Red Hat Enterprise Linux 7

网络指南

在 RHEL 7 中配置和管理网络、网络接口和网络服务

Last Updated: 2023-09-21

Red Hat Enterprise Linux 7 网络指南

在 RHEL 7 中配置和管理网络、网络接口和网络服务

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律通告

Copyright © 2023 | You need to change the HOLDER entity in the en-US/Networking_Guide.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

摘要

红帽企业 Linux 7 网络指南记录了有关红帽企业 Linux 中网络接口、网络和网络服务的配置和管理相关信息。它面向对 Linux 和网络具有基本了解的系统管理员。

目录

部分 I. 开始前 11

第 1 章 网络主题概述 12
1.1. IP 与非 IP 网络比较 12
网络通信的类型 12
1.2. 静态与动态 IP 地址比较 12
1.3. 配置 DHCP 客户端行为 13
请求 IP 地址 13
请求租用续订 13
1.3.1. 使 DHCPv4 持久 13
1.4. 设置 WIRELESS REGULATORY 域 14
1.5. 配置 NETCONSOLE 14
1.6. 使用带有 SYSCTL 的网络内核可调项 15
1.7. 使用 NCAT 工具管理数据 16
安装 ncat 16
ncat 用例的简短选择 16

部分 II. 管理 IP 网络 19

第 2 章 NETWORKMANAGER 入门 20
2.1. NETWORKMANAGER 概述 20
2.1.1. 使用 NetworkManager 的好处 20
2.2. 安装 NETWORKMANAGER 20
2.3. 检查 NETWORKMANAGER 的状态 20
2.4. 启动 NETWORKMANAGER 20
2.5. NETWORKMANAGER 工具 21
2.6. 将 NETWORKMANAGER 与网络脚本搭配使用 21
运行网络脚本 21
在网络脚本中使用自定义命令 22
运行 Dispatcher 脚本 23
2.7. 使用 NETWORKMANAGER 和 SYSCONFIG 文件 23
2.8. 其它资源 26

第 3 章 配置 IP 网络 27
3.1. 选择网络配置方法 27
3.2. 使用 NMTUI 配置 IP 网络 28
3.3. 使用 NMCLI 配置 IP 网络 33
3.3.1. nmcli 示例的简短选择 35
3.3.2. 使用 nmcli 启动和停止网络接口 37
3.3.3. 了解 nmcli 选项 38
3.3.4. 使用 nmcli Interactive Connection Editor 39
3.3.5. 使用 nmcli 创建和修改连接配置集 40
3.3.6. 使用 nmcli 连接到网络 43
3.3.7. 使用 nmcli 添加和配置动态以太网连接 44
添加动态以太网连接 44
配置动态以太网连接 45
3.3.8. 使用 nmcli 添加和配置静态以太网连接 46
添加静态以太网连接 46
3.3.9. 使用 nmcli 将配置集锁定到特定设备 49
3.3.10. 使用 nmcli 添加 Wi-Fi 连接 49
使用 nmcli 更改特定属性 50
3.3.11. 将 NetworkManager 配置为 Ignore Certain 设备 50

3.3.11.1. 将设备永久配置为 NetworkManager 中的非受管设备 51
流程 51
验证步骤 51
其它资源 52
3.3.11.2. 将设备临时配置为 NetworkManager 中的非受管设备 52
流程 52
验证步骤 52
其它资源 53
3.4. 使用 GNOME GUI 配置 IP 网络 53
3.4.1. 使用 control-center GUI 连接到网络 53
3.4.2. 使用 GUI 配置新连接并编辑现有连接 55
3.4.2.1. 使用 control-center 配置新的和编辑现有连接 55
使用 control-center 配置新连接 55
使用 control-center 编辑现有连接 57
3.4.2.2. 使用 nm-connection-editor 配置新的和编辑现有连接 58
使用 nm-connection-editor 配置新连接 58
使用 nm-connection-editor 编辑现有连接 60
3.4.3. 使用 nm-connection-editor 的通用配置选项 60
3.4.4. 使用 GUI 自动连接到网络 62
3.4.4.1. 使用 control-center 自动连接到网络 62
3.4.4.2. 使用 nm-connection-editor 自动连接到网络 63
3.4.5. 使用 GUI 管理系统范围以及专用连接配置集 63
3.4.5.1. 使用 nm-connection-editor 管理连接配置集的权限 63
3.4.5.2. 使用 control-center 管理连接配置集的权限 63
3.4.6. 使用 GUI 配置连线（以太网）连接 64
3.4.6.1. 使用 control-center 配置 Wired Connection 65
基本配置选项 65
生成更多连线配置 66
保存您的新（或修改的）连线连接 66
创建新 Wired 连接 66
3.4.6.2. 使用 nm-connection-editor 配置 Wired Connection 67
3.4.7. 使用 GUI 配置 Wi-Fi 连接 68
快速连接到可用接入点 68
连接到 Hidden Wi-Fi 网络 69
配置新的 Wi-Fi 连接 70
编辑现有 Wi-Fi 连接 71
Wi-Fi 连接的基本配置选项 71
进行进一步的 Wi-Fi 配置 73
保存您的新（或修改的）连接 73
3.4.8. 使用 GUI 配置 VPN 连接 74
3.4.8.1. 使用 control-center 建立 VPN 连接 74
添加新 IPsec VPN 连接 75
编辑现有 VPN 连接 78
保存您的新（或修改）连接并创建进一步配置 78
3.4.8.2. 使用 nm-connection-editor 配置 VPN 连接 79
3.4.9. 使用 GUI 配置移动带宽连接 79
3.4.9.1. 使用 nm-connection-editor 配置移动带宽连接 80
添加新移动带宽连接 80
编辑现有移动带宽连接 81
配置 Mobile Broadband 选项卡 81
保存您的新（或修改）连接并创建进一步配置 83
3.4.10. 使用 GUI 配置 DSL 连接 83
3.4.10.1. 使用 nm-connection-editor 配置 DSL 连接 83

添加新 DSL 连接 83
编辑现有 DSL 连接 84
配置 DSL 选项卡 84
保存您的新（或修改）连接并创建进一步配置 84
3.5. 使用 IFCFG 文件配置 IP 网络 85
使用 ifcfg 文件配置带有静态网络设置的接口 85
使用 ifcfg 文件配置带有动态网络设置的接口 86
3.5.1. 使用 ifcfg 文件管理系统范围以及专用连接配置集 88
3.6. 使用 IP 命令配置 IP 网络 88
使用 ip 命令分配静态地址 89
使用 ip 命令配置多个地址 90
3.7. 使用内核命令行配置 IP 网络 90
3.8. 使用 IGMP 启用 IP 多播 91
3.9. 其它资源 93
安装的文档 93
在线文档 93

第 4 章 配置静态路由和默认网关 95
4.1. 了解路由和网关简介 95
4.2. 使用 NMCLI 配置静态路由 95
4.3. 使用 GUI 配置静态路由 96
4.4. 使用 IP 命令配置静态路由 97
4.5. 在 IFCFG 文件中配置静态路由 98
使用 IP 命令参数格式的静态路由 98
使用网络/网络掩码指令格式的静态路由 100
4.5.1. 了解策略路由 101
4.6. 配置默认网关 102

第 5 章 配置网络连接设置 103
5.1. 配置 802.3 链接设置 103
忽略链路协商 103
强制自动协商激活 104
手动设置链接速度和双工 104
使用 nmcli 工具配置 802.3 链接设置 105
使用 nm-connection-editor 配置 802.3 链路设置 105
5.2. 配置 802.1X 安全性 107
5.2.1. 使用 nmcli 为 Wi-Fi 配置 802.1X 安全性 107
5.2.2. 使用 nmcli 为 Wired 配置 802.1X 安全性 108
5.2.3. 使用 GUI 为 Wi-Fi 配置 802.1X 安全性 108
5.2.4. 使用 nm-connection-editor 为 Wired 配置 802.1X 安全性 110
配置 TLS 设置 110
配置 FAST 设置 111
配置 Tunneled TLS 设置 112
配置受保护的 EAP(PEAP)设置 113
5.3. 使用带有 WPA_SUPPLICANT 和 NETWORKMANAGER的 MACSEC 114
5.4. 配置 IPV4 设置 115
使用 control-center 配置 IPv4 设置 116
使用 nm-connection-editor 设置 IPV4 的方法 116
5.5. 配置 IPV6 设置 120
5.6. 配置 PPP（点对点）设置 120

第 6 章 配置主机名 122
6.1. 了解主机名 122
6.1.1. 推荐的命名实践 122

6.2. 使用文本用户界面配置主机名，NMTUI 122
6.3. 使用 HOSTNAMECTL 配置主机名 123
6.3.1. 查看所有主机名 123
6.3.2. 设置所有主机名 124
6.3.3. 设置部分主机名 124
6.3.4. 清除具体主机名 124
6.3.5. 远程更改主机名 125
6.4. 使用 NMCLI 配置主机名 125
6.5. 其它资源 125

第 7 章 配置网络绑定 127
7.1. 了解控制器和端口接口的默认行为 127
7.2. 使用文本用户界面 NMTUI 配置绑定 128
7.3. 使用 NETWORKMANAGER 命令行工具 NMCLI 进行网络绑定 133
7.4. 使用命令行界面(CLI) 135
7.4.1. 检查 Bonding 内核模块是否已安装 135
7.4.2. 创建频道绑定接口 135
7.4.3. 创建端口接口 137
7.4.4. 激活频道绑定 138
7.4.5. 创建多个绑定 139
7.5. 验证冗余的网络配置绑定 140
7.6. 切换中绑定模式和所需设置概述 141
7.7. 使用频道绑定 142
7.7.1. 绑定模块指令 142
7.8. 使用 GUI 创建绑定连接 151
7.8.1. 建立绑定连接 151
保存您的新（或修改）连接并创建进一步配置 154
7.8.1.1. 配置 Bond 选项卡 155
7.9. 其它资源 157
安装的文档 157
在线文档 157

第 8 章 配置网络合作 159
8.1. 了解网络合作 159
8.2. 了解控制器和端口接口的默认行为 160
8.3. 网络团队与绑定的比较 161
8.4. 了解网络合作后台程序和"运行者" 162
8.5. 安装网络合作守护进程 163
8.6. 将 BOND 转换为团队 164
8.7. 选择用作网络组端口的接口 165
8.8. 选择网络团队配置方法 165
8.9. 使用文本用户界面 NMTUI 配置网络团队 166
8.10. 使用命令行配置网络团队 171
8.10.1. 使用 nmcli 配置网络合作 171
8.10.2. 使用 teamd 创建网络团队 175
8.10.3. 使用 ifcfg 文件创建网络团队 179
8.10.4. 使用 iputils 向网络团队添加端口 180
8.10.5. 使用 teamnl 列出团队的端口 180
8.10.6. 使用 teamnl 配置团队选项 180
8.10.7. 使用 iputils 为网络团队添加地址 181
8.10.8. 使用 iputils 打开网络团队的接口 181
8.10.9. 使用 teamnl 查看团队的活跃端口选项 181
8.10.10. 使用 teamnl 设置团队的活跃端口选项 181

8.11. 使用 TEAMDCTL 控制 TEAMD 182
8.11.1. 添加端口到网络团队 182
8.11.2. 从网络团队中删除端口 183
8.11.3. 将配置应用到网络组中的端口 183
8.11.4. 查看网络团队中的端口配置 183
8.12. 验证冗余的网络配置合作 184
8.13. 配置 TEAMD 运行程序 185
8.13.1. 配置广播运行程序 186
8.13.2. 配置随机运行程序 186
8.13.3. 配置 round-robin Runner 186
8.13.4. 配置 activebackup Runner 187
8.13.5. 配置 loadbalance Runner 189
8.13.6. 配置 LACP(802.3ad)运行程序 189
8.13.7. 配置链路状态的监控 190
8.13.7.1. 配置 Ethtool 进行链接状态监控 190
8.13.7.2. 为链路状态监控配置 ARP Ping 191
8.13.7.3. 为 Link-state Monitoring 配置 IPv6 NA/NS 192
8.13.8. 配置端口选择覆盖 193
8.13.9. 配置基于 BPF 的 Tx Port Selectors 194
8.14. 使用 GUI 创建网络团队 195
8.14.1. 建立团队连接 195
保存您的新（或修改）连接并创建进一步配置 199
8.14.1.1. 配置 Team 选项卡 199
8.15. 其它资源 199
安装的文档 199
在线文档 200

第 9 章 配置网络桥接 201
9.1. 使用文本用户界面 NMTUI 配置桥接 201
9.2. 使用 NETWORKMANAGER 命令行工具 NMCLI 205
9.3. 使用命令行界面(CLI) 208
9.3.1. 检查 Bridging 内核模块是否已安装 208
9.3.2. 创建网桥 208
9.3.3. 使用 Bond 的网桥 211
9.4. 使用 GUI 配置网络桥接 212
9.4.1. 使用 GUI 建立网桥连接 213
配置连接名称、自动连接行为和可用性设置 215
9.4.1.1. 配置 Bridge 选项卡 215
保存您的新（或修改）连接并创建进一步配置 218
9.5. 使用 IPROUTE 配置以太网桥接配置 219
9.6. 其它资源 220

第 10 章 配置 802.1Q VLAN 标记 221
10.1. 选择 VLAN 接口配置方法 222
10.2. 使用文本用户界面 NMTUI 配置 802.1Q VLAN 标记 222
10.3. 使用命令行工具 NMCLI 配置 802.1Q VLAN 标记 224
为 VLAN 接口分配地址 225
10.4. 使用命令行配置 802.1Q VLAN 标记 228
10.4.1. 使用 ifcfg 文件设置 802.1Q VLAN 标记 228
10.4.2. 使用 ip 命令配置 802.1Q VLAN 标记 229
10.5. 使用 GUI 配置 802.1Q VLAN 标记 230
10.5.1. 建立 VLAN 连接 230
保存您的新（或修改）连接并创建进一步配置 233

10.5.1.1. 配置 VLAN 选项卡 234
10.6. 使用 IP 命令绑定和桥接上的 VLAN 234
10.7. BOND 和 BRIDGE 上使用 NETWORKMANAGER 命令行工具 NMCLI 的 VLAN 235
10.8. 配置 VLAN 切换端口模式 236
10.9. 其它资源 237

第 11 章 一致的网络设备命名 238
11.1. 命名方案层次结构 238
11.2. 了解设备重命名过程 239
11.3. 了解可预测网络接口设备名称 240
11.4. SYSTEM Z 上 LINUX 可用的网络设备命名方案 241
11.5. VLAN 接口的命名方案 242
11.6. 使用 BIOSDEVNAME 的一致网络设备命名 242
11.6.1. 系统要求 243
11.6.2. 启用和禁用功能 243
11.7. 管理员备注 243
11.8. 控制网络设备名称的选择 244
11.9. 禁用一致的网络设备命名 244
11.10. 网络设备命名故障排除 246
11.11. 其它资源 249
安装的文档 249
在线文档 249

第 12 章 配置基于策略的路由来定义备用路由 251
12.1. 将流量从特定子网路由到不同的默认网关 251
先决条件 251
流程 252
验证步骤 254
故障排除步骤 255
其它资源 256

部分 III. INFINIBAND 和 RDMA 网络 257

第 13 章 配置 INFINIBAND 和 RDMA 网络 258
13.1. 了解 INFINIBAND 和 RDMA 技术 258
先决条件 259
13.2. 使用 ROCE 传输数据 259
先决条件 260
13.3. 配置 SOFT-ROCE 262
先决条件 263
删除 RXE 设备 264
验证 RXE 设备的连接性 265
13.4. INFINIBAND 和 RDMA 相关软件包 265
13.5. 配置基本 RDMA 子系统 267
13.5.1. 配置 rdma.conf 文件 267
13.5.2. 70-persistent-ipoib.rules 的使用 268
13.5.3. 为用户缓解 memlock 限制 268
13.5.4. 为以太网操作配置 Mellanox 卡 269
13.5.5. 连接到远程 Linux SRP 目标 269
连接到远程 Linux SRP 目标：高级别概述 270
13.6. 配置子网管理器 275
13.6.1. 确定需要 275
13.6.2. 配置 opensm 主配置文件 275
13.6.3. 配置 opensm 启动选项 275

13.6.4. 创建 P_Key 定义 275
13.6.5. 启用 opensm 277
13.7. 测试早期 INFINIBAND RDMA 操作 277
13.8. 配置 IPOIB 280
13.8.1. 了解 IPoIB 的角色 280
13.8.2. 了解 IPoIB 通信模式 280
13.8.3. 了解 IPoIB 硬件地址 281
13.8.4. 了解 InfiniBand P_Key 子网 281
13.8.5. 使用文本用户界面 nmtui 配置 InfiniBand 282
13.8.6. 使用命令行工具 nmcli 配置 IPoIB 284
13.8.7. 使用命令行配置 IPoIB 286
13.8.8. 配置 IPoIB 后测试 RDMA 网络 287
13.8.9. 使用 GUI 配置 IPoIB 288
保存您的新（或修改）连接并创建进一步配置 290
13.8.9.1. 配置 InfiniBand 选项卡 290
13.8.10. 其它资源 291
安装的文档 291
在线文档 291

部分 IV. 服务器 292

第 14 章 DHCP 服务器 293
14.1. 为什么使用 DHCP？ 293
14.2. 配置 DHCP 服务器 293
14.2.1. 配置文件 294
14.2.2. 租期数据库 298
14.2.3. 启动和停止服务器 299
14.3. DHCP 转发代理 300
14.3.1. 将 dhcrelay 配置为 DHCPv4 和 BOOTP 转发代理 301
14.3.2. 将 dhcrelay 配置为 DHCPv6 中继代理 302
14.4. 配置多HOMED DHCP 服务器 302
14.4.1. 主机配置 304
14.5. 用于 IPV6 的 DHCP(DHCPV6) 306
14.6. 为 IPV6 路由器配置 RADVD 守护进程 307
14.7. DHCPV6 和 RADVD 的比较 309
Manually 309
使用 radvd 守护进程 309
使用 DHCPv6 服务器 309
14.8. 其它资源 310

第 15 章 DNS 服务器 311
15.1. DNS 简介 311
15.1.1. 名称服务器区域 311
15.1.2. 名称服务器类型 312
15.1.3. BIND 作为名称服务器 312
15.2. BIND 312
15.2.1. 空区域 312
15.2.2. 配置指定服务 313
15.2.2.1. 在 chroot 环境中安装 BIND 315
15.2.2.2. 常见语句类型 316
15.2.2.3. 其他语句类型 325
15.2.2.4. 注释标签 327
15.2.3. 编辑区域文件 328
15.2.3.1. 常用指令 328

15.2.3.2. 通用资源记录 329
15.2.3.3. 注释标签 334
15.2.3.4. 用法示例 334
15.2.3.4.1. 简单区域文件 334
15.2.3.4.2. 反向名称解析区域文件 335
15.2.4. 使用 rndc 实用程序 336
15.2.4.1. 配置实用程序 336
15.2.4.2. 检查服务状态 337
15.2.4.3. 重新加载配置和区域 337
15.2.4.4. 更新区域密钥 338
15.2.4.5. 启用 DNSSEC 验证 339
15.2.4.6. 启用查询日志记录 339
15.2.5. 使用 dig 实用程序 339
15.2.5.1. 查找名称服务器 340
15.2.5.2. 查找 IP 地址 340
15.2.5.3. 查找主机名 341
15.2.6. BIND 的高级功能 342
15.2.6.1. 多个视图 342
15.2.6.2. 增量区域传输(IXFR) 342
15.2.6.3. 事务 SIGnatures(TSIG) 343
15.2.6.4. DNS 安全扩展(DNSSEC) 343
15.2.6.5. 互联网协议版本 6(IPv6) 343
15.2.7. 通用 Mistakes to Avoid 343
15.2.8. 其它资源 344
15.2.8.1. 安装的文档 344
15.2.8.2. 在线资源 345

第 16 章 配置 SQUID 缓存代理服务器 **347**
16.1. 在不使用身份验证的情况下将 SQUID 设置为缓存代理 347
先决条件 347
流程 347
验证步骤 350
16.2. 使用 LDAP 身份验证将 SQUID 设置为缓存代理 350
先决条件 350
流程 350
验证步骤 354
故障排除步骤 354
16.3. 使用 KERBEROS 身份验证将 SQUID 设置为缓存代理 355
先决条件 355
流程 355
验证步骤 359
故障排除步骤 359
16.4. 在 SQUID 中配置域黑名单 360
先决条件 360
流程 360
16.5. 将 SQUID 服务配置为在特定端口或 IP 地址上侦听 361
先决条件 361
流程 361
16.6. 其它资源 362

附录 A. 红帽客户门户网站 LABS 与网络相关 **363**
网桥配置 363
网络绑定帮助程序 363

数据包捕获语法生成器 363

附录 B. 修订历史记录 **364**
B.1. 致谢 364

索引 **365**

部分 I. 开始前

本文档部分提供了 Red Hat Enterprise Linux 中网络服务的基本概念概述。

第 1 章 网络主题概述

1.1. IP 与非 IP 网络比较

网络是互连的设备系统，可以沟通共享信息和资源，如文件、打印机、应用程序和互联网连接。每个设备都有一个唯一互联网协议(IP)地址，可以使用一组称为协议的规则在两个或多个设备之间发送和接收消息。

网络通信的类型

IP 网络

通过 Internet 协议地址通信的网络.IP 网络是在互联网和大多数内部网络中实现的。以太网、Cable Modems、DSL Modems、拨号调制解调器、无线网络和 VPN 连接都是典型示例。

非 IP 网络

用于通过较低层而不是传输层进行沟通的网络。请注意这些网络很少被使用。InfiniBand 是一个非 IP 网络，如 第 13 章 *配置 InfiniBand 和 RDMA 网络* 所述。

1.2. 静态与动态 IP 地址比较

静态 IP 地址

当为设备分配静态 IP 地址时，该地址不会随着时间变化，除非手动更改。如果需要，建议使用静态 **IP 地址** ：

- 确保 **DNS** 等服务器的网络地址一致性以及验证服务器。
- 使用独立于其他网络基础结构的带外管理设备。

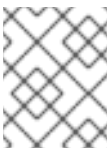
第 3.1 节 “选择网络配置方法” 中列出的所有配置工具都允许手动分配静态 **IP 地址**。**nmcli** 工具也适合，如 第 3.3.8 节 “使用 nmcli 添加和配置静态以太网连接” 中所述。

*有关自动配置和管理的更多信息，请参阅Red Hat Enterprise Linux 7 系统管理员指南中的 **OpenLMI** 章节。《Red Hat Enterprise Linux 7 安装指南》记录了* **Kickstart** 文件的使用，它也可用于自动分配网络设置。

动态 IP 地址

当为设备分配一个动态 IP 地址时，地址会随着时间推移而变化。因此，建议偶尔连接到网络的设备在重启计算机后可能会更改 IP 地址。

动态 IP 地址更灵活，更容易设置和管理。*动态主机控制协议* (DHCP)是一种传统方法，用于动态将网络配置分配到主机。如需更多信息，请参阅 第 14.1 节 “为什么使用 DHCP？”。您还可以使用 **nmcli** 工具，如 第 3.3.7 节 “使用 nmcli 添加和配置动态以太网连接” 所述。



注意

没有严格的规则来定义何时使用静态或动态 IP 地址。它取决于用户的需求、偏好和网络环境。

默认情况下，NetworkManager 调用 DHCP 客户端,dhclient。

1.3. 配置 DHCP 客户端行为

DHCP 客户端在每次连接到网络时都从 DHCP 服务器请求动态 IP 地址和对应配置信息。

请注意，NetworkManager 默认调用 DHCP 客户端,dhclient。

请求 IP 地址

当 **DHCP** 连接启动时，dhcp 客户端会从 **DHCP** 服务器请求 IP 地址。默认情况下，dhcp 客户端等待此请求完成的时间为 60 秒。您可以使用 **nmcli** 工具或 **/etc/sysconfig/network -scripts/ifcfg-*ifname*文件中的 IPV4_DHCP_TIMEOUT** 选项**来配置ipv4.dhcp- timeout** 属性。例如，使用 nmcli:

```
~]# nmcli connection modify enp1s0 ipv4.dhcp-timeout 10
```

如果在这个间隔内无法获取地址，则 IPv4 配置会失败。整个连接也可能失败，这取决于 **ipv4.may-fail** 属性：

- If **ipv4.may-fail** 被设置为 **yes** （默认），连接的状态取决于 IPv6 配置：
  1. 如果启用了 IPv6 配置并成功，连接将被激活，但无法再次重试 IPv4 配置。
  2. 如果禁用或未配置 IPv6 配置，连接会失败。
- If **ipv4.may-fail** 设置为 **没有** 停用连接。在这种情况下：
  1. 如果启用了 **连接的 autoconnect** 属性，NetworkManager 会重试以激活连接次数，次数与 **autoconnect-retries** 属性中设置的次数相同。默认值为 4。
  2. 如果连接仍然无法获得 dhcp 地址，则自动激活会失败。

     请注意，5 分钟后，自动连接过程将再次启动，dhcp 客户端会尝试从 dhcp 服务器获取地址。

请求租用续订

当 dhcp 地址被获取并且 IP 地址租期无法续订时，dhcp 客户端每 2 分钟重启三次，以尝试从 dhcp 服务器获得租用。每次都配置它，方法是将 **ipv4.dhcp-timeout** 属性设置为秒（默认为 60）以获取租用。如果在尝试过程中收到回复，则进程将停止，并且您的租期续订。

三次尝试失败后：

- If **ipv4.may-fail** 设置为 **yes** （默认）和 IPv6 配置成功，连接将被激活，dhcp 客户端每 2 分钟重新重新启动一次。
- If **ipv4.may-fail** 设置为 **no**，则连接将被停用。在这种情况下，如果连接启用了 **autoconnect** 属性，则从头激活连接。

1.3.1. 使 DHCPv4 持久

要使 DHCPv4 在启动和租期续订过程中持久化，请将 **ipv4.dhcp-timeout** 属性设置为 32 位整数 (MAXINT32)的最大值，即 **2147483647** 或 **infinity** 值：

```
~]$ nmcli connection modify enps1s0 ipv4.dhcp-timeout infinity
```

因此，NetworkManager 永远不会停止尝试从 DHCP 服务器获取或续订租期，直到成功为止。

要确保仅在租期续订过程中有 DHCP 持久性行为，您可以在 /etc/sysconfig/network-scripts/ifcfg-***enp1s0 配置文件中使用 nmcli 手动将静态 IP 添加到 /etc/sysconfig/network-scripts/ifcfg- enp1s0*** 配置文件中的 **IPADDR** 属性中：

```
~]$ nmcli connection modify enp1s0 ipv4.address 192.168.122.88/24
```

当 IP 地址租用到期时，静态 IP 会保留已配置或部分配置的 IP 状态（您可以拥有 IP 地址，但您未连接到互联网），确保 dhcp 客户端每 2 分钟重新启动一次。

1.4. 设置 WIRELESS REGULATORY 域

在 Red Hat Enterprise Linux 中，crda 软件包包含中央常规域代理，它为内核提供给定的无线监管规则。某些 udev 脚本使用它，除非调试 udev 脚本，否则不应该手动运行它。内核在新的规范域更改时发送 udev 事件来运行 crda。规范域更改由 Linux 无线子系统（IEEE-802.11）触发。该子系统使用 **Govern.bin** 文件来保存其规范数据库信息。

setregdomain 实用程序为您的系统设置规范域。**Setregdomain** 不使用任何参数，通常通过系统脚本（如 udev ）调用，而不是由管理员手动调用。如果国家代码查找失败，系统管理员可以在 **/etc/sysconfig/regdomain** 文件中定义 **COUNTRY** 环境变量。

有关规范域的更多信息，请参见以下手册页：

- **setregdomain(1)** 手册页 - 根据国家代码设置规范域.
- **CRDA(8)man** page - 向内核发送给定 ISO 或 IEC 3166 alpha2 的无线规范域.
- **Govern.bin(5)** 手册页 - 显示 Linux 无线法规数据库.
- **iw(8)手册页** - 显示或操作无线设备及其配置。

1.5. 配置 NETCONSOLE

如果磁盘日志记录失败或使用串行控制台，您可能需要使用内核调试。**netconsole** 内核模块允许通过网络将内核消息记录到另一台计算机。

为了能够使用 **netconsole**，您需要具有一个在网络上正确配置的 **rsyslog** 服务器。

过程 1.1. 为 netconsole 配置 rsyslog 服务器

1. 将 **rsyslogd** 守护进程配置为侦听 514/udp 端口，并通过取消注释 **/etc/rsyslog.conf** 文件的 **MODULES** 部分中的以下行从网络接收信息：

```
$ModLoad imudp
$UDPServerRun 514
```

2. 重启 **rsyslogd** 服务以使更改生效：

```
]# systemctl restart rsyslog
```

3. 验证 **rsyslogd** 是否在侦听 514/udp 端口：

```
]# netstat -l | grep syslog
udp        0      0 0.0.0.0:syslog          0.0.0.0:*
udp6       0      0 [::]:syslog             [::]:*
```

netstat -l 输出中的 **0.0.0.0:syslog 和 [::]:syslog** 值表示 **rsyslogd** 正在侦听 **/etc/services** 文件中定义的默认 **netconsole** 端口：

```
]$ cat /etc/services | grep syslog
syslog          514/udp
syslog-conn     601/tcp              # Reliable Syslog Service
syslog-conn     601/udp              # Reliable Syslog Service
syslog-tls      6514/tcp             # Syslog over TLS
syslog-tls      6514/udp             # Syslog over TLS
syslog-tls      6514/dccp            # Syslog over TLS
```

Netconsole 使用 **/etc/sysconfig/netconsole** 文件进行配置，该文件是 initscripts 软件包的一部分。默认情况下会安装此软件包，它还提供 **netconsole** 服务。

如果要配置发送机器，请按照以下步骤操作：

过程 1.2. 配置发送机器

1. 在 **/etc/sysconfig/netconsole** 文件中设置 **SYSLOGADDR** 变量的值，以匹配 **syslogd** 服务器的 IP 地址。例如：

```
SYSLOGADDR=192.168.0.1
```

2. 重启 **netconsole** 服务以使更改生效：

```
]# systemctl restart netconsole.service
```

3. 在重启系统后启用 **netconsole.service** 运行：

```
]# systemctl enable netconsole.service
```

4. 在 **/var/log/messages** 文件中或 **rsyslog.conf** 中指定的文件中查看来自客户端的 **netconsole** 消息。

```
]# cat /var/log/messages
```

注意

默认情况下，rsyslogd 和 **netconsole.service** 使用端口 514。要使用不同的端口，请将 **/etc/rsyslog.conf** 中的以下行改为所需的端口号：

```
$UDPServerRun <PORT>
```

在发送机器上，取消注释并编辑 **/etc/sysconfig/netconsole** 文件中的以下行：

```
SYSLOGPORT=514
```

有关 **netconsole** 配置和故障排除提示的更多信息，请参阅 Netconsole 内核文档。

1.6. 使用带有 SYSCTL 的网络内核可调项

通过 **sysctl** 实用程序使用某些内核可调项，您可以在运行中的系统上调整网络配置，并直接影响网络性能。

要更改网络设置，请使用 **sysctl** 命令。要进行系统重启后保留的永久性更改，请向 **/etc/sysctl.conf** 文件添加行。

要显示所有可用 **sysctl** 参数列表，以 **root** 用户身份输入：

```
~]# sysctl -a
```

有关使用 **sysctl** 的网络内核可调项的详情，请参阅《系统管理员指南》 的使用带有多个接口的 PTP 部分。

如需有关网络内核可调项的更多信息，请参阅内核管理指南中的网络接口可调项章节。

1.7. 使用 NCAT 工具管理数据

ncat 网络实用程序取代了红帽企业 Linux 7 中的 **netcat**。**ncat** 是可靠的后端工具，可向其他应用和用户提供网络连接。它从命令行在网络上读取和写入数据，并使用传输控制协议(TCP)、用户数据报协议(UDP)、流控制传输协议(SCTP)或 Unix 套接字进行通信。**ncat** 可以处理 **IPv4** 和 **IPv6**、打开连接、发送数据包、执行端口扫描，并支持更高级别的功能，如 **SSL** 和连接代理。

也可使用相同的选项，以 **ncat** 身份输入 Thenc **命令**。有关 **ncat** 选项的详情，请查看 *迁移规划指南和 ncat(1) man page 中的 New networking 工具(ncat) 部分* 。

安装 ncat

要安装 **ncat** 软件包，以 **root** 用户身份输入：

```
~]# yum install ncat
```

ncat 用例的简短选择

例 1.1. 启用客户端和服务器之间的通信

1. 将客户端机器设置为侦听 TCP 端口 *8080* 上的连接：

   ```
   ~]$ ncat -l 8080
   ```

2. 在服务器机器中指定客户端的 IP 地址并使用相同的端口号：

   ```
   ~]$ ncat 10.0.11.60 8080
   ```

   您可以在连接两侧发送消息，它们会显示在本地和远程计算机上。

3. 按 **Ctrl+D** 关闭 TCP 连接。

注意

要检查 UDP 端口，请使用带有 **-u** 选项的相同 **nc** 命令。例如：

```
~]$ ncat -u -l 8080
```

例 1.2. 发送文件

您可以将所有信息发送到文件，而不是在屏幕上打印信息，如上例中所述。例如，要通过 TCP 端口 *8080* 将文件从客户端发送到服务器：

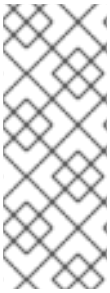
1. 在客户端机器上侦听将文件传输到服务器机器的特定端口：

```
~]$ ncat -l 8080 > outputfile
```

2. 在服务器机器上，指定客户端的 IP 地址、要传输的端口和文件：

```
~]$ ncat -l 10.0.11.60 8080 < inputfile
```

文件传输后，连接会自动关闭。



注意

您还可以在另一个方向传输文件：

```
~]$ ncat -l 8080 < inputfile
```

```
~]$ ncat -l 10.0.11.60 8080 > outputfile
```

例 1.3. 创建 HTTP 代理服务器

在 localhost 端口 *8080* 中创建 HTTP 代理服务器：

```
~]$ ncat -l --proxy-type http localhost 8080
```

例 1.4. 端口扫描

要查看哪些端口是开放的，请使用 **-z** 选项并指定要扫描的端口范围：

```
~]$ ncat -z 10.0.11.60 80-90
  Connection to 192.168.0.1 80 port [tcp/http] succeeded!
```

例 1.5. 使用 SSL 设置安全客户端-服务器通信

在服务器中设置 **SSL** ：

```
~]$ ncat -e /bin/bash -k -l 8080 --ssl
```

在客户端机器上：

```
~]$ ncat --ssl 10.0.11.60 8080
```

注意

为确保 **SSL** 连接的真正机密性，服务器需要 **--ssl-cert** 和 **--ssl-key** 选项，客户端需要 **--ssl-verify** 和 **--ssl-trustfile** 选项。有关 **OpenSSL** *的详情，请参考安全指南中的使用 OpenSSL 部分*。

更多示例请查看 *ncat(1)* man page。

部分 II. 管理 IP 网络

本文档部分提供了在 Red Hat Enterprise Linux 中配置和管理联网的详细说明。

第 2 章 NETWORKMANAGER 入门

2.1. NETWORKMANAGER 概述

在 Red Hat Enterprise Linux 7 中，默认网络服务由 **NetworkManager** 提供，后者是一个动态网络控制和配置守护进程，在网络设备和连接可用时保持启动和激活。传统的 **ifcfg** 类型配置文件仍受支持。如需更多信息，请参阅 第 2.6 节 “将 NetworkManager 与网络脚本搭配使用”。

2.1.1. 使用 NetworkManager 的好处

使用 NetworkManager 的主要优点是：

- 更轻松地进行网络配置： **NetworkManager** 确保网络连接正常工作。**当发现系统中没有网络配置但存在网络设备时，NetworkManager** 会创建临时连接以提供连接。
- **提供与用户的简单连接设置：NetworkManager 通过不同的工具（GUI、nmtui、nmcli -）**提供管理。请参阅 第 2.5 节 “NetworkManager 工具”。
- 支持配置灵活性。例如：配置 WiFi **接口，NetworkManager** 会扫描并显示可用的 wifi 网络。 **您可以选择一个接口，NetworkManager** 会显示在重启过程后提供自动连接所需的凭证。**NetworkManager** 可以配置网络别名、IP 地址、静态路由、DNS 信息和 VPN 连接，以及许多特定于连接的参数。您可以修改配置选项以反应您的需要。
- 通过 D-Bus 提供 API，允许应用程序查询和控制网络配置和状态。这样，应用程序可以通过 D-BUS 检查或配置网络。例如，通过 **Web** 浏览器监控和配置服务器的 Web 控制台界面使用 **NetworkManager** D-BUS 接口来配置网络。
- 重启过程后保持设备状态，并接管在重启过程中将其设定为受管模式的接口。
- 处理没有被显式设置但由用户或者其他网络设备手动控制的设备。

2.2. 安装 NETWORKMANAGER

默认情况下，NetworkManager 在 Red Hat Enterprise Linux 中安装。如果没有，以 **root** 用户身份输入：

```
~]# yum install NetworkManager
```

有关用户权限和获得权限的详情，请查看*Red Hat Enterprise Linux 系统管理员指南*。

2.3. 检查 NETWORKMANAGER 的状态

检查 **NetworkManager** 是否在运行：

```
~]$ systemctl status NetworkManager
NetworkManager.service - Network Manager
   Loaded: loaded (/lib/systemd/system/NetworkManager.service; enabled)
   Active: active (running) since Fri, 08 Mar 2013 12:50:04 +0100; 3 days ago
```

请注意，当 **NetworkManager** 没有运行时，systemctl **status** 命令会显示 **Active: inactive (dead)**。

2.4. 启动 NETWORKMANAGER

启动 **NetworkManager** ：

```
~]# systemctl start NetworkManager
```

在引导时自动启用 **NetworkManager** ：

```
~]# systemctl enable NetworkManager
```

有关启动、停止和管理服务的更多信息，请参阅*Red Hat Enterprise Linux 系统管理员指南*。

2.5. NETWORKMANAGER 工具

表 2.1. NetworkManager 工具和应用程序概述

应用程序或工具	描述
nmcli	命令行工具可让用户和脚本与 NetworkManager 交互。**请注意，** nmcli 可以在没有 GUI 的系统上使用，如服务器来控制 NetworkManager 的所有方面。它的功能与 GUI 工具相同。
nmtui	NetworkManager的基于 curses 的简单文本用户界面(TUI)
nm-connection-editor	**控制中心实用程序尚未处理的特定任务** （如配置绑定和成组连接）的图形用户界面工具。您可以添加、删除和修改 NetworkManager 存储的网络连接。要启动它，在终端中输入 **nm-connection-editor** ： `~]$ nm-connection-editor`
control-center	GNOME Shell 提供的图形用户界面工具，可供桌面用户使用。它整合了网络设置工具。要启动它，请按 **Super** 键进入 Activities Overview，键入 **Network**，然后按 **Enter 键**。此时会出现网络设置工具。
网络连接图标	GNOME Shell 提供的图形用户界面工具代表网络连接状态，如 NetworkManager 报告。该图标有多种状态，充当您当前使用的连接类型的可视化指示。

2.6. 将 NETWORKMANAGER 与网络脚本搭配使用

这部分论述了如何在网络脚本中运行脚本以及如何使用自定义命令。

术语 **网络脚本引用脚本 /etc/init.d/network** 及其调用的其他已安装脚本。虽然 NetworkManager 提供默认网络服务，但脚本和 NetworkManager 可以并行运行并协同工作。红帽建议先测试它们。

运行网络脚本

仅**使用 systemctl** 命令运行网络脚本：

```
systemctl start|stop|restart|status network
```

systemctl 工具会清除任何现有环境变量并确保正确执行。

在 Red Hat Enterprise Linux 7 中，首先启动 **NetworkManager**，并使用 NetworkManager 进行 **/etc/init.d/网络检查，以避免对 NetworkManager 的连接进行篡改。NetworkManager 旨在作为使用 sysconfig 配置文件的主要应用，/etc/init.d/network 则是辅助应用程序。**

/etc/init.d/network 脚本运行：

1. 手动 - 使用其中一个 **systemctl 命令start|stop|restart 网络**，

   或者

2. 如果网络服务已启用，则引导和关闭 - 作为 **systemctl enable network** 命令的结果。

这是一个手动过程，不会响应启动后发生的事件。用户也可以手动调用 **ifup** 和 **ifdown** 脚本。

注意

由于 initscripts 的技术限制，systemctl **reload network.service** 命令无法正常工作。要为网络服务应用新配置，请使用 **restart** 命令：

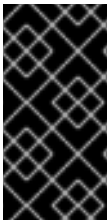
```
~]# systemctl restart network.service
```

这会关闭并启动所有网络接口卡(NIC)以加载新配置。如需更多信息，请参阅 Red Hat Knowledgebase 解决方案 Reload 和 force-reload 选项用于网络服务。

在网络脚本中使用自定义命令

只有在 /etc/init.d/network服务控制这些设备时，才会在 /sbin/ifup -local、ifdown-pre-local 脚本中执行自定义命令。默认情况下，ifup-local 文件不存在。如果需要，在 /sbin/ 目录下创建它。

ifup-local 脚本仅可由 initscripts 读取，NetworkManager 不可读取。要使用 NetworkManager 运行自定义脚本，请在分配程序.d/ 目录下创建它。请参阅 “运行 Dispatcher 脚本”一节。



重要

不建议修改 initscripts 软件包或相关 rpms 中包含的任何文件。如果用户修改这些文件，红帽不提供支持。

当网络连接上线和停机时，可以使用旧的 网络脚本 和 NetworkManager 运行自定义任务。如果启用了 NetworkManager ， ifup 和 ifdown 脚本将询问 NetworkManager 是否管理相关的接口，该接口可从 ifcfg 文件的 “DEVICE=” 行中找到。

由 NetworkManager 管理的设备：

调用 ifup

当您调用 ifup 且设备由 NetworkManager 管理时，有两个选项：

- 如果该设备尚未连接，则是否会询问 NetworkManager 启动连接。
- 如果设备已经连接，则不执行任何操作。

调用 ifdown

当您调用 ifdown 且设备由 NetworkManager 管理时 ：

- ifdown 要求 NetworkManager 终止连接。

网络管理器(NetworkManager)未管理的设备：

如果您调用 ifup 或 ifdown，该脚本将使用自 NetworkManager 存在之前的较旧的非网络管理器机制启动连接。

运行 Dispatcher 脚本

NetworkManager 提供了根据连接状态运行其他自定义脚本以启动或停止服务的方法。默认情况下，/etc/NetworkManager/dispatcher.d/ 目录存在，NetworkManager 会以字母顺序运行脚本。每个脚本必须是 root 拥有的 可执行文件，并且必须仅对文件所有者具有 写入权限。有关运行 NetworkManager 分配程序脚本的更多信息，请参阅红帽知识库解决方案如何编写 NetworkManager 分配程序脚本以应用 ethtool 命令。

2.7. 使用 NETWORKMANAGER 和 SYSCONFIG 文件

/etc/sysconfig/ 目录是配置文件和脚本的位置。除 VPN、移动宽带和 PPPoE 配置外，大多数网络配置信息都存储在 /etc/NetworkManager/ 子目录中。例如，特定于接口的信息存储在 /etc/sysconfig/network-scripts/ 目录下的 ifcfg 文件中。

对于全局设置，请使用 /etc/sysconfig/network 文件。VPN、移动宽带和 PPPoE 连接的信息存储在 /etc/NetworkManager/system-connections/ 中。

在 Red Hat Enterprise Linux 7 中，如果您编辑了 ifcfg 文件，NetworkManager 会自动不知道该更改，且必须提示您注意到更改。如果您使用其中一个工具更新 NetworkManager 配置集设置，NetworkManager 不会实现这些更改，直到您使用该配置集重新连接为止。例如，如果使用编辑器更改了配置文件，NetworkManager 必须再次读取配置文件。

要确定这一点，以 root 用户身份输入 以重新载入所有连接配置集：

```
~]# nmcli connection reload
```

或者，只重新载入一个更改的文件，ifcfg-*ifname*：

```
~]# nmcli con load /etc/sysconfig/network-scripts/ifcfg-ifname
```

请注意，您可以使用上述命令指定多个文件名。

使用 nmcli 等工具所做的更改不需要重新加载，但需要关闭关联的接口，然后再次启动：

```
~]# nmcli dev disconnect interface-name
```

```
~]# nmcli con up interface-name
```

有关 nmcli 的详情请参考 第 3.3 节 “使用 nmcli 配置 IP 网络”。

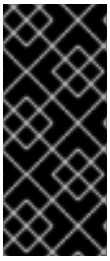
NetworkManager 不会触发任何网络脚本，尽管网络脚本在使用 ifup 命令时会尝试触发 NetworkManager。有关网络脚本的说明，请参阅 第 2.6 节 “将 NetworkManager 与网络脚本搭配使用”。

ifup 脚本是一个通用脚本，可执行一些操作，然后调用特定于接口的脚本，如 ifup-*device_name*、ifup-wireless、ifup-ppp 等。当用户手动运行 ifup enp1s0 时 ：

1. 如果查找名为 /etc/sysconfig/network-scripts/ifcfg-enp1s0 的文件；
2. 如果 ifcfg 文件存在，如果该文件中查找 TYPE 键，以确定要调用的特定类型脚本；
3. ifup 会调用 ifup-wireless 或 ifup-*device_name*，基于 TYPE ；
4. 特定于类型的脚本进行特定于类型的设置；
5. 特定于类型的脚本可让常见功能执行 IP相关任务，如 DHCP 或静态设置。

在引导时，/etc/init.d/network 会遍历所有 ifcfg 文件，以及每个具有 ONBOOT=yes 的文件读取所有 ifcfg 文件，它将检查 NetworkManager 是否已经从该 ifcfg 文件中启动 DEVICE。如果 NetworkManager 正在启动该设备或已启动该设备，则不再为该文件执行任何操作，并检查下一个 ONBOOT=yes 文件。如果 NetworkManager 尚未启动该设备，则 initscripts 会继续其传统行为并调用 ifup（如果是该 ifcfg 文件）。

其结果是，任何具有 ONBOOT=yes 的 ifcfg 文件应该在系统启动时(NetworkManager)或 initscripts 启动。这样可确保 NetworkManager 无法处理的某些传统网络类型（如 ISDN 或模拟的拨号模式ms）以及 NetworkManager 尚不支持的新应用程序仍能由 initscripts 正确启动，即使 NetworkManager 无法处理它们。



重要

建议不要将备份文件存储在 /etc 目录中的任何位置，或存储在与实时文件相同的位置，因为脚本在字面上执行 ifcfg-*。仅排除这些扩展：.old 、.orig 、.rpmnew 、.rpmorig 和.rpmsave。

有关使用 sysconfig 文件的详情请参考 第 3.5 节 “使用 ifcfg 文件配置 IP 网络” 和 *ifcfg(8)* man page。

2.8. 其它资源

- man(1) man page - 描述 man page 以及如何找到它们。
- NetworkManager(8)man page - 描述网络管理守护进程。
- NetworkManager.conf(5) 手册页 - 描述 NetworkManager 配置文件。
- /usr/share/doc/initscripts-*版本*/sysconfig.txt - 描述 ifcfg 配置文件及其指令，如传统网络服务所理解。
- /usr/share/doc/initscripts-*版本*/examples/networking/ - 包含示例配置文件的目录。
- ifcfg(8)手册页 - 简要描述 ifcfg 命令.

第 3 章 配置 IP 网络

作为系统管理员，您可以使用 NetworkManager 配置网络接口。

3.1. 选择网络配置方法

- 要使用 NetworkManager 配置网络接口，请使用以下工具之一：
  - 文本用户界面工具 nmtui。如需了解更多详细信息，请参阅 第 3.2 节 “使用 nmtui 配置 IP 网络”。
  - 命令行工具 nmcli。如需了解更多详细信息，请参阅 第 3.3 节 “使用 nmcli 配置 IP 网络”。
  - 图形用户界面工具 GNOME GUI。如需了解更多详细信息，请参阅 第 3.4 节 “ 使用 GNOME GUI 配置 IP 网络 ”。
- 在不使用 NetworkManager 的情况下配置网络接口：
  - 手动编辑 ifcfg 文件。如需了解更多详细信息，请参阅 第 3.5 节 “使用 ifcfg 文件配置 IP 网络”。
  - 使用 ip 命令。这可用于分配 IP 地址到接口，但更改在重新启动后不会保留；重新引导后，您将丢失任何更改。如需了解更多详细信息，请参阅 第 3.6 节 “使用 ip 命令配置 IP 网络”。
- 在 root 文件系统不是本地文件系统时配置网络设置 ：
  - 使用内核命令行。如需了解更多详细信息，请参阅 第 3.7 节 “使用内核命令行配置 IP

网络”。

3.2. 使用 NMTUI 配置 IP 网络

作为系统管理员，您可以使用 NetworkManager 的工具 nmtui 配置网络接口。请参阅 第 2.5 节 “NetworkManager 工具”。

这个步骤描述了如何使用文本用户界面工具 nmtui 配置网络。

先决条件

- nmtui 工具用于终端窗口中。它包含在 NetworkManager-tui 软件包中，但默认情况下不与 NetworkManager 一起安装。要安装 NetworkManager-tui：

```
~]# yum install NetworkManager-tui
```

- 要验证 NetworkManager 是否正在运行，请参阅 第 2.3 节 “检查 NetworkManager 的状态”。

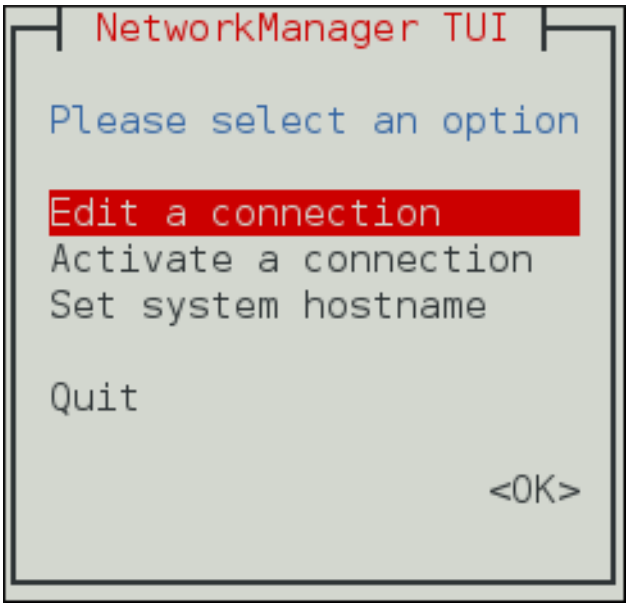
流程

1. 启动 nmtui 工具：

```
~]$ nmtui
```

此时将显示文本用户界面。

图 3.1. NetworkManager 文本用户界面起始菜单



[D]

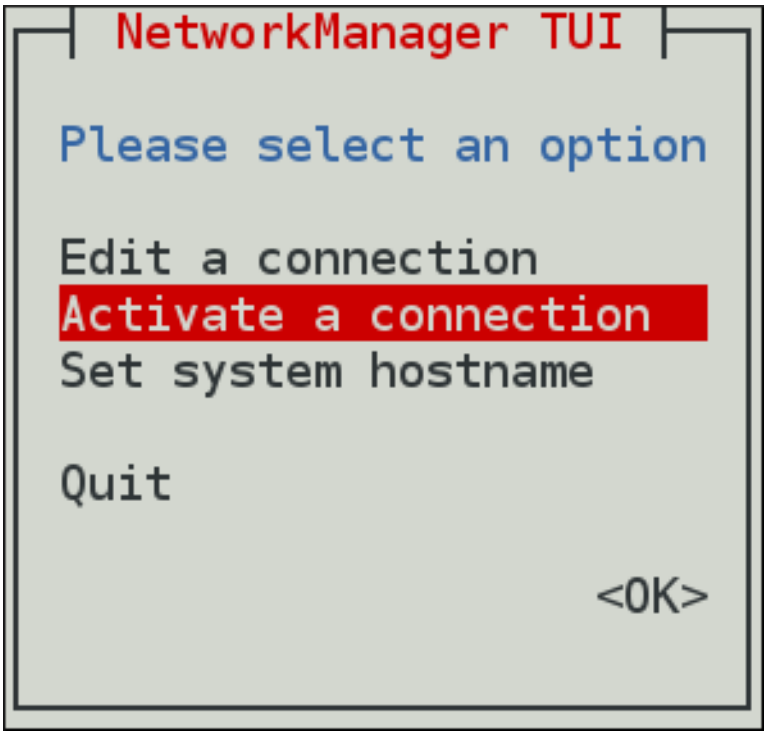
2. 要导航，请使用箭头键或按 Tab 键前进，然后按ShiftTab后退步浏览选项。按 Enter 键选择一个选项。Space bar 切换复选框的状态。

要在修改后已激活的连接后应用更改，需要重新激活连接。在这种情况下，请按照以下步骤操作：

流程

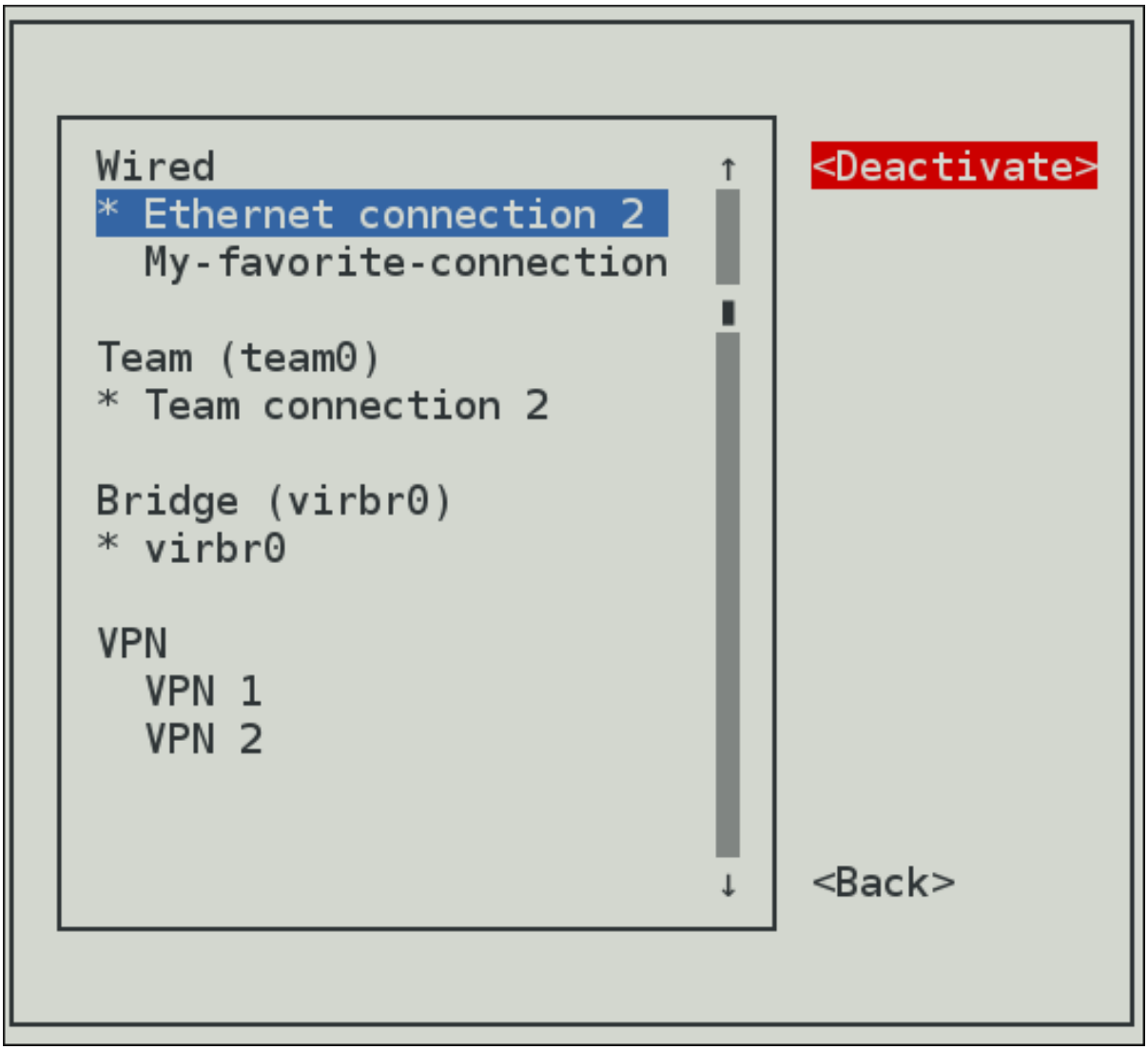
1. 选择 **Activate a connection** 菜单条目。

图 3.2. 激活连接



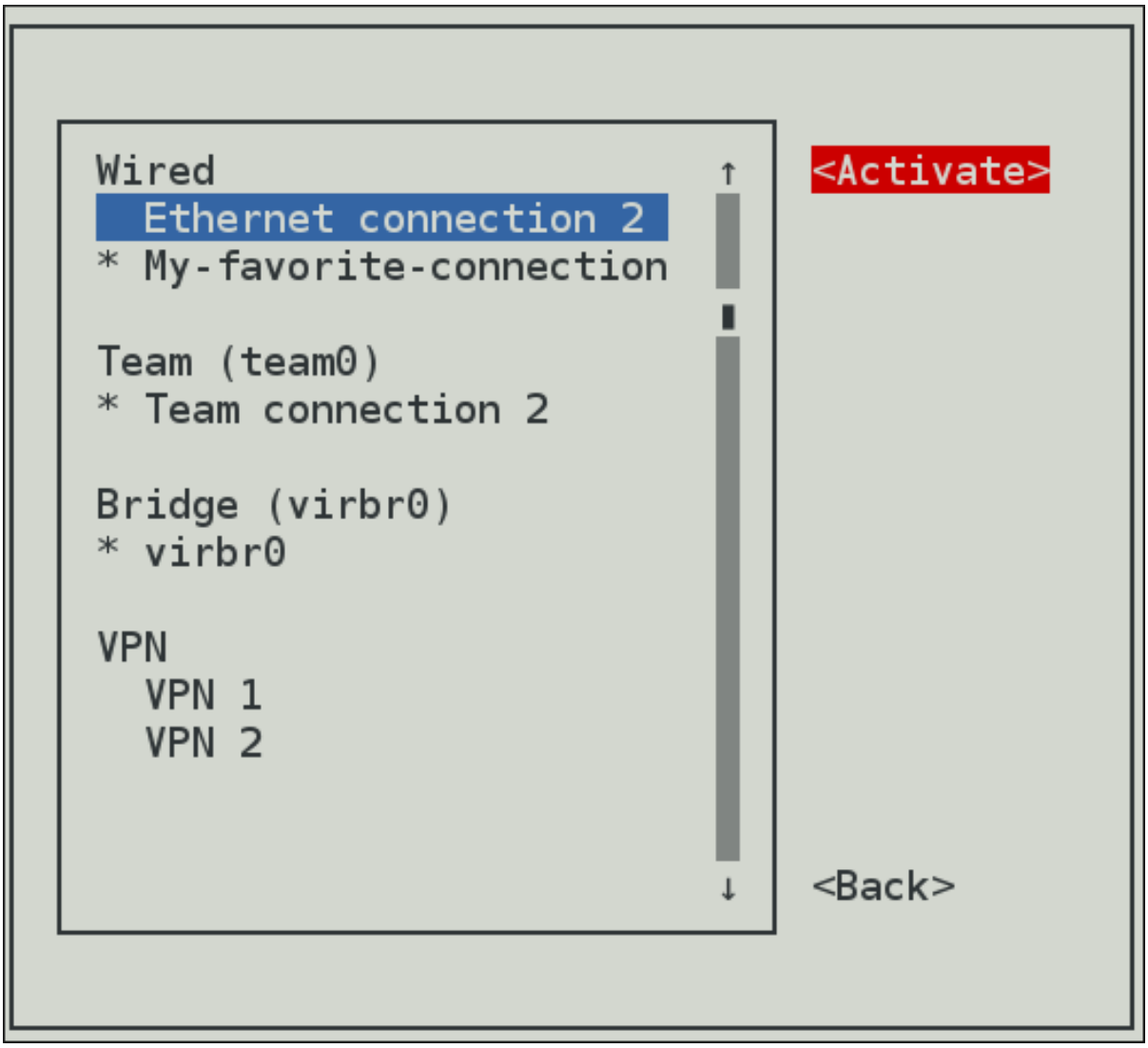
2. 选择**修改的连接。在右侧，**单击**取消激活按钮。**

图 3.3. 取消激活修改的连接



3. **再次选择连接并单击激活按钮。**

图 3.4. 重新激活修改的连接



还有一个可用的命令：

- nmtui edit *connection-name*

  如果没有提供连接名称，则会显示选择菜单。如果提供了连接名称并正确识别，则会出现相关的 Edit 连接屏幕。

- nmtui connect *connection-name*

如果没有提供连接名称，则会显示选择菜单。如果提供连接名称并正确识别，则会激活相关的连接。任何无效的命令都会打印用法消息。

请注意，nmtui 不支持所有类型的连接。特别是，您不能使用 WPA Enterprise 编辑 VPN、无线网络连接或使用 802.1X 的以太网连接。

3.3. 使用 NMCLI 配置 IP 网络

nmcli （NetworkManager 命令行界面）命令行工具用于控制 NetworkManager 和报告网络状态。它可用作 nm-applet 或其他图形客户端的替代。请参阅 第 2.5 节 “NetworkManager 工具”。nmcli 用于创建、显示、编辑、删除、激活和停用网络连接，以及控制和显示网络设备状态。

nmcli 工具可供用户和脚本用于控制 NetworkManager ：

- 对于服务器、无头计算机和终端，nmcli 可用于直接控制 NetworkManager，无需 GUI，包括创建、编辑、启动和停止网络连接，以及查看网络状态。
- 对于脚本，nmcli 支持更适合脚本处理的 terse 输出格式。它是集成网络配置而不是手动管理网络连接的一种方式。

nmcli 命令的基本格式如下：

```
nmcli [OPTIONS] OBJECT { COMMAND | help }
```

其中 OBJECT 可以是以下选项之一： 一般、联网、无线、连接、设备、代理 和监控 器。您可以在命令中使用这些选项的任何前缀。例如，nmcli con help、nmcli c help 和nmcli 连接帮助生成相同的输出。

要开始的一些有用的可选 OPTIONS 是：

-t, terse

此模式可用于计算机脚本处理，因为您可以看到仅显示值的 terse 输出。

例 3.1. 查看 terse 输出

```
nmcli -t device
ens3:ethernet:connected:Profile 1
lo:loopback:unmanaged:
```

-f, 字段

此选项指定输出中可以显示哪些字段。例如，NAME,UUID,TYPE,AUTOCONNECT,ACTIVE,DEVICE,STATE.您可以使用一个或多个字段。如果要使用更多，请不要在逗号后使用空格来分隔字段。

例 3.2. 指定输出中的字段

```
~]$ nmcli -f DEVICE,TYPE device
DEVICE  TYPE
ens3    ethernet
lo      loopback
```

甚至更适合编写脚本：

```
~]$ nmcli -t -f DEVICE,TYPE device
ens3:ethernet
lo:loopback
```

-p，相当好。

此选项可使 nmcli 生成人类可读的输出。例如，值是一致的，打印标头。

例 3.3. 以用户友善模式查看输出

```
nmcli -p device
=====================
```

```
 Status of devices
=====================
DEVICE  TYPE      STATE      CONNECTION
---------------------------------------------------------------
ens3    ethernet  connected  Profile 1
lo      loopback  unmanaged  --
```

-h, help

打印帮助信息.

nmcli 工具有一些内置上下文敏感的帮助：

nmcli help

此命令列出可在后续命令中使用的可用选项和对象名称。

nmcli ***对象帮助***

此命令显示与指定对象相关的可用操作列表。例如：

```
nmcli c help
```

3.3.1. nmcli 示例的简短选择

例 3.4. 检查 NetworkManager 的整体状态

```
~]$ nmcli general status
STATE      CONNECTIVITY  WIFI-HW  WIFI     WWAN-HW  WWAN
connected  full          enabled  enabled  enabled  enabled
```

在 terse 模式中：

```
~]$ nmcli -t -f STATE general
connected
```

例 3.5. 查看 **NetworkManager** 日志状态

```
~]$ nmcli general logging
 LEVEL  DOMAINS
 INFO   PLATFORM,RFKILL,ETHER,WIFI,BT,MB,DHCP4,DHCP6,PPP,WIFI_SCAN,IP4,IP6,A
UTOIP4,DNS,VPN,SHARING,SUPPLICANT,AGENTS,SETTINGS,SUSPEND,CORE,DEVICE,OL
PC,
WIMAX,INFINIBAND,FIREWALL,ADSL,BOND,VLAN,BRIDGE,DBUS_PROPS,TEAM,CONCHE
CK,DC
B,DISPATCH
```

例 3.6. 查看所有连接

```
~]$ nmcli connection show
 NAME       UUID                                  TYPE      DEVICE
Profile 1  db1060e9-c164-476f-b2b5-caec62dc1b05  ethernet    ens3
ens3       aaf6eb56-73e5-4746-9037-eed42caa8a65  ethernet    --
```

例 3.7. 仅查看当前活跃的连接

```
~]$ nmcli connection show --active
 NAME       UUID                                  TYPE      DEVICE
Profile 1  db1060e9-c164-476f-b2b5-caec62dc1b05  ethernet    ens3
```

例 3.8. 仅查看 NetworkManager 识别的设备及其状态

```
~]$ nmcli device status
DEVICE  TYPE      STATE      CONNECTION
ens3    ethernet  connected  Profile 1
lo      loopback  unmanaged  --
```

您还可以使用以下 nmcli 命令缩写：

表 3.1. 某些 nmcli 命令的缩写

nmcli 命令	缩写
nmcli 常规状态	nmcli g
nmcli 常规日志记录	nmcli g 日志
nmcli 连接显示	nmcli con show
nmcli connection show --active	nmcli con show -a
nmcli 设备状态	nmcli dev

更多示例请查看 *nmcli-examples(5)* man page。

3.3.2. 使用 nmcli 启动和停止网络接口

nmcli 工具可用于启动和停止任何网络接口，包括控制器。例如：

```
nmcli con up id bond0
nmcli con up id port0
nmcli dev disconnect bond0
nmcli dev disconnect ens3
```

注意

nmcli connection down 命令取消激活设备的连接，而不阻止设备进一步自动激活。nmcli device disconnect 命令断开连接设备并阻止设备自动激活进一步连接，而无需人工干预。

3.3.3. 了解 nmcli 选项

以下是一些重要 nmcli 属性选项：请参阅 *nmcli(1)* man page 中的综合列表：

connection.type

连接类型。允许的值有： adsl、bond-slave、bridge-slave、bridge-slave、bluetooth、cdma、以太网、gsm、infiniband、olpc-mesh、team、team-slave、vlan、wifi、wimax。每一连接类型具有特定于类型的命令选项。您可以在 *nmcli(1)* man page 中看到 TYPE_SPECIFIC_OPTIONS 列表。例如：

- **gsm 连接需要在 apn 中指定的接入点名称。**

```
nmcli c add connection.type gsm apn access_point_name
```

- **Wifi 设备需要在 ssid 中指定的服务集标识符。**

```
nmcli c add connection.type wifi ssid My identifier
```

connection.interface-name

与连接相关的设备名称。

```
nmcli con add connection.interface-name enp1s0 type ethernet
```

connection.id

用于连接配置集的名称。如果没有指定连接名称，将按如下方式生成一个连接名称：

```
connection.type -connection.interface-name
```

connection.id 是 *连接配置集* 的名称，不应与表示设备（wlp61s0、ens3、em1）的接口名称混淆。但是，用户可以在接口后命名连接，但它们不是同一事物。个设备可以有多个连接配置集。这对于移动设备或在不同设备之间来回切换网络电缆特别有用。根据需要创建不同的配置集并将它们应用到接口，而不是编辑配置。id 选项还引用连接配置集名称。

nmcli 命令（如 show、up 和down ）的最重要选项有：

id

用户分配给连接配置集的身份字符串。id 可用于 nmcli 连接命令来识别连接。命令输出中的 NAME 字段始终表示连接 ID。它引用了 con-name 执行的相同连接配置集名称。

uuid

系统分配给连接配置集的唯一标识字符串。可以在 nmcli 连接 命令中使用 uuid 来识别连接。

3.3.4. 使用 nmcli Interactive Connection Editor

nmcli 工具具有交互式连接编辑器。使用它：

```
~]$ nmcli con edit
```

系统将提示您从显示的列表中输入有效的连接类型。输入连接类型后，您将放置在 nmcli 提示符下。如果您熟悉连接类型，您可以在 nmcli con edit 命令中添加有效的连接类型选项，并直接转到 nmcli 提示符。格式用于编辑现有连接配置集：

```
nmcli con edit [id | uuid | path] ID
```

用于编辑新的连接配置集：

```
nmcli con edit [type new-connection-type] [con-name new-connection-name]
```

在 nmcli 提示符处键入 help 以查看有效命令的列表。使用 describe 命令获取设置及其属性的描述：

```
describe setting.property
```

，例如：

```
nmcli> describe team.config
```

3.3.5. 使用 nmcli 创建和修改连接配置集

连接配置集包含连接数据源所需的连接属性信息。

使用 nmcli 为 NetworkManager 创建新配置集 ：

```
nmcli c add {ARGUMENTS}
```

nmcli c add 接受两种不同类型的参数：

属性名称

NetworkManager 用于在内部描述连接的名称。最重要的是：

- connection.type

```
nmcli c add connection.type bond
```

- connection.interface-name

```
nmcli c add connection.interface-name enp1s0
```

- connection.id

```
nmcli c add connection.id "My Connection"
```

有关属性及其设置的更多信息，请参阅 nm-settings(5) man page。

别名名称

内部转换为属性的人类可读名称。最常用的是：

- type（ connection.type 属性）

```
nmcli c add type bond
```

- ifname（ connection.interface-name 属性）

```
nmcli c add ifname enp1s0
```

- con-name（ connection.id 属性）

```
nmcli c add con-name "My Connection"
```

在之前的 nmcli 版本中，要使用 别名 创建所需的连接。例如，ifname enp1s0 和 con-name My Connection。可以使用以下格式的命令：

```
nmcli c add type ethernet ifname enp1s0 con-name "My Connection"
```

在较新的版本中，属性名称和 别名 都可以互换使用。以下示例都是有效且等同的：

```
nmcli c add type ethernet ifname enp1s0 con-name "My Connection" ethernet.mtu 1600
```

```
nmcli c add connection.type ethernet ifname enp1s0 con-name "My Connection" ethernet.mtu 1600
```

```
nmcli c add connection.type ethernet connection.interface-name enps1s0 connection.id  "My Connection" ethernet.mtu 1600
```

参数因连接类型而异。只有 type 参数适用于所有连接类型，if name 是 bond、team、bridge 和 vlan 以外的所有类型。

type *type_name*

连接类型.例如：

```
nmcli c add type bond
```

ifname *interface_name*

要绑定连接的接口。例如：

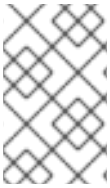
```
nmcli c add ifname interface_name type ethernet
```

要修改连接配置集的一个或多个属性，请使用以下命令：

```
nmcli c modify
```

例如，要将 connection.id 从 My Connection 改为 My favorite 连接，并将 connection .interface-name 改为 enp1s0，请按如下方式发出该命令：

```
nmcli c modify "My Connection" connection.id "My favorite connection" connection.interface-name enp1s0
```



注意

最好使用 属性名称。别名 仅用于兼容性原因。

另外，要将以太网 MTU 设置为 1600，请按如下方式修改大小：

```
nmcli c modify "My favorite connection" ethernet.mtu 1600
```

要使用 nmcli 在修改的连接后应用更改，请输入以下命令再次激活连接：

```
nmcli con up con-name
```

例如：

```
nmcli con up My-favorite-connection
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/16)
```

3.3.6. 使用 nmcli 连接到网络

列出当前可用的网络连接：

```
~]$ nmcli con show
NAME            UUID                                  TYPE           DEVICE
Auto Ethernet   9b7f2511-5432-40ae-b091-af2457dfd988  802-3-ethernet --
ens3            fb157a65-ad32-47ed-858c-102a48e064a2  802-3-ethernet ens3
MyWiFi          91451385-4eb8-4080-8b82-720aab8328dd  802-11-wireless wlp61s0
```

请注意，输出中的 NAME 字段始终表示连接 ID（名称）。它不是接口名称，即使它可能看起来相同。在上面显示的第二个连接中，NAME 字段中的 ens3 是用户提供给应用到接口的配置集的连接 ID ens3.在显示的最后一个连接中，用户已将连接 ID MyWiFi 分配给接口 wlp61s0.

