Learn RouterOS

Second Edition

by Dennis Burgess

ISBN: 978-1-105-06959-8

Table of Contents

Table of Contents ... 4
Introduction ... 10
Who Should Use This Book ... 11
About the Author ... 12
Link Technologies, Inc ... 13
What is RouterOS? ... 13
How This Book is Organized ... 15
RouterOS Reference Version ... 15
Second Edition ... 16
Special Thanks ... 16
Credits ... 16
RouterOS Hardware ... 18
RouterBOARD Devices ... 18
Solar Power and RouterBOARDs ... 22
X86 Based RouterOS Systems ... 23
RouterOS Fiber Optic Interfaces ... 24
T1/E1 Interfaces ... 24
Supported x86 Hardware ... 25
RouterOS Licensing ... 25
Extended Frequency Licenses ... 27
Ways to Lose your RouterOS License ... 28
RouterOS Installation ... 30
Using NetInstall on RouterBOARD Products ... 31
ISO – CD Installation ... 38
DOM / Flash Card / Hard Disk Installation via NetInstall ... 40
Accessing RouterOS ... 42
RouterOS Access Methods ... 43
Default User and Password ... 43
Using Neighborhood Viewer ... 43
Using Telnet ... 45
SSH – Secure Shell Access ... 45
WebBox ... 48
WebFig ... 58
Using WinBox ... 59
Managing RouterOS ... 66
User Management ... 66
RouterOS Services ... 69
Working with Files ... 71

Logging ... 76
Flashfig ... 78
Basic RouterOS Setup ... 80
Configuring IP Addresses ... 80
Default Routes ... 82
DNS Caching / Service ... 82
DHCP-Client ... 85
DHCP-Server ... 86
DHCP-Relay ... 94
Masquerading - NAT ... 96
Home Router ... 98
Common Wireless Configurations ... 101
RouterOS and IP ... 106
Why Routing ... 106
IP Addresses ... 107
Interface ARP – Address Resolution Protocol Settings ... 107
IPv6 ... 110
Routing and Routes ... 116
Policy Based Routing ... 120
Firewalling ... 124
Traffic Identification ... 124
Rules ... 125
Understanding Connection States ... 125
Packet Flow in RouterOS ... 128
Chains ... 131
Address-lists ... 134
How to Match Data ... 136
Protecting Your Router ... 148
Protecting Networks ... 149
Using Mangle ... 156
NAT – Network Address Translation ... 161
Basic Interfaces ... 170
Ethernet ... 170
Virtual Ethernet Interfaces ... 174
Bridge Interfaces ... 176
Virtual LAN (VLANs) ... 183
Bonding ... 185
MESH ... 188
VRRP ... 191
Wireless and RouterOS ... 194
WIC – Wireless Interface Cards ... 194
Basic Configuration of Wireless Interface Cards ... 195

Wireless Tools 197
Wireless Operational Modes 202
Security Profiles (Securing your Wireless Connection) 205
Access Lists 209
Registration Table 211
Connection Lists 212
Area / Area Prefixes 213
Virtual Access Points 214
Nstreme 216
Nstreme Dual 217
Using WDS (Wireless Distribution System) 219
Wireless Link Optimization / Best Practices 222
NV2 231
Troubleshooting Wireless Links 236
Tunnels 240
EoIP 241
IPIP 243
PPP System 244
L2TP/PPTP Servers 252
SSTP 256
PPPoE Server 258
OpenVPN 262
IPSec 265
Choosing a Tunnel Type 269
Traffic Control 270
Identifying Queue Data 271
Hierarchical Token Bucket– HTB 272
Queue Types 275
Simple Queues 284
Bursting 285
Creating Advanced Queues 289
Controlling P2P (Peer-to-Peer) Traffic 293
Hotspots 296
Wireless and Hotspots 296
Paid Hotspots 297
Free Hotspots 297
RouterOS and Hotspots 298
Setup of a Hotspot Interface in RouterOS 300
Radius Client 312
Multiple Radius Servers 313
Troubleshooting Radius Client Issues 314
MPLS 316

Getting Started with MPLS ... 317
MPLS Configuration ... 319
VPLS ... 323
BGP based VPLS ... 331
RouterOS Extras ... 334
IP Accounting ... 334
M3P – MikroTik Packet Packing Protocol ... 335
IP Pools ... 335
Socks ... 336
NTP ... 336
Clock ... 338
System Identity ... 338
TFTP Server ... 338
Traffic-Flow ... 339
Web Proxy ... 340
Universal Plug and Play - UPnP ... 344
Store System ... 345
CALEA ... 346
MetaRouters ... 349
Non-RouterOS MetaRouters ... 350
RouterOS Tools ... 352
Neighbors ... 352
Logging ... 353
System Configuration Reset ... 354
Scripting ... 355
Auto Upgrades ... 356
Watchdog ... 357
Bandwidth Test Server ... 357
Bandwidth Test Client ... 358
E-Mail System ... 359
Using Fetch Commands ... 359
Graphing ... 360
PPPoE Scan ... 361
Packet Sniffer ... 361
Profiler ... 362
IP Scan ... 363
Dynamic Routing ... 364
If Installed vs. Always ... 364
RIP ... 364
OSPF ... 366
BGP ... 368
Routing Filters ... 371

BGP Configurations 372
The Dude NMS 378
    Installation 379
    Dude Agents 380
    Dude Layout 381
    Running a Server 382
    Resetting Configuration 382
    Menus and Options 383
    Server Configuration 384
    Admins 389
    Charts 389
    Devices 390
    Files 394
    Links 395
    Logs 396
    Network Maps 397
    Notifications 403
    Outages 406
    Probes 406
    Tools 408
User Manager 412
    Hardware / License Requirements 412
    Installation of User Manager 413
    Configuration of User Manager 414
    User Sign-Ups 421
    User Sign-In Page 421
    Active Sessions 422
    Vouchers 422
    V5 of User Manager 424
Command Line Interface 430
    Command Line Hotkeys 432
Quick Reference Guide 434
    NetInstall of RouterBOARD Products 434
    NetInstall your Flash / DOM / Hard Disk 435
    Creating a Active/Backup Bridged Auto-Fail Link 435
    Setup Transparent Web Proxy System 436
    Redirect Non-Paying Customer 436
    Per Connection Load Balancing 437
    Create a Private VPN 438
Appendix 440
    Features Only Available via Command Line Interface 440
Index 442

Introduction

MikroTik RouterOS is a routing software that has been growing in popularity extremely quickly. When it is combined with reliable, powerful hardware, RouterOS can quickly surpass many routers that are currently available on the market. Many businesses, Wireless Internet Service Providers (WISPs) and other end-users have found that the cost savings that RouterOS offers is the key to their business success.

In this book, we are going to give you both knowledge plus examples of configuration of the MikroTik RouterOS software. You will end up learning RouterOS, and have working examples that you can emulate and change to meet your needs. We will cover many aspects of the software, including MikroTik specific systems, Wireless Networking, Routing, as well as virtually all of the features included in the RouterOS software.

We are going to give you code examples, screen shots and real world application designs that you can do right on your own RouterOS system. These items will enable you to apply RouterOS work in your business, or company. You will gain the knowledge to use RouterOS as a router, wireless access point, client premise device, web caching system, and even a VPN (Virtual Private Network) server.

Who Should Use This Book

This book is designed as a reference guide. I want to help you learn direction on what features you need to use, and why. If you need to know what a feature or command does, you will need to use the command reference that MikroTik offers on their website at http://www.MikroTik.com. If you want to learn how to take these features and put them together , common best practices, as well as how to ways of configure systems to make them do what you want them to do, then this book is for you.

We will cover lots of topics, some are simple topics and we will show you the options you have, but more importantly, we will show you why to use the features! Some features are packed with comments and suggestions on how to use that feature in combination with other features, and why to use these feature combinations.

This second edition expands upon the existing content of the original book; however, RouterOS is constantly changing, being updated, and adding new features. It's always improving, and therefore, we need to keep updating this book to ensure that we give you the latest information!

About the Author

Dennis Burgess started learning about computers at a young age. Using a TRS-80 Dennis started using Basic programming to create small computer programs. At the age of 13 he started a multi-line BBS (Bulletin Board System), using small Dell computers and 9600 baud modems. He was introduced to networking by the need to network his BBS computers together. After high school, Dennis attended a local technical college and graduated with an Associate's Degree in Computer Electronics and Networking Technologies.

Mr. Burgess went to work for a number of consulting companies, focusing on servers, and wide-area networks. He designed and deployed a number of networks for law firms, construction companies and other small-to-medium businesses. He deployed Microsoft solutions as well as Cisco routers on a routine basis. During this time, Dennis obtained his Microsoft Certified Professional status, as well as his A+ Computer Technician, N+ Network Technician, and also became a Cisco Certified Network Associate or (CCNA).

After working for a number of years as an Enterprise network and server consultant, Mr. Burgess worked for a number of dealerships in the St. Louis area building a private network for their needs. During this time he started his first wireless Internet Service Provider. This company introduced him into the world of MikroTik RouterOS. The WISP needed a method to control bandwidth for subscribers, so they built their first RouterOS x 86 systems.

After realizing the power and performance of RouterOS, including well as using them in tower installations for 802.11b/g access-points in the WISP, he continued to use RouterOS to deploy a fully redundant virtual network for the group of dealerships he worked for. This network, still using RouterOS, is working as intended, since 2001.

Mr. Burgess, ended up selling his Wireless Internet Service Provider Company later, and focused on creating a company that could assist other WISPs, businesses and ISPs with RouterOS. Dennis's company, Link Technologies, Inc, is now a world-wide MikroTik consulting company. Consulting clients include small WISPs as well as Enterprises using RouterOS.

Link Technologies, Inc. also started producing the PowerRouter series of RouterOS devices after seeing a need for Enterprise-Class RouterOS routers. These 1U carrier-grade systems are designed with Ethernet routing, and support high-performance applications, and web caching as well. The PowerRouter 732 is also a homeland security approved device.

Link Technologies, Inc

Link Technologies, Inc was formed with the purpose of helping Wireless ISPs as well as providing high-quality consulting services for RouterOS systems. In the USA, available options for RouterOS systems administration and consulting services for were very limited to small home businesses were very limited and technician level admins trying to help out businesses and ISPs with RouterOS. Dennis formed Link Technologies, Inc. to help give these businesses with the needed level of technical support, engineering and consulting services that they needed.

Link Technologies, Inc offers multiple certified RouterOS engineers, MikroTik Certified Trainers, RouterOS Training Programs, as well as general network engineering, consulting and support. We are one of the largest MikroTik consulting companies in the world, with clients ranging from start-up WISP operations, to publicly traded enterprises with over 35,000+ end-users.

In addition to On top of MikroTik, we also offer business support, Motorola Canopy, Cisco, Microsoft, mail servers, DNS servers and can help you with just about any type of consulting services that you may need for your networking business.

When you need any type some form of RouterOS consulting, engineering or training, be sure to contact us. We have several engineers' on-staff who can assist you!

Link Technologies, Inc.
House Springs, MO 63051
http://www.linktechs.net
Support@linktechs.net
314-735-0270

What is RouterOS?

Simply put, it is an infinitely configurable routing software package. This software allows you to use common hardware to perform high-end routing applications. MikroTik creates this software, as well as many different hardware platforms to run the software on. These industrial hardware platforms provide you many options including ultra-low cost business and home devices, all the way to core routing functions of large Internet providers and enterprises.

So what can you do with RouterOS? It can do virtually anything when it comes to Internet addressing and data traffic management. In the world of IP routing, there is not much that RouterOS cannot do! Many routers and network devices only perform certain functions. One device may be a PPPoE server/concentrator. Another device may control bandwidth and the way the data flows across your network. Yet another device may do caching of the data that flows to save bandwidth. All of these devices can add up in costs, not only the upfront hardware costs, but the upkeep, the maintenance, and the system administration for to understand each device.

RouterOS contains all of the above mentioned features! With all of this power in one device, you can immediately see the cost savings just in the initial hardware costs. Business owners now have to take a look at a cost-saving system that has the same reliability and performance that they are accustom to in more expensive hardware. In some cases, RouterOS devices and software can be less than one-quarter of the cost of similarly capable devices yet have more features than those more expensive devices.

How This Book is Organized.

The reason for writing this book to market is simply that there is nothing else that is written in non-tech-speak for RouterOS. I wanted to bring you a book that you can use as a reference for your own specific needs as well as to give you the ability to configure your own RouterOS system. As you know, there is already a command line reference and other support topics including some very good detailed topics on-line with MikroTik, but these topics simply don't give you the enough details about how to use RouterOS, nor how useful it can be when it is used for in your business. Hence, what you are now reading was created.

With all of the features of RouterOS it's very hard to create a book that is perfectly organized. I have noted when you should check other sections for other related topics. I have also included basic configuration instructions for you when possible even inside unrelated sections. I would recommend using either the index and/or the table of contents whenever possible. The index often will give you several references for the different facets of the same topic.

RouterOS Reference Version

In the second edition of "Learn RouterOS", we will use version v5.1 as a reference version. Some of the screen shots you will see come from the original first edition of "Learn RouterOS" but will still apply even in v5.x.

I have been asked how much has changed from version three to version five. The simple fact is that most of the improvements are either have been inside the interworking operating system of RouterOS or are new features. Most of the original feature set is still configured the same way as it was in v3.

Second Edition

This is the second edition of Learn RouterOS. The previous version, even though mostly still accurate, has become outdated. This version includes all information contained in the first edition book but I have updated the information, added new sections for new features, and improved the topics wherever possible.   New features as well as corrections were my priority, as well as updating sections requested with more information.

Special Thanks

I wished to thank everyone that made this book possible. The second edition also has notes, comments and updated content from several individuals that sent their copies back to me in the hopes that their requests would be fulfilled. I have looked over these and, where possible, made updates, corrections and improvements based on these recommendations. Thank you very much for the constructive comments that you provided.

Credits

Some images, graphics, and other content have been used from the wiki.MikroTik.com website. These images are copyright of their respective owners and have been licensed under the terms and conditions between said companies, and the author of this book. We appreciate the ability to use these fine examples and images.

RouterOS Hardware

RouterOS works on several different types of hardware. MikroTik produces their own hardware based on a single board computer approach, called RouterBOARDs. RouterBOARDs come in a number of different CPU types, number of Ethernet ports, wireless slots, memory configurations, and design types. RouterBOARDs can cost under $49 USD, and up to several hundred dollars depending on the hardware. These RouterBOARDs are specifically created for RouterOS software, and even come with RouterOS already installed, licensed and ready to use.

RouterBOARD Devices

To the right is a RouterBOARD 433AH. This board includes a 680 MHz processor, three 10/100 Ethernet Interfaces and three M-PCI Slots. This unit also includes a Micro-SD slot for web caching and other storage functions, as well as power-over-Ethernet support, and a 9-pin serial connection for console access.

MikroTik is constantly developing new products, so be sure to ask your MikroTik distributor, or sales channel about the latest products and where to use them. Experienced Engineers will know what board to use for what purpose. A big mistake many make is using underpowered equipment.

At the time of this writing, there are a number of board series in production. MikroTik's current main RouterBOARD is the 400 series. A number of versions exist, the 411 includes a RouterOS Level 3 license, one Ethernet and one M-PCI slot. This is great if you wish to add your own radio card. The RouterBOARD 433, as shown to the right, includes three Ethernet and Mini-PCI Slots.

MikroTik makes several boards, and typically follows a common naming scheme. Most RouterBOARD products will include three numerical digits followed by several letters depending on options. The first number is the RouterBOARD series number, so in the case of a 433, it would be a 400 series board. The second is the number of Ethernet interfaces and the last number is the number of mPCI slots on-board. In our previous example of the RB433, this would be a 400 series board, with three Ethernet interfaces and three mPCI slots.

The options on these boards include lettering such as A, AH, R, U and G. The AH option includes the higher end CPU at 680 MHz instead of the standard at 300 MHz. The AH boards also include a level 5 RouterOS license as well, and also include a Micro-SD memory card slot as well. The letter G stands for Gigabit Ethernet ports, but is typically only found on the 450 series boards. R stands for an integrated b/g radio. This means that they have an on-board, non-removable wireless radio card. Note though that these only run 2.4 GHz. Boards with the U option mean that they include a USB port as well.

