

桂林航天工业学院

课程设计存档材料

课程名称: 计算机网络课程设计

开课学期: 2019 — 2020 学年第二学期

专 业: 软件工程

班 级: 2017 软件工程 4 班

姓 名: 钟祯

学 号: 2017070030429

指导老师: 刘雷、张森森

报告日期: 2020 年 07 月 02 日

桂林航天工业学院课程设计任务书

设计题目：网络搭建仿真及聊天软件程序设计

学生姓名	钟祯		
课程名称	计算机网络课程设计	专业班级	2017 软件工程 4 班
地 点	巡天楼 324	起止时间	2020.06.29 —— 2020.07.02
设计内容	<p>一、搭建网络 设计一个网络拓扑结构，至少包含如下网络 网络 1：HTTP 服务器，能正常使用。 网络 2：FTP 服务器，能正常使用。 网络 3：DNS 服务器，能正常使用。 网络 4：一台主机。 网络 5：一个包含两个子网的网络，子网 1 有 50 个 IP 地址，子网 2 有 80 个 ip 地址，要划分子网，同时使用 vlan 将两个子网分类。 不同的网络通过若干个路由器连在一起，路由配置要求静态路由、RIP 和 OSPF 混合。 整个网络中任意两个机器都能交互数据。</p> <p>二、基于 TCP 或 UDP 的 C/S 网络聊天程序 基本要求：设计 Server 和 Client，有图形化界面，可多用户同时聊天。 最终效果：可实现聊天功能。</p>		
设计参数	<p>① 搭建的网络中路由器个数不得少于 4 个 ② 搭建的网络中划分的子网数不得少于 2 个 ③ 搭建的网络中至少包含 2 种应用层服务 ④ 聊天软件至少能 3 人同时在线</p>		
设计进度	<p>第一阶段：分析网络需求，设计网络拓扑结构。 第二阶段：完成网络拓扑结构设计、对服务器和交换机进行配置，包括 VLAN、静态路由、RIP、OSPF、HTTP 服务、FTP 服务、DNS 服务等。调试、测试搭建网络的连通性。 第三阶段：分析应用需求，搭建开发环境，进行程序的设计和编写工作。 第四阶段：对程序进行测试和调试。 第五阶段：整理资料，提交课程设计报告。</p>		
设计成果	课程设计报告		
参考资源	<p>[1] Andrew.S.计算机网络.清华大学出版社，2014 [2] 谢希仁.计算机网络（第七版）.电子工业出版社，2013 [3] James.计算机网络：自顶向下方法（第 7 版）.清华大学出版社，2018 [4] 李志远.计算机网络综合实验教程—协议分析与应用.机械工业出版社，2019</p>		

说明	<ol style="list-style-type: none">1. 本表应在每次实施前由指导教师填写一式 2 份，审批后所在教学单位和指导教师各留 1 份。2. 多名学生共用一题的，在设计内容、参数、要求等方面应有所区别。3. 若填写内容较多可另纸附后。
----	--

指导教师：刘雷、张森森

教研室主任：张新伦

2020 年 07 月 02 日



桂林航天工业学院
GUILIN UNIVERSITY OF AEROSPACE TECHNOLOGY

本科课程设计报告

课程名称: 计算机网络课程设计

开课学期: 2019 年 -- 2020 年第二学期

学 院: 计算机科学与工程学院

题 目: 网络搭建仿真及聊天软件程序设计

专业班级: 2017 软件工程 4 班

学 号: 2017070030429

学生姓名: 钟祯

指导教师: 刘雷、张森森

报告日期: 2020 年 07 月 02 日

成绩: _____ (五级)

课程设计评分标准

平时 (30%)	全勤、学习态度端正、实验认真、积极回答问题	优秀 (90-100)	
	偶有缺勤、实验认真、回答问题较积极	良好 (80-89)	
	旷课 2 次以内、偶有迟到、实验认真、回答问题较好	中 (70-79)	
	旷课 2 次以上、学习态度一般、基本能回答出问题	及格 (60-69)	
	经常旷课，实验过程不认真、问题回答不积极	不及格 (<60)	
考核 (70%)	作品功能完全实现、作品演示操作熟练、问题回答准确；设计论文结构合理、内容符合要求、论文格式符合规范要求、参考文献新	优秀 (90-100)	
	作品功能基本完全实现、作品演示操作熟练、问题回答准确；设计论文结构合理、内容符合要求、论文格式符合规范要求、参考文献新	良好 (80-89)	
	作品功能实现 70%、作品演示操作基较熟练、问题回答基本准确；设计论文结构基本合理、内容基本符合要求、论文格式基本符合规范要求、参考文献较新	中 (70-79)	
	作品功能实现 50%、作品演示操作较熟练、部分问题回答准确；设计论文结构较合理、论文格式基本符合规范要求、有参考文献	及格 (60-69)	
	作品功能不能实现、作品演示操作不熟练、问题回答效果差；设计论文结构不合理、论文格式不符合规范要求、无参考文献	不及格 (<60)	
总评			

目 录

第 1 章 绪论	10
1.1 计算机网络的发展历史	10
1.2 计算机网络的背景与意义	10
1.3 计算机网络的作用	11
第 2 章 网络拓扑结构设计	12
2.1 PC -- 主机设计	12
2.2 Router -- 路由器设计	12
2.3 Switch -- 交换机设计	13
2.4 Server -- 服务器设计	13
2.5 VLAN	13
第 3 章 网络拓扑图实现	14
3.1 总网络拓扑图	14
3.2 PC 机配置	14
3.3 Router 配置	15
3.4 VLAN 配置	17
3.5 子网划分	17
3.6 测试连通性	18
第 4 章 聊天室应用程序设计	20
4.1 程序工作流程图	20
4.2 程序源代码	20
第 5 章 聊天室应用程序实现	22
5.1 运行结果	22
5.2 抓包分析	23
第 6 章 总结	26
附录 A	28
参考文献	34

