应用说明

主要特征

TCP/TP 协议

Firewall 和 NAT-包状态过滤：P2P 协议过滤；源和目标 NAT;对源 MAC,IP 地址.端口.IP 协议. 协议（ICMP.TCP.MSS 等），接口，对内部的数据包和连接作标记，TOS 字节，内容过滤，顺序优先与数据频繁和时间控制，包长度控制….

路由-静态路由：多线路平衡路由；基于策列的路由（在防火墙中分类）；RIP vI/v2,OSPF v2,BGP v4

数据流控制-能对每个 IP,协议，子网，端口，防火墙标记做流量控制；支持 PCQ,RED,SFQ,FIFO 对列；Peer-to-peer 协议限制

Hotspot- Hotspot 认证网关支持 RADIUS 验证和记录；用户可用即用访问网络；流量控制功能；具备防火墙功能；实时信息状态显示；自定义 HTML 登录页；支持 IPass; 支持 SSL 安全验证, 支持广告功能。

点对点隧道协议-支持 PPTP,PPPOE 和 L2TP 访问控制和客户端；支持 CHAP,MSCHPv1 和 MSCHAPv2 验证协议；支持 RADIUS 验证和记录；MPPE 加密；PPPOE 压缩：数据流控制；具备防火墙功能；支持 PPPoE 按需拨号。

简单隧道-IPIP 隧道，EoIP 隧道（Ethernet over IP）

Ipsec-支持 IP 安全加密 AH 和 ESP 协议；

Proxy-支持 FTP 和 HTTP 缓存服务器；支持 HTTPS 代理；支持透明代理；支持 SOCKS 协议；DNS static entries;支持独立的缓存驱动器；访问控制列表；支持父系代理。

DHCP-DHCP 服务器；DHCP 接力；DHCP 客户端；多 DHCP 网络；静态和动态 DHCP 租.约；支持 RADIUS.

VRRP-高效率的 VRRP 协议

UpnP-支持即插即用

NTP-网络对时协议服务器和客户端；同步 GPS 系统

Monitoring/accounting-IP 传输日志记录；防火墙活动记录；静态 HTTP 图形资源管理。

SNMP-只读访问

M3P-mikrotik 分包协议，支持无限连接和以太网。

MNDP-mikrotik 邻近探测协议；同样支持思科的 CDP

Tools-ping;traceroute;bandwidth test;ping flood;telnet;SSH;packet sniffer;DDNS.

……

二层链接

Wireless-IEEE802.11a/b/g Wireless client 和访问节点（AP）;Nstreme2 协议；无限分布系统（WDS）;虚拟 AP 功能；40 和 104bit WEP;WEP pre-shared key 加密；访问控制列表；RADIUS 服务器验证；漫游功能（Wireless 客户端）；接入点桥接功能

Bridge-支持生成树协议（STP）:多桥接口：桥防火墙；MAC NAT 功能

VLAN-IEEE802.1q virtual LAN,支持以太网和无线连接；多 VLAN 支持；VLAN 桥接

Synchronous-V.35,V.24,E1/T1,X.21,DS3(T3)媒体类型；sync-ppp,cisco HDLC,帧中继协议；ANSI-617d(ANDI or annex D)和 Q933（CCITT or annex A）帧中继 LMI 类型

Synchronous-一串型 PPP dial-in/dial-out;PAP,CHAP,MSCHAPV1 和 MSCHAPV2 验证协议；RADIUS 验证和记录；支持串口；modem 池支持 128 个端口

ISDN-ISDN dial-in/dial-out;PAP,CHAP,MSCHAPv1 和 MSCHAPv2 验证协议；RADIUS 验证和记录;cisco HDLC,X75UI,X75bui 对列支持

…….

硬件要求

CPU 和主板-核心频率在 100MHZ 或更高的单核心 i386 处理器，以及与兼容的主板。

RAM-最小 32MiB,最大 1GiB;推荐 64 MiB 或更高。

ROM-标准 ATA/IDE 接口，USB 接口和 SATA 接口（SCSI 不支持；RAID 控制器驱动不支持；）最小需要 64Mb

空间：Flash 和一些微型驱动器使用 ATA 接口能连接使用。

MIPS 硬件要求

支持系统-4kc routerBOARD 500(532,512 和 511)与 routerBOARD100（133.133C.150.192）

支持系统-24kc routerBOARD 400（411/411U/411AH, 433/433AH/450/450G, 493/493AH）

RAM-最小 16MiB

ROM-板载 NAND 驱动，最小 64Mb

PPC 硬件要求

RouterBOARD 1100, RouterBOARD 800, RouterBOARD 600, RouterBOARD 333

配置

RouterOS 提供了强大的命令配置接口。你同样可以通过简易的 Windows 远程图形软件 WinBox 管理路由器。Web 配置提供了多数常用的功能上。主要特征：

完全一至德用户接口

运行时配置和监控

支持多个连接访问

用户侧列配置

活动历史记录，undo、redo 操作

安全模式操作

Scripts 能事先安排行内容，脚本支持所有的命令操作

路由器可用通过下面的接口进行管理：

本地 teminal console-PS/2 或 USB 键盘和 VGA 显示卡进行控制

Serial console-任何（默认使用 COM1）RS232 异步串口，串口默认设置为 9600bit/s,1stop

bit,no parity,hardware(RTS.CTS)flow control.

telnet- telnet 服务默认运行在 TCP 端口 23

SSH-SSH(安全 shell)服务默认运行在 TCP 端口 22

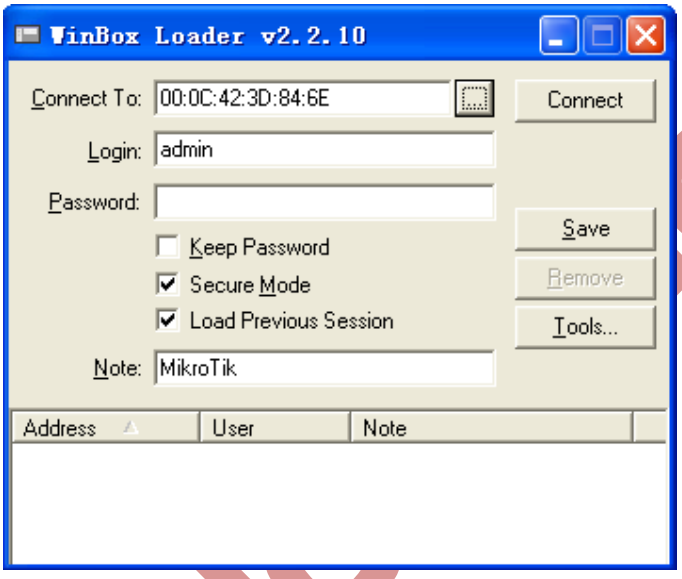
MAC Telnet-mikrotik MAC telnet 协议被默认启用在所以类以太网网卡接口上

Winbox- Winbox 是 routerOS 的一 windows 运程图形管理软件，使用 TCP 端口 8291(3.0rc13 版本后支持修改 Winbox 的端口)，同样也可用通过 MAC 地址连接。

第一章：routerOS 基本操作

RouterOS 各种登录方式

Mikrotik routerOS 内能通过运程配置各种参数，包括 Telnet,SSH,winbox 和 webbox.在这里我们将着介绍怎样使用 Winbox:



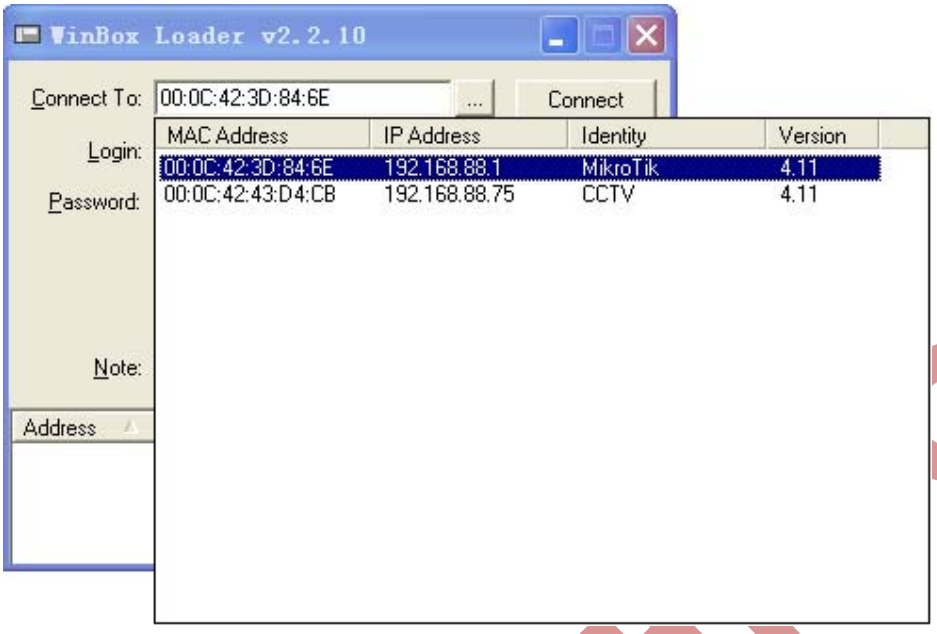
MAC-telnet 是在路由器没有 IP 地址的情况下或者配置防火墙参数后无法连接，通过路由器网卡 MAC 地址登陆的方式远程连接到路由器，MAC-telnet 仅能使用在来自同一个广播域中（因此在网络中不能有路由的存在），路由器的网卡应该被启用。注：在 winbox 中通过 MAC 地址连接路由器的功能，并内置了探测工具。这样在管理员忘记或复位了路由器后，同样可以通过 MAC 登录到 routerOS 上，进行图形界面操作。

Winbox 图形操作界面

Winbox 控制台适用于 mikrotik routerOS 的管理和配置，使用图形管理和配置，使用它型管理接口（GUI）。通过连接到 mikrotik 路由器的 HTTP(TCP 80 端口)欢迎界面下载 Winbox.exe 可行文件，下载并保存在你的 windows 中，之后直接在你 windows 电脑上运行 Winbox.exe 文件。

下面是对相应的键做介绍：

搜索和显示 MNDP（mikrotik Neighbor discovery protocol）或 CDP(cisco discovery protocol)设备。可以通过该功能键搜索同一子网内 mikrotik 和 cisco 设备。并能通过 MAC 地址登陆到 mikrotik routerOS 进行操作。



注：在 Winbox2.2.12 后的版本增加了 MAC 地址和 IP 地址选择功能，可根据搜索内容选择使用 MAC 地址连接或是 IP 地址连接。

Connect

通过指定的 IP 地址（默认端口为 80，不许特别指定，如果你修改了端口需要对具体访问端口做自定）或 MAC 地址（如果路由器在同一子网内）登陆路由器。

Save

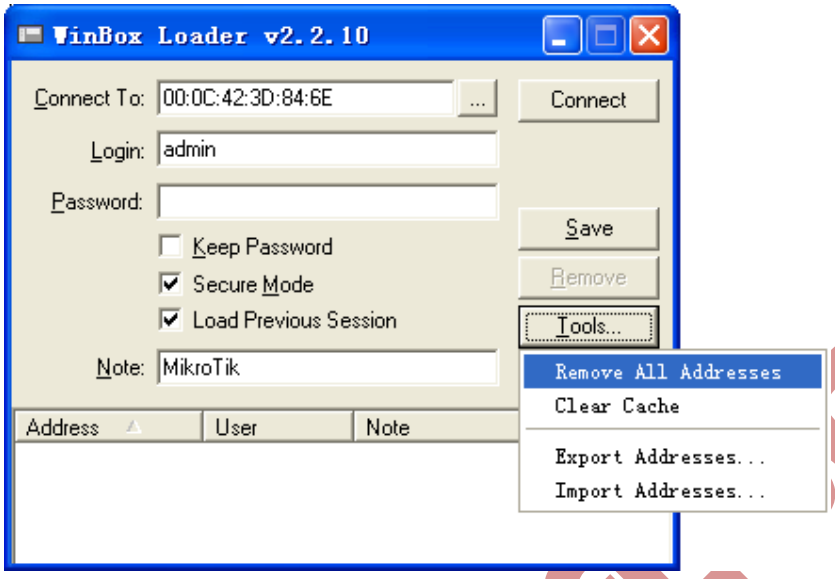
保存当前连接列表（当需要运行它们时。只需双击）

Remove

删除从列表中选择的项目

Tools...

所有列表中的项目，清除在本地的缓存，从 wbx 文件导入地址或导出为 wbx 文件

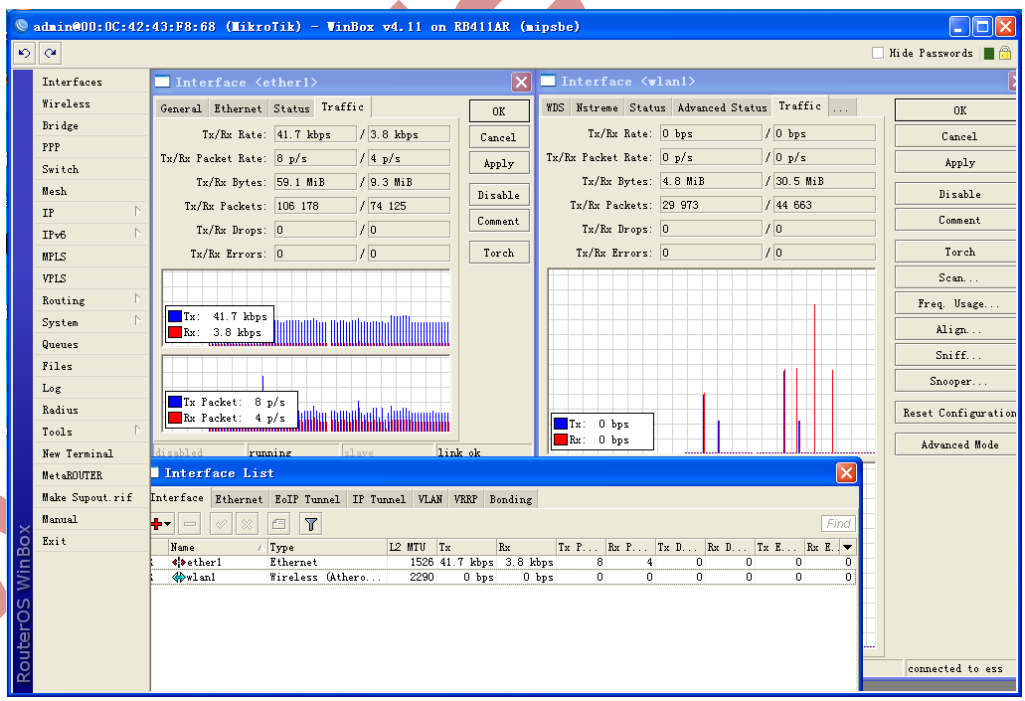


Secure mode (安全模式)：提供保密并在 Winbox 和 routerOS 之间使用 TLS(transport layer security)协议

Keep password(保存密码)保存密码到本地磁盘的文本文件中

注：在 winbox2.2.12 后增加了可选择死亡 MAC 登陆或者 IP 登陆的功能。

路由器的 winbox 控制台：



Winbox 控制台使用 TCP8291 端口，在登录到路由器后可以通过 Winbox 控制台操作 mikrotik 路由器的配置并实行与本地控制台同样的任务。

命了功能概述

下面是对 Winbox 控制台的操作建议：

图标	功能		图标	功能
	添加一条项目			定义或编辑一个注释
	删除一条存在项目			查询关键字
	启用一个项目			撤销操作
	禁用一条项目			恢复操作

故障分析

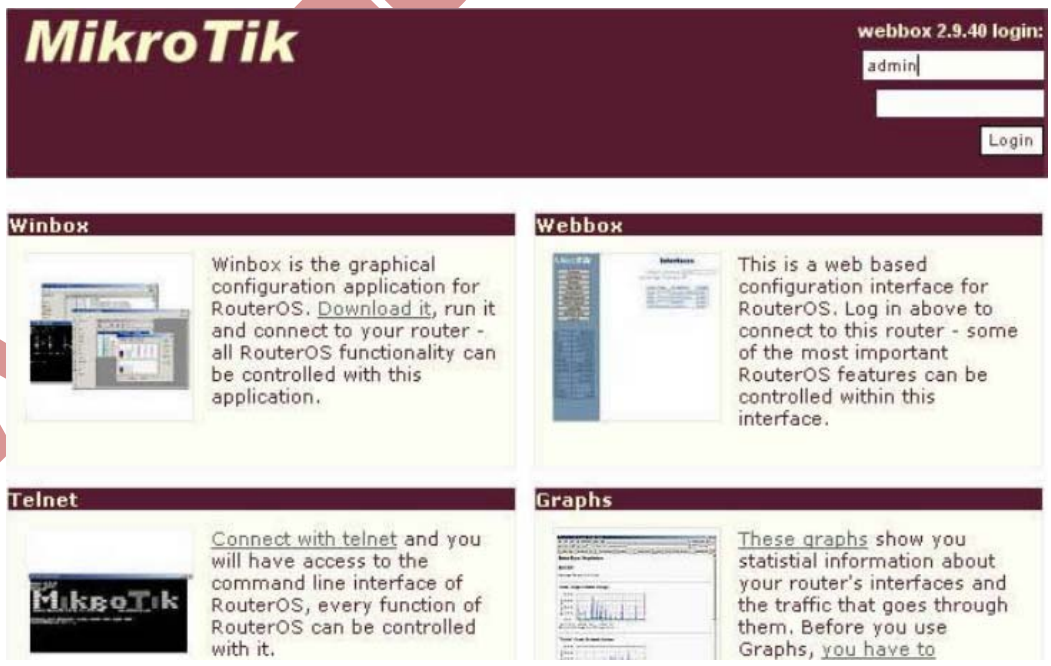
我能在 Linux 上运行 Winbox?
能，使用 wine 图形接口，可以运行 winbox 并连接到 routerOS

我不能打开 winbox 控制台

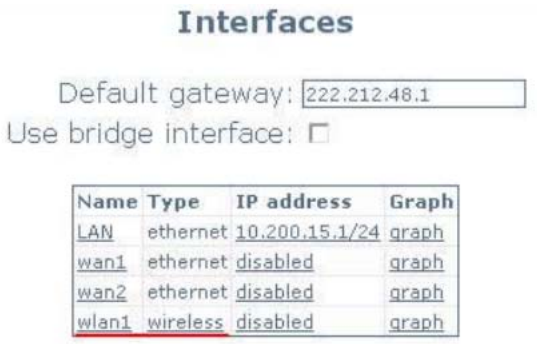
检查路由器上/ip service print 的 WWW 服务端口和地址是否正确，确定地址是你能链接到的指定网络。确定端口为指定的端口。如果你的服务端口和访问地址被修改，你可以通过下面的命令设置回默认值/ip service set www port=80 address=0.0.0.0/0. Winbox 控制台使用 TCP8291 端口在防火墙中是否做了访问限制。

Webbox 界面

当你配置好 routerOS 的 IP 地址后，通过在浏览器中输入 HTTP:// routerIP 可以访问到 routerOS 的 web 页面，我们早右上角输入 routerOS 的登陆账号和密码进入 webbox.

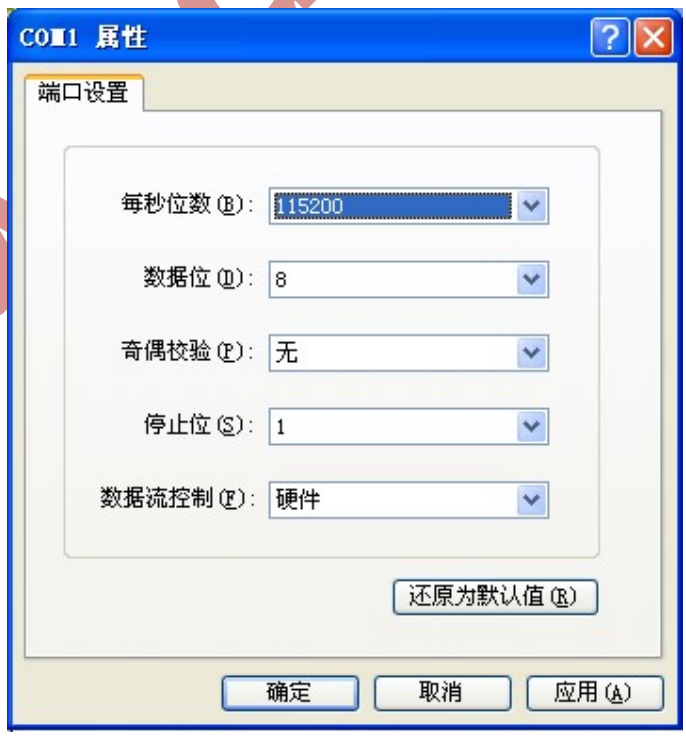


下面是 webbox 的页面，在 webbox 中只能对 routerOS 的一些基本参数配置，不能进行详细的网络配置。



串口访问

同样也能用任何 PC 通过标准的 DB9 模式串口线连接到路由器，串口连接的默认设置为每秒位数：9600 bits/s(routerBOARD 系列串口是 115211bits/s),使用终端仿真程序（如在 windows 中的超级终端或 secureCRT,UNIX/linux 得 minicom）连接到路由器，超级终端的具体参数设置如下：



在路由器启动完成后，会发出连续俩声短触“嘀嘀”的明鸣音，之后在显示屏上，出现登录的提示，如果在终端显示中，没有提示任何信息，需要检查一下网线或是串口线是否连接好。

串口控制（管理线）功能允许通过一个 mikrotik router 串行接口访问路由器的串口终端控台一个特殊的串行接口线通过工作站或者便式电脑的串口（COM）链接到路由器的串口。在 windows 电脑上常用的串口连接程序是超级终端（Hyperteminal）.

串口控制线配置

Mikrotik 定义的串口线连接串行接口（COM）,串口线为 DB9 接口，PC 和 RB532 的线序排列如下：

路由(DB9f)	Signal	(DB9f)
1, 6	CD, DSR	4
2	RxD	3
3	TxD	2
4	DTR	1, 6
5	GND	5
7	RTS	8
8	CTS	7

RB100 系列，RB400,RB300,RB600 的串口线序如下：

DB9f	Signal	DB9f	DB25f
1+4+6	CD+DTR+DSR	1+4+6	6+8+20
2	RxD	3	2
3	xD	2	3
5	GND	5	7
7+8	RTS+CTS	7+8	4+5

注：mikrotik routerOS 需要定义以上的串口线序，才可以正常通信。

当登录到终端控制台后，会出现 routerOS 的登录提示，第一次登录的时候用户为“admin”密码为空，直接敲回车键进入，如下面得所示：

```
MikroTik v3.0
Login: admin
Password:
```

修改密码可以使用/password 命令

```
[admin@MikroTik] > password
old password:
new password: ************
retype new password: ************
[admin@MikroTik] >
```

MAC 层访问（telnet 与 winbox）

通过 MAC 地址进行链接是用来访问没有设置 IP 地址的 routerOS 路由设备，这种连接类似于 IP 地址连接，通过 MAC 地址仅在限于 2 台 mikrotik routerOS 路由器之间进行。

操作路径：/tool mac-server

属性描述

Interface(name/all;默认:all)-连接 MAC 服务器客户端的接口名
All-所有接口
注：这是一个在菜单单选项的接口列表，如果你添加一些接口进列表，你就能充许通过 MAC 地址连接这些接口。Disabled(disabled=yes)状态的意思是不充许在接口列表中添加的接口通过 mac 地址进行访问。All interfaces 默认设置为充许任何接口进行 mac 地址运程访问。

使只有 ether1 interface 能通过 mac 运程访问服务器：

```
[admin@MikroTik] tool mac-server> print
Flags: X - disabled
 #   INTERFACE
 0   all
[admin@MikroTik] tool mac-server> remove 0
[admin@MikroTik] tool mac-server> add interface=ether1 disabled=no
[admin@MikroTik] tool mac-server> print
Flags: X - disabled
 #   INTERFACE
 0   ether1
[admin@MikroTik] tool mac-server>
```

MAC winbox server

操作路径：tool mac-server mac-winbox

属性描述

Interface(name/all;默认：all)-充许使用 MAC 地址的协议连接的接口名

ALL -所有接口

注：这是一个菜单选项的接口列表，如果你添加接口在列表中，即允许通过 MAC 地址访问到这个接口，disabled(disabled=yes)意思是在这些接口中是不允许使用 MAC 地址连接的接口。

仅启用 ether1 接口的 MAC 服务器

```
 [admin@MikroTik] tool mac-server mac-winbox> print
Flags: X - disabled
 #   INTERFACE
 0   all
[admin@MikroTik] tool mac-server mac-winbox> remove 0
[admin@MikroTik] tool mac-server mac-winbox> add interface=ether1 disabled=no
[admin@MikroTik] tool mac-server mac-winbox> print
Flags: X - disabled
 #   INTERFACE
 0   ether1
[admin@MikroTik] tool mac-server mac-winbox>
```

动态监控列表

操作路径：tool mac-server sessions

属性描述

Interface(只读：name)连接客户端的接口
Src-address(只读：MAC address)客户 MAC 地址（源地址）
Uptime(只读：time)客户端连接到服务器上的时间

查看 MAC 地址链接访问

```
[admin@MikroTik] tool mac-server sessions> print
 # INTERFACE SRC-ADDRESS      UPTIME
 0 wlan1     00:0B:6B:31:08:22 00:03:01
[admin@MikroTik] tool mac-server sessions>
```

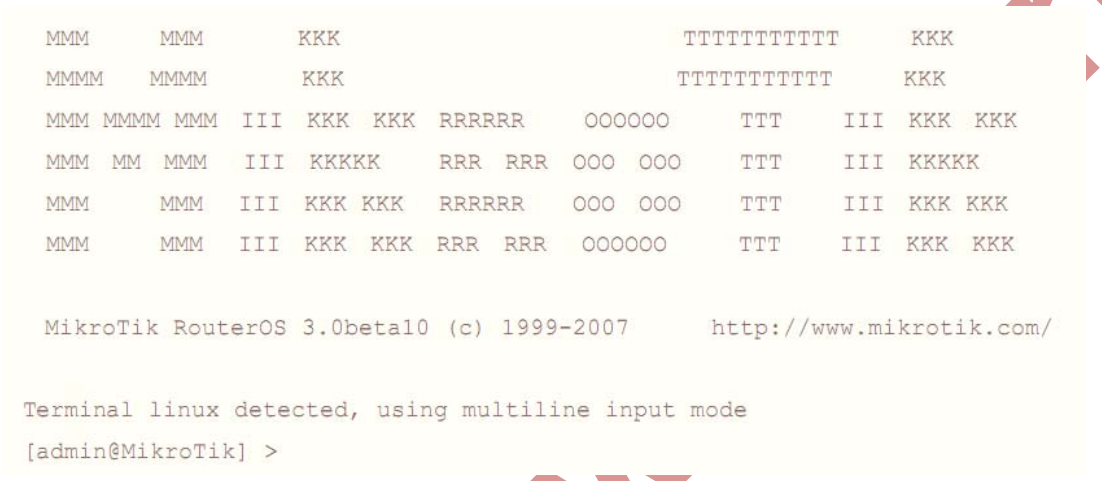
MAC telnet 访问客户端

操作路径：/tool mac-telnet

（MAC Address）兼容设备的 MAC 地址

通过 MAC 地址登陆远程的 routerOS:

[dmin@mikrotik] tool mac-telnet 00:02:6F:06:59:42
Login:admini
Password:
Trying 00:02:6F:06:59:42



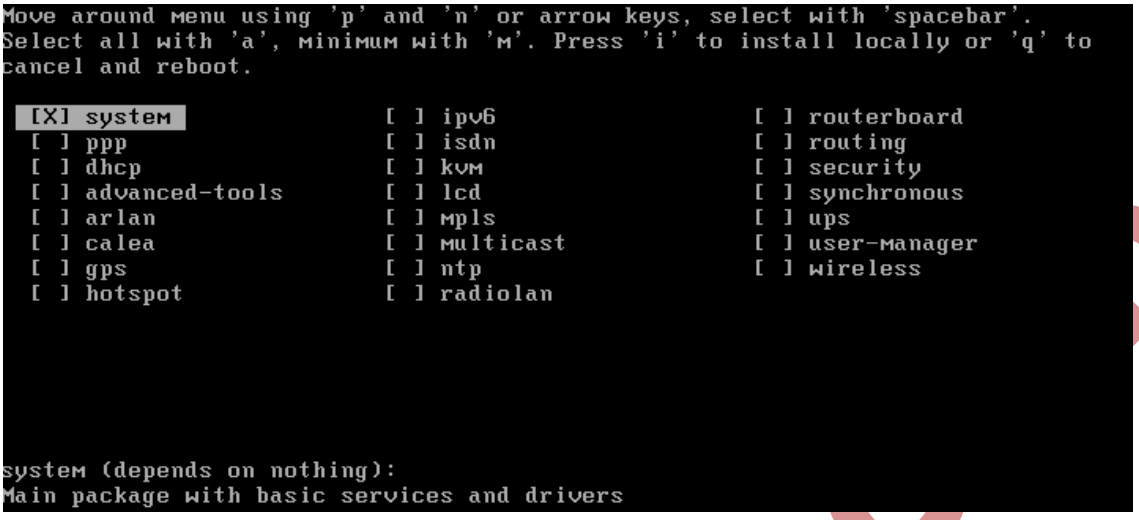
ROUTEROS 安装介绍

1. 使用带 ISO 的镜像文件通过光盘引导安装（用于 X86 系统）；支持 AMD,INTEL,VIA 以及其他 X86 系统，硬盘支持 IDE,SATA 硬盘接口
2. 使用 U 盘安装基于 X86（限 63.版本）
3. 使用 netinstall 网络安装程序（主要用于 RouterBOARD）；RB100、RB300、RB400、RB500、RB600、RB1000 系列

光盘安装

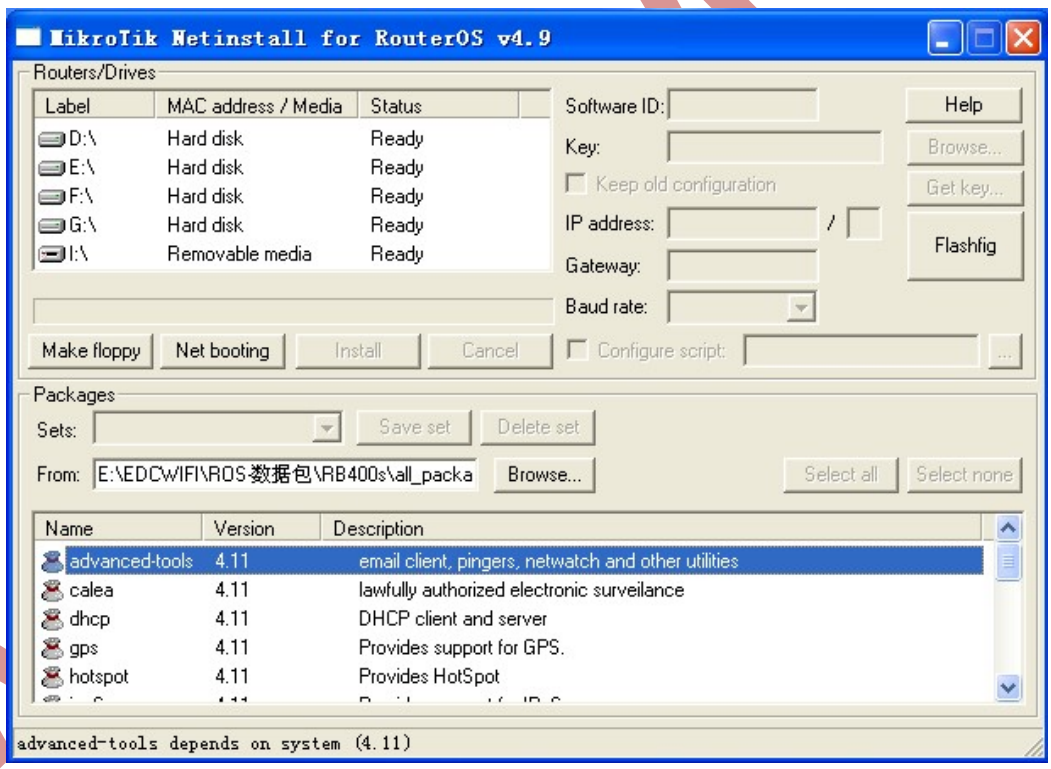
BIOS 设置光盘引导

自动进入安装页面，选择需要使用的功能包（system 包是必须默认安装）

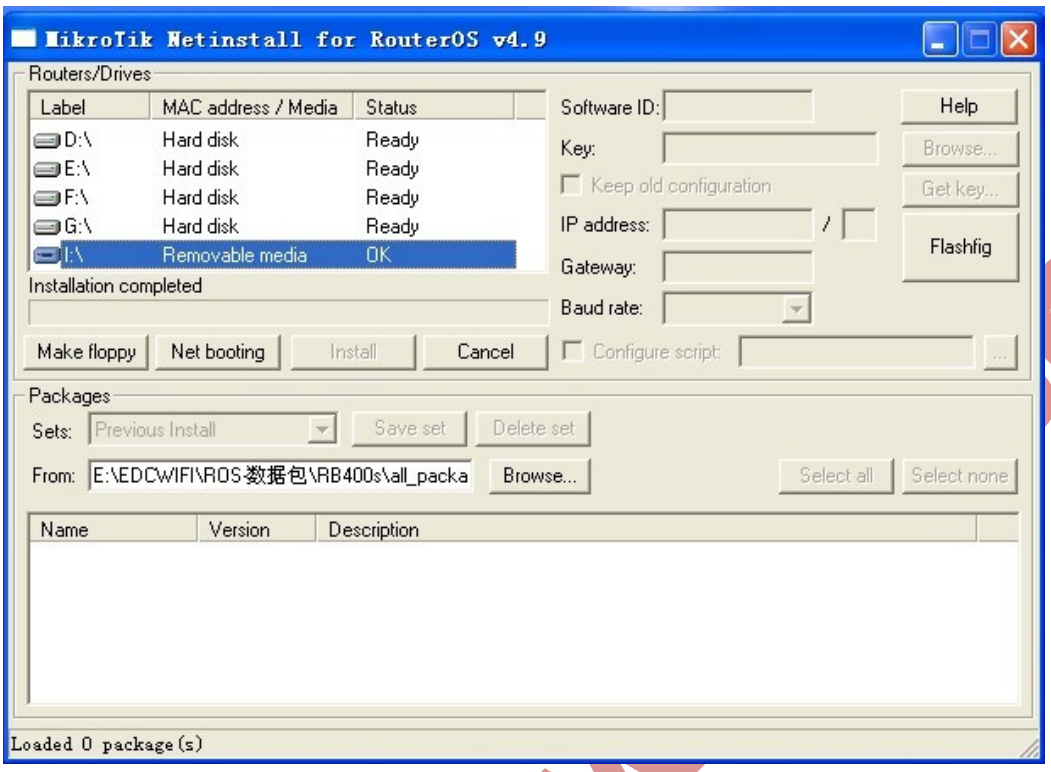


USB 安装

U 盘安装需要使用 3.0 以上版本 netinstall 软件，将 U 盘插入一台 windows 电脑的 USB 接口后，启动 netinstall 软件，选择 routerOS-x86 安装包：



通过 netintall 安装 routerOS 到 U 盘上：



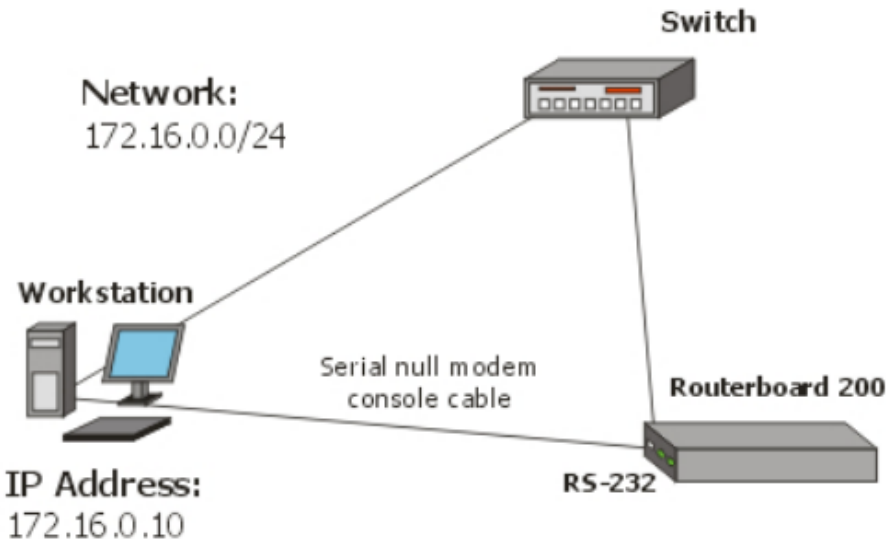
最后取出 U 盘，插入需要使用 U 盘的 PC 上，并设置 PC 通过 USB 引导启动，启动后可以看到系统正在安装。

Netintall 安装和复位 routerRoard

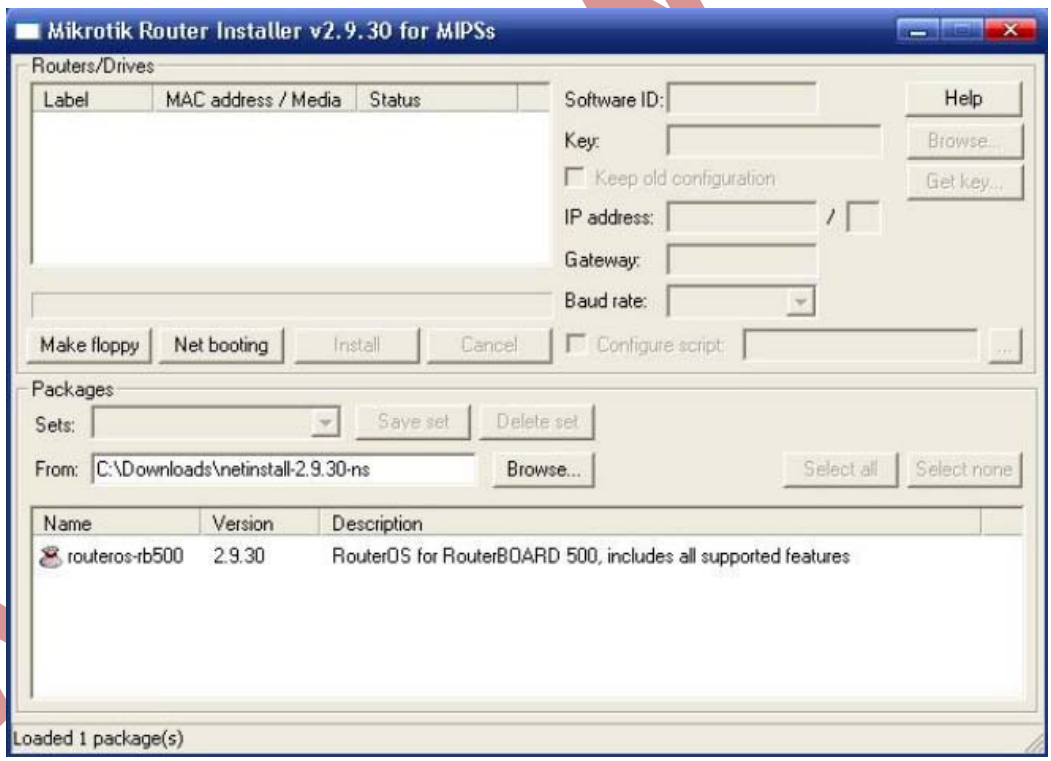
网络安装和复位 routerRoard

这个事列将介绍如何一步一步在一个 routerBoard 上安装软件，同样在你失去了 routerBoard 登录密码后，也可以通过该复位 RouterOS.

1. 使用 ether1 网卡通过交换机或者直接通过网线连接到 routerBoard 上，然后在使用串口线和 routerboare 相联接。

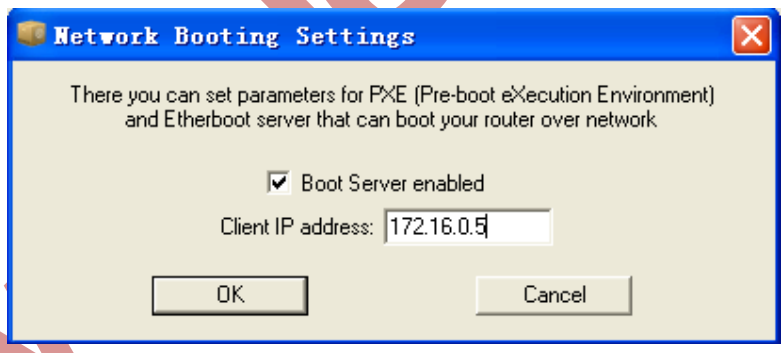


2. 在你的电脑上运行 netinstall for MIPS 程序，确定软件包（*.npk 文件）在你本地磁盘上。
   Netinstall for MIPS:



3. 设置好 windows 工作站的超级终端连接，每秒位数为 115200，其他参数为系统默认值：

4. 输入 boot server 客户端的 IP 地址。设置一个 IP 地址段，用于临时分配给 routerBoard 的 IP 地址（该事例的地址为 172.16.0.0/24）
注意：网线连接的是 routerBoard 的 ether1 网卡接口，不然无法获取引导信息。



4. 设置 routerBoard 从以太网卡引导，首先进入 routerBoard BIOS(重起 routerBoard)后，在超级终端下出现提示时 press any key…后按任意键进入 bios 设置）

RouterBoard 532
CPU frequency:330 MHz
Memory size:32MB

Press any key within 2 seconds to enter setup

RouterBOOT-1.13
What do you want to configure?

[键入文字]

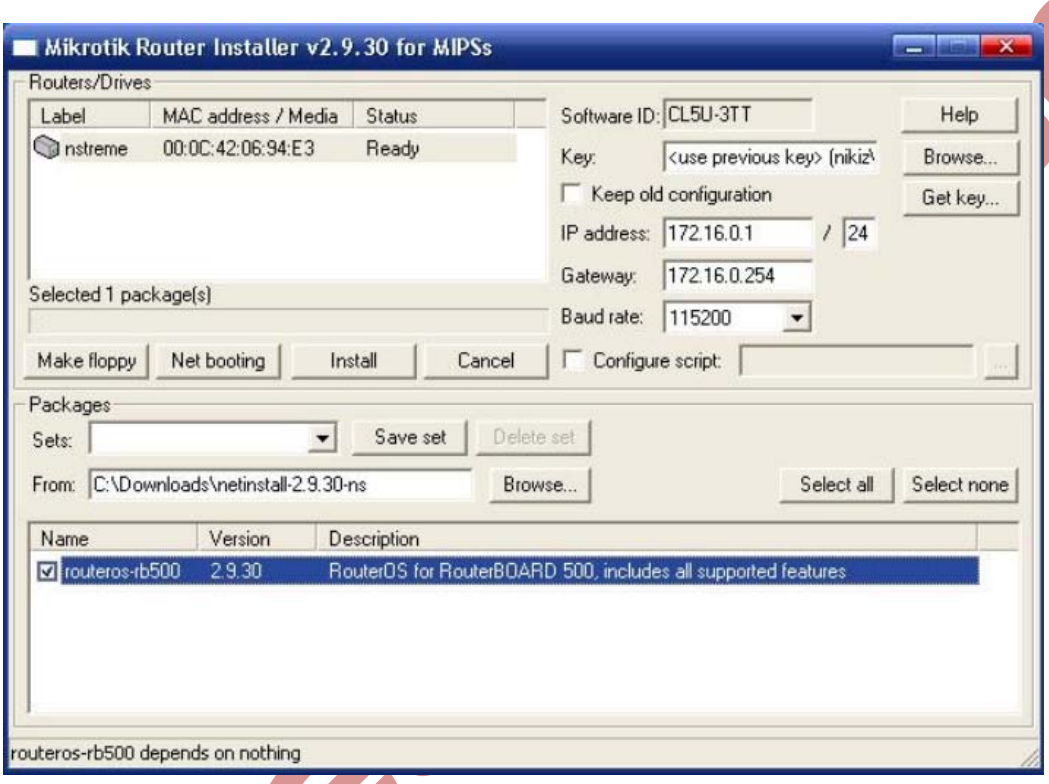
d-boot delay
k-boot key
s-serial console
o-boot device
u-cpu mode
f-try cpu frequency
c-keep cpu frequency
r-reset configuration
e-format nand
g-upgrade firmware
i-board info
p-boot protocol
t-do memory testing
  x-exit setup
your choice

进入 BIOS 后你可以看到可用命令的列表，设置引导设备，选择“boot device”,按”0“键可以进入
Routerboot-1.13
What do you want to configure?
d-boot delay
k-boot key
s-serial console
o-boot device
u-cpu mode
f-try cpu frequency
c-keep cpu frequency
r-reset configuration
e-format nand
g-upgrade firmware
i-board info
p-boot protocol
t-do memory testing
x-exit setup
your choice:o-boot device

按“e”键是选择从以太网卡引导 routerBoard:

Select boot device
e-boot over ethernet
n-boot from NAND,if fail then ethernet
c-boot from CF
1-boot ethernet once,then NAND
2-boot ethernet once,then cf

o-boot from NAND only
b-boot chosen device
your choice: e- etherboot
当选择完成后，返回 routerboard BIOS 首页，选择“X”，退出 BIOS。路由器将会重启
6.在启动时 routerboard 将试着从以太网卡上去引导信息。如果成功，运行 netinstall 的 windows 工作站。
在 windows 工作站，将会出现一个新的路由器列表，显示当前连接的 routerboard 设备。



连接完成后，需要选择安装的功能包或文件的路径，是否保留原来的配置，设置给路由器新的 IP 地址和网关。还有就是传输的波特率选择对应的 115200.
当完成设置后，就可以按 install 键开始安装 routerOS。
7.当安装工作完成，在安装程序中按“reboot”或在超级终端敲击“回车”，路由器将重启。记住设置完成后回到 routerboard BIOS 中设置为 boot from NAND only(仅从 routerboard 的存引导)。这样完成后，就能正常启动 routerOS。

CLI(COMMAND line interface)命令行操作

命令提示显示路由器的身份名称和当前的操作路径，如下：
[admin@mikrotik]
[admin@mikrotik] interface>ip address
[admin@mikrotik] ip address

命令

在任何操作目录使用‘？’都可用获在当前目录中的命令信息。

[admin@mikrotik]

Log/—系统日志
Quit—退出控制台
Radius/—radius-客户端设置
Certificate/—授权管理
Special-login/—特殊登录用户
Redo—返回以前的操作
Driver/—驱动管理
Ping-ping 命令
Setup—做基本的系统设置
Interface/—接口配置
Password—修改密码
Undo—辙销以前操作
Port/—串口控制
Impor—运行导入的配置脚本
Snmp/—SNMP 设置
User/—用户管理
File/—路由器本地文件存储
System/—系统信息和应用程序
Queue-/—带宽管理
Ip/—ip 选项
Tools/—工具
Ppp/)—点对点协议
Routing—各种路由协议设置
Export — 导出脚本

[admin@mikrotik]>
[admin@mikrotik] ip

•• —回到根目录
Service/—ip 服务
Socks/—socks4 代理
Arp/—ARP 项目管理
Upnp/—UPNP 管理
Dns/—DNS 设置
Address/—地址管理
Accounting—传输记录
The-proxy/—
Vrry/—虚拟路由协议
Pool/—ip 地址池
Packing/—数据包封装设置

Neighbor/--邻居
Route/—路由管理
Friewall/—防火墙管理
DHCP-client—DHCP 客户端设置
DHCP-serner/—DHCP 服务设置
Hotspot/—hotspot 管理
Ipsec/—ip 安全设置
Web-proxy/—http 代理
Export—

[admin@mikrotik]ip

上面是对可用命令和目录的简短描述，在下面的例子中。你可用通过输入目录名称移动到不同的目录中去

```
[admin@MikroTik] >                  | 根目录
[admin@MikroTik] > driver           | 输入'driver'进入到驱动管理目录中
[admin@MikroTik] driver> /          | 输入'/'从任何目录中回到根目录
[admin@MikroTik] > interface        | 输入'interface'进入接口管理目录中
[admin@MikroTik] interface> /ip     | 输入'/ip'从任何目录进入 IP 管理目录
[admin@MikroTik] ip>                |
```

一个指令或一个变量参数不需要完整的输入，如是含糊不清的指令或变量参数需要完整的输入。入输入 interface 时，你只要输入 in 或 int，需要显示完整的指令可以使用[tab]键

通过指令的组合，可以在当前的目录执行在不同目录操作 如：

```
[admin@MikroTik] ip route> print                  打印路由表
[admin@MikroTik] ip route> .. address print       打印 IP 地址列表
[admin@MikroTik] ip route> /ip address print      打印 IP 地址列表
```

指令执行概述

Command	指令
Command[Enter]	执行指令
[?]	显示该目录中的所有指令列表
Command[?]	显示指令的帮助和变量列表
Command argument[?]	显示指令的变量帮助
[Tab]	使指令/字段完整，如果输入的内容含糊不清，第二次键入 **[Tab]**就会给出存在的选项
/	移动到根目录
/command	执行根目录中的指令
..	移动到上一级目录
“”	指定一个空字符串

在配置 IP 地址中，配置 address 和 netmask 参数时，在许多事例中你可以将 IP 地址和子网掩码一起定义，也可以将子网掩码单独定义，这俩种方式是相同的，例如下面的俩个输入时等价的：.

```
/ip address add address 10.0.0.1/24 interface ether1
/ip address add address 10.0.0.1 netmask 255.255.255.0 interface ether1
```

基本操作命令

接口管理（interface management）

在配置 ip 地址和路由、前，如果你即插即用卡安装到路由器中，请检查 interface 中的接口列表，多数情况下设备驱动会自动安装，并相关的接口信息会显示在 interface print 列表中。例如

```
[admin@MikroTik] interface> print
Flags: X - disabled, D - dynamic, R - running
 #    NAME                TYPE           RX-RATE    TX-RATE    MTU
 0  R ether1               ether          0          0          1500
 1  R ether2               ether          0          0          1500
 2 X  wavelan1             wavelan        0          0          1500
 3 X  prism1               wlan           0          0          1500
[admin@MikroTik] interface>
```

如果你想使用这些设备，一般都需要启用，使用 interface enable name 指令给出接口名称或标号启用，例如：

```
[admin@MikroTik] interface> print
Flags: X - disabled, D - dynamic, R - running
 #    NAME                TYPE           RX-RATE    TX-RATE    MTU
 0 X  ether1              ether          0          0          1500
 1 X  ether2              ether          0          0          1500
[admin@MikroTik] interface> enable 0
[admin@MikroTik] interface> enable ether2
[admin@MikroTik] interface> print
Flags: X - disabled, D - dynamic, R - running
 #    NAME                TYPE           RX-RATE    TX-RATE    MTU
 0  R ether1              ether          0          0          1500
 1  R ether2              ether          0          0          1500
[admin@MikroTik] interface>
```

接口的名称能通过 interface set 指令来改变其描述：

```
[admin@MikroTik] interface> set ether1 name=Local; set ether2 name=Public
[admin@MikroTik] interface> print
Flags: X - disabled, D - dynamic, R - running
 #    NAME                      TYPE           RX-RATE    TX-RATE    MTU
 0  R Local                     ether          0          0          1500
 1  R Public                    ether          0          0          1500
[admin@MikroTik] interface>
```

通过 add 命令添加规则，如添加 ip 地址操作：

```
[admin@office] /ip address> prin
Flags: x - disabled,    I - invalid , D - dynamic
#         ADDRESS             NETWORK         BROADCAST       INTERFACE
0         10.200.15.1/24      10.200.15.0     10.200.15.255   lan
1    D    222.212.60.227.32   222.212.48.1    0.0.0.0         ADSL
[admin@office] /ip address> add address=192.168.10.1/24    interface=lan

[admin@office] /ip address> prin
Flags: x - disabled,    I - invalid , D - dynamic
#         ADDRESS             NETWORK         BROADCAST       INTERFACE
0         10.200.15.1/24      10.200.15.0     10.200.15.255   lan
1    D    222.212.60.227.32   222.212.48.1    0.0.0.0         ADSL
2         192.168.10.1/24     192.168.10.0    192.168.10.255  lan
[admin@office] /ip address>
```

通过 remove 命令删除怒需要规则

```
[admin@office] /ip firewall filter> prin
Flags: x - disabled,    I - invalid , D – dynamic

0 x chain=forward action=drop    layer7-protocol=qq

1 x chain= forward action=drop dst-address-list=qq

2 x chain= forward action=log log-prefix=””
[admin@office] /ip firewall filter> remove 2
[admin@office] /ip firewall filter> prinlags
Flags: X – disabled, I – invalid, D - dynamic
0 x chain=forward action=drop    layer7-protocol=qq

1 x chain= forward action=drop dst-address-list=qq
[admin@office] /ip firewall filter>
```

Setup 命令

当初始化路由器时，通过使用 setup 指令设置下列配置内容：

重新设置路由器配置
载入接口驱动
配置 ip 地址和网关
设置 DHCP 客户端
设置 pppoe 客户端
设置 pptp 客户端

使用 setup 指令，在路由器上配置 ip 地址，执行 setup 指令行：

```
[admin@MikroTik] > setup
 Setup uses Safe Mode. It means that all changes that are made during setup
are reverted in case of error, or if Ctrl-C is used to abort setup. To keep
changes exit setup using the 'x' key.
[Safe Mode taken]
 Choose options by pressing one of the letters in the left column, before
dash. Pressing 'x' will exit current menu, pressing Enter key will select the
entry that is marked by an '*'. You can abort setup at any time by pressing
Ctrl-C.
Entries marked by '+' are already configured.
Entries marked by '-' cannot be used yet.
Entries marked by 'X' cannot be used without installing additional packages.
   r - reset all router configuration
 + l - load interface driver
 * a - configure ip address and gateway
   d - setup dhcp client
   s - setup dhcp server
   p - setup pppoe client
   t - setup pptp client
   x - exit menu
your choice [press Enter to configure ip address and gateway]: a
```

配置 ip 地址和网关，输入 a 或[Enter]

```
 * a - add ip address
 - g - setup default gateway
   x - exit menu
 your choice [press Enter to add ip address]: a
```

选择 a 添加一个 ip 地址，首先，设置程序将要询问你选择哪一个接口添加 ip 地址，如果设置程序没有指定出，合适的接口，可以通过键入[Tab]俩次，查看可选的接口，在接口选择后，分配 ip 地址和子网掩码：

```
your choice: a
enable interface:
ether1  ether2  wlan1
enable interface: ether1
ip address/netmask: 10.1.0.66/24
#Enabling interface
/interface enable ether1
#Adding IP address
/ip address add address=10.1.0.66/24 interface=ether1 comment="added by setup"
 + a - add ip address
 * g - setup default gateway
   x - exit menu
your choice: x
```

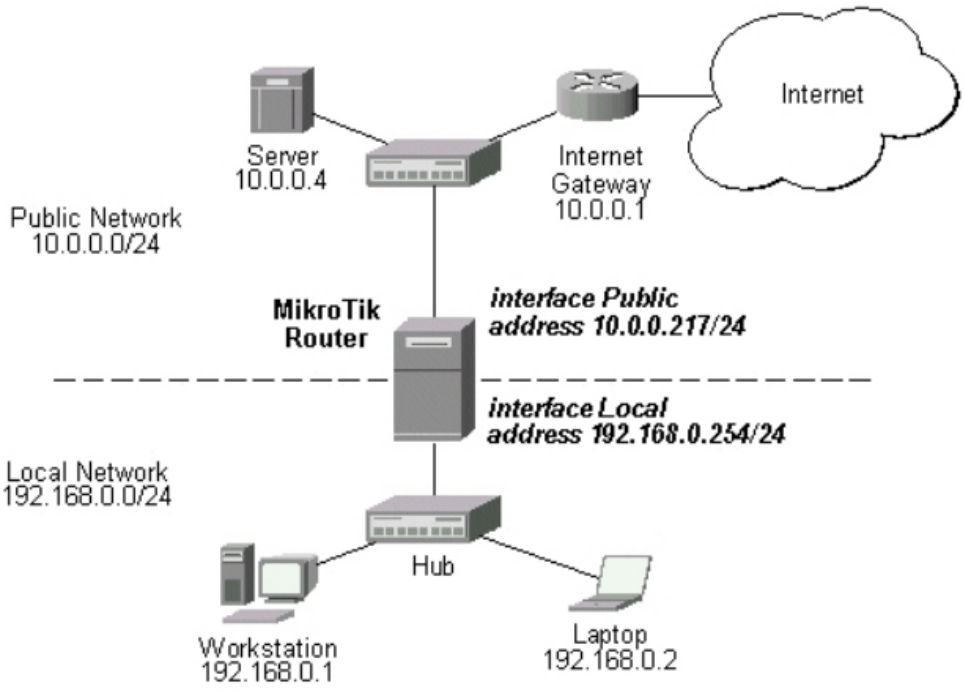
RouterOS 简单网络配置事例

配置一个 routerOS 分为三个步骤：

```
配置一个 RouteroS 分为三个步骤：
第一步：检查 Interface 上的网卡是否正确安装，然后在 ip  address 中配置 IP 地址；
第二步：在配置好 IP 地址后，在 ip  routes 中配置默认网关，配置完后检查是否到外网默认网关正常；
第三步：在 ip  firewall  nat 中配置 NAT 伪装，隐藏内部网络。
```

例如：

假如你需要同 mikrotik router 配置下面的网络：



在当前的事例中我们使用到俩个网络（公网和本地网络）：

本地网络使用地址为：192.168.0.0 子网掩码 24-bit(255.255.255.0)路由器的地址在这个网络中为：192.168.0.254

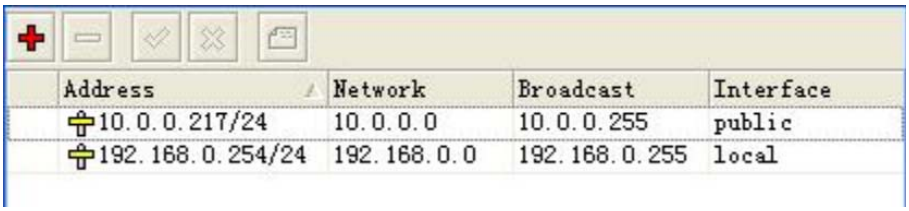
ISP 的网络为 10.0.0.0 子网掩码 24-bit（255.255.255.0）路由器的地址是在网络中为 10.0.0.217

通过下面的指令添加地址：

```
[admin@MikroTik] ip address> add address 10.0.0.217/24 interface Public
[admin@MikroTik] ip address> add address 192.168.0.254/24 interface Local
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS          NETWORK        BROADCAST      INTERFACE
  0   10.0.0.217/24     10.0.0.217     10.0.0.255     Public
  1   192.168.0.254/24   192.168.0.0    192.168.0.255   Local
[admin@MikroTik] ip address>
```

这里，子网掩码在 address 变量中指定，或者也可以通过在 natmask 变量中设置 255.255.255.0. 网段和广播地址在输入时没有指定，这些可以由 routerOS 自动计算出来

请注意：在 ip 地址被分配到路由器的不同网卡上时，营属于不同的网络。下面是 winbox 中的设置情况：



Address	Network	Broadcast	Interface
10.0.0.217/24	10.0.0.0	10.0.0.255	public
192.168.0.254/24	192.168.0.0	192.168.0.255	local

配置路由器

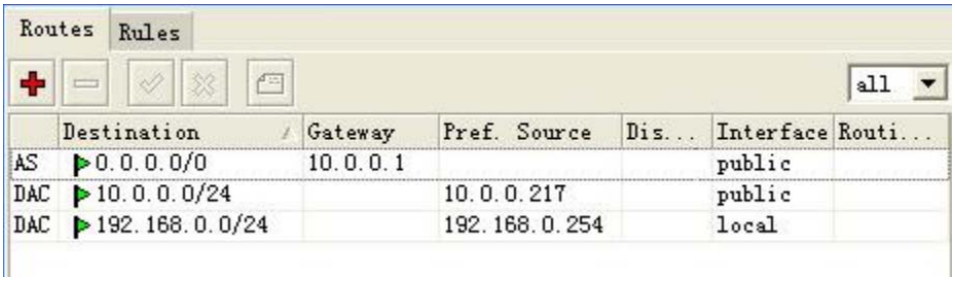
你可以看到俩个带有动态 dynamic(D)和连接 connected（C）的路由、当地址添加后会在路由中自动添加动态路由：

```
[admin@MikroTik] ip route> print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS       G GATEWAY        DISTANCE INTERFACE
    0 DC 192.168.0.0/24     r 0.0.0.0         0       Local
    1 DC 10.0.0.0/24        r 0.0.0.0         0       Public
[admin@MikroTik] ip route> print detail
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    0 DC dst-address=192.168.0.0/24 preferred-source=192.168.0.254
        gateway=0.0.0.0 gateway-state=reachable distance=0 interface=Local
    1 DC dst-address=10.0.0.0/24 preferred-source=10.0.0.217 gateway=0.0.0.0
        gateway-state=reachable distance=0 interface=Public
[admin@MikroTik] ip route>
```

添加默认路由

```
[admin@MikroTik] ip route> add gateway=10.0.0.1
[admin@MikroTik] ip route> print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS       G GATEWAY        DISTANCE INTERFACE
    0  S 0.0.0.0/0         r 10.0.0.1        1       Public
    1 DC 192.168.0.0/24     r 0.0.0.0         0       Local
    2 DC 10.0.0.0/24        r 0.0.0.0         0       Public
[admin@MikroTik] ip route>
```

这里。默认路由被例如标号#0.同样我们看到，网关 10.0.0.1，能在接口 public 通过。如果网关没有被正确的指定，interface 变量值将会无法确定（unknown）。Winbox 添加后情况如下：



测试网络连接

从现在起，ping 指令可以用来测试网络连接情况。

```
 [admin@MikroTik] ip route> /ping 10.0.0.4
10.0.0.4 64 byte ping: ttl=255 time=7 ms
10.0.0.4 64 byte ping: ttl=255 time=5 ms
10.0.0.4 64 byte ping: ttl=255 time=5 ms
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 5/5.6/7 ms
[admin@MikroTik] ip route>
[admin@MikroTik] ip route> /ping 192.168.0.1
192.168.0.1 64 byte ping: ttl=255 time=1 ms
192.168.0.1 64 byte ping: ttl=255 time=1 ms
192.168.0.1 64 byte ping: ttl=255 time=1 ms
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 1/1.0/1 ms
[admin@MikroTik] ip route>
```

如果路由器的地址 192.168.0.254 在 windows 工作站的 TCP/IP 协议中配置为默认网关这是你就能 PING 通路由器

```
C:\>ping 192.168.0.254

Reply from 192.168.0.254: bytes=32 time=10ms TTL=253
Reply from 192.168.0.254: bytes=32 time<10ms TTL=253
Reply from 192.168.0.254: bytes=32 time<10ms TTL=253
C:\>ping 10.0.0.217
Reply from 10.0.0.217: bytes=32 time=10ms TTL=253
Reply from 10.0.0.217: bytes=32 time<10ms TTL=253
Reply from 10.0.0.217: bytes=32 time<10ms TTL=253
C:\>ping 10.0.0.4
Request timed out.
Request timed out.
Request timed out.
```

注：你不能访问超过路由器的任何网络（10.0.0.0/24 的网络和 Internet）。你需要做下面的设置：

使用源地址翻译（masquerading）,通过 mikrotik 路由隐藏你的私有网络 192.168.0.0/24（查看下面的信息）

在 isp 的网关 10.0.0.1 上添加静态路由指明到目的地址 192.168.0.0/24 通过 10.0.0.217 的主机，这时所有在 isp 上的网络主机。包括服务器，将能连接你的私有网络

在设置路由时，你需要了解一些配置 TCP/IP 的网络知识，当你遇到配置网络安装困难时，我们建议不获得更多的网络知识。

下面我将讨论隐藏'hiding'私有的 LAN192.168.0.0/24 在 ISP 给的 10.0.0.217 的背后。

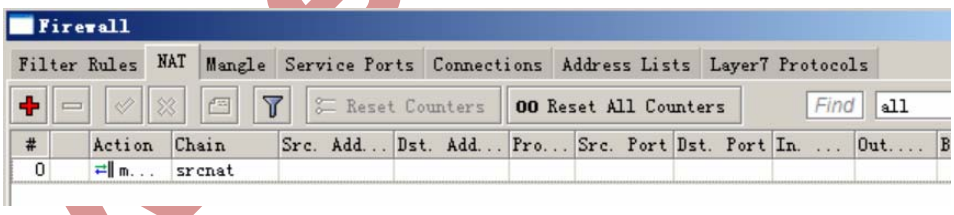
伪装的应用事例 (Masquerading)

如果你想隐藏'hiding'私有的 LAN192.168.0.0/24 在 ISP 给的 10.0.0.217 的背后，你需要使用 RouterOS 的源地址翻译。伪装将改变源 ip 地址和数据包端口，即将 192.168.0.0/24 改为 10.0.0.217 去回应 isp 的网络。

使用伪装时添加一条 NAT 规则在防火墙配置中，，执行 masquerade 如下面：

```
[admin@MikroTik] ip firewall nat> add chain=srcnat action=masquerade out-interface=Public
[admin@MikroTik] ip firewall nat> print
Flags: X - disabled, I - invalid, D - dynamic
 0   chain=srcnat out-interface=Public action=masquerade
```

Winbox 添加后如下：



注：如果需要了解很多的 NAT 信息，建议参阅 NAT 说明文档。

以上是如何配置一个简单 ROUTEROS 网络的应用实例。

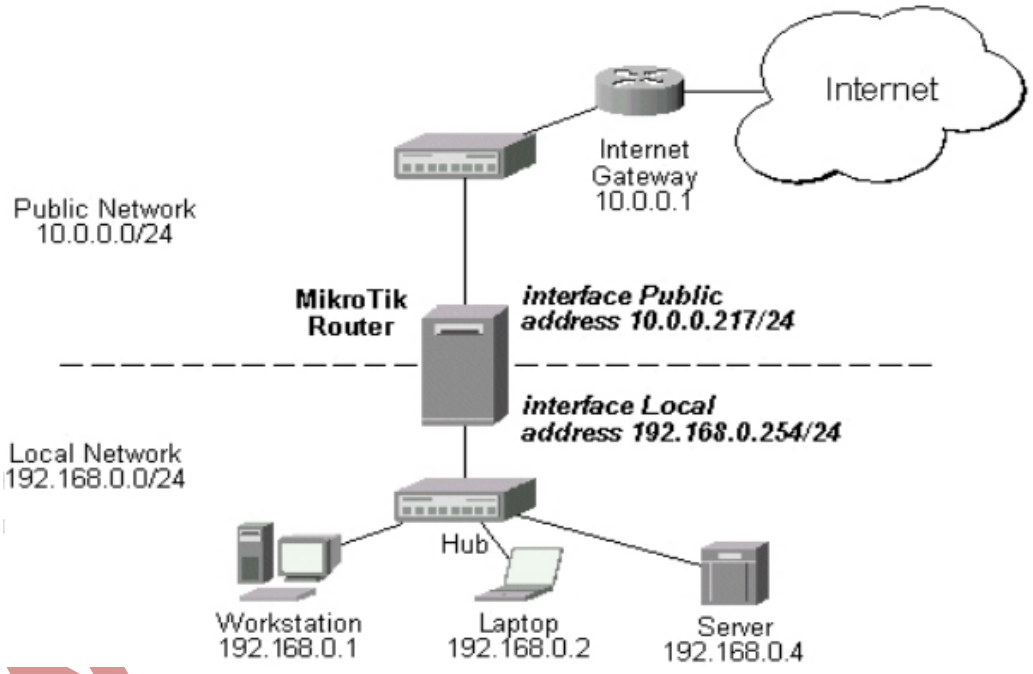
带宽管理事例

假如你想要限制所有的 LAN 内主机 192.168.0.88 的下行带宽为 128kbps 和上行带宽为 64kbps:

```
[admin@MikroTik] queue simple> add target-address=192.68.0.88 max-limit=64000/128000
interface=Local
[admin@MikroTik] queue simple> print
Flags: X - disabled, I - invalid, D - dynamic
 0   name="queue1" target-address=192.68.0.88 dst-address=0.0.0.0/0
     interface=Local queue=default priority=8 limit-at=0/0
     max-limit=64000/128000
[admin@MikroTik] queue simple>
```

NAT 端口映射事例

假如我们将以前的服务器从公网移动到私网中去，并想让其他的公网来访问我们的服务器，这时就需要用到 NAT：



现在服务器的地址是 192.168.0.4，我们在服务器上运行 80 端口监听 WEB 服务，我们想通过公网地址 10.0.0.217，80 端口访问该服务器，即我们就需要在 mikrotik 路由器上作静态的网络地址翻译（NAT），这样通过公网地址 10.0.0.217 端口 80 将数据传输到本地网络的 192.168.0.4：80，在目标地址和端口：

```
[admin@MikroTik] ip firewall nat> add chain=dstnat action=dst-nat protocol=tcp
dst-address=10.0.0.217/32 dst-port=80 to-addresses=192.168.0.4
[admin@MikroTik] ip firewall nat> pr
Flags: X - disabled, I - invalid, D - dynamic
 0   chain=dstnat dst-address=10.0.0.217/32 protocol=tcp dst-port=80
     action=dst-nat to-addresses=192.168.0.4 to-ports=0-65535
```

第二章：系统管理

RouterOS 备份与复位管理

RouterOS 备份，配置还原和复位如下

系统备份
系统通过备份文件还原
导入配置
导出配置
系统复位

mikrotik RouterOS 将配置备份为二进制文件，通过 FTP 访问或通过在 winbox 中的 file 中下载备份文件，并可以通过备份文件还原路由器设置。

Mikrotik RouterOS 通过导入配置文件，可生成文件（可编辑脚本），同样使用 FTP 或通过在 winbox 中的 file 中下载文件，导入配置则将脚本文本文件导入路由器。

系统复位是将所有的配置信息从 routerOS 中全部删除，在做此操作前 ，最好先将路由器的配置备份一次。

注：为了保证备份不会失败，请在讲备份的文件恢复到同样的软件版本和同样的硬件配置上去

系统备份

操纵路径：/system backup

Save 指令是保存当前配置到一个备份文件中，显示文件在 file 目录中，在 system reset 复位系统后，上传备份文件到 routerOS 中，并通过 system backup 中的 load 指令载入配置在还原系统配置

指令描述

Load　name={filename}-载入备份文件的配置
Save　name={filename}保存当前的配置到文件中

例如：

将当前的配置保存到文件 test：

```
[admin@MikroTik] system backup> save name=test
Saving system configuration
Configuration backup saved
[admin@MikroTik] system backup>
```

在路由器中查看保存的文件：

```
[admin@MikroTik] > file print
 # NAME                       TYPE        SIZE      CREATION-TIME
 0 test.backup                 backup      12567     aug/12/2002 21:07:50
[admin@MikroTik] >
```

导入备份文件 test:

```
[admin@MikroTik] system backup> load name=test
Restore and reboot? [y/N]: y
...
```

导出指令（EXPORT）

指令名称：export

Export 指令用于导出脚本配置信息，这个命令可以在任何目录被激活。Export 同样也可以通过 file 生成脚本配置文件，可用 FTP 下载下来。

指令描述

From=[number] - 指定需要导出的项目编号
File=[filename] - 保存的文件名称

例如：

```
[admin@MikroTik] > ip address print
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS          NETWORK        BROADCAST      INTERFACE
 0   10.1.0.172/24     10.1.0.0       10.1.0.255     bridge1
 1   10.5.1.1/24       10.5.1.0       10.5.1.255     ether1
[admin@MikroTik] >
```

制作一个导入文件：

```
[admin@MikroTik] ip address> export file=address
[admin@MikroTik] ip address>
```

制作一个仅一个项目的导出文件：

```
[admin@MikroTik] ip address> export file=address1 from=1
[admin@MikroTik] ip address>
```

在路由器中查看导出的文件

```
[admin@MikroTik] > file print
 # NAME                          TYPE        SIZE      CREATION-TIME
0  address.rsc                    script       315       dec/23/2003 13:21:48
1  address1.rsc                   script       201       dec/23/2003 13:22:57
[admin@MikroTik] >
```

在不建导出文件名，使用同样的指令导出显示出配置内容：

```
[admin@MikroTik] ip address> export from=0,1
# dec/23/2003 13:25:30 by RouterOS 2.8beta12
# software id = MGJ4-MAN
#
/ ip address
add address=10.1.0.172/24 network=10.1.0.0 broadcast=10.1.0.255 \
    interface=bridge1 comment="" disabled=no
add address=10.5.1.1/24 network=10.5.1.0 broadcast=10.5.1.255 \
    interface=ether1 comment="" disabled=no
[admin@MikroTik] ip address>
```

导入指令

操作路径：import

在根目录使用 import file-name 指令还原指定的导出文件。这种还原是用于部分的配置丢失。

注：导入指令不可能导入收有的路由器配置只能导入部分的配置如。Filewall rules 中的策略

指令描述

File=[filename] - 载入需要导入的路由器配置文件

例如：

使用下面的指令操作载入保存配置文件:

[键入文字]

```
[admin@MikroTik] > import address.rsc
Opening script file address.rsc
Script file loaded successfully
[admin@MikroTik] >
```

系统复位

操作路径：system>reset-configuration

这个指令将会清徐掉路由器的所有配置，包括登陆的账号和密码（恢复为：admin”和空密码）ip 地址和其他配置将会被抹去，接口将会被禁用，在 reset 指令执行后路由器将会重起。

例如：

```
[admin@MikroTik] > system reset
Dangerous! Reset anyway? [y/N]: n
action cancelled
[admin@MikroTik] >
```

系统重启与关机

操作路径：/system reboot

当升级或安装新软件功能包时需要新重启路由器，在整个系统重启周期中执行功能包安装。

重启命令将发送信息给运行中的处理器，并停止和卸载系统文件，重启路由器。

```
[admin@mikrotik] > system reboot
Reboot, yes? [Y/N] : Y
System will reboot shortly
[admin@mikrotik] >
```

操作路径：/system shutdown

在路由器电源关闭前，应停止路由系统的运行，重启命令将发送信息给运行中的处理器，并停止和卸载系统文件，停止路由器。

在一些系统需要大概 30 秒（如果没有升级操作，通常最少需要 10 秒）才能安全关闭电源。

```
[admin@mikrotik] > system shutdown
Shutdown, yes(Y/N): Y
System will shutdown promptly
[admin@mikrotik] >
```

RouterOS 身份

操作路径：/system identity

通过命令可以查看路由身份明，这个身份名同样被用于DHCP客户端的“主机名（host name）”
查看路由器身份名：

```
[admin@mikrotik] > system identity print
    Name: “mikrotik”
[admin@mikrotik] >
```

设置路由器身份证：

```
[admin@mikrotik] > system identity set name=Gateway
[admin@Gateway] >
```

系统资源管理

操作路径：/system resource

通过查看系统资源可以了解 routerOS 的运行情况

注：通过 monitor 命令显示 CPU 用率，内存和硬盘使用情况。

查看基本的系统资源情况：

```
[admin@MikroTik] system resource> print
                uptime: 5h26m12s
               version: "3.0"
            free-memory: 17000kB
           total-memory: 30200kB
                 model: "RouterBOARD 500"
                   cpu: "MIPS 4Kc V0.10"
             cpu-count: 1
         cpu-frequency: 333MHz
              cpu-load: 3
        free-hdd-space: 14208kB
       total-hdd-space: 61440kB
 write-sect-since-reboot: 1047
      write-sect-total: 379983
            bad-blocks: 0
[admin@MikroTik] system resource>
```

连续查看系统 CPU 和空内存使用情况：

```
[admin@MikroTik] > system resource monitor
      cpu-used: 0
   free-memory: 115676

[admin@MikroTik] >
```

IRQ 使用监测

命令路径：system resource irq print

显示当前 IRQ 使用情况

```
[admin@MikroTik] > system resource irq print
Flags: U - unused
  IRQ OWNER
  1   keyboard
  2   APIC
U 3
  4   serial port
  5   [Ricoh Co Ltd RL5c476 II (#2)]
U 6
U 7
U 8
U 9
U 10
  11  ether1
  12  [Ricoh Co Ltd RL5c476 II]
U 13
  14  IDE 1
[admin@MikroTik] >
```

IO 端口监视

操作路径：system resource io print

显示当前硬件的 IO（input/output）端口使用情况

```
[admin@MikroTik] > system resource io print
 PORT-RANGE       OWNER
 0x20-0x3F        APIC
 0x40-0x5F        timer
 0x60-0x6F        keyboard
 0x80-0x8F        DMA
 0xA0-0xBF        APIC
 0xC0-0xDF        DMA
 0xF0-0xFF        FPU
 0x1F0-0x1F7      IDE 1
 0x2F8-0x2FF      serial port
 0x3C0-0x3DF      VGA
 0x3F6-0x3F6      IDE 1
 0x3F8-0x3FF      serial port
 0xCF8-0xCFF      [PCI conf1]
 0x4000-0x40FF    [PCI CardBus #03]
 0x4400-0x44FF    [PCI CardBus #03]
 0x4800-0x48FF    [PCI CardBus #04]
 0x4C00-0x4CFF    [PCI CardBus #04]
 0x5000-0x500F    [Intel Corp. 82801BA/BAM SMBus]
 0xC000-0xC0FF    [Realtek Semiconductor Co., Ltd. RTL-8139/8139C/8139C+]
 0xC000-0xC0FF    [8139too]
 0xC400-0xC407    [Cologne Chip Designs GmbH ISDN network controller [HFC-PCI]
 0xC800-0xC87F    [Cyclades Corporation PC300/TE (1 port)]
 0xF000-0xF00F    [Intel Corp. 82801BA IDE U100]

[admin@MikroTik] >
```

USB 端口信息

操作路径：system resource usb print

显示所有路由器可用的 USB 端口

Device (只读：文本) - 设备编号
Name (只读：文本) - USB 端口名称
Speed (只读：整型) - 该端口工作的带宽速度
Vendor (只读：文本) - USB 设备销售商名称

显示所有可用 USB 端口：

```
[admin@MikroTik] system resource usb> print
 # DEVICE VENDOR                NAME                 SPEED
 0 1:1                        USB OHCI Root Hub     12 Mbps
[admin@MikroTik] system resource usb>
```

PCI 信息

操作路径：/system resource pci print
Category (只读：文本) - 设备类型
Device (只读：文本) - 设备编号
Device-id ((只读：整型) - 16 进制设备 ID
Irq (只读：整型) - 该设备使用的 IRQ 编号
Memory (只读：整型) - 该设备使用的内存长度
Name (只读：文本) - 设备名称
Vendor (只读：文本) - 设备销售商名称
Vendor-id (只读：整型) - 设备 16 进制销售商

查看 PCI 糟相信情况：

```
[admin@MikroTik] system resource pci> print
 # DEVICE   VENDOR                    NAME                       IRQ
 0 00:13.0  Compaq                    ZFMicro Chipset USB (rev... 12
 1 00:12.5  National Semi              SC1100 XBus (rev: 0)
 2 00:12.4  National Semi              SC1100 Video (rev: 1)
 3 00:12.3  National Semi              SCx200 Audio (rev: 0)
 4 00:12.2  National Semi              SCx200 IDE (rev: 1)
 5 00:12.1  National Semi              SC1100 SMI (rev: 0)
 6 00:12.0  National Semi              SC1100 Bridge (rev: 0)
 7 00:0e.0  Atheros Communications      AR5212 (rev: 1)           10
 8 00:0d.1  Texas Instruments          PCI1250 PC card Cardbus ... 11
 9 00:0d.0  Texas Instruments          PCI1250 PC card Cardbus ... 11
10 00:0c.0  National Semi              DP83815 (MacPhyter) Ethe... 10
11 00:0b.0  National Semi              DP83815 (MacPhyter) Ethe... 9
12 00:00.0  Cyrix Corporation          PCI Master (rev: 0)
[admin@MikroTik] system resource pci>
```

系统监测 watchdog

系统监督的特征是当系统软件万一出现错误会重启。

规格

[键入文字]

功能包需求：system
等级需求：Level 1
操作路径：/system watchdog
技术与标准：

当一个 IP 地址没有响应或者系统已被锁定这个菜单许可配置当前系统。软件监督计时器是用来提供商一次的选择记录，但是在特殊的情况下（由硬件故障引起的）它能锁定自己，对于 RouterBOARD 的硬件监督设备来说她囊在任何情况下重启。

属性描述

Auto-send-supout (yes/no;默认：no) - 帮助文件是自动产生它能通过邮件发送
Auto-send-supout (yes/no;默认：yes) – 当软件错误发生时，有一文件”autosupout.rif” 是自动产生。这个”autosupout.rif”文件是对”autosupout..old.rif”的重命名
No-ping-delay (时间；默认：5m) - 在重启以后多久去测试和 ping watch-address,默认设置是如果 watch-address 被设置为不可达，这时路由器将在每 6 分钟的时候重启。
Send-email-from (文本；默认：“”) - 邮件地址是发送来自于帮助文件，如果没有设置，可以通过操作路径/tool e-mail 开启其功能
Send-email-to((文本；默认：“”)邮件地址是向帮助文件发送
Send-amtp-server (文本；默认：“”) - SMTP 服务地址是向通过帮助文件发送，如果没有设置可以通过操作路径/tool e-mall 开启其功能
watch-address（IP 地址：默认：none）- 如果设置这功能了的话，万一 6 次按照顺序 ping 指定 ip 地址（10 秒发送一次）出现错误，系统会重启
none - 不可用的选项
watchdog-timer (yes/no:默认：no) - 是否重启决于在一段时间系统无法响应

万一系统崩溃系统产生的帮助文件并自动通过 192.0.2.1 发送到 support@example.com:

```
[admin@mikrotik] system watchdog > set auto-send-supout=yes \
\… send-to-email=support@example.com send-smtp-server=192.0.2.1
[admin@mikrotik] system watchdog> print
      Watch-address: none
    Watchdog-timer: yes
      No-ping-delay: 5m
  Automatic-supout: yes
  Auto-send-supout: yes
  Send-smtp-server: 192.0.2.1
       Send-email-to: support@example.com
[admin@mikrotik] system watchdog >
```

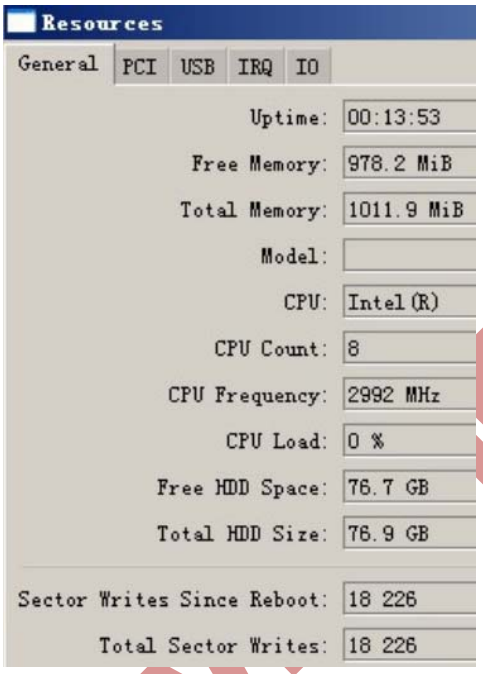
启用多 CPU

操作路径：/system hardware

如果你使用的是多 CPU，但你在启动 routerOSv3 后，在/system resources 中查看只有一个 CPU 在运行通过下面的命令启用多 CPU 功能：

```
 [admin@MikroTik] > system hardware
[admin@MikroTik] /system hardware>
..  /  :  edit  export  get  print  set
[admin@MikroTik] /system hardware> set multi-cpu=yes ;
[admin@MikroTik] /system hardware> prin
  multi-cpu: yes
[admin@MikroTik] /system hardware>
```

设置完成后。重启路由即可。



升级和降级 RouterOS

当你通过 BT 下载完 routerOS 软件后如 routerOS－ALL－3.5，里面有多个文件，每个文件对应不同的硬件做升级和降级设置，具体区如下：

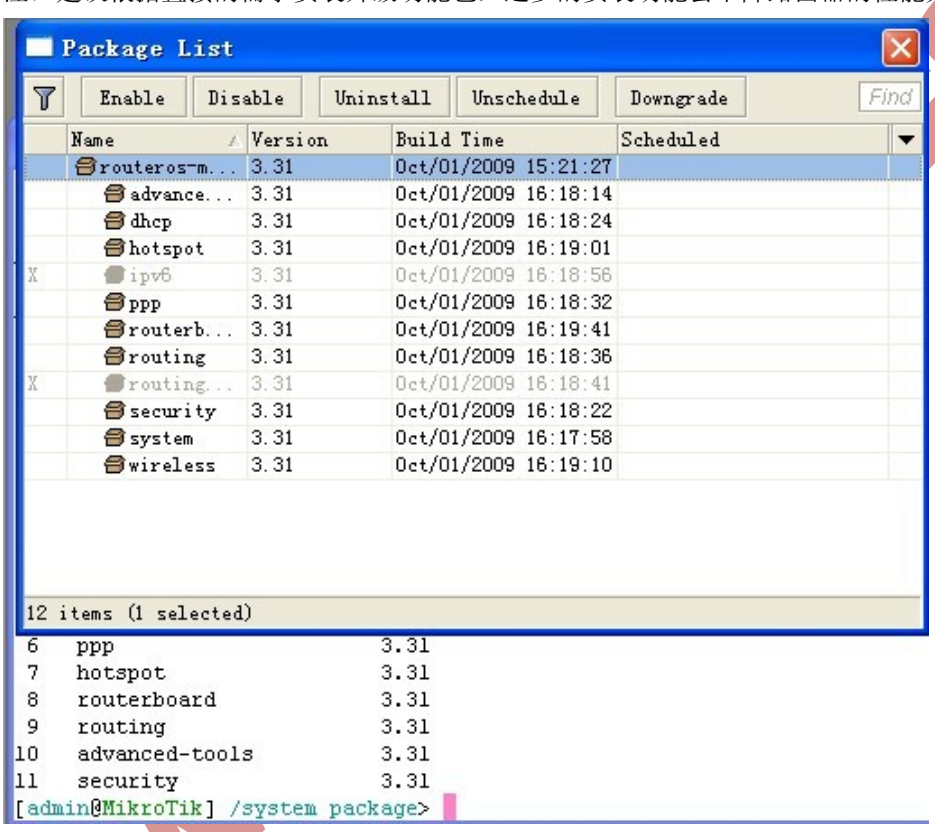
BT 包里面有四个文件名：
All_packages_mipsbe 对应所有 Atheros 芯片的 RB400 系列产品
All_packages_mipsle 对应 RB500 系列（RB133.RB133c,RB150,RB192.RB532）MIPS 4Kc 芯片
All_packages_ppc 对应 RB300,RB600 系列（RB333.RB600,RB1000,RB800）powerPC 芯片
All_packages_x86 对应所有 x86 构架的 PC 设备（AMD,INTEL,VIA 和其他 x86PC）

其他文件：

Mikrotik－X.Xiso 光盘镜像文件，用于 x86 安装

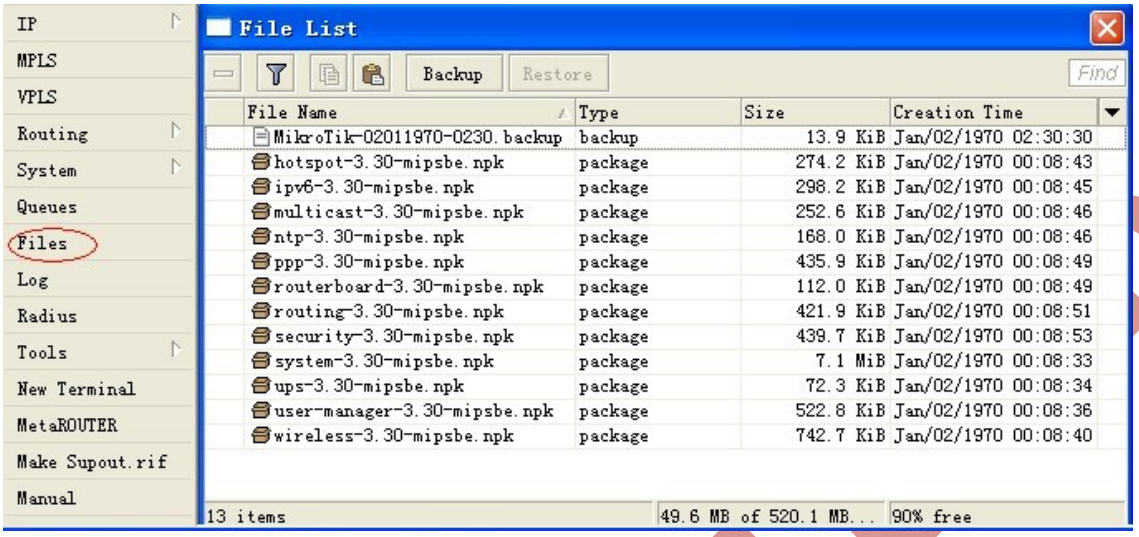
如果是 2.9 版本的 BT 文件区分如下：
All_packages_ns 对应 RB100 系列和 RB500 系列（RB133.RB133c,RB150,RB192.RB532）MIPS 4Kc 芯片
All_packages_x86 对应所有 x86 构架的 PC 设备（AMD,INTEL,VIA 和其他 x86PC

2．根据你使用 routerOS 的情况不同，选择上传的升级包（注：system-3.31.x.npk）的升级包是必须要的。如何来确定你使用的功能，可以通过在 system package>的目录中查询对照。
注：建议根据直接的需求安装升级功能包，过多的安装功能会下降路由器的性能如下图：

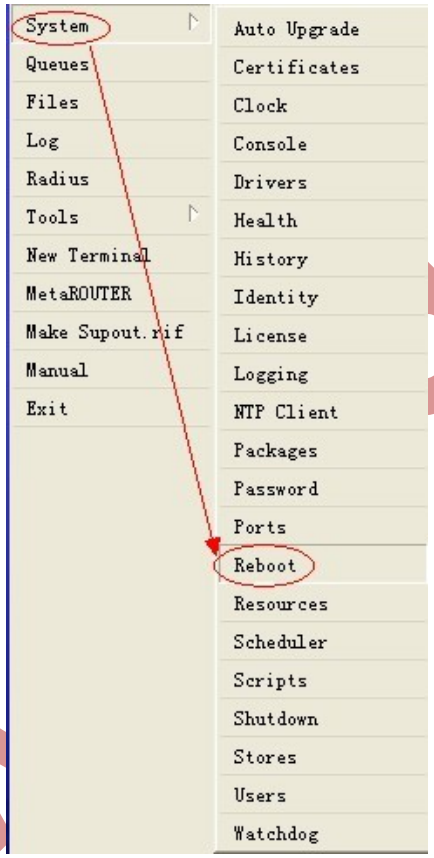


根据你在 system package 中的能够包选择－－对应的功能白进行升级，system 功能包是必须安装的。

3.选择好对应的功能包后，通过 winbox Files “上传导功能包：



4.功能包上传完成后，通过命令正常重启路由器：



RouterOS 在重启时。同时也在执行功能包的安装，在路由器本机的显示屏上可用看到安装进度条。重启完路由器后回看到路由器已经升级为新的版本。

降级选项

在 system package 中可以看的右上角有一个 dowgrade 的命令，这个将高版本降级到低版本的选项（需要同样将低版本的功能包上传到 routerOS 的 FTP 的 files 中）。

升级 RouterBOARD 固件

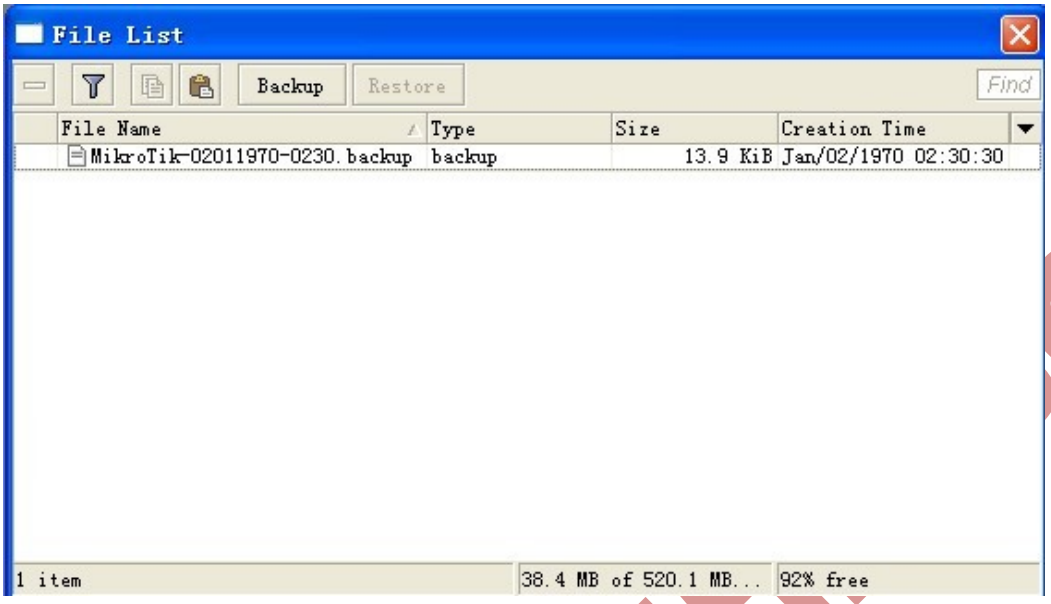
RouterBOARD 产品启动 BOIS 程序都有更新，对 routerBOARD 系列的启动引导和硬件兼容性进行修正和更新。
RouterBOARD 固件后缀名为:Fwf,每个系列的 routerBOARD 所使用的估计都不相同，固件下载地址可以到 www.mikrotik.com,如下面的列表：

RB 型号	固件前缀
RB1000	Mpc8548
RB600	Mpc8343
RB333	Mpc8323
RB400 系列（411/A/AH、433/AH、433AH、450/G、493/AH）	Ar7100
RB532	Rc32434
RB100 系列（112、133/C、150、192）	Adm5120

每隔一段时间，routerBOARD 的固件都会更新一次，所以通过 routerOS 中操作更新最新的 routerBOARD 固件，升级固件只能在命令型操作，首先我们需要查看 routerBOARD 当前的固件情况如下图：

```
[admin@office] /system> routerboard
[admin@office] /system routerboard> prin
  Routerboard: yes
        Model:450
Serial-namber: “188901ED9E57”
Current-firmware: “2.16”
Upgrade- firmware: “2.18”
[admin@office] /system routerboard>
```

如上面看到的，Current-firmware 是当前的固件信息：2.16，而我们最新的估计是 2.18 所以我们要通过上传固件到 routerOS 的 file 目录中（通过拖放的方式放到 winbox 中的 file list），当前的 RouterBOARD 是 RB450，所以这里我们上传的 Mikrotik 的固件文件如下图：



上次完后，然后我们通过 upgrade 命令升级：

```
[admin@office] /system routerboard> upgrade
Do you really want to upgrade firmware? (y / n): y
Firmware upgrade succeefully, please reboot for changes to take effect !
[admin@office] /system routerboard>
```

按照提示升级固件，升级后要求重启设备，才可以更新。

RouterOS 常用协议与端口

本文档列举了各种 Mikrotik RouterOS 服务用到的协议端口，它将帮助你决定为什么你的 mikrotik 路由器监听某些端口。以及如果你想要对某些服务禁止或者授权访问你都应该禁用/启用什么。请见其他相关的手册以获取更多解释。
操作路径：/ip service

属性描述

Name - 服务名称
Port (整型：1…65535) - 监听的端口
Addreaa (ip 地址 掩码；默认：0.0.0.0/0) - 可使用服务的 ip 地址
Certificate (名称：默认：none) - (对于不需要认证的服务缺省) 特定服务所使用的认证名称