添加以太网连接意味着创建分配给设备的配置配置集。在创建新配置集前，请查看可用设备，如下所示：

```
~]$ nmcli device status
DEVICE  TYPE      STATE         CONNECTION
ens3    ethernet  disconnected  --
ens9    ethernet  disconnected  --
lo      loopback  unmanaged     --
```

3.3.7. 使用 nmcli 添加和配置动态以太网连接

添加动态以太网连接

使用动态 IP 配置添加以太网配置配置集，允许 DHCP 分配网络配置：

```
nmcli connection add type ethernet con-name connection-name ifname interface-name
```

例如，创建名为 *my-office* 的动态连接配置集：

```
~]$ nmcli con add type ethernet con-name my-office ifname ens3
Connection 'my-office' (fb157a65-ad32-47ed-858c-102a48e064a2) successfully added.
```

打开以太网连接：

```
~]$ nmcli con up my-office
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/5)
```

检查设备和连接的状态：

```
~]$ nmcli device status
DEVICE  TYPE      STATE         CONNECTION
ens3    ethernet  connected     my-office
ens9    ethernet  disconnected  --
lo      loopback  unmanaged     --
```

配置动态以太网连接

要将主机发送的主机名更改为 DHCP 服务器，请修改 dhcp-hostname 属性：

```
~]$ nmcli con modify my-office my-office ipv4.dhcp-hostname host-name ipv6.dhcp-hostname host-name
```

要将主机发送的 IPv4 客户端 ID 更改为 DHCP 服务器，请修改 dhcp-client-id 属性：

```
~]$ nmcli con modify my-office my-office ipv4.dhcp-client-id client-ID-string
```

IPv6 没有 dhcp-client-id 属性，dhclient 为 IPv6 创建标识符。详情请查看 thedhclient(8) 手册页。

要忽略 DHCP 服务器发送到主机的 DNS 服务器，修改 ignore-auto-dns 属性：

```
~]$ nmcli con modify my-office my-office ipv4.ignore-auto-dns yes ipv6.ignore-auto-dns yes
```

有关属性及其设置的更多信息，请参阅 nm-settings(5) man page。

例 3.9. 使用交互编辑器配置动态以太网连接

使用互动编辑器配置动态以太网连接：

```
~]$ nmcli con edit type ethernet con-name ens3

===| nmcli interactive connection editor |===

Adding a new '802-3-ethernet' connection

Type 'help' or '?' for available commands.
Type 'describe [<setting>.<prop>]' for detailed property description.

You may edit the following settings: connection, 802-3-ethernet (ethernet), 802-1x, ipv4, ipv6, dcb
nmcli> describe ipv4.method

=== [method] ===
[NM property description]
IPv4 configuration method.  If 'auto' is specified then the appropriate automatic method (DHCP, PPP, etc) is used for the interface and most other properties can be left unset.  If
```

```
'link-local' is specified, then a link-local address in the 169.254/16 range will be assigned to
the interface.  If 'manual' is specified, static IP addressing is used and at least one IP
address must be given in the 'addresses' property.  If 'shared' is specified (indicating that
this connection will provide network access to other computers) then the interface is
assigned an address in the 10.42.x.1/24 range and a DHCP and forwarding DNS server are
started, and the interface is NAT-ed to the current default network connection.  'disabled'
means IPv4 will not be used on this connection.  This property must be set.

nmcli> set ipv4.method auto
nmcli> save
Saving the connection with 'autoconnect=yes'. That might result in an immediate activation
of the connection.
Do you still want to save? [yes] yes
Connection 'ens3' (090b61f7-540f-4dd6-bf1f-a905831fc287) successfully saved.
nmcli> quit
~]$
```

默认操作是将连接配置集保存为持久。如果需要，通过 save 临时命令，配置集只能保存在内存中，直到下次重启为止。

3.3.8. 使用 nmcli 添加和配置静态以太网连接

添加静态以太网连接

要使用静态 IPv4 配置添加以太网连接：

```
nmcli connection add type ethernet con-name connection-name ifname interface-name ip4
address gw4 address
```

IPv6 地址和网关信息可使用 ip6 和gw6 选项添加。

例如，创建仅使用 IPv4 地址和网关的静态以太网连接：

```
~]$ nmcli con add type ethernet con-name test-lab ifname ens9 ip4 10.10.10.10/24 \
gw4 10.10.10.254
```

另外，还可为设备指定 IPv6 地址和网关：

```
~]$ nmcli con add type ethernet con-name test-lab ifname ens9 ip4 10.10.10.10/24 \
gw4 10.10.10.254 ip6 abbe::cafe gw6 2001:db8::1
Connection 'test-lab' (05abfd5e-324e-4461-844e-8501ba704773) successfully added.
```

设置两个 IPv4 DNS 服务器地址：

```
~]$ nmcli con mod test-lab ipv4.dns "8.8.8.8 8.8.4.4"
```

请注意，这将取代任何先前设置的 DNS 服务器。设置两个 IPv6 DNS 服务器地址：

```
~]$ nmcli con mod test-lab ipv6.dns "2001:4860:4860::8888 2001:4860:4860::8844"
```

请注意，这将取代任何先前设置的 DNS 服务器。另外，要将额外的 DNS 服务器添加到任何先前设置中，请使用 + 前缀：

```
~]$ nmcli con mod test-lab +ipv4.dns "8.8.8.8 8.8.4.4"
```

```
~]$ nmcli con mod test-lab +ipv6.dns "2001:4860:4860::8888 2001:4860:4860::8844"
```

打开新的以太网连接：

```
~]$ nmcli con up test-lab ifname ens9
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/6)
```

检查设备和连接的状态：

```
~]$ nmcli device status
DEVICE  TYPE      STATE      CONNECTION
ens3    ethernet  connected  my-office
ens9    ethernet  connected  test-lab
lo      loopback  unmanaged  --
```

要查看有关新配置连接的详细信息，请按如下所示发出命令：

```
~]$ nmcli -p con show test-lab
===============================================================================

                Connection profile details (test-lab)
===============================================================================

connection.id:                          test-lab
connection.uuid:                        05abfd5e-324e-4461-844e-8501ba704773
connection.interface-name:              ens9
```

```
connection.type:                        802-3-ethernet
connection.autoconnect:                 yes
connection.timestamp:                   1410428968
connection.read-only:                   no
connection.permissions:
connection.zone:                        --
connection.master:                      --
connection.slave-type:                  --
connection.secondaries:
connection.gateway-ping-timeout:        0
[output truncated]
```

使用 **-p, --pretty** 选项会在输出中添加一个标题横幅和部分中断。

例 3.10. 使用交互编辑器配置静态以太网连接

使用互动编辑器配置静态以太网连接：

```
~]$ nmcli con edit type ethernet con-name ens3

===| nmcli interactive connection editor |===

Adding a new '802-3-ethernet' connection

Type 'help' or '?' for available commands.
Type 'describe [>setting<.>prop<]' for detailed property description.

You may edit the following settings: connection, 802-3-ethernet (ethernet), 802-1x, ipv4,
ipv6, dcb
nmcli> set ipv4.addresses 192.168.122.88/24
Do you also want to set 'ipv4.method' to 'manual'? [yes]: yes
nmcli>
nmcli> save temporary
Saving the connection with 'autoconnect=yes'. That might result in an immediate activation
of the connection.
Do you still want to save? [yes] no
nmcli> save
Saving the connection with 'autoconnect=yes'. That might result in an immediate activation
of the connection.
Do you still want to save? [yes] yes
Connection 'ens3' (704a5666-8cbd-4d89-b5f9-fa65a3dbc916) successfully saved.
nmcli> quit
~]$
```

默认操作是将连接配置集保存为持久。如果需要，通过 **save** 临时命令，配置集只能保存在内存中，直到下次重启为止。

NetworkManager 会将其内部参数 **connection.autoconnect** 设为 **yes**。NetworkManager 还会将

设置写出到 /etc/sysconfig/network-scripts/ifcfg-my-office，其中对应的 BOOTPROTO 将设置为 none，ONBOOT 设为 yes。

请注意，在接口下次启动之前，NetworkManager 不会注意到对 ifcfg 文件的手动更改。有关使用配置文件的更多信息，请参阅 第 2.7 节 “使用 NetworkManager 和 sysconfig 文件”、第 3.5 节 “使用 ifcfg 文件配置 IP 网络”。

3.3.9. 使用 nmcli 将配置集锁定到特定设备

要将配置集锁定到特定的接口设备：

```
nmcli connection add type ethernet con-name connection-name ifname interface-name
```

，要使配置集可用于所有兼容以太网接口：

```
nmcli connection add type ethernet con-name connection-name ifname "*"
```

请注意，即使您不想设置特定的接口，也必须使用 ifname 参数。使用通配符 * 指定配置集可用于任何兼容的设备。

将配置集锁定到特定的 MAC 地址：

```
nmcli connection add type ethernet con-name "connection-name" ifname "*" mac
00:00:5E:00:53:00
```

3.3.10. 使用 nmcli 添加 Wi-Fi 连接

查看可用的 Wi-Fi 接入点：

```
~]$ nmcli dev wifi list
  SSID          MODE  CHAN  RATE     SIGNAL  BARS  SECURITY
  FedoraTest    Infra  11    54 MB/s  98      ▂▄▆█  WPA1
  Red Hat Guest  Infra  6    54 MB/s  97      ▂▄▆_  WPA2
  Red Hat       Infra  6    54 MB/s  77      ▂▄▆_  WPA2 802.1X
* Red Hat       Infra  40   54 MB/s  66      ▂▄▆_  WPA2 802.1X
  VoIP          Infra  1    54 MB/s  32      ▂▄__  WEP
  MyCafe        Infra  11   54 MB/s  39      ▂▄__  WPA2
```

使用静态 IP 配置创建 Wi-Fi 连接配置集，但允许自动 DNS 地址分配：

```
~]$ nmcli con add con-name MyCafe ifname wlp61s0 type wifi ssid MyCafe \
ip4 192.168.100.101/24 gw4 192.168.100.1
```

设置 WPA2 密码，如 “caffeine”：

```
~]$ nmcli con modify MyCafe wifi-sec.key-mgmt wpa-psk
~]$ nmcli con modify MyCafe wifi-sec.psk caffeine
```

有关密码安全性的信息，请参阅Red Hat Enterprise Linux 7 安全指南。

更改 Wi-Fi 状态：

```
~]$ nmcli radio wifi [on | off ]
```

使用 nmcli 更改特定属性

检查特定属性，如 mtu ：

```
~]$ nmcli connection show id 'MyCafe' | grep mtu
802-11-wireless.mtu:                    auto
```

更改设置的属性：

```
~]$ nmcli connection modify id 'MyCafe' 802-11-wireless.mtu 1350
```

验证更改：

```
~]$ nmcli connection show id 'MyCafe' | grep mtu
802-11-wireless.mtu:                    1350
```

请注意，NetworkManager 指代 802-3-ethernet 和 802-11-wireless 等参数，mtu 作为设置的属性。有关属性及其设置的更多信息，请参阅 nm-settings(5) man page。

3.3.11. 将 NetworkManager 配置为 Ignore Certain 设备

默认情况下，NetworkManager 管理 lo （环回）设备以外的所有设备。但是，您可以将某些设备设置为非受管设备，以配置 NetworkManager 忽略这些设备。使用这个设置，您可以手动管理这些设备，

例如使用脚本。

3.3.11.1. 将设备永久配置为 NetworkManager 中的非受管设备

您可以根据多个条件将设备配置为非受管设备，如接口名称、MAC 地址或设备类型。这个步骤描述了如何永久将 enp1s0 接口设置为 NetworkManager 中的非受管接口。

要将网络设备临时配置为非受管设备，请参阅 第 3.3.11.2 节 “将设备临时配置为 NetworkManager 中的非受管设备”。

流程

1. 可选：显示要标识您要设置为非受管设备的设备列表 ：

```
# nmcli device status
DEVICE  TYPE      STATE         CONNECTION
enp1s0  ethernet  disconnected  --
...
```

2. 使用以下内容创建 /etc/NetworkManager/conf.d/99-unmanaged-devices.conf 文件：

```
[keyfile]
unmanaged-devices=interface-name:enp1s0
```

要将多个设备设置为非受管设备，请使用分号分隔 *unmanaged-devices* 参数中的条目：

```
[keyfile]
unmanaged-devices=interface-name:interface_1;interface-name:interface_2;...
```

3. 重新载入 NetworkManager 服务：

```
# systemctl reload NetworkManager
```

验证步骤

- 显示设备列表：

```
# nmcli device status
DEVICE  TYPE      STATE      CONNECTION
```

```
enp1s0  ethernet  unmanaged  --
...
```

enp1s0 设备旁边的非受管状态表示 NetworkManager 不管理此设备。

其它资源

有关用来将设备配置为非受管以及对应语法的标准列表，请查看 NetworkManager.conf(5) man page 中的 *设备列表格式* 部分。

3.3.11.2. 将设备临时配置为 NetworkManager 中的非受管设备

您可以根据多个条件将设备配置为非受管设备，如接口名称、MAC 地址或设备类型。这个步骤描述了如何临时将 enp1s0 接口设置为 NetworkManager 中的非受管接口。

可以使用这个方法用于特定目的，如测试。要将网络设备永久配置为非受管设备，请参阅 第 3.3.11.1 节 “将设备永久配置为 NetworkManager 中的非受管设备”。

流程

1. **可选：显示要标识您要设置为非受管设备的设备列表 ：**

```
# nmcli device status
DEVICE  TYPE      STATE         CONNECTION
enp1s0  ethernet  disconnected  --
...
```

2. **将 enp1s0 设备设置为非受管状态 ：**

```
# nmcli device set enp1s0 managed no
```

验证步骤

- **显示设备列表：**

```
# nmcli device status
DEVICE  TYPE      STATE      CONNECTION
enp1s0  ethernet  unmanaged  --
...
```

enp1s0 设备旁边的非受管状态表示 NetworkManager 不管理此设备。

其它资源

有关用来将设备配置为非受管以及对应语法的标准列表，请查看 NetworkManager.conf(5) man page 中的 *设备列表格式* 部分。

3.4. 使用 GNOME GUI 配置 IP 网络

在 Red Hat Enterprise Linux 7 中，NetworkManager 本身没有图形用户界面(GUI)。桌面右上角的网络连接图标作为 GNOME Shell 的一部分提供，网络设置配置工具作为支持有线、无线和 vpn 连接的新 GNOME 控制中心 GUI 的一部分提供。nm-connection-editor 是 GUI 配置的主要工具。除了 control-center 的功能外，它还应用并非由 GNOME 控制中心提供的功能，如配置绑定、团队、网桥连接。在本节中，您可以使用以下方法配置网络接口：

- GNOME control-center 应用程序
- GNOME nm-connection-editor 应用程序

3.4.1. 使用 control-center GUI 连接到网络

可以通过两种方式访问 control-center 应用程序的 Network settings 窗口：

- 按 Super 键进入 Activities Overview，键入 Settings，然后按 Enter 键。然后，选择左侧的网络 选项卡，并显示 网络设置 工具。继续 “使用 control-center 配置新连接”一节。
- 单击屏幕右上角的 GNOME Shell 网络连接图标，以打开其菜单。

图 3.5. 使用 control-center 应用进行网络配置

[D]

当您点击 GNOME Shell 网络连接图标时，您会看到：

- 您当前连接的分类网络列表（如 Wired 和 Wi-Fi）。
- NetworkManager 检测到的所有可用网络的列表。
- 连接任何已配置的虚拟专用网络(VPN)的选项.

  和

- 选择 Network Settings 菜单条目的选项。

如果您连接到某个网络，则通过连接名称左侧的黑色要点表示。

如果单击 Network Settings，则会出现 Network settings 工具。继续 “使用 control-center 配置新

连接”一节。

3.4.2. 使用 GUI 配置新连接并编辑现有连接

作为系统管理员，您可以配置网络连接。这允许用户应用或更改接口的设置。为此，您可以使用以下两种方式之一：

- GNOME control-center 应用程序
- GNOME nm-connection-editor 应用程序

3.4.2.1. 使用 control-center 配置新的和编辑现有连接

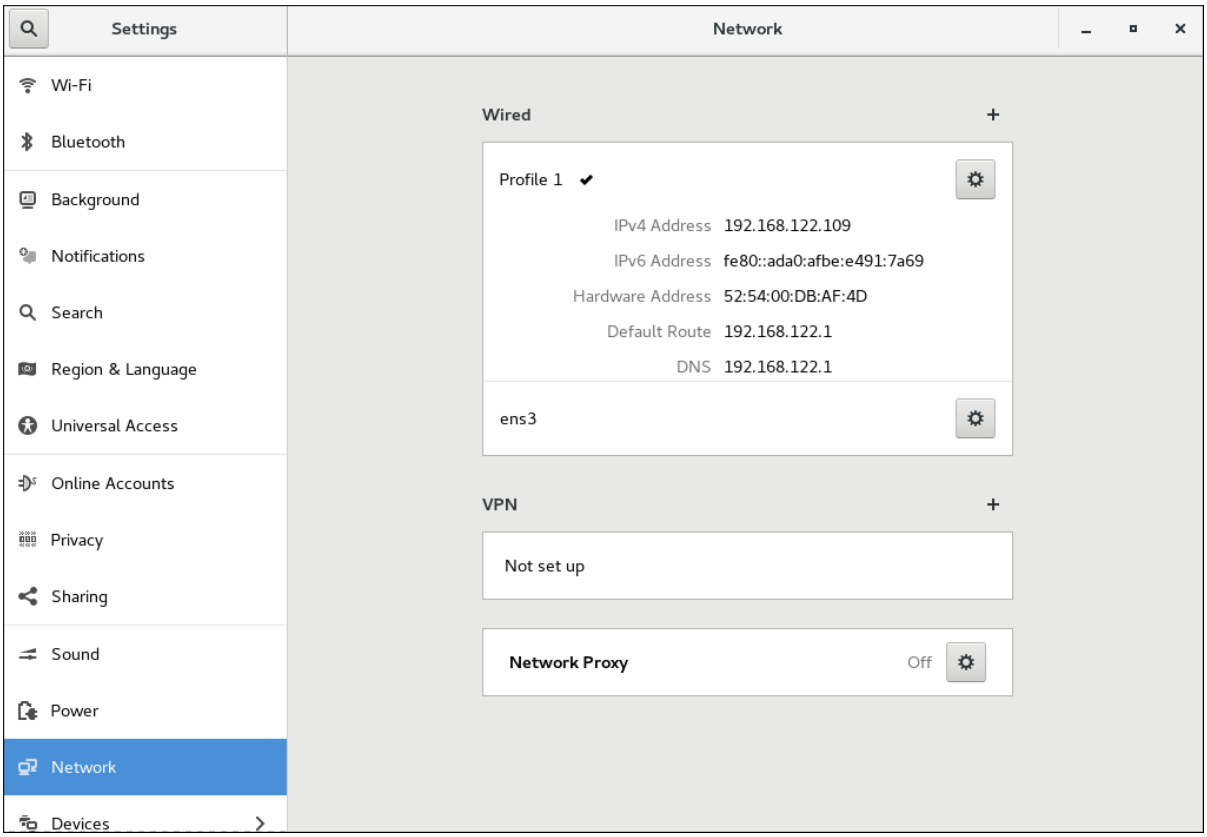
您可以使用 GNOME 控制中心应用创建和配置网络连接。

使用 control-center 配置新连接

要使用 control-center 应用程序配置新的有线、无线的 vpn 连接，请按以下操作：

1. 按 Super 键进入 Activities Overview，键入 Settings，然后按 Enter 键。然后，选择左侧的网络选项卡。Network settings 工具会出现在右侧菜单中：

图 3.6. 打开 Network Settings 窗口



2. 单击加号按钮来添加新连接。

配置：

- 有线连接，单击 Wired 条目旁边的加号按钮，再继续 第 3.4.6 节 “使用 GUI 配置连线（以太网）连接”。

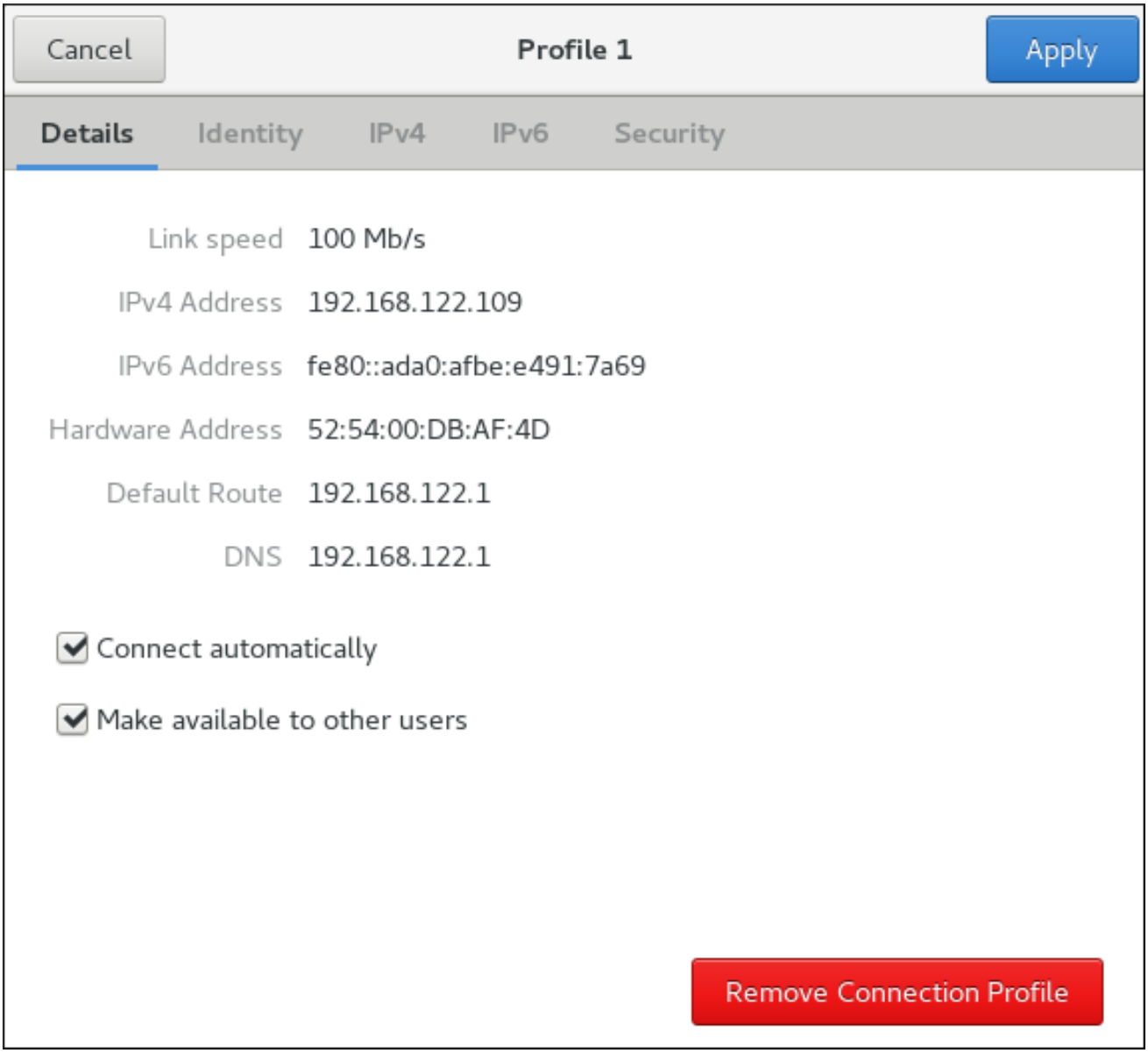
- VPN 连接，单击 VPN 条目旁边的加号按钮，然后继续 第 3.4.8.1 节 “使用 control-center 建立 VPN 连接”

对于 Wi-Fi 连接，点 Settings 菜单中的 Wi-fi 条目，然后继续 第 3.4.7 节 “使用 GUI 配置 Wi-Fi 连接”

使用 control-center 编辑现有连接

点击 Network settings 窗口中现有连接配置集的 gear wheel 图标将打开 Details 窗口，从中可以执行大部分网络配置任务，如 IP 寻址、DNS 和路由配置。

图 3.7. 使用网络连接详情窗口配置网络



[D]

对于您添加或配置的任何连接类型，您可以选择 NetworkManager 在可用时自动连接到该网络。为此，请选择 Connect 会在 NetworkManager 检测到可用时自动 连接到连接。如果您不希望 NetworkManager 自动连接，请清除复选框。如果复选框已清除，您必须在网络连接图标的菜单中手动选择该连接，使其连接。

要使连接可供其他用户使用，请选中 Make available to other users 复选框。

要在连接修改后应用更改，您可以点击连接窗口右上角的 Apply 按钮。

您可以通过点 Remove Connection Profile red 复选框来删除连接。

3.4.2.2. 使用 nm-connection-editor 配置新的和编辑现有连接

使用 nm-connection-editor GUI 应用程序，您可以使用 control-center 提供的附加功能来配置您想要的任何连接。此外，nm-connection-editor 应用并非由 GNOME 控制中心提供的功能，如配置绑定、网桥、VLAN 和组连接。

使用 nm-connection-editor 配置新连接

使用 nm-connection-editor 添加新连接类型：

流程

1. 在终端中输入 nm-connection-editor ：

```
~]$ nm-connection-editor
```

这时将显示 Network Connections 窗口。

2. 点击加号按钮选择一个连接类型：

图 3.8. 使用 nm-connection-editor 添加连接类型

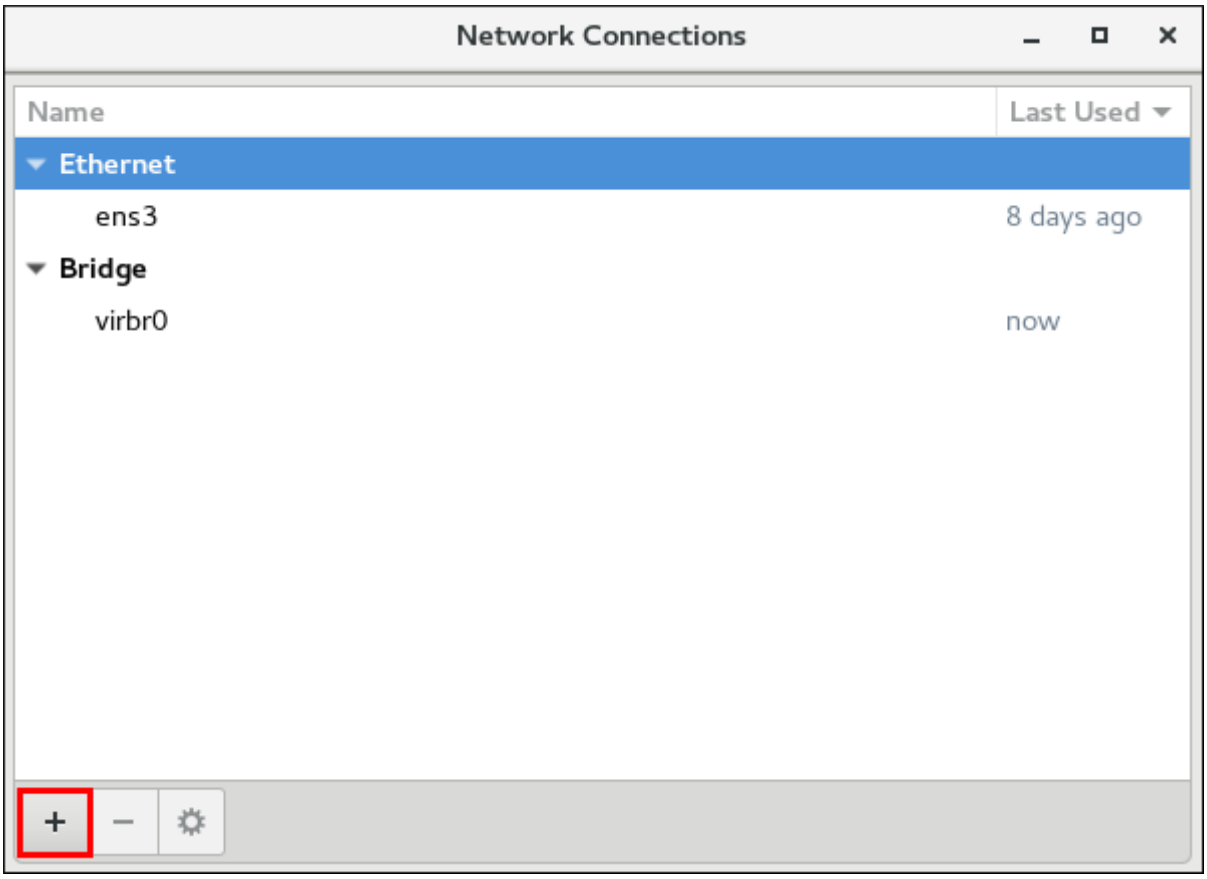


图 3.9. 使用 nm-connection-editor 选择连接类型

创建并配置：

- 绑定连接，单击 **Bond** 条目，再继续 第 7.8.1 节 “建立绑定连接”。
- 网桥连接，单击 **Bridge** 条目，再继续 第 9.4.1 节 “使用 GUI 建立网桥连接”；
- VLAN 连接，单击 **VLAN** 条目，再继续 第 10.5.1 节 “建立 VLAN 连接”；或者，
- 团队连接，单击 **Team** 条目，再继续 第 8.14 节 “使用 GUI 创建网络团队”。

使用 nm-connection-editor 编辑现有连接

有关现有连接类型，请点击网络连接对话框中的 gear wheel 图标，请参阅 “使用 nm-connection-editor 配置新连接”一节。

3.4.3. 使用 nm-connection-editor 的通用配置选项

如果您使用 nm-connection-editor 实用程序，请按照以下步骤操作最多连接类型（以太网、wifi、移动宽带、DSL）的五个常用配置选项：

流程

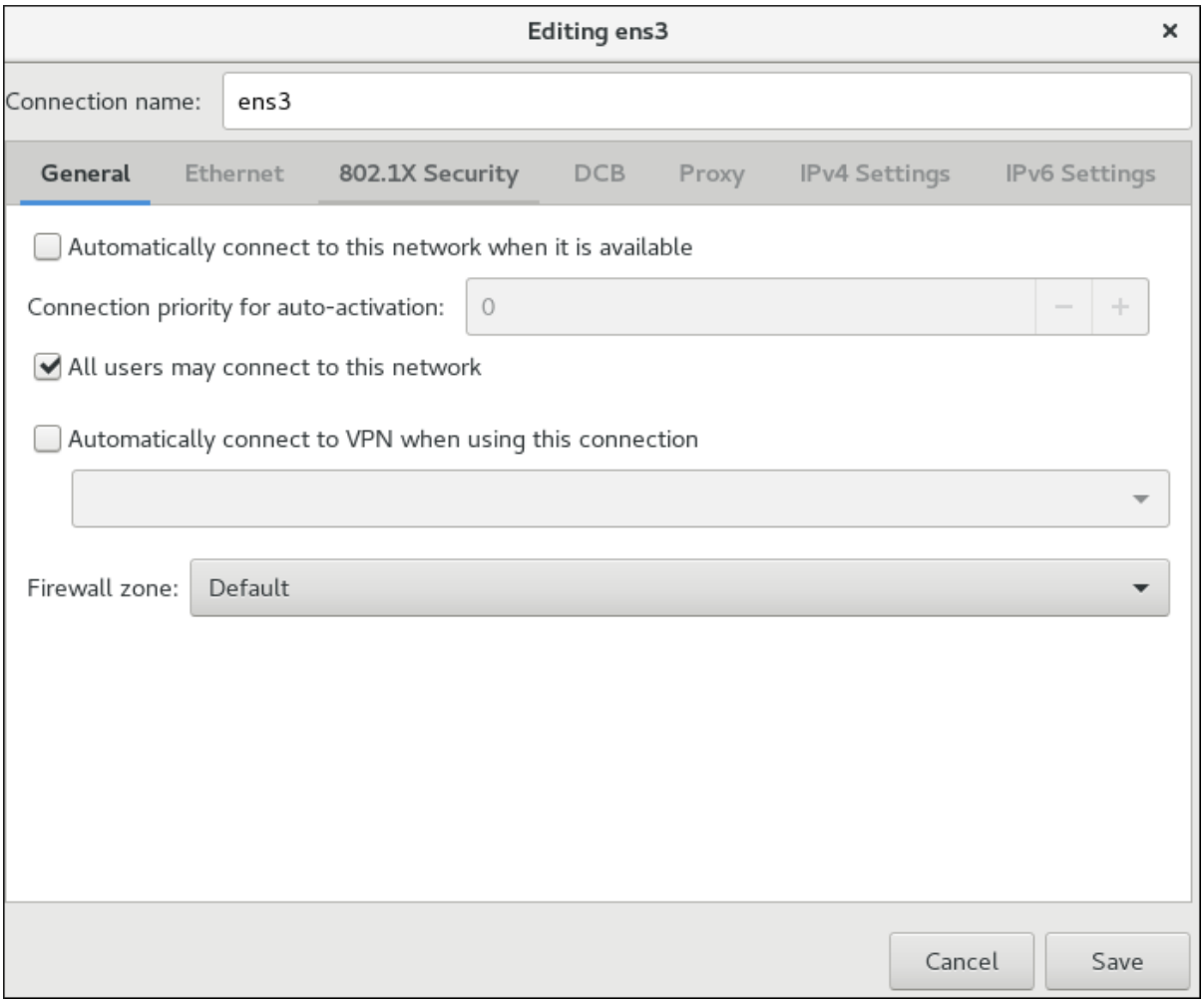
1. 在终端中输入 nm-connection-editor ：

```
~]$ nm-connection-editor
```

这时将显示 Network Connections 窗口。点击加号按钮选择连接类型或 gear wheel 图标以编辑现有连接。

2. 在 Editing 对话框中选择 General 选项卡：

图 3.10. nm-connection-editor 中的配置选项



- 连接名称 - 输入您的网络连接描述性名称。此名称用于在 Network 窗口的菜单中列出此连接。

- 自动激活的连接优先级 - 如果连接被设置为自动连接，则激活该数字（默认为0）。数值越高，表示优先级更高。

- 当这个网络可用时自动连接到这个网络 - 如果您希望 NetworkManager 在可用时自动连接到这个连接，请选择这个框。如需更多信息，请参阅 “使用 control-center 编辑现有连接”一节。

- 所有用户可以连接到此网络 - 选择此框可创建系统上所有用户可用的连接。更改此设置可能需要 root 特权。详情请查看 第 3.4.5 节 “使用 GUI 管理系统范围以及专用连接配置集”。

- 使用这个连接时自动连接到 VPN - 如果您希望 NetworkManager 在有可用时自动连接到 VPN 连接，请选择这个框。从下拉菜单中选择 VPN。

- firewall Zone - 从下拉菜单中选择防火墙区域。*有关防火墙区域的更多信息，请参阅Red Hat Enterprise Linux 7 安全指南*。

注意

对于 VPN 连接类型，只有三个配置选项可用：连接名称，所有用户都可以连接到此网络和 防火墙区。

3.4.4. 使用 GUI 自动连接到网络

对于您添加或配置的任何连接类型，您可以选择是否希望 NetworkManager 尝试在可用时自动连接到该网络。您可以使用以下方法之一：

- GNOME control-center 应用程序

- GNOME nm-connection-editor 应用程序

3.4.4.1. 使用 control-center 自动连接到网络

您可以使用 control-center 自动连接到网络：

流程

1. 按 Super 键进入 Activities Overview，键入 Settings，然后按 Enter 键。然后，选择左侧的网络选项卡。Network settings 工具会出现在右侧菜单中，请参阅 “使用 control-center 配置新连接”一节。

2. 从右侧菜单中选择网络接口。

3. 在右侧菜单点击连接配置集的 gear wheel 图标。这时将显示网络详细信息窗口。

4. 选择 Details 菜单条目，请参阅 “使用 control-center 编辑现有连接”一节。

5. 选择"连接"自动会导致 NetworkManager 每当 NetworkManager 检测到可用时自动连接到连接。如果您不希望 NetworkManager 自动连接，请清除复选框。如果复选框已清除，您必须在网络连接图标的菜单中手动选择该连接，使其连接。

3.4.4.2. 使用 nm-connection-editor 自动连接到网络

您还可以使用 GNOME nm-connection-editor 应用程序自动连接到网络。为此，请遵循 第 3.4.3 节 “使用 nm-connection-editor 的通用配置选项” 中取消的步骤，并在 General 选项卡中 选中 Automatically connect to this network。

3.4.5. 使用 GUI 管理系统范围以及专用连接配置集

NetworkManager *存储所有连接配置集。*配置文件是可应用于接口的设置的命名集合。NetworkManager *将这些连接配置集存储为系统范围使用（系统连接 ），以及所有用户连接配置集。对连接配置集的访问由 NetworkManager 存储的权限控制。有关连接设置 权限属性的更多信息，请参阅 nm-settings(5) man page。 您可以使用以下图形用户界面工具控制对连接配置集的访问：*

- *nm-connection-editor 应用程序*
- *control-center 应用程序*

3.4.5.1. 使用 nm-connection-editor 管理连接配置集的权限

要创建可供系统中所有用户使用的连接，请按照 第 3.4.3 节 “使用 nm-connection-editor 的通用配置选项” 中取消的步骤，并选中 General 标签页中的所有用户可以连接到这个网络复选框。

3.4.5.2. 使用 control-center 管理连接配置集的权限

要使连接可供其他用户使用，请按照 “使用 control-center 编辑现有连接”一节 中的步骤操作，然后在 GNOME control-center Network settings Details 窗口中选择" 可供其他用户 使用"复选框。

相反，清除 Make 可供其他用户使用复选框，使连接用户特定，而不是整个系统。

注意

根据系统的策略，您可能需要系统上的 root 权限，以更改连接是特定于用户还是系统范围。

NetworkManager 的默认策略是允许所有用户创建和修改系统范围的连接。启动时可用的配置文件不能是私有的，因为它们在用户登录之前将无法看到。例如，如果用户创建了连接配置集 user-em2，并选择了 Connect Automatically 复选框，但是其他未选择的用户可使用 Make，则在引导时无法使用连接。

要限制连接和网络，可以单独或组合使用两个选项：

- 清除 Make available to other users 复选框，该复选框将连接更改为仅可由用户进行更改才可修改和使用。
- 使用 polkit 框架根据每个用户限制常规网络操作的权限。

这两个选项的组合提供了精细的安全性和对网络的控制。有关 polkit 的更多信息，请参阅 polkit 手册页。

请注意，VPN 连接总是被创建为私有用户，因为它们比 Wi-Fi 或者以太网连接多。

3.4.6. 使用 GUI 配置连线（以太网）连接

您可以通过两种方式使用 GUI 配置有线连接：

- control-center 应用程序
- nm-connection-editor 应用程序

3.4.6.1. 使用 control-center 配置 Wired Connection

流程

1. 按 Super 键进入 Activities Overview，键入 Settings，然后按 Enter 键。然后，选择左侧的 Network 菜单条目，并显示 网络设置 工具，请参阅 “使用 control-center 配置新连接”一节。

2. 如果还没有突出显示，请选择 Wired 网络接口。

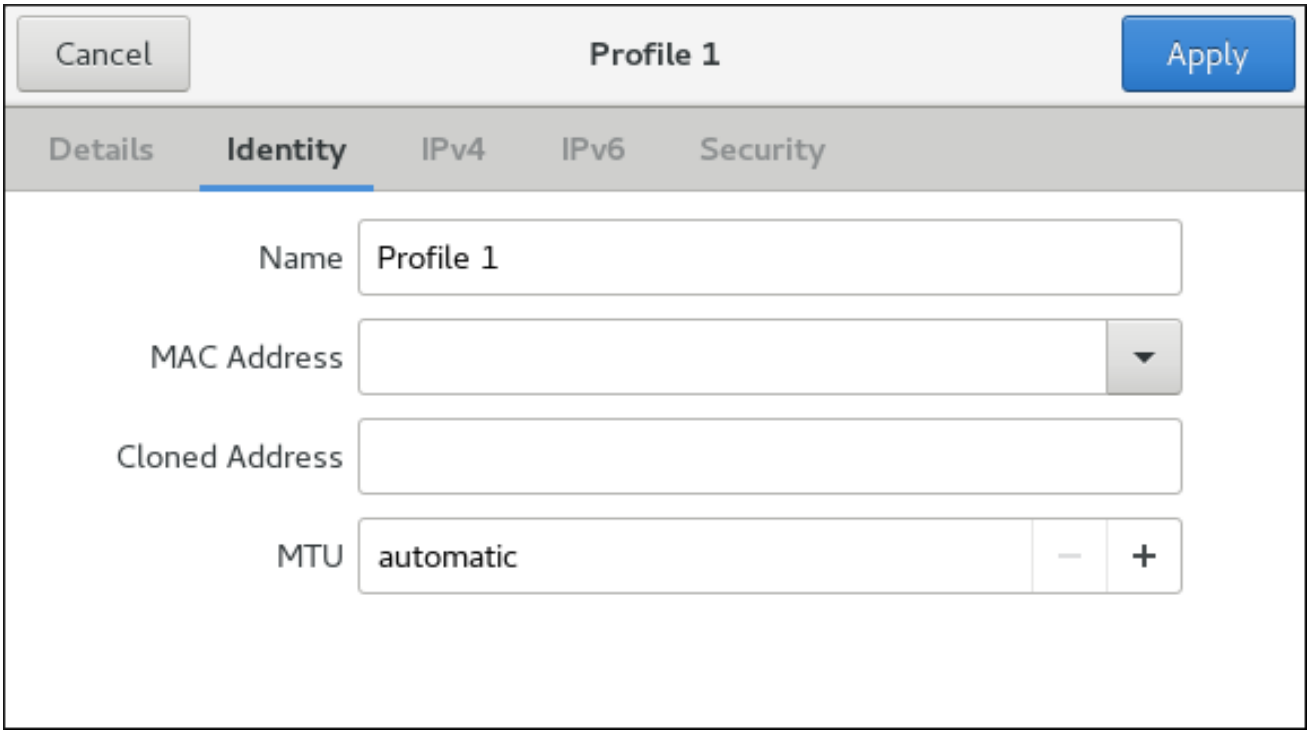
   默认情况下，系统会创建和配置一个名为 Wired 的有线连接配置集。配置文件是可应用于接口的设置的命名集合。可以为接口创建多个配置集并根据需要应用。默认配置集无法删除，但可以更改其设置。

3. 点击 gear wheel 图标编辑默认 Wired 配置集。

基本配置选项

您可以在 Wired 对话框中看到以下配置设置，方法是选择 Identity 菜单条目：

图 3.11. Wired Connection 的基本配置选项



- Name - 输入您的网络连接描述性名称。此名称将用于在 Network 窗口的菜单中列出此连接。

- MAC Address - 选择此配置集必须应用到的接口的 MAC 地址。

- 克隆地址 - 如果需要，请输入要使用的不同 MAC 地址。

- MTU - 如果需要，请输入要使用的特定最大传输单元 (MTU)。MTU 值表示链路层传输的最大数据包的字节大小。这个值默认为 1500，通常不需要指定或更改。

生成更多连线配置

您可以在编辑对话框中进一步配置现有连接。

配置：

- 连接的 IPv4 设置，单击 IPv4 菜单条目，然后继续 第 5.4 节 “配置 IPv4 设置”

  或者

- 连接的 IPv6 设置，单击 IPv6 菜单条目，再继续 第 5.5 节 “配置 IPv6 设置”。

- 基于端口的网络访问控制(PNAC)，单击 802.1X 安全菜单条目，再继续 第 5.2 节 “配置 802.1X 安全性”；

保存您的新（或修改的）连线连接

编辑完有线连接后，单击 Apply 按钮以保存自定义配置。如果在编辑期间使用配置集，请重启连接以使 NetworkManager 应用更改。如果配置集是 OFF，将其设置为 ON，或者在网络连接图标的菜单中选择它。有关使用新连接或更改的连接的详情，请查看 第 3.4.1 节 “使用 control-center GUI 连接到网络”。

创建新 Wired 连接

要创建新的有线连接配置集，请点击加号按钮，请参阅 “使用 control-center 配置新连接”一节。

当您点击加号按钮添加新连接时，NetworkManager 会为那个连接创建新配置文件，然后打开同一个对话框来编辑现有连接，请参阅 “使用 control-center 编辑现有连接”一节。这两个对话框之间的区别在于现有连接配置集有一个详情菜单条目。

3.4.6.2. 使用 nm-connection-editor 配置 Wired Connection

nm-connection-editor GUI 应用程序提供的配置选项比 control-center GUI 应用程序更多。使用 nm-connection-editor 配置有线连接：

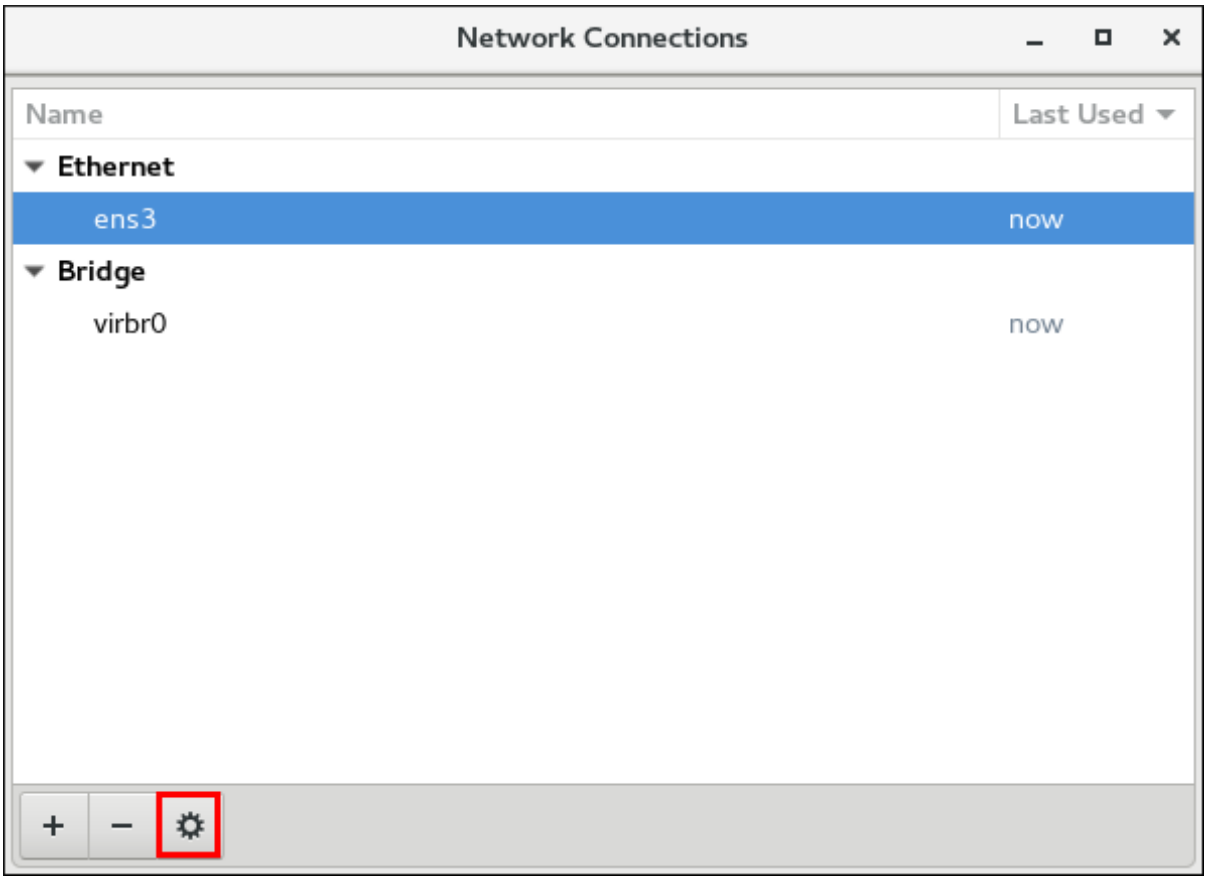
1. 在终端中输入 nm-connection-editor。

```
~]$ nm-connection-editor
```

这时将显示 Network Connections 窗口。

2. 选择您要编辑的以太网连接并点击 gear wheel 图标：

图 3.12. 编辑有线连接



此时将显示编辑对话框。

- 要自动连接到网络并限制连接，请点击 General 选项卡，请参阅第 3.4.3 节 “使用 nm-connection-editor 的通用配置选项”。
- 要配置网络设置，请点击 以太网 选项卡，请参阅 “使用 nm-connection-editor 配置 802.3 链路设置”一节。
- 要为有线连接配置 802.1X 安全性，点 802.1X 安全 选项卡，请参阅第 5.2.4 节 “使用 nm-connection-editor 为 Wired 配置 802.1X 安全性”。
- 要配置 IPV4 设置，请点击 IPV4 Settings 选项卡，请参阅 “使用 nm-connection-editor 设置 IPV4 的方法”一节。
- 要配置 IPV6 设置，请点击 IPV6 Settings 选项卡，请参阅第 5.5 节 “配置 IPv6 设置”。

3.4.7. 使用 GUI 配置 Wi-Fi 连接

这部分论述了如何使用 NetworkManager 配置 Wi-Fi （也称为无线或者 802.11a/b/g/n）到接入点的连接。Access Point 是一个允许无线设备连接到网络的设备。

要配置移动宽带（如 3G）连接，请参阅第 3.4.9 节 “使用 GUI 配置移动带宽连接”。

快速连接到可用接入点

流程

1. 点击网络连接图标激活网络连接图标的菜单，请参阅第 3.4.1 节 “使用 control-center GUI 连接到网络”。

2. 在 Wi-Fi 网络列表中，找到接入点的 Service Set Identifier (SSID)。

3. 单击网络的 SSID。挂锁符号表示接入点需要身份验证。如果访问点受到保护，对话框会提示您输入身份验证密钥或密码。

   NetworkManager 尝试自动检测接入点使用的安全性类型。如果存在多种可能性，NetworkManager 会猜测安全类型，并将其显示在 Wi-Fi 安全下拉菜单中。

   - 对于 WPA-PSK 安全（带有密码的 WPA）而言，无需选择。
   - 对于 WPA Enterprise(802.1X)，您必须特别选择安全性，因为无法自动检测安全性。

   请注意，如果您不确定，请依次尝试分别连接到每种类型。

4. 在"密码"字段中输入密钥或密码短语。某些密码类型（如 40 位 WEP 或 128 位 WPA 密钥）无效，除非它们具有必要长度。连接按钮将保持不活动状态，直到您输入了所选安全类型所需的长度的密钥。要了解更多有关无线安全性的信息，请参阅 第 5.2 节 “配置 802.1X 安全性”。

如果 NetworkManager 成功连接到访问点，网络连接图标将更改为无线连接信号强度的图形指示器。

您还可以编辑其中一个自动创建的访问点连接的设置，就像您自己添加一样。Network 窗口的 Wi-Fi 页面有一个 History 按钮。单击它可显示您曾尝试连接到的所有连接的列表。请查看 “编辑现有 Wi-Fi 连接”一节

连接到 Hidden Wi-Fi 网络

所有接入点都有一个 Service Set Identifier(SSID)来识别它们。但是，接入点可以被配置为不广播其 SSID，在这种情况下，它会被隐藏，且不会出现在 NetworkManager 的可用网络列表中。只要您知道其 SSID、验证方法和 secret，您仍然可以连接到无线访问点来隐藏其 SSID。连接到隐藏的无线网络：

流程

1. 按 Super 键进入 Activities Overview，键入 Settings，然后按 Enter 键。然后，选择左侧的 Wi-Fi 菜单条目。

2. 选择 Connect to Hidden Network。有两个选项：

   - 如果您在以下前连接到隐藏网络：

     1. 使用连接下拉菜单选择网络。
     2. 点 连接。

   - 如果没有，请按以下方法操作：

     1. 连接下拉列表保留为 新建。
     2. 输入隐藏网络的 SSID。
     3. 选择其 Wi-Fi 安全方法。
     4. 输入正确的身份验证 secret。
     5. 点 连接。

有关无线安全设置的详情请参考 第 5.2 节 “配置 802.1X 安全性”。

配置新的 Wi-Fi 连接

流程

1. 选择 Settings 的 Wi-Fi 菜单条目。

2. 点击您要连接到的 Wi-Fi 连接名称（默认情况下，与 SSID 相同）。

   - 如果 SSID 没有范围，请参阅 “连接到 Hidden Wi-Fi 网络”一节。

   - 如果 SSID 的范围不同，点右侧菜单中的 Wi-Fi 连接配置集。padlock 符号表示需要密钥或密码。如果需要，输入验证详情。

编辑现有 Wi-Fi 连接

您可以编辑过去您尝试或成功连接的现有连接。

流程

1. 按 Super 键进入 Activities Overview，键入 Settings 并按 Enter。

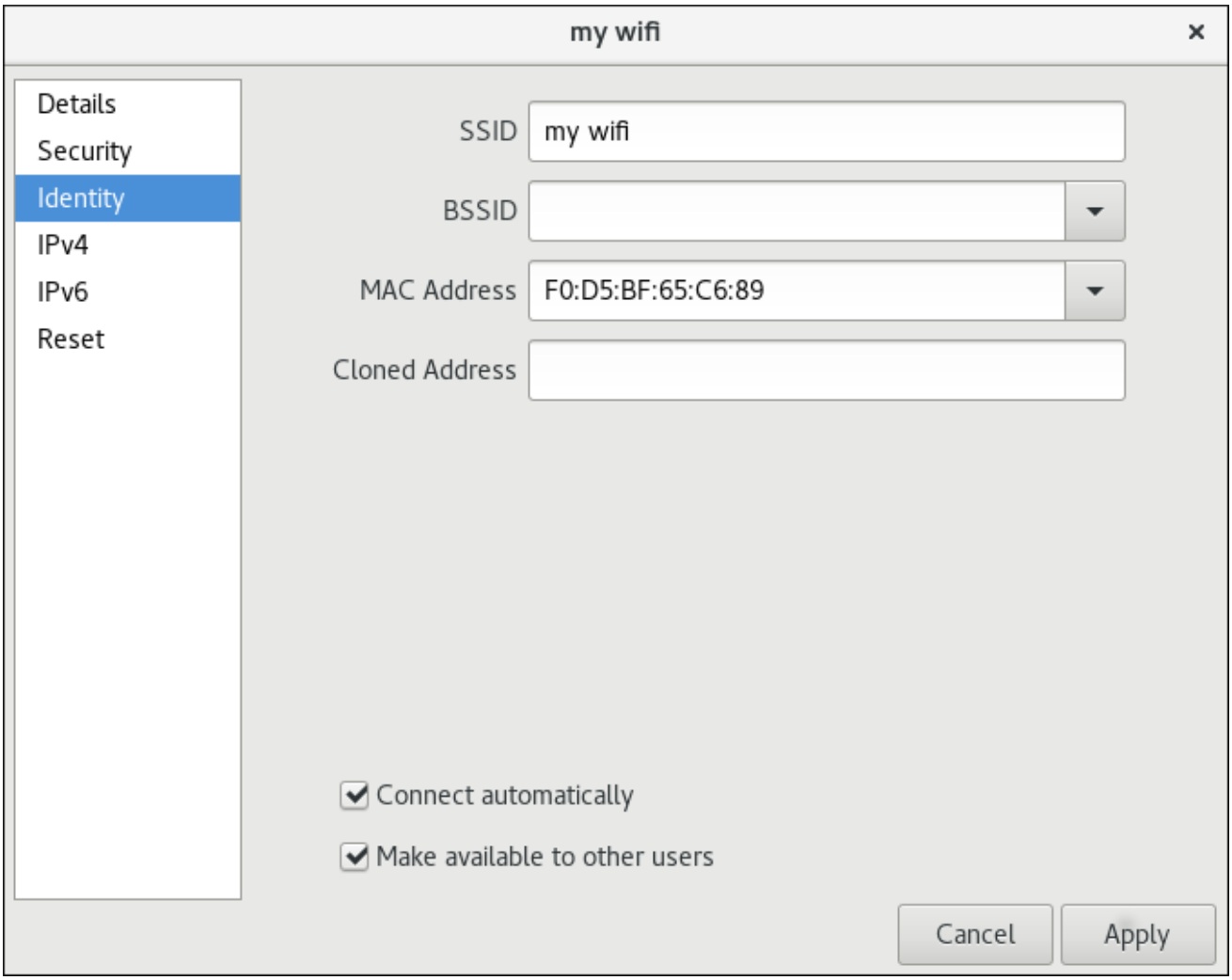
2. 从左侧的菜单条目中选择 Wi-Fi。

3. 选择您要编辑的 Wi-Fi 连接名称右侧的 gear wheel 图标，并会出现编辑连接对话框。请注意，如果网络当前不在范围内，请单击 History 以显示旧的连接。Details 窗口显示连接详情。

Wi-Fi 连接的基本配置选项

要编辑 Wi-Fi 连接的设置，请在编辑连接对话框中选择 Identity。可用的设置如下：

图 3.13. Wi-Fi 连接的基本配置选项



SSID

接入点(AP) 的服务集合标识符 (SSID)。

BSSID

基本服务集合标识符 (BSSID) 是您在基础架构模式中连接的特定无线访问点的 MAC 地址（也称为硬件地址 ）。默认情况下，此字段为空白，您可以通过 SSID 连接到无线访问点，而无需指定 BSSID。如果指定了 BSSID，它将强制系统只关联到特定的接入点。

对于临时网络，在创建 ad-hoc 网络时，BSSID 由 mac80211 子系统随机生成。NetworkManager不显示它

MAC 地址

选择要使用的 Wi-Fi 接口的 MAC 地址（也称为硬件地址 ）。

单个系统可以连接一个或多个无线网络适配器。因此，MAC 地址字段允许您将特定的无线适配器与特定连接 （或连接）关联。

克隆的地址

用于代替实际硬件地址的克隆 MAC 地址。保留空白，除非需要。

以下设置适用于大多数连接类型：

- "自动连接 " - 如果您希望 NetworkManager 在可用时自动连接到这个连接，请选择这个框。如需更多信息，请参阅 “使用 control-center 编辑现有连接”一节。

- 提供给其他用户 - 选择此框可创建可供系统上的所有用户使用的连接。更改此设置可能需要 root 特权。详情请查看 第 3.4.5 节 “使用 GUI 管理系统范围以及专用连接配置集”。

进行进一步的 Wi-Fi 配置

您可以在编辑对话框中进一步配置现有连接。

配置：

- 无线连接的安全身份验证，点 Security 并继续 第 5.2 节 “配置 802.1X 安全性”。

- 连接的 IPv4 设置，单击 IPv4 并继续 第 5.4 节 “配置 IPv4 设置”

  或者

- 连接的 IPv6 设置，单击 IPv6 并继续 第 5.5 节 “配置 IPv6 设置”。

保存您的新（或修改的）连接

编辑完无线连接后，点 Apply 按钮保存您的配置。如果配置正确，您可以通过从网络连接图标的菜单中选择它来连接到您修改的连接。有关选择和连接到网络的详情，请查看 第 3.4.1 节 “使用 control-center GUI 连接到网络”。

3.4.8. 使用 GUI 配置 VPN 连接

Libreswan 提供的 IPsec 是创建 VPN 的首选方法。Libreswan 是 VPN 的开源用户空间 IPsec 实现。使用命令行配置 IPsec VPN 信息包括在Red Hat Enterprise Linux 7 安全指南中。

3.4.8.1. 使用 control-center 建立 VPN 连接

Libreswan 提供的 IPsec 是在 Red Hat Enterprise Linux 7 中创建 VPN 的首选方法。如需更多信息，请参阅 第 3.4.8 节 “使用 GUI 配置 VPN 连接”。

以下描述的 GNOME 图形用户界面工具需要 NetworkManager-libreswan-gnome 软件包。要安装软件包，以 root 用户身份运行以下命令：

```
~]# yum install NetworkManager-libreswan-gnome
```

有关如何在Red Hat Enterprise Linux 中安装新软件包的更多信息，请参阅 Red Hat Enterprise Linux 系统管理员指南。

建立虚拟专用网络(VPN)可让您局域网(LAN)与其他远程 LAN 之间的通信。这可以通过在中间网络（如互联网）中设置隧道来实现。设置的 VPN 隧道通常使用身份验证和加密。在使用安全隧道成功建立 VPN 连接后，VPN 路由器或网关会对您传输的数据包执行以下操作：

1. 它为路由和验证目的添加了身份验证标头；
2. 它加密数据包数据；以及，
3. 它根据封装安全负载(ESP)协议将数据包含在数据包中，其中包括解密和处理指令。

接收 VPN 路由器会剥离标头信息，解密数据，并将其路由到预期的目标（工作站或其他节点在网络上）。使用网络连接时，本地网络上的接收节点会接收已解密并准备好处理的数据包。因此，网络到网络 VPN 连接中的加密和解密过程对客户端透明。

由于 VPN 使用了多个身份验证和加密层，因此 VPN 是连接多个远程节点以充当统一 Intranet 的安全有效方法。

添加新 IPsec VPN 连接

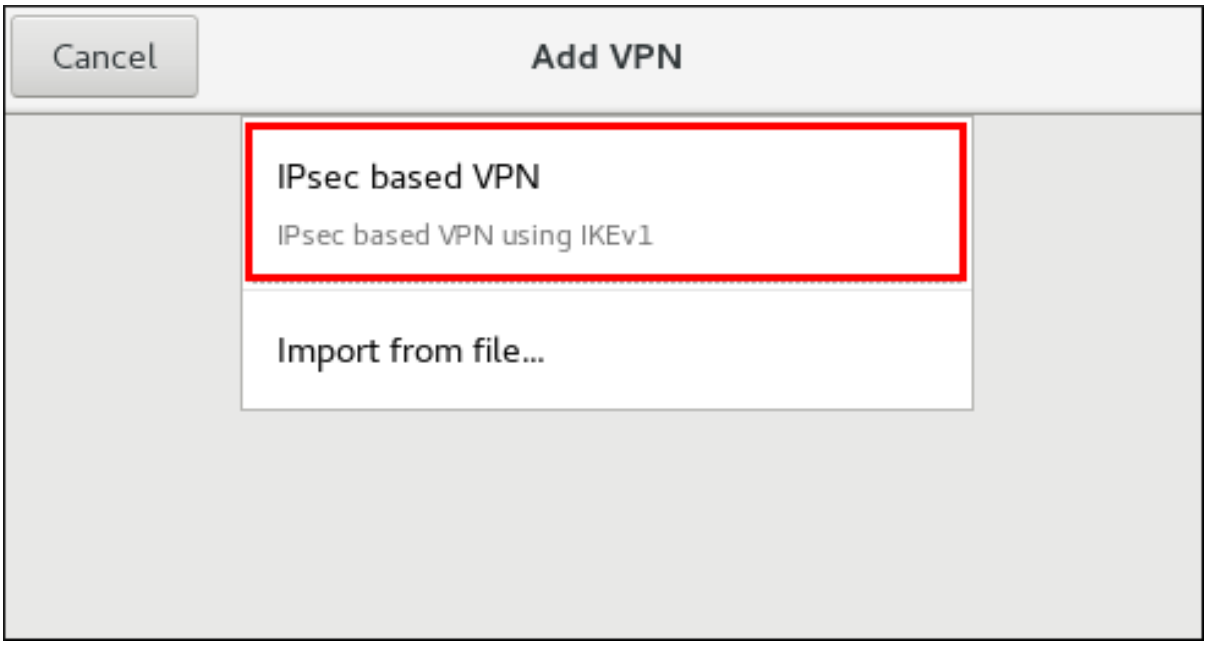
流程

1. 按 Super 键进入 Activities Overview，键入 Settings 并按 Enter。然后，选择 Network 菜单条目，并显示网络设置工具，请参阅 “使用 control-center 配置新连接”一节。

2. 点 VPN 条目中的加号按钮。

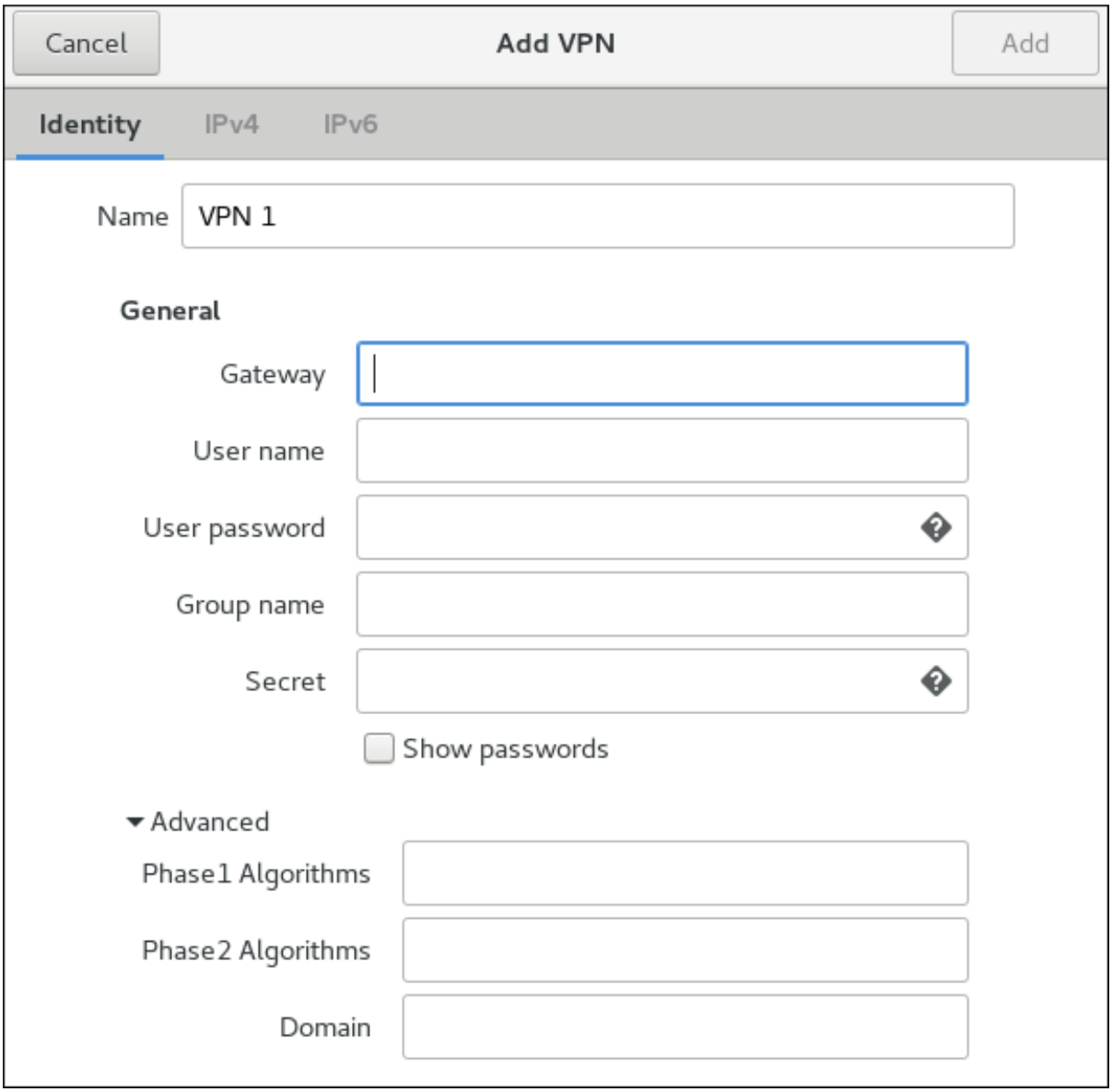
3. 此时会出现 Add VPN 窗口。 对于手动配置，请选择 基于 IPsec 的 VPN。

图 3.14. 在 IPsec 模式上配置 VPN



4. 在 Identity configuration 表单中，您可以在 General 和 Advanced 部分中指定字段：

图 3.15. General 和 Advanced 部分



- *在 General 部分，您可以指定：*

gateway

远程 VPN 网关的名称或 IP 地址。

用户名

如果需要，输入与 VPN 用户身份关联的用户名进行身份验证。

用户密码

如果需要，输入与 VPN 用户身份关联的密码进行验证。

组名称

在远程网关中配置的 VPN 组的名称。如果为空，则使用 IKEv1 主模式，而不是默认的 Aggressive 模式。

Secret

它是预共享的密钥，用于在用户身份验证之前初始化加密。如果需要，请输入与组名称关联的密码。

- *以下配置设置位于 Advanced 部分：*

phase1 算法

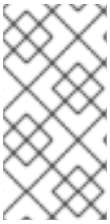
如果需要，输入用来验证和设置加密频道的算法。

phase2 算法

如果需要，请输入用于 IPsec 协商的算法。

域

如果需要，输入域名。



注意

在不使用 NetworkManager 的情况下配置 IPsec VPN，请参阅 第 3.4.8 节 “使用 GUI 配置 VPN 连接”。

编辑现有 VPN 连接

流程

1. *按 Super 键进入 Activities Overview，键入 Settings 并按 Enter。然后，选择 Network 菜单条目，并显示网络设置工具，请参阅 “使用 control-center 配置新连接”一节。*

2. *选择您要编辑的 VPN 连接并点击 gear wheel 图标并编辑 General 和 Advanced 部分，请参阅 第 3.4.8.1 节 “使用 control-center 建立 VPN 连接”。*

保存您的新（或修改）连接并创建进一步配置

编辑完新 VPN 连接后，点 Save 按钮保存自定义配置。如果在编辑期间使用配置集，请重启连接以使 NetworkManager 应用更改。如果配置集是 OFF，将其设置为 ON，或者在网络连接图标的菜单中选择它。有关使用新连接或更改的连接的详情，请查看 第 3.4.1 节 “使用 control-center GUI 连接到网络”。

您可以通过在网络窗口中选择现有连接并单击 Configure 来返回到编辑对话框来进一步配置现有连接。

然后，配置：

- 连接的 IPv4 设置，单击 IPv4 Settings 选项卡，再继续 第 5.4 节 “配置 IPv4 设置”。

3.4.8.2. 使用 nm-connection-editor 配置 VPN 连接

您还可以使用 nm-connection-editor 添加和配置 VPN 连接。要做到这一点，请按以下方式操作：

流程

1. 在终端中输入 nm-connection-editor。这时将显示 Network Connections 窗口，请参阅 第 3.4.3 节 “使用 nm-connection-editor 的通用配置选项”。
2. 单击加号按钮。此时会打开 Choose a Connection Type 菜单。
3. 从 VPN 菜单条目中选择，基于 IPsec 的 VPN 选项。
4. 点击 Create 打开 Editing 对话框，然后继续 “添加新 IPsec VPN 连接”一节 编辑 General 和 Advanced 部分。

3.4.9. 使用 GUI 配置移动带宽连接

您可以使用 NetworkManager 的移动宽带连接功能连接到以下 2G 和 3G 服务：

- 2G - GPRS（通用封包广播服务 ）、EDGE（GSM Evolution 的增强数据率 ）或 CDMA（代码划分多访问）。
- 3G - UMTS（通用移动电信系统 ）、HSPA（高分组访问 ）或 EVDO（仅 EVolution 数据）。

为了创建连接，您的计算机必须具有移动宽带设备(modem)，系统已发现并识别此设备。此类设备可以内建在您的计算机中（如许多笔记本电脑和笔记本电脑中），也可以作为内部或外部硬件单独提供。示

例包括 PC 卡、USB Modem 或 Dongle、手机或手机，能够充当调制解调器。

3.4.9.1. 使用 nm-connection-editor 配置移动带宽连接

您可以使用 GNOME nm-connection-editor 配置移动宽带连接。

添加新移动带宽连接

流程

1. 在终端中输入 nm-connection-editor。这时将显示 Network Connections 窗口，请参阅 第 3.4.3 节 “使用 nm-connection-editor 的通用配置选项”。
2. 单击加号按钮。此时会打开 Choose a Connection Type 菜单。
3. 选择 Mobile Broadband 菜单条目。
4. 单击 Create 以打开 Set up a Mobile Broadband Connection 助理。
5. 在创建此移动宽带设备的连接下，选择与连接一起使用的 2G 或 3G 功能设备。如果下拉菜单不活跃，这表明系统无法检测到能够移动宽带的设备。在这种情况下，点击 Cancel，请确保您有一个支持移动宽带的设备被计算机附加和识别，然后重试此过程。单击 Continue 按钮。
6. 从列表中选择您的服务提供商所在的国家，然后单击 Continue 按钮。
7. 从列表中选择您的供应商或手动输入您的供应商。单击 Continue 按钮。
8. 从下拉菜单中选择您的付款计划，并确认 Access Point Name (APN)正确无误。单击 Continue 按钮。
9. 检查并确认设置，然后单击应用按钮。
10. 通过参考编辑移动宽带特定设置 “配置 Mobile Broadband 选项卡”一节

编辑现有移动带宽连接

流程

1. 在终端中输入 nm-connection-editor。这时将显示 Network Connections 窗口。
2. 选择 Mobile Broadband 选项卡。
3. 选择您要编辑的连接并点击 gear wheel 图标。如需更多信息，请参阅 第 3.4.3 节 “使用 nm-connection-editor 的通用配置选项”。
4. 通过参考编辑移动宽带特定设置 “配置 Mobile Broadband 选项卡”一节

配置 Mobile Broadband 选项卡

如果您已经使用助理添加了一个新的移动宽带连接（请参阅 “添加新移动带宽连接”一节 ），则可以编辑 Mobile Broadband 选项卡来禁用 home 网络不可用，分配网络 ID，或者指示 NetworkManager 在使用该连接时首选特定技术（如 3G 或 2G）。

数字

与基于 GSM 的移动宽带网络建立 PPP 连接的数字。这个字段可以在首次安装宽带设备时自动填充。通常您可以将此字段留空并输入 APN。

username

输入用于通过网络进行身份验证的用户名。有些提供商不提供用户名，或者在连接网络时接受任何用户名。

密码

输入用于通过网络进行身份验证的密码。有些提供程序不提供密码或接受任何密码。

APN

输入用于建立与基于 GSM 网络的连接的接入点名称 (APN)。为连接输入正确的 APN 非常重要，因为它通常会决定：

- 如何为用户的网络使用计费；
- 用户是可以访问互联网、内部网还是子网。

网络 ID

输入网络 ID 会导致 NetworkManager 强制设备只注册到特定网络。这可用于确保连接不会在无法直接控制漫游时轮转。

类型

any - 默认值保留 modem 以选择最快的网络。

3G(UMTS/HSPA) - 强制连接仅使用 3G 网络技术。

2G(GPRS/EDGE) - 强制连接仅使用 2G 网络技术。

首选 3G(UMTS/HSPA) - 首先尝试使用 3G 技术（如 HSPA 或 UMTS）进行连接，并且仅在失败时回退到 GPRS 或 EDGE。

首选 2G(GPRS/EDGE) - 首先尝试使用 2G 技术（如 GPRS 或 EDGE）进行连接，并且仅在出现故障时回退到 HSPA 或 UMTS。

如果 home 网络不可用，允许轮转

如果您希望 NetworkManager 终止连接，而不是从 home 网络转换到漫游网络，请取消选中此框，从而避免可能的漫游收费。如果选中此框，NetworkManager 会尝试通过从 home 网络转换到漫游网络来维持良好的连接，反之亦然。

PIN

如果您设备的 SIM（订阅者身份模块）被锁定为 PIN（个人识别号），请输入 PIN，以便 NetworkManager 可以解锁该设备。如果需要 PIN 才能将该设备用于任何目的，NetworkManager 必须解锁 SIM。

CDMA 和 EVDO 的选项比较少。他们没有 APN、网络 ID 或 Type 选项。

保存您的新（或修改）连接并创建进一步配置

编辑完移动宽带连接后，单击 Apply 按钮以保存自定义配置。如果在编辑期间使用配置集，请重启连接以使 NetworkManager 应用更改。如果配置集是 OFF，将其设置为 ON，或者在网络连接图标的菜单中选择它。有关使用新连接或更改的连接的详情，请查看 第 3.4.1 节 “使用 control-center GUI 连接到网络”。

您可以通过在网络连接窗口中选择现有连接并单击 编辑 对话框来进一步配置现有连接。

然后，配置：

- 连接的点对点设置，点击 PPP Settings 选项卡，然后进入 第 5.6 节 “配置 PPP（点对点）设置”；
- 连接的 IPv4 设置，单击 IPv4 Settings 选项卡，再继续 第 5.4 节 “配置 IPv4 设置” ；或者，
- 连接的 IPv6 设置，单击 IPv6 Settings 选项卡，再继续 第 5.5 节 “配置 IPv6 设置”。

3.4.10. 使用 GUI 配置 DSL 连接

本节适用于在主机内装配 DSL 卡，而不是专用使用者或 SOHO 安装的外部组合 DSL modem 路由器组合的安装。

3.4.10.1. 使用 nm-connection-editor 配置 DSL 连接

您可以使用 GNOME nm-connection-editor 配置 DSL 连接。

添加新 DSL 连接

流程

1. 在终端中输入 nm-connection-editor。这时将显示 Network Connections 窗口，请参阅 第 3.4.3 节 “使用 nm-connection-editor 的通用配置选项”。

2. 单击加号按钮。

3. 此时将显示 Choose a Connection Type 列表。

4. 选择 DSL，然后按"创建 "按钮。

5. 此时将显示 Editing DSL Connection 1 窗口。

编辑现有 DSL 连接

流程

1. 在终端中输入 nm-connection-editor。这时将显示 Network Connections 窗口。

2. 选择您要编辑的连接并点击 gear wheel 图标。如需更多信息，请参阅 第 3.4.3 节 “使用 nm-connection-editor 的通用配置选项”。

配置 DSL 选项卡

username

输入用于与服务提供商进行身份验证的用户名。

服务

除非由服务提供商另有指示，否则留空。

密码

输入服务提供商提供的密码。

保存您的新（或修改）连接并创建进一步配置

编辑完 DSL 连接后，单击 Apply 按钮以保存自定义配置。如果在编辑期间使用配置集，请重启连接以使 NetworkManager 应用更改。如果配置集是 OFF，将其设置为 ON，或者在网络连接图标的菜单中选择它。有关使用新连接或更改的连接的详情，请查看 第 3.4.1 节 “使用 control-center GUI 连接到网络”。

您可以通过在网络连接窗口中选择现有连接并单击 编辑 对话框来进一步配置现有连接。

配置：

- MAC 地址和 MTU 设置，单击 Wired 选项卡，再继续 “基本配置选项 ”一节。
- 连接的点对点设置，点击 PPP Settings 选项卡，然后进入 第 5.6 节 “配置 PPP（点对点）设置”。
- 连接的 IPv4 设置，单击 IPv4 Settings 选项卡，再继续 第 5.4 节 “配置 IPv4 设置”。

3.5. 使用 IFCFG 文件配置 IP 网络

作为系统管理员，您可以通过编辑 ifcfg 文件手动配置网络接口。

接口配置（ifcfg）文件可控制不同网络设备的软件接口。当系统引导时，它使用这些文件来决定启动哪些界面以及如何进行配置。这些文件通常命名为 ifcfg-name ，后缀名称指的是配置文件控制的设备的名称。按照惯例，ifcfg 文件的后缀与配置文件中 DEVICE 指令提供的字符串相同。

使用 ifcfg 文件配置带有静态网络设置的接口

例如，若要使用 ifcfg 文件配置带有静态网络设置的接口，请为名为 enp1s0 的接口，在 /etc/sysconfig/network -scripts/ 目录中创建一个名为 ifcfg- enp1s0 的文件，该文件包含：

- 对于 IPv4 配置

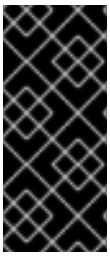
```
DEVICE=enp1s0
BOOTPROTO=none
ONBOOT=yes
PREFIX=24
IPADDR=10.0.1.27
```

- 对于 IPv6 配置

```
DEVICE=enp1s0
BOOTPROTO=none
ONBOOT=yes
IPV6INIT=yes
IPV6ADDR=2001:db8::2/48
```

您不需要指定网络或广播地址，因为这由 ipcalc 自动计算。

更多 IPv6 ifcfg 配置选项请查看 nm-settings-ifcfg-rh(5) man page。



重要

在 Red Hat Enterprise Linux 7 中，网络接口的命名规则已更改，如 第 11 章 一致的网络设备命名 中所述。使用HWADDR 指令指定硬件或 MAC 地址可能会影响设备命名过程。

使用 ifcfg 文件配置带有动态网络设置的接口

使用 ifcfg 文件配置名为 em1 的接口：

1. 在 /etc/sysconfig/network-scripts/ 目录中创建一个名为 ifcfg-em1 的文件，其中包含：

```
DEVICE=em1
BOOTPROTO=dhcp
ONBOOT=yes
```

2. 要将接口配置为向 DHCP 服务器发送其他主机名，请在 ifcfg 文件中添加以下行：

```
DHCP_HOSTNAME=hostname
```

要将接口配置为将不同的完全限定域名(FQDN)发送到 DHCP 服务器，请在 ifcfg 文件中添加以下行：

```
DHCP_FQDN=fully.qualified.domain.name
```

注意

在给定的 ifcfg 文件中只能使用一个指令（DHCP_HOSTNAME 或 DHCP_FQDN）。如果同时指定了 DHCP_HOSTNAME 和 DHCP_FQDN，则仅使用后者。

3. 要将接口配置为使用特定的 DNS 服务器，请在 ifcfg 文件中添加以下行：

```
PEERDNS=no
DNS1=ip-address
DNS2=ip-address
```

其中 ip-address 是 DNS 服务器的地址。这会导致网络服务使用指定的 DNS 服务器更新 /etc/resolv.conf。只需要一个 DNS 服务器地址，另一个是可选的。

4. 要在 ifcfg 文件中配置静态路由，请参阅 第 4.5 节 “在 ifcfg 文件中配置静态路由”。

默认情况下，当配置集被设置为通过在接口配置文件中将 BOOTPROTO 设置为 dhcp 来自动获取地址时，NetworkManager 会调用 DHCP 客户端, dhclient。如果需要 DHCP，则会在接口上为每个互联网协议 IPv4 和 IPv 6 启动 todhclient 实例。如果 NetworkManager 没有运行，或者未管理接口，则传统的网络服务将根据需要调用 fordhclient 实例。有关动态 IP 地址的详情，请参考 第 1.2 节 “静态与动态 IP 地址比较”。

5. 应用配置：

   a. 重新载入更新的连接文件：

   ```
   # nmcli connection reload
   ```

   b. 重新激活连接：

   ```
   # nmcli connection up connection_name
   ```

3.5.1. 使用 ifcfg 文件管理系统范围以及专用连接配置集

权限与 ifcfg 文件中的 USERS 指令对应。如果没有 USERS 指令，则网络配置文件将可供所有用户使用。例如，ifcfg 文件中的以下命令将仅使连接对列出的用户可用 ：

```
USERS="joe bob alice"
```

另外，您可以设置 USERCTL 指令来管理该设备：

- 如果您设置了 yes，则允许非root 用户控制这个设备。
- 如果没有，则 不允许非 root 用户控制这个设备。

3.6. 使用 IP 命令配置 IP 网络

作为系统管理员，您可以使用 ip 命令配置网络接口，但更改在重新引导后不会保留；重新引导后，您将丢失任何更改。

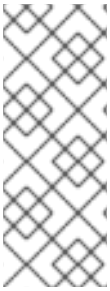
在上游软件包名称后，ip 实用程序的命令有时被称为 iproute2，它们记录在 man ip(8) 页面中。Red Hat Enterprise Linux 7 中的软件包名称为 iproute。如果需要，您可以通过检查其版本号来检查是否安装了 ip 工具程序，如下所示：

```
~]$ ip -V
ip utility, iproute2-ss130716
```

ip 命令可用于添加地址和路由到 NetworkManager 并行的接口，并在 nmcli、nmtui、control-center 和 D-Bus API 中识别它们。

关闭接口：

```
ip link set ifname down
```



注意

ip link set ifname 命令设置 IFF_UP 状态的网络接口，并从内核的范围中启用它。这与用于 initscripts 或 NetworkManager 设备的激活状态的 ifup ifname 命令不同。事实上，NetworkManager 始终会设置接口，即使它当前已断开连接。通过 nmcli 工具断开连接设备，不会删除 IFF_UP 标志。这样，NetworkManager 获取有关载波状态的通知。

请注意，ip 工具替换 ifconfig 工具，因为 net-tools 软件包（提供 ifconfig）不支持 InfiniBand 地址。

有关可用 OBJECT 的信息，请使用 ip help 命令。例如： ip link help 和 ip addr help。

注意

命令行中给定的 IP 命令在系统重启后不会保留。如果需要持久性时，请使用配置文件（ifcfg 文件）或向脚本添加命令。

nmtui 和 nmcli 示例后包含了为每项任务使用命令行和配置文件的示例，但解释了将其中一个图形用户界面用于 NetworkManager 之前，即 control-center 和 nm-connection-editor。

ip 工具可以用来为接口分配 IP 地址，格式如下：

```
ip addr [ add | del ] address dev ifname
```

使用 ip 命令分配静态地址

要为接口分配 IP 地址，请执行以下操作：

```
~]# ip address add 10.0.0.3/24 dev enp1s0
You can view the address assignment of a specific device:
~]# ip addr show dev enp1s0
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
qlen 1000
    link/ether f0:de:f1:7b:6e:5f brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.3/24 brd 10.0.0.255 scope global global enp1s0
```

```
    valid_lft 58682sec preferred_lft 58682sec
inet6 fe80::f2de:f1ff:fe7b:6e5f/64 scope link
    valid_lft forever preferred_lft forever
```

更多示例和命令选项可在 ip-address(8)手册页中找到。

使用 ip 命令配置多个地址

由于 ip 实用程序支持将多个地址分配到同一接口，因此不再需要使用别名接口方法将多个地址绑定到同一接口。分配地址的 ip 命令可以重复多次，以便能分配多个地址。例如：

```
~]# ip address add 192.168.2.223/24 dev enp1s0
~]# ip address add 192.168.4.223/24 dev enp1s0
~]# ip addr
3: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
qlen 1000
    link/ether 52:54:00:fb:77:9e brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.223/24 scope global enp1s0
    inet 192.168.4.223/24 scope global enp1s0
```

有关 ip 工具的命令的详情，请查看 ip(8)手册页。

注意

命令行中给定的 IP 命令在系统重启后不会保留。

3.7. 使用内核命令行配置 IP 网络

从接口连接到 iSCSI 目标上的 root 文件系统时，不会在安装的系统上配置网络设置。要解决这个问题：

1. 安装dracut 实用程序。有关使用dracut 的详情，请参考Red Hat Enterprise Linux 系统管理员指南

2. 使用内核命令行中的 ip 选项设置配置：

```
ip<client-IP-number>:[<server-id>]:<gateway-IP-number>:<netmask>:<client-
hostname>:<interface>:{dhcp|dhcp6|auto6|on|any|none|off}
```

- *DHCP - DHCP 配置*
- *dhpc6 - DHCP IPv6 配置*
- *auto6 - 自动 IPv6 配置*
- *在任何 - 内核中可用的任何协议（默认）*
- *none，off - 无自动配置，静态网络配置*

例如：

```
ip=192.168.180.120:192.168.180.100:192.168.180.1:255.255.255.0::enp1s0:off
```

3. *设置名称服务器配置：*

```
nameserver=srv1 [nameserver=srv2 [nameserver=srv3 [...]]]
```

Thedracut 实用程序设置网络连接并生成可复制到 /etc/sysconfig/network-scripts/ 文件中的新 ifcfg 文件。

3.8. 使用 IGMP 启用 IP 多播

利用 Internet 组管理协议(IGMP)，管理员可以管理网络、主机和路由器之间多播流量的路由和订阅。红帽企业 Linux 中的内核支持 IGMPv3。

要显示多播信息，请使用 ip maddr show 子命令，例如：

```
~]$ ip maddr show dev br0
8: br0
 inet  224.0.0.1
 inet6 ff02::1
 inet6 ff01::1
[output truncated]
```

或者，在 ip link show 命令输出中查找 MULTICAST 字符串，例如：

```
~]$ ip link show br0
8: br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode
DEFAULT qlen 1000
    link/ether 6c:0b:84:67:fe:63 brd ff:ff:ff:ff:ff:ff
```

禁用设备中的多播并检查 br0 设备中是否禁用多播：

```
~]# ip link set multicast off dev br0
~]$ ip link show br0
8: br0: <BROADCAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT qlen
1000
    link/ether 6c:0b:84:67:fe:63 brd ff:ff:ff:ff:ff:ff
```

缺少的 MULTICAST 字符串表示多播已被禁用。

在 br0 设备中启用多播并检查是否启用它：

```
~]# ip link set multicast on dev br0
~]$ ip link show br0
8: br0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode
DEFAULT qlen 1000
    link/ether 6c:0b:84:67:fe:63 brd ff:ff:ff:ff:ff:ff
```

如需更多信息，请参阅 ip Command Cheat Sheet for Red Hat Enterprise Linux 文章和 ip(8)man page。

要检查当前版本的 IGMP 和 IP 地址以获取多播，请查看 /proc/net/igmp 文件：

```
~]$ cat /proc/net/igmp
```

注意

默认情况下，firewalld 中不启用 IGMP。为区启用 IGMP：

```
~]# firewall-cmd --zone=zone-name --add-protocol=igmp
```

如需更多信息，请参阅《Red Hat Enterprise Linux 安全指南》中的使用 防火墙一章。

3.9. 其它资源

安装的文档

- ip(8) man page - 描述 ip 实用程序的命令语法。
- nmcli(1) man page - 描述 NetworkManager 的命令行工具.
- nmcli-examples(5) man page - 提供 nmcli 命令的示例.
- nm-settings(5) man page - 描述 NetworkManager 属性及其设置。
- nm-settings-ifcfg-rh(5) man page - 描述 ifcfg-rh 设置插件.

