点	测试步骤	实际结果
快速配置	设备 IP 地址 无线服务 Radius 服务器	1.在初始化页面导航栏选择快速配置	快速配置页面显示成功

表 5.1 (续) 无线配置测试用例

功能模块	测试点	测试步骤	实际结果
快速配置	Portal、密钥、AP 快速配置	2.在快速配置页面配置设备 IP 地址：IP 地址、掩码和网关，配置完成后，选择下一步 3.跳转到无线设置页面，配置主服务类型、无线服务和加密方式，选择下一步 4.跳转到 Radius 设置页面：设置主认证服务器 IP 地址、UDP 端口和密钥，设置计费服务器 IP 地址、UDP 端口和密钥，选择下一步 5.跳转到 Portal 设置页面，配置 Portal 认证服务器名称，服务器 IP 地址和端口号，选择下一步 6.跳转到密钥设置页面，配置密钥类型、密钥长度和密钥，选择下一步 7.跳转到 AP 设置页面，配置 AP 型号、名称和序列号，选择下一步 8.跳转到配置汇总页面，可以查看配置的全部信息，选择提交按钮	设备 IP 地址配置成功，无线服务设置成功，Radius 相关参数配置下发成功，Portal 认证配置成功，密钥下发成功，AP 配置成功且正常建立 CAPWAP 连接，以上测试结果都是通过型号为 WX5325 的 AC 设备用命令行监测成功
无线网络	MAC 地址表配置	1.导航栏选择网络，选择子项目 MAC 表项 2.选择 MAC 页签，该页面显示所有 MAC 地址表，该表项属性值有：MAC 地址、VLAN、端口和操作 3.删除/添加/查询该 MAC 地址表	删除 MAC 地址表项成功、添加 MAC 地址表项成功，可以正确筛选出对应的 MAC 地址表项
无线网络	MAC 地址表项详细配置	1.重复上述步骤 1，选择设置页面，选择不老化/填入老化时间，该时间要求在 10-630 秒，缺省为 300 秒	不老化时间下发成功/输入 400 秒的老化时间下发成功
无线网络	VLAN 配置	1.导航栏选择网络，选择子项目 VLAN 2.选择 VLAN 页签，该页面显示所有已建立的 VLAN 列表，表项有 VLAN 的 ID、描述、绑定物理端口 3.删除/添加/查询 VLAN 列表	删除 VLAN 表项成功、添加 VLAN 表项成功、可以根据 ID 筛选出 VLAN 表项

表 5.1 (续) 无线网络配置测试用例

功能模块	测试点	测试步骤	实际结果
无线网络	ARP 表项配置	1. 导航栏选择网络，然后选择子项目 ARP 2. 选择 ARP 页签，该页面显示所有已建立的 ARP 列表，表项有 IP 地址、MAC 地址、VLAN、绑定物理端口、类型 3. 添加静态 ARP 表项，删除动态/静态 ARP 表项，查询 ARP 表项	静态 ARP 表项添加成功，可以根据 IP 地址筛选出 ARP 表项、删除动态 ARP 表项成功，删除静态 ARP 表项成功
无线网络	WM2U 列表管理	1. 导航栏选择网络，然后选择子项目 WM2U 2. 选择 WM2U 页签，该页面显示所有服务模板的 WM2U 功能状态 3. 选择服务模板，使能/去使能服务模板	使能单个/多个服务模板成功，去使能单个/多个服务模板成功
无线网络	WM2U 全局参数配置	1. 重复上述步骤 1 2. 选择全局参数配置页签，该页面可以配置老化时间、组播组阈值、每一个客户端可以加入的组播组阈值、每一个组播组加入的客户端阈值以及达到阈值限制动作、速率间隔限制	WM2U 全局参数下发成功
无线网络	组播视频流监控	1. 重复上述步骤 1 2. 选择组播视频流页签，该页面显示组播视频流列表，一共有 6 个列属性：客户端地址、组播组 IP 地址、组播模式、组播协议版本、服务器地址和老化时间	组播视频流显示流成功
无线网络	接入服务	1. 导航栏选择无线服务，然后选择子项目接入服务 2. 选择新建无线服务，输入服务名，VLAN，加密方式 3. 选择绑定的 Radio 类型，使能 Radio，点击确定显示无线服务	无线接入服务配置成功