Some RouterBOARDs include an A option. These include a Level 4 license that allows them to run as an access point. Typically you will find this option only on boards that are designed for CPEs or clients, such as the RB411. These typically come with a Level 3 RouterOS license. The following table summarizes these options.

Option Letter	Option
A	Includes a Level 4 RouterOS License
AH	Includes 680 MHz CPU
U	Includes USB Port(s)
R	Includes Integrated B/G 2.4 GHz Radio
G	Includes Gigabit Ethernet Interfaces

The RouterBOARD 600 is considered an Extreme Performance Access-access point, providing three Gigabit Ethernet ports as well as four M-PCI slots for wireless connectivity. This unit runs a network processor that is much faster than the Atheros CPU on the 400 series boards. This unit also contains two compact flash slots for storage needs. One could be used for Web caching data, and another could be used to store Dude or User Manager Data. If you plan to run 802.11n you will typically need a board like this with GigE interfaces to use as the 802.11n protocol provides more allows for greater than 100 Mbps UDP throughput. Without the GigE interfaces, you will have a hardware limit at your Ethernet ports. Note though, that the RB600 has been replaced with the RB800 series board, and is no longer is in production.

The RB800 series is the replacement for the RB600. This board includes four mPCI slots, on-board mPCI-e slot, compact flash card slot, three Gigabit Ethernet ports, as well as two daughterboard ports. The RB800 also has a 800 MHz network processor, and two fan power headers for connecting either 5.5v or 3.3v DC fans. This board also draws more power than other RouterBOARD products at a max of 35 watts.

For core routing, with four Gigabit Ethernet interfaces as well as a rack-mountable case, you can purchase a RouterBOARD 1000 or 1000U. The U version is a rack-mountable model. This system is also based on a high performance network CPU running at 1333 MHz. You can also use compact flash storage cards, plus you have the ability to add more RAM via a SODIMM slot. This unit also comes with a level 6 RouterOS license, included with the cost of the hardware. Note that the RB1000 and RB1000U have been discontinued and are no longer in production.

The RB1100 is a replacement for the RB1000 and RB1000U. This RouterBOARD is designed with an included Rack Mountable case for indoor usage. Included are 13 Gigabit Ethernet interfaces, and on-board microSD card slot. The Processor is the same as the RB800, an 800 MHz network processor. It does have 512 Mbps of RAM though, as well as two switch groups. See Switch Groups for more information about their operation.

MikroTik also created the RB450 and RB450G, which these were Ethernet only (both 10/100 and GigE) desktop routers designed for home/office use. These products were discontinued shortly after the RB750 and RB750G units came out. The 700 series is a complete desktop router that includes a plastic desktop case, and power supply in a retail box. This is the only product packaged like this. This series has a 680 MHz CPU, and either five 10/100 Ethernet ports, or five Gigabit Ethernet ports depending on the model. These units also sport ultra-low retail pricing at $39.99 and $69.99 for each respective model.

RouterBOARDs all contain an on-board NAND. NAND is basically flash memory, just like your USB stick or Compact Flash card. This is an on-board chip on the RouterBOARD, giving the RouterBOARDs a non-removable flash memory area to load the operating system, in this case, RouterOS on. Most of the RouterBOARD products will have 64 megabytes of NAND storage or more, more than enough for RouterOS, its configuration, as well as typical files associated with RouterOS.

You can find out more information about current MikroTik RouterBOARD hardware, specifications, and details at http://www.RouterBOARD.com. You can also contact your local MikroTik Distributor.

Solar Power and RouterBOARDs

I have had quite a few requests about how to use RouterBOARDs with Solar systems; therefore I wanted to give you a few pointers. The key is power consumption; the newer RouterBOARDs, specifically the 400 series, are the most common boards used for solar powered sites. Most sites are powered by battery arrays at Ethernet 12, 24, or 48 volts. The 400 series of devices run from 10V to 28V DC power. When you install your RouterBOARDs with a long Ethernet run you will assume there will be some voltage drop, and you can do a web search on how to calculate this. If you are not doing a long Ethernet run, then 12 volt power may work out for you.

MikroTik also has an ultra-low wattage board; the 411R. This board only requires 5.6 watts of power and has an integrated b/g radio card. Recently, MikroTik has released several new boards, all with low power requirements. The RouterBOARD SXT, an integrated point to point or CPE MIMO solution, has a maximum power consumption of 7 watts. The OmniTik, an integrated MIMO 5 Gig solution only consumes 9 watts at most as well. The RouterBOARD Groove units consume no more than 4 watts. As you can see, the wattage used by this gear is great to use with solar powered setups.

If I had my choice, I would like to run 18-20V. The reason is that as the batteries drain, the voltage drops, and if you are running a 12-volt source, you will quickly drop below 10 volts and the RouterBOARDs will stop running. If you wanted to use 48-volt power, the RouterBOARD will not take that voltage that high so that won’t work either.

Some people have asked about using 24 volt solar systems. On a long Ethernet run this will work, but on a short run you have to take into consideration one other fact. Most of the solar charging controllers will output 26.5 volts or higher, so when you are running on the 24 volt batteries, and then the solar array is charging them, the voltage is higher than 24 volts. We have seen the voltage spike higher than what the RouterBOARDs are designed for so they power off to prevent damage from overvoltage. So I like to run a bit lower than 24 volts and a bit higher than the 12 volt systems as well. If your only choices are 12 and 24 volt, then run 12 volt!

Regardless, RouterBOARDs can run great on solar setups, consuming only 35 watts at max. A single car 12 volt battery can run a single board for several days without issues! Design the system correctly, and it can run for a long time! We have some solar-powered systems deployed and have never had to do more than to change batteries every few years.

X86 Based RouterOS Systems

The same software is available for x86 systems. X86 systems are the same hardware that common PCs and computers are based on. You can even load RouterOS on a basic computer, one that you may have in your home or office. Most of the features however, are based on a number of interfaces. With multi-port Ethernet cards and wireless cards available on the market as well as available through MikroTik, you can make an x86 RouterOS system with little effort and at low cost.

There are design issues with building your own systems. If you understand bus limitations, speeds and IRQ conflicts and how these items affect overall system performance, then you can build your own systems using off the shelf hardware just like any other computer would, typically creating a high-performance system.

There are a number of other companies out there as well; a simple Internet search will provide a number of results that sell complete, high-performance x86 systems with performance and reliability in mind. These systems are designed to use multiple bus channels, and high quality hardware to deliver reliable operation.

One such manufacturer is Link Technologies, Inc. Their PowerRouter series of devices provides gives you out-of-the-box, ready to run RouterOS performance. These are designed for high performance RouterOS systems take into account bus speed limitations, and even adding multi-core processors to further increase performance. These systems are designed to run a Routing Operating System. The PowerRouter 732, pictured above, includes seven Gigabit Ethernet ports, a Dual-Core CPU, along with options for SATA and SSD drives for storage. USB ports are also included for other data storage devices such as USB memory sticks, as well as cellular data cards. This offer this model in both AC and DC versions

Link Technologies also created an ultra-high-end system, called the PowerRouter 2200 series. These systems can run up to Dual Quad Core Xeon processors, and can support up to 22 GigE interfaces, including SFP interfaces that you can use fiber modules with. The 220 series also supports dual hot-swappable power supplies as well.

RouterOS Fiber Optic Interfaces

RouterOS does not have specific support for fiber optic interfaces. You won't get any data on the cabling; however, with the Ethernet Chipset support that RouterOS does offers, there are supported options that enable you to use for fiber optic interfaces. The PowerRouter 732 offers dual fiber interfaces in the back of the unit as a supported option. These interfaces are SFP, or small-factor pluggable, interfaces. These are very common and allow your device to hot-plug different types of interfaces, such as Gigabit Ethernet as well as fiber optic transceivers. The PowerRouter 2200 series hardware offers upwards of 20 Gigabit SFP interfaces if you require mostly fiber interfaces.

T1/E1 Interfaces

In RouterOS Version 3.15 MikroTik removed most support for T1/E1 interface cards. This was probably is mostly due to the problematic nature of the drivers that existed. FarSync, a brand of cards, are still supported, however, they are expensive compared to other T1/E1 interfaces on the market.

Supported x86 Hardware

It's important to note that RouterOS does not use "drivers" in the same way as other computers that most people know of. Most computer users are accustomed to installing an Operating System, and then they install drivers to make all of the hardware work. RouterOS is different. RouterOS contains all of the drivers that you will need right as part of the main installation. MikroTik though, chooses drivers based on popularity, usability, as well as what is in the contents of the latest Linux kernel to base select the drivers to include with each installation package.

RouterOS supports a wide range of Ethernet network adaptors, wireless interface cards, fiber interfaces, as well as 10 Gigabit interfaces. It supports a number of Mini-PCI and PCI adaptors, 3G or cellular data cards, and system boards. Before you start building your first RouterOS system, make sure you look at the supported hardware list. You can find that list by going to http://wiki.MikroTik .com/wiki/Supported_Hardware. This list is constantly updated by both MikroTik and RouterOS users.

With all of these options out there, sometimes it can be difficult to build your own system. If there is a known RouterBOARD or a pre-designed system that is already supported and tested with RouterOS, I would suggest purchasing these. The cost difference for a pre-designed system on these is typically minimal vs. the cost of router failures due to build-it-yourself hardware systems. I have seen this many times, customers wondering why their system does not constantly run reliably. I actually asked one customer what kind of hardware they were using, and their response was, "When my Windows 98 computer was too slow for me, we put it on the shelf. Later, we needed a router, so we plugged it in and put RouterOS on it. When the power supply died in it, we replaced it, with one of our standard fifteen dollar power supplies."

As a wise man said, "You get what you pay for". I tend to agree with this. If you put a $15 power supply in a system and think it is going to run 24 hours a day, 7 days a week for months or years without failure, then you need to rethink what business you are in. Get hardware that is supported, tested, and as well as designed for a long lifespan. Servers are built with higher grade components, power supplies, and better network cards typically, and that is why they tend to last longer. The same nomenclature should be used with your x86 RouterOS system. Don't skimp when you have to rely on it

RouterOS Licensing

RouterOS has five different licensing levels. Several are designed for evaluation of the RouterOS software. License levels 3 through 6 are the most common licenses. These are paid licenses. Most level 3 and 4 licenses come with RouterBOARD Products and other products designed to run RouterOS. The level 5 and 6 are extended licenses designed for high end applications.

License Level	4	5	6
Price/Cost	$45	$95	$250
Upgradable	ROS v4.x	ROS v5.x	ROS v5.x
Wireless AP	Yes	Yes	Yes
Wireless CPE/Bridge	Yes	Yes	Yes
Dynamic Routing	Yes	Yes	Yes
EoIP Tunnels	No Limit	No Limit	No Limit
PPPoE Sessions	200	500	No Limit
PPTP Tunnels	200	No Limit	No Limit
L2TP Tunnels	200	No Limit	No Limit
OVPN Tunnels	200	No Limit	No Limit
VLAN Interfaces	No Limit	No Limit	No Limit
P2P Firewall Rules	No Limit	No Limit	No Limit
NAT Rules	No Limit	No Limit	No Limit
Hotspot Clients	200	500	No Limit
Radius Client	Yes	Yes	Yes
Web Proxy	Yes	Yes	Yes
User Manager Sessions	20	50	No Limit

The level 3 Licenses are designed for Client or CPE devices. These are for wireless CPEs, or customer equipment. Typically you would purchase a Level 4 license or a WISP license. This license is included with many of the 400 series RouterBOARD products, as well as other x86 RouterOS products. There are no upgrades between licenses, so keep in mind the final usages. You can purchase another license and place it on-top of an existing license. An example of this may be that you have a hotspot that needs more than 200 active clients at one time. If this is the case, you can purchase another level 5 license, at full cost, and then apply it to the existing hardware.

Note that the licenses never expire, they support an unlimited number of interfaces, and each license is for only one installation. The installation is based on the disk drive or storage device that you use to install RouterOS on. You can install RouterOS on USB sticks, SATA and IDE hard drives, Disk on Modules or DOMs, as well as compact flash cards. You can move the storage device from one system to another, but you can't move the license. For example, you can move your compact flash card from one x86 system to another x86 system; however you cannot move the license from the existing compact flash card to another compact flash card. If you need a larger compact flash card, then you will have to purchase another license.

What is the Software ID? The software ID is the ID number associated with each specific RouterOS installation. It uses the hardware, disk information as well as other methods to generate a software ID Key. This key is then used to generate a license when paying or registering for a demo license.

What if your hard disk fails? MikroTik has the ability to replace a license for a nominal cost. You will need to contact them to receive a replacement key. They may need to know how or why the drive failed, and may request the drive before issuing a replacement key. In most cases though, it may be quicker and cheaper just to purchase another license.

Where is the license stored? RouterOS stores the license inside the MBR or the boot sector of your drive. Because of this, if you format the device with a non-MikroTik format utility, such as windows format, YOU WILL LOSE YOUR LICENSE! However, MikroTik has thought of this for us, and has provided the NetInstall Utility. The next section will cover the installation of RouterOS on many different devices

Extended Frequency Licenses

RouterOS also has the ability to add an extended frequency license, sometimes also called a custom frequency license. To determine if you have an extended frequency license, click on SYSTEM -> LICENSE. In the license window "extended frequency" shows in the features section. These license features allow RouterOS in conjunction with the right radio card, to operate in any frequency that the hardware can operate in. You will need to contact a reseller in your country to obtain this license feature. Some may have special paperwork for you to fill out to obtain this license feature. If you have a license or can operate in a band that is not normally allowed by RouterOS, you can obtain this license feature, install it and run on any frequency that the radio card supports.

With the above said, since version 4.3 of RouterOS, the "Conformance Testing Mode" formerly known as "Superchannel" or "Customer Frequency Upgrade", is no longer a paid licensing option. If you are using an older version of RouterOS you will need to purchase these, however, on versions higher than v4.3, you will no longer need too. In your frequency selection, you will have options for all channels that the radio card can operate in. Frequencies in BOLD are default standardized channels. If you operate your radio outside of these BOLD channels, you will have to modify the scan-list, or directly select the frequency on your clients. See scan-list for more options to allow you to find radios operating outside the standard frequencies for your selected country and frequency-mode.

Ways to Lose your RouterOS License

If you Format your flash drive, hard disk or DOM with anything other than MikroTik's NetInstall Utility, YOU WILL LOSE YOUR LICENSE!

DO NOT FORMAT YOUR DRIVE UNLESS IT IS WITH THE NETINSTALL UTILITY!

RouterOS Installation

Installation methods will depend on what hardware you are using. RouterOS can be installed on many different devices. These include x86 computers, and RouterBOARD products. RouterBOARDs typically come with not only the RouterOS software already loaded, and but has a license already installed as well. Contact your local distributor to find out what hardware comes with what license.

If you built your own PC and are planning to install RouterOS on it, then you have several choices for the installation. PC-based installations can use NetInstall to load an IDE or SATA DOM, or possibly a USB stick or other form of flash card, including Compact flash cards would be included with this. You can also, use three other methods; NetInstall using a bootable network interface card or (NIC), is one method, or using a CD-based installation.

For PC or x86 system installations, the recommended method is either NetInstall with a Compact Flash or DOM module, or the CD-based installation method.

For RouterBOARDs, we have one installation method, NetInstall. Note that RouterBOARDs will come with a RouterOS installation and a license; you typically will only need to use this method to either upgrade a device or to recover from a lost password. You can also reset the unit; see the “RouterBOARD Reset” Section. Since quite a few of the RouterBOARD products are put into lighting and static-intensive areas, such as radio towers, etc., as well as lightning discharges near where the RouterBOARD is installed, there are times that the RouterBOARD unit may stop functioning due to a NAND issue. Older RouterBOARD products had this issue, as the NAND was more susceptible to electromagnet interference; however, most of the newer RouterBOARD products have shown quite resilient to this type of issue. A reload of the NAND via the NetInstall program will reload the OS and allow the unit to restart in some cases. Keep in mind that if your hardware takes a direct lightning strike etc., the chances of it even having it powering on is slim. You may even need to look around to find for the pieces of the board.