实例

设置 WWW 服务能够从 10.10.10.0/24 网络 8081 端口可访问：

```
[admin@mikrotik] > ip service
[admin@mikrotik] > ip service>prin
Flags: X - disabled, I - invalid
#      name                 port        address          certificate
0      telent               23          0.0.0.0/0
1      ftp                  21          0.0.0.0/0
2      www                  80          0.0.0.0/0
3   x  www-ssl              443         0.0.0.0/0        none
4   x api                   8728        0.0.0.0/0
5   winbox                  8291        0.0.0.0/0
[admin@mikrotik] > ip service>
[admin@mikrotik]  ip service> set  www  port=8081  address=10.10.10.0/24
[admin@mikrotik]  ip service> print
Flags: X - disabled, I - invalid
#      name                 port        address          certificate
0      telent               23          0.0.0.0/0
1      ftp                  21          0.0.0.0/0
2   w   ww                  8081        10.10.10.0/24
3   x  www-ssl              44          0.0.0.0/0        none
4   x  api                  8728        0.0.0.0/0
5      winbox               8291        0.0.0.0/0
[admin@mikrotik]  ip service>
```

服务列表

以下便是 Mikrotik RouterOS 服务所用的协议和端口的列表。一些服务需要安装附加功能包，并且需要管理员启用，例如：带宽服务器。

端口/协议	描述
20/tcp	文件传输协议 FTP [数据连接]
21/tcp	文件传输协议 FTP [控制连接]
22/tcp	安全命令行解释 SSH 远程登录协议 （仅与安全封装一起）
23/tcp	远程通信网络协议
53/tcp	域名服务器 DNS
53/udp	域名服务器 DNS
67/udp	自举协议 或 DHCP 服务器 （仅与 dhcp 功能包一起）
68/udp	自举协议 或 DHCP 客户 （仅与 dhcp 功能包一起）
80/tcp	万维网（WWW）HTTP
123/udp	网络时间协议 NTP （仅与 ntp 功能包一起）
161/udp	简单网络管理协议 SNMP （仅与 snmp 功能包一起）
443/tcp	安全接口层 SSL 加密 HTTP（仅与 hotspot 功能包一起）
500/udp	Internet Key Exchange IKE protocol（仅与 IPsec 功能包一起）
520/udp	选路信息协议 RIP（仅与路由功能包一起）
521/udp	选路信息协议 RIP（仅与 routing 功能包一起）
179/tcp	边界网关协议 BGP（仅与 routing 功能包一起）
1080/tcp	SOCKS 代理协议
1701/udp	Layer 2 Tunnel Protocol L2TP（仅与 PPP 功能包一起）
1718/udp	H．323 Gatekeeper Discovery（仅与 telephony 功能包一起）
1719/tcp	H．323 Gatekeeper RAS（仅与 telephony 功能包一起）

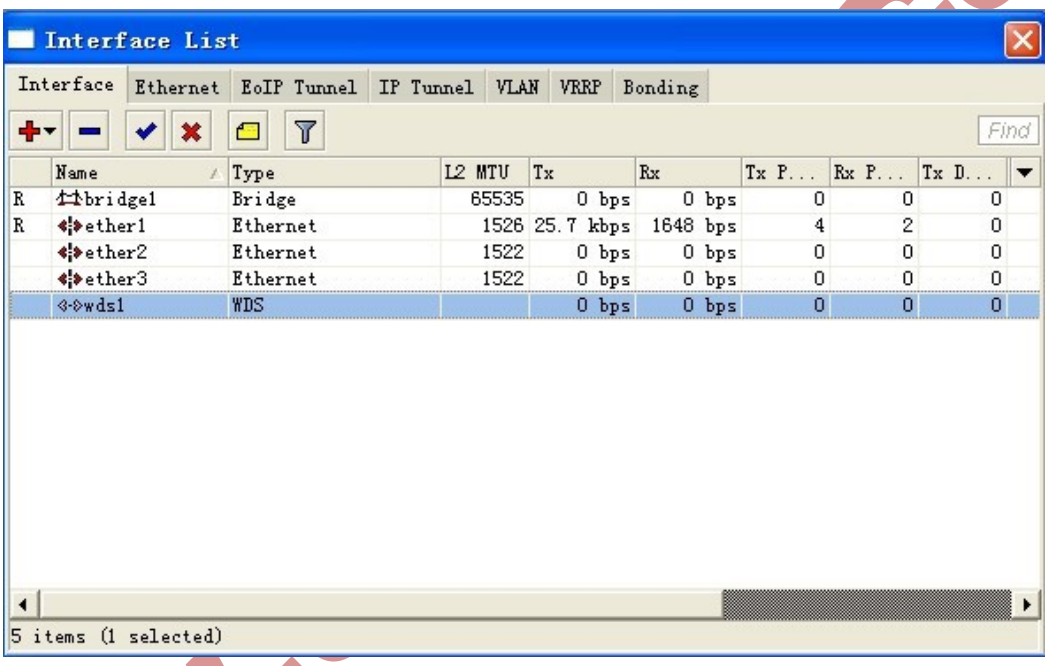
1720/tcp	H．323 呼叫安装（仅与 telephony 功能包一起）
1723/tcp	点对点隧道协议 PPTP （仅与 PPP 功能包一起）
1731/tcp	H．323 音频呼叫控制（仅与 telephony 功能包一起）
1900/udp	通用即插即用 upnp
2828/tcp	通用即插即用 upnp
2000/tcp	带宽测试服务器
3986/tcp	Winbox 代理
3987/tcp	安全 winbox SSL 代理（仅与安全功能包一起）
5678/udp	Mikrotik Neighbor Discovery Protocol
8080/tcp	HTTP 网络协议 （仅与 WEB 代理功能包一起）
8291/tcp	Winbox
20561/udp	MAC winbox
5000+/udp	H．323 RTP 音频流（仅与 telephony 功能包一起）
/1	ICMP – 网际控制报文协议
/4	IP – IP in IP（encapsulation）
/47	GRE – 普通路由封装（仅限 PPTP 与 EoIP）
/50	ESP – IPv4 压缩的安全有效载荷（仅与安全功能一起）
/51	AH – IPv4 认证标题（仅与安全功能包一起）
/89	OSPFIGP – OSPF 内部网关协议
/112	VRRP – 虚拟路由器冗余协议

第三章 接口配置（interface）

Interface 基本操作

在 interface 中包括物理接口网卡配置与虚拟接口的网卡配置，物理接口：ethernet,wireless,ISDN 等，虚拟接口：ppp,pppoE.PPTP,L2TP,EOIP,IPIP 和 bonding 等等。

Mikrotik RouterOS 支持各种网络接口卡，同样也支持一些虚拟接口像 VLAN,bridge 等。这些接口属性在接口列表中可以按你的需要进行配置。



操作路径：/interface

属性描述

Name(文本) - 接口名称
Status - 显示接口状态
Type (只读:arian/bridge/cyclages/eoip/ethernet/farsync/ipip/isdn-client/isdn-sever/12tp-client/12tp-server/moxe-c101/moxa-c502/mtsync/pc/ppp-client/pppoe-server/pptp-client/pptp-server/pvc/radiolan/sbe/vlan/wavelan/wireless/xpeed) - 接口类型
Mtu (整型) - 接口最大传输单位（bytes）
Rx-rate (整型：默认：0) - 最大数据接收率
O - no limits
Tx – rate (整型：默认：0) - 最大数据发送率

O - no limits

例如：

看下面得接口列表：

```
[admin@mikrotik] interface> print
Flags: x - disabled, D - dynamic, R - running
#      name            type            rx-rate      rx-rate        MTU
0   R  ether1          ether            0            0             1500
1   R  bridge1         bridge           0            0             1500
2   R  ether2          eth              0            0             1500
3   R  wlan1           wlan             0            0             1500
[admin@mikrotik] interface>
```

进入/interface bridge 桥接配置，添加一个桥：

```
[admin@mikrotik] /interface bridge> add
[admin@mikrotik] /interface bridge> prin
Flags: X - disabled,    R - running
0   R    name=”bridgel” mtu=1500 arp=enable mac-address=00:00:00:00:00:00
           Protocol-mode=none priority=0x8000    auto-mac=yes
           Admin-mac=00:00:00:00:00:00    max-message-age=20s    orward-delay=15s
           Transmit-hold-count=6    ageing-time=5m
[admin@mikrotik] /interface bridge>
```

流量监视

指令名称：/interface monitor-traffic

注： 可以监控通过接口的任何数据流量，并能同时监控多个网卡的流量情况

例如： 在命令行下的多接口监控：

```
[admin@mikrotik]    interface> monitor-traffic ether1, wlan1
  Received-packets-per-second:1        0
     Received-bits-per- secon:475bps      0bps
     Sent-packets- per- second:1          1
        Sent-bits-per- second: 2.43kbps 198bps
-- [Q quit | D dump | C-z    pause]
```

以太网接口（ethernet）

Mikrotik RouterOS 支持各种以太网。完全支持的以太网卡型号在 device driver list 可以找到。

功能包需要：system
等级需要：Level 1
操作路径：/interface ethernet
标准与技术协议：IEEE 802.3

以太网接口配置

操作路径：/interface ethernet

属性描述

Name (名称，默认：etherN) - 分配接口名称。
Arp（disabled | enabled | proxy-arp | reply-only; 默认：enabled）- 地址解析协议
Mtu (整型，默认：1500) - 最大传输单位
Disable-running-check (yes | no; 默认：yes) - 检测运行情况。
Mac-address (只读：MAC 地址) - 以太网卡的介质访问地址
Auto-megotiation (yes/no;默认：yes) - 当启用，接口会获取最好的网络连接。
Full-duplex (yes/no;默认：yes) - 定义数据传输全双工
Long-cable (yes/no;默认：no) - 改变电缆传输的长度设置（只用于 NS DP83815/6 卡）。电缆长度超过 50M，设备”long-cable=yes”
Speed (10Mbps/100mbps/1000mbps) - 设置以太网的书籍传输速度，参数由以太网卡支持的最大数据传输率确定。

例如

```
[admin@mikrotik] > interface print
Flags: X - disabled, D - dynamic, R - running
#      name            type          rx-rate       rx-rate          MTU
0   x ether1           ether         0             0                1500
[admin@mikrotik] > interface enable ether1
[admin@mikrotik] > interface print
Flags: X - disabled, D - dynamic, R - running
#      name            type          rx-rate       rx-rate          MTU
0   R  ether1          ether         0             0                1500
[admin@mikrotik] > interface ethernet
[admin@mikrotik] interface ethernet> print
Flags: X - disabled , R - running
#      name            MTU           MAC-ADDRESS             ARP
0   R  ether1          1500          00:0C:42:03:00:F2       enable
[admin@mikrotik] interface ethernet> print detail
Flags: X - disabled , R - running
0   R   name=”ether1”   mtu=1500   mac-address=00:0C:42:03:00:F2   arp= enable
        Disabled – running - check=yes auto - negotiation=yes full - duplex=yes
        Long - cable=no speed=100Mbps
```

[admin@mikrotik] interface ethernet>

接口状态监测命令
指令名称：/interface ethernet monitor

属性描述

Satus (link-ok | no-link | unknown) - 接口的状态，情况为：
Link-ok - 网卡以链接到网络
No-link - 网卡没有连接到网络
Unknown - 连接未确认
Rate（10Mbps | 100Mbps | 1000mbps）- 实际的连接速率
Auto-negotiation (done | incomplete) - 相邻连接的状态判断。
Done - 判断完成
Incomplete - 判断失败
Full-duplex (yes | no) – 是否为全双工数据传输

例如

通过 Monitor 命令可以查看现在以太网卡的连接状态，link-ok 为以太网络连接正常：

```
[admin@mikrotik] interface ethernet> monitor ether1，ether2
                  Status: link-ok    link-ok
      Auto—negotiation: done    done
                  Rate: 100mbps 100mbps
           Full-duplex:    yes      yes
```

修改以太网 MAC 地址：

[admin@mikrotik] interface ethernet>set 0 mac-address=00:0c:42:03:11:0A

第四章 **IP 配置与 ARP**

下面讨 IP 地址管理和地址解析协议设置，IP 地址在连接其它网络的设备使用 TCP/TP 协议，借助于地址解析协议（APP）连接在一个子网中的俩个设备。

功能规格

需要功能包：system
需要等级：level1
操作路径：/ip address, /ip arp
标准与技术：ip, ARP
硬件应用：无要求

IP 地址

操作路径：/ip address

在 IP 网络中，IP 地址是确认每个主机地址为目的，一个典型的 IP 大作（IPV4）由 4 个 8 位组成适当的路由地址还需要子网掩码，即那一段完整的 IP 地址位访问主机地址，那一段到网络地址，在大多数事例中。需要具体指名地址，掩码和接口参数。网络起始范围和广播地址能被自动计算出来。

能在以个接口上添加多个 ip 地址或在接口上不分配任何地址。当桥模式在俩个接口间被使用，在物理接口上添加 IP 地址并不是必须得（此从 routerOS 的 2.8 版本起），在桥模式的事例中，IP 地址能分配给属于桥模式的任何接口，但实际上带着将属于桥接口。你能使用/ip address print detall 查看地址归属的接口。

Mikrotik RouterOS 有下面的大作类型：

static - 用户手动分配给接口
dynamic - 确定 ppp, ppptp，或 pppoe 以连接，自动分配的接口

属性描述

Address (ip 地址) - 主机的 ip 地址
Broadcast (ip 地址；默认：255.255.255.255) - 广播 IP 地址，通过默认 IP 地址和子网掩码自动计算出的
Disable (yes/no;默认：no) - 指定那一个地址禁用或启用
Interface (名称) - 接口名称
Actual-interface (只读：名称) - 仅适用于逻辑接口，像桥（bridges）或隧道（tunnels）
Netmask (IP 地址；默认：0.0.0.0) - 指明网络地址，属于一个 IP 动作的一部分
Network（IP 地址；默认：0.0.0.0）- IP 地址网段。点对点连接时，网段到远端地址结束。

注：你不能有两个不同德 IP 地址来至相同的网段，例如：10.0.0.1/24 地址分配都 ether1 接口上，并且 10.0.0.132/24 地址分配到 ether2 接口上，这样是非法的。因为这两个属于同一个网段 10.0.0.0/24。

例如：添加 IP 地址 10.10.10.1/24 到 ether2 接口上

```
[admin@mikrotik] ip address > add    addreaa=10.10.10.1/24    interface=ether2
[admin@mikrotik] ip address > print
Flags: X - disabled, I – invalid, D - dynamic
 #    address            network         broadcast        interface
 0    2.2.2.1/24         2.2.2.0         2.2.2.255        ether2
 1    10.5.7.244/24      10.5.7.0        10.5.7.255       ether1
 2    10.10.10.1/24      10.10.10.0      10.10.10.255     ether2

[admin@mikrotik] ip address>
```

地址解析协议 ARP

操作路径：/ip arp

尽管 IP 数据包对话通过 IP 但硬件地址必须使用实际的传输数据从一个主机到另一个，通过地址解析协议从 OSI 第 3 层解析第 2 层的 MAC 地址。一个路由会有一个当前 ARP 等记表，通常这个表式建立为动态，但为增强网络安全性，建立一个静态的 ARP。

属性描述

Address（IP 地址）- 相应的 IP 地址
Interface（名称）- 被分配 IP 地址的接口名称
Mac-address (MAC 地址；默认:00：00：00：00：00：00) - 相应的 MAC 地址

注：最大的 ARP 的登记数为 1024.

如果 ARP 功能在接口上被关闭，例如：使用 arp=disabled,来至客户端的 ARP 请求将不被路由器回应，因此必须添加静态的 ARP 才行，例如：通过 arp 命令将路由器的 ip 和 MAC 地址必须添加到 windows 工作站中：

```
C:/> arp –s 10.5.8.254      00-aa-62-c6-09
```

如果在接口上的 arp 属性设置为 reply-only，这时路由器只应答来至静态 ARP 的请求，邻近的 MAC 大作将通过/ip arp 设置唯一的静态 ARP 列表。

例如：

```
[admin@mikrotik] ip arp> add address=10.10.10.10    interface=ether2
Mac-address=06:21:00:56:00:12
[admin@mikrotik] ip arp> print
Flags: X - disabled, I – invalid , H – dhcp, D - dynamic
#        ADDRESS            MAC-ADDRESS              INTERFACE
0   D    2.2.2.2            00.:30.:4F:1B:B3:D9      ether2
1   D    10.5.7.242         00:A0:24::9D:52:A4       ether1
2        10.10.10.10        06:21:00:56:00:16        ether2
[admin@mikrotik] ip arp>
```

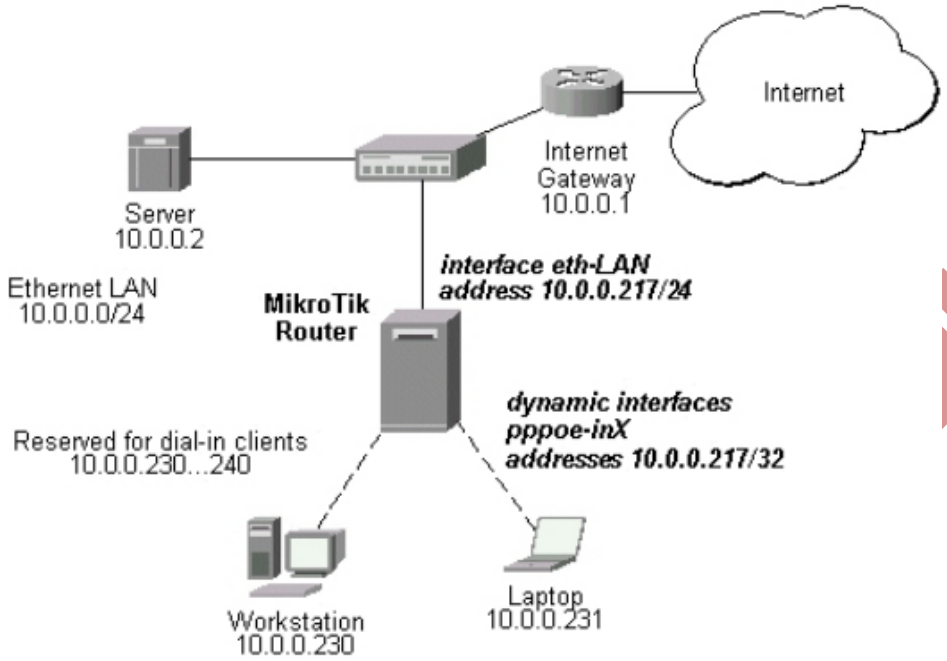
如果在一个接口上使用静态 ARP 记录会使网络更安全，你必须将该接口上的 arp 设置为'reply-only'，相关操作在下面的/interface 目录中：

```
[admin@mikrotik] ip arp> /interface ethernet set ether2 arp=reply-only
[admin@mikrotik] ip arp> print
Flags: X - disabled, I – invalid , H – dhcp, D - dynamic
#          ADDRESS            MAC-ADDRESS              INTERFACE
0   D    10.5.7.242           00.:A0.:24:9D:52:A4      ether1
1         10.10.10.10          06:12:00:56:00:12        ether2
[admin@mikrotik] ip arp>
```

ARP 代理

所有的物理接口，像以太网，atherOS 和 prism（wireless），aironet(PC)，waveLAN 等，都可设置地址解析或不设置。其它则设置使用 ARP 代理。如果 ARP 请求是从一个网络的主机发往另一个网络上的主机，那么连接这两个网络的路由器就可以回答该请求，这个过程称作委托 ARP 或 ARP 代理（proxyARP）。这样可以骗发起 ARP 请求的发送端，使它误以为路由器就是目的主机，而事实上目的主机是在路由器上网'另一边'。路由器的功能相当于门的主机的代理，把分组从其他之际转发给它

例如：看下列的网络配置：



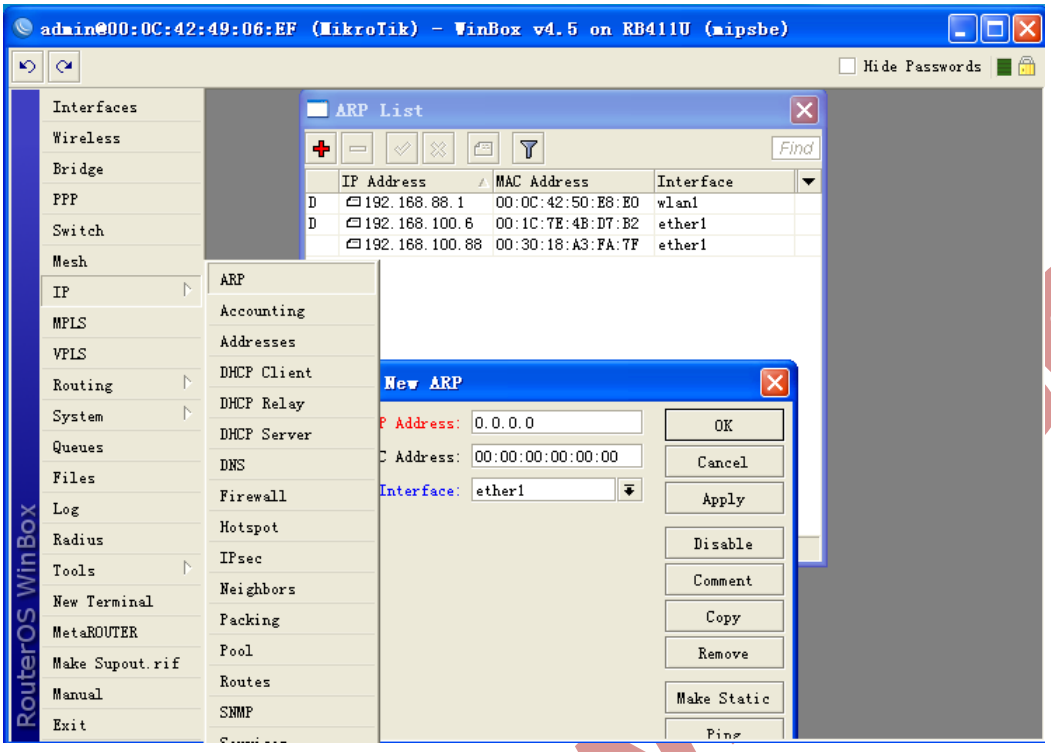
下面 RouterOS 设置

```
admin@MikroTik] ip arp> /interface ethernet print
Flags: X - disabled, R - running, S - slave
#    NAME        MTU     MAC-ADDRESS       ARP      MA.. SWITCH
0 R  ether1      1500  00:0C:42:11:54:F5 enabled    none 0
 [admin@MikroTik] ip arp> /interface print
Flags: X - disabled, R - running, D - dynamic, S - slave
  #    NAME                                      TYPE           MTU
  0 R  ether1                                    ether          1500
  1    prism1             prism          1500
  2 D  pppoe-in25          pppoe-in
  3 D  pppoe-in26          pppoe-in
[admin@MikroTik] ip arp> /ip address print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS           NETWORK        BROADCAST      INTERFACE
  0   10.0.0.217/24     10.0.0.0       10.0.0.255     eth-LAN
  1 D 10.0.0.217/32     10.0.0.230     0.0.0.0        pppoe-in25
  2 D 10.0.0.217/32     10.0.0.231     0.0.0.0        pppoe-in26
[admin@MikroTik] ip arp> /ip route print
Flags: X - disabled, I - invalid, D - dynamic, J - rejected,
C - connect, S - static, R - rip, O - ospf, B - bgp
    #    DST-ADDRESS       G GATEWAY        DISTANCE INTERFACE
    0  S 0.0.0.0/0         r 10.0.0.1       1        eth-LAN
    1 DC 10.0.0.0/24        r 0.0.0.0        0        eth-LAN
    2 DC 10.0.0.230/32      r 0.0.0.0        0        pppoe-in25
    3 DC 10.0.0.231/32      r 0.0.0.0        0        pppoe-in26
[admin@MikroTik] ip arp>
```

ARP 绑定操作

虽然主机在 IP 网络中是通过 IP 地址通话，但实际上以哦那个键地址（MAC）被用于主机到其他主机的数据传输。地址解析协议 address resolution protocol(ARP)是提供硬件之间的解析。每个路由器都一个 ARP 列表，记录 ARP 信息，由 ip 地址和相符合的 MAC 地址构成，一般 ARP 提供动态的 IP 与 MAC 地址对于关系，自动在 ARP 列表中产生、。路由器通过 ARP 列表的记录来回应各个主机的数据。我们也可通过静态的 ARP 记录，要求路由器只对静态的 ARP 做回应，这样就可以避免出现如有用户自己修改 IP 地址或者通过 ARP 病毒影响路由器工作，如通过下面的设置：

1. 在 winbox 中添加一个静态主机的 ARP 记录。



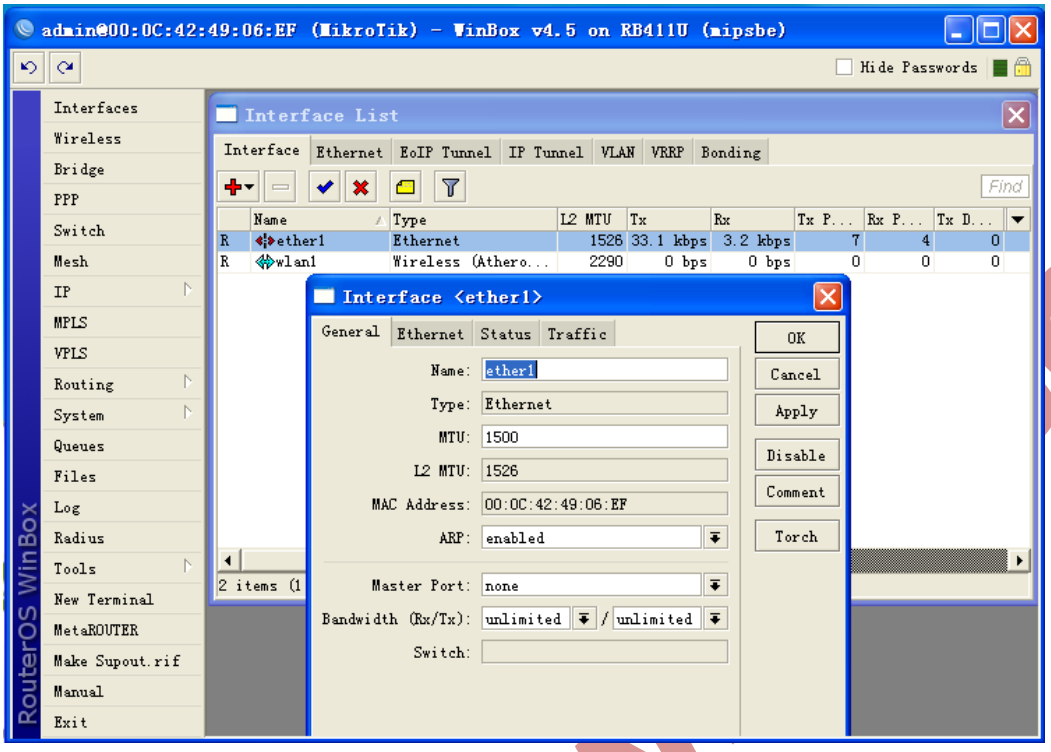
或者通过命令操作：

[admin@RB230].ip.arp> add address=10.10.10.10 interface=ether2 mac-address=00:21:56:00:12

同样我们可以使用：

将所有的 ARP 记录修改为静态。

2. 设置 ether2 interface 仅回应静态 ARP 的请求：



命令操作如下：

[admin@RB230] > interface ethernet set ether2　arp=reply-only

ARP 双向绑定事例

首先将所有/ip arp 列表中的所有 LAN 口德 ARP 信息变为静态的，我们可以同脚本做题处理修改。注意，可以通过脚本命令不一定将所有的内网的 ARP 参数修改完，可能需要手动添加。

```
:foreach i in [/ip arp find dynamic=yes interface=LAN] do={
  /ip arp add copy-from=$i}
```

然后设置 LAN 的网卡为：disabled

如果 ARP 功能在接口上被关闭，例如：使用 ARP=disabled 来至客户端的 ARP 请求将不被路由器回应，因此必须添加静态的 ARP 才行。例如。通过 arp 命令将路由器的 IP 和 MAC 地址必须添加到 windows 工作站中：

[admin@mikrotik] ip arp> /interface ethernet set LAN arp=disable

现在路由器已经绑定了内网主机的所有 IP 地址后，现在需要对 windows 电脑做对路由器绑订的设置

C:\> arp -s 10.5.8.254　00-aa-00-62-c6-09

也可以编辑 windows 自己的批处理文件（。Dat）操作

第五章 路由设置(Router)

下面的手册概述了 RouterOS 的理由的管理，针对目标。源地址和策略路由等，具体使用那一种理由方式，需要根据用户网络情况来选择。

操作规则

需要功能包：system
软件等级：Level 1
操作路径：/ip router./ip policy-routing

RouterOS 有下列类型的路由：

动态链接路由 是当在一个网卡上添加了 IP，会自动创建一个动态的路由（如 pppoE-client.pptp-client 和 DHCP-client 等自动添加网关）
静态路由 是用户自定义将数据传输到指定的网络区的路由，这需要手动指定默认的网关。

当添加一个 IP 地址后，会自动创建一动态的路由连接，你不需要手动添加连接路由器的路由配置，如果你使用一些路由协议（RIP 或 OSPF）你就需要定义静态路由到指定的网络，或指定默认网关。

负载均衡路由

当使用在到一个目标网络多于一个网关时，可以称为 Equal-cost multi-path routing 即将其作负载均衡。每一对新源/目标 IP 会选择一个新的网关。例如，一个 FTP 仅使用一个连接，但当一个新连接到不同德服务器就会使用其他的连接。

添加多网关的静态路由（格式如：gateway=x.x.x.x, y.y.y.y）路由协议会建立动态的多路路由。

基于策略的路由

在策略路由在 RouterOS 中配置多条路线，通过多种方式，操作方式：

标记期望的数据（源地址，目标地址和端口）设置一个 routing-mark
在 ip router 或 ip route rules 中配置目标和各种路由协议

基本路由操作

操作路径：/ip route

在路由子选项中，可以配置静态路由，负载均衡，路由标记

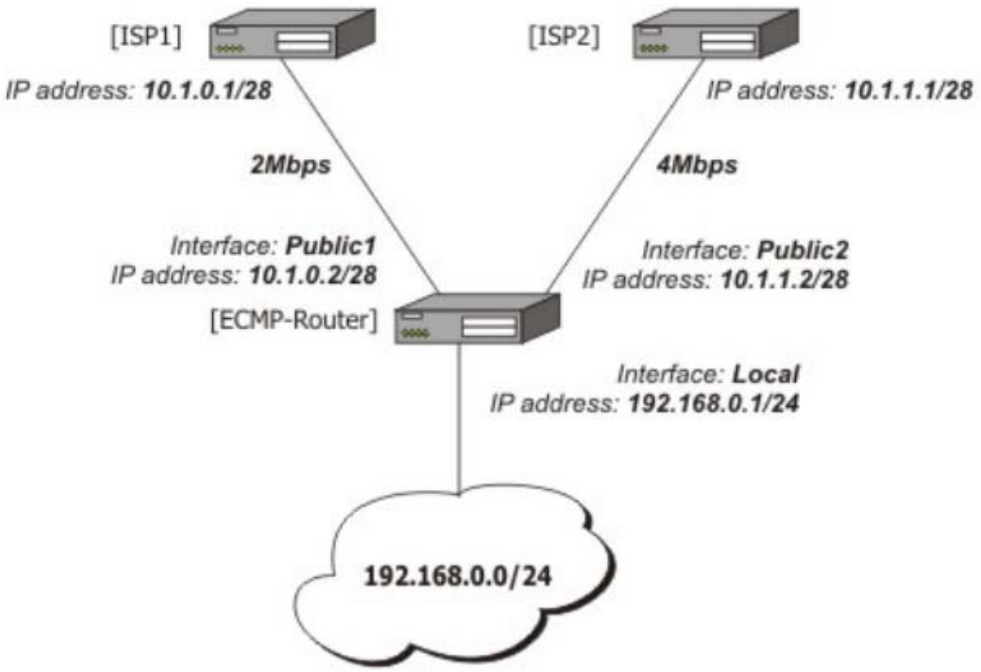
注：你能指定两个或更多的网关在路由中，而你能重复一些路由的不同类型的参数多次设置到一个网关上。

在一个路由器的两张网卡和两个 IP 地址中，添加两个静态路由到网络 10.1.12.0/24 和 0.0.0.0、0（默认的目标的地址）：

```
[admin@NAT] ip route> add dst-address=10.1.12.0/24 gateway=192.168.0.253
[admin@NAT] ip route> add gateway=10.5.8.1
[admin@NAT] ip route> print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf
 #     DST-ADDRESS       G GATEWAY        DISTANCE INTERFACE
 0 A S 10.1.12.0/24       r 192.168.0.253          Local
 1 ADC 10.5.8.0/24                                 Public
 2 ADC 192.168.0.0/24                              Local
 3 A S 0.0.0.0/0          r 10.5.8.1               Public
[admin@NAT] ip route>
```

负载均衡路由

考虑下面的网络环境，所有的数据都是从一个网络 192.168.0.0/24 到两个网关 10.1.0.1 和 10.1.1.1



注：ISPI 给我们的带宽是 2Mbps,ISP2 是 4Mbps,因此我们想要一个 1：2 的传输比（1/3 从 192.168.0.0/24 的数据走 ISP1,2/3 的通过）

路由器上网 IP 地址：

```
[admin@ECMP-Router] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         BROADCAST       INTERFACE
 0   192.168.0.254/24   192.168.0.0     192.168.0.255   Local
 1   10.1.0.2/28        10.1.0.0        10.1.0.15       Public1
 2   10.1.1.2/28        10.1.1.0        10.1.1.15       Public2
[admin@ECMP-Router] ip address>
```

添加默认路由，一个 ISP1 和两个 ISP2，这样我们得到了 1：3 的比例：

```
[admin@ECMP-Router] ip route> add gateway=10.1.0.1,10.1.1.1,10.1.1.1
[admin@ECMP-Router] ip route> print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf
 #     DST-ADDRESS       G GATEWAY        DISTANCE INTERFACE
 0 ADC 10.1.0.0/28                                 Public1
 1 ADC 10.1.1.0/28                                 Public2
 2 ADC 192.168.0.0/24                              Local
 3 A S 0.0.0.0/0         r 10.1.0.1                Public1
                       r 10.1.1.1                Public2
                       r 10.1.1.1                Public2
[admin@ECMP-Router] ip route>
```

源地址策略路由偶标记

在双线策略路由器情况下，早期的策略路由是通过标记一段地址走一条线路，如果我们标记 192.168.10.2-192.168.10.127 走线路 A,下的 IP 地址走 x 线路 B,这样的策略路由在一定得情况下出现效率不高的问题，如当用户 IP 地址是顺序增加，但没有到 127 的时候，线路 B 九不会起到流量分配的作用。

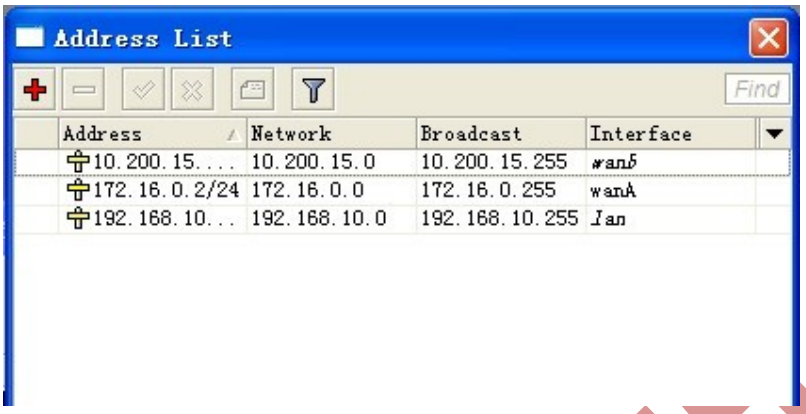
为了解决这样的问题，我们通过 routerOS 的 address-list 建立一个动作列表，分别将奇数偶数的 IP 地址分开，即奇数的 IP 地址走线路 A,而偶数的 IP 地址做线路 B，这样的策略路由便提高了双线的使用效率。

操作步骤如下：

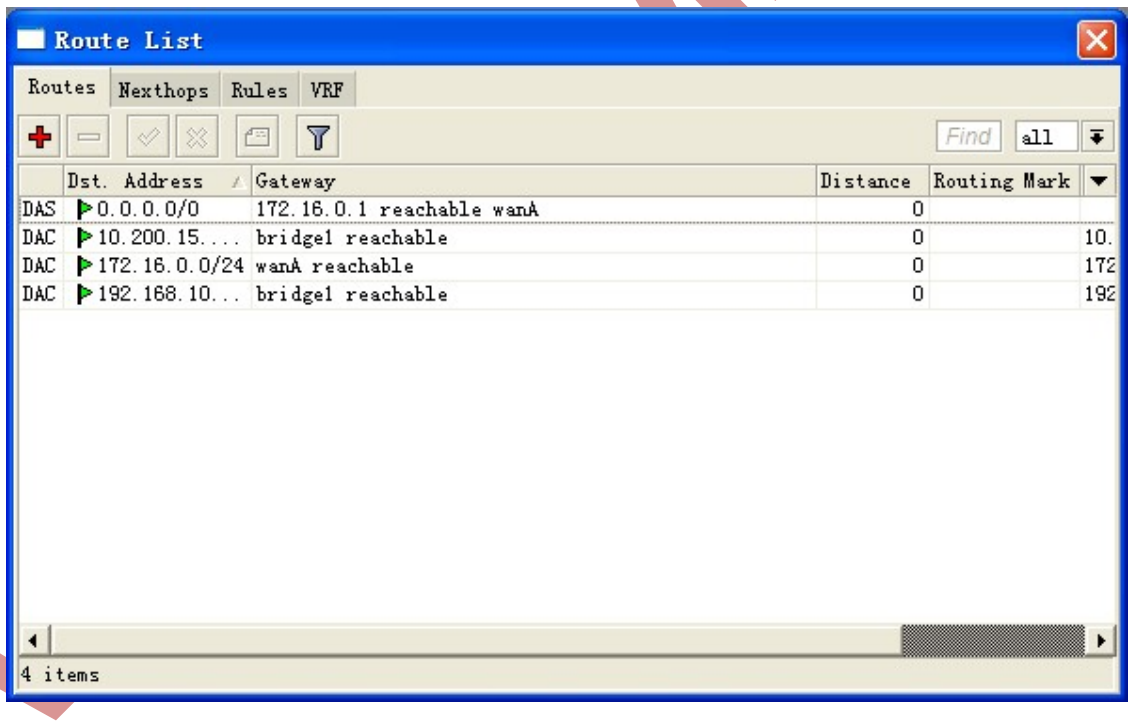
1. 配置好网络的 IP 地址和路由“
2. 在 IP firewall address-list 列表中建立奇数或者偶数地址列表；

3. 进入 IP firewall manngle 通过 src-address-list 标记数据包
4 在 IP route 中调用标记好的地址策略。配置路由。

步骤 1：我们首先进入路由器配置 IP 地址，假设我们有两条线路，分别是 A 和 B。A 的 IP 地址是 172.16.0.2，网关是 172.16.0.1；B 的 IP 地址是：10.200.15.20，网关是：10.200.15.1.

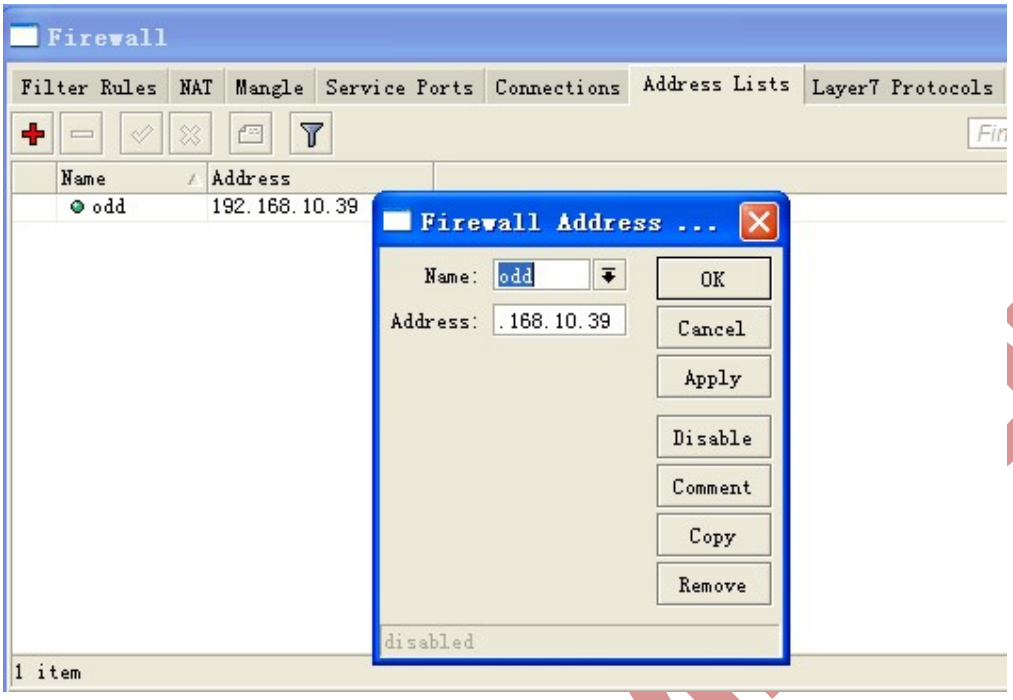


在 ip route 中配置以线路 A 的网关 172.16.0.1 为默认路由：



步骤 2：配置好的 IP 地址和路由后，按下在 ip filewall address-list 中添加奇数的地址列表，因为是双线路由，我们只需要配置一条奇数的列表，而偶数的列表可以不用配置，因为技术被标记后，以下的偶数地址。

我们将奇数列表地址名为：odd,并向地址列表里面添加你网络内所有的奇数的 IP 地址：



步骤 3：当奇数 IP 地址添加完成后，我们进入 ip filewall mangle 标记路由规则，我们选择 chain=prerouting:

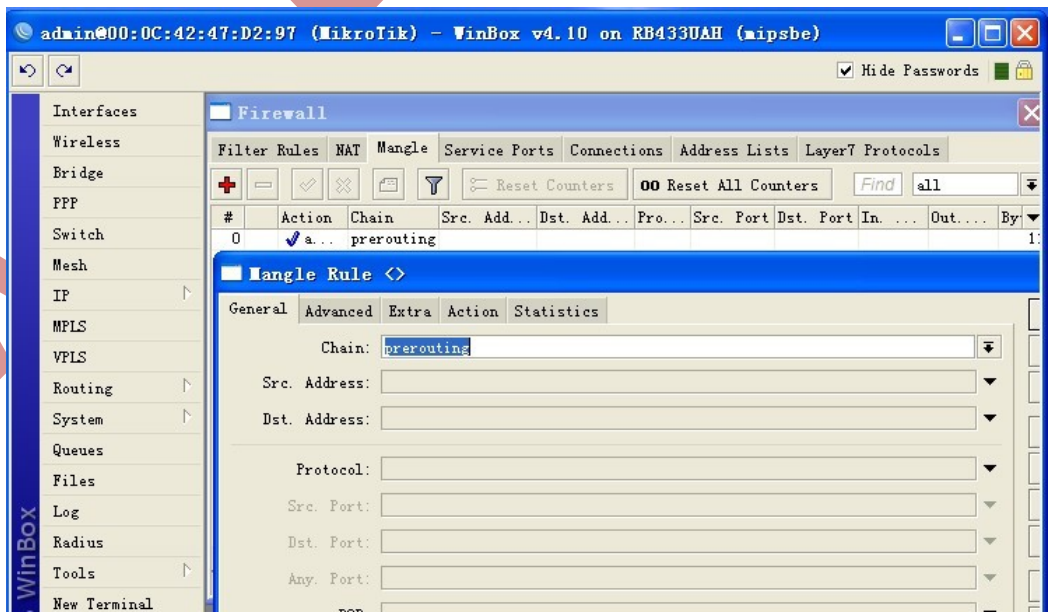
{admin@Mikrotik}/ipfirewallmangle>add chain=prerouting action=mart-routing new-routing-mark=odd src-address-list=odd

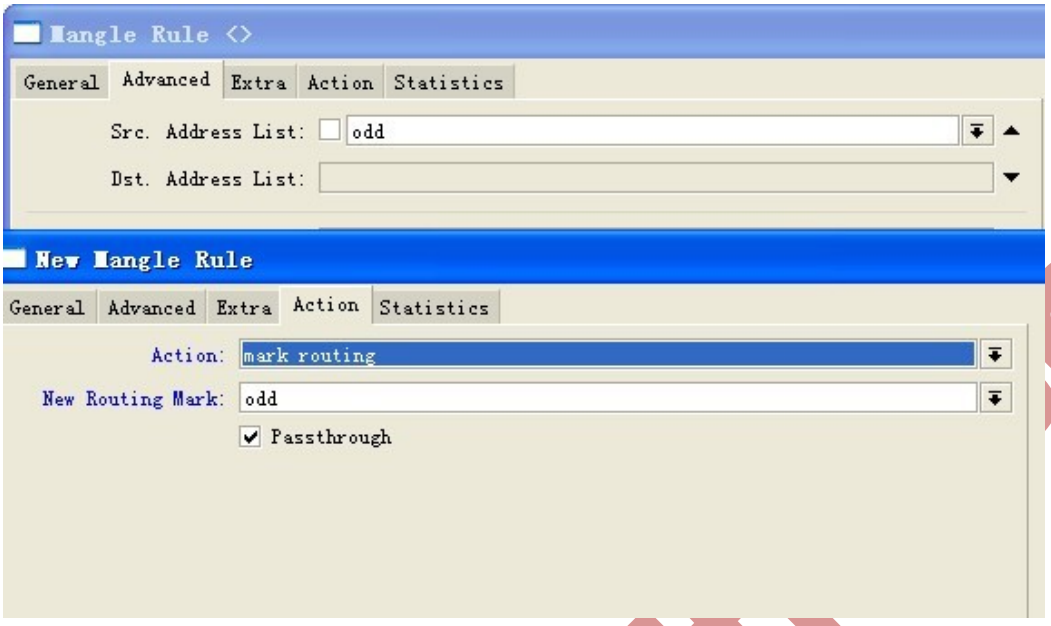
{admin@Mikrotik}/ ip firewall mangle>print

Flages:X-disable,i-invalid, D-dynamic

0 chain=prerouting action=mart-routing new-routing-mark=odd passthrough=yes

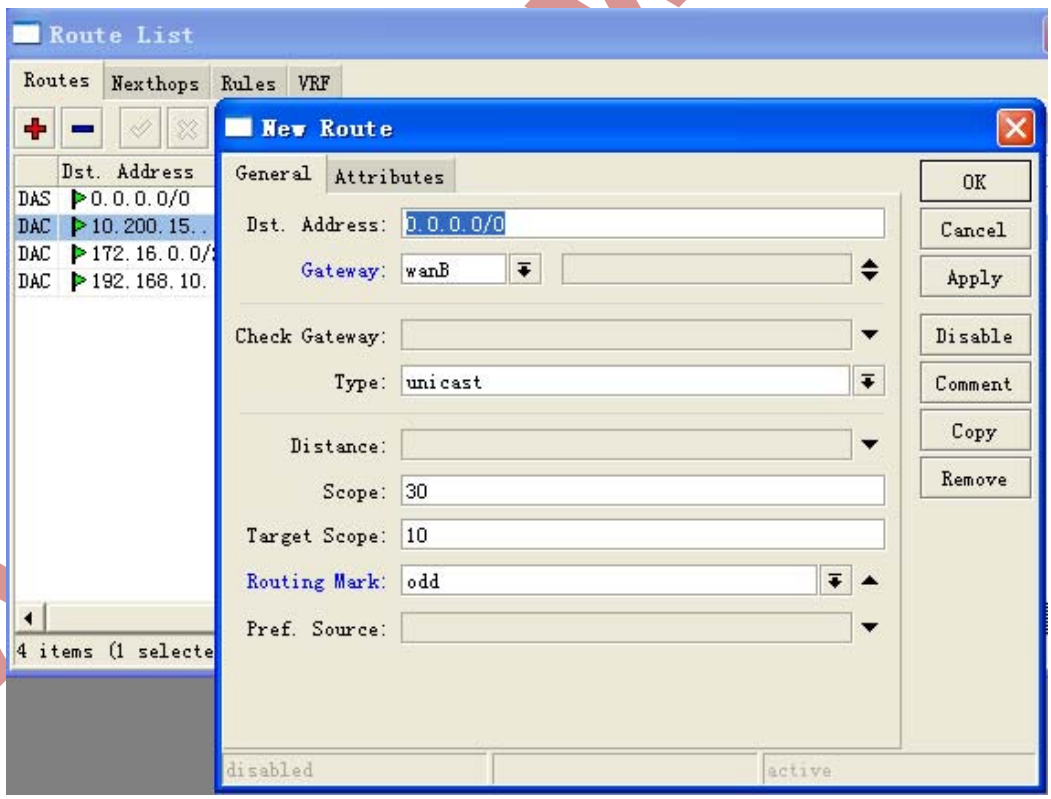
Src-address-list=odd





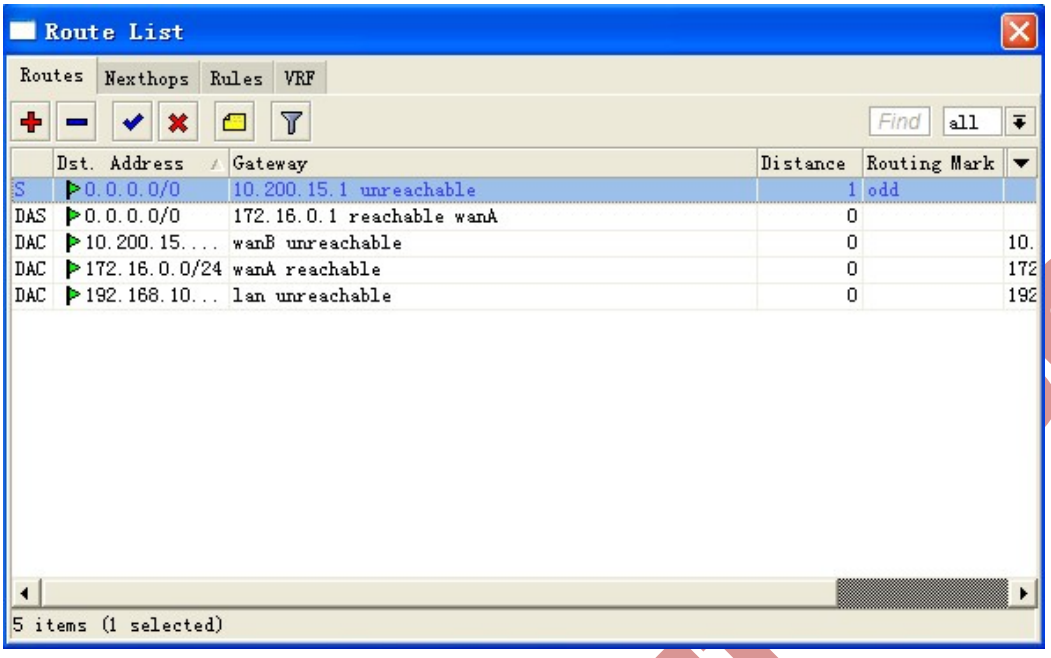
因为只标记了奇数的 ip 地址，其他的便是偶数的，所以不用再做配置，

步骤 4：配置完标记后，我们进入 ip route 配置路由，我们只需要将奇数的 ip 地址标记到线路 B 的网关即可，



配置只需要添加 gateway=10.200.15.1 和 routing-mark-odd,点确认：

[键入文字]



奇数的 ip 地址便从 B 线路上网 10.200.15.1 的网关出去，其他的偶数 IP 地址从默认的线路 A 的 172.16.0.1 网关出去。

光纤和 ADSL 静态路由

基本情况：假设用户有两条 Internet 线路，一条是使用固定调子的网通光纤 2M，另一条是使用电信拨号的 ADSL 通用为 2M。使用 NAT 伪装让局域网共享上网。在路由器上共有 3 块网卡，WAN1 用于网通光纤，WAN2 用于 ADSL 拨号，LAN 用于连接内网终端。

首先我们设置 WAN1 与 WAN2 的 ip 地址：ADSL 拨号大致如下：具体参考 pppoE 设置说明
配置 ADSL 线路
/interface pppoe-client 配置 ADSL 拨号信息
/interface pppoe-client add name pppoe-linel service CHN-telecom/user 以 c999@166password 123 interface WAN2 use-peer-dns yes mtu 1942 mru 1942

注：设置 pppoe-client 时当得到 ADSL 默认网关后，将 pppoe-client 中的 add-default-route=yes, 修改为 add-default-route=no 避免自动添加默认的电信路由。

```
{admin@mikrotik} ip address>add address 60.193.77.77/24 interface WAN1
{admin@mikrotik} ip address>print
Flages:X-disable,i-invalid, D-dynamic
#          ADDRESS          NETWORK              BROADCAST          INTERFACE
0    61.193.77.77/24       61.193.77.0          61.193.77.255      WAN1
D 1    218.88.32.10/24     218.88.32.1          0.0.0.0            PPPOE-OUT1
{admin@mikrotik} ip address>
```

下面配置内网地址为 192.168.0.1/24：

```
{admin@mikrotik} ip address> add address192.168.0.1/24 interface LAN
{admin@mikrotik} ip address>print
Flages:X-disable,i-invalid, D-dynamic
#          ADDRESS          NETWORK                BROADCAST            INTERFACE
0    61.193.77.77/24       61.193.77.0              61.193.77.255        WAN1
D 1 218.88.32.10/24        218.88.32.1               0.0.0.0             PPPOE-OUT1
2.    192.168.0.1/24      192.168.0.0              192.168.0.255            LAN
{admin@mikrotik} ip address>
```

下面我们配置一个默认网关，在这里我们以网通的 61.193.77.1 网关为默认网关，电信的作为静态路由：

```
{admin@mikrotik} ip route> add gateway=61.193.77.1
{admin@mikrotik} ip route>print
Flages:X-disable,A-active, D-dynamic
C-connect, s-static, r-rip b-bgp ,o-ospf
#    DST-ADDRESS       PREFSRC         GATEWAY              DISTANCE INTERFACE
0.ADC   61.193.77.0/24     61.193.77.77                        WAN1
1.ADC    218.88.32.1/32    218.88.32.10                       pppoe-out1
2.ADC    192.168.0.0/24    192.168.0.1                       LAN
3.A   S   0.0.0.0/0                  r 61.193.77.1            WAN1
{admin@mikrotik} ip route>
```

现在我们导入电信的静态路由表，电信和网通的路由表脚本在 www.mikrotikchina.com 的网站上可以下载到，操作根据说明要求设置，网通电信双线路由脚本操作方式：

添加脚本方式，请将你的正确的电信或网通的网关，使用用编辑-替换脚本里的“网关”，然后打开 winbox,点击 terminal(控制终端)然后复制脚本，并在 terminal l(控制终端)中点右键选择“paste”粘贴脚本，粘贴完后敲回车，即可完成！

这里我们将电信的网关 218.88.32.1 在“电信 ip 脚本”文本文件中使用替换操作将所有含“网关”的关键字替换为 218.88.32.1，然后复制并在 terminal 控制台中粘贴脚本，这样电信脚本即可导入。

```
{hcf@NAT}ip route>prin
Flages:X-disable,A-active, D-dynamic
C-connect, s-static, r-rip b-bgp ,o-ospf
#    DST-ADDRESS       PREFSRC       G GATEWAY              DISTANCE INTERFACE
0.ADC   61.193.77.0/24     61.193.77.77                        WAN1
1.ADC    218.88.32.1/32    218.88.32.10                       pppoe-out1
2.ADC    192.168.0.0/24    192.168.0.1                       LAN
3.A   S   0.0.0.0/0                  r 61.193.77.1            WAN1
```

```
 4 A   S   218.4.0.0/15          r218.88.32.1          pppoe-out1
 5 A   S   218.6.0.0/15          r218.88.32.1          pppoe-out1
 6 A   S   218.13.0.0/16         r218.88.32.1          pppoe-out1
 7 A   S   218.14.0.0/15         r218.88.32.1          pppoe-out1
 8 A   S   218.16.0.0/14         r218.88.32.1          pppoe-out1
 9 A   S   218.20.0.0/16         r218.88.32.1          pppoe-out1
 10 A   S   218.21.0.0/17        r218.88.32.1          pppoe-out1
 11 A   S   218.22.0.0/15        r218.88.32.1          pppoe-out1
 12 A   S   218.30.0.0/15        r218.88.32.1          pppoe-out1
 13 A   S   218.62128.0/15       r218.88.32.1          pppoe-out1
 14 A   S   218.63.0.0/16        r218.88.32.1          pppoe-out1
 15 A   S   218.64.0.0/15        r218.88.32.1          pppoe-out1
 16 A   S   218.66.0.0/16        r218.88.32.1          pppoe-out1
 .....
```

为保证一条线路断线时，到其他目标地址能正常连接，在/tool netwatch 中设置主机网关监控（具备设置参考 network 监控），并配置脚本编译。

如果当你使用静态路由制定网通或电信线路的时候，其中一条线路出现故障，需要切换到另外一条线路时我们需要设置以下脚本，如电信的线路出现故障，需要禁用掉电信网关的静态路由策略，让所有的数据周默认的网通线路，电信网关为：222.212.48.1.脚本设置如下：

```
当电信线路出现故障的时候，禁用掉所有到电信网关的策略
：foreach I in=[/ip router find gateway=218.88.32.1] do=[/ip rout disable si]
当电信线路正常后，启用所有电信策略
：foreach I in=[/ip router find gateway=218.88.32.1] do=[/ip rout disable si]
```

电信与网通目标地址路由

电信网通是现在比较常见的双线方式，通过路由指定分别让内网主机访问走电信和网通线路，该双线中，我们需要选择一条线路为主线，即默认路由出口，比如电信为主线，缺省网管网关为电信网关地址：网通线路需要通过导引路由表

上传电信或网通路由脚本后，在根目录下使用 import 命令导入：

```
[admin@mikrotik] >import cnc1.rsc
```

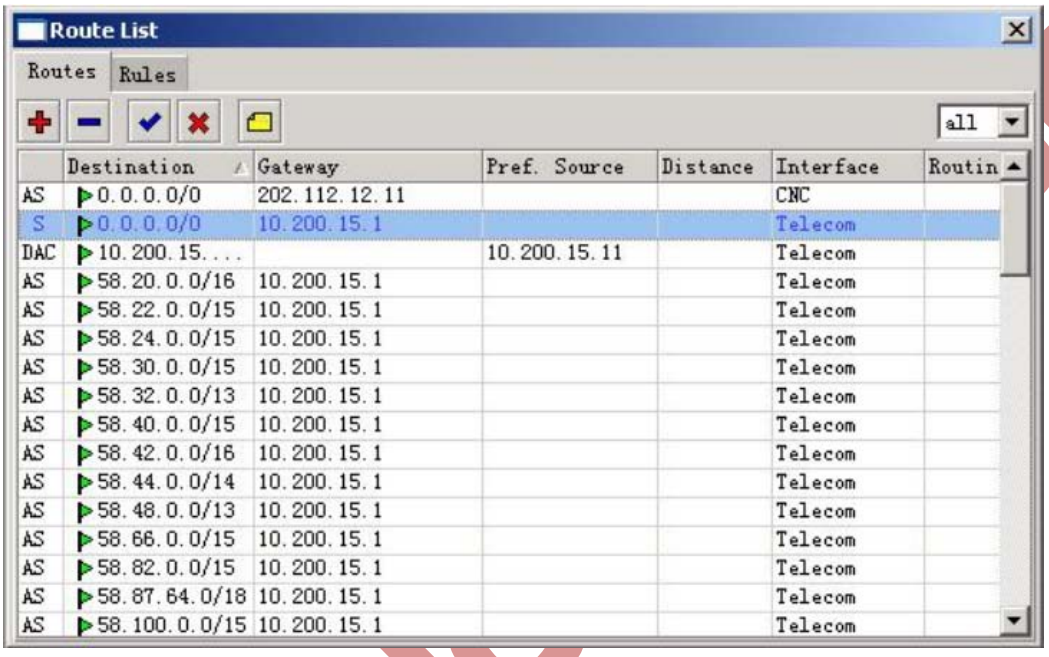
设置路由规则时命令如下：

```
/ip route add gatewall=对应网关地址 check-gatewall=ping routing-mark=telecom 或者 cnc
```

网关断线处理

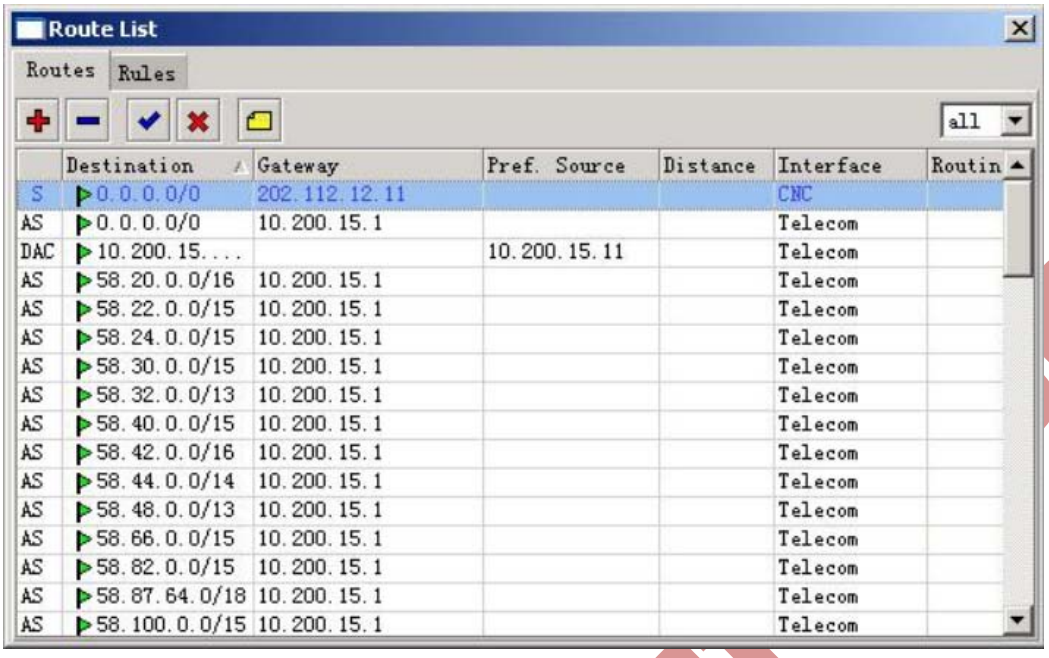
自动切换网关

RouterOS 中路由规则增加的两点功能：（！）在 RouterOS 在 v2.9 路由规则中增加了 check-gateway 的功能，能检测到网关的线路状态，如果网关无法探测到，便认为网关无法连接，会自动禁止访问网关的数据通过。check-gateway 功能的探测时间为 10s 一个周期。（2）在 RouterOS 中能够对缺省网关的判断，在 RouterOS 的如何一个路由表中只能存在一个缺省网关，即到如何目标地址为 0.0.0.0/0,没有做路由标记（routing-mark）的规则，如果存在另一个缺省网则认为是非法，路由将不予以执行，如下图：



	Destination	Gateway	Pref. Source	Distance	Interface	Routin
AS	0.0.0.0/0	202.112.12.11			CNC	
S	0.0.0.0/0	10.200.15.1			Telecom	
DAC	10.200.15....		10.200.15.11		Telecom	
AS	58.20.0.0/16	10.200.15.1			Telecom	
AS	58.22.0.0/15	10.200.15.1			Telecom	
AS	58.24.0.0/15	10.200.15.1			Telecom	
AS	58.30.0.0/15	10.200.15.1			Telecom	
AS	58.32.0.0/13	10.200.15.1			Telecom	
AS	58.40.0.0/15	10.200.15.1			Telecom	
AS	58.42.0.0/16	10.200.15.1			Telecom	
AS	58.44.0.0/14	10.200.15.1			Telecom	
AS	58.48.0.0/13	10.200.15.1			Telecom	
AS	58.66.0.0/15	10.200.15.1			Telecom	
AS	58.82.0.0/15	10.200.15.1			Telecom	
AS	58.87.64.0/18	10.200.15.1			Telecom	
AS	58.100.0.0/15	10.200.15.1			Telecom	

从上图我们可以看到，所有访问电信的 IP 段从 10.200.15.1 出去，其他的数据走网通的缺省网关出去，在我们可以这些网关的前缀都为“AS”，即确定的静态路由，而在第二排可以看到蓝色一行，他也是一个缺省网关，但因为一个路由表中只能存在一个缺省网关，所有前缀为“S”即静态但不确定的网关，被认为非法的。如果当 202.112.12.12.11 网关断线，则 10.200.15.1 会自动启用，变为缺省路由，实现现在的切换，如下：



当 202.112.12.11 断线后，check-gateway 在 10s 一个周期后探测到，并将 10.200.15.11 设置为缺省路由，如果 202.112.12.11 正常后，系统也将会将 201.112.12.11 设置为缺省路由，因为他是先于 10.200.15.1 添加入路由表中。

脚本处理方式

大多简单的方法是通过使用 netwatch 做监测。这里我们使用每间隔 5 秒钟 ping 一次路由器的默认网关（2.2.2.2），如果没有回应，我们将选择备用网（3.3.3.1）

```
/system script add name=down source=(/Ip route/
[…set [/ip route find dst-address=0.0.0.0]gateway3.3.3.1]
/system script add name=up source=(/Ip route/
[…set [/ip route find dst-address=0.0.0.0]gateway2.2.2.1]
/tool netwatch add host=2.2.2.2 interval=5s up-script=up down- script=down
```

现在有两条 internet 线路接入，我们需要同时使用两条线路并作负载均衡。因此我们需要在路由的默认路由上添加两个默认网关：

```
/ip route add gateway=1.1.1.1 ,2.2.2.1
```

下面是通过脚本合理的根据负载均衡的使用情况调整线路，保证在某条线路出现故障的时候其他线路能正常工作：

```
//system script add name=down source=(
:local R1
: local R2
:if([/tool netwatch get R1 satus]=up) do=[:set R1    1.1.1.1.]
```

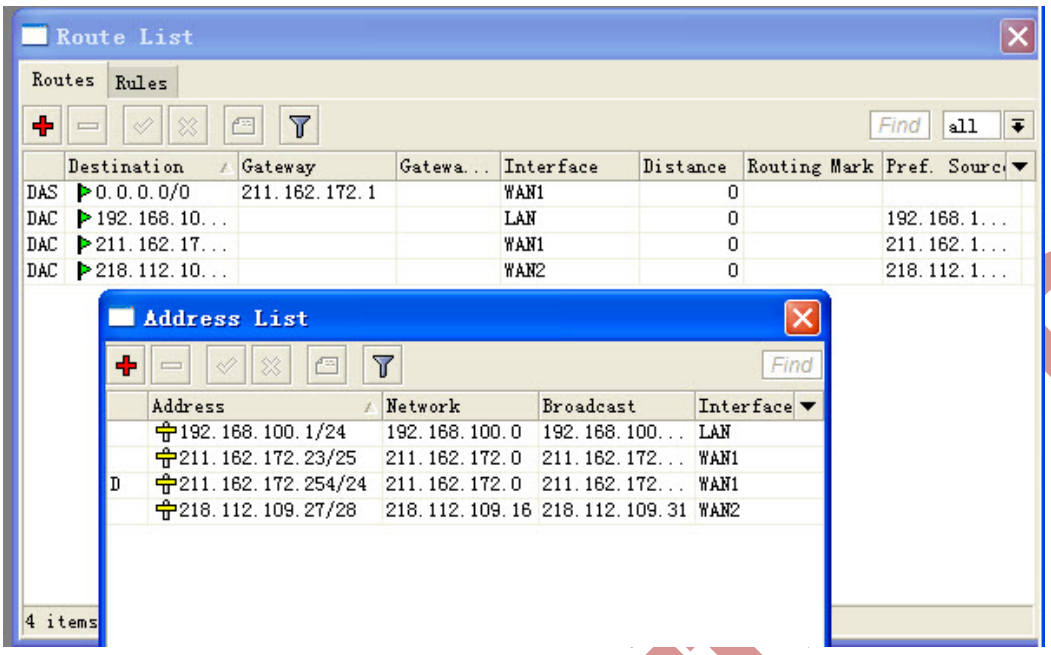
```
:if([/tool netwatch get R2 satus]=up) do=[:set R2    2.2.2.2]
/ip route set (/ip route find dst-addess=0.0.0.0/0)/
Gateway=(sR1…..Sr2)
/tool natwatch add comment=R1 host=1.1.1.1    interval=5s up-script=fo/
down- script=fo
/tool natwatch add comment=R2 host=2.2.2.2    interval=5s up-script=fo/
down- script=f0
```

通过修改脚本使其能使用三个或多个网关，例如：入我们第三个网关的地址是 3.3.3.1，脚本设置为：

```
//system script add name=down source=(
:local R1
: local R2
local R3
:if([/tool netwatch get R1 satus]=up) do=[:set R1    1.1.1.1.]
:if([/tool netwatch get R2 satus]=up) do=[:set R2    2.2.2.2]
:if([/tool netwatch get R3 satus]=up) do=[:set R3    3.3.3.3]
/ip route set (/ip route find dst-addess=0.0.0.0/0)/
Gateway=(sR1…..Sr2…..SR3)
/tool natwatch add comment=R1 host=1.1.1.1    interval=5s up-script=fo/
down- script=fo
/tool natwatch add comment=R2 host=2.2.2.2    interval=5s up-script=fo/
down- script=f0
/tool natwatch add comment=R2 host=3.3.3.3    interval=5s up-script=fo/
down- script=f0
```

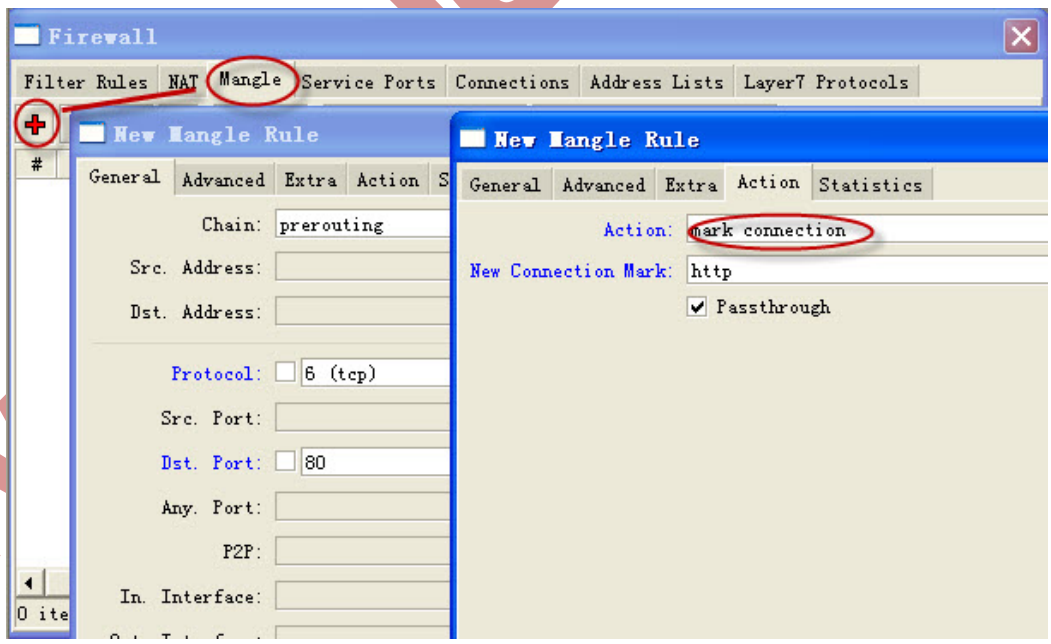
端口策略路由

Mikrotik RouterOS 可以支持多种策略路由，人我们常见的源地址，目标地址，同样支持端口的策略路由多种规则可以根据用户情况配合使用，现在我们通过下面的图解一步步实现端口的策略路由：

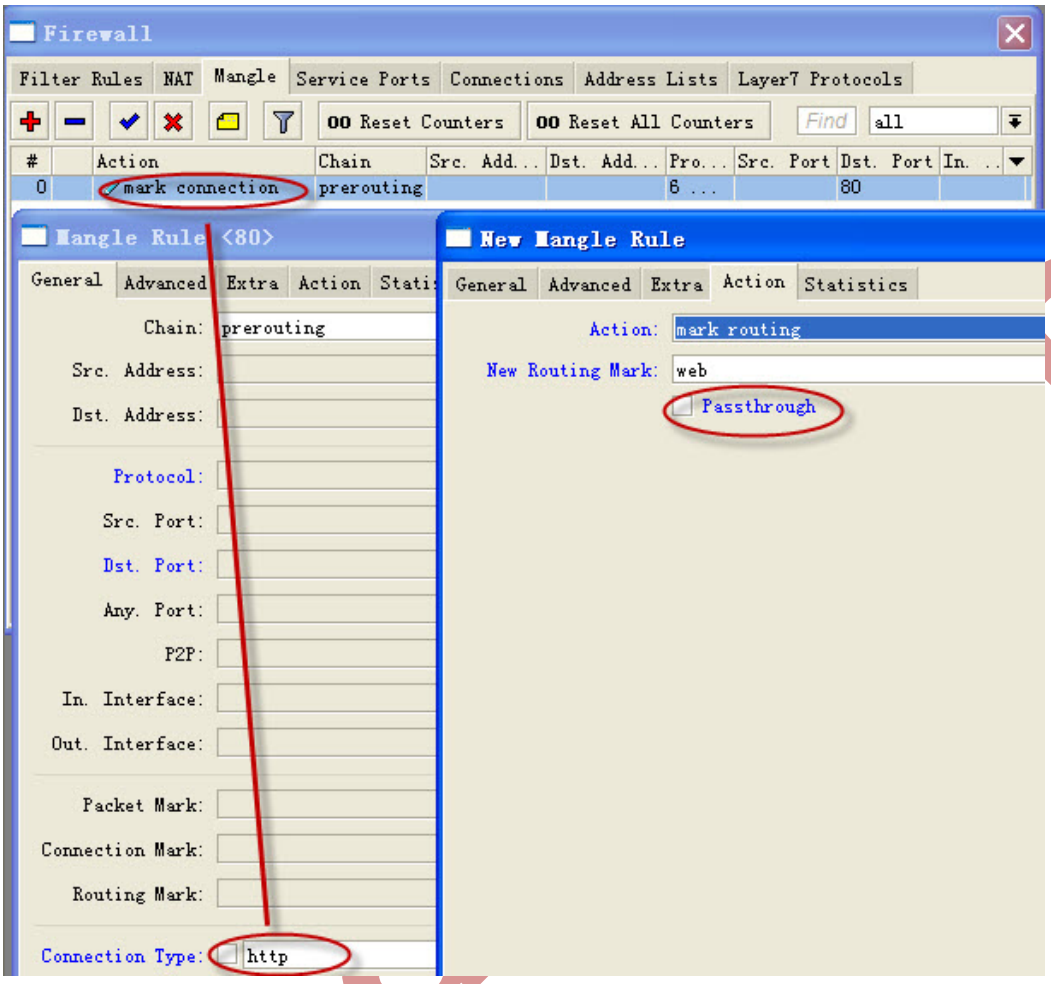


我们有两个ISP接入的线路，一个是WANI：200.162.172.23，一个是WAN2:218.112.109.27(地址为假设)，我们让默认的数据通过 WANI，让访问网页的数据通过 WAN2.

现在我们定义访问网页的端口，访问网页的端口是 TCP80,我们进入/ip filewall mangle 中数据标记

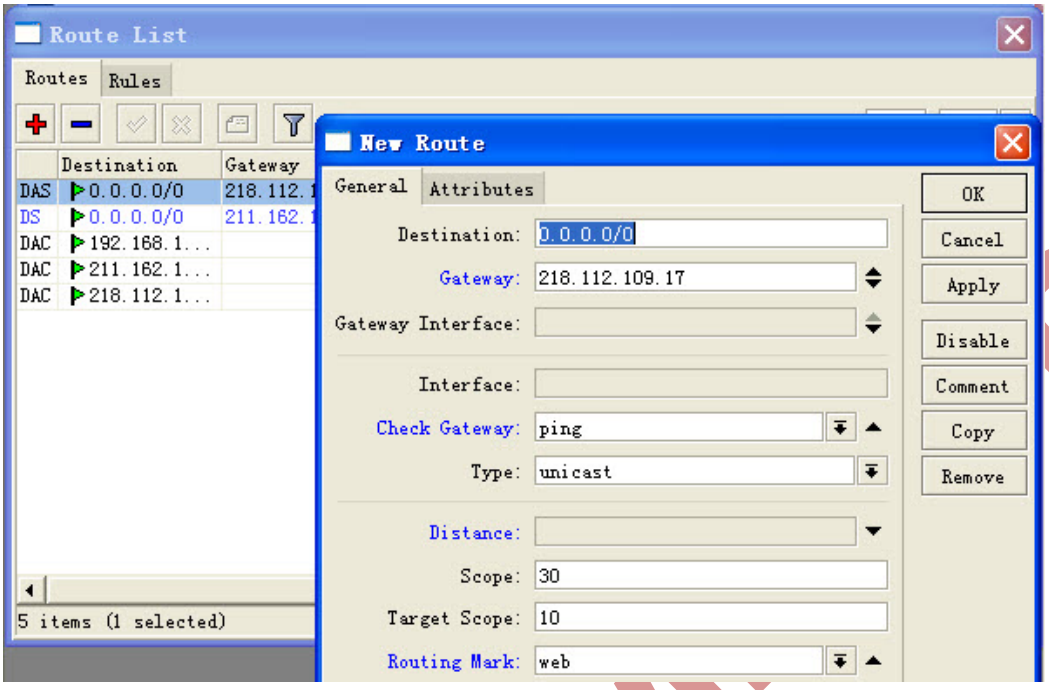


首先我们标记 80 端口的连接，标记为 HTTP,然后我们从这些连接中提取我们想要的数据：

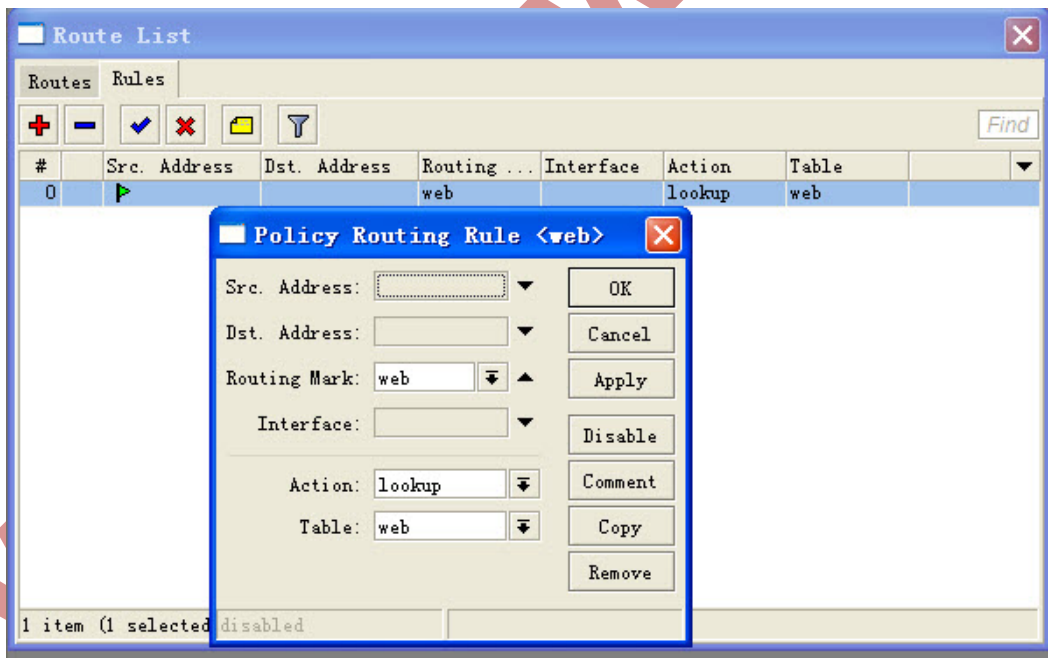


之后我们从标记提取路由标记，命名:web，因为我们在面前的连接标记中做过了 passthrough 的设置，在这里就不用在重复设置。

然后我们进入/ip route，配置路由我们让标记好的 80 端口路由区 WAN2 的线路：



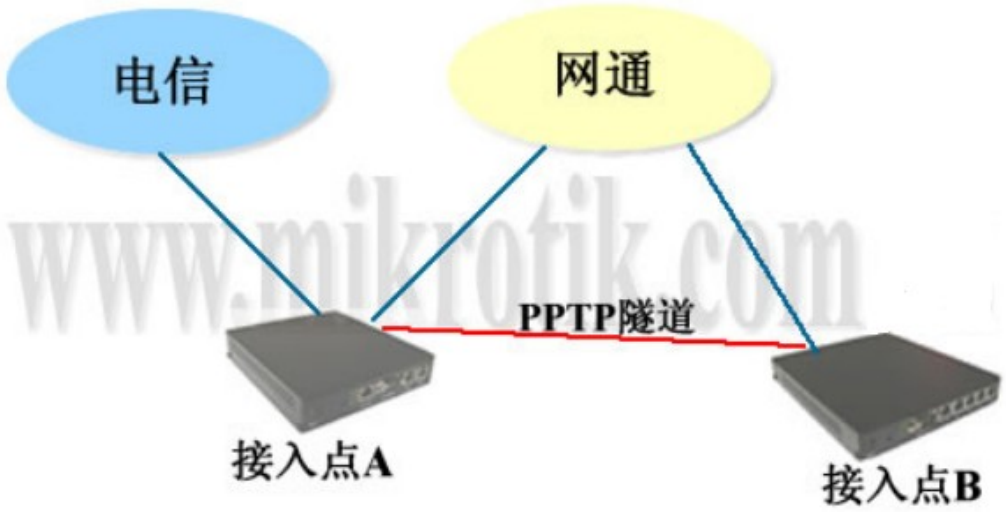
在这里，我们也可以通过/ip route rule 来低钠盐端口的规则：



让定义的 WEB 标记在一次回到 web 路由表中区查找网关。

PPTP 借线路路由操作

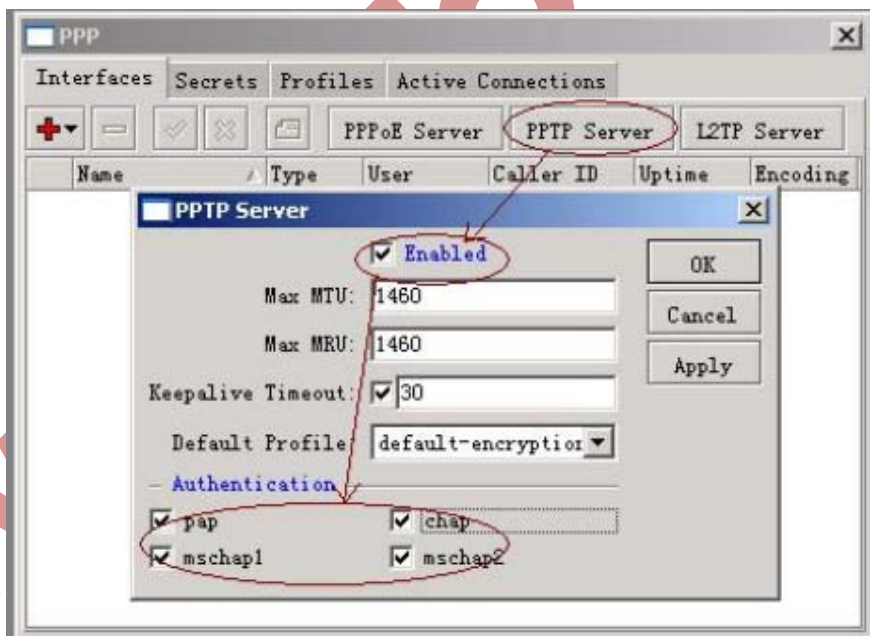
假设一个接入点 A 有电信和网通两条线路，并做了一网通为主，电信为静态路由策略设置。而另一个接入点 B 接线路，并想通过 PPTP 隧道的方式借用接入点 A 的电信线路，现在看下面的图列：



根据上面的案例，接入点 A 和 B 他们都是共同使用了网通的线路，这里网通两个点之间的延迟小于 10ms，网络延迟小才能保证足够的网速给 B 做电信的访问。首先建立从接入点 B 到 A 的 PPTP 隧道，我们在接入点 A 设置 PPTP 服务器，在接入点 B 设置客户端。这里接入点 A 的网通 IP 地址为 202.112.12.10，B 网通地址为 202.112.12.12.

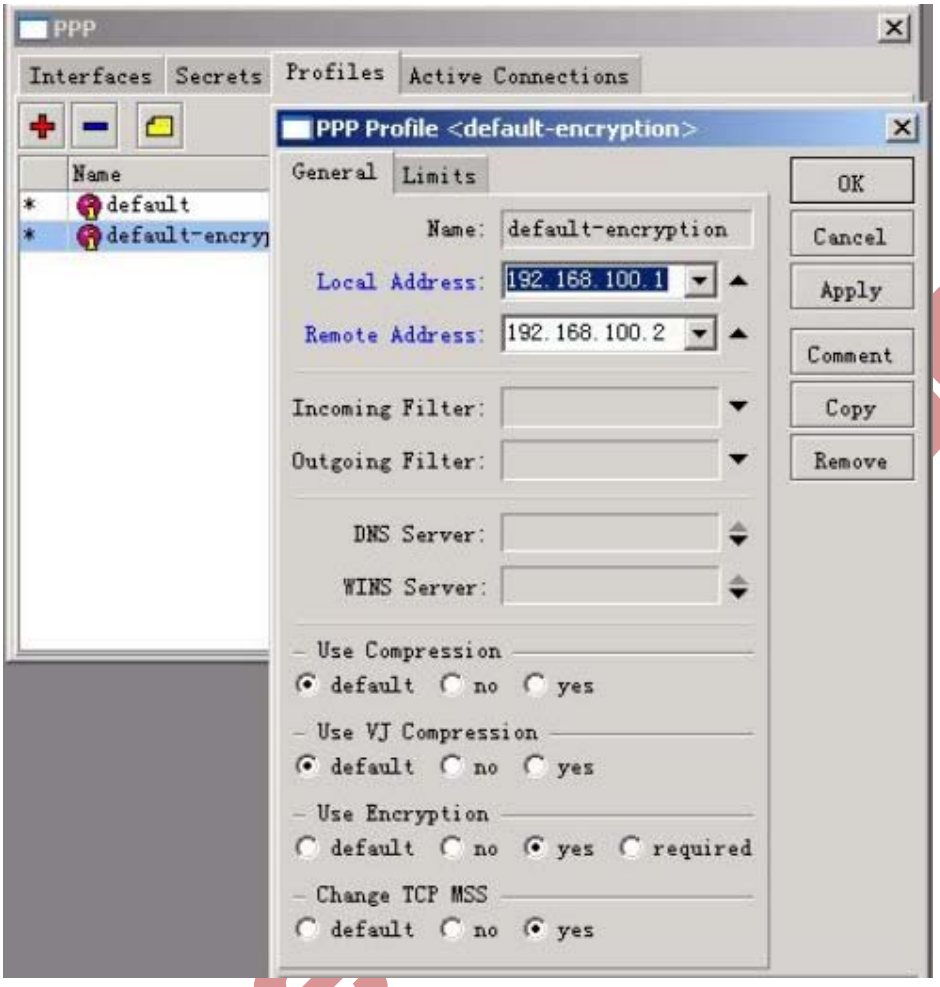
配置 PPPTP-server

在接入点 A 启用 PPPTP－server,并设置密码传输的加密类型：



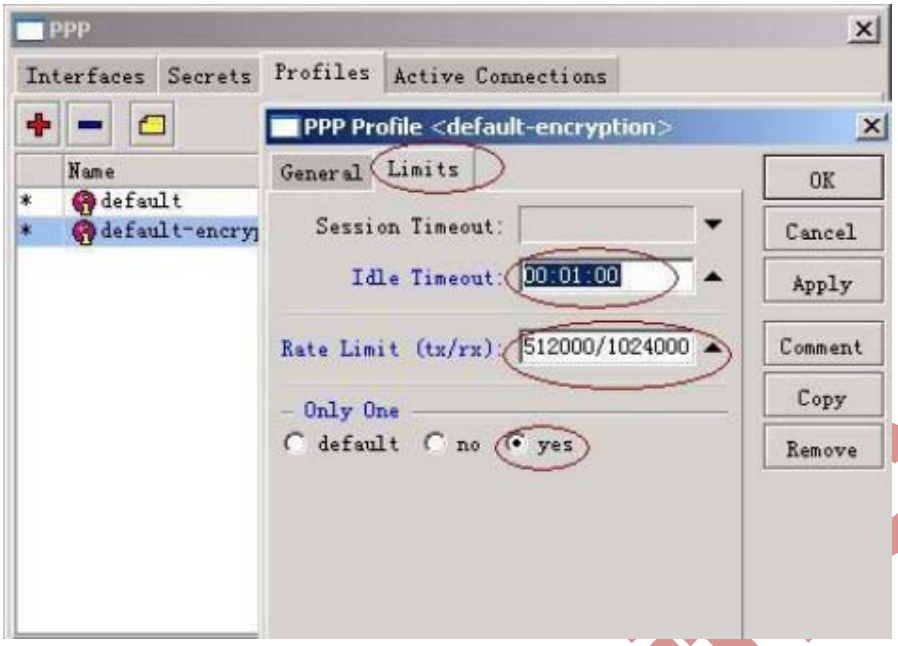
在这里 default-profile 我们采用 default-encyption.同样你也可以在 PPTP-server 的 profile 中创建自己的规则。Keepalive-timeout 是 pptp-server 主动使用 ICMP 协议探测客户端是否在线，如果客户端使用了防火墙或禁止 ICMP 探测，那无法探测到客户端，server 就会主动断开该客户端的连接，这个设置需要客户自己根据网络情况判断。

设置 profile 定义客户和主机的访问地址：



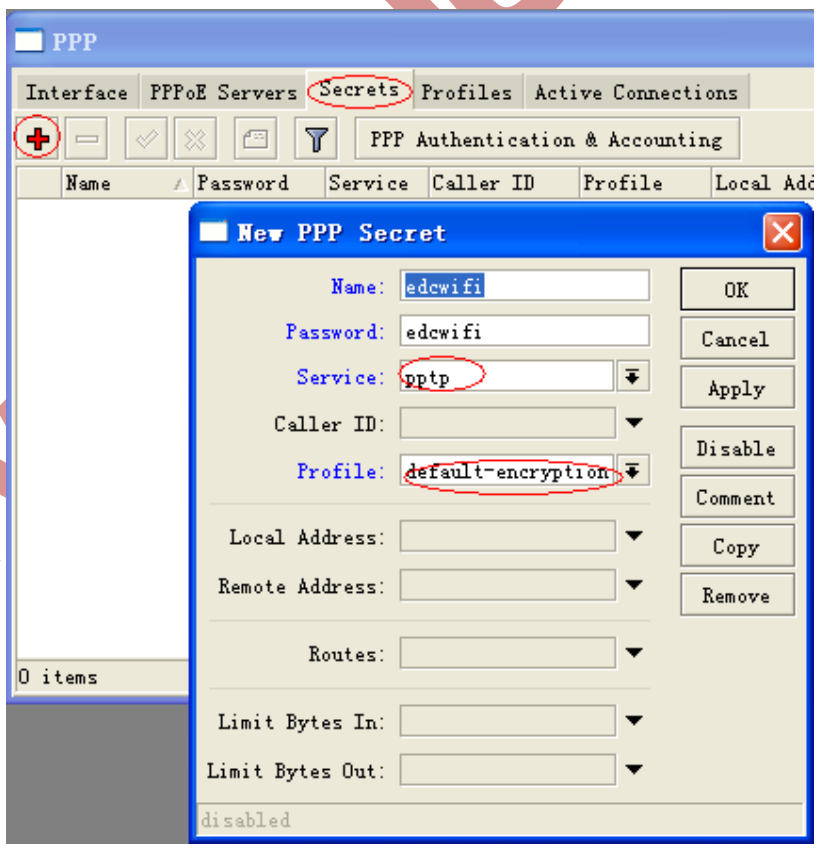
在这里我们给 PPTP－server 分配的 ip 地址为：192.168.100.1（local-address）,给客户端分配的地址为：192.168.100.2（remote-address）分配 ip 地址可以通过帐号设置 secrets 进行，在这里我们只有一个客户端所有可以直接通过 profile 中的规则设置，如果有多个客户端也可以通过/ip pool 中的地址池做 DHCP 的分配。

配置 limit 参数：



在 limit 参数中，我们可以看到 idle-timeout,这个客户端在没有流量超过 1 分钟后，就断开客户端，rate-limit 是对该类用户的流量控制这里设置的上行为 512K，下行为 IM 的带宽。最后是 only-one 该帐户是否为唯一，这里设置为 yes.