第 1 章 绪论

1.1 计算机网络的发展历史

追溯计算机网络的发展历史，它的演变可概括地分成四个阶段：

(1) 网络雏形阶段。从 20 世纪 50 年代中期开始，以单个计算机为中心的远程联机系统，构成面向终端的计算机网络，称为第一代计算机网络。

(2) 网络初级阶段。从 20 世纪 60 年代中期开始进行主机互联，多个独立的主计算机通过线路互联构成计算机网络，无网络操作系统，只是通信网。60 年代后期，ARPANET 网出现，称为第二代计算机网络。

(3) 20 世纪 70 年代至 80 年代中期，以太网产生，ISO 制定了网络互连标准 OSI，世界上具有统一的网络体系结构，遵循国际标准化协议的计算机网络迅猛发展，这阶段的计算机网络称为第三代计算机网络。

(4) 从 20 世纪 90 年代中期开始，计算机网络向综合化高速化发展，同时出现了多媒体智能化网络，发展到现在，已经是第四代了。局域网技术发展成熟。第四代计算机网络就是以千兆位传输速率为主的多媒体智能化网络。

1.2 计算机网络的背景与意义

进入 21 世纪以来，计算机网络技术完全是以日新月异的速度在发展，很多在 10 年前人们认为遥不可及甚至是不可能的事情，现在一台计算机的计算速度已经达到千万次每秒，一块硬盘的信息储存量可以达到 10 几个图书馆的馆藏量。计算机早已代替了以前的笔和纸，成为学习工作中最重要的工具，越来越多的信息被人们通过网络传播，信息量大、涉及面广、浏览方便等特点，使得网络信息正在悄悄的改变人们的信息获取习惯。如今科研人员不必翻阅大量的纸本资源来查找自己所需要的信息，也不用为了了解国际科研信息四处打听，而打听来的消息可能已经是几个月前的信息了。打开计算机，轻松的敲击键盘，想要的信息会准确无误的出现在眼前。一直以来，纸本图书期刊是科研人员获取情报文章的主要来源，去图书馆查资料也是科研过程中必不可少的一个重要环节。但随着计算机网络的飞速发展，电子版的资料信息以其易

获取、易储存、易利用的独特优势成为了人们的首选，使全球内的信息资源在短时间内得到了迅速传播，也使人们都越来越认识到信息的重要性，对信息的需求也逐渐增加。

1.3 计算机网络的作用

21世纪的一些重要特征就是数字化、网络化和信息化，它是一个以网络为核心的信息时代。要实现信息化就必须依靠完善的网络，因为网络可以非常迅速地传递信息。因此网络现在已经成为信息社会的命脉和发展知识经济的重要基础。网络对社会生活的很多方面以及对社会经济的发展产生了不可估量的影响。

第 2 章 网络拓扑结构设计

2.1 PC -- 主机设计

表 2.1 主机配置

PC	IP Address	Subnet Mask	Default Gateway	DNS Server
PC0	204.29.4.2	255.255.255.0	204.29.4.1	204.29.8.2
PC1	204.29.1.1	255.255.255.0	204.29.2.1	
PC2	204.29.1.2	255.255.255.0	204.29.2.1	
PC3	204.29.1.129	255.255.255.0	204.29.2.1	
PC4	204.29.1.130	255.255.255.0	204.29.2.1	

2.2 Router -- 路由器设计 (型号 2911, 需要扩展 HWIC - 2T 接口)

表 2.2 路由器配置

Router	Interface	IP Address	Subnet Mask	Routing
Router0	Gig 0/0	204.29.4.1	255.255.255.0	OSPF
	Se 0/0/0	204.29.3.2	255.255.255.0	
Router1	Gig 0/0	204.29.2.1	255.255.255.0	OSPF、RIP
	Se 0/0/0	204.29.3.1	255.255.255.0	
	Se 0/0/1	204.29.5.1	255.255.255.0	
Router2	Gig 0/0	204.29.6.1	255.255.255.0	OSPF、Static
	Se 0/0/0	204.29.5.2	255.255.255.0	
	Se 0/0/1	204.29.7.2	255.255.255.0	
Router3	Gig 0/0	204.29.8.1	255.255.255.0	OSPF、Static
	Se 0/0/0	204.29.7.1	255.255.255.0	
	Se 0/0/1	204.29.9.2	255.255.255.0	
Router4	Gig 0/0	204.29.10.1	255.255.255.0	Static
	Se 0/0/0	204.29.9.1	255.255.255.0	

2.3 Switch -- 交换机设计

表 2.3 交换机配置

Switch	Interface	IP Address / VLAN	Routing
Multilayer Switch	Fa 0/1	204.29.2.2	RIP
	Fa 0/2	VLAN50	
	Fa 0/3	VLAN80	
Switch0	Fa 0/1		
	Fa 0/2		
	Fa 0/3		
Switch1	Fa 0/1		
	Fa 0/2		
	Fa 0/3		

2.4 Server -- 服务器设计

表 2.4 服务器配置

Server	IP Address	Subnet Mask	Default Gateway	Services
HTTP	204.29.6.2	255.255.255.0	204.29.6.1	HTTP - ON index.html: 2017070030429
DNS	204.29.8.2	255.255.255.0	204.29.8.1	DNS - ON name: www.zhongzhen.com detail: 204.29.6.2
FTP	204.29.10.2	255.255.255.0	204.29.10.1	FTP - ON username: zhongzhen password: 123456