Using NetInstall on RouterBOARD Products

What you will need:

- Your RouterBOARD device
- Access to the Serial port on the RouterBOARD Device
- An Null Modem cable between your PC and the RouterBOARD device
- An Ethernet cable from the network interface on your computer to the RouterBOARDs Ethernet1 port
- The RouterOS NetInstall Utility, found on the MikroTik website
- The latest NPK file for your RouterBOARD Device
- The power supply for your RouterOS device as well, can be either POE or you can use the power jack.

Before you start, you will have to download the right file, depending on the model of your RouterBOARD. There are several CPU versions of RouterOS, and what RouterBOARD you have will determine what CPU version of RouterOS you need. For instance; if you have a RouterBOARD 400 series device, you will need the RouterOS version that supports the MIPSBE CPU. If you have a RouterBOARD 1000, you will need the PowerPC Processor version.

So let's get started:

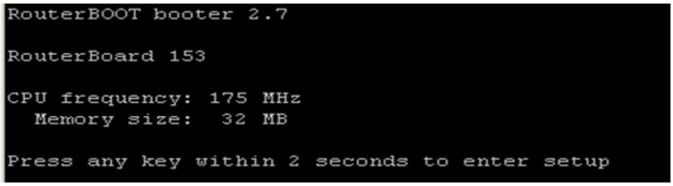
First, make sure you can use a terminal program to connect to the serial port of your RouterBOARD product. You should be able to power on the RouterBOARD, and see the boot process in your terminal program. Some common programs that you can use, are be Windows HyperTerminal, or Putty. You can download putty at http://www.chiark.greenend.org.uk/~sgtatham/putty/. You can also do a web search as well to find download locations for Putty.

Second, you will need to configure a PC with a network cable running to Ethernet 1 of your RouterBOARD product. You don't need a cross-over cable as RouterBOARDs are created with auto MDI-X ports to automatically cross over if necessary. It is possible to run through a switch, but this sometimes is problematic, so I suggest running a cable directly between your computer and the RouterBOARD.

Third, on your computer, place an IP address of 192.168.0.1 with a subnet mask of 255.255.255.0 on the Ethernet interface. You do not need a gateway or DNS servers. This may disconnect you from the Internet; however, you should have already downloaded all necessary files.

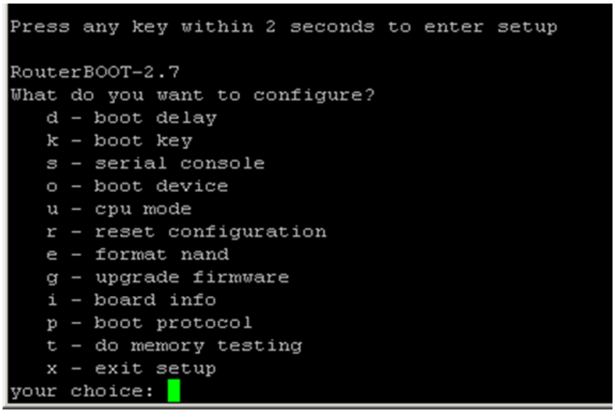
Fourth, ensure that your PC does not have any Firewalls turned on or active and that any active network defense software is disabled. NetInstall uses Layer 2 along with IP addresses that you identify; Firewalls could block the requests from the RouterBOARD and prevent the NetInstall Utility from running correctly. Anti-virus programs that have network or software Firewalls, and other similar applications should also be disabled, removed or turned off.

Now open your serial port, RouterBOARDs typically operates at 115200 baud. You MUST use a null-modem cable! You can use USB-to-serial converters if you need too. When you open your serial port, you should see the login prompt if your board has started up. If you have not applied power to your RouterBOARD, you can do so, and you should see the BIOS screen. During this BIOS screen, you should have an option to "*press any key to enter setup*". If you have already started your RouterOS and have a login prompt, you will need to unplug your RouterBOARD, wait a few seconds, and then reapply power so that the RouterBOOT booter comes up and you have the option to enter the BIOS configuration.



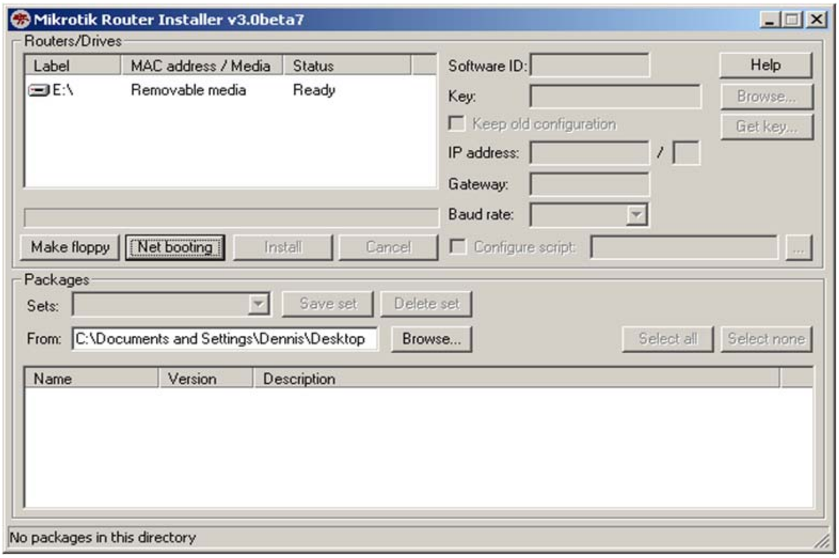
```
RouterBOOT booter 2.7

RouterBoard 153

CPU frequency: 175 MHz
  Memory size:  32 MB

Press any key within 2 seconds to enter setup
```

The screen above is an example of the RouterBOOT BIOS. Note that you have the option to *"Press any key within 2 seconds to enter setup".* Press any key to enter the BIOS setup.



```
Press any key within 2 seconds to enter setup

RouterBOOT-2.7
What do you want to configure?
   d - boot delay
   k - boot key
   s - serial console
   o - boot device
   u - cpu mode
   r - reset configuration
   e - format nand
   g - upgrade firmware
   i - board info
   p - boot protocol
   t - do memory testing
   x - exit setup
your choice:
```

Once you enter the RouterBOOT or BIOS of the RouterBOARD, now you will need to finish setting up your PC. Start your NetInstall utility.



Label	MAC address / Media	Status
E:\	Removable media	Ready

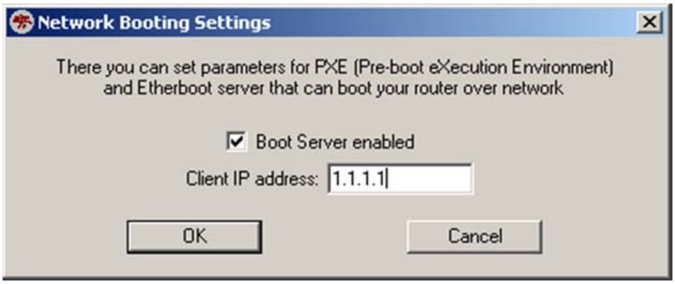
From: C:\Documents and Settings\Dennis\Desktop

Name	Version	Description

No packages in this directory

This utility will allow you to install via Netbooting of your RouterBOARD. It will use your Ethernet cable to boot your RouterBOARD, and enter an installation mode. Then you can select your installation package, or NPK file, and finish the installation.

Next, select your Net booting Button:



Network Booting Settings

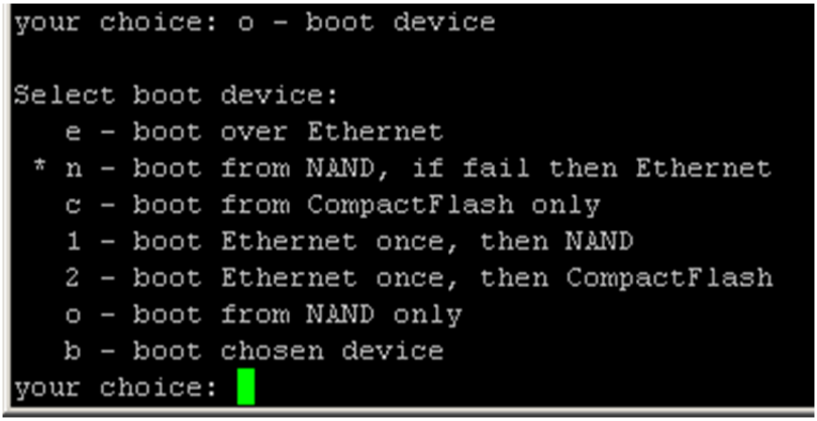
There you can set parameters for PXE (Pre-boot eXecution Environment) and Etherboot server that can boot your router over network

☑ Boot Server enabled

Client IP address: 1.1.1.1

Here, enter the IP address that you wish to give your RouterBOARDs Ethernet1 interface upon Netbooting. Remember, before we entered 192.168.0.1 as our IP on our PC. Just like any other IP based device, we need to make sure the IP that we give our RouterBOARD is in the same subnet as our NetInstall PC. My suggestion would be to use 192.168.0.2 and press OK.

Once we have the Installation server ready by using the NetInstall Utility, we need to tell our RouterBOARD to boot from the Ethernet interface. In where we left the terminal window, in the BIOS there is an option for *Boot Device*. The option to select this is *o.*

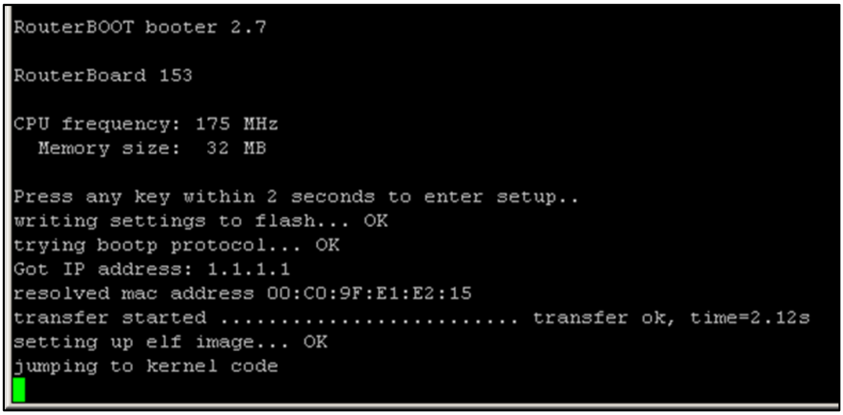


```
your choice: o - boot device

Select boot device:
   e - boot over Ethernet
 * n - boot from NAND, if fail then Ethernet
   c - boot from CompactFlash only
   1 - boot Ethernet once, then NAND
   2 - boot Ethernet once, then CompactFlash
   o - boot from NAND only
   b - boot chosen device
your choice:
```

Upon selecting *o,* we have a number of other options. Typically your RouterBOARD will boot from its NAND or its on-board flash memory. Since this is not working, or you don't want to load the existing version of RouterOS, we need to boot from another device. You can typically select *1* to boot from Ethernet once, and then boot from the NAND. I say typically, as your results may vary and if it's your first time, you might have to try the installation server a few times to understand its ins and outs.

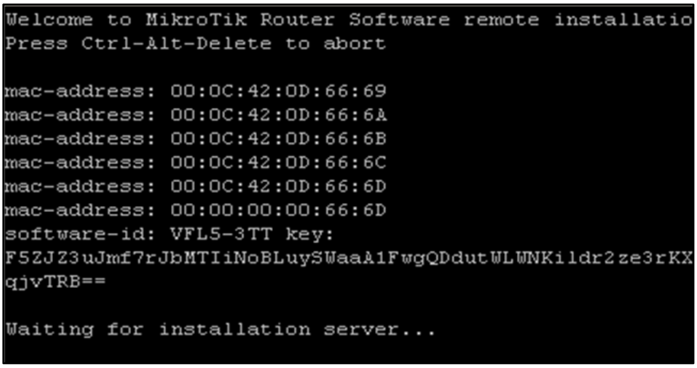
If you select 1, then you have one time to boot into the installation server mode, after that, it will continue booting to the NAND. This is usually what you want, as you want to boot via Ethernet, load the installation server, install RouterOS, and then it will reboot using the NAND and finish loading the OS. Another option would be to just boot over Ethernet, however, once your installation is complete, you will have to go back into the BIOS and select to boot from the NAND to finish the installation.

Once you choose your boot device, (remember we need Ethernet at least once to start the installation program), hit *x* to exit the BIOS setup on the RouterBOARD. This will cause your device to reboot, you should see the BIOS screen again, but this time, do not press any key to stop the board from booting.



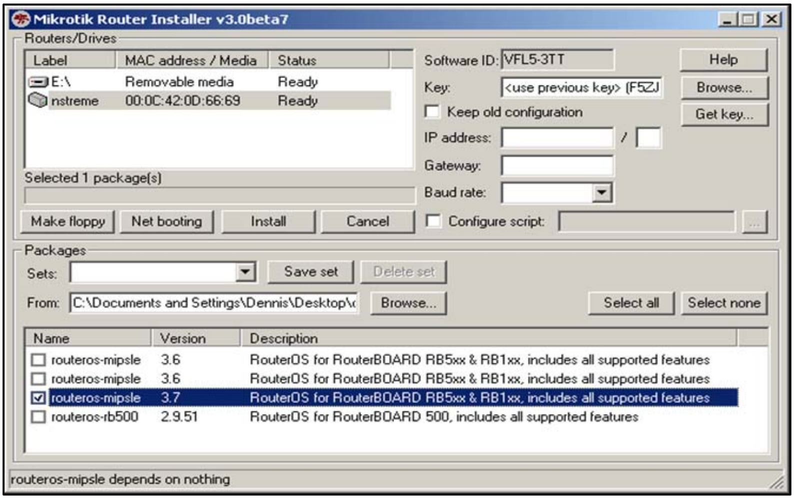
```
RouterBOOT booter 2.7

RouterBoard 153

CPU frequency: 175 MHz
  Memory size:  32 MB

Press any key within 2 seconds to enter setup..
writing settings to flash... OK
trying bootp protocol... OK
Got IP address: 1.1.1.1
resolved mac address 00:C0:9F:E1:E2:15
transfer started ........................ transfer ok, time=2.12s
setting up elf image... OK
jumping to kernel code
```

You should see the RouterBOARD trying bootp protocol to boot as shown above. Within a few seconds you should see the IP you put into your NetInstall Booter program, it should transfer the installation software, and come up with the MikroTik Router Software Remote Installer.



```
Welcome to MikroTik Router Software remote installatio
Press Ctrl-Alt-Delete to abort

mac-address: 00:0C:42:0D:66:69
mac-address: 00:0C:42:0D:66:6A
mac-address: 00:0C:42:0D:66:6B
mac-address: 00:0C:42:0D:66:6C
mac-address: 00:0C:42:0D:66:6D
mac-address: 00:00:00:00:66:6D
software-id: VFL5-3TT key:
F5ZJZ3uJmf7rJbMTIiNoBLuySWaaA1FwgQDdutWLWNKildr2ze3rKX
qjvTRB==

Waiting for installation server...
```

It is now waiting for the installation server; next we go back to our NetInstall Utility as the RouterBOARD is waiting for input.