设置客户端的帐号密码：

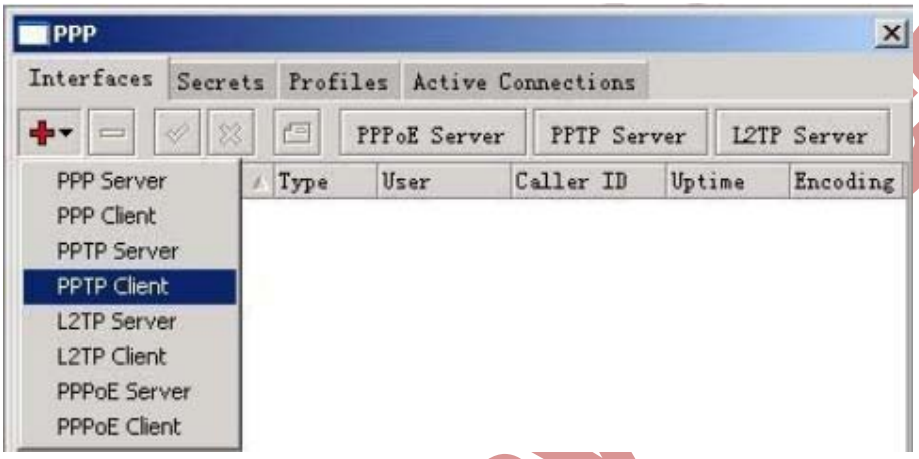


[键入文字]

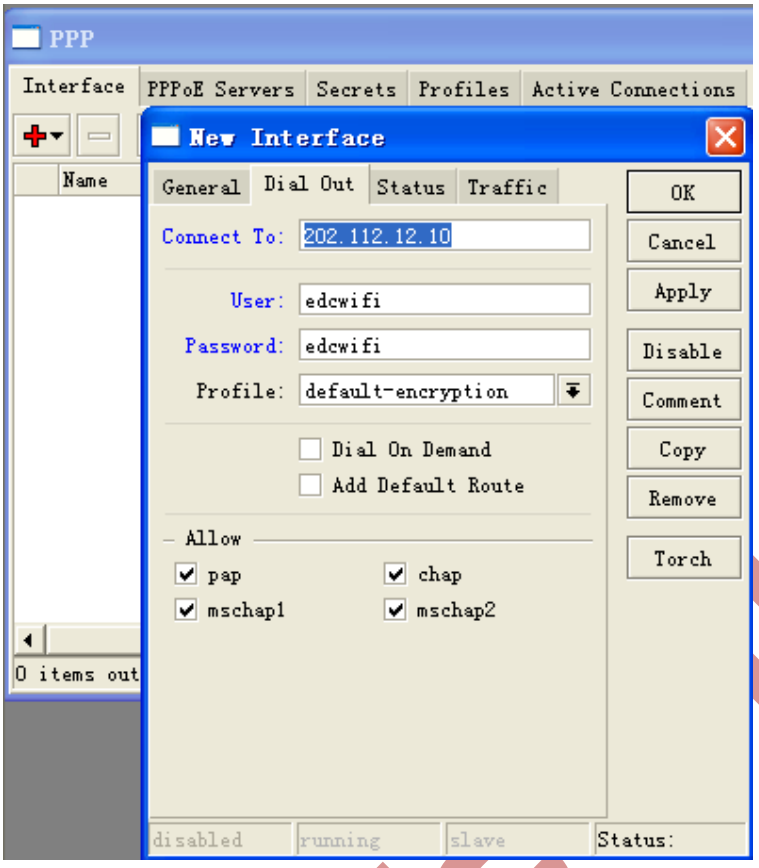
进入 secret 设置帐号和密码以及相关信息，设置好 name 和 password 后，选择 service 服务类型为 pptp,profile 规则为 default-encryption.这样 pptp-server 就已经设置完成。

配置 PPTP-client

完成 PPTP 服务设置后，现在开始设置接入点 B 的 PPTP－client,进入 PPP 选项添加 PPTP-client:



进入 dial-out 设置 PPTP 拨号信息，在 server-address 的地址为 202.112.12.10 级接入点 A 的网通地址：

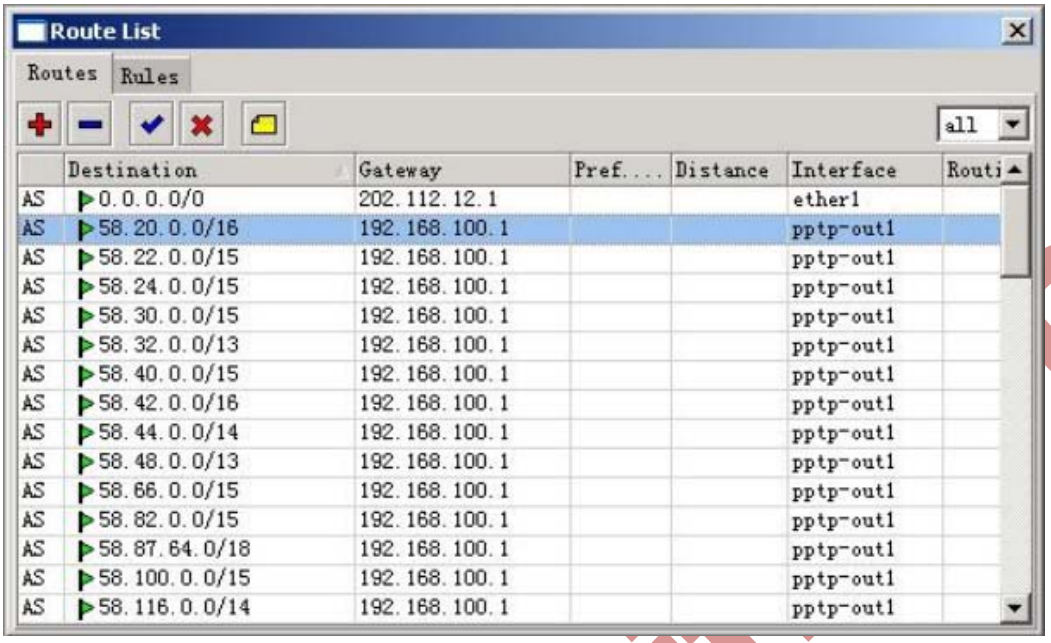


设置账号和密码分别为，设置完成后，便可以与接入点 A 的 PPTP-server 连接。

路由配置

在这里接点 A 和 B 都做了 IP 地址的 NAT 转换，并接点 A 已经做了电信的静态路由规则，即 A 点可以实现访问网通和电信的分流，在 A 点不需要在做任何设置。B 点就需要指定通过 AB 两点间的 PPTP 隧道到电信的线路，他指定的网关为 A 点的 PPTP 的 IP 大作（192.168.100.1）

设置电信访问的网关：



通过编辑电信的路由脚本， 导入路由表中，则实现了通过 PPTP 隧道使用 A 接入点的电信线路，完成了借线功能。

RouterOS 路由表操作原则

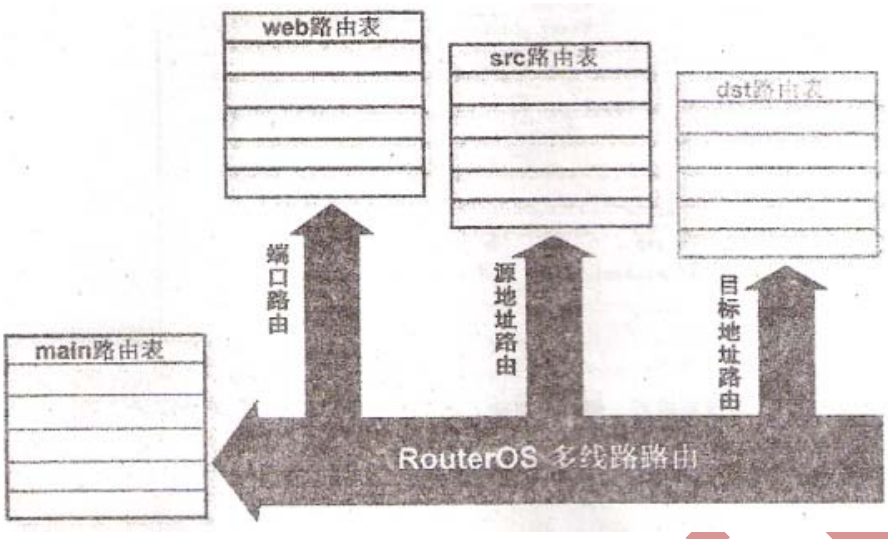
routerOS 能维护多个独立的理由表，能灵活的分配策略路由规则，通过下面的操作命令可以标记路由与定义路由策略表。

/ip filewall mangle mark-routing(支持源目标和端口路由)
/ip route routing-mark(支持源目标路由)
/ip route rule table (支持源目标路由)

他们之间关系式平等的：

Mark-routing=routing-mark=table

当他们被定义后都会在 ip route 中新建路由表，如图：



如果建立了多个路由表，routeOS 会首先处理新建的路由表，最后剩下的数据到 main 表，注意：在 IP route rule 中的规则是从上往下的执行，最上规则优先执行。

第 6 章 DHCP 操作配置

DHCP(动态主机分配协议)，负责分配和接收网络中的 IP 大作信息，能让网络内的主机动态获取地址，连接指定的网络，routerOS 支持服务端（server）和客户端（client）,同时支持 DHCP-relay 接力传输功能。

DHCP-client 设置

操作路径：/ip dhcp-client

Mikrotik routerOS DHCP- client 在一个以太网上启用，client 将接受一个地址，子网掩码，默认网关和两个 DNS 服务器地址。收到的 ip 和子网掩码将添加到选择的网卡上，默认网关将添加到路由表的中的动态项目，如果 DHCP-client 被禁用或没有更新一个地址，动态路由将自动删除。

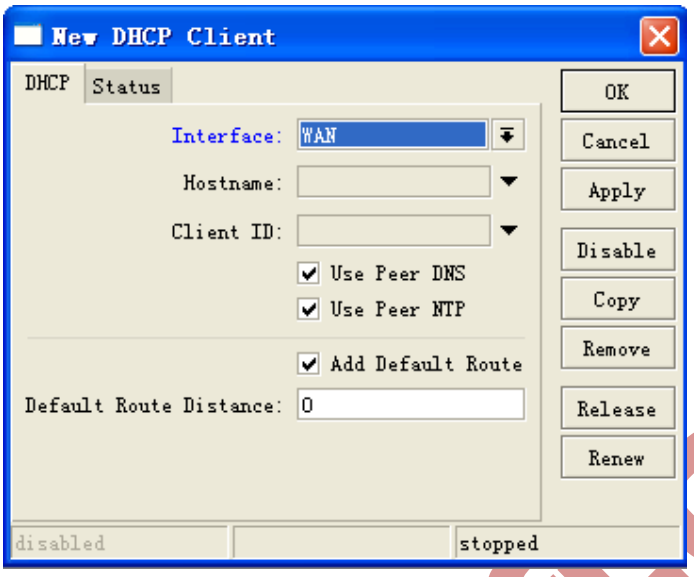
属性描述

Add-default-route(yes/no;默认：yes)是否添加指定的 DHCP 服务器的默认路由
Client-in(文本)与 administraor 或 ISP 相符合的参数
Enabled(yes/no;默认：no)是否启用 DHCP 客户端
Host-name(文本)客户端的主机名
Interface(名称，默认：（unknown）任何以太网 interface（这包括 wireless 和 Eoip 隧道）
Use-peer-dns((yes/no;默认：yes)是否接受 DHCP 服务器的 DNS 的设置（将会添加到/ip dnc 中）

命令描述

Renew 更新当前的租约，如果更新操作没有成功，客户端将试着初始化租约。
事例：在 ether1inferface 启用 DHCP-client：

{admin@mikrotik}ip dhcp-client>set enabled=yes interface=wan
{admin@mikrotik}ip dhcp-client>print
Enabled:yes
Interface:wan
Host-name:””
Client-id:””
Add-default-route:yes
Use-peer-dns:yes
{admin@mikrotik}ip dhcp-client>
Winbox 操作：



DHCP-server 设置

指令名称：/ip dhcp-server setup

属性描述

Dhcp server interface (名称)运行 DHCP 服务器的 interface

DHCP address space（ip 地址/掩码；默认：192.168.0.0/24）dchp 服务器将出租给客户端的网络地址段

Gateway（ip 地址；默认：0.0.0.0）分配给客户端的网关地址

DHCP relay（ip 地址；默认：0.0.0.0）在 DHCP 服务器与 DHCP 接力的 ip 地址

Addresses to give out （文本）DHCP 服务器分配给客户端的 ip 地址

Dns servers(ip 地址)分配给 DHCP 客户端的 DNS 服务器地址

Lease time (时间；默认：3d)使用租期时间

事例：配置 DCHP 服务器在 ether1 interface 上，并分配给 10.0.0.2 到 10.0.0.254 的网络地址段。设置网关为 10.0.0.1，DNS 服务器为 159.148.60.2，租约时间为 3 天：

{admin@mikrotik}ip dhcp-server>setup

选择 DHCP 服务器运行的 interface

DHCP server interface:ether1

选择 DHCP 网络地址段

Dhcp address space:10.0.0.0/24

设置网关地址

Dateway for dhcp netwok:10.0.0.1

选择 ip 地址池给 DHCP 服务器

Addresses to give out:10.0.0.2-10.0.0.254

设置 DNS 服务器

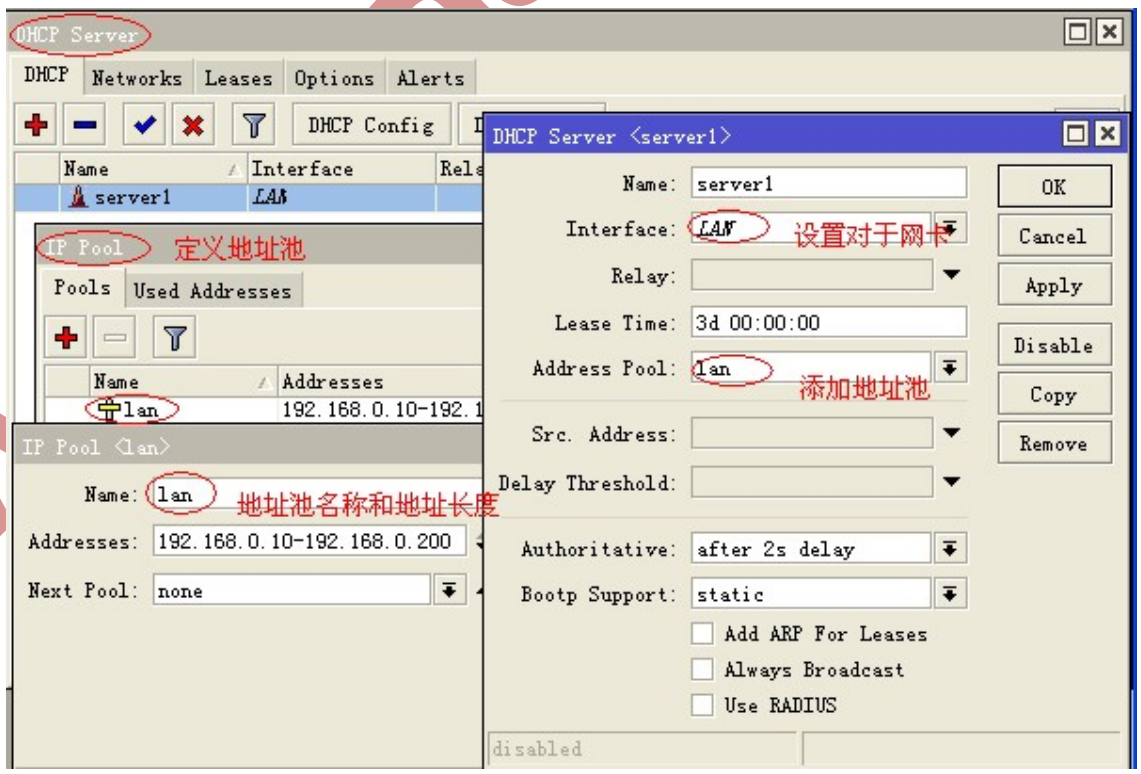
Dns servers:159.148.60.2
设置租约时间

Lease time:3d
{admin@mikrotik}ip dhcp-server>

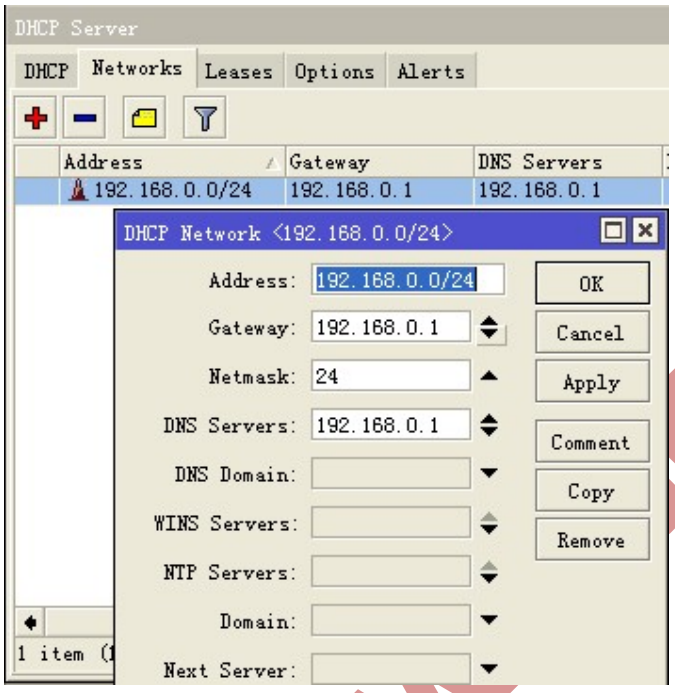
在上面向导中设置的内容，通过命令查看如下：

[admin@mikrotik] ip dhcp-server> print
Flags: X — disabled, I — invalid
NAME INTERFACE RELAY ADDRESS-POOL LEASE-TIME ADD-ARP
0 dhcp1 ether1 0.0.0.0 dhcp_pooll 3d no
[admin@mikrotik] ip dhcp-server> network print
ADDRESS GATEWAY DNS-SERVER WINS-SERVER DOMAIN
0 10.0.0.0/24 10.0.0.1 159.148.60.2
.[admin@mikrotik] ip dhcp-server> ip pool print
NAME RANGES
0 dhcp_pooll 10.0.0.2-10.0.0.254
[admin@mikrotik] ip dhcp-server>

Winbox 操作：添加 DHCP 服务，在 ip pool 中分配地址池，并指定 LAN 口上：



配置主机网络信息：



第七章 DNS 配置

DNS 缓存食使用最小的 DNS 请求时间连接到外部的 DNS 服务器，这相当于一个简单的本地 DNS 服务。

需要功能包：system
需要等级：level1
操作路径：/ip dns

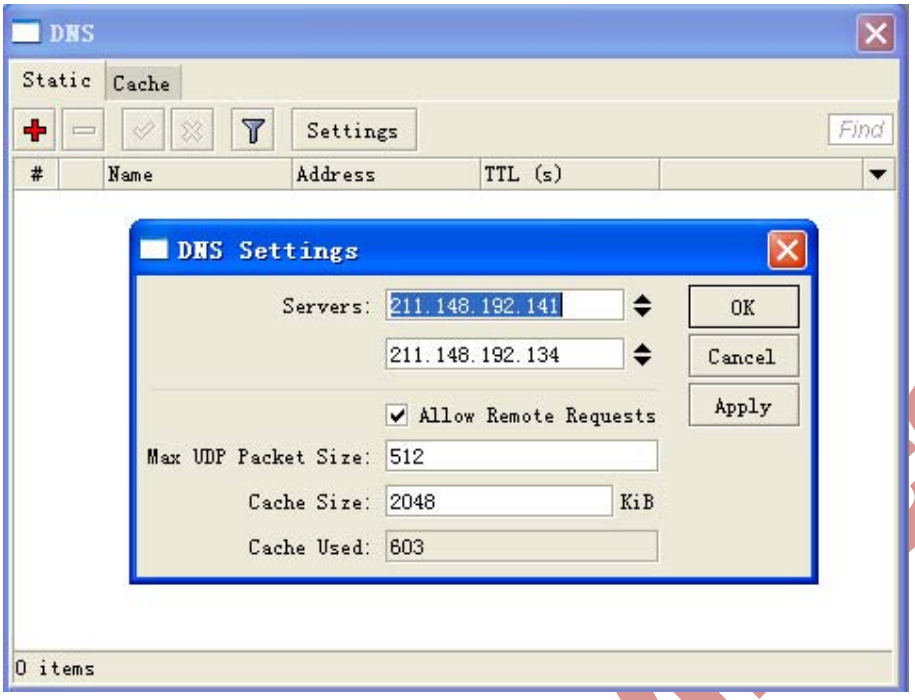
属性描述

Allow-remote-requests(yes/no)是否允许指定运程网络的请求
Primary-dns（ip 地址；默认：0.0.0.0）首选 DNS 服务器
Secondary-dns（ip 地址；默认：0.0.0.0）备用 DNS 服务器
Cache-size(整型：512…10240；默认：2048KB)指定 DNS 缓存的长度单位为 KB
Cache-max-ttl(时间；默认：7d)指定缓存记录的最大存活周期
Cache-used(只读：整型)显示当前使用的缓存大小 KB

注：如果/ip dhcp-client 属性下的 use-peer-dns 设置为 yes,这时./ip dns 下的 primary-dns 将会改变，并修改 DHCP 服务的 DNS 设置。

事例：设置首选 DNS 服务器为 159.148.60.2

```
[admin@mikrotik] ip dns > set primary-dns=159.148.60.2
{admin@mikrotik}ip dns > print
Resolve-mode: remote - dns
Primary-dns:159.148.60.2
Secondary-dns:0.0.0.0
[admin@mikrotik] ip dns >
```



缓存状态

操作路径：/ip dns static
Name(只读：名称)主机的 DNS 名称
Address（只读：ip 地址）主机 ip 地址
Tti(时间)剩余的存活周期

内部 DNS 域名解析

操作路径：/ip dns static

Mikrotik routerOS 在 DNS 缓存中进入 DNS 服务器的一些特征，如通过使用路由器的 DNS 作域名解析 ip 地址。

属性描述

Name(文本)分配给 ip 地址的 DNS 名称

Address（ip 地址）分配给域名的 ip 地址

事例：为 www.edcwifi.com 域名添加静态 DNS，ip 地址是 10.0.0.1

```
[admin@mikrotik] ip dns static> add name www.edcwifi.com    address=10.0.0.1
[admin@mikrotik] ip dns static> print
#     NAME                              ADDRESS                           TTL
```

0 aaa.aaa.a 123.123.123.123 1d
1 www.edcwifi.com 10.0.0.1 1d
[admin@mikrotik] ip dns static>

刷新 DNS 缓存

操作指令：/ip dns cache flush

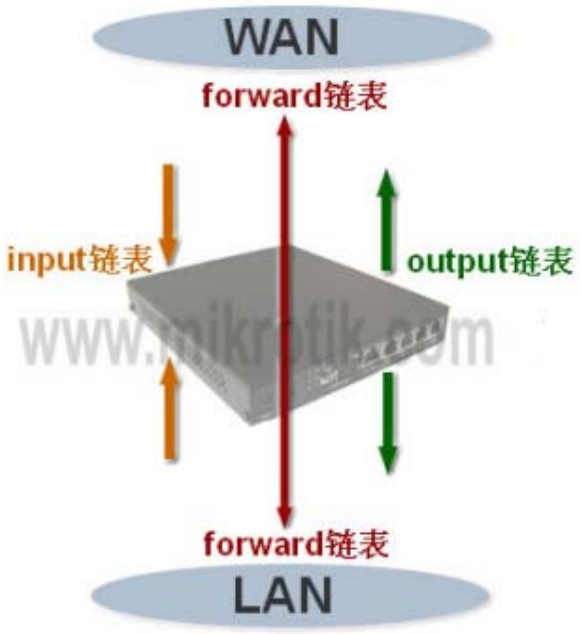
Flush-清徐内部 DNS 的缓存 clears internal DNS cache

[admin@mikrotik] ip dns> cache flush
[admin@mikrotik] ip dns> print
Primary-dns: 159.148.60.2
Secondary-dns: 0.0.0.0.
Allow-remote-requests:no
Cache-size:2048 KB
Cache-max-ttl:7d
Cache-used:10KB
[admin@mikrotik]ip dns>

第八章 防火墙过滤（firewall filte）

在 routerOS 通过 ip firewall 能对 ip 数据包过滤，源和目标 ip。端口。Ip 协议。协议（ICMP,TCP,MSS 等），网络接口，对内部的数据包河连接作标记，TOS 字节，内容过滤，顺序优先于数据频繁和时间控制，包长度控制….

从数据传输上分类：分为 input，foreward 和 output 三种链表（chain）过滤，不管是二层或者三层过滤上都包含着三个链表。



快速设置向导

添加一条 firewall 规则，将所有通过蓝牙器到目标协议为 TCP,端口为 135 的书籍包丢掉：
/ip firewall filter add chain=forward dst-port=135 protocol=tcp action=drop

拒绝通过 telnet 访问路由器（协议 TCP,端口 23）：
/ip firewall filter add chain=forward protocol=tcp dst-port=23 action=drop

Firewall 过滤

操作路径：/ip firewall filter
网络防火墙始终保持对那些有威胁敏感的数据进入内部网络中，不管怎样网络都是连接在

一起的，总是会有某些从外入你的 LAN。窃取资料和破坏内部网络。适当的配置防火墙可以保护网络。

Mikrotik routerOS 是功能非常强大的防火墙。包括以下特征：
包过滤功能
P2P 协议过滤
7 层协议过滤
Ipv6 防火墙过滤
数据传输分类：
源 MAC 地址
Ip 地址（网段或列表）和地址类型（广播，本地，组播）
端口和端口长度
Ip 协议
协议选择选项（ICMP 类型和代码字段。TCP 标记，ip 选项和 MSS）
Interface 的数据包从那里到达或同那里去
内部数据流于连接标记
TOS(DSCP)byte
数据包内容 packet content
Rate at which packets arrive and sequence nambers
数据包大小
包到达时间

基本过滤规则

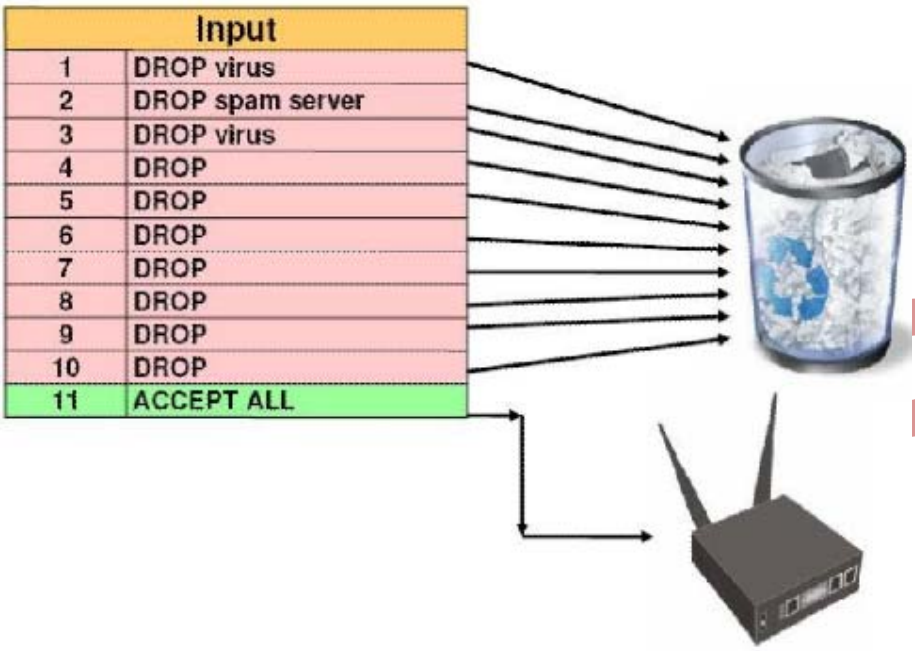
防火墙操作是借助于防火墙的策略，一个策略是告诉路由器如何处理一个 ip 数据包决定，每一条策略都由两部分组成，一部分是传输状态配置和定义如何操作数据包。数据链（chains）是为更好的管理组织策略。
过滤功能有三个默认的数据链（chains）：input，forward 和 output 他们分别负责从哪里进入路由器的，通过路由器转发的与从路由器发生的数据。用户也可用自定义添加链，当然这些链没有默认的传输配置，需要在三条默认的链中对 action=jump 策略中相关的 jump-target 进行配置。

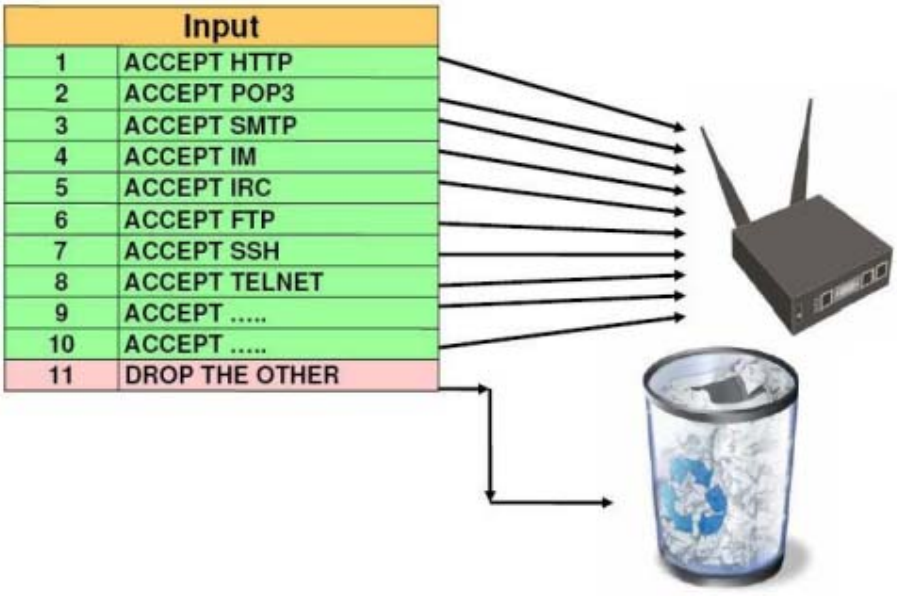
过滤链

下面是三条预先配置好了的 chains，他们是不被删除的：
Input 用于处理进入路由器的数据包，即数据包目标 ip 地址是到达路由器一个接口的 ip 地址，经过路由器的数据包不会在 input-chains 处理。
Forward 用于处理通过路由器的数据包
Output 用于处理源于路由器并从其中以个借口出去的数据包

当处理一个 chain（数据链），策略是从 chain 列表的顶部从上而下执行的，如果一个数据包满足策略的条件，这时会执行稿操作。

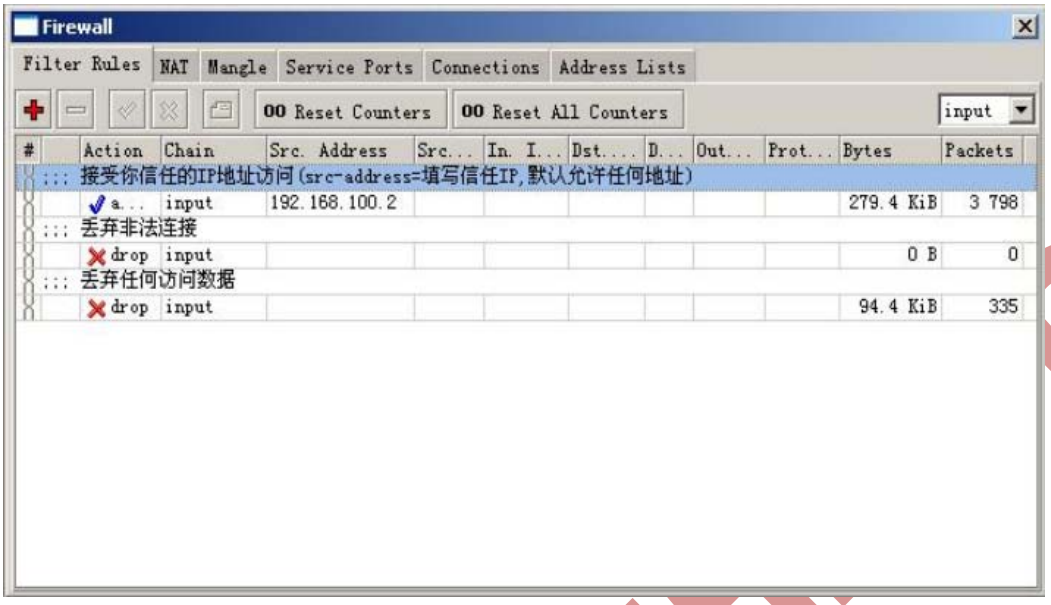
我们来看看防火墙过滤原则：





防火墙规则事例

我先从 input 链表开始，这里是对所有访问路由的数据进行过滤和处理：



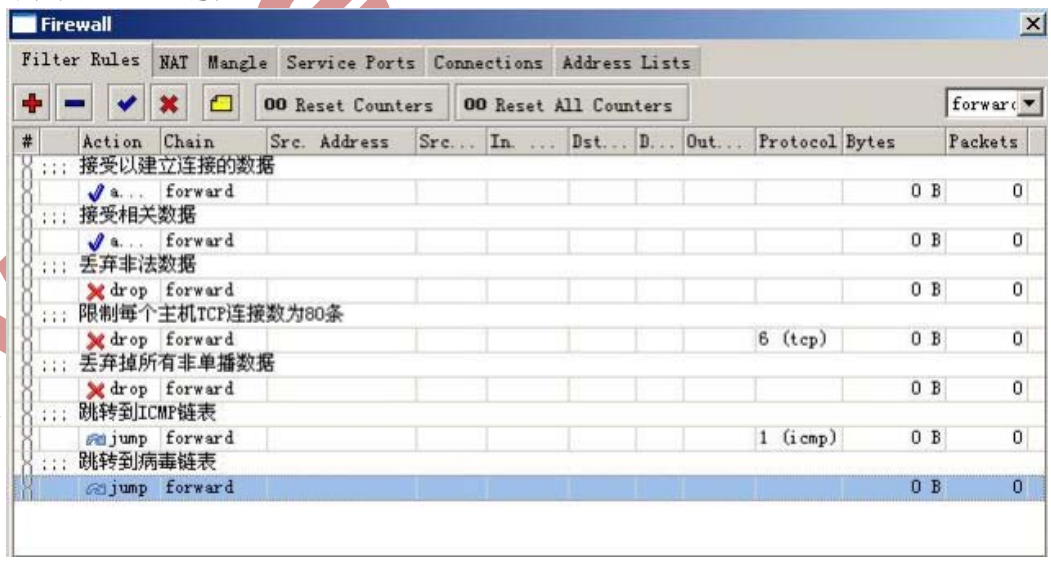
从 input 链表的第一条开始执行，这里一共有三条规则：

0 ;;; 接受你信任的 ip 地址访问（src-address=填写信任 ip，默认允许任何地址）
Chain=input src-address=192.168.100.2action=accept

1 ;;; .丢掉非法连接
Chain=input connection-state=invalid action=drop

2 ;;; 丢弃任何访问数据
Chain=input action=drop

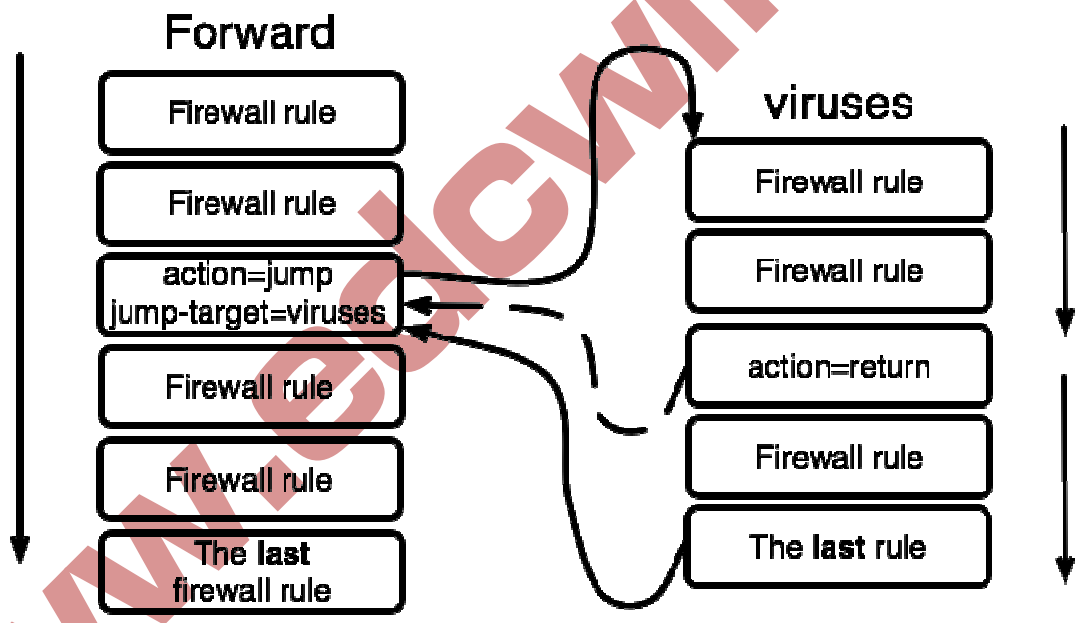
下面是 forward 链表：



Forward 链表，一共有 7 条规则，包括两个跳转到自定义链表 ICMP 和 virus 链表：

0 ;;; 接受已建立连接的数据
Chain=forward connection-sate=established action=accept
1 ;;; 接受相关数据
Chain=forward connection-sate=invalid action=drop
2 ;;; 丢弃非法数据包
Chain=forward connection-sate=established action=accept
3 ;;; 限制每个主机 TCP 连接数位 80 条
Chain=forward protocol=tcp connection-limit=80.32 action=drop
4 ;;; 丢弃掉所有非单播数据
Chain=forward src-address-type=lunicast action=drop
5 ;;; 跳转到 icmp 链表
Chain=forward protocol=icmp action=jump jump-target=icmp
6 ;;; 跳转到病毒链表
Chain=forward action=jump jump-target=virus

Forward 工作过程如下：



在自定义链表 icmp 中，是定义所有 icmp（Internet 控制报文协议），icmp 经常被认为是 ip 层的一个组成部分，他专递差错报文以及其他需要注意的信息，icmp 报文通常被 ip 层或更高层协议（TCP 或 UDP）使用，例如：ping,teaceroute,trace TTL 等。我们通过 icmp 链表来过滤所有的 icmp 协议：

Icmp 链表操作过程：

0 ;;; ping 应答限制为每秒 5 个包
Chains=icmp protocol=icmp icmp-options=0：0-255 limit=5，5 action=accept
1 ;;; traceroute 限制为每秒 5 个包
Chains=icmp protocol=icmp icmp-options=3：3 limit=5，5 action=accept
2 ;;; MTU 线路探测限制为每秒 5 个包
Chains=icmp protocol=icmp icmp-options=3：4 limit=5，5 action=accept
3 ;;; ping 请求限制为每秒 5 个包
Chains=icmp protocol=icmp icmp-options8：0-255 limit=5，5 action=accept
4 ;;; trace TTL 限制为每秒 5 个包
Chains=icmp protocol=icmp icmp-options=11：0-255 limit=5，5 action=accept
5 ;;; 丢弃掉任何 icmp 数据
Chains=icmp protocol=icmp action=drop

ICMP 类型：代码

通过指令保护你的路由器和相连接私有网络，你需要通过配置防火墙丢弃或拒绝 icmp 协议的传输。然而以下 icmp 数据包则需要用来维护和故障判断用。

下面是 icmp 类型列表：通常下面的 icmp 传输建议被允许通过

Ping
8:0—回应请求
0:0—回应答复
Trace
11：0—TTL 超出
3：3 端口不可到达

路径 MTU 探测

3：4—分段存储 fragmentation-DF-set

一般 icmp 过滤探测

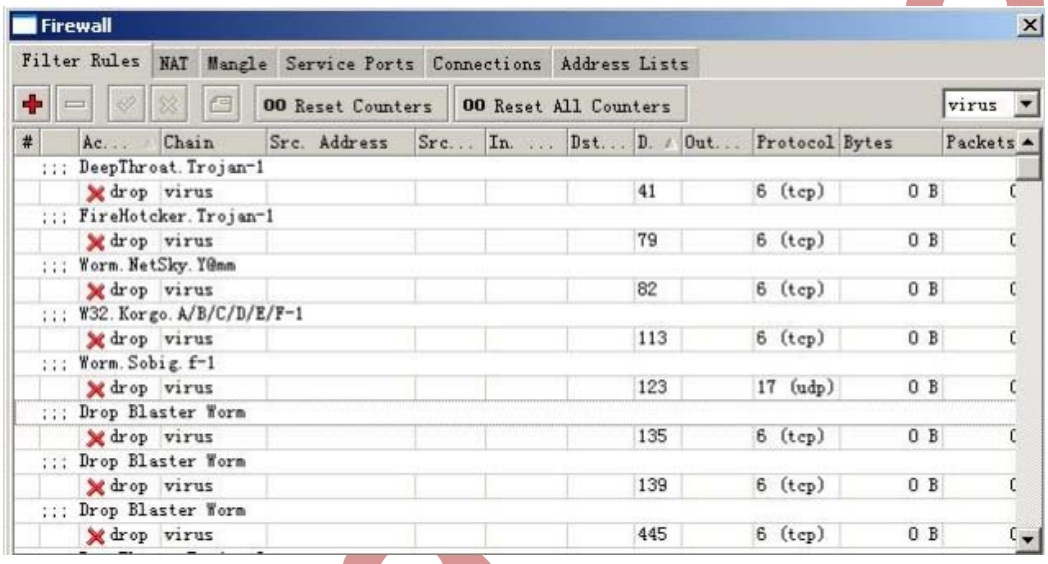
允许 ping-icmp 回应请求向外发送和回应答复进入

允许 traceroute-TTL 超出和端口不可到达信息进入

允许路径 MTU-icmp fragmentation-DF-Set 信息进入

阻止其他任何数据

在 virus 链表中过滤常见的病毒，我可以根据需要在链表中新的病毒对他们做过滤：



阻止不必要的 ip 广播：

Add chain=forward src-address=0.0.0.0/8 action=drop

Add chain=forward dst -address=0.0.0.0/8 action=drop

Add chain=forward src-address=127.0.0.0/8 action=drop

Add chain=forward dst-address=127.0.0.0/8 action=drop

Add chain=forward src-address=224..0.0.0/8 action=drop

Add chain=forward dst-address=224..0.0.0/8 action=drop

建立新的跳转数据链（chains）：

Add chain=forward protocol=tcp action=jump jump-targert=tcp

Add chain=forward protocol=udp action=jump jump-targert=udp

Add chain=forward protocol=icmp action=jcmp jump-targert=icmp

建立 tcp-chain 并拒绝一些 tcp 端口：

Add chain=tcp protocol=tcp dst-port=69 action=drop/

Comment=”deny TFTP”
Add chain=tcp protocol=tcp dst-port=111 action=drop/
Comment=”deny RPC portmapper”
Add chain=tcp protocol=tcp dst-port=135 action=drop/
Comment=”deny RPC portmapper”
Add chain=tcp protocol=tcp dst-port=139-139action=drop/
Comment=”deny NBT”
Add chain=tcp protocol=tcp dst-port=445 action=drop/
Comment=”deny CIFS”
Add chain=tcp protocol=tcp dst-port=2049 action=drop Comment=”deny NFS”
Add chain=tcp protocol=tcp dst-port=12345-12346 action=drop Comment=”deny netbus”
Add chain=tcp protocol=tcp dst-port=20034 action=drop Comment=”deny netbus”
Add chain=tcp protocol=tcp dst-port=3133 action=drop Comment=”deny backoriffice”
Add chain=tcp protocol=tcp dst-port=67-68 action=drop Comment=”deny DHCP”

在 UDP-chain 中拒绝非法的 UDP 端口 deny UDP ports in UDP chain：

Add chain=tcp protocol=tcp dst-port=69 action=drop Comment=”deny TFTP”
Add chain=tcp protocol=tcp dst-port=111action=drop Comment=”deny PRC portmapper”
Add chain=tcp protocol=tcp dst-port=135 action=drop Comment=”deny PRC portmapper”
Add chain=tcp protocol=tcp dst-port=137-139 action=drop Comment=”deny NBT”
Add chain=tcp protocol=tcp dst-port=2049 action=drop Comment=”denyNFS”
Add chain=tcp protocol=tcp dst-port=3133 action=drop Comment=”deny backoriffice”

在 icmp-chain 允许相应需要的 icmp 连接：

Add chain=icmp protacol=icmp icmp-options=0:0 action=accept/
Comment=”drop invalid connection”
Add chain=icmp protacol=icmp icmp-options=3:0 action=accept/
Comment=”allow established connection”
Add chain=icmp protacol=icmp icmp-options=3:1 action=accept/
Comment=”allow aleady established connections”
Add chain=icmp protacol=icmp icmp-options=4:0 action=accept/
Comment=”allow source quench”
Add chain=icmp protacol=icmp icmp-options=8:0action=accept/
Comment=”allow echo request”
Add chain=icmp protacol=icmp icmp-options=11:0 action=accept/
Comment=”allow time exceed”
Add chain=icmp protacol=icmp icmp-options=12:0 action=accept/
Comment=”allow parameter bad”
Add chain =icmp action=drop comment=”deny all other types”

Peer-to-peer 协议过滤

Peer-to-peer 协议即我们所说的用于主机间点对点传输 P2P，这个技术有许多优秀的应用如 Skype。但同时也带了需要的为许可的软件和媒体在网络中泛滥。影响到 http 和 e-mail 的正常使用。RouterOS 能识别大多 P2P 协议的连接，并能通过 QOS 进行过滤，丢弃所有的 P2P 协议：

```
[admin@mikrotik] /ip firewall filter> add chain=forward p2p=all-p2p action=drop
[admin@mikrotik] /ip firewall filter> add chain=forward
flags:x-disabled, I-invalid,D-dynamic
 0    chain=forward action=drop p2p=all-p2p
```

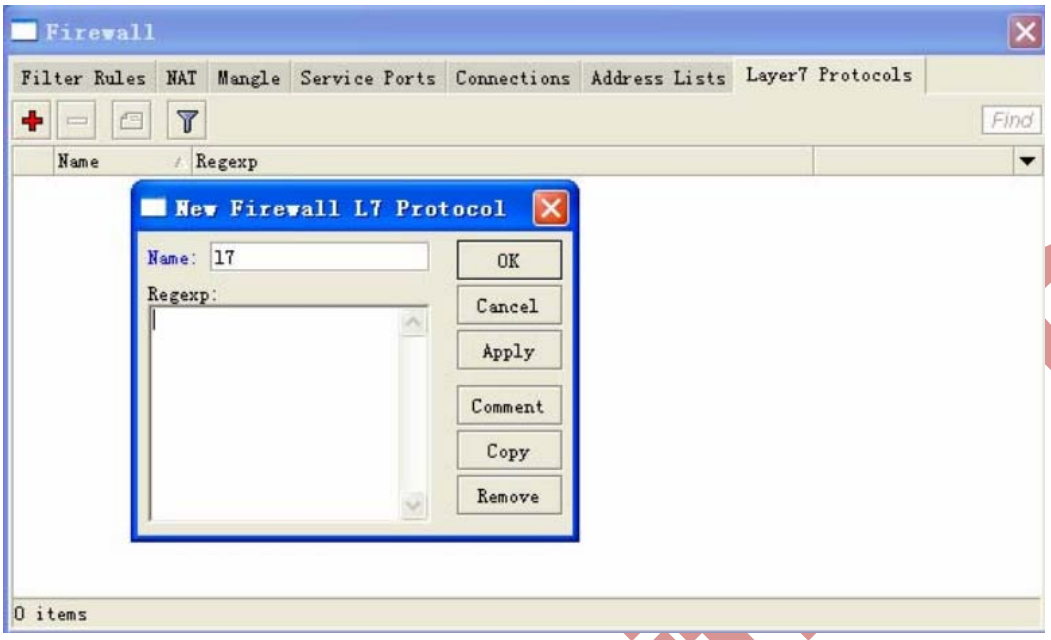
能探测到该协议的列表：

```
Fasttrack:(kazaalite,diet kazaa,grokster,imesh,giFT,poisond,mlmac)
Gnutella:(shareaza,XoLoX,Gnucleus,bearshare,limewire(jave),morpheus,phex,
Swapper,gtk-gnutella(linux),mutella(linux)qtella(linux),MLDonkey,acquisition(macOS)
Poisoned,swapper,shareaza.xolox.mlmac)

Gnutella2((shareaza, MLDonkey, Gnucleus, morpheus,adagio, mlmac)

Directconnect(Directconnect,(AKA DC++),MLDonkey,neomodus direct
connect,BCDC++,CZD
++)
eDonkey(eDonkey2000,eMuie,xMule(linux), shareaza, MLDonkey, mlmac,overnet,)
Soulseek(soulseek, MLDonkey)
BitTorrent(BitTorrent, BitTorrent++,uTorrent,Shareaza,MLDonkey,ABC,Azureus,
      BitAnarch,SimpleBT, BitTorrent,net,mlMac)
Blubster(Blubster,piolet)
WPNP(WinMX)
Warez(Warez,ares；starting from1.8.18)该协议能丢弃掉（drop），但不能被限制速度
```

RouterOS 7 层协议

RouterOSV3.在防火墙中增加了一个细胞的功能——7 层协议过滤。针对一些应用程序如 Skype，QQ，MSN，魔兽世界…..
网络程序做限制和过滤。下面介绍一下具体方法的使用：

7 层协议过滤增加在 ip filewall 中 layer7 protocols，我们可以在下面的图中看到：



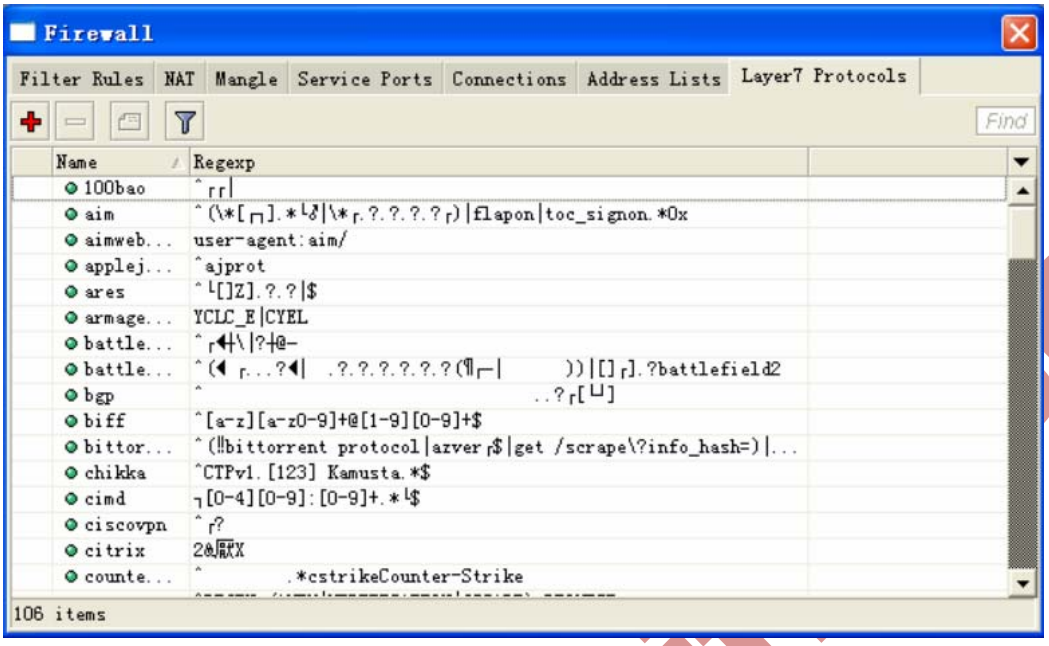
7 层协议通过 regexp 脚本编写相应 应用程序的过滤代码。Regexp 可以通过网上搜索相关资料了解，一些常用程序的 7 层协议脚本可以到我们的论坛去看看：http://mikrotikchina.cn/topic/40-l7-protos%E5%B8%B8%E7%94%A8%E8%84%9A%E6%9C%AC/

之后我们在命令（terminal）中导入 7 层协议脚本，用 import 17-protos.rsc 命令来导入脚本

```
[admin@mikrotik] >import 17 - protos.rsc
Opening script file 17-protos.rsc
script file loaded and executed successfully
[admin@mikrotik] >
```

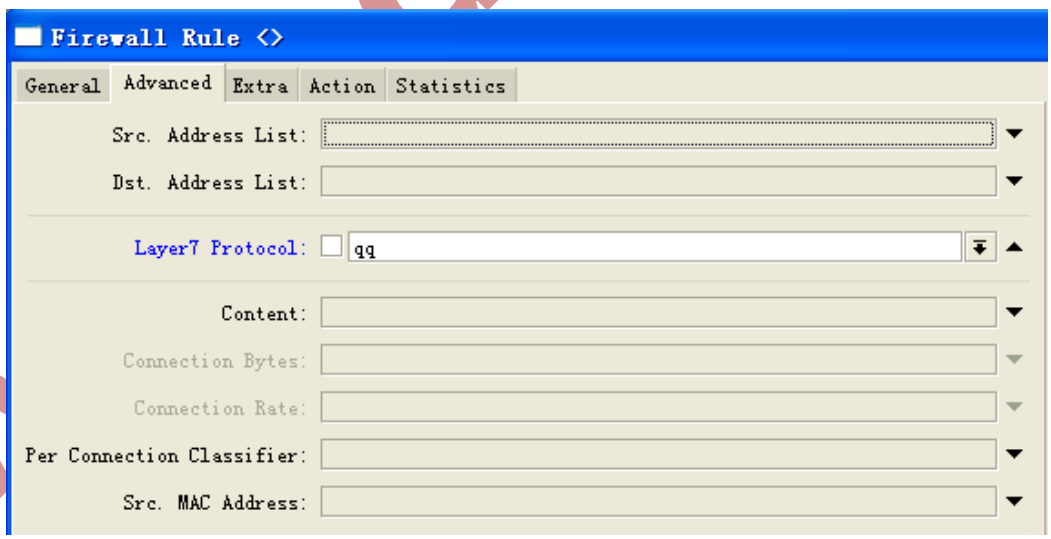
当系统提示 script file loaded and executed successful，说明脚本成功导入。

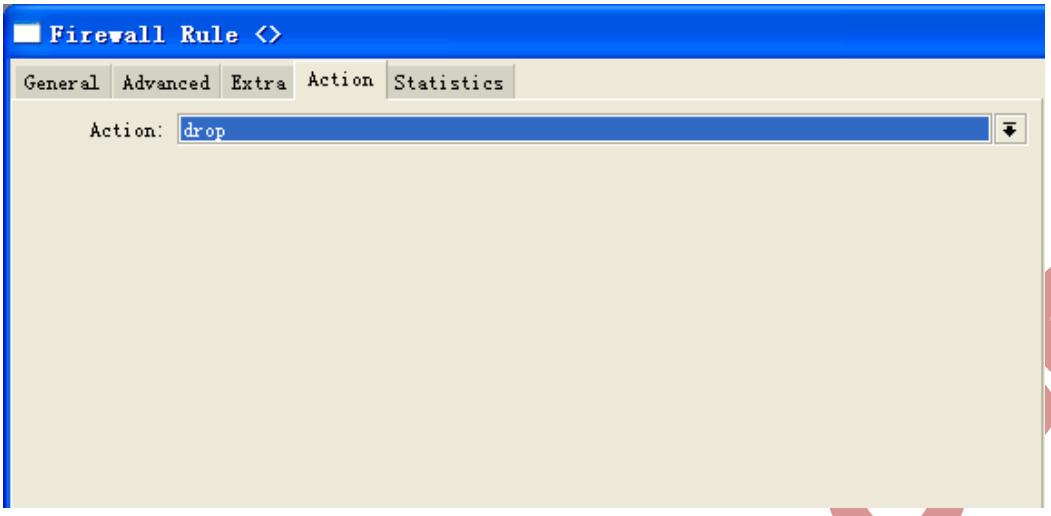
导入脚本后，我们可以在 layer7 protocol 中看到



导入后，我们就可以在 ip filewall 中通过 layer7 protocol 参数调用，并做相应的规则处理，下面是一个防火墙的 filter rules 里面调用 L7 脚本。

在这里我们通常禁止登陆 QQ 为例，在这里我们禁止所有无法登陆 QQ，添加一条规则后，进入 advanced 中的 layer7protocol 选项选择 QQ，然后在 action 中设置为 drop 丢弃，注意：L7 禁止 QQ 的规则设置好后，需要重启才能生效。





其他的操作也同以上设置类似，如果需要对 ip 地址或者 ip 段控制可以通过 src-address 或者 dst-address 进行设置。

DMZ 配置事例

DMZ 是英文 demilitarized zone 的缩写，中文名称为隔离区，也称为非军事化区。它是为了解决安装防火墙后外部网络不能访问内部网服务器的问题，而设立的一个非完全系统之间的缓冲区位于内部网络和外部网络之间的小网络区内，在这个小的网络区域内可以放置一些必须公开的服务器设施，如企业 WebFWQ ,ftp 服务器和论坛等，另一方面，通过这样一个 DMZ 区域，更加有效地保护了内部网络，因为这种网络部署，比起一般的防火墙方案，对攻击者来说又多了一道关卡。

路由器一般需要 3 张网卡（public 公网。Local 本地网络，DMZ-Zone 非军事区）：

```
[admin@gateway] interface> print
Flags: X - disabled, D - dynamic, R - running
#      NAME          TYPE       RX-RATE        TX-RATE        MTU
0   R  public        ether      0              0              1500
1   R  local         ether      0              0              1500
2   R  DMZ-zone      ether      0              0              1500
[admin@gateway]interface>
```

给相应的 interface 添加对应的 ip 地址，如下：

```
[admin@gateway] ip address> print
Flags: X - disabled, I – invalid,   D - dynamic
#   ADDRESS          NATWORK          BROADCAST          INTERACE
0   192.168.0.2/24   192.168.0.0      192.168.0.255      public
1   10.0.0.254/24    10.0.0.0         10.0.0.255         local
2   10.1.0.1/32      10.1.0.2         10.1.0.2           DMZ-Zone
```

```
3    192.168.0.3/24      192.168.0.0          192.168.0.255          public
[admin@gateway]ip address>
```

添加静态默认路由到本地路由器上

```
[admin@gateway] ip route> print
Flags: X - disabled, I – invalid, D – dynamic, J – rejected,
C - connect, S – static, r – rip, o – ospt, b - bgp
#          DST-ADDRESS        G GATEWAY          DISTANCE        INTERFACE
0   S      0.0.0.0/24          r 10.0.0.254      1               ether1
1   DC     10.0.0.0/24         r 0.0.0.0         0               ether2
[admin@gateway]ip route>
```

配置 DMZ 服务器的 ip 地址为 ip 地址 10.1.0.2，网络地址段 10.1.0.1、24，以及网关 10.1.0.1
配置能从因特网访问 DMZ 服务的 dst-nat 规则，将地址 192.168.0.3 配置给 DMZ 服务器：

```
[admin@gateway] ip firewall nat> add chain=dst-nat action=dst-nat\
\…dst-address=192.168.0.3    to-dst-address=10.1.0.2
[admin@gateway] ip firewall dst-nat> print
Flags: X - disabled, I - invalid, D - dynamic
0    china=dst-nat dst-address=192.168.0.3    action=dst-nat    to-dst-address=10.1.0.2

[admin@gateway] ip firewall nat>
```

第九章 routerOS 3.0 数据流（packet flow）

下面是 routerOS 3.0 数据流原理图，这里不可能将所有的原理放到一个图中，所以我们将原理图分解成 2 部分：

Bridging(桥接)或二层（MAC）路由的部分简化为一个 layer-3 的框
Routing or layer-3(ip)桥接的部分简化为一个 bridging 的框
Input interface - 数据包进入路由器的起点，不论什么样的接口（物理接口或虚拟接口）数据包都会从这里开始。
Output interface - 数据包离开路由器的终点，在之前运行的数据包确定被发送出路由器。
Local Process IN - 最终到达路由器自身的数据包。
Local Process OUT - 由路由器自身发出的数据包起点。

功能模块与结构

每一个功能模块在 RouterOS 中指定的目录下对应不同的功能，用户可以进入相应的目录配置属性。

Connection Tracking - /ip firewall connection tracking
Filter Input　filter forward　filter output - /ip firewall filter
Source NAT　destination NAT - /ip firewall nat
Mangle input　mangle forward　mangle output　mangle postrouting - /ip firewall mangle
Global-in global-total　global-out global-total　intertace HTB - /queue simeple 和 /queue tree
IPsec Poticy - /ip ipsec policy
Accounting - /ip accounting
Use IP firewall - /interface bridge settings 当其用桥接后，该功能才能生效，如果 use IP firewall 设置为 yes，则数据将进入 Layer-3 层处理。
Bridge input　bridge forward　bridge output - interface bridge filter
Bridge DST-NAT　bridge SRC-NAT - /interface bridge nat

主动处理

In-interface bridge - 检查输入接口是否为桥接数据（输入接口是一个端口到桥或者输入接口是桥）
Hotspot-in - 运行取传输 hotspot 特征数据，否则从连接跟踪丢弃掉。
Bridge decision - 桥通过 mac 地址列表查找数据包所匹配的目标 MAC 地址，当目标被找到，数据包将会被发送到相应的桥接口，如果没有匹配则会将数据包复制多份发送到所有的桥接端口。
Bridge decision - 这是一个变通方案，在实际桥属性判断前，允许使用 out-bridge-port
Routing decision - 路由器通过路由命令查找一个匹配数据的目标 ip 地址，当查找到后，数据包将发送到对应的端口或者路由器本身。如果在该事件中没有找到匹配路径，数据包将会被丢弃。

Routing adjustmenl- 这是一个变通的方案，允许在 mangle 中的 output 链表设置路由策略规则。

TTL=TTL-1 - 知名准确的地点，在什么时候路由包的 TTL 值被减少 1.如果 TTL 值变为 0 将会被丢弃掉。

IPsec decryption IPsec encryption - 自我处理 IPSec 判断和加密。

Out-interface bridge - 判断实际输出接口是否为桥接端口或判断输出接口是否为桥。

Hotspot-out - 撤销所有通过 Hotspot-in 的数据包操作，并发送回客户端。

第四章 带宽控制（Queue）

带宽控制是一套控制数据率分配，延迟易变性，及时转发（delivery）,可靠转发的机制。
Mikrotik routerOS 支持队列规则：
PFIFO-保先进先出
BFIFO 字节先进先出
SFQ 随机公平队列
RED 随机早先探测
PCQ 每次连接队列
HTB 等级令牌桶

规则功能包要求：system
等级要求：level1(limited to 1 queue),level3
操作路径：/queue

Queue 机制

服务质量（Qos）即路由器应该优先考虑保证数据流的质量毛病形成新的网络数据流，Qos 并非是只关于限流的，他更多的是与提供优良品质的服务相关，以下是一些 RouterOS 带宽控制机制的特征：

对特定 ip 地址，之王协议，端口以及其他参数限制数据率
限制 p2p 流量
优先考虑一些数据包流
为更快的 WEB 浏览使用列队脉冲串
对固定的时间间隔执行队列
在用户间平等的或者根据通道负担共享可用流量
队列应用在通过路由器真实接口的数据包上（比如：队列应用在向外的接口，像业务流）或者三个添加的虚拟接口中的如何一个或几个（global-in ,global-out,global-total）.
Qos 是通过掉包的方法工作的，被丢掉的包会被再次发送以防止丢弃了 TCP 协议，所以没必要担心会丢失 TCP 信息，用于描述网络应用的 Qos 等级的术语有：
Queuing disciping(qdisc)一个保存并维护队列包的算法，它指定了向外的数据包（也就是说队列规则可用对包再排序）以及在没有空间的情况下那些包需要丢弃。
CIP(committed information rate)约定好了的数据率，即通信量数据率，在不超过这个值的时候应该总是被转发
MIR(Maximal information rate)路由器可以提供的最大数据率
Priority –流量将处理的重要性顺序。你可以先设置优先级以便一些数据流可以在其他数据流之前被处理
Contention ratio 定义的数据率在用户中共享的比率（当数据率分配给许多用户时）正是用的数量拥有应用于她的简单速度限制。例如：连接比率是：1：4，即分配的数据率会在最大 4 个用户共享。

数据包在从接口发送之前会用对列规则进行处理。默认地，队列规则在物理接口的/queue interface 设置（对于虚拟接口没有默认的规则）。一旦我们对物理接口添加了一个队列（在/queue interface 定义的默认队列，对于特定接口将被忽视。就是说，当一个包没有匹配任何过滤器时，它将被发送到带有最高优先权的接口。

调度机和成型机 qdiscs
我们按照对业务流的影响分类队列规则如下：
调度机（schedulers）队列规则只根据他们的算法对数据包进行重新调度并丢弃在队列中不匹配的数据包，调度机队列规则包括：PFIFO,BFIFO,SFQ,PCQ,RED.
成型机（shapers）队列规则也履行限制规则，成型机有 PCQ 及 HTB。

虚拟接口

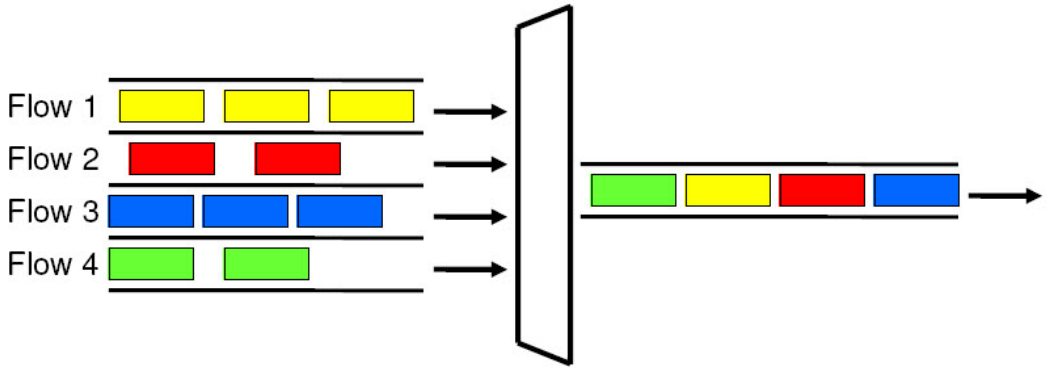
RouterOS 对实际接口增加了三个虚拟接口：
global-in 代表了所有普通的输入接口（INGRESS 队列），请注意在数据包过滤前与 giobal-in 相关的队列应用到路由器接的数据流。Global-in 排序就是在 manle 和 dst-nat 之后执行。
Global-out-代表了所有普通的输出接口。附属于他的队列会在附属于特定接口的队列之间应用。
Global-total-表了一个流经路由器的数据都能通过的虚拟接口。当把一个 qdisc 附属到 global-total 时，限制需要在两个方向起作用。例如，如果我们设置一个为 total-max-limit256000 限制，我们将得到
Upload+download=256kbps(最大值)

队列类型（Queue type）

操作路径：/ Queue type
在这个子目录你可以创建之间的客户队列类型，之后，将可以在/ Queue tree,/queue simple 或 queue interface 使用了

PFIFO 及 BFIFO

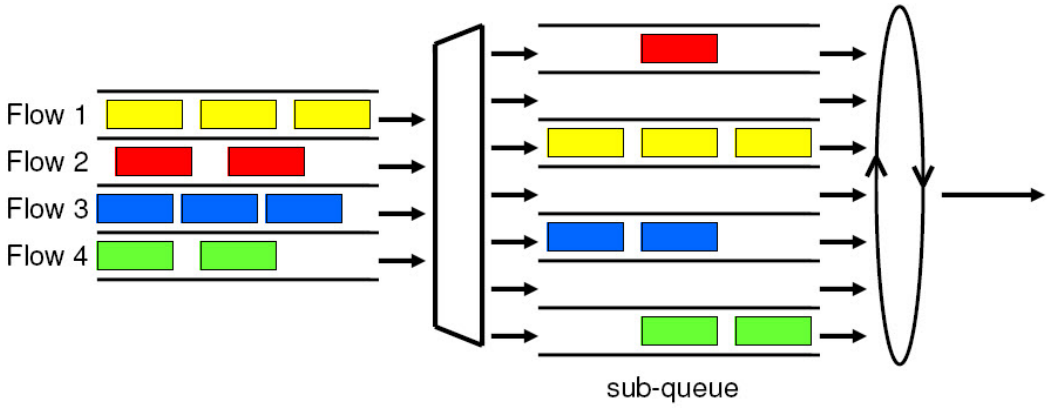
这些队列规则是基于先进出算法的（FIFO,FIST-IN FIRST-OUT）.PFIFO 和 BFIFO 的区别在于一个是以数据包为单位衡量的，而另一个是以字节为单位。其中只有一个叫做 pfifo-limit(bfifo-limit)的参数，它是用来定义一个 FIFO 队列可以容纳多少数据的，每个不能排队（如果排队满了）的包都会被丢弃，队列长度过大会增加执行时间。



如果你的连接不用塞的话，建议你使用 FIFO 队列规则。

SFQ

随机公平排序(SFQ)不会限制流量，它的宗旨是当你的连接完全满的时候均衡业务流（TCP）会话或者 udp 流，）
SFQ 的个公平性是由散列法和 round-robin 算法保证的，散例如算法把会话流分成一个有限数量的只队列。在 sfq-pertub 时间之间散列算法改变分会话流为其他只队列，round-robin 算法把从每个子队列的 pcq-allot 字节按照顺序出队列。

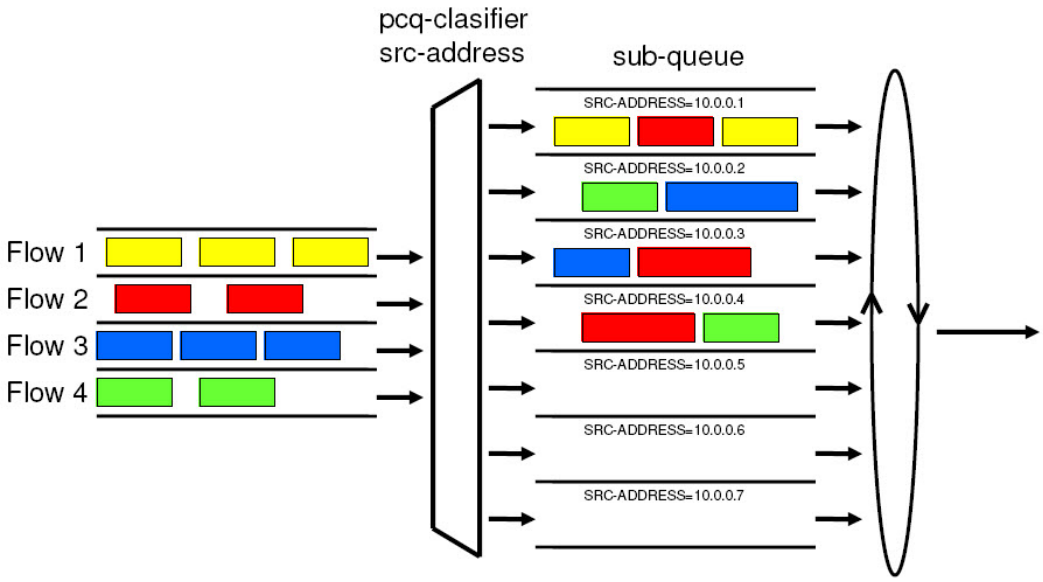


整个 SFQ 队列可以容纳 128 个数据包并且对这些包有 1024 个子队列可用，对拥挤的连接使用 SFQ 可以保证一些连接不至于空等待（STARVE）

PCQ

为了解决 SFQ 的不完美，每次连接排序 per connection queuing (PCQ)便产生了。它是唯一一种能限流的无等级排序类型。它是一个去掉了随机特性的进化版 SFQ，PCQ 也会根据 pcq-classifier 参数产生队列。每一个子队列都有一个 pcq-rate 的数据率限制和 pcq-linit 大小的数据包。PCQ 队列的中大小不能大于 pcq-total-limit 包。

以下实例说明了 PCQ 对数据包的用法，以他们的源地址分类。

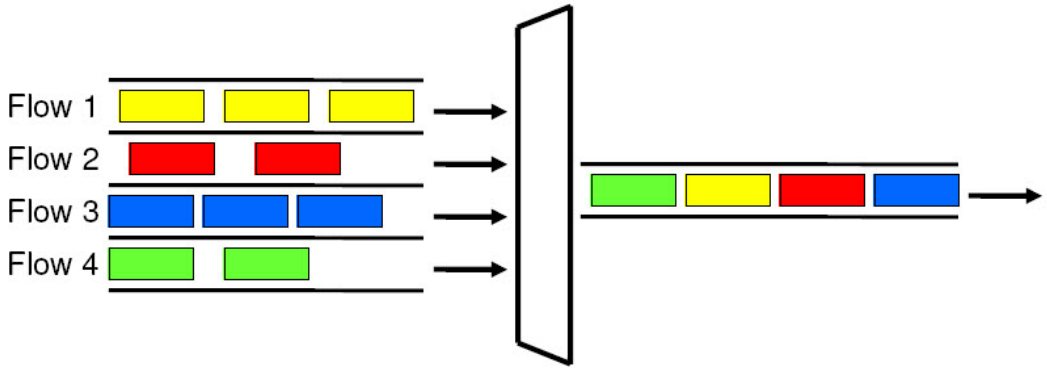


如果你以 src-address 对包分类那么带有不同源 ip 地址的包将被集合在不同的子队列中。现在你可以使用 pcq-rate 参数对每一个子队列限制或均衡。或许缀重要的部分是决定我们到底应该把这个队列附属到哪个接口上。如果我们把它依附在本地接口上，那么所有来自公网接口的数据都流将以 src-address(很可能这不是我们想要的)地址分组：相反的如果我们把他依附到公共接口，所有来自我们客户的数据都会一 src-address 分组－－于是我们可以很容易的限制或者均衡客户的上载。

有 pcq-classifiler 分类后为了在子队列中均衡速率，设置 pcq－rate 为 0 几乎不用管理，PCQ 也可以用来对多用户动态均衡或者形成流量。

RED

随机早先探测（RED）是一种通过控制平均队列长度避免网络拥塞的排序机制。当平均队列长度达到 red-min-threshold 是，RED 随机选择该丢弃哪个包，当平均队列长度变长时，堆砌多少包数的可能性会增加。如果平均队列长度达到 red-max-threshold.则丢弃该包。尽管如此，也存在真实队列长度（发平均的）远大于 red-max-threshold 时，丢弃所有超过 red-limit 的数据包的情况。



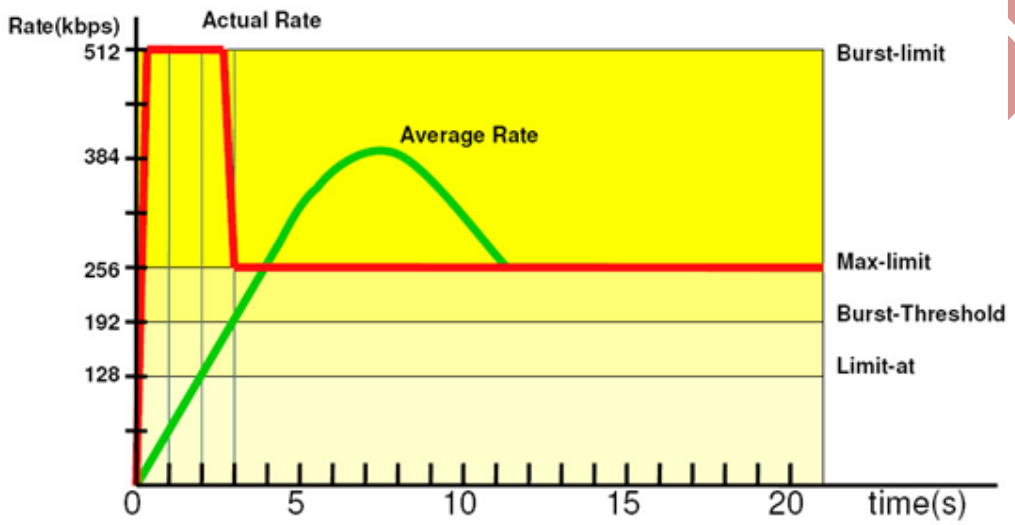
注意：RED 应用在高数据率的拥挤的连接上，他在 TCP 协议上工作的很好，但在 UDP 上就没那么理想了。

属性描述

Bfifo-limit(整型；默认：15000)－BFIFO 队列可以容纳的最大字节数
Kind(bfifo/pcq/pfifo/red/sfq)选择队列控制类型
Pfifo-字节先进先出
Pcq 每次连接队列
Pfifo-数据包先进先出
Red-随机早先探测
Afq-随机公平队列
Name(名称)队列类型相关名称
Pcq-classifier(dst-address/dst-port/src-address/src-port;默认：‘’‘’)PCQ 对其子队列进行分组的分类器。可以同时被数个分类器使用。例如：scr-address，src-port 可使用不同源地址和源端口把所有包分为独立的子队列
Pcq-limit(整数；默认：50)可以容纳一个单个 PCQ 子队列的包的数目
Pcq-rate(整数；默认：0)对每个子队列允许的最大数据率。0 值指的没有任何限制
Pcq-total-limit(整数；默认：2000) 可以容纳整个 PCQ 子队列的包的数目
Pfifo-limit(整数)PFIFP 队列可以容纳包的最大数目
Red-avg-packet(整数；默认：1000)被 RED 用来对平均队列长度计算
Red-burst（整数）用来决定平均队列长度被真实队列长度影响的快慢的字值。较长的值将减慢 RED 的计算速度－较长的脉冲串也是允许的。
Red-limit(整数)以字节计算。如果真实队列长度（非平均值）超过了这个值那么所有大于这个值的包都将被丢弃。
Red-max-threshold(整数)以字节计算。数据包标记概率最高的平均队列长度
Red-min-threshold(整数)当平均 RED 队列长度达到这个值时，数据包标记才有可能
Sfq-（整数；默认：1514）在一个 round-robin 中从子队列发出的字节数
Sfq-perturb(整数；默认：5)一秒计时。指定改变 SFQ 的散列算法的频率
Bursts 脉冲串
脉冲串用来在一段很短的时间允许更高数据率。每 1/6burst-time 时间，路由器都会计算每个类在上一个 burst-time 时间的平均数据率。如果这个平均数据率小于 burst-threshold,脉冲串

就会被启用且实际数据率达到 burst-bps.
否则实际数据率将跌至 max-limit 或 limit-at

让我们考虑如果我们有个 max-limit=256000， burst-time=8,burst-threshold=192000 以及 burst-time＝512000 的设置情况。当一个用户通过 HTTP 下载一个文件，我们可以观察到这样的现象：



在最开始的 8 秒中平均数据率是 0bps 因为在应用队列规则前没有流量通过，由于这个平均数据率小与 burst-threshold(192kbps),所有脉冲串会被使用，在第一秒之后，平均数据率为（0＋0＋0＋0＋0＋0＋0＋512）/8=64kbps,低于 burst-threshold，在第二秒后，平均数据率为（0＋0＋0＋0＋0＋0＋0＋512＋512）/128kbps,在第三秒之后达到临界点此时平均数据流变得大于 burst-threshold。这个时候脉冲串将被禁用且当前数据率降至 max-limit（256kbps）.

Simple queue 简单队列

限制数据率的 ip 地址和子网的简单方法就是使用简单队列。比也可以使用简单队列建立高级 Qos 应用：
Psp 流量队列
在选定时间间隔执行队列规则
FIFO 优先级
从/ip filewall mangle 使用多重包标记
形成双向流量（对上传和下载的带宽限制）

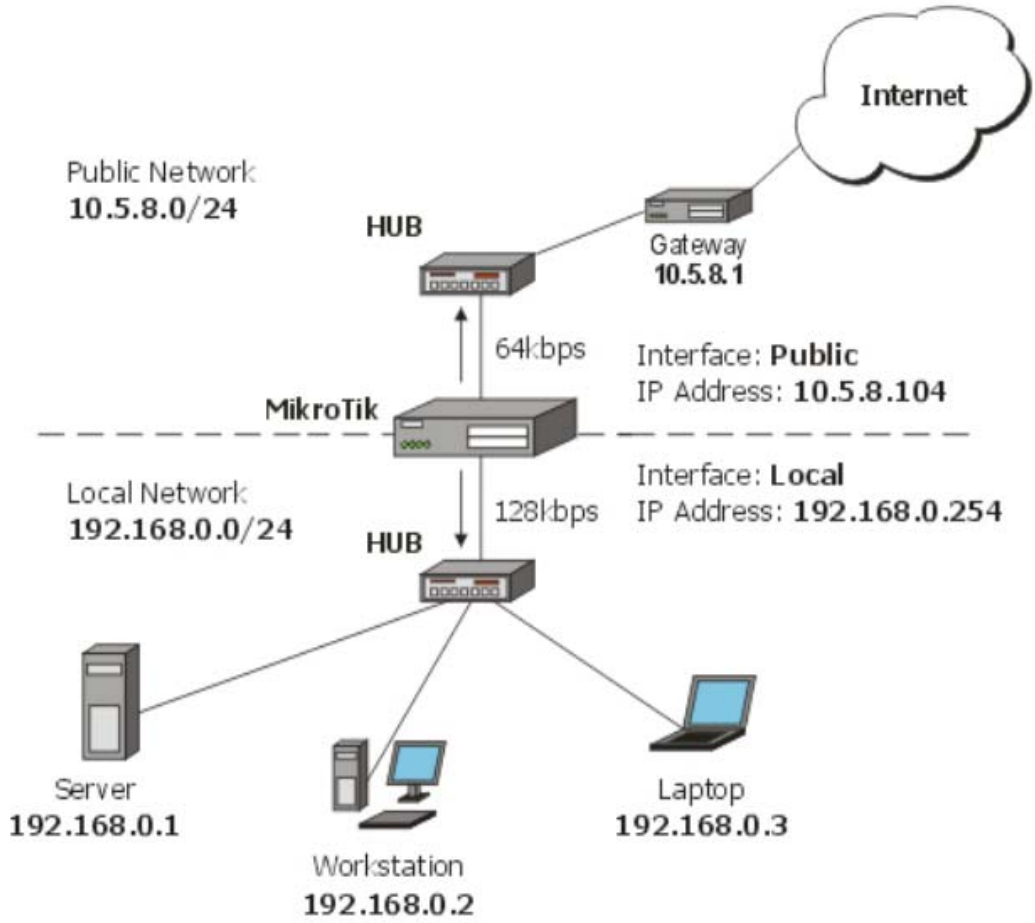
属性描述

burst-time（整数/整数）当脉冲串一 in/out（目标上传/下载）形式激活时可以达到的最大数据率
burst-threshold（整数/整数）整数/整数是否允许脉冲串。如果上一次脉冲时间的平均数据率

低于 burst-threshold 则实际数据率可能达到 burst-time。一 in/out（目标上传/下载）的形式。
burst-time（整数/整数）用于计算平均数据率。以 in/out（目标上传/下载）的形式。
Dirction(none both upload download)流量控制方向
None-队列停止有效的工作
Both-队列同时限制目标上行和目标下行
Upload 队列仅限制目标上行，下行的数据不会被限制
Download 队列仅限制目标下行，上行的数据不会被限制
Dst-address(ip 地址/子网掩码)dst-address 的掩码
Interface(文本)队列应用的对象端口。
Limit-at(整数/整数)该队列以 in/out（目标上传/下载）的形式定的数据率
Max-limit(整数/整数)在有足够带宽情况下可以达到的数据率，以 in/out（目标上传/下载）的形式
Name(文本)队列的描述性名称
P2p(any/all-p2p/bit-borrent/blubster/direnct-connect/edonkey/fasttrack/gnutella/souleek/winmx)控制匹配的 p2p 流量类型
All-p2p 匹配的所有 p2p 匹配
Any 匹配任何数据包（即不会检查该属性）
Packet-marks(名称；默认：‘’‘’)ip filewall 中的数据包标记
Mangle 更多数据包标记使用逗号（“ ,”）隔开
Parent(名称)父对列在等级制度中的名称。只能是其他简单队列
Priorty(整数：1…8)队列的优先级。1 是最高级的，8 是最低的
Queue(名称/名称；默认：default/default)以 in/out（目标上传/下载）的形式来自/queue type 的队列名称
Target-addresses(ip 地址/子网掩码限制目标 ip 地址（源地址）。使用源地址用逗号隔开
Time(时间。Sat/fri/thu/web/tue/mon/sun(+);默认：“ ”)限制队列在一个特定时间段的影响
Total-burst-limit(参数)global-total 队列脉冲串限制
Total-burst-threshold(整数)global-total/队列的脉冲串门限
Total-burst-limit（时间 global-total 队列脉冲时间
Total-limit-at(整数)限制累计的上传和下载为 Total-limit-at bps
Total-max-limit(整数) global-total 队列的限制上限（限制累计的上传和下载为 Total-max-limit bps）
Total-queue(名称)-globle-total 队列的队列规则

应用举例

下面假设我们想要对网络 192.168.0.0/24 流量限制为：下行 1MB 上行 512Kb，这里我们需要让服务器 192.168.0.1 不受流量控制。网络的基本设置如图：



这里我们使用（simple queue）简单队列，首先我们配置 RouterOS 的 ip 地址，网关和 NAT 等基本网络参数：

```
[admin@mikrotik] ip address> print

Flages:X-disabled  I-invalid  D-dynamic

#     ADDRESS          NETWORK          BROADCAST              INTERFACE
0     192.168.0.254/24   192..168.0.0     192.168.0.255          Local
1     10.5.8.104/24      10.5.8.0         10.5.8.255             public
[admin@mikrotik] ip address>
```

路由配置：

```
[admin@mikrotik] ip route> print
Flages: X – disabled, I – invalid, D – dynamic,
C – connect, S - static,   r - rip, b - bgp, o - ospf
#            DST-ADDRESS        G GATEWAY              DISTANCE   INTERFACE
0   ADC   10.5.8.0/24                                              public
```

1 ADC 192.168.0.0/24 Local
2 A S 0.0.0.0/0 r 10.5.8.1 public
[admin@mikrotik] ip route>

最后不要忘记在 ip filewall nat 中配置 src-nat 的伪装或 nat,做到这转换操作。

为网络 192.168.0.0/24 的所有客户端添加一个限制下载流量为 2MB 上传流量 1MB 的简单队列规则。

[admin@mikrotik] queue simple> add name=limit-local target-address=192.168.0.0/24
May-limit=1000000/2000000
[admin@mikrotik] queue simple> print
Flages: X - disabled, I - invalid, D - dynamic
0 name=limit-local target-address=192.168.0.0/24 dst-address=0.0.0.0/0
Parent=none priority=8 queue=default/default limit-at=0/0 max-limit=1000000/2000000 total-queue=default
[admin@mikrotik] queue simple>

Max-limit 限制了最大可用带宽，从客户的角度看，参数 traget-address 定义限制带宽的目标网络或者主机（也可以用逗号分隔开网络段或主机地址）。

这里不想让服务器受到我们添加上面规则的任何流量限制，我们可以通过添加一个没有任何限制的规则（max-limit=0/0 代表没有任何限制）并把它移到列表的顶部：

[admin@mikrotik] queue simple> add name=server target-address=192.168.0.1/32
[admin@mikrotik] queue simple> print
Flages: X – disabled, I – invalid, D - dynamic
0 name=”Limit-Local” target-address=192.168.0.0/24 dst-address=0.0.0.0/0
Parent=none priority=8 queue=default/default limit-at=0/0 max-limit=65536/131072 total-queue=default

1 name=”server” target-addresses=192.168.0.0/32 dst-address=0.0.0.0/0
Parent=none priority=8 queue=default/default limit-at=0/0 max-limit=0/0 total-queue=defaul
[admin@mikrotik] queue simple> move 1 0
[admin@mikrotik] queue simple> print
Flages: X – disabled, I – invalid, D - dynamic
0 name=”server” target-addresses=192.168.0.0/32 dst-address=0.0.0.0/0
Parent=none priority=8 queue=default/default limit-at=0/0 max-limit=0/0 total-queue=default

1 name=”Limit-Local” target-address=192.168.0.0/24 dst-address=0.0.0.0/0
Parent=none priority=8 queue=default/default limit-at=0/0 max-limit=65536/131072 total-queue=defaul

[admin@mikrotik] queue simple>

HTB 介绍

BTB 等级令牌桶允许创建一个等级队列结构，并确定队列之间的关系。就像“父亲与儿子”或“兄弟之间”。

一旦队列添加了一个 child(子队列)将会变为 inner（内部队列），所有向下没有 children(子队列)称为 leaf 队列（叶队列）。内部队列仅负责传输的分配，所有 leaf 队列对符合的数据进行处理。早 RouterOS 必须指定 parent(父级)选项并指导一个队列为子队列。

双重限制

每个队列在 HTB 有 2 个速率限制：

CIR（约定信息速率 committed information rate）-（在 routerOS 中的参数为 limit-at）最坏的情况下，无论任何都会将得到给定的 CIR 传输量（假设我们能发送那么多的数据量）
MIR（最大信息速率 maximal information rate）-（在 routerOS 中的参数为 max-limit）最好的情况下，如果父级有 带宽，将获得改速率值

换句话说，首先 limit-at(CIR)都会被满足，仅当子队列尝试借调必要的数据传输从他们的父级，以达到最大的带宽 max-limit(MIR)

注：无论如何 CIR 都会被分配到符合队列的带宽（即使父级的 max-limit 满载）。那就是为什么，确保最佳的使用双重现在功能，我们建议坚持这些规则：

CIR 约定速率之和，即所有子级速率必须小于或等于可获得父级传输量。

CIR（parent）*>CIR(child1)+….+CIR(childN)
*如果父级与主父级可以设置为 CIR(parent)=MIR(parent)

任何子级的最大速率必须小于或者等于父级的最大速率

CIR（parent）*>CIR(child1)& MIR(parent)>MIR(child2)&….& MIR(parent)> CIR(childN)

在 winbox 中队列的颜色变化：

0%-5%使用情况-绿色
51%-75%使用情况-黄色
76%-100%使用情况-红色

优先级

这里已经知道，所有队列的 limit-at(CIR)都有可能将会被耗尽，优先级则主要负责分配父级

队列剩余的带宽给 child(子队列)达到 max-limit.队列高的优先级最优先达到 max-limit,优先级低的则不会。8 是最低优先级，1 则最高。
注意，优先级工作环境：

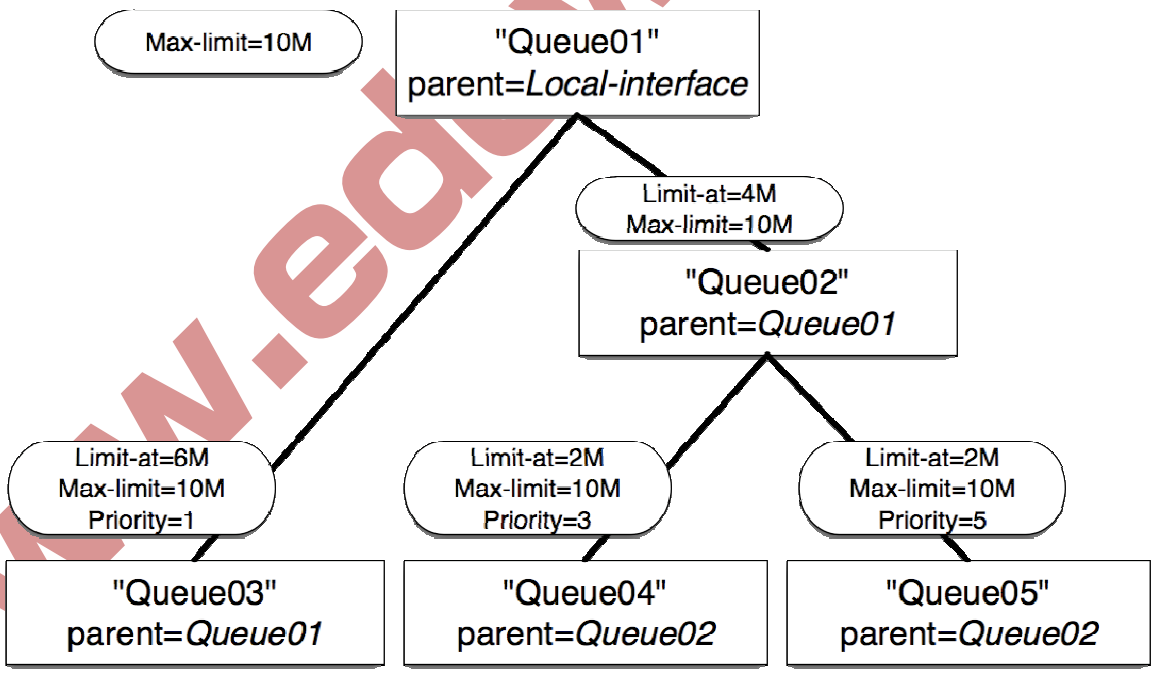
对于 leaf 叶队列-优先级对于 iner（内部队列）没有任何意义，即 inner 内部队列与 leaf（叶队列）的优先级不可比较
如果 max-limit 被设定（非 0）

下面这部分我们将分析 HTB 的操作，将演示一个 HTB 结构并将涵盖可能出现的所有情况和功能，我们的 HTB 结构由下面 5 个队列构成：

Queue01 内部队列有 2 个子级- Queue02 和 Queue03
Queue02 内部队列有 2 个子级- Queue04 和 Queue05
Queue03 叶队列
Queue04 叶队列
Queue05 叶队列

Queue03，Queue04 和 Queue05 的需要 10Mbps,我们接口处理能力在 10Mbps 的流量

事例 1：普通事例

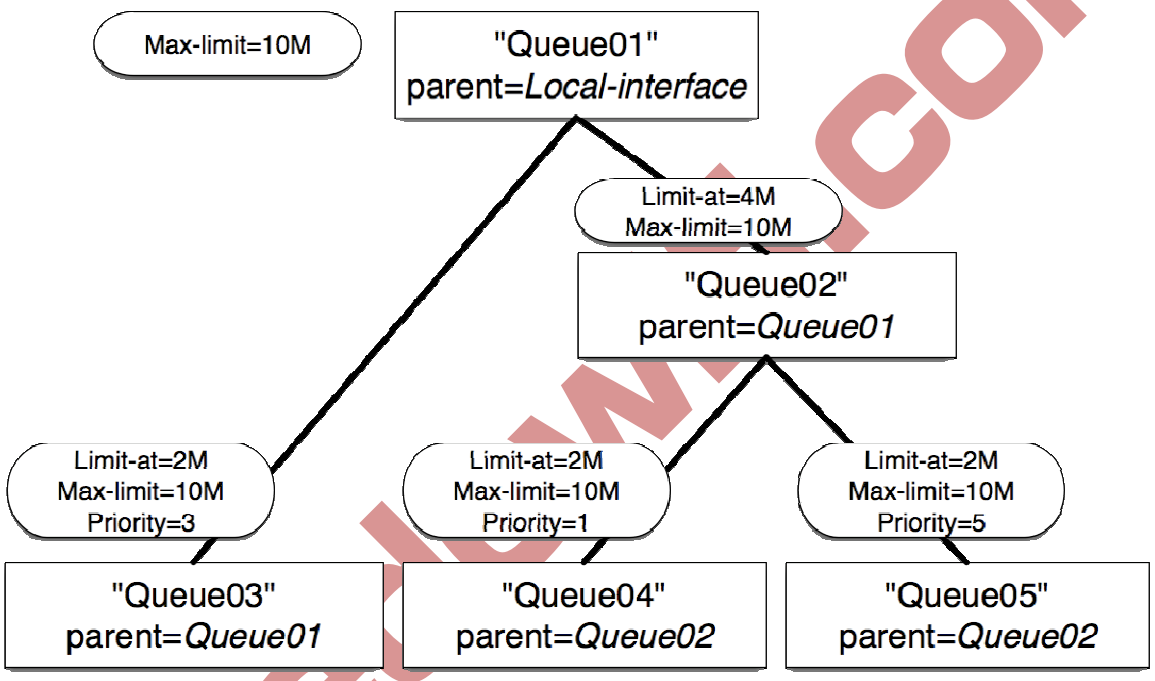


Queue01 limit-at=0Mbps max-limit=10Mbps
Queue02 limit-at=4Mbps max-limit=10Mbps
Queue03 limit-at=6Mbps max-limit=10Mbps priority=1
Queue04 limit-at=2Mbps max-limit=10Mbps priority=3
Queue05 limit-at=2Mbps max-limit=10Mbps priority=5

事例 1 结果：

Queue03 得到 6Mbps
Queue04 得到 2Mbps
Queue05 得到 2Mbps
结论：HTB 建立在一种方式上，通过满足所有的 limit-at，主队列已没有贷款进行分发。

事例 2：max-limit 事例

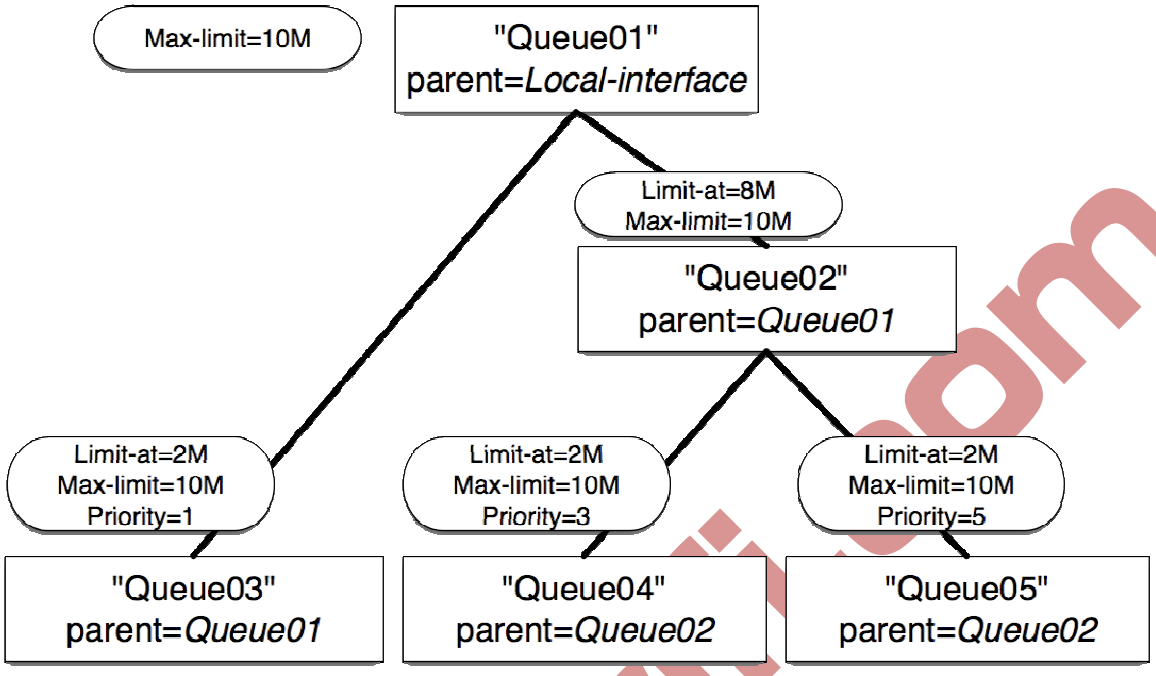


Queue01 limit-at=0Mbps max-limit=10Mbps
Queue02 limit-at=4Mbps max-limit=10Mbps
Queue03 limit-at=2Mbps max-limit=10Mbps priority=3
Queue04 limit-at=2Mbps max-limit=10Mbps priority=1
Queue05 limit-at=2Mbps max-limit=10Mbps priority=5

事例 2 结果

Queue03 得到 2Mbps
Queue04 得到 6Mbps
Queue05 得到 2Mbps
结论：在满足所有的 limit-at 后，HTB 将把剩余的带宽分配给优先级高的队列。

事例 3：inner 队列 limit-at



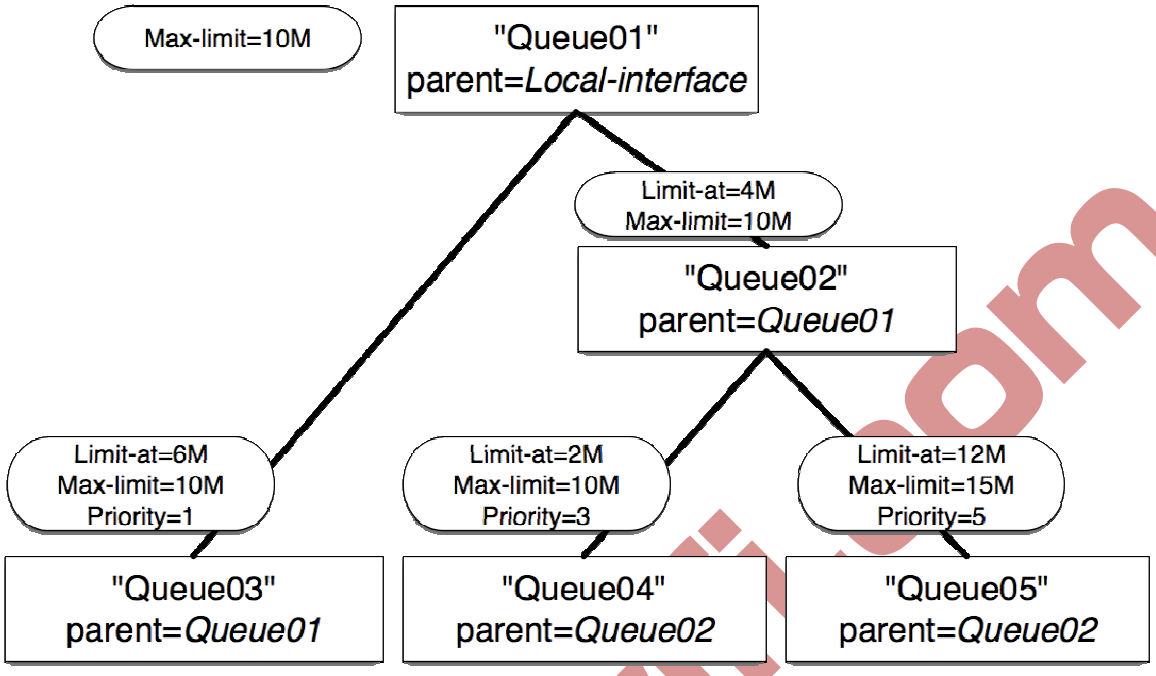
Queue01 limit-at=0Mbps max-limit=10Mbps
Queue02 limit-at=8Mbps max-limit=10Mbps
Queue03 limit-at=2Mbps max-limit=10Mbps priority=1
Queue04 limit-at=2Mbps max-limit=10Mbps priority=3
Queue05 limit-at=2Mbps max-limit=10Mbps priority=5

事例 3 结果

Queue03 得到 2Mbps
Queue04 得到 6Mbps
Queue05 得到 2Mbps

结论：在满足所有的 limit-at 后，HTB 将分配剩余带宽给优先级高的，但在这个事例中，内部对列 queue02 指定了 limit-at,这样他会保留 8Mbps 的流量给 Queue04 和 Queue05，Queue04 有更高的优先级，那就是为什么会得到更高的带宽。

事例 4：leaf 队列的 limit-at



Queue01 limit-at=0Mbps max-limit=10Mbps
Queue02 limit-at=4Mbps max-limit=10Mbps
Queue03 limit-at=6Mbps max-limit=10Mbps priority=1
Queue04 limit-at=2Mbps max-limit=10Mbps priority=3
Queue05 limit-at=12Mbps max-limit=15Mbps priority=5

事例 4 结果

Queue03 得到 3Mbps
Queue04 得到 1Mbps
Queue05 得到 6Mbps

结论：为了满足所有的 limit-at 后，HTB 被强迫分配 20Mbps, Queue03 为 6Mbps, Queue04 为 2Mbps，Queue05 为 12Mbps，但我们接口只能处理 10Mbps，因此接口队列通常 FIFO 带宽发分配将保持比例 6：2：12，即 3：1：6.