表 5.1（续） 无线网络配置测试用例

功能模块	测试点	测试步骤	实际结果
radio 管理	radio 配置	1.导航栏选择射频，再选择射频设置，选择 AP 进行射频详细配置 2.设置射频信道、功率和配置信道扫描	AP 射频配置生效，AP 射频参数配置成功

5.2.2 设备管理

设备管理主要包括两个子模块，其中一个设备信息配置，即 AC 本身设备；另一个是 AP 设备管理，以下针对这两个子模块设计测试用例，如表 5.2 所示。

表5.2 设备管理测试用例

功能模块	测试点	测试步骤	测试结果
设备信息	基本信息配置	1.导航栏选择设备，再次选择基本信息 2.设置系统名称，Web 闲置超时时间（该范围在 1-999，缺省值为 10），点击确定按钮	系统名称、Web 闲置超时时间配置成功
设备信息	日期和时间配置	1. .导航栏选择设备，再次选择日期和时间 2.填写日期和时间，点击确定	时间和日期配置成功
设备信息	日志管理	1.导航栏选择设备，再次选择日志管理 2.选择日志显示页签，显示当前日志按时间排列，通过输入日期可以查询日志	日志显示成功，可通过日期筛选日志成功
设备信息	配置管理	1.导航栏选择设备，再次选择配置管理 2.进行配置备份、配置恢复、保存配置和恢复出厂配置	配置备份操作成功、配置恢复成功、保存配置成功、恢复出厂配置成功
AP 设备管理	AP 设置	1.导航选择 AP，子项目再次选择 AP 配置 2.进行 AP 配置：AP 名称、序列号和型号	AP 配置成功
AP 设备管理	自动 AP 配置	1.导航栏选择 AP，选择自动 AP 配置	

表 5.2 (续) 设备管理测试用例

功能模块	测试点	测试步骤	测试结果
AP 设备管理	自动 AP 配置	2.进行自动 AP 配置：开启/关闭自动 AP 3.AP 列表显示：监测到所有 AP	自动 AP 开启成功、关闭自动 AP 成功
AP 设备管理	AP 组配置	1.导航栏选择 AP，选择 AP 组配置 2.进行 AP 组配置：AP 组 ID，AP 列表中选择 AP 加入	AP 组建立成功，AP 成功加入 AP 组

5.2.3 性能管理

性能管理分为性能监测和阈值设置两个小的子模块，由于性能监测是监测设备的性能，通常监测设备的 CPU 利用率、内存利用率、设备利用率等情况，利用设备参数分析系统资源利用率、设备接口信息等状态。如图 5.3 所示。