2.5 VLAN

表 2.5 VLAN 配置

VLAN	Network
VLAN50	204.29.1.0/24
VLAN80	204.29.22.0/24

第3章 网络拓扑图实现

3.1 总网络拓扑图

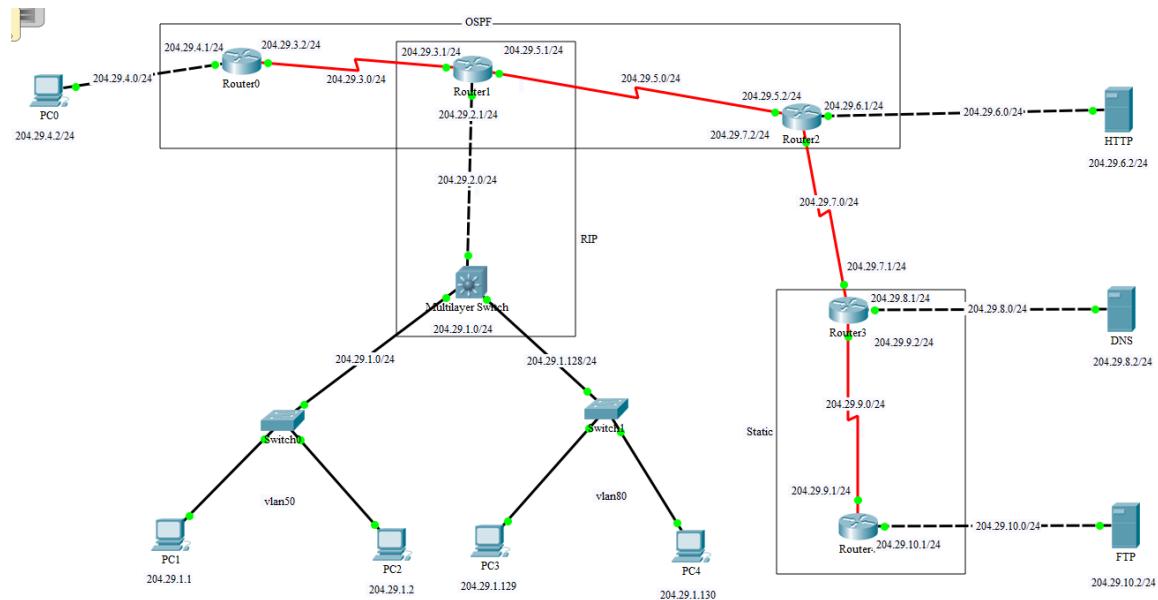


图 3.1 网络拓扑图

3.2 PC 机配置

其余主机配置均相同

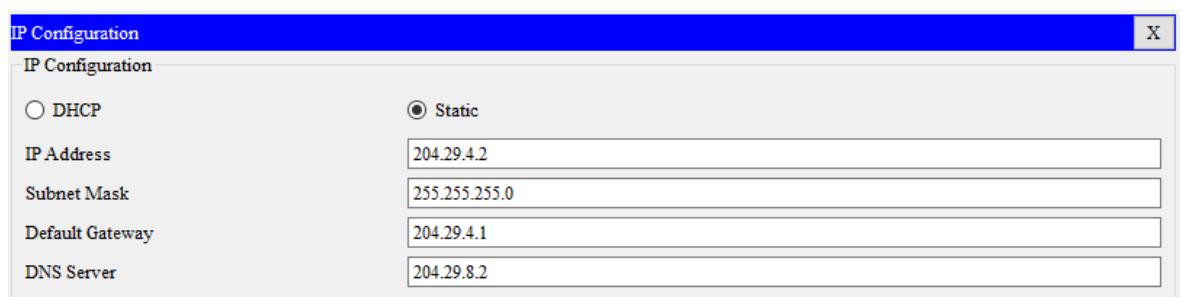


图 3.2 主机配置

3.3 Router 配置

```
Router>enable
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 0.0.0.0 0.0.0.0 204.29.10.2
Router(config)#end
Router#
```

图 3.3 默认路由的配置

```
Router>
Router>enable
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#network 204.29.2.0
Router(config-router)#end
Router#
```

图 3.4 RIP 的配置

```
Router#
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router ospf 29
Router(config-router)#network 204.29.5.0 0.0.0.255 area 29
Router(config-router)#end
Router#
```

图 3.5 OSPF 的配置

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#redistribute ospf 29 metric 3
Router(config-router)#exit
Router(config)#router ospf 29
Router(config-router)#redistribute rip subnets
Router(config-router)#exit
```

图 3.5 路由重分布配置

Routing Table for Router1				
Type	Network	Port	Next Hop IP	Metric
R	204.29.1.0/24	GigabitEthernet0/0	204.29.2.2	120/1
C	204.29.2.0/24	GigabitEthernet0/0	---	0/0
L	204.29.2.1/32	GigabitEthernet0/0	---	0/0
C	204.29.3.0/24	Serial0/0/0	---	0/0
L	204.29.3.1/32	Serial0/0/0	---	0/0
O	204.29.4.0/24	Serial0/0/0	204.29.3.2	110/65
C	204.29.5.0/24	Serial0/0/1	---	0/0
L	204.29.5.1/32	Serial0/0/1	---	0/0
O	204.29.6.0/24	Serial0/0/1	204.29.5.2	110/65
O	204.29.7.0/24	Serial0/0/1	204.29.5.2	110/128
O	204.29.8.0/24	Serial0/0/1	204.29.5.2	110/20
O	204.29.9.0/24	Serial0/0/1	204.29.5.2	110/20
O	204.29.10.0/24	Serial0/0/1	204.29.5.2	110/20

图 3.6 以 Router1 为例，观察相互学习而来的路由表信息

3.4 VLAN 配置