RouterOS 中的 HTB

在 routerOS 中有 4 个 HTB 树：

Global-in
Global-total
Global-out
Interface queue

当添加一个队列时，将产生 3 个 HTB 类（in global-in,global-total and global-out）,但在接口队列中不添加任何类。
当数据包通过路由器时，它将穿过所有 4 个 HTB 树-global-in, global-total,global-out 和 interface queue,如果是指向路由器的它将穿过 global-in 及 global-total HTB 树，如果数据包时从路由器发出的，它们将穿过 global-total,global-out 及 interface 队列。

Queue tree 队列树

操作路径：/queue tree

当你想使用基于协议，端口。Ip 地址等的复杂数据分配流量时，你需要使用队列树。首先通过在/ip firewall mangle 下标记数据包流然后使用这个标记作为在这个队列树的数据包流标识。

属性描述

Burst-limit(整数)当脉冲激活时可以达到的最大数据率
Burst-threshold(整数)用于计算是否允许脉冲，如果上一次脉冲时间的平均数据率低于 burst-threshold 则实际数据率可能达到 burst-limit.
Burst-limit(整数)用于计算平均数据率
Flow(文本)在/ip filewall mangle 下标记的数据包流，当前队列参数仅应用于用这个数据流标记了的数据包
Limit-at(整数)这个队列的约定流量
Max-limit(整数)在足够带宽可用的情况下可达到的流量
Name(文本)队列的描述性名称
Parent（文本）父队列的名称，顶级的队列是可用的接口（实际上是主 HTB）低级点的父队列可能是其他的队列，
Priority(整数，1…8)队列的优先级。1 是最高级等级，8 为最低。
Queue（文本）队列类型名称，类型是在/queue type 下定义的。这个参数仅应用于树等级制证中的子队列

Queue tree HTB 实例

这个事例中，设定 3 类数据 VIP,Web 和 other,这三类数据中 VIP 为网络内的重要用户优先级最高为 1，访问网页的数据 web 其次为 2，而剩下的数据 other 级别最低为 7，假设我们的网络是 1M 的 ADSL，我们通过配置 HTB 策略老保证网络内的优先数据。

通过用 new-connection-mark 标记向外的连接，并采取 mark-connection 动作。当这个完成时你可以使用 new-packet-mark 标记属于这个链接的所有数据包并采用 mark-packet.
首先 VIP 数据标记，我们通过 ip filewall address-list 定义 VIP 用户的地址列表，定义完成后通过 src-address-list 调用：

```
[admin@office] /ip firewall mangle> print
Flages: X – disabled, I – invalid, D - dynamic
```

0;;; vip
Chain=forward action=mark-connection new-connection-mark=vip
Passthrough=yes src-address-list=vip

1 Chain=forward action=mark-packet new-packet-mark=vip
Passthrough=no connection-mark=vip

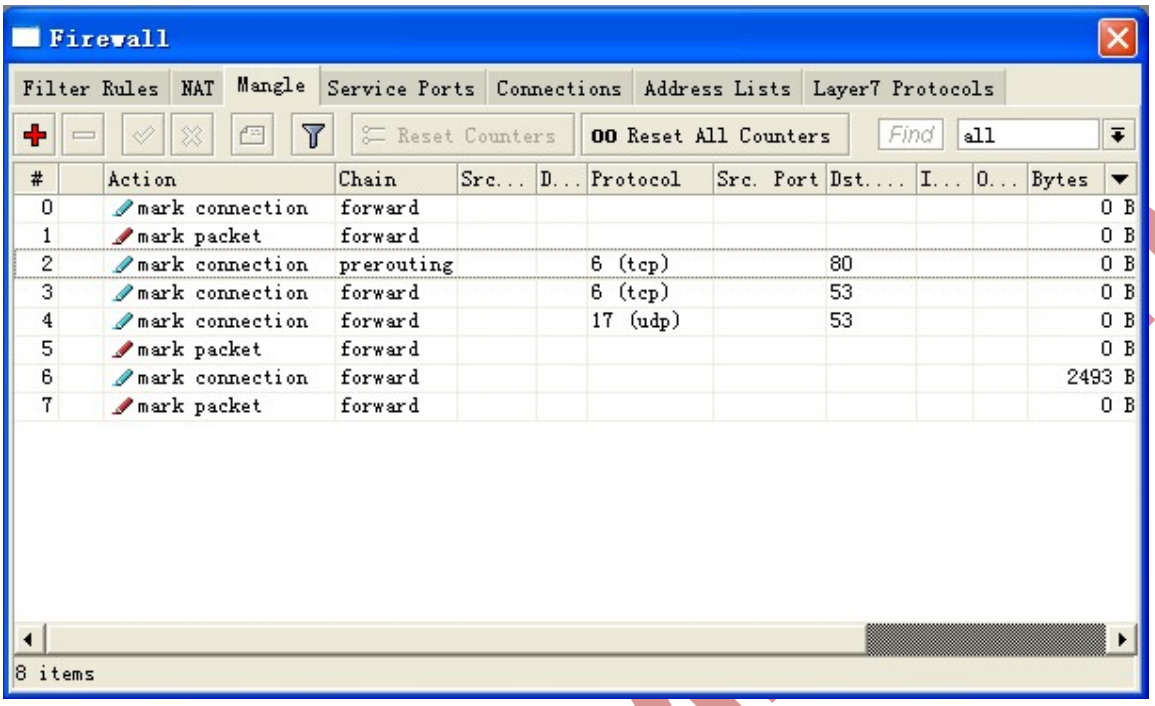
跟着定义 web 数据;这里我们需要针对访问网页的 tcp/80 端口和域名解析的 DNS 端口 tcp/53 端口标记:

2;;; web
Chain=forward action=mark-connection new-connection-mark=web
Passthrough=yes protocal=tcp dst-port=80

3 Chain=forward action=mark-connection new-connection-mark=web
Passthrough=yes protocal=tcp dst-port=53
4 Chain=forward action=mark-connection new-connection-mark=web
Passthrough=yes protocal=tcp dst-port=53
5 Chain=forward action=mark-packet new-packet-mark=web
Passthrough=no connection-mark=web

最后对剩下的 other 数据进行标记，因为前面已经标记了 VIP 和 web 的数据包，所有剩下数据就是其他的 other 数据:

6;;; other
Chain=forward action=mark-connection new-connection-mark=web
Passthrough=yes
7 Chain=forward action=mark-connection new-connection-mark=web Passthrough=no
onnection-mark=other



标记数据完成后，我们进入 queue tree 中，对数据进行优先级的配置，ADSL 总带宽为 1Mbps 下行，25kps 的上行，给三类数据带宽分配如下

VIP:下行 max-limit=800k limit-at=400k,上行 max-limit=2200k limit-at=200k,优先级 1
Web: max-limit=800k limit-at=400k,上行 max-limit=200k limit-at=200k,优先级 2
Other: max-limit=600k limit-at=200k,上行 max-limit=150k limit-at=50k,优先级 7

根据以上参数。我们在 queue tree 中配置队列优先级：

[admin@office] /queue tree > print
Flags: X - disable, I – invalid
0 name=totalup parent=pppoe-out1 packet=mark=”” limit-at=o queue=default
Priority=1 max-limit=250000 burst-limit=0 burst-threshold=0 burst-time=0s
1name=totaldown parent=other2 packet=mark=”” limit-at=o queue=default
Priority=8 max-limit=1000000 burst-limit=0 burst-threshold=0 burst-time=0s

2 name=vipdown parent=totaldown packet-mark=vip limit-at=o queue=default
Priority=2 max-limit=700000 burst-limit=0 burst-threshold=0 burst-time=0s

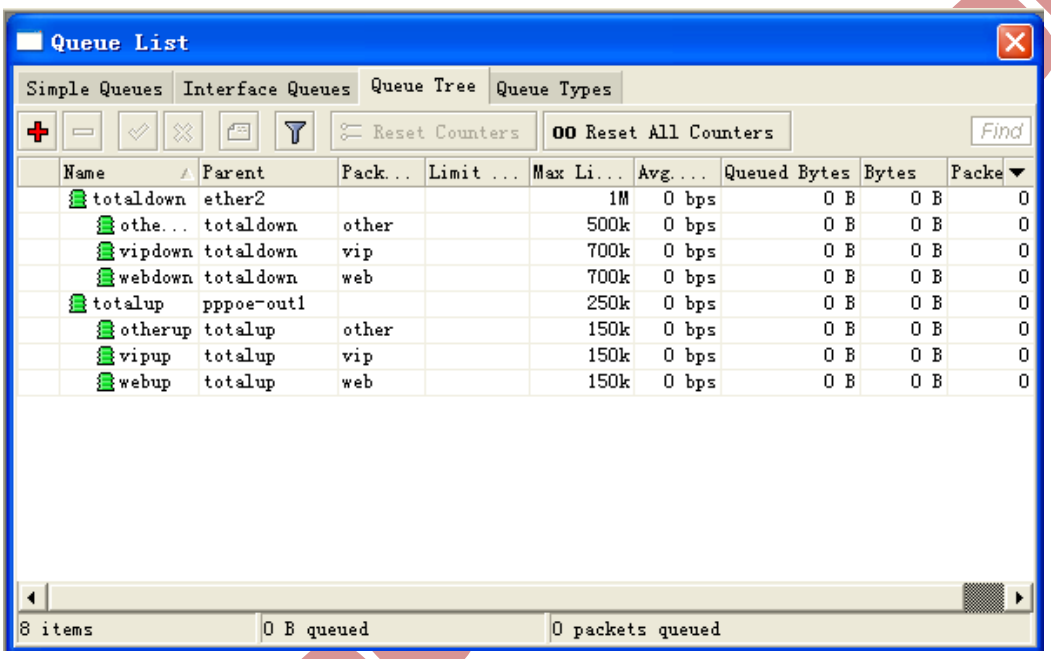
3 name=vipup parent=totalup packet-mark=vip limit-at=o queue=default
Priority=2 max-limit=150000 burst-limit=0 burst-threshold=0 burst-time=0s

4 name=otherdown parent=totaldown packet-mark=other limit-at=o queue=down
Priority=8 max-limit=500000 burst-limit=0 burst-threshold=0 burst-time=0s

5 name=otherup parent=totalup packet-mark=other limit-at=o queue=default Priority=8 max-limit=150000 burst-limit=0 burst-threshold=0 burst-time=0s

6 name=webup parent=totalup packet-mark=web limit-at=o queue=default Priority=1 max-limit=150000 burst-limit=0 burst-threshold=0 burst-time=0s

7 name=webup parent=totaldown packet-mark=web limit-at=o queue=default Priority=1 max-limit=700000 burst-limit=0 burst-threshold=0 burst-time=0s



PCQ 配置

PCQ 算法比较简单，首先利用分类器从相应数据流中区分一个子数据流，然后在每个一个子数据流上建立独立的 FIFO 队列长度和限制，再归类所有的子数据流在一起，并应用全局 FIFO 队列长度和限制。PCQ 参数：

Pcq-classifier(dst-address/dst-port/src-address/src-port;默认：‘‘’‘)：选择子数据流分类类型、
Pcq-rate(数字)每个子数据流可获得的最大数据带宽
Pcp-limit(数字)在数据包中一个数据流的队列长度
Pcq-total-limit(数字)：全局 FIFO 队列的队列长度

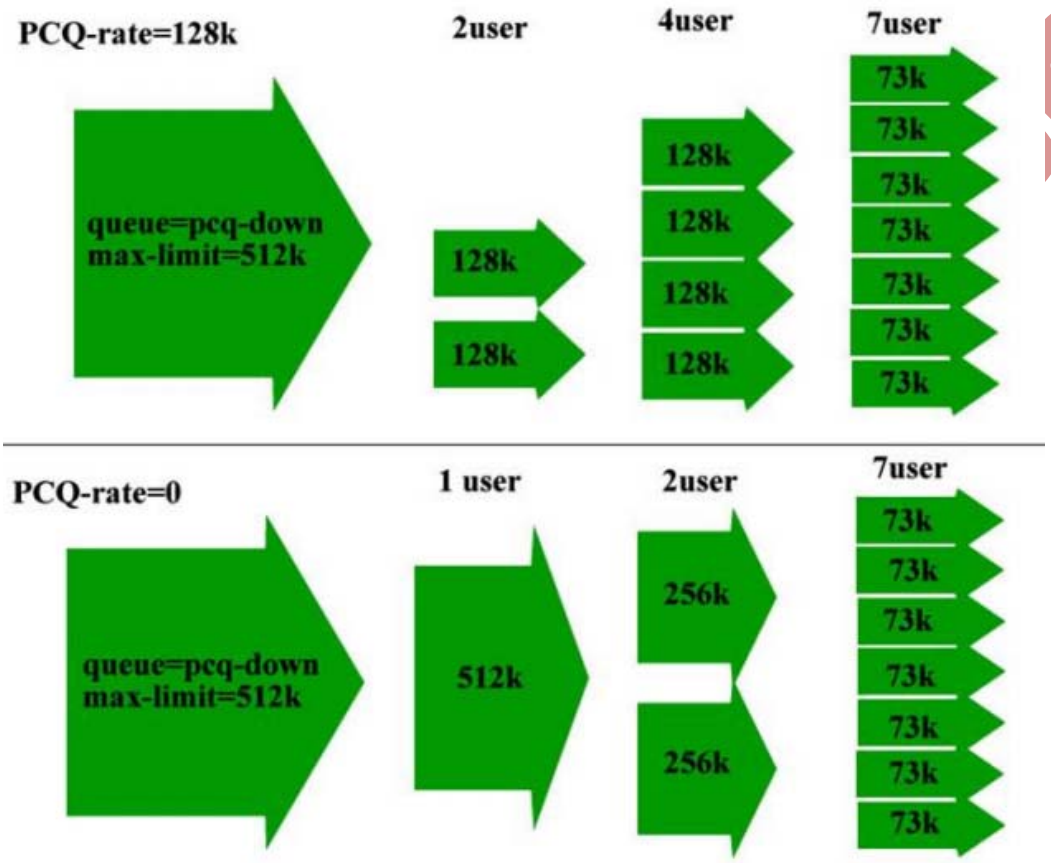
因此，等有 100 个队列需要限制 1000kbps 下载时，我们可以使用 1 个 PCQ 队列和该 PCQ 队列包含的 100 子数据流队列，

分类器

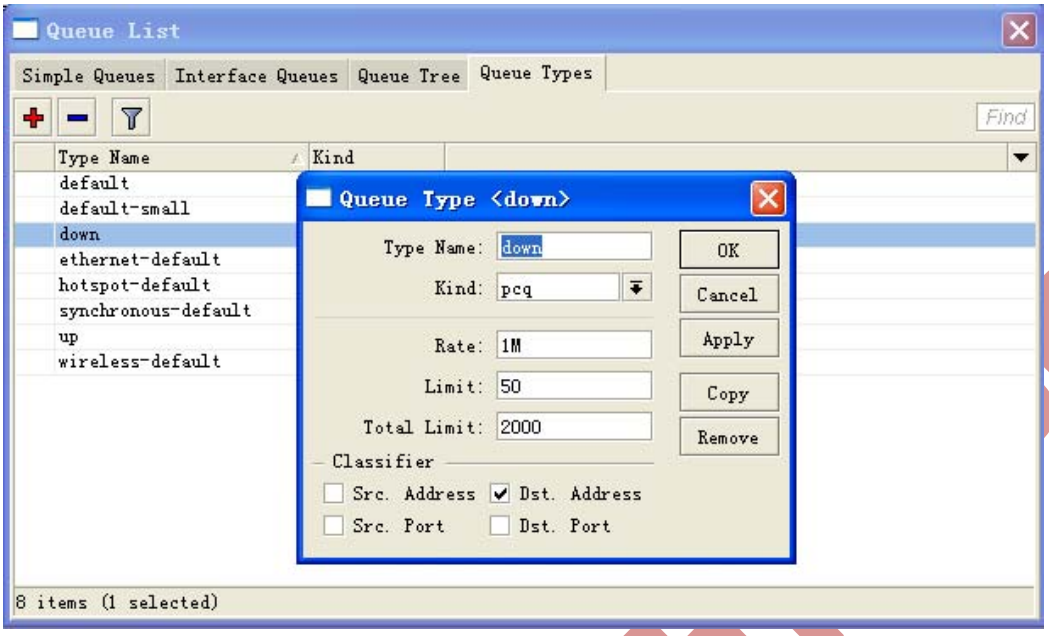
为了更好的理解分类器，将用一个从指定的地址和端口到一个指定的地址和端口 18 个数据

流，这时我们将选择一种分类器面饼通过 PCQ 将 18 个数据流分离带 PCQ 的子数据流。

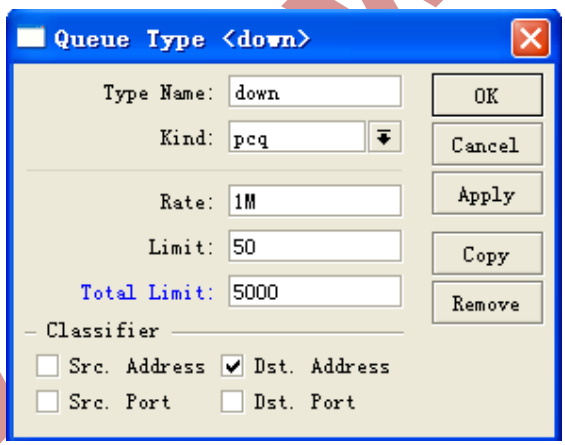
在局域网中因为网络带宽的问题，需要对网络流做控制，但又因为做固定的流量控制的时候，会造成在上网空时候带宽的浪费，这里我们可以同 routerOS 的 PCQ 算法完成对内部局域网流量的动态分配，如下图所示：



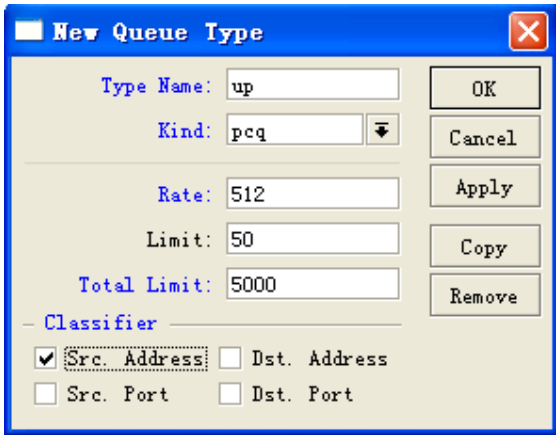
通过上图，我们可以看到当 PCQ 的速率设定为 128k 的时候，平均每个用户将会得到同样的带宽 128k，当上网高峰期的时候 PCQ 才会做二次流量分配，如果 PCQ 得速率在开始就设置为 OK，这样在一个用户的时候可以得到全部带宽，之后是 2 个用户平均分配，依次类推，但最后带宽会控制在 73K 的范围内，控制最小使用带宽，保证用户正常使用。
配置这里我们配置 192.168.10.0/24 这个段的 PCQ 流量控制，估计有 100 个用户在线，首先进入 queue type 中配置 PCQ 的上行和下行分别 512k 和 1m.



首先我们配置下行，每个用户获取 1m 的下行流量。由于是 100 个用户在线，所以在 limit 不变的情况下，total-limit 应该设置为 50*100=5000 下行指向的是目标地址，所以我们选择 dst-address:



上行选择 src-address,并配置 512K 的上行流量配置如下：

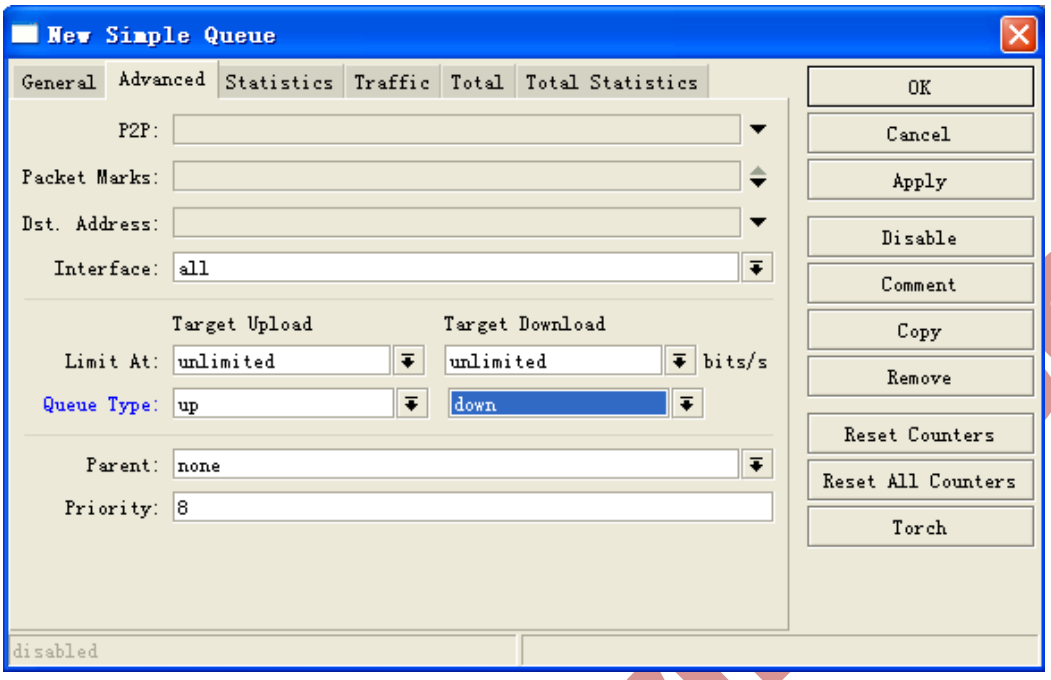


注意，limit 和 total-limit 的关系：

默认情况下 total-limit 是 2000，该规则仅能容纳 40 个用户（total-limit/limit=2000/50=40）
解决方法必须增加 total-limit 或者减少 limit
但必须保证每个用户队列（limit）获取 10-20 个数据包

在配置好 queue type 后我们进入 simple queue 中配置流量控制规则，这里我们在 general 中配置总出口带宽假设为 10M 上行带宽为 5M，内网地址段为 192.168.10.0/24：

接下来配置 queue-type 类型，进入 advanced 目录，选择上行和下行为刚才定一的 PCQ 类型 UP 和 down:

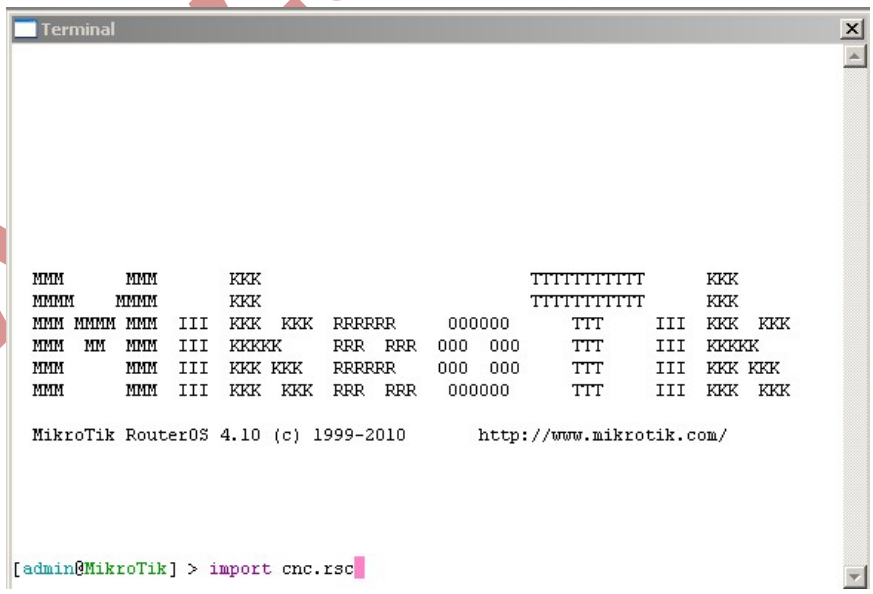


这样 PCQ 配置就完成，只需要在 simple queue 中配置一条规则，就可以控制所有用户的流量。

电信网通流量控制

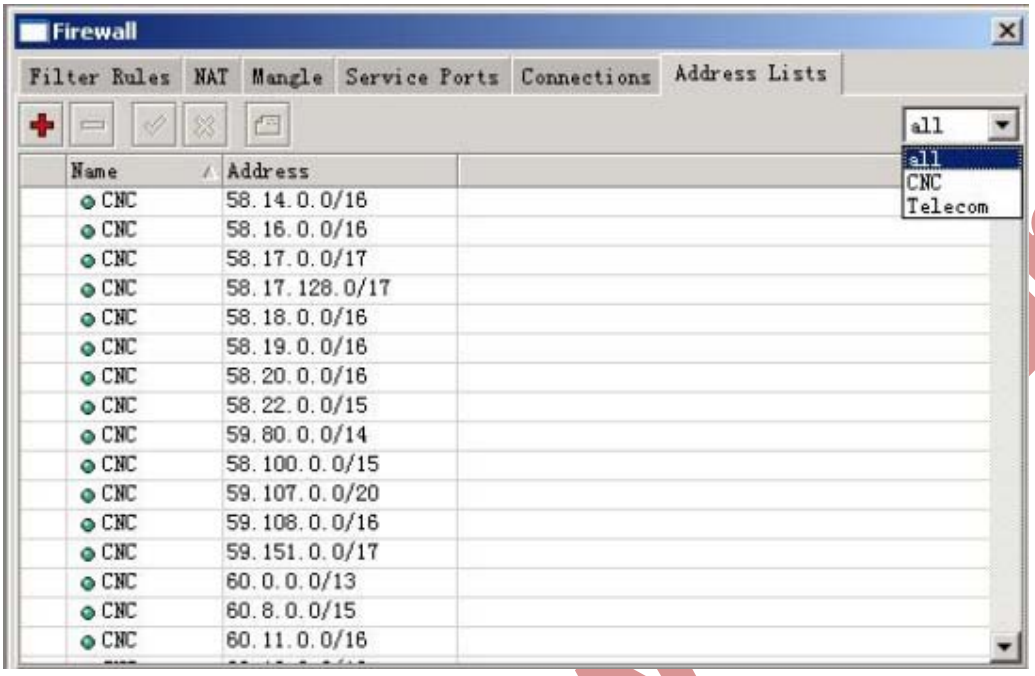
对于电信和网通的 ip 的地址段是已知，那么我们可以通过地址标记来实现对这些地址的浏览控制，首先我们将电信和网通的地址段导入 routerOS 的 address-list 中

通过 import 命令，导入地址列表：

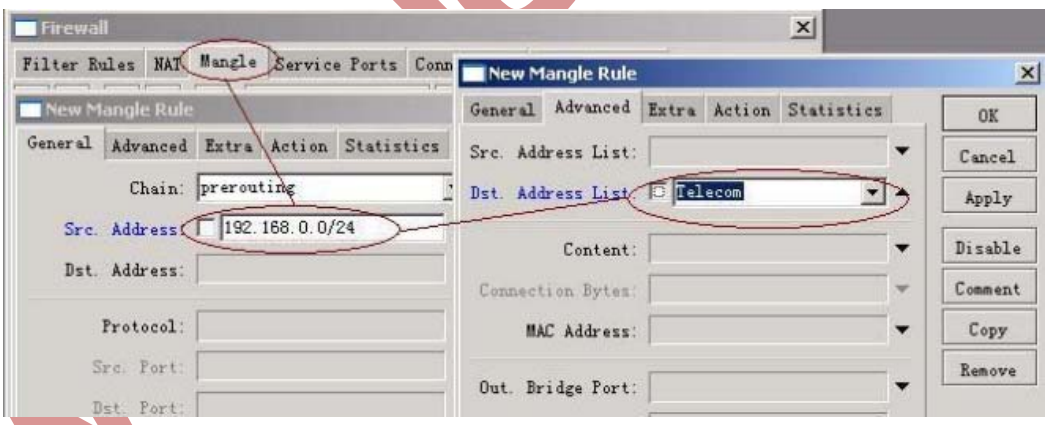


[键入文字]

导入后我们可以在/ip filewall address-list 中找到：



进入/ip firewall mangle 设置，这里我们定于访问电信的流量控制，我们的内网地址段位 192.168.0.0/24，所有这里我们配置源地址 src-address=192.168.0./24.在 mangle 中先标记连接，然后在从连接中提取数据包：



定义标记类型：

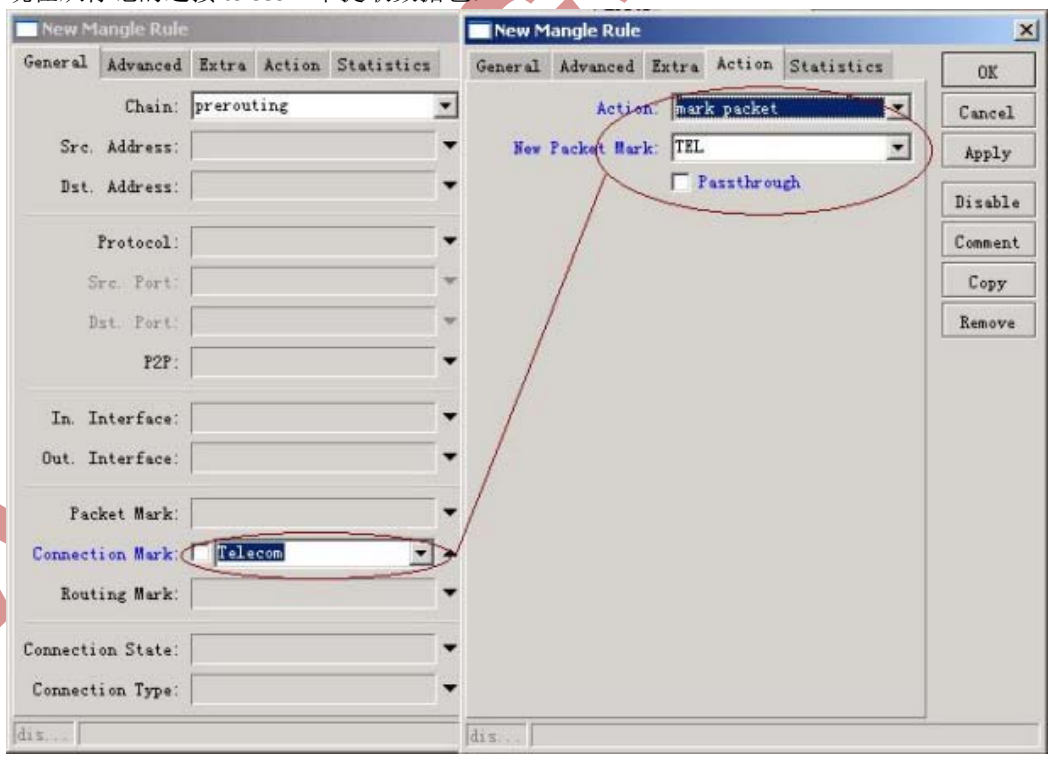
源代码

/ ip firewall mangle

Add chain=preouting src-address=192.168.0.0/24 dst-address-list=telecom

Action=mark-connection new-connection-mark=telecom passthough=yes comment= “” disabled=no

现在从标记的连接 telecom 中提取数据包：



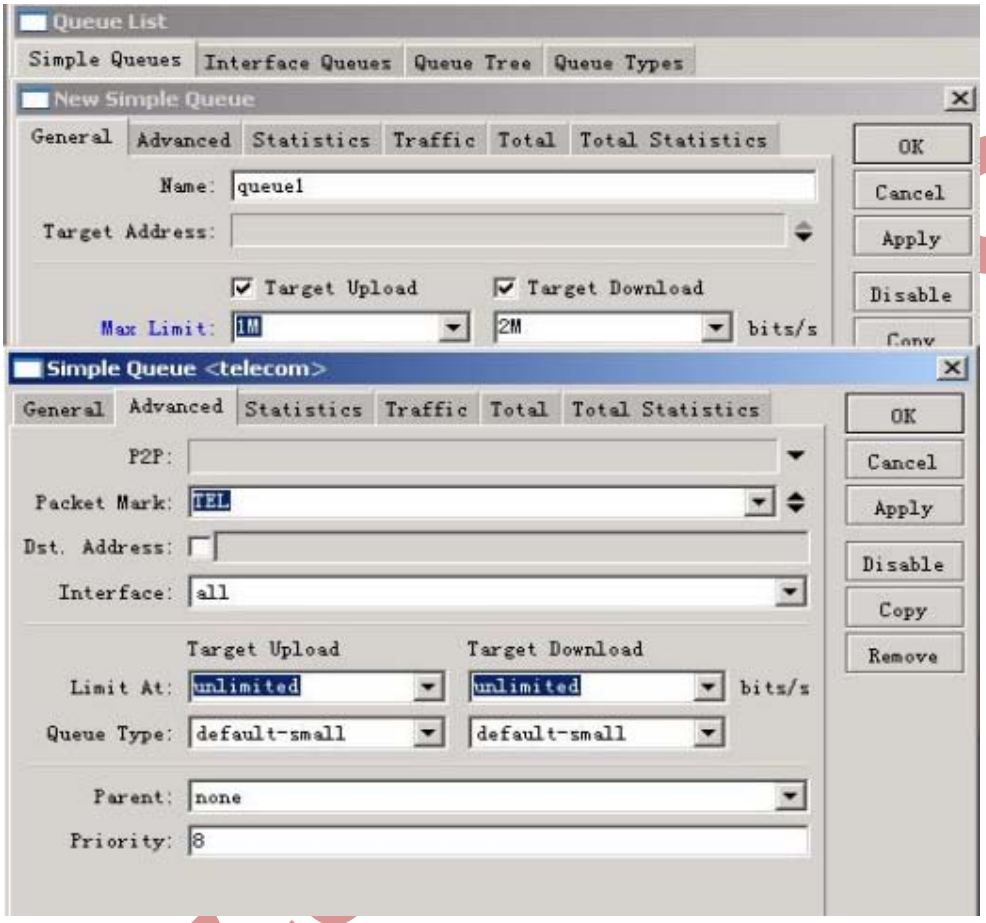
源代码：

/ ip firewall mangle

[键入文字]

Add chain=preouting connection-mark=telecom action=mark-packet
New-packet-mark=tel passthrough=no comment=”” disabled=no

现在我们进入.queue simple 队列中配置流控制规则，在这里我们把刀电信的带宽控制在 IM 上行和 2M 下行



源代码

/ queue simple
Add name=telecom dst-address=0.0.0.0/0 interface=all parent=none packet-mark=TEL
Direction=both priority=8 queue=default-small/default-small limit-at=0/0
Max-limit=100000/200000 total-queue=default-small disable=no

这样对电信的带宽便完成，控制网通带宽同样的

第十一章 网络地址翻译 NAT

网络地址翻译（nat）是一种当 ip 包通过路由器时取代其源和目标地址的理由协议。它通常被用来启用专用网络的多个主机使用一个公用 ipdz 访问因特网。

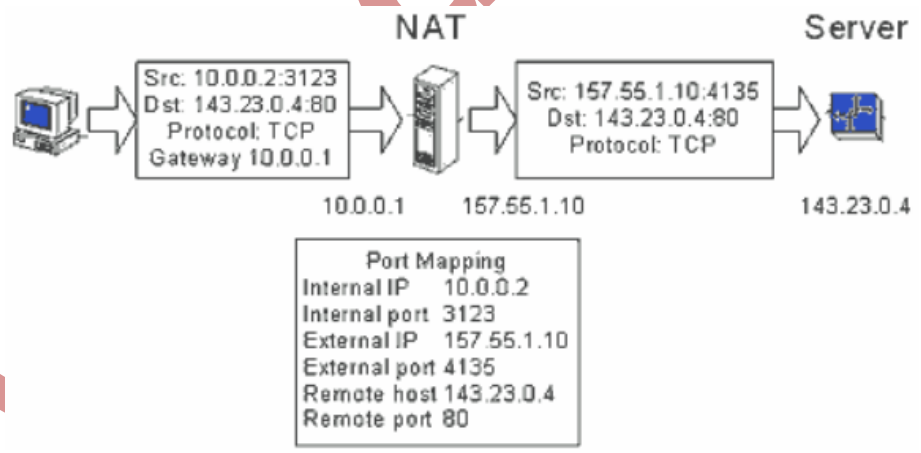
规格说明

功能包要求：system
等级要求：level1 level3
操作路径：/ip firewall nat
标准及技术：IP REC1631.REC2663
硬件使用：提升 CPU 和内存有助于 NAT 规则的处理

Nat 介绍

网络地址翻译是一种允许本地网络主机使用一段 ip 地址进行本地通信，使用令一段 ip 地址进行外部通信的因特网标准。一个使用网络地址翻译的局域网就内称为 natted（已翻译）网络。为了使网络地址翻译进行工作必须在每个 natted 网络都有一个 nat 网关。Nat 网关的作用就是在数据包进/出局域网同时重写 ip 地址的作用。

输出数据包转换的示范:



网络地址翻译包括两种类型：

源网络地址翻译或者 srcnat。这种类型的网络地址翻译工作在从一个 natted 网络产生的数据包上，nat 路由器在 ip 包通过它的时候用一个新的公网 ip 地址代替了其私有源地址。相反的操作适用于响应包从相反方向通过路由器时，
目标网络地址翻译或者 dstnat，这种类型的网络地址翻译工作在到达一个 natted 网络产生的

数据包上它通常用于使一个私有网络上的主机能够被因特网访问。Dstnat 路由器在 ip 包通过路由器到达私有网络时替换了 ip 包的目标 ip 地址。

Nat 缺点

在一个使用了网络地址翻译的路由器背后的主机并不拥有真实的端对端的连接。因此一些因特网协议就不在有网络地址翻译的情况下工作。一些来私有用网络外部或者无连接协议如 UDP 协议且需要 TCP 连接初始的服务将被打断。此外，一些内在于 NAT 不兼容，一个鲜明的事例就是 Ipsec 中的 AH 协议。

重定向与伪装

重定向和伪装分别是目的 nat 和源 nat 的特殊形式。重定向类似于普遍的目的网络地址翻译就好比伪装类似于源网络地址翻译—伪装是一整不需要指定 to-address 的源网络地址翻译的特殊形式-对外接口地址将被自动使用。重定向同理-进入接口地址将被使用。注意：to-ports 对于重定向规则来说很有意义-这就是在路由器起上处理这些请求的服务端口。（比如：web 代理）

当数据包进行了网络地址翻译（dst-nat）时（不论 action=nat 或者 action=redirect）.目的地址将会改变。有关地址翻译的任何信息（包括初始的目的地址）将被保存在路由器的内部维护表。当 web 请求被重定性到路由器的代理端口时，工作在路由器上的透明 web 代理将访问从内部表这个信息并西欧那个其中得 web 服务的地址。如果你正在对几个不同的代理服务器进行目的网络地址翻译，那你将不会从 ip 包头找到 web 服务器地址，因为 ip 包的目的地址之前是 web 服务器地址单现在已经变成了代理服务器的地址。从 HTTP/1.1 开始在 http 请求中出现了特殊的可以告知 web 服务器地址的包头，于是代理服务器使用它取代了 ip 包的目的地址。如没有这样的包头（如：老版本的 HTTP），代理服务器将不能确定 web 服务器地址也将无法工作。

这就是说，对 HTTP 流从一个路由器到其他一些透明代理服务器禁止正确的重定向式有可能的。只有在路由器本身添加透明代理并配置才是正确的方法。因此你的“真实的“代理就是上级代理。这种情况下你”真实的“代理再也不用时透明的，因为在路由器上的代理将成为透明的并将向”真正的“代理转交代代理方式请求（根据标准，这些请求包括 所有必须得 web 服务器信息）。

属性描述

action(accept/add-dst-to-address-list/and-src-to-address-list/dst-nat/jump/log/masquerade/natmap/passthrough./redirect/return/same/src-nat;default;accept)如果数据包与规则匹配 action 将启用

Accept-接收数据包。不进行任何动作。例如：数据包通过而且没有其他任何用于它的规则

Add-dst-to-address-list 向 address-list 参数指定的地址表中添加 ip 包的目的地址

Add-dst-to-address-list 向 address-list 参数指定的地址表中添加 ip 包的目的源地址

Dst-nat 用 to-addresses 及 to-ports 参数指定的变量取代 ip 包的目的地址

Jump 跳转到由 jump-target 参数指定的链

Log-action 的每个匹配都将对系统日志添加一条消息

Masquerade 以一个路由策略自动分配的 ip 地址取代 ip 包的源地址

Netmap 创造一个 ip 地址从一端到另一端的静态 1：1 映像。通常用于分配公用 ip 地址专用内网的主机上

Passthrough 忽列次条规则并转向下一条规则

Redirect 把 ip 包的目的地址替换成一个路由器本地地址

Return 返回到跳转发生的链

Same 从允许范围内分配给特定客户每个连接相同的源/目的 ip 地址。这种情况通常有来自期望相同客户的相同客户地址对多重连接的服务。

Src-nat 把 ip 包的源地址替换成由 to-address 和 to-ports 参数指定的值，

Address-list(name)指定地址列表的名称以收集使用了 action=add-dst-to-address-list 或

Action=add-arc-to-address-list 动作规则的 ip 地址

Address-list-timeout(time;default:00:00:00)在 address-list 参数指定的地址列表删除地址之间的时间间隔。

与 add-dst-to-address-list 或 add-src-to-address-list 动作一起使用

00：00：00 从地址列表中永久删除

Chain(dstnat/srcnat/name)定义一个具有特定规则的链。 由于不同德数据流通过不同的链，所以为新规则选择正确的链必须小心。如果输入与一个已定义好的链名不匹配，那么一个新的链将被生成。

Dstnat 在这个链中的规则会在路由前被应用。替代 ip 包目的地址的规则应放在这里

Dstnat 在这个链中的规则会在路由后被应用。替代 ip 包目的地址的规则应放在这里

Comment(text)对规则的描述性注解。一条注解能被用于从脚本中识别规则

Connection-bytes(整型-整型)当且仅当一定个定量字节从特定连接传输时与数据包进行匹配

0 代表无穷大，例如：connection-bytes=2000000-0 如果大于 2MB 数据从相关连接传输就与规则匹配

Connection-limit(整型。Netmask)限制每个地址或地址群的连接限度

Connection-mark(name)与通过 mangle 机制标记的特定连接数据包进行匹配

Connection-type(ftp/gre/h323/irc/mms/pptp/quake3/tftp)与基于连接跟踪助手信息相关链接的包进行匹配。相关链接助手必须在/ip firewall service-port 下启用

Content(text)文本数据包必须按顺序排列以与匹配规则

Dst-address(ip address/netmask/ip address-ip address)指定 ip 包的目的地址范围

Address/netmask 对合法网络地址的换算。例如：1.1.1.1/24

Dst-address-list(name)用户自定义的地址列表中匹配数据包的目的地址

Dst-address-type(unicast/local/broadcast/multicast)在 ip 包的目的地址类型中匹配其中之一

Unicast 用于点对点传输的 ip 地址，这种情况仅限于一个发送和一个接受者

Local 与分配到路由器接口的地址匹配

Broadcast 这个 ip 包从 ip 子网的一个点到其他所有点发送信号

Multicast 这种类型的 ip 地址负责从一个或多个点到其他一系列点的传输

Dst-limit(整型/time{0.1},整型，dst-address/dst-port/src-address{+},time(0.1))在每个目的 ip 或者每个目的端口库上限制每秒数据包绿（pps）。与 limit 匹配相反，每个目的 ip 地址/目的端口都有自己的限度。其选项如下（按出现次序）：

Count 最大平均包率，以 pps 衡量，除非跟随在 time 选项之后

Time 指定包率衡量的时间间隔

Burst 以成组方式匹配的包数量

Mode 包率限制分类方式

Expire 指定已记录的 ip/端口将被删除的过期时间，时间间隔。

Dst-port(整型：0..65535-整型：0…65535{*})目的端口数或范围

Hotspot(multiple chioce:from-client/auth/local-dst)从各种不同德 hot-spot 中匹配从客户获得的包。所有值都可以被取消

From-clicent 如果一个包来自于 hotspot 客户则为真

Auth 如果一个包来自验证用户则为真

Local-dst 如果一个包来拥有本地目的 ip 地址则为真

Icmp-options（整型：整型）与 icmp 的 type:code 域匹配

In-interface(name)-interface the packet has entered the router through

Ipv4-options(any/loose-source-routing/no-record-route/no-router-alert/no-source-routing/

No-timestamp/none/record-route/router-alert/strict-source-routing/timestamp)与 ivp4 标题选项匹配

Any 与 ipv4 选项中至少一个匹配

Loose-source-routing 与发射源路由选项的包进行匹配。次选项一般用于路由基于源提供信息的因特网数据报

No-record-alert 一以无路由警报选项匹配包

No-source-routing 一源路由选项匹配包

No-timestamp 以无时间印章选项匹配包

Record-route-以记录路由选项匹配包

Router-alert 以路由警报选项匹配包

Strict-source-routing 以严密的源路由选项匹配包

Timestamp 以时间印章选项匹配包

Jump-traget（dstnat/srcnatname）将要跳转的目标链名称，如果使用了动作 action=jump

Limit-（整型/time{0.1}，整型）按给定限度限制包匹配率。对于减少日志信息数量有用

Count 最大平均包率。以 PPS 衡量，除非跟随在 time 选项之后。

Time-指定包率衡量的时间间隔

Burst-以成组方式匹配的包数量

Log-prefix(text)所有写入日志的信息都包含次中指定的前缀，与 action=log 一起使用

Nth（整型，整型：0..15，整型{0.1}与特定的由规则获取的第 n 个包匹配。16 个可用计算器之一可被用来计算包数

Every-匹配每第 every+1 个包。例如：如果 every=1 那么规则匹配每第二包

Counter-指定要使用的计数器

Packet-一给定包的数量进行匹配。显然地，这个值必须在 0 和 every 之间。如果这个选项用于一个给定的计数器。那么在这个选项里必须至少有 every+1 个规则，一包含所有在 0 和 every 之间的值

Out-interface(name)离开路由包的接口

Packet-size（整型：0…65535-整型：0…65535{0，1}）按字节匹配指定大小或大小范围的包

Min 指定大小范围或独立的值的下限

Max-指定大小范围的上限

Phys-out-interface(name)与添加到一个桥设置的桥端口物理输入设备匹配，仅在数据包从桥到达并通过路由器时有用

Phys-in-interface(name) 与添加到一个桥设置的桥端口物理输出设备匹配，仅在数据包从

桥离开并通过路由器时有用

Protocol(ddp/egp/encap/ggp/gre/hmp/icmp/idrp-cmtp/igmp/ipencap/ipip/ipsec-ah/ipsec-esp/iso-tp4/ospf/pup/rdp/rspf/st/tcp/udp/vmtp/xns-idp/xtp/整型)与由协议名称或编号指定的特定 ip 协议匹配。如果你想指定端口就应该进行这个配置

Psd(整型，time,整型，整型)试图探测 TCP 及 udp 扫描。建议对高号码端口分配低权减少被误判的频率，例如来自被动模式的 FIP 迁移

Weighthreshold 来自不同主机且被作为端口扫描系列的带有不同目的端口的最新 TCP/UDP 包的总权重值

Delaythreshold 来自同意主机且被当作可能端口扫描子序列带有不同目的端口的包延迟

Lowportweight 特权目的端口（<＝1024）的数据包权重值

Highportweight 非特权目的端口（<＝1024）的数据包权重值

Random(整型)以给定概率随机匹配包

Router-mark(name)对 mangle 标记的特定路由的包权重进行匹配

Same-not-by-dst(yes/no)当选择要与 action=same 规则匹配的包的新源 ip 地址时指定是否对目的 ip 地址进行计算

Src-address/netmask/ip address)指定源 ip 包产生的地址范围

Src-address-list(name)与用户定义的地址列表中的数据包源地址匹配

Src-address-type(unicast/local/broadcast/multicast)与 ip 包的源地址类型中的一个匹配

Unicast-用于点对点传输的 ip 地址。这种情况仅限于一个发送者和一个接受者

Local 与分配到路由器接口的地址匹配

Broadcast-这种类型的 ip 地址负责重 hige 或多个点到其他一系列的传输

Src-mac-address(MAC address)源 MAC 地址

Src-port(整型：0…65535－整型：0…65535（*）)源端口数或范围

Tcp-mss(整型：0…65535)－与 ip 包的 TCP MSS 值匹配

Time(time-time.sat/fri/thu/web/tue/mon/sun{+})允许产生基于数据包达到时间和日期的过滤器，或者对于本地产生的数据包的离开时间日期

To-address(ip address-ip address{0.1};default:0.0.0.0.)取代初始 ip 包地址或地址范围

To-ports(整型：0..65535－整型 0…65535『0，1』) 取代初始 ip 包端口的端口或端口范围

Tos(max-rellabilty/max-throughput/min-cost/min-delay/normal)对 ip 头服务类型（TOS）域的值指定一个匹配

Max-reliablility 最大的可靠性（TOS=4）

Max-throughput 最大的吐量（TOS=8）

Min-cost 最低的成本的延迟（TOS=16）

Normal 普遍服务（TOS=0）

源地址 nat

如果你想在 ISP 给你的 10.5.8.109 地址后隐藏你的 192.168.0.0/24 的专用区域网，你应使用 mikrotik 路由器的源网络地址翻译特性。当数据包通过路由器时，伪装将把重 192.168.0.0？24 产生的源 ip 地址和包端口改变成路由器的 10.5.8.109 地址，为了使用伪装，必须向 nat 配置中添加一个带有隐藏动作的源网络动作翻译规则：

/ip firewall nat add chain=srcnat action=masquerade out-interface=pulic

所有从 192.168.0.0/24 出去的向外连接都将使用路由器的 10.5.8.109 作为源端口。因特网将不可能访问本地地址。如果你允许对本地网络服务器访问，你应使用目的网络地址翻译（nat）。

RouterOS 支持两种隐藏私有网络方式，masquerade 与 src-nat 都是改变源 ip 地址后一个数据包的端口，masquerade 和 source nat 典型的应用都是私有网络隐藏在一个或多个外网后，设置一个新的源地址 nat

masquerade 使用的是路由器默认的 ip 地址
src-nat 需要指定 to-address

masquerade 操作

add chail=srcnat src-address=192.168.0.0/24 action=masqerade out-interface=WAN

src-nat 操作

add chail=srcnat src-address=192.168.0.0/24 action=src-nat to-address=10.5.8.109 out-interface=WAN

目的地址 nat

如果你想使用公网 ip 地址 10.5.8.200 访问本地地址 192.168.0.109，你应该使用 mikrotik 有了钱的目的地址翻译特性，同样的，如果你允许本地服务器与公网 ip 进行通信，你就需要使用源地址翻译。

对公用接口添加公用 ip：

/ip address add adress =10.5.8.200/24 interface=public

添加允许外部网络访问本地服务器的规则：

/ip firewall nat add chain=dstnat dst-address=10.5.8.00 action=dst-nat / to-addresses=192.168.0.109

添加规则本地服务器能够与外部网络通信，并将其源地址翻译为 10.5.8.200

/ip firewall nat add chain =srcnat src-address=192.168.0.109 action=src-nat / to-address=10.5.8.200

Dst-nat 数据转移

通过使用 dst-nat 操作转移 ip 数据或端口到指定的主机上，如我们可以将内网所有访问 tcp/80 端口的数据转移到令外一个主机的 192.168.0.100

/ip firewall nat chain =src — nat protocol=tcp dst-port=80 action=dst-nat to-address=192.168.0.100

Dst-nat 数据重定向

Redirect 是改变目标 ip 地址或 ip 数据的端口，指定访问数据转移到本地，与 DST-NAT 不同的是 Redirect 不需要指明 to-address,一个将 tcp/80 端口重定向到本地的测试

Chain=dstnat action=Redirect protocol=tcp dst-port=80

1：1nat 实例
如果你想重公用 ip 子网 11.11.11.1/32 访问本地的 2.2.2.2/32，你应该使用目的地址翻译以及源地址翻译特性设置 action=natmap。

/ip firewall nat add chain=dstnat dst-address=11.11.11.1 \ action=netmap to-addresses=2.2.2.2
/ip firewall nat add chain=dstnat dst-address=2.2.2.2 \ action=netmap to-addresses=11.11.11.1

连接状态

操作路径：/ip firewall connection

连接追踪用于维护连接状态信息，例如源目的 ip 地址和端口，连接状态，协议类型和超时。特定连接的状态包含：
Established 意思即数据包时已知连接的一部分，new 意思为数据包开启了一个新连接，related 意为数据包开始了一个新连接，每与一个已存在连接想联系，如 FTP 数据传输或 ICMP 错误信息，invalid 意为数据包不属于任何一个已建立的连接。

注：连接追踪是对本地产生的数据包在 perouting 链或者 output 链完成的。

另一个不能被过高估计追踪功能是 nat 对其的需要。你应该清徐非你启用了连接追踪否则 nat 是不能完成的，对 p2p 协议识别也一样。连接追踪业在进一步处理前会从碎片中收集 ip 包。

/ip firewall connection 状态列表包含的最大数连接时由路由器的初始物理内存大小决定的。因此，例如一个 64M RAM 的路由器可以容纳最多 65535 连接的信息，128M RAM 的路由器就可以增加到 130000 以上。因此请确定你的路由器配置了足够量的内存以便可以适宜地处理所有连接。

属性描述

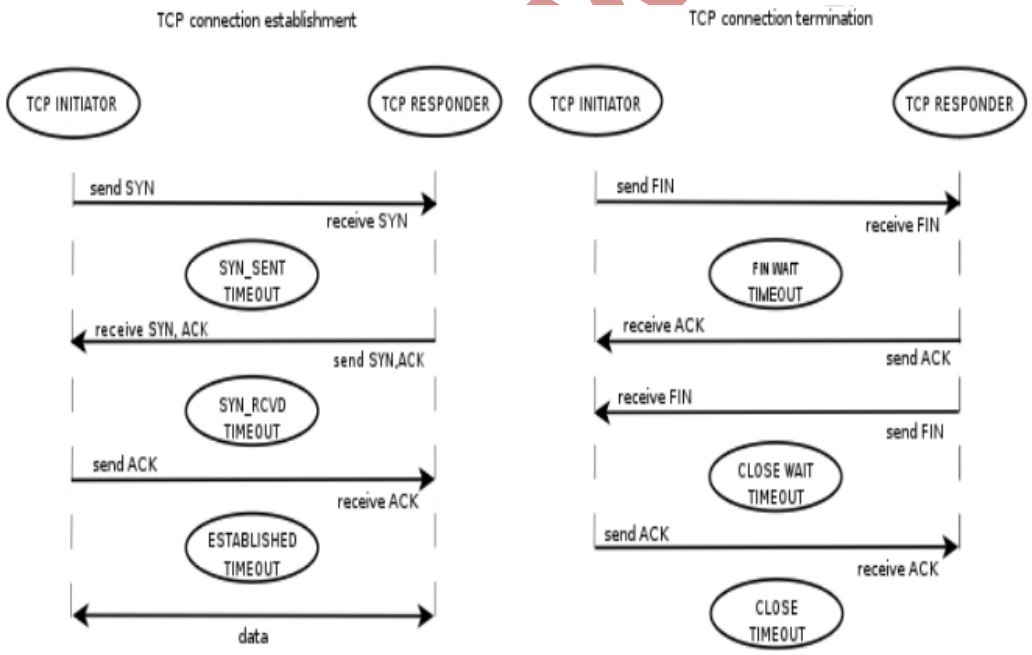
Connection-mark(只读：text)mangle 中设置的连接标记
Dst-address(只读：ip address:port)连接建立到得目的地址和端口
Protocol(只读：text)p2p 协议
Reply-src-address(只读：ip address：port)从源地址和端口建立的响应连接

Reply-dst-address（只读：ip address：port）连接建立到得目的地址和端口
Src-addres(只读：ip address：port)从源地址和端口建立的连接
Tcp-state(只读：text)TCP 连接状态
Timeout(只读：time)直到连接超时的时间量
Assured(只读：true/false)显示时否看到对该条登记的最后一个包的回应
Icmp-id(只读：整型)每个 ICMP 包都会在被发送时得到一个为其设定的 ID，并且当接收器收到了 ICMP 信息时，它会在新的 ICMP 信息内设定同样的 ID 以使发送器能识别回应并能够用适当的 ICMP 请求连接它。
Icmp-option(只读：整型)icmp 类型和代码域
Reply-icmp-id(只读：整型)包含已接收包的 ICMP ID
Reply-icmp-option(只读：整型)已接收包的 ICMP 类型和代码域
Unrlied(只读：true/false)显示是否请求未被回应

连接跟踪

操作路径：/ip firewall connection tracking

连接追踪提供了几个链接超时（timeout）当特定的超时超过了相应的条目将会从连接状态列表中删除。下面的图描绘了典型的 TCP 连接建立和终端以及在这些处理过程中发生地 tcp 超时：



属性描述

Count-curent(只读：整数)在连接状态列表中记录的当前连接数
Count-max(只读：整数)取决于总内存量的连接状态列表，自动计算出最低连接数
Enable(yes/no;默认：yes)允许或禁止链接追踪，nat 被使用的情况下必须开启
Generic-timeout(时间；默认：10M)；连接列表中追踪既非 TCP 有非 UDP 包的条目的最大时

间量将会在看到匹配此条目最后一个之后存活
Icmp-timeout(时间；默认：10S)连接追踪条目将在看到 ICMP 请求后存活最大时间量
Tcp-close-timeout(时间；默认：10s)TCP 连接追踪条目在看到连接复位（RST）或来自连接释放初始化机连接终端请求确认通知（ACK）之后存活的最大时间
Tcp-close-wait-timeout t(时间；默认：10s)当来自应答器的终端请求（FIN）之后连接追踪条目存活的最大时间
Tcp-established-timeout t(时间；默认：1d)当来自拦截初始化机的确认通知后连接追踪条目存活的最大时间
Tcp-fin-wait-timeout t(时间；默认：10s)当来自连接释放初始化机的连接终端请求（FIN）后存后连接追踪条目存活的最大时间
Tcp-syn-received-timeout t(时间；默认：1M)当匹配连接请求（SYN）之后连接追踪条目存活的最大时间
Tcp-syn-sent-timeout(时间；默认：1M)当来自连接初始化机的连接请求（SYN）后连接追踪条目存活的最大时间
Tcp-time-wait-timeout (时间；默认：10S)当紧随连接请求（SYN）的连接终端请求（FIN）之后或在看到来自连接释放初始机的其他终端请求（FIN）之后连接追踪条目存活的最大时间
Udp-timeout (时间；默认：10S)当匹配此条目的最后一个之后连接追踪天哦么存活的最大时间
Udp-stream-timeout (时间；默认：3m)在匹配此链接（连接追踪条目是确定的）最后一个包的响应被看到之后连接追踪条目存活的最大时间。他用与增加对 H323,VoIP 当连接的超时】

注：最大超时值取决于再连接状态列表中的连接数量。如果在列表中连接数量大于：
连接的最大数量的 1/16,超时值将为一天
连接的最大数量的 3/16,超时值将为 1 小时
连接的最大数量的 1/2,超时值将为 10 分钟
连接的最大数量的 13/16,超时值将为 1 分钟

如果超时值超过了上面列出的值，那么将使用更小的值，如果链接追踪超时值小于数据包率。比如：在下一个包到达之前超时就过期了，那么 nat 和 statefull-firewalling 将停止工作。

注：tracking 功能被关闭，nat 功能叶将会失效：如果你在不考虑启用 nat 功能情况，可以关闭掉 tracking。

第十二章 分类标记（mangle）

Mangle 允许对 ip 数据包做特殊的标记，mangle 是通过修改指定的 ip 数据包头字段，去标记 ip 数据包的特征能标记端口，ip，协议，TCP 协议和相应的 ip 数据流。Mangle 属于综合性功能，所以在路由，流量控制和其他相应功能中都会涉及到。

需要功能包：system
需要等级：level1
操作路径：/ip firewall mangle
协议标准：ip

Mangle 介绍

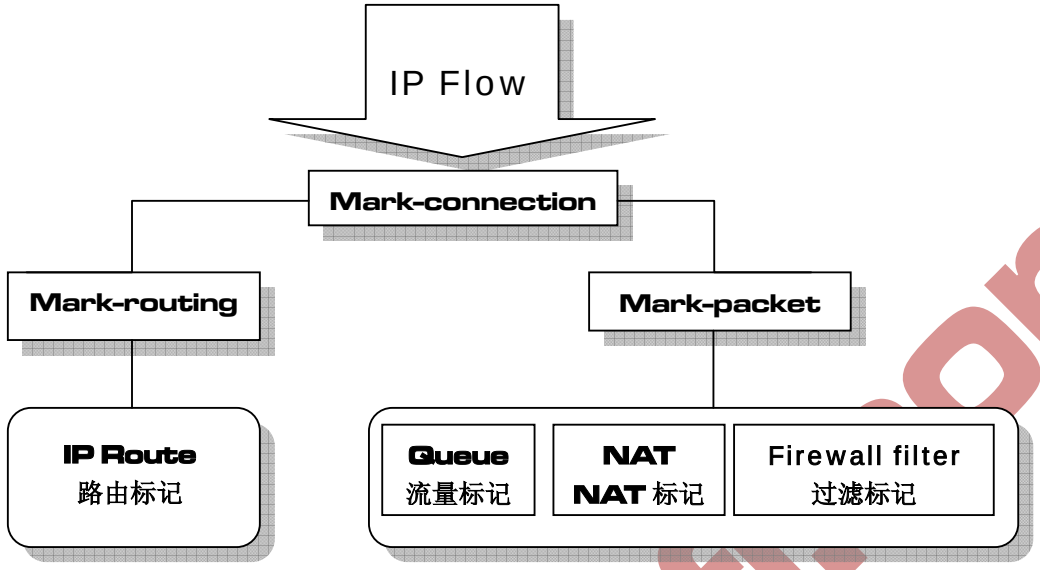
Mangle 是一种标记器，标记特殊的数据包等待将来处理，在 routerOS 中许多其他功能组件会使用到他，如 queue-trees 和 nat,他们识别到一个数据包了标记的便会做相应的处理。Mangle 标记仅存在该路由器中，他们无法传输到网络中去。
根据数据传输方式不同可以选择：

Prerouting 两样前，常用于标记策略和端口路由
Input 进入路由器的数据
Foreward 通路由转发，用于修改 TTL.TCP-MSS 和浏览控制规则
Output 数据输出
Prostrouting：路由后

标记 ip 数据流的三种类型：

Mark-connection: 标记所有 IP 流的链接
Mark-packet: 标记 IP 流中数据包
Mark-routing: 标记 IP 流中 IP 数据包的路由信息

三种类型的关系，所有的在 IP 数据包传输前，首先需要通过建立 TCP/UDP 链接，进行传输。所以当数据通过 IP 流进入 Mangle 后，建立相应的链接标记，并从链接标记中提取数据包，做处理。图示如下：



Mangle 应用

peer-to-peer 传输标记

保证优质的网络链接，如 VoIP 和 HTTP 等为最优先级，将 P2P 的优先级设置为最低 RouterOS QOS 操作首先使用 mangle 标记不同类型的传输，然后把它们放入的 queues 做不同的限制。下面的事例是强迫 P2P 的总的传输不能超过 1Mbps，其他的传输链接则扩大链接带宽和优先级：

[admin@NAT] > /ip firewall mangle add chain=forward p2p=all-p2p action=mark-connection new-connection-mark=p2p_conn
[admin@NAT] > /ip firewall mangle add chain=forward connection-mack=p2p_conn action=mark-packet new-mark=p2p
[admin@NAT] > /ip firewall mangle add chain=forward packet-mark=!p2p_conn action=mark-packet new-packet-mark=other
[admin@NAT] > /ip firewall mangle print
Flags: X – disabled, I – invalid, D – dynamic
0 chain=forward p2p=all-p2p action=mark-connection new-connection-mark=p2p_conn

1 chain=forward connection-mark=p2p_conn action=mark-packet new-packet-mark=p2p

2 chain=forward packet-mark=!p2p_conn action=mark-packet new-packet-mark=other

[admin@NAT] >
[admin@NAT] > /queue tree add parent=Public packet-mark=p2p limit-at=1000000

max-limit=100 000 000 priority=8

[admin@NAT] > /queue tree add parent=Local packet-mark=p2p limit-at=1000000 max-limit=100000000 priorty=8

[admin@NAT] > /queue tree add parent=Public packet-mark=other limit-at=1000000 max-limit=100000000 priority=1

[admin@NAT] > /queue tree add parent=Local packet-mark=other limit-at=1000000 max-limit=100000000 priorty=1

Mangle 限制 2 级代理

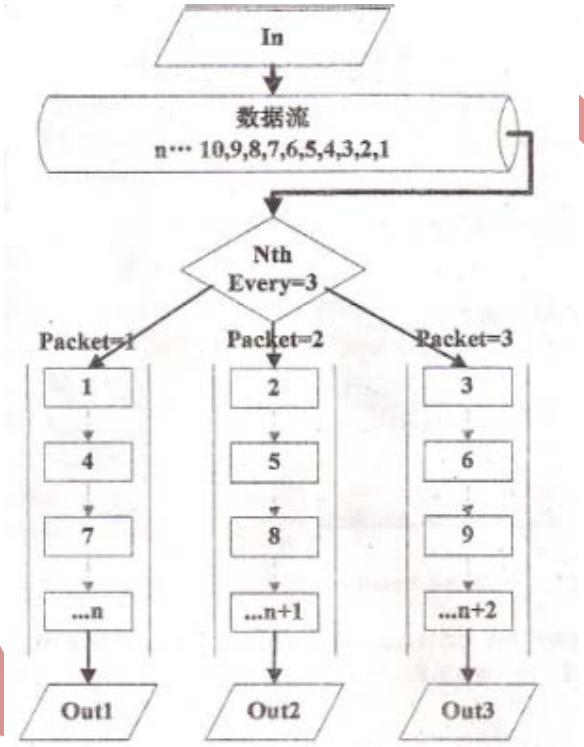
通过 mangle 限制 2 级代理，但对端口的 http 代理无效，我们可以指定 in-interface 或者指定目标数据到内网的 IP 地址，即 dst-address

[admin@Mikrotik] /ip firewall mangle> add chain=forward out-interface=lan action=change-ttl new-ttl=set:1
[admin@Mikrotik] /ip firewall mangle>print chain=forward
Flags: X – disard, I – invalid, D – dynamic
8 chain=forward action=change-ttl new-ttl=set:1 out-interface=lan

地十三章 RouterOS v3 Nth

在 v3.0 中 NTH 工具做了一点修改，仅只有两个参数”every”和”packet”。每个规则都有自己的计数器。
当规则收到数据包，当前规则的计数器会增加 1，如果计数器匹配值”every”与数据包匹配，计数器将重新设置为 0。使用 Nth 我们可以将一串链接通过计数器分离，比如可以将链接分配为多个组，重新排列链接序列。

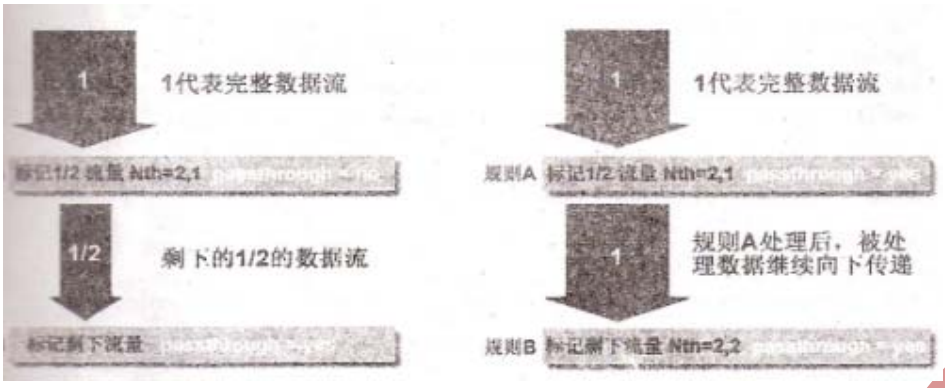
Nth – 匹配特定的第 N 次收到的数据包的规则。一个计数器最多可以计数 16 个数据包
Every – 匹配每 every 数据包，同时指定 counter（计数器值）
Packet – 匹配给定的数据数，例如，Nth=3,1，匹配 3 个数据包的第 1 个



上图，可以看到数据流从 1-n 的数据，被 Nth 分为 3 个计数器，并根据 Packet 重新排列数据流的队列。Nth 我们可以应用的范围，包括多线路的负载均衡、内网多台 ftp 访问、以及其他的应用。

Passthrough 对 Nth 的控制

实现相同的 Nth 结果时，改变 Passthrough 参数（Passthrough 为是否将该规则数据继续向下传递，no 为停止向下传递，yes 则相反，具体参考 Mangle 章节）会得到不同的规则配置，首先要知道 Mangle 标记捕获数据时先进先出算法，即从上往下执行，我们在配置 Mangle 的 Nth 规则，需要注意前后顺序。如我们把数据流标记为两个组，即一条为 1/2， 另一条也为 1/2，把一个数据流看成“1”，而我们把可以通过两种方法配置：



当我们需要将数据流标记为 3 组时，即每条规则为 1/3。配置方法同样有两种，如下图

如同从下面的图上看到，使用和不使用 Passthrough 的区别，在于流量是否继续向下传递。

例如，有双线接入，并采用 Nth 的双线负载均衡。首先我们需要在 mangle 里标记链接，如果配置 Passthrough=no 参数，Nth 参数配置仅需要一条规则，即标记置 50%流量，首先我们需要标记链接：

/ip firewall mangle
Add chain=prerouting new-connection-mark=AAA nth=2,1 action=mark-connection passthrough=no;

抓取完前 50%的数据后，剩下的流量只需要做一个默认的标记剩下的数据即可。

Add chain=prerouting new-connection-mark=BBB action=mark-connection

当变成 3 条线时，第一条规则标记所有数据包并对比所有流量的 1/3，第二条规则标记剩下 2/3 数据包的 50%，第三条规则标记和对比所有剩下的数据包（所有数据包的 1/3）

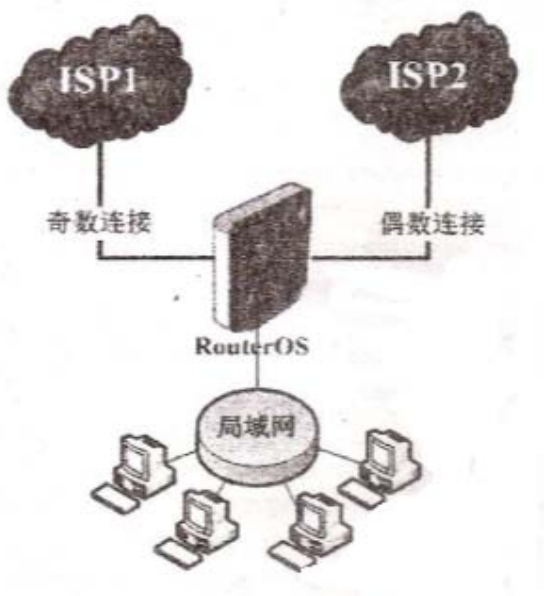
/ip firewall mangle
Add action=mark-connection chain=prerouting new-connection-mark=AAA nth=3,1 passthrough=no;
Add action=mark-connection chain=prerouting new-connection-mark=BBB nth=2,1 passthrough=no;
Add action=mark-connection chain=prerouting new-connection-mark=CCC;

同样我们有的数据包并且每个规则对比每 3 个数据包。

/ip firewall mangle
Add action=mark-connection chain=prerouting new-connection-mark=AAA nth=3,1 passthrough=yes;
Add action=mark-connection chain=prerouting new-connection-mark=BBB nth=3,2 passthrough=yes;
Add action=mark-connection chain=prerouting new-connection-mark=CCC nth=3,3 passthrough=yes;

Nth 在负载均衡的应用

下面我们看一个世纪的双线接入的 Nth 应用实例，假设我们有两条 ISP 的线路，我们通过 Nth 的方法实现负载均衡，让 2 条同样 ISP 线路达到合并带宽的作用。



根据 Nth 的原理我们可以将来至内网的连接分为两组，即一组为奇数连接、一组为偶数连接，即奇数走一条线路，偶数走另一条线路。因为我们定义的是连接状态为 new，即新建立的链接，对正常的访问没有任何影响，每个新建立所产生的后续数据都会按照原来的线路链接运行。

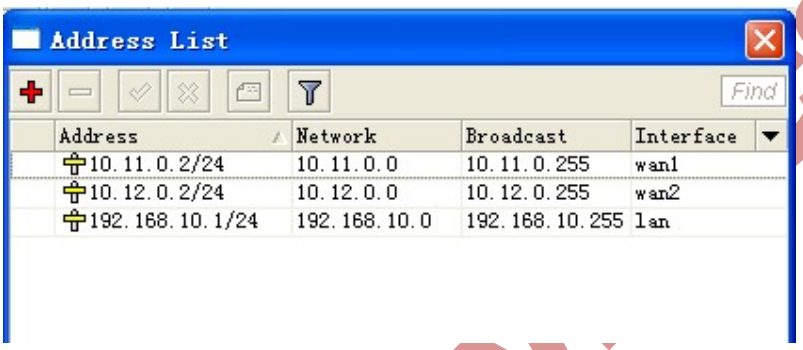
我们从所有的链接中，提取每次新建立的链接 connection=new，并对他们做 Nth 的标记，将

这些链接中相关的奇数（odd）包和偶数（even）包分离开，并走两个不同的网关（ISP 与 ISP2）出去。这样就能保持每次链接的持续性。

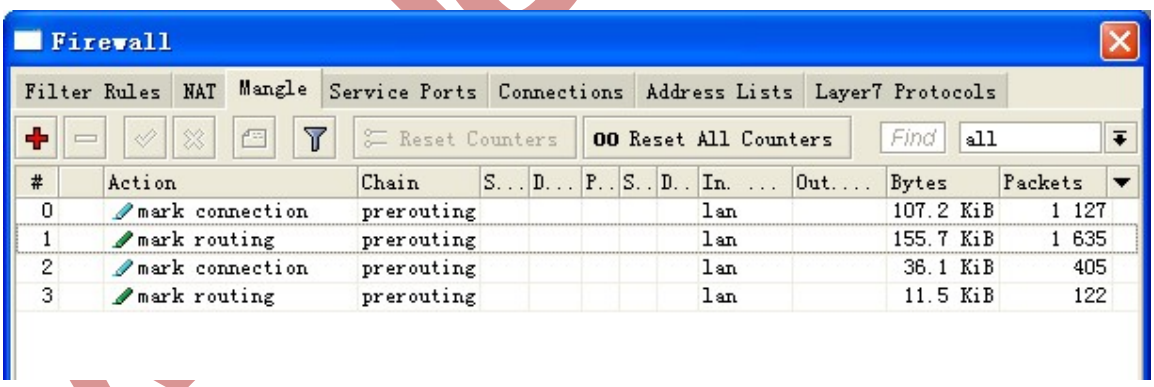
网络参数如下：

Wan1:：ip 地址 10.11.0.2/24，网关 10.12.0.1
Wan：ip 地址 10.12.0.2/24， 网关 10.12.0.1
Lan：192.168.10.1/24

首先配置 IP



接下来在 ip firewall mangle 中标记奇数和偶数的 Nth，并配置路由标记，奇数 Nth 链接标记取名为 odd，偶数连接标记取名为 even，将奇数的路由标记取名为 ISP1，将偶数的路由标记取名为 ISP2，如下：



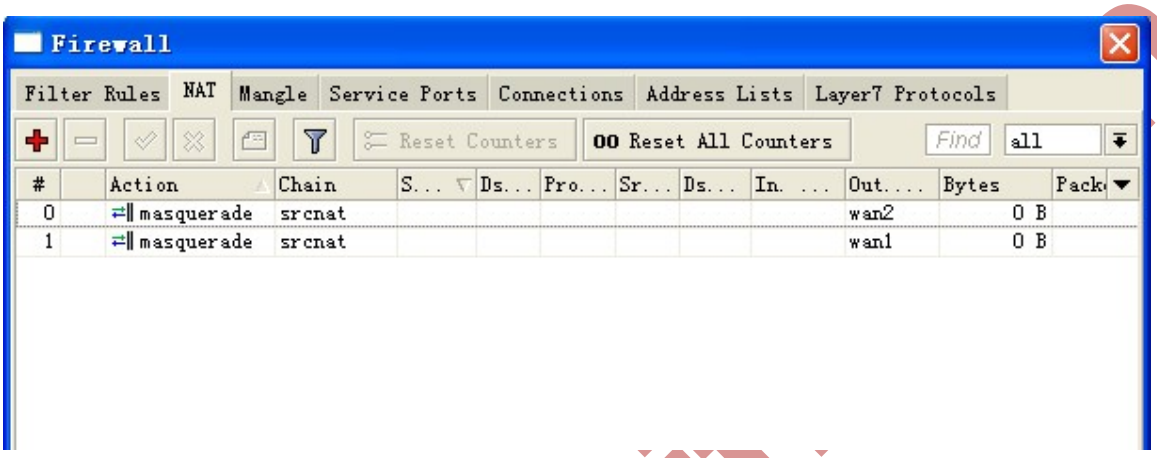
命令行配置如下：

[admin@Mikrotik] /ip firewall mangle> print
Flags: X – disabled, I – invalid, D – dynamic
0 chain=prerouting action=mark-connection new-connection-mark=odd passthrough=yes connection-state=new in-interface=lan nth=2,1
1 chain=prerouting action=mark-routing new-routing-mark=odd passthrough=yes in-interface=lan connection-mark=odd
2 chain=prerouting action=mark-connection new-connection-mark=even passthrough=yes

connection-state=new in-interface=lan nth=2,2

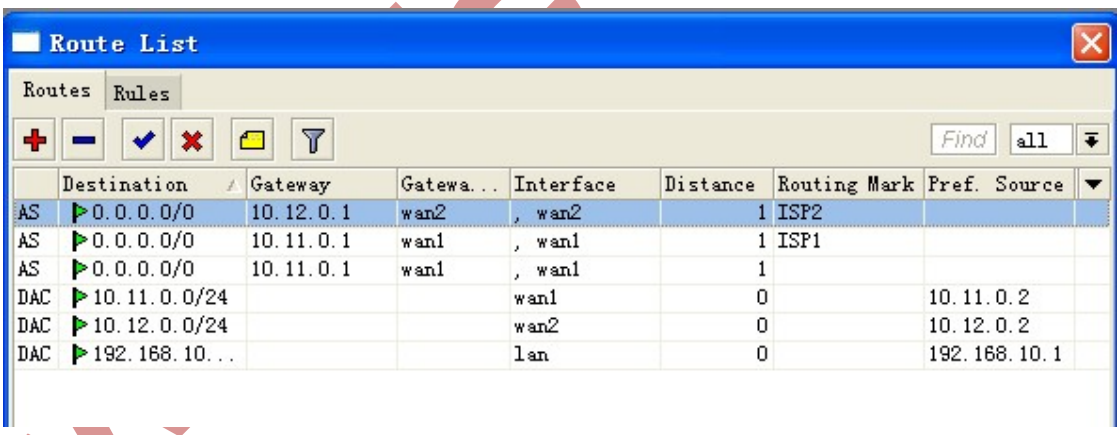
3 chain=prerouting action=mark-connection new-connection-mark=odd passthrough=yes in-interface=lan connection-mark=even

NAT 配置



路由配置

进入 ip route 中配置路由规则，配置 10.12.0.1 对应 ISP2 的路由标记，10.11.0.1 对应 ISP1 的路由标记，我们用 10.11.0.1 作为路由器本身的默认网关。



命令行配如下：

/ ip route

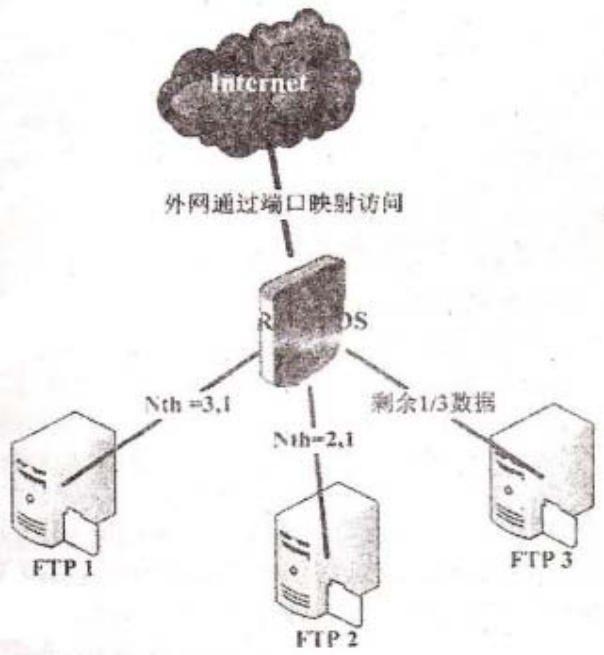
Add gateway=10.11.0.1 routing-mark=ISP1

Add gateway=10.12.0.1 routing-mark=ISP2

这样双线的 Nth 负载均衡就配置完成，建议这样的负载均衡使用在相同 ISP 的线路上，并且带宽接近。采用 Nth 线路时有使用者反映出现某些网银无法打开的问题。

Nth 在端口映射的应用

通过 Nth 的原理我可以实现一些特定的应用，比如应用于 FTP 服务的端口映射，当我们有大量信息需要向互联网共享时，可能我们一台 FTP 服务器无法承担所有的数据流量，我们可以通过建立多台服务器来分担流量，在不必修改 FTP 端口的情况下，通过 Nth 均衡分流数据到 3 台 FTP 服务器上，如下图内网的 3 个 FTP 服务器：



我们通过建立 3 条 nat 规则，区别 3 个不同的服务器连接，在 nat 中没有同时做 Passthrough 的选项，而且在 nat 规则中采用的是先进先出算法，所以我们只能采用先标记 1/3， 在标记 1/2，最后标记剩下的数据的方法处理 3 条线路的均衡操作。

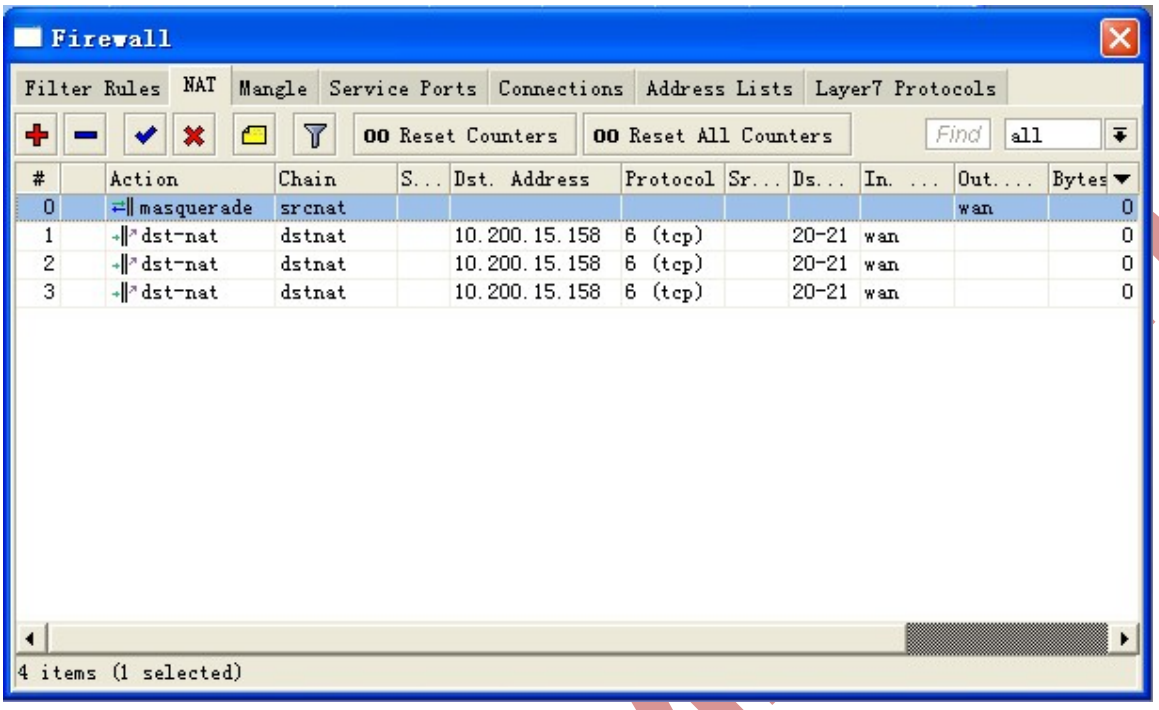
我们的网络环境如下

Wan：ip 地址为 10.200.15.158/24 网关为 10.200.15.1
Lan：ip 地址为 192.168.10.1/24
内网的 3 个 FTP 服务器的 IP 地址分别是 192.168.10.2 , 192.168.10.3 , 192.168.10.4

在配置完 IP 地址后，我们进入 ip firewall nat 配置 nat 规则，首先我们需要配置基本的 nat 伪装规则，将内网的私有 IP 地址转换为公网 IP。

/ip firewall nat
Add action=masquerade chain=srcnat disabled=no out-interface=wan

接着，设置端口映射，FTP 使用的是 TCP，20-21 端口，我们配置 3 条 nat 的 Nth 端口映射的规则，分别指向 192.168.10.2、192.168.10.3 和 192.168.10.4 三个服务器的 IP 地址：



通过命令行配置如下：

标记前 1/3 的端口映射

Add action=dst chain=dstnat dst-address=10.200.15.158 dst-port=20-21 in-interface=wan nth=3,1 protocol=tcp to-addresses=192.168.10.2 to-ports=20-21

标记剩下 1/2 的端口映射

Add action=dst-nat chain=dstnat dst-address=10.200.15.158 dst-port=20-21 in-interface=wan nth=2,1 protocol=tcp to-addresses=192.168.10.3 to-ports=20-21

标记最后 1/3 的端口映射

Add action=dst-nat chain=dstnat dst-address=10.200.15.158 dst-port=20-21 in-interface=wan nth=2,1 pratocol=tcp to-addresses=192.168.10.4 to-ports=20-21

这样通过 Nth 分流的端口映射配置完成，这样的 Nth 操作仅适合于一次性提交和访问的数据连接。如果是带登陆验证的访问，不建议使用这种方式，会出现连接后在不同服务器上的重复认证。

第十四章 Bridge 网桥

支持以太网 MAC 等级桥接，EoIP(Ethernet over IP)，Prisn, Atheros 以及广播局域网。所有的 802.11a, 802.11b, and 802.11g 客户无线接口（ad-hoc, infrastructure 或 station 模式）都不支持这个因为 802.11 的限制。然而，在 Prism 和基于 Atheros 链接之间使用 WDS 特性（对基于卡片的 Atheros 和 Prism 芯片组）或 EoIP 进行桥接还是可能的。

为防止网络中国的环路，你可以使用生成树协议（STP/RSTP）。这个协议也可以作为备份链接的配置。主要特征：

生成树协议（STP）
快速生成树协议（RSTP）
多重桥接口
MAC 地址可以被实时监控
为路由访问的 IP 地址分配
桥接口可以被过滤及网络地址翻译
支持基于桥数据包过滤器的桥路由

快速配置指南

把接口 ether1 和 ether2 放在一个桥里：

1. 添加一个桥接口，命名为 MyBridge：
/interface bridge add name=”MyBridge” disabled=no

2. 把 ether1 和 ether2 添加到 MyBridge 接口：
/interface bridge port set ether1, ether2 bridge=MyBridge

规格

功能包需要：system
认证需要：Level 3
子目录需要：/interface bridge
标准和技术：IEEE801.1D

类似以太网的网络（Ethernet, Ethernet over IP, IEEE802.11 in ap-bridge 或 bridge 模式, WDS, VLAN）可以通过使用 MAC 桥链接在一起。桥特性允许这些不同局域网的主机互连（使用 EoIP，如果任何种类的 IP 网络互连存在其中则地理分布式网络也可以被桥接起来）好像他们是连接在一个局域网中。由于桥是透明的，他们不会再追踪路由表中出现，并且没有使用程序可以使工作在一个局域网中主机和工作在另一个局域网的主机有区别如果这些局域网桥接起来了（由于局域网互连方式的不同，不同主机间的延迟和数据率会有不同）。

网络环路可能以复杂的拓扑形式出现（有意或无意的）。如果没有特殊的处理，环路将组织网络的正常工作，因为他们可能导致雪崩一样的数据包倍增。每一个桥都运行一个计算如何组织环路的算法。STP 允许桥之间进行通信，于是他们可以协商无环路的拓扑。所有其他可能 环路的链接被当成备用，所以如果主链接失败其他的链接就可以取代他的位置。这个算法定期地互相交换配置信息（BPDU—Bridge Protocol Data Unit: 桥协议数据单元），因此所有的桥都可以用网络拓扑中最新变化的信息进行更新。STP 选择负责网络配置的根桥，像关闭和打开其他桥端口的桥。根桥是拥有最低桥 ID 的桥。

网络桥配置

操作路径：/interface bridge

为了把许多网络链接到一个桥上；必须建立一个桥接口（一会，所有需要的接口都应该像他的端口一样配置）。一个 MAC 地址将会被分配给岁有的桥接口（最小的 MAC 地址将会被自动选择）

属性描述

Ageing-time (时间；默认：5m) – 一个主机信息可以被保存在桥数据库的时间
Arp (disabled | enabled | proxy-arp | reply-only；默认：enabled) – 地址解析协议设置
Forward-delay（时间；默认：15s）- 在桥接口初始化阶段（例如：在路由器启动或启用接口之后）桥正常工作之前监听/学习状态所有的时间
Garbage-collection-interval（时间；默认：4s）- 丢弃桥数据库中老的（过期的）主机词条的频率。无用存储单元收集过程消除比 ageing-time 属性定义的更老的词条。
Hello-time（时间；默认：2s）- 给其他桥发送 hello 包的频率
Mac-address (只读：MAC 地址) – 接口的 MAC 地址
Max-message-age (时间；默认 20s) – 保留从其他桥接受 hello 信息的时间长短
Mtu (整型；默认：1500) – 最大传输单元
Name（名称；默认：bridgeN）- 桥接口的描述性名称
Prlority (整型：0..65535；默认：32768) – 桥接口优先级。STP 使用优先级参数决定如果最后两个端口形成了环路应保留哪个
Stp(no | yes；默认：no) – 是否启用生成树协议。桥环路仅在这个属性启用时才会被阻止。

添加并启用一个转发所有协议的桥接口：

```
[admin@Mikrotik] interface bridge> add; print
Flags: X – disabled, R – running
    0 R name="bridgel" mtu=1500 arp=enabled mac-address=61:64:64:72:65:73 stp=no
  priority=32768 ageing-time=5m forward-delay=15s
    Garbage-collection-interval=4s hello-time=2s max-message=20s
[admin@Mikrotik] interface bridge> enable 0
```

端口设置

操作路径：/interface bridge port

子目录用于使接口受制于一个特殊的桥接口。

属性描述

Bridge（名称；默认：none）- 那些接口被定义为 Bridge 接口
None – 接口没有被定义到任何桥中
Interface（只读：名称）- 接口名，包含在一个桥内
Path-cost(整型：0..65535；默认：10) – STP 使用的用以决定最佳路径代价
Priority（整型：0..255；默认：128）- 同一网络中相比较于其他接口的接口优先级