在线文档

Red Hat Enterprise Linux 7 安全指南

描述基于 IPsec 的 VPN 及其配置。描述使用 DNSSEC 进行身份验证的 DNS 查询。

RFC 1518 - 无类别域间路由(CIDR)

描述 CIDR Address Assignment 和 Aggregation 策略，包括可变长度子网。

RFC 1918 - 专用互联网的地址分配

描述保留给私有使用的 IPv4 地址范围。

RFC 3330 - 特殊使用 IPv4 地址

描述互联网编号分配机构(IANA)分配的全局和其他专用 IPv4 地址块。

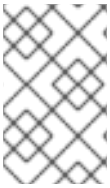
第 4 章 配置静态路由和默认网关

本章涵盖了静态路由和默认网关的配置。

4.1. 了解路由和网关简介

路由是一种允许系统查找到其他系统的网络路径的机制。路由通常由专用于路由的网络上的设备处理（尽管任何设备都可以配置为执行路由）。因此，通常不需要在 Red Hat Enterprise Linux 服务器或客户端上配置静态路由。例外包括必须通过加密 VPN 隧道的流量或出于成本或安全性而应采用特定路由的流量。主机的路由表将自动填充到直接连接网络的路由。“网络接口启动时检查路由”。若要访问远程网络或主机，系统被授予应发送到流量的网关地址。

当主机的接口由 DHCP 配置时，通常会分配一个网关地址，用于上游网络或互联网。此网关通常称为默认网关，因为如果系统不知道更好的路由（路由表中已存在），则使用该网关。网络管理员通常使用网络中的第一个或最后一个主机 IP 地址作为网关地址 ；例如 192.168.10.1 或 192.168.10.254。不要被代表网络本身的地址混淆；在本示例中，192.168.10.0 或子网的广播地址；在本例中为 192.168.10.255。默认网关通常为网络路由器。默认网关适用于不是以本地网络为目标的任何和所有流量，路由表中没有为其指定首选路由。



注意

为了扩展您的专业知识，您可能还对红帽系统管理 I(RH124) 培训课程感兴趣。

4.2. 使用 NMCLI 配置静态路由

要使用 nmcli 工具配置静态路由，请使用以下之一：

- nmcli 命令行
- nmcli 互动编辑器

例 4.1. 使用 nmcli 配置静态路由

使用命令行为现有以太网连接配置静态路由：

```
~]# nmcli connection modify enp1s0 +ipv4.routes "192.168.122.0/24 10.10.10.1"
```

```
~]# nmcli connection modify enp1s0 +ipv4.routes "192.168.122.0/24 10.10.10.1"
```

这会将 192.168.122.0/24 子网的流量定向到网关 10.10.10.1。

例 4.2. 使用 nmcli Editor 配置静态路由

使用互动编辑器为以太网连接配置静态路由：

```
~]$ nmcli con edit ens3
===| nmcli interactive connection editor |===

Editing existing '802-3-ethernet' connection: 'ens3'

Type 'help' or '?' for available commands.
Type 'describe [<setting>.<prop>]' for detailed property description.

You may edit the following settings: connection, 802-3-ethernet (ethernet), 802-1x, dcb, ipv4, ipv6,
tc, proxy
nmcli> set ipv4.routes 192.168.122.0/24 10.10.10.1
nmcli> save persistent
Connection 'ens3' (23f8b65a-8f3d-41a0-a525-e3bc93be83b8) successfully updated.
nmcli> quit
```

4.3. 使用 GUI 配置静态路由

要设置静态路由，请打开您要配置的连接的 IPv4 或 IPv6 设置窗口。有关如何做到这一点的说明，请参阅 第 3.4.1 节 “使用 control-center GUI 连接到网络”。

Routes

地址 - 输入远程网络、子网络或主机的 IP 地址。

子网掩码 - 上面输入的 IP 地址的子网掩码或前缀长度。

网关 - 上面输入的远程网络、子网络或主机的网关的 IP 地址。

指标 - 网络成本，赋予此路由的首选值。数值越低，优先级越高。

自动

当 Automatic 为 ON 时，会使用来自 RA 或 DHCP 的路由，但您也可以添加其他静态路由。OFF 时，只使用您定义的静态路由。

仅将此连接用于其网络上的资源

选择这个复选框以防止连接成为默认路由。典型的示例是连接是 VPN 隧道或向头办公室的租期线，并且您不希望任何互联网入站流量通过连接。选择这个选项意味着只有专门用于路由的流量才会通过连接自动获得，或者手动输入到连接上。

4.4. 使用 IP 命令配置静态路由

作为系统管理员，您可以使用 ip route 命令配置静态路由。

要显示 IP 路由表，请使用 ip route 命令。例如：

```
~]$ ip route
default via 192.168.122.1 dev ens9  proto static  metric 1024
192.168.122.0/24 dev ens9  proto kernel  scope link  src 192.168.122.107
192.168.122.0/24 dev enp1s0  proto kernel  scope link  src 192.168.122.126
```

ip route 命令的格式如下：

```
ip route [ add | del | change | append | replace ] destination-address
```

有关选项和格式的详情，请查看 ip-route(8)man page。

将静态路由添加到主机地址，换句话说到单个 IP 地址：

```
~]# ip route add 192.0.2.1 via 10.0.0.1 [dev interface]
```

其中 192.0.2.1 是带点十进制表示法的主机的 IP 地址， 100.0.1 是下一个跃点地址和 interface 是导致下一跃点的退出接口。

将静态路由添加到网络，换句话说是添加到代表一系列 IP 地址的 IP 地址中 ：

```
~]# ip route add 192.0.2.0/24 via 10.0.0.1 [dev interface]
```

其中 192.0.2.0 是目标网络的 IP 地址 （点十进制表示法），/24 是网络前缀。网络前缀是子网掩码中已启用的位数。这种网络地址斜杠网络前缀长度的格式有时称为无类别域间路由 (CIDR)表示法。

删除分配的静态路由：

```
~]# ip route del 192.0.2.1
```

您使用 ip route 对路由表所做的任何更改在系统重启后不会保留。要永久配置静态路由，您可以通过在 /etc/sysconfig/network-scripts/ 目录中为接口创建路由接口文件来配置这些路由。例如，enp1s0 接口的静态路由将存储在 /etc/sysconfig/network-scripts/route-enp1s0 文件中。在重启网络服务或接口前，您对路由接口文件进行的任何更改都不会生效。route-接口文件有两种格式 ：

- IP 命令参数，请参阅 “使用 IP 命令参数格式的静态路由”一节。

  和

- 网络/子网掩码指令，请参见 “使用网络/网络掩码指令格式的静态路由”一节。

有关 ip route 命令的详情，请查看 ip-route (8) man page。

4.5. 在 IFCFG 文件中配置静态路由

如果系统关机或重启，使用 ip 命令设置的静态路由会在命令提示符下丢失。要将静态路由配置为在系统重启后永久保留，必须将其放置在 /etc/sysconfig/network-scripts/ 目录中的每个接口配置文件中。文件名应当为 route-接口的格式。配置文件中有两种命令类型：

使用 IP 命令参数格式的静态路由

如果每个接口配置文件中需要（例如 /etc/sysconfig/network-scripts/route-enp1s0），请在第一行中定义指向默认网关的路由。只有在没有通过 DHCP 设置网关且没有在 /etc/sysconfig/network 文件中全局设置时才需要此项：

```
default via 192.168.1.1 dev interface
```

其中 192.168.1.1 是默认网关的 IP 地址。接口是连接到或可以访问默认网关的接口。dev 选项可以省略，它是可选的。请注意，此设置优先于 /etc/sysconfig/network 文件中的设置。

如果需要路由到远程网络，可以按照如下所示指定静态路由：每行都解析为单个路由：

```
10.10.10.0/24 via 192.168.1.1 [dev interface]
```

其中 10.10.10.0/24 是远程或目标网络的网络地址和前缀长度。地址 192.168.1.1 是导致远程网络的 IP 地址。最好使用下一个跃点地址，但退出接口的地址可以正常工作。“下一跃点表示链路的远程末尾”，如网关或路由器。dev 选项可用于指定退出接口接口，但这不是必需的。根据需要添加任意数量的静态路由。

以下是使用 ip 命令参数格式的路由接口文件示例。默认网关为 192.168.0.1，接口 enp1s0 以及租用的行或 WAN 连接位于 192.168.0.10。这两个静态路由到达 10.10.10.0/24 网络和 172.16.1.10/32 主机：

```
default via 192.168.0.1 dev enp1s0
10.10.10.0/24 via 192.168.0.10 dev enp1s0
172.16.1.10/32 via 192.168.0.10 dev enp1s0
```

在上面的示例中，进入本地 192.168.0.0/24 网络的数据包将被定向出连接到该网络的接口。进入 10.10.10.0/24 网络和 172.16.1.10/32 主机的数据包将被定向到 192.168.0.10。至未知的远程网络将使用默认网关，因此只有在默认路由不合适时，才应为远程网络或主机配置静态路由。此上下文中的 remote 表示没有直接连接到系统的任何网络或主机。

对于 IPv6 配置，ip route 格式的 route6-接口 文件示例 ：

```
2001:db8:1::/48 via 2001:db8::1 metric 2048
2001:db8:2::/48
```

指定退出接口是可选的。如果要强制流量离开特定接口，它可以很有用。例如，对于 VPN，您可以强制到远程网络的流量通过某个网络。 tun0 接口，即使接口位于目标网络的不同子网中。

ip route 格式可用于指定源地址。例如：

```
10.10.10.0/24 via 192.168.0.10 src 192.168.0.2
```

要定义基于策略的现有路由配置，指定多个路由表，请参阅 第 4.5.1 节 “了解策略路由”。

重要

如果 DHCP 已分配了默认网关，并且配置文件中指定了具有相同指标的同一网关，启动过程中或启动接口时会出现错误。可能会显示以下错误消息："RTNETLINK 答案："文件存在"。此错误可能会被忽略。

使用网络/网络掩码指令格式的静态路由

您还可以对路由接口文件使用 network/netmask 指令格式。以下是网络/子网掩码格式的模板，之后提供说明：

```
ADDRESS0=10.10.10.0
NETMASK0=255.255.255.0
GATEWAY0=192.168.1.1
```

- *ADDRESS0=10.10.10.0 是要访问的远程网络或主机的网络地址。*
- *NETMASK0=255.255.255.0 是使用 ADDRESS0=10.10.10.0 定义的网络地址的子网掩码。*
- *GATEWAY0=192.168.1.1 是默认网关，或者可用于访问的 IP 地址 ADDRESS0=10.10.10.0*

以下是使用 network/netmask 指令格式的路由接口文件示例。默认网关为 192.168.0.1，但可通过 192.168.0.10 获得租用的行或 WAN 连接。这两个静态路由到达 10.10.10.0/24 和 172.16.1.0/24 网络：

```
ADDRESS0=10.10.10.0
NETMASK0=255.255.255.0
GATEWAY0=192.168.0.10
```

```
ADDRESS1=172.16.1.10
NETMASK1=255.255.255.0
GATEWAY1=192.168.0.10
```

后续静态路由必须按顺序编号，且不得跳过任何值。例如： ADDRESS0、ADDRESS1、ADDRESS2 等。

出于安全原因，默认禁用从一个接口将数据包转发到另一个接口，或从同一接口转发数据包。

这可以防止系统充当外部流量的路由器。如果您需要系统路由外部流量，比如在共享连接或配置 VPN 服务器时，您需要启用 IP 转发。详情请查看红帽企业 Linux 7 安全指南。

4.5.1. 了解策略路由

策略路由 也称为源路由，是一种更灵活的路由配置机制。路由决策通常根据软件包的目标 IP 地址做出。策略路由 可根据其他路由属性（如源 IP 地址、源端口和协议类型）提供更多灵活性来选择路由。路由表存储关于网络的路由表信息。它们通过数值或名称标识，可在 /etc/iproute2/rt_tables 文件中配置。默认表使用 254 来标识。使用 策略路由时，您还需要规则。规则用于根据数据包的特定属性选择路由表。

对于 initscripts，路由表是路由的属性，可以通过 table 参数进行配置。ip route 格式可以用来定义基于策略的现有路由配置，该配置指定了多个路由表：

```
10.10.10.0/24 via 192.168.0.10 table 1
10.10.10.0/24 via 192.168.0.10 table 2
```

要在 initscripts 中指定路由规则，请将它们编辑到 IPv4 的 /etc/sysconfig/network-scripts/rule-enp1s0 文件，或编辑到 IPv4 的 /etc/sysconfig/network-scripts/rule6-enp1s0 文件。

NetworkManager 支持 policy-routing，但尚不支持规则。规则必须由运行自定义脚本的用户进行配置。对于每个手动静态路由，可以选择路由表：

- ipv4.route-table for IPv4

  和

- ipv6.route-table （适用于 IPv6 ）

通过设置到特定表的路由，来自 DHCP、autoconf6、DHCP6 的所有路由都放置到该特定表中。此外，已配置地址的子网的所有路由都放置在相应的路由表中。例如，如果您配置 192.168.1.10/24 地址，则 192.168.1.0/24 子网包含在 ipv4.route-table 中。

有关 policy-routing 规则的详情，请查看 ip-rule(8)man page。有关路由表，请查看 ip-route(8)man page。

4.6. 配置默认网关

默认网关由网络脚本决定，该脚本首先解析 /etc/sysconfig/network 文件，然后由启动的接口的网络接口 ifcfg 文件 “确定”。ifcfg 文件按照数字顺序解析，最后一个要读取的 GATEWAY 指令则用于编写路由表中的默认路由。

您可以使用 GATEWAY 指令指定默认路由，可以是全局指令，也可以在特定于接口的配置文件中指定。但是，在 Red Hat Enterprise Linux 中，全局 /etc/sysconfig/network 文件已被弃用，现在应仅在每个接口配置文件中指定网关。

在动态网络环境中，移动主机由 NetworkManager 管理，网关信息可能是特定于接口的，最好由 DHCP 分配。在需要影响网络管理器(NetworkManager)要用于访问网关的退出接口选择的特殊情况下，在 ifcfg 文件中为那些不导致默认网关的接口使用 DEFROUTE=no 命令。

第 5 章 配置网络连接设置

本章描述了各种网络连接设置的配置，并演示了如何使用 NetworkManager 配置它们。

5.1. 配置 802.3 链接设置

您可以通过修改以下配置参数来配置以太网连接的 802.3 链接设置：

- 802-3-ethernet.auto-negotiate
- 802-3-ethernet.speed
- 802-3-ethernet.duplex

您可以将 802.3 链接设置配置为三个主要模式：

- 忽略链路协商
- 强制自动协商激活
- 手动设置 速度和 双工 链接设置

忽略链路协商

在这种情况下，NetworkManager 会忽略以太网连接的链接配置，使设备中已配置保留。

要忽略链路协商，请设置以下参数：

```
802-3-ethernet.auto-negotiate = no
802-3-ethernet.speed = 0
802-3-ethernet.duplex = NULL
```

重要

如果 auto-negotiate 参数设置为 no，但没有设置 速度和 双工 值，这并不表示禁用自动协商。

强制自动协商激活

在这种情况下，NetworkManager 在设备中强制自动协商。

要强制自动协商激活，请设置以下选项：

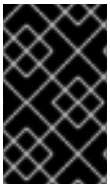
```
802-3-ethernet.auto-negotiate = yes
802-3-ethernet.speed = 0
802-3-ethernet.duplex = NULL
```

手动设置链接速度和双工

在这种情况下，您可以在链路上 手动配置 速度和双工 设置。

要手动设置 速度和 双工 链路设置，请按如下所示设置上述参数：

```
802-3-ethernet.auto-negotiate = no
802-3-ethernet.speed = [speed in Mbit/s]
802-3-ethernet.duplex = [half |full]
```



重要

确保同时设置 speed 和 duplex 值，否则 NetworkManager 不会更新链接配置。

作为系统管理员，您可以使用以下选项之一配置 802.3 链接设置：

- nmcli 工具
- nm-connection-editor 工具

使用 nmcli 工具配置 802.3 链接设置

流程

1. 为 enp1s0 设备创建一个新的以太网连接。
2. 将 802.3 链接设置设置为您选择的配置。详情请查看 第 5.1 节 “配置 802.3 链接设置”

   例如，手动将 速度选项 100 Mbit/s 和 duplex 设置为 full ：

```
nmcli connection add con-name MyEthernet type ethernet ifname enp1s0 \
802-3-ethernet.auto-negotiate no \
802-3-ethernet.speed 100 \
802-3-ethernet.duplex full
```

使用 nm-connection-editor 配置 802.3 链路设置

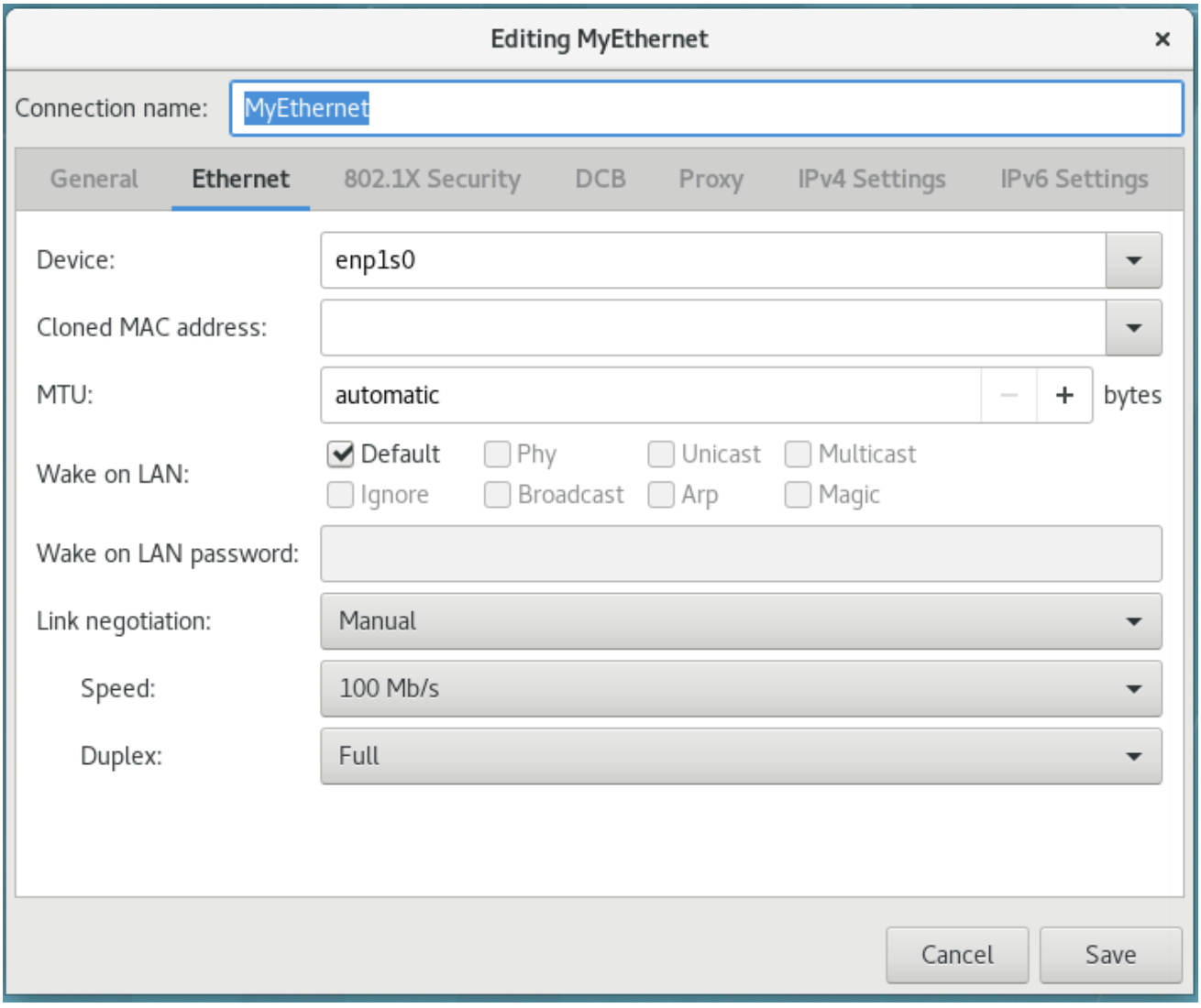
流程

1. 在终端中输入 nm-connection-editor。
2. 选择您要编辑的以太网连接，然后单击 gear wheel 图标以进入编辑对话框。如需更多信息，请参阅 第 3.4.3 节 “使用 nm-connection-editor 的通用配置选项”。

3. *选择您选择的链接协商。*

   - ***ignore* ：跳过链接配置（默认）。**
   - ***Automatic* ：在该设备中强制链接自动协商。**
   - ***Manual* ：可以指定 *Speed* 和 *Duplex* 选项来强制实施链路协商。**

图 5.1. 使用 nm-connection-editor 配置 802.3 链接设置



5.2. 配置 802.1X 安全性

802.1X 安全是用于基于端口的网络访问控制(PNAC)的 IEEE 标准的名称。它也称为 WPA Enterprise。802.1X 安全性是控制从物理网络访问逻辑网络的一种方式。要加入逻辑网络的所有客户端都必须使用正确的 802.1X 身份验证方法与服务器（例如路由器）进行身份验证。

802.1X 安全性最常与保护无线网络(WLAN)相关，但也可用于防止对网络(LAN)具有物理访问权限的入侵者获得进入。

过去，DHCP 服务器不配置为将 IP 地址租给未授权用户，但出于各种原因，这种做法既不现实又不安全，因此不再推荐使用。取而代之，可使用 802.1X 安全性来确保通过基于端口的身份验证实现逻辑安全网络。

802.1X 为 WLAN 和 LAN 访问控制提供了一个框架，并且充当执行一种可扩展验证协议(EAP)类型的信封。EAP 类型是一种协议，用于定义如何在网络上实现安全性。

5.2.1. 使用 nmcli 为 Wi-Fi 配置 802.1X 安全性

流程

1. 设置经过身份验证 的 key-mgmt （密钥管理）协议。它为安全 Wifi 连接配置密钥机制。有关属性的详情，请查看 nm-settings(5) man page。

2. 配置 802-1x 身份验证设置。有关传输层安全性(TLS)身份验证，请参阅 “配置 TLS 设置”一节。

表 5.1. 802-1x 身份验证设置

802-1X 身份验证设置	名称
802-1x.identity	身份
802-1x.ca-cert	CA 证书
802-1x.client-cert	用户证书

802-1X 身份验证设置	名称
802-1x.private-key	私钥
802-1x.private-key-password	私钥密码

例如，要使用 EAP-TLS 身份验证方法配置 WPA2 Enterprise，请应用以下设置：

```
nmcli c add type wifi ifname wlo61s0 con-name 'My Wifi Network' \
    802-11-wireless.ssid 'My Wifi' \
    802-11-wireless-security.key-mgmt wpa-eap \
    802-1x.eap tls \
    802-1x.identity identity@example.com \
    802-1x.ca-cert /etc/pki/my-wifi/ca.crt \
    802-1x.client-cert /etc/pki/my-wifi/client.crt \
    802-1x.private-key /etc/pki/my-wifi/client.key \
    802-1x.private-key-password s3cr3t
```

5.2.2. 使用 nmcli 为 Wired 配置 802.1X 安全性

要使用 nmcli 工具配置有线连接，请遵循与无线连接相同的步骤，但 802-11-wireless.ssid 和 802-11-wireless-security.key-mgmt 设置除外。

5.2.3. 使用 GUI 为 Wi-Fi 配置 802.1X 安全性

流程

1. 打开网络窗口（请参阅 第 3.4.1 节 “使用 control-center GUI 连接到网络”）。
2. 从右侧菜单中选择 Wireless 网络接口。如有必要，将符号链接电源按钮设置为 ON，并检查您的硬件交换机是否已开启。
3. 选择新连接的连接名称，或者点击现有连接配置集的 gear wheel 图标，您要为其配置 802.1X 安全性。如果是新连接，请完成任何身份验证步骤来完成连接，然后单击 gear wheel 图标。

4. 选择 **Security**。

可用的配置选项如下：

安全性

none - 不要加密 Wi-Fi 连接。

WEP 40/128 位密钥 - Wired Equivalent Privacy(WEP)，来自于 IEEE 802.11 标准。使用单一预共享密钥（PSK）。

WEP 128 位密码 - 密码短语的 MD5 哈希将用于生成 WEP 密钥。

LEAP - Cisco 系统的轻量级可扩展身份验证协议.

动态 **WEP(802.1X) - WEP 密钥动态更改.配合使用** **“配置 TLS 设置”一节**

WPA 和 WPA2 个人 - Wi-Fi Protected Access(WPA)，来自 IEEE 802.11i 标准草案.一个 WEP 的替换。Wi-Fi Protected Access II（WPA2），来自于 802.11i-2004 标准。个人模式，使用预共享密钥（WPA-PSK）。

WPA 和 WPA2 Enterprise - WPA 与 RADIUS 身份验证服务器配合使用，以提供 IEEE 802.1X 网络访问控制。配合使用 **“配置 TLS 设置”一节**

密码

输入要在身份验证过程中使用的密码。

5. **从下拉菜单中选择以下安全方法之一：LEAP、Dynamic WEP(802.1X)或 WPA & WPA2 Enterprise。**

如需了解在安全下拉菜单中选择哪些 可扩展身份验证协议(EAP)类型与您的选择对应，请参阅 “配置 TLS 设置”一节。

5.2.4. 使用 nm-connection-editor 为 Wired 配置 802.1X 安全性

流程

1. 在终端中输入 nm-connection-editor。

```
~]$ nm-connection-editor
```

这时将显示 Network Connections 窗口。

2. 选择您要编辑的以太网连接并点击 gear wheel 图标，请参阅 第 3.4.6.2 节 “使用 nm-connection-editor 配置 Wired Connection”。

3. 选择安全性并将符号链接电源按钮设置为 ON 以启用设置配置。

4. 从以下验证方法中选择：

- 选择 TLS 用于传输层安全性，然后继续 “配置 TLS 设置”一节。
- 选择 FAST for Flexible Authentication via Secure Tunneling，再继续 “配置 Tunneled TLS 设置”一节;
- 为 Tunneled Transport Layer Security 选择 Tunneled TLS，否则称为 TTLS 或 EAP-TTLS，然后继续 “配置 Tunneled TLS 设置”一节 ；
- 选择 Protected EAP(PEAP) for Protected Extensible Authentication Protocol，再继续 “配置受保护的 EAP(PEAP)设置”一节。

配置 TLS 设置

通过传输层安全性(TLS)，客户端和服务器使用 TLS 协议进行相互身份验证。服务器演示其持有数字证书、客户端使用其客户端证书证明自己的身份，并交换关键信息。身份验证完成后，不再使用 TLS 隧道。客户端和服务器使用交换密钥来使用 AES、TKIP 或 WEP 加密数据。

证书必须分发到希望进行身份验证的所有客户端的事实，这表示 EAP-TLS 身份验证方法非常强大，但设置起来也更为复杂。使用 TLS 安全性需要公钥基础架构(PKI)开销来管理证书。使用 TLS 安全性的好处是被泄露的密码不允许访问(W)LAN：入侵者还必须有权访问客户端的私钥。

NetworkManager 不决定支持 TLS 版本。NetworkManager 收集用户输入的参数，并将它们传递给守护进程 wpa_supplicant，该守护进程负责处理该过程。反过来，它使用 OpenSSL 来建立 TLS 隧道。OpenSSL 本身协商 SSL/TLS 协议版本。它使用最高版本的两端支持。

要配置 TLS 设置，请按照 第 5.2.4 节 “使用 nm-connection-editor 为 Wired 配置 802.1X 安全性” 中描述的步骤操作。可用的配置设置如下：

身份

提供此服务器的身份。

用户证书

单击可浏览并选择个人 X.509 证书文件，这些证书文件编码为 Distinguished Encoding Rules (DER) 或隐私增强邮件(PEM)。

CA 证书

单击可浏览并选择 X.509 证书颁发机构证书文件，这些证书颁发机构文件编码为 Distinguished Encoding Rules (DER)或 Privacy Enhanced Mail(PEM)。

私钥

点击以浏览并选择使用 Distinguished Encoding Rules (DER)、隐私增强邮件(PEM) 或个人信息交换语法标准 (PKCS #12) 编码的私钥文件。

私钥密码

在 Private key 字段中输入私钥的密码。选择 Show password 使密码在您键入时可见。

配置 FAST 设置

要配置 FAST 设置，请遵循 第 5.2.4 节 “使用 nm-connection-editor 为 Wired 配置 802.1X 安全性” 中描述的步骤。可用的配置设置如下：

匿名身份

提供此服务器的身份。

PAC 置备

选中要启用的复选框，然后从 Anonymous、Authentic ated 和 Both 中选择。

PAC 文件

单击可浏览并选择受保护的访问凭据(PAC)文件。

内部身份验证

GTC - 通用令牌卡.

MSCHAPv2 - Microsoft Challenge Handshake Authentication Protocol 版本 2.

username

输入要在身份验证过程中使用的用户名。

密码

输入要在身份验证过程中使用的密码。

配置 Tunneled TLS 设置

要配置 Tunneled TLS 设置，请按照 第 5.2.4 节 “使用 nm-connection-editor 为 Wired 配置 802.1X 安全性” 中所述的步骤操作。可用的配置设置如下：

匿名身份

这个值被用作未加密的身份。

CA 证书

单击浏览并选择证书认证机构的证书。

内部身份验证

PAP - 密码身份验证协议.

MSCHAP - 挑战处理身份验证协议.

MSCHAPv2 - Microsoft Challenge Handshake Authentication Protocol 版本 2.

CHAP - 挑战处理身份验证协议.

username

输入要在身份验证过程中使用的用户名。

密码

输入要在身份验证过程中使用的密码。

配置受保护的 EAP(PEAP)设置

要配置受保护的 EAP(PEAP)设置，请按照 第 5.2.4 节 “使用 nm-connection-editor 为 Wired 配置 802.1X 安全性” 中描述的步骤操作。可用的配置设置如下：

匿名身份

这个值被用作未加密的身份。

CA 证书

单击浏览并选择证书认证机构的证书。

PEAP 版本

要使用的受保护 EAP 版本。自动、0 或 1.

内部身份验证

MSCHAPv2 - Microsoft Challenge Handshake Authentication Protocol 版本 2.

MD5 - 消息摘要 5，一种加密哈希功能。

GTC - 通用令牌卡.

username

输入要在身份验证过程中使用的用户名。

密码

输入要在身份验证过程中使用的密码。

5.3. 使用带有 WPA_SUPPLICANT 和 NETWORKMANAGER的 MACSEC

介质访问控制安全 （MACsec、IEEE 802.1AE）使用 GCM-AES-128 算法加密并验证 LAN 中的所有流量。MACsec 不仅可以保护 IP，还可以保护地址解析协议(ARP)、邻居发现(ND)或 DHCP。IPsec 在网络层（层 3）上运行，而 SSL 或 TLS 在应用层（层 7）上运行，而 MACsec 在数据链路层（层 2）中运行。将 MACsec 与其他网络层的安全协议相结合，以利用这些规则提供的不同安全功能。

使用使用预共享连接关联密钥/CAK 名称(CAK/CKN)对执行身份验证的交换机启用 MACsec ：

流程

1. *创建 CAK/CKN 对。例如，以下命令在十六进制表示中生成 16 字节密钥：*

```
~]$ dd if=/dev/urandom count=16 bs=1 2> /dev/null | hexdump -e '1/2 "%02x"'
```

2. *创建 wpa_supplicant.conf 配置文件并添加以下行：*

```
ctrl_interface=/var/run/wpa_supplicant
eapol_version=3
```

```
ap_scan=0
fast_reauth=1

network={
    key_mgmt=NONE
    eapol_flags=0
    macsec_policy=1

    mka_cak=0011... # 16 bytes hexadecimal
    mka_ckn=2233... # 32 bytes hexadecimal
}
```

使用上一步中的值完成 **wpa_supplicant.conf** 配置文件中的**ka_cak** 和**mka_ckn** 行。

如需更多信息，请参阅 **wpa_supplicant.conf(5) man page**。

3. 假设您使用 **wlp61s0** 连接到您的网络，使用以下命令启动 **wpa_supplicant** ：

```
~]# wpa_supplicant -i wlp61s0 -Dmacsec_linux -c wpa_supplicant.conf
```

红帽建议使用 **nmcli** 命令配置 **wpa_supplicant.conf** 文件，而不是创建和编辑 **wpa_supplicant.conf** 文件。以下示例假定您已有一个 **16** 字节十六进制 **CAK($MKA_CAK)**和 **32** 字节十六进制 **CKN($MKA_CKN)**：

```
~]# nmcli connection add type macsec \
  con-name test-macsec+ ifname macsec0 \
  connection.autoconnect no \
  macsec.parent wlp61s0 macsec.mode psk \
  macsec.mka-cak $MKA_CAK \
  macsec.mka-cak-flags 0 \
  macsec.mka-ckn $MKA_CKN

~]# nmcli connection up test-macsec+
```

在这一步后，应配置 **macsec0** 设备并用于联网。

如需了解更多详细信息，请参阅 **MACsec 中的新功能：使用 wpa_supplicant 和（可选）NetworkManager 设置 MACsec**。另外，请参阅 **MACsec：加密网络流量文章的不同解决方案，以了解有关 MACsec 网络架构、用例场景和配置示例的更多信息。**

5.4. 配置 IPV4 设置

使用 control-center 配置 IPv4 设置

流程

1. 按 Super 键进入 Activities Overview，键入 Settings，然后按 Enter 键。然后，选择左侧的网络 选项卡，并显示 网络设置 工具。继续 “使用 control-center 配置新连接”一节。

2. 选择您要编辑的连接并点击 gear wheel 图标。此时 将显示编辑 对话框。

3. 单击 IPv4 菜单条目。

IPv4 菜单条目允许您配置用于连接到网络的方法，根据需要输入 IP 地址、DNS 和路由信息。当您创建和修改以下连接类型之一时，IPv4 菜单条目可用：有线、无线、移动宽带、VPN 或 DSL。

如果您使用 DHCP 从 DHCP 服务器获取动态 IP 地址，只需将地址设置为 Automatic(DHCP)。

如果您需要配置静态路由，请参阅 第 4.3 节 “使用 GUI 配置静态路由”。

使用 nm-connection-editor 设置 IPV4 的方法

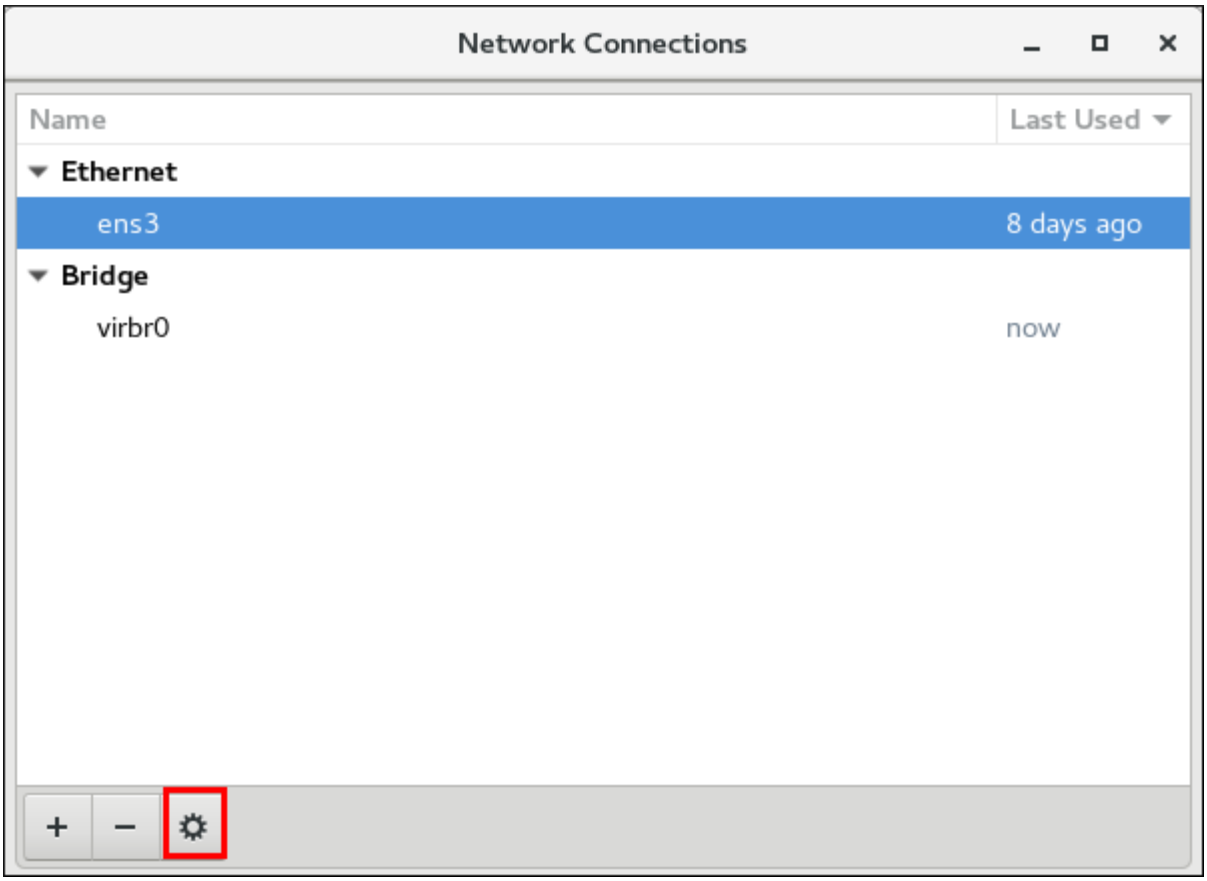
您可以使用 nm-connection-editor 编辑和配置连接设置。这个步骤描述了如何配置 IPv4 设置：

流程

1. 在终端中输入 nm-connection-editor。

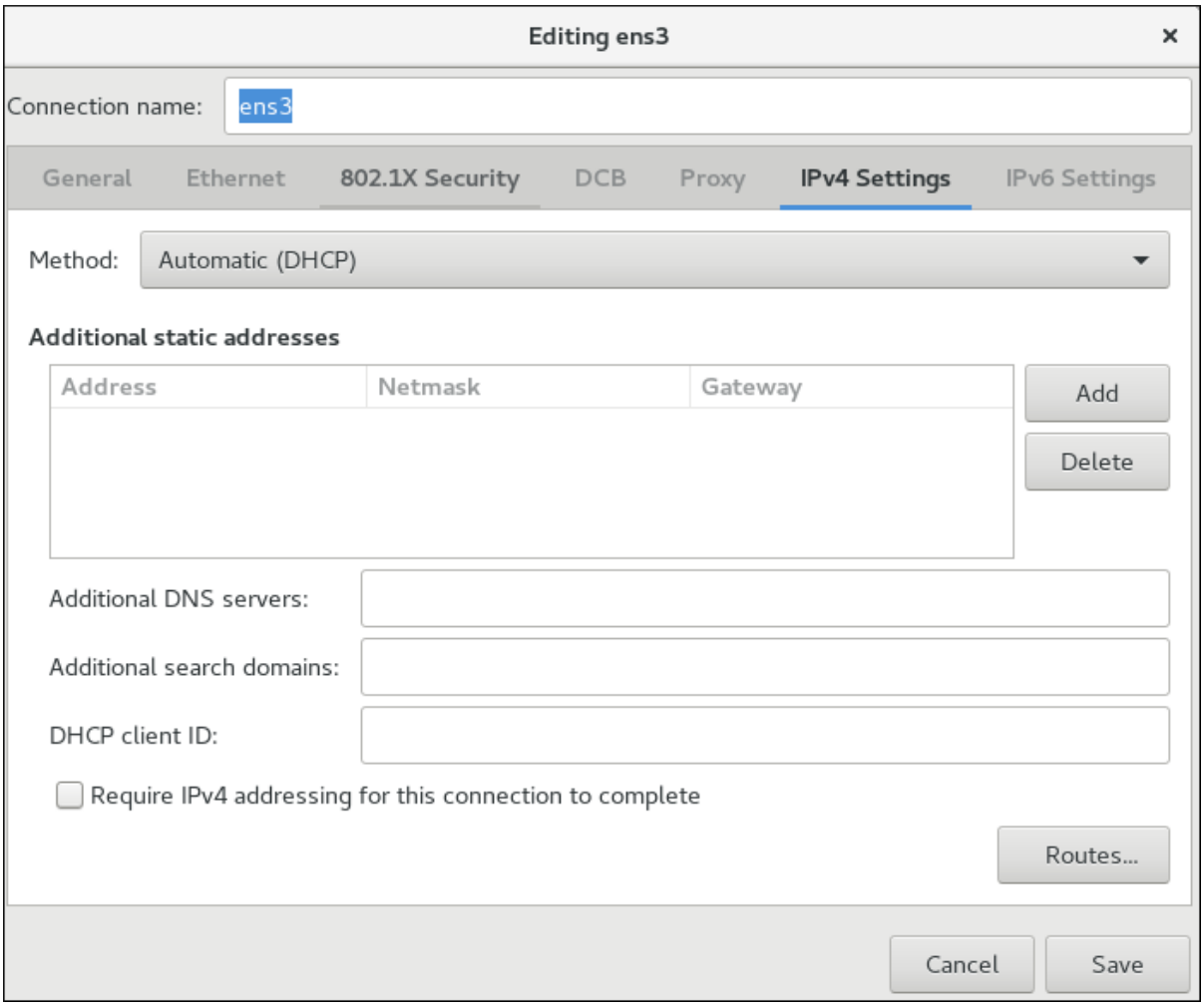
2. 对于现有连接类型，点击 gear wheel 图标。

图 5.2. 编辑连接



3. *单击 **IPv4 Settings**。*

图 5.3. 配置 IPv4 设置



按连接类型提供的可用 IPv4 方法

单击 Method 下拉菜单时，根据您要配置的连接类型，您可以选择以下 IPv4 连接方法之一：所有方法都根据与之关联的连接类型或类型在此处列出：

wired、连线和 DSL 连接方法

自动(DHCP) - 如果您要连接的网络使用 DHCP 服务器分配 IP 地址，请选择这个选项。您无需填写 DHCP 客户端 ID 字段。

仅自动(DHCP)地址 - 如果您要连接的网络使用 DHCP 服务器分配 IP 地址，但您想要手动分配 DNS 服务器，请选择这个选项。

Manual - 如果要手动分配 IP 地址，请选择这个选项。

仅链接 - 如果您要连接的网络没有 DHCP 服务器且您不想手动分配 IP 地址，请选择这个选项。随机地址将根据RFC 3927 分配前缀 169.254/16。

共享到其他计算机 - 如果您要配置的界面是共享互联网或 WAN 连接，请选择这个选项。该接口分配了一个范围为 10.42.x.1/24 范围、DHCP 服务器和 DNS 服务器，并且接口连接到系统上具有网络地址转换 (NAT)的默认网络连接。

禁用 - 此连接禁用 IPv4。

移动带宽连接方法

自动(PPP) - 如果您要连接的网络会自动分配 IP 地址和 DNS 服务器，请选择这个选项。

仅自动(PPP)地址 - 如果您要连接的网络会自动分配 IP 地址，请选择这个选项，但您想要手动指定 DNS 服务器。

VPN 连接方法

自动(VPN) - 如果您要连接的网络会自动分配 IP 地址和 DNS 服务器，请选择这个选项。

仅自动(VPN)地址 - 如果您要连接的网络会自动分配 IP 地址，请选择这个选项，但您想要手动指定 DNS 服务器。

DSL 连接方法

自动(PPPoE) - 如果您要连接的网络会自动分配 IP 地址和 DNS 服务器，请选择这个选项。

仅自动(PPPoE)地址 - 如果您要连接的网络自动分配 IP 地址，请选择这个选项，但您想要手动指定 DNS 服务器。

如果您使用 DHCP 从 DHCP 服务器获取动态 IP 地址，只需将 Method 设置为 Automatic(DHCP)。

如果您需要配置静态路由，请点击 Routes 按钮，并参阅 第 4.3 节 “使用 GUI 配置静态路由”。

5.5. 配置 IPV6 设置

要配置 IPv6 设置，请按照 第 5.4 节 “配置 IPv4 设置” 中描述的步骤并点击 IPv6 菜单条目。

方法

Ignore - 如果要忽略这个连接的 IPv6 设置，请选择这个选项。

自动 - 选择此选项以使用 SLAAC 根据硬件地址和路由器公告 (RA)创建自动无状态配置。

仅自动地址 - 如果您要连接的网络使用路由器公告 (RA)创建自动无状态配置，但您想要手动分配 DNS 服务器，请选择此选项。

自动，仅 DHCP - 选择此选项以不使用 RA，而是直接从 DHCPv6 请求信息以创建有状态配置。

Manual - 如果要手动分配 IP 地址，请选择这个选项。

仅链接 - 如果您要连接的网络没有 DHCP 服务器且您不想手动分配 IP 地址，请选择这个选项。随机地址将根据RFC 4862 进行分配，前缀为 FE80::0。

地址

DNS 服务器 - 输入以逗号分隔的 DNS 服务器列表。

搜索域 - 输入以逗号分隔的域控制器列表。

如果您需要配置静态路由，请点击 Routes 按钮，并参阅 第 4.3 节 “使用 GUI 配置静态路由”。

5.6. 配置 PPP（点对点）设置

身份验证方法

在大多数情况下，供应商的 PPP 服务器支持所有允许的身份验证方法。如果连接失败，用户应根据

PPP 服务器配置禁用对某些方法的支持。

使用点对点加密(MPPE)

Microsoft Point-To-Point 加密协议(RFC 3078)。

允许 BSD 数据压缩

PPP BSD 压缩协议(RFC 1977)。

允许取消数据压缩

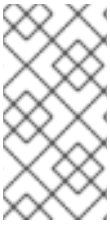
PPP 调试协议(RFC 1979)。

使用 TCP 标头压缩

低速串行链接(RFC 1144)压缩 TCP/IP 标头。

发送 PPP echo 数据包

用于回环测试的 LCP Echo-Request 和 Echo-ReplyCodes(RFC 1661)。



注意

由于 NetworkManager 中的 PPP 支持是可选的，要配置 PPP 设置，请确保已安装了 NetworkManager-ppp 软件包。

第 6 章 配置主机名

6.1. 了解主机名

主机名有三种类型：静态、友好和瞬态。

“静态主机名是传统主机名”，可由用户选择，存储在 /etc/hostname 文件中。“临时主机名是由内核维护的动态主机名”。默认情况下，它将初始化为静态主机名，其值默认为 “localhost”。它可以在运行时由 DHCP 或 mDNS 更改。“用户友善的主机名是一个自由格式” UTF8 主机名，供用户使用。

注意

主机名可以是长度最多 64 个字符的自由格式字符串。但是，红帽建议静态和临时名称与 DNS 中用于完全限定域名(FQDN)匹配，如 host.example.com。此外，建议静态和临时名称仅包含 7 位 ASCII 小写字符、没有空格或点，并将自身限制为 DNS 域名标签允许的格式，即使这不是严格的要求。旧规格不允许下划线，因此不建议使用它们。

hostnamectl 工具将强制实施以下内容：静态和瞬态主机名包含 a-z、A-Z、0-9、-、_ “和” “.” 仅（不以“点” 开始或结束），且不要紧接点后有两个点。强制实施 64 个字符的大小限制。

6.1.1. 推荐的命名实践

互联网编号分配公司(ICANN) 有时会在公共寄存器中添加以前未注册的顶级域（如.yourcompanyany）。因此，红帽强烈建议您不要使用没有委托给您的域名，即使使用私有网络中也是如此，因为这可能导致根据网络配置的不同解析域名。因此，网络资源可能会不可用。使用未委派的域名也会增加 DNSSEC 部署和维护的难度，因为域名冲突需要手动配置来启用 DNSSEC 验证。 有关此问题的更多信息，请参阅域名冲突上的 ICANN 常见问题解答。

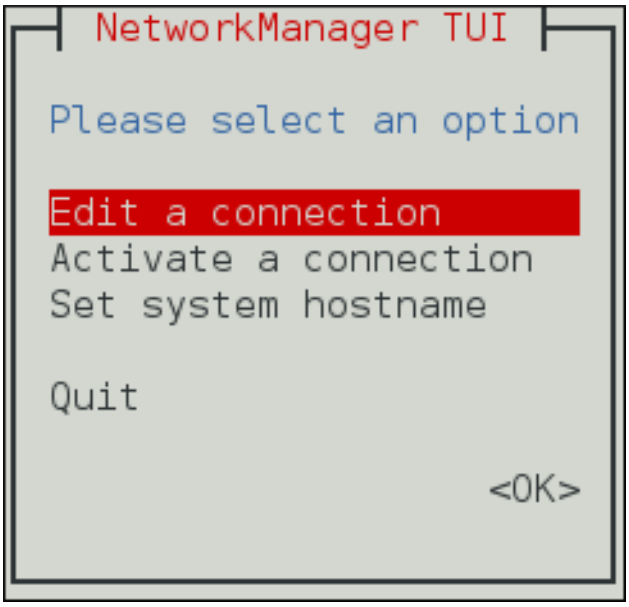
6.2. 使用文本用户界面配置主机名，NMTUI

文本用户界面工具 nmtui 可用于在终端窗口中配置主机名。使用以下命令启动该工具：

```
~]$ nmtui
```

此时将显示文本用户界面。任何无效的命令都会打印用法消息。

图 6.1. NetworkManager 文本用户界面起始菜单



[D]

要导航，请使用箭头键或按 Tab 键前进，然后按ShiftTab后退步浏览选项。按 Enter 键选择一个选项。Space bar 切换复选框的状态。

有关安装 nmtui 的详情请查看 第 3.2 节 “使用 nmtui 配置 IP 网络”。

NetworkManager 文本用户界面工具 nmtui 可用于查询和设置 /etc/hostname 文件中的静态主机名。

重要

在 Red Hat Enterprise Linux 中，NetworkManager 使用 systemd-hostnamed 服务读取和写入静态主机名，该主机名存储在 /etc/hostname 文件中。因此，NetworkManager 不再自动对 /etc/hostname 文件进行手动修改；您应该通过 hostnamectl 实用程序更改系统主机名。另外，在 /etc/sysconfig/network 文件中使用 HOSTNAME 变量现已弃用。

6.3. 使用 HOSTNAMECTL 配置主机名

提供 hostnamectl 工具用于管理给定系统上使用的三个独立主机名类别。

6.3.1. 查看所有主机名

要查看所有当前主机名，请输入以下命令：

```
-
```

```
~]$ hostnamectl status
```

如果未指定 选项，则默认表示 **status** 选项。

6.3.2. 设置所有主机名

要在系统中设置所有主机名，以 **root** 用户身份输入以下命令：

```
~]# hostnamectl set-hostname name
```

这将改变好奇、静态和瞬态主机名等。静态和临时主机名将简化用户友善主机名的形式。空格将被替换为 “-”，并且将删除特殊字符。

6.3.3. 设置部分主机名

要设置特定主机名，以 **root** 用户身份输入以下命令并带有相关选项：

```
~]# hostnamectl set-hostname name [option...]
```

其中 选项是 **--pretty**、**--static** 和 **-- transient** 的一个或多个。

如果 **--static** 或 **--transient** 选项与 **--pretty** 选项一同使用，静态和临时主机名将会简化用户友善主机名的形式。空格将被替换为 “-”，并且将删除特殊字符。如果未提供 **--pretty** 选项，则不会进行简化。

在设置用户友善的主机名时，如果主机名包含空格或单个引号，请记住使用相应的引号。例如：

```
~]# hostnamectl set-hostname "Stephen's notebook" --pretty
```

6.3.4. 清除具体主机名

要清除特定主机名并允许它恢复到默认主机名，以 **root** 用户身份使用相关选项输入以下命令：

```
~]# hostnamectl set-hostname "" [option...]
```

其中""是一个带引号的空字符串，其中 选项为一个或多个 ：**--pretty**、**--static** 和 **--transient**。

6.3.5. 远程更改主机名

要在远程系统中执行 hostnamectl 命令，请使用 -H, --host 选项，如下所示：

```
~]# hostnamectl set-hostname -H [username]@hostname
```

其中 hostname 是您要配置的远程主机。用户名是可选的。hostnamectl 工具将使用 SSH 连接到远程系统。

6.4. 使用 NMCLI 配置主机名

NetworkManager 工具 nmcli 可用于查询和设置 /etc/hostname 文件中的静态主机名。

要查询静态主机名，请运行以下命令：

```
~]$ nmcli general hostname
```

要将静态主机名设置为 my-server，以 root 用户身份运行以下命令：

```
~]# nmcli general hostname my-server
```

6.5. 其它资源

- hostnamectl(1) man page - 描述 hostnamectl，包括命令和命令选项。
- hostname(1) 手册页 - 包含主机名和 域名命令的说明。
- hostname(5) 手册页 - 包含主机名文件及其内容和使用的说明.
- hostname(7) man page - 包含主机名解析的说明。
- machine-info(5) 手册页 - 描述本地计算机信息文件及其包含的环境变量。

- *machine-id(5) 手册页 - 描述本地机器 ID 配置文件。*

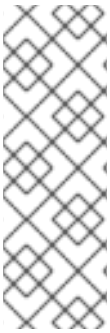
- *systemd-hostnamed.service(8)man page - 描述 hostnamectl 使用的 systemd-hostnamed 系统服务。*

第 7 章 配置网络绑定

红帽企业 Linux 7 允许管理员将多个网络接口绑定到单个绑定通道中。通道绑定使两个或多个网络接口能够充当一个接口，同时增大带宽并提供冗余。

警告

不支持在不使用网络交换机的情况下使用直接电缆连接来绑定。如果没有网络交换机，此处描述的故障转移机制将无法按预期工作。如需更多信息，请参阅红帽知识库文章，为什么不支持直接连接的绑定？



注意

active-backup、balance-tlb 和 balance-alb 模式不需要对交换机进行任何具体的配置。其他绑定模式需要配置交换机来聚合链接。例如，Cisco 交换机需要 EtherChannel for Modes 0、2 和 3，但对于模式 4 LACP 和 EtherChannel 是必需的。请参阅您的交换机提供的文档并查看 https://www.kernel.org/doc/Documentation/networking/bonding.txt

7.1. 了解控制器和端口接口的默认行为

使用 NetworkManager 守护进程控制绑定端口接口时，特别是在查找故障时，请牢记以下几点：

1. 启动控制器接口不会自动启动端口接口。
2. 启动端口接口总会启动控制器接口。
3. 停止控制器接口也会停止端口接口。
4. 没有端口的控制器可以启动静态 IP 连接。
5. 没有端口的控制器在启动 DHCP 连接时会等待端口。

6.
当添加具有载波的端口时，等待端口且具有 DHCP 连接的控制器会完成。

7.
当添加不具有载体的端口时，等待端口且具有 DHCP 连接的控制器将继续等待。

7.2. 使用文本用户界面 NMTUI 配置绑定

文本用户界面工具 nmtui 可用于在终端窗口中配置绑定。使用以下命令启动该工具：

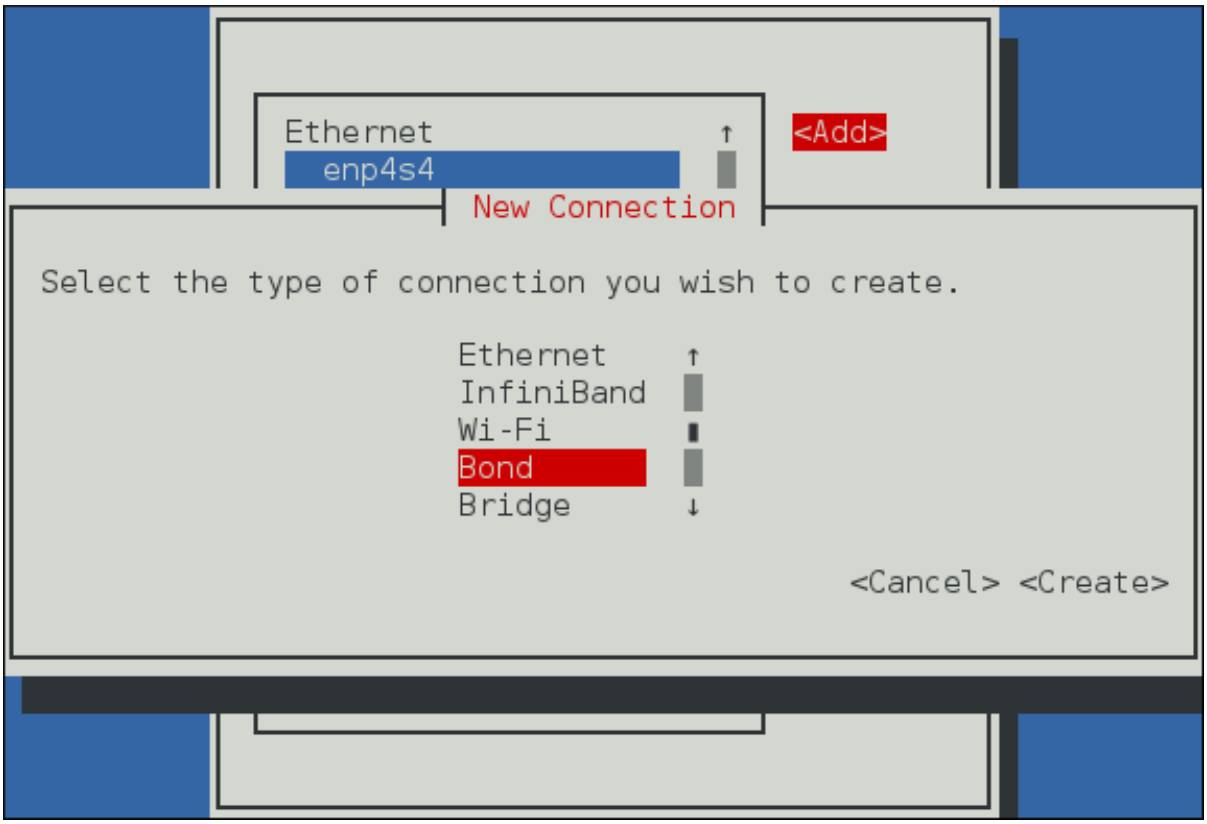
```
~]$ nmtui
```

此时将显示文本用户界面。任何无效的命令都会打印用法消息。

要导航，请使用箭头键或按 Tab 键前进，然后按ShiftTab后退步浏览选项。按 Enter 键选择一个选项。Space bar 切换复选框的状态。

1.
在起始菜单中选择 Edit a connection。选择 Add，这会打开 New Connection 屏幕。

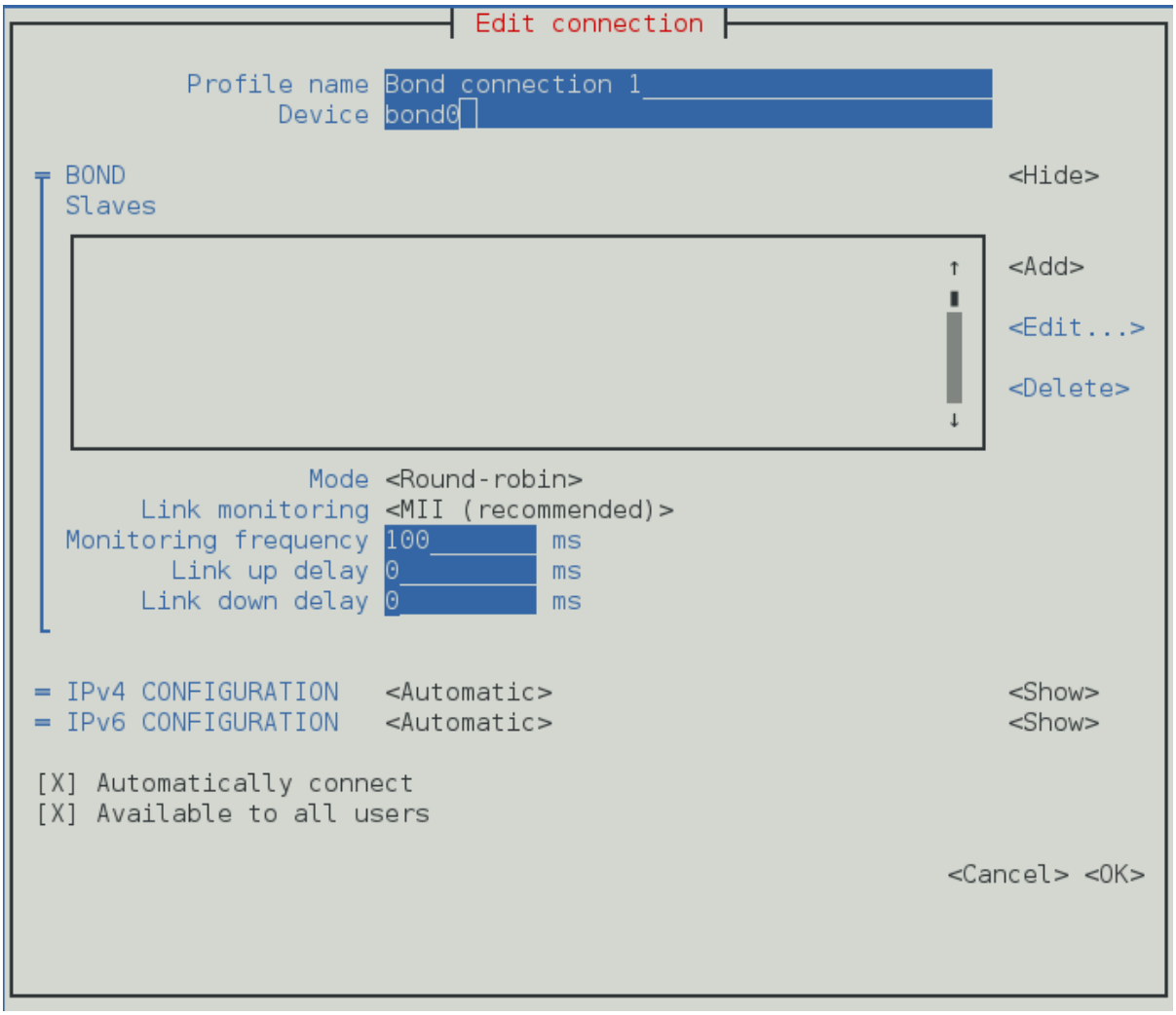
图 7.1. NetworkManager 文本用户界面添加绑定连接菜单



[D]

2. *选择 **Bond**，然后选择 **Create** ；绑定的编辑连接屏幕将打开。*

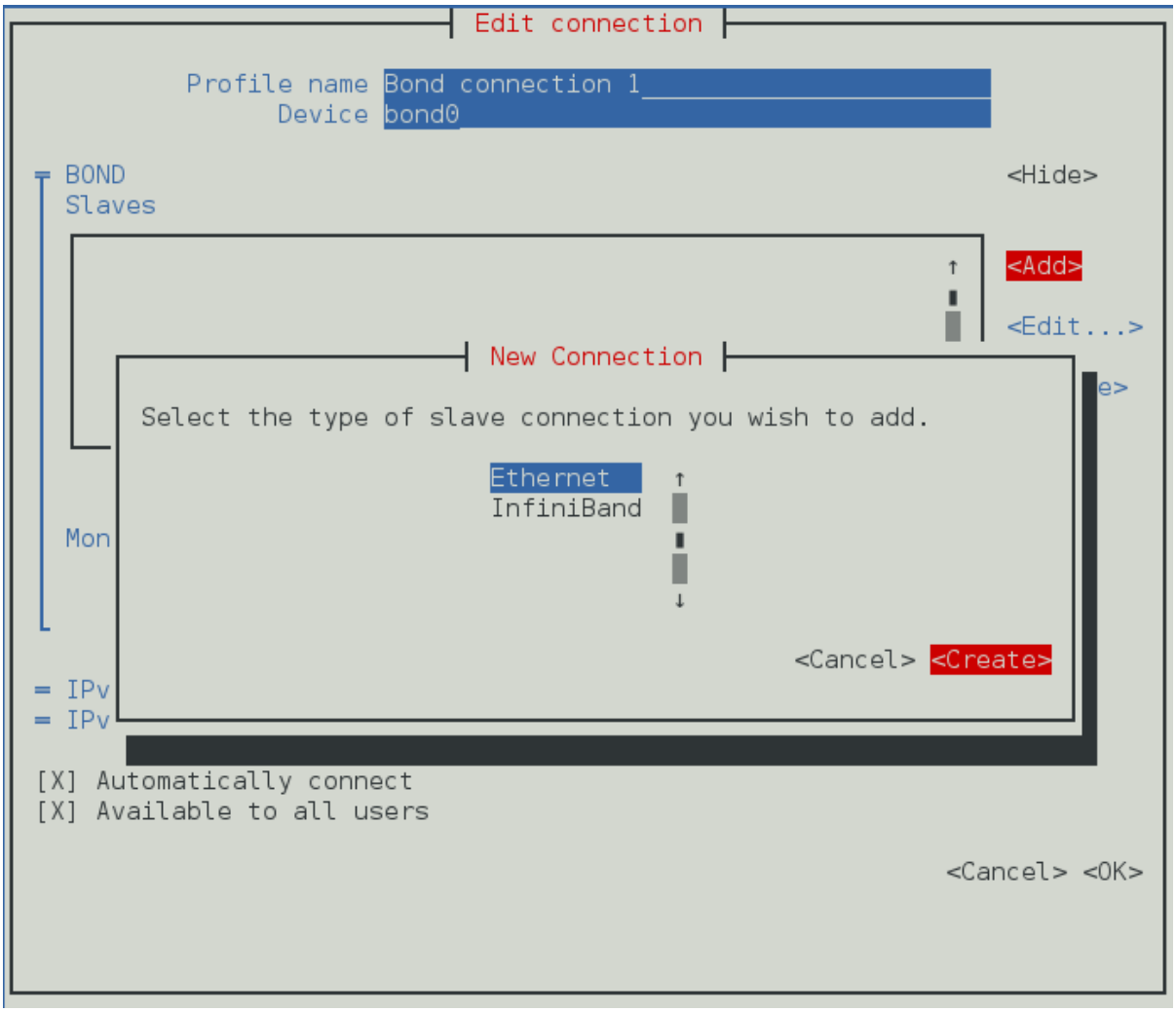
图 7.2. NetworkManager 文本用户界面配置 Bond Connection 菜单



[D]

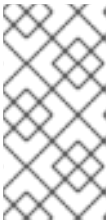
3. *此时，需要将端口接口添加到绑定中；要添加这些选择添加添加 添加，可打开 New Connection 屏幕。选择连接类型后，选择创建按钮。*

图 7.3. NetworkManager 文本用户界面配置新绑定从连接菜单



[D]

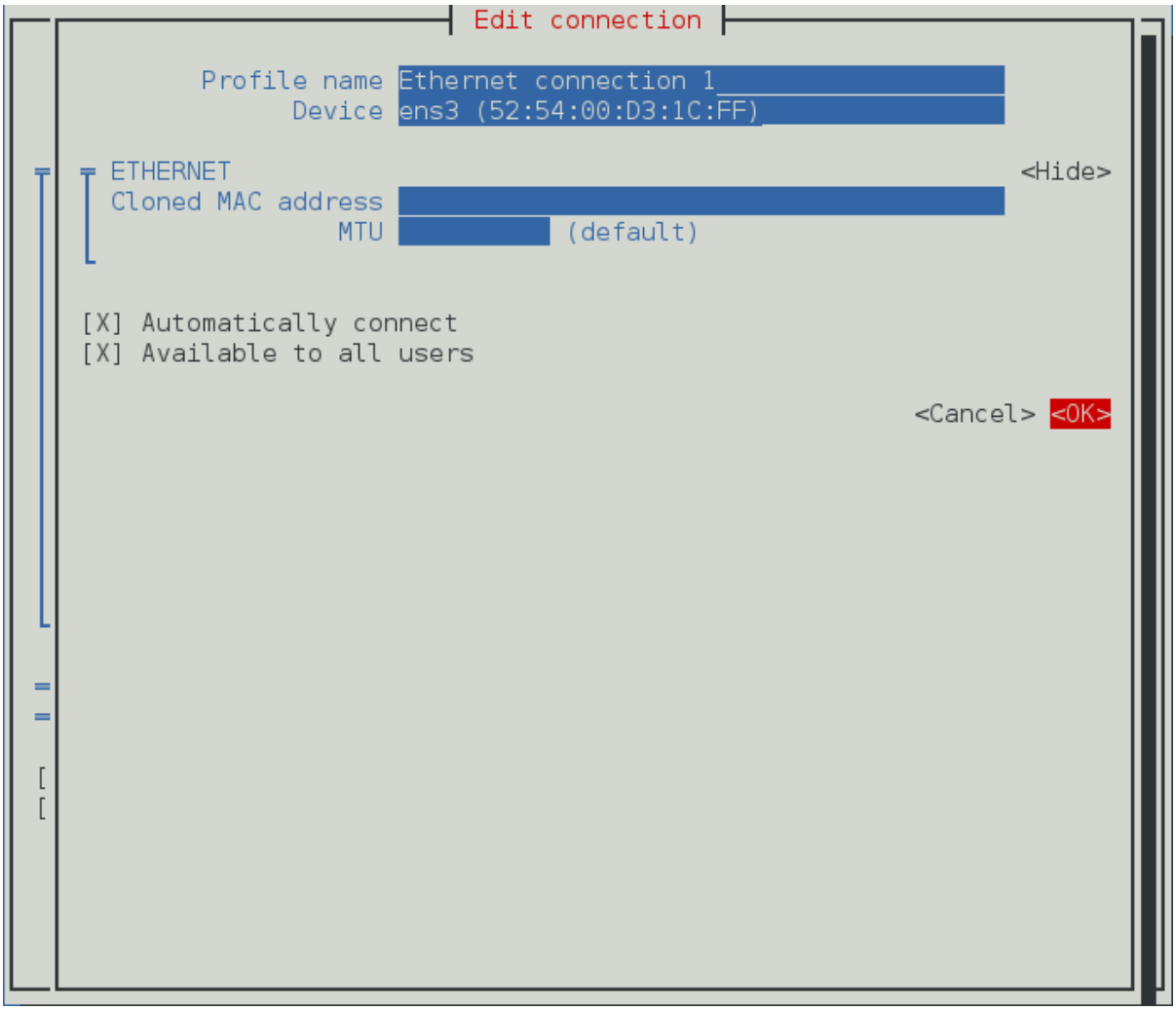
4. *端口的 Edit Connection 显示会出现；在 Device 部分输入所需端口的设备名称或 MAC 地址。如果需要，通过选择 Show to the Ethernet 标签右侧输入要用作绑定 MAC 地址的克隆 MAC 地址。选择 OK 按钮以保存端口。*



注意

如果设备没有 MAC 地址指定，则当 Edit Connection 窗口重新加载后，仅当成功找到该设备时，会自动填充 Device 部分。

图 7.4. NetworkManager 文本用户界面配置 Bond Slave Connection 菜单

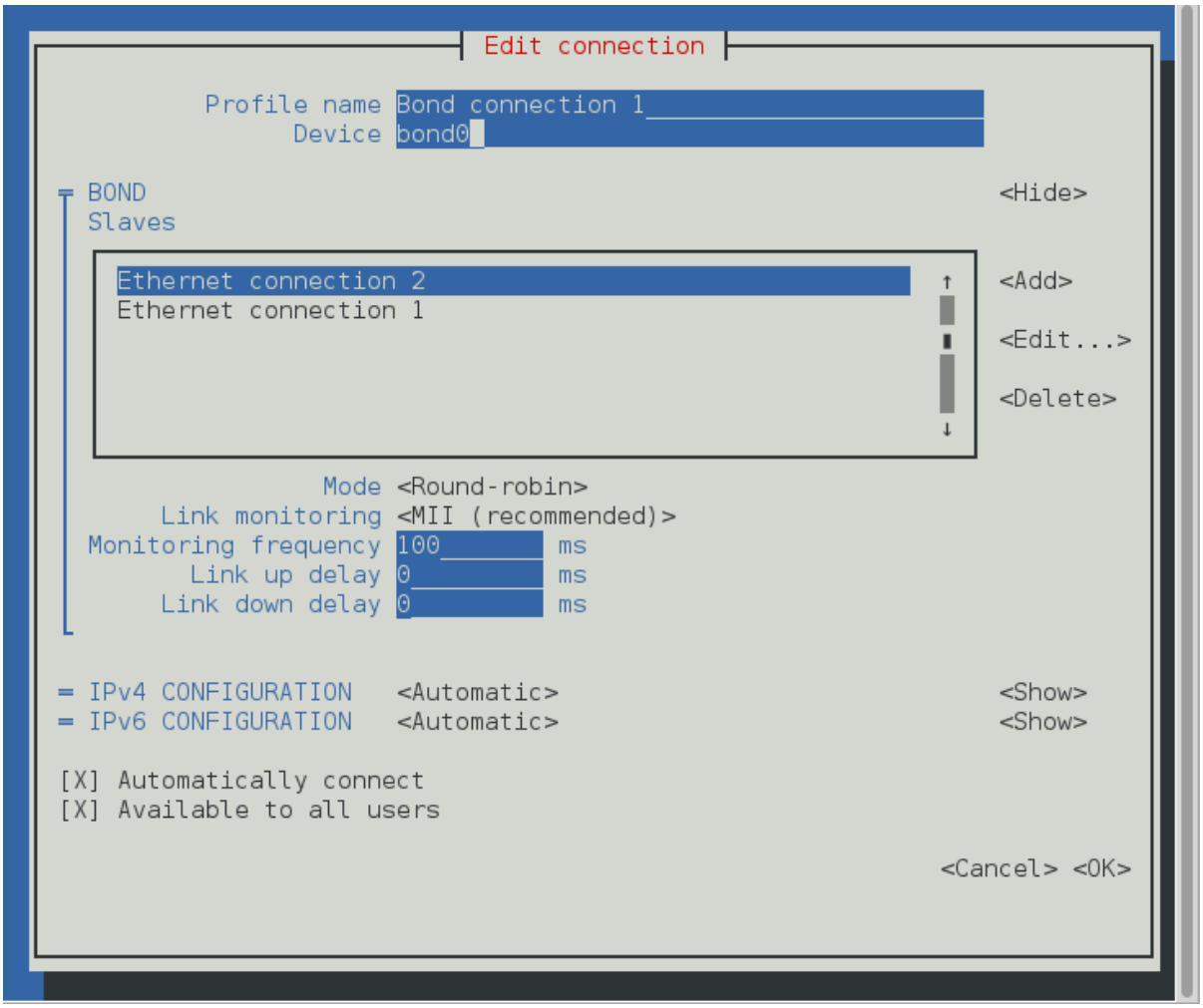


[D]

5. *绑定端口的名称会出现在 Slaves 部分。重复上述步骤，以添加其他端口连接。*

6. *在选择确定按钮之前，检查并确认设置。*

图 7.5. NetworkManager 文本用户界面完成绑定



[D]

有关绑定术语的定义，请参阅 第 7.8.1.1 节 “配置 Bond 选项卡”。

有关安装 nmtui 的详情请查看 第 3.2 节 “使用 nmtui 配置 IP 网络”。

7.3. 使用 NETWORKMANAGER 命令行工具 NMCLI 进行网络绑定

注意

有关 nmcli 简介，请参阅 第 3.3 节 “使用 nmcli 配置 IP 网络”。

要使用 nmcli 工具创建 绑定 连接，请运行以下命令：

```
~]$ nmcli con add type bond ifname mybond0
Connection 'bond-mybond0' (5f739690-47e8-444b-9620-1895316a28ba) successfully added.
```