```

Switch>
Switch>enable
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#vlan 50
Switch(config-vlan)#name VLAN50
Switch(config-vlan)#exit
Switch(config)#vlan 80
Switch(config-vlan)#name VLAN80
Switch(config-vlan)#exit
Switch(config)#interface fa0/2
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 50
Switch(config-if)#exit
Switch(config)#interface fa0/3
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 80
Switch(config-if)#exit

```

图 3.7 配置 VLAN 并分配接口

3.5 子网划分

子网 2 有 80 个 ip 地址，80 大于 2 的 6 次方，小于 2 的 7 次方，所以主机位数取 7 位。

网络位数就是 $32 - 7 = 25$ 位。

子网掩码为 11111111.11111111.11111111.10000000 换算成 10 进制的就是 255.255.255.128，这就是第一个子网的子网掩码，网络号为 192.168.0.0/25，网络地址 192.168.0.0，主机地址 192.168.0.1~192.168.0.126，广播地址 192.168.0.127。

子网 1 有 50 个 ip 地址，50 大于 2 的 5 次方，小于 2 的 6 次方，所以主机位数取 6 位。

网络位数就是 $32 - 6 = 26$ 位。

子网掩码为 11111111.11111111.11111111.11000000 换算成 10 进制的就是 255.255.255.192，这就是第二个子网的子网掩码，网络号为 192.168.0.128/26，网络地址 192.168.0.128，主机地址 192.168.0.129~192.168.0.190，广播地址 192.168.0.191。

3.6 测试连通性

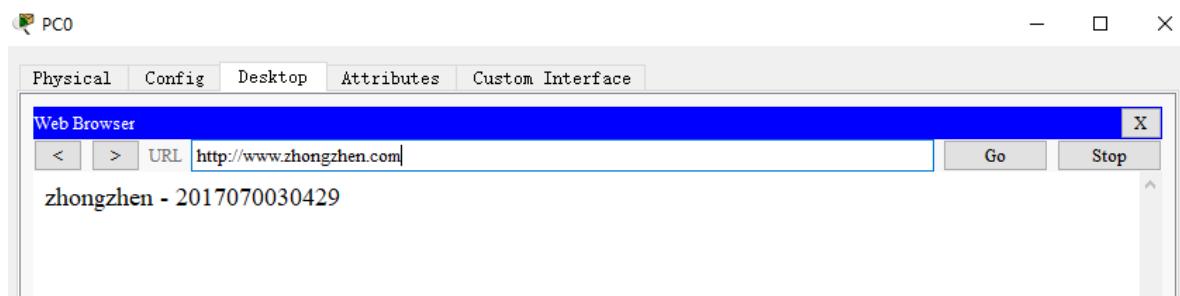


图 3.8 测试与 DNS 服务器的连接（域名解析指向了 HTTP 服务器）

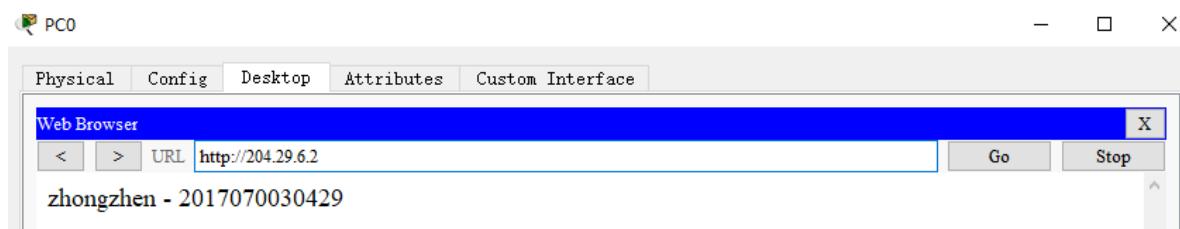


图 3.9 测试与 HTTP 服务器的连接

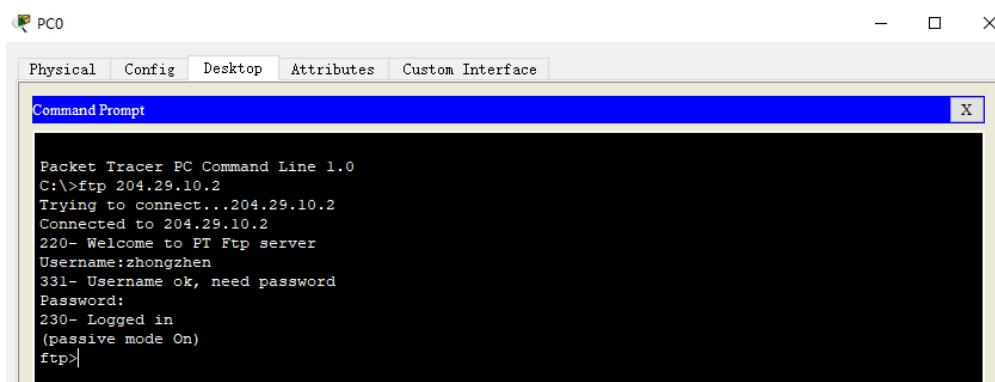


图 3.10 测试与 FTP 服务器的连接

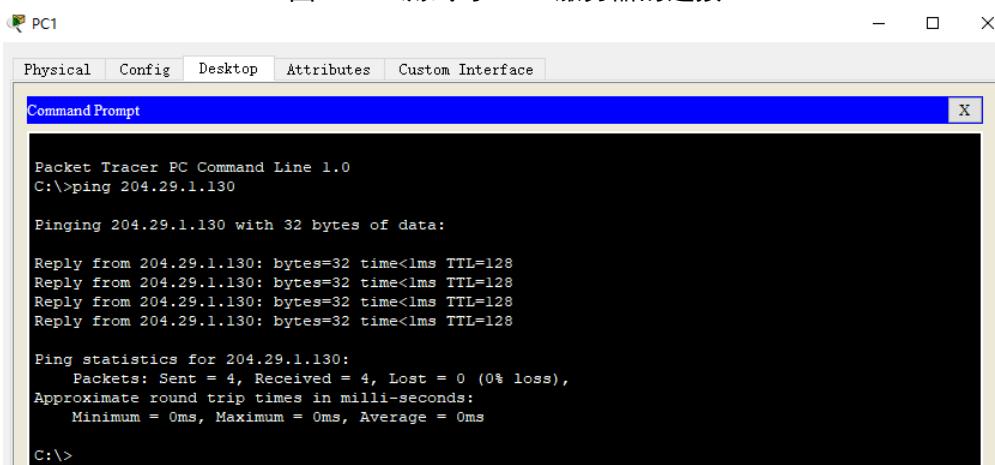


图 3.11 测试主机与主机之间的连接

第 4 章 聊天室应用程序设计

4.1 程序工作流程图

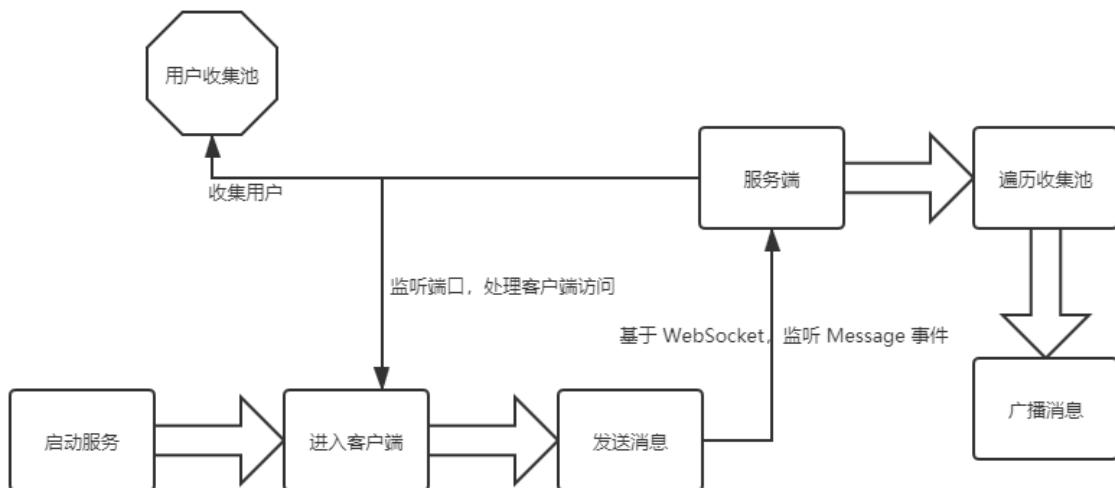


图 4.1 聊天室工作流程图

4.2 程序源代码（客户端代码记录于附录 A）