注：从 v2.9.9 版本起，列表中的端口应被添加（add）而非设置（set），请看下面的例子：

把 ether1 和 ether2 分到已创建的桥 Bridge1 中（v2.9.9 以前）：

```
[admin@MikroTik] interface bridge port> set ether1,ether2 bridge=bridge1
[admin@MikroTik] interface bridge port> print
 # INTERFACE   BRIDGE PRIORITY PATH-COST    HORIZON
 0 ether1      bri... 0x80     10           none
 1 ether2      bri... 0x80     10           none
 2 wlan1       none     128     10             none
[admin@MikroTik] interface bridge port>
```

把 ether1 和 ether2 分到已创建的桥 Bridge1 中（V2.9.9 起）：

```
[admin@MikroTik] interface bridge port> add ether1,ether2 bridge=bridge1
[admin@MikroTik] interface bridge port> print
 # INTERFACE   BRIDGE PRIORITY PATH-COST     HORIZON
 0 ether1      bri... 0x80     10            none
 1 ether2      bri... 0x80      10            none
[admin@MikroTik] interface bridge port>
```

桥接口查看

命令名：/interface bridge monitor

用于监听一个桥的当前状态。

属性描述

Bridge-id（文本）- 桥 ID，以如下形式 Bridge-priorith，Bridge-MAC-address
Designated-root（文本）- 根桥的 ID
Path-cost（整型）- 到根桥所需总代价

Root-Port（名称）- 根桥链接的端口

监听一个桥：

```
[admin@MikroTik] interface bridge> monitor bridge1
       state: enabled
   current-mac-address: 00:00:00:00:00:00
         root-bridge: yes
       root-bridge-id: 0x8000.00:00:00:00:00:00
       root-path-cost: 0
           root-port: none
          port-count: 2
  designated-port-count: 0

[admin@MikroTik] interface bridge>
```

桥端口监测

命令名：/interface bridge port monitor

属性描述

Designated-Port（文本）- 指定根桥端口
Designated-root（文本）- 最靠近根桥的桥 ID
Port-id（整型）- 端口 ID，代表端口优先级和端口号且是唯一的
Status（disabled | blocking | listening | forwarding）- 桥端口的状态：
Disabled – 端口被禁用。没有帧被转发，没有桥协议数据单元（BPDUs）被收到
Blocking – 端口不转发任何帧但监听 BPDU
Listening – the port does not forward any frames, but listens to them 端口不转发任何帧但监听
Learning – 端口不转发任何帧但学习 MAC 地址
Forwarding – 端口转发帧并学习 MAC 地址

监听一个桥端口：

```
[admin@MikroTik] interface bridge port> mo 0
            state: enabled
   current-mac-address: 00:00:00:00:00:00
          root-bridge: yes
       root-bridge-id: 0x8000.00:00:00:00:00:00
       root-path-cost: 0
           root-port: none
          port-count: 2
  designated-port-count: 0
-- [Q quit|D dump|C-z pause]
```

桥主机列表

命令名：/interface bridge host

属性描述

Age (只读：时间) – 从主机获得最后一个包开始的时间
Bridge（只读：名称）- 属于词条（entry）的桥
Local（只读：标记）- 主机词条是否是桥本身的
Mac-address (只读：MAC 地址) – 主机 MAC 地址
On-interface （只读：名称）- 主机所连接的桥接的接口

获得活动的主机列表：

```
[admin@MikroTik] interface bridge host> print
Flags: L - local, E - external-fdb
   BRIDGE            MAC-ADDRESS       ON-INTERFACE      AGE
   bridge1           00:00:B4:5B:A6:58 ether1            4m48s
   bridge1           00:30:4F:18:58:17 ether1            4m50s
 L bridge1           00:50:08:00:00:F5 ether1            0s
 L bridge1           00:50:08:00:00:F6 ether2            0s
   bridge1           00:60:52:0B:B4:81 ether1            4m50s
   bridge1           00:C0:DF:07:5E:E6 ether1            4m46s
   bridge1           00:E0:C5:6E:23:25 prism1            4m48s
   bridge1           00:E0:F7:7F:0A:B8 ether1            1s
[admin@MikroTik] interface bridge host>
```

桥防火墙

操作路径：/interface bridge filter, /interface bridge nat, /interface bridge broute

桥防火墙执行包过滤因此提供了用于管理数据流进，流出和流经桥的安全功能。

注：在桥接接口之间的数据包就像其他 IP 流一样，也要经过类属的/ip firewall 规则（但桥过滤器总是在 IP 过滤器/NAT 之前应用，除了在 IP 防火墙输出之后执行的 output）。这些规则可以同真实的物理接受/发送接口一起使用，也可以和简单对桥接在一起的接口划分的桥接口同时使用。

有三种桥过滤器列表：

Filter – 有三个预先设定的桥防火墙链表：

Input – 其目的地市桥（进入桥设备的数据包，无论什么情况下以本地桥 MAC 地址为目标的数据）。

Output – 来自于桥（由桥设备本身处理发出的数据）。

Forward – 通过桥转发（即有桥设备转发到另外网络的数据）。

Nat – 桥网络地址翻译提供了改变遍历桥的数据包的源/目的 MAC 地址的方法。它有连条内置的链：

Scnat – 用于在一个不同的 MAC 地址后“隐藏”一个主机或者一个网络。这个链适用于通过一个桥接口离开路由器的数据包

Dstnat – 用于把一些包重定向到另一个目的地址

Broute – 使一个桥变为一个桥路由器 — 一种在一些包上其路由作用而在其他包起桥作用的路由器。它有一个预定义链：brouting，当一个包进入一个受控接口后它便进行遍历（在“Bridging Decision”之前）。

注：桥的目标网络地址翻译在桥接判定之前执行。当需要涉及到三层过滤时或者流量控制，需要将桥的 use-ip-firewall 启用，否则三层过滤和流量控制将无法工作。

你可以在桥防火墙（filter，broute and NAT）中设置数据包标记，就像用 mangle 在 IP 防火墙中设置数据包标记一样。所以用桥防火墙设置的包标记可以在 IP 防火墙中使用，反之亦然。普通桥防火墙属性在这部分描述。一些在 nat, broute 和 filter rules 之间有区别的参数将在后面的部分描述。

属性描述

802.3-sap（整型）- DSAP（目的文件服务访问点）和 SSAP（源端业务接入点）是两个 1 字节域，她们识别使用链路层服务的网络协议实体。这些字节总是相等的。两个十六进制数字可以在这里指定以匹配 SAP 字节。

802.3-type（整型）- 以太网协议类型，放置在 IEEE802.2 帧标题后面。仅当 802.3-sap 为 0xAA (SNAP — 子网链接点标题)时才生效。例如：AppleTalk 可以由跟随在 0x8098 SNAP 类型码后面的 0xAA SAP 码说明。

Arp-dst—address (IP 地址；默认：0.0.0.0/0) – ARP 目的地址

Arp-dst-mac-address（MAC 地址；默认：00:00:00:00:00:00）- ARP 目的 MAC 地址

Arp-hardware-type（整型；默认：1）- ARP 硬件类型

Arp-opcode（arp-nak | drarp-error | drarp-reply | drarp-request | inarp-request | reply | reply-reverse | request | request-reverse）- ARP opcode（数据包类型）

Arp-nat – 消极 ARP 应答（很少使用，主要在 ATM 网络中使用）
Drarp-error – 动态 RARP 错误代码，saying that an IP address for the given MAC address can not be allocated 表明一个给定 MAC 地址的 IP 地址不能分配
Drarp-error – 动态 RARP 应答，带有一个主机临时地址分配
Drarp-request – 动态 RARP 请求一个对给定 MAC 地址的临时 IP 地址
Reply – 带有一个 MAC 地址的标准 ARP 应答
Reply-reverse – 带有一个以分配 IP 地址的反向 ARP（RARP）应答
Request – 向一个已知 IP 地址的标准 ARP 请求
Request-reverse – reverse ARP(RARP) request to a known MAC address to find out unknown IP 向已知 MAC 地址询问未知 IP 地址的凡响 ARP（RARP）请求（intended to be used by hosts to find out thelr own IP address 主机有意用来查明其本身 IP 地址，类似于 DHCP 服务）
Arp-src-address（IP 地址；默认：0.0.0.0/0）- ARP 源 IP 地址
Arp-src-mac-address（MAC 地址；默认：00:00:00:00:00:00）- ARP 源 MAC 地址
Chain（文本）- 过滤器工作其中的桥防火墙链（内置或用户定义的）
Dst-address（IP 地址；默认：0.0.0.0/0）- 目的 IP 地址（仅当 MAC 协议设置为 IPv4 时）
Dst-mac-address（MAC 地址；默认：00:00:00:00:00:00）- 目的 MAC 地址
Dst-port（整型：0..65535）- 目标端口号或范围（仅对 TCP 或 UDP 协议）
In-bridge（名称）- 数据包进入的桥接口
In-intenrface（名称）- 数据包进入的物理接口（例如：桥端口）
Ip-protocol（ipsec-ah | ipsec-esp | ddp | egp | ggp | gre | hmp | idpr-cmtp | icmp | igmp | ipencap | encap | ipip | iso-tp4 | ospf | pup | rspf | rdp | st | tcp | udp | vmtp | xns-idp | xtp）- IP 协议（仅当 MAC 协议设置为 IPv4）
Ipsec-ah – IPsec AH 协议
Ipsec-esp – IPsec ESP 协议
Ddp – 数据报投递协议
Egp – 外部网关协议
Ggp – 网关-网关协议
Gre – 通过路由压缩
Hmp – 宿主监督协议
Idpr-cmtp – idp 控制报文传输
Icmp – 因特网控制报文协议
Igmp – 因特网分组管理协议
Ipencap – ip 压缩至 ip
Encap – ip 压缩
Ipip – ip 压缩
Iso-tp4 – iso 传输协议类型 4
Ospf – 开放式最短路径优先
Rdp – 靠数据包协议
St – st 数据报模式
Tcp – 传输控制协议
Udp – 用户数据报协议
Vmtp – 通用信息传输
Xns-idp – Xerox ns idp
Xtp – xpress 传输协议

Jump-target（名称）- 如果指定 action=jump，那么指定用户定义的防火墙链来处理数据包
Limit - （整型/time{0,1}，整型）- 以给定值限制包匹配率，有助于减少日志消息的总量
Count – 除非跟随在 Time 选项之后否则以包每秒（pps）衡量最大平均包率
Time – 指定包率测量的时间间隔
Burst – 要匹配的脉冲串中的包数量 8
Log-prefix（文本）- 在日志信息之前定义用于打印的前缀
Mac-protocol（整型 | 802.2 | ip | ipv6 | ipx | rarp | vlan）- 以太网有效负载类型（MAC）
Mark-flow（名称）- mark existing flow
Packet-type（broadcast | host multicast | other-host）- MAC 帧类型：
Broadcast – 广播 MAC 包
Host – 目的为桥本身的数据包
Multicast – 多重 MAC 包
Other-host – 定位到其他联合广播地址而非到桥本身的数据包
Src-address（IP 地址；默认：0.0.0.0/0）- 源 IP 地址（仅当 MAC 协议设置为 IPv4 时）
Src-mac-address（MAC 地址；默认：00:00:00:00:00:00）- 源 MAC 地址
Src-port（整型：0..65535）- 端口号或范围（仅对 TCP 或 UDP 协议）
Stp-flags（topology-change | topology-change-ack）- BPDU（网桥协议数据单元）标志。桥之间为阻止环路定期地互相交换名为 BPDU 的配置信息。
Topology-change – 拓扑变化标志是当一个桥检测到端口状态改变时设置，它命令所有其他桥丢弃它们的主机列表并重新计算网络拓扑
Topology-change-ack – 拓扑变化确认标志是作为通告数据包回应而设置的
Stp-forward-delay（time：0..65535）- forward delay timer 转发延迟计时器
Stp-hello-time（time：0..65535）- stp hello 数据包时间
Stp-max-age（time：0..65535）- 最大 STP 信息年龄
Stp-msg-age（time：0..65535）- STP 信息年龄
Stp-port（整型：0..65535）- stp 端口识别
Stp-root-address（MAC address）- 根桥 MAC 地址
Stp-root-cost（整型：0..65535）- 根桥代价
Stp-root-priority（time：0..65535）- 根桥优先级
Stp-sender-address（MAC address）- stp 信息发射机 MAC 地址
Stp-sender-priority（整型：0..65535）- 发射机优先级
Stp-type（config | tcn）- BPDU 类型
Config – 配置 BPDU
Tcn – 拓扑变化通告
Vlan-encap (802.2 | arp | ip | ipv6 | ipx | rarp | vlan) - 压缩在 VLAN 帧中的 MAC 协议类型
Vlan-id（整型：0..4095）- VLAN 识别域
Vlan-priority（整型：0..7）- 用户优先级域

注：仅当目的 MAC 地址为 01:80:C2:00:00:00/FF:FF:FF:FF:FF:FF（桥组地址）时，stp 匹配器才有效，同时 stp 应被启用。仅当 mac-protocol 为 arp 或 rarp 时 ARP 匹配器才有效。VLAN 匹配器仅对 vlan 以太网协议有效。IP 相关匹配器仅当 mac-protocol 被设置为 ipv4 时才有效

如果实际帧和 IEEE 802.2 和 IEEE 802.3 标准一致时，802.3 匹配器就会被询问（注意：它并不是在全世界网络使用的工业标准以太网帧格式）。这些匹配器对其他包会被忽视。

桥数据包过滤

操作路径：/interface bridge filter

这部分描述的是桥数据包过滤器详细的过滤选项，在一般的防火墙描述中这部分通常被省略掉了。

属性描述

Action（accept | drop | jump | log | mark | passthrough | return; default:accept）- 如果数据包匹配了其中一个规则就采取动作：
Accept – 接收包，无动作。例如：数据包通过而没有任何动作，并且没有其他规则会在相关列表/链中处理。
Drop – 悄然地丢弃包（不发送 ICMP 拒绝信息）
Jump – 跳转到有 jump-target 变量指定的链
Log – 记录数据包
Mark – 标记数据包以便后面使用
Passthrough – 忽视这条规则并到下一个。除了对包计数外像一个被禁用的规则一样动作
Return – 从跳转发生的地方回到前一个链
Out-bridge（name）- 流出桥的接口
Out-interface（name）- 数据包离开桥的接口

桥网络地址翻译 Bridge nat

操作路径：/interface bridge nat

本部分描述了在一般防火墙描述中省略了的桥 nat 选项。

属性描述

Action（accept | arp-reply | drop | dst-nat | jump | log | mark | passthrough | redirect | retun | src-nat; default:accept）- 如果数据包匹配了其中一个规则就采取动作：
Accept – 接收包，无动作。例如：数据包通过而没有任何动作，并且没有其他规则会在相关列表/链中处理。
Arp-reply – 发送一个带有指定 MAC 地址的 ARP 应答（任何其他包都会被这条规则忽略，仅在 dstnat 链内有效）
Drop – 悄然丢弃数据包（不发送 ICMP 拒绝信息）
Dst-nat – 改变一个包的目的 MAC 地址（仅在 dstnat 链有效）
Jump – 跳转到由 jump-target 变量指定的链
Log – 记录数据包
Mark – 标记数据包以便后面使用
Passthrough – 忽视这条规则并到下一个。输了对包计数外像一个被禁用的规则一样动作
Redirect – 把数据包重新定位到桥本身（仅在 srcnat 链中有效）

Out-bridge（name）- 流出桥接口
To-arp-reply-mac-address（MAC address）- 当选中 action=arp-reply 时，把源 MAC 地址加入以太网帧及 ARP 有效负载
To-dst-mac-address（MAC address）- 当选中 action=dst-nat 时，把目的 MAC 地址假如以太网帧
To-src-mac-address (MAC address) – 当选中 action=src-nat 时，把源 MAC 地址加入以太网帧

桥路路由

操作路径：/interface bridge broute

这部分描述在一般防火墙描述省略了的桥路设施具体选项，桥路表应用于进入一个转发受控接口的每个包（例如：它不会工作在普通的接口，因为它们没有包含在桥里）。

属性描述

Action（accept | drop | dst-nat | jump | log | mark | passthrough | redirect | return; default: accept）- action to undertake if the packet matches the rule, one of the:
如果数据包匹配了其中一个规则就采取动作：
Accept – 由桥接代码决定对数据包做哪种处理
Drop – 从桥接代码中提取数据包，使它看起来像来自一个非桥接的接口（不会再有其他桥判定或过滤被应用于这个包除非数据包被路由出到一个桥接的接口，这种情况下包将和其他路由包一样被正常处理）
Dst-nat – 改变一个包的目的 MAC 地址（仅在 dstnat 链中有效）
Jump – 跳转到由 jump-target 变量指定的链
Log – 记录数据包
Mark – 标记数据包以便后面使用
Passthrough – 忽视这条规则并到下一个。除了对包计数外像一个被禁用的规则一样动作
Redirect – 把数据包重新定位到桥本身（仅在 dstnat 链中有效）
Return – 从跳转发生的地方回到之前的链
To-dst-mac-address（MAC address）- 当选中 action=dst-nat 时，把目的 MAC 地址加入以太网帧

故障分析

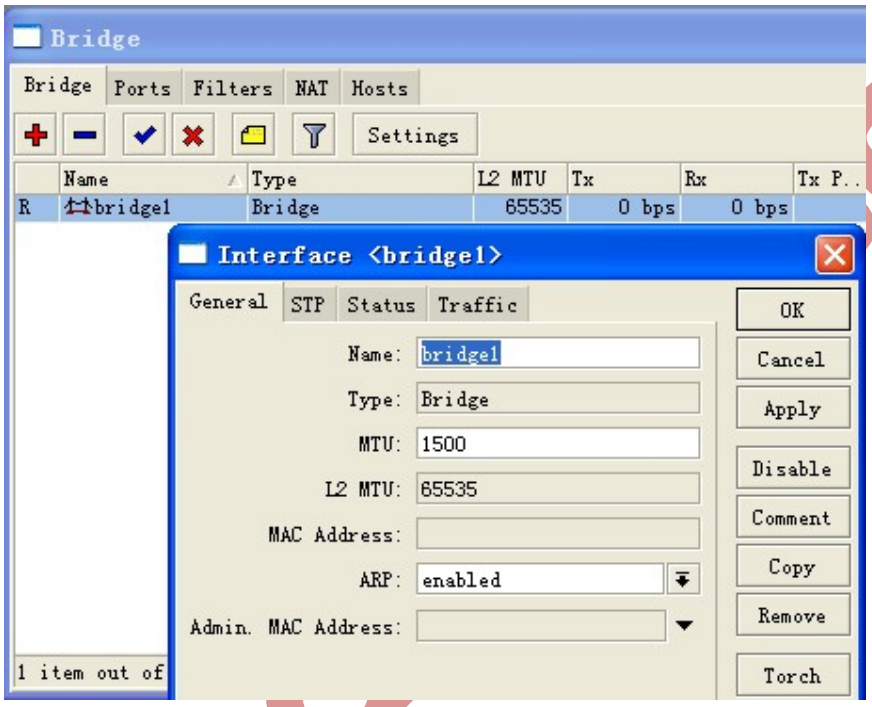
路由器显示我的规则不合法
In-interface, in-bridge（或 in-bridge-port）被指定，但并不存在这样的接口
有一条 action=mark-packet 的动作，但没有 new-packet-mark
有一条 action=mark-connection 的动作，但没有 new-connection-mark
有一条 action=mark-routing 的动作，但没有 new-routing-mark

网桥应用事例

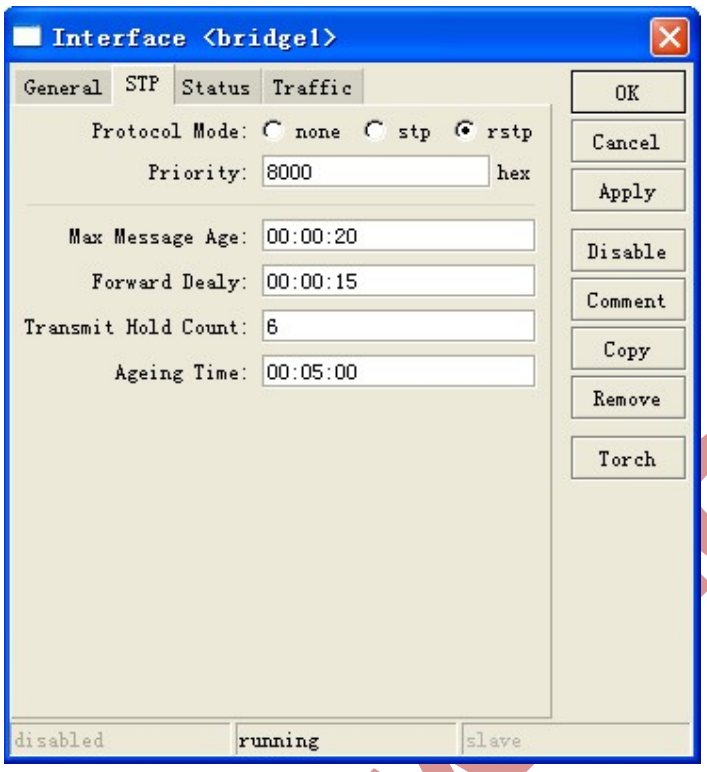
Bridge 实现二层端口隔离

RouterOS 具有 Bridge 的桥接功能，在配置多网口的情况下可以实现二层数据的转发，既可以实现交换机功能，加上 RouterOS 支持 Bridge filter 的过滤，同样也支持对二层数据的管理，通过配置 Bridge 的防火墙规则实现多网口的端口隔离。

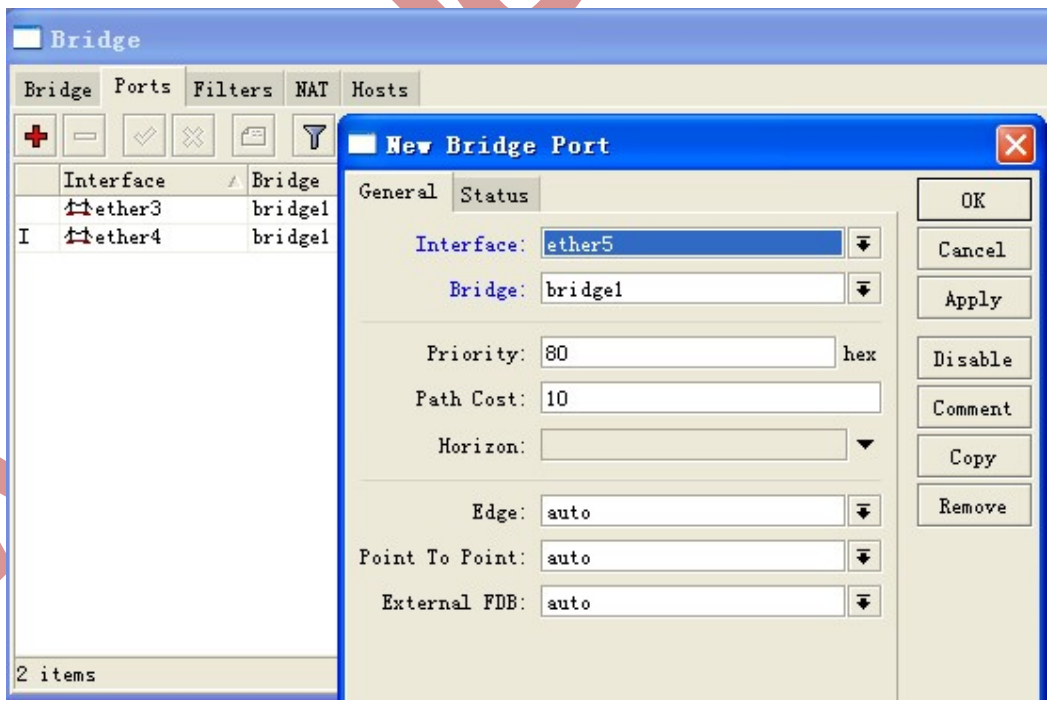
在这里我们通过 RB450 的操作为实例，配置二层端口隔离。首先我们在 Bridge 中添加一个网桥 bridge 1：



在 Bridge 中启用 rstp 快速生成树协议，防止二层的回环出现，同样也是支持二层的冗余功能，在这里我们选择 rstp：



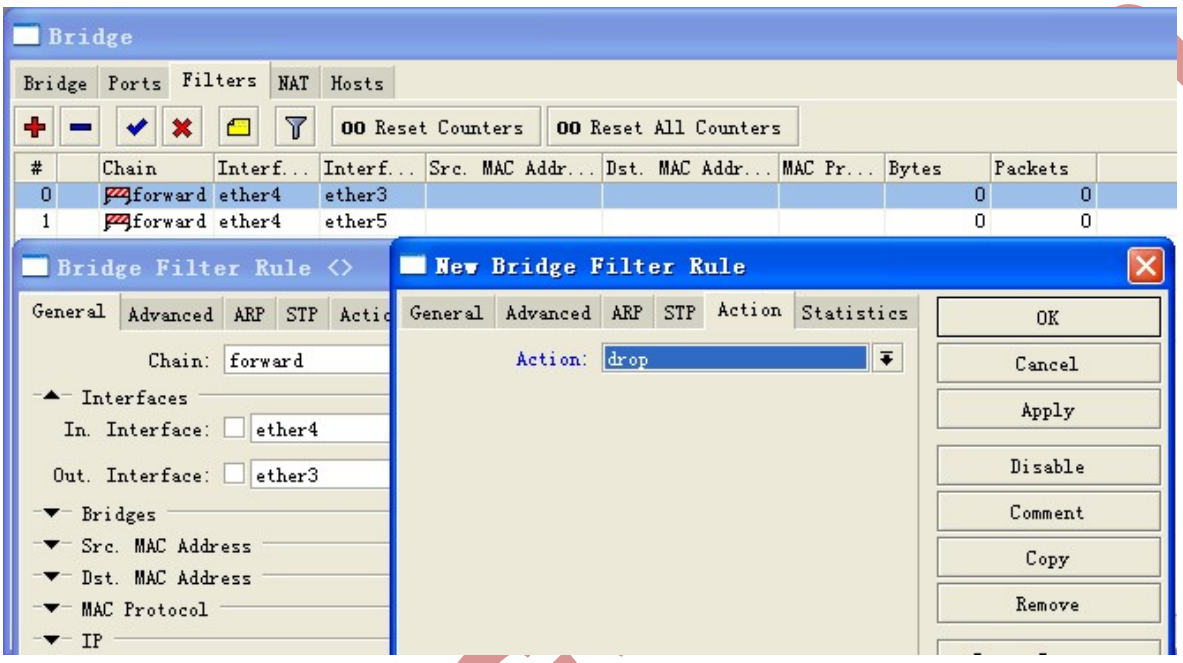
添加完桥接功能后，需要将对应的网卡添加入 Bridge 1 中，进入 Port 中设置，我们将 3 个网卡 ether3、ether4 和 ether5 一个一个添加到 Bridge 1 中：



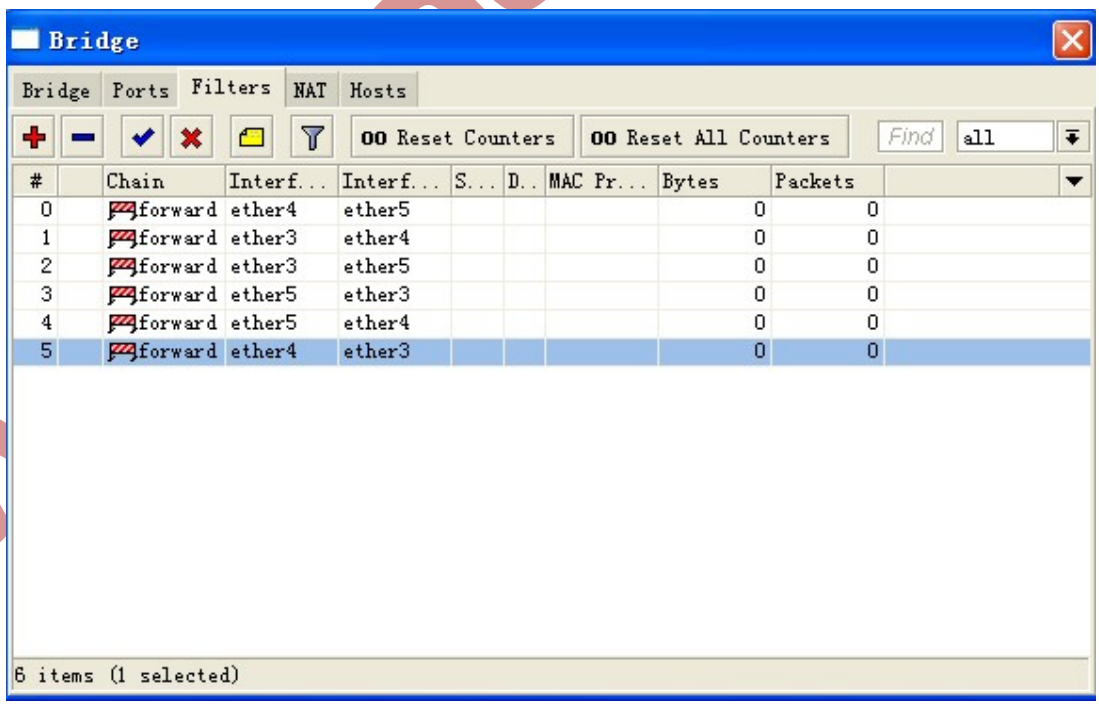
添加完每个端口后，现在 RB450 的 3 个以太网口就完成了桥接的设置，这样 3 个口就实现了二层的交换功能。

[键入文字]

这里我们禁止 ether3、ether4 和 ether5 进行通信，我们进入 filter 中设置防火墙过滤规则，我们首先配置 ether3 与 ether4 的数据隔离我们在 interface 选项中设置 in-interface 和 out-interface(in-interface 为数据进入的网口，out-interface 为数据出去的网口，数据时双向传输的，两个接口需要做两条规则)，然后选择 action 设置 action 参数为 drop，丢弃数据：



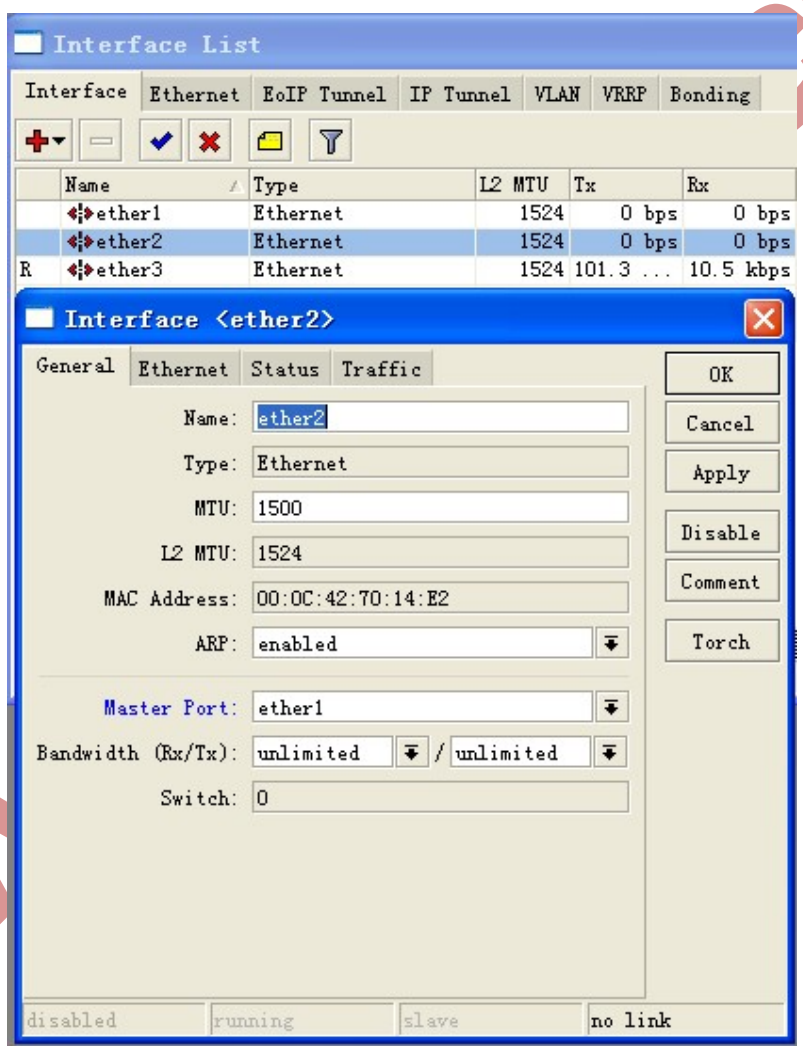
下面是设置好的状态：



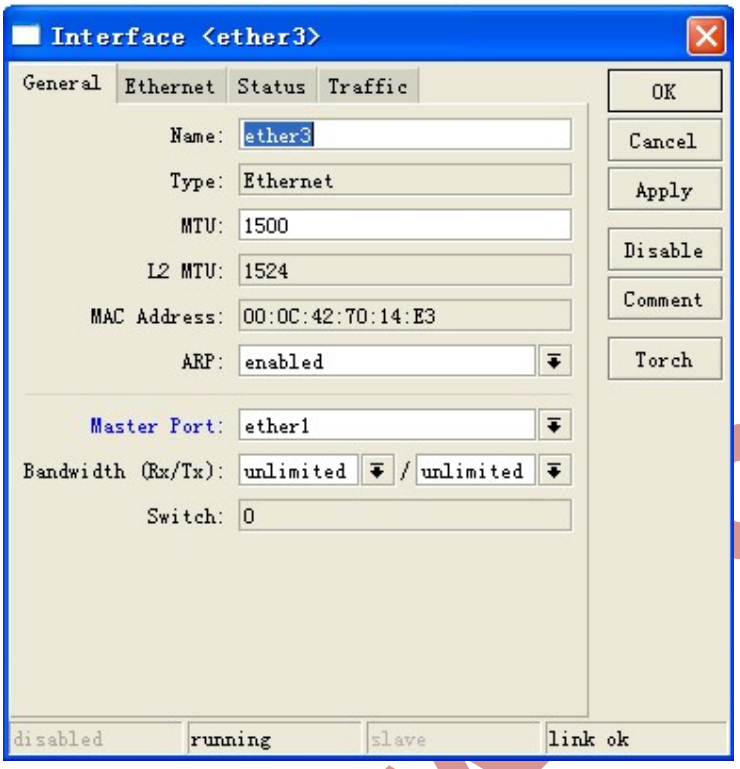
RouterBOARD 设置硬交换 Switch

随着 RouterOS 3.0 发布后，RouterBOARD 系列路由产品开始支持以太网口的硬件交换，如 RouterBOARD450 迷你路由器，5 个以太网口能设置为 5 个硬件交换口，即数据通过二层转发，不在经过 RouterOS 路由软件处理，完全和交换机转发相同。Switch 功能仅支持 RouterBOARD100 和 400 系列产品，需要 3.0 以上的软件版本支持。

下面我们用 RB433 为例，RB433 一共有 3 个以太网口，分别为 ether1、ether2 和 ether3，这里我们需要将三个网卡配置为交换口。设置硬件交换，需要将 1 个网口设置为主端口（Master Port），其他口为从端口（Slave Port），我们已 ether1 为 Master，其他网口为从端口。我们就只需要配置 ether2 和 ether3 的参数，配置 ether2 接口：



配置 ether3 接口



这样 ether1、ether2 和 ether3 设置为 Switch 交换口，三个口可以多到数据的硬件转发，同时可以通过 interface 中的 Bandwidth 设置每个端口的带宽。

如何建立一个透明传输整形器

属性描述

你想用在一个以太网中做一个 Mikrotik RouterOS 透明传输整形器。你可以在两个网络中间加入。要达到这样 RouterOS™应该如下配置（这里假设为没有其他配置在整形器上，并且安装了两张以太网卡）：

1. 启用并命名以太网卡。链接到内部网络的网卡命令为 int，链接到上级路由器的网卡为 ext：

/interface set ether1, ether2 disabled=no
/interface set ether1 name=int
/interface set ether2 name=ext

2. 让我们假设 10.0.0.1 的 IP 地址是网关。那我们添加 IP 地址为 10.0.0.2/24 到相应的网卡上（以后你将需要这个地址远程配置整形器），设置好后你可以通过 ping 来检查你的网关。如果不能通，你可以换一下网线（例如：将插在 ext 网卡上的线换到 int 上，看是否网卡设置反了）注：如果一个都没有工作，可能在网关上设置了防火墙策略或是地址绑定，先暂时删除它们再使一次。

/ip address add interface=ext address=10.0.0.2/24

3. 创建一个桥接口，并将两个物理网卡 int 和 ext 做桥接：

/interface bridge add name=bridge
/interface bridge port add interface=ext bridge=bridge
/interface bridge port add interface=ext bridge=bridge

注：现在前面设置的 IP 地址应被改变到 Bridge 接口上：

/ip address set [/ip address find] interface=bridge

现在你可以简单的添加期望的队列。注：你可以在队列中使用真实的网卡名称。例如，限制所有下载为 256Kbit/s 和所有上传为 128Kbit/s，仅需要添加两条队列就可以了：

/queue simple add limit-at=131072 interface=ext
/queue simple add linit-at=262144 interface=int

第十五章 虚拟路由冗余协议（VRRP）

虚拟路由冗余协议 Virtual Router， Redundancy Protocol（VRRP），MikroTik RouterOS VRRP 协议遵循 RFC2338。VRRP 协议时保证访问一些资源不会中断，即通过多台路由器组成一个网关集合，如果其中一台路由器出现故障，会自动启用另一台。两个或多个路由器建立起一个动态的虚拟集合，每一个路由器都可以参与处理数据，这个集合最大不能超过 255 个虚拟路由器（可参考虚拟路由协议）。一般现在的路由器都支持该协议。

利用 VRRP 集合功能提供高效的路由器运行方式，不在需要复杂的脚本 ping 监测

规格

需要功能包：system
软件等级：Level 1
操作路径： /interface vrrp

许多 VRRP 路由器可用组成一个虚拟路由器集合。在一个网络中最大可用支持相同 VRID（虚拟路由 IP）255 个。每个路由器都必须设置一个优先参数，每个 VRRP 配置通一个虚拟的网卡绑定在一个真实的网卡上。VRRP 地址放入虚拟的 VRRP 网卡上。VRRP Master 状态显示为 running 标志，虚拟网卡上的地址被激活，其他属于 backup（即优先级低的 VRRP 路由）停止运行。

虚拟路由冗余协议时一种为路由提供高效率的路由选择协议。一个或多个 IP 地址可以分配到一个虚拟路由上，一个虚拟路由节点应该具备以下状态：

MASTER 状态，一个节点回答所有的请求给相应请求的 IP 地址。仅只有一个 MASTER 路由器在虚拟路由中。每隔一段时间这个节点发出 VRRP 广播包所有 backup 路由器。
BACKUP 状态，VRRP 路由器监视 Master 路由器的状态。它不会回答任何来至相应 IP 地址的请求，当 MASTER 路由器无法工作时（假设至少三次 VRRP 数据链接丢失），选择过程发生，新的 MASTER 会根据优先级产生。

注：VRRP 不能运行在 VLAN 接口上，VLAN 的接口 MAC 地址于与运行在物理网卡 MAC 地址是不同的。

VRRP 路由

操作路径：/interface vrrp

属性描述

Arp（disabled | enabled | proxy-arp | reply-only；默认：enabled）- 地址解析协议 Address Resolution Protocol

Authentication（none | simple | ah；default：none）- 使用 VRRP 消息数据包的验证方式。
None – 没有证明
Simple – 纯正文验证
Ah – 验证头使用 HMAC-MD5-96 算法
Backup（只读：flag）- 是否为备份状态
Interface（name）- 运行接口的名称
Interval（整型：1..255；默认 t：1）- VRRP 状态更新间隔秒针。定义多少频率发送 VRRP 信息数据包。
Mac-address（MAC address）- VRRP 的 MAC 地址 address。符合 RFC 协议，任何 VRRP 都应该只有唯一的 MAC 地址。
Master（只读：flag）- 是否为 Master 状态
Mtu（整型；默认：1500）- 最大传输单位
Name（name）- VRRP 分配的名称
On-backup（name；默认：“”）- 当节点为 backup 状态执行的脚本
On-master（name；默认：“”）- 当节点为 Master 状态执行的脚本
Password（文本；默认：“”）- -需要验证时的密码，不使用验证时可以被忽略。8 位字符长文本字符串（为纯文本验证方式）；16 位字符长文本字符串（为需要 128 位 key 的 AH 验证）
Preemption-mode（Yes | no；默认：Yes）- 是否启用优先模式。
On – 一个 backup 节点在当前的 Master 失效之前，是不会选择 Master，即使该 backup 的优先高于当前 Master 的级别
Yes – 该节点总是拥有最高优先级。
Vrid（整型：0..255；默认：1）- 虚拟路由的身份号（必须是在接口（interface）上是唯一的）

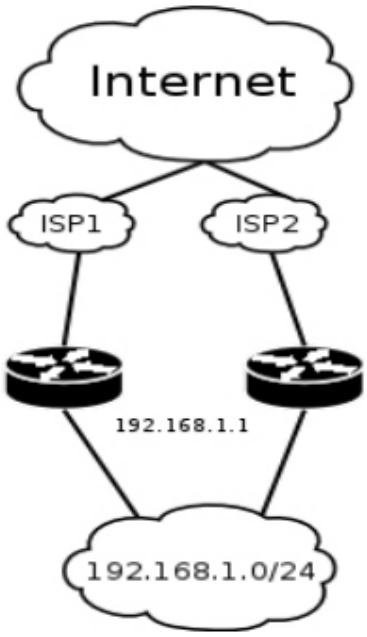
Priority（整型：1-255；默认：100）- 当前节点的优先级（高的数值代表高的优先级）

注：所有同一个集合的节点，必须使相同的 vrid，interval，preemption-mode，authentication 和 password 第 255 的优先级被保留为真正的虚拟路由的主机 IP 地址。

添加一个 VRRP 事例在 ether1 的接口上，一个虚拟路由的 vrid 设置为 1，因为是虚拟路由的主机，所有优先级为 255：

```
[admin@MikroTik] interface vrrp> add interface=ether1 vrid=1 priority=255
[admin@MikroTik] interface vrrp> print
Flags: X - disabled, I - invalid, R - running, M - master, B - backup
 0   RM name="vrrp1" mtu=1500 mac-address=00:00:5E:00:01:01 arp=enabled
       interface=ether1 vrid=1 priority=255 interval=1 preemption-mode=yes
       authentication=none password="" on-backup="" on-master=""
[admin@MikroTik] ip vrrp>
```

简单的 VRRP 事例



VRRP 协议能被用于一个冗余的无缝 Internet 链接，让我们假设有 192.168.1.0/24 网络和我们需要提供高效的 Internet 链接。这个网络需要启用 NAT（VRRP 网络需要使用公网 IP，使用动态路由协议如 BGP 或 OSPF）。我们链接到两个不同的 ISP，且一个被设置为最优先（如，价格便宜或者速度更快的）。

这个事例讲解如何配置 VRRP 在两个路由器上。路由器必须初始化配置：网卡已被启用、每个网卡配置好了 IP 地址、路由表这种正确（至少一个默认路由）。SRC-NAT 或 masquerading（伪装）应配置好。具体设置请参见相关的内容

我们将 192.168.1.0/24 的网络连接到名为 local 网卡的两台 VRRP 路由器上

配置 Master VRRP 路由器

首先我们应创建一个 VRRP 在这个路由器上。我们将使用 255 的优先值，该路由器将被设置为优先路由器

```
[admin@MikroTik] interface vrrp> add interface=local priority=255
[admin@MikroTik] interface vrrp> print
Flags: X - disabled, I - invalid, R - running, M - master, B - backup
 0   RM name="vrrp1" mtu=1500 mac-address=00:00:5E:00:01:01 arp=enabled
       interface=local vrid=1 priority=255 interval=1 preemption-mode=yes
       authentication=none password="" on-backup="" on-master=""
[admin@MikroTik] interface vrrp>
```

下一步，IP 地址应被添加到 VRRP 中

```
[admin@MikroTik] ip address> add address=192.168.1.1/24 interface=vrrp1
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.1/24        10.0.0.0        10.0.0.255      public
  1   192.168.1.2/24     192.168.1.0     192.168.1.255   local
  2   192.168.1.1/24     192.168.1.0     192.168.1.255   vrrp1
[admin@MikroTik] ip address>
```

配置 Backup VRRP 路由器

现在我们将创建一个低优先级的 VRRP 路由（我们可以使用默认值 100），因此路由器将优先选择 backup：

```
[admin@MikroTik] interface vrrp> add interface=local
[admin@MikroTik] ip vrrp> print
Flags: X - disabled, I - invalid, R - running, M - master, B - backup
 0    B name="vrrp1" mtu=1500 mac-address=00:00:5E:00:01:01 arp=enabled
       interface=local vrid=1 priority=100 interval=1 preemption-mode=yes
       authentication=none password="" on-backup="" on-master=""
[admin@MikroTik] interface vrrp>
```

现在我们添加同样的地址到备份 VRRP 路由中：

```
[admin@MikroTik] ip address> add address=192.168.1.1/24 interface=vrrp1
```

现在，当我们断开 Master 路由器，在几秒钟后备份路由将选择 Master 状态：

```
[admin@MikroTik] interface vrrp> print
Flags: X - disabled, I - invalid, R - running, M - master, B - backup
 0   RM name="vrrp1" mtu=1500 mac-address=00:00:5E:00:01:01 arp=enabled
       interface=local vrid=1 priority=100 interval=1 preemption-mode=yes
       authentication=none password="" on-backup="" on-master=""
[admin@MikroTik] interface vrrp>
```

第十六章 Hotspot 热点认证网关

Hotspot 介绍

Hotspot 是一种通过要求用户认证来访问某些网络资源的方法。用户可以使用几乎任何网页浏览器（HTTP 或 HTTPS 协议）登陆，所以他们不需要安装任何附件的插件。RouterOS 会自动计算正常运行时间以及每个客户使用的流量，并且也能把这个信息发送到 RADIUS 服务器。Hotspot 系统可以限制每个特定用户的比特率，总流量，运行时间以及涉及的其他参数。

Hotspot 热点 web 服务认证时一种友好的 web 方式的认证系统，在此种认证方式中，系统将自动要求未认证用户打开认证网页，验证通过后，便可链接到因特网，未认证用户无论输入任何一个网站地址，都会被强制到一个认证界面，要求用户进行认证。配置了 Walled Garden 特性后，允许用户不需要提前认证就可以访问一些网页。

获取地址

首先，一个客户必须先获得一个 IP 地址。它可以通过设置被静态 IP 地址，或者获取一个 DHCP 服务器的所分配的 IP 地址。如果可以的话，DHCP 服务器可以换提供绑定分发 IP 地址到客户 MAC 地址的途径。

此外，Hotspot 服务器能自动分配给任何客户端的虚拟 IP 地址，分配来自 Hotspot 建立的 IP 池。这个特性对那些不愿意修改 IP（活不清楚，缺乏网络技术）。如果用户和 Hotspot 网关不在相同子网，Hotspot 通过 ARP 广播的方式强迫分配一个 IP 给用户，用户不会注意到这个转变（例如：在用户配置不会有任何改变），但路由器本身则看到完全不同（在 Hotspothost 中可以看到被转化了的地址），这项技术叫做一对一 NAT，但它也以 RouterOS2.8 版本中叫做的“即插即用”。

一对一 NAT 接收来自己连接网络接口的任何向内地址，并完成一个网络地址翻译。客户可以使用任何预先配置的地址（注意要求配置网关）。如果一对一 NAT 特性被设置为翻译一个客户的地址为一个公网 IP 地址，那么这个客户就甚至可以运行一个服务器或任何其他需要公网 IP 地址的服务。这个 NAT 将在数据包被路由器接收后就立即改变包的源地址。

注意，你使用一对一 NAT 时，必须在该接口上启用 arp 模式。

认证之前

当在一个接口上启用 HotSpot 时，系统自动配置对所有未登陆用户显示登录页面。这个是通过添加动态目的 NAT 规则完成的，你可以在一个运行中的 Hotspot 系统上观察得到。这些规则是用来把未认证用户的所有 HTTP 及 HTTPS 请求重定向到 Hotspot servlet（认证过程，都会跳转到登录页面）。其他一些规则将在该章后面专门部分进行讲述。

在配置好 Hotspot 后，打开任何 HTTP 页面都会产生 Hotspot servlet 登陆页面（可以通过自行定义登陆页面），所有访问 Hotspot 网关以外的资源，都会跳转到登陆页面，因此必须在 Hotspot 网关配置一个合法的 DNS

Walled Garden

有时希望对某些服务不要求认证（例如让客户不需要认证的访问公司的服务器），或者一些服务要求认证（例如，用户访问一个内部文件服务器或其他限制区域）。这些都可以通过 Walled Garden 系统实现。

当一个未登陆用户请求 Walled Garden 中允许的服务时，Hotspot 网关不会阻拦它，或者如果是 HTTP，就简单地把请求重定向到原来的目的（或定向到一个指定的父级代理）。

为了执行 Walled Garden 对 HTTP 请求的特性，专门设计了一个嵌入的 web 代理服务器，所有来自未认证用户的请求时从这个代理通过。注意嵌入的代理服务器还没有高速缓存功能。还要注意这个嵌入代理服务器是在 system 软件功能包里并不需要 web-proxy 功能包。它是在 /ip proxy 下面配置的。

认证

现在有 5 中不同的认证方法。你可以同时使用一个或多个：

HTTP PAP – 最简单的方法。显示 Hotspot 登陆页并以纯文本格式获取认证信息（如：用户名和密码）。注意当在网络传输时，密码是没有加密的。

HTTP CHAP – 标准方式，在登陆页包含了 CHAP 询问。CHAP MD5 散列询问与用户密码一起使用来计算将被发送到 Hotspot 网关的字符串。散列结果（作为一个密码）与用户名一起通过网络发送到 Hotspot 服务器（所以，密码是从来不以纯文本格式通过 IP 网络发送的）。在客户端，MD5 算法通过 JavaScript applet 执行，所以如果一个浏览器不支持 JavaScript（比如，Internet explorer 2.0 或一些 PDA 浏览器），将不能认证用户。可以允许未加密密码，即开打 HTTP PAP 认证方式被接受，但并不推荐使用这个特性（出于安全考虑）。

HTTPS – 与 HTTP PAP 一样，但对加密传输使用了 SSL 协议。Hotspot 用户只发送没有附加散列的密码（注意没有必要担心纯文本密码在网络上的暴露，因为传输本身是加密的）。在另一种情况，HTTP POST 方法（如果不可能，那么用 HTTP GET 方法）用于向 Hotspot 网关发哦说那个数据。

HTTP cookie – 在每次成功登陆之后，会有一个 cookie 发送到 web 浏览器，同时被添加到活动 HTTP cookie 列表。这个 cookie 将与存储在 Hotspot 网关的相比较，并仅当源 MAC 地址及随机生成的 ID 与存储在网关的相匹配。这个方法只可以与 HTTP PAP，HTTP CHAP 或 HTTPS 方法一起使用，不然的话没有其他方式可以产生 cookie。

MAC address – 将用客户端的 MAC 地址与用户账户同时作为用户名。

Hotspot 可以通过询问本地用户数据库或 RADIUS 服务器认证用户(本地数据库会被先询问，然后是 RADIUS 服务器)。如果通过 RADIUS 服务器认证 HTTP cookie，那么路由器将在 cookie 被第一次产生时发送相同的信息到服务器。如果认证在本地完成，那么符合该用户的信息将会被调用，否则将会调用 RADIUS 中的参数。如果要知道更多关于 RADIUS 服务器

工作的信息，请参见其相应的 Radius 手册。

HTTP PAP 方法也使得通过请求页 /login?username=username&password=password。如果你想使用 telnet 连接登陆，准确的 HTTP 请求应该这样：GET /login?username=username&password=password HTTP / 1.0

配置菜单

/ip hotspot – Hotspot 上的特定界面（每个界面一个服务器）。Hotspot 服务器必须添加在这个目录中，Hotspot 系统才能够在一个界面上工作。

/ip hotspot profile – Hotspot 服务器概要。影响 Hotspot 客户登陆过程的设置在这里进行。多个 Hotspot 服务器可以使用同样的概要信息。

/ip hotspot host – 所有 Hotspot 接口上的活动网络主机的动态列表。在这里你可以找到 IP 地址与一对一 NAT 的绑定

/ip hotspot ip-binding – 将 IP 地址绑定到主机 Hotspot 接口的规则

/ip hotspot service-port – 一对一 NAT 地址翻译助手

/ip hotspot walled-garden – HTTP 等级的 Walled Garden 规则（DNS 名，HTTP 请求字串）

/ip hotspot walled-garden ip – IP 等级的 Walled Garden 规则（IP 地址，IP 协议）

/ip hotspot user – 本地 Hotspot 系统用户

/ip hotspot user profile – 本地 Hotspot 系统用户组规则

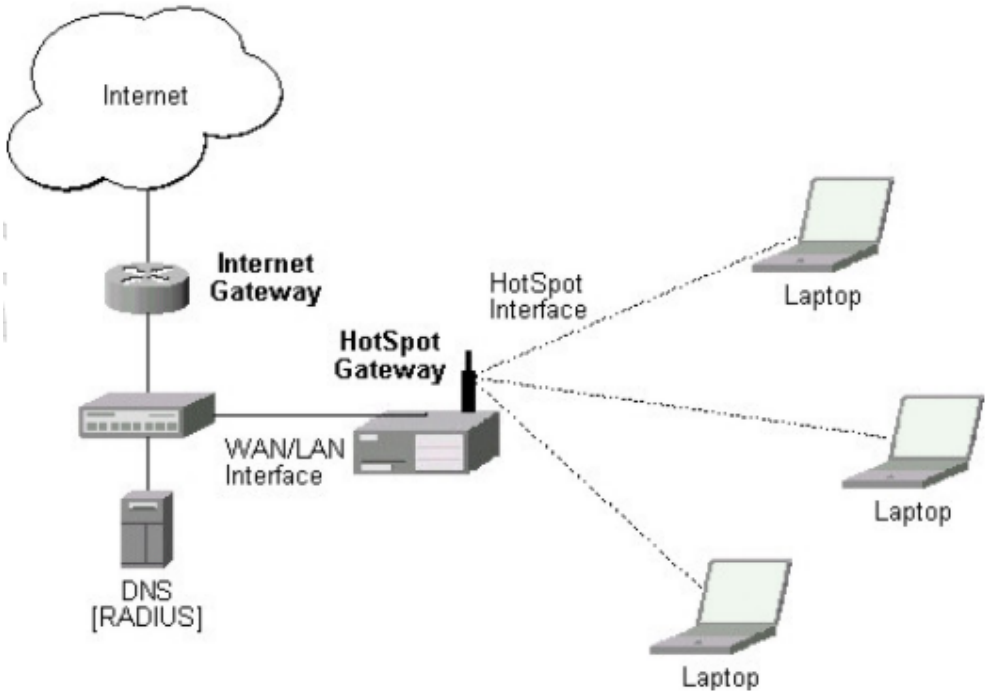
/ip hotspot active – 所有已认证 Hotspot 用户的动态列表

/ip hotspot cookie – 所有合法的 HTTP cookie 动态列表

下面是一个简单的 Hotspot 事例，Hotspot 网关应该至少有两个网络接口：

1. Hotspot 接口，用于连接 Hotspot 客户
2. LAN/WAM 接口，用于访问网络资源。例如：DNS 和 RADIUS 服务器应该可达

下面的图表显示了一个简单的 Hotspot 设置。



Hotspot 接口应该分配一个 IP 地址。物理网络连接应该建立在 Hotspot 用户的电脑和网关之间。它可以使无线（无线网卡需要在 AP 上注册），或者有线的（MIC 网卡需要连接到一个集线器或一个交换机）。

当 ISP 需要在有线或者无线网络中建立 Hotspot 热点认证系统，如：小区、酒店、机场和其他公共场所。一个普通的 Hotspot 网络建立在一个外网接口和一个内部网络接口下，我们需要对内网用户作认证上网。

注：在 2.9 版本的 RouterOS Hotspot 功能包采用的是端口代理的方式连接，在启用 Hotspot 接口后 UpNp 即插即用功能自动启动，通过在/ip hotspot host 列表中可以查询相应的信息。

Hotspot 接口设置

操作路径：/ip hotspot

Hotspot 系统建立在一个独立的网络接口，你可以在不同的网络接口（以太网卡、无线网卡等）上配置不同的 Hotspot 服务器。

属性描述

Address-per-mac（整型 | unlimited；默认：2）- 允许与特定 MAC 地址绑定的 IP 地址数量（降低一个 IP 模拟多个 MAC 的攻击）
Unlimited – 每个 MAC 对应 IP 地址数量无限制
Address-pool（name | none；默认：none）- 运行一对一 NAT 的 IP 地址。你可以选择不使用

一对一 NAT
None – 对这个 Hotspot 接口的客户不使用一对一 NAT
HTTPS（只读：flag）- HTTPS 服务是否在这个接口上实际在运行（它在这个服务器概要中设置，并且在路由器中输入了一个合法的认证）
Idle-timeout（time | none；默认：00:05:00）- 对未认证客户的空闲超时时间（非活动的最大时间）。它用于探测客户没有使用外部网络（因特网），例如，没有收到来自某个客户的流量也没有流出路由器的流量。达到超时时间后，用户将被主机注销清除，用户所使用的地址也将被释放
None – 不切断空闲用户
Interface（name）- 运行 Hotspot 的接口
Ip-of-dns-name（只读：IP address）- Hotspot 接口概要中设置的 Hotspot 网关 DNS 名称的 IP 地址
Keepalive-timeout（time | none；默认：none）- 对未认证客户的持活超时时间。用于探测客户的计算机是活动的并且是可达的。如果在这个期间探测失败，那么用户将被主机列表清除并且用户使用的地址也将被释放
None – 不切断不可达用户
Profile（name；默认：default）- 接口的默认 Hotspot 概要
Reset-html (name) – 以原始的 HTML 文件重新覆盖已有的 Hotspot servlet。它用于你改变 servlet 之后且它不工作。

注：addresses-per-mac – 只有当地址池定义后，属性才能生效。

为了把 Hotspot 系统添加到本地接口，允许系统对每个客户进行一对一 NAT（来自 HS-real 地址池的地址将被用于 NAT）：

```
[admin@MikroTik] ip hotspot> add interface=local address-pool=HS-real
[admin@MikroTik] ip hotspot> print
Flags: X - disabled, I - invalid, S - HTTPS
 #   NAME                    INTERFACE    ADDRESS-POOL PROFILE IDLE-TIMEOUT
 0   hs-local                local       HS-real      default 00:05:00
[admin@MikroTik] ip hotspot>
```

Hotspot 服务

操作路径：/ip hotspot profile

属性描述

Dns-name（text）- Hotspot 服务器的 DNS 名称。与 Hotspot 服务器名类似的 DNS 名。（它看起来像登陆页面位置）。这个名字会被自动地在 DNS 缓存中添加为一个静态 DNS。
Hotspot-address（IP address；default：0.0.0.0）- Hotspot 服务器的 IP 地址
Html-directory（text；default：“”）- 目录的名称（以 FTP 访问），它存储了 HTML servlet 页面（当改变路径时，如果路径不存在，默认页面会自动被复制到指定的目录中）

http-cookie-lifetime（time；default：3d）- HTTP cookie 的有效时间

http-proxy（IP address；default：0.0.0.0）- Hotspot 服务器将作为一个代理服务器使用的对所有被通用代理系统打断并没在/ip proxy direct 列表中定义的代理服务器地址。如果没有特别指明，地址将在/ip proxy 下面的

parent-proxy 参数定义。如果这个也空缺，请求将被本地代理处理。

Login-by（multiple choice：cookie | http-chap | http-pap | https | mac | trial；default：cookie，http-chap）- 使用的认证方法

Cookie – 使用 HTTPcookie 认证，而不询问用户证明。以防客户没有 cookie，或者存储的用户名和密码对从上一次认证后不再合法，就将使用其他方法认证。可能仅和其他 HTTP 认证方法一同使用（HTTP-PAP，HTTP-CHAP 或 HTTPS），因为第一次 cookie 是没有办法产生的。

http-chap – 对密码使用 MD5 散列算法的 CHAP 询问-回答的方法。这种方法很容易避免在一个不安全网络上发送清楚的文本密码。这个方法是默认的认证方法。

http-pap – 在网络中使用纯文本认证。请注意如果使用了这个方法，你的用户密码将在本地网络中暴露，所有可够侦听它们。

https – 使用加密了的 SSL 通道来传输用户与 Hotspot 服务器的通信。注意，为了使它能工作，必须对路由器输入一个合法的认证（参见认证管理的手册）。

mac – 试着先使用客户的 MAC 地址作为它的用户名。如果与本地用户数据库或 RADIUS 服务器匹配了，那么客户将不会被要求填写登陆表格就可以通过认证。

trial – 在一定时间内不会要求认证。

Radius-interim-update（time | received；default：received）- 发送累计账户报告的频率

Os – 与 received 相同

Received – 使用接收自 RADIUS 服务器的任何值

Rate-limit（text；default：“”）- 从路由器角度考虑以 rx-rate[/tx-rate] [rx-burst-rate[/tx-burst-rate] [rx-burst-threshold[/tx-burst-threshold][rx-burst-time[/tx-burst-time]]]格式表示的速率限制（其中“tx”是客户上传，“tx”是客户下载）。所有的速率都应该是带有‘k’（1,000s）或‘M’（1,000,000s）的数字。如果 tx-rate 没有指定，rx-rate 和 tx-rate 一样。对于 tx-burst-rate 和 tx-burst-threshold 以及 tx-burst-time 也同理。如果 rx-burst-threshold 和 tx-burst-threshold 都没有指定（但是 burst-rate 已指定），rx-rate 和 tx-rate 将被做为 burst threshold 使用。如果 tx-burst-time 和 tx-burst-time 都没有指定，那么 1s 将会作为默认值使用。

Smtp-server（IP address；default：0.0.0.0）- 默认 SMTP 服务器无条件地用于重定向

Split-user-domain（yes | on；default：no）- 当用户名以“user@domain”或“domain\user”格式给出时，是否把用户名从域名中分离出来

Ssl-certificate（name | none；default：none）- 对 HTTPS 认证使用的 SSL 认证名。不用于其他认证方法

Trial-uptime（time/time；default：30m/1d）- 仅当认证方式为询问时使用。

Tridl-user-profile（name；default：default）- 仅当认证方法为询问时使用。指定询问用户将使用的用户概要

Use-radius（yes | on；default：on）- 是否使用 RADIUS 认证 Hotspot 用户

注：如果 dns-name 属性没有指定，则 Hotspot-address 将代替使用。如果 Hotspot-address 也没有指定，那么将自动探测这两个值。如果启用了 RADIUS 验证，/radius 下的参数应正确配置。

属性描述

Domain（只读：text）- 域名（如果从用户名中分离出来的话）
Expires-in（只读：time）- cookie 合法存在的时间
Mac-address（只读：MAC address）- 用户的 MAC 地址
User（只读：name）- 用户名

注：如果 dns-name 属性没有指定，则 Hotspot-address 将代替使用。如果 Hotspot-address 也没有指定，那么将自动探测这两个值。如果启用了 RADIUS 验证，/radius 下的参数应正确配置。

属性描述

Domain（只读：text）- 域名（如果从用户名中分离出来的话
Expires-in（只读：time）- cookie 合法存在的时间
Mac-address（只读：MAC address）- 用户的 MAC 地址
User（只读：name）- 用户名

注：可以在相同的 MAC 地址上有多重的 cookie。例如，在同一台电脑上对每个 web 浏览器都可以有一个单独的 cookie。

Cookie 是可以过期的。默认的 cookie 合法时间为 3 天（72 小时），但对每个 Hotspot 服务是可以修改的，例如：
/ip hotspot profile set default http-cookie-lifetime=1d

获取合法 cookie 列表：

```
[admin@MikroTik] ip hotspot cookie> print
  # USER            DOMAIN          MAC-ADDRESS       EXPIRES-IN
  0 ex                            01:23:45:67:89:AB 23h54m16s
[admin@MikroTik] ip hotspot cookie>
```

HTTP 方式 Walled Garden

操作路径：/ip hotspot walled-garden

Walled Garden 是在允许未认证下访问某些资源，同样能用于需要认证访问的其他资源。例如：访问一些 Hotspot 服务提供商的基本信息或账单选项。

这个目录只管理对 HTTP 和 HTTPS 协议的 Walled Garden。其他协议也可以包含进 Walled Garden，但要在其他地方配置（/ip hotspot walled-garden ip，参考本手册的下一部分）。

属性描述

Action（allow | deny；default：allow）- 如果数据包和规则匹配则执行动作：
Allow – 无需优先认证就允许访问页面
Deny – 需要认证才能访问页面
Dst-address（IP address）- 目的 web 服务器的 IP 地址
Dst-host（wildcard；default：“”）- 目的 web 服务器的域名（这是一个通配符）
Dst-port（整型；default：“”）- 客户发送请求的目的 TCP 端口
Method（text）- 请求的 HTTP 方法
Path（text；default：“”）- 请求的路径（这是一个通配符
Server（name）- 应用该规则的 Hotspot 服务器名
Src-address（IP address）- 发送请求的用户 IP 地址

注：通配符属性（dst-host 和 dst-path）匹配一个完整的串（如：若设置为“example”，则它们不会匹配“example.com”）。可用的通配符为‘*’（匹配任意字符的任意数量）并且‘？’（匹配任何一个字符）。正则表达式也在这里接受，但如果属性做为一个正则表达式对待，那么它应该以图表（‘:’）开始。

关于使用正则表达式： ：
\\符号序列是用于在控制台输入\字符的
\. 样式的意思为只是 .(在正则表达式单独的点表示任何符号)
显示在给出样式之前任何符号都不允许，我们在样式开始使用^符号
指定在给出样式之后任何符号都不允许，我们在样式结束的地方使用符号$

由于路由器不能解密请求，你也就不能对 HTTPS 请求使用 path 属性（也不应该使用——这就是 HTTPS 协议被创造的目的）。

允许未认证用户到 www.example.com 域/paynow.html 页面的请求：

```
[admin@MikroTik] ip hotspot walled-garden> add path="/paynow.html" \
\... dst-host="www.example.com"
[admin@MikroTik] ip hotspot walled-garden> print
Flags: X - disabled, D - dynamic
 0   dst-host="www.example.com" path="/paynow.html" action=allow
[admin@MikroTik] ip hotspot walled-garden>
```

IP 方式 Walled Garden

操作路径：/ip hotspot walled-garden ip

这个目录管理类属 IP 请求的 Walled Garden。参见前面 HTTP 和 HTTPS 协议属性的部分（像实际的 DNS 名，HTTP 方法和在请求中使用的路径）。

属性描述

Action（allow | deny；default：allow）- 如果数据包和规则匹配则执行动作：
Allow – 无需认证就允许访问页面
Deny – 需要认证才能访问页面
Reject – 需要认证才能访问该页面，以防页面会被没有认证的 ICMP 拒接信息访问，主机不可达将被产生
Dst-address（IP address）- 目的 web 服务器的 IP 地址
Dst-host（wildcard；default：“”）- 目的 web 服务器的域名（这是一个通配符）
Dst-port（整型；default：“”）- 客户发送请求的目的 TCP 端口
Protocol（整型 | ddp egp encap ggp gre hmp icmp idpr-cmtp igmp ipip ipsec-ah ipsec-esp iso-tp4 ospf pup rdp rspf st tcp udp vmtp xns-idp xtp）- IP 协议名
Server（name）- 应用该规则的 Hotspot 服务器名
Src-address（IP address）- 发送请求的用户 IP 地址

IP 绑定

操作路径：/ip hotspot ip-binding

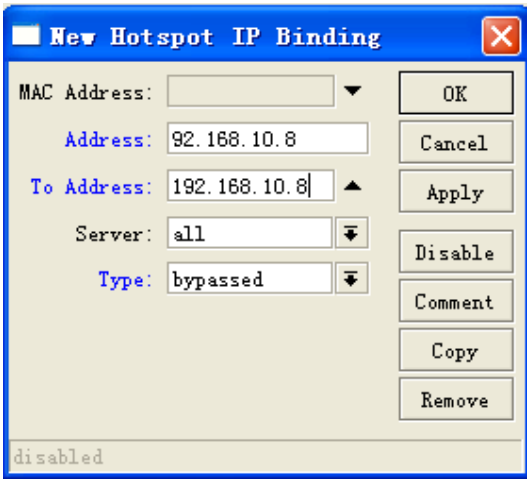
你可以静态地设置源 IP 地址（或 IP 网络）或源 MAC 地址的 NAT 翻译。你也可以允许一些地址绕过 Hotspot 认证（如：它们可以不必认证登陆就能访问外部资源），并阻止指定的地址认证登陆。

属性描述

Address（IP address / [netmask]；default：“”）- 源 IP 地址或客户网络
Mac-address（MAC address；default：“”）- 客户的源 MAC 地址
Server（name | all；default：all）- 客户将连接到的服务器名
To-address（IP address；default：“”）- 把源始客户与地址翻译成的 IP 地址。如果 address 属性是作为一个网络给定，那么这个将是翻译的开始地址（例如：第一个 address 被翻译为 to-address，address+1 翻译为 to-address+1，以此类推）
Type（regular | bypassed | blocked）- 静态绑定条目类型
Regular – 根据条目中设定的值进行一对一 NAT 翻译
Bupassed – 绕过认证，即不需要通过 Hotspot 认证，扩音访问资源
Blocked – 不会执行翻译，并且所有来自主机的数据包被丢弃

注：这是一个有序列表，所以你可以把更详细的条目放在里表的顶部以超越比较低的普通条目。

例如，让 192.168.10.8 不通过认证，即可上网，并访问外部资源：



Hotspot 主机列表

操作路径：/ip hotspot host

这个目录显示了所有链接到 Hotspot 服务下的活动主机，这个列表包含所有一对一 NAT 翻译。

属性描述

Address（只读：IP address）- 客户的原始 IP 地址
Authorized（只读：flag）- 客户是否成功地被 Hotspot 系统认证
Blocked（只读：flag）- 如果访问在 Walled-Garden 中因为广告超时时间过期被阻止，则为真
Bridge-port（只读：name）- 主机链接的真实物理接口。当 Hotspot 服务被放在一个桥接口以判定在桥中的主机实际的端口时，使用该值
Bypass-hotspot（只读：flag）- 是否客户不需要 Hotspot 系统的认证
Bytes-in（只读：整型）- 路由器从客户接受的字节数
Bytes-out（只读：整型）- 路由器发送到客户的字节数
Host-dead-time（只读：time）- 路由器没有从主机接收任何数据包（包括 ARP 回应，持活回应及用户流量）的时间。
Idle-time（只读：time）闲置的时间
Idle-timeout（只读：time）- 应用于用户的确切 idle-timeout 值。这个属性显示了用户空闲多久会被自动登出。
Keepalive-timeout（只读：time）应用于用户的 keepalive-timeout 精确值。这个属性显示了用户的电脑在不可达状态多久会被自动登出
Mac-address（只读：MAC address）- 实际的用户 MAC 地址
Packets-in（只读：整型）- 路由器接收客户的包数
Packets-out（只读：整型）- 路由器发送到客户的数据包数
Server（只读：name）- 主机链接到的服务器名
Static（只读：flag）- 翻译是否是来自静态 IP 绑定列表

To-address（只读：IP address）- 主机翻译成的原始 IP 地址
Uptime（只读：time）- 用户的当前会话时间（如：用户在活动用户列表中已经多久了？）

命令描述

Make-binding – 把可以个动态项目从这个列表复制到静态 IP 绑定列表
Unnamed（name）- 项目编号
Comment（text）- 产生客户对静态条目的评论
Type（regular | bypassed | blocked）- 静态项目的类型

Hotspot 用户管理

主要对 Hotspot 的用户账户、权限和用户参数分组进行管理。

操作路径： /ip hotspot user
操作路径： /ip hotspot user profile

热点用户管理用于普通用户分类设置，profile 用户组根据需要能将不同用户分类管理。

属性描述

Address-pool（name | none；默认值：none）- 用户用来分配 IP 的 IP 池名称。这个就像 MikroTik RouterOS 早期版本的 dhcp-pool 一样工作。
None – 不向这个服务中的用户再分配 IP 地址
Advertise（yes | no；默认值：no）- 是否对此服务启用强制广告弹出
Advertise-interval（multiple choice：time；默认值：30m，10m）- 显示广告弹出之间间隔的设置。在列表完成后，最后一项值会后面所有广告使用
Advertise-timeout（time | immediately never；默认值：1m）- 在使用 Walled Garden 阻止网络访问之前等待广告显示的时间长度
Advertise-url （multiple choice：text；默认值：http://www.mikrotik.com/ ，http://www.routerboard.com/）- 广告弹出显示的 URL 列表。这个列表时循环的，所以当到达最后一项时，下次显示的将是第一项
Idle-timeout（time | none；默认值：none）- 授权用户空闲超时时间（未活动状态的最长时间）。它用于探测用户没有使用向外部网络或 Hotspot 主机发送数据（如因特网），比如：没有任何流量从用户进入或从路由器流出。当达到超时时间，用户会被登出，丢出主机列表，用户使用的地址也会被清空，记录的会话时间也会由这个值减少。
None – 不切断空闲用户
Incoming-filter（name）- 应用于来自此服务用户向内数据包的防火墙链表的名称
Incoming-packet-mark（name）- 自动置于来自此服务每个用户所有数据包的包标记
Keepalive-timeout（time | none；默认值：00:02:00）- 授权客户的持续活超时时间。用于探测客户的电脑是在线。如果在这个期间检测失败，那么用户会被注销，用户使用的地址也会被清空。
None – 关闭此功能，不切断不在线用户
Name（name）- 服务参考名

On-login（text；默认值：“”）- 用户登入后运行的脚本后
On-logout（text；默认值：“”）- 用户登出后运行的脚本名
Open-status-page（always | http-login；默认值：always）- 是否为授权用户显示状态页面使用 MAC 登入方法。如果你想放一些信息（例如：横幅或弹出窗口）在 alogin.html 页面将会很有用，这样所有的用户都可以看到它。
http-login – 如果 http 登入打开状态页面（包括 cookie 和 http 登入方法）
always – 如果 mac 登入打开 http 状态页面
outgoing-filter（name）- 应用于此服务用户的向外流出的包的防火墙链表名称
outgoing-packet-mark（name）- 自动设置在此概要每个用户的所有数据包的包标记
rate-limit（text；默认值：“”）- 从路由器角度来看的 rx-rate[/tx-rate]格式的速率限制。[rx-burst-rate[/tx-burst-rate][rx-burst-threshold[/tx-burst-threshold]]][rx-burst-time[/tx-burst-time][priority][rx-rate-min[/tx-rate-min]]]（所以“rx”客户的上传，“tx”客户的下载）。所有速率必须以可选的‘k’（1,000s）或‘M’（1,000,000s）计算。如果 tx-rate 没有指定，则 rx-rate 和 tx-rate 一样。对于 tx-burst-rate 和 tx-burst-threshold 以及 tx-burst-time 也同理。如果 both rx-burst-threshold 和 tx-burst-threshold 都没有指定（但 burst-rate 指定了），那么 rx-burst 和 tx-rate 会作为脉冲串门限使用。如果 rx-burst-time 和 tx-burst-time 都没有指定，那么 1s 将设置为默认值。优先级从 1 到 8 取值，1 代表最高优先级，而 8 代表最低的。如果 rx-rate-min tx-rate-min 都没有指定那么 rx-rate 和 tx-rate 的值将被使用。Rx-rate-min 和 tx-rate-min 的值不能超过 rx-rate 和 tx-rate。
Session-timeout（time；默认值：os）- session timeout（maximal allowed session time）for client. After this time, the user will be logged out unconditionally 把客户会话切断（最大允许的会话时间）。在这个时间过后，用户将会被无条件地登出。
0 – 不切断
Shared-users（整型；默认值：1）- 同时登陆切实用同一个用户名的最大用户数量
Shatus-autorefresh（time | none；默认值：none）- 热点 servlet 状态页面自动刷新间隔
Transparent-proxy（yes | no；默认值：yes）- 是否对该概要授权用户使用透明的 HTTP 代理

注：当 idle-timeout 或者 session-timeout 到时，对该用户的连接会话将会被从 Hotspot 认证中注销，减少用户闲置对系统的超载。

操作路径：/ip hotspot user

属性描述

Address（IP address；默认值：0.0.0.0）-静态 IP 地址。如果不是 0.0.0.0，那么客户将总是得到相同的 IP 地址。也就是说，对该用户只允许一个同时的登陆。任何一个已存在的地址都将使用嵌入的一对一 NAT 被这个地址取代。
Bytes-in（只读：整型）- 接收用户的总字节数
Bytes-out（只读：整型）- 发送给用户的总字节数
Limit-bytes-in（整型；默认值：0）- 用户可以传输的最大字节数（例如：从接收到的字节数）
0 – 无限制
Limit-bytes-out（整型；默认值：0）- 用户可以接收的最大字节数（例如：发送给用户的字节数）

0 – 无限制

Limit-uptime（time；默认值：0s）- 用户的总正常运行时间限制

0s – 无限制

Mac-address（MAC address；默认值：00:00:00:00:00:00）- 静态 MAC 地址。如果不是 00:00:00:00:00:00，那么用户仅能从该 MAC 地址登陆

Name（name）- 用户名

Packets-in（只读：整型）- 接收到用户的最大包数量

Packets-out（只读：整型）- 发送给用户的最大包数

Password（text）- 用户口令

Profile（name；默认值：默认值）- 用户资料

Router（text）- 当用户连接上后将在热点网关注册的路由器。路由格式为“dst-address 网关公制”（例如：“10.1.0.0/24 10.0.0.1”）。数个路由应用逗号分开指定。

Server（name | all；默认值：all）- 该用户允许登陆的服务器

Uptime（只读：time）- 用户登陆的总时间

注：如果 MAC 认证方法使用，客户的 MAC 地址可以被当作用户名使用（不需要口令）

字节限制是对每个用户的总限制（不像在/ip hotspot active 中的对每个会话的限制）。所以，如果一个用户已经下载了些东西，那么会话限制将显示总限制 -（minus）已下载的。例如：如果对一个用户的下载限制为 100MB，并且用户已经下载了 30MB，那么在/ip hotspot active 中的登陆后会话下载限制将为 100MB-30MB=70MB。

如果一个用户达到了他的限制是对每个用户的总限制（bytes-in >=limit-bytes-in 或 bytes-out >=limit-bytes-out），它将再也不能登陆。如果用户通过本地用户数据库认证，那么每次他登出时统计就会被更新。意思是说如果一个用户现在登陆，那么统计现在也不会显示当前总的值。使用/ip hotspot active 子目录以查看当前用户会话的统计。

如果用户的 IP 地址被指定了，则仅允许一个同时的登陆。如果同一个认证的用户为激活时被再次使用，那么活动用户将自动被登出。

添加一个仅允许 01:23:45:67:89:AB MAC 地址登陆的用户名和密码都为 ex 的用户，并限制 1 小时工作时间：

```
[admin@MikroTik] ip hotspot user> add name=ex password=ex \
\... mac-address=01:23:45:67:89:AB limit-uptime=1h
[admin@MikroTik] ip hotspot user> print
Flags: X - disabled
 #   SERVER    NAME                  ADDRESS      PROFILE   UPTIME
 0             ex                                 default   00:00:00
[admin@MikroTik] ip hotspot user> print detail
Flags: X - disabled
  0    name="ex" password="ex" mac-address=01:23:45:67:89:AB profile=default
    limit-uptime=01:00:00 uptime=00:00:00 bytes-in=0 bytes-out=0
    packets-in=0 packets-out=0
[admin@MikroTik] ip hotspot user>
```

Hotspot 在线用户

操作路径：/ip hotspot active

现时用户列表显示当前已登陆了的用户。这里不能修改任何信息，除了使用 remove 命令将用户登出。

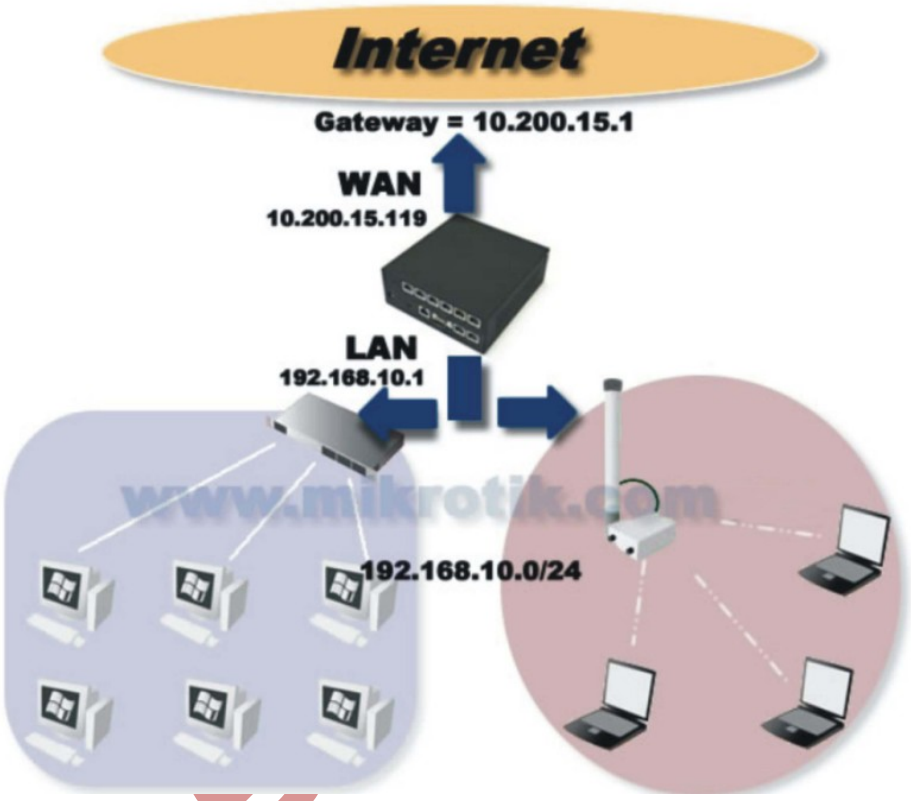
属性描述

Address（只读：IP address）- 用户的 IP 地址
Blocked（只读：flag）- 是否以广告将用户阻挡（例如：通常适当的广告未决）。
Bytes-in（只读：整型）- 路由器从客户收到的字节数
Bytes-out（只读：整型）- 路由器发送到客户的字节数
Domain（只读：text）- 用户范围（如果从用户名中分离出来）
Idle-time（只读：time）- 用户被闲置的时间
Idle-timeout（只读：time）- 应用于该用户的 idle-timeout 精确值。这个属性显示他被自动登出的闲置时间
Keepalive-timeout（只读：time）- 应用于该用户的 keepalive-timeout 精确值。该属性描述了用户的电脑不可达多久才会被自动登出
Limit-bytes-in（只读：整型）- 用户被允许发送给路由器的最大字节数
Limit-bytes-out（只读：整型）- 路由器被允许发送到客户的最大字节数
Login-by（multiple choice，只读：cookie | http-chap | https | mac trial）- 用户用的认证方法
Mac-address（只读：MAC address）- 用户的实际 MAC 地址
Packets-in（只读：整型）- 路由器接受来自客户的包数量
Packets-out（只读：整型）- 路由器发送给客户的包数量
Radius（只读：yes | no）- 用户是否通过 RADIUS 认证
Server（只读：name）- 用户登陆所指定的服务器
Session-time-left（只读：time）- 应用于该用户的 session-time-left 精确值。这个属性显示了用户在自动被注销前保持的登入状态时间
Uptime（只读：time）- 当前用户的会话时间（例如：用户登入的时间）

User（只读：name）- 用户名

Hotspot 配置事例

我们根据下面的网络拓扑结构为例：

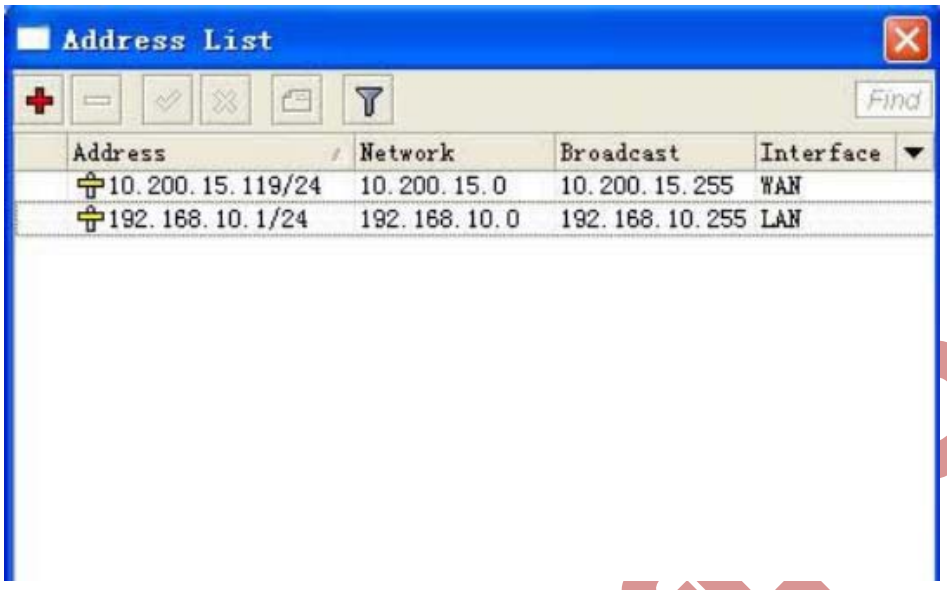


一个网关路由器的网络参数如下：

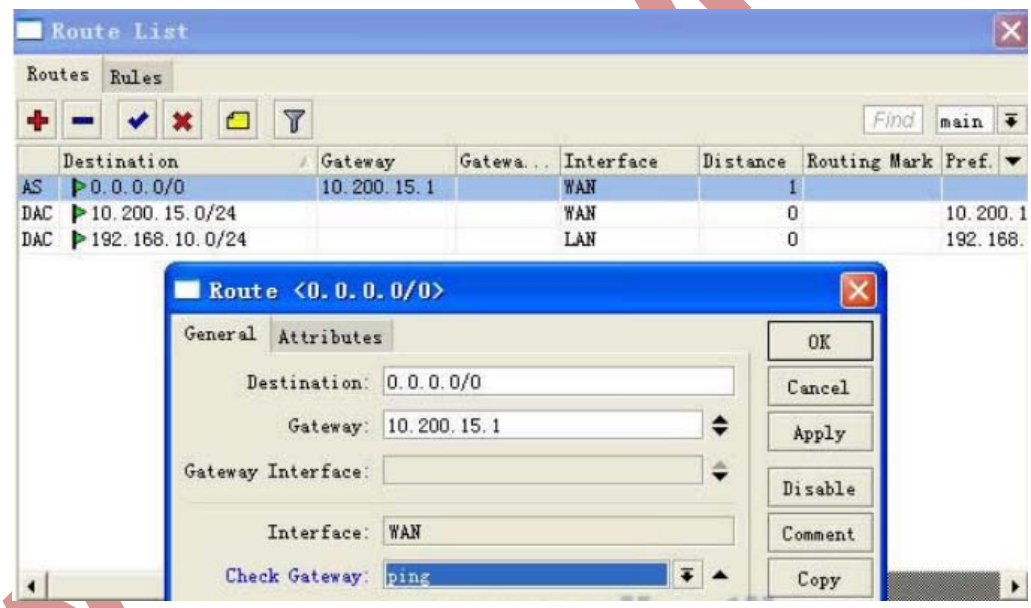
WAN 口对应外网 IP 为 10.200.15.119/24，网关为 10.200.15.1
LAN 口对应内网 IP 为 192.168.10.1/24
DNS：61.139.2.69

在根据这些参数我们需要先配置好 IP 地址、网关和 DNS，并打开 DNS 缓存等。

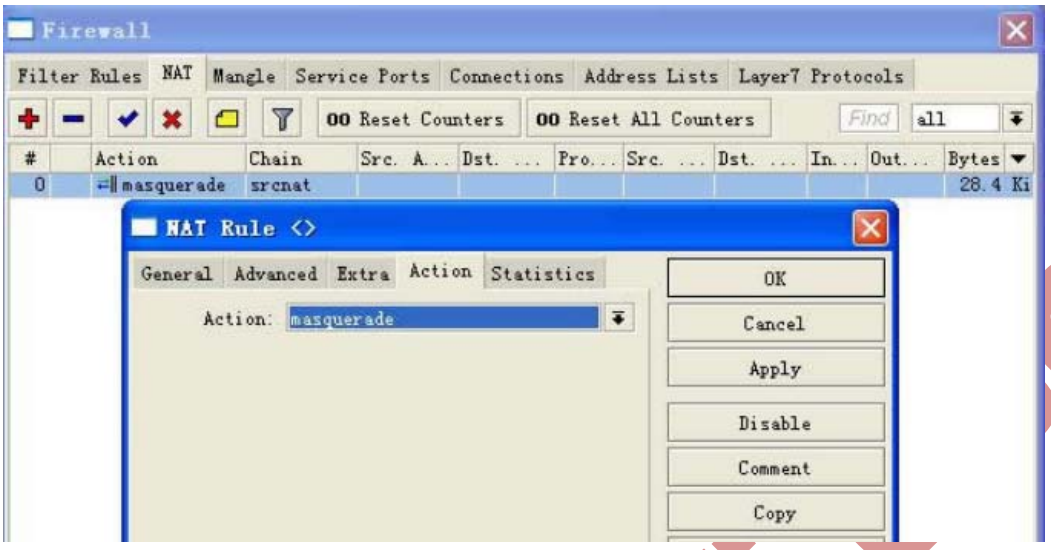
进入 ip address 配置 IP 地址：



进入 ip route 配置网关



进入 ip firewall nat 设置好 NAT 伪装：

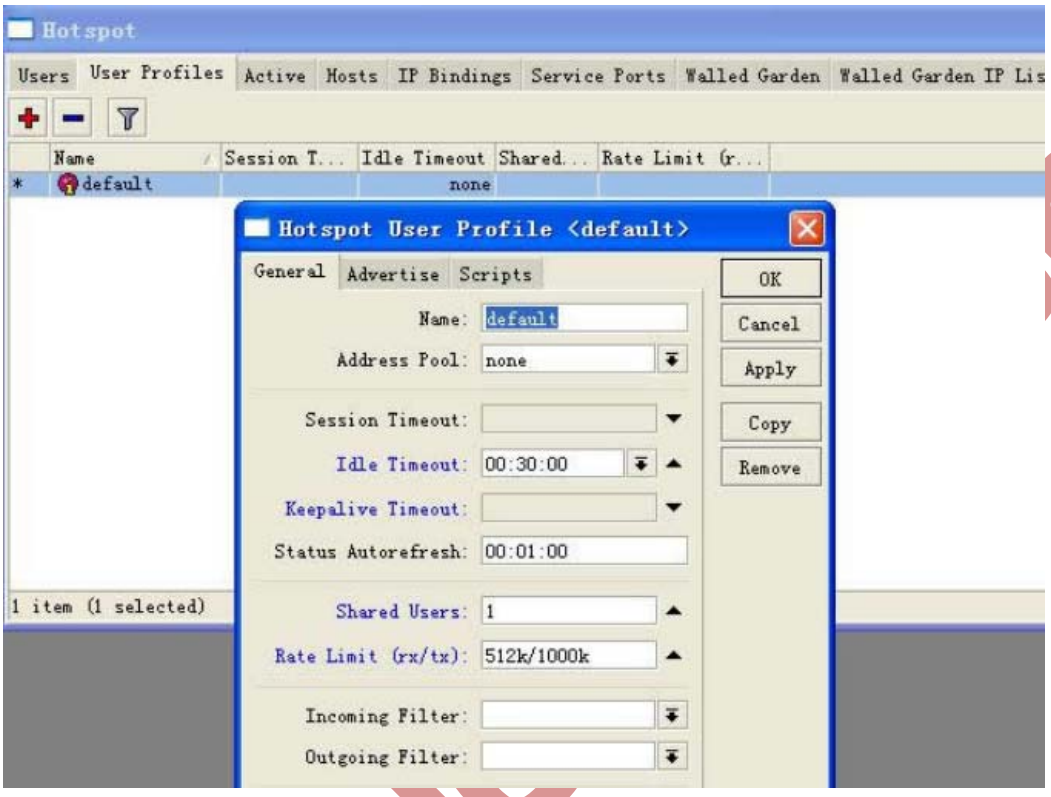


进入 ip dns 配置 DNS 缓存：

现在我们的基本参数已经配置完成，现在我们需要配置的 Hotspot 参数，配置 Hotspot 参数的基本流程是：

1、 先进入 ip hotspot user profile 设置用户分组规则
2、 然后再 ip hotspot user 添加用户的账号
3、 进入 ip hotspot server profile 配置服务器规则
4、 在 ip pool 中分配 IP 地址段，根据需要启用 DHCP 服务
5、 在 ip hotspot server 添加并启用 Hotspot 服务

现在我们进入 ip hotspot，并配置 ip hotspot use profile



在 user profile 里面一般配置如下几个参数：

Idle-timeout：用户在一定时间内没有任何流量发出后自动注销链接

Keepalive-timeout：路由器主动通过 ICMP 探测主机是否在线，如果在一定时间为探测到自动注销链接（如果用户机开启防火墙，路由器无法探测到）

Shared-users：账号的分享用户多少，默认为 1，即仅一个用户使用该账号。

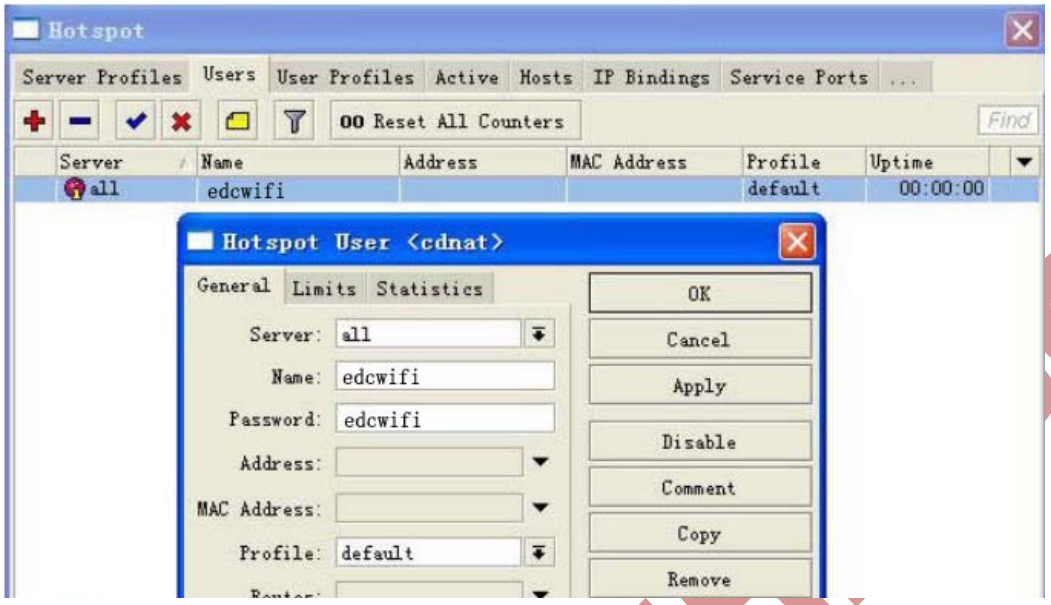
Rate-limit：分配每个账号宽带，格式为“上行/下行”

Transparent-proxy：透明代理功能是否开启，一般使用 Hotspot 认证建议不用打开此参数。

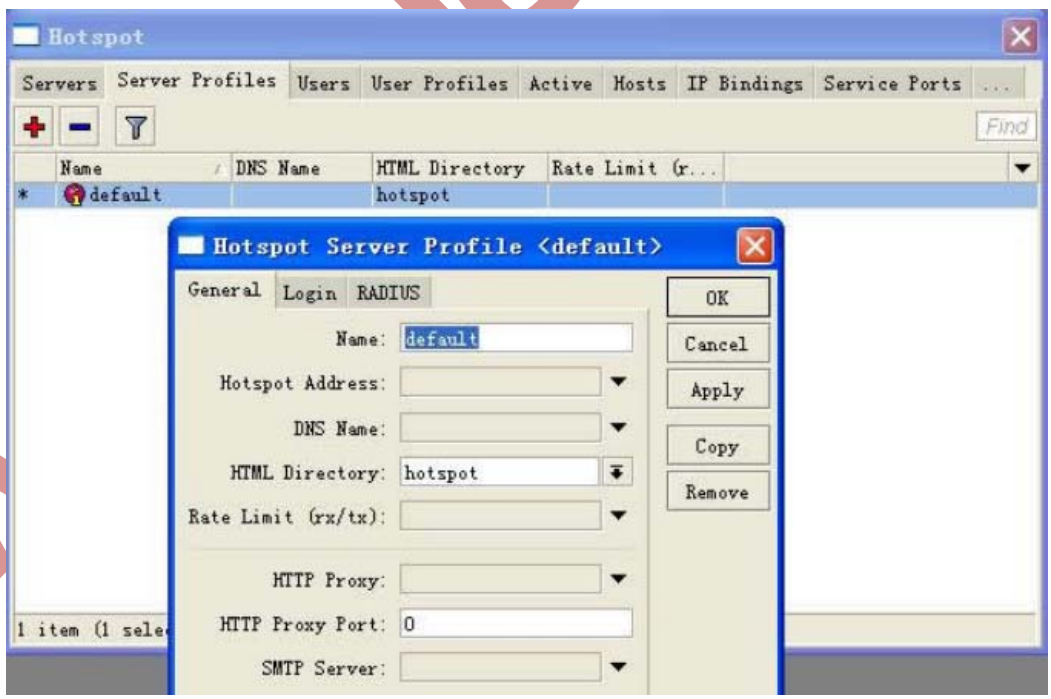
其他参数请参考具体 Hotspot 手册。

Address pool 这个是 DHCP 的地址池，给用户分配 IP，我们可以在 ip pool 中分配地址段，具体操作请参考 RouterOS 的 DHCP 操作。

在 user 配置用户登录账号和密码，以及所属的 profile 类型：

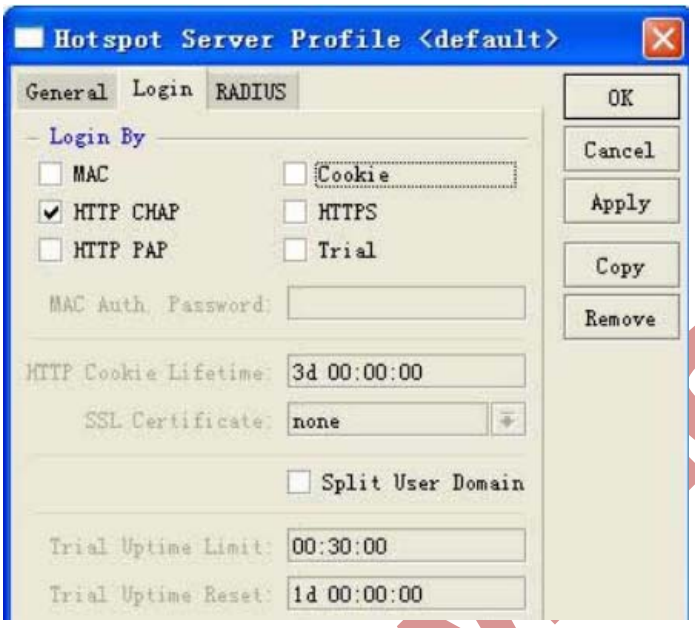


这里默认 server 服务器 all，
Name 用户名：edcwifi
Password：edcwifi
Profile：用户组规则，这里选择我们之前设置的 default 规则
配置完用户规则后，进入 ip hotspot server profile，配置服务器规则。



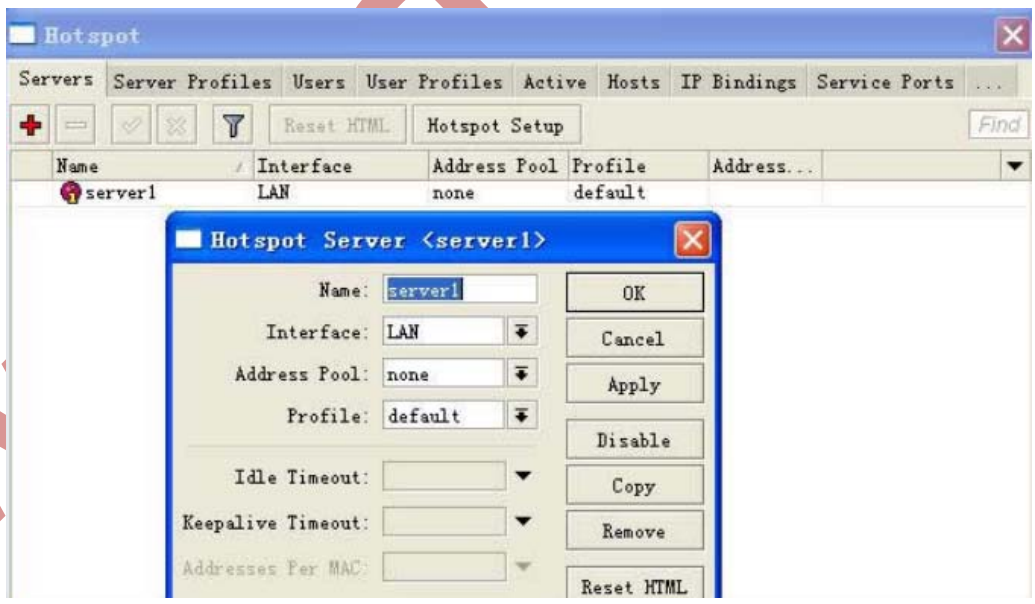
在 general 选项中选择 HTML directory 为默认的 Hotspot 文件路径，同时也可以选择自己定义的文件名路径。

配置 login 登录方式，一般只启用 http chap 即可，其他选项根据需要开启。



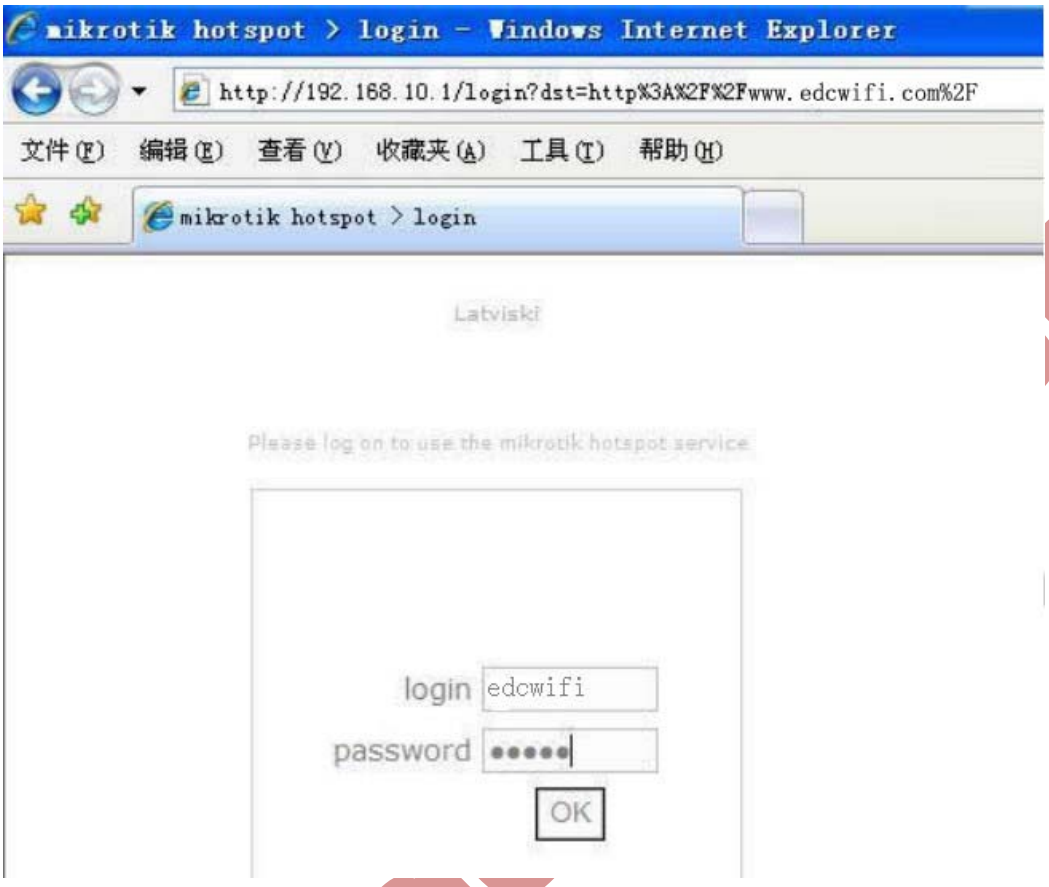
至于 radius 根据需要开启。

配置完成以上参数后，最后我们开启 Hotspot 服务器：



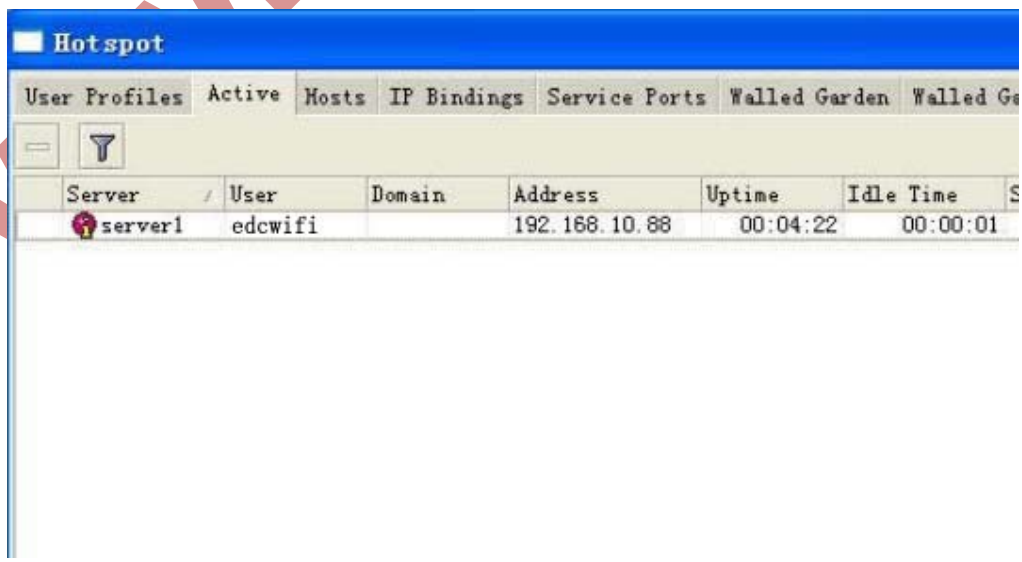
当开启完成后，所有对路由器或者外网访问都需要通过 web 认证，在用户没有认证的情况下，当用户随便输入一个网站都会跳转到认证页面。

如当输入 www.edcwifi.com 的网站，Hotspot 会强制用户的 web 页面跳转到认证页，如图：



用户输入账号 edcwifi 和密码 edcwifi 后，点 ok 按钮即可通过认证，当认证通过后，页面自动跳转到 www.edcwifi.com 的网站.