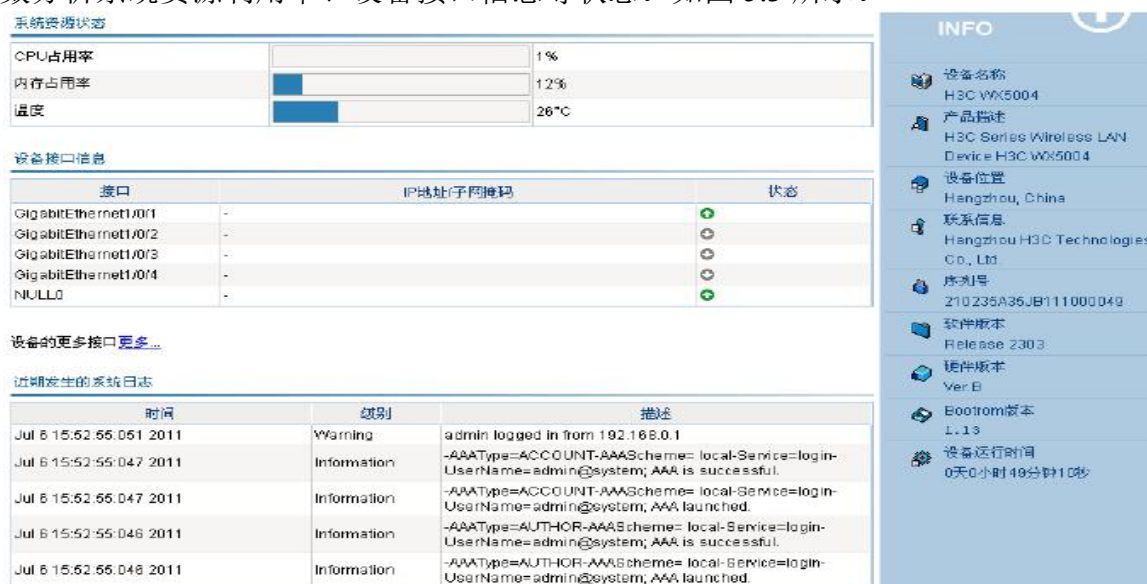


图5.3 性能管理

导航栏选择概览，设备信息，即可监测设备性能参数。从图 5.3 中可以看出，设备性能管理的主要指标都是通过设备接口信息和必要的设备参数来反应，从端口的状态可以查看到连接到 AC 设备上的各种无线网络组件的性能状态。

通过端口相关参数的状态，网络管理员可以优化网络状态，例如采用负载均衡技术等。

选择设备接口信息，如图 5.4 所示，选择第一个接口信息，如图 5.5 所示为 GigabitEthernet1/0/1 端口统计信息。通过在端口配置阈值，可以有效改善设备的设备

性能。



图5.4 设备接口信息

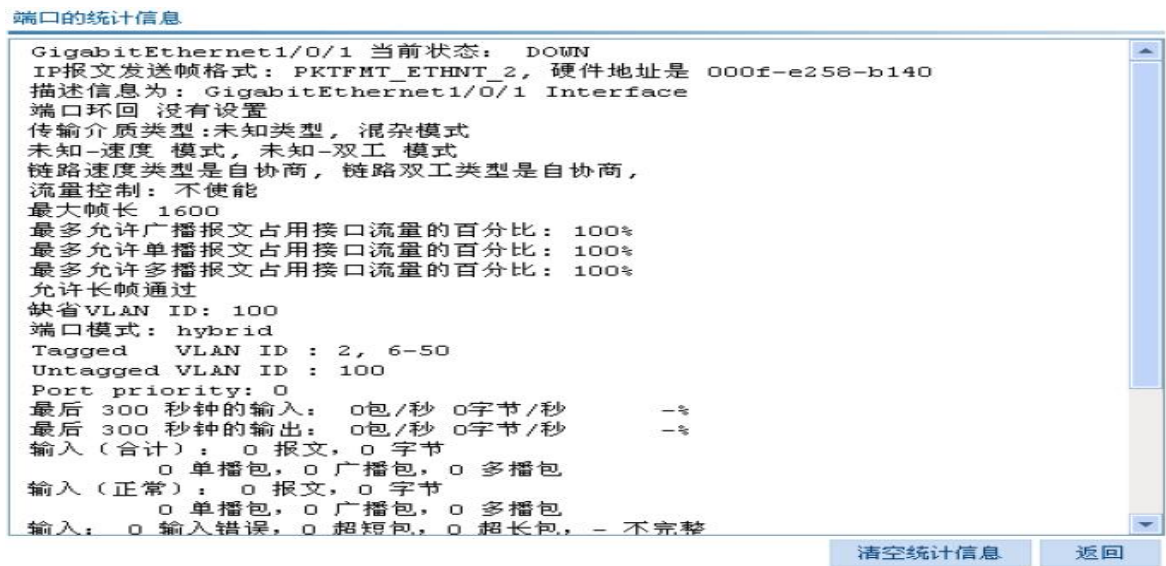


图5.5 端口统计信息

5.2.4 告警和安全管理

告警和安全管理细分为三个子模块，分别是认证管理、安全管理和 Qos，下面分别按照这三个子模块设计测试用例来验证告警和安全管理模块的功能的正确性。如表 5.3 所示为告警和安全管理测试用例设计。