```
import { chat } from './chat.ts'
import { acceptable, listenAndServe, acceptWebSocket } from './deps.ts'

listenAndServe({ port: 3721 }, async (req) => {
  if (req.method == 'GET' && req.url == '/') {
    req.respond({
      status: 200,
      body: await Deno.open('./index.html'),
      headers: new Headers({ 'content-type': 'text/html' })
    })
  }

  if (req.method == 'GET' && req.url == '/ws') {
    if (acceptable(req)) {
      acceptWebSocket({ conn: req.conn, bufReader: req.r, bufWriter: req.w, headers: req.headers }).then(chat)
    }
  }
})

console.log('run http://localhost:3721')
```

图 4.1 服务端入口文件 mod.ts

```
export { v4 } from 'https://deno.land/std/uuid/mod.ts'
export { listenAndServe } from 'https://deno.land/std/http/server.ts'
export { acceptable, WebSocket, acceptWebSocket, isWebSocketCloseEvent } from 'https://deno.land/std/ws/mod.ts'
```

图 4.2 服务端依赖文件 deps.ts

```
import { v4, WebSocket, isWebSocketCloseEvent } from './deps.ts'

const users = new Map<string, WebSocket>()

function broadcast(message: string, senderID?: string): void {
    if (!message) return

    for (const user of users.values()) {
        user.send(senderID ? `用户 [${senderID}]说: ${message}` : message)
    }
}

export async function chat(ws: WebSocket): Promise<void> {
    const userID = v4.generate()

    users.set(userID, ws)
    broadcast(`用户 [${userID}] 已连接.....`)

    for await (const event of ws) {
        const message = typeof event === 'string' ? event : ''
        broadcast(message, userID)