这时我们可以在 ip hotspot active 中看到用户登录的在线情况：



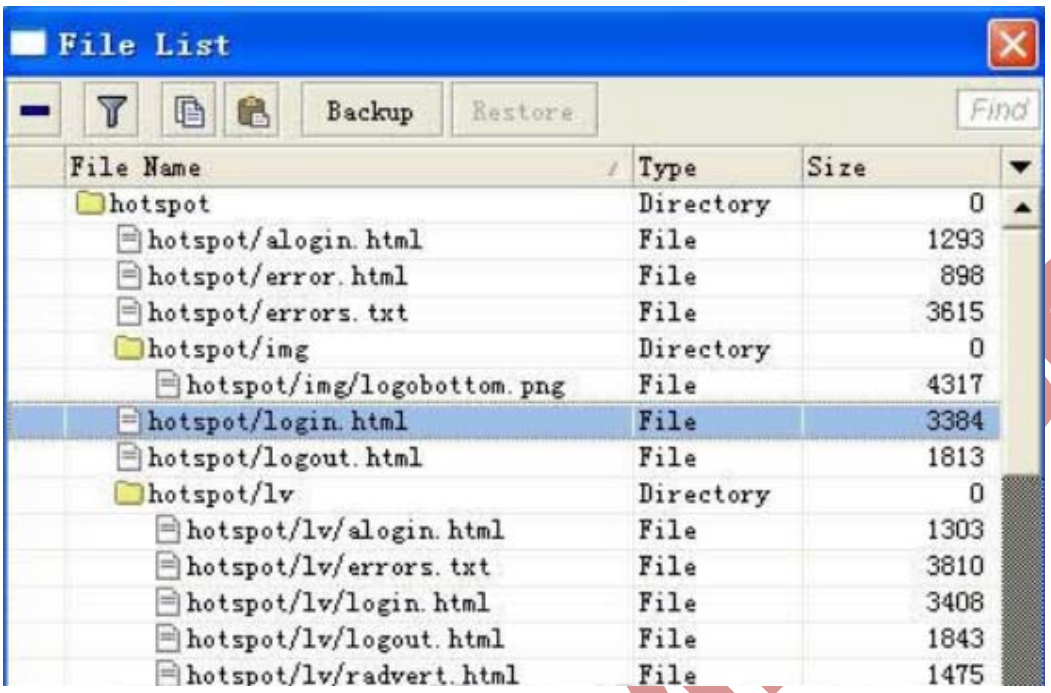
获取现时用户列表：

```
[admin@MikroTik] ip hotspot active> print
Flags: R - radius, B - blocked
 #    USER           ADDRESS         UPTIME      SESSION-TIMEOUT IDLE-TIMEOUT
 0    EDCWIFO      192.168.10.88      4m17s       55m43s
[admin@MikroTik] ip hotspot active>
```

用户如果需要注销，通过输入 192.168.10.1 Hotspot 认证网关，点击 log off 推出登录页面

认证页面

Hotspot 的认证登录页面时开放式的，既可以通过 RouterOS 的 files 目录下找到这些文件，Hotspot 在 files 中的默认文件名“Hotspot”，



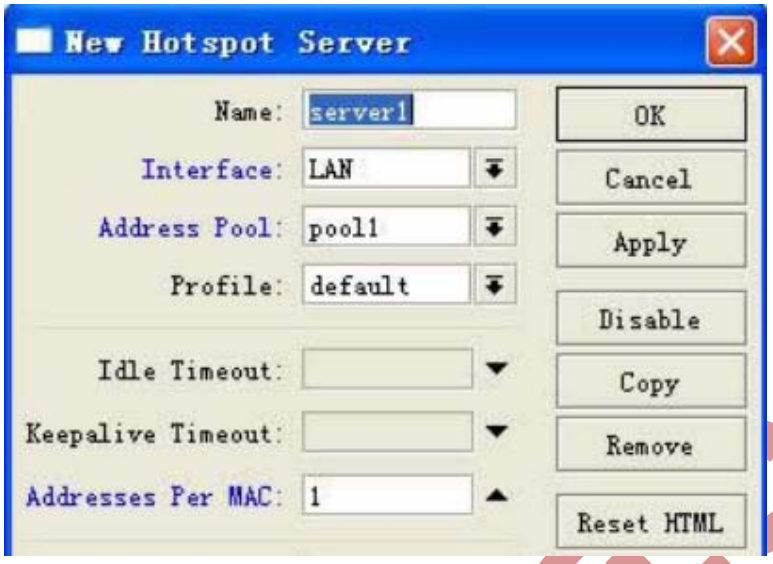
认证页面我们可以通过修改 login.html、logout.html 和 status.html 的 web 界面得到你想要的网页画面或者 log。

Hotspot 即插即用功能

从 2.7 的版本就开始支持 upnp 的即插即用功能，即当用户和 Hotspot 认证服务器在同一局域网内，不管局域网用户设置任何的 IP 地址（前提是用户必须设置任意的 IP 地址、网关和 DNS）都可以被 Hotspot 认证服务器获取，并在 Hotspot 的 host 中分配一个新的虚拟 IP 地址，并对用户作一对一的 NAT 转换。Hotspot 的即插即用方式分成适用于：流动性较强的公共场所，如机场、车站、公园，也可以应用到酒店和小区中。

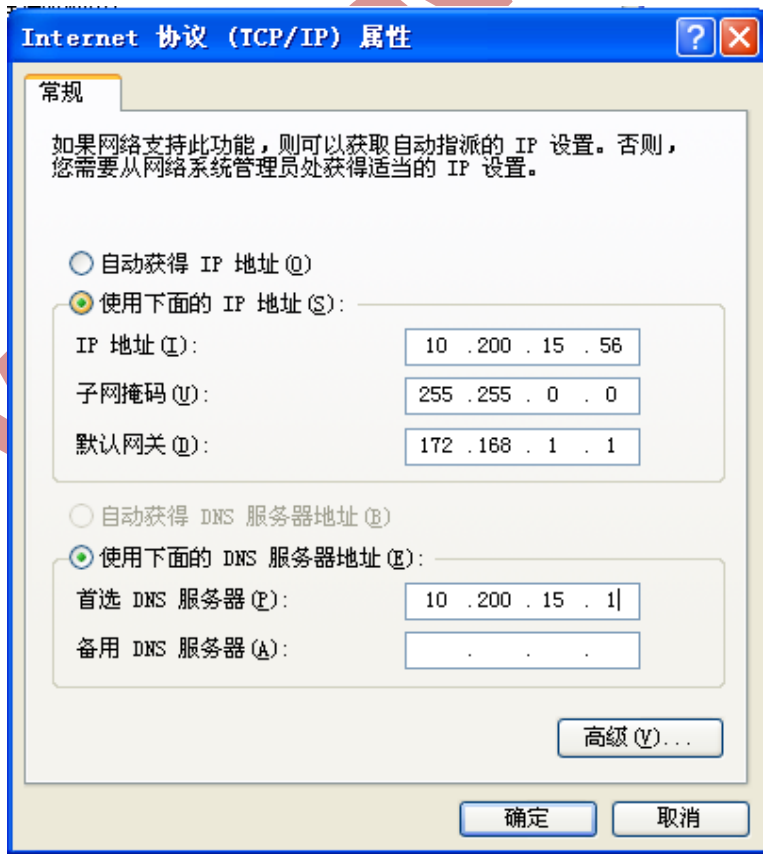
Hotspot 服务器会在同一局域网内发送 ARP 广播，告诉局域网内的所有主机自己的网关设备，并为在线的主机分配一个虚拟的 IP 地址，这样客户主机在没有配置正确的 IP 地址情况下页能连接到 Hotspot 网关服务器，并认证上网。

在 2.9 和 3.0 的 Hotspot 启用 server 服务后，即插即用功能默认是打开的，但配置 Hotspot 需要在 Hotspot server 中将 address pool 的地址池设置好，如图：

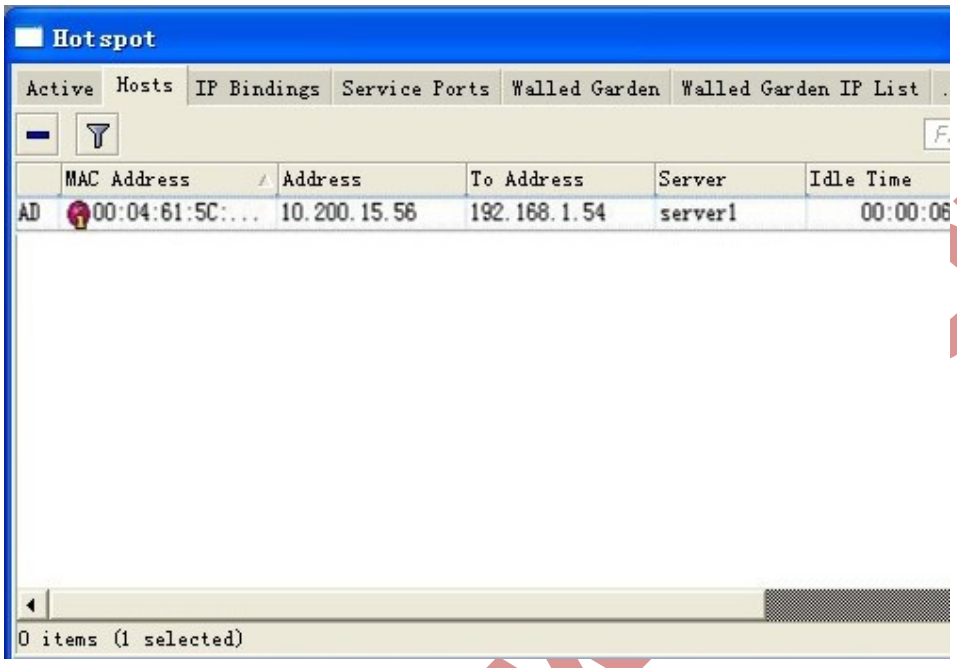


Address Per MAC 这个是每个 IP 对应的 MAC 地址，这里我们设置为 1，即一个 IP 对应一个 MAC 地址。

我们 windows 电脑的 IP 地址配置如下：



在 Hotspot 的 host 列表中，我们可以看到，在同一局域网内的 windows 主机被 Hotspot 捕获后，自动为其分配 IP 地址，并做了对应关系



注：如果 Hotspot 没有工作，可能的情况如下：

检查/ip dns 包含的合法 DNS 服务器，在命令或者 tools ping 中是否能解析/ping www.edcwifi.com，并确定 DNS 的缓存功能打开
确保连接追踪已经启用：/ip firewall connection tracking set enabled=yes

Hotspot 防火墙部分

除了在/ip hotspot 子目录本身的明显的动态规则（像主机及动态用户），一些附加的规则会在激活一个 Hotspot 服务时被添加到防火墙表中。不像 RouterOS 2.8 版本，只有相对较少的防火墙规则添加在防火墙中，因为主要的工作时有一对一 nat 算法完成的。

Nat 规则

从/ip firewall nat print dynamic 命令你可以获取如下（在每条规则后跟有评注）：

```
 0  D Chain=dstnat hotspot=from-client action=jump jump-target=hotspot
```

把对数据包的所有 Hotspot 相关任务从 Hotspot 客户放到一个单独的链中：

```
1  D Chain=Hotspot Protocol=udp dst-port=53 action=redirect to-ports=64872
2  D Chain=Hotspot Protocol=tcp dst-port=53 action=redirect to-ports=64872
```

把所有 DNS 请求都重定向到 Hotspot 服务。64872 端口对所有 Hotspot 用户提供 DNS 服务。如果你想要 Hotspot 服务器也监听其他端口，在这里以同样方式添加规则，改变 dst-port 属性。

```
3   D Chain=Hotspot Protocol=dst-port=80 Hotspot=local-dst action=redirect
    to-ports=64872
```

把所有 HTTP 登陆请求定向到 HTTP 登陆 servlet。64873 就是 Hotspot HTTP servlet 端口。

```
4   D Chain=Hotspot Protocol=tcp dst-port=443 hotspot=local-dst action=redirect
    to-ports=64875
```

把所有 HTTPS 登陆请求定向到 HTTPS 登陆 servlet。64875 是 Hotspot HTTPS servlet 端口。

```
5   D Chain=Hotspot Protocol=tcp action-jump hotspot=!auth jump-target=hs-unauth
```

所有其他的数据包除了 DNS 及来自未认证客户的登陆请求以外都应该通过 hs-unauth 链。

```
6   D Chain=Hotspot Protocol=tcp action=jump hotspot=auth jump-target=hs-auth
```

来自认证用户的数据包通过 hs-auth 链

```
7   D ;;; www.edcwifi.com
      Chain=hs-unauth dst-address=69.195.74.100 protocol=tcp dst-port=80
      Action=return
```

首先在 hs-unauth 链中把所有影响 TCP 协议的都放到/ip hotspot walled-garden ip 子目录中。现在我们把 www.edcwifi.com 从重定向到登陆页面的排除。

```
8   D Chain=hs-unauth Protocol=tcp dst-port=80 action=redirect to-ports=64874
```

所有其他 HTTP 请求都被定向到监听 64874 的 Walled Garden 代理服务器。如果在/ip hotspot walled-garden 子目录有一个 HTTP 请求的 allow 条目，它将被转发到目的。否则，请求将会自动被重定向到 Hotspot 登陆 servlet（端口 64873）。

```
9   D Chain=hs-unauth Protocol=tcp dst-port=3128 action=redirect to-ports=64874
10  D Chain=hs-unauth protocol=tcp dst-port=8080 action=redirect to-ports=64874
```

默认设置的 Hotspot 假设只有这些端口才能用于 HTTP 代理请求。这两个条目用于“捕捉”客户到未知代理的请求。如：使的有可能让带有未知代理设置的客户与 Hotspot 系统能够一起工作。这个特性叫做“通用代理”。如果探测到一个客户正在使用某个代理服务器，系统将自动以 http hotspot 标志对数据包进行标记以便处理未知代理问题。注意已使用的端口（64874）与#8 规则中对 HTTP 请求的一样（所以 HTTP 和 HTTP 代理请求都由相同的代码处理）。

11 D Chain=hs-unauth Protocol=tcp dst-port=443 action=redirect to-port=64875

HTTPS 代理监听 64875 端口

12 D Chain=hs-unauth Protocol=tcp dst-port=25 action=jump jump-target=hs-smtp

对 SMTP 协议协议的重定向也可以在 Hotspot 配置中定义。如果是这样，那么一个重定向规则将被放在 hs-smtp 链中。这个完成后以便带有未知 SMTP 配置的用户能通过服务提供商（你们的）的 SMTP 服务器发送邮件，而代替了用户在自己电脑配置的 SMTP 服务器。

13 D Chain=hs-auth Protocol=tcp Hotspot=http action=redirect to-ports=smtp

对认证用户提供 HTTP 代理服务。认证用户的请求可能需要透明的代理（“通用代理”技术以及广告特征）。http 标志会自动的放在被 Hotspot HTTP 代理探测到的服务器的 HTTP 代理请求（监听 64874 端口的）。这个完成后以便有代理设置的用户可以使用 Hotspot 网关代理用户在自己电脑上配置的代理服务器。这个标志也会被放在任何概要被配置为透明代理的用户所做的 HTTP 请求上。

14 D Chain=hs-auth Protocol=tcp dst-port=25 action=jump jump-target=hs-smtp

对授权用户提供 SMTP 代理（同#12 规则的一样）

包过滤规则

从/ip firewall filter print dynamic 命令，你可以获得：

0 D Chain=forward Hotspot=from-client, !action action=jump jump-target=hs-unauth-to

任何来自未认证且通过路由器的数据包都将被发送到 hs-unauth 链。Hs-unauth 执行基于 IP 的 Walled Garden 过滤器。

1 D Chain=forward Hotspot=to-client, !auth action=jump jump-target=hs-unauth-to

任何通过路由器到达客户的包都将被重定向到另一个叫做 hs-unauth-to 的链。这个链会拒绝到达客户的未认证请求。

2 D Chain=input Hotspot=from-client action=jump jump-target=hs-input

任何从客户到达路由器本身的包将重定向到另一个叫 hs-input 的链。

3 D Chain=hs-input Protocol=udp dst-port=64872 action=accept
4 D chain=hs-input protocol=tcp dst-port=64872-64875 action=accept

允许客户访问本地认证和代理服务。

```
5   D Chain=hs-input Hotspot=!auth action=jump jump-target=hs-unauth
```

所有其他来自未认证客户到路由器本身的数据流都将会与通过路由器的数据流一样的方式被处理。

```
6   D Chain=hs-unauth Protocol=icmp action=return
7   D ;;; www.edcwifi.com
    Chain=hs-unauth dst-address=69.195.74.100 protocol=tcp dst-port=80
    Action=return
```

不仅在 TCP 协议相关的 Walled Garden 条目被添加的 NAT 列表中，在包过滤器中 hs-unauth 链表也会添加在/ip hotspot walled-garden ip 目录中设置的东西。这就是为什么，尽管你只在 NAT 表中添加了一个条目却有两条规则的原因。

```
8   D Chain=hs-unauth Protocol=tcp action=reject reject-with=tcp-reset
9   D Chain=hs-unauth action=reject reject-with=icmp-net-prohibited
```

任何没有被 Walled Garden 记录在表格上的都将被拒绝。注意拒绝 TCP 连接的 TCP 重启的使用。

```
10  D Chain=hs-unauth-to action=reject reject-with=icmp-host-prohibited
```

用 ICMP 拒绝信息拒绝所有到达客户的包。

第十七章 PPoE 配置

PPPoE 基于以太网的点对点协议（point to point protocol over ethernet）当前的 PPPoE 主要被 ISP 商用于 xDSL 和 Cable modems 与用户端的链接，他们几乎与以太网一样。PPPoE 是一种标准的点对点协议（PPP）他们之间只是传输上的差异：PPPoE 使用 modem 链接来代替普通的以太网。一般来说，PPPoE 是基于与用户认证和通过分发 IP 地址给客户端。

RouterOS 能做一个的 RADIUS 客户端 - 你能使用一台 RADIUS 服务器去验证 PPPoE 的客户端和对他们计费

一个 PPPoE 连接由客户端和一个访问集线服务器组成，客户端可以是一个安装了 PPPoE 协议的 windows 电脑。PPPoE 客户端和服务器能工作在任何以太网等级的路由器接口（interface）- wireless 802.11（Aironet，Cisco，WaveLAN，Prism，Atheros），10/100/1000 Mbit/s Ethernet，RadioLAN 和 EoIP（Ethernet over IP tunnel）都支持。

支持的链接

MikroTik RouterOS PPPoE 客户端到任何 PPPoE 服务器（access concentrator）
MikroTik RouterOS PPPoE 服务器（access concentrator）到多个 PPPoE 客户端（客户端包括几乎所有的操作系统和大部分路由器）

多链接 PPP 协议支持 MP，提供 MRRU 协议（能够传输 1500 和大数据包）和基于 PPP 连接的桥接 bridging（使用桥接控制协议 BCP，能发送基于 PPP 连接的原始以太网帧）这样能在没有 EoIP 协议的支持下，设置桥接。

注：当 RADIUS 服务器验证一个用户 CHAP、MS-CHAPv1 或 MS-CHAPv2，RADIUS 戏院不会使用共享密码（shared secret），仅验证回复（authentication reply）被使用。因此如果你有一个错误的共享密码，RADIUS 服务器将接受请求。你可以使用/radius monitor 命令查看 bad-replies 参数，无论什么时候客户在试图连接时这个值都会增加。

规格

需要功能包：PPP
需要等级：Level 3（限制 1 个连接），Level 4（限制 200 个连接），Level 5（限制 500 个连接），Level 6（无限制）
操作路径：/interface pppoe-server, /interface pppoe-client
协议标准和技术： PPPoE （RFC 2516）
硬件要求：PPPoE 服务器的需要增加 RAM 和提高 CPU 性能，每个连接使用 9kib（如果限流被使用额外还需要增加 10KiB）

PPPoE Client 设置

操作路径：/interface pppoe-client

属性描述

Name（名称；默认：pppoe-out1）- PPPoE 的接口名称
Interface（名称）- 选择 PPPoE 服务器的接口使连接通过
Mtu(整型;默认:1480)- 最大传输单位。最适合的 MTU 值(以避免以太网链接的 1500-byte，设置为 1480 以避免数据包的重复存储)
Mru(整型;默认:1480)- 最大接收单位。最适合的 MRU 值(以避免以太网连接的 1500-byte，设置为 1480 以避免数据包的重复存储)
user（文本；默认：“”）- 链接 PPPoE 服务器的用户账号
Password（文本；默认：“”）- 连接 PPPoE 服务器的用户密码
Profile（名称）- 连接的默认策略
Allow（multiple choice：mschap2，mschap1，chap，pap；default：mschap2，mschap1，chap，pap）- 客户端使用的验证协议
Service-name（文本；默认：“”）在访问集线器上设定指定服务名（AC）
Ac-name（文本；默认：“”）- 这条可以为空白，当客户端与任何一个访问集线器相连，会选取该服务名
Add-default-route（yes | no；默认：no）- 是否添加动态默认路由
Dial-no-demand（yes | no；默认：no）- 当连接唯一的 AC 时，传输数据产生，在断开连接，没有传输数据时，idle-timeout 将被设置
Use-peer-dns（yes | no；默认：no）- 是否使用路由器的默认 DNS 给 PPP 的 DNS

注：如果存在一条默认的 PPPoE 的路由，add-default-rouote 将不会创建一个新的路由

在 gig 接口上添加和启用客户端，连接 AC 提供的 testsn 服务名，使用的用户账号 John 和密码 password：

```
[admin@RemoteOffice] interface pppoe-client> add interface=gig \
\... service-name=testSN user=john password=password disabled=no
[admin@RemoteOffice] interface pppoe-client> print
Flags: X - disabled, R - running
 0  R name="pppoe-out1" mtu=1480 mru=1480 interface=gig user="john"
      password="password" profile=default service-name="testSN" ac-name=""
      add-default-route=no dial-on-demand=no use-peer-dns=no
```

监视 PPPoE 客户端

命令名称：/interface pppoe-client monitor

属性描述

Status（文本）- 客户端的状态

Dialing - 拨号连接的情况
Verifying password… - 确定连接到服务器，密码正在核对处理
Terminated - 接口没有启用，或是另一端未建立连接
Encoding（文本）- 在该条连接中使用加密和编码
Uptime（时间）- 连接时间显示为天、时、分、秒
Service-name（文本）- 客户端连接的服务器名称
Ac-mac（MAC 地址）- 客户端已经连接的访问集线器（AC）MAC 地址

监视 pppoe-out1 连接情况：

```
[admin@MikroTik] interface pppoe-client> monitor pppoe-out1
        status: "connected"
        uptime: 10s
      encoding: "none"
  service-name: "testSN"
       ac-name: "10.0.0.1"
        ac-mac: 00:C0:DF:07:5E:E6

[admin@MikroTik] interface pppoe-client>
```

PPPoE server 设置

操作路径：/interface pppoe-server server

PPPoE server（access concentrator）支持在每一个接口上的多服务，需要设置不同的 service 名称，当前 PPPoE server 的吞吐量在一个 celeron 600 CPU 测试达到 160 Mb/s，如果使用更高性能的 CPU，吞吐量将会程比例的增加。

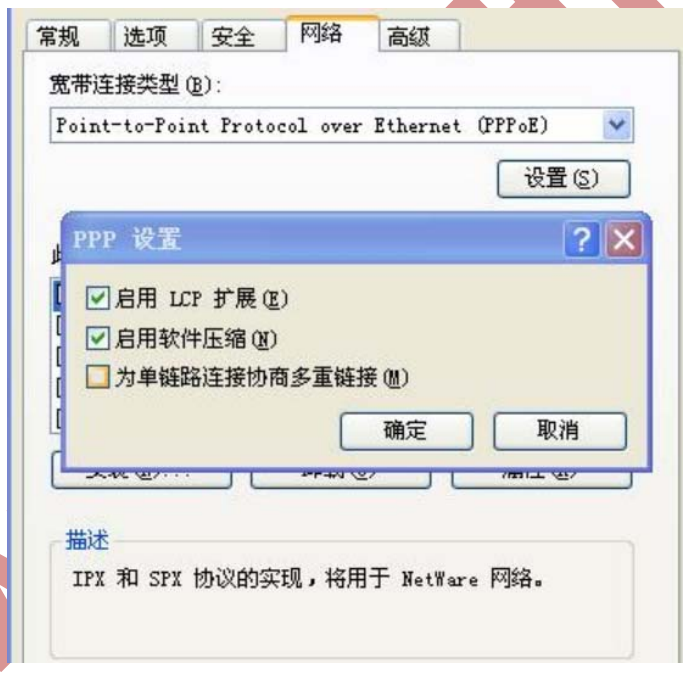
Service-name（文本）- PPPoE 服务名称
Mtu（整型；默认：1480）- 最大传输单位。最适合的 MTU 值（以避免以太网连接的 1500-byte，设置为 1480 以避免数据包的重复存储）
Mru（整型；default：1480）- 最大接受单位。最适合的 MRU 值（以避免以太网连接的 1500-byte，设置为 1480 以避免数据包的重复存储）
Mrru（整型：512..65535；默认：bisabled）- 在连接中能被接收的最大数据包长度。如一个数据包比隧道的 MTU 值大时，将会被分割到多个数据包中，允许实际大小的 IP 或以太网的数据包发送到隧道。
Authentication（多种选择：mschap2 | mschap1 | chap | pap；默认：mschap2，mschap1，chap，pap）- 验证算法
Keepalive-timeout - 定义实际周期（秒）连接开始后路由器每秒钟会发出 keepalive 数据包。如果在设定的时间周期内没有传输和没有 keepalive 回应，客户端将会被认为失去连接
One-session-per-host（yes | no；默认：no）- 每次只允许一个主机对话连接（MAC 地址被确定）。如果主机将试着去建立一个新的对话连接，旧的一个将会被关闭

Default-profile（name；默认：default）- 使用默认的策略配置
Max-sessions（整型；默认：0）- AC 能福能的最大客户端数量
  0 - 没有限制
Interface（名称）- 客户端连接的网卡接口

注：keepalive-timeout 值通常情况下设置为 10。如果你设置为 0，路由器将不会断开客户端，知道他们自己注销或是路由器重启该用户账户才会断开。解决这个问题，one-session-per-host 属性需启用

安全提示：请不要分配一个 IP 地址到 PPPoE 的物理网卡上，避免出现用户不通过 PPPoE 验证即可上网的情况。

明确的讲 MRRU 意思为基于单连接的 MP，该协议被为拆分大数据包为更小的。在 windows 下在网络属性下，设置按钮中打开“为单链路连接协商多重链接”MRRU 是强行设置为 1614。这个设置有益于超载路线 MTU 探测失败。且 MP 协议应在双方都被启用。



ADSL 拨号上网事例

ADSL 用户名：user@169
　　密码：1234
Service Name：CHN-Telecom

1：添加 PPPoE clients

```
[admin@Router] interface pppoe-client>
[admin@Router] interface pppoe-client> add interface=ether1 mtu=1492 mru=1492
service-name=CHN-Telecom user= user@169 password=1234
add-default-route=yesuse-peer-dns=yes
[admin@ROUTER] interface pppoe-client> print
Flags: X - disabled, R - running
0  X name="pppoe-out1" mtu=1492 mru=1492 interface=ether1 user=user@169
    password=1234 profile=default service-name=CHN-Telecom ac-name=""
    add-default-route=yes dial-on-demand=no use-peer-dns=yes
```

PPPoE 拨号已经配置好，接下来将 ADSL MODEM 的网线连接号进行以下操作就可以连通了。

```
[admin@Router] interface pppoe-client>enable 0
[admin@Router] interface pppoe-client> monitor pppoe-out1
        status: "connected"
        uptime: 10s
      encoding: "none"
   service-name: "CHN-Telecom"
       ac-name: ""
        ac-mac: 00:C0:DF:07:5E:E6
```

之后还需在 ip firewall mangle 中添加一条规则：

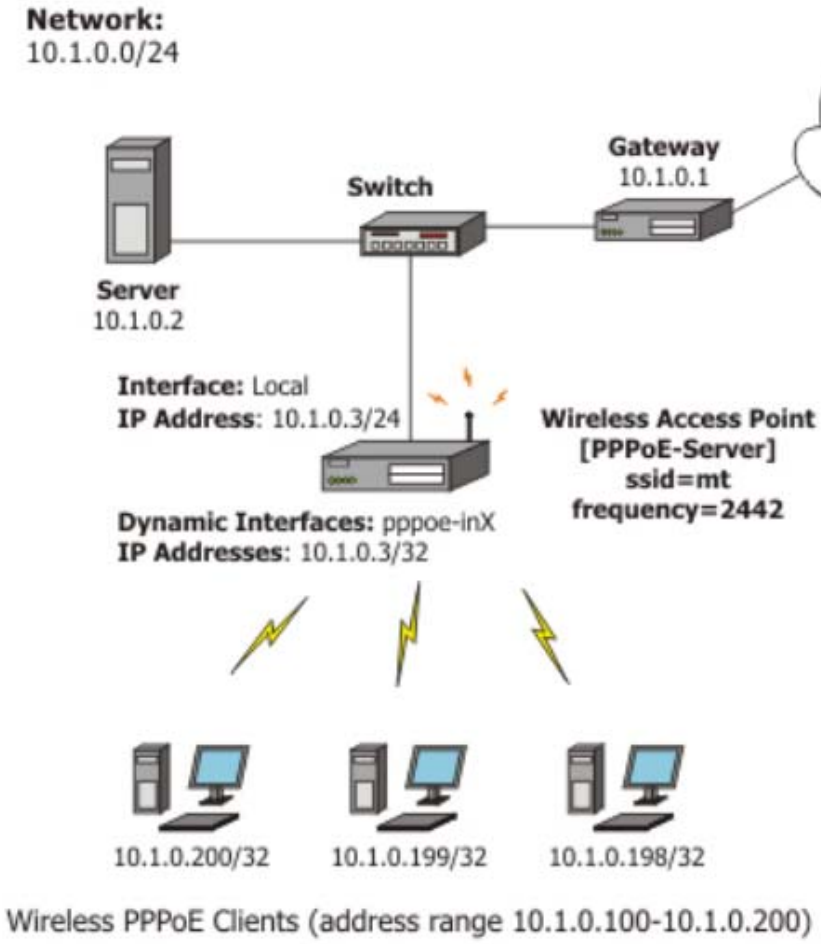
```
[admin@Router] ip firewall mangle> add chain=forward protocol=tcp tcp-flags=syn
action=change-mss new-mss=1440
[admin@Router] ip firewall mangle> print
Flags: X - disabled, I - invalid
0   chain=forward protocol=tcp tcp-flags=syn action=change-mss
    new-mss=1440
```

最后如果你要起用 nat 功能，不要忘了在 ip firewall nat 设置 IP 伪装。

基于 802.11g 无线网络的 PPPoE 服务

在无线网络中，服务器可以设置在一个访问节点（Access Point），任意一个 RouterOS 客户端或是 windows 客户端都可以连接访问节点的 PPPoE 认证。无线网卡的 MTU 可以设置为 1600，因此接口上的 MTU 设置为 1500，这可以充分利于 1500byte 传输数据包，并避免 MTU 比 1500 低出现的任何问题。

让我们考虑下面的配置，mikrotik 无线 AP 能使无线用户端通过验证后访问到本地的网络：



首先，需要配置无线网卡：

```
[admin@PPPoE-Server] interface wireless> set 0 mode=ap-bridge \
   frequency=2442 band=2.4ghz-b/g ssid=mt disabled=no
[admin@PPPoE-Server] interface wireless> print
Flags: X - disabled, R - running
 0    name="wlan1" mtu=1500 mac-address=00:01:24:70:53:04 arp=enabled
      disable-running-check=no interface-type=Atheros AR5211
      radio-name="000124705304" mode=station ssid="mt" area=""
      frequency-mode=superchannel country=no_country_set antenna-gain=0
      frequency=2412 band=2.4ghz-b scan-list=default rate-set=default
      supported-rates-b=1Mbps,2Mbps,5.5Mbps,11Mbps
      supported-rates-a/g=6Mbps,9Mbps,12Mbps,18Mbps,24Mbps,36Mbps,48Mbps,
                         54Mbps
      basic-rates-b=1Mbps basic-rates-a/g=6Mbps max-station-count=2007
      ack-timeout=dynamic tx-power=default tx-power-mode=default
      noise-floor-threshold=default periodic-calibration=default
      burst-time=disabled fast-frames=no dfs-mode=none antenna-mode=ant-a
      wds-mode=disabled wds-default-bridge=none wds-ignore-ssid=no
      update-stats-interval=disabled default-authentication=yes
      default-forwarding=yes default-ap-tx-limit=0 default-client-tx-limit=0
      hide-ssid=no security-profile=default disconnect-timeout=3s
      on-fail-retry-time=100ms preamble-mode=both
[admin@PPPoE-Server] interface wireless>
```

现在，配置以太网卡，添加默认 IP 地址和设置默认路由：

```
[admin@PPPoE-Server] ip address> add address=10.1.0.3/24 interface=Local
[admin@PPPoE-Server] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         BROADCAST       INTERFACE
 0   10.1.0.3/24         10.1.0.0        10.1.0.255      Local
[admin@PPPoE-Server] ip address> /ip route
[admin@PPPoE-Server] ip route> add gateway=10.1.0.1
[admin@PPPoE-Server] ip route> print
Flags: X - disabled, A - active, D - dynamic,
C - connect, S - static, r - rip, b - bgp, o - ospf
 #     DST-ADDRESS       G GATEWAY         DISTANCE INTERFACE
 0 ADC 10.1.0.0/24                                  Local
 1 A S 0.0.0.0/0          r 10.1.0.1         1       Local
[admin@PPPoE-Server] ip route> /interface ethernet
[admin@PPPoE-Server] interface ethernet> set Local arp=proxy-arp
[admin@PPPoE-Server] interface ethernet> print
Flags: X - disabled, R - running
 #    NAME                                MTU   MAC-ADDRESS       ARP
 0  R Local                                1500  00:0C:42:03:25:53 proxy-arp
[admin@PPPoE-Server] interface ethernet>
```

添加 PPPoE server 到无线网卡上：

```
[admin@PPPoE-Server] interface pppoe-server server> add interface=wlan1 \
  service-name=mt one-session-per-host=yes disabled=no
[admin@PPPoE-Server] interface pppoe-server server> print
Flags: X - disabled
 0   service-name="mt" interface=wlan1 max-mtu=1480 max-mru=1480
     authentication=pap,chap,mschap1,mschap2 keepalive-timeout=10
     one-session-per-host=yes max-sessions=0 default-profile=default
[admin@PPPoE-Server] interface pppoe-server server>
```

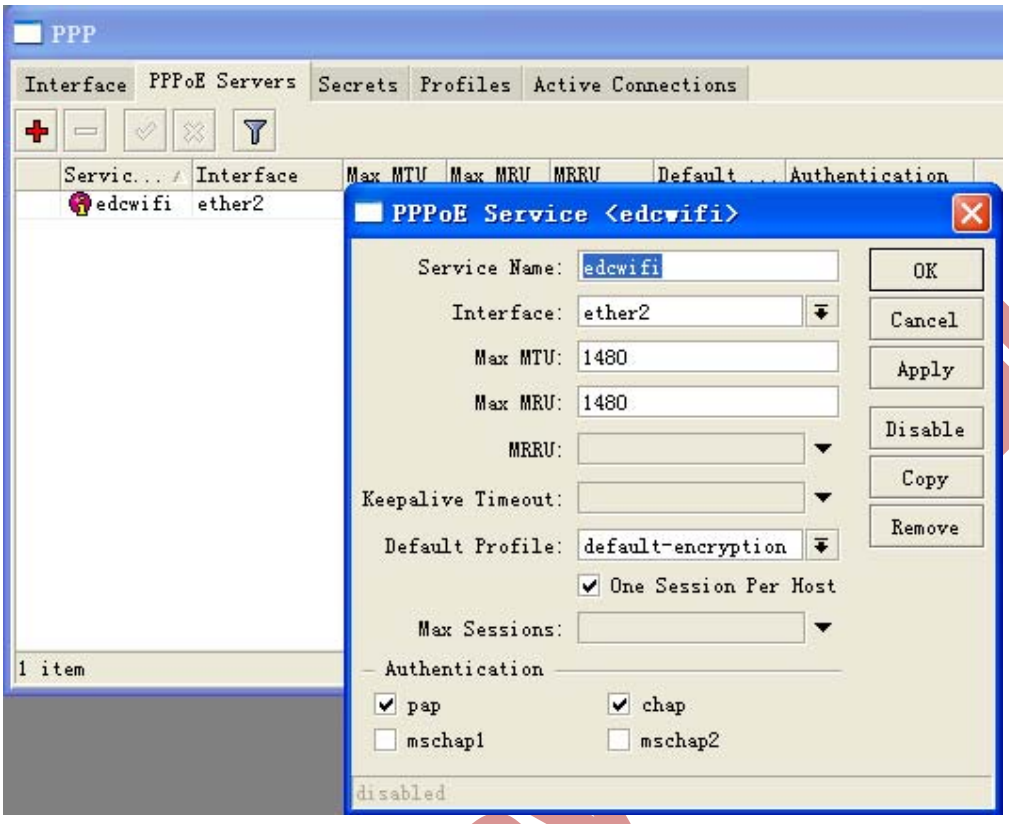
最后，设置 PPPoE clients：

```
[admin@PPPoE-Server] ip pool> add name=pppoe ranges=10.1.0.100-10.1.0.200
[admin@PPPoE-Server] ip pool> print
 # NAME                                    RANGES
 0 pppoe                                   10.1.0.100-10.1.0.200
[admin@PPPoE-Server] ip pool> /ppp profile
[admin@PPPoE-Server] ppp profile> set default use-encryption=yes \
  local-address=10.1.0.3 remote-address=pppoe
[admin@PPPoE-Server] ppp profile> print
Flags: * - default
 0 * name="default" local-address=10.1.0.3 remote-address=pppoe
    use-compression=no use-vj-compression=no use-encryption=yes only-one=no
    change-tcp-mss=yes

 1 * name="default-encryption" use-compression=default
    use-vj-compression=default use-encryption=yes only-one=default
    change-tcp-mss=default
[admin@PPPoE-Server] ppp profile> .. secret
[admin@PPPoE-Server] ppp secret> add name=w password=wkst service=pppoe
[admin@PPPoE-Server] ppp secret> add name=l password=ltp service=pppoe
[admin@PPPoE-Server] ppp secret> print
Flags: X - disabled
 #   NAME       SERVICE CALLER-ID PASSWORD  PROFILE          REMOTE-ADDRESS
 0   w          pppoe             wkst      default          0.0.0.0
 1   l          pppoe             ltp       default          0.0.0.0
[admin@PPPoE-Server] ppp secret>
```

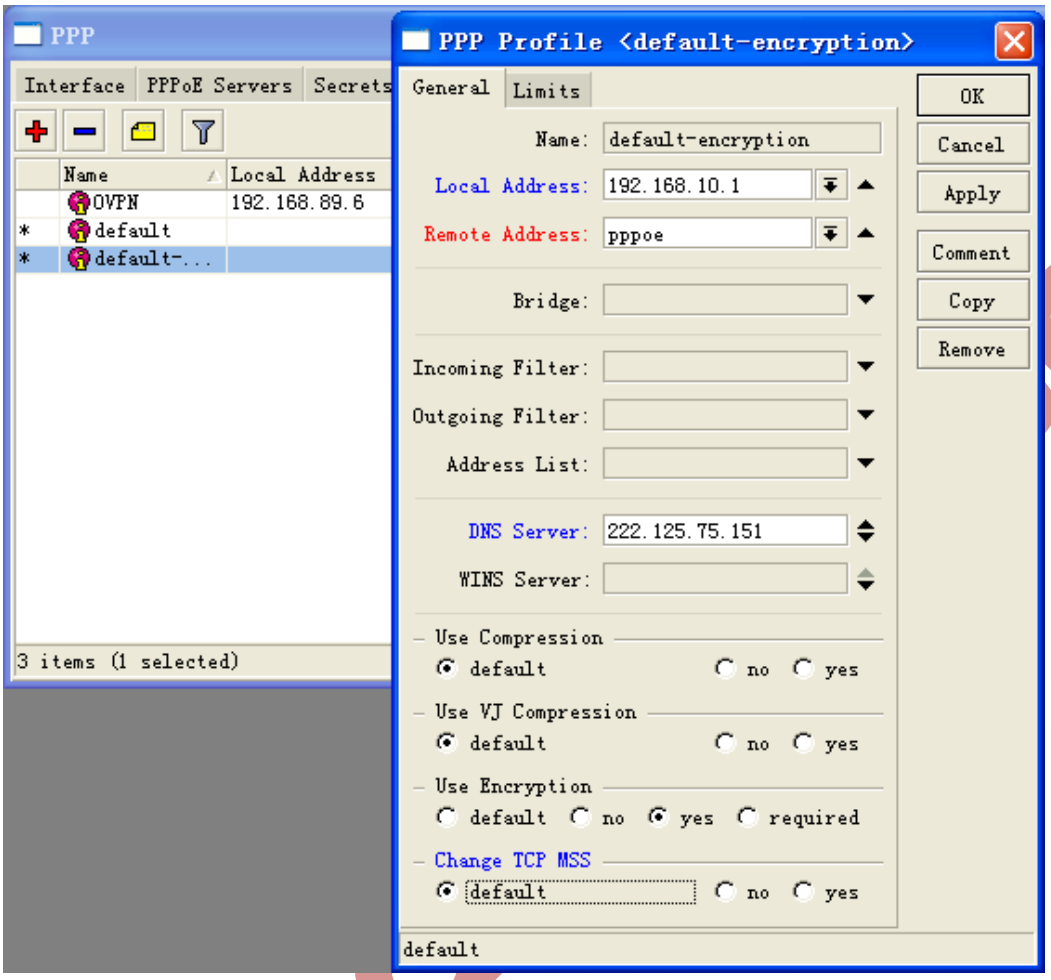
注：在 windows XP 中的 PPPoE 客户端内建加密功能，但 RASPPPoE 没有。因此，如果计划不在支持比 windows XP 老的 windows 客户端，推荐在 default 规则配置把 require-encryption 值选择为 yes。在其他一些应用中，可以服务设置为接受为加密的数据。

Winbox 配置 PPPoE 服务

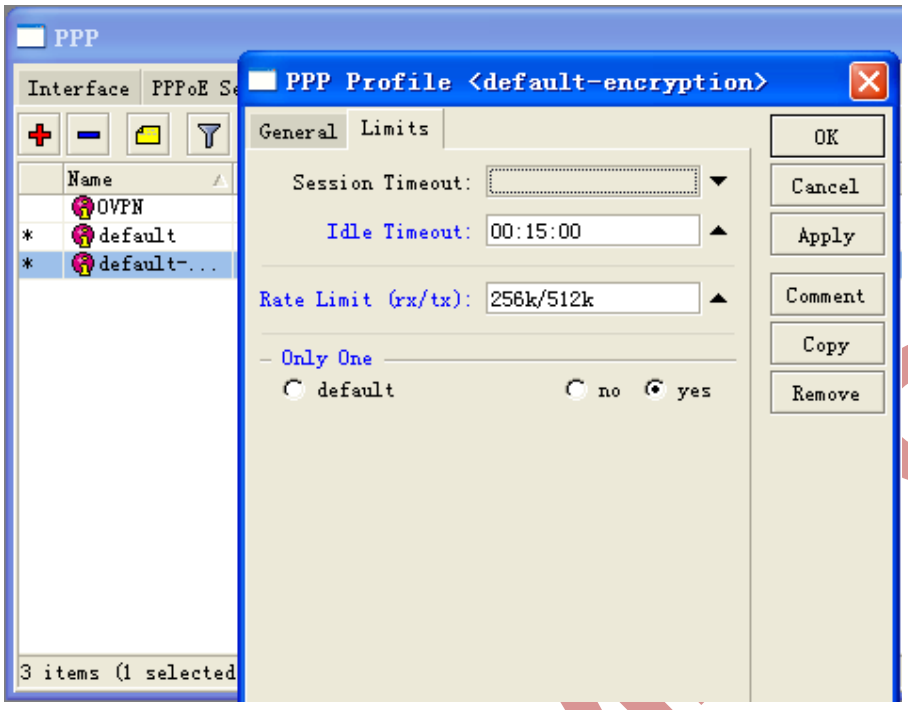
通过 winbox 配置 PPPoE 服务器，这里我们首先通过进入 PPP 目录下的 PPPoE server，配置 service name 为 edcwifi，用于 PPPoE 服务器名，并把 PPPoE 服务指向 ether2 的网卡上，其他参数如图所示：



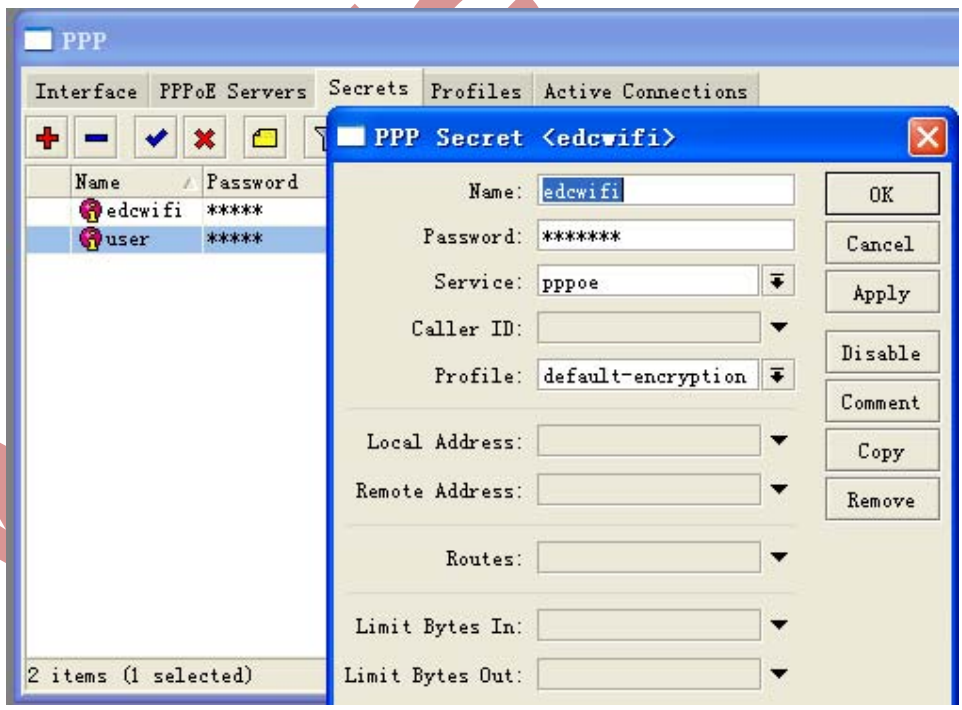
这里我们选择的是 default-encryption 的 profile 规则，所有我们需要进入 profiles 中配置该规则的参数，loacl-address 为本地路由器网关 IP，remote-address 则是远程客户端 IP 地址。这里我们设置 local-address 为 192.168.10.1 ，remote-address 添加在 ip pool 中设置好的地址池 PPPoE，然后配置 DNS 参数，其他配置如图：



下面配置 limits 参数：



这样用户的组规则配置完成，根据需要也可以增加其他的组规则到 profile 中。接下来配置每个用户信息，进入 PPP secrets 添加用户账号：



这里 name 为用户账号名，password 为用户密码。Profile 选择刚才设置好的 default-encryption，根据情况也可以调用其他相应的 profile 规则。配置完用户的账号和密码后，PPPoE 服务就可以启动。

[键入文字]

大型 PPPoE 服务的综合应用

PPPoE 认证由于是基于 OSI 七层参考模型第二层运行，所以不会受 IP 曾数据的影响，特别是 ARP 协议,这样可以避免现在比较常见的 ARP 病毒攻击。PPPoE 是让每一个用户在二层 MAC 地址间建立一个虚拟的隧道，即保证了数据的安全，又保证了稳定。比起通过 IP 方式认证的 web 页面要稳定安全的多。在一些网吧为了避免 ARP 病毒的侵扰，也在网吧内部建立的 PPPoE 认证方式，避免 ARP 对网吧带来上网电脑频繁掉线问题。

现在几乎所有的人都在使用 windows XP 或以上的操作系统，这些操作系统都自带了 PPPoE 拨号软件，即用户不需要太复杂的操作，就可以建立一个虚拟拨号连接。

1、 首先采用 RouterOS 作为接入路由器和外网防火墙,这里我们采用 RB1000 设备或者 RouterOS x86PC 系统作为外网接入路由器，可以实现多线路多运营商的路由器，并作为 nat 转换设备，减轻 PPPoE 认证服务器的压力。
2、 在接入路由器下面，我们可以根据用户数量，建立多个 PPPoE 服务器，采用 PPPoE 集群服务器方式均衡用户，一般通过一台高性能的 PC，如双核的服务器支持 1000 个左右 PPPoE 认证用户同时在线，更高的 PC 配置可以获得更高的在线用户数量。
3、 通过核心交换 VLAN 的 Trunk 连接用户层交换机，并分配每个用户连接那个 PPPoE 服务器，这样可以通过 VLAN 划分用户区域，隔离不必要的数据，减少广播风暴。用户接入可以通过以太网的有线连接，也可以通过无线的 AP 接入网络，连接方式灵活多样化。
4、 所有 PPPoE 服务器都采用同一个 radius 服务器，这样账号便于管理，特别在多 PPPoE 的集群认证下有利于冗余的工作，在一台 PPPoE 服务器停机后，其余的设备可以接替工作。配置 radius 服务器也可以分担 RouterOS 在账号管理的负荷。

故障分析

我能够连接到服务器，ping 也能完全通过，但我仍然不能开打 web 页面？

确定你在路由器上指定了正确的 DNS 服务器（在/ip dns 或在/ppp profile 中的 dns-server 参数）

我能使 PPPoE 连接小点的数据包（例如 pings）

你需要改变所以经过 PPPoE 连接的 mss 数据包为 1440：

```
[admin@MT] interface pppoe-server server> set 0 max-mtu=1440 max-mru=1440
[admin@MT] interface pppoe-server server> print
Flags: X - disabled
 0   service-name="mt" interface=wlan1 max-mtu=1440 max-mru=1440
     authentication=pap,chap,mschap1,mschap2 keepalive-timeout=10
     one-session-per-host=yes max-sessions=0 default-profile=default
[admin@MT] interface pppoe-server server>
```

我的 windows PPPoE 客户端得到了来至 mikrotik pppoe server 的 IP 地址和默认网关，但不能出 PPPoE server 并且不能连接外部网络。

PPPoE 服务器没有与客户端连接，为 PPPoE 客户端的地址配置伪装（masquerading）或是确定你为客户端分配的地址段指定了正确的路由，或是你在以太网卡上启用了 proxy-ARP（请看 IP 地址和地址解析协议 Address Resolution Protocol（ARP））

我的 windows XP 不能连接到 PPPoE 服务器

你要在 XP 的 PPPoE 客户端属性中指明“service name”。或是没有在 mikrotik PPPoE 服务器配置服务名（server name），这样你会得到“line is busy”错误或是系统显示“verifying password - unknown error”

我想要记录连接建立的日志

在/system logging facility 中配置日志信息并启用 PPP 日志类型

第十八章 PPTP

PPTP（点对点隧道协议）支持 IP 上的加密隧道。Mikrotik RouterOS 工具包含对 PPTP 客户和服务器的支持。PPTP 隧道的基本应用：

因特网上的安全路由器-路由器隧道
连接（桥接）本地企业网或 LAN（当使用了 EoIP 时）
对移动或远程客户远程访问企业网/公司的 LAN（参见 windows 的 PPTP 设置以获取更多信息）

每个 PPTP 连接都包含一个服务器和客户。Mikrotik RouterOS 可能作为一个服务器或者客户工作——或者，对多种配置，它可以对某些连接是服务器而对其他连接时客户。例如，下面创建的客户可以连接到 windows 2000 服务器，另一个 mikrotik touter，或另一个支持 PPTP 服务器的路由器。

快速设置向导

在两个 IP 地址为 10.5.8.104（PPTP 服务器）及 10.1.0.172（PPTP 客户）的 mikrotik 路由器之间创建一个 PPTP 隧道，参考下面的步骤：

PPTP 服务器上的设置：

1. 添加一个用户：

```
[admin@PPTP-server] PPP secret> add name=jack password=pass
local-address=10.0.0.1 remote-address=10.0.0.2
```

2. 启用 PPTP 服务器：

```
[admin@PPTP-server] interface pptp-server server> set enabled=yes
```

PPTP 客户的设置：

1. 添加 PPTP 客户：

```
[admin@PPTP-Client] interface pptp-client> add user=jack password=pass
connect-to=10.5.8.104 disabled=no
```

规格

功能包要求：PPP
等级要求：Level 3（限制 1 个在线），Level 4（限制 200 在线），Level 5、6（无限制）
操作路径：/interface pptp-server, /interface pptp-client
标准与技术：PPTP（RFC 2637）

点对点隧道协议（PPTP）是一种支持多协议虚拟专用网络的网络技术。通过该协议，远程用户能够通过 windows 客户端或者路由器，以及其他装有点对点协议的系统安全访问公司网络，并能拨号连入本地 ISP，通过 Internet 安全连接到公司网络。

PPTP 包含了 PPP 认证及对每个 PPTP 连接的账户管理。全部的认证和每个链接的账户管理可以通过 RADIUS 客户或本地完成。支持 MPPE 40bit RC4 以及 MPPE 128bit RC4 加密。

PPTP 的连接采用的是 TCP 端口 1723 和 IP 协议 GRE（类属路由封装，IP 协议 ID 47）。PPTP 可以通过启用定为 TCP 端口 1723 和 47，注意让相应的路由器不会对这两个端口做防火墙过滤等操作，否则 PPTP 连接会失效。

PPTP 客户设置

操作路径：/interface pptp-client

属性描述

Add-default-route（yes | no；default：no）- 是否像使用默认路由器（网关）一样使用该客户连接到的服务器
Allow（multiple choice：mschap2，mschap1，chap，pap；default：mschap2，mschap1，chap，pap）- 允许客户用来认证的协议
Connect-to（IP address）- PPTP 服务器连接到的 IP 地址
Mru（整型；default：1460）- 最大接收单元。最优值是隧道工作的接口 MRU 减少 40（所以，1500 字节以太网连接设置 MRU 为 1460 以避免包的分割）
Mtu（整型；default：1460）- 最大传输单元。最优值是隧道工作的接口 MTU 减少 40（所以，1500 字节以太网连接设置 MTU 为 1460 以避免包的分割）
Name（name；default：pptp-outN）- 参考接口名
Password（text；default：“”）- 当登陆远程服务器时用户的密码
Profile（name；default：default）- 当连接到远程服务器时使用的概要简介
User（text）- 当登陆到远程服务器时使用的用户名

使用用户名为 John 密码 John，设置 PPTP 名为 text2 的客户连接到 10.1.1.12PPTP 服务器并使用它作为默认网关：

```
[admin@MikroTik] interface pptp-client> add name=test2 connect-to=10.1.1.12 \
\... user=john add-default-route=yes password=john
[admin@MikroTik] interface pptp-client> print
Flags: X - disabled, R - running
 0 X  name="test2" mtu=1460 mru=1460 connect-to=10.1.1.12 user="john"
      password="john" profile=default add-default-route=yes

[admin@MikroTik] interface pptp-client> enable 0
```

属性描述

Encoding（text）- 加密及编码（如果非对称，使用‘/’分隔）在该连接中使用
Status（text）- status of the client
Dialing - 试图进行连接
Verifying password… - 连接已建立到服务器，正在核实密码
Connected - 毋需解释的
Terminated - 没有启用借口或另一端不能建立连接
Uptime（time）- 以天，小时，分钟以及秒钟显示的连接时间
命令名：/interface pptp-client monitor

一个已建立连接的实例：

```
[admin@MikroTik] interface pptp-client> monitor test2
     uptime: 4h35s
   encoding: MPPE 128 bit, stateless
     status: Connected
[admin@MikroTik] interface pptp-client>
```

PPTP 服务器设置

操作路径：/interface pptp-server server

PPTP 服务器为每个链接的 PPTP 客户创建了一个动态的接口。PPTP 连接依靠你所有的证书登记从客户计数。Level 1 证书允许一个 PPTP 客户，Level 3、4 证书最多允许 200 客户，Level 5、6 证书没有 PPTP 客户限制。

为了创建 PPTP 用户，你应该咨询 PPP secret 以及 PPP profile 手册。也可以使用 mikrotik 路由器作为 RADIUS 客户来注册 PPTP 用户。

属性描述

Authentication（multiple choice：pap | chap | mschap1 | mschap2；default：mschap2）- 认证算法
Default-profile - 默认概要信息
Enabled（yes | no；default：no）- 定义 PPTP 服务器是否启用
Keepalive-timeout（time；default：30）- 定义路由器开始每秒发送持买入活时间数据包之后的时间段（以秒计算）。如果没有流量并且没有保持活动，在那段时间将出现反应（例如，2 * keepalive-timeout），没有反应的客户将被宣布为断开连接
Mru（整型；default：1460）- 最大接收单元。最优值是隧道工作的接口 MRU 减少 40(所以，1500 字节以太网连接设置 MRU 为 1460 以避免包的封装问题)

Mtu（整型；default：1460）- 最大传输单元。最优值是隧道工作的接口 MTU 减少 40（所以，1500 字节以太网连接设置 MTU 为 1460 以避免包的封装问题）

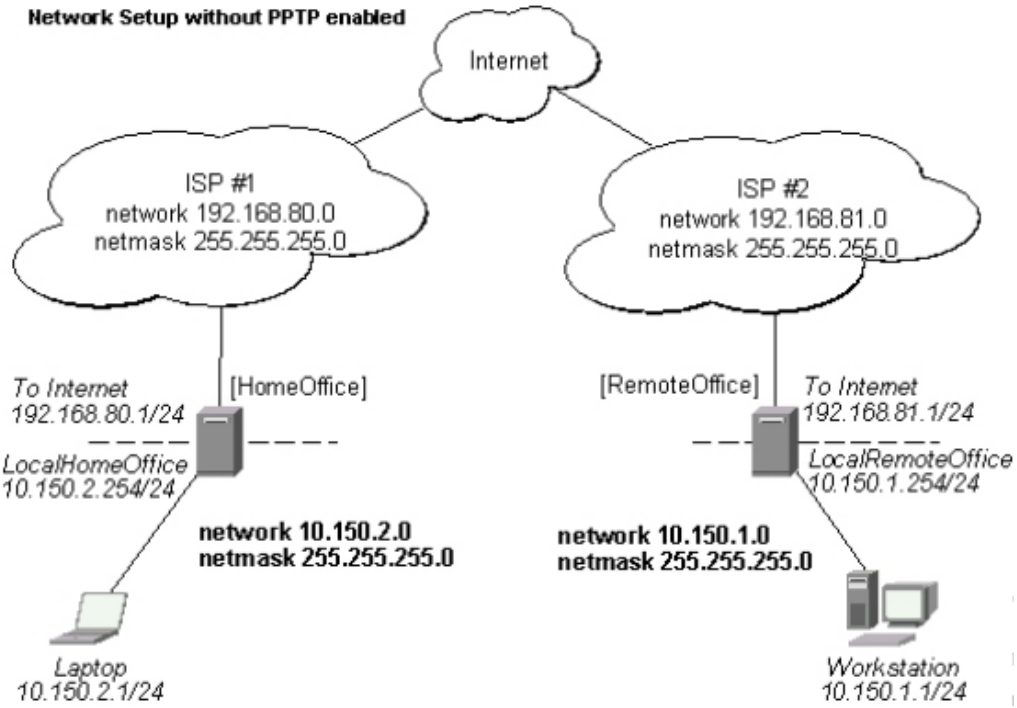
启用 PPTP 服务器：

```
[admin@MikroTik] interface pptp-server server> set enabled=yes
[admin@MikroTik] interface pptp-server server> print
            enabled: yes
                mtu: 1460
                mru: 1460
     authentication: mschap2,mschap1
   keepalive-timeout: 30
     default-profile: default
[admin@MikroTik] interface pptp-server server>
```

PPTP 应用实例

Router-to-router 安全隧道实例

以下是一个使用因特网上的加密 PPTP 隧道连接两个企业网的例子：



在这个例子中有两个路由器：

[HomeOffice]

接口 LocalHomeOffice 10.150.2.254/24
接口 ToInternet 192.168.80.1/24

[RemoteOffice]

接口 ToInternet 192.168.81.1/24
接口 LocalRemoteOffice 10.150.1.254/24

每个路由器连接到一个不同的 ISP。任何一个路由器可以通过因特网访问其他的路由器。

在 preforma PPTP 服务器，用户必须对客户设置：

```
[admin@HomeOffice] ppp secret> add name=ex service=pptp password=lkjrht
local-address=10.0.103.1 remote-address=10.0.103.2
[admin@HomeOffice] ppp secret> print detail
Flags: X - disabled
 0   name="ex" service=pptp caller-id="" password="lkjrht" profile=default
     local-address=10.0.103.1 remote-address=10.0.103.2 routes==""

[admin@HomeOffice] ppp secret>
```

然后应该在 PPTP 服务器列表中添加用户：

```
[admin@HomeOffice] interface pptp-server> add user=ex
[admin@HomeOffice] interface pptp-server> print
Flags: X - disabled, D - dynamic, R - running
 #    NAME                USER        MTU   CLIENT-ADDRESS  UPTIME   ENC...
 0    pptp-in1             ex

[admin@HomeOffice] interface pptp-server>
```

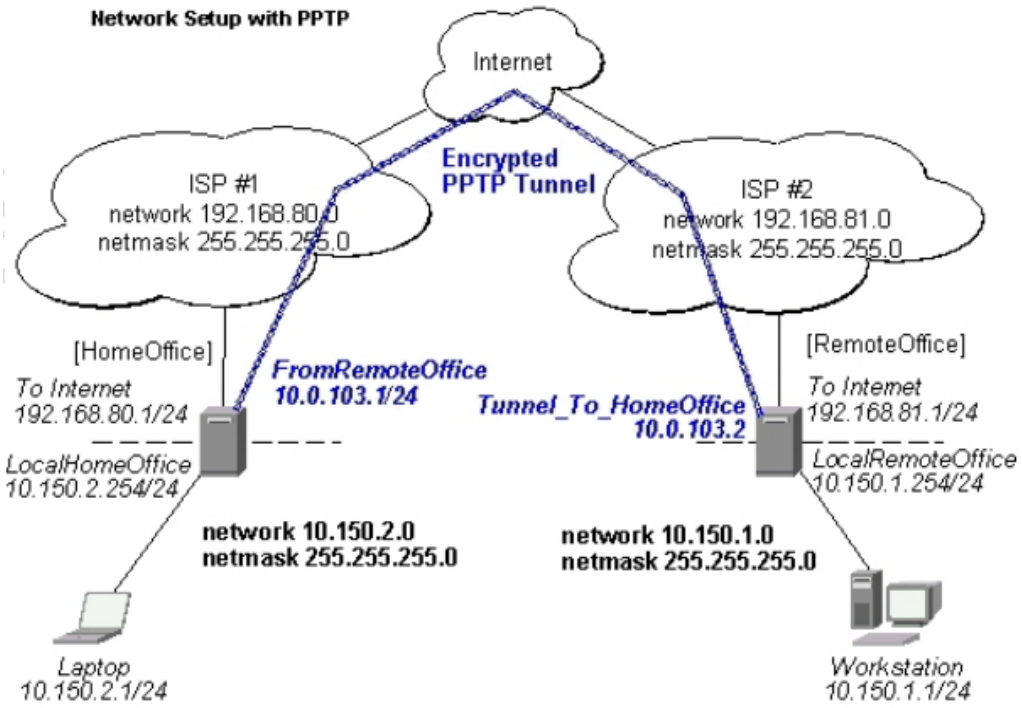
最后，启用服务器：

```
[admin@HomeOffice] interface pptp-server server> set enabled=yes
[admin@HomeOffice] interface pptp-server server> print
          enabled: yes
              mtu: 1460
              mru: 1460
    authentication: mschap2
   default-profile: default
[admin@HomeOffice] interface pptp-server server>
```

在 RemoteOffice 路由器添加一个 PPTP 客户：

```
[admin@RemoteOffice] interface pptp-client> add connect-to=192.168.80.1 user=ex \
\... password=lkjrht disabled=no
[admin@RemoteOffice] interface pptp-client> print
Flags: X - disabled, R - running
 0  R name="pptp-out1" mtu=1460 mru=1460 connect-to=192.168.80.1 user="ex"
     password="lkjrht" profile=default add-default-route=no

[admin@RemoteOffice] interface pptp-client>
```

这样，一个 PPTP 隧道就在路由器之间创建好了。这个隧道就像在 IP 地址为 10.0.103.1 及 10.0.103.2 的路由器之间的以太网点对点连接。它使得在第三网络部分上的路由器之间能够直接通信。



为了在 PPTP 隧道上路由本地企业网你需要添加以下路由：

```
[admin@HomeOffice] > ip route add dst-address 10.150.1.0/24 gateway 10.0.103.2
[admin@RemoteOffice] > ip route add dst-address 10.150.2.0/24 gateway 10.0.103.1
```

在 PPTP 服务器上它或者可以通过使用用户配置的 router 参数完成：

```
[admin@HomeOffice] ppp secret> print detail
Flags: X - disabled
 0   name="ex" service=pptp caller-id="" password="lkjrht" profile=default
     local-address=10.0.103.1 remote-address=10.0.103.2 routes==""

[admin@HomeOffice] ppp secret> set 0 routes="10.150.1.0/24 10.0.103.2 1"
[admin@HomeOffice] ppp secret> print detail
Flags: X - disabled
 0   name="ex" service=pptp caller-id="" password="lkjrht" profile=default
     local-address=10.0.103.1 remote-address=10.0.103.2
     routes="10.150.1.0/24 10.0.103.2 1"

[admin@HomeOffice] ppp secret>
```

测试 PPTP 隧道连接：

```
[admin@RemoteOffice]> /ping 10.0.103.1
10.0.103.1 pong: ttl=255 time=3 ms
10.0.103.1 pong: ttl=255 time=3 ms
10.0.103.1 pong: ttl=255 time=3 ms
ping interrupted
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 3/3.0/3 ms
```

测试通过 PPTP 隧道到 LocalHomeOffice 接口的连接：

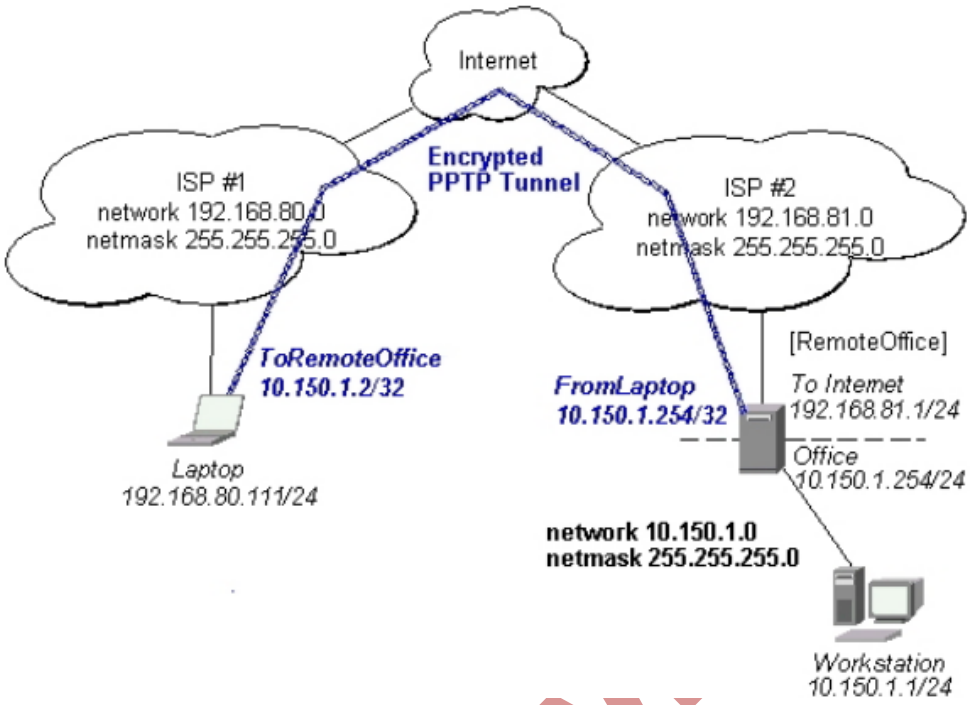
```
[admin@RemoteOffice]> /ping 10.150.2.254
10.150.2.254 pong: ttl=255 time=3 ms
10.150.2.254 pong: ttl=255 time=3 ms
10.150.2.254 pong: ttl=255 time=3 ms
ping interrupted
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 3/3.0/3 ms
```

要在这个安全隧道上桥接一个 LAN，请参考 EoIP 部分手册的例子。要想对该隧道上的流量设置最大速度，请查询 Queues 部分。

通过 PPTP 隧道连接一个远程客户

下面的例子显示了如果通过给定电脑和远程办公网络同一网络 IP 地址的 PPTP 加密隧道把一个电脑连接到一个远程办公网络（不需要在 EoIP 隧道上桥接）

请查询如何设置一个你使用的软件的 PPTP 客户的手册。



这个例子中的路由器：

[RemoteOffice]

接口 ToInternet 192.168.81.1/24
接口 Office 10.150.1.254/24

客户电脑可以通过因特网访问路由器：

在 PPTP 服务器上设置用户账号：

```
[admin@RemoteOffice] ppp secret> add name=ex service=pptp password=lkjrht
local-address=10.150.1.254 remote-address=10.150.1.2
[admin@RemoteOffice] ppp secret> print detail
Flags: X - disabled
 0   name="ex" service=pptp caller-id="" password="lkjrht" profile=default
     local-address=10.150.1.254 remote-address=10.150.1.2 routes==""

[admin@RemoteOffice] ppp secret>
```

然后应该在 PPTP 服务器里表中添加用户：

```
[admin@RemoteOffice] interface pptp-server> add name=FromLaptop user=ex
[admin@RemoteOffice] interface pptp-server> print
Flags: X - disabled, D - dynamic, R - running
 #    NAME               USER        MTU   CLIENT-ADDRESS  UPTIME   ENC...
 0    FromLaptop          ex
[admin@RemoteOffice] interface pptp-server>
```

并且启用服务：

```
[admin@RemoteOffice] interface pptp-server server> set enabled=yes
[admin@RemoteOffice] interface pptp-server server> print
          enabled: yes
              mtu: 1460
              mru: 1460
   authentication: mschap2
   default-profile: default
[admin@RemoteOffice] interface pptp-server server>
```

Windows 的 PPTP 设置

对 Windows NT，2000， 98SE 以及 98 支持 PPTP 客户。Windows98 SE，2000，以及 ME 包括 Windows 设置中的支持或者自动安装 PPTP。对 95，NT，及 98，安装需要从 Microsoft 下载。很多 ISP 都制作了帮助页面以帮助客户进行 Windows PPTP 安装。

PPTP（VPN）安装的简单说明及客户设置 - Windows 98 SE

如果 VPN（PPTP）套件已经安装，选择‘Dial-up Networking’和‘Create a new connection’。创建一个 VPN 的选项应该选择。如果没有 VPN 选项，那么按照下面的安装说明进行。当询问 VPN 服务器主机名或 IP 地址时，输入路由器的 IP 地址。双击‘new’图标并输入正确的用户名和密码（必须在路由器或用于认证的用户数据库中）。

连接的设置在选择了‘connect’按钮后需要 9 秒钟。建议把连接属性进行编辑以便‘NetBEUI’，‘IPX/SPX compatible’，及‘Log on to network’为未选择的。连接的设置时间为在‘connect’按钮选择后 2 秒钟。

为了安装 Windows 98 SE 的 VPN 套件，从‘Start’主目录中选择‘setting’。选择‘control panel’，选择‘Add/Remove Program’，选择‘Windows setup’标签，选择‘communications’软件安装以及‘Details’。在软件列表的底部选择‘Virtual Private Networking’安装。

故障分析

我使用了防火墙但我不能建立 PPTP 建立

确定 TCP 链接到 1723 端口可以通过你的两个站点。而且，TCP 协议 47 应该通过。

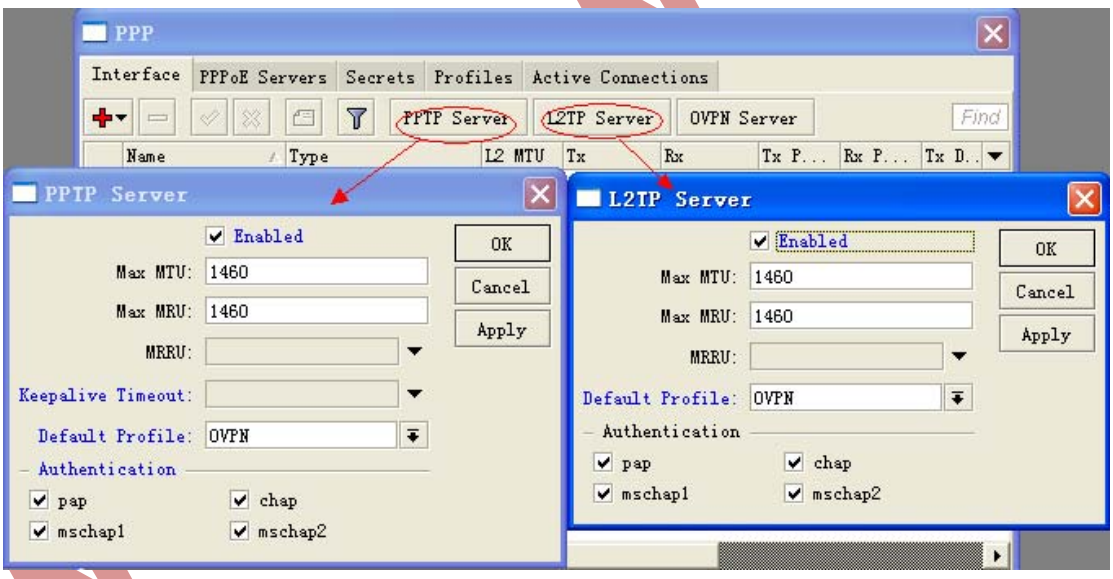
第十九章 PPTP 与 L2TP 服务

PPTP 和 L2TP 都使用 PPP 协议对数据进行封装，然后添加附加包头用于数据在互联网络上的传输。尽管两个协议非常相似，但是仍存在以下几方面的不同：

PPTP 要求互联网络为 IP 网络。L2TP 只要求隧道媒介提供面向数据包的点对点的连接。PPTP 只能在两端点间建立单一隧道。L2TP 支持在两端点间使用多隧道。使用 L2TP，用户可以针对不同的服务质量创建不同的隧道。L2TP 可以提供包头压缩。当压缩包头时，系统开销（overhead）占用 4 个字节，而 PPTP 协议下要占用 6 个字节。L2TP 可以提供隧道验证，而 PPTP 则不支持隧道验证。但是当 L2TP 或 PPTP 与 IPSEC 共同使用时，可以由 IPSEC 共同使用时，可以由 IPSEC 提供隧道验证，不需要在第 2 层协议上验证隧道

同时建立 PPTP 和 L2TP 服务器

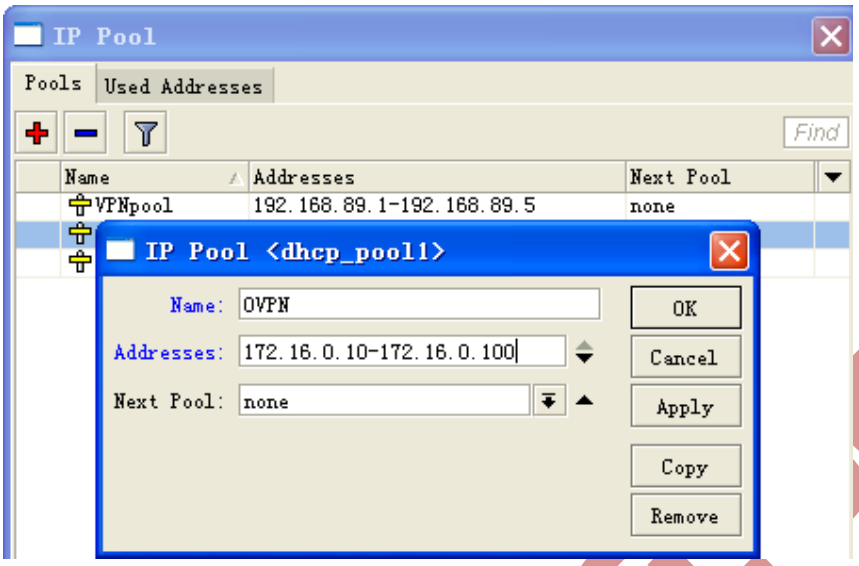
首先，我们看一下 PPTP 和 L2TP 的建立，同样是在 PPP 的目录下，只是选择的服务不同，一个是 PPTP 服务，一个是 L2TP 服务：



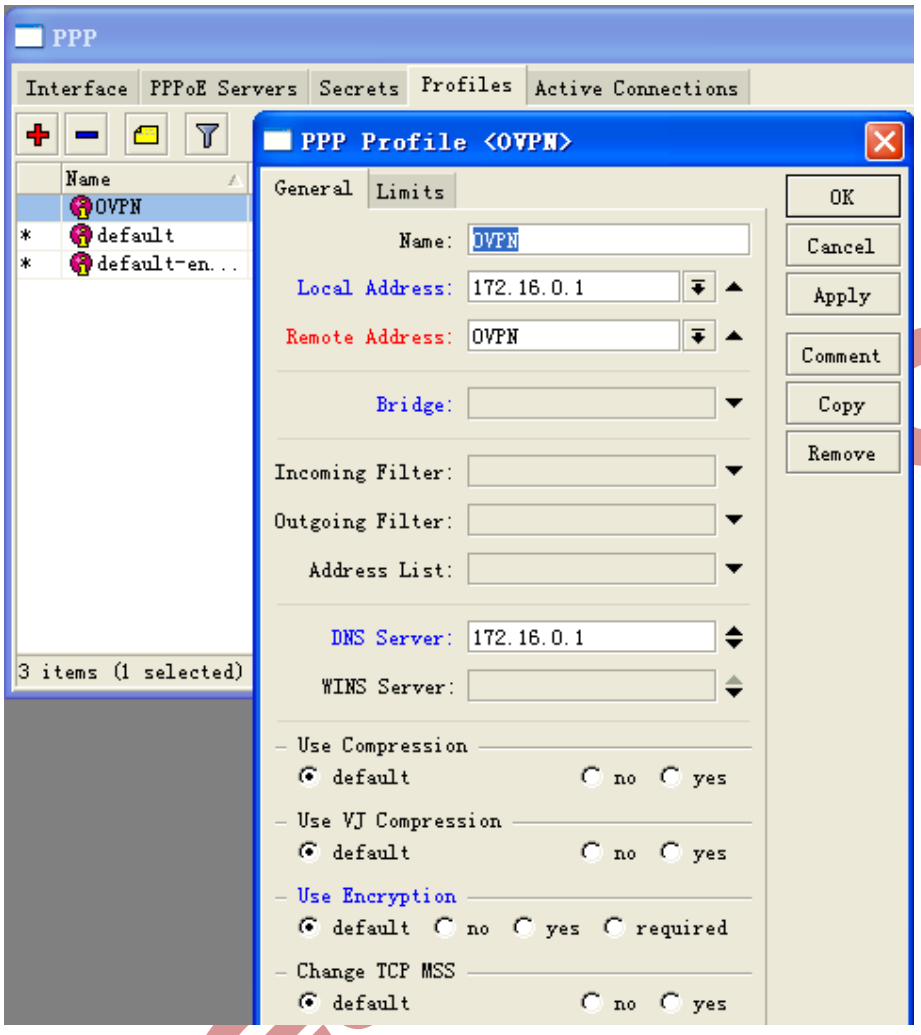
这里选择的 profile 类型完全相同，都为 OVPN 相应，Autentication 认证方式也可以选择相同方式。

这里我们举一个实例，我们建立了一个主机的 OVPN 服务，同时启用 PPTP 和 L2TP 方式，分配远程 IP 为 172.16.0.10-172.16.0.100 的地址池，我用 172.16.0.11 做为 OVPN 隧道的本地 IP。

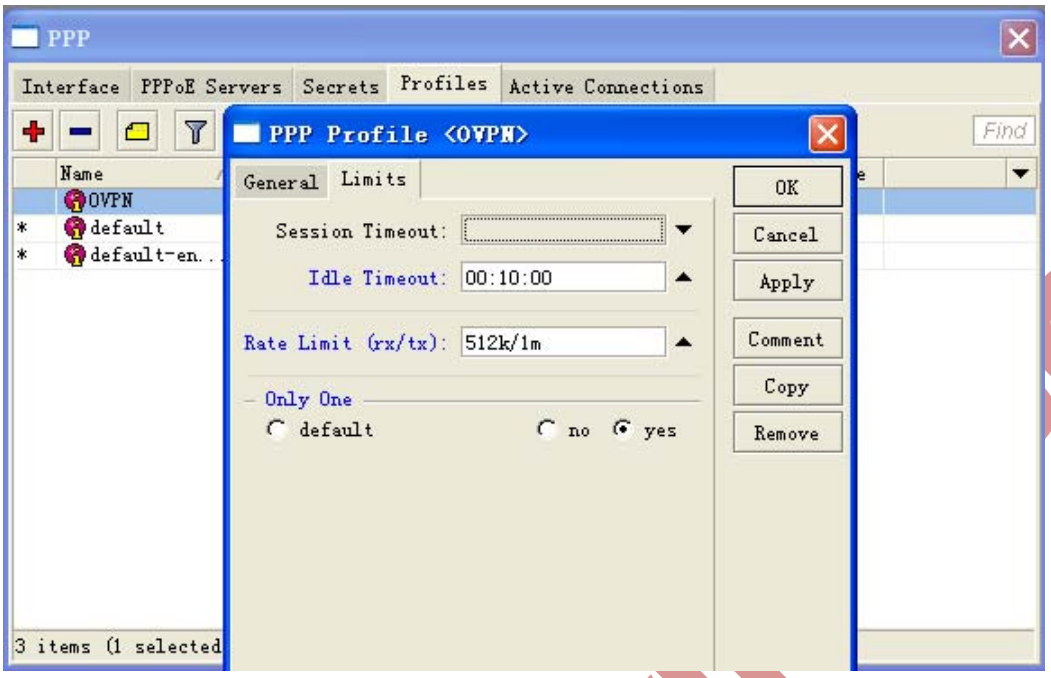
首先我们进入 ip pool 中配置地址池：



配置好地址池后，在 PPP profiles 中添加用户组规则，这里我们添加一个组规则取名 OVPN，配合本地 IP 地址 172.16.0.1 ，在远程 remote-address 中配置之前添加号的地址池 OVPN，设置 DNS 为 172.16.0.1 ，其他配置参数如下：



在limits选项中,配置相应的Idle-timeout(空闲超时时间)、Rate-limit(宽带)和only-one（账号是否唯一性）：



在 PPP secret 中配置用户账号信息，service 参数用于选择，启用服务器的类型，这里我们添加了 edcwifi 和 pptp 两种类型的账号，分别对应 L2TP 和 PPTP 登陆方式，profile 类型选择 OVPN。

配置完成后，我们便可以通过 PPTP 或者 L2TP 连接 RouterOS 的 OVPN 服务，在 Windows 下可以通过 PPTP 的方式直接连接 RouterOS 的 OVPN 服务，但 L2TP 不行，因为 Windows 要求 L2TP 进行 IPsec 的加密方式连接，这里我们可以通过修改 Windows 注册表连接

[键入文字]

L2TP 的 Windows 注册表修改

L2TP 修改注册表，缺省的 Windows XP L2TP 传输策略不允许 L2TP 传输不使用 IPsec 加密。可以通过修改 Windows XP 注册表来禁用缺省的行为，手工修改：

1） 进入 Windows XP 的“开始”“运行”里面输入“Regedt32”，打开“注册表编辑器”，定位“HKEY_Local_Machine\System\CurrentControl Set\Services\RasMan\Parameters”主键。
2） 为该主键添加以下键值：

键值：prohibitipsec
数据类型：reg-dword
值：1

修改后即可通过 Windows 正常连接到 L2TP 服务。

OVPN 的几种应用方式

Mikrotik 的 OVPN 系统支持多种方式的应用，能够实现企业对等访问、企业与多分支点、多点与移动办公和 OVPN 数据转移等多种 OVPN 连接方式。在 OVPN 数据转移方面用处最多的方式是在 VoIP 方面，因为受某些 ISP 网络的限制，使得正常的 VoIP 通讯受到影响，所以可以通过 OVPN 的方式实现数据的转移。