Mikrotik Router Installer v3.0beta7

Routers/Drives

Label	MAC address / Media	Status
E:\	Removable media	Ready
nstreme	00:0C:42:0D:66:69	Ready

Selected 1 package(s)

Software ID: VFL5-3TT

Key: <use previous key> (F5ZJ

□ Keep old configuration

IP address: /

Gateway:

Baud rate:

□ Configure script:

Packages

Sets:

From: C:\Documents and Settings\Dennis\Desktop\

Name	Version	Description
□ routeros-mipsle	3.6	RouterOS for RouterBOARD RB5xx & RB1xx, includes all supported features
□ routeros-mipsle	3.6	RouterOS for RouterBOARD RB5xx & RB1xx, includes all supported features
☑ routeros-mipsle	3.7	RouterOS for RouterBOARD RB5xx & RB1xx, includes all supported features
□ routeros-rb500	2.9.51	RouterOS for RouterBOARD 500, includes all supported features

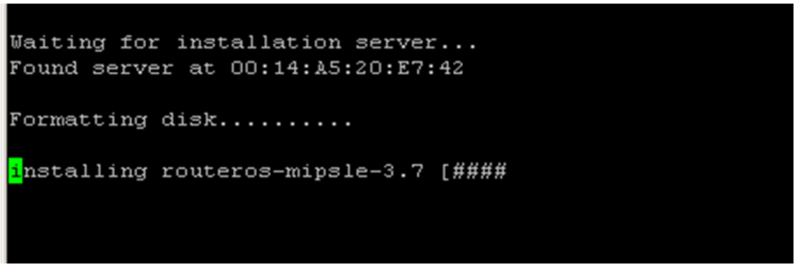
routeros-mipsle depends on nothing

Note that we now have a device, typically labeled *Nstreme*, along with its MAC Address. This is the RouterBOARD, and it's waiting for installation. We then use the browse button under the packages section and find the location of our NPK installation file is at. Upon selecting the folder, we can then check the box with the proper installation file and version. You may only have one file in this box, if it's the only one you may have downloaded.

Once you have the package selected, you have a few other options. In the upper right corner, you can select to keep old configuration, this will keep the existing configuration, but write over the RouterOS Operating System. It

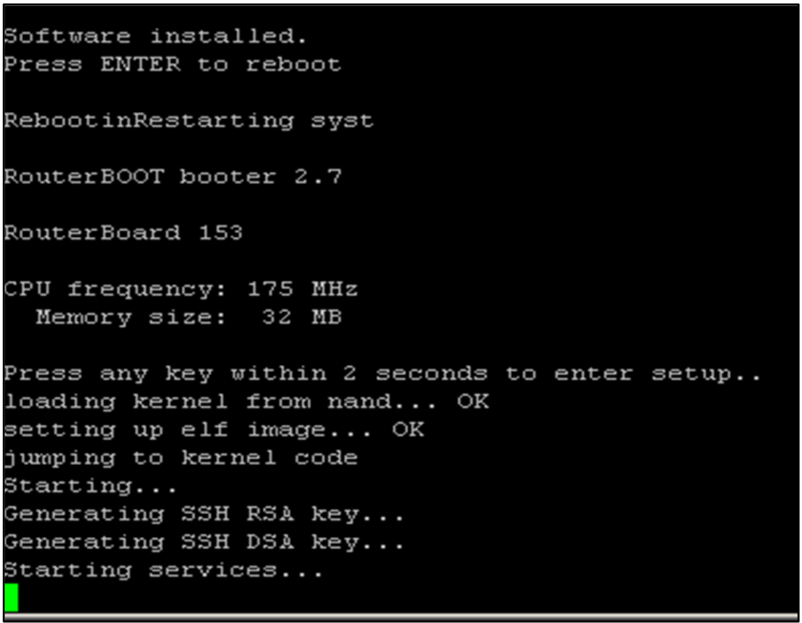
WILL NOT remove any passwords on your system. You also have the option of specifying the default baud rate for the serial port, or including a configuration script.

Once you are ready to do the installation, simply press the Install button!



```
Waiting for installation server...
Found server at 00:14:A5:20:E7:42

Formatting disk..........

installing routeros-mipsle-3.7 [####
```

The NetInstall Utility will then format the disk, in this case it will be the NAND of the RouterBOARD, and perform the initial installation of the RouterOS installation package. Once this is complete, you can press any key and the RouterBOARD will reboot. If you selected to boot from Ethernet once, and then the NAND, upon rebooting, it will finish the load of RouterOS. If you selected Ethernet only, it will come back to the installation server, unless you go into the BIOS and set it to boot from the NAND.

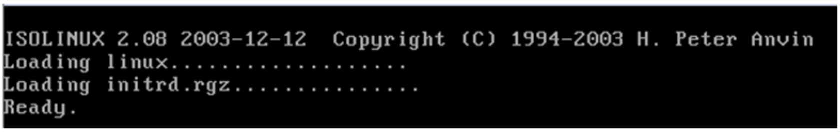


```
Software installed.
Press ENTER to reboot

RebootinRestarting syst

RouterBOOT booter 2.7

RouterBoard 153

CPU frequency: 175 MHz
  Memory size:  32 MB

Press any key within 2 seconds to enter setup..
loading kernel from nand... OK
setting up elf image... OK
jumping to kernel code
Starting...
Generating SSH RSA key...
Generating SSH DSA key...
Starting services...
```

Above the system has restarted, booted from the NAND, generated the SSH Keys, and started the RouterOS Services. At this point, you have a working RouterOS system!

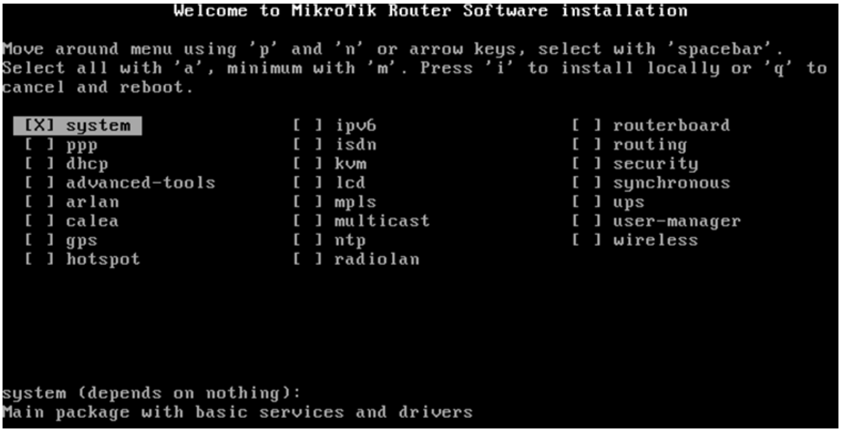
ISO – CD Installation

Installation of RouterOS via an ISO image and/or CD image is very simple. If you are installing your system via a CD, there is an ISO included in the all packages download with RouterOS. This ISO should be burned with an ISO burner and is a bootable CD. Upon booting from the CD and/or ISO, you will be presented with the ISOLINUX startup system to load the OS as seen below.



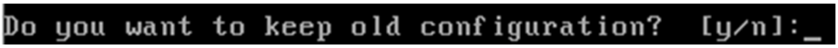
```
ISOLINUX 2.08 2003-12-12  Copyright (C) 1994-2003 H. Peter Anvin
Loading linux...................
Loading initrd.rgz...............
Ready.
```

Once booted from CD you will then see the Software installation menu for RouterOS. As you can see below, there are a number of options that you can install. Here you can select exactly what packages you wish to install. Simply use your arrow keys to highlight the packages that you wish to install and hit the spacebar to put a X beside each package to select it for installation.



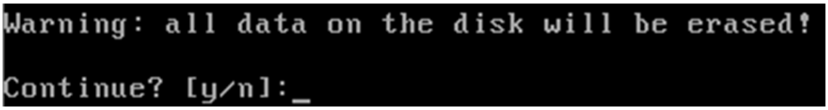
```
                Welcome to MikroTik Router Software installation

Move around menu using 'p' and 'n' or arrow keys, select with 'spacebar'.
Select all with 'a', minimum with 'm'. Press 'i' to install locally or 'q' to
cancel and reboot.

  [X] system                [ ] ipv6                  [ ] routerboard
  [ ] ppp                   [ ] isdn                  [ ] routing
  [ ] dhcp                  [ ] kvm                   [ ] security
  [ ] advanced-tools        [ ] lcd                   [ ] synchronous
  [ ] arlan                 [ ] mpls                  [ ] ups
  [ ] calea                 [ ] multicast             [ ] user-manager
  [ ] gps                   [ ] ntp                   [ ] wireless
  [ ] hotspot               [ ] radiolan






system (depends on nothing):
Main package with basic services and drivers
```

Once you have selected all of your packages, you can finish your installation. To do this, hit the I key, I as in Igloo. This will allow you to finish the installation. As a shortcut method, you can install all the packages by selecting A or just the minimum packages by selecting M. Q will cancel your installation and reboot your system.



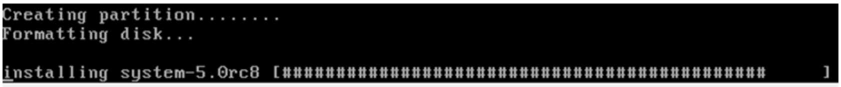
```
Do you want to keep old configuration?  [y/n]:_
```

The system will ask if you wish to keep your existing or old configuration. If you are using a licensed version of RouterOS on the existing disk that you are installing too, then this is an option. By selecting Y, to you will keep that configuration. If you do not have a license, then hitting Y will yield a license issue and it will continues on without keeping your old configuration.



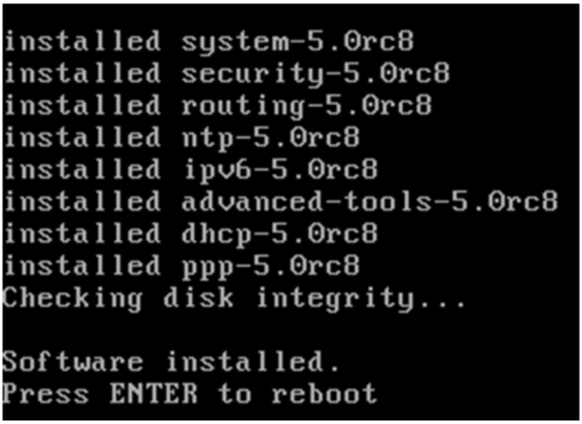
```
Warning: all data on the disk will be erased!

Continue? [y/n]:_
```

It will ask if you wish to format the disk as well.



```
Creating partition........
Formatting disk...

installing system-5.0rc8 [#############################################     ]
```

Once the disk if formatted (remember you will not lose your license by doing this installation if you do have a license)



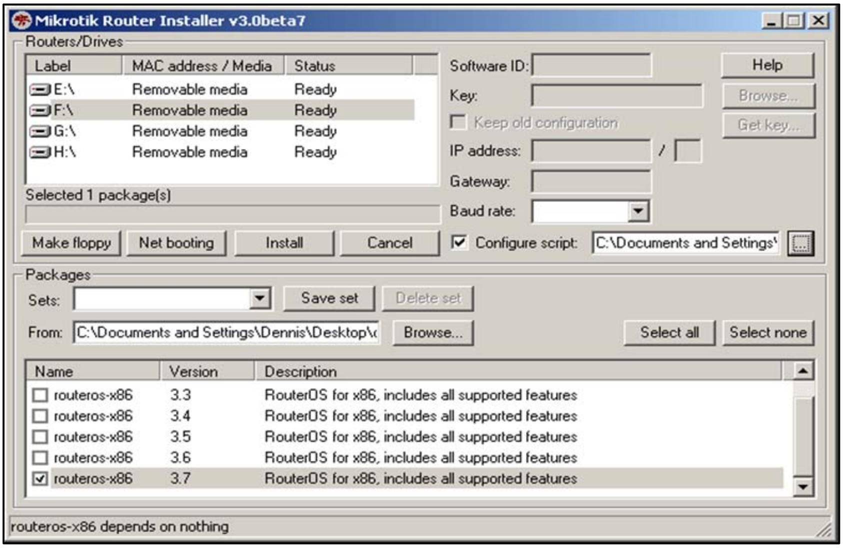
```
installed system-5.0rc8
installed security-5.0rc8
installed routing-5.0rc8
installed ntp-5.0rc8
installed ipv6-5.0rc8
installed advanced-tools-5.0rc8
installed dhcp-5.0rc8
installed ppp-5.0rc8
Checking disk integrity...

Software installed.
Press ENTER to reboot
```

When the installation completes, it will ask you to press enter to reboot your system. These will then bring up the OS will load, generate any SSH keys you may need and then bring you to a RouterOS login prompt.

DOM / Flash Card / Hard Disk Installation via NetInstall

RouterOS Installation via NetInstall is very similar to the NetInstall installation of RouterBOARDs, but it is simpler! For your flash card, you will need some kind of reader. I commonly use Compact Flash cards, and use a simple USB Flash reader. If you are using a DOM module or hard disk, you will need to install this like any other device inside your PC. Of course, you will need your PC's BIOS to recognize it. If you can start by formatting it via windows then this will ensure that it is working prior to using the NetInstall Utility. Remember though, if you format an already licensed drive using anything BUT NetInstall, you WILL LOSE YOUR LICENSE.

Once you have the disk ready to go, start your NetInstall Utility. Just like with the RouterBOARD products, you will need the NPK file that goes with the system you are installing. Chances are this will be an x86 system, so you will need the x86 version of RouterOS NPK. You can download this along with the NetInstall Utility right from MikroTik's webpage.



Label	MAC address / Media	Status
E:\	Removable media	Ready
F:\	Removable media	Ready
G:\	Removable media	Ready
H:\	Removable media	Ready

Selected 1 package(s)

☑ Configure script: C:\Documents and Settings\

From: C:\Documents and Settings\Dennis\Desktop\

Name	Version	Description
□ routeros-x86	3.3	RouterOS for x86, includes all supported features
□ routeros-x86	3.4	RouterOS for x86, includes all supported features
□ routeros-x86	3.5	RouterOS for x86, includes all supported features
□ routeros-x86	3.6	RouterOS for x86, includes all supported features
☑ routeros-x86	3.7	RouterOS for x86, includes all supported features

routeros-x86 depends on nothing

As you can see I have several *Removable Media* drives. In this image, we have a USB flash reader with four slots, for different types of media. Only one is my Compact Flash. I formatted the Compact Flash with Windows prior to starting NetInstall, so I know it is drive F on my system. I select my F drive, then browse to the folder where my NPK file is located at, and select the correct NPK file for installation. This is just like the final steps when using the NetInstall Utility with a RouterBOARD. Once you select those options, including your baud rate and script selected, you can simply press Install to format and install the RouterOS System.

Once the installation is completed, it will say installation is complete in the NetInstall Utility; you will be able to shut down your PC or stop the necessary flash drive and remove it. Insert the storage device into your new RouterOS system, and power on. The first boot will finish the installation of RouterOS on the storage device. This may take a few minutes. Once complete, the system will restart, generate the SSH keys, start the RouterOS Services, and then display a login prompt.

Note, when you have an existing licensed device, with DOMs, and flash cards, there is no way to keep the old configuration!

Accessing RouterOS

RouterOS is not your normal Router. Typical access methods such as SSH and Telnet access are offered in RouterOS. However, there are two other methods that allow you to configure your RouterOS system. MAC Telnet gives you the ability to login to a RouterOS system that has no IP addresses configured. In fact, this is one of the strongest admin abilities of RouterOS. As long as there is Layer 2 connectivity, you can access your RouterOS system!

Now, you might ask, it's a router, it should be doing TCP/IP Layer 3 routing etc., why do I need to access it via Layer 2? Simple; if it's not configured, you will have the ability to access and configure your RouterOS without needing a console or serial cable! I have done complete configurations of several RouterOS devices across long range wireless links. The installers basically configured RouterOS to connect wirelessly to an existing access point or backhaul radio and then I was able to access the other RouterOS devices, even without IPs, and configure them remotely!

The most common way to access your RouterOS configuration is with a utility called WinBox. You can download this from MikroTik's webpage, or if you have IP connectivity to your router, use your favorite web browser and go to the router's IP address. This will bring up a configuration page, which allows you to download WinBox at. I would suggest though, getting the latest WinBox version via MikroTik's webpage.

Just like the Net Install Utility, WinBox will function at either Layer 2 or Layer 3. You can connect to your RouterOS system via a MAC address or an IP address. IF you are using the MAC, make sure you have your Firewall turned off, as well as any network protection software that you may have loaded on your PC.