        if (!message && isWebSocketCloseEvent(event)) {
            users.delete(userID)
            broadcast(`用户 [${userID}] 已断开连接.....`)
            break
        }
    }
}
```

图 4.3 服务端核心逻辑文件 chat.ts

第 5 章 聊天室应用程序实现

5.1 运行结果

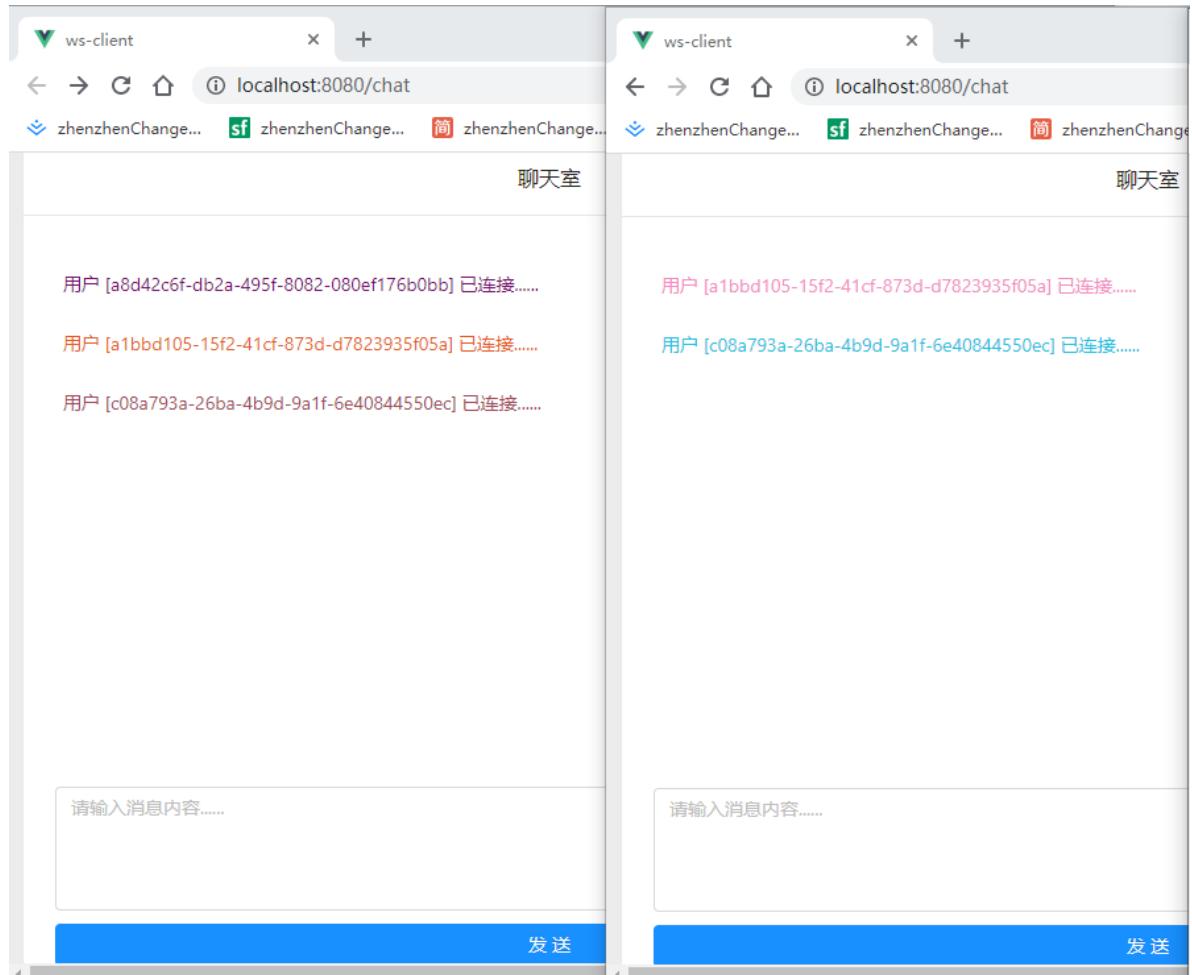


图 5.1 群聊

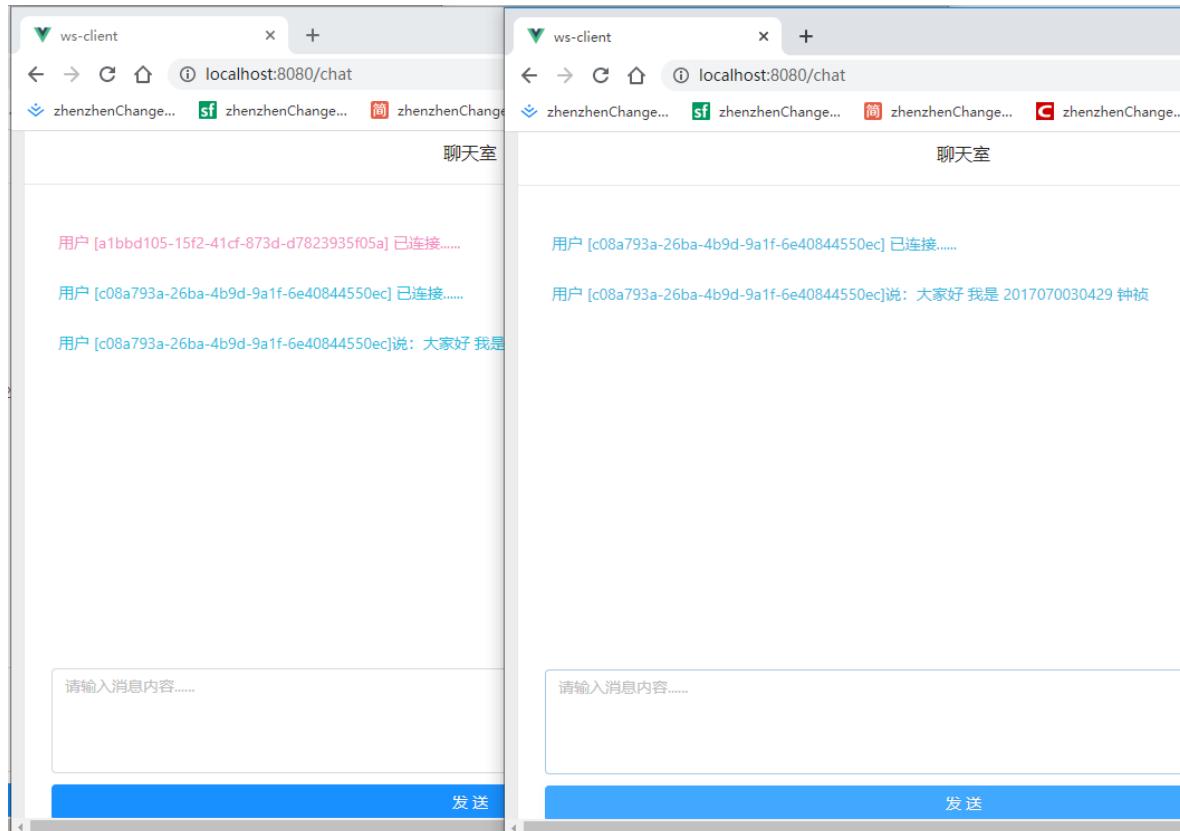


图 5.2 详情

5.2 抓包分析

100 6.416473	127.0.0.1	127.0.0.1	TCP	84 51469 → 3721 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
101 6.416904	127.0.0.1	127.0.0.1	HTTP	581 GET /ws HTTP/1.1 ←
102 6.416923	127.0.0.1	127.0.0.1	TCP	84 3721 → 51469 [ACK] Seq=1 Ack=498 Win=2619648 Len=0
103 6.417999	127.0.0.1	127.0.0.1	HTTP	232 HTTP/1.1 101 Switching Protocols ←
104 6.418025	127.0.0.1	127.0.0.1	TCP	84 51469 → 3721 [ACK] Seq=498 Ack=149 Win=2619648 Len=0
105 6.418219	127.0.0.1	127.0.0.1	WebSocket	147 WebSocket Text [FIN]
106 6.418234	127.0.0.1	127.0.0.1	TCP	84 51469 → 3721 [ACK] Seq=498 Ack=212 Win=2619392 Len=0
129 24.899634	127.0.0.1	127.0.0.1	WebSocket	96 WebSocket Text [FIN] [MASKED]
130 24.899666	127.0.0.1	127.0.0.1	TCP	84 3721 → 51469 [ACK] Seq=212 Ack=510 Win=2619648 Len=0
131 24.899997	127.0.0.1	127.0.0.1	WebSocket	143 WebSocket Text [FIN]
132 24.900024	127.0.0.1	127.0.0.1	TCP	84 51469 → 3721 [ACK] Seq=510 Ack=271 Win=2619392 Len=0
141 38.901110	127.0.0.1	127.0.0.1	WebSocket	110 WebSocket Text [FIN] [MASKED]
142 38.901138	127.0.0.1	127.0.0.1	TCP	84 3721 → 51469 [ACK] Seq=271 Ack=536 Win=2619648 Len=0
143 38.901467	127.0.0.1	127.0.0.1	WebSocket	157 WebSocket Text [FIN]
144 38.901485	127.0.0.1	127.0.0.1	TCP	84 51469 → 3721 [ACK] Seq=536 Ack=344 Win=2619392 Len=0

图 5.3 TCP 连接建立成功

分析：通过抓包结果可知，客户端与服务端建立 TCP 连接后，访问了接口地址 /ws，随后服务端发送初始化数据，用户进入聊天室即可开始聊天。

101 6.416904	127.0.0.1	127.0.0.1	HTTP	581 GET /ws HTTP/1.1
102 6.416923	127.0.0.1	127.0.0.1	TCP	84 3721 → 51469 [ACK] Seq=1 Ack=498 Win=2619648 Len=0
103 6.417999	127.0.0.1	127.0.0.1	HTTP	232 HTTP/1.1 101 Switching Protocols
104 6.418025	127.0.0.1	127.0.0.1	TCP	84 51469 → 3721 [ACK] Seq=498 Ack=149 Win=2619648 Len=0
105 6.418219	127.0.0.1	127.0.0.1	WebSocket	147 WebSocket Text [FIN]
106 6.418234	127.0.0.1	127.0.0.1	TCP	84 51469 → 3721 [ACK] Seq=498 Ack=212 Win=2619392 Len=0
129 24.899634	127.0.0.1	127.0.0.1	WebSocket	96 WebSocket Text [FIN] [MASKED]
130 24.899666	127.0.0.1	127.0.0.1	TCP	84 3721 → 51469 [ACK] Seq=212 Ack=510 Win=2619648 Len=0
131 24.899997	127.0.0.1	127.0.0.1	WebSocket	143 WebSocket Text [FIN]
132 24.900024	127.0.0.1	127.0.0.1	TCP	84 51469 → 3721 [ACK] Seq=510 Ack=271 Win=2619392 Len=0
141 38.901110	127.0.0.1	127.0.0.1	WebSocket	110 WebSocket Text [FIN] [MASKED]
142 38.901138	127.0.0.1	127.0.0.1	TCP	84 3721 → 51469 [ACK] Seq=271 Ack=536 Win=2619648 Len=0
143 38.901467	127.0.0.1	127.0.0.1	WebSocket	157 WebSocket Text [FIN]
144 38.901485	127.0.0.1	127.0.0.1	TCP	84 51469 → 3721 [ACK] Seq=536 Ack=344 Win=2619392 Len=0