请注意，由于绑定未给出任何 con-name，因此连接名称是通过在类型前前从接口名称衍生而来的。

NetworkManager 支持内核提供的大多数绑定选项。例如：

```
~]$ nmcli con add type bond ifname mybond0 bond.options "mode=balance-rr,miimon=100"
Connection 'bond-mybond0' (5f739690-47e8-444b-9620-1895316a28ba) successfully added.
```

添加端口接口：

1. 创建新连接，详情请参阅 第 3.3.5 节 “使用 nmcli 创建和修改连接配置集”。

2. 将 controller 属性设置为 绑定接口 名称，或设置为控制器连接的名称：

```
~]$ nmcli con add type ethernet ifname ens3 master mybond0
Connection 'bond-slave-ens3' (220f99c6-ee0a-42a1-820e-454cbabc2618) successfully added.
```

若要添加新的端口接口，可使用新接口重复上一命令。例如：

```
~]$ nmcli con add type ethernet ifname ens7 master mybond0
Connection 'bond-slave-ens7' (ecc24c75-1c89-401f-90c8-9706531e0231) successfully added.
```

要激活端口，请按如下所示发出命令：

```
~]$ nmcli con up bond-slave-ens7
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/14)
```

```
~]$ nmcli con up bond-slave-ens3
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/15)
```

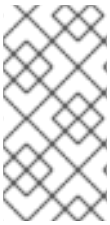
激活端口时，控制器连接也会启动。更多信息请参阅 第 7.1 节 “了解控制器和端口接口的默认行为”。在这种情况下，不需要手动激活控制器连接。

可以在运行时更改 active_slave 选项和绑定的主选项，而不激活连接。例如要更改 active_slave 选项，请使用以下命令：

```
~]$ nmcli dev mod bond0 +bond.options "active_slave=ens7"
Connection successfully reapplied to device 'bond0'.
```

或者更改主选项 ：

```
~]$ nmcli dev mod bond0 +bond.options "primary=ens3"
Connection successfully reapplied to device 'bond0'.
```



注意

active_slave 选项设定当前活动端口，而绑定 的主 选项指定了在添加新端口或发生活跃端口时，内核会自动选择的活动端口。

7.4. 使用命令行界面(CLI)

使用 bonding 内核模块和名为通道绑定接口的特殊网络接口创建绑定。

7.4.1. 检查 Bonding 内核模块是否已安装

在 Red Hat Enterprise Linux 7 中，默认情况下不会加载 bonding 模块。您可以以 root 用户身份运行以下命令来载入该模块：

```
~]# modprobe --first-time bonding
```

此激活在系统重启后不会保留。有关永久模块加载的解释，请参阅Red Hat Enterprise Linux 内核管理指南。请注意，如果配置文件使用了 BONDING_OPTS 指令，则 bonding 模块将根据需要加载，因此不需要单独加载。

要显示模块信息，请运行以下命令：

```
~]$ modinfo bonding
```

有关更多命令选项，请参阅 modprobe(8)手册页。

7.4.2. 创建频道绑定接口

要创建通道绑定接口，请在 /etc/sysconfig/network-scripts/ 目录中创建一个名为 ifcfg-bondN 的文件，将 N 替换为接口的数字，如 0。

文件的内容可以基于绑定的任何接口类型的配置文件，如以太网接口。基本区别在于 DEVICE 指令是 bondN，将 N 替换为接口的编号，而 TYPE=Bond 则替换为 TYPE=Bond。此外，设置 BONDING_MASTER=yes。

例 7.1. ifcfg-bond0 接口配置文件示例

通道绑定接口示例：

```
DEVICE=bond0
NAME=bond0
TYPE=Bond
BONDING_MASTER=yes
IPADDR=192.168.1.1
PREFIX=24
ONBOOT=yes
BOOTPROTO=none
BONDING_OPTS="bonding parameters separated by spaces"
NM_CONTROLLED="no"
```

NAME 指令可用于在 NetworkManager 中命名连接配置集。ONBOOT 说明在引导时是否应该启动配置集（或者更常见的是，在自动连接设备时）。

重要

bonding 内核模块的参数必须指定为 BONDING_OPTS="bonding parameters" 指令（在 ifcfg-bondN 接口文件中）中的空格分隔列表。不要在 /etc/modprobe.d/bonding.conf 中或已弃用的 /etc/modprobe.conf 文件中指定绑定设备的选项。

max_bonds 参数不特定于接口，在使用带有 BONDING_OPTS 指令的 ifcfg-bondN 文件时，不应设置，因为此指令将导致网络脚本根据需要创建绑定接口。

有关配置 bonding 模块和查看绑定参数列表的详情请参考 第 7.7 节 “使用频道绑定”。

请注意，如果没有 NM_CONTROLLED="no" 设置，NetworkManager 可能会覆盖此配置文件中的设置。

7.4.3. 创建端口接口

频道绑定接口是控制器（也称为 主），要绑定的接口被称为端口（从）。创建通道绑定接口后，必须通过在端口的配置文件中添加 MASTER 和 SLAVE 指令来配置要绑定的网络接口。每个端口接口的配置文件几乎相同。

例 7.2. 端口接口配置文件示例

例如，如果两个以太网接口被通道绑定，enp1s0 和 enp2s0，它们可以类似以下示例：

```
DEVICE=device_name
NAME=bond0-slave
TYPE=Ethernet
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
NM_CONTROLLED="no"
```

在本例中，将 device_name 替换为接口的名称。请注意，如果一个接口存在多个配置集或配置文件，则可以通过 ONBOOT=yes 进行接口，它们可能会相互竞争，而普通的 TYPE=Ethernet 配置集可能会被激活，而不是绑定端口。

注意

请注意，如果没有 NM_CONTROLLED="no" 设置，NetworkManager 可能会覆盖此配置文件中的设置。

7.4.4. 激活频道绑定

要激活绑定，请打开所有端口。以 root 身份运行以下命令：

```
~]# ifup ifcfg-enp1s0
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/7)
```

```
~]# ifup ifcfg-enp2s0
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/8)
```

请注意，如果编辑当前为 的接口的接口，请首先将其设置为：

```
ifdown device_name
```

在完成后，打开所有端口，这将打开绑定（如果尚未设置）。“” “”

要让 NetworkManager 知道这些更改，请以 root 用户身份为每个更改的接口发出命令 ：

```
~]# nmcli con load /etc/sysconfig/network-scripts/ifcfg-device
```

另外，重新载入所有接口：

```
~]# nmcli con reload
```

默认行为是网络管理器(NetworkManager)不知晓这些更改并继续使用旧配置数据。这由 NetworkManager.conf 文件中的 monitor-connection-files 选项设置。如需更多信息，请参阅 NetworkManager.conf(5) 手册页。

要查看绑定接口的状态，请运行以下命令：

```
~]# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp1s0: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
master bond0 state UP mode DEFAULT qlen 1000
    link/ether 52:54:00:e9:ce:d2 brd ff:ff:ff:ff:ff:ff
3: enp2s0: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
master bond0 state UP mode DEFAULT qlen 1000
    link/ether 52:54:00:38:a6:4c brd ff:ff:ff:ff:ff:ff
4: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP mode DEFAULT
    link/ether 52:54:00:38:a6:4c brd ff:ff:ff:ff:ff:ff
```

7.4.5. 创建多个绑定

在 Red Hat Enterprise Linux 中，为每个绑定创建一个频道绑定接口，其中包括 BONDING_OPTS 指令。使用这个配置方法以便多个绑定设备可以有不同的配置。要创建多个通道绑定接口，请按如下操作：

- 使用 BONDING_OPTS 指令创建多个 ifcfg-bondN 文件；此指令将导致网络脚本根据需要创建绑定接口。
- 创建或编辑要绑定的现有接口配置文件，并包含 SLAVE 指令。
- 通过 MASTER 指令，将接口绑定到通道绑定接口。

例 7.3. ifcfg-bondN 接口配置文件示例

以下是频道绑定接口配置文件的示例：

```
DEVICE=bondN
NAME=bondN
TYPE=Bond
BONDING_MASTER=yes
IPADDR=192.168.1.1
PREFIX=24
```

```
ONBOOT=yes
BOOTPROTO=none
BONDING_OPTS="bonding parameters separated by spaces"
```

在本例中，将 N 替换为绑定接口的编号。例如，要创建两个绑定，可创建两个配置文件 ifcfg-bond0 和 ifcfg-bond1 （具有适当的 IP 地址 ）：

创建要按 例 7.2 “端口接口配置文件示例” 绑定的接口，并使用 MASTER=bondN 指令将其分配给绑定接口。例如，继续前面的示例，如果每个绑定需要两个接口，那么两个绑定创建了四个接口配置文件，并使用 MASTER=bond0 分配前两个接口，后两个使用 MASTER=bond1。

7.5. 验证冗余的网络配置绑定

如果设备用于备份目的，以防止或恢复特定系统故障，则网络冗余是一个过程。以下流程描述了如何验证冗余中绑定的网络配置：

流程

1. 从绑定接口 Ping 目标 IP。例如：

```
~]# ping -I bond0 DSTADDR
```

2. 查看处于 活跃 模式的接口：

```
~]# cat /sys/class/net/bond0/bonding/active_slave
enp1s0
```

enp1s0 是 活动 端口接口。

3. 设置 活跃 端口接口缩减：

```
~]# ip link set enp1s0 down
```

4. 检查 备份 接口是否已启动：

```
~]# cat /sys/class/net/bond0/bonding/active_slave
enp2s0
```

enp2s0 现在是 活动的 端口接口。

5. 检查您是否仍然可以从绑定接口 ping 目标 IP:

```
~]# ping -I bond0 DSTADDR
```

7.6. 切换中绑定模式和所需设置概述

下表描述了您必须对上游交换机应用的必要配置，具体取决于绑定模式：

表 7.1. 根据绑定模式切换配置设置

绑定模式	Switch 上的配置
0 - balance-rr	需要启用静态的 Etherchannel（未启用 LACP 协商）
1 - active-backup	需要可自主端口
2 - balance-xor	需要启用静态的 Etherchannel（未启用 LACP 协商）
3 - broadcast	需要启用静态的 Etherchannel（未启用 LACP 协商）
4 - 802.3ad	需要启用 LACP 协商的 Etherchannel
5 - balance-tlb	需要可自主端口
6 - balance-alb	需要可自主端口

有关在交换机中配置这些设置，请查看交换机文档。

7.7. 使用频道绑定

要提高性能，请调整可用的模块选项以确定哪个组合最适合。请特别注意 miimon 或 arp_interval 和 arp_ip_target 参数。有关可用选项列表以及如何快速确定绑定接口的最佳选项，请参阅 第 7.7.1 节 “绑定模块指令”。

7.7.1. 绑定模块指令

在将绑定模块参数添加到 BONDING_OPTS="绑定参数" 指令（例如ifcfg-bond0）之前，最好测试哪个通道绑定模块参数 最适合您的绑定接口参数。通过处理 sysfs 文件系统中的文件，可以在不卸载（和重新加载）绑定模块的情况下配置绑定接口的参数。

sysfs 是将内核对象表示为目录、文件和符号链接的虚拟文件系统。sysfs 可用于查询有关内核对象的信息，也可通过使用普通文件系统命令来操控这些对象。sysfs 虚拟文件系统挂载到 /sys/ 目录下。通过与 /sys/class/net/ 目录交互和管理文件，可以动态配置所有绑定接口。

要确定绑定接口的最佳参数，请按照 第 7.4.2 节 “创建频道绑定接口” 中的说明创建通道绑定接口文件，如 ifcfg-bond0。在绑定到 bond 0 的每个接口的配置文件中插入 SLAVE=yes 和 MASTER=bond 0 指令。完成此操作后，您可以继续测试参数。

首先，以 root 用户身份运行 ifup bondN 来打开您创建的绑定：

```
~]# ifup bond0
```

如果您正确创建了 ifcfg-bond0 bonding 接口文件，您可以看到运行 ip 链接的输出中列出的 bond0 以 root 用户身份显示 ：

```
~]# ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp1s0: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master
bond0 state UP mode DEFAULT qlen 1000
    link/ether 52:54:00:e9:ce:d2 brd ff:ff:ff:ff:ff:ff
3: enp2s0: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master
bond0 state UP mode DEFAULT qlen 1000
    link/ether 52:54:00:38:a6:4c brd ff:ff:ff:ff:ff:ff
4: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue state
UP mode DEFAULT
    link/ether 52:54:00:38:a6:4c brd ff:ff:ff:ff:ff:ff
```

要查看所有现有绑定，即使它们没有启动，请运行：

```
~]$ cat /sys/class/net/bonding_masters
bond0
```

您可以通过操作位于 /sys/class/net/bondN/bonding/ 目录中的文件来单独配置每个绑定。首先，您要配置的绑定必须关闭：

```
~]# ifdown bond0
```

例如，要在 bond0 上以 1 秒间隔启用 MII 监控，以 root 用户身份运行：

```
~]# echo 1000 > /sys/class/net/bond0/bonding/miimon
```

要为 balance-alb 模式配置 bond0，请运行：

```
~]# echo 6 > /sys/class/net/bond0/bonding/mode
```

...or，使用模式名称：

```
~]# echo balance-alb > /sys/class/net/bond0/bonding/mode
```

在为相关的绑定配置选项后，您可以通过运行 ifup bondN 来启动并对其进行测试。如果您决定更改选项，请关闭接口，使用 sysfs 修改其参数，将其备份并重新测试。

为绑定确定最佳参数集后，将这些参数作为空格分隔的列表添加到您要配置的绑定接口的 /etc/sysconfig/network-scripts/ifcfg-bondN 文件的 BONDING_OPTS= 指令中。每当该绑定被启动（例如，当设置了 ONBOOT=yes 指令时，系统在引导序列中，在 BONDING_OPTS 中指定的绑定选项将会对那个绑定生效）。

以下列表提供了许多更常用的通道绑定参数的名称，以及它们的操作说明。如需更多信息，请参阅 modinfo 绑定 输出中每个 parm 的简短描述，或者参阅 https://www.kernel.org/doc/Documentation/networking/bonding.txt。

绑定接口参数

ad_select=value

指定要使用的 802.3ad 聚合选择逻辑。可能的值有：

-

stable 或者 0 - 默认设置。活动聚合器由最大的聚合带宽来选择。只有在活跃聚合器的所有端口都停机或活跃聚合器没有端口时，才会重新选择活跃的聚合器。

- **bandwidth 或者 1 - 活跃的聚合器由最大聚合带宽选择。在以下情况下进行重新选择：**
  - **在绑定中添加或删除了端口；**
  - **任何端口的链接状态更改；**
  - **任何端口的 802.3ad 关联状态更改；**
  - **绑定的管理状态变为 up。**
- **count 或者 2 - 活跃的聚合器由最大端口数量选择。重新选择按照上述带宽设置所述进行。**

当活跃聚合器发生部分故障时，带宽和 计数选择策略允许故障转移 802.3ad 聚合。这会使聚合器具有最高的可用性，可以是带宽或端口数量，随时激活。

arp_interval=time_in_milliseconds

以毫秒为单位指定 ARP 监控的频率。

重要

务必要同时指定 arp_interval 和 arp_ip_target 参数，或者指定 miimon 参数。否则可能会导致链接失败时的网络性能下降。

如果在 mode =0 或 mode =2（ 两种负载平衡模式）中使用这个设置，网络交换机必须配置为在 NIC 间平均分发数据包。有关如何完成此操作的详情请参考

https://www.kernel.org/doc/Documentation/networking/bonding.txt。

该值默认设置为 0，它会禁用它。

arp_ip_target=ip_address[,ip_address_2,...ip_address_16]

在启用 arp_interval 参数时，指定 ARP 请求 的目标 IP 地址。最多可在逗号分隔列表中指定最多 16 个 IP 地址。

arp_validate=value

验证 ARP 探测的来源/分发；默认为 none。其他有效值包括 active、backup 和 all。

downdelay=time_in_milliseconds

指定在禁用链接失败前需要等待多久（以毫秒为单位）。该值必须是 miimon 参数中指定的值的倍数。该值默认设置为 0，它会禁用它。

fail_over_mac=value

指定 active-backup 模式是否应该在分配时（传统行为）点把所有端口设置为相同的 MAC 地址（启用时），或者根据所选策略执行绑定 MAC 地址的特殊处理。可能的值有：

- none 或者 0 - 默认设置。此设置禁用 fail_over_mac，并使绑定在分配时将 active-backup 绑定的所有端口设置为同一 MAC 地址。
- active 或者 1 - “活跃的” fail_over_mac 策略表示绑定的 MAC 地址应该始终是当前活跃端口的 MAC 地址。端口的 MAC 地址没有改变，而是在故障切换过程中更改绑定的 MAC 地址。

  此策略对无法更改 MAC 地址的设备或拒绝传入广播的设备和自己的源 MAC（会干扰 ARP 监控器）的设备很有用。此策略的缺点在于，网络中的每个设备都必须通过粒度 ARP 更新，而非检测传入流量以更新其 ARP 表的正常交换机方法。如果发行版 ARP 丢失，通信可能会中断。

  当此策略与 MII 监控器结合使用时，在实际传输和接收之前确认链接的设备特别容易遭

受无饱和性 ARP 的损失，并且可能需要适当的延迟设置。

- follow 或者 2 - 以下 fail_over_mac 策略会导致正常选择绑定的 MAC 地址（通常是添加到绑定的第一个端口的 MAC 地址）。“”但是，第二个和后续端口没有在备份角色中被设置为这个 MAC 地址；一个端口在故障转移时被编程为带有绑定的 MAC 地址（而前者活跃端口接收新活跃端口的 MAC 地址）。

  此策略对多端口设备很有用，当多个端口被编程为相同的 MAC 地址时，这些设备会变得混乱或造成性能损失。

lacp_rate=value

指定链接合作伙伴应在 802.3ad 模式下传输 LACPDU 数据包的速率。可能的值有：

- slow 或者 0 - 默认设置。这规定合作伙伴应每 30 秒传输一次 LACPDU。
- fast 或 1 - 指定合作伙伴应每 1 秒传输一次 LACPDU。

miimon=time_in_milliseconds

指定 MII 链接监控的频率（以毫秒为单位）。如果需要高可用性，这很有用，因为 MII 用于验证 NIC 是否活跃。要验证特定 NIC 的驱动程序是否支持 MII 工具，以 root 用户身份输入以下命令：

```
~]# ethtool interface_name | grep "Link detected:"
```

在这个命令中，将 interface_name 替换为设备接口的名称，如 enp1s0，而不是绑定接口。如果支持 MII，命令会返回：

```
Link detected: yes
```

如果使用绑定接口来实现高可用性，则每个 NIC 的模块必须支持 MII。将值设置为 0 （默认值），关闭此功能。在配置此设置时，此参数的良好起点是 100。

重要

务必要同时指定 arp_interval 和 arp_ip_target 参数，或者指定 miimon 参数。否则可能会导致链接失败时的网络性能下降。

mode=值

允许您指定绑定策略。该值可以是 ：

- balance-rr 或 0 - 为容错和负载均衡设置循环策略。每个绑定接口中按顺序接收和发送传输时，从第一个可用端口接口开始。
- active-backup 或 1 - 为容错设置主动备份策略。通过第一个可用的绑定接口接收和发送传输。只有在活跃的绑定端口接口失败时，才会使用另一个绑定的端口接口。
- balance-xor 或者 2 - 传输基于所选的 hash 策略。默认值为根据源的 XOR 和目的地 MAC 地址生成哈希值，以端口接口的模数乘以模数。在此模式中，目标为特定同级的通信始终将通过同一个接口发送。由于目的地由 MAC 地址决定，此方法最适合发送到同一链路或本地网络上同级设备的通信。如果流量必须穿过单个路由器，这种流量均衡模式将不理想。
- broadcast 或 3 - 为容错设置广播策略。所有传输都将在所有端口接口上发送。
- 802.3ad 或 4 - 设置 IEEE 802.3ad 动态链路聚合策略。创建共享相同速度和双工设置的聚合组。在活跃的聚合器中的所有端口上传输并接收接收。需要兼容 802.3ad 的交换机。
- balance-tlb 或 5 - 为容错和负载均衡设置传输负载平衡(TLB)策略。传出流量会根据每个端口接口的当前负载进行分发。传入流量由当前端口接收。如果接收端口失败，则另一个端口通过失败的端口的 MAC 地址。此模式仅适用于内核绑定模块已知的本地地址，因此无法与虚拟机在桥接后面使用。
- balance-alb 或者 6 - 为容错和负载均衡设置自适应负载均衡(ALB)策略。包括 IPv4 流量的传输和接收负载平衡。通过 ARP 协商实现负载平衡。此模式仅适用于内核绑定模块已知的本地地址，因此无法与虚拟机在桥接后面使用。

有关上游交换机上所需设置的详情，请参考 第 7.6 节 “切换中绑定模式和所需设置概述”。

primary=interface_name

指定主设备的接口名称，如 enp1s0。主 设备是要使用的绑定接口中的第一个，除非失败，否则不会被恢复。当绑定接口中的一个 NIC 速度更快，因此能够处理更大的负载时，此设置特别有用。

这个设置只有在绑定接口处于 active-backup 模式时才有效。如需更多信息，请参阅 https://www.kernel.org/doc/Documentation/networking/bonding.txthttps://www.kernel.org/doc/Documentation/networking/bonding.txt。

primary_reselect=value

指定主端口的 reselection 策略。这会影响选择在主端口故障或恢复主端口时如何成为活动端口的激活端口。这个参数用于防止在主端口和其他端口之间进行软盘。可能的值有：

- *always 或者 0 （默认）- 主端口在备份时都会变为活跃端口。*
- *better 或者 1 - 当主端口恢复时，主端口成为活跃端口，如果主端口的速度和双工比当前活跃端口的速度和 duplex 更好。*
- *failure 或者 2 - 只有当前活跃端口失败且主端口启动时才会激活端口。*

在两种情况下，primary_reselect 设置会被忽略：

- *如果没有激活端口，则要恢复的第一个端口就会变为活动端口。*
- *最初分配给绑定时，主端口始终由活动端口组成。*

通过 sysfs 更改 main _reselect 策略将导致根据新策略立即选择最佳活跃端口。根据具体情况，这可能会造成活动端口的更改

resend_igmp=range

指定要在故障转移事件后发布的 IGMP 成员资格报告的数量。故障转移后会立即发布一份成员报告，后续数据包每 200ms 间隔发送一次。

有效范围为 0 到 255 ；默认值为 1。值为 0 可防止为响应故障转移事件发布 IGMP 成员资格报告。

这个选项对于绑定模式 balance-rr （模式 0）、active-backup （模式 1）、balance-tlb （模式 5）和 balance-alb （模式 6）来说非常有用，其中的故障转移可将 IGMP 流量从一个端口切换到另一个端口。因此，必须发布全新的 IGMP 报告以使交换机通过新选定的端口转发传入的 IGMP 流量。

updelay=time_in_milliseconds

指定（以毫秒为单位）在启用链接前等待多长时间。该值必须是 miimon 参数中指定的值的倍数。该值默认设置为 0，它会禁用它。

use_carrier=number

指定 miimon 是否应该使用 MII/ETHTOOL ioctls 或 netif_carrier_ok() 来确定链接状态。netif_carrier_ok() 功能依赖于设备驱动程序使用 netif_carrier_on/off 维护其状态；大多数设备驱动程序都支持这个功能。

MII/ETHTOOL ioctls 工具使用内核中已弃用的调用序列。但是，如果您的设备驱动程序不支持 netif_carrier_on/off，这仍然可以被配置。

有效值为：

- 1 - 默认设置.启用使用 netif_carrier_ok()。
- 0 - 启用使用 MII/ETHTOOL ioctls。

注意

如果绑定接口坚持链接不应启动，则您的网络设备驱动程序可能不支持 netif_carrier_on/off。

xmit_hash_policy=value

选择在 balance-xor 和 802.3ad 模式中用于端口选择的传输哈希策略。可能的值有：

- 0 或者 layer2 - 默认设置。此参数使用硬件 MAC 地址的 XOR 来生成哈希。使用的公式是：

```
(source_MAC_address XOR destination_MAC) MODULO slave_count
```

此算法会将所有流量都放在同一端口上的特定网络对等点上，兼容 802.3ad。

- 1 或者 layer3+4 - 使用上层协议信息（可用时）生成哈希。这允许到特定网络对等的流量跨越多个端口，但单一连接不跨越多个端口。

使用的 TCP 和 UDP 数据包的公式为：

```
((source_port XOR dest_port) XOR
  ((source_IP XOR dest_IP) AND 0xffff)
    MODULO slave_count
```

对于碎片 TCP 或 UDP 数据包以及所有其他 IP 协议流量，将省略源和目标端口信息。对于非IP 流量，公式与 L2 传输哈希策略相同。

该政策旨在模拟某些交换机的行为；特别是，Cisco 使用 PFC2 以及一些 Foundry 和 IBM 产品进行切换。

此策略使用的算法不兼容 802.3ad。

- 2 或 layer2+3 - 使用 layer2 和 layer3 协议信息的组合来生成哈希。

使用硬件 MAC 地址和 IP 地址 XOR 生成哈希。公式是：

```
(((source_IP XOR dest_IP) AND 0xffff) XOR
  ( source_MAC XOR destination_MAC ))
    MODULO slave_count
```

此算法会将所有流量都放在同一端口上的特定网络对等点。对于非IP 流量，公式与 L2 传输哈希策略相同。

此策略旨在提供比 L2 更均衡的流量分布，特别是在需要 L3 网关设备到达大多数目的地的环境中。

这个算法符合 802.3ad。

7.8. 使用 GUI 创建绑定连接

您可以使用 GNOME 控制中心实用程序指示 NetworkManager 从两个或多个 Wired 或 InfiniBand 连接创建 Bond。不需要先创建要绑定的连接。它们可以配置为流程的一部分，以配置绑定。您必须具有可用接口的 MAC 地址，才能完成配置过程。

7.8.1. 建立绑定连接

过程 7.1. 添加新 Bond Connection_Using nm-connection-editor

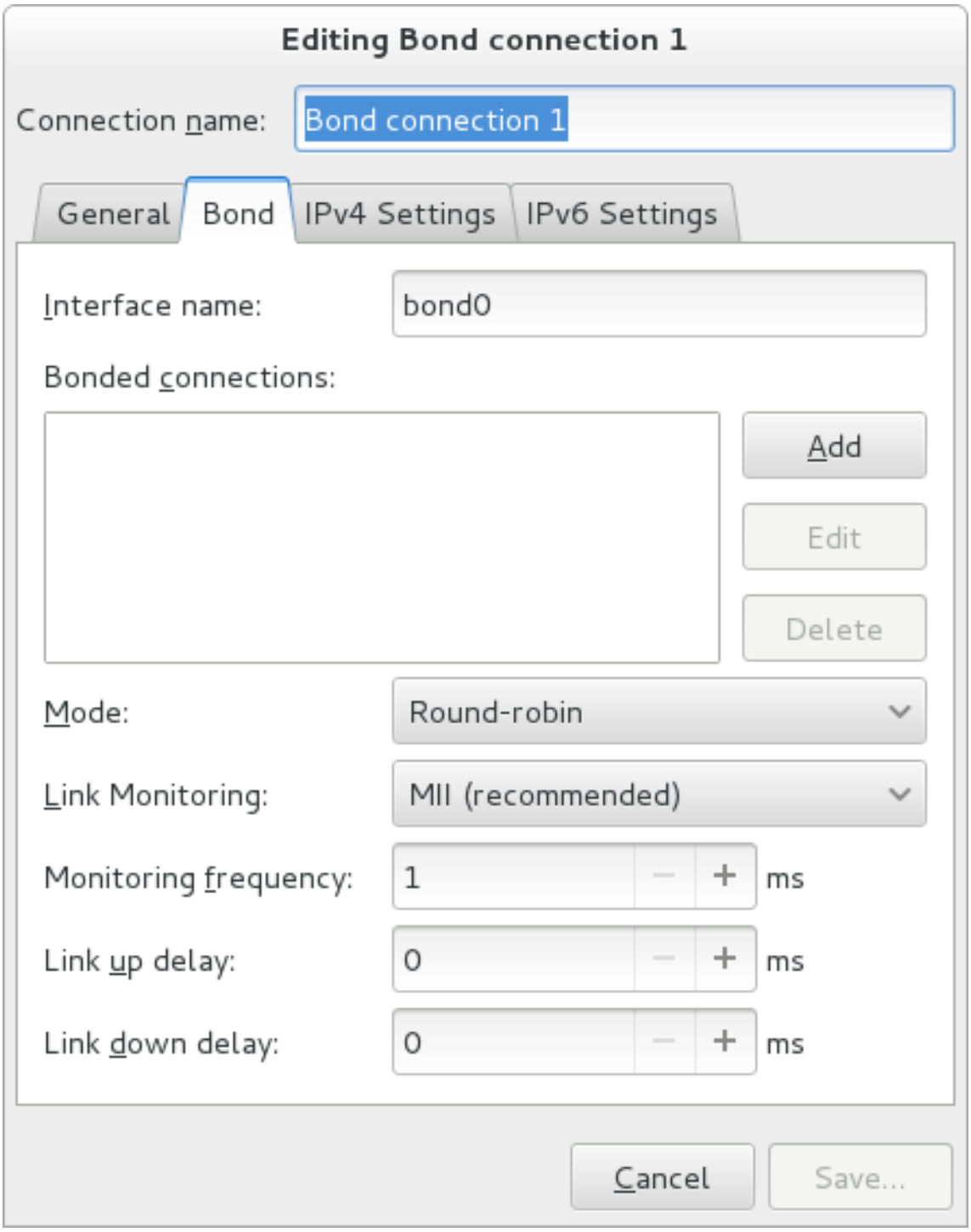
按照以下步骤创建新绑定连接。

1. 在终端中输入 nm-connection-editor ：

```
~]$ nm-connection-editor
```

2. 单击添加按钮。此时将显示 Choose a Connection Type 窗口。选择 Bond，再单击 Create。此时将显示 Editing Bond connection 1 窗口。

图 7.6. NetworkManager 图形用户界面添加 Bond 菜单

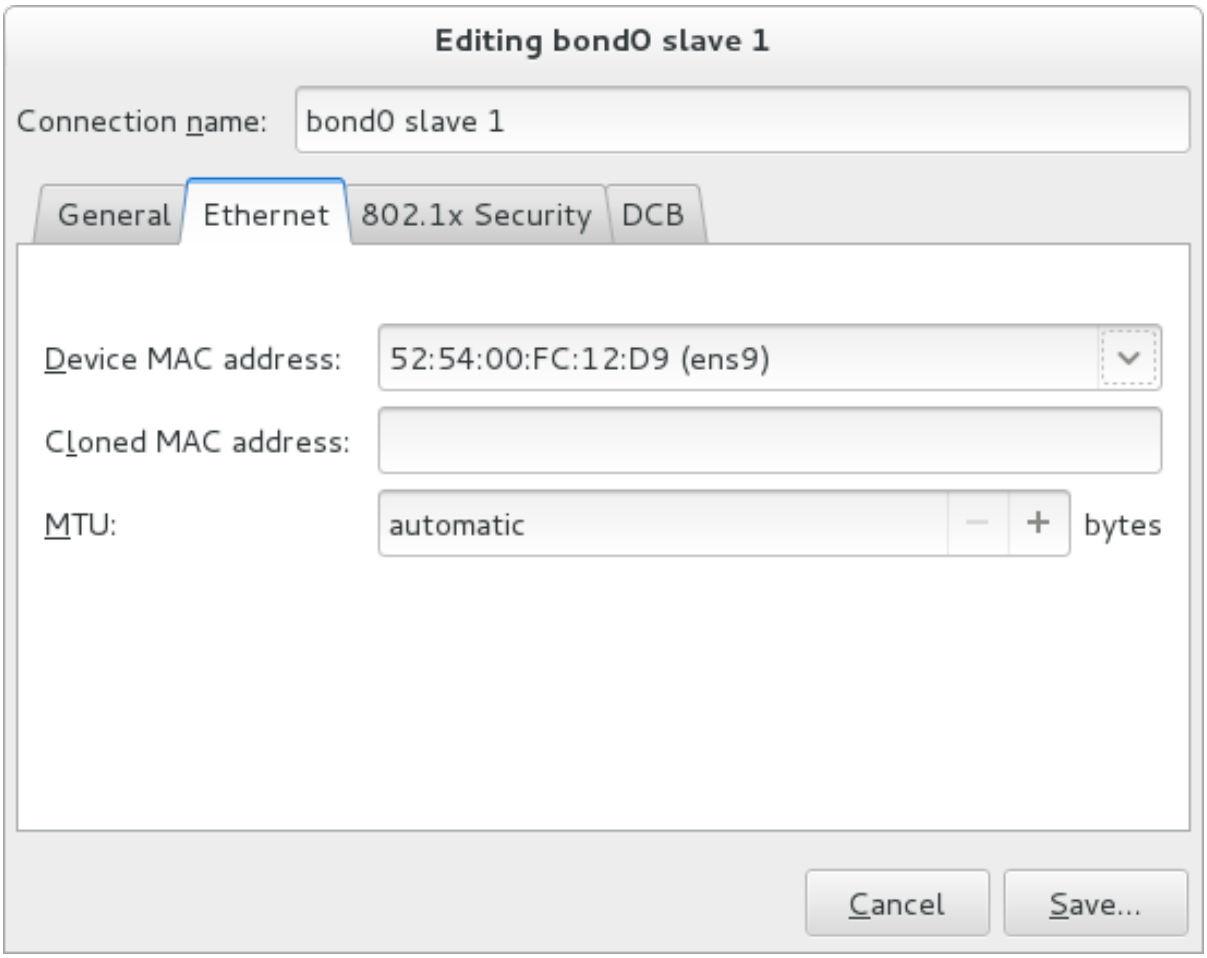


[D]

3. *在 Bond 选项卡中，点 Add 并选择您要用于绑定连接的接口类型。单击创建按钮。请注意，仅当您创建第一个端口时才会出现选择端口类型的对话框；之后，它将自动将同一类型用于所有后续端口。*

4. *此时将显示编辑 bond0 从属 1 窗口。使用设备 MAC 地址下拉菜单选择要绑定的接口的 MAC 地址。第一个端口的 MAC 地址将用作绑定接口的 MAC 地址。如果需要，输入要用作绑定 MAC 地址的克隆 MAC 地址。单击保存按钮。*

图 7.7. NetworkManager 图形用户界面添加 Bond 连接菜单



[D]

5. *绑定端口的名称会出现在绑定连接 窗口中。单击添加按钮以添加更多端口连接。*

6. *检查并确认设置，然后单击保存按钮。*

7. *请参考以下 第 7.8.1.1 节 “配置 Bond 选项卡” 来编辑特定于绑定的设置。*

过程 7.2. 编辑现有的 Bond 连接

按照以下步骤编辑现有绑定连接。

1. *在终端中输入 nm-connection-editor ：*

```
~]$ nm-connection-editor
```

2. 选择您要编辑的连接并点击 Edit 按钮。

3. 选择常规选项卡。

4. 配置连接名称、自动连接行为和可用性设置。

   Editing 对话框中的五个设置对所有连接类型通用，请参阅 General 选项卡：

   - 连接名称 - 输入您的网络连接描述性名称。此名称将用于在 Network 窗口的菜单中列出此连接。
   - 当这个网络可用时自动连接到这个网络 - 如果您希望 NetworkManager 在可用时自动连接到这个连接，请选择这个框。如需更多信息，请参阅 “使用 control-center 编辑现有连接”一节。
   - 所有用户可以连接到此网络 - 选择此框可创建系统上所有用户可用的连接。更改此设置可能需要 root 特权。详情请查看 第 3.4.5 节 “使用 GUI 管理系统范围以及专用连接配置集”。
   - 使用这个连接时自动连接到 VPN - 如果您希望 NetworkManager 在有可用时自动连接到 VPN 连接，请选择这个框。从下拉菜单中选择 VPN。
   - firewall Zone - 从下拉菜单中选择防火墙区域。有关防火墙区域的更多信息，请参阅Red Hat Enterprise Linux 7 安全指南。

5. 请参考以下 第 7.8.1.1 节 “配置 Bond 选项卡” 来编辑特定于绑定的设置。

保存您的新（或修改）连接并创建进一步配置

编辑完绑定连接后，点 Save 按钮保存自定义配置。

然后，配置：

- 连接的 IPv4 设置，单击 IPv4 Settings 选项卡，然后继续 第 5.4 节 “配置 IPv4 设置”

  或者

- 连接的 IPv6 设置，单击 IPv6 Settings 选项卡，再继续 第 5.5 节 “配置 IPv6 设置”。

7.8.1.1. 配置 Bond 选项卡

如果您已添加了新的绑定连接（请参阅 过程 7.1, “添加新 Bond Connection_Using nm-connection-editor”），您可以编辑 Bond 选项卡，以设置负载共享模式和用于检测端口连接故障的链接监控类型。

模式

用来通过组成绑定的端口连接共享流量的模式。默认值为 round-robin。可以通过下拉列表来选择其他负载共享模式，如 802.3ad。

链接监控

监控端口可以传输网络流量的方法。

以下负载共享模式可从 Mode 下拉列表中选择：

round-robin

为容错和负载平衡设置循环策略。每个绑定接口中按顺序接收和发送传输时，从第一个可用端口接口开始。如果没有额外的交换机配置，此模式可能无法与虚拟机桥接之后工作。

活跃备份

设置用于容错的 active-backup 策略。通过第一个可用的绑定接口接收和发送传输。只有在活跃的绑定端口接口失败时，才会使用另一个绑定的端口接口。请注意，这是 InfiniBand 设备绑定的唯一模式。

XOR

设置 XOR（独占-或）策略。传输基于所选哈希策略。默认值为根据源的 XOR 和目的地 MAC 地址生成哈希值，以端口接口的模数乘以模数。在此模式中，目标为特定同级的通信始终将通过同一个接口发送。由于目的地由 MAC 地址决定，此方法最适合发送到同一链路或本地网络上同级设备的通信。如果流量必须穿过单个路由器，这种流量均衡模式将不理想。

broadcast

为容错设置广播策略。所有传输都将在所有端口接口上发送。如果没有额外的交换机配置，此模式可能无法与虚拟机桥接之后工作。

802.3ad

设置 IEEE 802.3ad 动态链路聚合策略。创建共享相同速度和双工设置的聚合组。在活跃的聚合器中的所有端口上传输并接收接收。需要兼容 802.3ad 的网络交换机。

自适应传输负载平衡

设置用于容错和负载平衡的自适应传输负载平衡(TLB)策略。传出流量会根据每个端口接口的当前负载进行分发。传入流量由当前端口接收。如果接收端口失败，则另一个端口通过失败的端口的 MAC 地址。此模式仅适用于内核绑定模块已知的本地地址，因此无法与虚拟机在桥接后面使用。

自适应负载平衡

设置用于容错和负载平衡的自适应负载平衡(ALB)策略。包括 IPv4 流量的传输和接收负载平衡。通过 ARP 协商实现负载平衡。此模式仅适用于内核绑定模块已知的本地地址，因此无法与虚拟机在桥接后面使用。

以下类型的链接监控可以从 Link Monitoring 下拉列表中选择。最好测试哪个通道绑定模块参数最适合您的绑定接口。

MII（媒体独立接口）

接口的载波状态受到监控。这可以通过查询驱动程序、直接查询 MII 寄存器或使用 ethtool 查询设备来完成。有三个可用的选项：

监控频率

查询驱动程序或 MII 寄存器之间的时间间隔，以毫秒为单位。

链接延迟

尝试使用报告为 up 的链接（以毫秒为单位）等待的时间（以毫秒为单位）。如果紧接在报告为 “up” 的链路之后的期间内丢失了一些饱和的 ARP 请求，则可以使用这一延迟。例如，可以在切换初始化时发生这种情况。

链接延迟

当之前活动链接报告为 “down” 时，在更改为另一个链接之前要等待的时间（毫秒）。如果连接的交换机需要相对较长的时间才会更改为备份模式，则可以使用这一延迟。

ARP

地址解析协议(ARP)用于探测一个或多个同级服务器，以确定链路层连接的工作原理。它依赖于提供传输开始时间和上次接收时间的设备驱动程序。

有两个可用的选项：

监控频率

发送 ARP 请求之间的时间间隔，以毫秒为单位。

ARP 目标

用于发送 ARP 请求的 IP 地址的逗号分隔列表。

7.9. 其它资源

安装的文档

- nmcli(1) man page - 描述 NetworkManager 的命令行工具。
- nmcli-examples(5) 手册页 - 提供 nmcli 命令的示例.
- nm-settings(5) 手册页 - NetworkManager 连接的设置和参数的说明.

在线文档

Red Hat Enterprise Linux 系统管理员指南

说明内核模块功能的使用情况。

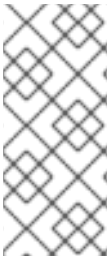
https://access.redhat.com/site/node/28421/Configuring_VLAN_devices_over_a_bonded_interface

有关通过绑定接口配置 VLAN 设备的红帽知识库文章。

第 8 章 配置网络合作

8.1. 了解网络合作

多种名称（如通道绑定、以太网绑定、端口中继、通道合作、NIC团队或链路聚合）的结合或 聚合，以提供吞吐量更高的逻辑链接或提供冗余。这个概念最初在 Linux 内核中实施，广泛称为 绑定。网络合作这一术语已被选为指这个新的概念实施。现有绑定驱动程序不受影响，网络合作作为备选方式提供，不会替代 Red Hat Enterprise Linux 7 中的绑定。



注意

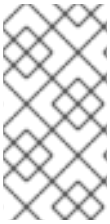
关于模式 4 链路聚合控制协议(LACP)合作模式，需要配置交换机来聚合链接。如需了解更多详细信息，请参阅 https://www.kernel.org/doc/Documentation/networking/bonding.txt

网络合作或团队旨在通过提供小型内核驱动程序来实现数据包流的快速处理，以及各种用户空间应用程序来执行用户空间中的所有其他操作，从而以不同的方式实施该概念。驱动程序具有应用程序编程接口 (API)，称为 “Team Netlink API”，用于实施 Netlink 通信。用户空间应用可以使用此 API 与驱动程序通信。提供了一个称为 “lib” 的库，用于对团队 Netlink 通信和 RT Netlink 消息进行用户空间换行。还提供使用 libteam 库的应用守护进程 teamd。一个 teamd 实例可以控制一个团队驱动程序实例。“守护进程通过使用称为运行程序的额外代码来实施负载平衡和” active-backup 逻辑，如循环。通过以这种方式隔离代码，网络合作实施提供了易于扩展且可扩展的解决方案，以满足负载平衡和冗余要求。例如，自定义运行程序可以相对轻松地编写以通过 teamd 实施新逻辑，即使 teamd 是可选的，用户可以自行编写应用以使用 libteam。

teamdctl 实用程序可用于使用 D-bus 控制正在运行的 teamd 实例。teamdctl 围绕 teamd D-Bus API 提供 D-Bus 打包程序。默认情况下，teamd 使用 Unix 域套接字侦听和通信，但仍监控 D-Bus。这是为了确保 teamd 可用于 D-Bus 不存在或尚未加载的环境中。例如，当通过 teamd 链接引导时，D-Bus 尚未加载。teamdctl 实用程序可在运行时用于读取配置、Link-watchers 状态、检查和更改端口状态、添加和删除端口，以及更改活动状态和备份状态之间的端口。

团队 Netlink API 使用 Netlink 消息与用户空间应用通信。libteam 用户空间库不直接与 API 交互，而是使用 libnl 或 teamnl 与驱动程序 API 交互。

总之，在内核中运行的团队驱动程序实例不会被直接配置或控制。所有配置都借助用户空间应用来完成，如 teamd 应用。然后，应用程序会相应地指示内核驱动程序部分。



注意

在网络合作环境中，术语 端口 也称为 从设备。当 直接使用 teamd 时，首选使用 从 接口，而使用 NetworkManager 来指代创建组的接口。

8.2. 了解控制器和端口接口的默认行为

使用 NetworkManager 守护进程控制合作端口接口时，特别是在查找故障时，请牢记以下几点：

1. 启动控制器接口不会自动启动端口接口。
2. 启动端口接口总会启动控制器接口。
3. 停止控制器接口也会停止端口接口。
4. 没有端口的控制器可以启动静态 IP 连接。
5. 没有端口的控制器在启动 DHCP 连接时会等待端口。
6. 当添加具有载波的端口时，等待端口且具有 DHCP 连接的控制器会完成。
7. 当添加不具有载体的端口时，等待端口且具有 DHCP 连接的控制器将继续等待。

警告

不支持在不使用网络交换机的情况下使用直接电缆连接，不支持合作。如果没有网络交换机，此处描述的故障转移机制将无法按预期工作。如需更多信息，请参阅红帽知识库文章，为什么不支持直接连接的绑定？

8.3. 网络团队与绑定的比较

表 8.1. 绑定和团队功能比较

功能	bonding	Team
广播 Tx 策略	是	是
循环 Tx 策略	是	是
Active-backup Tx 策略	是	是
LACP（802.3ad）支持	是（仅活动）	是
基于 hash 的 Tx 策略	是	是
用户可以设置哈希功能	否	是
TX 负载均衡支持（TLB）	是	是
LACP 哈希端口选择	是	是
LACP 支持的负载均衡	否	是
ethtool 链接监控	是	是
ARP 链路监控	是	是
NS/NA（IPv6）链路监控	否	是
端口启动/关闭延迟	是	是

功能	bonding	Team
“端口优先级和粘性（主要选项增强”）	否	是
单独设置每个端口的链接监控设置	否	是
多个链路监控设置	有限	是
Lockless Tx/Rx 路径	否（rwlock）	是（RCU）
VLAN 支持	是	是
用户空间运行时控制	有限	full
用户空间中的逻辑	否	是
可扩展性	难	易
模块化设计	否	是
性能开销	低	非常低
D-Bus 接口	否	是
多设备堆栈	是	是
使用 LLDP 时零配置	否	（在计划中）
NetworkManager 支持	是	是

8.4. 了解网络合作后台程序和"运行者"

Team 守护进程 teamd 使用 libteam 来控制一个团队驱动程序实例。“这个团队驱动程序实例添加硬件设备驱动程序实例组成一个网络链接组”。团队驱动程序提供网络接口，team0 例如，到内核的其他部分：团队驱动程序实例创建的接口具有名称，如 team0,team1等等。这是为了便于理解，也可以使用其他名称。所有合作方法的常见逻辑都由 teamd “实施；那些特定于不同负载共享和备份方法的功能（如循环）由称为运行程序的独立代码单元实施”。“由于模块和 ‘’ 模式等词语对内核已具有特定含义 ‘’，因此选择了运行程序一词来引用这些代码单元”。用户以 JSON 格式配置文件指定运行程序，然后在创建实例时将代码编译到 teamd 实例中。运行程序不是插件，因为运行程序的代码编译到 teamd 的实例中，因为它在创建之时。如果需要，可作为 teamd 的插件创建。

编写时，可以使用以下运行程序：

- broadcast（数据通过所有端口传输）
- 循环（数据依次在所有端口上传输）
- Active-backup（使用一个端口或链接，而其他端口或链接作为备份）
- 负载均衡（具有活跃的 Tx 负载平衡和基于 BPF 的 Tx 端口选择器）
- LACP（实施 802.3ad 链接聚合控制协议）

另外，还有以下 link-watchers 可用：

- ethtool （Libteam lib 使用 ethtool 监视链接状态变化）。如果配置文件中未指定任何其他 link-watcher，则这是默认设置。
- arp_ping （arp_ping 实用程序用于监控使用 ARP 数据包的远端硬件地址是否存在。）
- nsna_ping （Neighbor Advertisements 和邻居请求来自 IPv6 Neighbor Discovery 协议）用于监控邻居接口的存在性。）

代码中没有阻止特定 link-watcher 与特定运行程序使用的限制，但在使用 lacp 运行程序时，ethtool 是唯一推荐的 link-watcher。

8.5. 安装网络合作守护进程

默认情况下，网络合作守护进程 teamd 不会被安装。要安装 teamd，以 root 用户身份运行以下命令：

```
~]# yum install teamd
```

8.6. 将 BOND 转换为团队

可以使用 bond2team 工具将现有绑定配置文件转换为团队配置文件。它可以将 ifcfg 格式的绑定配置文件转换为 ifcfg 或 JSON 格式的团队配置文件。请注意，重命名后可能会中断防火墙规则、别名接口以及任何可能绑定到原始接口名称的任何内容，因为工具将仅更改 ifcfg 文件，而任何其他内容都不受影响。

要查看命令格式的一些示例，请运行以下命令：

```
~]$ bond2team --examples
```

新文件将在名称以 /tmp/bond2team.XXXXXX/ 开头的目录中创建，其中 XXXXXX 是随机字符串。创建新配置文件后，将旧绑定文件移动到备份文件夹，然后将新文件移到 /etc/sysconfig/network-scripts/ 目录中。

例 8.1. 将 Bond 转换为团队

要将当前 bond0 配置转换为 team ifcfg，以 root 用户身份发出命令：

```
~]# /usr/bin/bond2team --master bond0
```

请注意，这将保留名称 bond0。要使用新名称保存配置，请使用 --rename，如下所示：

```
~]# /usr/bin/bond2team --master bond0 --rename team0
```

添加 --json 选项以输出 JSON 格式文件，而不是 ifcfg 文件。有关 JSON 格式示例，请参阅 teamd.conf(5) man page。

例 8.2. 将 Bond 转换为团队并指定文件路径

要将当前 bond0 配置转换为团队 ifcfg，并手动指定 ifcfg 文件的路径，以 root 身份发出命令：

```
~]# /usr/bin/bond2team --master bond0 --configdir /path/to/ifcfg-file
```

添加 --json 选项以输出 JSON 格式文件，而不是 ifcfg 文件。

例 8.3. 使用 Bond2team 创建团队配置

也可以通过为 bond2team 工具提供绑定参数列表来创建团队配置。例如：

```
~]# /usr/bin/bond2team --bonding_opts "mode=1 miimon=500"
```

您也可以在命令行中提供端口，如下：

```
~]# /usr/bin/bond2team --bonding_opts "mode=1 miimon=500 primary=enp1s0 \
 primary_reselect-0" --port enp1s0 --port enp2s0 --port enp3s0 --port enp4s0
```

详情请查看 bond2team(1) 手册页。有关绑定参数的说明，请参阅 第 7.7 节 “使用频道绑定”

8.7. 选择用作网络组端口的接口

要查看可用的接口，请运行以下命令：

```
~]$ ip link show
1: lo:  <LOOPBACK,UP,LOWER_UP > mtu 65536 qdisc noqueue state UNKNOWN mode
DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: em1:  <BROADCAST,MULTICAST,UP,LOWER_UP > mtu 1500 qdisc pfifo_fast state UP
mode DEFAULT qlen 1000
    link/ether 52:54:00:6a:02:8a brd ff:ff:ff:ff:ff:ff
3: em2:  <BROADCAST,MULTICAST,UP,LOWER_UP > mtu 1500 qdisc pfifo_fast state UP
mode DEFAULT qlen 1000
link/ether 52:54:00:9b:6d:2a brd ff:ff:ff:ff:ff:ff
```

从可用接口中，确定哪些接口适合添加到您的网络团队，然后继续 第 8.8 节 “选择网络团队配置方法”

8.8. 选择网络团队配置方法

要使用 NetworkManager 的文本用户界面工具 nmtui 配置网络组，请继续 第 8.9 节 “使用文本用户界面 nmtui 配置网络团队”

要使用命令行工具nmcli 创建网络团队，可使用 第 8.10.1 节 “使用 nmcli 配置网络合作”。

要使用 Team 守护进程创建网络团队，teamd 将继续 第 8.10.2 节 “使用 teamd 创建网络团队”。

要使用配置文件创建网络团队，请继续 第 8.10.3 节 “使用 ifcfg 文件创建网络团队”。

要使用图形用户界面配置网络团队，请参阅 第 8.14 节 “使用 GUI 创建网络团队”

8.9. 使用文本用户界面 NMTUI 配置网络团队

文本用户界面工具 nmtui 可用于在终端窗口中配置合作。使用以下命令启动该工具：

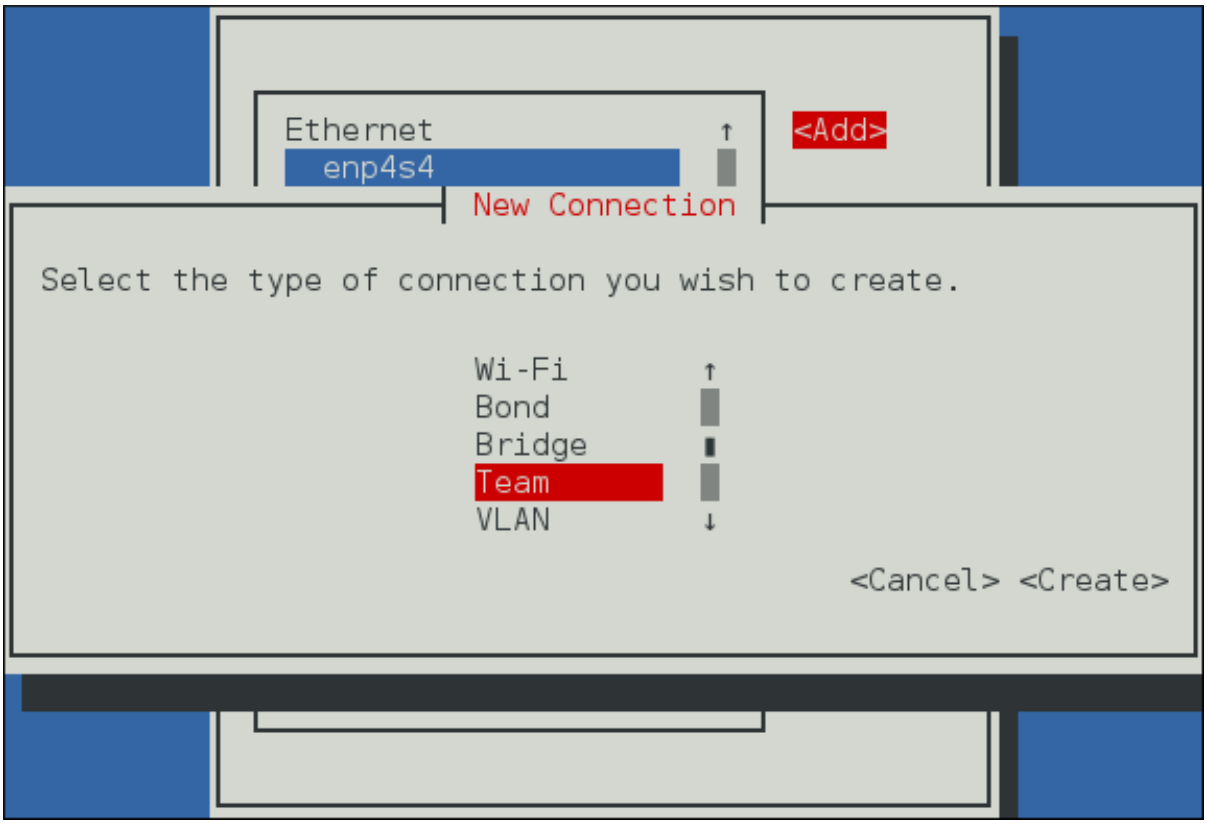
```
~]$ nmtui
```

此时将显示文本用户界面。任何无效的命令都会打印用法消息。

要导航，请使用箭头键或按 Tab 键前进，然后按ShiftTab后退步浏览选项。按 Enter 键选择一个选项。Space bar 切换复选框的状态。

1. 在起始菜单中选择 Edit a connection。选择 Add，这会打开 New Connection 屏幕。

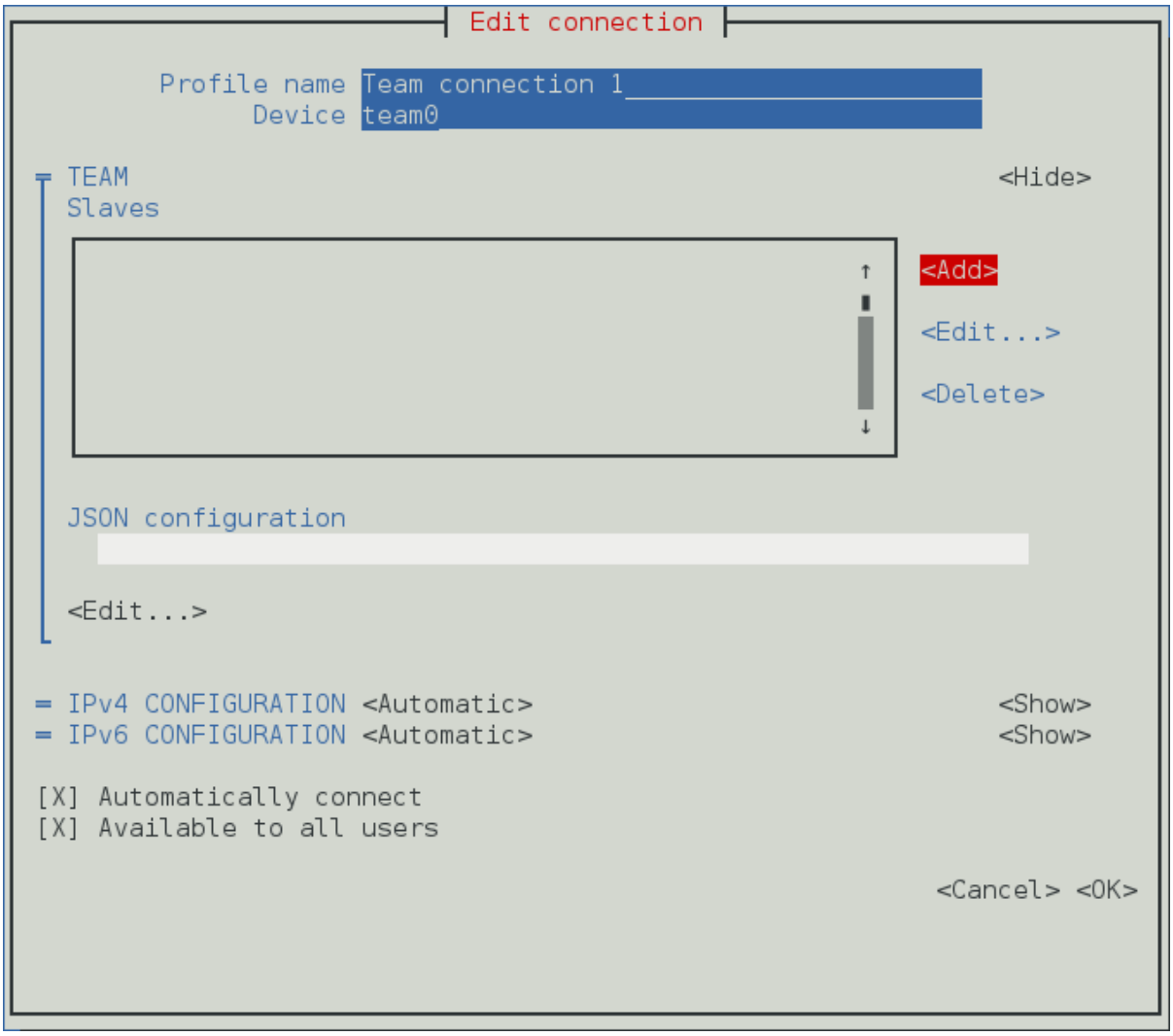
图 8.1. NetworkManager 文本用户界面添加团队连接菜单



[D]

2. *选择 **Team**，打开 **Edit connection** 屏幕。*

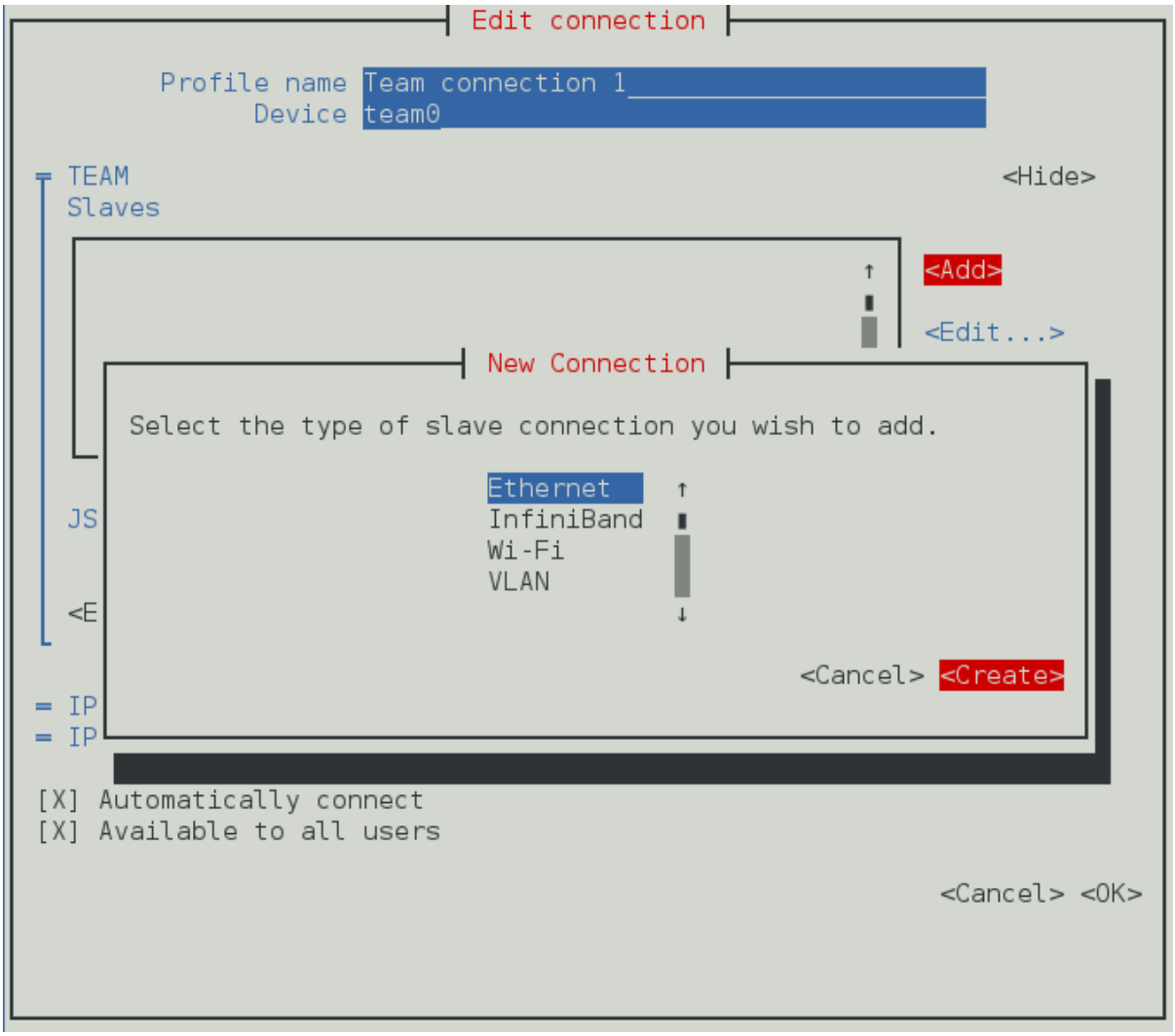
图 8.2. NetworkManager 文本用户界面配置团队连接菜单



[D]

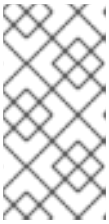
3. *要向团队添加端口接口，请选择 **Add**，这会打开 **New Connection** 屏幕。选择连接类型后，选择创建按钮，使组的 **Edit Connection** 显示出现。*

图 8.3. NetworkManager 文本用户界面配置新团队端口连接菜单



[D]

4. *在 Device 部分输入所需端口的设备名称或 MAC 地址。如果需要，通过选择 Show to the Ethernet 标签右侧输入要用作团队 MAC 地址的克隆 MAC 地址。选择确定按钮。*



注意

如果设备没有 MAC 地址指定，则当 Edit Connection 窗口重新加载后，仅当成功找到该设备时，会自动填充 Device 部分。

图 8.4. NetworkManager 文本用户界面配置团队端口接口连接菜单

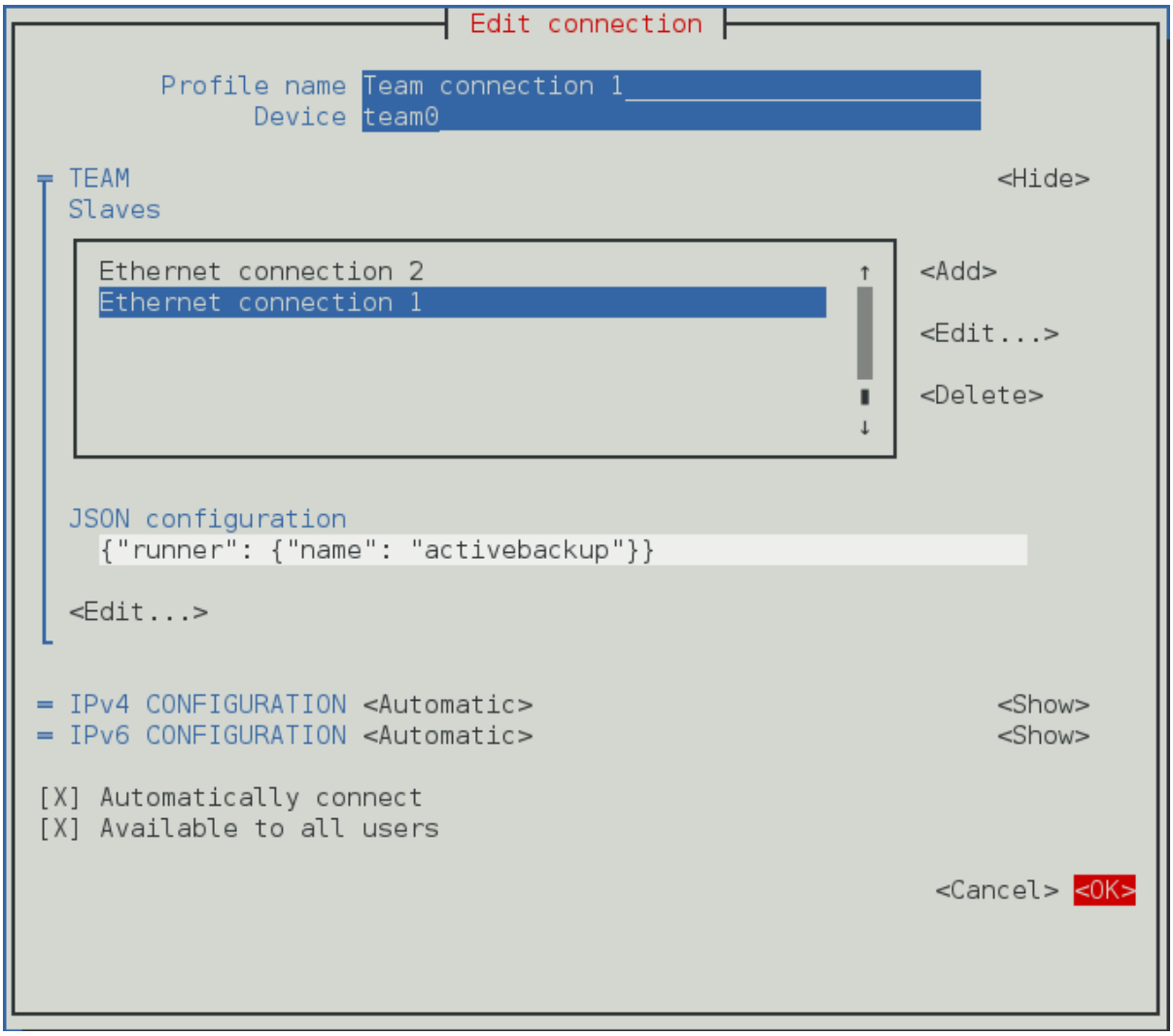
[D]

5. *合作端口的名称会出现在 Slaves 部分中。重复上述步骤，以添加其他端口连接。*

6. *如果要应用自定义端口设置，请选择 JSON 配置部分下的 Edit 按钮。这将启动可以应用更改的 vim 控制台。完成后，从 vim 写入更改，然后确认 JSON 配置下显示的 JSON 字符串是否与预期相符。*

7. *在选择确定按钮之前，检查并确认设置。*

图 8.5. NetworkManager 文本用户界面配置团队连接菜单



[D]

有关 JSON 字符串示例，请参阅 第 8.13 节 “配置 teamd 运行程序”。请注意，应该只使用示例字符串的相关部分使用 nmtui 进行团队或端口配置。不要将 “Device 指定为” JSON 字符串的一部分。例如，在 Team JSON “配置字段中，仅应使用设备之后但 “ 端口前的” JSON 字符串。与端口相关的所有 JSON 字符串仅必须在 port 配置字段中添加。

有关安装 nmtui 的详情请查看 第 3.2 节 “使用 nmtui 配置 IP 网络”。

8.10. 使用命令行配置网络团队

8.10.1. 使用 nmcli 配置网络合作

查看系统中可用的连接：

```
~]$ nmcli connection show
NAME  UUID                                  TYPE          DEVICE
enp2s0  0e8185a1-f0fd-4802-99fb-bedbb31c689b  802-3-ethernet  --
```

```
enp1s0  dfe1f57b-419d-4d1c-aaf5-245deab82487  802-3-ethernet  --
```

查看系统中可用的设备：

```
~]$ nmcli device status
DEVICE     TYPE     STATE                      CONNECTION
virbr0     bridge   connected                  virbr0
ens3       ethernet connected                  ens3
```

要创建新团队接口，名称为 ServerA ：

```
~]$ nmcli connection add type team ifname ServerA
Connection 'team-ServerA' (b954c62f-5fdd-4339-97b0-40efac734c50) successfully added.
```

NetworkManager 会将其内部参数 connection.autoconnect 设为 yes，因为没有为 ipv4.method 指定任何 IP 地址 。NetworkManager 还将配置文件写入 /etc/sysconfig/network-scripts/ifcfg-team-ServerA，其中对应的 ONBOOT 设为 yes， BOOTPROTO 将设置为 dhcp。

请注意，在接口下次启动之前，NetworkManager 不会注意到对 ifcfg 文件的手动更改。有关使用配置文件的更多信息，请参阅 第 2.7 节 “使用 NetworkManager 和 sysconfig 文件”。

查看分配的其他值：

```
~]$ nmcli con show team-ServerA
connection.id:                          team-ServerA
connection.uuid:                        b954c62f-5fdd-4339-97b0-40efac734c50
connection.interface-name:              ServerA
connection.type:                        team
connection.autoconnect:                 yes
...
ipv4.method:                            auto
[output truncated]
```

由于没有指定 JSON 配置文件，因此应用默认值。如需有关团队 JSON 参数及其默认值的更多信息，请参阅 teamd.conf(5) man page。请注意，名称是从接口名称中派生而来的，方法是在类型前附加。或者，使用 con-name 选项指定一个名称，如下所示：

```
~]$ nmcli connection add type team con-name Team0 ifname ServerB
Connection 'Team0' (5f7160a1-09f6-4204-8ff0-6d96a91218a7) successfully added.
```

要查看刚才配置的组接口，请输入以下命令：

```
~]$ nmcli con show
NAME            UUID                                  TYPE           DEVICE
team-ServerA    b954c62f-5fdd-4339-97b0-40efac734c50  team           ServerA
enp2s0          0e8185a1-f0fd-4802-99fb-bedbb31c689b  802-3-ethernet --
enp1s0          dfe1f57b-419d-4d1c-aaf5-245deab82487  802-3-ethernet --
Team0           5f7160a1-09f6-4204-8ff0-6d96a91218a7  team           ServerB
```

要更改分配给团队的名称，以以下格式输入命令：

```
nmcli con mod old-team-name connection.id new-team-name
```

要为已存在的团队加载团队配置文件：

```
nmcli connection modify team-name team.config JSON-config
```

，您可以将团队配置指定为 JSON 字符串，或者提供包含配置的文件名。文件名可以包含路径。在这两种情况下，存储在 team.config 属性中的 JSON 字符串。如果是 JSON 字符串，请在字符串两边使用单引号，并将整个字符串粘贴到命令行。

查看 team.config 属性：

```
nmcli con show team-name | grep team.config
```

设置 team.config 属性时，所有其他团队属性将相应地更新。

也可以更灵活地公开和设置特定团队选项，而无需直接修改对应的 JSON 字符串。您可以使用其他可用的团队属性来逐个将相关的团队选项设置为所需的值。因此，team.config 属性被更新为与新值匹配。

例如，要设置允许指定一个或多个 link -watchers 的 team.link -watchers 属性，以以下格式输入命令：

```
nmcli connection modify team-name team.link-watchers "name=ethtool delay-up=5,
name=nsna_ping target-host=target.host"
```

所需的 link-watchers 是用逗号分开的，并且属于同一 link-watcher 的属性由空格分隔。

要设置 team.runner 和 team.link-watchers 属性，以以下格式输入命令：

```
nmcli connection modify team-name team.runner activebackup team.link-watchers
"name=ethtool delay-up=5, name=nsna_ping target-host=target.host"
```

这等同于将 team.config 属性设置为对应的 JSON 字符串：

```
nmcli connection modify team-name team.config '{"runner": {"name": "activebackup"},
"link_watch": [{"name": "ethtool", "delay_up": 5},{"name": "nsna_ping", "target_host ":
"target.host"}]'
```

要将接口 enp1s0 添加到 Team0，名称为 Team0-port1，请按以下方式发出命令：

```
~]$ nmcli con add type ethernet con-name Team0-port1 ifname enp1s0 slave-type team master
Team0
Connection 'Team0-port1' (ccd87704-c866-459e-8fe7-01b06cf1cffc) successfully added.
```

同样，要添加另一个接口 enp2s0，名称为 Team0-port2，请按如下方式发出命令：

```
~]$ nmcli con add type ethernet con-name Team0-port2 ifname enp2s0 slave-type team master
Team0
Connection 'Team0-port2' (a89ccff8-8202-411e-8ca6-2953b7db52dd) successfully added.
```

nmcli 仅支持以太网端口。

要打开团队，必须首先启动端口：

```
~]$ nmcli connection up Team0-port1
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/2)
```

```
~]$ nmcli connection up Team0-port2
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/3)
```

您可以验证组接口是否通过激活端口激活，如下所示：

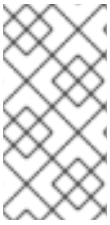
```
~]$ ip link
3:  Team0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
mode DEFAULT
    link/ether 52:54:00:76:6f:f0 brd ff:ff:ff:ff:ff:f
```

另外，发出以下命令以打开团队，如下所示：

```
~]$ nmcli connection up Team0
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/4)
```

nmcli简介请查看 第 3.3 节 “使用 nmcli 配置 IP 网络”

8.10.2. 使用 teamd 创建网络团队



注意

使用 teamd 创建的配置不具有持久性，因此可能需要使用 第 8.10.1 节 “使用 nmcli 配置网络合作” 或 第 8.10.3 节 “使用 ifcfg 文件创建网络团队” 中定义的步骤创建团队。

若要创建网络组，需要 JSON 格式配置文件作为端口组或链接的接口的虚拟接口。快速方法是复制示例配置文件，然后使用以 root 特权运行的编辑器编辑这些文件。要列出可用的示例配置，请输入以下命令：

```
~]$ ls /usr/share/doc/teamd-*/example_configs/
activebackup_arp_ping_1.conf  activebackup_multi_lw_1.conf   loadbalance_2.conf
activebackup_arp_ping_2.conf  activebackup_nsna_ping_1.conf  loadbalance_3.conf
activebackup_ethtool_1.conf   broadcast.conf                 random.conf
activebackup_ethtool_2.conf   lacp_1.conf                    roundrobin_2.conf
activebackup_ethtool_3.conf   loadbalance_1.conf             roundrobin.conf
```

要查看包含的文件之一，如 activebackup_ethtool_1.conf，请输入以下命令：

```
~]$ cat /usr/share/doc/teamd-*/example_configs/activebackup_ethtool_1.conf
{
 "device": "team0",
 "runner": {"name": "activebackup"},
 "link_watch": {"name": "ethtool"},
 "ports": {
  "enp1s0": {
   "prio": -10,
   "sticky": true
  },
  "enp2s0": {
   "prio": 100
  }
 }
}
```

创建用于存储 teamd 配置文件的工作配置目录。例如，作为普通用户，输入带有以下格式的命令：

```
~]$ mkdir ~/teamd_working_configs
```

将您选择的文件复制到您的工作目录中，并根据需要进行编辑。例如，您可以使用带有以下格式的命令：

```
~]$ cp /usr/share/doc/teamd-*/example_configs/activebackup_ethtool_1.conf \
~/teamd_working_configs/activebackup_ethtool_1.conf
```

要编辑该文件以适应您的环境，例如要将接口更改为网络团队的端口，请打开该文件以进行编辑，如下所示：

```
~]$ vi ~/teamd_working_configs/activebackup_ethtool_1.conf
```

进行任何必要的更改并保存文件。有关使用 vi 编辑器或使用您首选编辑器的帮助，请参见 vi (1) 手册页。

“请注意，要用作组中端口的接口不得处于活动状态，也就是说，在将它们添加到组设备时，它们必须关闭”。要检查其状态，请运行以下命令：

```
~]$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode
DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: em1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode
DEFAULT qlen 1000
    link/ether 52:54:00:d5:f7:d4 brd ff:ff:ff:ff:ff:ff
```

```
3: em2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode
DEFAULT qlen 1000
  link/ether 52:54:00:d8:04:70 brd ff:ff:ff:ff:ff:ff
```

在本示例中，我们计划使用的接口都是 “UP”。

要关闭接口，以 root 用户身份以以下格式发出命令：

```
~]# ip link set down em1
```

根据需要为每个接口重复此操作。

要以 root 用户身份基于配置文件创建组接口，请更改到工作配置目录（本例中为teamd_working_configs ）：

```
~]# cd /home/userteamd_working_configs
```

然后以以下格式发出命令：

```
~]# teamd -g -f activebackup_ethtool_1.conf -d
Using team device "team0".
Using PID file "/var/run/teamd/team0.pid"
Using config file "/home/user/teamd_working_configs/activebackup_ethtool_1.conf"
```

g 选项用于调试消息， -f 选项指定要加载的配置文件，而 -d 选项则使进程在启动后作为守护进程运行。有关其他选项，请参阅 teamd(8) 手册页。

要检查团队的状态，以 root 身份运行以下命令：

```
~]# teamdctl team0 state
setup:
  runner: activebackup
ports:
  em1
    link watches:
      link summary: up
      instance[link_watch_0]:
        name: ethtool
        link: up
  em2
    link watches:
      link summary: up
      instance[link_watch_0]:
        name: ethtool
```

```
      link: up
runner:
  active port: em1
```

要将地址应用到网络组接口，team0，以 root 用户身份以以下格式发出命令：

```
~]# ip addr add 192.168.23.2/24 dev team0
```

要检查组接口的 IP 地址，请按如下所示发出命令：

```
~]$ ip addr show team0
4: team0:  <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
    link/ether 16:38:57:60:20:6f brd ff:ff:ff:ff:ff:ff
    inet 192.168.23.2/24 scope global team0
       valid_lft forever preferred_lft forever
    inet6 2620:52:0:221d:1438:57ff:fe60:206f/64 scope global dynamic
       valid_lft 2591880sec preferred_lft 604680sec
    inet6 fe80::1438:57ff:fe60:206f/64 scope link
       valid_lft forever preferred_lft forever
```

“要激活组接口，或者要启动它”，以 root 用户身份以以下格式发出命令：

```
~]# ip link set dev team0 up
```

“要临时停用组接口或将其关闭”，以 root 用户身份以以下格式发出命令：

```
~]# ip link set dev team0 down
```

以 root 用户身份终止或终止团队守护进程实例，以以下格式发出命令：

```
~]# teamd -t team0 -k
```

```
~]# teamd -t team0 -k
```

k 选项指定与设备关联的守护进程实例 team0 将被终止。有关其他选项，请参阅 teamd(8) 手册页。

要获得 teamd 命令行选项的帮助，请运行以下命令：

```
~]$ teamd -h
```

此外，请参阅 teamd(8)man page。

8.10.3. 使用 ifcfg 文件创建网络团队

要使用 ifcfg 文件创建网络团队，请在 /etc/sysconfig/network-scripts/ 目录中创建一个文件，如下所示：

```
DEVICE=team0
DEVICETYPE=Team
ONBOOT=yes
BOOTPROTO=none
IPADDR=192.168.11.1
PREFIX=24
TEAM_CONFIG='{"runner": {"name": "activebackup"}, "link_watch": {"name": "ethtool"}}'
```

这会为团队创建接口，换句话说是主设备。

创建要成为成员的端口 team0，在 /etc/sysconfig/network-scripts/ 目录中创建一个或多个文件，如下所示：

```
DEVICE=enp1s0
HWADDR=D4:85:64:01:46:9E
DEVICETYPE=TeamPort
ONBOOT=yes
TEAM_MASTER=team0
TEAM_PORT_CONFIG='{"prio": 100}'
```

根据需要添加与上述类似的额外端口接口，更改 DEVICE 和 HWADDR 字段以匹配正在添加的端口（网络设备）。如果未由 prio 指定端口优先级，则默认为 0 ；它接受范围 -32,767 到 +32,767 的负值和正数。

使用HWADDR 指令指定硬件或 MAC 地址将影响设备命名过程。这在 第 11 章 一致的网络设备命名 中进行了解释。

要打开网络团队，以 root 用户身份运行以下命令：

```
~]# ifup team0
```

要查看网络团队，请运行以下命令：

```
~]$ ip link show
```

8.10.4. 使用 iputils 向网络团队添加端口

添加端口 em1 到网络团队 team0请使用 ip 实用程序以 root 用户身份运行以下命令：

```
~]# ip link set dev em1 down
~]# ip link set dev em1 master team0
```

根据需要添加其他端口。组驱动程序将自动启动端口。

8.10.5. 使用 teamnl 列出团队的端口

要使用 teamnl 工具查看或列出网络组中的端口，以 root 用户身份运行以下命令：

```
~]# teamnl team0 ports
em2: up 100 fullduplex
em1: up 100 fullduplex
```

8.10.6. 使用 teamnl 配置团队选项

要使用 teamnl 工具查看或列出所有当前可用的选项，以 root 用户身份运行以下命令：

```
~]# teamnl team0 options
```

要将团队配置为使用活跃备份模式，以 root 用户身份运行以下命令：

```
~]# teamnl team0 setoption mode activebackup
```

8.10.7. 使用 iputils 为网络团队添加地址

要为团队添加地址，请执行以下操作： team0请使用 ip 实用程序以 root 用户身份运行以下命令：

```
~]# ip addr add 192.168.252.2/24 dev team0
```

8.10.8. 使用 iputils 打开网络团队的接口

“要激活或打开网络组的接口”，team0请使用 ip 实用程序以 root 用户身份运行以下命令：

```
~]# ip link set team0 up
```

8.10.9. 使用 teamnl 查看团队的活跃端口选项

要使用 teamnl 程序查看或列出网络组中的 activeport 选项，以 root 用户身份运行以下命令：

```
~]# teamnl team0 getoption activeport
0
```

8.10.10. 使用 teamnl 设置团队的活跃端口选项

要使用 teamnl 工具在网络团队中设置 activeport 选项，以 root 用户身份运行以下命令：

```
~]# teamnl team0 setoption activeport 5
```

要检查团队端口选项中的更改，以 root 用户身份运行以下命令：

```
~]# teamnl team0 getoption activeport
5
```

8.11. 使用 TEAMDCTL 控制 TEAMD

要查询正在运行的 teamd 实例以获取统计信息或配置信息，或者进行更改，需要使用控制工具 teamdctl。

查看团队的当前团队状态 team0,以 root 用户身份输入以下命令：

```
~]# teamdctl team0 state view
```

要获得更详细的输出：

```
~]# teamdctl team0 state view -v
```

获取 JSON 格式的完整状态转储（可用于机器处理） team0使用以下命令：

```
~]# teamdctl team0 state dump
```

用于 JSON 格式的配置转储： team0使用以下命令：

```
~]# teamdctl team0 config dump
```

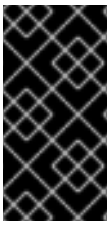
查看端口的配置 em1这是团队的一部分 team0输入以下命令：

```
~]# teamdctl team0 port config dump em1
```

8.11.1. 添加端口到网络团队

添加端口 em1 到网络团队 team0以 root 身份运行以下命令：

```
~]# teamdctl team0 port add em1
```



重要

如果直接使用 teamdctl 添加端口，则必须将端口设为 down。否则，teamdctl team0 端口添加 em1 命令将失败。

8.11.2. 从网络团队中删除端口

要删除接口 em1 来自网络团队 team0以 root 身份运行以下命令：

```
~]# teamdctl team0 port remove em1
```

8.11.3. 将配置应用到网络组中的端口

要将 JSON 格式配置应用到端口 em1 在网络团队中 team0，以 root 用户身份以以下格式发出命令：

```
~]# teamdctl team0 port config update em1 JSON-config-string
```

其中 JSON-config-string 是 JSON 格式作为文本字符串的配置。这将使用提供的 JSON 格式字符串更新端口的配置。配置端口的有效 JSON 字符串示例如下：

```
{
  "prio": -10,
  "sticky": true
}
```

在 JSON 配置字符串两边使用单引号，并省略换行符。

请注意，旧配置将被覆盖，省略的任何选项都将重置为默认值。有关更多团队守护进程控制工具命令示例，请参阅 teamdctl(8)手册页。

8.11.4. 查看网络团队中的端口配置

复制端口的配置 em1 在网络团队中 team0以 root 身份运行以下命令：

```
~]# teamdctl team0 port config dump em1
```

这会将端口的 JSON 格式配置转储到标准输出。

8.12. 验证冗余的网络配置合作

如果设备用于备份目的，以防止或恢复特定系统故障，则网络冗余是一个过程。以下流程描述了如何验证冗余团队的网络配置：

流程

1. 从组接口 Ping 目标 IP。例如：

```
~]# ping -I team0 DSTADDR
```

2. 查看处于 活跃 模式的接口：

```
~]#  teamdctl team0 state
setup:
  runner: activebackup
ports:
  enp1s0
    link watches:
      link summary: up
      instance[link_watch_0]:
        name: ethtool
        link: up
        down count: 0
  enp2s0
    link watches:
      link summary: up
      instance[link_watch_0]:
        name: ethtool
        link: up
        down count: 0
runner:
  active port: enp1s0
```

enp1s0 是 活动 接口。

3. 设置 活跃 端口接口缩减：

```
~]# ip link set enp1s0 down
```

4. 检查 备份 接口是否已启动：

```
~]#  teamdctl team0 state
setup:
 runner: activebackup
ports:
 enp1s0
   link watches:
     link summary: down
     instance[link_watch_0]:
       name: ethtool
       link: down
       down count: 1
 enp2s0
   link watches:
     link summary: up
     instance[link_watch_0]:
       name: ethtool
       link: up
       down count: 0
runner:
 active port: enp2s0
```

enp2s0 现在是 活动 接口。

5. 检查您是否仍然可以从团队接口 ping 目标 IP:

```
~]# ping -I team0 DSTADDR
```

8.13. 配置 TEAMD 运行程序

运行程序是代码单元，在创建守护进程实例时编译到团队守护进程中。有关 teamd 运行程序介绍，请参阅 第 8.4 节 “了解网络合作后台程序和"运行者""。

8.13.1. 配置广播运行程序

要配置广播运行程序，以 root 用户身份使用编辑器，将以下内容添加到团队 JSON 格式配置文件中：

```
{
 "device": "team0",
 "runner": {"name": "broadcast"},
 "ports": {"em1": {}, "em2": {}}
}
```

有关详细信息，请参阅 teamd.conf(5) 手册页。

8.13.2. 配置随机运行程序

随机运行程序的行为与循环运行程序相似。

要配置随机运行程序，以 root 用户身份使用编辑器，将以下内容添加到团队 JSON 格式配置文件中：

```
{
 "device": "team0",
 "runner": {"name": "random"},
 "ports": {"em1": {}, "em2": {}}
}
```

有关详细信息，请参阅 teamd.conf(5) 手册页。

8.13.3. 配置 round-robin Runner

要配置循环运行程序，以 root 用户身份使用编辑器，将以下内容添加到团队 JSON 格式配置文件中：

```
{
 "device": "team0",
```

```
 "runner": {"name": "roundrobin"},
 "ports": {"em1": {}, "em2": {}}
}
```

循环的非常基本配置.