RouterOS Access Methods

- Layer 2
- MAC Telnet
- Via MAC in WinBox
- Layer 3
- IP-Based Telnet
- Via IP in WinBox
- SSH -- Secure Shell
- Webpage
- API – Application Programming Interface
- Serial Interface

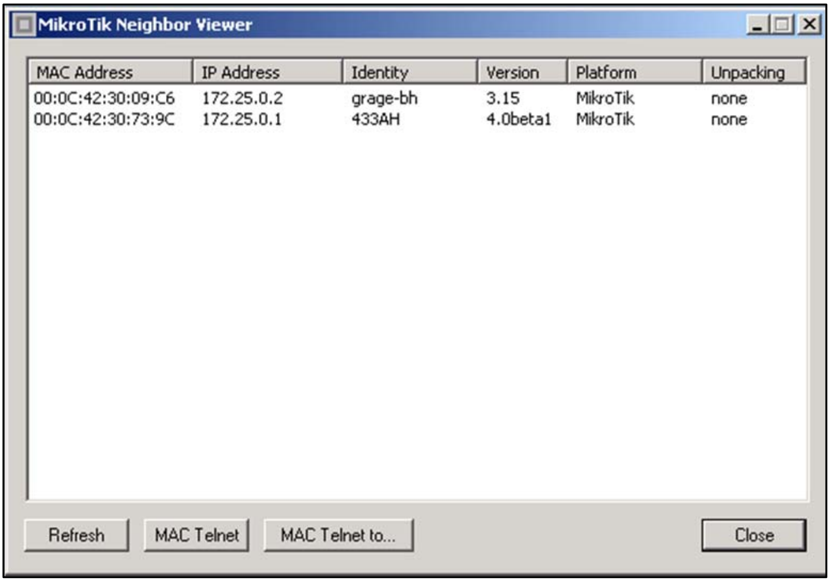
Default User and Password

RouterOS will default to the administrator username of 'admin' and the password will be blank.

Using Neighborhood Viewer

MikroTik has software called Neighbor Viewer. You can download this software via MikroTik's webpage. There are actually two applications; one is the Neighbor Viewer application. This will look for MAC addresses that are broadcasting MNDP packets. These MikroTik Network Discovery Packets are broadcast so that other neighboring MikroTik devices, WinBox and the Neighbor Viewer can find them. This is very similar to CDP, Cisco Discovery Protocol. This feature is enabled by default and we will talk about this more in the "RouterOS Services" Section.

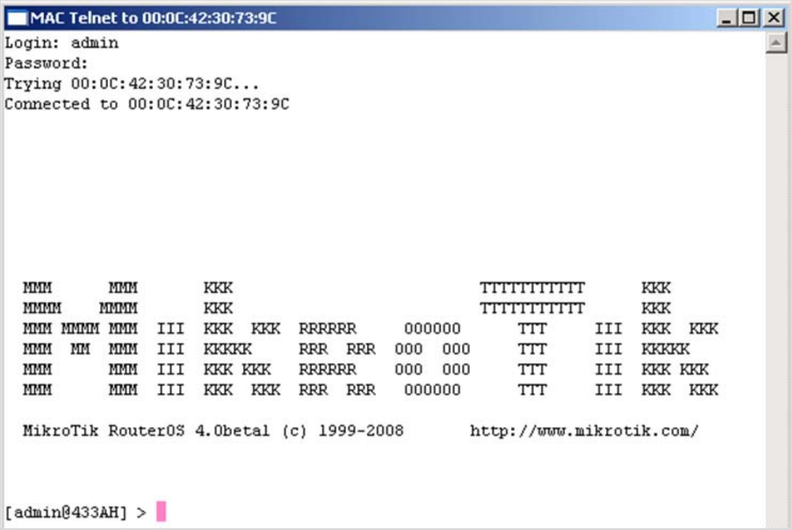
By running the Neighbor Viewer, you can see RouterOS devices that have Layer 2 connectivity with your PC. By selecting one of these devices, you have the option to open a MAC telnet session with it. This opens the terminal program, the second application, which is included in the ZIP file that Neighbor Viewer came in, and connects you to your RouterOS device via a MAC Telnet session. Once your MAC telnet opens, you will be prompted for a login and the password to your device. Once entered, you will receive a terminal prompt and will be able to issue terminal commands.



MikroTik Neighbor Viewer

MAC Address	IP Address	Identity	Version	Platform	Unpacking
00:0C:42:30:09:C6	172.25.0.2	grage-bh	3.15	MikroTik	none
00:0C:42:30:73:9C	172.25.0.1	433AH	4.0beta1	MikroTik	none

By Selecting the RouterOS system that you wish to connect to, you can then click on the MAC Telnet button, and it will open the Terminal program. This program connects you into your router via MAC Telnet.



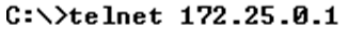
MAC Telnet to 00:0C:42:30:73:9C

```
Login: admin
Password:
Trying 00:0C:42:30:73:9C...
Connected to 00:0C:42:30:73:9C




  MMM      MMM       KKK                          TTTTTTTTTTT      KKK
  MMMM    MMMM       KKK                          TTTTTTTTTTT      KKK
  MMM MMMM MMM  III  KKK  KKK  RRRRRR     000000      TTT     III  KKK  KKK
  MMM  MM  MMM  III  KKKKK     RRR  RRR  000  000     TTT     III  KKKKK
  MMM      MMM  III  KKK KKK   RRRRRR    000  000     TTT     III  KKK KKK
  MMM      MMM  III  KKK  KKK  RRR  RRR   000000      TTT     III  KKK  KKK

  MikroTik RouterOS 4.0beta1 (c) 1999-2008       http://www.mikrotik.com/


[admin@433AH] >
```

Using Telnet

By default, RouterOS has a Telnet server enabled. You can use any Telnet application via the IPs on your RouterOS device to connect. Upon connecting you will receive a login prompt and then will be able to login and issue terminal commands. RouterOS by default runs Telnet sessions on the default Telnet port of 23.

Using windows you can type *telnet ip_address* of router. In windows, you can type Start → Run → CMD. This will open a command prompt window and allow you to type your telnet command.



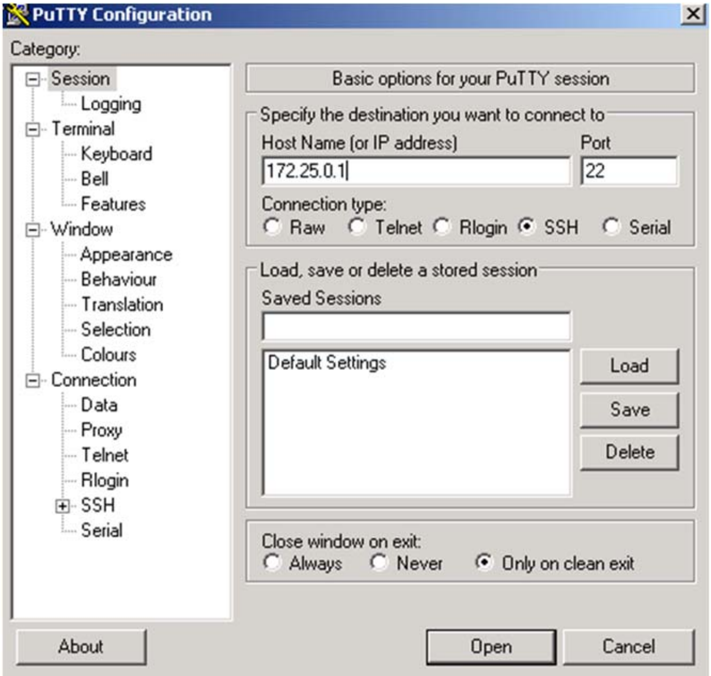
```
C:\>telnet 172.25.0.1
```

Note you must have Layer 3 connectivity. You will need a same-subnet IP on your PC as well as on your RouterOS System. Telnet sessions are typically not secure, as they provide no data encryption, and keystrokes and text are sent in clear text.

SSH – Secure Shell Access

RouterOS also offers Secure Shell Access to the terminal. This access is the exact same as using a telnet session, however, during the SSH connection, the data exchanged uses a secure channel between your PC and the RouterOS device. In version 4.x of RouterOS, there is support for version 1 and version 2 SSH. In version 5.x, only SSH version 2 will be supported. Upon loading your RouterOS device, you will note that it generates SSH security keys. These keys are used to the secure the connection. This means that text that is transmitted or received by your SSH client is encrypted, and not sent in clear text.

SSH though does run on the IP layer, so you will need to have Layer 3 connectivity to your Router. There are a number of FREE SSH clients that you can use. Putty is one of them, as well as OpenSSH, and other applications. We will show the Putty application here.



Host Name (or IP address): 172.25.0.1

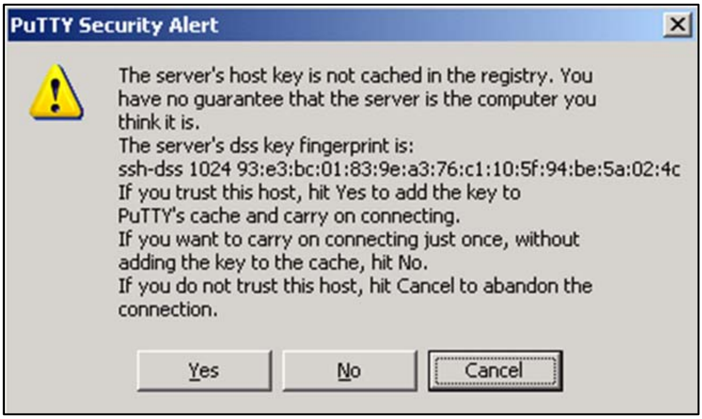
Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Saved Sessions: Default Settings

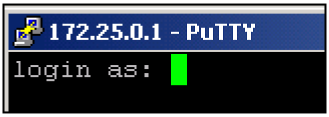
Close window on exit: ○ Always ○ Never ● Only on clean exit

As you can see, there are a number of options, but for basic SSH connectivity, you will need to put in the host name, or IP address into Putty. RouterOS defaults to the standard SSH port of 22. You will use the connection type of SSH. You can also store sessions if you wish as well. Once you have the proper IP information and port, you can click Open to start your SSH session.



The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
The server's dss key fingerprint is:
ssh-dss 1024 93:e3:bc:01:83:9e:a3:76:c1:10:5f:94:be:5a:02:4c
If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the connection.

The first time you connect to your RouterOS system, you will see a host key that is not cached. This is the SSH Key that is generated upon the initial installation of your RouterOS system. Putty will cache the key, if you wish, so that you don't get this message again if you wish. You would hit yes to cache the key. If you hit no, you will continue connecting, but it will not cache they key.



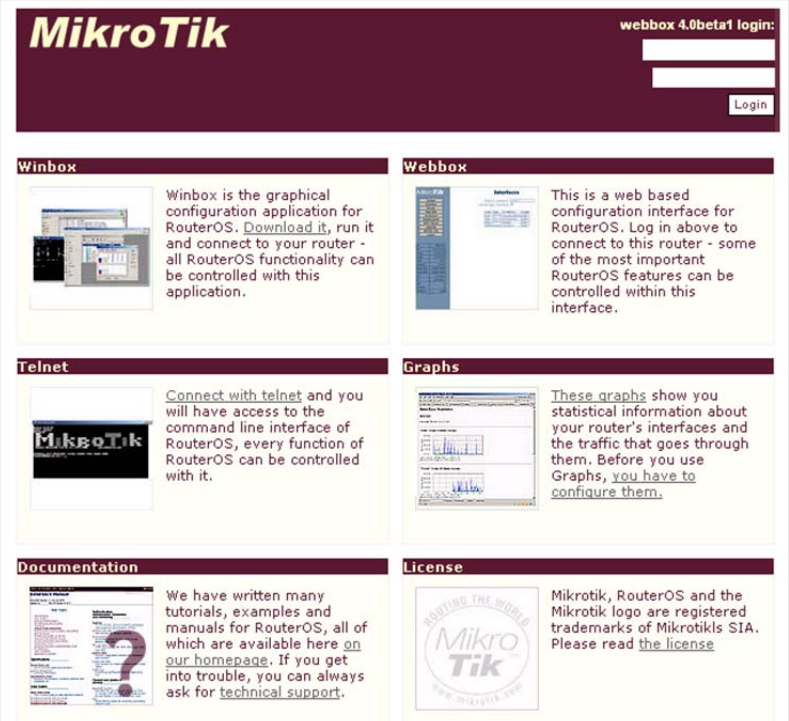
```
login as: 
```

Once you connect, you will get a login prompt. From this point on, your connection will be just like a telnet session. You will be presented the terminal window for programming RouterOS.

WebBox

RouterOS allows you to us a webpage for basic configuration. To get to this page, you will need Layer 3 connectivity or IP connectivity to your Router. Your PC must be on the same subnet as the RouterOS system. In V5.x MikroTik has completed the initial cross platform access method using a new service call WebFig. WebFig is described in more detail in its section of the book. The WebBox interface is a simplified system, and the WebFig system is meant to replace WinBox as the primary configuration tool at Layer 3.

Simply browse to the IP address using your favorite web browser.



Winbox

Winbox is the graphical configuration application for RouterOS. Download it, run it and connect to your router - all RouterOS functionality can be controlled with this application.

Webbox

This is a web based configuration interface for RouterOS. Log in above to connect to this router - some of the most important RouterOS features can be controlled within this interface.

Telnet

Connect with telnet and you will have access to the command line interface of RouterOS, every function of RouterOS can be controlled with it.

Graphs

These graphs show you statistical information about your router's interfaces and the traffic that goes through them. Before you use Graphs, you have to configure them.

Documentation

We have written many tutorials, examples and manuals for RouterOS, all of which are available here on our homepage. If you get into trouble, you can always ask for technical support.

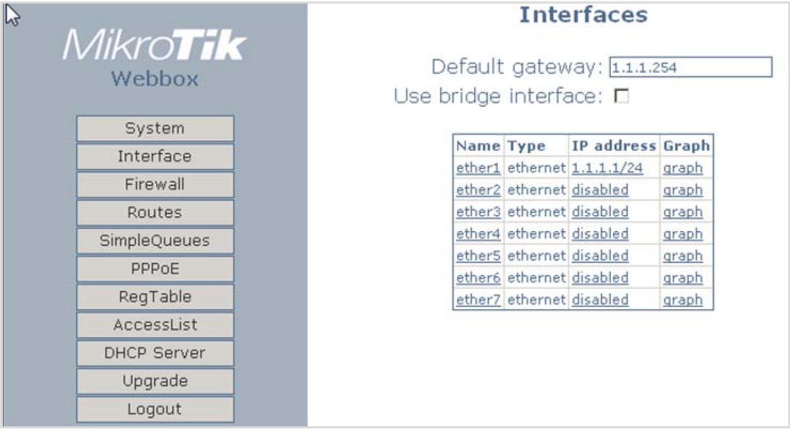
License

Mikrotik, RouterOS and the Mikrotik logo are registered trademarks of Mikrotikls SIA. Please read the license

As you can see, there are a number of options here. If you click on the WinBox image, you can download it, right from your RouterOS. The WebBox is the web based configuration, you will need to use the WebBox login at the top of the screen.

From here you also can open a telnet window, by clicking on the telnet section. Graphs are explained in detail in the "Graphing" section. You also have options for the on-line MikroTik documentation, as well as the licensing information directly from MikroTik's website.

In this section, we will discuss where items and features are at in WebBox, but not go into detail about the usages each one has. WinBox is the primary access method in many cases, so we will go much further in depth in those sections using WinBox.

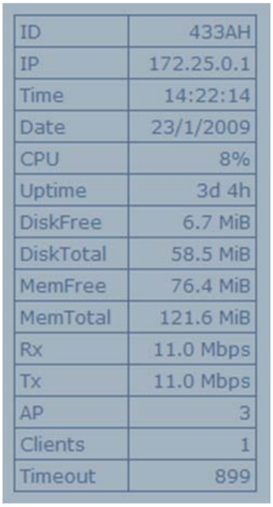


Interfaces

Default gateway: 1.1.1.254
Use bridge interface: □

Name	Type	IP address	Graph
ether1	ethernet	1.1.1.1/24	graph
ether2	ethernet	disabled	graph
ether3	ethernet	disabled	graph
ether4	ethernet	disabled	graph
ether5	ethernet	disabled	graph
ether6	ethernet	disabled	graph
ether7	ethernet	disabled	graph

Interfaces and IP addresses

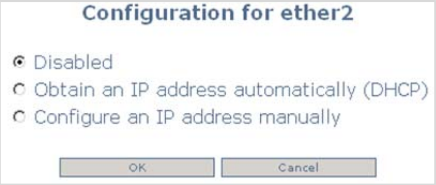


ID	433AH
IP	172.25.0.1
Time	14:22:14
Date	23/1/2009
CPU	8%
Uptime	3d 4h
DiskFree	6.7 MiB
DiskTotal	58.5 MiB
MemFree	76.4 MiB
MemTotal	121.6 MiB
Rx	11.0 Mbps
Tx	11.0 Mbps
AP	3
Clients	1
Timeout	899

Once logged into the WebBox you will be presented with a number of options. On the left side are the basic buttons for each section of the router. Below that, you will have a table that shows a number of statistics including your CPU usage, number of APs, clients, as well as other information.