表5.3 告警和安全管理测试用例

功能模块	测试点	测试步骤	测试结果
认证管理	802.1x	1.导航栏选择认证，子菜单选择	

表 5.3（续） 告警和安全管理测试用例

功能模块	测试点	测试步骤	测试结果
认证管理	802.1x	802.1x 认证 2.进行 802.1x 认证相关配置	802.1x 认证配置成功
认证管理	Portal 认证	1. 导航栏选择设备，子菜单选择 Portal 认证 2.进行 Portal 认证相关参数配置	Portal 认证配置成功
认证管理	Radius 认证	1.导航栏选择设备，子菜单选择 Radius 认证 2.进行 Radius 认证相关参数配置	Radius 认证配置成功
安全管理	安全配置	1.导航栏选择安全，子菜单选择安全管理 2.进行监测非法设备的参数配置 3.进行入侵监测参数配置 4.进行黑白名单参数配置	各个安全配置选项参数配置成功
安全管理	用户隔离	1.导航栏选择安全，子菜单选择用户隔离 2.进行用户隔离参数配置	在用户隔离列表中加入的用户被成功隔离
Qos	ACL	1.导航栏选择 Qos，子菜单选择 ACL 2.配置 ACL 基本/高级/链路层规则	ACL 规则配置成功
Qos	Qos 策略	1.导航栏选择 Qos，子菜单选择 Qos 策略 2.创建策略，绑定类和流行为 3.在端口/服务上应用该策略	策略配置成功，并正确应用

5.2.5 漫游移动管理

漫游移动管理是相比于有线网络管理配置，无线网络自己特有的地方。由第三章的需求分析可以知道，测试漫游移动管理主要从三个方面：漫游组、漫游组成员和客户端信息。

漫游分为 AC 内漫游和 AC 间漫游两种形式，所以在这个方面的测试用例的设计就要从两个方面来进行，也就是设计两种测试用例来测试三个方面的内容。

如表 5.4 为漫游移动管理的测试用例。

表5.4 漫游移动管理测试用例

功能模块	测试点	测试步骤	测试结果
漫游组	漫游组配置	1.导航栏选择漫游，子菜单选择漫游组设置 2.创建漫游组（关闭/开启漫游、地址类型、源地址、地址类型和认证密钥）	漫游组创建成功
漫游组	漫游组成员管理	1.导航栏选择漫游，子菜单选择漫游组设置 2.根据成员 IP 地址，添加进漫游组	Portal 认证配置成功
漫游客户端	漫游客户端	1.导航栏选择漫游，子菜单选择客户端信息 2.客户端在漫游组成员之间漫游	漫游成功的客户端成功在列表中显示，如图 5.6 所示。

客户端信息				
<input type="text"/> 客户端MAC地址 <input type="button" value="查询"/> 高级查询				
客户端MAC地址	BSSID	VLAN ID	宿主IP	漫游方向
0040-96b3-8a77	000f-e2b1-7bd0	1	192.168.1.101	漫入
<input type="button" value="刷新"/>				

图5.6 漫游客户端

5.3 本章小结

本章对无线网络管理系统的管理的内容进行功能测试，主要还是分为无线网络管理、设备管理、性能管理、告警和安全管理、漫游移动管理进行管理配置下发，测试用例全部通过。

第六章 总结与展望

6.1 总结

以前无线网络只是有线网络的辅助手段，但是伴随着最近几年，3G、4G 网络的快速建设，无线网络的地位不仅仅只是作为有线网络的补充了，那么面对当前这种情况，无线网络的管理就显得尤为重要了。虽然近几年，无线网络取得了骄人的进步，但是无线网络还有很长的路要走，而且无线网络管理的内容每天都在发生着变化。所以本文提出了建立无线网络管理系统的不仅仅要求它有稳定性、高效性，而且要有很好的可扩展性。