有关详细信息，请参阅 teamd.conf(5) 手册页。

8.13.4. 配置 activebackup Runner

活动备份运行程序可以使用所有 link-watchers 来确定团队中链接的状态。以下示例中的任何一个都可以添加到团队 JSON 格式配置文件中：

```
{
  "device": "team0",
  "runner": {
    "name": "activebackup"
  },
  "link_watch": {
    "name": "ethtool"
  },
  "ports": {
    "em1": {
      "prio": -10,
      "sticky": true
    },
    "em2": {
      "prio": 100
    }
  }
}
```

这个示例配置使用以 ethtool 作为链路监视器的 active-backup 运行程序。端口 em2 具有更高优先级。sticky 标志可确保如果 em1 变为活动状态，只要链接保持启动，它就会保持活动状态。

```
{
  "device": "team0",
  "runner": {
    "name": "activebackup"
  },
  "link_watch": {
```

```
    "name": "ethtool"
  },
  "ports": {
    "em1": {
      "prio": -10,
      "sticky": true,
      "queue_id": 4
    },
    "em2": {
      "prio": 100
    }
  }
}
```

这个示例配置添加了队列 ID 4。它使用 active-backup 运行程序，并将 ethtool 用作链路监视器。端口 em2 具有更高优先级。但是粘滞标志可确保如果 em1 变为活动状态，只要链接保持启动，它就会保持活动状态。

要使用 ethtool 作为链接监视器配置 activebackup 运行程序并应用延迟，以 root 用户身份使用编辑器，将以下内容添加到团队 JSON 格式配置文件中：

```
{
  "device": "team0",
  "runner": {
    "name": "activebackup"
  },
  "link_watch": {
    "name": "ethtool",
    "delay_up": 2500,
    "delay_down": 1000
  },
  "ports": {
    "em1": {
      "prio": -10,
      "sticky": true
    },
    "em2": {
      "prio": 100
    }
  }
}
```

这个示例配置使用以 ethtool 作为链路监视器的 active-backup 运行程序。端口 em2 具有更高优先级。但是粘滞标志可确保如果 em1 变为活动状态，它在链接保持启动时保持活动状态。链接更改不会立即传播到运行程序，但会应用延迟。

有关详细信息，请参阅 teamd.conf(5) 手册页。

8.13.5. 配置 loadbalance Runner

此运行程序可用于两种类型的负载平衡，即主动和被动。在活动模式中，流量的持续重新平衡通过利用最近流量的统计信息尽可能均匀地共享流量来实现。在被动模式中，流量流在可用的链路上随机分布。由于处理开销降低，因此这具有速度优势。在高容量流量应用中，这通常是首选的，因为流量通常由多个流组成，该流将在可用链接之间随机分发，从而无需 teamd 干预即可完成负载共享。

要配置负载均衡器运行程序以进行被动传输(Tx)负载平衡，以 root 用户身份使用编辑器，将以下内容添加到团队 JSON 格式配置文件中：

```
{
 "device": "team0",
 "runner": {
   "name": "loadbalance",
   "tx_hash": ["eth", "ipv4", "ipv6"]
 },
 "ports": {"em1": {}, "em2": {}}
}
```

配置基于哈希的被动传输(Tx)负载平衡.

要配置负载均衡器运行程序以进行活跃传输(Tx)负载平衡，以 root 用户身份使用编辑器，将以下内容添加到团队 JSON 格式配置文件中：

```
{
   "device": "team0",
   "runner": {
     "name": "loadbalance",
     "tx_hash": ["eth", "ipv4", "ipv6"],
     "tx_balancer": {
       "name": "basic"
     }
   },
   "ports": {"em1": {}, "em2": {}}
}
```

使用基本负载平衡器进行活动传输(Tx)负载平衡的配置。

有关详细信息，请参阅 teamd.conf(5) 手册页。

8.13.6. 配置 LACP(802.3ad)运行程序

要使用 ethtool 作为链接监视器配置 LACP 运行程序，以 root 用户身份使用编辑器，将以下内容添加到团队 JSON 格式配置文件中：

```
{
  "device": "team0",
  "runner": {
     "name": "lacp",
     "active": true,
     "fast_rate": true,
     "tx_hash": ["eth", "ipv4", "ipv6"]
  },
    "link_watch": {"name": "ethtool"},
    "ports": {"em1": {}, "em2": {}}
}
```

用于连接到支持链路聚合控制协议(LACP)的对应对象的连接配置。LACP 运行程序应使用 ethtool 来监控链接的状态。请注意，只有 ethtool 可用于链路监控，因为如果是 arp_ping，链接永远不会出现。原因在于，必须首先建立链接，且只能在数据包（包括 ARP）之后才能进行。使用 ethtool 可以防止这种情况，因为它单独监控每个链路层。

此运行程序可以采用与针对 loadbalance 运行程序相同的方式进行主动负载平衡。要启用活跃传输(Tx)负载均衡，请添加以下部分：

```
"tx_balancer": {
     "name": "basic"
}
```

有关详细信息，请参阅 teamd.conf(5) 手册页。

8.13.7. 配置链路状态的监控

可用的链路状态监控方法如下：若要实施其中一种方法，请使用以 root 特权运行的编辑器，将 JSON 格式字符串添加到团队 JSON 格式配置文件。

8.13.7.1. 配置 Ethtool 进行链接状态监控

要在即将推出的链接和运行程序之间添加或编辑现有延迟，请按如下所示添加或编辑部分：

```
"link_watch": {
     "name": "ethtool",
     "delay_up": 2500
}
```

要在关闭的链接和运行程序之间添加或编辑现有延迟，请按如下方式添加或编辑部分：

```
"link_watch": {
      "name": "ethtool",
      "delay_down": 1000
}
```

8.13.7.2. 为链路状态监控配置 ARP Ping

组守护进程 teamd 将 ARP REQUEST 发送到链路远程末尾的地址，以确定链接是否已启动。使用的方法与 arping 实用程序相同，但它不使用该实用程序。

准备包含 JSON 格式的新配置的文件，如下例所示：

```
{
      "device": "team0",
      "runner": {"name": "activebackup"},
      "link_watch":{
         "name": "arp_ping",
         "interval": 100,
         "missed_max": 30,
         "source_host": "192.168.23.2",
         "target_host": "192.168.23.1"
      },
        "ports": {
           "em1": {
             "prio": -10,
             "sticky": true
           },
           "em2": {
             "prio": 100
           }
       }
}
```

此配置使用 arp_ping 作为链路监视器。missing_max 选项是允许最多允许回复数的限值（例如，ARP 回复）。它应当与 interval 选项一同选择，以确定链接报告为 down 前的总时间。

要加载组端口的新配置 em2在包含 JSON 配置的文件中，以 root 用户身份运行以下命令：

```
~]# teamdctl port config update em2 JSON-config-file
```

请注意，旧配置将被覆盖，省略的任何选项都将重置为默认值。有关更多团队守护进程控制工具命令示例，请参阅 teamdctl(8)手册页。

8.13.7.3. 为 Link-state Monitoring 配置 IPv6 NA/NS

```
{
  "device": "team0",
    "runner": {"name": "activebackup"},
    "link_watch": {
    "name": "nsna_ping",
    "interval": 200,
    "missed_max": 15,
    "target_host": "fe80::210:18ff:feaa:bbcc"
    },
      "ports": {
        "em1": {
          "prio": -10,
          "sticky": true
        },
        "em2": {
          "prio": 100
        }
      }
}
```

要配置发送 NS/NA 数据包之间的间隔，请添加或编辑部分，如下所示：

```
"link_watch": {
    "name": "nsna_ping",
    "interval": 200
}
```

值以毫秒为单位为正数。它应当与 missing_max 选项结合使用，以确定链接报告为 down 前的总时间。

要在将链接报告为 down 前配置允许的最大丢失的 NS/NA 回复数据包数量，请添加或编辑部分，如下所示：

```
"link_watch": {
    "name": "nsna_ping",
```

```
    "missed_max": 15
}
```

丢失的 NS/NA 回复数据包的最大数量.如果超过这个数字，链接将报告为 down。missing_max 选项是允许最多允许回复数的限值（例如，ARP 回复）。它应当与 interval 选项一同选择，以确定链接报告为 down 前的总时间。

要配置解析到 NS/NA 数据包的 IPv6 地址的主机名，请添加或编辑部分，如下所示：

```
"link_watch": {
    "name": "nsna_ping",
    "target_host": "MyStorage"
}
```

“target_host” 选项包含要转换为 IPv6 地址的主机名，该地址将用作 NS/NA 数据包的目标地址。可以使用 IPv6 地址代替主机名。

有关详细信息，请参阅 teamd.conf(5) 手册页。

8.13.8. 配置端口选择覆盖

传输帧的物理端口通常由团队驱动程序的内核部分选择，与用户或系统管理员无关。使用所选团队模式（teamd 运行程序）的策略选择输出端口。但默认情况下，将特定类别的传出流量定向到某些物理接口以实施略为复杂的策略会很有帮助。默认情况下，团队驱动程序是 multiqueue aware，并在驱动程序初始化时创建 16 个队列。如果需要更多或更少的队列，则可以使用 Netlink 属性 tx_queues 在创建团队驱动程序实例期间更改此值。

端口的队列 ID 可以通过端口配置选项 queue_id 设置，如下所示：

```
{
  "queue_id": 3
}
```

这些队列 ID 可与 tc 实用程序结合使用，以配置多队列队列强制和过滤器，以抑制要传输在特定端口设备上的某些流量。例如，如果使用上述配置并希望强制绑定到 192.168.1.100 的所有流量都使用 enp1s0 在团队的输出设备中，以以下格式以 root 身份发出命令：

```
~]# tc qdisc add dev team0 handle 1 root multiq
~]# tc filter add dev team0 protocol ip parent 1: prio 1 u32 match ip dst \
  192.168.1.100 action skbedit queue_mapping 3
```

这种覆盖运行程序选择逻辑的机制以便绑定到特定端口的流量可以用于所有运行程序。

8.13.9. 配置基于 BPF 的 Tx Port Selectors

loadbalance 和 LACP 运行程序使用数据包的哈希来排序网络流量。哈希计算机制基于 Berkeley Packet Filter(BPF)代码。BPF 代码用于生成哈希值，而不是对传出数据包做出策略决策。哈希长度为 8 位，给出 256 个变体。这意味着，许多不同的套接字缓冲区 (SKB)可以具有相同的哈希值，因此流量通过同一链接传递。使用简短哈希是一种将流量划分到不同流的快速方法，以满足在多个链接之间进行负载平衡的目的。在静态模式中，哈希仅用于决定流量应发送的端口。在活动模式下，运行程序将持续将哈希重新分配给不同的端口，以尽力达到完美平衡。

以下片段类型或字符串可用于数据包 Tx 哈希计算：

- eth - 使用源和目标 MAC 地址。
- VLAN - 使用 VLAN ID。
- ipv4 - 使用源和目标 IPv4 地址。
- ipv6 - 使用源和目标 IPv6 地址。
- ip - 使用源和目标 IPv4 和 IPv6 地址。
- l3 - 使用源和目标 IPv4 和 IPv6 地址。
- TCP - 使用源和目标 TCP 端口。
- UDP - 使用源和目标 UDP 端口。
- SCTP - 使用源和 destinationSCTP 端口。
- l4 - 使用源和目标 TCP，以及 UDP 和SCTP 端口。

这些字符串可通过在负载平衡运行程序中添加以下格式的行来使用：

```
"tx_hash": ["eth", "ipv4", "ipv6"]
```

请参阅 第 8.13.5 节 “配置 loadbalance Runner”。

8.14. 使用 GUI 创建网络团队

8.14.1. 建立团队连接

您可以使用 nm-connection-editor 指示 NetworkManager 从两个或多个 Wired 或 InfiniBand 连接创建团队。无需先创建合作连接。它们可以配置为流程的一部分，以配置该团队。您必须具有可用接口的 MAC 地址，才能完成配置过程。

过程 8.1. 使用 nm-connection-editor 添加新团队连接

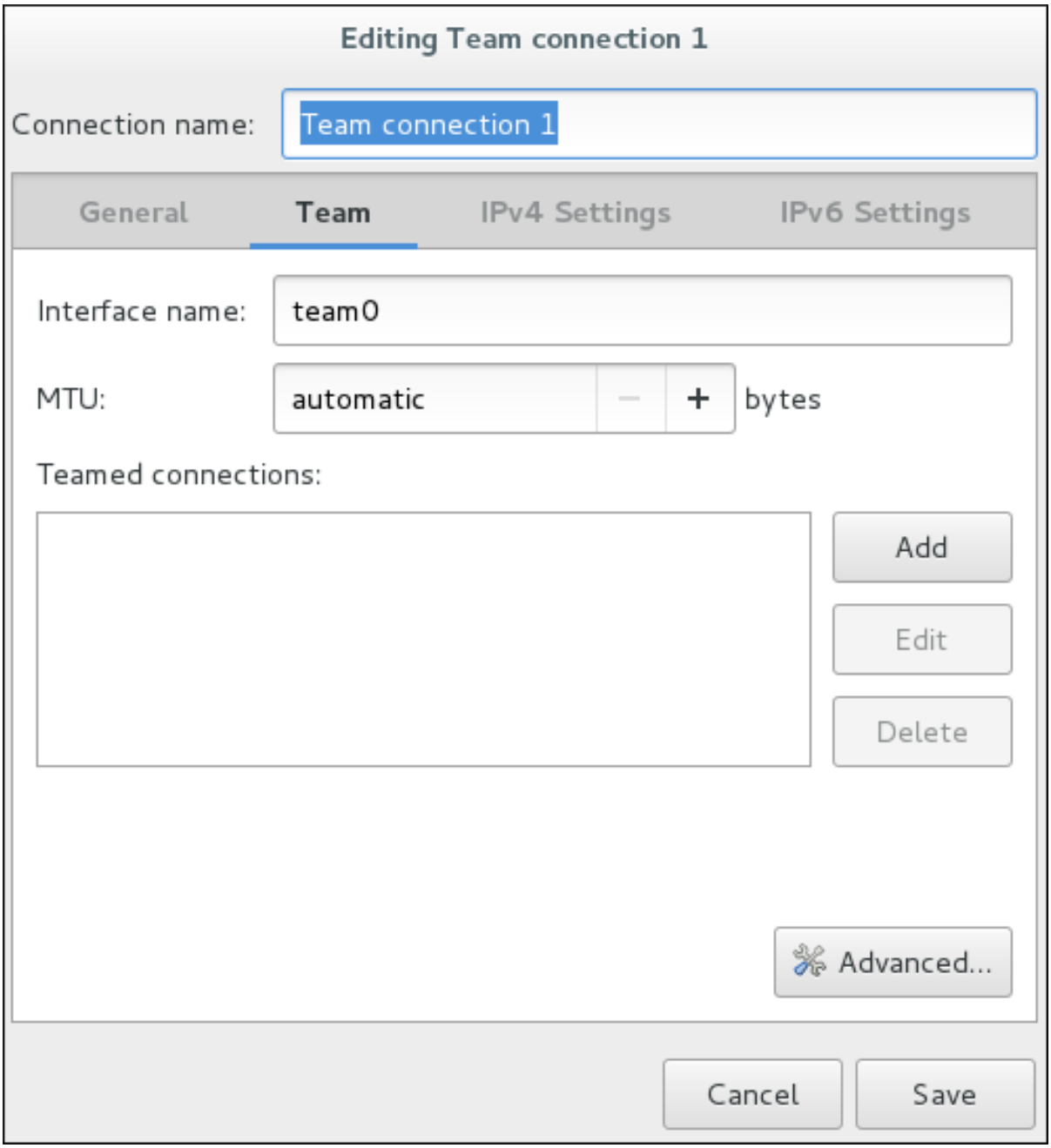
按照以下步骤添加新团队连接。

1. 在终端中输入 nm-connection-editor ：

```
~]$ nm-connection-editor
```

2. 单击添加按钮。此时将显示 Choose a Connection Type 窗口。选择 Team，再单击 Create。此时将显示 Editing Team connection 1 窗口。

图 8.6. NetworkManager 图形用户界面添加菜单

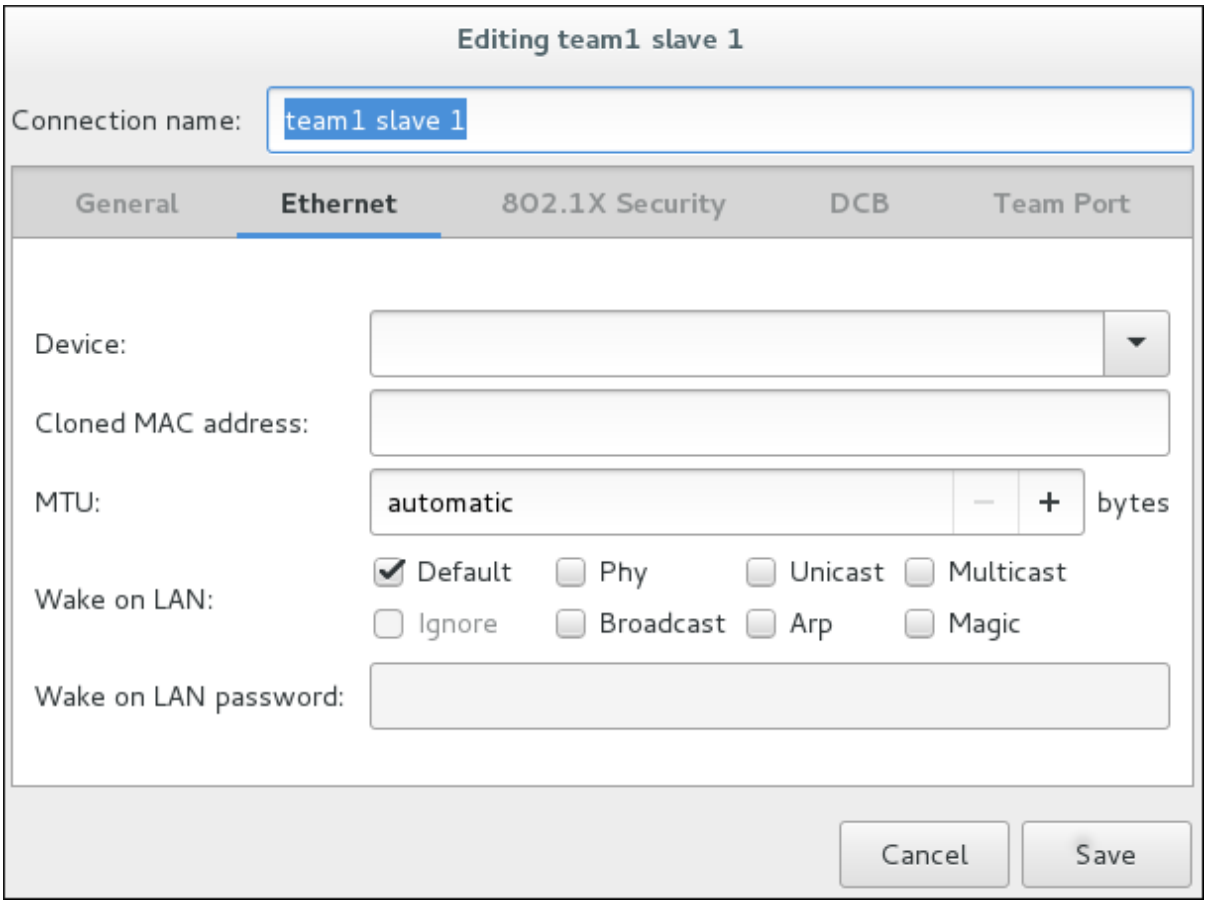


[D]

3. *在 Team 选项卡上，点 Add 并选择您要用于团队连接的界面类型。单击创建按钮。请注意，仅当您创建第一个端口时才会出现选择端口类型的对话框；之后，它将自动将同一类型用于所有后续端口。*

4. *此时将显示 Editing team0 slave 1 窗口。*

图 8.7. NetworkManager 图形用户界面添加 Slave 连接



[D]

5. *如果要应用自定义端口设置，请单击 Team Port 选项卡，再输入 JSON 配置字符串或从文件中导入。*

6. *单击 保存 按钮。*

7. *团队端口的名称会出现在 Teamed connections 窗口中。单击添加按钮以添加更多端口连接。*

8. *检查并确认设置，然后单击保存按钮。*

9. *通过引用以下 第 8.14.1.1 节 “配置 Team 选项卡” 来编辑特定于团队的设置。*

过程 8.2. 编辑现有团队连接

按照以下步骤编辑现有团队连接。

1. 在终端中输入 nm-connection-editor ：

```
~]$ nm-connection-editor
```

2. 选择您要编辑的连接并点击 Edit 按钮。

3. 选择常规选项卡。

4. 编辑对话框中的五个设置对大多数连接类型很常见。请参见 General 选项卡：

- 连接名称 - 输入您的网络连接描述性名称。此名称用于在 Network 窗口的菜单中列出此连接。
- 自动激活的连接优先级 - 如果连接被设置为自动连接，则激活该数字（默认为0 ）。数值越高，表示优先级更高。
- 当这个网络可用时自动连接到这个网络 - 如果您希望 NetworkManager 在可用时自动连接到这个连接，请选择这个框。如需更多信息，请参阅 “使用 control-center 编辑现有连接”一节。
- 所有用户可以连接到此网络 - 选择此框可创建系统上所有用户可用的连接。更改此设置可能需要 root 特权。详情请查看 第 3.4.5 节 “使用 GUI 管理系统范围以及专用连接配置集”。
- 使用这个连接时自动连接到 VPN - 如果您希望 NetworkManager 在有可用时自动连接到 VPN 连接，请选择这个框。从下拉菜单中选择 VPN。
- firewall Zone - 从下拉菜单中选择防火墙区域。有关防火墙区域的更多信息，请参阅Red Hat Enterprise Linux 7 安全指南。

5. 通过引用以下 第 8.14.1.1 节 “配置 Team 选项卡” 来编辑特定于团队的设置。

保存您的新（或修改）连接并创建进一步配置

编辑完团队连接后，点 Save 按钮保存自定义配置。

然后，配置：

- 连接的 IPv4 设置，单击 IPv4 Settings 选项卡，然后继续 第 5.4 节 “配置 IPv4 设置”

  或者

- 连接的 IPv6 设置，单击 IPv6 Settings 选项卡，再继续 第 5.5 节 “配置 IPv6 设置”。

8.14.1.1. 配置 Team 选项卡

如果您已添加新的团队连接，您可以在文本框中输入自定义 JSON 配置字符串或导入配置文件。单击 Save，将 JSON 配置应用到组接口。

有关 JSON 字符串示例，请参阅 第 8.13 节 “配置 teamd 运行程序”

有关如何添加新团队的说明，请参阅 过程 8.1, “使用 nm-connection-editor 添加新团队连接”。

8.15. 其它资源

安装的文档

- teamd(8)手册页 - 描述 teamd 服务。
- teamdctl(8)手册页 - 描述 teamd 管理工具。
- teamd.conf(5) 手册页 - 描述 teamd 配置文件。
- teamnl(8)手册页 - 描述 teamd Netlink 库。

- ***bond2team(1) 手册页 - 描述将绑定选项转换为 team 的工具。***

在线文档

http://www.w3schools.com/js/js_json_syntax.asp

JSON 语法说明。

第 9 章 配置网络桥接

网桥是一种链路层设备，可根据 MAC 地址在网络之间转发流量。它根据其构建的 MAC 地址表做出转发决策，该表通过侦听网络流量并了解连接到每个网络的主机。可以在 Linux 主机中使用软件网桥来模拟硬件桥，例如在虚拟化应用程序中，用于与一个或多个虚拟 NIC 共享 NIC。

请注意，不能通过以 Ad-Hoc 或 Infrastructure 模式运行的 Wi-Fi 网络来构建桥接。这是因为 IEEE 802.11 标准指定在 Wi-Fi 中使用 3 个地址帧以高效使用空时间。

9.1. 使用文本用户界面 NMTUI 配置桥接

文本用户界面工具 nmtui 可用于在终端窗口中配置桥接。使用以下命令启动该工具：

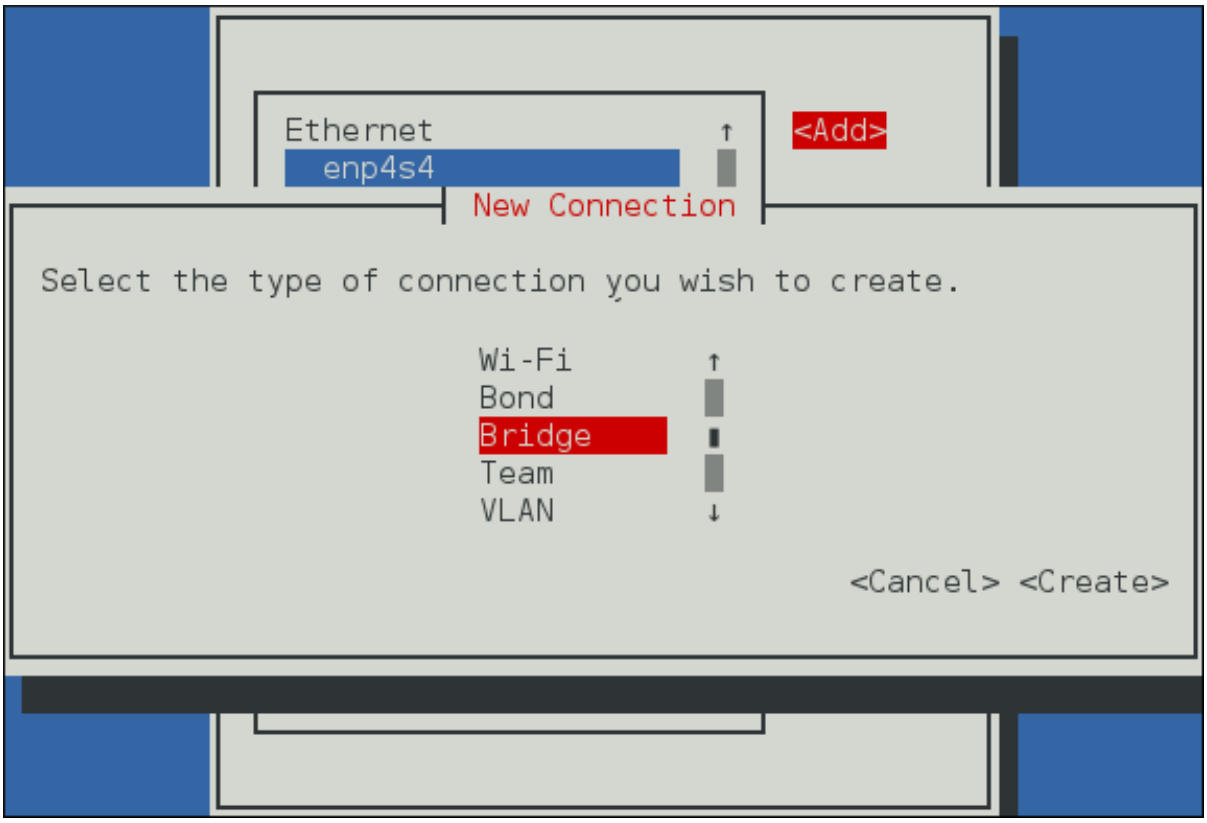
```
~]$ nmtui
```

此时将显示文本用户界面。任何无效的命令都会打印用法消息。

要导航，请使用箭头键或按 Tab 键前进，然后按ShiftTab后退步浏览选项。按 Enter 键选择一个选项。Space bar 切换复选框的状态。

1. 在起始菜单中选择 Edit a connection。选择 Add，这会打开 New Connection 屏幕。

图 9.1. NetworkManager 文本用户界面添加网桥连接菜单

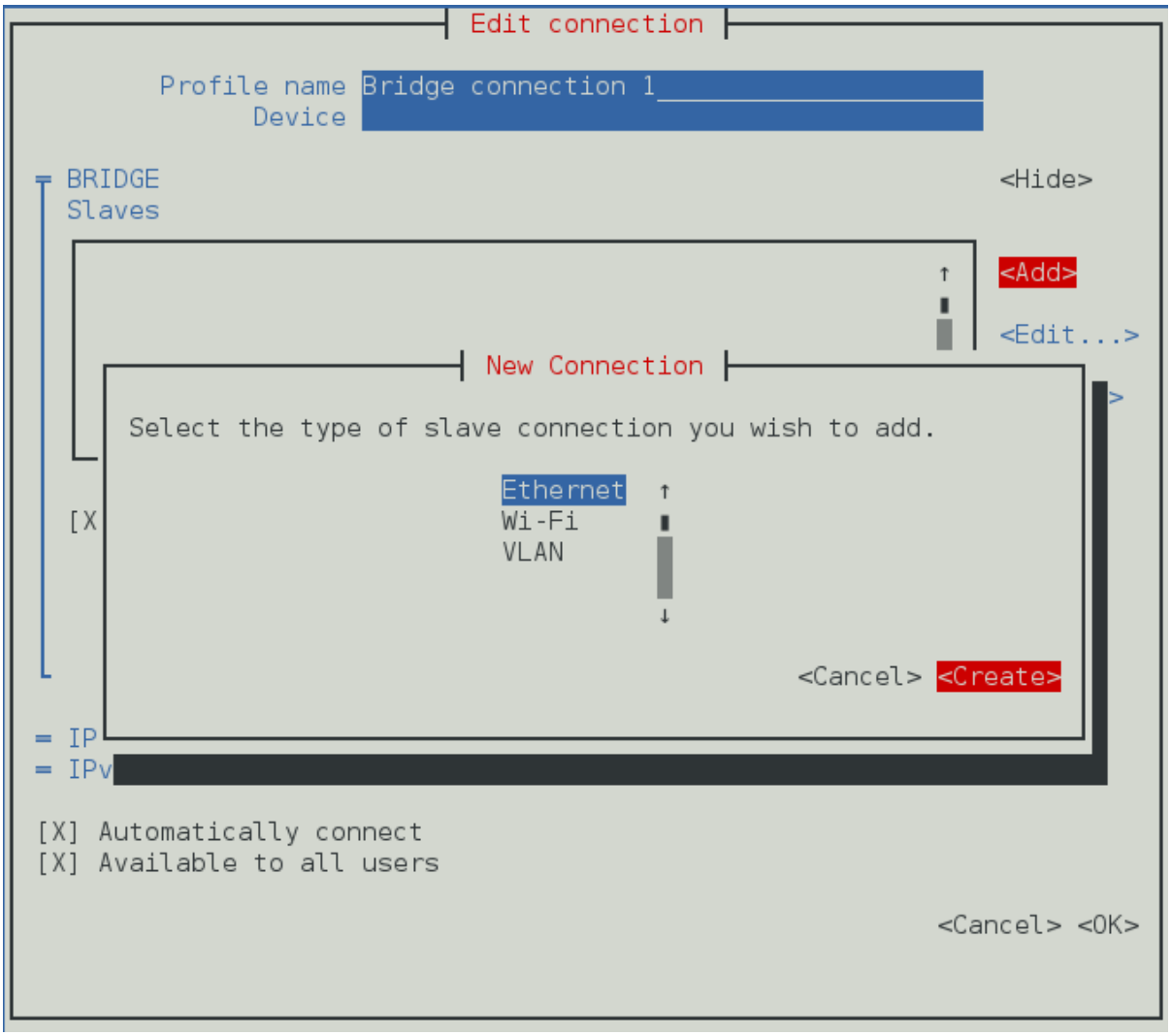


[D]

2. *选择 **Bridge**，会打开 **Edit** 连接屏幕。*

3. *要向网桥添加端口接口，请选择 **Add**，打开 **New Connection** 屏幕。选择连接类型后，选择创建按钮，使网桥的 **Edit Connection** 显示出现。*

图 9.2. NetworkManager 文本用户界面添加了新的 Bridge Slave Connection 菜单



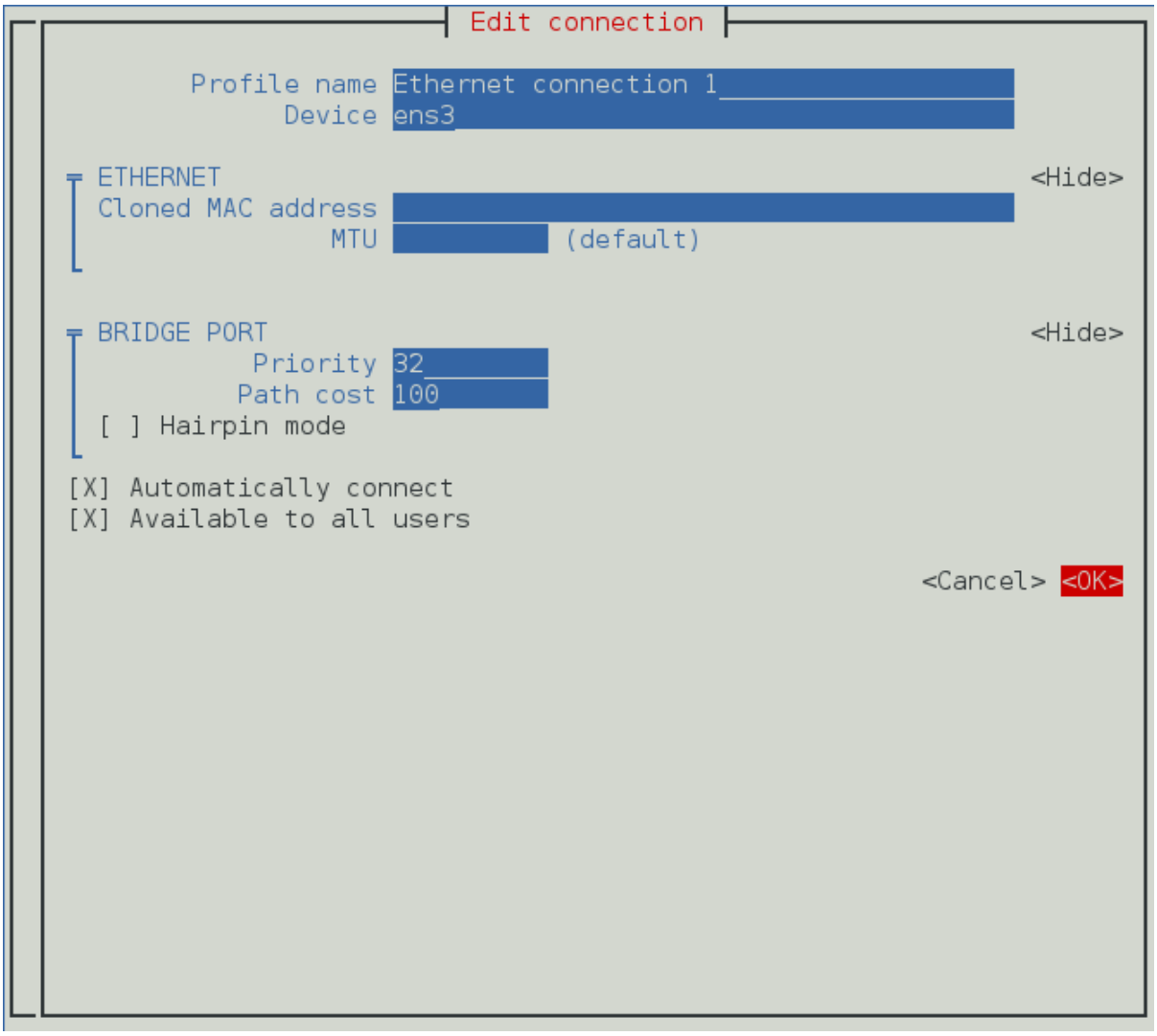
[D]

4. *在 Device 部分输入所需端口的设备名称或 MAC 地址。如果需要，通过在以太网标签右侧选择 Show 来输入要用作网桥 MAC 地址的克隆 MAC 地址。选择确定按钮。*

注意

如果设备没有 MAC 地址指定，则当 Edit Connection 窗口重新加载后，仅当成功找到该设备时，会自动填充 Device 部分。

图 9.3. NetworkManager 文本用户界面配置 Bridge Slave Connection 菜单

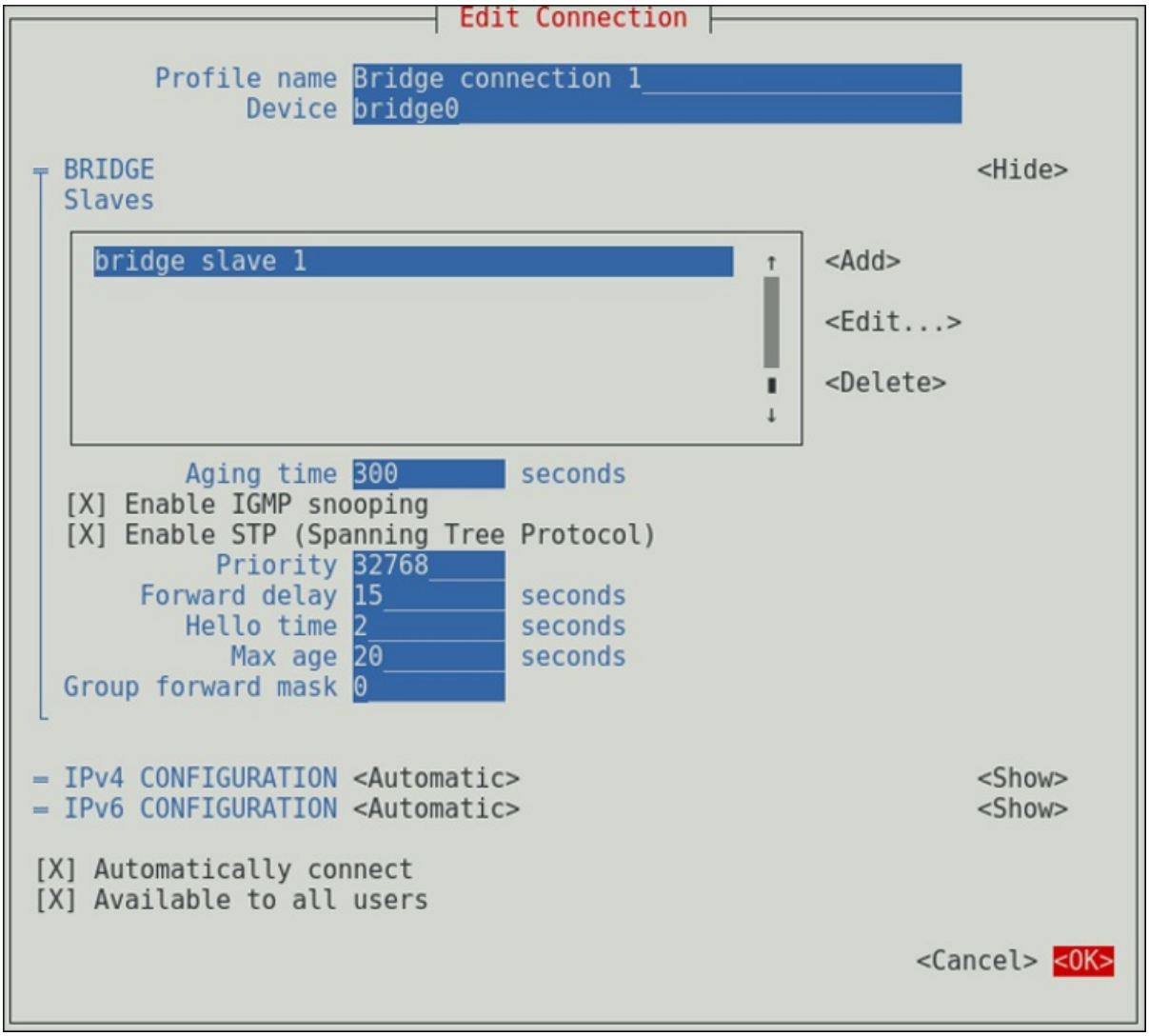


[D]

5. *网桥端口的名称会出现在 Slaves 部分中。重复上述步骤，以添加其他端口连接。*

6. *在选择确定按钮之前，检查并确认设置。*

图 9.4. NetworkManager 文本用户界面配置网桥菜单



[D]

有关网桥术语的定义，请参阅 第 9.4.1.1 节 “配置 Bridge 选项卡”。

有关安装 nmtui 的详情请查看 第 3.2 节 “使用 nmtui 配置 IP 网络”。

9.2. 使用 NETWORKMANAGER 命令行工具 NMCLI

要创建名为 的网桥，请执行以下操作： bridge-br0，以 root 身份发出命令 ：

```
~]# nmcli con add type bridge ifname br0
Connection 'bridge-br0' (6ad5bba6-98a0-4f20-839d-c997ba7668ad) successfully added.
```

如果没有指定接口名称，名称将默认为 bridge,bridge-1,bridge-2等等。

要查看连接，请运行以下命令：

```
~]$ nmcli con show
NAME        UUID                                  TYPE            DEVICE
bridge-br0  79cf6a3e-0310-4a78-b759-bda1cc3eef8d  bridge          br0
enp1s0      4d5c449a-a6c5-451c-8206-3c9a4ec88bca  802-3-ethernet  enp1s0
```

默认启用跨树协议 (STP)。使用的值来自 IEEE 802.1D-[1] 标准。要禁用这个网桥的 STP，以 root 身份发出一个命令：

```
~]# nmcli con modify bridge-br0 bridge.stp no
```

要为此网桥重新启用 802.1D STP，请以 root 身份发出命令 ：

```
~]# nmcli con modify bridge-br0 bridge.stp yes
```

802.1D STP 的默认网桥优先级为 32768。在 root 网桥选择中首选数值较低。例如，优先级为 28672 的网桥将被选为 root 网桥，而不是优先级为 32768 （默认值）的网桥。要使用非默认值创建桥接，请按如下所示发出命令：

```
~]$ nmcli con add type bridge ifname br5 stp yes priority 28672
Connection 'bridge-br5' (86b83ad3-b466-4795-aeb6-4a66eb1856c7) successfully added.
```

允许的值介于 0 到 65535 之间。

要将现有网桥的网桥优先级更改为非默认值，请以以下格式发出命令：

```
~]$ nmcli connection modify bridge-br5 bridge.priority 36864
```

允许的值介于 0 到 65535 之间。

要将网桥连接配置为在 01 :80:C2:00:00:00 到 01:80: C2:00:00:0F 范围内转发组地址，请更改 group-forward-mask 属性。此属性是 16 位的掩码。每个位对应于上方必须转发的组地址。例如：

```
~]$ nmcli connection modify bridge-br5 bridge.group-forward-mask 8
```

重要

group-forward-mask 属性无法将 0、 1、2 位中的任何一个设置为 1，因为这些地址用于生成树协议(STP)、链路聚合控制协议(LACP)和以太网 MAC 暂停帧。

要查看网桥设置，请运行以下命令：

```
~]$ nmcli -f bridge con show bridge-br0
```

nmcli(1) man page 的 bridge 部分列出了 802.1D STP 的其他选项。

要添加或分配接口，例如 enp1s0，到网桥 bridge-br0，发出如下所示的命令：

```
~]$ nmcli con add type ethernet ifname enp1s0 master bridge-br0
Connection 'bridge-slave-enp1s0' (70ffae80-7428-4d9c-8cbd-2e35de72476e) successfully
added.
```

要分配现有到网桥的连接，请遵循以下步骤：

1. **更改其控制器和端口类型属性。例如，分配名为 vlan100 的现有 VLAN 连接：**

   ```
   ~]$ nmcli connection modify vlan100 master bridge-br0 slave-type bridge
   ```

2. **重新激活连接以应用更改：**

   ```
   ~]$ nmcli connection up vlan100
   ```

要使用互动模式更改值，请运行以下命令：

```
~]$ nmcli connection edit bridge-br0
```

您将进入 nmcli 提示符。

```
nmcli> set bridge.priority 4096
nmcli> save
Connection 'bridge-br0' (79cf6a3e-0310-4a78-b759-bda1cc3eef8d) successfully saved.
nmcli> quit
```

有关 nmcli 简介，请参阅 第 3.3 节 “使用 nmcli 配置 IP 网络”。

9.3. 使用命令行界面(CLI)

9.3.1. 检查 Bridging 内核模块是否已安装

在 Red Hat Enterprise Linux 7 中，默认载入桥接模块。如果需要，您可以以 root 用户身份运行以下命令来确保载入该模块：

```
~]# modprobe --first-time bridge
modprobe: ERROR: could not insert 'bridge': Module already in kernel
```

要显示模块信息，请运行以下命令：

```
~]$ modinfo bridge
```

有关更多命令选项，请参阅 modprobe(8)手册页。

9.3.2. 创建网桥

要创建网桥，请在 /etc/sysconfig/network-scripts/ 目录中创建一个名为 ifcfg-brN 的文件，将 N 替换为接口的数字，如 0。

文件的内容与任何类型的接口类似，如以太网接口。这个示例中的区别如下：

- DEVICE 指令以 brN 格式指定接口名称作为其参数，其中 N 被接口编号取代。
- 为 TYPE 指令指定参数 Bridge。此指令决定设备类型，参数区分大小写。
- 网桥接口配置文件被赋予一个 IP 地址，而物理接口配置文件只能具有 MAC 地址（参见下方）。
- 添加了一个额外的指令 DELAY=0，以防止该网桥在监控流量时等待该网桥，了解主机所在的位置，并构建 MAC 地址表，以基于该网桥的过滤决策。如果没有路由循环，则不需要 15 秒的默认延迟。

例 9.1. ifcfg-br0 接口配置文件示例

以下是使用静态 IP 地址的网桥接口配置文件示例 ：

```
DEVICE=br0
TYPE=Bridge
IPADDR=192.168.1.1
PREFIX=24
BOOTPROTO=none
ONBOOT=yes
DELAY=0
```

若要完成网桥另一接口，或修改现有的接口，请将其指向网桥接口。

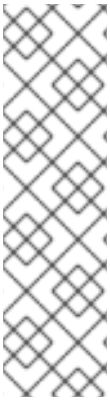
例 9.2. ifcfg-enp1s0 接口配置文件示例

以下是指向网桥接口配置文件的以太网接口配置文件示例：在 /etc/sysconfig/network-scripts/ifcfg-device_name 中配置您的物理接口，其中 device_name 是接口的名称

```
DEVICE=device_name
TYPE=Ethernet
HWADDR=AA:BB:CC:DD:EE:FF
```

```
BOOTPROTO=none
ONBOOT=yes
BRIDGE=br0
```

（可选）使用 NAME 指令指定名称。如果没有指定名称，NetworkManager 插件 ifcfg-rh 将以“Type Interface”格式为连接配置集创建一个名称。在本例中，这意味着网桥将命名为 Bridge br0。或者，如果 NAME=bridge-br0 添加到 ifcfg-br0 文件中，则连接配置集将命名为 bridge-br0。



注意

对于 DEVICE 指令，几乎可使用任何接口名称，因为它不能确定设备类型。TYPE=Ethernet 不严格要求。如果未设置 TYPE 指令，则该设备将被视为以太网设备（除非其名称明确匹配不同的接口配置文件）。

指令区分大小写。

使用HWADDR 指令指定硬件或 MAC 地址将影响设备命名过程，如 第 11 章 一致的网络设备命名 中所述。

警告

如果您在远程主机上配置桥接，并且通过您要配置的物理 NIC 连接到该主机，请考虑在继续操作前丢失连接的影响。您在重新启动服务时将断开连接，并且如果出现任何错误，可能无法重新连接。建议控制台或带外访问。

要打开新的或最近配置的接口，以以下格式以 root 身份发出命令：

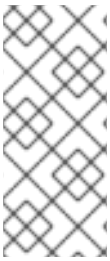
```
ifup device
```

该命令将检测 NetworkManager 是否正在运行并调用 nmcli con load UUID，然后调用 nmcli con up UUID。

另外，要重新载入所有接口，以 root 用户身份运行以下命令：

```
~]# systemctl restart network
```

此命令将停止网络服务，启动网络服务，然后为所有 ifcfg 文件调用 ifup，使用 ONBOOT=yes。



注意

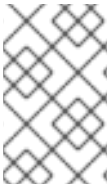
默认行为是让 NetworkManager 不知晓对 ifcfg 文件的更改，并继续使用旧配置数据，直到接口下次启动为止。这由 NetworkManager.conf 文件中的 monitor-connection-files 选项设置。如需更多信息，请参阅 NetworkManager.conf(5) 手册页。

9.3.3. 使用 Bond 的网桥

现在，将提供一个由两个或多个绑定以太网接口组成的网桥示例，因为这是虚拟化环境中的另一个常用应用程序。如果您不熟悉绑定接口的配置文件，请参阅 第 7.4.2 节 “创建频道绑定接口”

创建或编辑两个或多个以太网接口配置文件，它们将绑定，如下所示：

```
DEVICE=interface_name
TYPE=Ethernet
SLAVE=yes
MASTER=bond0
BOOTPROTO=none
HWADDR=AA:BB:CC:DD:EE:FF
```



注意

将interface_name 用作接口名称是常见的做法，但几乎可以使用任何名称。

创建或编辑一个接口配置文件 /etc/sysconfig/network-scripts/ifcfg-bond0，如下所示：

```
DEVICE=bond0
ONBOOT=yes
BONDING_OPTS='mode=1 miimon=100'
BRIDGE=brbond0
```

有关配置 bonding 模块和查看绑定参数列表的详情请参考 第 7.7 节 “使用频道绑定”。

创建或编辑一个接口配置文件 /etc/sysconfig/network-scripts/ifcfg-brbond0，如下所示：

```
DEVICE=brbond0
ONBOOT=yes
TYPE=Bridge
IPADDR=192.168.1.1
PREFIX=24
```

现在，我们有两个或者多个接口配置文件，包含 MASTER=bond0 指令。它们指向名为 /etc/sysconfig/network-scripts/ifcfg-bond0 的配置文件，其中包含 DEVICE=bond0 指令。此 ifcfg-bond0 依次指向包含 IP 地址的 /etc/sysconfig/network-scripts/ifcfg-brbond0 配置文件，并充当主机内虚拟网络的接口。

要打开新的或最近配置的接口，以以下格式以 root 身份发出命令：

```
ifup device
```

该命令将检测 NetworkManager 是否正在运行并调用 nmcli con load UUID，然后调用 nmcli con up UUID。

另外，要重新载入所有接口，以 root 用户身份运行以下命令：

```
~]# systemctl restart network
```

此命令将停止网络服务，启动网络服务，然后为所有 ifcfg 文件调用 ifup，使用 ONBOOT=yes。

注意

默认行为是让 NetworkManager 不知晓对 ifcfg 文件的更改，并继续使用旧配置数据，直到接口下次启动为止。这由 NetworkManager.conf 文件中的 monitor-connection-files 选项设置。如需更多信息，请参阅 NetworkManager.conf(5) 手册页。

9.4. 使用 GUI 配置网络桥接

启动网桥接口时，NetworkManager 在启动任何网络独立 IP 配置（如 DHCP 或 IPv6 “自动配置”）前，会等待至少一个端口进入转发状态”。在连接端口或端口之前，可以在连接或开始转发数据包前执行静态 IP 寻址。

9.4.1. 使用 GUI 建立网桥连接

过程 9.1. 使用 nm-connection-editor 添加新网桥连接

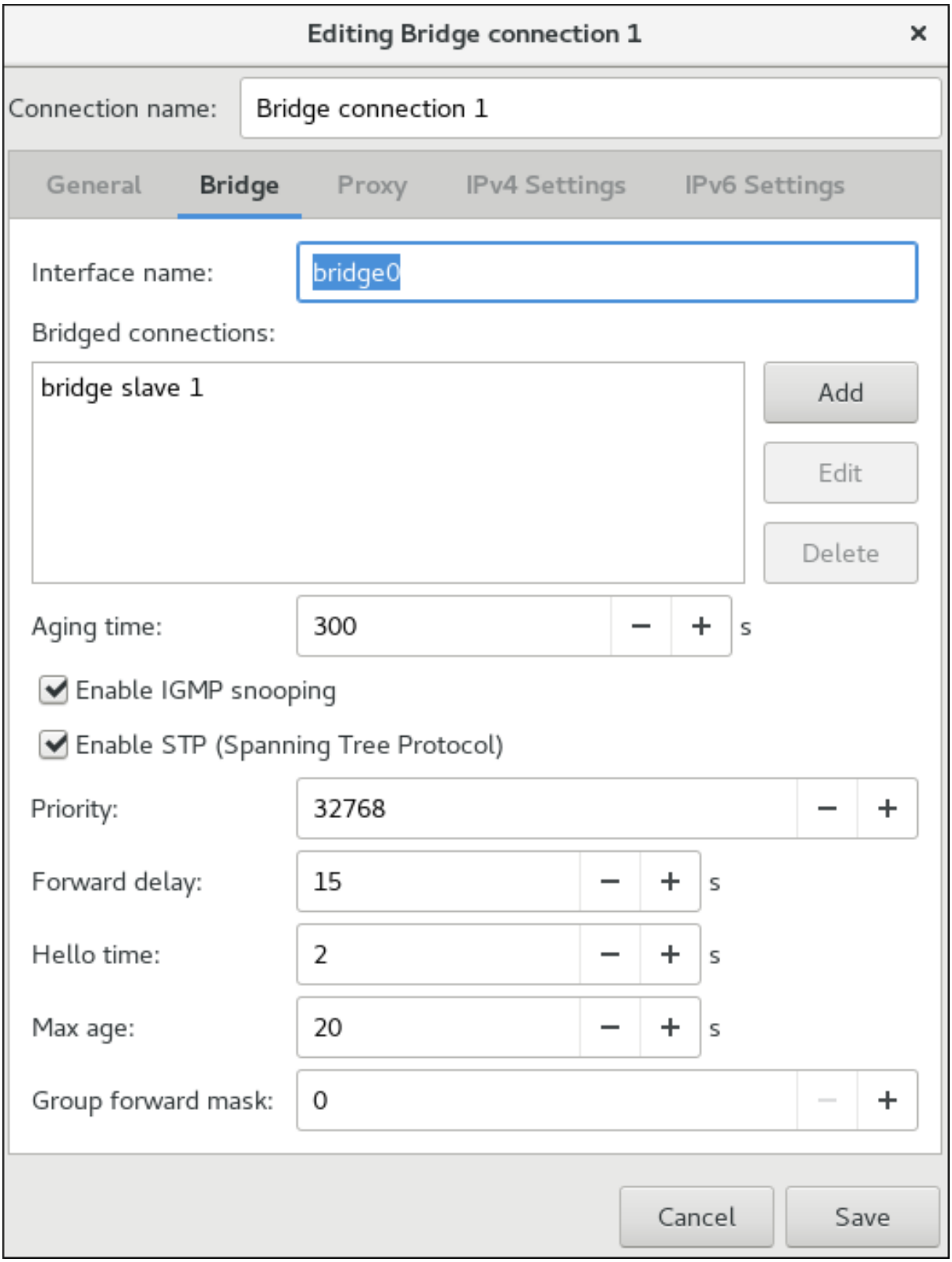
按照以下步骤创建新网桥连接：

1. 在终端中输入 nm-connection-editor ：

```
~]$ nm-connection-editor
```

2. 单击添加按钮。此时将显示 Choose a Connection Type 窗口。选择 Bridge 并点 Create。此时会出现 Editing Bridge connection 1 窗口。

图 9.5. 编辑网桥连接 1



[D]

3. *通过下面的 过程 9.3, “在网桥中添加端口接口” 来添加端口设备。*

过程 9.2. 编辑现有网桥连接

1. 在终端中输入 nm-connection-editor ：

```
~]$ nm-connection-editor
```

2. 选择您要编辑的网桥连接。

3. 单击编辑按钮。

配置连接名称、自动连接行为和可用性设置

Editing 对话框中的五个设置对所有连接类型通用，请参阅 General 选项卡：

- 连接名称 - 输入您的网络连接描述性名称。此名称将用于在 Network 窗口的菜单中列出此连接。

- 当这个网络可用时自动连接到这个网络 - 如果您希望 NetworkManager 在可用时自动连接到这个连接，请选择这个框。如需更多信息，请参阅 “使用 control-center 编辑现有连接”一节。

- 所有用户可以连接到此网络 - 选择此框可创建系统上所有用户可用的连接。更改此设置可能需要 root 特权。详情请查看 第 3.4.5 节 “使用 GUI 管理系统范围以及专用连接配置集”。

- 使用这个连接时自动连接到 VPN - 如果您希望 NetworkManager 在有可用时自动连接到 VPN 连接，请选择这个框。从下拉菜单中选择 VPN。

- 防火墙区域 - 从下拉菜单中选择防火墙区域。有关防火墙区域的更多信息，请参阅Red Hat Enterprise Linux 7 安全指南。

9.4.1.1. 配置 Bridge 选项卡

接口名称

网桥的接口名称。

桥接连接

个或多个端口接口。

老化时间

MAC 地址的时间（以秒为单位）保存在 MAC 地址转发数据库中。

启用 IGMP 侦听

如果需要，选中复选框以在设备上启用 IGMP 侦听。

启用 STP（生成树协议）

如果需要，选择要启用 STP 的复选框。

优先级

优先级最低的网桥；将选择优先级最低的网桥作为根网桥。

forward delay

在进入转发状态前，以秒为单位花费在列表和学习状态上的时间。默认值为 15 秒。

您好的时间

在网桥协议数据单元(BPDU)中发送配置信息之间的时间间隔，单位为秒。

最长期限

存储来自 BPDU 的配置信息的最长时间，以秒为单位。这个值应该是 Hello Time 和 1 的两倍，但小于转发延迟减号的两倍。

组转发掩码

此属性是组地址的掩码，允许转发组地址。在大多数情况下，将范围中的地址分组为 01:80:C2:00:00:00 到 01:80:C2:00:00:0F，不会由网桥设备转发。此属性是 16 位的掩码，各自对应于上述范围内的组地址，必须转发。请注意，组转发掩码 属性无法将 0、 1、 2 位中的任何一个设置为 1，因为这些地址用于生成树协议(STP)、链路聚合控制协议(LACP)和以太网 MAC 暂停帧。

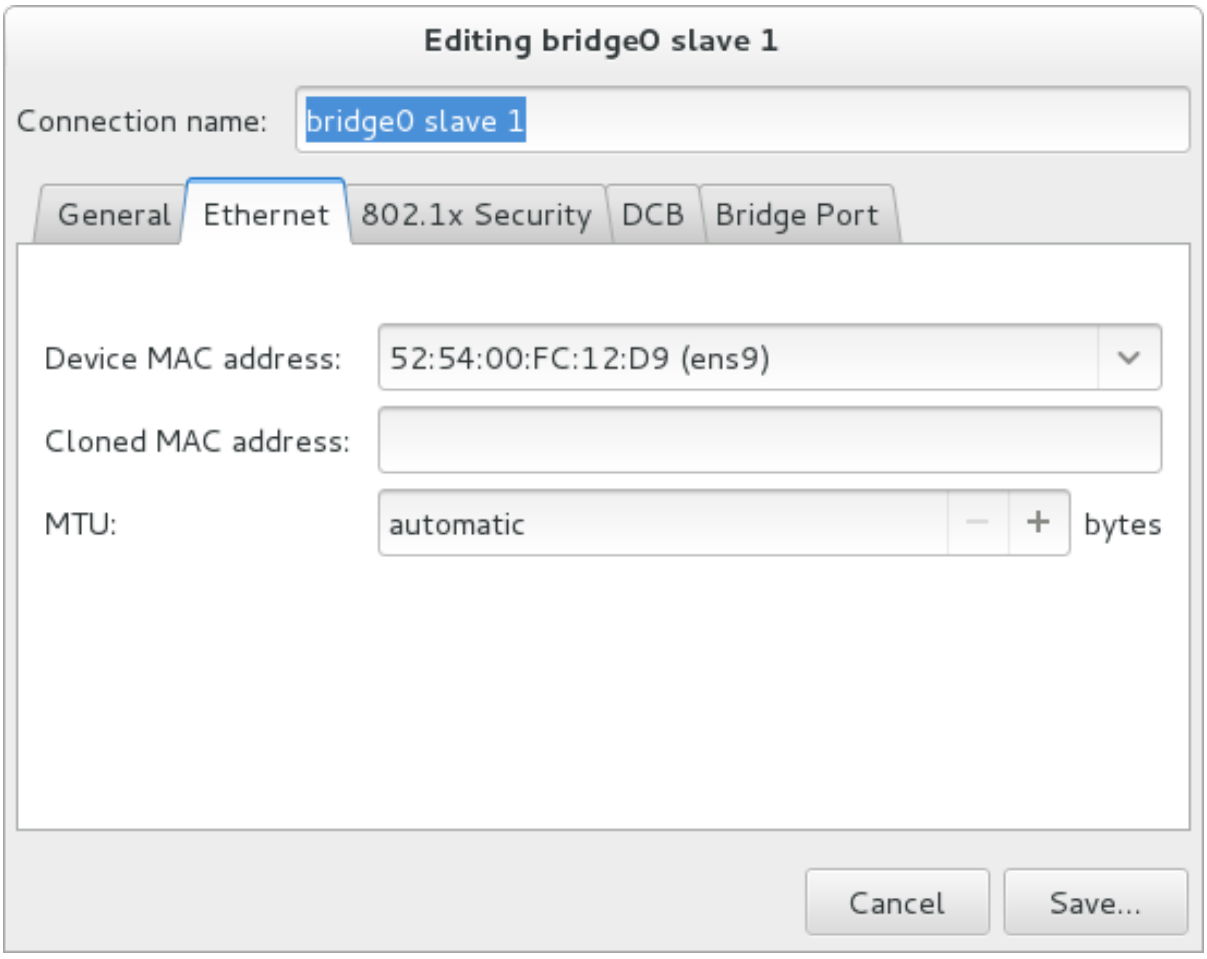
过程 9.3. 在网桥中添加端口接口

1. 要向网桥添加端口，请在 Editing Bridge connection 1窗口中选择 Bridge 选项卡。如有必要，请按照 过程 9.2, “编辑现有网桥连接” 中的步骤打开此窗口。

2. 点添加。此时将显示 Choose a Connection Type 菜单。

3. 从列表中选择要创建的连接类型。点 Create。此时会显示适合所选连接类型的窗口。

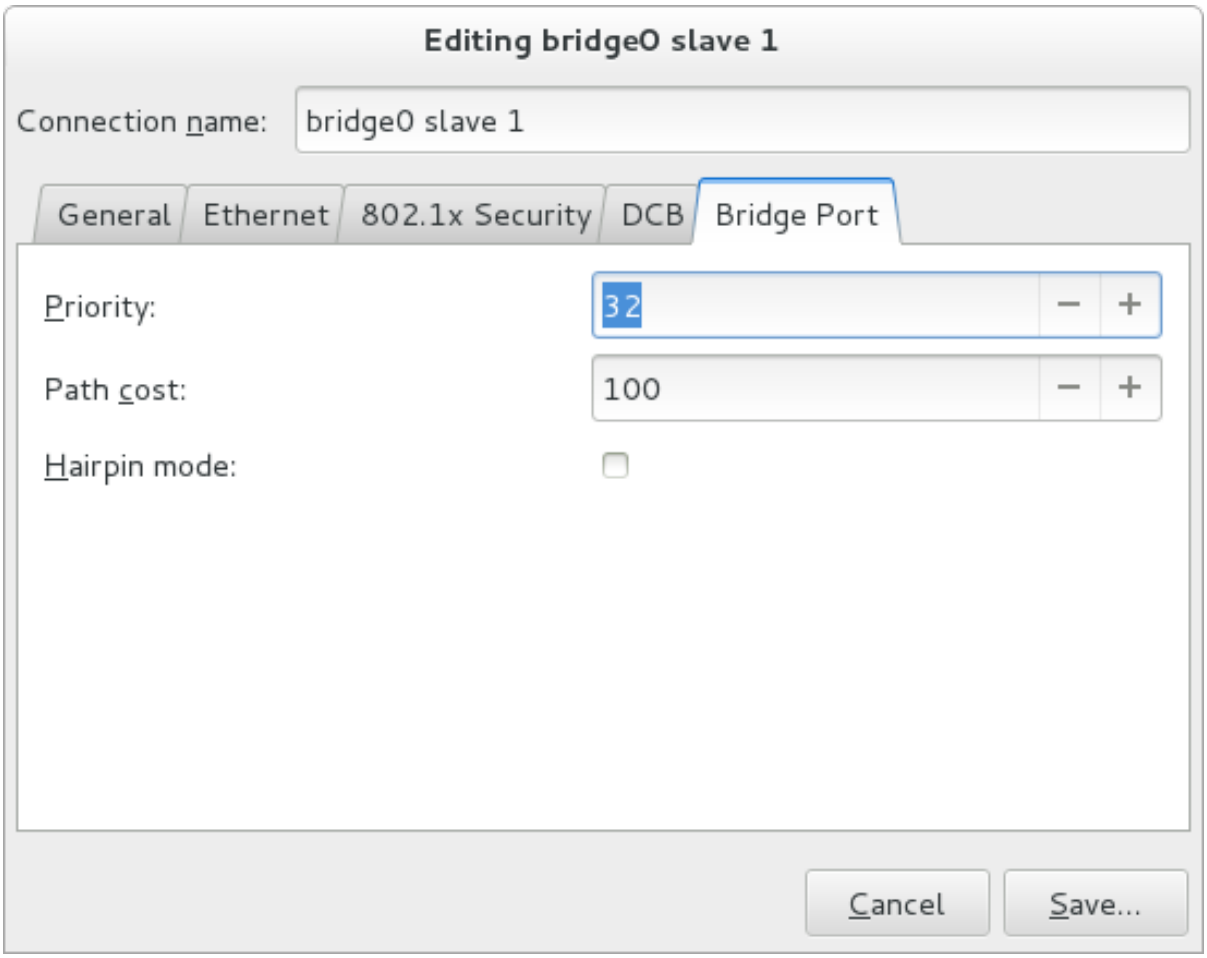
图 9.6. NetworkManager 图形用户界面添加网桥连接



[D]

4. 选择"网桥端口"选项卡。根据需要配置优先级 和路径成本.请注意，网桥端口的 STP 优先级受 Linux 内核的限制。虽然标准允许 0 到 255 范围， 但 Linux 仅允许 0 到 255。本例中默认为 32。

图 9.7. NetworkManager 图形用户界面网桥端口选项卡



[D]

5. *如果需要，选中 Hairpin 模式复选框，以启用外部处理帧转发。也称为虚拟以太网端口聚合器 (VEPA)模式。*

然后，配置：

- *以太网端口，点击 以太网 选项卡并继续 “基本配置选项”一节 ；*
- *绑定端口，点 Bond 选项卡并继续 第 7.8.1.1 节 “配置 Bond 选项卡” ；*
- *团队端口，点击 团队 选项卡并继续 第 8.14.1.1 节 “配置 Team 选项卡” ；*
- *VLAN 端口，点 VLAN 选项卡，再继续 第 10.5.1.1 节 “配置 VLAN 选项卡” ；*

保存您的新（或修改）连接并创建进一步配置

编辑完新网桥连接后，点 Save 按钮保存自定义配置。如果在编辑期间使用配置集，请重启连接以使 NetworkManager 应用更改。如果配置集是 OFF，将其设置为 ON，或者在网络连接图标的菜单中选择它。有关使用新连接或更改的连接的详情，请查看 第 3.4.1 节 “使用 control-center GUI 连接到网络”。

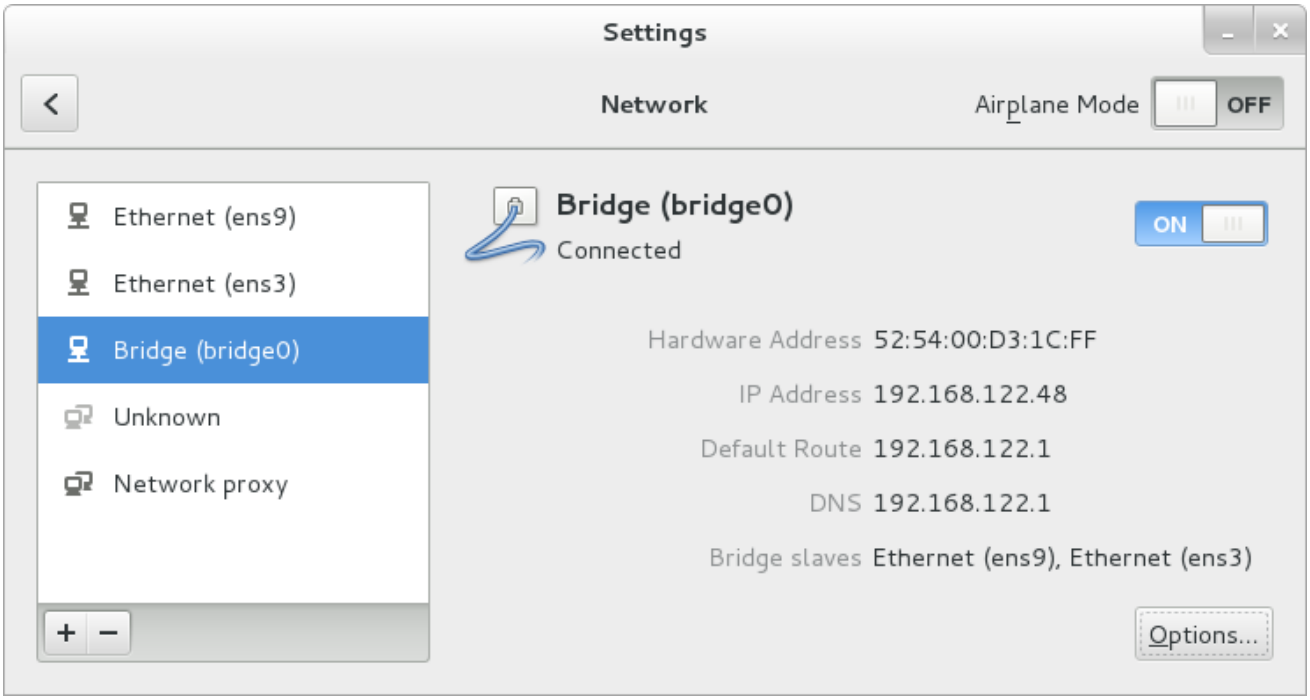
您可以通过在网络窗口中选择现有连接并单击 Options 返回编辑对话框来进一步配置现有连接。

然后，配置：

- 连接的 IPv4 设置，点击 IPv4 Settings 选项卡，然后继续 第 5.4 节 “配置 IPv4 设置” 或；
- 连接的 IPv6 设置，单击 IPv6 Settings 选项卡，再继续 第 5.5 节 “配置 IPv6 设置”。

保存后，网桥将会显示在网络设置工具中，每个端口显示在显示中。

图 9.8. 带有 Bridge 的 NetworkManager 图形用户界面



[D]

9.5. 使用 IPROUTE 配置以太网桥接配置

iproute 软件包可用作 bridge-utils 的替代选择。它允许设置网桥端口选项，如 优先级、成本 或 状态。

要使用 ip 实用程序为分配给网桥设备的接口设置端口选项，以 root 身份运行以下命令：

```
~]# ip link set enp1s0 type bridge_slave option
```

要选择可用选项，使用 ip 实用程序以 root 用户身份运行以下命令：

```
~]# ip link help bridge_slave
    Usage: ... bridge_slave [ state STATE ] [ priority PRIO ] [cost COST ]
                [ guard {on | off} ]
                [ hairpin {on | off} ]
                [ fastleave {on | off} ]
                [ root_block {on | off} ]
                [ learning {on | off} ]
                [ flood {on | off} ]
```

有关端口选项的详情，请查看 ip-link(8)man page。

9.6. 其它资源

- nmcli(1) man page - 描述 NetworkManager 的命令行工具。
- nmcli-examples(5) 手册页 - 提供 nmcli 命令的示例.
- nm-settings(5) 手册页 - NetworkManager 连接的设置和参数的说明.
- ip-link(8)手册页 - 网桥端口选项的描述.