```

> Frame 105: 147 bytes on wire (1176 bits), 147 bytes captured (1176 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 3721, Dst Port: 51469, Seq: 149, Ack: 498, Len: 63
> WebSocket
> Line-based text data (1 lines)
    用戶 [8b54166e-430b-4d5c-bd17-cd16408091de] 已连接.....
```

图 5.4 用户进入聊天室

100 6.416473	127.0.0.1	127.0.0.1	TCP	84 51469 → 3721 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
101 6.416904	127.0.0.1	127.0.0.1	HTTP	581 GET /ws HTTP/1.1
102 6.416923	127.0.0.1	127.0.0.1	TCP	84 3721 → 51469 [ACK] Seq=1 Ack=498 Win=2619648 Len=0
103 6.417999	127.0.0.1	127.0.0.1	HTTP	232 HTTP/1.1 101 Switching Protocols
104 6.418025	127.0.0.1	127.0.0.1	TCP	84 51469 → 3721 [ACK] Seq=498 Ack=149 Win=2619648 Len=0
105 6.418219	127.0.0.1	127.0.0.1	WebSocket	147 WebSocket Text [FIN]
106 6.418234	127.0.0.1	127.0.0.1	TCP	84 51469 → 3721 [ACK] Seq=498 Ack=212 Win=2619392 Len=0
129 24.899634	127.0.0.1	127.0.0.1	WebSocket	96 WebSocket Text [FIN] [MASKED]
130 24.899666	127.0.0.1	127.0.0.1	TCP	84 3721 → 51469 [ACK] Seq=212 Ack=510 Win=2619648 Len=0
131 24.899997	127.0.0.1	127.0.0.1	WebSocket	143 WebSocket Text [FIN]
132 24.900024	127.0.0.1	127.0.0.1	TCP	84 51469 → 3721 [ACK] Seq=510 Ack=271 Win=2619392 Len=0
141 38.901110	127.0.0.1	127.0.0.1	WebSocket	110 WebSocket Text [FIN] [MASKED]
142 38.901138	127.0.0.1	127.0.0.1	TCP	84 3721 → 51469 [ACK] Seq=271 Ack=536 Win=2619648 Len=0
143 38.901467	127.0.0.1	127.0.0.1	WebSocket	157 WebSocket Text [FIN]
144 38.901485	127.0.0.1	127.0.0.1	TCP	84 51469 → 3721 [ACK] Seq=536 Ack=344 Win=2619392 Len=0