If you click the interface name, you will be presented with an option to change the interface name. If you click the IP address, or disabled option under IP addresses for an interface, you will be prompted on how you wish to add IP addresses on that interface. You can have no IP address, or you can obtain an IP via DHCP. You can also configure an IP address manually. You can return to the interfaces section by selecting the Interface button on the left side of WebBox.

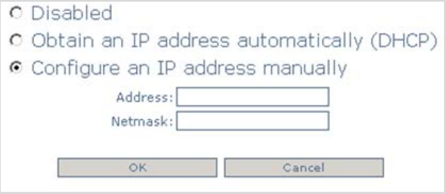
If you select DHCP, it will take you back to the main screen, and you may see ‘*searching…*’ as the IP address, as it is looking for a DHCP-Server. If you refresh the screen, it should change to an IP address as long as a DHCP-Server was found.



Configuration for ether2

● Disabled
○ Obtain an IP address automatically (DHCP)
○ Configure an IP address manually

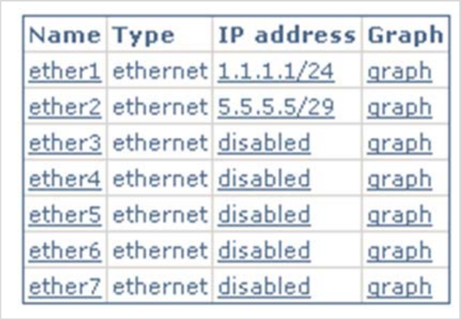
Configuring the interface manually is simple enough as well. Simply Enter your IP address, and in the Netmask, enter the Dotted Decimal Subnet mask, ex. would for example - 255.255.255.0. Once you have entered this information, go ahead and press OK.



○ Disabled
○ Obtain an IP address automatically (DHCP)
● Configure an IP address manually

Address:
Netmask:

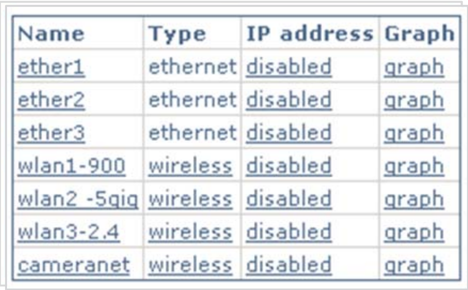
Note in the example on the right, you have the static IP addresses are shown. Here you can also click on *Graph* to view the interface graphing, if you have this enabled.



Name	Type	IP address	Graph
ether1	ethernet	1.1.1.1/24	graph
ether2	ethernet	5.5.5.5/29	graph
ether3	ethernet	disabled	graph
ether4	ethernet	disabled	graph
ether5	ethernet	disabled	graph
ether6	ethernet	disabled	graph
ether7	ethernet	disabled	graph

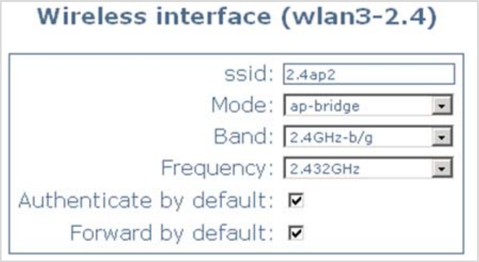
Wireless Interfaces

If you select a wireless interface, we can choose the type wireless to pull up the basic wireless information and configure the wireless interface.



Name	Type	IP address	Graph
ether1	ethernet	disabled	graph
ether2	ethernet	disabled	graph
ether3	ethernet	disabled	graph
wlan1-900	wireless	disabled	graph
wlan2 -5gig	wireless	disabled	graph
wlan3-2.4	wireless	disabled	graph
cameranet	wireless	disabled	graph

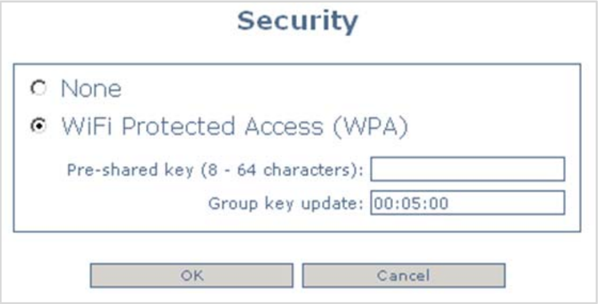
You can see the wireless interface settings and you can configure the basic options of your Wireless interface here. You can setup your SSID, Mode, and Band as well as what frequency to use. You can also disable or enable the default Authenticate and/or Forwards. You will also have options to specify a wireless security method as well.



Wireless interface (wlan3-2.4)

ssid: 2.4ap2
Mode: ap-bridge
Band: 2.4GHz-b/g
Frequency: 2.432GHz
Authenticate by default: ☑
Forward by default: ☑

You can specify either no security or Wi-Fi Protected access via WPA in the security section on your wireless interface as well. Note that you can enter your Pre-shared key or PSK, as well as your group key update. Note you cannot setup WPA2 in WebBox.

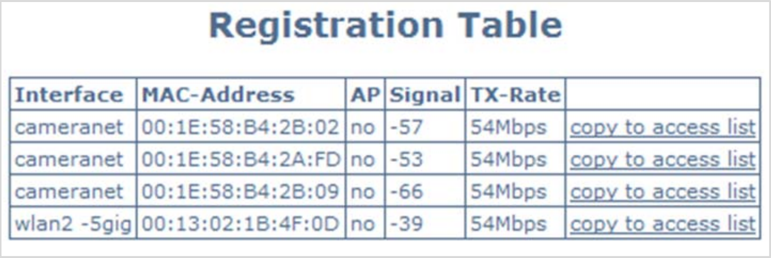


Security

○ None
● WiFi Protected Access (WPA)
Pre-shared key (8 - 64 characters):
Group key update: 00:05:00

Registration Table

The Registration Table button on the left side, gives you the ability to view the wireless registration table. This shows what interface wireless radios are connected to, as well what the MAC, signal level, TX-Rate and the ability to copy the MAC to the access list.

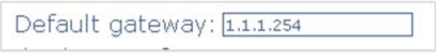


Registration Table

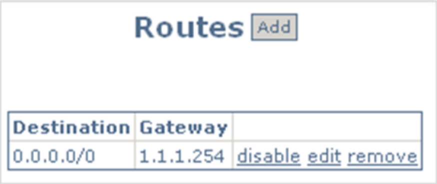
Interface	MAC-Address	AP	Signal	TX-Rate	
cameranet	00:1E:58:B4:2B:02	no	-57	54Mbps	copy to access list
cameranet	00:1E:58:B4:2A:FD	no	-53	54Mbps	copy to access list
cameranet	00:1E:58:B4:2B:09	no	-66	54Mbps	copy to access list
wlan2 -5gig	00:13:02:1B:4F:0D	no	-39	54Mbps	copy to access list

Routing

You can also specify the default gateway for your RouterOS system, right here by typing the default gateway IP address on the main interface page of WebBox. If you click on the Routes section on the left side, you will have the option to create other routes as well.



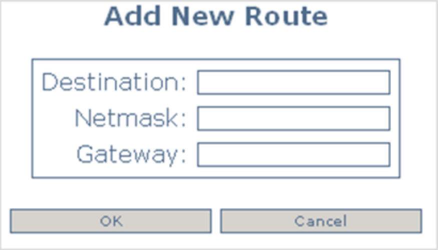
Default gateway: 1.1.1.254



Routes

Destination	Gateway	
0.0.0.0/0	1.1.1.254	disable edit remove

To add routes you can click on the add button. Once on the Add New Route screen, adding routes is as simple as specifying, the destination network, the dotted-decimal Netmask, as well as and the gateway to use.



Add New Route

Destination:

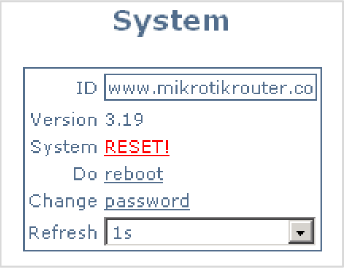
Netmask:

Gateway:

You can also disable, edit and remove routes by selecting the corresponding options.

System Options

Under the System option on the left side, you can setup the system ID; this is the identity of the RouterOS system. This screen also displays your version, allows you to reboot your RouterOS device, or allows you to change your user's password from this screen.



System

ID www.mikrotikrouter.co

Version 3.19

System RESET!

Do reboot

Change password

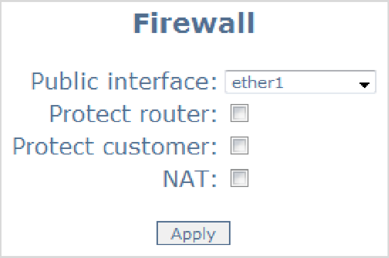
Refresh 1s

The refresh timer specifies how often to refresh the WebBox software page to show information such as usages, CPU time, etc.

There is also an option to perform a software reset; this resets the device to a factory default configuration. This will simply reset the configuration, not your licensing information. It does, however, reset your passwords and IP addresses.

Basic Firewall

Inside the RouterOS WebBox Firewall, you have a few simple options. You can specify a public Interface. Note that this is the ONLY time that you can specify a "public" interface. You also have a number of check boxes, to protect the router, the customer and to perform NAT out the public interface. Making these selections enters specific commands into RouterOS to perform the selected actions.



Firewall

Public interface: ether1

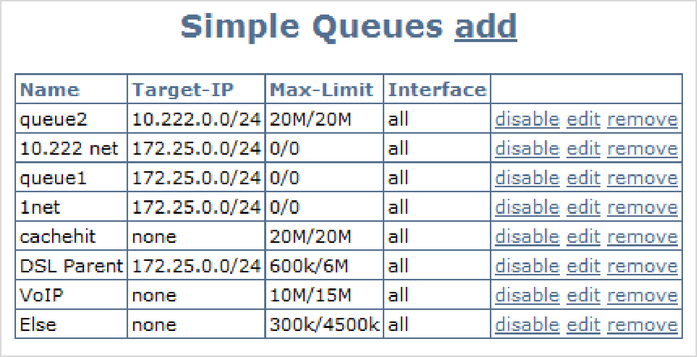
Protect router: □

Protect customer: □

NAT: □

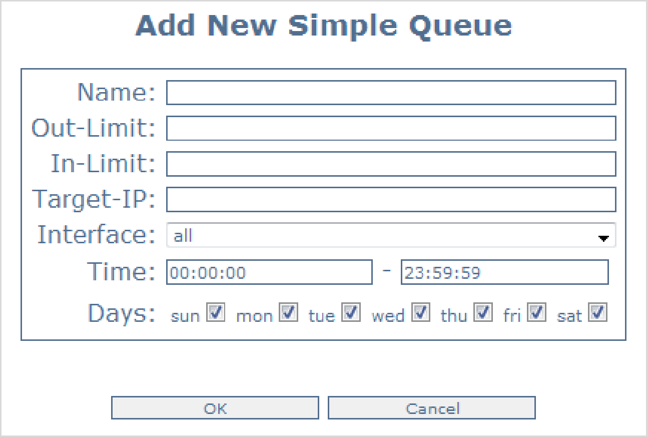
Simple Queues

Inside WebBox you can also specify simple queues. The interface is the same as specifying routes as well. Once you click on Add, you can specify a queue name, in and out limits, and your target IP. You can also specify time and days that the queue is effective.



Simple Queues add

Name	Target-IP	Max-Limit	Interface	
queue2	10.222.0.0/24	20M/20M	all	disable edit remove
10.222 net	172.25.0.0/24	0/0	all	disable edit remove
queue1	172.25.0.0/24	0/0	all	disable edit remove
1net	172.25.0.0/24	0/0	all	disable edit remove
cachehit	none	20M/20M	all	disable edit remove
DSL Parent	172.25.0.0/24	600k/6M	all	disable edit remove
VoIP	none	10M/15M	all	disable edit remove
Else	none	300k/4500k	all	disable edit remove



Add New Simple Queue

Name:

Out-Limit:

In-Limit:

Target-IP:

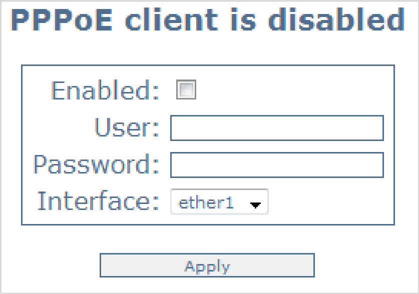
Interface: all

Time: 00:00:00 - 23:59:59

Days: sun ☑ mon ☑ tue ☑ wed ☑ thu ☑ fri ☑ sat ☑

PPPoE-Client

RouterOS has the ability to become a PPPoE-Client. In the PPPoE section, you can select if you wish to enable the PPPoE-Client. You will specify if you wish to enable the client, what interface it will run on and the username and password.

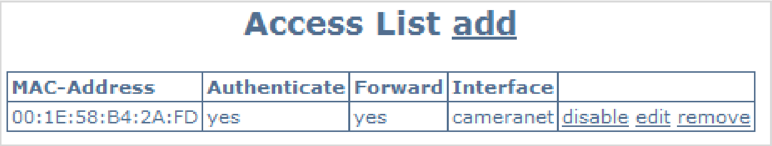


PPPoE client is disabled

Enabled: □
User:
Password:
Interface: ether1

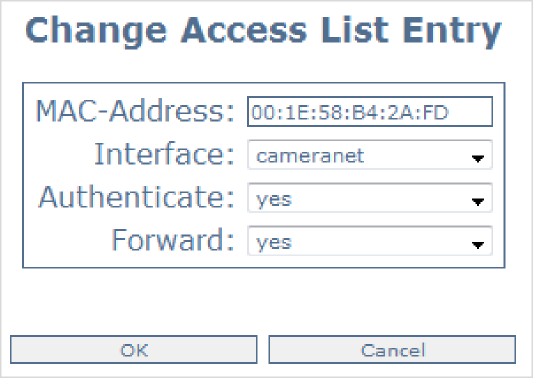
Access List

The RouterOS Access list specifies what interface and what MACs can either Authenticate or Forward. The interface defaults will apply if you do not have the MAC address in the access list. This is the basic MAC access control in MikroTik. Here you can add MAC addresses, select if you wish to authenticate or allow the client to forward as well. You can also specify an interface as well. It is possible to specify the same MAC on multiple interfaces; for example, one interface could not allow the client to register, while another might not, etc.



Access List add

MAC-Address	Authenticate	Forward	Interface	
00:1E:58:B4:2A:FD	yes	yes	cameranet	disable edit remove

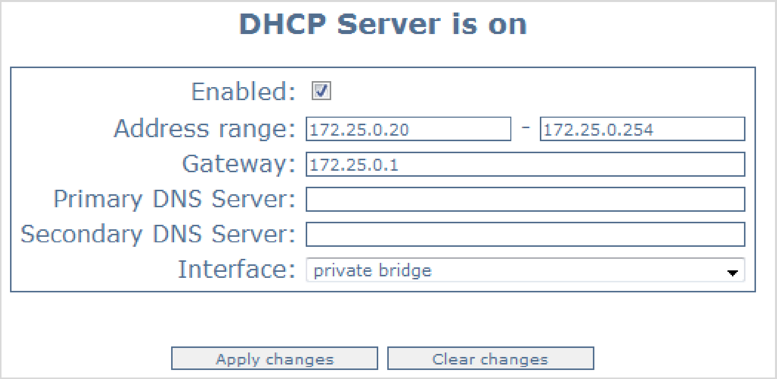


Change Access List Entry

MAC-Address: 00:1E:58:B4:2A:FD
Interface: cameranet
Authenticate: yes
Forward: yes

DHCP-Server

WebBox has options to specify basic DHCP-Server information. You can enable the DHCP-Server; specify the IP address range and gateway to hand out, and specify as well as the DNS servers to use. You will also need to specify the proper interface where you want DHCP enabled.