企业对等互访

通过 OVPN 隧道协议如 PPTP、L2TP 或者 IPIP 可以建立一个对等的隧道，使得两个办公室能互相访问公司数据和文件。这种方式我们首先建立 PPP 服务器，并给客户端分配账号和固定 IP 地址、建立 PPP 的拨号和分配 IP 地址，最后设置两个远程局域网的 IP 地址路由（操作可以参考 PPTP 章节）

企业与分支点总分连接

我们可以使用 PPTP 或者 L2TP 等隧道协议，建立多个客户端账号，使多个分公司能连接懂啊总公司的中心服务器，并能通过公司管理各个点的数据和信息互联。这样总公司做到对分公司的有效管理，可以即时发布信息到各个分公司。并能实现数据的安全传输。

企业移动办公与综合应用

RouterOS 支持普通用户的 PPTP 和 L2TP 的 Windows 的拨号连接，所以对于在外出差和家庭办公的用户就可以方便的连接到总部的中心服务器和分支点的文件服务器，特别对需要即时处理问题的公司员工最为合适。这种方式也适用于建立 VoIP 数据的转移和企业内部的 IP 语言通讯。

第二十章 EoIP 隧道

EoIP（Ethernet over IP）隧道是一个建立在两个路由器的 IP 传输层之间的以太网隧道协议，是专属于 Mikrotik RouterOS 协议。EoIP 接口表现的类似以太网传输，当路由器的桥接功能被启用后，所有的以太网数据流量（所有的以太网协议）将被桥接就如同在两个路由器（启用了桥接功能）之间有物理以太网接口和光纤一样。

有 EoIP 接口的网络设置：

可以在因特网上桥接 LAN
可以在加密的隧道桥接 LAN
可以在 802.11b‘ad-hoc’无线网络上桥接 LAN

快速设置向导

在 IP 地址为 10.5.8.1 和 10.1.0.1 的两个路由器之间做 EoIP 隧道：

1. 在 IP 地址为 10.5.8.1 的路由器上添加一个 EoIP 接口并设置它的 MAC 地址：

```
/interface eoip add remote-address=10.1.0.1 tunnel-id=1
mac-address=00-00-5E-80-00-01 disabled=no
```

2. 在 IP 地址为 10.1.0.1 的路由器上添加一个 EoIP 接口并设置它的 MAC 地址：

```
/interface eoip add remote-address=10.5.8.1 tunnel-id=1
mac-address=00-00-5E-80-00-02 disabled=no
```

现在你可以从同一子网添加 IP 地址以创建 EoIP 接口。

规格

功能包要求：system
等级要求：Level 1（limited to 1 tunnel），Level 3
操作路径：/interface eoip

EoIP 接口应该在有 IP 等级连接可能的两个路由器上配置。EoIP 通道可以在 IPIP 隧道,PPTP 128bit 加密隧道，PPPoE 连接或任何传输 IP 的连接上运行。具体属性：

每个上运行隧道接口可以与一个有相同“隧道 ID”的相应接口配置的远程路由器相连接 EoIP 接口就好像接口列表下的所有特征。IP 地址及其他隧道可以在这个接口上运行 EoIP 协议封装以太网帧在 GRE（IP 协议号 47）数据包中，并把它们发送到 EoIP 隧道的远程端

EoIP 隧道的最大计数为 65536

注：WDS 在很大程度上比 EoIP 快（最多达可达到 10-20%，在 touterBOARD 500 系统上），所以推荐在可能时使用 WDS。

EoIP 配置

操作路径：/interface eoip
Arp（disabled | enabled | proxy-arp | reply-only；default：enabled）- 地址解析协议
Mac-address(MAC address)- EoIP 接口的 MAC 地址。你可以自由的使用从 00-00-5E-80-00-00 到 00-00-5E-FF-FF-FF 范围的 MAC 地址
Mtu（整型；default：1500）- 最大传输单元。默认值提供了最大的兼容性
Name（name；default：eoip-tunnelN）- 作为参考的接口名
Remote-address - EoIP 隧道 IP 地址的另一端——必须是 Mikrotik 路由器
Tunnel-id（整型）- a unique tunnel identifier

注：tunnel-id 是一种识别隧道的方法。在同一个路由器上不应该有相同 tunnel-id 的隧道。在参与的两个路由器的 tunnel-id 必须是平等的。

Mtu 必须设置为 1500 以消除隧道内的数据包分段存储（它允许类似以太网络的透明桥接，因此有可能在隧道上传输满长度的以太网帧）。

当桥接 EoIP 隧道时，推荐对每个隧道设置唯一的 MAC 地址以使桥接算法正常工作。对于 EoIP 接口你可以使用从 00-00-5E-80-00-00 到 00-00-5E-FF-FF-FF 范围的 MAC 地址，IANA 就是为这些情况保留的。或者，你可以设置第一字节的第二位来标记地址为由网络管理员指定的本地管理的地址，并使用任何 MAC 地址，你只需要确定它们再连接到一个桥的主机之间是唯一的。

添加并启用名为 to_mt2 连接到 10.5.8.1 路由器的 EoIP 隧道，指定 tunnel-id 为 1：

```
[admin@MikroTik] interface eoip> add name=to_mt2 remote-address=10.5.8.1 \
\... tunnel-id 1
[admin@MikroTik] interface eoip> print
Flags: X - disabled, R - running
 0 X  name="to_mt2" mtu=1500 arp=enabled remote-address=10.5.8.1 tunnel-id=1

[admin@MikroTik] interface eoip> enable 0
[admin@MikroTik] interface eoip> print
Flags: X - disabled, R - running
 0 R  name="to_mt2" mtu=1500 arp=enabled remote-address=10.5.8.1 tunnel-id=1

[admin@MikroTik] interface eoip>
```

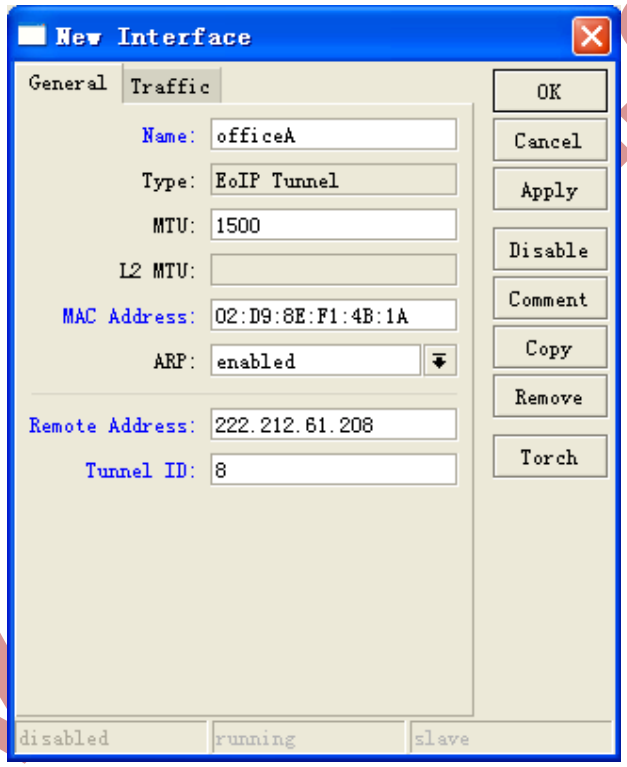
EoIP 应用实例

这里我们假设有两个异地的办公点，OfficeA 和 OfficeB，我们通过 EoIP 隧道将他们连接起来，建立 2 层的安全隧道通信

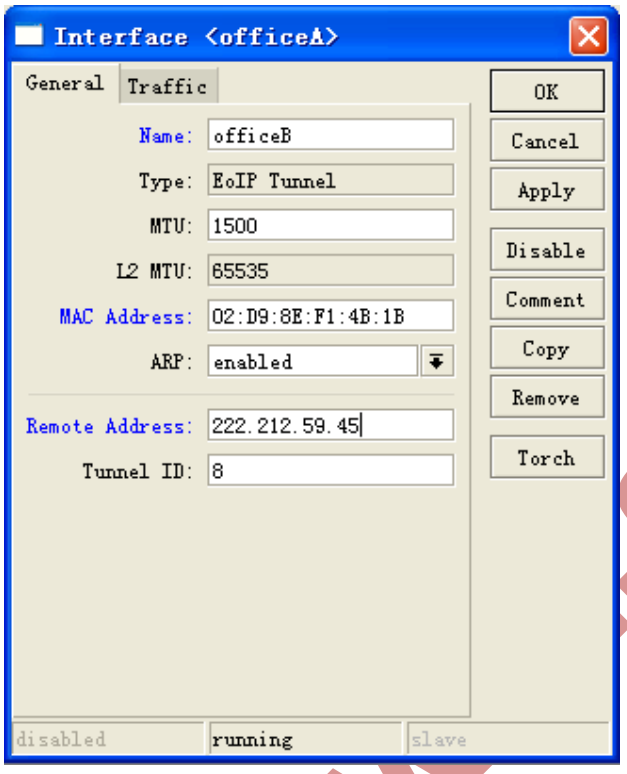
网络参数：

officeA 的 IP 地址为 222.212.61.208，桥接分配地址 10.0.0.1
officeB 的 IP 地址是 222.212.59.45。桥接分配地址 10.0.0.2

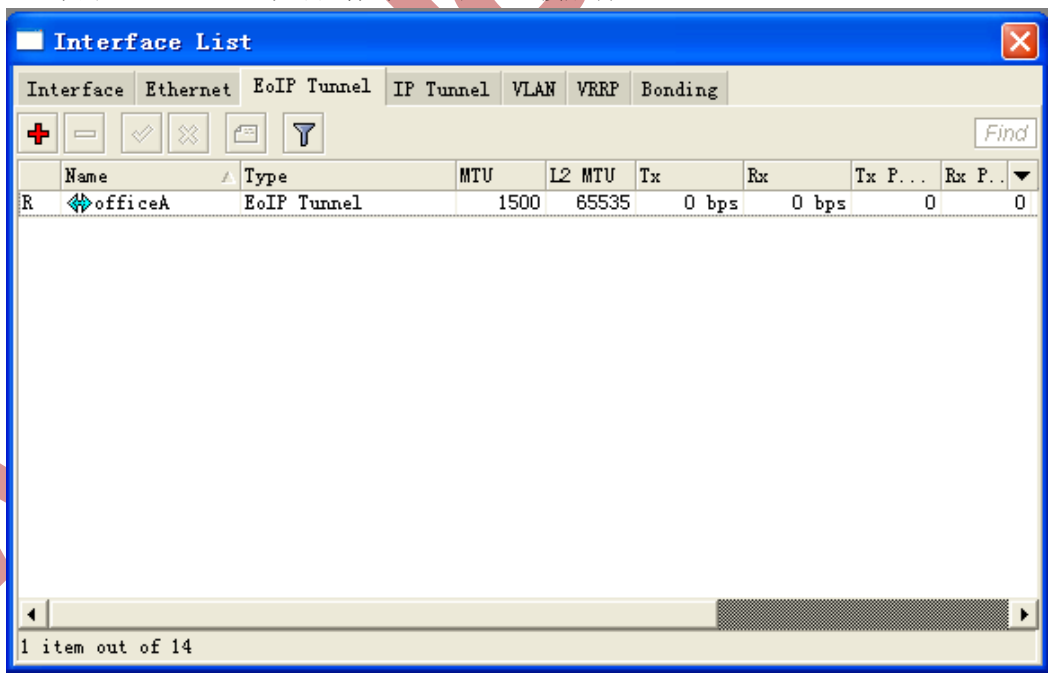
1、 这里我们将配置两个 EoIP 隧道的 tunnel ID 为 8，首先在 interface 里建立 EoIP 隧道



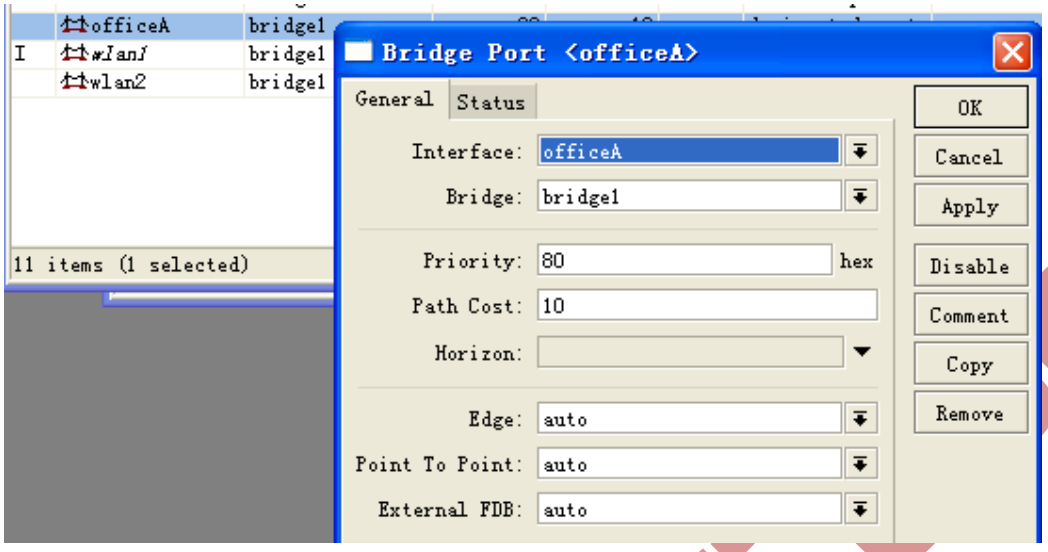
建立 officeB 的 EoIP 隧道：



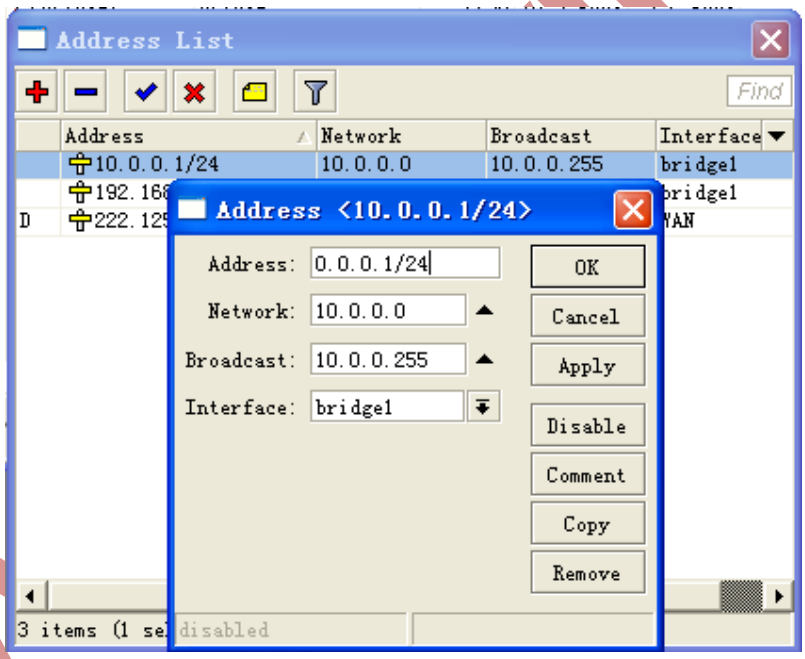
2、 接下来，officeA 和 officeB 的配置基本相同，现在我们来看 officeA 配置，在 interface 中的 EoIP tunnel 中可以看到 EoIP 隧道连接成功：



进入 Bridge，并在 Bridge 中添加一个 Bridge1 的桥，然后并在 Port 中将 ether5 网卡和建立 EoIP 隧道的 officeA 绑定到 Bridge1 中：



3、绑定完桥后，进入 ip address，设置桥 IP 地址为 10.0.0.1/24 ，同样在 officeB 的路由器则设置为 10.0.0.2/24：



注：在做 NAT 规则的时候，特别是伪装，需要指明伪装的端口，如果默然伪装，将会把 EoIP 隧道隐藏，使其二层透穿出现问题，为了避免影响 EoIP 的连接，要选择 out-interface 为 WAN 口。

故障分析

路由器可以相互之间 ping 通但 EoIP 隧道依然不能正常工作！

检查 EoIP 接口的 MAC 地址——它们不应该一样！

第二十一章 Bonding

Bonding 是通过汇聚多个接口到一个虚拟的连接上，这种方式可以获得更高的带宽或提供失效转移接管。
Bonding 操作必须用于二层链路层，不支持三层 IP 层的应用。

让我们假设每个路由器有 2 张网卡（router1 和 router2）并且我们想在两个路由器之间得到最大的传输速率。通过 bonding 配置可以让该设想成为可能。如下配置：

1. 确定你没有 IP 地址在相应的接口，这将被从属到 bonding 接口上！
2. 在 router1 上添加 bonding 接口：

   ```
   [admin@touter1] interface bonding> add slaves=ether1,ether2
   ```

   在 touter2 上添加：

   ```
   [admin@router2] interface bonding> add slaves=ether1, ether2
   ```

3. 添加地址到 bonding 接口上：

   ```
   [admin@router1] ip address> add address=172.16.0.1/24 interface=bonding1
   ```

   ```
   [admin@router2] ip address> add address=172.16.0.2/24 interface=bonding1
   ```

4. 在 router1 上测试连接：

   ```
   [admin@router1] interface bonding> /ip 172.16.0.2
   172.16.0.2 ping timeout
   172.16.0.2 ping timeout
   172.16.0.2 ping timeout
   172.16.0.2 64 byte ping: ttl=64 time=2 ms
   172.16.0.2 64 byte ping: ttl=64 time=2 ms
   ```

注意：note bonding 接口需要几秒钟时间的连通时间。

规格

需要功能包：system
需要等级：Level 1
操作路径：/interface bonding

提供了最佳的失效转移管理，你需要指定 link-monitoring 参数：

MII（媒体独立接口 Media Independent Interface）type1 or type2 - 媒体独立接口是一个在操作系统与 NIC 之间的理论层，探测连接是否运行（执行可以通过其他功能实现，但在我们的事例中这个是非常重要的）。
ARP - 地址解析协议（通过 arp-interval 时间）检测连接状态。

Link-monitoring 被用于检测是否连接。

属性描述

Arp（disabled | enabled | proxy-arp | reply-only；默认：enabled）- 接口的地址解析协议
Disabled - 接口不使用 ARP
Enabled - 接口使用 ARP
Proxy-arp - 接口使用 ARP 代理功能
Reply-only - 接口将只回应/ip arp 的静态 MAC 地址
Arp-interval（time；默认：00:00:00.100）- 通过定义多少毫秒监测 ARP 请求。
Arp-ip-targets（IP address；默认：“”）- IP 目标地址，如果 link-monitoring 被设置 arp 目标 IP 地址将会被监视。你也可以指定多个 IP 地址。
Down-delay（time；默认：00:00:00）- 如果一个连接失效被探测到，bonding 接口通过 down-delay 时间禁用配置。
Lacp-rate（1sec | 30secs；默认：30secs）- 连接聚合控制协议速率是指定多久将 bonding 端的 LACPDUs 进行交换。被用于确定是否连接或进行其他变化。LACP 试着适应这些变化并提供失效管理。
Link-monitoring（arp | mii-type1 | mii-type2 | none；默认：none）- 连接监视是否使用（是否设置启用）
Arp - 使用地址解析协议，探测远程地址是否到达。
mii-type1 - 使用 MII type1 协议确认连接状态。连接状态探测依赖设备驱动。如果 bonding 显示状态为 up，但运行时并未启动，说明该卡可能不支持 bonding 功能。
mii-type2 - 使用 MII type2 探测连接状态（被用于如果 mii-type1 不支持 NIC）
none - 没有任何模式监测，如果一个连接失效，不会被关闭（但没有传输通过）。
Mac-address（只读：MAC address）- bonding 接口的 MAC 地址
mii-interval（time；默认：00:00:00.100）- 多久监测一次连接失效（此参数被用于在 link-monitoring 设置为 mii-type1 或 mii-type2）
mode（802.3ad | active-backup | balance-alb | balance-rr | balance-tlb | balance-xor | broadcast；默认：balance-rr）- 接口绑定模式，如下：
802.3ad - IEEE 802.3ad 动态连接聚合，提供容错和负载均衡。在这个模式下，接口被聚合到一个组里，每个 slave 共享同样的速度。如果你在两个 bonding 路由器之间使用一个交换机，必须确定这个交换机支持 IEEE 802.3ad。active-backup - 提供连接备份。在同一时间仅一个 slave 可以运行。如果一个失效，另外一个 slave 自动连接。
Balance-alb - 自适应负载均衡。该模式包含 balance-tlb，通过接收传输负载均衡。设备驱动应支持设置 MAC 地址，不需要指定的交换机支持
Balance-rr - 轮询负载均衡。在 bonding 接口里 slaves 将依次序的传输和接收。提供负载均衡和容错

Balance-tlb - 输出传输同分布式方式分配负荷到当前的每个 slave 上，传入数据被接收通过当前 slave。如果接收 slave 失败，这时另外一个 slave 带走实效的 MAC 地址。不需要任何特殊的交换机支持
Balance-xor - 为传输使用 XOR 策略。仅提供失效管理，但不支持负载均衡
Broadcast - 同样的数据在所有接口广播一次。这样提供失效容错，但在一些慢的机器上降低了传输吞吐量。
Mtu（整型：68..1500；默认：1500）- 最大传输单元，单位 btyes
Name（name）- 至少 2 个 Ethernet 接口被用于 bonding 接口
Up-delay（time；默认：00:00:00）- 如果一个链路已经连接，bonding 接口被 up-delay 时间禁用，在这个时间过后 bonding 接口启用。

基于两个 EoIP 隧道的 bonding

假设你需要通过 MikroTik 路由器配置以下的网络设置，你有 2 个办公室，并同时接入了相同的 2 个 ISP 线路，你想绑定 2 条线路，得到双倍的贷款速度，并提供失效管理。

两个路由器直接通过 2 个 ISP 连接到 Internet，并配置这两个路由器连接上网。

配置 Office 路由器：

```
[admin@office1] > /interface print
Flags: X - disabled, D - dynamic, R - running
 #    NAME    TYPE   MTU
 0 R isp1     ether  1500
 1 R isp2     ether  1500

[admin@office1] > /ip address print
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS     NETWORK    BROADCAST    INTERFACE
 0 1.1.1.1/24   1.1.1.0    1.1.1.255    isp2
 1  10.1.0.111/24  10.1.0.0   10.1.0.255   isp1
```

配置 Office2 的路由器

```
[admin@office2] interface> print
Flags: X - disabled, D - dynamic, R - running
 #   NAME                TYPE       MTU
 0 R isp2                ether      1500
 1 R isp1                ether      1500
[admin@office2] interface> /ip add print
Flags: X - disabled, D - dynamic, R - running
 #   ADDRESS          NETWORK      BROADCAST      INTERFACE
 0    2.2.2.1/24       2.2.2.0      2.2.2.255      isp2
 1    10.1.0.112/24    10.1.0.0     10.1.0.255     isp1
```

通过 EoIP 隧道连接,实现一个虚拟的二层网络连接,用于 bonding 的连接(由于 bonding 基于二层链路层的链路聚合，所以必须使用 2 层接口）。先配置 Office1 通过 ISP1 连接的 EoIP 隧道：

```
[admin@office1] > interface eoip add remote-address=10.1.0.112 tunnel-id=2 \...
mac-address=FE:FD:00:00:00:04
[admin@office1] > interface eoip print
Flags: X - disabled, R - running
0 R name=" eoip-tunnel2" mtu=1500 mac-address=FE:FD:00:00:00:04 arp=enabled \...
remote-address=10.1.0.112 tunnel-id=2
```

在 Office2 路由器上配置 ISP1 线路的 EoIP

```
[admin@office1] > interface eoip add remote-address=2.2.2.1 tunnel-id=1 \...
mac-address=FE:FD:00:00:00:02
[admin@office1] interface eoip> print
Flags: X - disabled, R - running
0 R name=" eoip-tunnel1" mtu=1500 mac-address=FE:FD:00:00:00:02 arp=enabled \...
remote-address=10.1.0.111 tunnel-id=2
```

在 Office1 路由器上配置 ISP2 的 EoIP 隧道

```
[admin@office1] > interface eoip add remote-address=2.2.2.1 tunnel-id=1 \...
mac-address=FE:FD:00:00:00:03
[admin@office1] > interface eoip> print
Flags: X - disabled, R - running
 0 R name=" eoip-tunnel1" mtu=1500 mac-address=FE:FD:00:00:00:03 arp=enabled
remote-address=2.2.2.1 tunnel-id=1
 1 E name=" eoip-tunnel2" mtu=1500 mac-address=FE:FD:00:00:00:04 arp=enabled
remote-address=10.1.0.112 tunnel-id=2
```

在 Office2 路由器上配置 ISP2 的 EoIP 隧道

```
[admin@office2] > interface eoip add remote-address=1.1.1.1 tunnel-id=1 \...
mac-address=FE:FD:00:00:00:01
[admin@office2] > interface eoip> print
Flags: X - disabled, R - running
 0   R name=" eoip-tunnel2" mtu=1500 mac-address=FE:FD:00:00:00:01 arp=enabled
Remote-address=1.1.1.1 tunnel-id=1

1   R name=" eoip-tunnel2" mtu=1500 mac-address=FE:FD:00:00:00:02 arp=enabled
Remote-address=10.1.0.111 tunnel-id=2
```

设置 bonding，在 Office1

```
[admin@office1] interface bonding> add slaves=eoip-tunnel1,eoip-tunnel2
[admin@office1] interface bonding> print
Flags: X - disabled, R - running
 0 R name=” bonding1” mtu=1500 mac-address=00:0C:42:03:20:E7 arp=enabled
slaves=eoip-tunnel1,eoip-tunnel2 mode=balance-rr primary=none
link-montioring=none arp-interval=00:00:00.100 arp-ip-targets=” ”
mii-interval=00:00:00.100 down-delay=00:00:00 up-delay=00:00:00
lacp-tate=30secs
[admin@office1] ip address> add address=3.3.3.1/24 interface=bonding1
[admin@office1] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
 #    ADDREDD          NETWORK      BROADCAST     INTERFACE
 0    1.1.1.1/24       1.1.1.0      1.1.1.255     isp2
 1    10.1.0.111/24    10.1.0.0     10.1.0.255    isp1
 2    3.3.3.1/24       3.3.3.0      3.3.3.255     bonding1
```

在 Office2 上配置

```
 [admin@office2] interface bonding> add slaves=eoip-tunnel1,eoip-tunnel2
 [admin@office2] interface bonding> print
 Flags: X - disabled, I - invalid, D - dynamic
 #    ADDRESS          NETWORK      BROADCAST     INTERFACE
 0    2.2.2.1/24       2.2.2.0      2.2.2.255     isp2
 1    10.1.0.112/24    10.1.0.0     10.1.0.255    isp1
 2    3.3.3.2/24       3.3.3.0      3.3.3.255     bonding1
 [admin@office2] ip address> /ping 3.3.3.1
          3.3.3.1 64 byte ping: ttl=64 time=2 ms
          3.3.3.2 64 byte ping: ttl=64 time=2 ms
  2    packets transmitted, 2 packets received, 0% packet loss
 round-trip min/avg/max=2/2.0/2 ms
```

第二十二章 VLAN

VLAN 是基于 802.1Q VLAN 协议。它允许你在单个以太网或无线接口上拥有多个虚拟 LAN，给予了高效分离 LAN 的能力。它最多可以支持 4095 个 VLAN 接口，每个以太网的每个接口都有唯一的 VLAN ID。很多路由器，包括 Cisco 或华为，以及很多二层交换机也都支持。

VLAN 是一个允许终端用户如同雾里连接到一个隔离 LAN 一样相互通信的逻辑分组，独立于网络的物理配置。VLAN 支持添加新的安全尺度并对允许当在不相关用户间逻辑地维持分割时共享一个物理网络收取开支。

规格

功能包要求：system
等级要求：Level 1（limited to 1 vlan），Level 3
子目录要求：/interface vlan
标准与技术：VLAN（IEEE 802.1Q）

VLAN 是一个简单的对一套交换机端口进行分组以形成一个逻辑的方法。在一个交换机内这是一个简单的逻辑配置。当 VLAN 延伸到多个交换机时，内部交换机连接就成为主干，在它上面数据包会被标记以指明它们属于哪个 VLAN。

你可以使用 Mikrotik RouterOS（也可以是 Cisco IOS 和 Linux）来标记这些数据包也可以用来接受并路由标记了的包。

由于 VLAN 工作于 OSI 的第二层，它可以作为另一个没有任何显示的网络接口使用。VLAN 成功地通过以太网桥（对 Mikrotik RouterOS 桥你应该设置 forward-protocols 为 ip，arp 以及 other；对其他桥也应该有类似设置）。

你可以在无线连接上传输 VLAN 并把多个 VLAN 接口放在一个无线接口上。注意 VLAN 不是一个全隧道协议（例如，它没有附加域来传输发送者和接收者的 MAC 地址），相同的限制适用于 VLAN 上的桥接也适用于普通的无线接口桥接。换句话说，当无线客户参与放置在无线接口的 VLAN 时，就没有可能使放置在一个无线接口站模式的 VLAN 与其他任何接口进行桥接。

当前支持的以太网接口

这是一个 VLAN 经过测试并能工作的网络接口列表。注意也存在很多其他支持 VLAN 的接口，但它们并没有被检测。

Realtek 8139
Intel PRO/100
Intel PRO1000 server adapter

National semiconductor DP83816 based cards （RouterBOARD200 onboard Ethernet，RouterBOARD 24 card）
National semiconductor DP83815（soekris onboard Ethernet）
VIA VT6105M based cards（RouterBOARD 44 card）
VIA VT6105
VIA VT6102（VIA EPIA onboard Ethernet）

这是一个 VLAN 经过测试并能工作的网络接口列表，但不支持大数据包（>1496 字节）：

3Com 3c59x PCI
DEC 21140（tulip）

VLAN 配置

操作路径：/interface vlan

属性描述

Arp（disabled | enabled | proxy-arp | reply-only；默认：enabled）- 地址解析协议设置
Disabled - 接口不使用 ARP 协议
Enabled - 接口使用 ARP 协议
Proxy-arp - 接口将成为 ARP 代理
Reply-only - 接口将只对源于它本身 IPD地址的请求回应，但邻居 MAC 地址将仅从/ip arp 静态设置表收集。
Interface（名称）- VLAN 网络的物理接口
Mtu（整型；默认：1500）- 最大传输单元
Name（name）- 参考接口名
Vlan-id（整型；默认：1）- 虚拟 LAN 用于区别 VLAN 的标识符或标记。必须在一个 VLAN 中所有电脑是平等的。

注：MTU 必须像在以太网接口那样设置为 1500 字节。但这样也可能不能与一些不支持接收/传输长度带有 VLAN 标题的以太网数据包的以太网卡一起工作（1500 字节数据+4 字节 VLAN 标题+14 字节以太网标题）。这种情况下使用 MTU1496，但要注意如果较长的数据包药在接口发送的话这会引起数据包的分割。同样要记得如果路径 MTU 搜索在源和目的间不能正常工作，MTU1496 可能引起一些问题。

实例：在接口 ether1 添加并启用名为 test 且 vlan-id=1 的 VLAN：

```
[admin@MikroTik] interface vlan> add name=test vlan-id=1 interface=ether1
[admin@MikroTik] interface vlan> print
Flags: X - disabled, R - running
 #    NAME               MTU  ARP      VLAN-ID INTERFACE
 0 X  test               1500 enabled  1       ether1
[admin@MikroTik] interface vlan> enable 0
[admin@MikroTik] interface vlan> print
Flags: X - disabled, R - running
 #    NAME               MTU  ARP      VLAN-ID INTERFACE
 0  R test               1500 enabled  1       ether1
[admin@MikroTik] interface vlan>
```

VLAN 应用事例

我们假设我们有两个或更多连接到 hub 的 Mikrotik RouterOS 路由器。在 VLAN 将被创建的连到物理网络的接口是 ether1（它只是为了例子简单化才需要，不是必须的）。

要通过 VLAN 连接电脑它们就必须物理上连接并且唯一的 IP 地址应该分配给它们以便它们可以互相 ping 通。然后分别在它们创建 VLAN 接口：

```
[admin@MikroTik] interface vlan> add name=test vlan-id=32 interface=ether1
[admin@MikroTik] interface vlan> print
Flags: X - disabled, R - running
 #    NAME               MTU  ARP      VLAN-ID INTERFACE
 0  R test               1500 enabled  32      ether1
[admin@MikroTik] interface vlan>
```

如果接口成功的创建，那么它们都能够运行。如果电脑没有正确的连接（通过不再传输或转发 VLAN 包的网络设备），则两个或者一个接口不能运行。当接口运行时，IP 地址可以分配给 VLAN 接口。

在 router1 上：

```
[admin@MikroTik] ip address> add address=10.10.10.1/24 interface=test
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
 #   ADDRESS            NETWORK         BROADCAST       INTERFACE
 0   10.0.0.204/24      10.0.0.0        10.0.0.255      ether1
 1   10.20.0.1/24       10.20.0.0       10.20.0.255     pc1
 2   10.10.10.1/24      10.10.10.0      10.10.10.255    test
[admin@MikroTik] ip address>
```

在 router2 上：

```
[admin@MikroTik] ip address> add address=10.10.10.2/24 interface=test
[admin@MikroTik] ip address> print
Flags: X - disabled, I - invalid, D - dynamic
  #   ADDRESS            NETWORK         BROADCAST       INTERFACE
  0   10.0.0.201/24      10.0.0.0        10.0.0.255      ether1
  1   10.10.10.2/24      10.10.10.0      10.10.10.255    test
[admin@MikroTik] ip address>
```

如果设置得正确，那么从 router1 可以 ping 通 router2，反之亦然：

```
[admin@MikroTik] ip address> /ping 10.10.10.1
10.10.10.1 64 byte pong: ttl=255 time=3 ms
10.10.10.1 64 byte pong: ttl=255 time=4 ms
10.10.10.1 64 byte pong: ttl=255 time=10 ms
10.10.10.1 64 byte pong: ttl=255 time=5 ms
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 3/10.5/10 ms
[admin@MikroTik] ip address> /ping 10.10.10.2
10.10.10.2 64 byte pong: ttl=255 time=10 ms
10.10.10.2 64 byte pong: ttl=255 time=11 ms
10.10.10.2 64 byte pong: ttl=255 time=10 ms
10.10.10.2 64 byte pong: ttl=255 time=13 ms
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 10/11/13 ms
[admin@MikroTik] ip address>
```

多 VLAN 下的 PPPoE 服务

在大型局域网络中，会建立多个 VLAN track 隧道，而 PPPoE 服务只能运行在一个局域网中，如果有多个 VLAN 网络，这样给个 VLAN 下的客户就是相互独立的，这样就只能在 RouterOS 中多建立几个基于 VLAN 的 PPPoE 服务器，如下图，建立多个 VLAN 后，在 PPPoE-server 中对每个 VLAN 建立一个 PPPoE 服务：

PPP

Interface | PPPoE Servers | Secrets | Profiles | Active Connections

Find

Servic...	Interface	Max MTU	Max MRU	MRRU	Default ...	Authentication
service1	vlan1	1480	1480		default	pap chap msc...
service2	vlan2	1480	1480		default	pap chap msc...
service3	vlan3	1480	1480		default	pap chap msc...
service4	vlan4	1480	1480		default	pap chap msc...
service5	vlan5	1480	1480		default	pap chap msc...
service6	vlan6	1480	1480		default	pap chap msc...

6 items

下面是在 interface 中独立 VLAN 下的 PPPoE 运行情况：

Interface List

Interface | Ethernet | EoIP Tunnel | IP Tunnel | VLAN | VRRP | Bonding

Find

	Name	Type	L2 MTU	Tx	Rx	Tx P...	Rx P...	Tx D...
R	ether1	Ethernet	1526	43.2 kbps	1545 bps	5	2	0
R	vlan1	VLAN	1522	0 bps	0 bps	0	0	0
R	vlan2	VLAN	1522	0 bps	0 bps	0	0	0
R	vlan3	VLAN	1522	0 bps	0 bps	0	0	0
R	vlan4	VLAN	1522	0 bps	0 bps	0	0	0
R	vlan5	VLAN	1522	0 bps	0 bps	0	0	0
R	vlan6	VLAN	1522	0 bps	0 bps	0	0	0
	ether2	Ethernet	1522	0 bps	0 bps	0	0	0
	ether3	Ethernet	1522	0 bps	0 bps	0	0	0
	pppoe-in1	PPPoE Server		0 bps	0 bps	0	0	0
	pppoe-in2	PPPoE Server		0 bps	0 bps	0	0	0
	pppoe-in3	PPPoE Server		0 bps	0 bps	0	0	0
	pppoe-in4	PPPoE Server		0 bps	0 bps	0	0	0
	pppoe-in5	PPPoE Server		0 bps	0 bps	0	0	0

14 items (1 selected)

第二十三章 Web 代理

Mikrotik RouterOS 支持下面的代理服务器功能：

常规 HTTP 代理
透明代理。可以同时透明代理和常规代理
源、目的、URL 及请求方法的访问列表
缓存访问列表（指定哪些对象需要缓存，哪些不需要）
直径访问列表（指定哪些资源应该直接访问，哪些需要通过其他代理服务器）
日志功能

设置 1Gib 的 web 缓存，并通过 8000 端口监听，操作如下：

```
[admin@MikroTik] ip proxy> set enabled=yes port=8000 max-cache-size=1048576
[admin@MikroTik] ip proxy> print
               enabled: yes
           src-address: 0.0.0.0
                  port: 8000
          parent-proxy: 0.0.0.0
     parent-proxy-port: 0
           cache-drive: system
    cache-administrator: "webmaster"
        max-cache-size: 1048576KiB
         cache-on-disk: no
  max-client-connections: 600
  max-server-connections: 600
        max-fresh-time: 3d
   serialize-connections: no
       always-from-cache: no
         cache-hit-dscp: 4
[admin@MikroTik] ip proxy>
```

记住保护你的代理，被未验证的用户所访问，这样会变为一个开放的代理。同样你需要设置目标 NAT 启用透明代理功能：

```
[admin@MikroTik] ip firewall nat> add chain=dstnat protocol=tcp dst-port=80 action=redirect
to-ports=8000
[admin@MikroTik] ip firewall nat>
```

规格

功能包要求：web-proxy
许可等级：Level 3
操作路径：/ip proxy (winbox: ip web-proxy)
技术标准：HTTP/1.0, HTTP/1.1, FTP
硬件需求：需要内存和硬盘空间（具体情况下面的属性）

这个服务履行代理 HTTP 以及 HTTP 代理（对 FTP，HTTP 及 HTTPS 协议）请求。Web 代理通过存储被请求的因特网对象，以起到网页缓存功能的作用，例如，通过在一个网络数据产生的站点更接近接受者的系统上的 HTTP 及 FTP 协议数据的可用数据。这里“更接近”指的是增加的路径可靠度，或速度或者两者都有。Web 浏览器可以使用本地代理缓存来加快访问并减少宽带消耗。

当设置代理服务时，确定它只为你的客户服务，而不是误用为继电器。请阅读访问列表部分的安全注意。

注意保持 web 代理一直运行，即使当你想使用它作为像 HTTP 及 FTP 防火墙（例如，拒绝访问 MP3 文件）或把请求透明的重定向到外部代理时也没有缓存，这样做会是很有用处的。

属性描述

Cache-administrator（text；default：webmaster）- 显示在代理错误页面的管理员 e-mail
Cache-drive（system | name；default：system）- 指定用于存储缓存对象的目标磁盘机。你可以使用控制台完成来查看可用驱动器列表
Cache-only-on-disk（yes | no；default：yes）- 是否在描述磁盘上缓存目录的内存中创建的数据库。这样会减少内存消耗，但会影响速度
Enabled（yes | no；default：no）- 代理服务器是否启用
Max-disk-cache-size（none | unlimited | 整型：0..4294967295；default：none）- 指定最大磁盘缓存大小，以 kb 计算
Max-fresh-time（time；default：3d）- 存储缓存对象的最大时间，一个目标的合法时间一般是由对象本身定义的，但以防太长，你可以覆盖最大值
Maximal-client-connecions（整型；default：1000）- 客户接受的最大连接数（任何更多的连接都将被拒绝）
Maximal-server-connectons（整型；default：1000）- 到服务器的最大连接数（任何更多来自客户的连接都将被挂起直到一些服务器连接结束）
Max-object-size（整型；default：2000Kib）- 大于指定长度的对象将不会保存在磁盘上。以 kb 计算。如果你想获得一个更高的比特命中率，你应该增加该值（一个 2Mib 对象撞击代表 2048 个 1kib 撞击）。如果你更想增加速度而不是节省带宽，你应该把这个值设的低一些
Max-ram-cache-size（none | unlimited | 整型：0..4294967295；default：none）- 指定最大 RAM 缓存大小，以 kb 计算
Parent-proxy（IP address：Port；default：0.0.0.0）- 把所有请求定向到的 IP 地址及其他 HTTP 代理端口（异常会在“direct Access”列表中定义）
0.0.0.0:0 - 没有使用父级代理

Port（Port；default：8080）- 代理服务器将监听的 TCP 端口。这个会在所有想使用该服务器作为 HTTP 代理端口的客户上定义。透明（对客户使用零配置）代理设置可以通过使用目的 NAT 特性在 IP 防火墙重定向 HTTP 请求到该端口完成
Src-address（IP address；default：0.0.0.0）- web 代理将使用这个地址连接父级代理或 web 站点
0.0.0.0 - 合适的 src-address 将会自动从路由列表中取出

注：这个 web 代理监听所有路由器 IP 地址列表中包含的 IP 地址。

在端口 8000 上启用代理：

```
[admin@MikroTik] ip proxy> set enabled=yes port=8000
[admin@MikroTik] ip proxy> print
                  enabled: yes
              src-address: 0.0.0.0
                     port: 8000
             parent-proxy: 0.0.0.0:0
              cache-drive: system
       cache-administrator: "dmitry@mikrotik.com"
       max-disk-cache-size: none
        max-ram-cache-size: 100000KiB
        cache-only-on-disk: yes
 maximal-client-connections: 1000
 maximal-server-connections: 1000
          max-object-size: 2000KiB
           max-fresh-time: 3d
[admin@MikroTik] ip proxy>
```

访问列表

访问列表像普通防火墙规则一样配置。规则从顶到底的处理。第一条匹配的规则制定对连接做何处理。一共有 6 个制定匹配显示的分类器。如果没有指定其中任何一个，那么特定规则将与每一条连接进行匹配。

如果连接被一条规则匹配，该规则的 action 属性就指定是否连接应被允许。如果特定连接没有匹配任何规则，那么它将被允许。

属性描述

Action（allow | deny；default：allow）- 指定通过或拒绝已匹配的包
Dst-adress（IP address/netmask）- IP 包的目的地址

Dst-host（wildcard）- IP 地址或用于连接目标服务器的 DNS 名（这是一个在指定端口与到特定网址路径之前写在他的浏览器的字符串）
Dst-port（Port{1,10}）- 包到达的列表或端口范围
Hits（只读：整型）- 被规则修正的请求数
Local-port（Port）- 指定包接收的 web 代理端口。这个值应该匹配 web 代理监听的其中一个端口
Method（any | connect | delete | get | head | options | post | put | trace）- 用于请求的 HTTP 方法（参见本文档最后面的 HTTP 方法部分）
Path（wildcard）- 在目标服务器中的被请求页面名（例如，特定网页的名称或不含它存在的服务器名称的文档）
Redirect-to（text）- 以防访问被该规则拒绝，用户应被重定向到这里指定的 URL
Src-address（IP address/netmask）- IP 包的源地址

注：统配符属性（dst-host 和 dst-path）匹配一个完整的字符串（例如，如果设置为“example”，则他们不会匹配“example.com”）。可用的统配符‘*’（匹配任何数量的任何字符）以及‘？’（匹配任何一个字符）。这里也接受常规表达，但是如果属性被当作常规表达处理，那就应该以冒号（‘：’）开始。

在常规表达式中的低命中：

\\ 符号顺序用于在控制台中输入\字符
\. 样式仅表示 .（在常规表达式中单独一个点表示任何符号）
表示在给定样式之前不允许任何符号，我们在样式的开头使用^符号
指定在给定样式之后不允许任何符号，我们在样式的结尾使用$
输入[or]符号，你可以用反斜杠对它们转义

强烈建议拒绝所有 IP 地址除了在路由器之后的那些，因为代理仍然可以访问你的 internal-use-only（企业网）web 服务器。在 firewall manual 中查询如果保护你的路由器。

直接访问列表

操作路径：/ip proxy direct

如果指定了 parent-proxy 属性，就很可能告诉代理服务器是否尝试通过请求到父级代理或通过直接连接到被请求的服务器以解决问题。直接访问列表就像前一章节描述的 dialing 访问列表一样管理，除了 action 参数。

属性描述

Action（allow | deny；default：allow）- 指定对已匹配包的动作
Allow - 总是直接绕过父级路由器解决匹配的请求
Deny - 通过父级代理以解决匹配请求。如果没有指定则这个与 allow 的效果相同
Dst-address（IP address/netmask）- IP 包的目的地址

Dst-host（wildcard）- 用于连接到目标服务器的 IP 地址或 DNS 名（这是在指定特定网页到达的端口与路径之前用户写在他的浏览器中的字符串）
Dst-port（Port{1,10}）- 包到达的列表或端口范围
Hits（只读：整型）- 被规则修正过的请求数
Local-port（Port）- 指定包接受的 web 服务器端口。这个值应该与 web 代理监听的其中一个匹配
Method（any | connect | get | head | options | post | put | trace）- 用于请求中的 HTTP 方法（参见本文档最后的 HTTP 方法部分）
Path（wildcard）- 目标服务器中的被请求页面名（例如，特定 web 页面名或不含它存在的服务器名的文档）
Src-address（IP address/netmask）- IP 包的源地址

注：不像访问列表，直接代理访问列表有与 deny 相等价的默认动作。当没有规则指定或一个特定请求没有匹配任何规则时发生。

缓存管理

操作路径：/ip web-proxy cache

缓存访问列表指定哪个请求（域、服务器、页面）应该由 web 代理本地缓存，而哪个不用。这个列表与 web 代理访问列表完全一样地执行。

属性描述

Action（allow | deny；default：allow）- 指定对已匹配包的动作
 Allow - 从匹配的请求缓存对象
 Deny - 不从匹配的请求缓存对象
Dst-address（IP address/netmask）- IP 包的目的地址
Dst-host（wildcard）- 用于连接到目标服务器的 IP 地址或 DNS 名（这是在指定特定网页到达的端口与路径之前用户写在他的浏览器中的字符串）
Dst-port（Port{1,10}）- 包到达的列表或端口范围
Hits（只读：整型）- 被规则修正过的请求数
Local-port（Port）- 指定包接收的 web 服务器端口。这个值应该与 web 代理监听的其中一个匹配
Method（any | connect | delete | get | head | options | post | put | trace）- 用于请求中的 HTTP 方法（参见本文档最后的 HTTP 方法部分）
Path（wildcard）- 目标服务器中的被请求页面名（例如，特定 web 页面名或不含它存在的服务器名的文档）
Src-address（IP address/netmask）- IP 包的源地址

代理监视

命令名：/ip proxy monitor

这个命令显示代理服务器的一些状态

属性描述

Cache-used（只读：整型）- 用于缓存的磁盘空间
Hits（只读：整型）- 在缓存中找到并开始被服务的请求数
Hits-sent-to-clients（只读：整型）- 由缓存服务的数据量
Ram-cache-used（只读：整型）- 用于存储缓存的 RAM 空间
Received-from-servers（只读：整型）- 从其他服务器接收的数据量
Requests（只读：整型）- 已处理的请求量
Sent-to-clients（只读：整型）- 发送到该代理服务器客户的数据量
Status（只读：text；default：stopped）- 显示代理服务器的状态信息
Stopped - 代理被禁用且没有运行
Rebuilding-cache - 代理被启用并运行，存在的缓存被核实
Running - 代理被启用并运行
Stopping - 代理关闭（最大 10s）
Clearing-cache - 代理停止，缓存文件被删除
Creating-cache - 代理停止，缓存目录结构被创建
Dns-missing - 代理被启用，但没有运行因为未知的 DNS 服务器（你应该改变地址或端口）
Invalid-cache-administrator - 代理被启用，但没有运行因为非法的缓存管理员的 e-mail 地址
Invalid-hostname - 代理被启用，但没有运行因为非法的主机名（你应该设置一个合法的主机名）
Error-logged - 主机没有运行因为未知的错误。这个错误会被日志标记为系统错误。请发把错误、描述以及如何发生的送给我们
Reserved-for-cache（整型）- 最大缓存大小，可以访问 web 代理
Total-ram-used（只读：整型）- 用于代理的总 RAM 大小
Uptime（只读：time）- 代理最近一次启动后的时间

连接列表

操作路径：/ip proxy connections

这个目录包含代理存储的当前连接的列表。

属性描述

Dst-address（只读：IP address）- 连接的 IP 地址
Protocol（只读：text）- 协议名
Rx-bytes（只读：整型）- 客户接收的字节量
Src-address（只读：IP address）- 连接源发站的 IP 地址
State(只读:closing | connecting | converting | Hotspot | idle | resolving | rx-header | tx-body | tx-eof | tx-header | waiting |)- 打开连接的状态
Closing - 数据传输完成，连接正在最终完成

Connecting - 建立 toe 连接
Hotspot - 检查是否 Hotspot 认证允许继续（对 Hotspot 代理）
Idle - 闲置状态
Resolving - 分辨服务器的 DNS 名
Rx-header - 接受 HTTP 标题
Tx-body - 传输 HTTP 正文给客户
Tx-eof - 写祖块端（当转换为分组的回应）
Tx-header - 传输 HTTP 标题给客户
Waiting - 等待来自同等体的传输
Tx-bytes（只读：整型）- 由客户发送的字节数

缓存插页

操作路径：/ip proxy inserts

这个目录显示存储在缓存中的对象的统计数据（缓存插页）

属性描述

Denied（只读：整型）- 被缓存列表拒绝的插页数
Errors（只读：整型）- 磁盘或其他系统相关的错误数量
No-memory（只读：整型）- 由于没有足够内存而没有存储的对象数量
Successes（只读：整型）- 成功缓存插页的数量
Too-large（只读：整型）- 过大而不能存储的对象数量

缓存查检

操作路径：/ip proxy lookups

这个目录相识从缓存读取的对象的统计数据（缓存查检）

属性描述

Denied（只读：整型）- 被访问列表拒绝的请求数
Expired（只读：整型）- 在缓存中发现的过期请求数，这样将会被外部服务器请求
No-expiration-info（只读：整型）- 接收没有能与请求相比较的信息的页面的有条件请求
Non-chacheable（只读：整型）- 来自外部服务器的无条件请求数（由于它们的缓存被缓存访问列表拒绝了）
Not-found（只读：整型）- 没有在缓存中发现的请求数，这样将被一个外部服务器请求（或者是父级代理，如果进行过相应的配置）
Successes（只读：整型）- 在缓存中发现的请求数

补充工具

操作路径：/ip proxy

Web 代理有附加的命令来处理用于缓存目的的非系统驱动器和从严重的文件系统错误中恢复代理。

Check-drive - 检查非系统缓存驱动器的错误
Clear-cache - 删除存在缓存并建立新缓存目录
Format-drive - 格式化非系统缓存驱动器并为容纳缓存做准备

HTTP 方式

OPTIONS

这个方法是一个关于客户与 request-URI 定义的服务器之间的链上的可用通信项信息的请求。这个方法允许客户决定选项以及（或）与没有初始化任何资源检索的资源相关的需求。

GET

这个方法检索 Request-URI 定义的任何星系。如果 Request-URI 设计数据处理过程然后 GET 方法的回应应该包含处理产生的数据而不是处理过程的源代码，除非源是这个处理的结果。

如果请求信息包含 Range 标题字段，那么 GET 方法就会成为一个部分 GET。这个部分 GET 方法通过只请求不传送已被客户包含的数据的尸体的一部分来减少不必要的网路使用。

当却仅当达到 HTTP 缓存要求时一个 GET 请求的回应为可缓存的。

HEAD

这个方法共享 GET 方法的所有特征除了服务器不一定必须在回应中返回一个信息体。它检索蕴含在导致广泛应用于测试合法的超文本连接，可访问性，及最近修改的请求中的实体的元信息。

HEAD 请求的回应可能以这样的途径仍为可缓存的：包含于响应的信息可能用于更新先前缓存的被 Request-URI 识别的实体。

POST

这个方法需要起源服务器接受包含在请求中的实体，就像 Request-URI 定义的新的下级资源一样。

POST 方法执行的实际动作由起源服务器判定，并且通常是依赖 Request-URI 的

POST 方法的回应不可缓存，除非回应包含合适的 cache-control 或 expires 标题字段。

PUT

这个方法需要被包含的实体存储在提供的 Request-URI 中。如果另一个实体存在于指定的 Request-URI 中，被包含的实体应该被认为是存在于起源服务器上的新版本。如果 request-URI 没有指向一个存在的资源，那么起源服务器应该创建一个有 URI 的资源。

如果请求通过了一个缓存并且 Request-URI 识别了一个或多个当前缓存的实体，那么这写实体就应作为过时的处理。这个方法的回应不可缓存。

TRACE

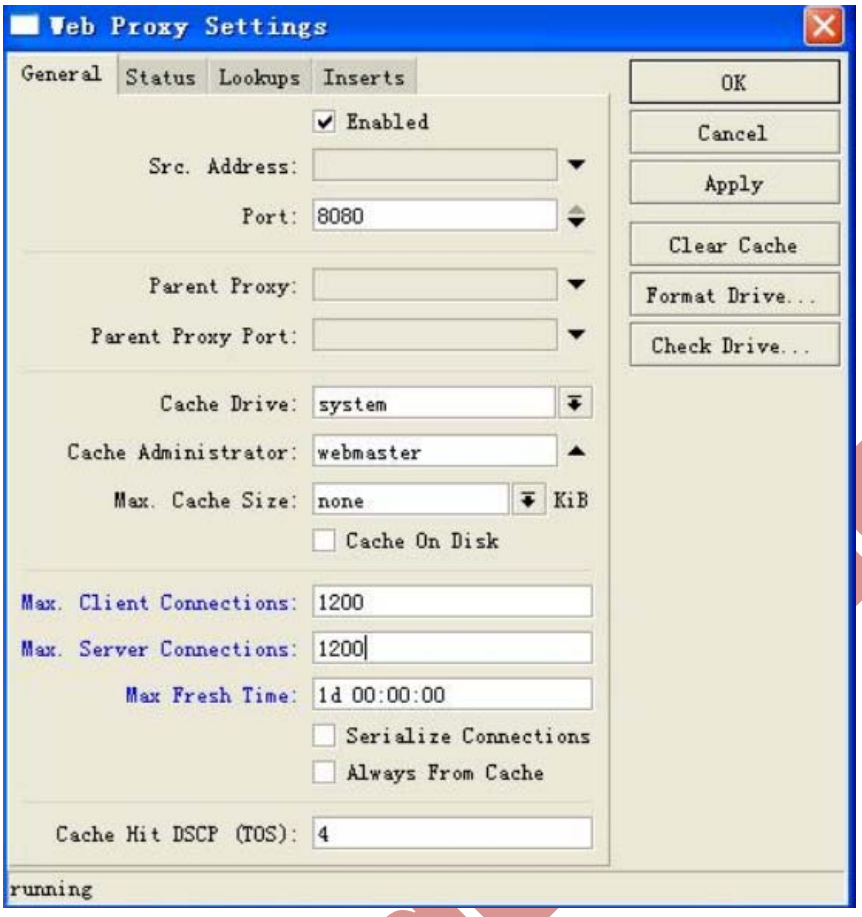
这个方法调用一个远程的，请求信息的应用层循环。最终的请求接收者应该把接收到的信息作为一个 200（ok）回应的实体正文反射给客户。最终接收者是起源服务器或者第一个代理或者在请求中接收 0 值的 Max-Forwards 的网关。一个 TRACE 请求不一定要包含一个实体。

Web 代理应用事例

通过使用 web-proxy 禁止网站和禁止下载

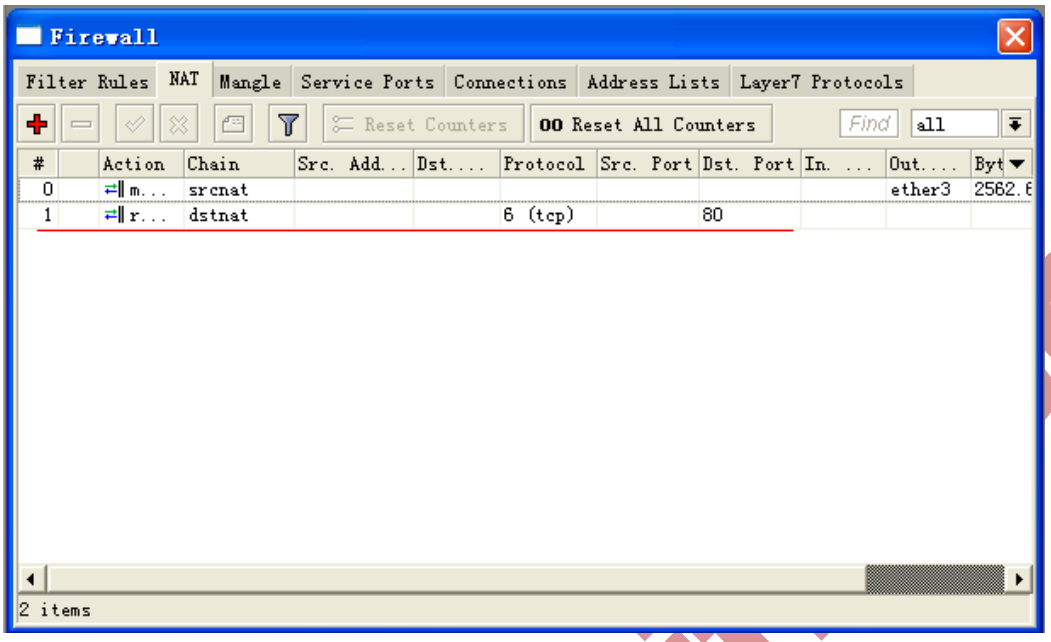
首先配置 web-proxy，配置参数如下：

```
[admin@MikroTik] /ip proxy> prin
              enabled: yes
          src-address: 0.0.0.0
                 port: 8080
         parent-proxy: 0.0.0.0
    parent-proxy-port: 0
          cache-drive: system
   cache-administrator: "webmaster"
       max-cache-size: none
        cache-on-disk: no
max-client-connections: 1200
max-server-connections: 1200
       max-fresh-time: 1d
 serialize-connections: no
    always-from-cache: no
       cache-hit-dscp: 4
```



现在，设置透明传输数据重定向，将所有访问 80 端口的数据重定向到 Web-proxy 的 8080 端口上：

```
/ip firewall nat
chain=dstnat protocol=tcp dst-port=80 action=redirect to-ports=8080
```



确定你路由器本地的 proxy 没有打开代理，并禁止外网通过路由器代理上网：

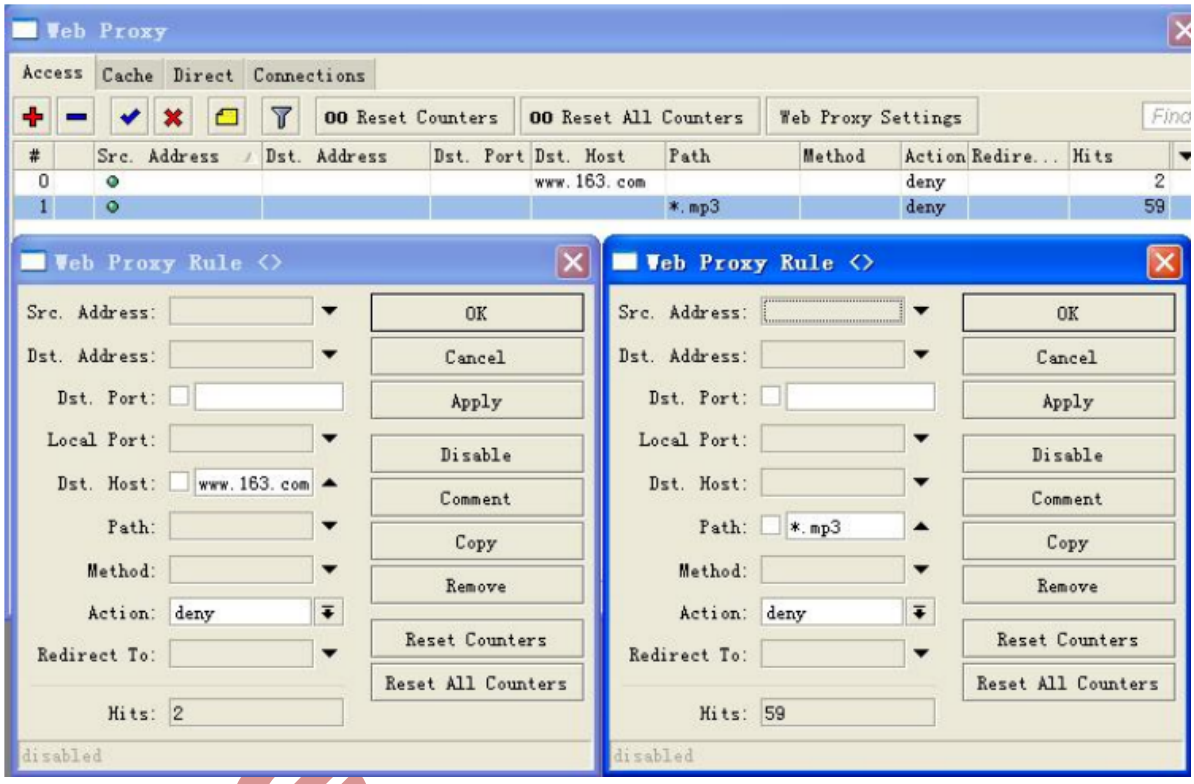
```
/ip firewall filter
chain=input in-interface=<Your WAN Port> src-address=0.0.0.0/0 protocol=tcp dst-port=8080
action=drop
```

设置禁止访问网站，该设置将禁止访问 http://www.163.com

```
/ip proxy access
dst-host=www.163.com action=deny
```

我们可用阻止文件如：“.mp3, .exe, .dat, .avi”的下载。

```
/ip proxy access
path=*.exe action=deny
path=*.mp3 action=deny
path=*.zip action=deny
path=*.rar action=deny.
```



同样我可用阻止所有含“mail”的关键字链接

```
/ip proxy access
dst-host=:mail action=deny
```

第二十四章 MetaRouter

RouterOS 现在支持 2 种不同的虚拟化技术分别是：metarouter 和 Xen

Metarouter

Metarouter 是 Mikrotik 开发的，当前仅支持 RouterBOARD 400 系列（mips-be），也只能创建 RouterOS 的虚拟机。Mikrotik 计划添加更多的功能懂啊 metarouter 中，因此新的硬件支持将会添加到 metarouter 中，甚至会超过 Xen 的功能。

Xen

Xen 是基于 Linux Xen 虚拟机项目，应用于当前的 RouterOS x86 系统（PC），Xen 虚拟机能创建不同的操作系统。

虚拟机技术的应用

下面是一些虚拟机的可行方案（一些方案现在只指出 Xen，但 metarouter 将会添加更多的功能）：

数据管理中心

加强了一些路由器的硬件平台
加强路由器的服务，并更高等级的服务器如 VOIP 交换在同一台设备上
使用客户机上的一个路由器为定制功能，例如日志记录、LDAP 或者传统网络
冗余路由器更加简单和便宜

托管中心

通过 RouterOS 的虚拟化技术配合各种网络功能，如各种服务 mail、HTTP、FTP 等
提供虚拟机路由器的 VPN 解决方案，这样能使网络管理员拥有自己的路由器，在高速骨干网络建立各种隧道或者 VPN 访问系统

无线 ISP 客户端

设置两个独立的路由器，并设置 WISP 的无线控制，并交由以太网端的客户进行控制

多客户端（例如办公楼）

分布在多个点的客户从一个骨干以太网连接（有线或无线），让每个客户能控制自己的独立的虚拟路由器，并配置自己的办公的路由。

网络规划与测试

建立一个虚拟的网络在一台设备上，相同环境的可对一个网络测试计划配置，进行微调，起到在实验室的作用，而不需要在外假设，通过脚本和 the dude 网络管理器模拟和检测网络

Metarouter 介绍

Metarouter 是 RouterOS 从 4.0beta1 和 3.21 版本开始新增加的功能，当前 metarouter 只能用于 RB400 系列，用于创建虚拟机，在以后会有更多硬件平台增加此功能。每一个 metarouter 是使用设备相同的资源，建立独立的 RouterOS 系统。每一个 metarouter 至少需要 16M 的 RAM。16M 是绝对最小的值，建立为每一个 metarouter 使用更大的 RAM。

当前可以创建 8 个 metarouter 虚拟机，将来新的版本会增加到 16 个。在主设备上，你可以创建 8 个虚拟机接口连接到 metarouter，唯一可以增加接口的方式只能通过 VLAN。现在 metarouter 虚拟机还不能支持外部存储设备。

Metarouter 功能常用于允许客户或者低特权用户访问自己的“路由”，并根据他们需要自己配置参数，这样不需要另外一个真实的路由器。例如：一个 ISP 能创建一个虚拟路由器，允许客户特定的用户通过以太接口访问，并定义他们自己的防火墙规则，但只有又不会影响主设备的运行

在/metarouter 目录下给出了一下命令：

Add - 允许你创建一个新的虚拟路由器
Print - 通过列表显示当前所有虚拟路由器
Enable - 启用一个虚拟路由器
Disable - 禁用一个虚拟路由器
Console - 访问一个虚拟路由器的控制台
Interface - 映射相应的网络接口

创建一个 metarouter

```
[admin@edcwifi] /matarouter> add name=mr0 memory-size=32 disk-size=3200
disabled=no
[admin@edcwifi] /metarouter> print
Flags: X - disabled
#   NAME        MEMORY-SIZE   DISK-SIZE    USED-DISK    STATE
0   mr0         16Mib         0kib         377kib       running
```

Name：虚拟路由器的名称
Memory-size：分配给虚拟路由器的 RAM 大小
Disk-size：HDD 的容量，通过 KB 分配给虚拟路由器（如果设置为 0，容量默认为动态分配）*

Used-disk：当前使用的硬盘空间 currently used disk space
State：metarouter 运行的状态

注意：metarouter 在使用的动态 HDD 空间时，启用代理功能会占用你所有的 HDD 存储！

默认配置

如果你添加一个新的 metarouter 没有指定任何参数，默认会添加动态的 HDD 长度，和 16M 的 RAM：

```
[admin@edcwifi] /metarouter> add name=me1
[admin@edcwifi] /metarouter> print
Flags: X - disabled
 #   NAME         MEMORY-SIZE     DISK-SIZE     USED-DISK   STATE
 1   mr1          16MiB           0KiB          3KiB        running
```

添加接口

首先需要添加一个新的接口懂啊你的虚拟路由器上，这个操作在 interface 目录完成，interface 命令如下面：

```
[admin@edcwifi] /metarouter> interface add
Comment disabled dynamic-mac-address type virtual-machine copy-from
dynamic-bridge static-interface vm-mac-address
```

我们添加一个接口：

```
 [admin@edcwifi] /metarouter> interface add virtual-machine=mr1 type=dynamic
```

在物理路由器的 interface 出现一个虚拟接口：

```
 [admin@edcwifi] > /interface print
 Flags: D - dynamic, X - disabled, R - running, S - slave
  #     NAME                TYPE       MTU
  8  R  ether9              ether      1500
  9  R  test                bridge     1500
  10 DR vif1                vif        1500
```

连接虚拟机

连接你的虚拟机，使用 console 命令：

```
 /metarouter console 0
```

你可以看到你最新添加的虚拟接口：

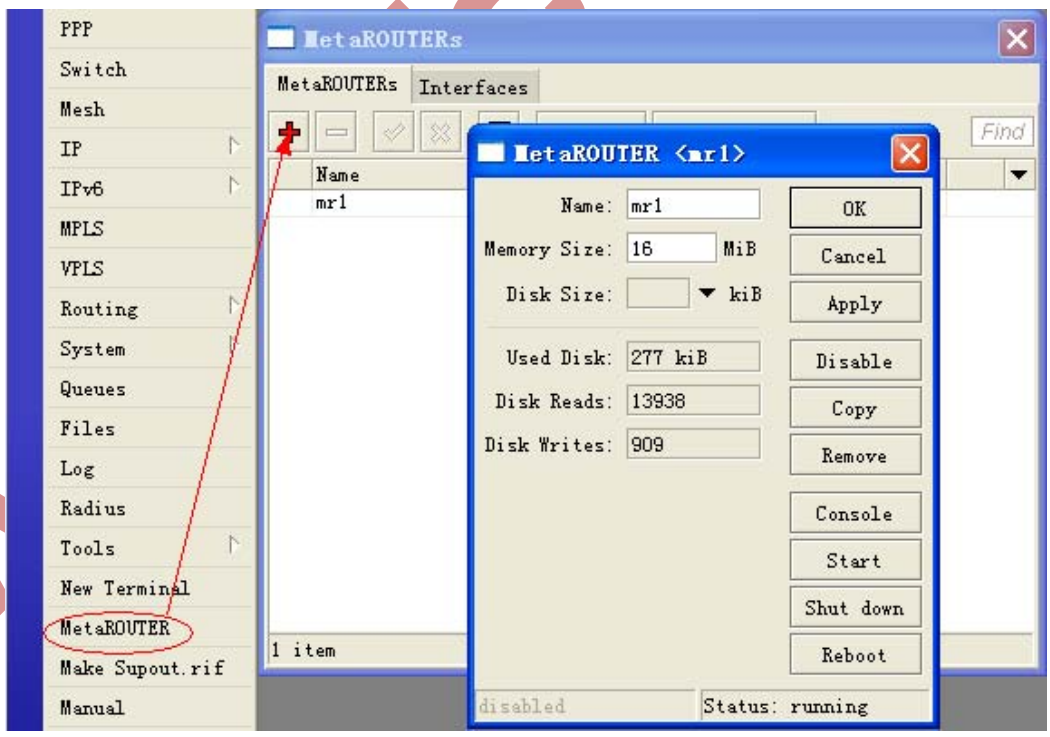
```
[admin@edcwifi] > interface print
Flags: D - dynamic, X - disabled, R - running, S - slave
 #     NAME                TYPE      MTU
 0  R  ether1              ether     1500
```

从 metarouter 的虚拟机控制台断开，按 CTRL+A 和 Q 退回到物理路由器：

```
[admin@edcwifi] >
[Q - quit connection]    [B - send break]
[A - send ctr-A prefix]  [R - autoconfigure rate]

Q

Welcome back!
```

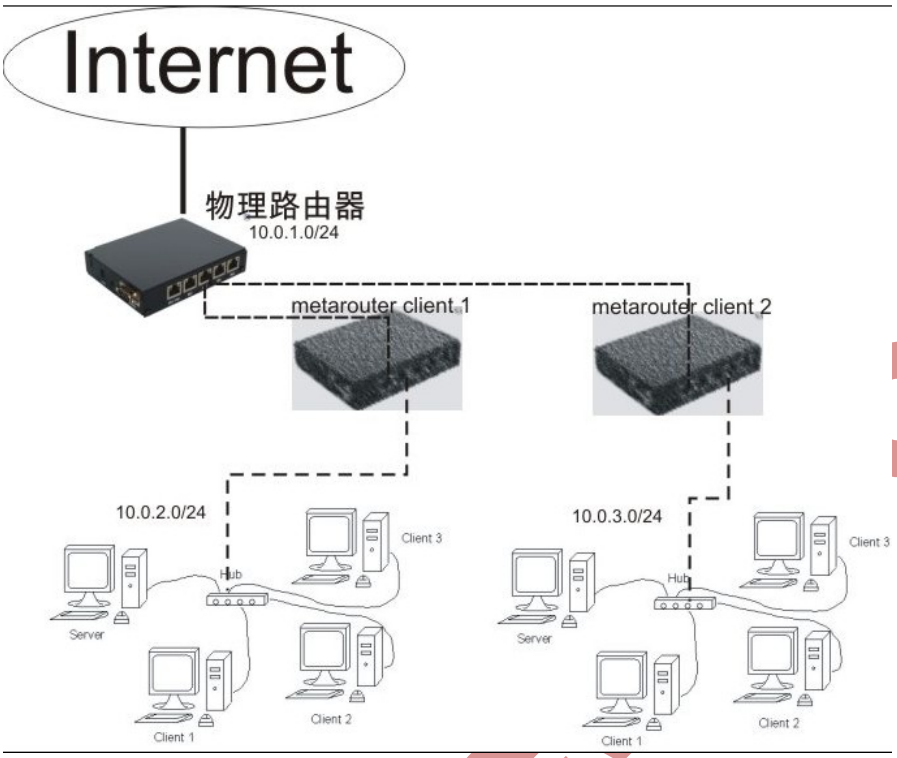
Metarouter 事例

现在你看到之前添加的虚拟接口在物理路由器的 interface 目录中显示为 vif1，当然在 metarouter 的接口中显示为 ether1，你可以在 2 个接口上配置 IP 地址，并连接网络。创建一个 Bridge 在允许传输的物理接口和虚拟接口上。下面是 winbox 操作界面



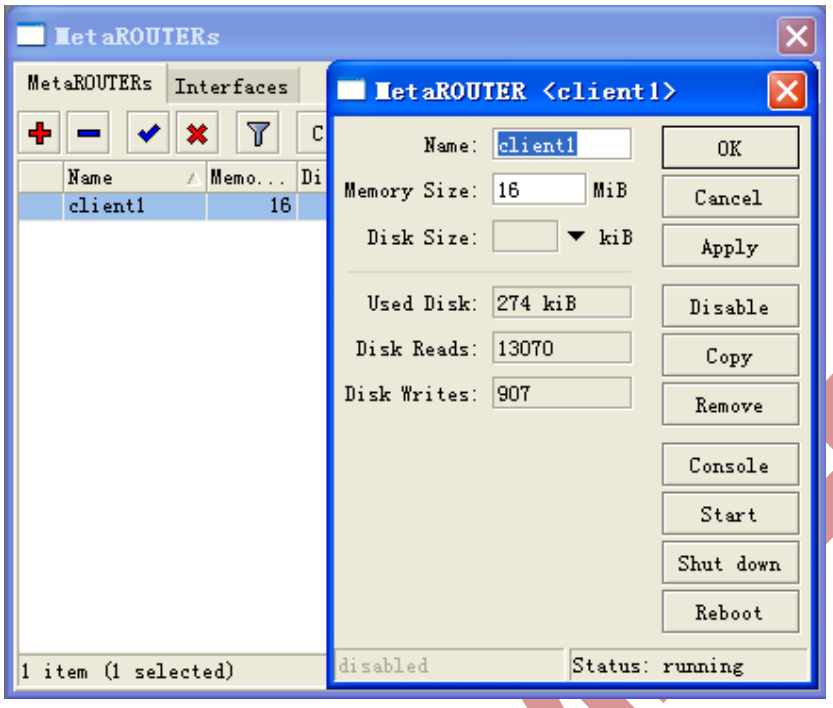
这个事例将介绍如何配置 metarouter 功能，为局域网内部独立 RouterOS 虚拟路由器，基于 RB450 配置 metarouter，我们将建立两个 client 虚拟路由，并管理两个不同的局域网。目

的是让两个局域网的管理员可以管理自己的路由器（metarouter），并根据自己的需要配置他们自己的防火墙、流量控制和 nat 规则，当然他们不能连接物理路由器（没有分配权限）：



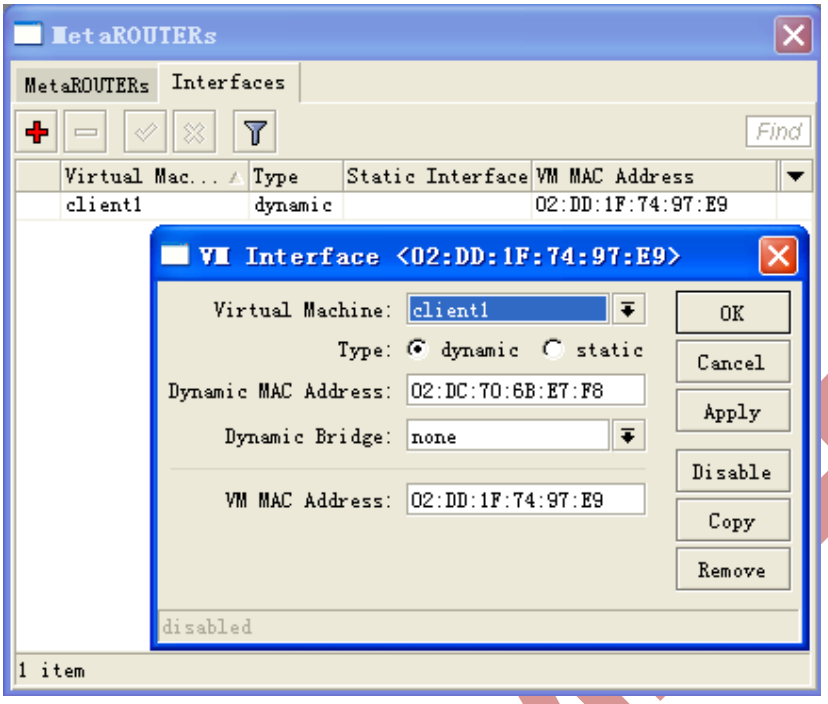
1. 给客户添加一个 metarouter：

```
[admin@EDCwifi] /matarouter> add name=client1 memory-size=16
[admin@EDCwifi] /metarouter> print
Flags: X - disabled
 #   NAME            MEMORY-SIZE   DISK-SIZE    USED-DISK     STATE
 0   client1         16MiB         0KiB         221KiB        running
[admin@EDCwifi] /metarouter>
```



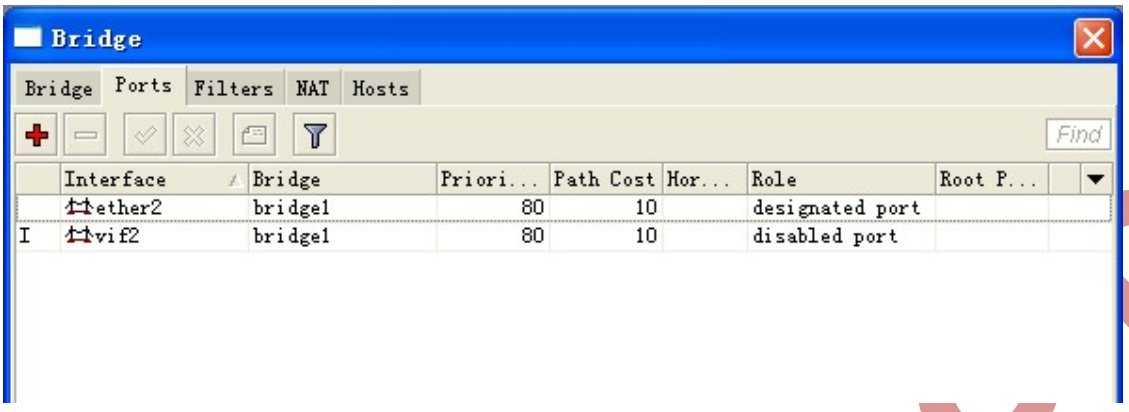
2. 添加 metarouter 虚拟机的接口：

```
[admin@EDCwifi] /metarouter interface> add virtual-machine=client1
[admin@EDCwifi] /metarouter interface> add virtual-machine=client1
[admin@EDCwifi] /metarouter interface> print
Flags: X - disabled, A - active
 #   VIRTUAL-MACHINE                TYPE       VM-MAC-ADDRESS
 0 A client1                        dynamic    02:DC:70:6B:E7:F8
 1 A client1                        dynamic    02:DD:1F:74:97:E9
[admin@EDCwifi] /metarouter interface>
```



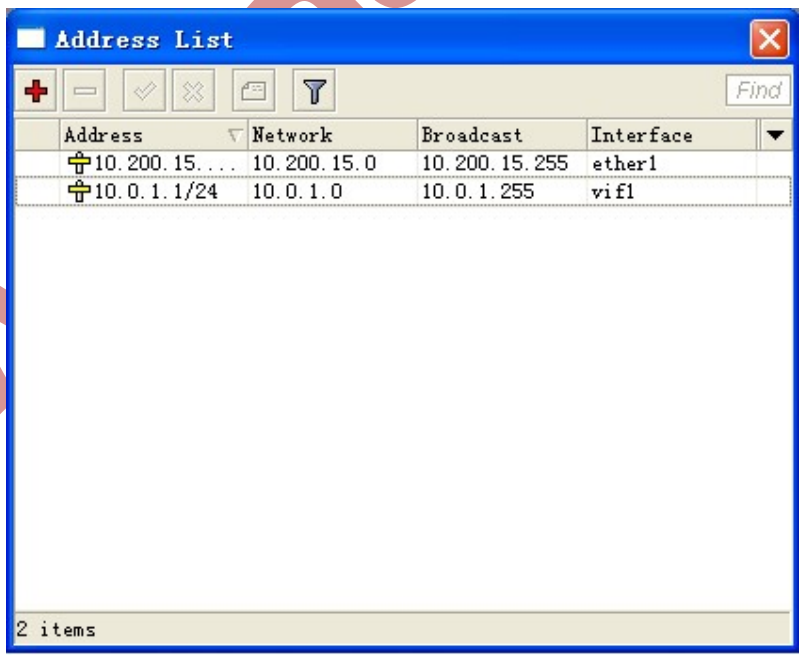
3. 创建一个桥接口，将 metarouter 接口与以太网接口桥接，这里我们将 vif2 的虚拟接口放入内网的桥中，用于客户端通过物理接口连接（使用桥接目的是将虚拟路由的内网接口，通过桥接透穿到真实的网络中）：

```
[admin@EDCwifi] /interface bridge> add
[admin@EDCwifi] /interfacee bridge> print
Flags: X - disabled, R - running
 0 R name=”bridge1” arp=enabled mac-adress=00:00:00:00:00:00
protocol-mode=none priority=0x8000 auto-mac=yes admin-mac=00:00:00:00:00:00
max-message-age=20s forward-delay=15s transmit-hold-count=6 ageing-time=5m

[admin@EDCwifi] /interface bridge port> add interface=ether2 bridge=bridge1
[admin@EDCwifi] /interface bridge port> add interface=vif2 bridge=bridge1
[admin@EDCwifi] /interface bridge port> print
Flags: X - disabled, I - inactive, D - dynamic
 #   INTERFACE    BRIDGE        PRIORITY   PATH-COST   HORIZON
 0   ether2       bridge1       0x80       10          none
 1   vif2         bridge1       0x80       10          none
```



4. 为新的 metarouter 接口添加 IP 地址，ether1 作为物理外网连接（假设我们已经配置好物理路由器的网络连接），vif1 用于连接 metarouter 主机系统（vif1 可以认为是一个 lan 接口连接）：

```
[admin@EDCwifi] /ip address> add address=10.0.1.1/24 interface=vif1
[admin@EDCwifi] /ip address> print
Flags: X - disbled, I - invalid, D - dynamic
 #   ADDRESS              METWORK         BROADCAST       INTERFACE
 0   10.200.15.56/24      10.200.15.0     10.200.15.255   ether1
 1   10.0.1.1/24          10.0.1.0        10.0.1.255      vif1
[admin@EDCwifi] /ip address>
```



5. 进入 metarouter 控制平台，通过 console 命令：

```
[admin@EDCwifi] /metarouter> console client1

[ctrl-A is the prefix key]

Starting...
Starting services...

Mikrotik 3.22
Mikrotik login:admin
Password:

[admin@mikrotik] > /sys identity set name=client1
```

6. 配置 metarouter 的参数，设置以太网接口名称，让客户明白设备的连接情况：

```
[admin@MikroTik] /interface Ethernet> print
Flags: X - disabled, R - running, S - slave
 #   NAME              MTU      MAC-ADDRESS       ARP
 0 R ether1            1500     02:49:E8:55:8E:E8  enabled
 1 R lan               1500     02:16:16:90:EF:0E  enabled
[admin@MikroTik] /interface Ethernet>
```

为外网和内网接口配置 IP 地址

```
[admin@MikroTik] /ip address> add address=10.0.1.2/24 interface=wan
[admin@MikroTik] /ip address> add address=10.0.2.1/24 interface=1
[admin@MikroTik] /ip address> print
Flags: X - disabled, I - invalid, D - dynamic
 #  ADDRESS         NETWORK       BROADCAST      INTERFACE
 0  10.0.1.2/24     10.0.1.0      10.0.1.255      wan
 1  10.0.2.1/24     10.0.2.0      10.0.2.255      lan
```

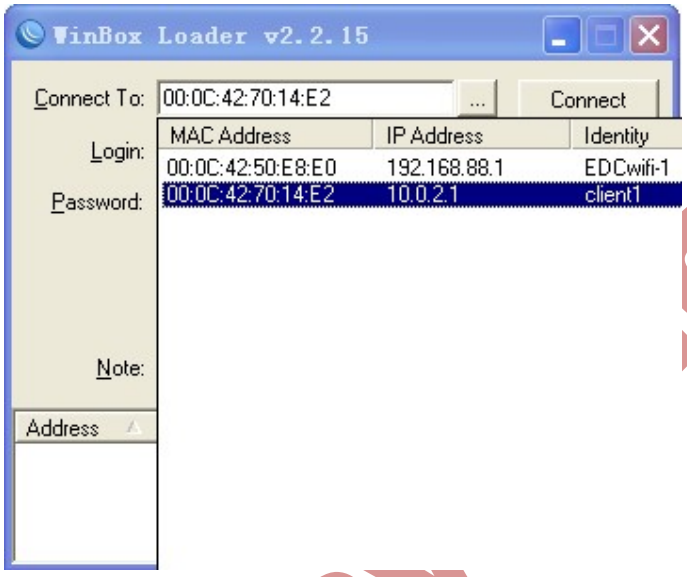
添加默认网关

```
[admin@MikroTik] /ip router> add gateway=10.0.1.1
[admin@MikroTik] /ip router> print
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r -
rip, b - bgp, o - ospf, m - mme, B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS      PREF-SRC    G GATEWAY    DISTANCE   INTERFACE
 0 A s  0.0.0.0/0        r 10.0.1.1                1          wan
 1 ADC  10.0.1.0/24      10.0.1.2                  0          wan
 2 ADC  10.0.2.0/24      10.0.2.1                  0          wan
[admin@MikroTik] /ip router>
```

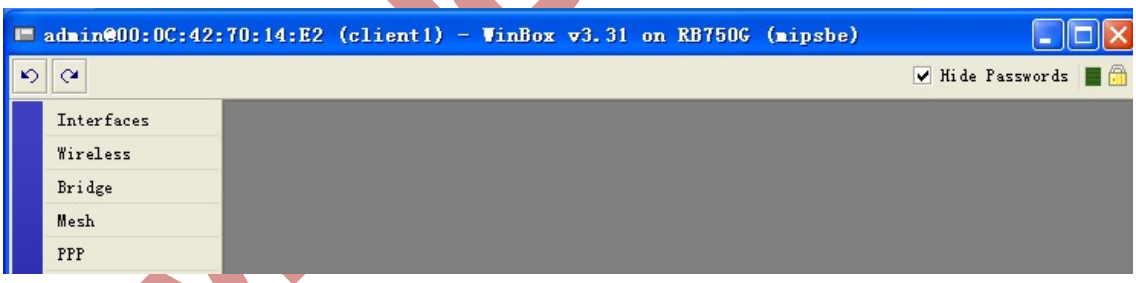
配置 nat 转换

```
[admin@MikroTik] /ip firewall nat> add action=masquerade out-interface=wan
chain=srcnat
```

配置完成后，我们可以通过局域网的 winbox 扫描到配置好的 metarouter

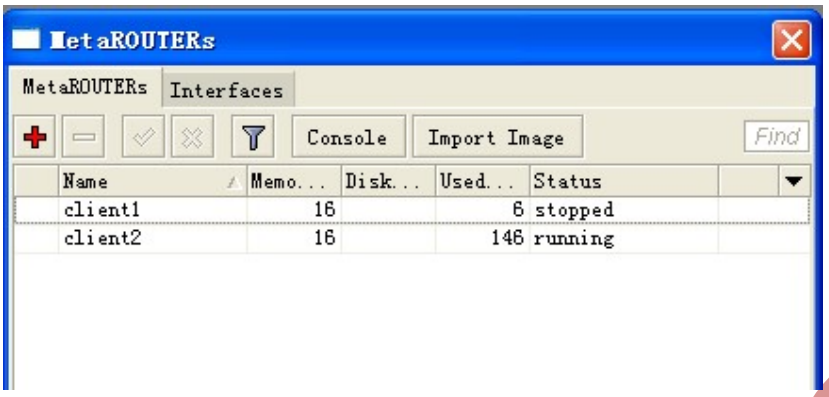


通过连接后，在 winbox 中显示 metarouter 信息



这样我们可以通过局域网内部，连接虚拟的 RouterOS 上网。这样我们可以让局域网 1 的管理进入 client1 的 metarouter 配置自己的网络参数。

我们在用同样的方法建立 10.0.3.0/24 网络的第二个 metarouter 在 R 偶特瑞 BOARD 上，只要硬件性能允许，为不同客户提供多个自主路由器管理：



第二十五章 Log 日志管理

不同的系统事件和状态信息都能被 RouterOS 的 log 记录下，日志能被存储到本地路由器的内存或者文件中。也可以通过发送 Email 或者通过运行在远程的 syslog 下载程序存储存储到其他的系统硬盘上。

RouterOS 中的日志有不同的分组或者项目，日志来至于每个项目运行状态，可通过配置进行每个组或项目的记录。局部日志文件能存储到内存中（内存记录为默认并会显示到/log 目录下，在重启或者断电后日志会丢失），以及远程记录等。

操作路径：/system logging

属性描述

Action（name；默认：memory）- 用户可选择在/system logging action 指定操作的类型
Prefix（文本）- 本地日志前缀
Topics（info | critical | firewall | keepalive | packet | read | time | write | ddns | Hotspot | l2tp | PPP | route | update | account | debug | ike | manager | PPPoE | script | warning | async | DHCP | notification | pptp | state | watchdog | bgp | error | IPsec | radius | system | web-proxy | calc | event | isdn | ospf | raw | telephony | wireless | e-mail | gsm | mme | ntp | open | ovpn | pim | radvd | rip | sertcp | ups；默认：info）- 指定日志组或者日志信息类型

在 logging 中通过记录 firewall 产生的日志信息，存储到本地缓存中。

```
[admin@mikrotik] system logging> add topics=firewall action=memory
[admin@mikrotik] system logging> print
Flags: X - disabled, I - invalid
#  TOPICS                    ACTION PREFIX
0  info                      memory
1  error                     memory
2  warning                   memory
3  critical                  echo
4  firewall                  memory
[admin@mikrotik] system logging>
```

Logging 执行

操作：/system logging action

属性描述

Disk-lines（整型；默认：100）- 在日志文件存储到硬盘的记录数量（仅在 action 设置为 disk）
Disk-stop-on-full（yes | no；默认：no）- 是否在 disk-lines 数量达到后停止存储日志信息
Email-to（name）- 发送到指定的 Email 地址（仅在 action 设置为 Email）
Memory-lines（整型；默认：100）- 在本地缓存记录的数量（仅在 action 设置为 memory）
Memory-stop-on-full（yes | no；默认：no）- 是否在 memory-lines 数量达到后停止存储日志信息
Name（name）- 一个 action 操作的名称
Remember（yes | no；默认：yes）- 是否保存日志信息，其中尚未显示在控制台的（仅在 action 设置为 echo）
Remote（IP address：Port；默认：0.0.0.0:514）- 远程日志服务器的 IP 地址和 UDP 端口（仅在 action 设置为 remote）
Target（disk | echo | Email | memory | remote；默认：memory）- 记录存储设备或目标
Disk - 日志记录到硬盘
Echo - 日志显示在控制台屏幕上
Email - 日志通过 Email 发送
Memory - 日志被存储到本地内存
Remote - 日志发送到远端服务主机

注：你不能删除或重命名默认 action 规则

添加一个新的 action 取名为 long，将日志记录到本地内存，在内存中的记录为 1000 条，这样在/log 中会显示 1000 条记录，用于查看很多的信息：

```
[admin@mikrotik] system logging action> add name=long \
\... target=memory memory-lines=50 memory-stop-on-full=yes
[admin@mikrotik] system logging action> print
Flags: * - defaull
 #   NAME                      TARGET   REMOTE
 0 * memory                    mamory
 1 * disk                      disk
 2 * echo                      echo
 3 * remote                    remote 0.0.0.0:514
 4   long                      memory
[admin@mikrotik] system logging action>
```

通过 ip firewall filter 记录所有访问 80 端口，并在 log 中添加前缀“80Port”的相关信息：

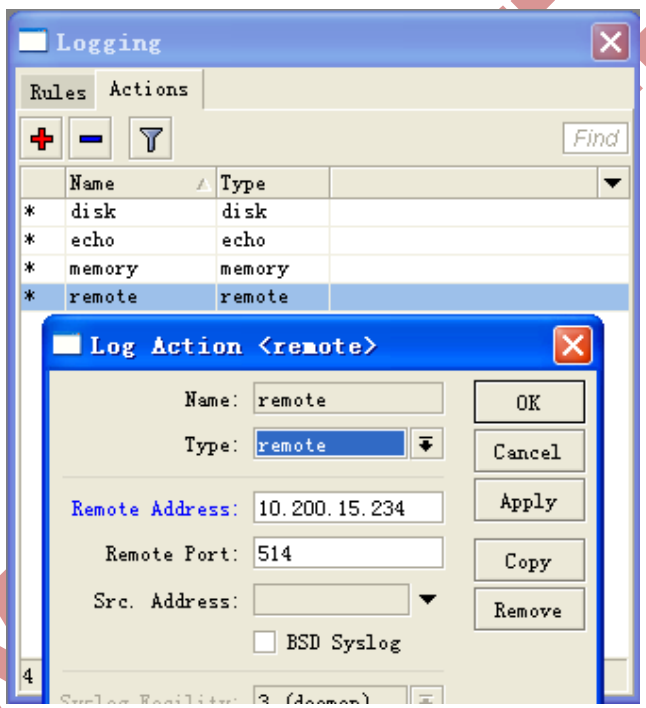
```
[admin@mikrotik] /ip firewall filter> add chain=forward protocol=tcp dst-port=80
Action=log log-prefix=80port
[admin@mikrotik] /ip firewall filter> print
```

```
Flags: X - disabled, I - invalid, D - dynamic

    0 chain=forward action=log protocol=tcp dst-port=80 log-prefix=” 80port”
```

使用 dude 管理器记录系统日志

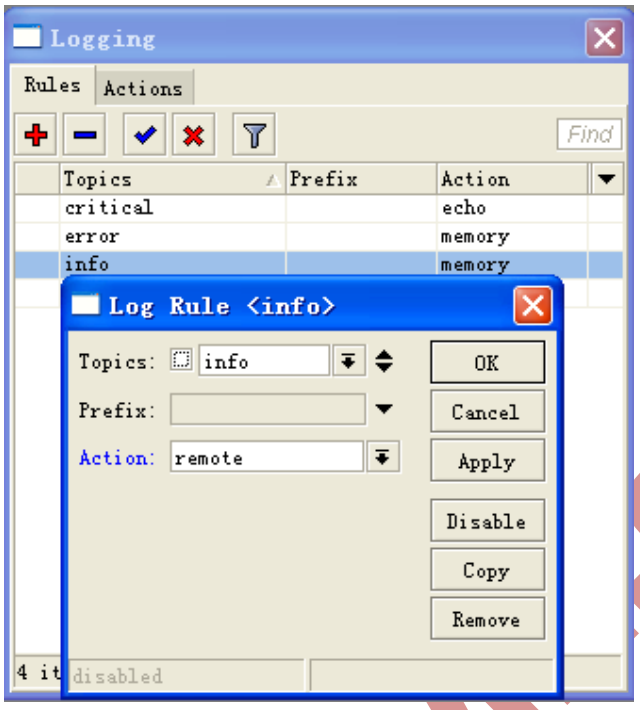
在 Mikrotik 提供的 syslog 软件，用于记录 RouterOS 的系统日志信息，但这个软件只能记录 1000 条，不能做长时间记录和定期存储。在新版本的 the dude 网络管理软件中增加了系统日志记录和下载的功能，这个我们可以通过 the dude 管理器对我们需要的 RouterOS 日志信息进行记录和管理。