本文从有线网络管理的成熟方案为切入点，深入研究了建立网络管理系统所需要的结构模型，以及当前无线设备厂商公用的网络配置管理协议 NETCONF 等技术点。做了充足的技术储备以后，又规划了无线网络管理配置的具体内容。根据具体的内容，从当前的实际情况出发，整合了可以利用的资源，设计出了无线网络管理系统的整体结构。由于无线网络管理系统的服务器端以及具体模块的 XML 报文驱动模块都已经成熟的设计实现，所以在整个论文中只是简单介绍了该部分的内容。本文的侧重点在于浏览器的管理端部分的设计与实现，利用了层次设计的思想，将整体结构分为四个层次：通信会话层、报文表示层、控制接口层和业务配置层，有效降低了各层次模块之间的耦合性，在数据处理之前增加了数据验证阶段，数据的安全性也得到了保障。管理端这样设计还有一个好处是降低了嵌入式 Web 服务器端的负载，也满足硬件设备对嵌入式系统资源的要求。在本文的最后，针对完成的整个网管系统进行了功能方面的测试，完成整个毕业设计的工作。

6.2 展望

受限于设备的硬件资源的限制，当前建立的 Web 无线网络管理系统只能同时支持 5 个用户进行网络配置和操作，而且基于 Apache 的 httpd 2.4.17 的 Web 服务器在整个嵌入式系统的内存占用较高，这些问题都是在嵌入式 Web 网管系统中需要努力的方向。

参考文献

- [1] 肖德宝,陈历淼,艾翔等.下一代网络配置管理协议 NETCONF 的研究与实现[J].华中师范大学学报(自然科学版),2008,42(4):530-534.DOI:10.3321/j.issn:1000-1190.2008.04.007.
- [2] 王伟. NETCONF 下 DHCP Snooping 的设计与实现[D]. 北京: 北京交通大学, 2015.
- [3] 曾伟超,黄政力,丁三弟等.WLAN 网管系统规范的研究与应用[J].电信工程技术与标准化,2011,24(4):13-17.DOI:10.3969/j.issn.1008-5599.2011.04.003.
- [4] 刘会芬.基于 netconf 的网络管理者的研究与实现[D]. 武汉: 华中师范大学 2008.
- [5] <.Network configuration management using NETCONF and YANG[J].IEEE Communications Magazine: Articles, News, and Events of Interest to Communications Engineers,2010,48(9):166-173.
- [6] Network configuration management using NETCONF and YANG[J].IEEE Communications Magazine: Articles, News, and Events of Interest to Communications Engineers,2010,48(9):166-.
- [7] 刘成业,熊庆国,熊伟等.网络数据监控及管理[J].现代电子技术,2004,27(17):4-5.DOI:10.3969/j.issn.1004-373X.2004.17.001.
- [8] 赖杨灯.基于 NETCONF 的策略路由配置系统的设计与实现[D]. 北京: 北京交通大学 2014.
- [9] 孙铭浩. VXLAN 隧道的设计与实现[D]. 哈尔滨: 哈尔滨工业大学 2014.
- [10] 吴飞,何剑波,陈先宏. 嵌入式 Web 网管系统及其交互方法: 中国. [P]. 2007-03-14.
- [11] 师鸿博.基于 SNMP 协议的 Web 监控系统[D].南京邮电大学,2011.DOI:10.7666/d.d177413.
- [12] 姜本臣.基于嵌入式 Web 服务器应用技术的研发[D]. 沈阳: 沈阳工业大学 2012 .
- [13] 李捷,王汝传.基于 Web 的嵌入式网络管理平台的研究和设计[J].计算机科学,2004,31(1):62-65.DOI:10.3969/j.issn.1002-137X.2004.01.017.
- [14] 鉴纪凯.基于 NETCONF 的域名系统的设计与实现[D]. 北京: 北京交通大学 2015 .
- [15] 汤万博.企业党员信息管理系统的设计与实现[D]. 大连: 大连海事大学 2014 .
- [16] 刘正勇. 纸媒体客服信息管理系统的设计与实现[D]. 湖南: 湖南大学 2014.
- [17] 黄磊. 基于 Internet 的远程 PLC 控制研究[D]. 西安: 西安工业大学 2014 .
- [18] 梁伟民. 基于 itil 的 netconf 网络配置管理系统的研究与实现[D]. 武汉: 华中师范大学 2011.
- [19] 王金一,阎保平.基于 NETCONF 的 WSN 网络管理系统设计[J].计算机工程与科学,2011,33(8):1-7.DOI:10.3969/j.issn.1007-130X.2011.08.001.
- [20] 杨颖莹. 高性能 Web 框架的分析与应用[D]. 北京: 北京邮电大学 2012.
- [21] 乔彩霞. 前段设备 SNMP 网管系统的设计与实现[D]. 成都: 电子科技大学 2012.
- [22] 李志广. 基于 Web 的嵌入式网络管理系统的设计与实现[D]. 成都: 西南交通大学 2003.