DHCP Server is on

Enabled: ☑

Address range: 172.25.0.20 - 172.25.0.254

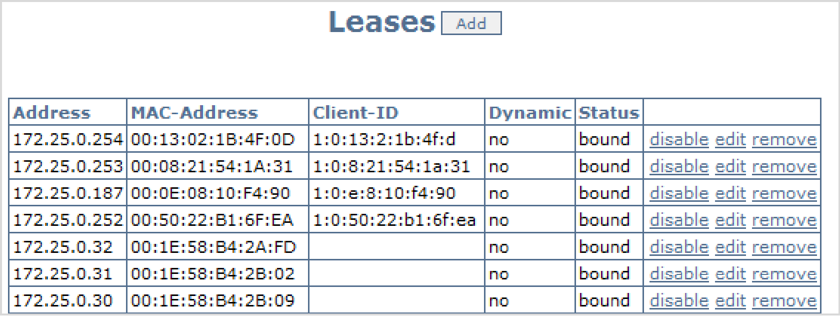
Gateway: 172.25.0.1

Primary DNS Server:

Secondary DNS Server:

Interface: private bridge

Below the DHCP-Server options is the lease information. You can view what MAC has what IP, as well as other information, and you can specify to add a static lease if you wish.

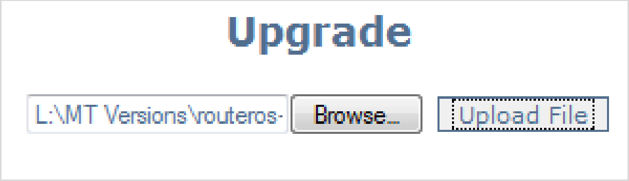


Leases

Address	MAC-Address	Client-ID	Dynamic	Status	
172.25.0.254	00:13:02:1B:4F:0D	1:0:13:2:1b:4f:d	no	bound	disable edit remove
172.25.0.253	00:08:21:54:1A:31	1:0:8:21:54:1a:31	no	bound	disable edit remove
172.25.0.187	00:0E:08:10:F4:90	1:0:e:8:10:f4:90	no	bound	disable edit remove
172.25.0.252	00:50:22:B1:6F:EA	1:0:50:22:b1:6f:ea	no	bound	disable edit remove
172.25.0.32	00:1E:58:B4:2A:FD		no	bound	disable edit remove
172.25.0.31	00:1E:58:B4:2B:02		no	bound	disable edit remove
172.25.0.30	00:1E:58:B4:2B:09		no	bound	disable edit remove

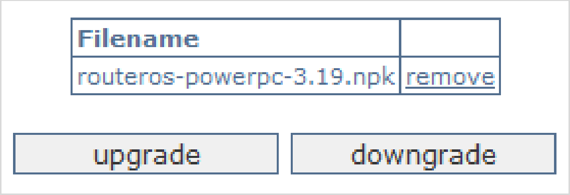
Upgrades

The upgrade button allows you to specify a NPK file, upload the file and upgrade your RouterOS device to the latest version. Be sure that you have the proper file for the CPU version of RouterOS that you are using.



L:\MT Versions\routeros-

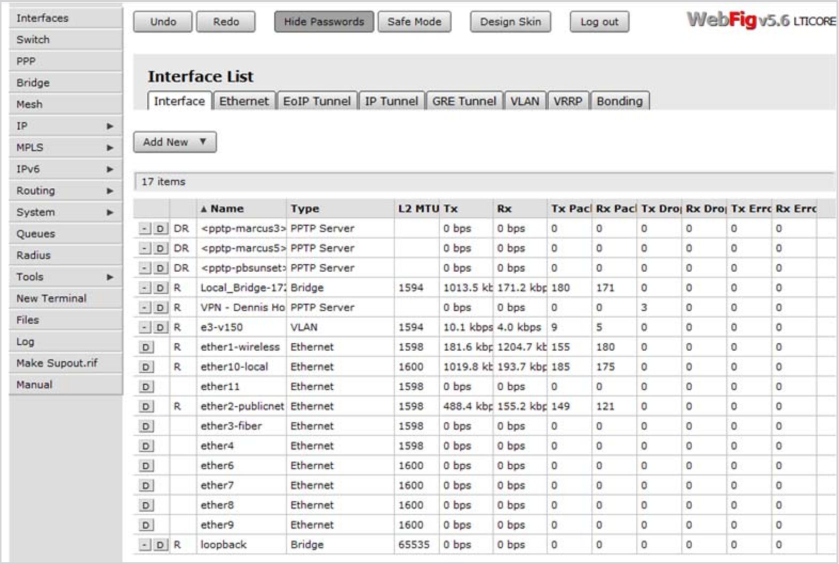
First specify the filename that you wish to upload. This will upload the file via the web browser. Once the file is uploaded, then you have the option to remove the file, upgrading it, or in some cases downgrading RouterOS versions. Click on whatever action that you wish to perform. Keep in mind that either function will require the RouterOS device to reboot.



Filename	
routeros-powerpc-3.19.npk	remove

WebFig

In RouterOS v5, MikroTik has introduced a new method of configuration, WebFig, on RouterOS. This is designed to be a web based configuration tool, which mimics the WinBox application.



Interface List

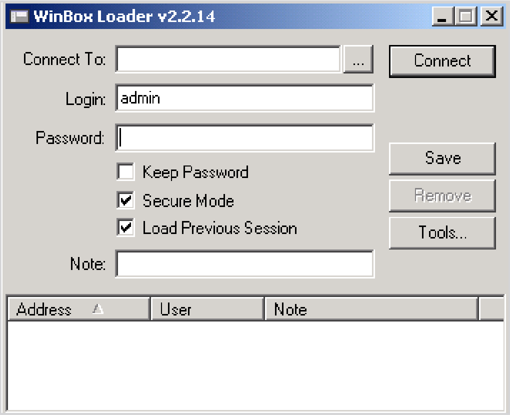
17 items

		▲ Name	Type	L2 MTU	Tx	Rx	Tx Pacl	Rx Pacl	Tx Dro	Rx Dro	Tx Errc	Rx Errc
- D	DR	<pptp-marcus3>	PPTP Server		0 bps	0 bps	0	0	0	0	0	0
- D	DR	<pptp-marcus5>	PPTP Server		0 bps	0 bps	0	0	0	0	0	0
- D	DR	<pptp-pbsunset:	PPTP Server		0 bps	0 bps	0	0	0	0	0	0
- D	R	Local_Bridge-17:	Bridge	1594	1013.5 kt	171.2 kbr	180	171	0	0	0	0
- D	R	VPN - Dennis Ho	PPTP Server		0 bps	0 bps	0	0	3	0	0	0
- D	R	e3-v150	VLAN	1594	10.1 kbps	4.0 kbps	9	5	0	0	0	0
D	R	ether1-wireless	Ethernet	1598	181.6 kbr	1204.7 kt	155	180	0	0	0	0
D	R	ether10-local	Ethernet	1600	1019.8 kt	193.7 kbr	185	175	0	0	0	0
D		ether11	Ethernet	1598	0 bps	0 bps	0	0	0	0	0	0
D	R	ether2-publicnet	Ethernet	1598	488.4 kbr	155.2 kbr	149	121	0	0	0	0
D		ether3-fiber	Ethernet	1598	0 bps	0 bps	0	0	0	0	0	0
D		ether4	Ethernet	1598	0 bps	0 bps	0	0	0	0	0	0
D		ether6	Ethernet	1600	0 bps	0 bps	0	0	0	0	0	0
D		ether7	Ethernet	1600	0 bps	0 bps	0	0	0	0	0	0
D		ether8	Ethernet	1600	0 bps	0 bps	0	0	0	0	0	0
D		ether9	Ethernet	1600	0 bps	0 bps	0	0	0	0	0	0
- D	R	loopback	Bridge	65535	0 bps	0 bps	0	0	0	0	0	0

The WebFig is intended to replace WinBox as a web based configuration. Looking very much like WinBox, WebFig should be able to be used across all platforms, including Linux and MacOS vs. The existing WinBox application only runs only under Windows and Wine on Linux systems. You will need to refer to the “Using WinBox” section, as the operation of WebFig is designed similar to WinBox.

Using WinBox

RouterOS has a great utility, WinBox, that comes free of charge, and which allows you to have a graphical interface for RouterOS. You can download WinBox you can download from MikroTik's website, or, if you have IP access to your router, you can use your web browser and connect to the IP address of your RouterOS system. This page will allow you to download a version of WinBox. I do recommend that you visit their MikroTik website for the latest version though. The webpage will deliver the latest version, if you have the latest RouterOS version on your router.



WinBox Loader v2.2.14

Connect To:

Login: admin

Password:

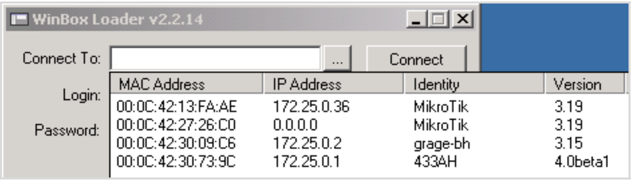
☐ Keep Password

☑ Secure Mode

☑ Load Previous Session

Note:

WinBox uses either the Router's MAC address or an IP address to communicate. In IP mode, it will use TCP port 8291 for the connection to the router. You can enter the MAC or IP address in the "Connect To" box or you can browse for this. There is a button with three periods (ellipsis), to the right of the Connect To box. By clicking on this box, WinBox will use the MNDP packets sent out from RouterOS devices on the local network, and display the MAC addresses of the devices found.



WinBox Loader v2.2.14

Connect To:

Login:

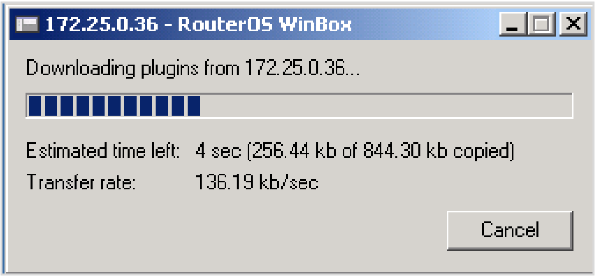
Password:

MAC Address	IP Address	Identity	Version
00:0C:42:13:FA:AE	172.25.0.36	MikroTik	3.19
00:0C:42:27:26:C0	0.0.0.0	MikroTik	3.19
00:0C:42:30:09:C6	172.25.0.2	grage-bh	3.15
00:0C:42:30:73:9C	172.25.0.1	433AH	4.0beta1

Inside the WinBox Display screen, you have several pieces of information, including the MAC address of your RouterOS device, the Identity and the IP on the interface closest to your PC and the RouterOS version well.

If you click on a MAC address, it will place that MAC into the Connect To window for you. If you click on the IP Address, it will place the IP address into the Connect To window. Be sure that you have IP connectivity if you use the IP address, otherwise, WinBox will use the MAC address to connect. Make sure you have the proper username and password.

NOTE: That the MAC address connect feature, really should be used only to get an initial IP onto your RouterOS device. Some functions, such as file transfers etc., are problematic at best while connected with a MAC address through WinBox.



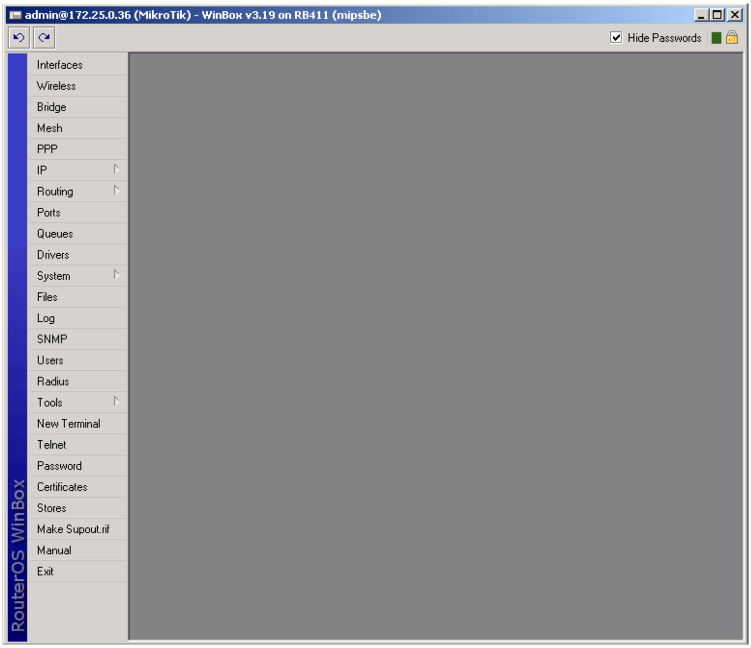
172.25.0.36 - RouterOS WinBox

Downloading plugins from 172.25.0.36...

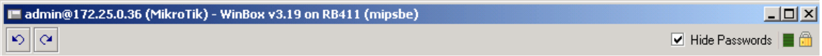
Estimated time left: 4 sec (256.44 kb of 844.30 kb copied)

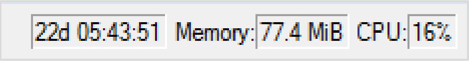
Transfer rate: 136.19 kb/sec

Upon connecting you may need to download the plug-ins from the RouterOS device. This typically should be very quick. Once it is done, it will open the full WinBox Graphic Interface.

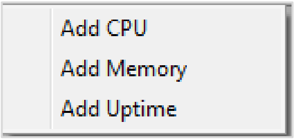


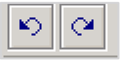
As you can see there are a number of options inside WinBox.


admin@172.25.0.36 (MikroTik) - WinBox v3.19 on RB411 (mipsbe)
☑ Hide Passwords



22d 05:43:51 Memory: 77.4 MiB CPU: 16%





At the top of WinBox in the title bar are a number of details. The username@IP or MAC address of the RouterOS device will be listed at the top. Next, the system identity is displayed, then the WinBox title, along with the current RouterOS version number and what RouterBOARD or system the RouterOS device is. Next to that is the CPU type.

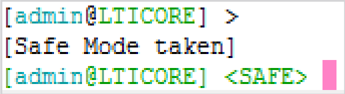
The two arrows on the left side of the screen are undo and redo command buttons. On the right side, we have options to hide passwords, a small green box that shows the CPU load, and a lock to show if we are logged in securely to the RouterOS or not. This is determined inside the WinBox Application before you connect.

Between the redo and undo commands and the hide password option, you have a nice long blank bar. If you right-click in there you will have the options to add some other common stats. You can add CPU, Free Memory and Uptime information to your top bar. As you can see below, it will show this information in your task bar. You can also right-click again and remove each one of these as needed.

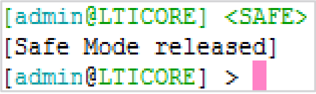
Safe Mode

In V5rc6, MikroTik added a new feature in the WinBox system. This feature was available in the command line for an extremely long time, however, it was finally added into the WinBox system upon the version 5 release candidate 6 update. Now, in the upper left of WinBox, near your undo and redo buttons, we now have a "Safe Mode" Option . Safe Mode is a simple operations mode that says, after activing Safe Mode, any changes that you make will take effect, however, if you loose your WinBox connection, (i.e., if your changes causes you to loose connection to the router via WinBox) then those changes from after you activated safe mode will be removed. This hopefully gives you the option of logging back into the WinBox and reentering your changes correctly without loosing connecitivity.

Prior to v5rc6, we had to use safe mode via the command line. To use this, it was quite simple. We hit [CTRL-X] from a command line or a terminal window. This would to enter Safe Mode. Again the feautre works as described above. To exit Safe Mode in the command line, simply hit [CTRL-X] a second time. This will release Safe Mode and all of your changes will be saved. As another option though, You can also hit [CTRL-D], to this will release safe mode, but any changes to the router during this time will be discarded.