这里我们使用的是 the dude 3.0beta8 的版本，首先我们需要进入 RouterOS 的 system logging 配置系统日志的远程记录参数：

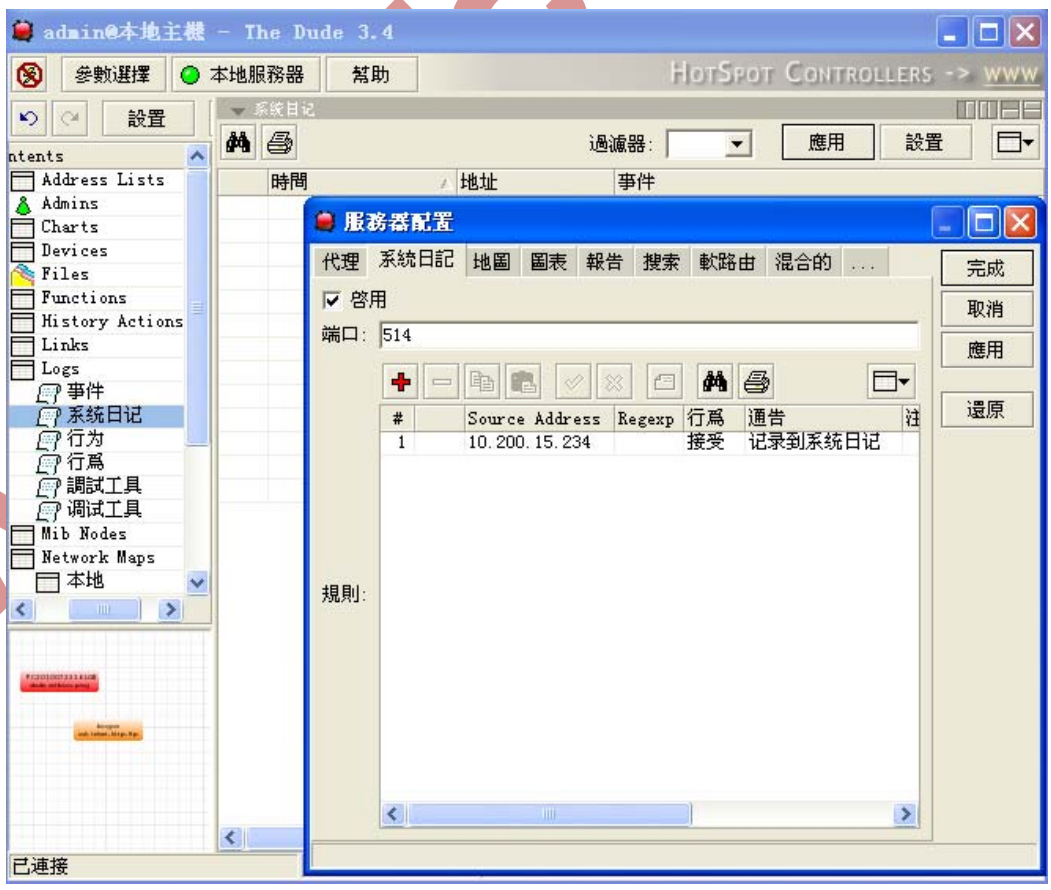


进入 system logging 后配置 action 里的 remote 参数，将 remote address 配置为指定的系统信息接受的 the dude 服务器 IP 地址。

然后将需要记录的日志信息，设置为 remote：

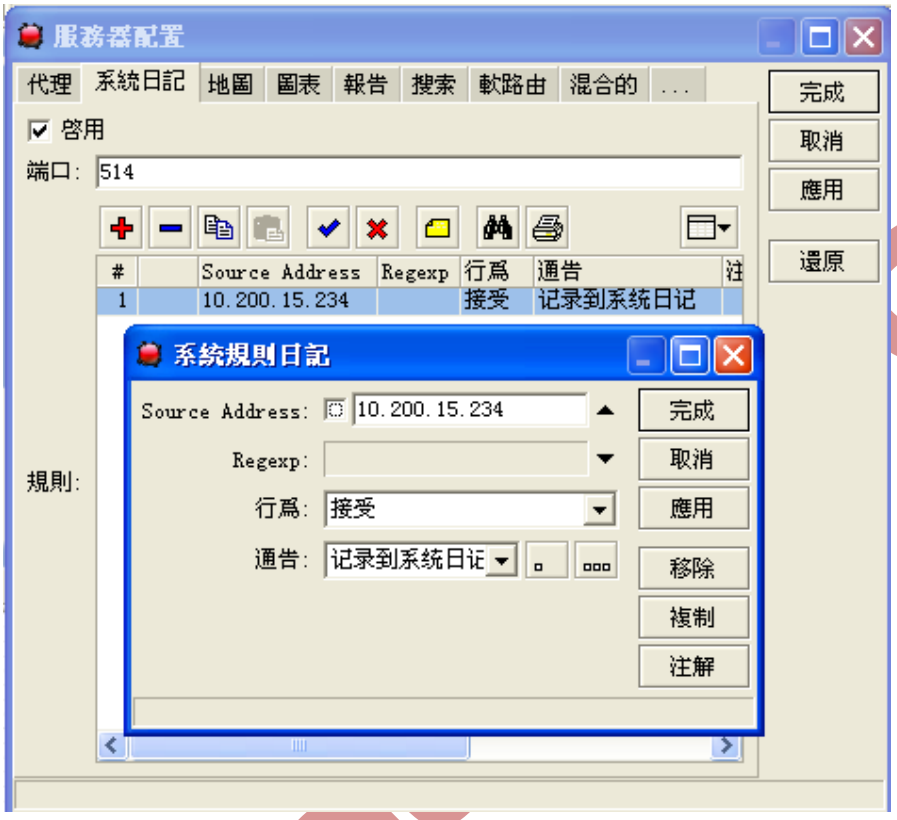


下面是在 10.200.15.234 的 the dude 网络管理器上配置系统日志信息，进入“设置”，选择“系统日志”，并配置相应的端口和 IP 地址。

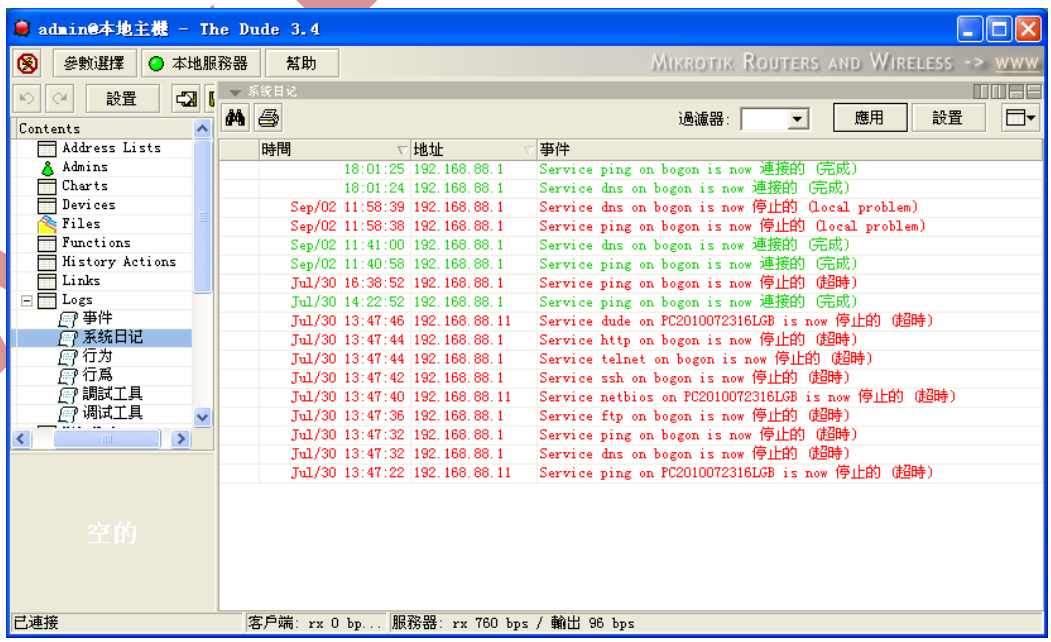


[键入文字]

配置 the dude 系统日志记录参数：

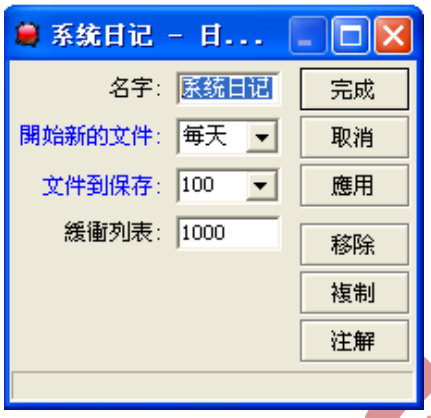


配置完成后，我们在 the dude 管理器的 log 下系统日志看到接受到 IP 地址为 10.200.15.1 的 RouterOS 的日志：



[键入文字]

在 the dude 的 log 记录中，有一个“设置”选项，可以配置日志记录的存储参数：



这里我们设置日志记录存储的文件名、产生新的日志文件时间间隔等参数。

通过 the dude 网络管理软件，可以方便的记录每台 RouterOS 的日志信息和情况，同时达到监控的目的。这样对你的网络进行综合的管理和监控，分析网络运行情况和状态，为你及时对网络环境进行处理和改造提供及时的信息。

Log 信息

操作路径：/log

用于显示/system logging 记录的日志信息

属性描述

Message（只读：文本）- 信息文本
Time（只读：文本）- 事件的日期和时间
Topics（只读：文本）- 项目信息的从属

查看本地日志：

```
[admin@MikroTik] > log print
 TIME                MESSAGE
 dec/24/2003 08:20:36 log configuration changed by admin
 dec/24/2003 08:20:36 log configuration changed by admin
 dec/24/2003 08:20:36 log configuration changed by admin
 dec/24/2003 08:20:36 log configuration changed by admin
 dec/24/2003 08:20:36 log configuration changed by admin
 dec/24/2003 08:20:36 log configuration changed by admin
-- [Q quit|D dump]
```

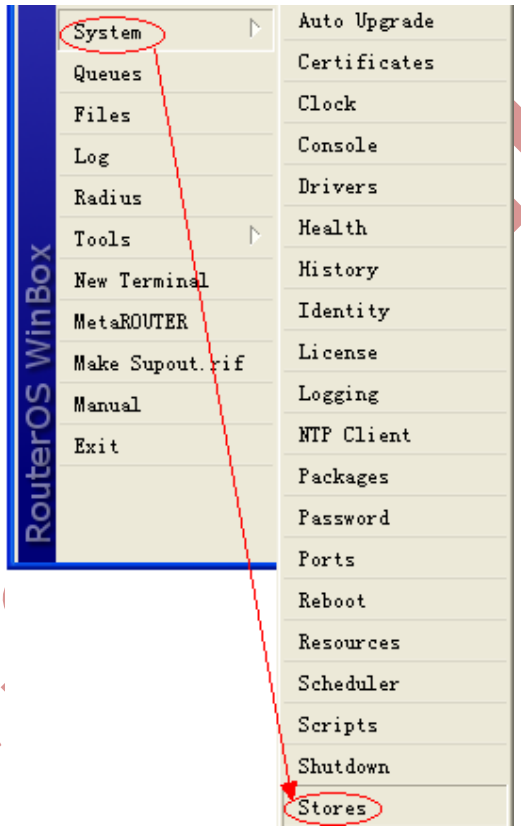
监视系统日志：

```
[admin@MikroTik] > log print follow
 TIME                MESSAGE
 dec/24/2003 08:20:36 log configuration changed by admin
 dec/24/2003 08:24:34 log configuration changed by admin
 dec/24/2003 08:24:51 log configuration changed by admin
 dec/24/2003 08:25:59 log configuration changed by admin
 dec/24/2003 08:25:59 log configuration changed by admin
 dec/24/2003 08:30:05 log configuration changed by admin
 dec/24/2003 08:30:05 log configuration changed by admin
 dec/24/2003 08:35:56 system started
 dec/24/2003 08:35:57 isdn-out1: initializing...
 dec/24/2003 08:35:57 isdn-out1: dialing...
 dec/24/2003 08:35:58 Prism firmware loading: OK
 dec/24/2003 08:37:48 user admin logged in from 10.1.0.60 via telnet
-- Ctrl-C to quit. New entries will appear at bottom.
```

第二十六章 RouterOS store 功能

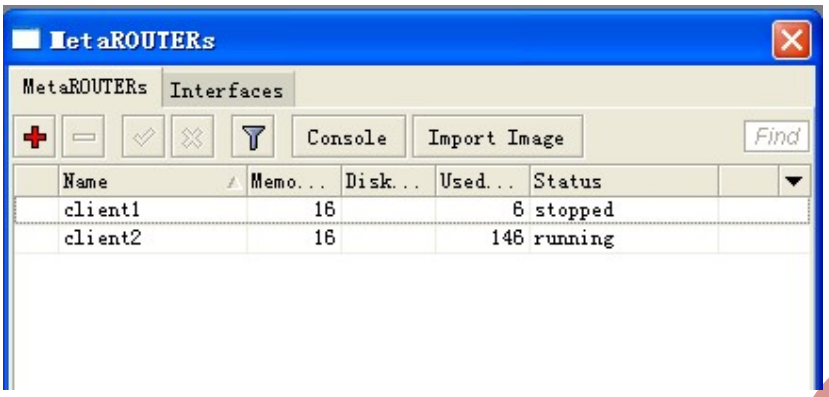
RouterOS 在 3.15 后增加了 store 存储功能，支持各种本地系统存储和外部设备存储，主要应用于 web-proxy、user-manager 和 the dude 数据存储，在 3.23 后由于 RouterOS 支持 log 日志的本地存储，所以 store 的应用有所增加。

除了 RouterOS 使用本地系统盘存储外，我们可以在 PC 或者 RouterBOARD 上增加各种存储设备，比如 RouterBOARD 可以选择 CF/MircoSD 方式存储，而 PC 可以选择增加硬盘、U 盘等方式。我们进入 winbox 后可以选择 store 目录，进入存储管理。

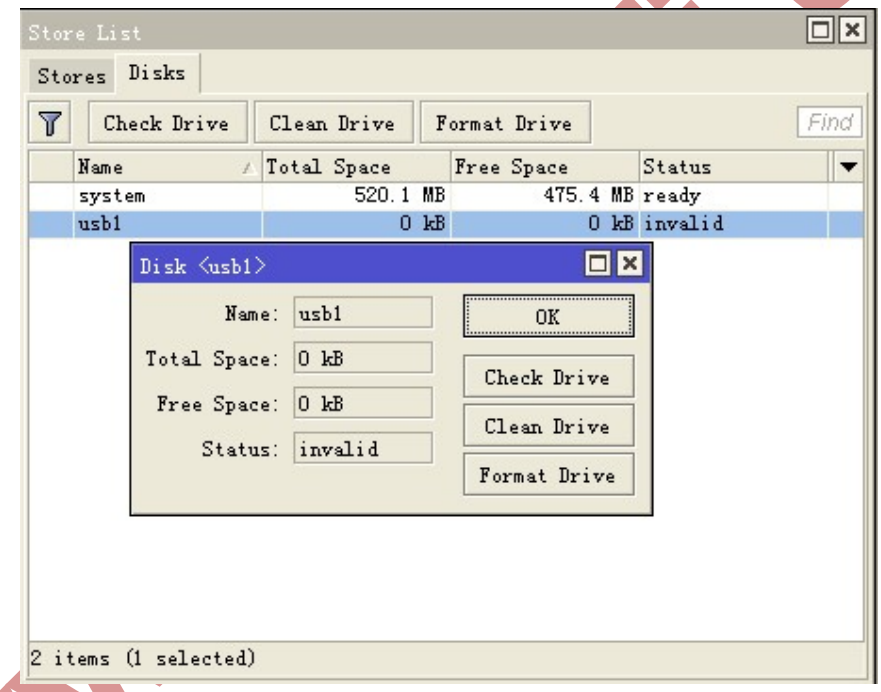


RouterOS 使用 U 盘扩展存储

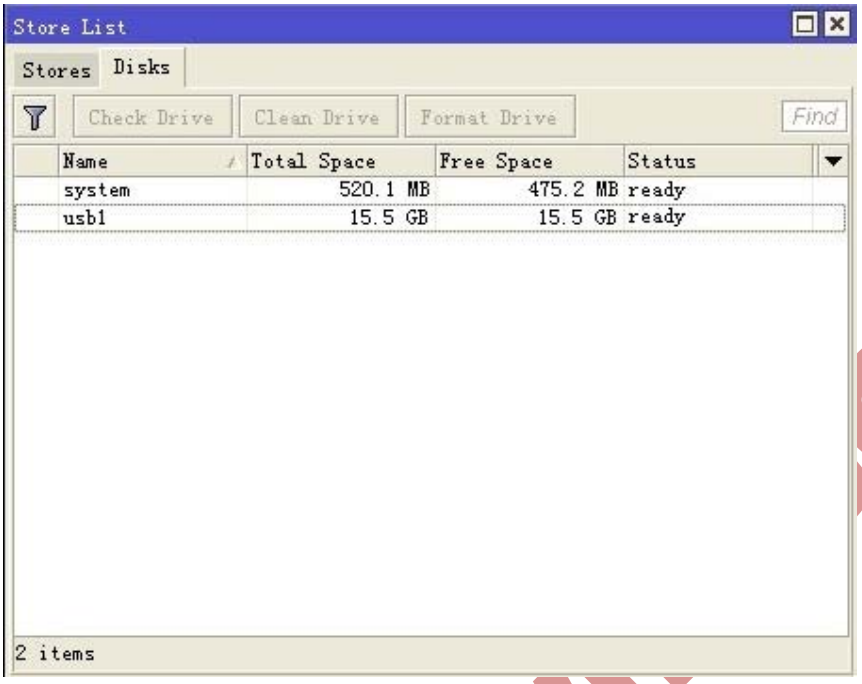
这里我们通过使用 U 盘来演示，在 PC 上增加外部存储的操作，我们将一个 16G 的 U 盘，插入 RouterOS PC 的 USB 接口，这个时候，我们可以在 store 的 disk 目录中找到 USB1 的硬盘信息：



当前状态为 invalid，即无法识别，因为 RouterOS 的硬盘分区和我们常用的 U 盘分区不同，所以我们需要选择 USB1，对 U 盘做格式化操作，选择 format driver 的选项



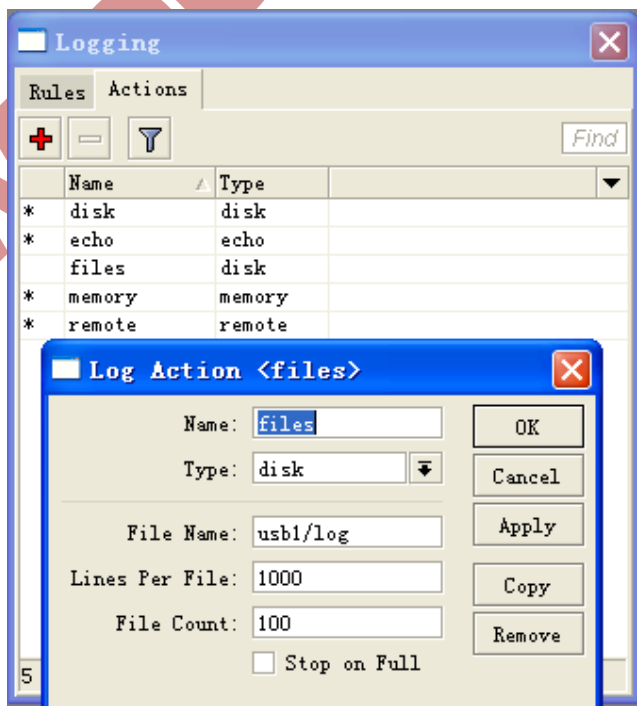
在格式化完成后，我们可以看到当前的 USB1 状态为 ready，能够正常识别到容量和空闲存储空间：



存储 log 日志信息

在 RouterOS 3.23 后增加了可以将 log 日志存储到 RouterOS 上的存储设备里，由于本地系统存储空间有限，我们可以通过外部存储的 U 盘扩展，这里我们通过我们可以使用日志记录

首先我进入 system logging 设置 action，并新建立一个 files 规则，并定义存储方式 type 为 disk：



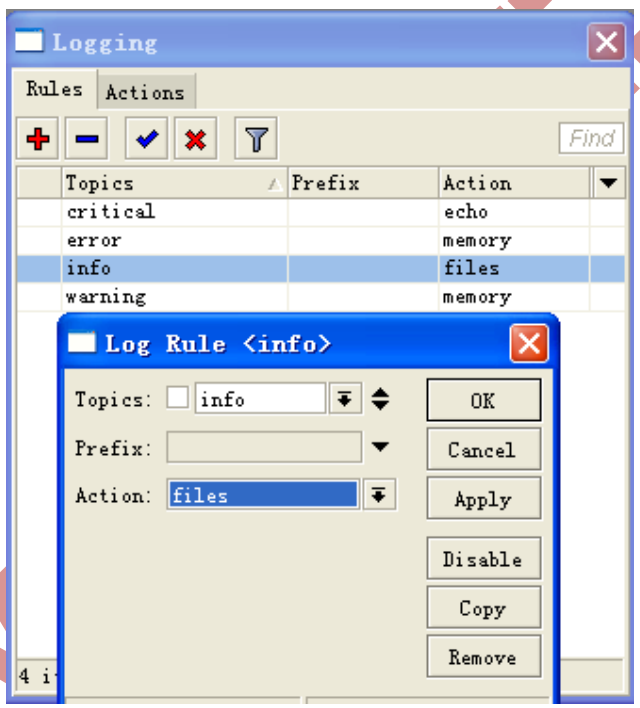
[键入文字]

Disk 类型几个参数如下：

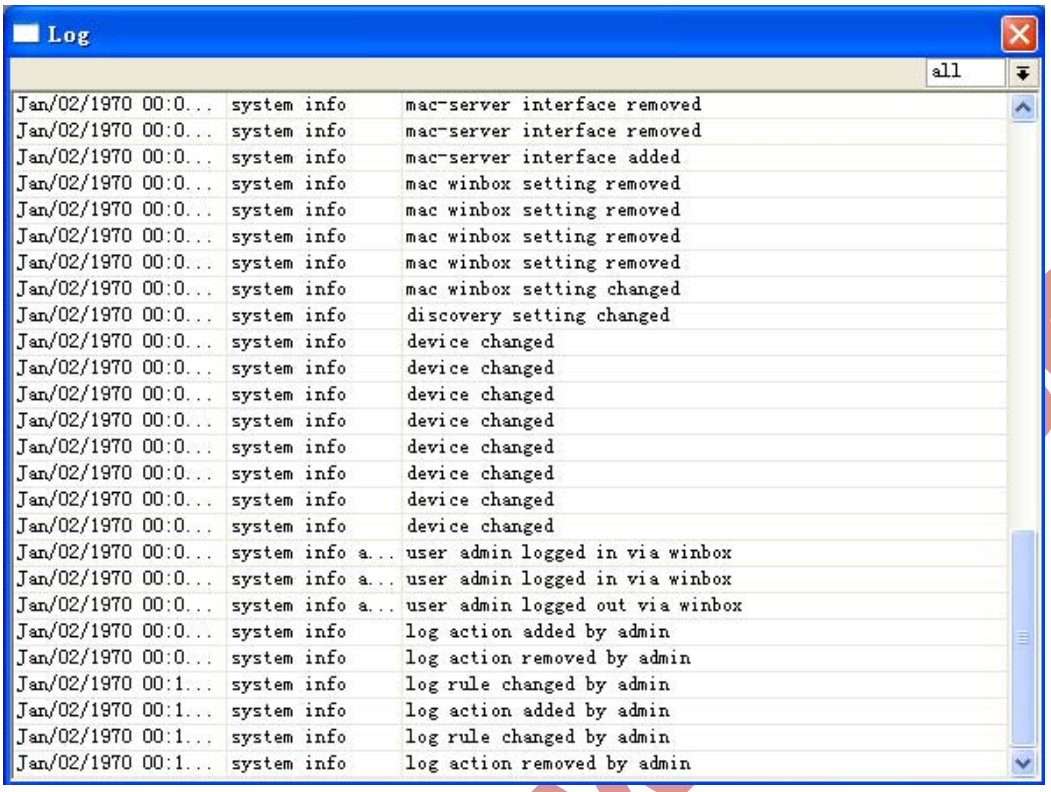
Type：log 日志记录方式，这里我们选择 disk
File name：文件存储的路径，如果是 USB1 的 U 盘，我给的路径时 USB1/log
Lines per file：每个文件记录多少条信息
File count：log 日志一共建立多少文件，如果日志记录超出文件数量，将会从 log0 号从头开始记录并覆盖原来的文件
Stop on full：当 log 建立的文件到达后，停止向文件写入 log 日志

注意：文件名建立的原则，即<filename>.0.txt,<filename>.1.txt,<filename>.n.txt 的顺序建立，文件的大小可以自行定义。

设置 logging 的 info（信息记录）为 files 操作，即记录到 USB1 中



我们可以看到在 log 中记录的防火墙信息

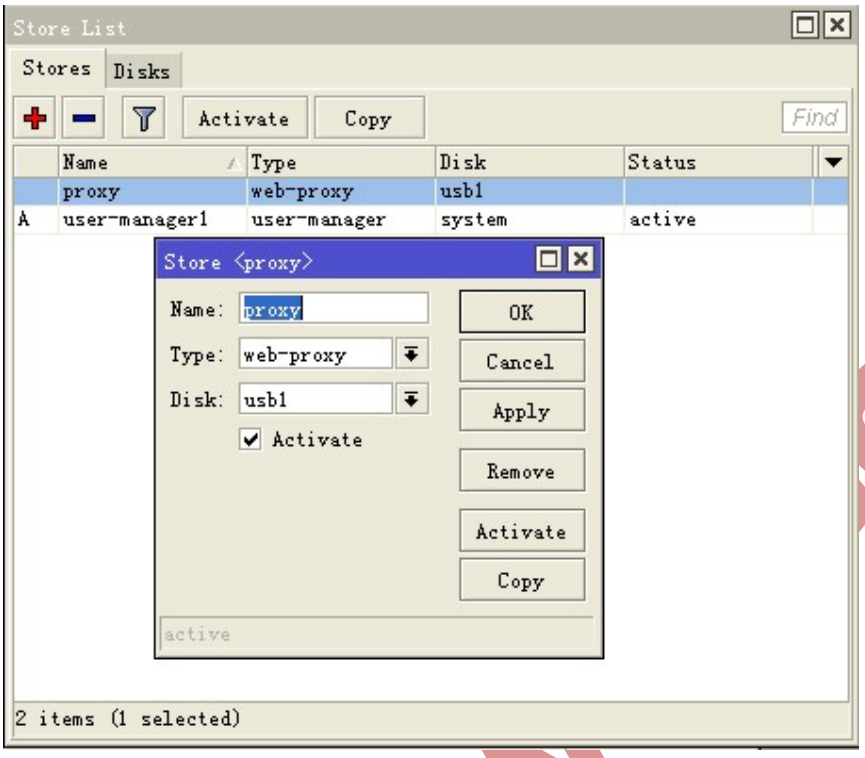


注意，当在记录大量的日志信息，使用本地存储设备写入数据时，会出现 CPU 占用较大的情况，需要注意合理分配你的系统资源，建议尽量做远程日志记录的存储。

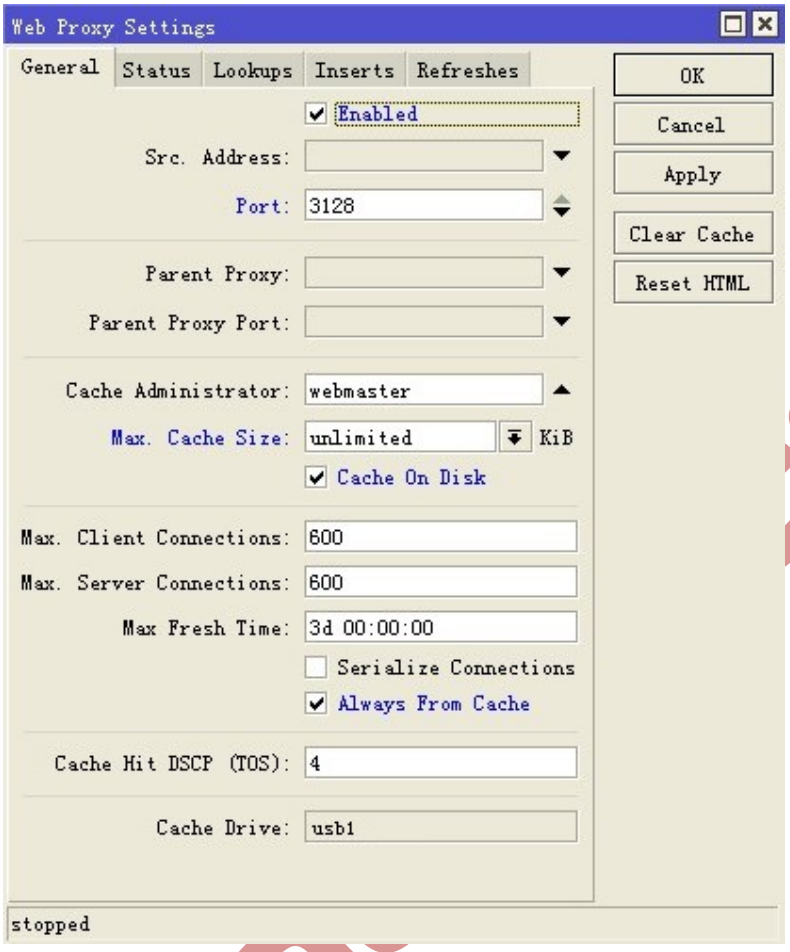
Web-proxy 使用 U 盘存储

一些特殊网络环境，可能会用到 Web 缓存功能，需要将访问过的静态页面缓存到硬盘中，做二次访问。由于系统盘空间有限，我们可以使用 U 盘做为网页数据的外部存储。

下面在 store 目录下添加一个名 proxy 的规则，选择类型为 web-proxy，指定硬盘为 USB1

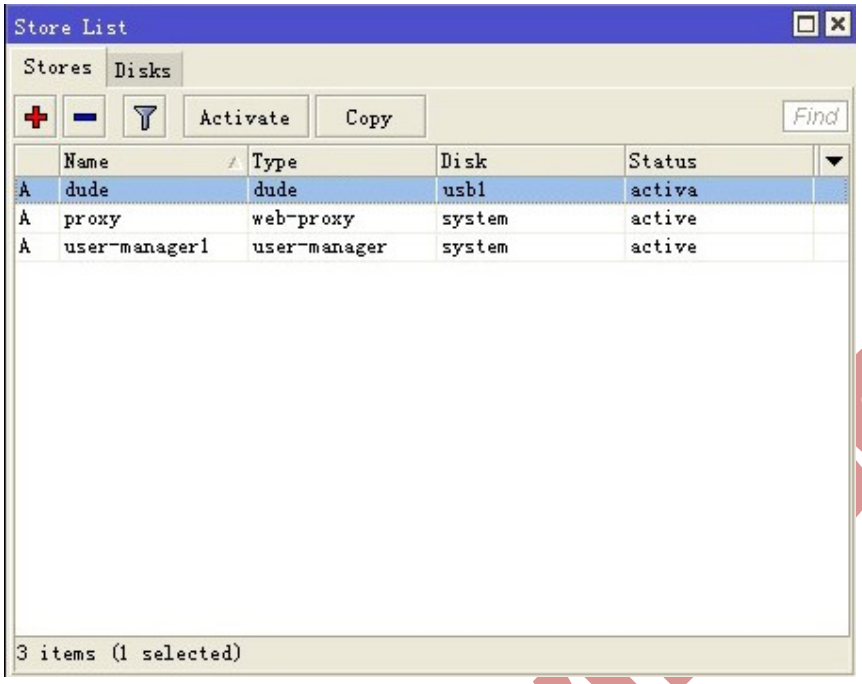


设置 web-proxy 的配置，这里可以看到 cache_drive 会根据 store 的配置，调用 USB1 的外部存储



Store 的其他应用

Store 可以建立 the dude 网络管理器的拓扑结构图的数据存储，如下图



Store 功能也可以用户存储 user-manager 的数据信息，建立 user-manager 的数据库。

第二十七章 IP 访问日志记录

IP 访问日志记录，用于内向外或者外向内发送的所有连接（包括源地址、目标地址、数据包和字节）都会被记录下来。同时当启用 Hotspot 和 PPP 认证时，账号也随之被记录到相应的连接中。在 RouterOS 中主要应用于记录内网与外网之间的访问日志，以便对网络中的所有数据做记录进行检查和分析，或者出于安全考虑为以后非法连接提供依据。

规格

功能包要求：system
认证等级：Level 1
操作路径：/user, /ppp, /ip accounting, /radius
硬件使用：传输记录需要根据记录内容大小增加内存

IP 访问记录

操作路径：/ip accounting

当每个包通过路由器时，匹配 IP 数据包源和目的地址会成对的在访问列表中并且这个对的流量会增加。PPP，PPTP，PPPoE，ISDN，以及 Hotspot 客户的流量也可以在每个用户的基础上计算。数据包的数量和字节的数量都会被计算。如果没有与之前的 IP 或用户对匹配，那么新的记录将被添加到里表中。

属性描述

Enabled（yes | no；默认：no）- 是否启用了本地 IP 访问记录日志
Accounting-local-traffic（yes | no；默认：no）- 是否计算来自/到达路由器的流量访问
Threshold（整型；默认：256）- 在管理列表中的 IP 对的最大数量（最大值为 8192）

当临界值限制达到时，没有新的 IP 对将被添加到管理列表中。在管理列表中没有计算的每个包都将被添加到 uncounted 计数器。启用 IP 访问管理：

```
[admin@MikroTik] ip accounting> set enabled=yes
[admin@MikroTik] ip accounting> print
              enabled: yes
  account-local-traffic: no
              threshold: 256
[admin@MikroTik] ip accounting>
```

IP 访问快照

[键入文字]

操作路径：/ip accounting snapshot

当数据收集的快照做好后，管理列表会被清空并且新的 IP 对与流量数据会被添加进来。更经常的数据会被收集。

属性描述

Bytes（只读：整型）- 字节总数，以条目匹配
Dst-address（只读：IP address）- 目的 IP 地址
Dst-user（只读：text）- 接受者的名称（如果可应用）
Packets（只读：整型）- 包的总数，以这个条目匹配
Src-address（只读：IP address）- 源 IP 地址
Src-user（只读：text）- 发送者的名称（如果可用）

注：仅当用户通过一个 PPP 隧道连接到路由器或被 Hotspot 认证时才显示用户名。在获取快照之前，列表是空的。

取一个新 IP 访问的快照：

```
[admin@MikroTik] ip accounting snapshot> take
[admin@MikroTik] ip accounting snapshot> print
 # SRC-ADDRESS     DST-ADDRESS     PACKETS    BYTES      SRC-USER    DST-USER
 0 192.168.0.2      159.148.172.197 474         19130
 1 192.168.0.2      10.0.0.4          3         120
 2 192.168.0.2      192.150.20.254   32         3142
 3 192.150.20.254   192.168.0.2      26         2857
 4 10.0.0.4         192.168.0.2       2          117
 5 159.148.147.196  192.168.0.2       2          136
 6 192.168.0.2      159.148.147.196   1          40
 7 159.148.172.197  192.168.0.2      835         1192962
[admin@MikroTik] ip accounting snapshot>
```

Web 获取 IP 访问信息

操作路径：/ip accounting web-access

Web 页面报告似的使用标准的 Unix/linux wget 工具收集流量数据并存储到文件或者使用 Mikrotik 的日志下载软件。如果 Web 报告启用且 Web 页面被查看，那么当连接起始为 Web 页面时，snapshot 将被生成。Snapshot 将在 Web 页面上显示。被有 wget 工具 http 连接使用的 TCP 协议保证任何一点的流数据都不会丢失。Snapshot 图像将在来自 wget 的连接被初始化时生成。Web 浏览器或 wget 可以连接到 URL：http://[routerIP]/accounting/ip.cgl

属性描述

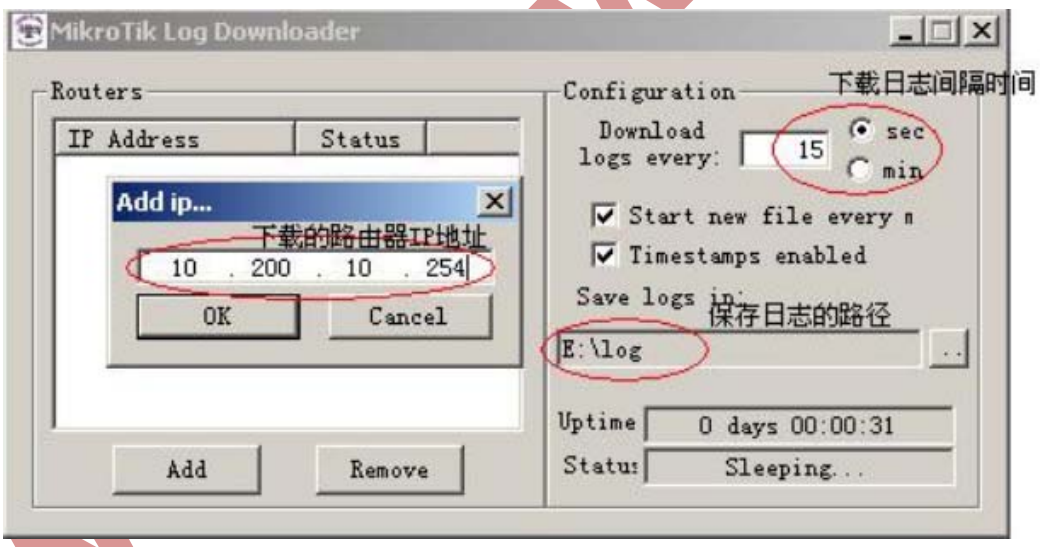
Accessible-via-web（yes | no；default：no）- 是否 snapshot 通过 Web 可用
Address（IP address/netmask；default：0.0.0.0）- 允许存取 snapshot 的 IP 地址范围

仅启用来自 192.168.10.10 主机对流量日志的 Web 访问：

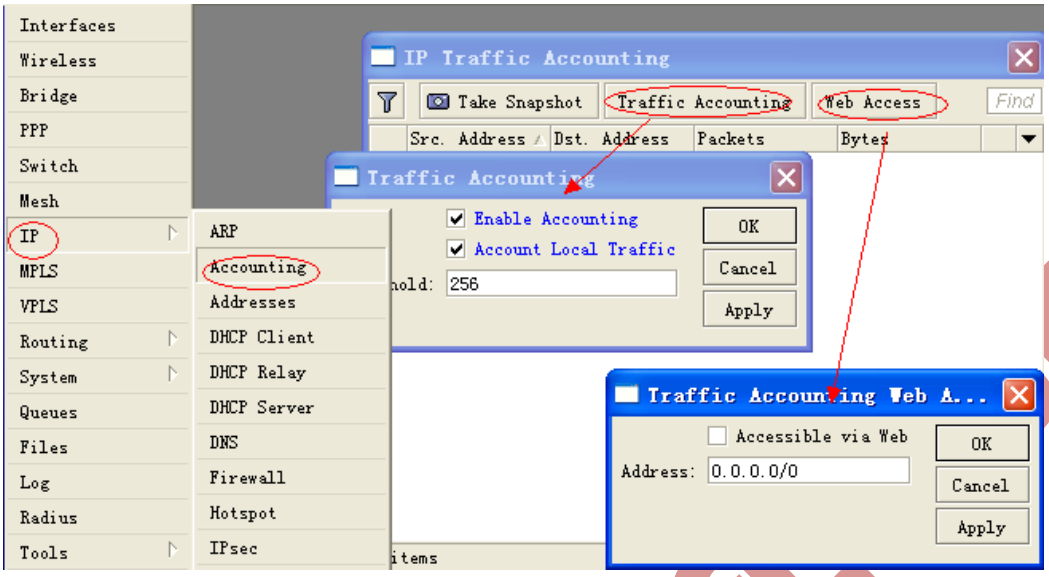
```
[admin@MikroTik] ip accounting web-access> set accessible-via-web=yes \
\... address=192.168.10.10/32
[admin@MikroTik] ip accounting web-access> print
   accessible-via-web: yes
              address: 192.168.10.10/32
[admin@MikroTik] ip accounting web-access>
```

下面是通过 log downloader 和 winbox 操作的事例

首先打开 log downloader 程序，如图所示，添加需要记录 RouterOS 的日志的 IP 地址，并配置相应的参数：



然后再 RouterOS 打开，并启用日志的远程记录，在 ip accounting 中设置：



注意：该软件只能通过 RouterOS 的 Web 端口记录，即 Web 端口默认必须是 80。

第二十八章 Scheduler（计划任务）

设定的计划任务，并通过时间安排执行相应的脚本操作。

规格

功能包需求：system
等级需求：Level 1
操作路径：/system scheduler

计划任务配置

计划任务列表能触发脚本执行，在指定的时间段或者是在指定的时间间隔。

属性描述

Interval（time；默认：0s）- 脚本执行的间隔时间，脚本反复执行在一个指定的时间间隔
Name（name）- 任务名
On-event（name）- 脚本执行名。通过调用/system script 里的脚本规则名称
Run-count（只读：整型）- 开始脚本执行的日期
Start-date（date）- 开始脚本执行的日期
Start-time（time）- 开始脚本执行的时间
Startup - 默认在系统启用 3 秒后执行脚本。

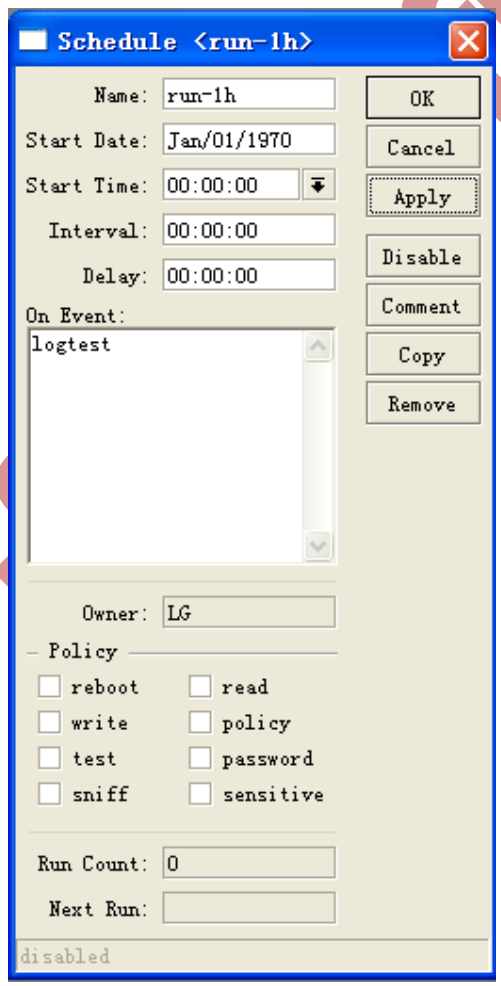
注：重启路由器时将重置 run-count 计数器。

如果计划表选项里面对 start-time 设置了 startup，则在控制台开启后 3 秒运行。这意味着所有的脚本设置为 start-time=startup 和 interval=0，当路由器启动就会被执行。

事例 1：我们添加一个任务执行系统日志记录测试，并间隔 1 小时执行一次，这个操作为 logtest：

```
[admin@MikroTik] system script> add name=log-test source=:log message=test
[admin@MikroTik] system script> print
   0 name="log-test" source=":log messgae=test" owner=admin run-count=0
[admin@MikroTik] system script> .. scheduler
[admin@MikroTik] system scheduler> add name=run-1h interval=1h
on-event=log-test
[admin@MikroTik] system scheduler> print

Flags: X - disabled
 #   NAME      ON-EVENT  START-DATE  START-TIME INTERVAL     RUN-COUNT
 0   run-1h    log-test  mar/30/2004 06:11:35   1h          0
[admin@MikroTik] system scheduler>
```

Schedule 在 winbox 的配置如下：



事例 2：另外一个例子是添加 2 个脚本改变带宽设置队列规则“cust0”，每天上午 9 点限制为 64kb/s，下午 5 点限制为 128kb/s。这个队列的规则、脚本和计划任务如下（注：在 2.9 种 cust0 是不需要加双引号的，但在 3.0 中需要注明字符串，要加上双引号“cust0”）：

[键入文字]

```
[admin@MikroTik] queue simple> add name=Cust0 interface=ether1 \
\... dst-address=192.168.0.0/24 limit-at=64000
[admin@MikroTik] queue simple> print
Flags: X - disabled, I - invalid
  0   name="Cust0" target-address=0.0.0.0/0 dst-address=192.168.0.0/24
      interface=ether1 limit-at=64000 queue=default priority=8 bounded=yes

[admin@MikroTik] queue simple> /system script
[admin@MikroTik] system script> add name=start_limit source={/queue simple set \
\... Cust0 limit-at=64000}
[admin@MikroTik] system script> add name=stop_limit source={/queue simple set \
\... Cust0 limit-at=128000}
[admin@MikroTik] system script> print
  0 name="start_limit" source="/queue simple set Cust0 limit-at=64000"
    owner=admin run-count=0

  1 name="stop_limit" source="/queue simple set Cust0 limit-at=128000"
    owner=admin run-count=0

[admin@MikroTik] system script> .. scheduler
[admin@MikroTik] system scheduler> add interval=24h name="set-64k" \
\... start-time=9:00:00 on-event=start_limit
[admin@MikroTik] system scheduler> add interval=24h name="set-128k" \
\... start-time=17:00:00 on-event=stop_limit
[admin@MikroTik] system scheduler> print
Flags: X - disabled
  #   NAME      ON-EVENT START-DATE  START-TIME INTERVAL           RUN-COUNT
  0   set-64k   start... oct/30/2008 09:00:00   1d                 0
  1   set-128k  stop_... oct/30/2008 17:00:00   1d                 0

[admin@MikroTik] system scheduler>
```

事例 3：下面的例子安排了一个通过电子邮件发送每周备份路由器配置信息的脚本：

```
[admin@MikroTik] system script> add name=e-backup source={/system backup
{... save name=email; /tool e-mail send to="root@host.com" subject=([/system
{... identity get name] . " Backup") file=email.backup}
[admin@MikroTik] system script> print
  0 name="e-backup" source="/system backup save name=ema... owner=admin
    run-count=0

[admin@MikroTik] system script> .. scheduler
[admin@MikroTik] system scheduler> add interval=7d name="email-backup" \
\... on-event=e-backup
[admin@MikroTik] system scheduler> print
Flags: X - disabled
  #   NAME      ON-EVENT START-DATE  START-TIME INTERVAL            RUN-COUNT
  0   email-... e-backup oct/30/2008 15:19:28   7d                  1
[admin@MikroTik] system scheduler>
```

不要忘记去设置电子邮件参数，即 SMTP 服务的配置，操作路径/tool e-mail 例如（注：建议是自己的 SMTP 服务器，一些正规网站的邮件服务器可能会将发送信息屏蔽）：

```
[admin@MikroTik] tool e-mail> set server=159.148.147.198 from=SysAdmin@host.com
[admin@MikroTik] tool e-mail> print
   server: 159.148.147.198
     from: SysAdmin@host.com
[admin@MikroTik] tool e-mail>
```

事例 4

下面的例子是从午夜 12 点到正午 12 点的每个小时里把“×”加进日志中：

```
[admin@MikroTik] system script> add name=enable-x source={/system scheduler
{... enable x}
[admin@MikroTik] system script> add name=disable-x source={/system scheduler
{... disable x}
[admin@MikroTik] system script> add name=log-x source={:log message=x}
[admin@MikroTik] system script> .. scheduler
[admin@MikroTik] system scheduler> add name=x-up start-time=00:00:00 \
\... interval=24h on-event=enable-x
[admin@MikroTik] system scheduler> add name=x-down start-time=12:00:00
\... interval=24h on-event=disable-x
[admin@MikroTik] system scheduler> add name=x start-time=00:00:00 interval=1h \
\... on-event=log-x
[admin@MikroTik] system scheduler> print
Flags: X - disabled
 #   NAME     ON-EVENT START-DATE  START-TIME INTERVAL           RUN-COUNT
 0   x-up     enable-x oct/30/2008 00:00:00   1d                 0
 1   x-down   disab... oct/30/2008 12:00:00   1d                 0
 2   x        log-x    oct/30/2008 00:00:00   1h                 0
[admin@MikroTik] system scheduler>
```

第二十九章 RouterOS 常用工具

1、 netwatch 监控

netwatch 工具通过 ping 监控网络中的主机，并能通过状态的改变产生定义的事件。

规格

需要功能包：advanced-tools
等级：Level 1
操作路径：/tool netwatch
协议标准：none

Netwatch 监控的是在网络上的主机状态。通过在列表中指定 IP 地址，并发送间隔的 ICMP 的 ping 探测和执行控制脚本。在主机状态改变时根据 netwatch 的情况下命令。

属性描述

Down-script（名称）- 当一个主机的状态从 unknown 或 up 改变为 down。
Host（IP 地址；默认：0.0.0.0）- 需要监视的主机 IP 地址
Interval（时间；默认：1S）- ping 间隔时间
Status（只读：up | down | unknown）- 显示主机的当前状态
Up - 主机状态为 up
Down - 主机状态为 down
Unknown - 在列表项目属性被改变后或是项目被启用或禁用
Timeout（时间；默认：1s）- 每个 ping 的 timeout 值。在这个时钟周期内没有收到来至主机的回应，将认为该主机为 down
Up-script（名称）- 当一个主机的状态从 unknown 或 down 改变 up

事例

这个事例将运行脚本 gw_1 或 gw_2 根据网关的状态来修改默认网关：

```
[admin@mikrotik] system script> add name=gw_1 source={/ip route set
{... [/ip route find dst 0.0.0.0] gateway 10.0.0.1}
[admin@mikrotik] system script> add name=gw_2 source={/ip route set
{..[/ip route find dst 0.0.0.0] gateway 10.0.0.217}
[admin@mikrotik] system script> /tool netwatch
[admin@mikrotik] tool netwatch> add host=10.0.0.217 interval=10s timeout=998ms
\\... up-script=gw_2 down-script=gw_1
[admin@mikrotik] tool netwatch> print
Flags: X - disabled
```

```
 #  HOST          TIMEOUT       INTERVAL      STATUS
 0  10.0.0.217    997ms         10s           up
[admin@mikrotik] tool netwatch> print detail
Flags: X - disabled
0  host=10.0.0.217 timeout=997ms interval=10s since=feb/27/2003 14:01:03
status=up up-script=gw_2 down-script=gw_1

[admin@mikrotik] tool netwathch
```

让我们来看上面的例子，如果网关变为无法到达改变默认路由。有两个脚本，当主机状态改变为 up 脚本“gw_2”执行一次。在这个事例中，相当于进入控制台执行下面的命令：

```
[admin@mikrotik] > /ip route set [/ip route find dst 0.0.0.0] gateway 10.0.0.217
```

/io route find dst 0.0.0.0 命令是返回在路由表中 dst-address 值为 0.0.0.0 的参数，通常这种值为默认路由。用于代替/ip route set 命令后的第一个变量

当主机状态变为 down 脚本“gw_1”执行一次。如下面：

```
[admin@mikrotik] > /ip route set [/ip route find dst 0.0.0.0] gateway 10.0.0.1
```

如果 10.0.0.217 地址无法到达，改变默认网关。

下面是另一个事例，无论什么时候 10.0.0.215 主机断线，发送 e-mail 通知到你指定的邮箱：

```
[admin@mikrotik] system script> add name=e-down source={/tool e-mail send
{... from=” rieks@mt.lv”  server=” 159.148.147.198”  body=” router down”
{... subject=” router at second floor is down”  to=” rieks@latnet.lv”}
[admin@mirotik] system script> add name=e-up souter={/tool e-mail send
{... from=” ridks@mt.lv”  server=” 159.148.147.198”  body=” router up”
{... subject=” router at second floor is up”  to=” rieks@latnet.lv”}
[admin@mikrotik] system script>
[admin@mikrotik] system sctipt> /tool netwatch
[admin@mikrotik] system netwatch> add host=10.0.0.215 timeout=999ms \
\... interval=20s up-sctipt=e-up down-script=e-down
[admin@mikrotik] tool netwatch> print detail

Flags: X - disabled
    0 host=10.0.0.215 timeout=998ms interval=20s since=feb/27/2003 14:15:36
    status=up up-script=e-up down-script=e-down

[admin@mikrotik] tool netwatch>
```

2、 图形显示（graphing）

[键入文字]

Graphing 是一个监视工具，用于监视 RouterOS 在一段时期内不同参数的情况。

需要功能包：system，RouterBOARD（optional）
等级需要：Level 1
操作路径：/tool graphing
Graphing 工具可以显示的图形为：

RouterBOARD 健康状态（电压和温度）
资源使用（CPU，内存和硬盘使用 disk usage）
通过 interfaces 的传输情况
Simple queues 中的传输情况

Graphing 由两部分构成 - 第一部分是手机数据信息，另一部分在一个 Web Page 中显示数据访问图形的地址为 http://[router_IP_address]/graphs/或是通过浏览 RouterOS 的默认网页进入。

在路由器中数据收集每间隔 5 分钟，但保存到系统驱动中是每隔一个 store-every 时间，当重启路由后，显示的信息在重启前为最后一次存储到磁盘中的数据。

RouterOS 每一个项目产生四种图标 generates four graphics for each item：

“daily”graph（5 minute Average）
“weekly”graph（30 minute Average）
“monthly”graph（2 hour Average）

从一个网路去访问每个图形，可以通过 allow-address 指定这个网络的访问项目。

操作路径：/tool graphing

属性描述

Store-every（5min | hour | 24hours；默认：5min）- 多长时间将信息存储到系统驱动上。

存储信息到系统驱动上为每小时：

```
/tool graphing set store-every=hout
[admin@MikroTik] tool graphing> print
  Store-every: hour
[admin@MikroTik] tool graphing>
```

健康情况

操作路径：/tool graphing health

这个子目录提供关于 RouterBOARD 的电压和温度的信息，但你必须安装 RouterBOARD 功能包和使用 RouterBOARD：

属性描述

Allow-address（IP 地址/掩码；默认：0.0.0.0/0）- 运行访问图形显示的网络地址段
Store-on-disk（yes | no；默认：yes）- 是否将信息存储到系统驱动上，如果选择为‘no’，这些信息将存储到 RAM 中，重启后会丢失

接口图标

操作路径：/tool graphing interface

显示有多少流量传输在一段时期内通过了一个 interface

属性描述

Allow-address（IP 地址/掩码；默认：0.0.0.0/0）- 运行访问图形显示的网络地址段，被允许的地址可以试着打开 http://[router_IP_address]/graphs/，如果没有允许将无法看到
Interface（名称；默认：all）- interface 的名称
Store-on-disk（yes | no；默认：yes）- 是否将信息存储到系统驱动上，如果选择为‘no‘，这些信息将存储到 RAM 中，重启后会丢失

仅 192.168.0.0/24 的网段监视通过 ether1 的传输情况，并将信息写入到磁盘中：

```
[admin@MikroTik] tool graphing interface> add interface=ether1
Allow-address=192.168.0.0/24 store-on-disk=yes
[admin@MikroTik] tool graphing interface> print
Flags: X - disabled
 #   INTERFACE   ALLOW-ADDRESS       STORE-ON-DISK
 0   ether1      192.168.0.0/24      yes
[admin@MikroTik] tool graphing interface>
```

带宽 Graphing

操作路径：/tool graphing queue

在这个子选项中尼可以指定一个队列/queue simple 到图形显示中去。

属性描述

Allow-address（IP 地址/掩码；默认：0.0.0.0/0）- 运行访问图形显示的网络地址段，被允许的地址可以试着打开 http://[router_IP_address]/graphs/，如果没有允许将无法看到
Allow-target（yes | no；默认：yes）- 允许在/queue simple target-address 中那些 IP 段访问 graphing Web
Simple-queue（名称；默认：all）- 要监测的 simple queue 名称
Store-on-disk（yes | no；默认：yes）- 是否将信息存储到系统驱动上，如果选择为‘no‘，这些信息将存储到 RAM 中，重启后会丢失

添加一个 simple queue 到图形列表，simple-queue 名称为 queue1，限制访问网段，并存储相关信息到磁盘中：

```
[admin@NAT tool graphing queue> add simple-queue=queue1
allow-address=192.168.0.0/24 store-on-disk=yes
```

资源图表

操作路径：/tool graphing resource

提供路由器在一段时期内资源使用情况：

- CPU usage
- Memory usage
- Disk usage

属性描述

Allow-address（IP 地址/掩码；默认：0.0.0.0/0）- 运行访问图形显示的网络地址段，被允许的地址可以试着打开 http://[router_ip_address]/graphs/，如果没有允许将无法看到
Store-on-disk（yes | no；默认：yes）- 是否将信息存储到系统驱动上，如果选择为‘no’，这些信息将存储到 RAM 中，重启后会丢失

添加允许监视者的 IP 地址段为 192.168.0.0/24：

```
[admin@NAT] tool graphing resource> add allow-address=192.168.0.0/24
store-on-disk=yes
[admin@NAT] tool graphing resource> print
Flags: X - disabled
 #  ALLOW-ADDRESS     STORS-ON-DISK
 0  192.168.0.0/24    yes
[admin@NAT] tool graphing resource>
```

3、Bandwidth-text 带宽测试

带宽测试用于监测远程 Mikrotik 路由器的吞吐量（有线或无线），从而去发现网络瓶颈。

协议属性

TCP 测试使用 TCP 协议标准，根据 TCP 算法得出有多少包延迟，被丢弃和其他 TCP 算法特性。关于内部速度设定和状态分析请查看 TCP 协议。吞吐量的统计是用来计算整个 TCP 数据流的大小。TCP 内部链接的大小和使用没有包含在吞吐量的统计中。因此当在测算吞吐量时，这个统计并不像 UDP 协议标准所要求的。通过这样设置，便可以得到近似最大吞吐量。

注：bandwidth text 会使用所有可获得的带宽（by default），并做可能冲击网络的使用性。

Bandwidth text 比较占资源。如果需要测试路由器的真实吞吐量，你应该运行 bandwidth text 通过所测路由器。

注：如果用 UDP 协议，那么 bandwidth text 所测的数据是 IP header+UDP header+UDP。如果用 TCP 协议，那么 bandwidth text 所测的数据仅为 TCP 数据。（不包含 TCP 数据报头和 IP 数据报头）。

Server 配置

操作路径：/tool bandwidth-server

属性描述

Allocate-udp-ports-from - 分配 UDP 端口
Authenticate（yes | no；默认：yes）- 通信要求验证客户端（通过账号和密码）
Enable（yes | no；默认：no）- 为客户端启用连接
Max-sessions - bandwidth-text 最大的客户端连接数

Bandwidth 服务器：

```
[admin@mikrotik] tool bandwidth-server> print
                 Enabled: yes
                 Authenticate: yes
  Allocate-udp-ports-from: 2000
                 Max-sessions: 10
[admin@mikrotik] tool>
```

查看会话连接

```
[admin@mikrotik] tool> bandwidth-server session print
 # CLLENT        PROTCOL    DIRECTION   USER
```

```
 0 35.35.35.1     udp          send         admin
 1 25.25.25.1     udp          send         admin
 2 36.36.36.1     udp          send         admin
[admin@mikrotik] tool>
```

开启没有客户端的 bandwidth-test 服务器

```
[admin@mikrotik] tool bandwidth-server> set enabled=yes authenticate=no
[admin@mikrotik] tool bandwidth-server> print
                 Enabled: yes
            Authenticate: no
 Allocate-udp-ports-from: 2000
           Max-sessions: 10
[admin@mikrotik] tool>
```

Client 配置

操作路径：/tool bandwidth-text

属性描述

（IP address）- 目标主机 IP 地址
Assume-lost-time（time；默认：0s）- 设定如果 bandwidth server 无响应多久后丢弃连接
Direction（receive/transmit/both；默认：receive）- 测试方式
Do（name | string；默认：“”）- 脚本代码
Duration（time；默认：0s）- 测试时长
 0s - 测试时间没有被限制
Interval（time：20ms..5s；默认：1s）- 报告间隔时间（秒钟计算）
Local-tx-speed（整型；默认：0）- 本地发送最大速率（bits per second）
 0 - 没有速率限制
Local-udp-tx-size（整型：40..6400）- 本地 UDP 发送最大数据包
Password（text；默认：“”）- 测试的密码
Protocol（UDP | TCP；默认：UDP）- 使用的网络协议
Random-data（yes | no；默认：no）- 如果随即数据设置为 yes，bandwidth 测试数据包的有效载荷，将有不可随机数据流，使连接利用数据压缩，将不会扭曲结果（如果较低性能的 CPU，random-data 应设置为 no）
Remote-tx-speed（整型；默认：0）- 远端接收测试的最大速率（bits per second）
 0 - 没有速率限制
Remote-udp-tx-size（整型：40..6400）- 远端 UDP 发送最大数据包
User（name；默认：“”）- 远程用户名

在 10.0.0.211 主机上运行 15 秒发送和接收 1000-byte UDP 数据包的带宽测试，用户名为 admin。

[键入文字]

```
[admin@mikrotik] tool> bandwidth-terx 10.0.0.211 duration=15s diretion=both
\... size=1000 protocol=udp user=admin
                  Atatus: done testing
                Duration: 15s
              Tx-current: 3.62Mbps
    Tx-10-second-average: 3.87Mbps
        Tx-total-average: 3.53Mbps
              Rx-current: 3.33Mbps
    Tx-10-second-average: 3.68Mbps
        Rx-total-average: 3.49Mbps
[admin@mikrotik] tool>
```

4、 Torch（即时通信监听）

即时通信监听被称为 torch 它是用来监听正在运行的一个接口的通信情况。你可以监视通过协议名、源地址、目的地址、端口来分类监视通信情况。Torch 能显示出你已经关闭和发送接受的每个数据流的情况。

操作路径：/tool torch

属性描述

（name）- 用于监视的接口名
Dst-address（IP address/netmask）- 目的地址和子网掩码是用来通信，任意的目的地址是：0.0.0.0/0
Freeze-frame-interval（time）- 屏幕输出暂停的立即时间
Port（Name | 整型）- 端口的名
Protocol（any | any-ip | DDP | egp | encap | ggp | gre | hmp | ICMP | idpr-cmtp | igmp | ipencap | ipip | ipsec-ah | ipsec-esp | iso-tp4 | OSPF | pup | rdp | rspf | st | tcp | udp | vmtp | xns-idp | xtp）- 协议名
Any - 任何以太网和网络协议
Any-ip - 任何网络协议
Src-address（IP address/netmask）- 源地址和子网掩码是用来进行通信，所有源地址是：0.0.0.0/0

注：如果规定了一个特殊的端口，仅有 TCP 和 UDP 协议将被过滤，这就是说协议包含 any any-ip tcp udp。除了上行和下行，你已经用命令指定输出（例如，你将得到协议族仅是以防万一如果协议被明确指出）。

下面的例子是利用 telnet 协议监视通过 ether1 接口的通信情况：

```
[admin@mikrotik] tool> torch ether1 port=telnet
  SRC-PORT             DST-PORT          TX          RX
```

```
                1439          23 (telnet)          1.7kbps     368bps

 [admin@mikrotik] tool>
```

IP 协议通过 ether1 接口所显示的情况：

```
 [admin@mikrotik] tool> torch ether1 protocol=any-ip
  PRO..  TX           RX
  Tcp    1.06kbps     608bps
  Udp    896bps       3.7kbps
  Icmp   480bps       480bps
  Ospf   0bps         192bps

 [admin@mikrotik] tool>
```

IP 协议作用于 10.0.0.144/32 这台主机链接 ether1 接口所显示的情况：

```
 [admin@mikrotik] tool> torch ether1 src-address=10.0.0.144/32 protocol=any
  PRO..  SRC-ADDRESS     TX         RX
  Tcp    10.0.0.144      1.01kbps   608bps
  Icmp   10.0.0.144      480bps     480bps

 [admin@mikrotik] tool>
```

Tcp/udp 协议通过 ether1 接口所显示的情况：

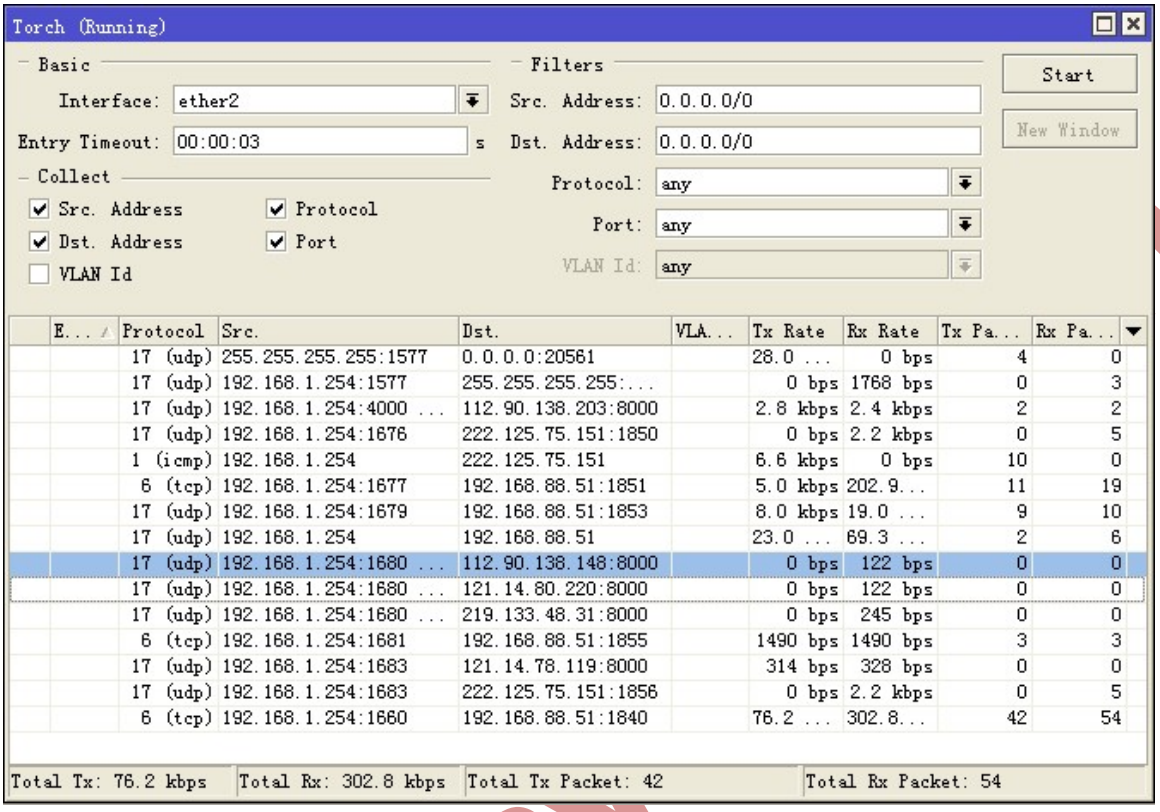
```
 [admin@mikrotik] tool> torch ether1 protocol=any-ip port=any
  PRO..  SRC-PORT        DST-PORT              TX          RX
  Tcp    3430            22 (ssh)              1.06kbps    608bps
  Udp    2812            1813 (radius-acct)    512bps      2.11kbps
  Tcp    1059            139 (netbios-ssn)     248bps      260bps
 [admin@mikrotik] tool>
```

禁止 QQ 连接到服务器

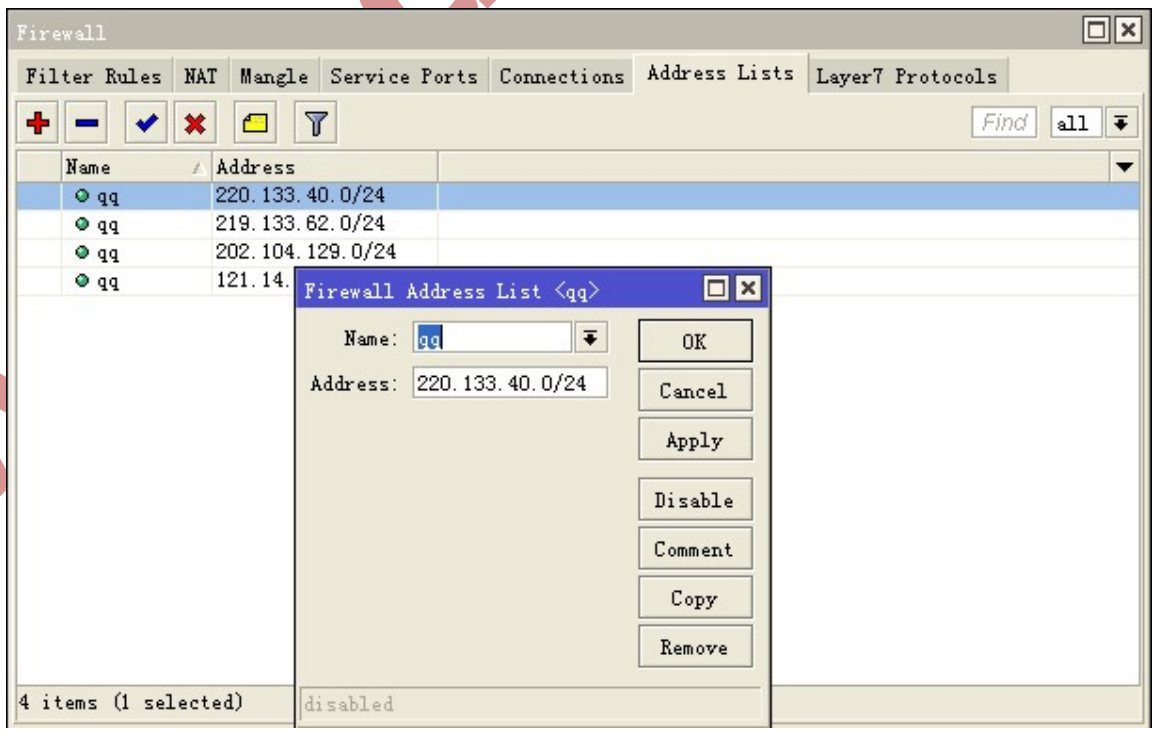
通过端口禁止 QQ 连接时没有作用的，因为 QQ 可以更改连接端口，最好的办法是通过禁止 QQ 连接相应的 QQ 服务器，实现对 QQ 的上网连接。

首先我需要通过 RouterOS 的工具查找每次 QQ 连接服务器的 IP 地址，在这里我们使用的是 RouterOS 自带的 torch 工具，torch 是用于监测相应网卡的数据连接状态、协议、端口和流量的工具，在/tool 中可以找到 torch。

我们需要监测 interface 为内网网卡，QQ 连接通常使用 8000 端口连接服务器，我们通过查看 dst-port 为 8000 的连接，当 8000 端口无法连接时，QQ 会使用 80 端口，如下图：



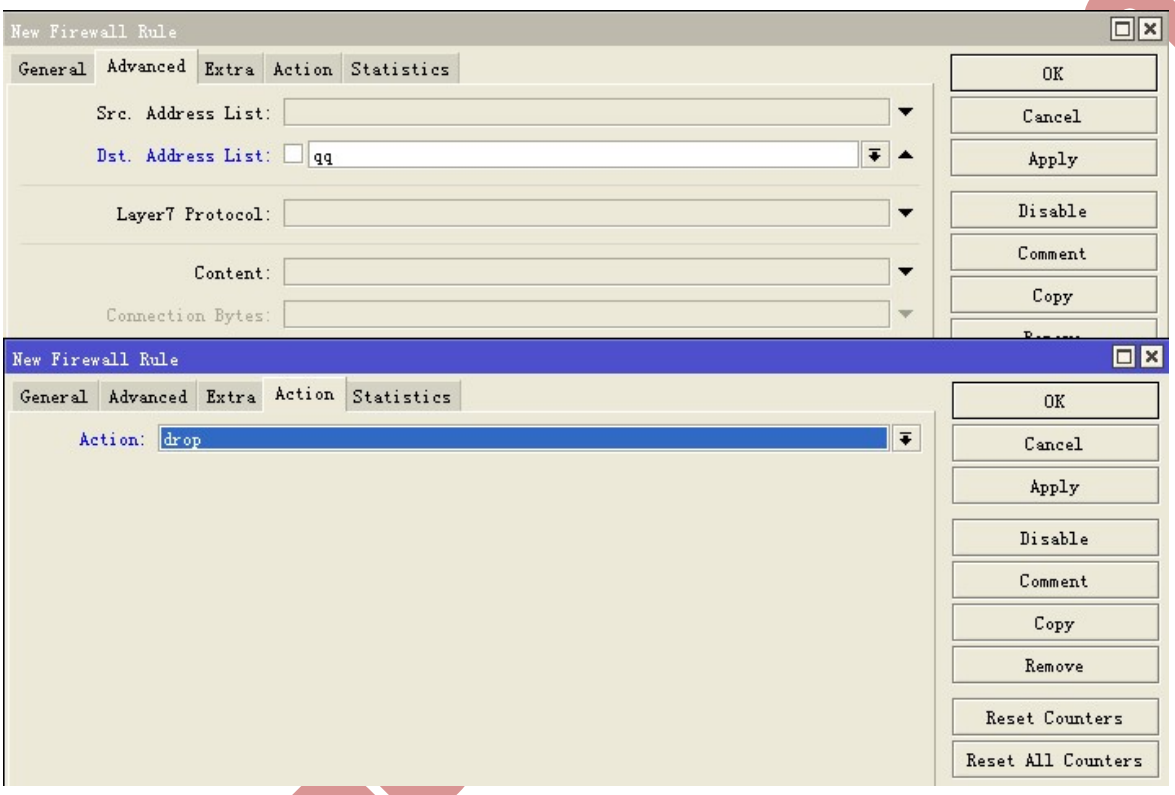
当看到 8000 端口的连接的 dst-address，可以判断为 QQ 服务器，我通过 ip firewall address-list 填写 QQ 服务器的 IP 地址，比如 220.133.40.11 是 QQ 服务器地址，而添加到 address-list 名字取为 qq，填写 address：220.133.40.0.24，用于规则调用：



当添加完 QQ 服务器 IP 地址后，在 ip firewall filter 的 forward 链表添加过滤规则：

```
/ip firewall filter add chain=forward dst-address-list=qq action=drop
```

Winbox 操作如下：



通过 torch 得到 QQ 服务器的 IP 地址，需要反复测试，知道 QQ 不能连接到服务器为止。

5、User Manager 操作

User manager 是一套类 radius 管理系统，它主要应用于：

Hotspot 用户管理；
PPP（PPTP/PPPoE）用户管理；
DHCP 用户管理；
无线用户管理；
RouterOS 登录账号管理

User manager 操作主要通过 Web 界面进行管理，方便的添加、删除和查询用户信息，现在的 user manager 仍然在开发阶段，许多功能仍然在补偿。使用 user manager 最少需要 32M 内存和 2M 硬盘空间。

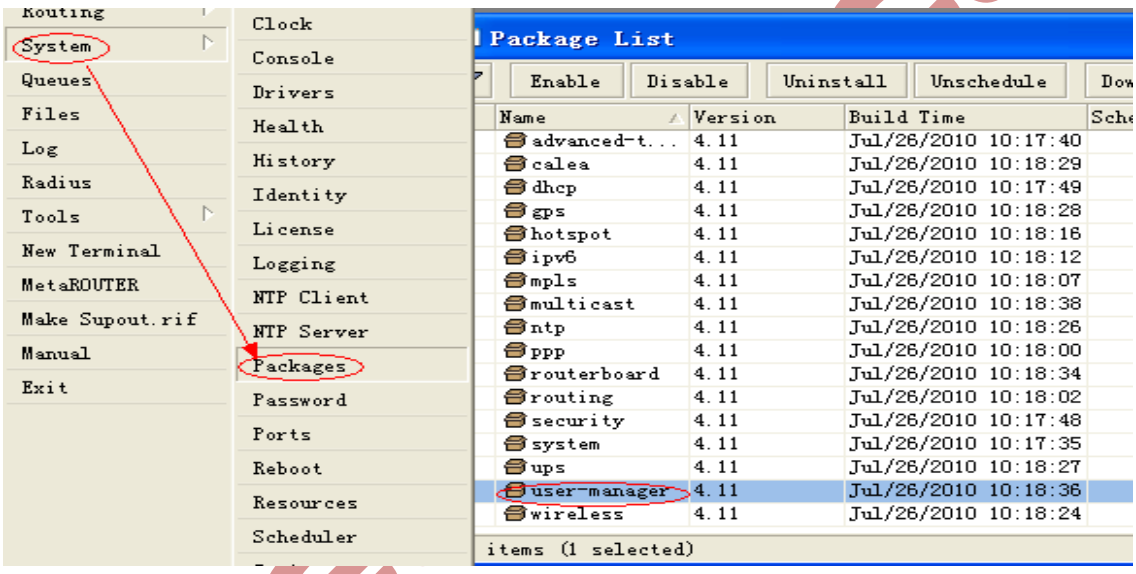
在 RouterOS v3.0 修改为在线用户许可方式：

- Level 3 - 10 active users
- Level 4 - 20 active users
- Level 5 - 50 active users
- Level 6 - unlimited active users

在 Hotspot 中通过设置 user manager 认证上网

初始化 user manager

首先确定你是否安装了 user manager 的功能包，我们可以在 RouterOS 中的/system packages 中查询到：

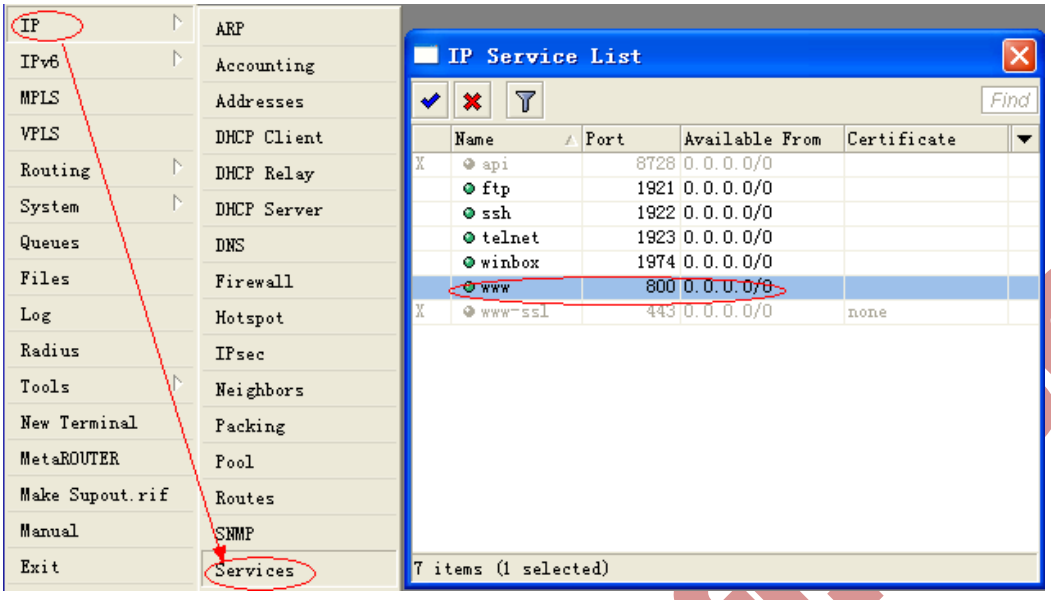


正确安装 user manager 后，我们进入/tool usermanager 的目录配置相应的参数和启用管理账号，我们需要进入命令行操作，才能初始化 user manager 的管理账号，如下面我们进入 customer 目录添加 user manager 的客户账号：

```
[admin@mikrotik] > tool
[admin@mikrotik] tool> user-manager
[admin@mikrotik] tool user-manager> customer
[admin@mikrotik] tool user-manager customer> add login=yus password=yus
```

登录名为 yus，登录密码为 yus。

当我们设置好后，我们可以同 Web 页面登录到 user manager 的管理页面，我们将 RouterOS 的 www 端口设置为 800：



设置 www 端口为 800，是为了在设置 Hotspot 后，能通过 Web 页面正常访问 user manager。

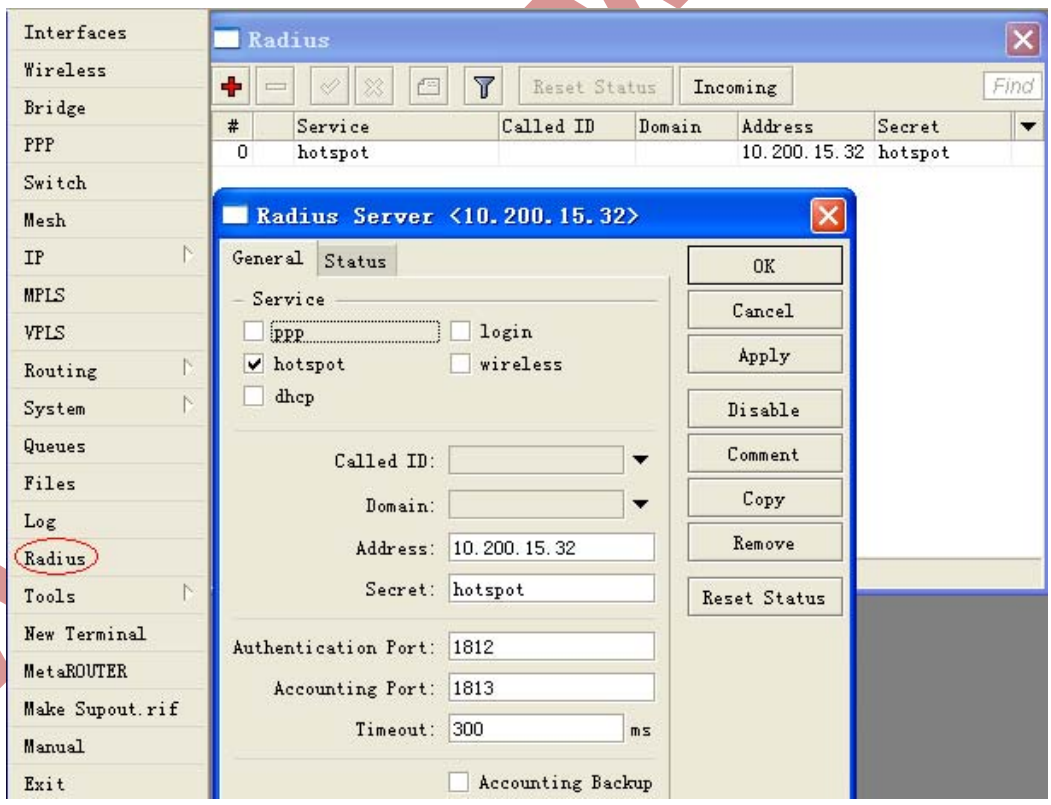
在设置完成后我们进入 IE 浏览器，打开 http://routerIP:800/userman 便可以访问 user manager 的登陆页面：

在这里我们的 RouterOS IP 地址为 10.200.15.32，通过 Web 方式即能登录到 user manager。

在完成 user manager 的初始化配置后，我们需要和 RouterOS 建立 radius 服务的连接，并且配置 Hotspot 的账号和类型通过 user manager 来认证。

设置 RouterOS 中的 radius

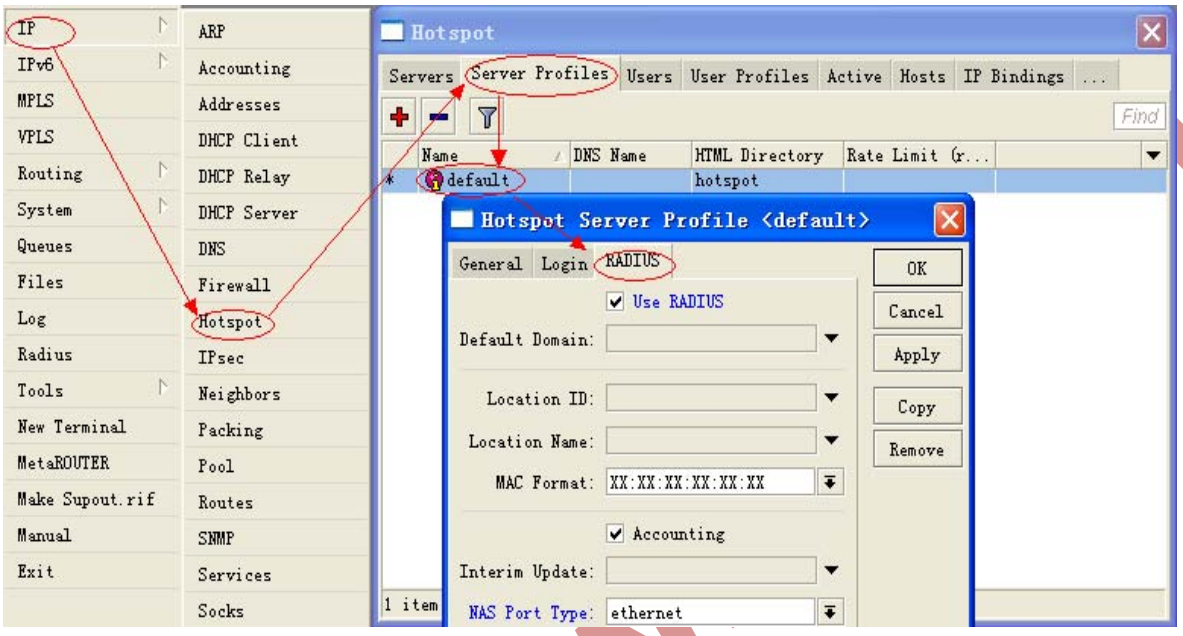
首先设置 RouterOS 中的 radius 参数和 Hotspot 的配置，进入路由器的 radius 目录设置 radius 服务器的 IP 地址和访问密码，并配置 Hotspot 需要通过 radius 认证：



这里是通过本来 radius 做认证，所以 address 输入的是本地的 IP 地址，并设置 secret 为 Hotspot。

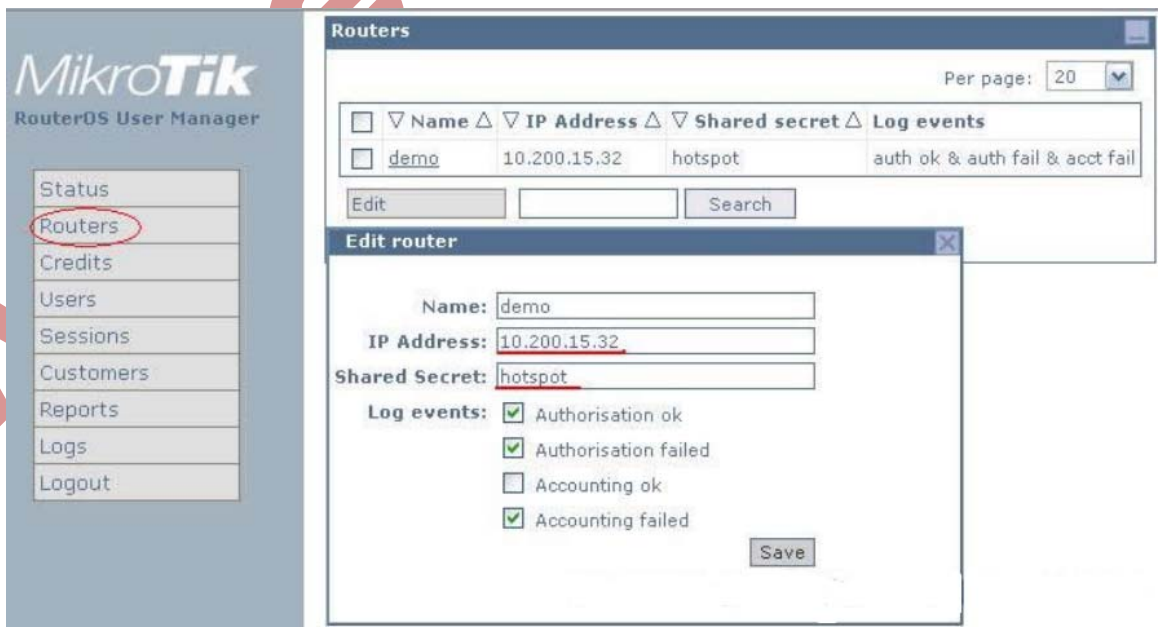
[键入文字]

然后进入/ip hotspot 目录，在 servers 的 profile 中配置 radius 服务：



设置完 Hotspot profile 的 radius 参数后，这样 RouterOS 的 radius 参数就配置完成，下面需要配置 user manager 的参数：

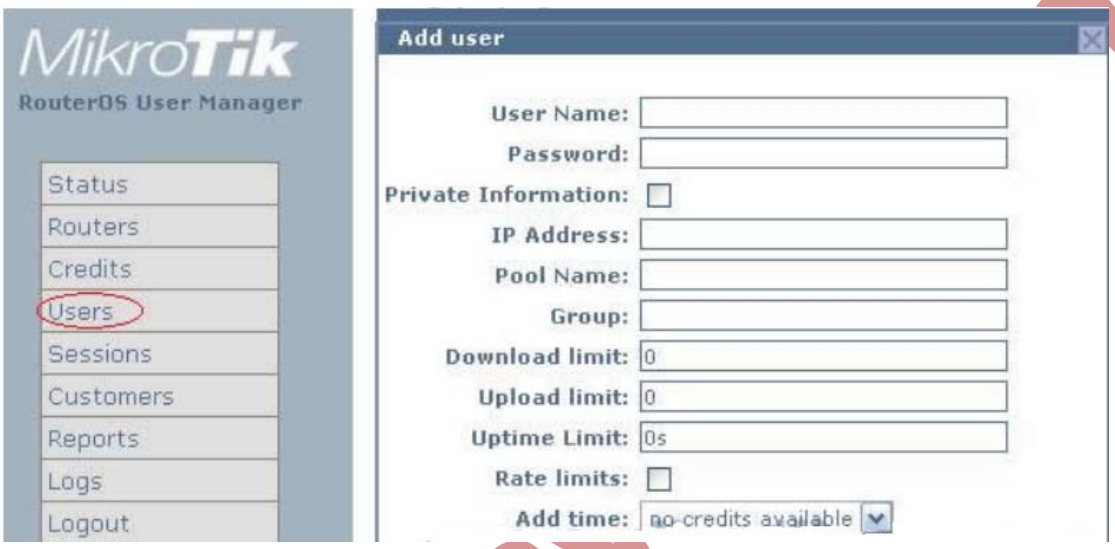
进入 user manager 中的 router 项配置与本地的 RouterOS 连接参数，我们添加一个项目，将名称取名为“Demo”，在 user manager 中同样的我们将 IP 地址设置为 RouterOS 的本地 IP，secret 为 Hotspot。在 router 项目中可以添加多个 radius 客户端，并能同时为多个 radius 客户端提供认证。



当 user manager 中的 routers 参数配置完成后，RouterOS 就可以和 user manager 相互通信传递用户认证信息了。

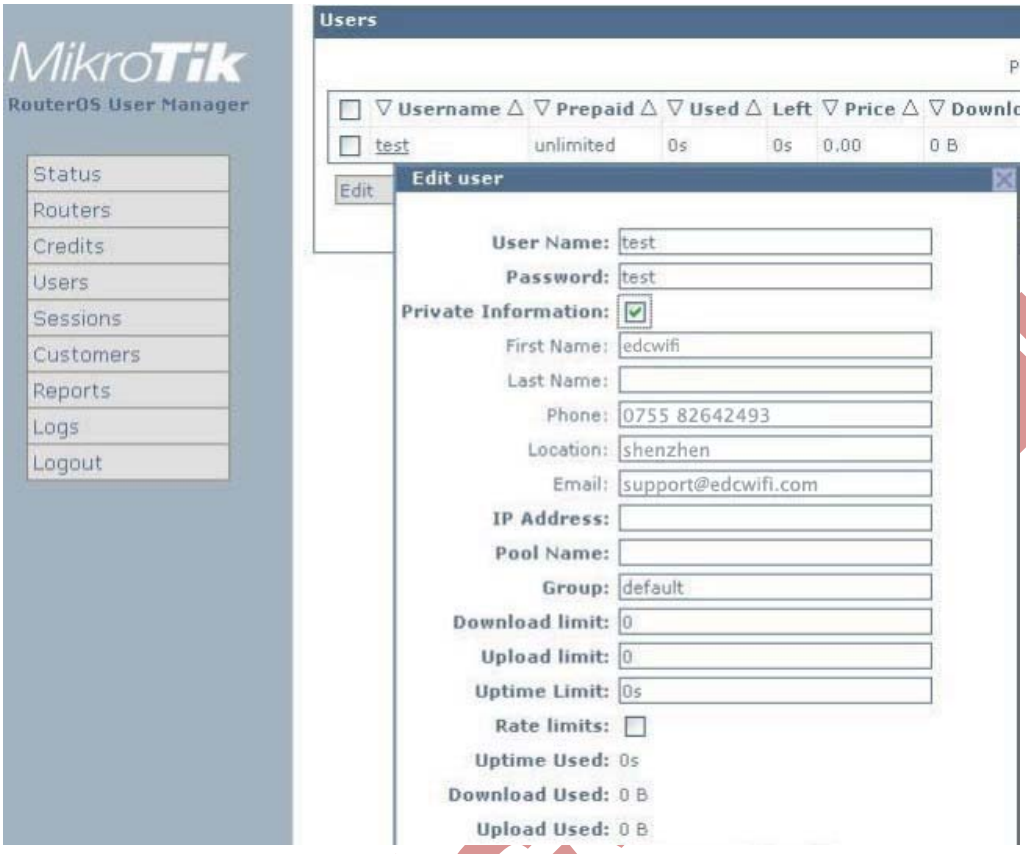
添加认证用户账号

最后就是我们在 user manager 的 users 项目中配置用户的账号：

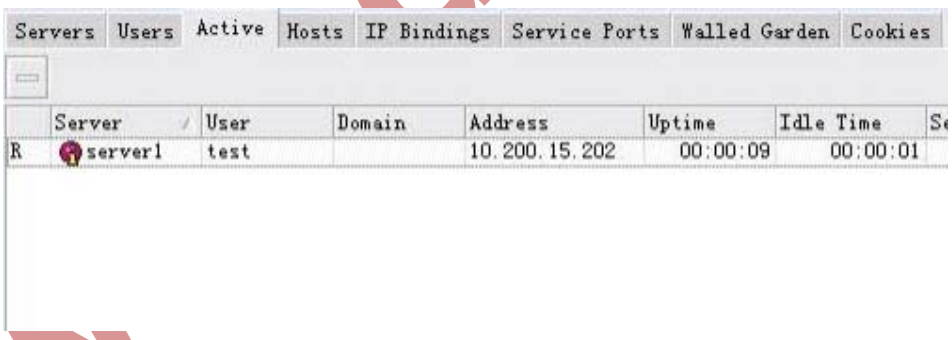


User name：用户账号名称
Password：用户密码
Private information：是否设置用户的个人信息资料
IP address：分配给用户的 IP 地址
Pool name：分配给用户的地址池（地址池从 RouterOS 中获取）
Group：设置 Hotspot 用户的 profile 规则，仅限 Hotspot 使用
Download limit：按照下行流量计费
Upload limit：按照上行流量计费
Uptime limit：按照在线时间计费
Rate limit：是否设置流量控制规则
Add time：添加时间控制规则

下面我们添加一个账号，名称为 test，密码：test，配置 group 为 Hotspot 上的默认规则 default。



现在我们可以通过 user manager 添加的账号，已经可以在 Hotspot 认证上通过，如下图：



如果是通过 radius 认证登陆到 Hotspot 上的，在该 test 登录行最前面会有一个“R”出现。

用户修改自己的密码

User manager 支持用户自主修改自己的密码，这样能让用户自己管理自己的账号，在这里需要用户登录到指定的路径去：http://routerIP:800/user

当用户登陆到该页面后，只有输入自己的用户名和密码，就可以登陆到设置页面，并修改自己的密码和个人信息资料：