```

> Frame 141: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 51469, Dst Port: 3721, Seq: 510, Ack: 271, Len: 26
> WebSocket
> Line-based text data (1 lines)
    钟桢 2017070030429
```

图 5.5 用户发送消息

WebSocket 是一种在单个 TCP 连接上进行全双工通信的协议。

它使得客户端和服务器之间的数据交换变得更加简单，允许服务端主动向客户端推送数据。在 WebSocket API 中，浏览器和服务器只需要完成一次握手，两者之间就直接可以创建持久性的连接，并进行双向数据传输。

在本程序中，服务端监听端口号，将访问该端口的用户（客户端）收集起来存放，同时 WebSocket 监听消息事件，当有消息传输时，广播给收集起来的每一个用户，达到群聊效果。

第 6 章 总结

经过短短一星期的计算机网络课程设计，我深深体会到网络的神奇和奥妙，也意识到自己在这方面还有许多不足。在以后的学习中，要多多实践，把理论的知识运用到实践中。

本次实践，做的是一个小型的局域网，把整个网络做通，并在各个链路上抓包，测试连接（ping），理解抓包结果。

实践过程中，总是出现很多的失误，根本原因在于对各种协议没有理解好。但是，学习最重要的是要有耐心，不要怕出现失误，当出现失误时，说明那个知识点没有掌握好，并且要反思一下为什么出现错误。要有自己的逻辑思维，脑袋里要有网络拓扑图的清晰结构，在进行配置时，要清楚哪个端口需要配置的内容是什么，不可忙乱连接，或者照着老师的，做完了都不理解是什么。实践中遇到了困难，可以问老师或者同学，相互学习，取长补短，对知识点的掌握是很有益处的。

“纸上得来终觉浅，绝知此事要躬行！”在短暂的实践过程中，让我深深的感觉到自己在实际运用中的专业知识的匮乏，让我真正领悟到“学无止境”的含义。这次课设让我学到的东西太多，使我受益非浅，它让我知道了工作上的辛苦，让我知道工作并不像在学校里学习一样轻松。不过，虽然辛苦了点，但能让我学到更多的东西，我心里还是高兴的。人非生而知之，要学得知识，一靠学习，二靠实践。没有实践，学习就是无源之水，无本之木。

附录 A

APP.vue

```
<template>
  <router-view></router-view>
</template>
```

Home.vue

```
<template>
  <div style="background:#ECECEC; padding:30px; height: 100vh; text-align: center;">
    <a-card class="ws-card" title="基于 TCP 的 WebSocket 聊天室" style="width: 400px">
      <a-form :form="form" :label-col="{ span: 5 }" :wrapper-col="{ span: 15 }" @submit="handleSubmit">
        <a-form-item label="昵称">
          <a-input v-decorator="['note', { rules: [{ required: true, message: '请输入昵称~' }] }]" />
        </a-form-item>
        <a-form-item :wrapper-col="{ span: 24 }">
          <a-button type="primary" shape="round" html-type="submit">进入</a-button>
        </a-form-item>
      </a-form>
    </a-card>
  </div>
</template>
```

```
<script>
export default {
  name: 'Home',
  data() {
    return {
      form: this.$form.createForm(this, { name: 'coordinated' })
    }
  },
  methods: {
    handleSubmit(e) {
      e.preventDefault()
      this.form.validateFields((err, values) => {
        if (!err) {
```

```
        console.log('Received values of form: ', values)
        this.$router.push('/chat')
    }
})
}
}
}
}
</script>
```

```
<style lang="scss" scoped>
.ws-card {
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
}
</style>
```

Chat. vue

```
<template>
<div style="background:#ECECEC; padding:30px; height: 100vh;">
    <a-card class="ws-card" title="聊天室" style="width: 800px">
        <a-list id="scroll" :split="false" class="ws-list" item-layout="horizontal" :data-source="data">
            <a-list-item :style="{ color: getColor(item.id) }" slot="renderItem" slot-scope="item">
                {{ item.message }}
            </a-list-item>
        </a-list>
        <a-textarea class="mt-10" placeholder="请输入消息内容....." :rows="4" v-model="message" />
        <a-button class="mt-10" type="primary" block @click="send">发送</a-button>
    </a-card>
</div>
</template>
```

```
<script>
export default {
    name: 'chat',
    data() {
        return {
            ws: '',
            data: [],
        }
    }
}
```

```

    message: '',
    mapsColor: new Map()
  }
},
created() {
  this.ws = new WebSocket(`ws://localhost:3721/ws`)
  this.ws.addEventListener('message', this.onReceiveMessage)
},
mounted() {
  const listScroll = document.getElementById('scroll')
  listScroll.scrollTop = listScroll.scrollHeight
},
methods: {
  send() {
    this.ws.send(this.message)
    this.message = ''
  },
  onReceiveMessage({ data }) {
    const match = data.match(/\[(.+?)\]/)
    const id = match[1]

    if (!this.mapsColor.has(id)) this.mapsColor.set(id, this.getRandomColor())

    this.data.push({ id, message: data })

    // 数据更新后 DOM 还未发生改变，需在下一轮事件循环再操作 DOM
    this.$nextTick(() => {
      const listScroll = document.getElementById('scroll')
      listScroll.scrollTop = listScroll.scrollTop + listScroll.scrollHeight
    })
  },
  getColor(id) {
    return this.mapsColor.get(id)
  },
  getRandomColor() {
    const rgb = []
    for (let i = 0; i < 3; ++i) {
      let color = Math.floor(Math.random() * 256).toString(16)
      color = color.length == 1 ? '0' + color : color
      rgb.push(color)
    }
  }
}

```

```
        return '#' + rgb.join('')
    }
}
}

</script>
<style lang="scss" scoped>
.ws-loading {
    width: 100%;
    bottom: 40px;
    position: absolute;
    text-align: center;
}

.ws-card {
    top: 50%;
    left: 50%;
    float: left;
    transform: translate(-50%, -50%);
    ::v-deep .ant-card-head {
        text-align: center;
    }
}

.ws-list {
    height: 400px;
    padding: 6px;
    overflow-y: scroll;
}

.mt-10 {
    margin-top: 10px;
}
</style>
```

router/index.js

```
import Vue from 'vue'
import VueRouter from 'vue-router'
import Home from '../views/Home.vue'
```

```
Vue.use(VueRouter)
```

```
const routes = [
  {
    path: '/',
    redirect: '/chat'
  },
  {
    path: '/home',
    name: 'Home',
    component: Home
  },
  {
    path: '/chat',
    name: 'Chat',
    component: () => import('../views/Chat.vue')
  }
]
const router = new VueRouter({
  mode: 'history',
  base: process.env.BASE_URL,
  routes
})

export default router
```


参考文献

- [1] Andrew.S.计算机网络.清华大学出版社，2014
- [2] 谢希仁.计算机网络（第七版）.电子工业出版社，2013
- [3] James.计算机网络：自顶向下方法（第7版）.清华大学出版社，2018
- [4] 李志远.计算机网络综合实验教程—协议分析与应用.机械工业出版社，2019
- [5] 竹下隆史.图解 TCP/IP.人民邮电出版社，2013

学生签名：
年 月 日