第 10 章 配置 802.1Q VLAN 标记

若要创建 VLAN，需要在另一个接口上创建一个接口，该接口称为父接口。当 VLAN 接口通过接口时，VLAN 接口将使用 VLAN ID 标记数据包，而返回的数据包将被取消标记。VLAN 接口可以配置与任何其他接口类似。父接口不需要是以太网接口。可以在网桥、绑定和组接口之上创建一个 802.1Q VLAN 标记接口，但要注意一些事项：

- 对于绑定上的 VLAN，在打开 VLAN 接口之前，绑定具有端口且启动它们非常重要。“”在没有端口的绑定中添加 VLAN 接口无法正常工作。

- 无法在带有 fail_over_mac=follow 选项的绑定上配置 VLAN 端口，因为 VLAN 虚拟设备无法更改其 MAC 地址以匹配父 MAC 地址。在这种情况下，流量仍会与不正确的源 MAC 地址一同发送。

- 通过网络交换机发送 VLAN 标记的数据包需要正确配置交换机。例如，Cisco 交换机上的端口必须分配到一个 VLAN 或配置为中继端口，以接受来自多个 VLAN 的已标记数据包。某些供应商交换机允许通过中继端口处理原生 VLAN 的未标记帧。某些设备允许您启用或禁用原生 VLAN，其他设备则默认禁用它。因此，这种差异可能会导致两个不同交换机之间的原生 VLAN 配置错误，从而造成安全风险。例如：

  个交换机使用原生 VLAN 1，另一个交换机使用原生 VLAN 10。如果允许帧在不插入标签的情况下通过，攻击者可以跳过 VLAN - 这种常见的网络渗透技术也称为 VLAN 跳接。

  要最小化安全风险，请按以下方式配置接口：

  switch

  - 除非需要它们，否则禁用中继端口。
  - 如果您需要中继端口，请禁用原生 VLAN，以便不允许使用未标记的帧。

  Red Hat Enterprise Linux 服务器

- 使用 nftables or ebtables 实用程序在入口过滤中丢弃未标记帧。

- 些较旧的网络接口卡、环回接口、Wimax 卡和某些 InfiniBand 设备都被认为是 VLAN 挑战，这意味着它们不支持 VLAN。这通常是因为设备无法适应 VLAN 标头和与标记数据包关联的最大 MTU 大小。

注意

红帽不支持 VLAN 之上的绑定。如需更多信息，请参阅 Red Hat 知识库文章，是否在 VLAN 上配置绑定作为端口接口的有效配置？

10.1. 选择 VLAN 接口配置方法

- 要使用 NetworkManager 的文本用户界面工具 nmtui 配置 VLAN 接口，请继续 第 10.2 节 “使用文本用户界面 nmtui 配置 802.1Q VLAN 标记”

- 要使用 NetworkManager 的命令行工具 nmcli 配置 VLAN 接口，请继续 第 10.3 节 “使用命令行工具 nmcli 配置 802.1Q VLAN 标记”

- 要手动配置网络接口，请参阅 第 10.4 节 “使用命令行配置 802.1Q VLAN 标记”。

- 要使用图形用户界面工具配置网络，请继续 第 10.5 节 “使用 GUI 配置 802.1Q VLAN 标记”

10.2. 使用文本用户界面 NMTUI 配置 802.1Q VLAN 标记

文本用户界面工具 nmtui 可用于在终端窗口中配置 802.1Q VLAN。使用以下命令启动该工具：

```
~]$ nmtui
```

此时将显示文本用户界面。任何无效的命令都会打印用法消息。

要导航，请使用箭头键或按 Tab 键前进，然后按ShiftTab后退步浏览选项。按 Enter 键选择一个选项。Space bar 切换复选框的状态。

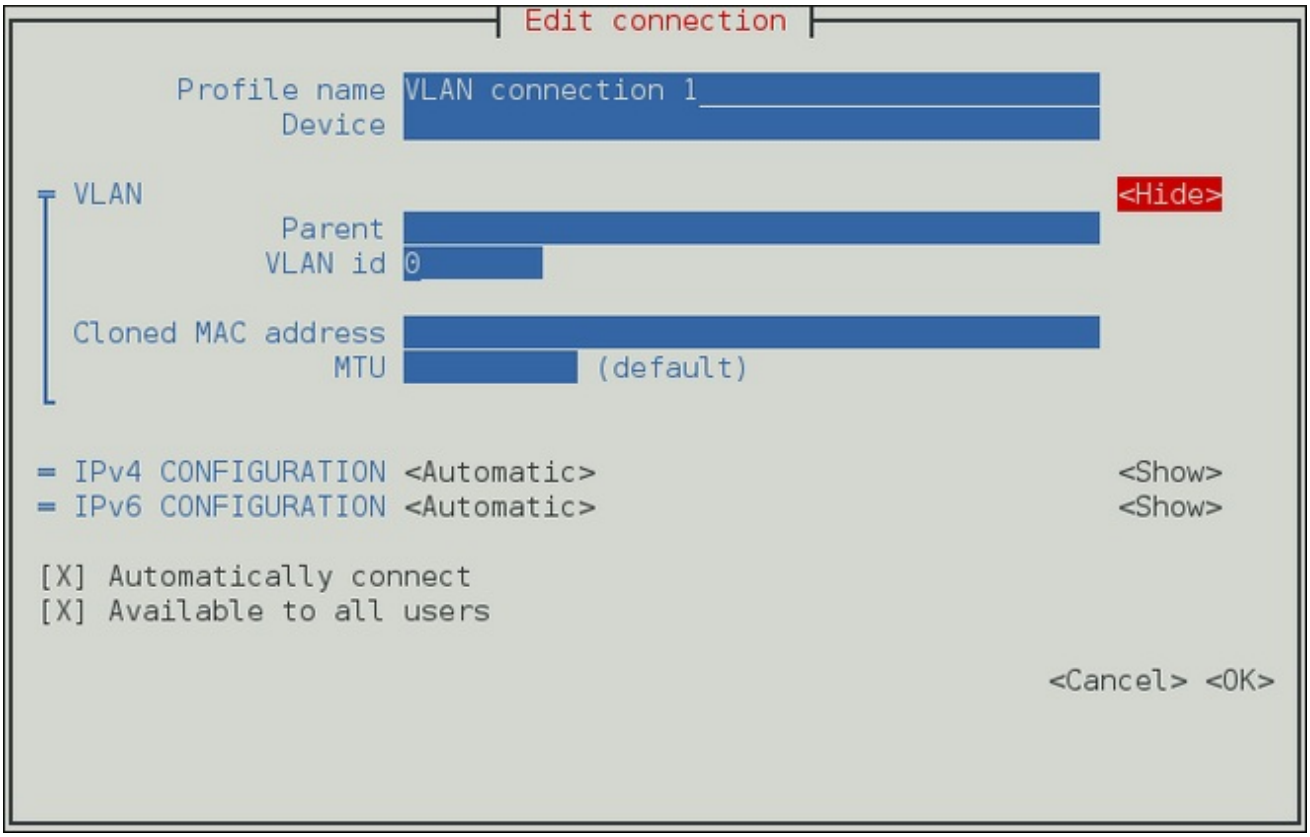
在起始菜单中选择 Edit a connection。选择 Add，这会打开 New Connection 屏幕。

图 10.1. NetworkManager 文本用户界面添加 VLAN 连接菜单

[D]

选择 VLAN，将打开 Edit 连接屏幕。按照屏幕提示完成配置。

图 10.2. NetworkManager 文本用户界面配置 VLAN 连接菜单



[D]

有关 VLAN 术语的定义，请参阅 第 10.5.1.1 节 “配置 VLAN 选项卡”。

有关安装 nmtui 的详情请查看 第 3.2 节 “使用 nmtui 配置 IP 网络”。

10.3. 使用命令行工具 NMCLI 配置 802.1Q VLAN 标记

要查看系统中的可用接口，请使用以下命令：

```
~]$ nmcli con show
NAME        UUID                                  TYPE            DEVICE
System enp2s0  9c92fad9-6ecb-3e6c-eb4d-8a47c6f50c04  802-3-ethernet  enp2s0
System enp1s0  5fb06bd0-0bb0-7ffb-45f1-d6edd65f3e03  802-3-ethernet  enp1s0
```

请注意，输出中的 NAME 字段始终表示连接 ID。它不是接口名称，即使它可能看起来相同。该 ID 可用于 nmcli 连接命令来标识连接。将 DEVICE 名称用于其他应用，如 firewalld。

要在以太网接口 enp1s0 上创建一个 802.1Q VLAN 接口，带有 VLAN 接口 VLAN10 和 ID 10，请发出以下命令：

```
~]$ nmcli con add type vlan ifname VLAN10 dev enp1s0 id 10
```

```
Connection 'vlan-VLAN10' (37750b4a-8ef5-40e6-be9b-4fb21a4b6d17) successfully added.
```

请注意，由于 VLAN 接口没有给出任何 con-name，因此名称是通过在类型前前从接口名称派生而来的。或者，使用 con-name 选项指定一个名称，如下所示：

```
~]$ nmcli con add type vlan con-name VLAN12 dev enp1s0 id 12
Connection 'VLAN12' (b796c16a-9f5f-441c-835c-f594d40e6533) successfully added.
```

为 VLAN 接口分配地址

您可以使用与任何其他接口相同的 nmcli 命令分配静态和动态接口地址。

例如，创建具有静态 IPv4 地址和网关的 VLAN 接口的命令如下：

```
~]$ nmcli con add type vlan con-name VLAN20 dev enp1s0 id 20 ip4 10.10.10.10/24 \
gw4 10.10.10.254
```

要创建带有动态分配寻址的 VLAN 接口，请按如下所示发出命令：

```
~]$ nmcli con add type vlan con-name VLAN30 dev enp1s0 id 30
```

有关使用 nmcli 命令配置接口的示例，请参阅 第 3.3.6 节 “使用 nmcli 连接到网络”。

要查看创建的 VLAN 接口，请按如下方式发出命令：

```
~]$ nmcli con show
NAME        UUID                                  TYPE           DEVICE
VLAN12      4129a37d-4feb-4be5-ac17-14a193821755  vlan           enp1s0.12
System enp2s0  9c92fad9-6ecb-3e6c-eb4d-8a47c6f50c04  802-3-ethernet  enp2s0
System enp1s0  5fb06bd0-0bb0-7ffb-45f1-d6edd65f3e03  802-3-ethernet  enp1s0
vlan-VLAN10    1be91581-11c2-461a-b40d-893d42fed4f4  vlan           VLAN10
```

要查看有关新配置连接的详细信息，请按如下所示发出命令：

```
~]$ nmcli -p con show VLAN12
===============================================================================

                Connection profile details (VLAN12)
===============================================================================

connection.id:                          VLAN12
connection.uuid:                        4129a37d-4feb-4be5-ac17-14a193821755
connection.interface-name:              --
connection.type:                        vlan
connection.autoconnect:                 yes
…
-------------------------------------------------------------------------------
802-3-ethernet.port:                    --
802-3-ethernet.speed:                   0
802-3-ethernet.duplex:                  --
802-3-ethernet.auto-negotiate:          yes
802-3-ethernet.mac-address:             --
802-3-ethernet.cloned-mac-address:      --
802-3-ethernet.mac-address-blacklist:
802-3-ethernet.mtu:                     auto
…
vlan.interface-name:                    --
vlan.parent:                            enp1s0
vlan.id:                                12
vlan.flags:                             0 (NONE)
vlan.ingress-priority-map:
vlan.egress-priority-map:
-------------------------------------------------------------------------------
===============================================================================

    Activate connection details (4129a37d-4feb-4be5-ac17-14a193821755)
===============================================================================

GENERAL.NAME:                           VLAN12
GENERAL.UUID:                           4129a37d-4feb-4be5-ac17-14a193821755
GENERAL.DEVICES:                        enp1s0.12
GENERAL.STATE:                          activating
[output truncated]
```

nmcli(1) man page 的 VLAN 部分列出了 VLAN 命令的其他选项。在 man page 中，创建 VLAN 的设备称为父设备。在上面的示例中，设备通过其接口名称 enp1s0 指定，它也可以通过连接 UUID 或 MAC 地址来指定。

在以太网接口 enp2s0 上创建一个带有入口优先级映射的 802.1Q VLAN 连接配置集，名称 VLAN1 ID 13，发出如下命令：

```
~]$ nmcli con add type vlan con-name VLAN1 dev enp2s0 id 13 ingress "2:3,3:5"
```

要查看与上述创建的 VLAN 关联的所有参数，请按如下所示发出命令：

```
~]$ nmcli connection show vlan-VLAN10
```

要更改 MTU，请按如下所示发出命令：

```
~]$ nmcli connection modify vlan-VLAN10 802.mtu 1496
```

MTU 设置决定了网络层数据包的最大大小。链路层帧可以附带的载荷的最大大小反过来限制了网络层 MTU。对于标准以太网帧，这意味着 MTU 为 1500 字节。设置 VLAN 时，无需更改 MTU，因为链路层标头的大小会增加 4 字节，以适应 802.1Q 标签。

编写本文时，connection.interface-name 和 vlan.interface-name 必须相同（如果已设置）。因此，必须使用 nmcli 的交互模式同时更改它们。要更改 VLAN 连接名称，请使用以下命令：

```
~]$ nmcli con edit vlan-VLAN10
nmcli> set vlan.interface-name superVLAN
nmcli> set connection.interface-name superVLAN
nmcli> save
nmcli> quit
```

nmcli 实用程序可用于设置和清除 ioctl 标志，以改变 802.1Q 代码函数的方式。NetworkManager 支持以下 VLAN 标记：

- 0x01 - 重新排序输出数据包标头
- 0x02 - 使用 GVRP 协议

- *0x04 - 接口及其 master 的松散绑定*

VLAN 的状态同步到父接口或主接口（在其上创建 VLAN 的接口或设备）的状态。如果父接口设置为“down”管理状态，则所有关联的 VLAN 将被设置，并且所有路由都会从路由表中清除。标志 0x04 可启用 松散绑定 模式，其中仅操作状态从父对象传递到关联的 VLAN，但不更改 VLAN 设备状态。

要设置 VLAN 标记，请按如下方式发出命令：

```
~]$ nmcli connection modify vlan-VLAN10 vlan.flags 1
```

有关 nmcli 简介，请参阅 第 3.3 节 “使用 nmcli 配置 IP 网络”。

10.4. 使用命令行配置 802.1Q VLAN 标记

在 Red Hat Enterprise Linux 7 中，默认载入 8021q 模块。如果需要，您可以以 root 用户身份运行以下命令来确保载入该模块：

```
~]# modprobe --first-time 8021q
modprobe: ERROR: could not insert '8021q': Module already in kernel
```

要显示模块信息，请运行以下命令：

```
~]$ modinfo 8021q
```

有关更多命令选项，请参阅 modprobe(8)手册页。

10.4.1. 使用 ifcfg 文件设置 802.1Q VLAN 标记

1. *在 /etc/sysconfig/network-scripts/ifcfg-device_name 中配置父接口，其中 device_name 是接口的名称：*

   ```
   DEVICE=interface_name
   TYPE=Ethernet
   BOOTPROTO=none
   ONBOOT=yes
   ```

2.

在 /etc/sysconfig/network-scripts/ 目录中配置 VLAN 接口配置。配置文件名称应当是父接口加上 a . 字符加上 VLAN ID 号。例如，如果 VLAN ID 为 192，并且父接口为 enp1s0，则配置文件名称应为 ifcfg-enp1s0.192 ：

```
DEVICE=enp1s0.192
BOOTPROTO=none
ONBOOT=yes
IPADDR=192.168.1.1
PREFIX=24
NETWORK=192.168.1.0
VLAN=yes
```

如果需要在同一接口 enp1s0 上配置第二个 VLAN（如 VLAN ID 193），请添加一个名称为 enp1s0.193 的新文件，并提供 VLAN 配置详细信息。

3. 重新启动网络服务，以使更改生效。作为 root 发出以下命令：

```
~]# systemctl restart network
```

10.4.2. 使用 ip 命令配置 802.1Q VLAN 标记

要在以太网接口 enp1s0 上创建一个 802.1Q VLAN 接口，名称为 VLAN8 和 ID 8，请以 root 身份发出命令，如下所示：

```
~]# ip link add link enp1s0 name enp1s0.8 type vlan id 8
```

要查看 VLAN，请运行以下命令：

```
~]$ ip -d link show enp1s0.8
4: enp1s0.8@enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP mode DEFAULT
    link/ether 52:54:00:ce:5f:6c brd ff:ff:ff:ff:ff:ff promiscuity 0
    vlan protocol 802.1Q id 8 <REORDER_HDR>
```

请注意，ip 实用程序将 VLAN ID 解释为十六进制值（如果它前面带有 0x ）和数值（如果具有前导 0 ）。这意味着，若要分配一个十进制值为 22 的 VLAN ID， 不得添加任何前导零。

要删除 VLAN，以 root 身份发出命令，如下所示：

```
~]# ip link delete enp1s0.8
```

要使用属于多个 VLAN 的多个接口，请在物理接口 enp1s0 上使用适当的 VLAN ID 创建本地 enp1s0.1 和 enp1s0.2 ：

```
~]# ip link add link enp1s0 name enp1s0.1 type vlan id 1
    ip link set dev enp1s0.1 up
~]# ip link add link enp1s0 name enp1s0.2 type vlan id 2
    ip link set dev enp1s0.2 up
```

请注意，在物理设备上运行网络嗅探器，您可以捕获到达物理设备的标记帧，即使 enp1s0 上没有配置 VLAN 设备也是如此。例如：

```
tcpdump -nnei enp1s0 -vvv
```

注意

如果系统关机或重启，在命令提示符处使用 ip 命令创建的 VLAN 接口将会丢失。要将 VLAN 接口配置为在系统重启后永久保留，请使用 ifcfg 文件。请查看 第 10.4.1 节 “使用 ifcfg 文件设置 802.1Q VLAN 标记”

10.5. 使用 GUI 配置 802.1Q VLAN 标记

10.5.1. 建立 VLAN 连接

您可以使用 nm-connection-editor 创建使用现有接口作为父接口的 VLAN。请注意，只有父接口设置为自动连接时，才会自动创建 VLAN 设备。

过程 10.1. 使用 nm-connection-editor 添加新 VLAN 连接

1. 在终端中输入 nm-connection-editor ：

```
~]$ nm-connection-editor
```

2. 单击添加按钮。此时将显示 Choose a Connection Type 窗口。选择 VLAN 并单击 Create。此时将显示 Editing VLAN 连接 1 窗口。

3. 在 VLAN 选项卡上，从您要用于 VLAN 连接的下拉列表中选择父接口。

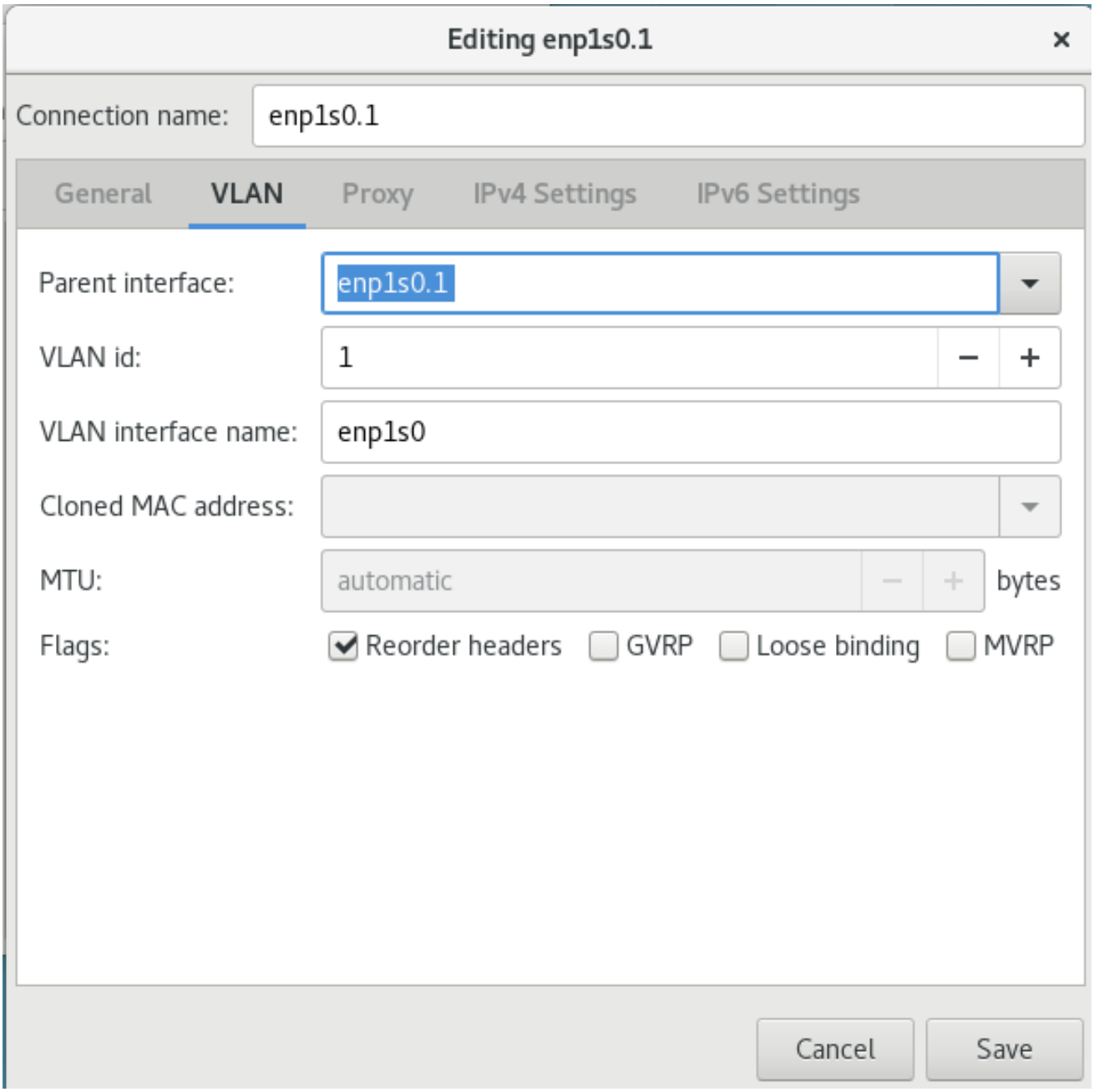
4. 输入 VLAN ID

5. 输入 VLAN 接口名称。这是将要创建的 VLAN 接口的名称。例如，enp1s0.1 或 vlan2。（通常这是父接口名称加上 “.” 和 VLAN ID，或者 “vlan” 加上 VLAN ID。）

6. 检查并确认设置，然后单击保存按钮。

7. 要编辑特定于 VLAN 的设置，请参阅 第 10.5.1.1 节 “配置 VLAN 选项卡”。

图 10.3. 使用 nm-connection-editor 添加新 VLAN 连接



[D]

过程 10.2. 编辑现有的 VLAN 连接

按照以下步骤编辑现有的 VLAN 连接。

1. **在终端中输入 nm-connection-editor ：**

```
~]$ nm-connection-editor
```

2. 选择您要编辑的连接并点击 Edit 按钮。

3. 选择常规选项卡。

4. 配置连接名称、自动连接行为和可用性设置。

   Editing 对话框中的这些设置对所有连接类型通用：

   - 连接名称 - 输入您的网络连接描述性名称。此名称将用于在网络窗口的 VLAN 部分中列出此连接。

   - 当这个网络可用时自动连接到这个网络 - 如果您希望 NetworkManager 在可用时自动连接到这个连接，请选择这个框。如需更多信息，请参阅 “使用 control-center 编辑现有连接”一节。

   - 可供所有用户使用 - 选择此复选框创建系统上所有用户可用的连接。更改此设置可能需要 root 特权。详情请参考 第 3.4.5 节 “使用 GUI 管理系统范围以及专用连接配置集”。

5. 要编辑特定于 VLAN 的设置，请参阅 第 10.5.1.1 节 “配置 VLAN 选项卡”。

保存您的新（或修改）连接并创建进一步配置

编辑完 VLAN 连接后，单击 Save 按钮以保存自定义配置。

然后，配置：

- 连接的 IPv4 设置，单击 IPv4 Settings 选项卡，再继续 第 5.4 节 “配置 IPv4 设置”。

  或者

- 连接的 IPv6 设置，单击 IPv6 Settings 选项卡，再继续 第 5.5 节 “配置 IPv6 设置”。

10.5.1.1. 配置 VLAN 选项卡

如果您已经添加了一个新的 VLAN 连接（具体步骤请参阅 过程 10.1, “使用 nm-connection-editor 添加新 VLAN 连接”），您可以编辑 VLAN 标签来设置父接口和 VLAN ID。

父接口

可在下拉列表中选择之前配置的接口。

VLAN ID

用于标记 VLAN 网络流量的标识号。

VLAN 接口名称

要创建的 VLAN 接口的名称。例如，enp1s0.1 或 vlan2。

克隆的 MAC 地址

（可选）设置用于标识 VLAN 接口的备用 MAC 地址。这可用于更改在此 VLAN 上发送的数据包的源 MAC 地址。

MTU

（可选）设置用于通过 VLAN 连接发送的数据包的最大传输单元(MTU)大小。

10.6. 使用 IP 命令绑定和桥接上的 VLAN

要通过绑定和网桥使用 VLAN，请按如下所示操作：

1. 以 root 用户身份添加绑定设备：

```
# ip link add bond0 type bond
# ip link set bond0 type bond miimon 100 mode active-backup
# ip link set em1 down
# ip link set em1 master bond0
# ip link set em2 down
# ip link set em2 master bond0
# ip link set bond0 up
```

2. 在绑定设备中设置 VLAN：

```
# ip link add link bond0 name bond0.2 type vlan id 2
# ip link set bond0.2 up
```

3. 添加桥接设备并为其附加 VLAN：

```
# ip link add br0 type bridge
# ip link set bond0.2 master br0
# ip link set br0 up
```

10.7. BOND 和 BRIDGE 上使用 NETWORKMANAGER 命令行工具 NMCLI 的 VLAN

要通过绑定和网桥使用 VLAN，请按如下所示操作：

1. 添加绑定设备：

```
~]$ nmcli connection add type bond con-name Bond0 ifname bond0 bond.options
"mode=active-backup,miimon=100" ipv4.method disabled ipv6.method ignore
```

请注意，在这种情况下，绑定连接仅充当 VLAN 的"低接口"，并且不获取任何 IP 地址。因此，已在命令行中 禁用ipv4.method 忽略参数和ipv6.method 忽略参数。

2. 在绑定设备中添加端口：

```
~]$ nmcli connection add type ethernet con-name Slave1 ifname em1 master bond0 slave-
type bond
~]$ nmcli connection add type ethernet con-name Slave2 ifname em2 master bond0 slave-
type bond
```

3. 添加桥接设备：

```
~]$ nmcli connection add type bridge con-name Bridge0 ifname br0 ip4 192.0.2.1/24
```

4. 在分配给网桥设备的绑定之上添加 VLAN 接口：

```
~]$ nmcli connection add type vlan con-name Vlan2 ifname bond0.2 dev bond0 id 2 master
br0 slave-type bridge
```

5. 查看创建的连接：

```
~]$ nmcli connection show
NAME     UUID                                  TYPE           DEVICE
Bond0    f05806fa-72c3-4803-8743-2377f0c10bed  bond           bond0
Bridge0  22d3c0de-d79a-4779-80eb-10718c2bed61  bridge         br0
Slave1   e59e13cb-d749-4df2-aee6-de3bfaec698c  802-3-ethernet  em1
Slave2   25361a76-6b3c-4ae5-9073-005be5ab8b1c  802-3-ethernet  em2
Vlan2    e2333426-eea4-4f5d-a589-336f032ec822  vlan           bond0.2
```

10.8. 配置 VLAN 切换端口模式

红帽企业 Linux 计算机通常用作路由器，并在其网络接口上启用高级 VLAN 配置。当以太网接口连接到 交换机并且通过物理接口运行 VLAN 时，您需要设置交换机端口模式。红帽企业 Linux 服务器或工作站通常仅连接到一个 VLAN，这使得切换模式访问适合和默认设置。

在某些情况下，多个标记的 VLAN 使用相同的物理链接，在交换机和 Red Hat Enterprise Linux 机器之间是以太网，这需要两端配置交换模式中继。

例如，当红帽企业 Linux 计算机用作路由器时，计算机需要将路由器后各种 VLAN 标记的数据包转发到同一物理以太网上的交换机，并保持这些 VLAN 之间的隔离。

在描述的设置中，例如 第 10.3 节 “使用命令行工具 nmcli 配置 802.1Q VLAN 标记”，使用 Cisco switchport 模式中继。如果您仅在接口上设置 IP 地址，请使用 Cisco switchport 模式访问。

10.9. 其它资源

- ip-link(8)手册页 - 描述 ip 实用程序的网络设备配置命令。
- nmcli(1) man page - 描述 NetworkManager 的命令行工具。
- nmcli-examples(5) 手册页 - 提供 nmcli 命令的示例.
- nm-settings(5) 手册页 - NetworkManager 连接的设置和参数的说明.
- nm-settings-ifcfg-rh(5) 手册页 - /etc/sysconfig/network-scripts/ifcfg-* 文件中的 ifcfg-rh 设置的说明。

第 11 章 一致的网络设备命名

红帽企业 Linux 为网络接口提供了一致且可预测的网络设备命名方法。这些功能更改了系统中网络接口名称，以便更轻松地查找和区分接口。

传统上，Linux 中的网络接口枚举为 eth[0123...]s0但是这些名称不一定与机箱上的实际标签对应。具有多个网络适配器的现代服务器平台，可能会遇到这些接口的命名不明确的情况。这会影响主板上嵌入的网络适配器（Lan-on-Motherboard 或LOM）和附加（单点和多端口）适配器。

在 Red Hat Enterprise Linux 中，udev 支持许多不同的命名方案。默认设置是基于固件、拓扑和位置信息分配固定名称。这具有完全自动化、完全可预知的优点，即即使在添加或删除硬件时仍能修复这些名称（不会重新枚举），而且可以无缝替换损坏的硬件。缺点是它们有时比. eth 或者 wla 传统名称.例如：enp5s0.

警告

不要禁用一致的网络设备命名，因为它允许系统使用 ethX 样式名称，其中 X 是与特定接口对应的唯一编号，在引导过程中可能具有不同的网络接口名称。如需了解更多详细信息，请参阅 第 11.10 节 “网络设备命名故障排除”。

11.1. 命名方案层次结构

默认情况下，systemd 将使用以下策略命名接口以应用支持的命名方案：

- 方案 1 ：合并固件或 BIOS 的名称提供板载设备的索引号(example: eno1)，如果固件或 BIOS 中的信息适用并可用，则应用该编号，否则返回方案 2。
- 方案 2 ：合并固件或 BIOS 的名称提供 PCI Express 热插拔插槽指数号（例如：ens1）如果固件或 BIOS 中的信息适用并可用，则将应用，否则回退到方案 3。
- 方案 3 ：合并硬件连接器的物理位置的名称（例如： enp2s0）会被应用（如果适用），否则在所有其他情况下直接回退到方案 5。
-

方案 4 ：默认情况下不使用合并接口的 MAC 地址（例如： enx78e7d1ea46da），但如果用户选择，则可用。

- 方案 5 ：传统的不可预测的内核命名方案（如果所有其他方法都失败）将使用 （例如： enp1s0）。

此策略（上面概述的步骤）是默认设置。如果系统启用了 biosdevname，则会使用它。请注意，启用 biosdevname 需要将 biosdevname=1 作为内核命令行参数传递，但使用 Dell 系统时除外，只要安装了 biosdevname，则会默认使用 biosdevname。如果用户添加了 udev 规则来更改内核设备的名称，则优先使用这些规则。

11.2. 了解设备重命名过程

设备名称过程如下：

1. /usr/lib/udev/rules.d/60-net.rules 中的规则指示 udev helper 实用程序 /lib/udev/rename_device 查看所有 /etc/sysconfig/network-scripts/ifcfg-后缀文件。如果找到具有与接口 MAC 地址匹配的 aHWADDR 条目的 ifcfg 文件，它会将接口重命名为 ifcfg 文件中由 DEVICE 指令指定的名称。

2. /usr/lib/udev/rules.d/71-biosdevname.rules 中的规则指示 biosdevname 根据命名策略重命名接口，只要在上一步中未重命名该接口，则会安装 biosdevname，在引导命令行中未提供 biosdevname=0 作为内核命令。

3. /lib/udev/rules.d/75-net-description.rules 中的规则指示 udev 通过检查网络接口设备来填写内部 udev 设备属性值 ID_NET_NAME_ONBOARD、ID_NET_NAME_SLOT、ID_NET_NAME_PATH、ID_NET_NAME_MAC。请注意，某些设备属性可能尚未定义。

4. 根据以下优先级，/usr/lib/udev/rules.d/80-net-name-slot.rules 中的规则指示 udev 重新命名该接口，但第 1 或 2 步中未对其进行重命名，且未提供内核参数 net.ifnames=0，这根据以下优先级：ID_NET_NAME_ONBOARD、ID_NET_NAME_SLOT、ID_NET_NAME_PATH。如果一个未设置，则会进入列表中的下一个。如果未设置这些接口，则不会重命名接口。

第 3 步和第 4 步正在实施命名方案 1、2、3 和可选 4，如 第 11.1 节 “命名方案层次结构” 所述。第 2 步在 第 11.6 节 “使用 biosdevname 的一致网络设备命名” 中进行了更详细的说明。

11.3. 了解可预测网络接口设备名称

名称具有基于接口类型的双字符前缀：

1. 以太网，
2. 用于无线 局域网(WLAN)，
3. 用于 无线局域网(WWAN)的 WW。

名称具有以下类型：

O<index>

板载设备索引号

s<slot>[f<function>][d<dev_id>]

热插拔插槽索引号.所有多功能 PCI 设备将在设备名称中承载 [f<功能>] 编号，包括功能 0 设备。

x<MAC>

MAC 地址

[P<domain>]p<bus>s<slot>[f<function>][d<dev_id>]

PCI 地理位置.在 PCI 地理位置中，只有在值不是 0 时才会提及 [P<domain>] 号。例如：

ID_NET_NAME_PATH=P1enp5s0

[P<domain>]p<bus>s<slot>[f<function>][u<port>][.][c<config>][i<interface>]

USB 端口号链.对于 USB 设备，hub 端口号的完整链由 hub 的端口号组成。如果名称超过最多 15 个字符数，则不会导出该名称。如果链中有多个 USB 设备，则隐藏 USB 配置描述符 (c1) 和 USB 接口描述符 (i0) 的默认值。

11.4. SYSTEM Z 上 LINUX 可用的网络设备命名方案

使用 bus-ID 在 System z 实例上的 Linux 中为网络接口创建可预测的设备名称。bus-ID 标识 s390 频道子系统中的一个设备。总线 ID 标识 Linux 实例范围内的设备。对于 CCW 设备，总线 ID 是设备的设备号，前面带有 0.n，其中 n 是子频道集 ID。例如： 0.1.0ab1。

设备类型以太网的网络接口命名如下：

```
enccw0.0.1234
```

设备类型为 SLIP 的网络设备命名如下：

```
slccw0.0.1234
```

使用 znetconf -c 命令或 lscs -a 命令显示可用的网络设备及其总线 ID。

表 11.1. System z 中的 Linux 设备名称类型

格式	描述
enccw*bus-ID*	设备类型以太网
slccw*bus-ID*	设备类型为 SLIP 的网络设备

11.5. VLAN 接口的命名方案

通常而言，格式为 VLAN 接口名称： interface-name。使用VLAN-ID。VLAN-ID 范围从 0 到 4096 个，最多四个字符，接口名称总数则限制为 15 个字符。最大接口名称长度由内核标头定义，是一个全局限制，会影响所有应用程序。

在 Red Hat Enterprise Linux 7 中，支持 VLAN 接口名称的四种命名约定：

VLAN 加上 VLAN ID

单词 vlan 加上 VLAN ID。例如：vlan0005

VLAN 加上 VLAN ID 但不填充

通过额外的前导零，单词 vlan 加上 VLAN ID 而无需 padding。例如：vlan5

设备名称加上 VLAN ID

父接口的名称加上 VLAN ID。例如： enp1s0.0005

设备名称加上 VLAN ID，但不进行 padding

父接口的名称加上 VLAN ID 而无需通过额外的前导零进行 padding。例如： enp1s0.5。

11.6. 使用 BIOSDEVNAME 的一致网络设备命名

此功能通过 biosdevname udev helper 实用程序实施，将更改现有网络接口、PCI 卡网络接口和虚拟功能网络接口的名称 eth[0123...] 表 11.2 “biosdevname 命名约定” 所示的新命名规则。请注意，除非系统是 Dell 系统，否则 biosdevname 被显式启用，如 第 11.6.2 节 “启用和禁用功能” 所述，systemd 命名方案将优先使用。

表 11.2. biosdevname 命名约定

设备	旧名称	新名称
嵌入式网络接口(LOM)	eth[0123...]	em[1234...][a]
PCI 卡网络接口	eth[0123...]	p<*slot*>p<*以太网端口*>[b]
虚拟功能	eth[0123...]	p<*slot*>p<*以太网端口*>_<*虚拟接口*>[c]

设备	旧名称	新名称

[a] 新的枚举从 **1** 开始.

[b] 例如：**p3p4**

[c] 例如：**p3p4_1**

11.6.1. 系统要求

biosdevname 程序使用系统 BIOS 的信息，特别是类型 9 (System Slot) 和类型 41 （板设备扩展信息）字段包含在 SMBIOS 中。如果系统的 BIOS 没有 SMBIOS 版本 2.6 或更高版本，且此数据不会使用，则不会使用新的命名规则。大多数较旧的硬件不支持此功能，因为缺少包含正确的 SMBIOS 版本和字段信息的 BIOS。有关 BIOS 或 SMBIOS 版本信息，请联系您的硬件供应商。

要使这个功能生效，还必须安装 biosdevname 软件包。 要安装它，以 root 身份运行以下命令：

```
~]# yum install biosdevname
```

11.6.2. 启用和禁用功能

要禁用此功能，在引导命令行中在安装过程中和安装后传递以下选项：

```
biosdevname=0
```

要启用此功能，在引导命令行中在安装过程中和安装后传递以下选项：

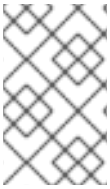
```
biosdevname=1
```

除非系统满足最低要求，否则这个选项将被忽略，系统将使用如章节开头所述的 systemd 命名方案。

如果指定了 biosdevname install 选项，它必须保留为系统生命周期的引导选项。

11.7. 管理员备注

许多系统自定义文件可以包括网络接口名称，因此如果将系统从旧约定移到新约定，则需要更新。如果您使用新的命名约定，您还需要更新自定义 iptables 规则、更改 irqbalance 的脚本和其他类似配置文件的网络接口名称。此外，启用此更改以进行安装需要修改通过 ksdevice 参数使用设备名称的现有 kickstart 文件；这些 kickstart 文件将需要更新，才能使用网络设备的 MAC 地址或网络设备的新名称。



注意

最大接口名称长度由内核标头定义，是一个全局限制，会影响所有应用程序。

11.8. 控制网络设备名称的选择

可通过以下方式控制设备命名：

通过识别网络接口设备

使用HWADDR 指令在 ifcfg 文件中设置 MAC 地址可使其由 udev 标识。名称取自 DEVICE 指令提供的字符串，按照惯例，该字符串与 ifcfg 后缀相同。例如，ifcfg-enp1s0。

通过打开或关闭 biosdevname

将使用 biosdevname 提供的名称（如果 biosdevname 可以确定一个名称）。

通过打开或关闭 systemd-udev 命名方案

将使用 systemd-udev 提供的名称（如果 systemd-udev 可以确定一个名称）。

11.9. 禁用一致的网络设备命名

要禁用一致的网络设备命名，建议仅在特殊情况下使用。如需更多信息，请参阅 第 11 章 一致的网络设备命名 和 第 11.10 节 “网络设备命名故障排除”。

要禁用一致的网络设备命名，请从以下选项之一中选择：

- 通过"masking"udev 的默认策略的规则文件禁用分配固定名称。这可以通过创建指向 /dev/null 的符号链接来完成。因此，将使用无法预测的内核名称。以 root 用户身份输入以下命令：

```
~]# ln -s /dev/null /etc/udev/rules.d/80-net-name-slot.rules
```

- 创建您自己的手动命名方案，例如命名您的接口 internet0、dmz0 or lan0。为此，请创建自己的 udev 规则文件，并为设备设置 NAME 属性。确保对默认策略文件上方的新文件进行排序，例如为其命名 /etc/udev/rules.d/70-my-net-names.rules。

- 更改默认策略文件以选择不同的命名方案，例如，默认为 MAC 地址后命名所有接口。作为 root 用户，复制默认策略文件，如下所示：

```
~]# cp /usr/lib/udev/rules.d/80-net-name-slot.rules /etc/udev/rules.d/80-net-name-slot.rules
```

  编辑 /etc/udev/rules.d/ 目录中的文件并根据需要更改行。

- 打开 /etc/default/grub 文件并查找 GRUB_CMDLINE_LINUX 变量。

注意

GRUB_CMDLINE_LINUX 是一个变量，其中包含添加到内核命令行中的条目。它可能已经包含其他配置，具体取决于您的系统设置。

将 net.ifnames=0 和 biosdevname=0 作为内核参数值添加到 GRUB_CMDLINE_LINUX 变量：

```
~]# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel_7/swap rd.luks.uuid=luks-cc387312-6da6-
469a-8e49-b40cd58ad67a crashkernel=auto  vconsole.keymap=us
vconsole.font=latarcyrheb-sun16 rd.lvm.lv=rhel_7/root rhgb quiet net.ifnames=0
biosdevname=0"
GRUB_DISABLE_RECOVERY="true"
```

运行 grub2-mkconfig 命令重建 /boot/grub2/grub.cfg 文件：

```
~]# grub2-mkconfig -o /boot/grub2/grub.cfg
```

注意

对于使用 UEFI 引导的系统：

```
~]# grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

查看当前的设备名称。例如，eno1：

```
~]# nmcli connection show
NAME   UUID                                  TYPE           DEVICE
Wired  63cba8b2-60f7-4317-bc80-949e800a23cb  802-3-ethernet  eno1
```

将设备名称修改为 enp1s0，并重启系统：

```
~]# nmcli connection modify Wired connection.interface-name enp1s0
```

```
~]# reboot
```

grubby 实用程序用于更新和显示关于 grub 引导加载器的配置文件的信息。详情请查看 grubby(8) 手册页。有关使用 GRUB2 的更多信息，请参阅 Red Hat Enterprise Linux 系统管理员指南。

11.10. 网络设备命名故障排除

将根据 第 11.2 节 “了解设备重命名过程” 中描述的步骤为每个接口分配可预测的接口名称（如果适用）。要查看 udev 将使用的可能名称列表，以 root 用户身份以以下格式发出命令 ：

```
~]# udevadm info /sys/class/net/ifname | grep ID_NET_NAME
```

其中 ifname 是以下命令列出的接口之一：

```
~]$ ls /sys/class/net/
```

udev 将根据 第 11.2 节 “了解设备重命名过程” 中描述的规则应用其中一个可能的名称，并在此总结：

- /usr/lib/udev/rules.d/60-net.rules - 从 initscripts 开始，
- /usr/lib/udev/rules.d/71-biosdevname.rules - from biosdevname,
- /usr/lib/udev/rules.d/80-net-name-slot.rules - from systemd

从上面的规则文件列表中，可以看到，如果通过 initscripts 或 biosdevname 进行接口命名，它总是优先于 udev 原生命名。但是，如果没有出现 initscripts 重命名并禁用 biosdevname，那么要更改接口名称，将 80-net-name-slot.rules 从 /usr 复制到 /etc，并相应地编辑该文件。换句话说，注释掉或安排按特定顺序使用的方案。

例 11.1. 某些接口来自内核命名空间(eth[0,1,2...])，而 Others Are Successfully Renamed 为 udev

混合方案最可能意味着，对于某些硬件，内核无法向 udev 提供可用信息，因此无法找出任何名称，或者提供给 udev 的信息并不合适，例如非唯一设备 ID。后者更为常见，解决方案是在 ifcfg 文件中使用自定义命名方案，或通过编辑 80-net-name-slot.rules 更改正在使用的 udev 方案。

例 11.2. 在 /var/log/messages 或 systemd Journal 中，Renaming Is Seen to Be Done Twice for E每个接口

带有命名方案的系统在 ifcfg 文件中编码，但没有重新生成的 initrd 镜像可能会遇到这个问题。在

早期启动期间，仍使用 initrd 分配接口名称（通过 biosdevname 或 udev 或 dracut 参数）。然后，在切换到真实 rootfs 后，将再次执行重命名，并且新接口名称由 udev 生成的 /usr/lib/udev/rename_device 二进制文件决定，因为需要处理 60-net.rules。您可以安全地忽略此类消息。

例 11.3. 在 ifcfg 文件中使用 ethX 名称不起作用的命名方案

Red Hat Enterprise Linux 不提供持续应用 ethX 命名约定的方法，除非在非常特殊的情况下。

如果某个其他接口已在使用请求的名称，将接口设置为特定名称的 udev 规则将失败。这包括 /usr/lib/udev/rules.d/60-net.rules 文件提供的功能。

当内核枚举网络设备时，内核在引导时使用 ethX 命名约定。ethX 名称在不同的重新启动之间不一致，因此它们无法预测。因此，尝试使用 udev 将接口重命名为可预测的名称或重新排序内核给出的不可预测的 ethX 名称会失败。

在以下情况下，使用 ethX 名称可以正常工作：

- 系统只有一个网络接口。
- 用于红帽企业 Linux 7 虚拟机中的 virtio NIC.请参阅《虚拟化部署和管理指南》中的 KVM 半虚拟化(virtio)驱动程序和网络配置章节

例 11.4. 在 Inconsistent enpXxX Names 中设置 net.ifnames=0 结果

如果 systemd 可预测的接口命名(net.ifnames)和 biosdevname 命名方案都已禁用，网络接口继续使用最初由内核提供的 ethX 名称无法预计且可能不一致。

当内核枚举网络设备时，内核始终在启动时使用 enpXxX 命名约定。由于并行化，内核接口枚举顺序应该在每次重启后会有所不同。Red Hat Enterprise Linux 依赖于 systemd 可预测的接口命名方案或 biosdevname 命名方案，以可预测的方式将内核无法预计的 ethX 接口重命名为在重新引导后始终一致的名称。

有关网络适配器命名约定的更多信息，请参阅在 RHEL7 中设置 net.ifnames=0?红帽客户门户上的知识中心支持文章.

例 11.5. 以太网接口前缀的限制

您选择的前缀必须满足以下要求：

- 它由 ASCII 字符组成。
- 它是一个字母数字字符串。
- 它少于 16 个字符。
- 它不与用于网络接口命名的任何其他已知的前缀冲突，如 eth、eno、ens 和 em。

11.11. 其它资源

安装的文档

- udev(7) 手册页 - 描述 Linux 动态设备管理守护进程 udevd。
- systemd(1) man page - 描述 systemd 系统和服务管理器。
- biosdevname(1) man page - 描述用于获取设备的 BIOS 给定名称的实用程序。

在线文档

- 红帽企业 Linux 7 上的 IBM 知识库 SC34-2710-00 设备驱动程序、功能和命令包括用于 IBM System z “设备和附加的可预测网络设备名称的信息”。

第 12 章 配置基于策略的路由来定义备用路由

默认情况下，RHEL 中的内核决定使用路由表根据目标地址转发网络数据包。基于策略的路由允许您配置复杂的路由场景。例如，您可以根据各种条件来路由数据包，如源地址、数据包元数据或协议。

本节论述了如何使用 NetworkManager 配置基于策略的路由。

注意

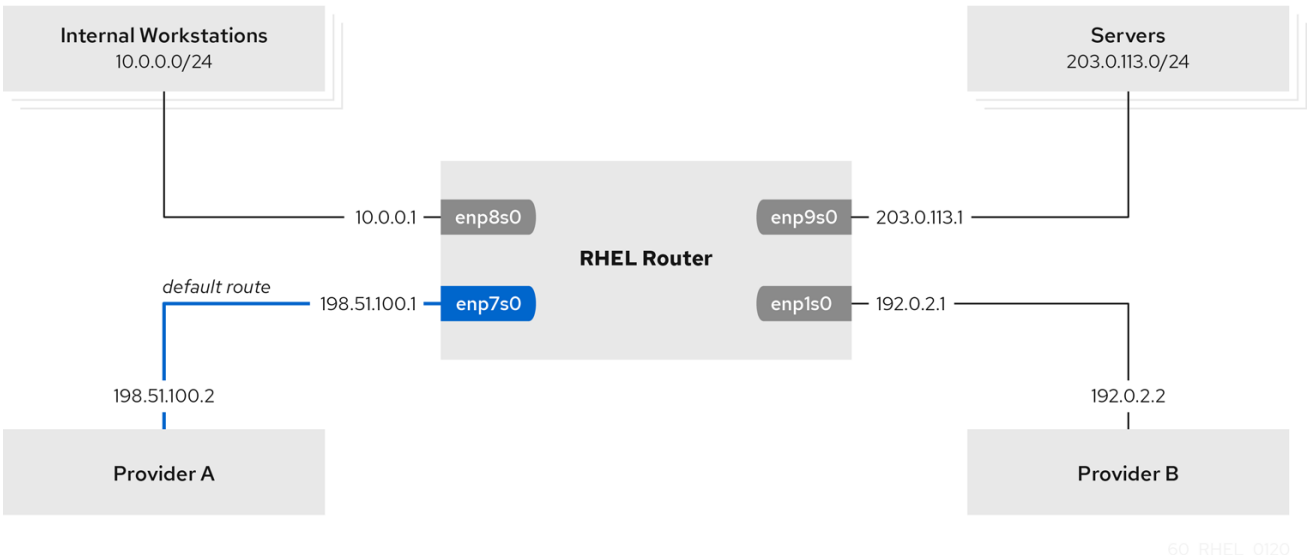
在使用 NetworkManager 的系统中，只有 nmcli 实用程序支持设置路由规则，并将路由分配到特定表。

12.1. 将流量从特定子网路由到不同的默认网关

本节论述了如何将 RHEL 配置为默认路由将所有流量路由到互联网供应商 A 的路由器。使用基于策略的路由，RHEL 会将从内部工作站子网接收的流量路由到供应商 B。

该流程假设以下网络拓扑：

图 12.1. 激活连接



先决条件

- 要在流程中设置的 RHEL 路由器有四个网络接口：

- *enp7s0 接口连接到供应商 A 的网络。提供商网络中的网关 IP 是 198.51.100.2，网络使用 a/30 网络掩码。*
- *enp1s0 接口连接到提供商 B 的网络。提供商网络中的网关 IP 是 192.0.2.2 ，网络使用一个/30 网络掩码。*
- *enp8s0 接口通过内部工作站连接到 10.0.0.0/24 子网。*
- *enp9s0 接口与公司的服务器一起连接到 203.0.113.0/24 子网。*

- *内部工作站子网中的主机使用 10.0.0.1 作为默认网关。在此过程中，您要将此 IP 地址分配给路由器的 enp8s0 网络接口。*
- *服务器子网中的主机使用 203.0.113.1 作为默认网关。在此过程中，您要将此 IP 地址分配给路由器的 enp9s0 网络接口。*
- *firewalld 服务已启用并激活，这是默认设置。*

流程

1. *将网络接口配置为供应商 A：*

```
# nmcli connection add type ethernet con-name Provider-A ifname enp7s0 ipv4.method manual ipv4.addresses 198.51.100.1/30 ipv4.gateway 198.51.100.2 ipv4.dns 198.51.100.200 connection.zone external
```

nmcli connection add 命令创建 NetworkManager 连接配置集。以下列表描述了该命令的选项：

- *键入 ethernet ：定义连接类型为 Ethernet。*
- *con-name connection_name ：设置配置集的名称。使用有意义的名称以避免混淆。*

- ifname network_device ：设置网络接口。
- ipv4.method 手册 ：启用配置静态 IP 地址。
- ipv4.addresses IP_address/subnet_mask ：设置 IPv4 地址和子网掩码。
- ipv4.gateway IP_address ：设置默认网关地址。
- ipv4.dns IP_of_DNS_server ：设置 DNS 服务器的 IPv4 地址。
- connection.zone firewalld_zone ：将网络接口分配给定义的 firewalld 区域。请注意，firewalld 自动启用分配给外部区的伪装接口。

2. 将网络接口配置为供应商 B:

```
# nmcli connection add type ethernet con-name Provider-B ifname enp1s0 ipv4.method manual ipv4.addresses 192.0.2.1/30 ipv4.routes "0.0.0.0/1 192.0.2.2 table=5000, 128.0.0.0/1 192.0.2.2 table=5000" connection.zone external
```

这个命令使用 ipv4.routes 参数而不是 ipv4.gateway 来设置默认网关。这需要将此连接的默认网关分配到不同于默认值的路由表(5000)。当连接被激活时，NetworkManager 会自动创建这个新的路由表。

注意

nmcli 实用程序不支持将0.0.0.0/0 用于 ipv4.gateway 中的默认网关。为临时解决这个问题，命令为 0.0.0.0/1 和 128.0.0.0/1 子网创建单独的路由，其中也包括完整的 IPv4 地址空间。

3. 将网络接口配置为内部工作站子网：

```
# nmcli connection add type ethernet con-name Internal-Workstations ifname enp8s0 ipv4.method manual ipv4.addresses 10.0.0.1/24 ipv4.routes "10.0.0.0/24 src=192.0.2.1 table=5000" ipv4.routing-rules "priority 5 from 10.0.0.0/24 table 5000" connection.zone trusted
```

这个命令使用 **ipv4.routes** 参数将静态路由添加到路由表中 **ID** 为 **5000**。**10.0.0.0/24** 子网的这一静态路由使用本地网络接口的 **IP** 地址到供应商 **B(192.0.2.1)**作为下一跃点。

另外，该命令使用 **ipv4.routing-rules** 参数添加优先级为 **5** 的路由规则，该规则将 **10.0.0.0/24** 子网的流量路由到表 **5000**。低的值具有更高的优先级。

请注意，**ipv4.routing-rules** 参数中的语法与 **ip route add** 命令中的语法相同，但 **ipv4.routing-rules** 始终需要指定优先级。

4. 将网络接口配置为服务器子网：

```
# nmcli connection add type ethernet con-name Servers ifname enp9s0 ipv4.method manual
ipv4.addresses 203.0.113.1/24 connection.zone trusted
```

验证步骤

1. 在内部工作站子网的 **RHEL** 主机上：

   1. 安装 **traceroute** 软件包：

      ```
      # yum install traceroute
      ```

   2. 使用 **traceroute** 工具显示到互联网主机的路由：

      ```
      # traceroute redhat.com
      traceroute to redhat.com (209.132.183.105), 30 hops max, 60 byte packets
       1  10.0.0.1 (10.0.0.1)     0.337 ms  0.260 ms  0.223 ms
       2  192.0.2.1 (192.0.2.1)   0.884 ms  1.066 ms  1.248 ms
       ...
      ```

      命令的输出显示路由器通过 **192.0.2.1**（提供商 **B** 的网络）发送数据包。

2. 在服务器子网的 **RHEL** 主机上：

   1. 安装 **traceroute** 软件包：

```
# yum install traceroute
```

2. 使用 **traceroute** 工具显示到互联网主机的路由：

```
# traceroute redhat.com
traceroute to redhat.com (209.132.183.105), 30 hops max, 60 byte packets
 1  203.0.113.1 (203.0.113.1)    2.179 ms  2.073 ms  1.944 ms
 2  198.51.100.2 (198.51.100.2)  1.868 ms  1.798 ms  1.549 ms
 ...
```

命令的输出显示路由器通过 **198.51.100.2** 发送数据包，即供应商 **A** 的网络。

故障排除步骤

在 RHEL 路由器中：

1. 显示规则列表：

```
# ip rule list
0: from all lookup local
5: from 10.0.0.0/24 lookup 5000
32766: from all lookup main
32767: from all lookup default
```

2. 显示表 **5000** 中的路由：

```
# ip route list table 5000
0.0.0.0/1 via 192.0.2.2 dev enp1s0 proto static metric 100
10.0.0.0/24 dev enp8s0 proto static scope link src 192.0.2.1 metric 102
128.0.0.0/1 via 192.0.2.2 dev enp1s0 proto static metric 100
```

3. 显示将哪些接口分配给哪些防火墙区：

```
# firewall-cmd --get-active-zones
external
  interfaces: enp1s0 enp7s0
trusted
  interfaces: enp8s0 enp9s0
```

4. 验证外部区是否启用了伪装 ：

```
# firewall-cmd --info-zone=external
external (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp1s0 enp7s0
  sources:
  services: ssh
  ports:
  protocols:
  masquerade: yes
  ...
```

其它资源

- **有关您可以在 nmcli connection add 命令中设置的 nm-settings(5) 参数的详情，请查看 ipv4.* man page 中的 IPv4 设置 部分。**
- **有关您可以在 nmcli connection add 命令中设置的 nm-settings(5) 参数的详情，请查看 connection.* man page 中的 Connection 设置 部分。**
- **有关使用 nmcli 管理 NetworkManager 连接的详情，请查看 nmcli(1) man page 中的 连接管理命令 部分。**

部分 III. INFINIBAND 和 RDMA 网络

这部分论述了如何通过 InfiniBand 网络连接设置 RDMA、InfiniBand 和 IP。

第 13 章 配置 INFINIBAND 和 RDMA 网络

13.1. 了解 INFINIBAND 和 RDMA 技术

Infiniband 指的是两个不同的因素。第一个是 InfiniBand 网络的物理链路层协议。第二个是更高级别的编程 API，称为 InfiniBand Verbs API。InfiniBand Verbs API 是一个远程直接访问(RDMA)技术。

RDMA 提供从一台计算机的内存直接访问另一台计算机的内存，而不涉及计算机的操作系统。这个技术启用了高吞吐量、低延迟的网络，该网络使用率较低，在大规模并行计算机集群中特别有用。

在典型的 IP 数据传输中，计算机 A 上的应用程序 X 会将一些数据发送到计算机 B 上的应用程序 Y。作为传输的一部分，机器 B 上的内核必须首先接收数据，解码数据包标头，确定数据属于应用程序 Y，唤醒应用程序 Y，等待应用程序 Y 执行读取系统调用到内核中，然后它必须手动将内核本身的内部内存空间中的数据复制到应用程序 Y 提供的缓冲区中。此过程意味着大多数网络流量必须至少复制到系统的主内存总线上（当主机适配器使用 DMA 将数据放入内核提供的内存缓冲区时，以及内核将数据移动到应用程序内存缓冲时），这意味着计算机必须执行大量上下文切换以在内核上下文和应用程序 Y 上下文之间切换。当网络流量以非常高的速度流动并可能导致其他任务减慢时，这两种情况都会给系统带来非常高的 CPU 负载。

RDMA 通信不同于普通 IP 通信，因为它们绕过内核在通信过程中的干预，并大大降低了处理网络通信通常需要的 CPU 开销。RDMA 协议允许机器中的主机适配器知道数据包何时来自网络，哪个应用程序应接收该数据包，并在应用程序的内存空间中移动。它不将数据包发送到内核，然后将其复制到用户应用的内存中，而是将数据包的内容直接置于应用的缓冲区中，而无需进一步干预。但是，无法使用大多数 IP 网络应用程序所基于的标准 Berkeley Sockets API 来实现，因此它必须提供自己的 API、InfiniBand Verbs API 和应用程序，然后才能直接使用 RDMA 技术。

Red Hat Enterprise Linux 7 支持 InfiniBand 硬件和 InfiniBand Verbs API。另外，还有另外两种支持的技术允许在非 InfiniBand 硬件中使用 InfiniBand Verbs API：

- 互联网广域 RDMA 协议(iWARP)

  iWARP 是一种计算机网络协议，实施远程直接内存访问(RDMA)，以通过 Internet 协议(IP)网络进行高效数据传输。

- RDMA over Converged Ethernet(RoCE)协议，之后通过以太网(IBoE)重命名为 InfiniBand。

RoCE 是一种网络协议，允许通过以太网网络进行远程直接内存访问(RDMA)。

先决条件

iWARP 和 RoCE 技术都具有普通的 IP 网络链路层作为其底层技术，因此它们的大部分配置实际上都包含在 第 3 章 配置 IP 网络 中。 在大多数情况下，一旦正确配置了 IP 网络功能，它们的 RDMA 功能都是自动的，只要安装了适用于硬件的正确驱动程序，就会显示。 每个红帽提供的内核都始终包括内核驱动程序，但是，如果在机器安装时未选择 InfiniBand 软件包组，则必须手动安装用户空间驱动程序。

从 Red Hat Enterprise Linux 7.4 开始，所有 RDMA 用户空间驱动程序都合并到 rdma-core 软件包中。要安装所有支持的 iWARP、RoCE 或 InfiniBand 用户空间驱动程序，以 root 用户身份输入：

```
~]# yum install libibverbs
```

如果您使用基于 Priority Flow Control(PFC)和 mlx4 的卡，请编辑 /etc/modprobe.d/mlx4.conf “以指示为以太网上未丢弃服务配置哪些数据包优先级的驱动程序”，这些卡将插入到并重新构建 initramfs 以包含修改的文件。较新的基于 mlx5 的卡会自动与交换机协商 PFC “设置，不需要任何模块选项告知它们未丢弃优先级或优先级”。

要将 Mellanox 卡设置为在以太网模式中使用一个或多个端口，请参阅 第 13.5.4 节 “为以太网操作配置 Mellanox 卡”。

安装了这些驱动程序软件包（除了通常安装任何 InfiniBand 的普通 RDMA 软件包外，用户应该能够利用大多数正常的 RDMA 应用程序来测试和查看其适配器上发生的 RDMA 协议通信。 但是，并非红帽企业 Linux 7 中包含的所有程序都正确支持 iWARP 或 RoCE/IBoE 设备。 这是因为 iWARP 上的连接建立协议与实际 InfiniBand 链路层连接上的不同。 如果相关的程序使用 librdmacm 连接管理库，它会以静默的方式处理 iWARP 和 InfiniBand 之间的差异，并且程序应正常工作。 如果应用尝试执行自己的连接管理，则必须特别支持 iWARP 或其他它不起作用。

13.2. 使用 ROCE 传输数据

RDMA over Converged Ethernet(RoCE)是一个网络协议，它允许通过以太网网络进行远程直接内存访问(RDMA)。RoCE v1 和 RoCE v2 有两个版本，具体取决于所使用的网络适配器。

RoCE v1

RoCE v1 协议是一个带有 ethertype 0x8915 的以太网链路层协议，它允许同一以太网广播域中任何两个主机之间进行通信。使用 ConnectX-3 网络适配器时，RoCE v1 是 RDMA 连接管理器

(RDMA_CM)的默认版本。

RoCE v2

RoCE v2 协议在 IPv4 或 IPv6 协议的 UDP 上存在。RoCE v2 保留 UDP 目的地端口号 4791。从 Red Hat Enterprise Linux 7.5 开始，RoCE v2 是 RDMA_CM 的默认版本，在使用 ConnectX-3 Pro、ConnectX-4、ConnectX-4 Lx 和 ConnectX-5 网络适配器时。硬件支持 RoCE v1 和 RoCE v2。

RDMA 连接管理器(RDMA_CM)用于在客户端和服务器之间设置可靠连接以传输数据。RDMA_CM 为建立连接提供了一个与 RDMA 传输相关的接口。通信通过特定的 RDMA 设备，数据传输基于消息。

先决条件

RDMA_CM 会话需要以下之一：

- 客户端和服务器都支持相同的 RoCE 模式。
- 客户端支持 RoCE v1 和服务器 RoCE v2。

由于客户端决定连接模式，因此可以出现以下情况：

连接成功：

如果客户端处于 RoCE v1 或 RoCE v2 模式，取决于使用的网卡和驱动程序，对应的服务器必须具有相同的版本才能创建连接。另外，如果客户端处于 RoCE v1，并且服务器采用 RoCE v2 模式，则连接会成功。

连接失败：

如果客户端在 RoCE v2 中，并且对应的服务器在 RoCE v1 中，则无法建立连接。在本例中，更新对应服务器的驱动程序或网络适配器，请参考 第 13.2 节 “使用 RoCE 传输数据”

表 13.1. 使用 RDMA_CM 的 RoCE 版本默认值

客户端	Server	默认设置
RoCE v1	RoCE v1	连接

客户端	Server	默认设置
RoCE v1	RoCE v2	连接
RoCE v2	RoCE v2	连接
RoCE v2	RoCE v1	无连接

客户端上的 RoCE v2 和服务器上的 RoCE v1 不兼容。要解决这个问题，请强制服务器和客户端环境通过 RoCE v1 进行通信。这意味着强制支持 RoCE v2 使用 RoCE v1 的硬件：

过程 13.1. 当在 Roce v2 中运行的硬件 AIready 时更改默认 RoCE 模式

1. 进入 /sys/kernel/config/rdma_cm 目录，使其具有 RoCE 模式：

```
~]# cd /sys/kernel/config/rdma_cm
```

2. 输入带有以太网网络设备的ibstat 命令显示状态。例如，对于 mlx5_0:

```
~]$ ibstat mlx5_0
   CA 'mlx5_0'
      CA type: MT4115
      Number of ports: 1
      Firmware version: 12.17.1010
      Hardware version: 0
      Node GUID: 0x248a0703004bf0a4
      System image GUID: 0x248a0703004bf0a4
      Port 1:
         State: Active
         Physical state: LinkUp
         Rate: 40
         Base lid: 0
         LMC: 0
         SM lid: 0
         Capability mask: 0x04010000
         Port GUID: 0x268a07fffe4bf0a4
         Link layer: Ethernet
```

3. 为 mlx5_0 设备创建一个目录：

```
~]# mkdir mlx5_0
```

4. 在 default_roce_mode 文件中以树格式显示 RoCE 模式：

```
~]# cd mlx5_0
```

```
~]$ tree
└── ports
    └── 1
        ├── default_roce_mode
        └── default_roce_tos
```

```
~]$ cat /sys/kernel/config/rdma_cm/mlx5_0/ports/1/default_roce_mode
  RoCE v2
```

5. 更改默认 RoCE 模式：

```
~]# echo "RoCE v1" > /sys/kernel/config/rdma_cm/mlx5_0/ports/1/default_roce_mode
```

6. 查看更改：

```
~]$ cat /sys/kernel/config/rdma_cm/mlx5_0/ports/1/default_roce_mode
  RoCE v1
```

13.3. 配置 SOFT-ROCE

RoCE 可以在硬件和软件中实施。soft-RoCE 是 RDMA 传输的软件实施。

先决条件

从 Red Hat Enterprise Linux 7.4 开始，Soft-RoCE 驱动程序已合并到内核中。用户空间驱动程序也合并到 rdma-core 软件包中。soft-RoCE 也称为 RXE。要启动、停止和配置 RXE，请使用 rxe_cfg 脚本。要查看 rxe_cfg 的选项，请输入 rxe_cfg 帮助。

过程 13.2. 配置 Soft-RoCE

1. 以 root 用户身份，输入以下命令显示 RXE 的当前配置状态：

```
~]# rxe_cfg
 rdma_rxe module not loaded
 Name      Link  Driver   Speed   NMTU  IPv4_addr  RDEV  RMTU
 igb_1     yes   igb
 mlx4_1    no    mlx4_en
 mlx4_2    no    mlx4_en
```

2. 要载入 RXE 内核模块并启动 RXE，以 root 用户身份输入：

```
~]# rxe_cfg start
 Name      Link  Driver   Speed   NMTU  IPv4_addr  RDEV  RMTU
 igb_1     yes   igb
 mlx4_1    no    mlx4_en
 mlx4_2    no    mlx4_en
```

另外，要验证是否载入了 RXE 内核模块，请输入：

```
~]# lsmod |grep rdma_rxe
 rdma_rxe            111129  0
 ip6_udp_tunnel       12755  1 rdma_rxe
 udp_tunnel           14423  1 rdma_rxe
 ib_core             236827  15
rdma_cm,ib_cm,iw_cm,rpcrdma,mlx4_ib,ib_srp,ib_ucm,ib_iser,ib_srpt,ib_umad,ib_uve
rbs,rdma_rxe,rdma_ucm,ib_ipoib,ib_isert
```

3. 在通过以太网接口添加新 RXE 设备之前，应打开对应的接口并分配有有效的 IP 地址。添加新的 RXE 设备，如 igb_1 ：

```
~]# rxe_cfg add igb_1
```

```
~]# rxe_cfg add igb_1
```

```
~]# rxe_cfg status
 Name      Link  Driver   Speed   NMTU  IPv4_addr  RDEV  RMTU
 igb_1     yes   igb                    rxe0  1024  (3)
 mlx4_1    no    mlx4_en
 mlx4_2    no    mlx4_en
```

RDEV 列中的 rxe0 表示为 igb_1 设备启用了 rxe。

4. 要验证 RXE 设备的状态，请使用 ibv_devices 命令：

```
~]# ibv_devices
 device              node GUID
 ------          ----------------
 mlx4_0          0002c90300b3cff0
 rxe0            a2369ffffe018294
```

或者，输入 aibstat 作为详细状态：

```
~]# ibstat rxe0
CA 'rxe0'
   CA type:
   Number of ports: 1
   Firmware version:
   Hardware version:
   Node GUID: 0xa2369ffffe018294
   System image GUID: 0x0000000000000000
   Port 1:
      State: Active
      Physical state: LinkUp
      Rate: 2.5
      Base lid: 0
      LMC: 0
      SM lid: 0
      Capability mask: 0x00890000
      Port GUID: 0xa2369ffffe018294
      Link layer: Ethernet
```

删除 RXE 设备

如果要删除 RXE 设备，请输入：

```
~]# rxe_cfg remove igb_1
```

验证 RXE 设备的连接性

以下示例演示了如何在服务器和客户端端验证 RXE 设备的连接。

例 13.1. 在服务器幻灯片中验证 RXE 设备的连接性

```
~]$ ibv_rc_pingpong -d rxe0 -g 0
  local address:  LID 0x0000, QPN 0x000012, PSN 0xe2965f, GID fe80::290:faff:fe29:486a
  remote address: LID 0x0000, QPN 0x000011, PSN 0x4bf206, GID fe80::290:faff:fe29:470a
8192000 bytes in 0.05 seconds = 1244.06 Mbit/sec
1000 iters in 0.05 seconds = 52.68 usec/iter
```

例 13.2. 在客户端幻灯片中验证 RXE 设备的连接性

```
~]$ ibv_rc_pingpong -d rxe0 -g 0 172.31.40.4
  local address:  LID 0x0000, QPN 0x000011, PSN 0x4bf206, GID fe80::290:faff:fe29:470a
  remote address: LID 0x0000, QPN 0x000012, PSN 0xe2965f, GID fe80::290:faff:fe29:486a
8192000 bytes in 0.05 seconds = 1245.72 Mbit/sec
1000 iters in 0.05 seconds = 52.61 usec/iter
```

13.4. INFINIBAND 和 RDMA 相关软件包

因为 RDMA 应用程序与基于 Berkeley 的应用程序和普通 IP 网络如此不同，所以 IP 网络中使用的大多数应用程序无法直接在 RDMA 网络上使用。红帽企业 Linux 7 随附多种不同的软件包，用于 RDMA 网络管理、测试和调试、高级别软件开发 API 以及性能分析。

要使用这些网络，需要安装这些软件包的部分或全部（此列表并不详尽，但涵盖了与 RDMA 相关的最重要的软件包）。

所需的软件包：

- rdma - 负责 RDMA 堆栈的内核初始化。
- libibverbs - 提供 InfiniBand Verbs API。
- opensm - 子网管理器（仅在一台机器上需要，且仅在该结构上没有激活子网管理器的情况下）。
- 安装的硬件的用户空间驱动程序 - infinipath-psm、libcxgb3、libcxgb4、libehca、libipathverbs、libmthca、libmlx4、libmlx5、libnes 和 libocrdma。请注意，libehca 仅适用于 IBM Power 系统服务器。

推荐的软件包：

- librdmacm、librdmacm-utils 和 ibacm - 连接管理库了解 InfiniBand、iWARP 和 RoCE 之间的不同，能够在所有这些硬件类型中正确打开连接，一些简单的测试程序用于验证网络的操作，以及与库集成的缓存守护进程，以便在大型集群中加快远程主机解析速度。
- libibverbs-utils - 基于 Verbs 的简单程序，用于查询安装的硬件和通过光纤验证通信。
- infiniband-diags 和 ibutils - 为 InfiniBand 光纤管理提供很多有用的调试工具。它们仅提供 iWARP 或 RoCE 的功能，因为大多数工具在 InfiniBand 链路层工作，而不是 Verbs API 层。
- perftest 和 qperf - 各种 RDMA 通信的性能测试应用程序。

可选软件包：

这些软件包在 Optional 频道中可用。在从可选频道安装软件包前，请参阅覆盖范围详情。有关订阅可选频道的信息，请参阅红帽知识库解决方案如何访问可选和补充频道。

- dapl、dapl-devel 和 dapl-utils - 为 RDMA 提供与 Verbs API 不同的 API。这些软件包同时有一个运行时组件和开发组件。

- openmpi、mvapich2 和 mvapich2-psm - 能够使用 RDMA 通信的 MPI 堆栈。写入这些堆栈的用户空间应用程序不一定知道 RDMA 通信正在发生。

13.5. 配置基本 RDMA 子系统

rdma 服务的启动是自动的。当 RDMA 的硬件（无论是 InfiniBand 或 iWARP 或 RoCE/IBoE）被检测到时，udev 会指示 systemd 启动rdma 服务。

```
~]# systemctl status rdma
● rdma.service - Initialize the iWARP/InfiniBand/RDMA stack in the kernel
   Loaded: loaded (/usr/lib/systemd/system/rdma.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: file:/etc/rdma/rdma.conf
```

用户不需要启用rdma 服务，但如果他们想随时强制使用该服务。要做到这一点，以 root 用户身份输入以下命令：

```
~]# systemctl enable rdma
```

13.5.1. 配置 rdma.conf 文件

Therdma 服务读取 /etc/rdma/rdma.conf 找出管理员默认载入的内核级和用户级 RDMA 协议。用户应编辑此文件，以打开或关闭各种驱动程序。

可以启用或禁用的各种驱动程序有：

- IPoIB - 这是一个 IP 网络模拟层，允许 IP 应用程序通过 InfiniBand 网络运行。

- SRP - 这是 SCSI 请求协议。它允许计算机挂载通过SRP 协议导出的远程驱动器或驱动器阵列，就像是本地硬盘一样。

- SRPT - 这是SRP 协议的目标模式或服务器模式。这会加载导出驱动器或驱动器阵列以便其他机器进行挂载所必需的内核支持，就像它是其计算机的本地计算机一样。在实际导出任何设备之前，需要进一步配置目标模式支持。如需更多信息，请参阅 targetd 和 targetcli 软件包中的文档。

- ISER - 这是 Linux 内核通用 iSCSI 层的一个低级驱动程序，可为 iSCSI 设备通过 InfiniBand 网络进行传输。

- RDS - 这是 Linux 内核中可靠的数据报服务。它没有在 Red Hat Enterprise Linux 7 内核中启用，因此无法加载。

13.5.2. 70-persistent-ipoib.rules 的使用

rdma 软件包提供文件 /etc/udev.d/rules.d/70-persistent-ipoib.rules。此 udev 规则文件用于将 IPoIB 设备从其默认名称（如 ib0 和 ib1）重命名为更描述性的名称。用户必须编辑此文件，以更改其设备的命名方式。首先，找到要重命名设备的 GUID 地址：

```
~]$ ip link show ib0
8: ib0: >BROADCAST,MULTICAST,UP,LOWER_UP< mtu 65520 qdisc pfifo_fast state UP mode
DEFAULT qlen 256
    link/infiniband 80:00:02:00:fe:80:00:00:00:00:00:00:00:f4:52:14:03:00:7b:cb:a1 brd
00:ff:ff:ff:ff:12:40:1b:ff:ff:00:00:00:00:00:00:00:ff:ff:ff:ff
```

link/infiniband 后马上是 IPoIB 接口的 20 字节硬件地址。地址的最后 8 字节（标在上方标出）是生成新名称所需的一切。用户可以做出适合他们的任何命名方案。例如，如果 mlx4 设备连接到 aib 0 子网结构，则使用 device_ fabric 命名约定，如 mlx4 _ib 0。不推荐的唯一内容是使用标准名称，如 ib0 orib 1，因为它们可能会与分配的内核自动名称冲突。接下来，在规则文件中添加一个条目。复制规则文件中的现有示例，将 ATTR{address} 条目中的 8 字节替换为要重命名的设备中突出显示的 8 字节，然后在 NAME 字段中输入要使用的新名称。

13.5.3. 为用户缓解 memlock 限制

RDMA 通信要求固定计算机的物理内存（也就是说，当整个计算机在可用内存上启动不足时，内核不允许将该内存交换到分页文件）。固定内存通常是非常特权的操作。为了允许 root 之外的用户运行大型 RDMA 应用程序，可能需要增加非root 用户在系统中被允许的内存量。这可以通过在 /etc/security/limits.d/ 目录中添加一个包含以下内容的文件来实现：

```
~]$ more /etc/security/limits.d/rdma.conf
# configuration for rdma tuning
*       soft    memlock         unlimited
*       hard    memlock         unlimited
# rdma tuning end
```

13.5.4. 为以太网操作配置 Mellanox 卡

Mellanox 的某些硬件能够在 InfiniBand 或以太网模式下运行。这些卡通常默认为 InfiniBand。用户可以将卡设置为以太网模式。目前，仅支持在 ConnectX 系列硬件（使用 mlx5 或 mlx4 驱动程序）中设置模式。

要配置 Mellanox mlx5 卡，请使用 mstflint 软件包中的 mstconfig 程序。详情请查看红帽客户门户网站中的 Red Hat Enterprise Linux 7 知识库中的配置 Mellanox mlx5 卡的内容。

要配置 Mellanox mlx4 卡，请使用 mstconfig 设置卡上的端口类型，如知识库文章中所述。如果 mstconfig 不支持您的卡，请编辑 /etc/rdma/mlx4.conf 文件，并按照该文件中的说明为 RoCE/IBoE 使用正确设置端口类型。在这种情况下，还需要重建 initramfs，以确保将更新的端口设置复制到 initramfs 中。

设置了端口类型后，如果一个或多个端口都被设置为 Ethernet，且 mstconfig 没有用于设置端口类型，则用户可能会在其日志中看到此消息：

```
mlx4_core 0000:05:00.0: Requested port type for port 1 is not supported on this HCA
```

这是正常的，不会影响操作。负责设置端口类型的脚本不知道驱动程序何时在内部切换到请求的类型，从脚本发出端口 2 以切换至该交换机的时间，尝试将端口 1 设置为不同类型都会被拒绝。该脚本会重试，直到命令成功或超时传递之后，指示端口交换机永不完成。

13.5.5. 连接到远程 Linux SRP 目标

SCSI RDMA 协议(SRP)是一个网络协议，它允许系统使用 RDMA 访问附加到另一个系统的 SCSI 设备。要允许 SRP 启动器在 SRP 目标端连接 SRP 目标，您必须为启动器中使用的主机通道适配器(HCA)端口添加访问控制列表(ACL)条目。

HCA 端口的 ACL ID 不唯一。ACL ID 取决于 HCA 的 GID 格式。使用同一驱动程序的 HCA（如 exampleib_qib ）可以具有不同的 GID 格式。ACL ID 还取决于您启动连接请求的方式。

连接到远程 Linux SRP 目标：高级别概述

1. *准备目标侧：*

   a. *创建存储后端。例如，获取 /dev/sdc1 分区：*

   ```
   /> /backstores/block create vol1 /dev/sdc1
   ```

   b. *创建 SRP 目标：*

   ```
   /> /srpt create 0xfe80000000000000001175000077dd7e
   ```

   c. *根据步骤中创建的后端创建一个 LUN：*

   ```
   /> /srpt/ib.fe80000000000000001175000077dd7e/luns create /backstores/block/vol1
   ```

   d. *为远程 SRP 客户端创建节点 ACL：*

   ```
   /> /srpt/ib.fe80000000000000001175000077dd7e/acls create
   0x7edd77000075110000 1175000077d708
   ```

   请注意，srp_daemon 和ibsrpdm 的节点 ACL 不同。

2. *为客户端发起与 srp_daemon 或 ibsrpdm 的 SRP 连接：*

   ```
   [root@initiator]# srp_daemon -e -n -i qib0 -p 1 -R 60 &
   ```

```
[root@initiator]# ibsrpdm -c -d /dev/infiniband/umad0 > /sys/class/infiniband_srp/srp-qib0-1/add_target
```

3. 可选。建议您使用不同的工具（如 lsscsi 或 dmesg ）验证 SRP 连接。

过程 13.3. 使用 srp_daemon 或 ibsrpdm 连接到远程 Linux SRP 目标

1. 对目标使用 ibstat 命令，以确定 状态 和端口 GUID 值。HCA 必须处于 Active 状态。ACL ID 基于 端口 GUID:

```
[root@target]# ibstat
CA 'qib0'
 CA type: InfiniPath_QLE7342
 Number of ports: 1
 Firmware version:
 Hardware version: 2
 Node GUID: 0x001175000077dd7e
 System image GUID: 0x001175000077dd7e
 Port 1:
  State: Active
  Physical state: LinkUp
  Rate: 40
  Base lid: 1
  LMC: 0
  SM lid: 1
  Capability mask: 0x0769086a
  Port GUID: 0x001175000077dd7e
  Link layer: InfiniBand
```

2. 获取 SRP 目标 ID，它基于 HCA 端口的 GUID。请注意，您需要将专用磁盘分区作为 SRP 目标的后端，如 /dev/sdc1。以下命令替换 fe80 的默认前缀，删除冒号，并在字符串的其余部分中添加新前缀：

```
[root@target]# ibstatus | grep '<default-gid>' | sed -e 's/<default-gid>://' -e 's/://g' | grep 001175000077dd7e
fe80000000000000001175000077dd7e
```

3. 使用 targetcli 工具在块设备中创建 LUN vol1，创建一个 SRP 目标并导出 LUN：

```
[root@target]# targetcli
```

```
/> /backstores/block create vol1 /dev/sdc1
Created block storage object vol1 using /dev/sdc1.
/> /srpt create 0xfe80000000000000001175000077dd7e
Created target ib.fe80000000000000001175000077dd7e.
/> /srpt/ib.fe80000000000000001175000077dd7e/luns create /backstores/block/vol1
Created LUN 0.
/> ls /
o- / ..... [...]
  o- backstores ..... [...]
  | o- block ..... [Storage Objects: 1]
  | | o- vol1 ..... [/dev/sdc1 (77.8GiB) write-thru activated]
  | o- fileio ..... [Storage Objects: 0]
  | o- pscsi ..... [Storage Objects: 0]
  | o- ramdisk ..... [Storage Objects: 0]
  o- iscsi ..... [Targets: 0]
  o- loopback ..... [Targets: 0]
  o- srpt ..... [Targets: 1]
    o- ib.fe80000000000000001175000077dd7e ..... [no-gen-acls]
      o- acls ..... [ACLs: 0]
      o- luns ..... [LUNs: 1]
        o- lun0 ..... [block/vol1 (/dev/sdc1)]
/>
```

4. ***在启动器中使用 ibstat 命令检查状态是否为 Active，并确定 端口 GUID:***

```
[root@initiator]# ibstat
CA 'qib0'
 CA type: InfiniPath_QLE7342
 Number of ports: 1
 Firmware version:
 Hardware version: 2
 Node GUID: 0x001175000077d708
 System image GUID: 0x001175000077d708
 Port 1:
  State: Active
  Physical state: LinkUp
  Rate: 40
  Base lid: 2
  LMC: 0
  SM lid: 1
  Capability mask: 0x07690868
  Port GUID: 0x001175000077d708
  Link layer: InfiniBand
```

5. ***使用以下命令扫描，但不连接到远程 SRP 目标：目标 GUID 显示启动器已找到远程目标。ID 字符串显示远程目标是一个 Linux 软件目标(ib_srpt.ko)。***

```
[root@initiator]# srp_daemon -a -o
IO Unit Info:
    port LID:        0001
    port GID:        fe80000000000000001175000077dd7e
```

```
change ID:       0001
max controllers: 0x10

controller[  1]
    GUID:      001175000077dd7e
    vendor ID: 000011
    device ID: 007322
    IO class : 0100
    ID:        Linux SRP target
    service entries: 1
        service[  0]: 001175000077dd7e / SRP.T10:001175000077dd7e
```

6. **要验证 SRP 连接，请使用 lsscsi 命令列出 SCSI 设备，并在启动器连接到目标之前和之后比较 lsscsi 输出。**

```
[root@initiator]# lsscsi
[0:0:10:0]   disk    IBM-ESXS ST9146803SS      B53C  /dev/sda
```

7. **要在没有为 initiator 端口配置有效 ACL 的情况下连接到远程目标（应该会失败），请对 srp_daemon or ibsrpdm 使用以下命令：**

```
[root@initiator]# srp_daemon -e -n -i qib0 -p 1 -R 60 &
[1] 4184
```

```
[root@initiator]# ibsrpdm -c -d /dev/infiniband/umad0 > /sys/class/infiniband_srp/srp-qib0-
1/add_target
```

8. **dmesg 的输出显示 SRP 连接操作失败的原因。在后续步骤中，使用目标端的 dmesg 命令使情况变得明确。**

```
[root@initiator]# dmesg -c
[ 1230.059652] scsi host5: ib_srp: REJ received
[ 1230.059659] scsi host5: ib_srp: SRP LOGIN from
fe80:0000:0000:0000:0011:7500:0077:d708 to fe80:0000:0000:0000:0011:7500:0077:dd7e
REJECTED, reason 0x00010006
[ 1230.073792] scsi host5: ib_srp: Connection 0/2 failed
[ 1230.078848] scsi host5: ib_srp: Sending CM DREQ failed
```

9. **由于 LOGIN 失败，lsscsi 命令的输出与上一步中的输出相同。**

```
[root@initiator]# lsscsi
[0:0:10:0]   disk    IBM-ESXS ST9146803SS      B53C  /dev/sda
```

10. **在目标端使用 dmesg (ib_srpt.ko)提供了有关 LOGIN 失败的原因。此外，输出中还包含**

srp_daemon 提供的有效 ACL ID：0x7edd77000075110000117500007 7d708。

```
[root@target]# dmesg
[ 1200.303001] ib_srpt Received SRP_LOGIN_REQ with i_port_id
0x7edd770000751100:0x1175000077d708, t_port_id
0x1175000077dd7e:0x1175000077dd7e and it_iu_len 260 on port 1
(guid=0xfe80000000000000:0x1175000077dd7e)
[ 1200.322207] ib_srpt Rejected login because no ACL has been configured yet for initiator
0x7edd770000751100001175000077d708.
```

11. ***使用 targetcli 工具添加有效的 ACL：***

```
[root@target]# targetcli
targetcli shell version 2.1.fb41
Copyright 2011-2013 by Datera, Inc and others.
For help on commands, type 'help'.