- [23] 刘博. 嵌入式 Web 服务器的设计与实现[D]. 西安: 西安电子科技大学 2006.
- [24] 候洁. 嵌入式 Web 远程设备监控软件设计[D]. 成都: 电子科技大学 2011.
- [25] 王海军. 基于 Web 的网络管理系统的研究[D]. 武汉: 武汉理工大学 2005.
- [26] 史涛. 防火墙嵌入式 Web 网管服务器的设计与实现[D]. 北京: 北京邮电大学 2010.
- [27] 冯永亮.HTML5 本地数据存储技术研究[J].西安文理学院学报（自然科学版）,2014,(3):66-69.DOI:10.3969/j.issn.1008-5564.2014.03.018.
- [28] 李焯民.基于 HTML5 的前端本地化存储技术[J].成都大学学报：自然科学版,2012,31(1):67-69.DOI:10.3969/j.issn.1004-5422.2012.01.020.
- [29] 张坤,吕义正 .HTML5 客户端存储的安全性探析 [J]. 黄山学院学报,2014,(5):13-15.DOI:10.3969/j.issn.1672-447X.2014.05.004.
- [30] 王明超,陈榕.基于脚本的客户端软件结构化存储技术研究[J].计算机技术与发展,2010,20(2):25-28.DOI:10.3969/j.issn.1673-629X.2010.02.007.

致谢

时间过的很快，马上研究生的生活就要结束了。伴随着论文的完成，离校的日子也越来越近了。

感谢王献青老师在本论文的写作过程中的细心指导，王老师在论文的写作过程中给了我很大的帮助和鼓励，不仅指出了论文过程中应该注意的细节和研究方法，还对论文撰写过程中出现的问题应该如何修改、完善提出了很多的建设性意见，让我的论文撰写更加流畅和自然，也学习到了很多的解决问题的方法。对王老师不厌其烦的指导，只能在这里表达非常深刻的感谢，谢谢老师。

回顾两年半的研究生生活，有太多的人需要感谢，太多的事情值得回忆。感谢我的同学、舍友，在学习上和生活上对我的帮助。感谢研究生二年级的时候，在 H3C 实习的时候，我的企业指导老师牛建、项目经理赵廷海、同事袁晓杰、王炳志、田小丽、秦顺宇等对我工作和论文选题的帮助，真是太多的人需要感谢，只想说认识你们是我最大的荣幸。

最后想说，谢谢我的父母，是你们的爱，让我学会了坚强，懂得了学习和生活。

作者简介

1. 基本情况

路皓宇，男，甘肃庆阳人，1988 年 4 月出生，西安电子科技大学软件学院软件专业 2013 级硕士研究生。

2. 教育背景

2007.09~2011.07	长安大学	本科	机械设计制造自动化
2013.09~至今	西安电子科技大学	硕士研究生	软件工程