/> /srpt/ib.fe80000000000000001175000077dd7e/acls create
0x7edd770000751100001175000077d708
Created Node ACL for ib.7edd770000751100001175000077d708
Created mapped LUN 0.
```

12. *验证 **SRP LOGIN 操作：***

    a. ***等待 60 秒，以允许 srp_daemon 重新登录：***

    ```
    [root@initiator]# sleep 60
    ```

    b. *验证 **SRP LOGIN 操作：***

    ```
    [root@initiator]# lsscsi
    [0:0:10:0]   disk    IBM-ESXS ST9146803SS      B53C  /dev/sda
    [7:0:0:0] disk LIO-ORG vol1 4.0 /dev/sdb
    ```

    c. *对于 **SRP 目标发现的内核日志，请使用：***

    ```
    [root@initiator]# dmesg -c
    [ 1354.182072] scsi host7: SRP.T10:001175000077DD7E
    [ 1354.187258] scsi 7:0:0:0: Direct-Access     LIO-ORG  vol1            4.0  PQ: 0 ANSI: 5
    [ 1354.208688] scsi 7:0:0:0: alua: supports implicit and explicit TPGS
    [ 1354.215698] scsi 7:0:0:0: alua: port group 00 rel port 01
    [ 1354.221409] scsi 7:0:0:0: alua: port group 00 state A non-preferred supports TOlUSNA
    [ 1354.229147] scsi 7:0:0:0: alua: Attached
    ```

```
[ 1354.233402] sd 7:0:0:0: Attached scsi generic sg1 type 0
[ 1354.233694] sd 7:0:0:0: [sdb] 163258368 512-byte logical blocks: (83.5 GB/77.8 GiB)
[ 1354.235127] sd 7:0:0:0: [sdb] Write Protect is off
[ 1354.235128] sd 7:0:0:0: [sdb] Mode Sense: 43 00 00 08
[ 1354.235550] sd 7:0:0:0: [sdb] Write cache: disabled, read cache: enabled, doesn't
support DPO or FUA
[ 1354.255491] sd 7:0:0:0: [sdb] Attached SCSI disk
[ 1354.265233] scsi host7: ib_srp: new target: id_ext 001175000077dd7e ioc_guid
001175000077dd7e pkey ffff service_id 001175000077dd7e sgid
fe80:0000:0000:0000:0011:7500:0077:d708 dgid
fe80:0000:0000:0000:0011:7500:0077:dd7e
xyx
```

13.6. 配置子网管理器

13.6.1. 确定需要

大多数 InfiniBand 交换机都附带嵌入式子网管理器。但是，如果需要更新的子网管理器而不是交换机固件中的子网管理器，或者需要比交换机管理器允许的更多控制，Red Hat Enterprise Linux 7 包含 opensm 子网管理器。所有 InfiniBand 网络都必须运行子网管理器才能正常工作。即使执行由两个没有交换机的计算机组成的简单网络并且卡被重新插入，也需要子网管理器才能显示卡上的链接。如果多个子网管理器将充当控制器，并且其他任何子网管理器将充当控制器，那么控制器子网管理器应该会失败。

13.6.2. 配置 opensm 主配置文件

opensm 程序在 /etc/rdma/opensm.conf 中保留其主配置文件。用户可以随时编辑此文件，并且编辑操作会在升级过程中保留。文件本身中有大量选项文档。但是，对于最常用的两个编辑，将 GUID 设置为绑定到 并且使用 来运行 PRIORITY，强烈建议不要编辑 opensm.conf 文件，而是编辑 /etc/sysconfig/opensm。如果没有对基础 /etc/rdma/opensm.conf 文件的编辑，则每当 opensm 软件包升级时都会升级。随着新选项定期添加到此文件中，这有助于使当前配置保持最新。如果 opensm.conf 文件已被更改，然后在升级时，可能需要将新选项合并到编辑的文件中。

13.6.3. 配置 opensm 启动选项

/etc/sysconfig/opensm 文件中的选项控制子网管理器实际的启动方式，以及子网管理器的启动数量。例如，一个双端口 InfiniBand 卡（每个端口插入物理独立的网络中）将需要在每个端口上运行的子网管理器的副本。opensm 子网管理器将仅管理每个应用程序实例一个子网，并且必须为每个需要管理的子网启动一次。此外，如果有多个打开服务器，那么请在各服务器上设置优先级来控制要控制哪些端口以及控制器是控制器。

文件 /etc/sysconfig/opensm 用来提供设置子网管理器优先级以及控制子网管理器绑定到哪个 GUID 的简单方法。/etc/sysconfig/opensm 文件中的选项有广泛的说明。用户只需要读取并遵循文件本身的指示，即可启用 openm 的故障转移和多结构操作。

13.6.4. 创建 P_Key 定义

默认情况下，openm.conf 会查找文件 /etc/rdma/partitions.conf 以获取要在结构上创建的分区列表。所有光纤必须包含 0x7fff 子网，并且所有交换机和所有主机必须属于该结构。除此之外，还可以创建任何其他分区，并且所有主机和所有交换机都不必是这些额外分区的成员。这样，管理员可以创建与 InfiniBand 光纤上以太网 VLAN 类似的子网。如果分区定义了给定速度（如 40 Gbps），并且网络上有一个主机无法执行 40 Gbps，那么该主机即使有权限加入分区也将无法加入分区，因为它不能满足速度要求，因此建议将分区速度设置为任何有权加入分区的主机速度最慢。如果某些主机子集需要更快的分区，请创建一个速度较高的不同分区。

以下分区文件将导致默认的 0x7fff 分区速度降低 10 Gbps，以及速度为 40 Gbps 的 0x0002 分区：

```
~]$ more /etc/rdma/partitions.conf
# For reference:
# IPv4 IANA reserved multicast addresses:
#   http://www.iana.org/assignments/multicast-addresses/multicast-addresses.txt
# IPv6 IANA reserved multicast addresses:
#   http://www.iana.org/assignments/ipv6-multicast-addresses/ipv6-multicast-addresses.xml
#
# mtu =
#   1 = 256
#   2 = 512
#   3 = 1024
#   4 = 2048
#   5 = 4096
#
# rate =
#   2  =   2.5 GBit/s
#   3  =  10   GBit/s
#   4  =  30   GBit/s
#   5  =   5   GBit/s
#   6  =  20   GBit/s
#   7  =  40   GBit/s
#   8  =  60   GBit/s
#   9  =  80   GBit/s
#   10 = 120   GBit/s

Default=0x7fff, rate=3, mtu=4, scope=2, defmember=full:
      ALL, ALL_SWITCHES=full;
Default=0x7fff, ipoib, rate=3, mtu=4, scope=2:
      mgid=ff12:401b::ffff:ffff      # IPv4 Broadcast address
      mgid=ff12:401b::1              # IPv4 All Hosts group
      mgid=ff12:401b::2              # IPv4 All Routers group
      mgid=ff12:401b::16             # IPv4 IGMP group
      mgid=ff12:401b::fb             # IPv4 mDNS group
      mgid=ff12:401b::fc             # IPv4 Multicast Link Local Name Resolution group
      mgid=ff12:401b::101            # IPv4 NTP group
      mgid=ff12:401b::202            # IPv4 Sun RPC
      mgid=ff12:601b::1              # IPv6 All Hosts group
      mgid=ff12:601b::2              # IPv6 All Routers group
      mgid=ff12:601b::16             # IPv6 MLDv2-capable Routers group
      mgid=ff12:601b::fb             # IPv6 mDNS group
      mgid=ff12:601b::101            # IPv6 NTP group
      mgid=ff12:601b::202            # IPv6 Sun RPC group
      mgid=ff12:601b::1:3            # IPv6 Multicast Link Local Name Resolution group
```

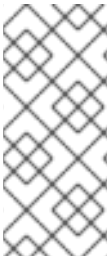
```
        ALL=full, ALL_SWITCHES=full;

ib0_2=0x0002, rate=7, mtu=4, scope=2, defmember=full:
        ALL, ALL_SWITCHES=full;
ib0_2=0x0002, ipoib, rate=7, mtu=4, scope=2:
        mgid=ff12:401b::ffff:ffff       # IPv4 Broadcast address
        mgid=ff12:401b::1               # IPv4 All Hosts group
        mgid=ff12:401b::2               # IPv4 All Routers group
        mgid=ff12:401b::16              # IPv4 IGMP group
        mgid=ff12:401b::fb              # IPv4 mDNS group
        mgid=ff12:401b::fc              # IPv4 Multicast Link Local Name Resolution group
        mgid=ff12:401b::101             # IPv4 NTP group
        mgid=ff12:401b::202             # IPv4 Sun RPC
        mgid=ff12:601b::1               # IPv6 All Hosts group
        mgid=ff12:601b::2               # IPv6 All Routers group
        mgid=ff12:601b::16              # IPv6 MLDv2-capable Routers group
        mgid=ff12:601b::fb              # IPv6 mDNS group
        mgid=ff12:601b::101             # IPv6 NTP group
        mgid=ff12:601b::202             # IPv6 Sun RPC group
        mgid=ff12:601b::1:3             # IPv6 Multicast Link Local Name Resolution group
        ALL=full, ALL_SWITCHES=full;
```

13.6.5. 启用 opensm

用户需要启用 opensm 服务，因为它在安装时不会被默认启用。以 root 用户身份运行以下命令：

```
~]# systemctl enable opensm
```

13.7. 测试早期 INFINIBAND RDMA 操作



注意

这部分只适用于 InfiniBand 设备。由于 iWARP 和 RoCE/IBoE 设备是基于 IP 的设备，因此用户应在配置了 IPoIB 并且设备具有 IP 地址后继续执行有关测试 RDMA 操作的部分。

启用rdma 服务并且启用了 opensm 服务（如果需要）并且安装了适用于特定硬件的适当用户空间库后，应该可以执行用户 spacerdma 操作。libibverbs-utils 软件包中的简单测试程序有助于确定 RDMA 操作是否正常工作。Theibv_devices 程序将显示系统中有哪些设备，sibv_devinfo 命令将提供关于每个设备的详细信息。例如：

```
~]$ ibv_devices
    device                 node GUID
    ------              ----------------
    mlx4_0              0002c903003178f0
    mlx4_1              f4521403007bcba0
~]$ ibv_devinfo -d mlx4_1
hca_id: mlx4_1
        transport:                  InfiniBand (0)
        fw_ver:                     2.30.8000
        node_guid:                  f452:1403:007b:cba0
        sys_image_guid:             f452:1403:007b:cba3
        vendor_id:                  0x02c9
        vendor_part_id:             4099
        hw_ver:                     0x0
        board_id:                   MT_1090120019
        phys_port_cnt:              2
                port:   1
                        state:              PORT_ACTIVE (4)
                        max_mtu:            4096 (5)
                        active_mtu:         2048 (4)
                        sm_lid:             2
                        port_lid:           2
                        port_lmc:           0x01
                        link_layer:         InfiniBand

                port:   2
                        state:              PORT_ACTIVE (4)
                        max_mtu:            4096 (5)
                        active_mtu:         4096 (5)
                        sm_lid:             0
                        port_lid:           0
                        port_lmc:           0x00
                        link_layer:         Ethernet
~]$ ibstat mlx4_1
CA 'mlx4_1'
        CA type: MT4099
        Number of ports: 2
        Firmware version: 2.30.8000
        Hardware version: 0
        Node GUID: 0xf4521403007bcba0
        System image GUID: 0xf4521403007bcba3
        Port 1:
                State: Active
                Physical state: LinkUp
                Rate: 56
                Base lid: 2
                LMC: 1
                SM lid: 2
                Capability mask: 0x0251486a
                Port GUID: 0xf4521403007bcba1
                Link layer: InfiniBand
        Port 2:
                State: Active
                Physical state: LinkUp
                Rate: 40
                Base lid: 0
```

```
LMC: 0
SM lid: 0
Capability mask: 0x04010000
Port GUID: 0xf65214fffe7bcba2
Link layer: Ethernet
```

The ibv_devinfo and ibstat 命令输出的信息略有不同（如端口 MTU 存在 in ibv_devinfo 但没有 in ibstat 输出，端口 GUID 存在 in ibstat 输出，但不存在 in ibv_devinfo 输出），另一些名称会有所不同（例如： 基本 本地标识符(LID)in ibstat 输出与 port_lid 输出 of ibv_devinfo相同。

简单的 ping 程序（如从 infiniband-diags 软件包进行对比）可用于测试 RDMA 连接。ibping 程序使用客户端-服务器模型。您必须首先在一台机器上启动同级服务器，然后作为客户端在另一台机器上运行并告诉它连接到 同级服务器。由于我们要测试基础 RDMA 功能，因此我们需要使用特定于 RDMA 的地址解析方法，而不是使用 IP 地址来指定服务器。

在服务器计算机上，用户可以使用 the ibv_devinfo 和 ibstat 命令打印出 port_lid （或基本 lid）和要测试的端口的端口 GUID（假设上述接口的端口 1，port _lid/Base LID 为 2，Port GUID 为 0xf4521403007bba1）。然后，以所需选项开始，以专门绑定到要测试的卡和端口，并且指定应该以服务器模式运行。您可以通过传递 -? 或 --help 来查看可用的选项，但在本例中，我们需要 -S 或 --Server 选项，并需要 -C 或 --Ca 和 - P 或 -- Port 绑定到特定卡和端口。注意：此实例中的端口不表示网络端口号，但在使用多端口卡时表示卡上的物理端口号。要使用多端口卡的第二个端口测试与 RDMA 结构的连接，需要告诉 ibping 绑定到卡 上 的端口 2。当使用单个端口卡或测试卡上的第一个端口时，不需要这个选项。例如：

```
~]$ ibping -S -C mlx4_1 -P 1
```

然后，更改到客户端计算机并运行ibping。记下 serveribping 程序绑定到的端口的端口 GUID，或者 serveribping 程序绑定到的端口的本地标识符 (LID)。此外，请注意客户端计算机上哪些卡和端口实际连接到与服务器上绑定至卡和端口相同的网络。例如，如果服务器上第一个网卡的第二个端口已绑定，并且该端口连接到辅助 RDMA 结构，则客户端上将指定需要哪些卡和端口才能连接到该次要结构。旦知道这些内容，即可运行作为客户端的同级程序，并使用服务器上收集的端口 LID 或 GUID 连接到服务器，作为要连接的地址。例如：

```
~]$ ibping -c 10000 -f -C mlx4_0 -P 1 -L 2
--- rdma-host.example.com.(none) (Lid 2) ibping statistics ---
10000 packets transmitted, 10000 received, 0% packet loss, time 816 ms
rtt min/avg/max = 0.032/0.081/0.446 ms
```

或者

```
~]$ ibping -c 10000 -f -C mlx4_0 -P 1 -G 0xf4521403007bcba1 \
```

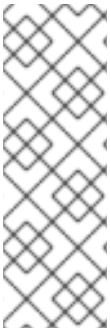
```
--- rdma-host.example.com.(none) (Lid 2) ibping statistics ---
10000 packets transmitted, 10000 received, 0% packet loss, time 769 ms
rtt min/avg/max = 0.027/0.076/0.278 ms
```

此结果验证结束 RDMA 通信是否适用于用户空间应用程序。

可能会出现以下错误：

```
~]$ ibv_devinfo
libibverbs: Warning: no userspace device-specific driver found for
/sys/class/infiniband_verbs/uverbs0
No IB devices found
```

此错误表示未安装必要的用户空间库。管理员将需要安装第 第 13.4 节 “InfiniBand 和 RDMA 相关软件包” 节中列出的一个用户空间库（适合其硬件）。在个别情况下，当用户为驱动程序安装错误的 arch 类型或 libibverbs 时会出现这种情况。例如，如果 libibverbs 是 arch x86_64，并且已安装 libmlx4，但类型为 i686，则可能会造成这个错误。



注意

许多示例应用更喜欢使用主机名或地址而不是 LID 来打开服务器和客户端之间的通信。对于这些应用程序，需要先设置 IPoIB，然后才能测试端到端 RDMA 通讯。Theibping 应用程序并不常见，因为它将接受简单的 LID 作为寻址形式，这允许它只是一个简单的测试，消除了测试场景中 IPoIB 寻址可能出现的问题，因此让我们更加孤立地查看简单的 RDMA 通讯是否正常工作。

13.8. 配置 IPOIB

13.8.1. 了解 IPoIB 的角色

如 第 1.1 节 “IP 与非 IP 网络比较” 所述，大多数网络都是 IP 网络。InfiniBand 不是.IPoIB 的角色是在 InfiniBand RDMA 网络之上提供 IP 网络模拟层。这允许现有应用程序在不修改的情况下通过 InfiniBand 网络运行。但是，这些应用程序的性能要比编写应用程序原生使用 RDMA 通讯时要低得多。由于大多数 InfiniBand 网络都有某些应用程序必须能从网络获取所有性能，而其他一些应用程序如果表示不需要修改应用程序来使用 RDMA 通讯，就可以接受降低性能率的其他应用程序。

由于 iWARP 和 RoCE/IBoE 网络实际上都是 IP 网络，其 IPDMA 层顶端的 IP 网络，所以不需要 IPoIB。因此，内核将拒绝在 iWARP 或 RoCE/IBoE RDMA 设备之上创建任何 IPoIB 设备。

13.8.2. 了解 IPoIB 通信模式

IPoIB 设备可以配置为在数据报或连接模式下运行。其区别在于，IPoIB 层尝试在通信的另一端与计算

机打开的队列对类型。对于数据报模式，打开不可靠、断开连接的队列对。对于连接的模式，打开可靠、连接的队列对。

使用数据报模式时，不可靠、断开连接的队列对类型不允许任何大于 InfiniBand link-layer 的 MTU 的数据包。IPoIB 层在传输的 IP 数据包之上添加一个 4 字节 IPoIB 标头。因此，IPoIB MTU 需要比 InfiniBand link-layer MTU 小 4 字节。因为 2048 是一个常见的 InfiniBand 链路层 MTU，数据报模式中常见的 IPoIB 设备 MTU 是 2044。

使用连接模式时，可靠、连接的队列对类型允许大于 InfiniBand link-layer MTU 的消息，主机适配器在每个端处理数据包分段和重新集合。因此，InfiniBand 适配器以连接模式发送的 IPoIB 信息大小没有限制。但是，IP 数据包只具有 16 位大小字段的限制，因此限制为 65535 的最大字节数。允许的最大 MTU 实际上比这小，因为我们还必须考虑还必须符合该大小的各种 TCP/IP 标头。因此，连接模式中的 IPoIB MTU 被上限为 65520，以确保有足够的空间用于所有需要的 TCP 标头。

连接的模式选项通常具有更高的性能，但也消耗更多的内核内存。由于大多数系统比内存消耗更关注性能，因此连接模式是最常用的模式。

但是，如果为连接模式配置系统，它仍然必须以数据报模式发送多播流量（InfiniBand 交换机和光纤无法以连接模式传递多播流量），并且在与未为连接模式配置的任何主机通信时，它也会回退到数据报模式。管理员应注意，如果他们打算运行发送多播数据的程序，并且这些程序尝试将多播数据发送到接口上的最大 MTU，那么有必要为数据报操作配置接口，或者找到配置多播应用以将数据包发送大小调整到数据存储大小数据包的大小。

13.8.3. 了解 IPoIB 硬件地址

IPoIB 设备具有 20 个字节硬件地址。如果config 无法读取所有 20 字节，则已弃用的实用程序绝对不能用于尝试查找 IPoIB 设备的正确硬件地址。iproute 软件包中的 ip 工具可以正常工作。

IPoIB 硬件地址的前 4 字节是标志和队列对号。下一个 8 字节是子网前缀。首次创建 IPoIB 设备时，它的默认子网前缀为 0xfe:80:00:00:00:00:00:00:00。该设备将使用默认子网前缀 (0xfe80000000000000)，直到它与子网管理器取得联系，此时它将重置子网前缀，使其与子网管理器已将其配置为匹配。最后 8 字节是 IPoIB 设备所附加的 InfiniBand 端口的 GUID 地址。因为前 4 字节和接下来 8 字节可能会随时变化，所以在为 IPoIB 接口指定硬件地址时，不会使用或匹配它们。第 13.5.2 节 “70-persistent-ipoib.rules 的使用” 部分解释了如何通过将 udev 规则文件中的 ATTR{address} 字段的前 12 字节离开 ATTR{address} 字段来生成地址，从而使设备匹配可靠。在配置 IPoIB 接口时，配置文件的 HWADDR 字段可以包含所有 20 字节，但实际只使用最后 8 字节来匹配并查找配置文件指定的硬件。但是，如果设备配置文件中 TYPE=InfiniBand 条目未正确拼写，如果up-ib 不是用于打开 IPoIB 接口的实际脚本，则系统将无法找到配置指定的硬件的错误将会发出。对于 IPoIB 接口，配置文件的 TYPE= 字段必须是 InfiniBand 或 infiniband （条目区分大小写，但脚本将接受这两个特定拼写）。

13.8.4. 了解 InfiniBand P_Key 子网

InfiniBand 结构可以通过使用不同的 P_Key 子网在逻辑上细分为虚拟子网。这与在以太网接口上使用 VLAN 非常相似。所有交换机和主机必须是默认 P_Key 子网的成员，但管理员可以创建额外的子网，并将这些子网的成员限制为 fabric 中的主机或交换机的子集。P_Key 子网必须由子网管理器定义，然后主机才能使用它。有关如何使用 opensm 子网管理器定义 P_Key 子网的信息，请参阅 第 13.6.4 节 “创建 P_Key 定义” 部分。对于 IPoIB 接口，一旦创建了 P_Key 子网，我们可以为这些 P_Key 子网创建额外的 IPoIB 配置文件。正如以太网设备上的 VLAN 接口一样，每个 IPoIB 接口的行为就像它位于与其他 IPoIB 接口完全不同的结构上一样，这些接口共享同一链路，但具有不同的 P_Key 值。

IPoIB P_Key 接口的名称有特殊要求。所有 IPoIB P_Keys 范围从 0x0000 到 0x7fff，高位 0x8000 表示 P_Key 成员资格是完全成员资格，而不是部分成员身份。Linux 内核的 IPoIB 驱动程序只支持 P_Key 子网中的完全成员资格，因此对于 Linux 可以连接的任何子网而言，会始终设置 P_Key 编号的高位。这意味着，如果 Linux 计算机加入 P_Key 0x0002，一旦加入，实际的 P_Key 编号将为 0x8002，表示我们是 P_Key 0x0002 的所有成员。因此，当在 opensm partitions.conf 文件中创建 P_Key 定义时，如 第 13.6.4 节 “创建 P_Key 定义” 部分所述，需要指定不使用 0x8000 的 P_Key 值，但在 Linux 客户端上定义 P_Key IPoIB 接口时，将 0x8000 值添加到基本 P_Key 值。

13.8.5. 使用文本用户界面 nmtui 配置 InfiniBand

文本用户界面工具 nmtui 可用于在终端窗口中配置 InfiniBand。使用以下命令启动该工具：

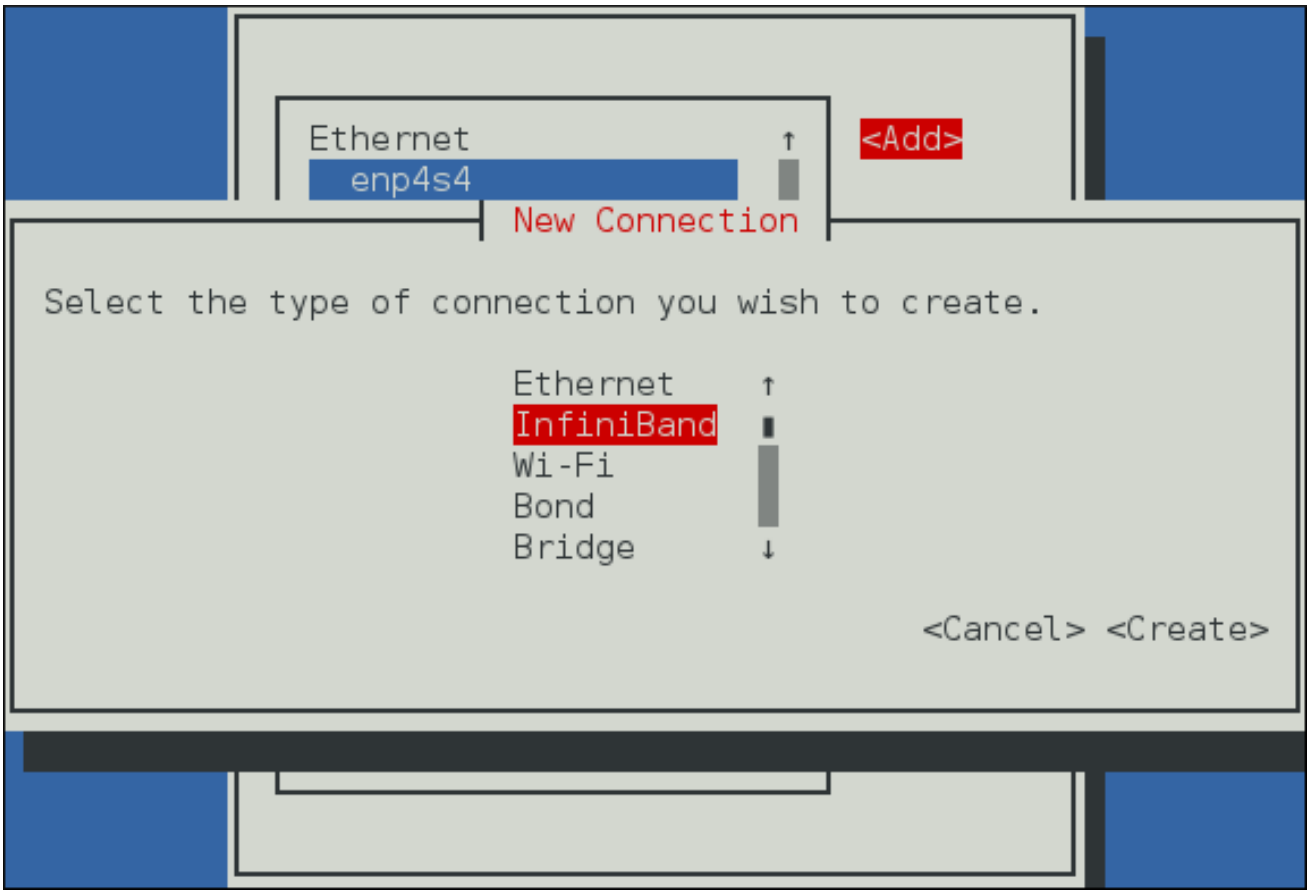
```
~]$ nmtui
```

此时将显示文本用户界面。任何无效的命令都会打印用法消息。

要导航，请使用箭头键或按 Tab 键前进，然后按ShiftTab后退步浏览选项。按 Enter 键选择一个选项。Space bar 切换复选框的状态。

在起始菜单中选择 Edit a connection。选择 Add，这会打开 New Connection 屏幕。

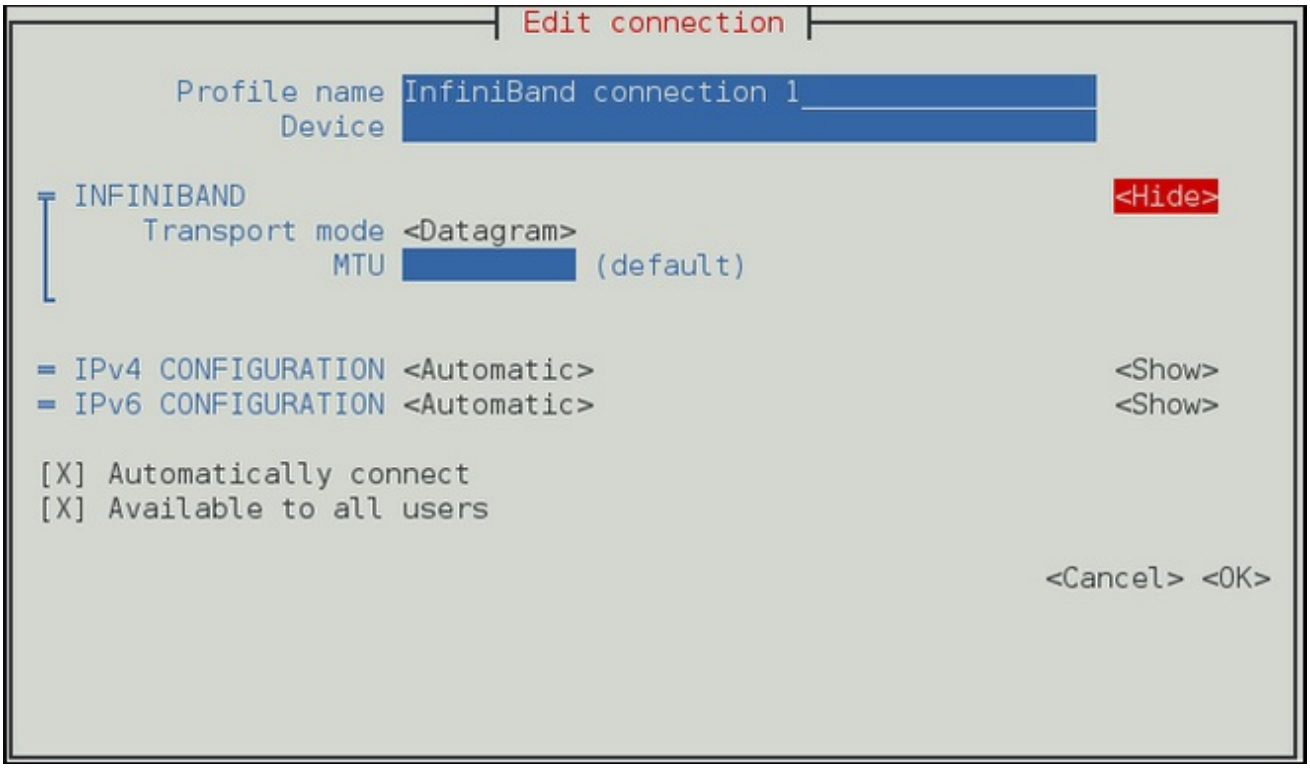
图 13.1. NetworkManager Text 用户界面添加 InfiniBand 连接菜单



[D]

选择 InfiniBand，会打开 Edit connection 屏幕。按照屏幕提示完成配置。

图 13.2. NetworkManager 文本用户界面配置 InfiniBand 连接菜单



[D]

有关 InfiniBand 术语的定义，请参阅 第 13.8.9.1 节 “配置 InfiniBand 选项卡”。

有关安装 nmtui 的详情请查看 第 3.2 节 “使用 nmtui 配置 IP 网络”。

13.8.6. 使用命令行工具 nmcli 配置 IPoIB

首先确定是否需要重命名默认 IPoIB 设备，如果需要，请按照 第 13.5.2 节 “70-persistent-ipoib.rules 的使用” 部分中的说明使用 udev 重命名规则重命名设备。通过删除 the ib_ipoib 内核模块，然后重新载入它，用户可以强制重命名 IPoIB 接口而不重新启动：

```
~]$ rmmod ib_ipoib
~]$ modprobe ib_ipoib
```

设备具有所需的名称后，使用 nmcli 工具创建 IPoIB 接口。以下示例显示了两种方式：

例 13.3. 在两个单独的命令中创建和修改 IPoIB。

```
~]$ nmcli con add type infiniband con-name mlx4_ib0 ifname mlx4_ib0 transport-mode connected
mtu 65520
Connection 'mlx4_ib0' (8029a0d7-8b05-49ff-a826-2a6d722025cc) successfully added.
~]$ nmcli con edit mlx4_ib0

===| nmcli interactive connection editor |===

Editing existing 'infiniband' connection: 'mlx4_ib0'

Type 'help' or '?' for available commands.
Type 'describe [>setting<.>prop<]' for detailed property description.

You may edit the following settings: connection, infiniband, ipv4, ipv6
nmcli> set infiniband.mac-address
80:00:02:00:fe:80:00:00:00:00:00:00:00:f4:52:14:03:00:7b:cb:a3
nmcli> save
Connection 'mlx4_ib3' (8029a0d7-8b05-49ff-a826-2a6d722025cc) successfully updated.
nmcli> quit
```

或者您可以在一个命令中运行 **nmcli c add** 和 **nmcli c modify**，如下所示：

例 13.4. 通过一个命令创建和修改 IPoIB.

```
nmcli con add type infiniband con-name mlx4_ib0 ifname mlx4_ib0 transport-mode connected mtu
65520  infiniband.mac-address 80:00:02:00:fe:80:00:00:00:00:00:00:00:f4:52:14:03:00:7b:cb:a3
```

在这些点上，已创建了名为 **mlx4_ib0** 的 **IPoIB** 接口，并设置为使用连接模式、最大连接模式 **MTU**、**DHCP**（**IPv 4** 和 **IPv6**）。如果将 **IPoIB** 接口用于集群流量和用于集群外通信的以太网接口，则可能需要在 **IPoIB** 接口上禁用默认路由和任何默认名称服务器。这可按如下方式完成：

```
~]$ nmcli con edit mlx4_ib0

===| nmcli interactive connection editor |===

Editing existing 'infiniband' connection: 'mlx4_ib0'

Type 'help' or '?' for available commands.
Type 'describe [>setting<.>prop<]' for detailed property description.
```

```
You may edit the following settings: connection, infiniband, ipv4, ipv6
nmcli> set ipv4.ignore-auto-dns yes
nmcli> set ipv4.ignore-auto-routes yes
nmcli> set ipv4.never-default true
nmcli> set ipv6.ignore-auto-dns yes
nmcli> set ipv6.ignore-auto-routes yes
nmcli> set ipv6.never-default true
nmcli> save
Connection 'mlx4_ib0' (8029a0d7-8b05-49ff-a826-2a6d722025cc) successfully updated.
nmcli> quit
```

如果需要 P_Key 接口，请使用 nmcli 创建一个，如下所示：

```
~]$ nmcli con add type infiniband con-name mlx4_ib0.8002 ifname mlx4_ib0.8002 parent mlx4_ib0 p-key 0x8002
Connection 'mlx4_ib0.8002' (4a9f5509-7bd9-4e89-87e9-77751a1c54b4) successfully added.
~]$ nmcli con modify mlx4_ib0.8002 infiniband.mtu 65520 infiniband.transport-mode connected ipv4.ignore-auto-dns yes ipv4.ignore-auto-routes yes ipv4.never-default true ipv6.ignore-auto-dns yes ipv6.ignore-auto-routes yes ipv6.never-default true
```

13.8.7. 使用命令行配置 IPoIB

首先确定是否需要重命名默认 IPoIB 设备，如果需要，请按照 第 13.5.2 节 “70-persistent-ipoib.rules 的使用” 部分中的说明使用 udev 重命名规则重命名设备。通过删除 the ib_ipoib 内核模块，然后重新载入它，用户可以强制重命名 IPoIB 接口而不重新启动：

```
~]$ rmmod ib_ipoib
~]$ modprobe ib_ipoib
```

设备具有所需的名称后，管理员可以使用首选编辑器创建 ifcfg 文件，以控制设备。带有静态 IPv4 寻址的典型 IPoIB 配置文件如下：

```
~]$ more ifcfg-mlx4_ib0
DEVICE=mlx4_ib0
TYPE=InfiniBand
ONBOOT=yes
HWADDR=80:00:00:4c:fe:80:00:00:00:00:00:00:00:f4:52:14:03:00:7b:cb:a1
BOOTPROTO=none
IPADDR=172.31.0.254
PREFIX=24
```

```
NETWORK=172.31.0.0
BROADCAST=172.31.0.255
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
MTU=65520
CONNECTED_MODE=yes
NAME=mlx4_ib0
```

DEVICE 字段必须与任何 udev 重命名规则中创建的自定义名称匹配。NAME 条目不需要与设备名称匹配。如果启动 GUI 连接编辑器，则 NAME 字段用于向用户显示此连接的名称。TYPE 字段必须是 InfiniBand，才能正确处理 InfiniBand 选项。CONNECTED_MODE 为 yes 或 no，其中 yes 将使用连接模式，并且 no 将使用数据报模式进行通信（请参阅 第 13.8.2 节 “了解 IPoIB 通信模式”部分）。

对于 P_Key 接口，这是典型的配置文件：

```
~]$ more ifcfg-mlx4_ib0.8002
DEVICE=mlx4_ib0.8002
PHYSDEV=mlx4_ib0
PKEY=yes
PKEY_ID=2
TYPE=InfiniBand
ONBOOT=yes
HWADDR=80:00:00:4c:fe:80:00:00:00:00:00:00:00:f4:52:14:03:00:7b:cb:a1
BOOTPROTO=none
IPADDR=172.31.2.254
PREFIX=24
NETWORK=172.31.2.0
BROADCAST=172.31.2.255
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
MTU=65520
CONNECTED_MODE=yes
NAME=mlx4_ib0.8002
```

对于所有 P_Key 接口文件，PHYSDEV 指令是必需的，且必须是父设备的名称。PKEY 指令必须设置为 yes，而 PKEY_ID 必须是接口的编号（添加有或不带 0x8000 个成员身份）。但是，设备名称必须是 PKEY_ID 的四位十六进制表示法和 0x8000 成员资格位（使用逻辑 OR 运算符）：

```
NAME=${PHYSDEV}.$((0x8000 | $PKEY_ID))
```

默认情况下，文件中的 PKEY_ID 被视为十进制数字，转换为十六进制数，然后使用逻辑 OR 运算符与 0x8000 运算符合并，以达到该设备的正确名称，但用户可以通过将标准的 0x 前缀添加到该数字，以十六进制指定 PKEY_ID。

13.8.8. 配置 IPoIB 后测试 RDMA 网络

配置 IPoIB 后，就可以使用 IP 地址指定 RDMA 设备。由于使用 IP 地址和主机名指定机器具有无处不在的性质，因此大多数 RDMA 应用程序将此用作首选，或在某些情况下仅使用它来指定要连接的远程计算机或本地设备。

要测试 IPoIB 层的功能，可以使用任何标准 IP 网络测试工具并提供要测试的 IPoIB 设备的 IP 地址。例如，IPoIB 设备的 IP 地址之间的 ping 命令现在应该可以正常工作。

Red Hat Enterprise Linux qperf 和 perftest 包括两个不同的 RDMA 性能测试软件包。它们中的任何一个都可用于进一步测试 RDMA 网络的性能。

但是，在使用作为 perftest 软件包一部分的任何应用程序，或使用 qperf 应用程序时，地址解析会有一个特殊备注。尽管远程主机是使用 IPoIB 设备的 IP 地址或主机名指定的，但测试应用程序可以通过不同的 RDMA 接口进行实际连接。这是因为从主机名或 IP 地址转换为 RDMA 地址的过程允许在两台机器之间使用有效的 RDMA 地址对。如果客户端可以通过多种方式连接到服务器，那么如果指定的路径出现问题，则程序可以选择使用其他路径。例如，如果每台机器上有两个端口连接到同一 InfiniBand 子网，并且每台机器上提供了第二个端口的 IP 地址，则该程序可能会发现每台机器上的第一个端口是有效的连接方法并使用它们。在这种情况下，任何 perftest 程序的命令行选项都可以用来告诉他们要绑定的卡和端口，如 第 13.7 节 “测试早期 InfiniBand RDMA 操作” 中的同级操作一样，以确保测试在需要测试的特定端口上进行。对于 qperf，绑定到端口的方法略有不同。qperf 程序作为一台计算机上的服务器运行，侦听所有设备（包括非 RDMA 设备）。客户端可以使用服务器的任何有效 IP 地址或主机名连接到 qperf。qperf 将首先尝试打开数据连接，并通过客户端命令行上给定的 IP 地址或主机名运行请求的测试，但是如果使用该地址出现问题，qperf 将回退到尝试在客户端和服务器之间的任何有效路径上运行测试。因此，要强制 qperf 通过特定链接进行测试，请将 -loc_id 和 -rem_id 选项用于 qperf 客户端，以强制测试在特定的链接上运行。

13.8.9. 使用 GUI 配置 IPoIB

要使用图形工具配置 InfiniBand 连接，请使用 nm-connection-editor

过程 13.4. 使用 nm-connection-editor 添加新的 InfiniBand 连接

1. 在终端中输入 nm-connection-editor ：

```
~]$ nm-connection-editor
```

2. 单击添加按钮。此时将显示 Choose a Connection Type 窗口。选择 InfiniBand 并点 Create。此时会出现 Editing InfiniBand 连接 1 窗口。

3. 在 InfiniBand 标签页中，从您要用于 InfiniBand 连接的下拉菜单中选择传输模式。

4. 输入 InfiniBand MAC 地址。

5. 检查并确认设置，然后单击保存按钮。

6. 使用 第 13.8.9.1 节 “配置 InfiniBand 选项卡” 编辑与 InfiniBand 相关的设置。

过程 13.5. 编辑现有的 InfiniBand 连接

按照以下步骤编辑现有的 InfiniBand 连接。

1. 在终端中输入 nm-connection-editor ：

```
~]$ nm-connection-editor
```

2. 选择您要编辑的连接并点击 Edit 按钮。

3. 选择常规选项卡。

4. 配置连接名称、自动连接行为和可用性设置。

Editing 对话框中的五个设置对所有连接类型通用，请参阅 General 选项卡：

- 连接名称 - 输入您的网络连接描述性名称。此名称将用于在 Network 窗口的菜单中列出此连接。

- 当这个网络可用时自动连接到这个网络 - 如果您希望 NetworkManager 在可用时自动连接到这个连接，请选择这个框。如需更多信息，请参阅 “使用 control-center 编辑现有连

接”一节。

- 所有用户可以连接到此网络 - 选择此框可创建系统上所有用户可用的连接。更改此设置可能需要 root 特权。详情请查看 第 3.4.5 节 “使用 GUI 管理系统范围以及专用连接配置集”。

- 使用这个连接时自动连接到 VPN - 如果您希望 NetworkManager 在有可用时自动连接到 VPN 连接，请选择这个框。从下拉菜单中选择 VPN。

- firewall Zone - 从下拉菜单中选择防火墙区域。有关防火墙区域的更多信息，请参阅Red Hat Enterprise Linux 7 安全指南。

5. 通过引用 第 13.8.9.1 节 “配置 InfiniBand 选项卡” 来编辑特定于 InfiniBand 的设置。

保存您的新（或修改）连接并创建进一步配置

编辑完 InfiniBand 连接后，点 Save 按钮保存自定义配置。

然后，配置：

- 连接的 IPv4 设置，单击 IPv4 Settings 选项卡，然后继续 第 5.4 节 “配置 IPv4 设置”

  或者

- 连接的 IPv6 设置，单击 IPv6 Settings 选项卡，再继续 第 5.5 节 “配置 IPv6 设置”。

13.8.9.1. 配置 InfiniBand 选项卡

如果您已经添加了一个新的 InfiniBand 连接（详情请参阅 过程 13.4, “使用 nm-connection-editor 添加新的 InfiniBand 连接”），您可以编辑 InfiniBand 标签来设置父接口和 InfiniBand ID。

传输模式

数据报或连接模式可从下拉列表中选择。选择您的 IPoIB 网络其余部分正在使用的相同模式。

设备 MAC 地址

如果已安装 InfiniBand 硬件，则会预先填充 InfiniBand 功能设备的 MAC 地址，用于 InfiniBand 网络流量。这个硬件地址字段会被预先填充。

MTU

（可选）为通过 InfiniBand 连接发送的数据包设置最大传输单元(MTU)大小。

13.8.10. 其它资源

安装的文档

- */usr/share/doc/initscripts-版本/sysconfig.txt - 描述配置文件及其指令.*

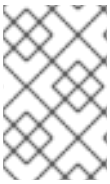
在线文档

https://www.kernel.org/doc/Documentation/infiniband/ipoib.txt

IPoIB 驱动程序的描述。包括对相关 RFC 的引用.

部分 IV. 服务器

本节讨论如何设置联网通常所需的服务器。



注意

要通过 Web 浏览器监控和管理服务器，请参阅使用 RHEL 7 web 控制台管理系统。

第 14 章 DHCP 服务器

动态主机配置协议(DHCP)是一种网络协议，可自动将 TCP/IP 信息分配给客户端计算机。每个 DHCP 客户端连接到中央位置的 DHCP 服务器，后者返回该客户端的网络配置（包括 IP 地址、网关和 DNS 服务器）。

14.1. 为什么使用 DHCP？

DHCP 对于自动配置客户端网络接口非常有用。在配置客户端系统时，您可以选择 DHCP，而不是指定 IP 地址、子网掩码、网关或 DNS 服务器。客户端从 DHCP 服务器检索此信息。如果您要更改大量系统的 IP 地址，DHCP 也非常有用。您无需重新配置所有系统，只需在服务器上为新 IP 地址集编辑一个配置文件。如果组织的 DNS 服务器有变化，则更改发生在 DHCP 服务器上，而不是在 DHCP 客户端上。当您重新启动网络或重新引导客户端时，更改将生效。

如果组织有一个正常工作的 DHCP 服务器连接到网络，笔记本电脑和其他移动计算机用户可以将这些设备从办公室移到办公室。

请注意，DNS 和 DHCP 服务器的管理员以及任何调配应用都应就组织中使用的主机名格式达成一致。有关主机名格式的详情，请查看 第 6.1.1 节 “推荐的命名实践”。

14.2. 配置 DHCP 服务器

dhcp 软件包包含一个互联网系统 Consortium(ISC)D 服务器。以 root 用户身份安装软件包：

```
~]# yum install dhcp
```

安装 dhcp 软件包会创建一个文件 /etc/dhcp/dhcpd.conf，该文件只是一个空配置文件。以 root 身份运行以下命令：

```
~]# cat /etc/dhcp/dhcpd.conf
#
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.example
#   see dhcpd.conf(5) man page
#
```

示例配置文件可以在 /usr/share/doc/dhcp-版本;/dhcpd.conf.example 中找到。您应该使用此文件来帮助您配置 /etc/dhcp/dhcpd.conf，详情如下。

DHCP 还使用文件 /var/lib/dhcpd/dhcpd.leases 来存储客户端租期数据库。如需更多信息，请参阅第 14.2.2 节 “租期数据库”。

14.2.1. 配置文件

配置 DHCP 服务器的第一步是创建 配置文件，该文件存储客户端的网络信息。使用此文件声明客户端系统的选项。

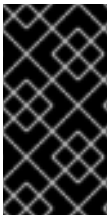
配置文件可以包含额外的标签页或空白行，以方便格式化。关键字不区分大小写，以哈希符号(#)开头的行被视为注释。

配置文件中有两种语句类型：

- 参数 - 说明如何执行任务，无论是执行任务，还是要将哪些网络配置选项发送到客户端。
- 声明 - 描述网络拓扑、描述客户端、为客户端提供地址，或者对一组声明应用一组参数。

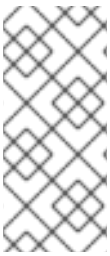
以 keyword 选项开头的参数称为选项。这些选项控制 DHCP 选项；而参数配置不是可选的值或控制 DHCP 服务器的行为方式。

在大括号({ })括起的部分之前声明的参数（包括选项）被视为全局参数。全局参数应用到它下面的所有部分。



重要

如果配置文件已更改，则更改不会生效，直到 DHCP 守护进程使用 systemctl restart dhcpd 命令重启为止。



注意

使用omshell命令不更改 DHCP 配置文件和每次重新启动服务，而是提供了一种交互式方式来连接、查询和更改 DHCP 服务器的配置。通过使用omshell，可在服务器运行期间进行所有更改。有关omshell的详情请参考 theomshell man page。

在 例 14.1 “子网声明” 中，路由器、subnet-mask、domain-search、domain-name-servers 和

time-offset 选项都会用于它下面声明的任何主机语句。

对于将提供服务的每个子网，对于 DHCP 服务器连接到的每个子网，必须有一个子网声明，该公告告诉 DHCP 守护进程如何识别某个地址在该子网上。每个子网都需要子网声明，即使没有地址动态分配到该子网。

在本例中，子网中的每个 DHCP 客户端都有全局选项，并且声明了一个范围。为客户端分配 范围内的 IP 地址。

例 14.1. 子网声明

```
subnet 192.168.1.0 netmask 255.255.255.0 {
        option routers                  192.168.1.254;
        option subnet-mask              255.255.255.0;
        option domain-search            "example.com";
        option domain-name-servers      192.168.1.1;
        option time-offset              -18000;     # Eastern Standard Time
 range 192.168.1.10 192.168.1.100;
}
```

要配置将动态 IP 地址租给子网中的系统的 DHCP 服务器，请从 例 14.2 “范围参数” 修改示例值。它为客户端声明默认的租用时间、最长租用时间和网络配置值。 本例将范围 192.168.1.10 和 192.168.1.100 范围 分配给客户端系统。

例 14.2. 范围参数

```
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
option domain-search "example.com";
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.10 192.168.1.100;
}
```

要根据网络接口卡的 MAC 地址为客户端分配 IP 地址，请在主机声明中使用硬件以太网 参数。如 例 14.3 “使用 DHCP 的静态 IP 地址” 中所示，主机 apex 声明指定 MAC 地址为 00:A0:78:8E:9E:AA 的网络接口卡始终接收 IP 地址 192.168.1.4。

请注意，您也可以使用可选参数 host-name 为客户端分配主机名。

例 14.3. 使用 DHCP 的静态 IP 地址

```
host apex {
  option host-name "apex.example.com";
  hardware ethernet 00:A0:78:8E:9E:AA;
  fixed-address 192.168.1.4;
}
```

Red Hat Enterprise Linux 7 支持为 InfiniBand IPoIB 接口分配静态 IP 地址。但是，由于这些接口没有普通的硬件以太网地址，因此必须使用不同的方法来为 IPoIB 接口指定唯一标识符。标准是使用选项 dhcp-client-identifier= 构造来指定 IPoIB 接口的 dhcp-client-identifier 字段。DHCP 服务器主机结构最多支持一个硬件以太网和每个主机段一个 dhcp-client-identifier 条目。但是，可能有多个固定地址条目，DHCP 服务器将自动使用适合 DHCP 请求所接收网络的地址来响应。

例 14.4. 在多个接口中使用 DHCP 的静态 IP 地址

如果机器有复杂的配置，例如每个物理接口上有两个 InfiniBand 接口和 P_Key 接口以及以太网连接，则可以使用以下静态 IP 结构来满足此配置：

```
Host apex.0 {
   option host-name “apex.example.com”;
   hardware ethernet 00:A0:78:8E:9E:AA;
   option dhcp-client-identifier=ff:00:00:00:00:00:02:00:00:02:c9:00:00:02:c9:03:00:31:7b:11;
   fixed-address 172.31.0.50,172.31.2.50,172.31.1.50,172.31.3.50;
}

host apex.1 {
   option host-name “apex.example.com”;
   hardware ethernet 00:A0:78:8E:9E:AB;
   option dhcp-client-identifier=ff:00:00:00:00:00:02:00:00:02:c9:00:00:02:c9:03:00:31:7b:12;
   fixed-address 172.31.0.50,172.31.2.50,172.31.1.50,172.31.3.50;
}
```

要找到适合您的设备的 dhcp-client-identifier，您通常可以使用 prefix ff:00:00:00:02:00:00:02:c9:00，然后添加 IPoIB 接口的最后 8 字节（这也会是 IPoIB 接口所在的 InfiniBand 端口的 8 字节 GUID）。在一些较旧的控制器中，这个前缀不正确。在这种情况下，我们建议在 DHCP 服务器上使用 tcpdump 来捕获传入的 IPoIB DHCP 请求，并从该捕获中收集正确的 dhcp-client-identifier。例如：

```
]$ tcpdump -vv -i mlx4_ib0
tcpdump: listening on mlx4_ib0, link-type LINUX_SLL (Linux cooked), capture size 65535
bytes
```

```
23:42:44.131447 IP (tos 0x10, ttl 128, id 0, offset 0, flags [none], proto UDP (17), length 328)
    0.0.0.0.bootpc > 255.255.255.255.bootps: [udp sum ok] BOOTP/DHCP, Request, length
300, htype 32, hlen 0, xid 0x975cb024, Flags [Broadcast] (0x8000)
          Vendor-rfc1048 Extensions
            Magic Cookie 0x63825363
            DHCP-Message Option 53, length 1: Discover
            Hostname Option 12, length 10: "rdma-qe-03"
            Parameter-Request Option 55, length 18:
              Subnet-Mask, BR, Time-Zone, Classless-Static-Route
              Domain-Name, Domain-Name-Server, Hostname, YD
              YS, NTP, MTU, Option 119
              Default-Gateway, Classless-Static-Route, Classless-Static-Route-Microsoft, Static-
Route
              Option 252, NTP
            Client-ID Option 61, length 20: hardware-type 255,
00:00:00:00:00:02:00:00:02:c9:00:00:02:c9:02:00:21:ac:c1
```

以上转储显示 Client-ID 字段。然后，硬件类型 255 与 ID 的 initial ff: 对应，而 ID 的其余部分将完全按照需要显示在 DHCP 配置文件中进行引用。

共享同一物理网络的所有子网都应在共享网络声明中声明，如 例 14.5 “shared-network lack” 所示。shared-network 中的参数以及括起的子网声明之外的参数被视为全局参数。分配给 shared-network 的名称必须是网络的描述性标题，例如使用标题 “test-lab” 来描述测试实验环境中的所有子网。

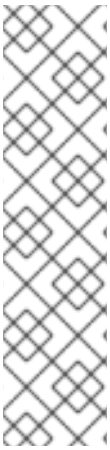
例 14.5. shared-network lack

```
shared-network name {
    option domain-search           "test.redhat.com";
    option domain-name-servers      ns1.redhat.com, ns2.redhat.com;
    option routers                  192.168.0.254;
    #more parameters for EXAMPLE shared-network
    subnet 192.168.1.0 netmask 255.255.252.0 {
        #parameters for subnet
        range 192.168.1.1 192.168.1.254;
    }
    subnet 192.168.2.0 netmask 255.255.252.0 {
        #parameters for subnet
        range 192.168.2.1 192.168.2.254;
    }
}
```

如 例 14.6 “grouprigger” 所示，组声明用于将全局参数应用到一组声明。例如，可以对共享网络、子网和主机进行分组。

例 14.6. grouprigger

```
group {
  option routers                192.168.1.254;
  option subnet-mask            255.255.255.0;
  option domain-search          "example.com";
  option domain-name-servers    192.168.1.1;
  option time-offset            -18000;     # Eastern Standard Time
  host apex {
    option host-name "apex.example.com";
    hardware ethernet 00:A0:78:8E:9E:AA;
    fixed-address 192.168.1.4;
  }
  host raleigh {
    option host-name "raleigh.example.com";
    hardware ethernet 00:A1:DD:74:C3:F2;
    fixed-address 192.168.1.6;
  }
}
```



注意

您可以将提供的示例配置文件用作起点，并在该文件中添加自定义配置选项。要将此文件复制到正确的位置，以 root 用户身份运行以下命令：

```
~]# cp /usr/share/doc/dhcp-version_number/dhcpd.conf.example /etc/dhcp/dhcpd.conf
```

... 其中 version_number 是 DHCP 版本号。

有关选项语句的完整列表及其作用，请参阅 dhcp-options(5) man page。

14.2.2. 租期数据库

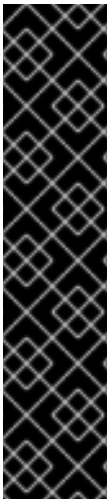
在 DHCP 服务器上，文件 /var/lib/dhcpd/dhcpd.leases 存储 DHCP 客户端租期数据库。不要更改此文件。每个最近分配的 IP 地址的 DHCP 租期信息自动存储在租期数据库中。这些信息包括租期的长度、IP 地址已分配给它的租期、租用开始和结束日期，以及用于检索租期的网络接口卡的 MAC 地址。

租期数据库中的所有时间都是统一的通用时间(UTC)，而不是本地时间。

租期数据库会不定期重新创建，因此不会太大。首先，所有已知的租期都保存在一个临时租期数据库中。dhcpd.leases 文件重命名为 dhcpd.leases~，临时租期数据库写入 dhcpd.leases。

DHCP 守护进程可能会被终止，或者在租期数据库重命名为备份文件但在写入新文件之前系统可能会崩溃。如果发生这种情况，则 dhcpd.leases 文件不存在，但需要启动服务。不要创建新的租用文件。如果您这样做，所有旧租期都会丢失，从而导致很多问题。正确的解决方案是将 dhcpd.leases~ 备份文件重命名为 dhcpd.leases，然后启动守护进程。

14.2.3. 启动和停止服务器



重要

当首次启动 DHCP 服务器时，它将失败，除非 dhcpd.leases 文件存在。如果文件不存在，您可以使用 touch /var/lib/dhcpd/dhcpd.leases 命令创建该文件。如果同一服务器也将 BIND 作为 DNS 服务器运行，则不需要这一步，因为启动命名服务会自动检查 dhcpd.leases 文件。

不要在之前运行的系统中创建新的租用文件。如果您这样做，所有旧租期都会丢失，从而导致很多问题。正确的解决方案是将 dhcpd.leases~ 备份文件重命名为 dhcpd.leases，然后启动守护进程。

要启动 DHCP 服务，请使用以下命令：

```
systemctl start dhcpd.service
```

要停止 DHCP 服务器，请输入：

```
systemctl stop dhcpd.service
```

默认情况下，DHCP 服务在引导时不启动。有关如何将守护进程配置为在引导时自动启动的详情，请参考 Red Hat Enterprise Linux 系统管理员指南。

如果有多个网络接口附加到系统，但 DHCP 服务器应仅在其中一个接口上侦听 DHCP 请求，请将 DHCP 服务器配置为仅侦听该设备。DHCP 守护进程将仅侦听在 /etc/dhcp/dhcpd.conf 文件中找到子网声明的接口。

这对于具有两个网卡的防火墙计算机非常有用。一个网卡可以配置为 DHCP 客户端，以检索到互联网的 IP 地址。其他网卡可用作防火墙后面的内部网络的 DHCP 服务器。仅指定连接到内部网络的网卡可以使系统更安全，因为用户无法通过互联网连接到守护进程。

要指定命令行选项，请以 root 用户身份复制并编辑 dhcpd.service 文件。例如，如下所示：

-

```
~]# cp /usr/lib/systemd/system/dhcpd.service /etc/systemd/system/
~]# vi /etc/systemd/system/dhcpd.service
```

以 root 用户身份编辑 [Service]:

```
ExecStart=/usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user dhcpd -group dhcpd --no-pid
your_interface_name(s)
```

部分下的行，重启该服务：

```
~]# systemctl --system daemon-reload
~]# systemctl restart dhcpd
```

在 [Service] 部分下的 /etc/systemd/system/dhcpd 中，可以将命令行选项附加到 ExecStart=/usr/sbin /dhcpd.dhcpd。它们包括：

- -p portnum - 指定 dhcpd 应侦听的 UDP 端口号。默认值为 port 的值。DHCP 服务器通过端口号传输到 DHCP 客户端的响应，其端口号大于指定的 UDP 端口。例如，如果使用默认端口 67，服务器侦听端口 67 中的请求并在端口 67 上响应客户端。如果此处指定了端口并使用 DHCP 中继代理，则必须指定 DHCP 中继代理应侦听的同一端口。详情请查看 第 14.3 节 “DHCP 转发代理”。
- -f - 将守护进程作为前台进程运行。这主要用于调试。
- -d - 将 DHCP 服务器守护进程记录到标准错误描述符。这主要用于调试。如果未指定，日志将写入 /var/log/messages。
- -cf filename - 指定配置文件的位置。默认位置为 /etc/dhcp/dhcpd.conf。
- -lf 文件名 - 指定租期数据库文件的位置。如果租期数据库文件已存在，每次启动 DHCP 服务器时使用相同的文件非常重要。强烈建议仅将此选项用于非生产计算机上的调试目的。默认位置为 /var/lib/dhcpd/dhcpd.leases。
- -Q - 在启动后台程序时，请勿打印完整的版权消息。

14.3. DHCP 转发代理

DHCP 转发代理(dhcrelay) 允许从没有 DHCP 服务器的子网将 DHCP 和 BOOTP 请求中继到其他子网上的一个或多个 DHCP 服务器。

当 DHCP 客户端请求信息时，DHCP 转发代理会将请求转发到 DHCP 转发代理启动时指定的 DHCP 服务器列表。当 DHCP 服务器返回回复时，回复将在发送原始请求的网络上广播或单播。

IPv4 的 DHCP转发代理,dhcrelay 侦听所有接口上的 DHCPv4 和 BOOTP 请求，除非使用 INTERFACES 指令在 /etc/sysconfig/dhcrelay 中指定了接口。请参阅 第 14.3.1 节 “将 dhcrelay 配置为 DHCPv4 和 BOOTP 转发代理”。IPv6 的 DHCP转发代理dhcrelay6 没有这种默认行为，并且必须指定侦听 DHCPv6 请求的接口。请参阅 第 14.3.2 节 “将 dhcrelay 配置为 DHCPv6 中继代理”。

dhcrelay 可以作为 DHCPv4 和 BOOTP 转发代理（默认）运行，也可以作为 DHCPv6 转发代理（使用 -6 参数）运行。要查看使用消息，请发出 commanddhcrelay -h。

14.3.1. 将 dhcrelay 配置为 DHCPv4 和 BOOTP 转发代理

要在 DHCPv4 中运行，BOOTP 模式指定应将请求转发到的服务器。以 root 用户身份复制并编辑 dhcrelay.service 文件：

```
~]# cp /lib/systemd/system/dhcrelay.service /etc/systemd/system/
~]# vi /etc/systemd/system/dhcrelay.service
```

编辑 [Service] 部分下的ExecStart 选项，并将一个或多个服务器 IPv4 地址添加到行末，例如：

```
ExecStart=/usr/sbin/dhcrelay -d --no-pid 192.168.1.1
```

如果您还想指定 DHCP 转发代理侦听 DHCP 请求的接口，请使用 -i 参数将它们添加到ExecStart 选项中（否则它将监听所有接口），例如：

```
ExecStart=/usr/sbin/dhcrelay -d --no-pid 192.168.1.1 -i em1
```

，查看 thedhcrelay(8)man page。

要激活所做的更改，以 root 用户身份重启该服务：

```
~]# systemctl --system daemon-reload
~]# systemctl restart dhcrelay
```

14.3.2. 将 dhcrelay 配置为 DHCPv6 中继代理

要在 DHCPv6 模式中运行dhcrelay，请添加 -6 “参数，并指定从客户端或其他中继代理接收查询的较低接口 ‘’ ”（应转发来自客户端和其他中继代理的查询）。Copydhcrelay.service todhcrelay6.service 并以 root 用户身份编辑它 ：

```
~]# cp /lib/systemd/system/dhcrelay.service /etc/systemd/system/dhcrelay6.service
~]# vi /etc/systemd/system/dhcrelay6.service
```

编辑 [Service] add -6 参数下的ExecStart “选项，并添加较低接口和 ‘’ 上级接口”，例如 ：

```
ExecStart=/usr/sbin/dhcrelay -d --no-pid -6 -l em1 -u em2
```

有关其他选项，请参见 thedhcrelay(8)man page。

要激活所做的更改，以 root 用户身份重启该服务 ：

```
~]# systemctl --system daemon-reload
~]# systemctl restart dhcrelay6
```

14.4. 配置多HOMED DHCP 服务器

多主 DHCP 服务器为多个网络提供服务，即多个子网。这些部分中的示例详细介绍了如何配置 DHCP 服务器为多个网络服务、选择要侦听的网络接口以及如何为移动网络的系统定义网络设置。

在进行任何更改之前，备份现有的 /etc/dhcp/dhcpd.conf 文件。

DHCP 守护进程将仅侦听在 /etc/dhcp/dhcpd.conf 文件中找到子网声明的接口。

以下是具有两个网络接口的服务器的基本 /etc/dhcp/dhcpd.conf 文件：enp1s0 在 10.0.0.0/24 网络中，以及 enp2s0 在 172.16.0.0/24 网络中。通过多个 subnet 声明，您可以为多个网络定义不同的设置：

```
default-lease-time 600;
max-lease-time 7200;
subnet 10.0.0.0 netmask 255.255.255.0 {
 option subnet-mask 255.255.255.0;
 option routers 10.0.0.1;
 range 10.0.0.5 10.0.0.15;
}
subnet 172.16.0.0 netmask 255.255.255.0 {
 option subnet-mask 255.255.255.0;
 option routers 172.16.0.1;
 range 172.16.0.5 172.16.0.15;
}
```

subnet 10.0.0.0 netmask 255.255.255.0;

您的 DHCP 服务器服务的每个网络都需要 subnet 声明。多个子网需要多个 subnet 声明。如果 DHCP 服务器在 subnet 声明范围中没有网络接口，DHCP 服务器不为该网络提供服务。

如果只有一个 subnet 声明，且该子网的范围中没有网络接口，DHCP 守护进程无法启动，且以下错误会记录到 /var/log/messages ：

```
dhcpd: No subnet declaration for enp1s0 (0.0.0.0).
dhcpd: ** Ignoring requests on enp1s0.  If this is not what
dhcpd:    you want, please write a subnet declaration
dhcpd:    in your dhcpd.conf file for the network segment
dhcpd:    to which interface enp2s0 is attached. **
dhcpd:
dhcpd:
dhcpd: Not configured to listen on any interfaces!
```

option subnet-mask 255.255.255.0;

option subnet-mask 选项定义子网掩码，并覆盖 subnet 声明中的 netmask 值。在简单情况下，子网和子网掩码值相同。

option routers 10.0.0.1;

option routers 选项定义子网的默认网关。对于系统而言，这是必需的，才能访问不同子网和外部网络上的内部网络。

range 10.0.0.5 10.0.0.15;

range 选项指定可用 IP 地址池。从指定的 IP 地址范围内为系统分配地址。

如需更多信息，请参阅 dhcpd.conf(5) man page。

警告

为避免在 DHCP 服务器向另一个物理以太网网段提供 IP 地址时出现错误配置，请确保您没有在共享网络声明中包含更多子网。

14.4.1. 主机配置

在进行任何更改之前，备份现有的 /etc/sysconfig/dhcpd 和 /etc/dhcp/dhcpd.conf 文件。

为多个网络配置单一系统

以下 /etc/dhcp/dhcpd.conf 示例创建两个子网，并根据它连接到的网络为同一系统配置 IP 地址 ：

```
default-lease-time 600;
max-lease-time 7200;
subnet 10.0.0.0 netmask 255.255.255.0 {
 option subnet-mask 255.255.255.0;
 option routers 10.0.0.1;
 range 10.0.0.5 10.0.0.15;
}
subnet 172.16.0.0 netmask 255.255.255.0 {
 option subnet-mask 255.255.255.0;
 option routers 172.16.0.1;
 range 172.16.0.5 172.16.0.15;
}
host example0 {
 hardware ethernet 00:1A:6B:6A:2E:0B;
 fixed-address 10.0.0.20;
}
host example1 {
 hardware ethernet 00:1A:6B:6A:2E:0B;
 fixed-address 172.16.0.20;
}
```

host example0

host 声明为单个系统定义特定参数，如 IP 地址。要为多个主机配置特定参数，请使用多个 host 声明。

大多数 DHCP 客户端会忽略 host 声明中的名称，因此该名称可以是任意名称，只要它对于其他 host 声明是唯一的。要为多个网络配置相同的系统，请为每个 host 声明使用不同的名称，否则 DHCP 守护进程无法启动。系统由 hardware ethernet 选项识别，而不是 host 声明中的名称。

hardware ethernet 00:1A:6B:6A:2E:0B;

hardware ethernet 选项标识系统。要查找此地址，请运行 ip link 命令。

fixed-address 10.0.0.20;

fixed-address 选项为 hardware ethernet 选项指定的系统分配有效的 IP 地址。这个地址必须在 range 选项指定的 IP 地址池之外。

如果 option 语句没有以分号结尾，DHCP 守护进程无法启动，且以下错误会记录到 /var/log/messages ：

```
/etc/dhcp/dhcpd.conf line 20: semicolon expected.
dhcpd: }
dhcpd: ^
dhcpd: /etc/dhcp/dhcpd.conf line 38: unexpected end of file
dhcpd:
dhcpd: ^
dhcpd: Configuration file errors encountered -- exiting
```

使用多个网络接口配置系统

以下 host 声明配置了一个系统，它具有多个网络接口，以便每个接口接收相同的 IP 地址。如果两个网络接口同时连接到同一网络，则此配置无法正常工作：

```
host interface0 {
 hardware ethernet 00:1a:6b:6a:2e:0b;
 fixed-address 10.0.0.18;
}
host interface1 {
```

```
 hardware ethernet 00:1A:6B:6A:27:3A;
 fixed-address 10.0.0.18;
}
```

在本例中，interface0 是第一个网络接口，interface1 是第二个接口。不同的 hardware ethernet 选项标识每个接口。

如果此类系统连接到另一个网络，请添加更多 host 声明，请记住：

- 为主机连接到的网络分配有效的 fixed-address。
- 使 host 声明中的名称唯一。

当 host 声明中指定的名称不唯一时，DHCP 守护进程无法启动，且以下错误会记录到 /var/log/messages ：

```
dhcpd: /etc/dhcp/dhcpd.conf line 31: host interface0: already exists
dhcpd: }
dhcpd: ^
dhcpd: Configuration file errors encountered -- exiting
```

这个错误是由 /etc/dhcp/dhcpd.conf 中定义的多个 host interface0 声明造成的。

14.5. 用于 IPV6 的 DHCP(DHCPV6)

自带有DHCPv 6 服务器、客户端和中继代理功能的 4.x 版本以来，ISC DHCP 包括对 IPv6(DHCPv6)的支持。代理同时支持 IPv4 和 IPv6，但代理一次只能管理一个协议；对于双重支持，必须为 IPv4 和 IPv 6 单独启动它们。例如，通过编辑对应的配置文件 /etc/dhcp/dhcpd.conf 和 /etc/dhcp/dhcpd6.conf 来配置 DHCPv 4 和 DHCPv6，然后发出以下命令：

```
~]# systemctl start dhcpd
~]# systemctl start dhcpd6
```

DHCPv6 服务器配置文件可以在 /etc/dhcp/dhcpd6.conf 中找到。

示例服务器配置文件可以在 /usr/share/doc/dhcp-版本/dhcpd6.conf.example 中找到。

简单的 DHCPv6 服务器配置文件类似如下：

```
subnet6 2001:db8:0:1::/64 {
        range6 2001:db8:0:1::129 2001:db8:0:1::254;
        option dhcp6.name-servers fec0:0:0:1::1;
        option dhcp6.domain-search "domain.example";
}
```

要为客户端分配 fixed-address，基于网络接口卡的 MAC 地址，请使用 hardware ethernet 参数：

```
host otherclient {
        hardware ethernet 01:00:80:a2:55:67;
        fixed-address6 3ffe:501:ffff:100::4321;
 }
```

shared-network 中的配置选项和 IPv6 的组声明与 IPV4 相同。如需了解更多详细信息，请参阅 例 14.5 “shared-network lack” 和 例 14.6 “grouprigger” 中演示的示例。

14.6. 为 IPV6 路由器配置 RADVD 守护进程

路由器公告守护进程(radvd)发送路由器公告消息，这是 IPv6 无状态自动配置所需要的。这允许用户根据这些公告自动配置其地址、设置、路由并选择默认路由器。配置 radvd 守护进程：

1. 安装 radvd 守护进程：

```
~]# sudo yum install radvd
```

2. 设置 /etc/radvd.conf 文件。例如：

```
interface enp1s0
{
AdvSendAdvert on;
MinRtrAdvInterval 30;
MaxRtrAdvInterval 100;
```

```
prefix 2001:db8:1:0::/64
{
AdvOnLink on;
AdvAutonomous on;
AdvRouterAddr off;
};

};
```

注意

如果要额外公告 DNS 解析器和路由器公告，请在 /etc/radvd.conf 文件中添加RDNSS <ip> <ip> { }; 选项。要为您的子网配置 DHCPv6 服务，您可以将 AdvManagedFlag 设置为 on，因此路由器公告允许客户端在 DHCPv6 服务可用时自动获取 IPv6 地址。有关配置 DHCPv6 服务的详情，请参考 第 14.5 节 “用于 IPv6 的 DHCP(DHCPv6)”

3. 启用 radvd 守护进程：

```
~]# sudo systemctl enable radvd.service
```

4. 立即启动 radvd 守护进程：

```
~]# sudo systemctl start radvd.service
```

要显示路由器公告软件包的内容以及 radvd 守护进程发送的值，请使用 radvdump 命令：

```
~]# radvdump
Router advertisement from fe80::280:c8ff:feb9:cef9 (hoplimit 255)
        AdvCurHopLimit: 64
        AdvManagedFlag: off
        AdvOtherConfigFlag: off
        AdvHomeAgentFlag: off
        AdvReachableTime: 0
        AdvRetransTimer: 0
```

```
Prefix 2002:0102:0304:f101::/64
        AdvValidLifetime: 30
        AdvPreferredLifetime: 20
        AdvOnLink: off
        AdvAutonomous: on
        AdvRouterAddr: on
Prefix 2001:0db8:100:f101::/64
        AdvValidLifetime: 2592000
        AdvPreferredLifetime: 604800
        AdvOnLink: on
        AdvAutonomous: on
        AdvRouterAddr: on
AdvSourceLLAddress: 00 80 12 34 56 78
```

有关 radvd 守护进程的更多信息，请参阅 radvd(8)、radvd.conf(5)、radvdump(8)man page。

14.7. DHCPV6 和 RADVD 的比较

IPv4 的动态主机配置主要应用于 DHCPv4。但是，对于 IPv6，可以使用以下选项：

- Manually
- 使用 radvd 守护进程
- 使用 DHCPv6 服务器

Manually

手动寻址始终可用。您可以使用 第 3.3.6 节 “使用 nmcli 连接到网络”、第 7.2 节 “使用文本用户界面 nmtui 配置绑定”、第 3.6 节 “使用 ip 命令配置 IP 网络” 中描述的工具为系统分配 IPv6 地址。

使用 radvd 守护进程

符合标准的 IPv6 网络必须提供路由器公告，因此可应用 IPv6 配置选项来运行路由器公告守护进程 (radvd)。路由器公告提供链接信息，这些信息实际上可在物理 LAN 上本地使用。在路由器播发上，您可以选择手动 IPv6 配置、通过路由器播发自动 IPv6 配置或动态主机配置协议(DHCPv6)。有关配置 radvd 守护进程的详情请参考 第 14.6 节 “为 IPv6 路由器配置 radvd 守护进程”。

使用 DHCPv6 服务器

当地址管理处于中央管理之下时，用户可以设置 DHCPv6 服务器。DHCPv6 的可用性通过路由器广告数据包中的标志来宣布。

表 14.1. DHCPv6 和 radvd 的比较

DHCPv6	radvd
确保随机地址以保护隐私.	提供有关默认网关的信息。
向客户端发送其他网络配置选项。例如，网络时间协议(NTP)服务器、会话启动协议(SIP)服务器、预启动执行环境(iPXE)配置。	
将 MAC 地址映射到 IPv6 地址.	

注意

若要正确配置网络，请将 DHCPv6 与 radvd 结合使用，因为只有路由器公告提供有关默认网关的信息。

14.8. 其它资源

- dhcpd(8)手册页 - 描述 DHCP 守护进程的工作原理。
- dhcpd.conf(5) 手册页 - 说明如何配置 DHCP 配置文件；包含一些示例。
- dhcpd.leases(5) man page - 描述租期的持久数据库。
- dhcp-options(5) 手册页 - 说明在 dhcpd.conf 中声明 DHCP 选项的语法；包含一些示例。
- dhcrelay(8)手册页 - 介绍 DHCP 转发代理及其配置选项。
- /usr/share/doc/dhcp-版本/ - 包含当前 DHCP 服务版本的示例文件、README 文件和发行注记。

第 15 章 DNS 服务器

DNS (Domain Name System)是一种分布式数据库系统，用于将主机名与对应的 IP 地址关联。对于用户，其优势在于他们可以通过名称来指代网络上的计算机，这些名称通常比数字网络地址更容易记忆。对于系统管理员而言，使用 DNS 服务器（也称为名称服务器）可以更改主机的 IP 地址，而不会影响基于名称的查询。DNS 数据库的使用不仅仅是用于将 IP 地址解析为域名，随着 DNSSEC 部署，它们的使用越来越广泛。

15.1. DNS 简介

通常使用对特定域具有权威的一个或多个集中式服务器来实施 DNS。当客户端主机请求名称服务器的信息时，它通常连接到端口 53。然后，名称服务器尝试解析请求的名称。如果名称服务器已配置为递归名称服务器并且没有权威答案，或者尚未从之前的查询中缓存答案，它将查询其他名称服务器（称为 root 名称服务器），以确定哪些名称服务器对于该名称具有权威，然后查询它们以获取请求的名称。配置为完全权威的名称服务器（禁用递归）不会代表客户端进行查找。

15.1.1. 名称服务器区域

在 DNS 服务器中，所有信息都存储在称为资源记录 (RR)的基本数据元素中。资源记录在 RFC 1034 中定义。域名组织为树结构。层次结构的每个级别都以句点(.)分隔。例如：由 . 表示的根域是 DNS 树的根域，其级别为零。称为 顶级域(TLD)的域名 com 是根域(.)的子级，因此它是层次结构的第一层。域名 example.com 处于层次结构的第二个级别。

> **例 15.1.** 简单资源记录
>
> 一个简单的资源记录示例 (RR)：
>
> ```
> example.com.      86400    IN        A          192.0.2.1
> ```
>
> 域名 example.com 是 RR 的所有者。值 86400 是 生存时间 (TTL)。字母 IN 表示 “Internet 系统”，表示 RR 的类。字母 A 表示 RR 的类型 （在本例中为主机地址）。主机地址 192.0.2.1 是此 RR 最后一个部分中的数据。这一行示例为 RR。组具有相同类型、所有者和类的 RR 称为资源记录集(RRSet)。

区域通过使用区域文件在权威名称服务器上定义，该文件包含各个区域中资源记录的定义。区域文件存储在主名称服务器 （也称为主名称服务器 ）中，对文件进行了更改，以及从主名称服务器接收区域定义的次要名称服务器 （也称为从属名称服务器 ）。主名称服务器和次要名称服务器对区域具有权威，并且对客户端而言看起来相同。根据配置，任何名称服务器也可以同时充当多个区域的主或次要服务器。

请注意，DNS 和 DHCP 服务器的管理员以及任何调配应用都应就组织中使用的主机名格式达成一

致。有关主机名格式的详情，请查看 第 6.1.1 节 “推荐的命名实践”。

15.1.2. 名称服务器类型

有两种名称服务器配置类型：

权威

权威名称服务器应答仅属于其区域一部分的资源记录。此类别包括主要（主）和次要（从属）名称服务器。

递归

递归名称服务器提供解析服务，但它们对任何区域都不是权威的。所有分辨率的应答在固定时间段内缓存在内存中，由检索的资源记录指定。

虽然名称服务器可以同时具有权威和递归性，但建议不要组合配置类型。要能够执行其工作，权威服务器应始终对所有客户端可用。另一方面，由于递归查找的时间比权威响应的时间要多得多，因此递归服务器只能供受限数量客户端使用，否则它们容易出现分布式拒绝服务(DDoS)攻击。

15.1.3. BIND 作为名称服务器

BIND 由一组与 DNS 相关的程序组成。它包含名为 的名称服务器、名为 rndc 的管理实用程序，以及一个名为 dig 的调试工具。有关如何在 Red Hat Enterprise Linux 中运行服务的更多信息，请参阅 Red Hat Enterprise Linux 系统管理员指南。

15.2. BIND

本节介绍 BIND （Berkeley Internet 名称域），Red Hat Enterprise Linux 中包含的 DNS 服务器。它侧重于其配置文件的结构，并描述了如何在本地和远程管理该文件。

15.2.1. 空区域

BIND “配置多个空区域”，以防止递归服务器向无法处理它们的 Internet 服务器发送不必要的查询（从而给查询它们的客户端创建延迟和 SERVFAIL 响应）。这些空区域可确保返回即时和权威 NXDOMAIN 响应。配置选项 empty-zones-enable 控制是否创建了空区域，同时也可使用 disable-empty-zone 选项，同时从要使用的默认前缀列表中禁用一个或多个空区域。

为 RFC 1918 前缀创建的空区域数量已增加，如果未指定空区域（默认为 yes），BIND 9.9 及更高版本的用户都会看到 RFC 1918 空区域。

15.2.2. 配置指定服务

当 named 服务启动时，它会读取来自文件的配置，如 表 15.1 “命名的服务配置文件” 所述。

表 15.1. 命名的服务配置文件

路径	描述
/etc/named.conf	主配置文件。
/etc/named/	主配置文件中包含的配置文件的辅助目录。

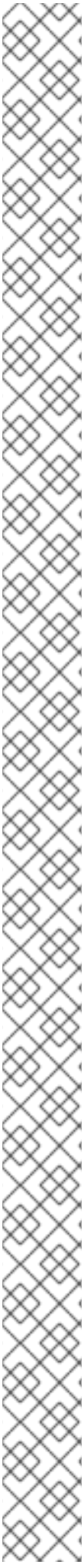
配置文件由一组语句组成，其嵌套选项通过打开和关闭大括号（{ 和 }）括起。请注意，编辑 文件时，您必须小心不要犯任何语法错误，否则 named 服务将不会启动。典型的 /etc/named.conf 文件按以下方式组织：

```
statement-1 ["statement-1-name"] [statement-1-class] {
  option-1;
  option-2;
  option-N;
};
statement-2 ["statement-2-name"] [statement-2-class] {
  option-1;
  option-2;
  option-N;
};
statement-N ["statement-N-name"] [statement-N-class] {
  option-1;
  option-2;
  option-N;
};
```

注意

如果您已安装 bind-chroot 软件包，BIND 服务将在 chroot 环境中运行。在这种情况下，初始化脚本将使用 mount --bind 命令挂载上述配置文件，以便您可以管理此环境外的配置。无需将任何内容复制到 /var/named/chroot/ 目录中，因为它是自动挂载的。这简化了维护过程，因为您不需要在 chroot 环境中运行时对 BIND 配置文件进行任何特殊处理。您可以像 BIND 一样整理所有内容，而不是在 chroot 环境中运行。

如果 /var/named/chroot/ 下对应的挂载点目录为空，则以下目录会自动挂载到



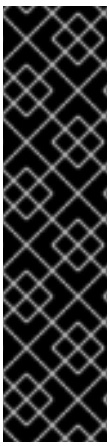
/var/named/chroot/ 目录中：

- */etc/named*
- */etc/pki/dnssec-keys*
- */run/named*
- */var/named*
- */usr/lib64/bind 或 /usr/lib/bind （依赖架构）。*

如果 /var/named/chroot/ 中不存在目标文件，则还会挂载以下文件：

- */etc/named.conf*
- */etc/rndc.conf*
- */etc/rndc.key*
- */etc/named.rfc1912.zones*
- */etc/named.dnssec.keys*
- */etc/named.iscdlv.key*

- /etc/named.root.key



重要

编辑已在 chroot 环境中挂载的文件需要创建备份副本，然后编辑原始文件。或者，使用禁用 “edit-a-copy 模式的编辑器”。例如，要在 chroot 环境中使用 Vim 编辑 BIND 的配置文件 /etc/named.conf，以 root 用户身份运行以下命令：

```
~]# vim -c "set backupcopy=yes" /etc/named.conf
```

15.2.2.1. 在 chroot 环境中安装 BIND

要安装 BIND 在 chroot 环境中运行，以 root 用户身份运行以下命令：

```
~]# yum install bind-chroot
```

要启用 named-chroot 服务，请运行以下命令来检查命名的服务是否在运行 ：

```
~]$ systemctl status named
```

如果正在运行，则必须禁用它。

要禁用命名，以 root 用户身份运行以下命令：

```
~]# systemctl stop named
```

```
~]# systemctl disable named
```

然后，要启用 named-chroot 服务，以 root 用户身份运行以下命令：

```
~]# systemctl enable named-chroot
```

```
~]# systemctl start named-chroot
```

要检查 named-chroot 服务的状态，以 root 身份运行以下命令：

```
~]# systemctl status named-chroot
```

15.2.2.2. 常见语句类型

在 /etc/named.conf 中通常使用以下语句类型：

acl

acl (Access Control List)语句允许您定义主机组，以便允许或拒绝访问名称服务器。它采用以下形式：

```
acl acl-name {
  match-element;
  ...
};
```

acl-name 语句名称是访问控制列表的名称，而 match-element 选项通常是一个单独的 IP 地址（如 10.0.1.1）或无 类别间路由 (CIDR)网络表示法（如 10.0.1.0/24）。有关已定义的关键字列表，请参阅 表 15.2 “预定义的访问控制列表”。

表 15.2. 预定义的访问控制列表

关键字	描述
任意	匹配每个 IP 地址.
localhost	匹配本地系统正在使用的任何 IP 地址。

关键字	描述
localnets	匹配本地系统所连接的任何网络上的任何 IP 地址。
none	不匹配任何 IP 地址。

acl 语句特别适用于与其他 语句（如 选项 ）结合使用。例 15.2 “使用 Conjunction 中的 acl 和 Options” 定义两个访问控制列表 ：黑语和 红色内容，并在授予 红色 内容普通访问时在黑名单中添加黑色 内容。

例 15.2. 使用 Conjunction 中的 acl 和 Options

```
acl black-hats {
  10.0.2.0/24;
  192.168.0.0/24;
  1234:5678::9abc/24;
};
acl red-hats {
  10.0.1.0/24;
};
options {
  blackhole { black-hats; };
  allow-query { red-hats; };
  allow-query-cache { red-hats; };
};
```

Include

include 语句允许您在 /etc/named.conf 中包含文件，以便将潜在的敏感数据放入具有受限权限的单独文件中。它采用以下形式：

```
include "file-name"
```

file-name 语句名称是文件的绝对路径。

例 15.3. 包含到 /etc/named.conf 的文件

```
include "/etc/named.rfc1912.zones";
```

选项

options 语句允许您定义全局服务器配置选项，以及设置其他语句的默认值。它可用于指定指定工作目录的位置、允许的查询类型，等等。它采用以下形式：

```
options {
  option;
  ...
};
```

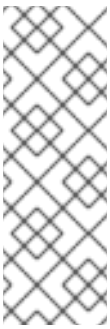
有关常用选项指令列表，请查看以下 表 15.3 “常用配置选项”。

表 15.3. 常用配置选项

选项	描述
allow-query	指定允许哪些主机查询权威资源记录的名称服务器。它接受 CIDR 表示法中的访问控制列表、IP 地址集合或网络。默认情况下允许所有主机。
allow-query-cache	指定允许哪些主机查询名称服务器以获取非权威数据，如递归查询。默认情况下，仅允许 localhost 和 localnets。
黑色	指定不允许哪些主机查询名称服务器。当特定主机或网络充满请求时，应使用此选项。默认选项为 none。
directory	指定 named 服务的工作目录。默认选项为 /var/named/。
disable-empty-zone	用于从要使用的默认前缀列表中禁用一个或多个空区域。可以在 options 语句中指定，也可以在 语句中指定。它可以多次使用。
dnssec-enable	指定是否返回与 DNSSEC 相关的资源记录。默认选项为 yes。
dnssec-validation	指定是否通过 DNSSEC 验证资源记录。默认选项为 yes。
empty-zones-enable	控制是否创建空区域。只能在 options 语句中指定。
forwarders	指定应向其转发请求以进行解析的名称服务器的有效 IP 地址列表。

选项	描述
forward	指定 forwarders 指令的行为。它接受以下选项： • 首先 - 服务器在尝试自行解析名称之前，将查询 forwarders 指令中列出的名称服务器。 • 仅 - 当无法查询 forwarders 指令中列出的名称服务器时，服务器将不会尝试自行解析该名称。
listen-on	指定要侦听查询的 IPv4 网络接口。在同样充当网关的 DNS 服务器上，您可以使用此选项来应答源自单个网络的查询。默认情况下，所有 IPv4 接口都使用。
listen-on-v6	指定要侦听查询的 IPv6 网络接口。在同样充当网关的 DNS 服务器上，您可以使用此选项来应答源自单个网络的查询。默认情况下，所有 IPv6 接口都使用。
max-cache-size	指定用于服务器缓存的最大内存量。达到限制时，服务器会导致记录永久过期，从而不会超过限制。在有多个视图的服务器中，限制将单独应用于每个视图的缓存。默认选项为 32M。

选项	描述
notify	指定是否在更新区域时通知次要名称服务器。它接受以下选项： • 是 - 服务器将通知所有次要名称服务器. • 否 - 服务器不会通知任何次要名称服务器。 • 仅主服务器 - 服务器将仅通知区域的主服务器。 • 显式 - 服务器将仅通知 zone 语句中指定的 second 服务器。
pid-file	指定由指定服务创建的进程 ID 文件的位置。
递归	指定是否充当递归服务器。默认选项为 **yes**。
statistics-file	指定统计信息文件的备用位置。默认使用 **/var/named/named.stats** 文件。



注意

named 用于运行时数据的目录已从 BIND 默认位置 **/var/run/named/** 移到新位置 **/run/named/**。因此，PID 文件已从默认位置 **/var/run/named/named.pid** 移到新位置 **/run/named/named.pid**。此外，**session-key** 文件已移到 **/run/named/session.key**。这些位置需要通过 **options** 部分中的语句来指定。请参阅 *例 15.4 “使用选项声明”*。

重要

为防止分布式拒绝服务(DDoS)攻击，建议您使用 allow-query-cache 选项来限制特定客户端子集的递归 DNS 服务。

有关可用选项的完整列表，请参阅 第 15.2.8.1 节 “安装的文档” 中引用的 BIND 9 管理员参考手册和 named.conf 手册页。

例 15.4. 使用选项声明

```
options {
  allow-query       { localhost; };
  listen-on port    53 { 127.0.0.1; };
  listen-on-v6 port 53 { ::1; };
  max-cache-size    256M;
  directory         "/var/named";
  statistics-file   "/var/named/data/named_stats.txt";

  recursion         yes;
  dnssec-enable     yes;
  dnssec-validation yes;

  pid-file          "/run/named/named.pid";
  session-keyfile   "/run/named/session.key";
};
```

zone

zone 语句允许您定义区域的特征，如其配置文件和区域特定选项的位置，并可用于覆盖全局 options 语句。它采用以下形式：

```
zone zone-name [zone-class] {
  option;
  ...
};
```

zone-name 属性是区的名称，zone-class 是区域的可选类，选项是一个 zone 语句选项，如 表 15.4 “区域语句中常用选项” 所述。

zone-name 属性尤为重要，因为它是为 /var/named/ 目录中对应区域文件中使用的 $ORIGIN 指令分配的默认值。指定守护进程将区域名称附加到区域文件中列出的任何非限定域名。例如，如果区域 语句定义了 example.com 的命名空间，请使用 example.com 作为 zone-name，以便将其放置在 example.com 区域文件中的主机名的末尾。

有关区文件的详情请参考 第 15.2.3 节 “编辑区域文件”。

表 15.4. 区域语句中常用选项

选项	描述
allow-query	指定允许哪些客户端请求有关该区域的信息。此选项覆盖全局 allow-query 选项。默认情况下允许所有查询请求。
allow-transfer	指定允许哪些次要服务器请求传输区域信息。默认情况下允许所有传输请求。
allow-update	指定允许哪些主机动态更新其区域中的信息。默认选项是拒绝所有动态更新请求。 请注意，在允许主机更新其区域信息时，您应该小心。不要在此选项中设置 IP 地址，除非服务器位于可信网络中。反之，使用 第 15.2.6.3 节 “事务 SIGnatures(TSIG)” 所述 TSIG 密钥。
file	指定包含区域配置数据的指定工作目录中的文件名称。
Master	指定从哪个 IP 地址请求权威区域信息。只有在区域定义为 slave 类型时才使用这个选项。

选项	描述
notify	指定是否在更新区域时通知次要名称服务器。它接受以下选项： • 是 - 服务器将通知所有次要名称服务器. • 否 - 服务器不会通知任何次要名称服务器。 • 仅主服务器 - 服务器将仅通知区域的主服务器。 • 显式 - 服务器将仅通知 zone 语句中指定的 second 服务器。

选项	描述
type	指定 zone 类型。它接受以下选项： • 仅委派 - 强制实施基础架构区域的委派状态，如 COM、NET 或 ORG。收到的任何没有显式或隐式委派的答案都将被视为 NXDOMAIN。这个选项仅适用于递归或缓存实施中使用的 TLD（顶级域）或 root 区域文件。 • forward - 将有关该区域信息的所有请求转发到其他名称服务器。 • 提示 - 特殊类型的区域，用于指向根名称服务器，当否则不知道区域时解析查询。不需要使用 hint 区域进行默认配置。 • Master - 将名称服务器指定为该区域的权威。如果区域的配置文件驻留在系统上，则应将区域设置为主区域。 • slave - 将名称服务器设计为这个区域的次要服务器。主服务器在 masters 指令中指定。

对于主名称服务器或从属名称服务器的 /etc/named.conf 文件的大部分更改都涉及添加、修改或删除区域语句，并且通常需要一小部分区域声明选项才能使名称服务器高效工作。

在 例 15.5 “主名称服务器的区域语句” 中，区被标识为 example.com，类型被设置为 master，并且指示 named 服务读取 /var/named/example.com.zone 文件。它还仅允许次要名称服务器(192.168.0.2)传输区域。

例 15.5. 主名称服务器的区域语句

```
zone "example.com" IN {
  type master;
  file "example.com.zone";
  allow-transfer { 192.168.0.2; };
};
```

次要服务器的 zone 语句略有不同。type 设置为 slave，master 指令会告知主服务器的 IP 地址。

在 例 15.6 “次要名称服务器的区域语句” 中，named 服务配置为查询 192.168.0.1 IP 地址 的主服务器，以获取有关 example.com 区域的信息。然后，收到的信息会保存到 /var/named/slaves/example.com.zone 文件中。请注意，您必须将所有次要区域放在 /var/named/slaves/ 目录中，否则该服务将无法传输区域。

例 15.6. 次要名称服务器的区域语句

```
zone "example.com" {
  type slave;
  file "slaves/example.com.zone";
  masters { 192.168.0.1; };
};
```

15.2.2.3. 其他语句类型

在 /etc/named.conf 中使用以下类型的语句：

controls

通过 control 语句，您可以配置使用 therndc 命令来管理指定服务所需的各种安全要求。

有关 Therndc 工具及其用法的详情，请查看 第 15.2.4 节 “使用 rndc 实用程序”。

key

key 语句允许您按名称定义特定密钥。密钥用于验证各种操作，如安全更新或使用 rndc 命令。将两个选项与键一起使用 ：

- algorithm-name - 要使用的算法类型（如 hmac-md5）。
- Secret "key-value" - 加密的密钥.

有关 Therndc 工具及其用法的详情，请查看 第 15.2.4 节 “使用 rndc 实用程序”。

logging

logging 语句允许您使用多种类型的日志，因此称为频道。通过使用 语句中的 channel 选项，您可以使用自己的文件名（文件）、大小限制（大小限制）、版本号（版本）和重要级别（严重性）来构造自定义的日志类型。定义了自定义频道后，可以使用 category 选项对频道进行分类，并在重启指定服务时开始日志记录。

默认情况下，named 将标准消息发送到 rsyslog 守护进程，守护进程将它们放置在 /var/log/messages 中。BIND 中内置了多个具有不同严重性级别的标准通道，如 default_syslog （处理信息日志消息）和 default_debug （专门处理调试消息）。默认类别名为 default，使用内置通道进行正常日志记录，而无需特殊配置。

自定义日志记录过程可能是一个非常详细的过程，并且不在本章的讨论范围之内。有关创建自定义 BIND 日志的详情，请参考 第 15.2.8.1 节 “安装的文档” 中引用的 BIND 9 管理员参考手册。

server

server 语句允许您指定影响指定服务应如何响应远程名称服务器的选项，特别是通知和区域传送方面。

transfer-format 选项控制随每条消息一起发送的资源记录数量。它可以是 一次性回答 （仅一个资源记录）或 多个回答（ 多个资源记录）。请注意，尽管 many-answers 选项效率更高，但旧版本的 BIND 不支持此选项。

trusted-keys

trusted-keys 语句允许您指定用于安全 DNS (DNSSEC)的分类公钥。有关此主题的更多信息，请参阅 第 15.2.6.4 节 “DNS 安全扩展(DNSSEC)”。

view

view 语句允许您根据主机查询名称服务器的网络来创建特殊视图。这允许某些主机收到一个关于区域的答案，而其他主机会收到完全不同的信息。或者，某些区域可能仅可供特定的可信主机使用，不可信主机则只能对其他区域进行查询。

只要它们的名称是唯一的，就可以使用多个视图。match-clients 选项允许您指定应用到特定视图的 IP 地址。如果在 视图中使用 options 语句，它将覆盖已配置的全局选项。最后，大多数视图语句包含应用到 match-clients 列表的多个区域语句。

请注意，列出 view 语句的顺序非常重要，因为将使用与特定客户端 IP 地址匹配的第一个语句。有关此主题的详情请参考 第 15.2.6.1 节 “多个视图”。

15.2.2.4. 注释标签

除了语句外，/etc/named.conf 文件还可以包含注释。注释将被指定服务忽略，但可在向用户提供附加信息时证明非常有用。以下是有效的注释标签：

//

行末尾 // 字符后面的任何文本都将被视为注释。例如：

```
notify yes;  // notify all secondary nameservers
```

#

行末尾 # 字符后面的任何文本都将被视为注释。例如：

```
notify yes;  # notify all secondary nameservers
```

/ 和 */*

包括在 / 和 */ 中的任何文本块均被视为注释。例如：*

```
notify yes;  /* notify all secondary nameservers */
```

15.2.3. 编辑区域文件

如 第 15.1.1 节 “名称服务器区域” 所述，区文件包含有关命名空间的信息。默认情况下，它们存储在位于 /var/named/ 中的指定工作目录中。每个区域文件都根据 zone 语句中的 file 选项进行命名，通常方式与 中的域相关，并将该文件识别为包含区域数据，如 example.com.zone。

表 15.5. 指定的服务区文件

路径	描述
/var/named/	named 服务的工作目录。名称服务器不允许写入此目录。
/var/named/slaves/	次要区域的目录。此目录可由 named 服务写入。
/var/named/dynamic/	其他文件的目录，如动态 DNS (DDNS)区域或托管的 DNSSEC 密钥。此目录可由 named 服务写入。
/var/named/data/	用于各种统计信息和调试文件的目录。此目录可由 named 服务写入。

区域文件由指令和资源记录组成。指令告知名称服务器执行任务或应用特殊设置到区域，资源记录定义区域的参数并将身份分配给各个主机。指令是可选的，但需要资源记录才能为区域提供名称服务。

所有指令和资源记录应在单独的行中输入。

15.2.3.1. 常用指令

指令以美元符号字符($)开头，后跟 指令的名称，通常显示在文件的顶部。以下指令通常用于区文件：

$INCLUDE

$INCLUDE 指令允许您在显示该文件的位置包含另一个文件，以便其他区域设置可以存储在单独的区域文件中。

例 15.7. 使用 $INCLUDE 指令

```
$INCLUDE /var/named/penguin.example.com
```

$ORIGIN

$ORIGIN 指令允许您将域名附加到不限定的记录，例如仅具有主机名的记录。请注意，如果在 /etc/named.conf 中指定区域，则不需要使用此指令，因为默认使用区域名称。

在 例 15.8 “使用 $ORIGIN 指令” 中，资源记录中使用的所有不以结尾名称（ 字符 ）都附加到 example.com。

例 15.8. 使用 $ORIGIN 指令

```
$ORIGIN example.com.
```

$TTL

$TTL 指令允许您将区域的默认 Time(TTL)值设置为 Live (TTL)值，即区域记录的有效期。每一资源记录可以包含自己的 TTL 值，以覆盖此指令。

增加此值可让远程名称服务器在较长时间内缓存区域信息，从而减少区域的查询次数并延长传播资源记录更改所需的时间。

例 15.9. 使用 $TTL 指令

```
$TTL 1D
```

15.2.3.2. 通用资源记录

以下资源记录通常在区文件中使用：

A

Address 记录指定要分配给某个名称的 IP 地址。它采用以下形式：

```
hostname IN A IP-address
```

如果省略主机名值，记录将指向最后指定的主机名。

在 例 15.10 “使用 A 资源记录” 中，对 server1.example.com 的请求指向 10.0.1.3 或 10.0.1.5。

例 15.10. 使用 A 资源记录

```
server1  IN  A  10.0.1.3
         IN  A  10.0.1.5
```

CNAME

Canonical Name 记录将一个名称映射到另一个名称。因此，这种记录有时被称为别名记录。它采用以下形式：

```
alias-name IN CNAME real-name
```

CNAME 记录最常用于指向使用常见命名方案的服务，例如 www （Web 服务器）。但是，它们的使用有多项限制：

- CNAME 记录不应指向其他 CNAME 记录。这主要是为了避免可能的无限循环。
- CNAME 记录不应包含其他资源记录类型（如 A、NS 和 MX 等）。唯一的例外是签署区域时 DNSSEC 相关记录（RRSIG、NSEC 等）。
- 指向主机（NS、MX 和 PTR）完全限定域名(FQDN)的其他资源记录不应指向 CNAME 记录。

在 例 15.11 “使用 CNAME 资源记录” 中，A 记录将主机名绑定到 IP 地址，而 CNAME 记录将常用的 www 主机名指向它。

例 15.11. 使用 CNAME 资源记录

```
server1  IN  A      10.0.1.5
www      IN  CNAME  server1
```

MX

Mail Exchange 记录指定发送到该区域控制的特定命名空间的邮件应前往哪里。它采用以下形式：

```
IN MX preference-value email-server-name
```

email-server-name 是一个完全限定域名(FQDN)。**preference-value** 允许对命名空间的电子邮件服务器进行数字排名，优先选择某些电子邮件系统而不是其他电子邮件系统。优先级最低的 **MX** 资源记录优先于其他值。但是，多个电子邮件服务器可以拥有相同的值，在其中均匀分发电子邮件流量。

在 *例 15.12 “使用 MX 资源记录”* 中，在收到目标为 **example.com** 域的电子邮件时，第一个 **mail .example.com** 电子邮件服务器优先于 **mail2 .example.com** 电子邮件服务器。

例 15.12. 使用 MX 资源记录

```
example.com.  IN  MX  10  mail.example.com.
              IN  MX  20  mail2.example.com.
```

NS

Nameserver 记录宣布特定区域的权威名称服务器。它采用以下形式：

```
IN NS nameserver-name
```

nameserver-name 应为完全限定域名(FQDN)。请注意，如果两个名称服务器被列为域的权威，那么这些名称服务器是次要名称服务器还是其中一个名称服务器是否为主服务器并不重要。它们仍然被视为具有权威性。

例 15.13. 使用 NS 资源记录

```
IN  NS  dns1.example.com.
IN  NS  dns2.example.com.
```

PTR

Pointer 记录指向命名空间的另一部分。它采用以下形式：

```
last-IP-digit IN PTR FQDN-of-system
```

最后 IP 位指令是 IP 地址中的最后一个数字，而 FQDN-of-system 是完全限定域名(FQDN)。

PTR 记录主要用于反向名称解析，因为它们将 IP 地址指向特定名称。有关正在使用的 PTR 记录示例，请查看 第 15.2.3.4.2 节 “反向名称解析区域文件”。

SOA

授权起始记录向名称服务器宣布有关命名空间的重要权威信息。它位于指令的后面，是区域文件中的第一个资源记录。它采用以下形式：

```
@  IN  SOA  primary-name-server hostmaster-email (
    serial-number
    time-to-refresh
    time-to-retry
    time-to-expire
    minimum-TTL )
```

这些指令如下：

- @ 符号将 $ORIGIN 指令（如果未设置 $ORIGIN 指令，则放置区域名称），作为此 SOA 资源记录所定义的命名空间。
- primary-name-server 指令是对该域具有权威的主名称服务器的主机名。
- hostmaster-email 指令是有关命名空间的联系人的电子邮件。
- serial-number 指令是一个数字值，在每次更改区域文件时递增，以指示 named 服务重新加载区域的时间。
- time-to-refresh 指令是数字值次要名称服务器，用于确定在询问主名称服务器是否对区域进行了任何更改之前需要等待多久。
- 时间到重试指令是一个数字值，供次要名称服务器用于确定在主名称服务器未应答的情况下发出刷新请求前等待的时长。如果主服务器未在时间到扩展指令所指定的时间之前回复刷新请求，则次要服务器停止响应与该命名空间相关的请求。

- 在 BIND 4 和 8 中，min-TTL 指令是其他名称服务器缓存区域信息的时间。在 BIND 9 中，它定义了将负答案缓存多长时间。负答案的缓存最多可设置为 3 小时(3H)。

配置 BIND 时，会以秒为单位指定所有时间。但是，在指定秒之外的时间单位时，可以使用缩写，如分钟(M)、小时(H)、天(D)和周(W)。表 15.6 “与其他时间单位相比的秒” 以秒为单位显示一个时间，以其他格式显示等同的时间。

表 15.6. 与其他时间单位相比的秒

秒	其他时间单位
60	1M
1800	30M
3600	1H
10800	3H
21600	6H
43200	12H
86400	1D
259200	3D
604800	1W
31536000	365D

例 15.14. 使用 SOA 资源记录

```
@  IN  SOA  dns1.example.com.  hostmaster.example.com. (
      2001062501  ; serial
      21600       ; refresh after 6 hours
      3600        ; retry after 1 hour
      604800      ; expire after 1 week
      86400 )     ; minimum TTL of 1 day
```

15.2.3.3. 注释标签

除了资源记录和指令外，区域文件也可以包含注释。指定服务将忽略注释，但可以在向用户提供附加信息时证明很有用。行末尾分号字符后面的任何文本都将被视为注释。例如：

```
604800  ; expire after 1 week
```

15.2.3.4. 用法示例

以下示例显示了区域文件的基本使用情况。

15.2.3.4.1. 简单区域文件

例 15.15 “简单区域文件” 演示标准指令和 SOA 值的使用。

例 15.15. 简单区域文件

```
$ORIGIN example.com.
$TTL 86400
@       IN  SOA  dns1.example.com.  hostmaster.example.com. (
            2001062501  ; serial
            21600       ; refresh after 6 hours
            3600        ; retry after 1 hour
            604800      ; expire after 1 week
            86400 )     ; minimum TTL of 1 day
;
;
        IN  NS     dns1.example.com.
        IN  NS     dns2.example.com.
dns1    IN  A      10.0.1.1
        IN  AAAA   aaaa:bbbb::1
dns2    IN  A      10.0.1.2
        IN  AAAA   aaaa:bbbb::2
;
;
@       IN  MX     10  mail.example.com.
        IN  MX     20  mail2.example.com.
mail    IN  A      10.0.1.5
        IN  AAAA   aaaa:bbbb::5
mail2   IN  A      10.0.1.6
        IN  AAAA   aaaa:bbbb::6
;
;
; This sample zone file illustrates sharing the same IP addresses
; for multiple services:
;
services  IN  A      10.0.1.10
        IN  AAAA   aaaa:bbbb::10
        IN  A      10.0.1.11
```

```
        IN  AAAA   aaaa:bbbb::11

ftp     IN  CNAME  services.example.com.
www     IN  CNAME  services.example.com.
;
;
```

在本示例中，权威名称服务器设置为 dns1.example.com 和 dns2.example.com，并且使用 A 记录分别连接到 10.0.1.1 和 10.0.1.2 IP 地址。

配置有 MX 记录的电子邮件服务器通过 A 记录指向邮件和 mail 2。由于这些名称不以尾随期结束，因此 $ORIGIN 域在后面放置，将其扩展到 mail.example.com 和 mail2.example.com。

标准名称中可用的服务，如 www.example.com (WWW)，使用 CNAME 记录指向相应的服务器。

这个区文件会被命名为 service，且 /etc/named.conf 中的 zone 语句类似如下：

```
zone "example.com" IN {
  type master;
  file "example.com.zone";
  allow-update { none; };
};
```

15.2.3.4.2. 反向名称解析区域文件

反向名称解析区域文件用于将特定命名空间中的 IP 地址转换为完全限定域名 (FQDN)。它看起来与标准区文件非常相似，但使用 PTR 资源记录将 IP 地址链接到完全限定域名，如 例 15.16 “反向名称解析区域文件” 所示。

例 15.16. 反向名称解析区域文件

```
$ORIGIN 1.0.10.in-addr.arpa.
$TTL 86400
@  IN  SOA  dns1.example.com.  hostmaster.example.com. (
       2001062501  ; serial
       21600       ; refresh after 6 hours
       3600        ; retry after 1 hour
       604800      ; expire after 1 week
       86400 )     ; minimum TTL of 1 day
;
@  IN  NS   dns1.example.com.
;
1  IN  PTR  dns1.example.com.
2  IN  PTR  dns2.example.com.
```

```
;
5  IN  PTR  server1.example.com.
6  IN  PTR  server2.example.com.
;
3  IN  PTR  ftp.example.com.
4  IN  PTR  ftp.example.com.
```

在本例中，IP 地址 10.0.1.1 到 10.0.1.6 将指向对应的完全限定域名。

这个区文件在服务中使用 /etc/named.conf 文件中的 zone 语句，如下所示：

```
zone "1.0.10.in-addr.arpa" IN {
  type master;
  file "example.com.rr.zone";
  allow-update { none; };
};
```

除了区域名称外，本示例和标准区域语句几乎没有区别。请注意，反向名称解析区域需要 IP 地址的前三个块反向 ，后跟.in-addr.arpa。这允许将反向名称解析区域文件中使用的单个 IP 编号块与 区域关联。

15.2.4. 使用 rndc 实用程序

Therndc 实用程序是一种命令行工具，允许您在本地和远程计算机管理指定服务。其用法如下：

```
rndc [option...] command [command-option]
```

15.2.4.1. 配置实用程序

要防止对服务的未授权访问，必须将 named 配置为侦听所选端口（默认为953 ），并且 服务和 rndc 实用程序必须使用相同的密钥。

表 15.7. 相关文件

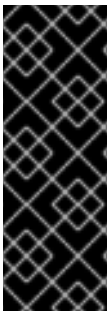
路径	描述
/etc/named.conf	named 服务的默认配置文件。
/etc/rndc.conf	rndc 实用程序的默认配置文件。

路径	描述
/etc/rndc.key	默认密钥位置。

Therndc 配置位于 /etc/rndc.conf 中。如果文件不存在，实用程序将使用位于 /etc/rndc.key 中的密钥，该密钥是在安装过程中使用 therndc-confgen -a 命令自动生成的。

named 服务使用 /etc/named.conf 配置文件中的 control 语句进行配置，如 第 15.2.2.3 节 “其他语句类型” 所述。除非存在此语句，否则仅允许来自回环地址(127.0.0.1)的连接，并且将使用位于 /etc/rndc.key 中的密钥。

有关此主题的更多信息，请参阅 第 15.2.8 节 “其它资源” 中列出的 man page 和 BIND 9 管理员参考手册。



重要

要防止非特权用户向服务发送控制命令，请确保只允许 root 用户读取 /etc/rndc.key 文件：

```
~]# chmod o-rwx /etc/rndc.key
```

15.2.4.2. 检查服务状态

要检查指定服务的当前状态，请使用以下命令：

```
~]# rndc status
version: 9.7.0-P2-RedHat-9.7.0-5.P2.el6
CPUs found: 1
worker threads: 1
number of zones: 16
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
recursive clients: 0/0/1000
tcp clients: 0/100
server is up and running
```

15.2.4.3. 重新加载配置和区域

要重新载入配置文件和区域，在 shell 提示符后输入以下内容：

```
~]# rndc reload
server reload successful
```

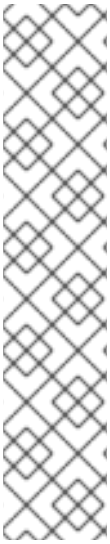
这会重新加载区域，同时保留所有之前缓存的响应，以便您可以更改区域文件而不丢失所有存储的名称解析。

要重新载入一个区，在重新载入命令后指定它的名称，例如：

```
~]# rndc reload localhost
zone reload up-to-date
```

最后，要重新载入配置文件和新添加的区域，请输入：

```
~]# rndc reconfig
```



注意

如果要手动修改使用 Dynamic DNS (DDNS) 的区域，请确保先运行停止命令 ：

```
~]# rndc freeze localhost
```

完成后，运行 thaw 命令再次允许 DDNS 并重新载入区：

```
~]# rndc thaw localhost
The zone reload and thaw was successful.
```

15.2.4.4. 更新区域密钥

若要更新 DNSSEC 密钥并签署区域，请使用 sign 命令。例如：

```
~]# rndc sign localhost
```

请注意，若要使用上述命令为区域签名，必须将 auto-dnssec 选项设置为在 zone 语句中 维护。例如：

-

```
zone "localhost" IN {
  type master;
  file "named.localhost";
  allow-update { none; };
  auto-dnssec maintain;
};
```

15.2.4.5. 启用 DNSSEC 验证

要启用 DNSSEC 验证，以 root 用户身份运行以下命令：

```
~]# rndc validation on
```

同样，要禁用这个选项，请输入：

```
~]# rndc validation off
```

有关如何在 /etc/named.conf 中配置这个选项的信息，请参阅 第 15.2.2.2 节 “常见语句类型” 中描述的选项语句。

红帽企业 Linux 7 安全指南包含有关 DNSSEC 的综合部分。

15.2.4.6. 启用查询日志记录

要启用（或者在当前启用时禁用）查询日志，以 root 用户身份运行以下命令：

```
~]# rndc querylog
```

要检查当前的设置，请使用 status 命令，如 第 15.2.4.2 节 “检查服务状态” 所述。

15.2.5. 使用 dig 实用程序

dig 实用程序是一种命令行工具，允许您执行 DNS 查找和调试名称服务器配置。其典型用法如下：

```
dig [@server] [option...] name type
```

有关类型常用值的列表，请参阅 第 15.2.3.2 节 “通用资源记录”。

15.2.5.1. 查找名称服务器

要查找特定域的名称服务器，请使用以下格式的命令：

```
dig name NS
```

在 例 15.17 “名称服务器查找示例” 中，dig 工具用于显示 example.com 的名称服务器。

例 15.17. 名称服务器查找示例

```
~]$ dig example.com NS

; <<>> DiG 9.7.1-P2-RedHat-9.7.1-2.P2.fc13 <<>> example.com NS
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57883
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;example.com.                   IN      NS

;; ANSWER SECTION:
example.com.            99374   IN      NS      a.iana-servers.net.
example.com.            99374   IN      NS      b.iana-servers.net.

;; Query time: 1 msec
;; SERVER: 10.34.255.7#53(10.34.255.7)
;; WHEN: Wed Aug 18 18:04:06 2010
;; MSG SIZE  rcvd: 77
```

15.2.5.2. 查找 IP 地址

要查找分配给特定域的 IP 地址，请使用以下格式 命令：

```
dig name A
```

在 例 15.18 “IP 地址查找示例” 中，dig 工具用于显示 example.com 的 IP 地址。

例 15.18. IP 地址查找示例

```
~]$ dig example.com A

; <<>> DiG 9.7.1-P2-RedHat-9.7.1-2.P2.fc13 <<>> example.com A
;; global options: +cmd
;; Got answer:
```

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4849
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 0

;; QUESTION SECTION:
;example.com.                IN      A

;; ANSWER SECTION:
example.com.          155606  IN      A       192.0.32.10

;; AUTHORITY SECTION:
example.com.          99175   IN      NS      a.iana-servers.net.
example.com.          99175   IN      NS      b.iana-servers.net.

;; Query time: 1 msec
;; SERVER: 10.34.255.7#53(10.34.255.7)
;; WHEN: Wed Aug 18 18:07:25 2010
;; MSG SIZE  rcvd: 93
```

15.2.5.3. 查找主机名

要查找特定 IP 地址的主机名，请使用以下命令：

```
dig -x address
```

在 例 15.19 “示例主机名查找” 中，dig 工具用于显示分配给 192.0.32.10 的主机名。

例 15.19. 示例主机名查找

```
~]$ dig -x 192.0.32.10

; <<>> DiG 9.7.1-P2-RedHat-9.7.1-2.P2.fc13 <<>> -x 192.0.32.10
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29683
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 5, ADDITIONAL: 6

;; QUESTION SECTION:
;10.32.0.192.in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
10.32.0.192.in-addr.arpa. 21600 IN      PTR     www.example.com.

;; AUTHORITY SECTION:
32.0.192.in-addr.arpa.  21600   IN      NS      b.iana-servers.org.
32.0.192.in-addr.arpa.  21600   IN      NS      c.iana-servers.net.
32.0.192.in-addr.arpa.  21600   IN      NS      d.iana-servers.net.
32.0.192.in-addr.arpa.  21600   IN      NS      ns.icann.org.
32.0.192.in-addr.arpa.  21600   IN      NS      a.iana-servers.net.
```

```
;; ADDITIONAL SECTION:
a.iana-servers.net.     13688   IN      A       192.0.34.43
b.iana-servers.org.     5844    IN      A       193.0.0.236
b.iana-servers.org.     5844    IN      AAAA    2001:610:240:2::c100:ec
c.iana-servers.net.     12173   IN      A       139.91.1.10
c.iana-servers.net.     12173   IN      AAAA    2001:648:2c30::1:10
ns.icann.org.           12884   IN      A       192.0.34.126

;; Query time: 156 msec
;; SERVER: 10.34.255.7#53(10.34.255.7)
;; WHEN: Wed Aug 18 18:25:15 2010
;; MSG SIZE  rcvd: 310
```

15.2.6. BIND 的高级功能

大多数 BIND 实施仅使用指定服务来提供名称解析服务或充当特定域的权威。但是，BIND 版本 9 具有许多高级功能，允许更加安全、高效的 DNS 服务。

重要

在尝试使用 DNSSEC、TSIG 或 IXFR（创建区域传输）等高级功能之前，请确保网络环境中所有名称服务器均支持特定功能，特别是您使用较旧版本的 BIND 或非 BIND 服务器时。

第 15.2.8.1 节 “安装的文档” 引用的 BIND 9 管理员参考手册中将更详细地阐述上述所有功能。

15.2.6.1. 多个视图

（可选）可以根据请求源自的网络向客户端呈现不同的信息。这主要用于拒绝来自本地网络外的客户端的敏感 DNS 条目，同时允许来自本地网络内的客户端进行查询。

要配置多个视图，请将 view 语句添加到 /etc/named.conf 配置文件。使用 match-clients 选项匹配 IP 地址或整个网络，并为它们提供特殊选项和区域数据。

15.2.6.2. 增量区域传输(IXFR)

增量区域传输 (IXFR)允许次要名称服务器仅下载主名称服务器上修改的区域的更新部分。与标准传输流程相比，这提高了通知和更新流程的效率。

请注意，只有使用动态更新来更改主区域记录时，IXFR 才可用。 如果手动编辑区域文件进行更改，

则使用自动区域传输(AXFR)。

15.2.6.3. 事务 SIGnatures(TSIG)

事务 SIGnatures(TSIG)确保在允许传输之前在主名称服务器和次要名称服务器上都存在共享密钥。这增强了基于标准 IP 地址的转让授权方法，因为攻击者不仅需要访问 IP 地址来传输区，还需要知道 secret 密钥。

自版本 9 起，BIND 也支持 TKEY，这是授权区域传送的另一种共享密钥方法。

重要

在通过不安全的网络进行通信时，请勿依赖于基于 IP 地址的身份验证。

15.2.6.4. DNS 安全扩展(DNSSEC)

域名系统安全扩展(DNSSEC)提供 DNS 数据的来源验证、验证存在拒绝和数据完整性。当特定域标记为安全时，会针对未通过验证的每个资源记录返回 SERVFAIL 响应。

请注意：要调试 DNSSEC 签名的域或 DNSSEC 识别解析器，您可以使用 dig 工具，如 第 15.2.5 节 “使用 dig 实用程序” 所述。有用的选项包括 +dnssec （请求 DNSSEC 相关资源记录是通过设置 DNSSEC OK 位）、+cd （递归名称服务器不验证响应）和 +bufsize=512 （将数据包大小更改为 512B 以通过某些防火墙）。

15.2.6.5. 互联网协议版本 6(IPv6)

使用 AAAA 资源记录和 listen-on-v6 指令支持互联网协议版本 6(IPv6)，如 表 15.3 “常用配置选项” 所述。

15.2.7. 通用 Mistakes to Avoid

以下是如何避免用户在配置名称服务器时出现常见错误的建议列表：

正确使用分号和大括号

/etc/named.conf 文件中的省略的分号或不匹配 curly 括号可能会阻止指定服务启动。

正确使用句点（ . 字符）

在区域文件中，域名末尾的句点表示完全限定域名。如果省略，named 服务将附加区域名称或 $ORIGIN 值来完成它。

编辑区域文件时增加序列号

如果未递增序列号，主名称服务器将具有正确的新信息，但是次要名称服务器永远不会收到更改的通知，并且不会尝试刷新该区域的数据。

配置防火墙

如果防火墙阻止了从指定服务到其他名称服务器的连接，建议的做法是更改防火墙设置。

警告

将固定 UDP 源端口用于 DNS 查询是一个潜在的安全漏洞，可能会让攻击者更轻松地执行缓存逐出攻击。为防止这种情况，默认情况下 DNS 发送自随机临时端口。将防火墙配置为允许来自随机 UDP 源端口的传出查询。默认情况下，使用范围 1024 到 65535。

15.2.8. 其它资源

以下信息来源提供了有关 BIND 的其他资源。

15.2.8.1. 安装的文档

BIND 具有涵盖许多不同的主题的完整已安装文档，每个文档都放置在其自身的主题目录中。对于以下每个项目，将 version 替换为系统中安装的 bind 软件包的版本：

/usr/share/doc/bind-version/

包含最新文档的主目录。目录包含 HTML 和 PDF 格式的 BIND 9 管理员参考手册，详细介绍 BIND 资源要求、如何配置不同类型的名称服务器、如何执行负载平衡和其他高级主题。

/usr/share/doc/bind-version/sample/etc/

包含指定配置文件示例的目录。

rndc(8)

rndc 名称服务器控制实用程序的 man page，其中包含其用法的文档。

named(8)

名为 的 Internet 域名服务器的 man page，其中包含可用于控制 BIND 名称服务器守护进程的分类参数文档。

lwresd(8)

轻量级解析器守护进程 man page lwresd，包含有关 守护进程及其用法的文档。

named.conf(5)

包含指定配置文件中可用选项的完整列表的 man page。

rndc.conf(5)

包含 rndc 配置文件中可用选项的完整列表的 man page。

15.2.8.2. 在线资源

https://access.redhat.com/site/articles/770133

有关在 chroot 环境中运行 BIND 的红帽知识库文章，包括与 Red Hat Enterprise Linux 6 相比的区别。

https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/

红帽企业 Linux 7 安全指南包含有关 DNSSEC 的综合部分。

https://www.icann.org/namecollision

域名冲突上的 ICANN 常见问题.

第 16 章 配置 SQUID 缓存代理服务器

Squid 是一种代理服务器，可缓存内容以降低带宽并更快地加载 Web 页面。本章论述了如何将 Squid 设置为 HTTP、HTTPS 和 FTP 协议的代理，以及验证和限制访问。

16.1. 在不使用身份验证的情况下将 SQUID 设置为缓存代理

这部分论述了 Squid 的基本配置在没有身份验证的情况下作为缓存代理。此流程会根据 IP 范围限制对代理的访问。

先决条件

- 这个过程假设 /etc/squid/squid.conf 文件是 squid 软件包提供的。如果您在之前编辑了这个文件，请删除该文件并重新安装该软件包。

流程

1. 安装 squid 软件包：

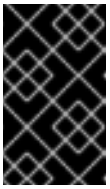
```
# yum install squid
```

2. 编辑 /etc/squid/squid.conf 文件：

   a. 调整 localnet 访问控制列表(ACL)，使其与允许使用代理的 IP 范围匹配：

```
acl localnet src 192.0.2.0/24
acl localnet 2001:db8::/32
```

默认情况下，/etc/squid/squid.conf 文件包含 http_access allow localnet 规则，允许使用 localnet ACL 中指定的所有 IP 范围内的代理。请注意，您必须在 http_access allow localnet 规则之前指定所有 localnet ACL。



重要

删除所有与您的环境不匹配的现有 acl localnet 条目。

b.

默认配置中存在以下 ACL，并将 443 定义为使用 HTTPS 协议的端口：

```
acl SSL_ports port 443
```

如果用户也可以在其它端口上使用 HTTPS 协议，请为每个端口添加 ACL：

```
acl SSL_ports port port_number
```

c.

更新 acl Safe_ports 规则列表，以配置 Squid 可以建立连接的端口。例如，若要配置使用代理的客户端只能访问端口 21(FTP)、80(HTTP)和 443(HTTPS)上的资源，在配置中仅保留以下 acl Safe_ports 语句：

```
acl Safe_ports port 21
acl Safe_ports port 80
acl Safe_ports port 443
```

默认情况下，配置中包含 http_access deny !Safe_ports 规则，该规则定义对 Safe_ports ACL 中没有定义的端口的访问拒绝。

d.

在 cache_dir 参数中配置缓存类型、缓存目录的路径、缓存大小以及进一步缓存特定于类型的设置：

```
cache_dir ufs /var/spool/squid 10000 16 256
```

使用这些设置：

- Squid 使用ufs 缓存类型.
- Squid 将其缓存存储在 /var/spool/squid/ 目录中。
- 缓存最多增长 10000 MB。
- Squid 在 /var/spool/squid/ 目录中创建 16 级-1 子目录。

- Squid 在每个级别 1 目录中创建 256 个子目录。

如果您没有设置 cache_dir 指令，Squid 会将缓存保存在内存中。

3. 如果您在 cache_dir 参数中设置与 /var/spool/squid/ 不同的缓存目录：

a. 创建缓存目录：

```
# mkdir -p path_to_cache_directory
```

b. 配置缓存目录的权限：

```
# chown squid:squid path_to_cache_directory
```

c. 如果您以 enforcing 模式运行 SELinux，请为缓存目录设置 squid_cache_t 上下文：

```
# semanage fcontext -a -t squid_cache_t "path_to_cache_directory(/.*)?"
# restorecon -Rv path_to_cache_directory
```

如果您的系统中没有 semanage 工具，请安装 policycoreutils-python-utils 软件包。

4. 在防火墙中打开 3128 端口：

```
# firewall-cmd --permanent --add-port=3128/tcp
# firewall-cmd --reload
```

5. 启动 squid 服务：

```
# systemctl start squid
```

6. 在系统引导时启用 squid 服务自动启动：

```
# systemctl enable squid
```

验证步骤

要验证代理是否正常工作，请使用 curl 实用程序下载网页：

```
# curl -O -L "https://www.redhat.com/index.html" -x "proxy.example.com:3128"
```

如果 curl 没有显示任何错误，并且 index.html 文件已下载到当前目录中，代理可以正常工作。

16.2. 使用 LDAP 身份验证将 SQUID 设置为缓存代理

本节论述了 Squid 作为使用 LDAP 验证用户身份的缓存代理的基本配置。此流程配置仅经过身份验证的用户可以使用代理。

先决条件

- **这个过程假设 /etc/squid/squid.conf 文件是 squid 软件包提供的。如果您在之前编辑了这个文件，请删除该文件并重新安装该软件包。**
- **LDAP 目录中有一个服务用户，如 uid=proxy_user,cn=users,cn=accounts,dc=example,dc=com。Squid 只使用此帐户搜索验证用户。如果存在身份验证用户，Squid 会以此用户的身份绑定到该目录以验证身份验证。**

流程

1. **安装 squid 软件包：**

```
# yum install squid
```

2. **编辑 /etc/squid/squid.conf 文件：**

   a. **要配置 basic_ldap_auth 帮助程序，请在 /etc/squid/squid.conf 顶部添加以下配置条目：**

```
auth_param basic program /usr/lib64/squid/basic_ldap_auth -b
"cn=users,cn=accounts,dc=example,dc=com" -D
"uid=proxy_user,cn=users,cn=accounts,dc=example,dc=com" -W
/etc/squid/ldap_password -f "(&(objectClass=person)(uid=%s))" -ZZ -H
ldap://ldap_server.example.com:389
```

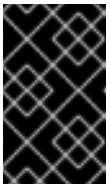
下面描述了传递给上例中 basic_ldap_auth helper 工具的参数：

- -B base_DN 设置 LDAP 搜索基础。
- -D proxy_service_user_DN 设置 Squid 所用帐户的可分辨名称(DN)，以在 目录中搜索身份验证用户。
- -W path_to_password_file 设置包含代理服务用户密码的文件路径。使用密码文件可防止密码在操作系统的进程列表中可见。
- -f LDAP_filter 指定 LDAP 搜索过滤器。Squid 将 %s 变量替换为身份验证用户提供的用户名。

  示例中的（&(¹=person)(uid=%s) 过滤器定义用户名必须与 uid 属性中设置的值匹配，并且目录条目包含用户对象类。
- -ZZ 使用 STARTTLS 命令通过 LDAP 协议强制实施 TLS 加密的连接。在以下情况下省略 -ZZ ：
  - LDAP 服务器不支持加密的连接。
  - URL 中指定的端口使用 LDAPS 协议。
- -H LDAP_URL 参数以 URL 格式指定协议、主机名或 IP 地址以及 LDAP 服务器的端口。

b. 添加以下 ACL 和规则来配置 Squid 只允许经过身份验证的用户使用代理：

```
acl ldap-auth proxy_auth REQUIRED
http_access allow ldap-auth
```



重要

在 http_access 拒绝所有规则前指定这些设置。

c.

删除以下规则以禁用从 localnet ACL 中指定的 IP 范围禁用代理身份验证：

```
http_access allow localnet
```

d.

默认配置中存在以下 ACL，并将 443 定义为使用 HTTPS 协议的端口：

```
acl SSL_ports port 443
```

如果用户也可以在其它端口上使用 HTTPS 协议，请为每个端口添加 ACL：

```
acl SSL_ports port port_number
```

e.

更新 acl Safe_ports 规则列表，以配置 Squid 可以建立连接的端口。例如，若要配置使用代理的客户端只能访问端口 21(FTP)、80(HTTP)和 443(HTTPS)上的资源，在配置中仅保留以下 acl Safe_ports 语句：

```
acl Safe_ports port 21
acl Safe_ports port 80
acl Safe_ports port 443
```

默认情况下，配置中包含 http_access deny !Safe_ports 规则，该规则定义对 Safe_ports ACL 中没有定义的端口的访问拒绝。

f.

在 cache_dir 参数中配置缓存类型、缓存目录的路径、缓存大小以及进一步缓存特定于类型的设置：

```
cache_dir ufs /var/spool/squid 10000 16 256
```

使用这些设置：

- Squid 使用ufs 缓存类型.

- Squid 将其缓存存储在 /var/spool/squid/ 目录中。
- 缓存最多增长 10000 MB。
- Squid 在 /var/spool/squid/ 目录中创建 16 级-1 子目录。
- Squid 在每个级别 1 目录中创建 256 个子目录。

如果您没有设置 cache_dir 指令，Squid 会将缓存保存在内存中。

3. 如果您在 cache_dir 参数中设置与 /var/spool/squid/ 不同的缓存目录：

a. 创建缓存目录：

```
# mkdir -p path_to_cache_directory
```

b. 配置缓存目录的权限：

```
# chown squid:squid path_to_cache_directory
```

c. 如果您以 enforcing 模式运行 SELinux，请为缓存目录设置 squid_cache_t 上下文：

```
# semanage fcontext -a -t squid_cache_t "path_to_cache_directory(/.*)?"
# restorecon -Rv path_to_cache_directory
```

如果您的系统中没有 semanage 工具，请安装 policycoreutils-python-utils 软件包。

4. 将 LDAP 服务用户的密码存储在 /etc/squid/ldap_password 文件中，并为该文件设置适当的权限：

```
# echo "password" > /etc/squid/ldap_password
# chown root:squid /etc/squid/ldap_password
# chmod 640 /etc/squid/ldap_password
```

5. 在防火墙中打开 3128 端口：

```
# firewall-cmd --permanent --add-port=3128/tcp
# firewall-cmd --reload
```

6. 启动 squid 服务：

```
# systemctl start squid
```

7. 在系统引导时启用 squid 服务自动启动：

```
# systemctl enable squid
```

验证步骤

要验证代理是否正常工作，请使用 curl 实用程序下载网页：

```
# curl -O -L "https://www.redhat.com/index.html" -x
"user_name:password@proxy.example.com:3128"
```

如果 curl 没有显示任何错误，并且 index.html 文件已下载到当前目录中，代理可以正常工作。

故障排除步骤

验证 helper 工具是否正常工作：

1. 使用您在 auth_param 参数中使用的相同设置手动启动帮助程序：

```
# /usr/lib64/squid/basic_ldap_auth -b "cn=users,cn=accounts,dc=example,dc=com" -D
"uid=proxy_user,cn=users,cn=accounts,dc=example,dc=com" -W /etc/squid/ldap_password -
f "(&(objectClass=person)(uid=%s))" -ZZ -H ldap://ldap_server.example.com:389
```

2. 输入一个有效的用户名和密码，然后按 Enter 键：

```
user_name password
```

如果帮助程序返回 OK，则身份验证成功。

16.3. 使用 KERBEROS 身份验证将 SQUID 设置为缓存代理

这部分论述了 Squid 作为缓存代理的基本配置，它使用 Kerberos 向 Active Directory(AD)验证用户。此流程配置仅经过身份验证的用户可以使用代理。

先决条件

- 这个过程假设 /etc/squid/squid.conf 文件是 squid 软件包提供的。如果您在之前编辑了这个文件，请删除该文件并重新安装该软件包。
- 要安装 Squid 的服务器是 AD 域的成员。详情请参阅《Red Hat Enterprise Linux 7 系统管理员指南》中的将 Samba 设置为域成员 。

流程

1. 安装以下软件包：

```
# yum install squid krb5-workstation
```

2. 以 AD 域管理员身份进行身份验证：

```
# kinit administrator@AD.EXAMPLE.COM
```

3. 为 Squid 创建一个 keytab，并将其存储在 /etc/squid/HTTP.keytab 文件中：

```
# export KRB5_KTNAME=FILE:/etc/squid/HTTP.keytab
# net ads keytab CREATE -U administrator
```

4. 在 keytab 中添加 HTTP 服务主体：

```
# net ads keytab ADD HTTP -U administrator
```

5. 将 keytab 文件的所有者设置为 squid 用户：

```
# chown squid /etc/squid/HTTP.keytab
```

6. 另外，还可验证 **keytab** 文件是否包含代理服务器的完全限定域名(FQDN)的 **HTTP** 服务主体：

```
#  klist -k /etc/squid/HTTP.keytab
Keytab name: FILE:/etc/squid/HTTP.keytab
KVNO Principal
---- --------------------------------------------------------------------------
...
   2 HTTP/proxy.ad.example.com@AD.EXAMPLE.COM
...
```

7. 编辑 **/etc/squid/squid.conf** 文件：

   a. 要配置**consberos_auth** 帮助程序程序，请在 **/etc/squid/squid.conf** 的顶部添加以下配置条目：

```
auth_param negotiate program /usr/lib64/squid/negotiate_kerberos_auth -k
/etc/squid/HTTP.keytab -s HTTP/proxy.ad.example.com@AD.EXAMPLE.COM
```

下面描述了传递给上例中协商_**kerberos_auth** 帮助程序的参数：

- **-k file** 设置密钥选项卡文件的路径。请注意，**squid** 用户必须具有此文件的读取权限。
- **-s HTTP/host_name@kerberos_realm** 设置 **Squid** 使用的 **Kerberos** 主体。

另外，您可以通过将以下一个或多个参数传递给帮助程序来启用日志：

- **-i** 记录信息性消息，如身份验证用户。
- **-d** 启用调试日志记录。

Squid 将 **helper** 实用程序中的调试信息记录到 **/var/log/squid/cache.log** 文件。

b.

添加以下 ACL 和规则来配置 Squid 只允许经过身份验证的用户使用代理：

```
acl kerb-auth proxy_auth REQUIRED
http_access allow kerb-auth
```

重要

在 http_access 拒绝所有规则前指定这些设置。

c.

删除以下规则以禁用从 localnet ACL 中指定的 IP 范围禁用代理身份验证：

```
http_access allow localnet
```

d.

默认配置中存在以下 ACL，并将 443 定义为使用 HTTPS 协议的端口：

```
acl SSL_ports port 443
```

如果用户也可以在其它端口上使用 HTTPS 协议，请为每个端口添加 ACL：

```
acl SSL_ports port port_number
```

e.

更新 acl Safe_ports 规则列表，以配置 Squid 可以建立连接的端口。例如，若要配置使用代理的客户端只能访问端口 21(FTP)、80(HTTP)和 443(HTTPS)上的资源，在配置中仅保留以下 acl Safe_ports 语句：

```
acl Safe_ports port 21
acl Safe_ports port 80
acl Safe_ports port 443
```

默认情况下，配置中包含 http_access deny !Safe_ports 规则，该规则定义对 Safe_ports ACL 中没有定义的端口的访问拒绝。

f.

在 cache_dir 参数中配置缓存类型、缓存目录的路径、缓存大小以及进一步缓存特定于类型的设置：

```
cache_dir ufs /var/spool/squid 10000 16 256
```

使用这些设置：

- Squid 使用ufs 缓存类型.
- Squid 将其缓存存储在 /var/spool/squid/ 目录中。
- 缓存最多增长 10000 MB。
- Squid 在 /var/spool/squid/ 目录中创建 16 级-1 子目录。
- Squid 在每个级别 1 目录中创建 256 个子目录。

如果您没有设置 cache_dir 指令，Squid 会将缓存保存在内存中。

8. 如果您在 cache_dir 参数中设置与 /var/spool/squid/ 不同的缓存目录：

   a. 创建缓存目录：

   ```
   # mkdir -p path_to_cache_directory
   ```

   b. 配置缓存目录的权限：

   ```
   # chown squid:squid path_to_cache_directory
   ```

   c. 如果您以 enforcing 模式运行 SELinux，请为缓存目录设置 squid_cache_t 上下文：

   ```
   # semanage fcontext -a -t squid_cache_t "path_to_cache_directory(/.*)?"
   # restorecon -Rv path_to_cache_directory
   ```

   如果您的系统中没有 semanage 工具，请安装 policycoreutils-python-utils 软件包。

9. 在防火墙中打开 3128 端口：

```
# firewall-cmd --permanent --add-port=3128/tcp
# firewall-cmd --reload
```

10. 启动 squid 服务：

```
# systemctl start squid
```

11. 在系统引导时启用 squid 服务自动启动：

```
# systemctl enable squid
```

验证步骤

要验证代理是否正常工作，请使用 curl 实用程序下载网页：

```
# curl -O -L "https://www.redhat.com/index.html" --proxy-negotiate -u : -x
"proxy.ad.example.com:3128"
```

如果 curl 没有显示任何错误，并且 index.html 文件存在于当前目录中，则代理可以正常工作。

故障排除步骤

手动测试 Kerberos 身份验证：

1. 为 AD 帐户获取 Kerberos ticket：

```
# kinit user@AD.EXAMPLE.COM
```

2. 显示 ticket（可选）：

```
# klist
```

3. 使用 talk_kerberos_auth_test 工具来测试身份验证：

```
# /usr/lib64/squid/negotiate_kerberos_auth_test proxy.ad.example.com
```

如果帮助程序返回令牌，则身份验证成功。

```
Token: YIIFtAYGKwYBBQUCoIIFqDC...
```

16.4. 在 SQUID 中配置域黑名单

通常,管理员想要阻止对特定域的访问。这部分论述了如何在 Squid 中配置域黑名单。

先决条件

- squid 被配置，用户可以使用代理。

流程

1. 编辑 /etc/squid/squid.conf 文件并添加以下设置：

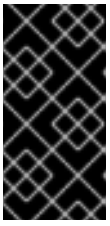
```
acl domain_blacklist dstdomain "/etc/squid/domain_blacklist.txt"
http_access deny all domain_blacklist
```

重要

在允许访问用户或客户端的第一个 http_access allow 语句前面添加这些条目。

2. 创建 /etc/squid/domain_blacklist.txt 文件，并添加您要阻止的域。例如，要阻止对 example.com 包括子域和 block example.net 的访问，请添加：

```
.example.com
example.net
```



重要

如果您在 squid 配置中引用了 /etc/squid/domain_blacklist.txt 文件，则此文件不能为空。如果文件为空，Squid 无法启动。

3. 重启 squid 服务：

```
# systemctl restart squid
```

16.5. 将 SQUID 服务配置为在特定端口或 IP 地址上侦听

默认情况下，Squid 代理服务侦听所有网络接口上的 3128 端口。这部分论述了如何更改端口并配置 Squid 在特定 IP 地址中侦听。

先决条件

- 已安装 squid。

流程

1. 编辑 /etc/squid/squid.conf 文件：

   - 要设置 Squid 服务侦听的端口，请在 http_port 参数中设置端口号。例如，要将端口设置为 8080，请设置：

     ```
     http_port 8080
     ```

   - 要配置 Squid 服务侦听的 IP 地址，请在 http_port 参数中设置 IP 地址和端口号。例如，要配置 Squid 侦听端口 3128 中的 192.0.2.1 IP 地址，请设置：

     ```
     http_port 192.0.2.1:3128
     ```

   在配置文件中添加多个 http_port 参数以配置 Squid 侦听多个端口和 IP 地址：

   ```
   http_port 192.0.2.1:3128
   http_port 192.0.2.1:8080
   ```

2. 如果您配置了 Squid，则使用其他端口作为默认值(3128)：

   a. 在防火墙中打开端口：

   -

```
# firewall-cmd --permanent --add-port=port_number/tcp
# firewall-cmd --reload
```

b. 如果您以 **enforcing** 模式运行 **SELinux**，请将端口分配给 **squid_port_t** 端口类型定义：

```
# semanage port -a -t squid_port_t -p tcp port_number
```

如果您的系统中没有 **semanage** 工具，请安装 **policycoreutils-python-utils** 软件包。

3. 重启 **squid** 服务：

```
# systemctl restart squid
```

16.6. 其它资源

- 有关您可以在 **/etc/squid/squid/squid.conf** 文件中设置的所有配置参数列表以及详细描述，请参阅/**usr/share/doc/squid- <version> /squid.doc.documented** 文件。

附录 A. 红帽客户门户网站 LABS 与网络相关

红帽客户门户 Labs 是旨在帮助您提高性能、解决问题、识别安全问题和优化配置的工具。本附录提供与网络相关的红帽客户门户实验室概述。所有红帽客户门户 Labs 均可通过以下网址访问：

网桥配置

网桥配置旨在为使用 Red Hat Enterprise Linux 5.4 或更高版本的应用程序（如 KVM）配置桥接网络接口。

网络绑定帮助程序

网络绑定帮助器允许管理员使用绑定内核模块和绑定网络接口将多个网络接口控制器绑定到单个频道。

使用网络绑定帮助程序启用两个或多个网络接口，以充当一个绑定接口。

数据包捕获语法生成器

Packet 捕获语法生成器可帮助您捕获网络数据包。

使用 Packet 捕获语法生成器，以生成选择接口的 tcpdump 命令，然后将信息打印到控制台。您需要 root 访问权限才能输入命令。

附录 B. 修订历史记录

修订 0.10-06 Tue 03 Mar 2020 Marc Muehlfeld
将配置基于策略的路由添加到 Define alternatives Routes 部分。

修订 0.10-05 Fri 22 Nov 2019 Marc Muehlfeld
重写"配置 Squid 缓存代理服务器"章节.

修订 0.10-04 Tue 06 Aug 2019 Marc Muehlfeld
7.7 GA 发行的版本.

修订 0.10-03 Thu 22 Mar 2018 Ioanna Gkioka
7.5 GA 发行的版本.

修订 0.10-02 Mon 14 Aug 2017 Ioanna Gkioka
带有 misc 更新的异步版本。

修订 0.10-01 Tue 25 Jul 2017 Mirek Jahoda
7.4 GA 发行的版本.

修订 0.9-30 Tue 18 Oct 2016 Mirek Jahoda
7.3 GA 发布版本.

修订 0.9-25 Wed 11 Nov 2015 Jana Heves
7.2 GA 版本.

修订 0.9-15 Tue 17 Feb 2015 Christian Huffman
7.1 GA 版本

修订 0.9-14 Fri Dec 05 2014 Christian Huffman
更新了 nmtui 和 NetworkManager GUI 部分。

修订 0.9-12 Wed Nov 05 2014 Stephen Wadeley
改进了 *IP 网络*、*802.1Q VLAN 标记和合作*.

修订 0.9-11 Tues Oct 21 2014 Stephen Wadeley
改进绑定、*桥接和团队*.

修订 0.9-9 Tue Sep 2 2014 Stephen Wadeley
改进了绑定和一致网络设备命名.

修订 0.9-8 Tue July 8 2014 Stephen Wadeley
《网络指南》的红帽企业 Linux 7.0 GA 版本.

修订 0-0 Wed Dec 12 2012 Stephen Wadeley
红帽企业 Linux 7 网络指南的初始化.

B.1. 致谢

该文本的某些部分首先出现在《红帽企业 Linux 6 部署指南》中，版权所有 © 2014 Red Hat, Inc.，网址为.

索引

符号

/etc/named.conf (见 BIND)

主名称服务器 (见 BIND)

内核模块
  模块参数
    绑定模块参数，绑定模块指令

  绑定模块，使用频道绑定
    description，使用频道绑定
    到绑定接口的参数，绑定模块指令

动态主机配置协议 (见 DHCP)

权威名称服务器 (见 BIND)

根名称服务器 (见 BIND)

次要名称服务器 (见 BIND)

资源记录 (见 BIND)

递归名称服务器 (见 BIND)

频道绑定
  description，使用频道绑定
  到绑定接口的参数，绑定模块指令
  配置，使用频道绑定

频道绑定接口 (见 内核模块)

B

Berkeley Internet 名称域 (见 BIND)

BIND
  zones
    $INCLUDE 指令，常用指令
    $ORIGIN 指令，常用指令
    $TTL 指令，常用指令

A (Address)资源记录，通用资源记录

CNAME (Canonical Name)资源记录，通用资源记录

description，名称服务器区域

MX （邮件交换）资源记录，通用资源记录

NS (Nameserver)资源记录，通用资源记录

PTR (Pointer)资源记录，通用资源记录

SOA （授权起始）资源记录，通用资源记录

注释标签，注释标签

示例用法，简单区域文件，反向名称解析区域文件

其他资源，在线资源

安装的文档，安装的文档

功能

DNS 安全扩展(DNSSEC)，DNS 安全扩展(DNSSEC)

事务 SIGnature(TSIG)，事务 SIGnatures(TSIG)

互联网协议版本 6(IPv6)，互联网协议版本 6(IPv6)

增量区域传输(IXFR)，增量区域传输(IXFR)

多个视图，多个视图

自动区传输(AXFR)，增量区域传输(IXFR)

工具

dig，BIND 作为名称服务器，使用 dig 实用程序，DNS 安全扩展(DNSSEC)

named，BIND 作为名称服务器，配置指定服务

rndc，BIND 作为名称服务器，使用 rndc 实用程序

常见错误，通用 Mistakes to Avoid

文件

/etc/named.conf，配置指定服务，配置实用程序

/etc/rndc.conf，配置实用程序

/etc/rndc.key，配置实用程序

目录

/etc/named/，配置指定服务

/var/named/，编辑区域文件

/var/named/data/，编辑区域文件

/var/named/dynamic/，编辑区域文件

/var/named/slaves/，编辑区域文件

类型

主（主）名称服务器，名称服务器区域，名称服务器类型

权威名称服务器，名称服务器类型

次要（从）名称服务器，名称服务器区域，名称服务器类型

递归名称服务器，名称服务器类型

资源记录，名称服务器区域

配置

ACL 声明，常见语句类型

Control 语句，其他语句类型

include 语句，常见语句类型

Logging 语句，其他语句类型

options 声明，常见语句类型

trusted-keys 语句，其他语句类型

zone 语句，常见语句类型

服务器声明，其他语句类型

查看声明，其他语句类型

注释标签，注释标签

键声明，其他语句类型

bonding (见 频道绑定)

D

DHCP，DHCP 服务器

dhcpd.conf，配置文件

dhcpd.leases，启动和停止服务器

dhcpd6.conf，用于 IPv6 的 DHCP(DHCPv6)

DHCPv6，用于 IPv6 的 DHCP(DHCPv6)

dhcrelay，DHCP 转发代理

group，配置文件

shared-network，配置文件

subnet，配置文件

使用原因，为什么使用 DHCP？

停止服务器，启动和停止服务器

全局参数，配置文件

其他资源，其它资源

启动服务器，启动和停止服务器

命令行选项，启动和停止服务器

服务器配置，配置 DHCP 服务器

转发代理，DHCP 转发代理

选项，配置文件

dhcpd.conf，配置文件

dhcpd.leases，启动和停止服务器

dhcrelay，DHCP 转发代理

dig (见 BIND)

DNS

定义，DNS 服务器

(参见 BIND)

M

Multihomed DHCP

主机配置，主机配置

服务器配置，配置多homed DHCP 服务器

N

name server (见 DNS)

named (见 BIND)

NIC

绑定到单个频道，使用频道绑定

R

rndc (见 BIND)