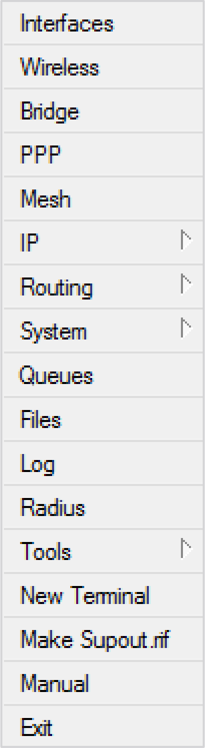
```
[admin@LTICORE] >
[Safe Mode taken]
[admin@LTICORE] <SAFE>
```

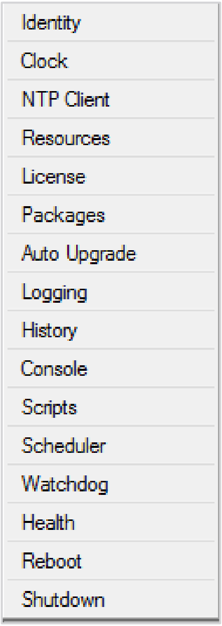


```
[admin@LTICORE] <SAFE>
[Safe Mode released]
[admin@LTICORE] >
```

WinBox Menus

WinBox is organized into different menus, that allow you to access each of the RouterOS features from. For example, the Interfaces menu will give you access to the interface options, and settings, while the IP menu choice, will gives you access to the IP-related commands and features. There are a number of features that go directly to several other menu choices as well.

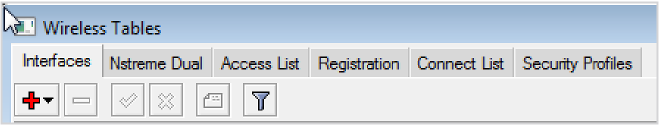




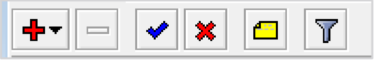
RouterOS orginizes its features inside a directory-structure-like system. Each object on the WinBox menu, has other sub-menus. For instance, if you click on System, you will get the menu to the right. Notice, that we can setup parameters such as clock settings, view system resources and even reboot or shutdown the system.

This system is mirrored in the command line interface. We discuss more command line options and features in the command line section. The simplest method of understanding this is by using the menu structure. If you wished to access the system reboot command, in WinBox you click, System, then Reboot. In the command line, you would type, *system reboot*.

WinBox also uses sub sections via tabs. In the wireless section, we see a number of tabs and each tab represents another level of commands. Below, you will see we have interfaces, access-lists, and other tabs. In the command line these are represented just like folders again. If you wanted to see the wireless interfaces, in WinBox you would click on Wireless, then click on the Interfaces tab. MikroTik just thought it would be better to have a tab approch for these items versus having a listing like in the system command. In the command line, you would simply type, *wireless interfaces*.

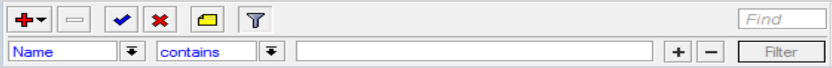


WinBox Interface Options

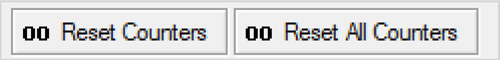


Inside each of these tabbed sections, and all throughout RouterOS, you will find these icons. The left-most icon is an Add icon. If you have a small down arrow, then there are other options than in addition to Add. In the wireless section, you can add Virtual APs or WDS links. In the Ethernet section; you have the ability to add VLANS or other types of interfaces that pertain to the associated section. You will also see these arrows on different types of rules; again, they mean the same thing.

The Minus icon is for removing an object. If you have a VLAN that you wish to remove, you can highlight the item, and then remove it by using this icon. The Check Mark and X, are used to enable and disable the object. These again, will show up in many different locations in RouterOS and their function is the same. The Note or Comment button is next and this will allow you to add a comment to the object you have selected. This may be an interface or a Firewall rule as well.

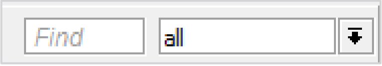


The Filter button is the rightmost icon. This allows you to filter your objects in the list by some method. Depending on the location in RouterOS, you may be able to filter based on Name, MAC, or maybe Action type, SRC address, etc. You can filter several ways as well, by selecting if it contains, does not contain, or is or is not as well. You can type in the text that you wish to filter. You also have a Plus and Minus button to the left of the text. This will add or remove another filter, so that you can filter your objects by several different criteria.



You will also find sections in RouterOS that contain counter resets. In some sections, such as Firewall rules, you will have counters that count packets or bytes. If you select an object, you can reset that individual object's counters with the Reset Counters button. If you wish to reset all counters in the list, you can use the Reset All Counters button.

Some sections may have a Find, as well as a dropdown listing of some type. We will cover each of the dropdowns as we get to that section. The Find will find the selected text and highlight it in the object window below to help you locate objects with certain text.

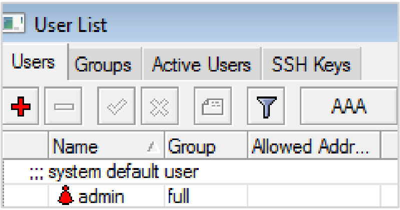


Managing RouterOS

In this section we will cover how to manage your RouterOS installation. This includes managing user's access to your router, controlling basic services that your RouterOS offers, and managing the logging that your RouterOS system generates. This is sometimes a full time job if you have quite a few of RouterOS Routers out there. If you use MikroTik's Dude Application, covered in the Dude section, then you will have some great abilities to help manage large numbers of systems.

User Management

RouterOS has a built in user management system, this is located under the Users section of RouterOS. This system was moved when WinBox was revamped to accommodate smaller screen resolutions. Currently, the user's session is under system → users. However, in the command line, it is simply under */users*.

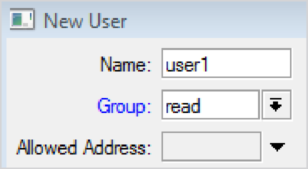


Name	Group	Allowed Addr...
;;; system default user		
admin	full	

In the user section, you will have a number of tabs, just like the rest of RouterOS. These tabs include the list of users, the groups of users, current active users, and any SSH Keys that you generate.

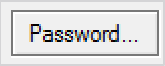
Adding/Removing/Changing Local Users

RouterOS provides you with a user list for router management. This list is on the users tab inside your user list. You can add, remove, disable, and enable users just like any other table object in RouterOS.



Name: user1
Group: read
Allowed Address:

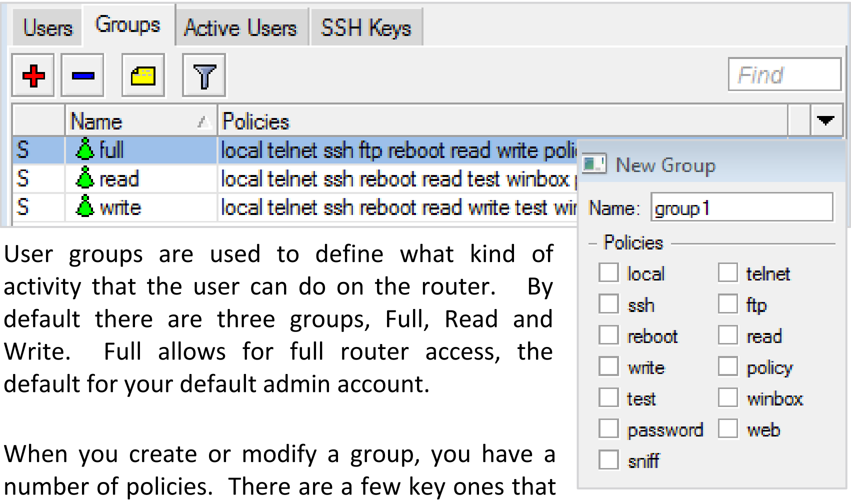
When creating users here, you have to create the username, and select an access group that you want the user to be in. The allowed address is the IP or subnet that you will allow that user to login with. This of course, is only for Layer 3 connectivity. Once you create this user, you will need to



setup a password. I typically would hit apply and then click on the password button to set the password for the user. This is the same process that you would use to reset a user password as well.

You cannot see the passwords for these users, you can NOT see these. You can reset them, by using the password button, but you can't unhide them or view the user passwords in any way. This is done so that another user that logs in can't view passwords. If they make a change, you will know, because the passwords don't work. The idea is that at least you know that a change was made, instead of someone getting the admin username and password, and logging in without your knowledge.

RouterOS User Groups



	Name	Policies
S	full	local telnet ssh ftp reboot read write poli
S	read	local telnet ssh reboot read test winbox p
S	write	local telnet ssh reboot read write test wir

New Group

Name: group1

Policies

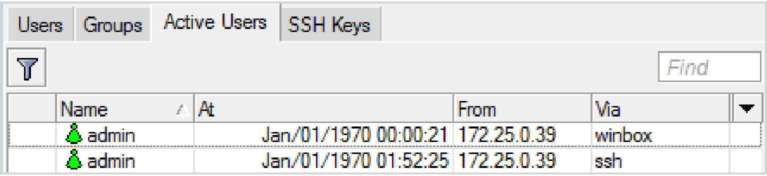
□ local □ telnet
□ ssh □ ftp
□ reboot □ read
□ write □ policy
□ test □ winbox
□ password □ web
□ sniff

User groups are used to define what kind of activity that the user can do on the router. By default there are three groups, Full, Read and Write. Full allows for full router access, the default for your default admin account.

When you create or modify a group, you have a number of policies. There are a few key ones that you should know about. Reboot will allow a user with this right to reboot your RouterOS system. Password allows you to see or unhide passwords inside RouterOS. Sniff allows the users to access the packet sniffing features of RouterOS. The last one I recommend you knowing about is the policy. This one allows users to change user settings, such as adding users, etc.

Active Users

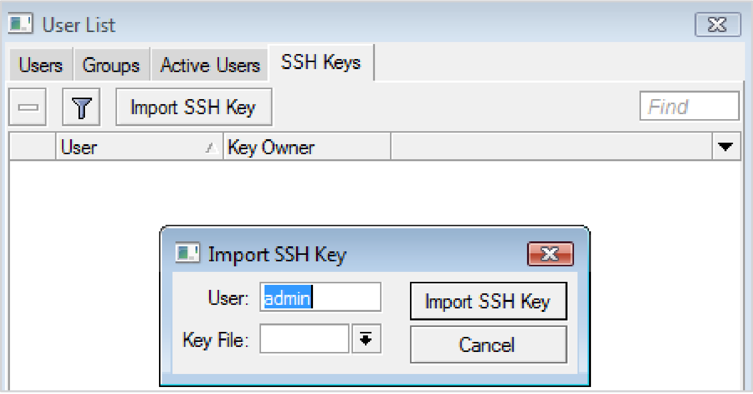
The active user section simply shows you what current active users are connected to your router. In this case, we have a WinBox connection from an IP. We also have a SSH connection from the same IP address.



Name	At	From	Via
admin	Jan/01/1970 00:00:21	172.25.0.39	winbox
admin	Jan/01/1970 01:52:25	172.25.0.39	ssh

SSH Keys

SSH keys are used to authenticate sessions without using a username/password. When most SSH sessions are started, a shared SSH key is generated to secure the communications between the client and the RouterOS system. In the SSH Keys section, we have the option to import a SSH that is tied to a username. This would be done instead of having the key generated. Import these by clicking the Import SSH key button, then specifying what user will use this key, and select the key file. You will have to have uploaded your key file already. See the Files section of Managing RouterOS for information on how to do this.

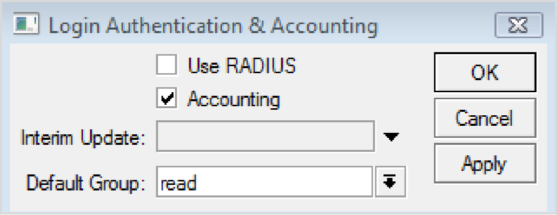


User: admin

Key File:

Once you have imported your key, you can use your DSA key on your client without having to login. It will use that key with that user.

AAA Settings – Radius RouterOS Users



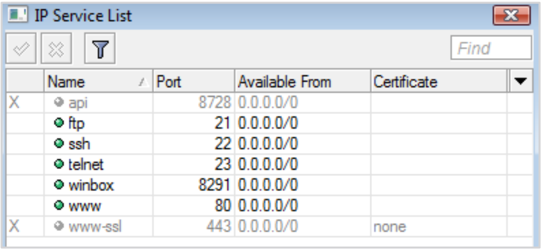
☐ Use RADIUS
☑ Accounting
Interim Update:
Default Group: read

With the AAA system, you can set your RouterOS to use a Radius server to allow users to login. By doing this, you can have a centralized Radius system for router management. The users that you have in the RAIDUS system can access your routers and make changes, but you are not giving out the default Admin passwords to your engineers and techs. This will help you in a large scale deployment of RouterOS. One thing to keep in mind when you do this; you typically will need to create a local group, that allows everything but the Policy function. That way other users that login via Radius cannot change the users locally in the router.

RouterOS Services

RouterOS has a number of services that it runs right out of the box. These services allow remote access, and management of your router. Some of these services include your WinBox and WebBox access.

By going to IP → Services, you will be able to turn on and off these services as well as change ports, and change from the IPs addresses that they are available from. Each one of these objects can be turned off or on, by disabling or enabling them. By double-clicking on one, you will get the individual item context window. This will give you options to select what port you wish it to run on. By default these ports are setup to the most common port numbers.



	Name	Port	Available From	Certificate
X	api	8728	0.0.0.0/0	
	ftp	21	0.0.0.0/0	
	ssh	22	0.0.0.0/0	
	telnet	23	0.0.0.0/0	
	winbox	8291	0.0.0.0/0	
	www	80	0.0.0.0/0	
X	www-ssl	443	0.0.0.0/0	none

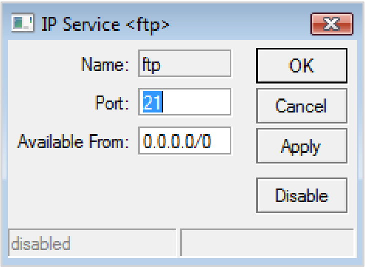
You also have the ability to setup the Available From field. This field allows you to restrict access to the selected service down to an IP or a subnet range. If you wished to only allow 192.168.0.0/16 IPs to access your FTP server, you would enter 192.168.0.0/16 into the Available From field. A good recommendation though is to disable any unused services. I have found that on larger networks, there are multiple, non-sequential IP ranges for management, thus, I typically will use my Firewall to restrict access by admin ranges.

Securing Services

As mentioned in the above section, I would really recommend that you use your IP Based Firewall to properly secure your RouterOS services.

FTP Service

FTP is used to allow the transfer of files to and from RouterOS. There are however, other ways as well to transfer files that do not rely on a technology that is outdated and/or routinely scanned. By default, your FTP server is turned on; I recommend turning it off! To do this, simply disable it in the object list under IP → Services.



IP Service <ftp>

Name: ftp

Port: 21

Available From: 0.0.0.0/0

disabled

API Service

RouterOS offers an Application Programming Interface. This interface allows you to create custom applications to program your routers. This service is turned off by default, but just like the rest of the services, you can change the default port from 8728 to another port, and change the Available From IP or IP range.

SSH / Telnet Services

Just like other routers, you can SSH or telnet into the command line interface. Using telnet, the information, like your username/password is sent in clear text. I would recommend turning off telnet, and only allow SSH. SSH sessions generate a key that will be used to secure the communications between your SSH Client application and your router. The default port for SSH is 22, and it is commonly scanned. If possible, change this port or use the availability list to secure this further.

WWW Service / WWW-SSL Service

This allows you to access your WebBox application, as well as the on-line graphing etc. Here, I normally do not change the port, unless I don't want someone seeing this. If you need for this router to be more secure, I would turn this off and just use SSH and WinBox to manage the RouterOS. You can change the default port to whatever you wish.

The WWW-SSL service allows this system to be accessed via HTTPS. For the webpage to function with a SSL certificate you must have imported the certificate already. Once the certificate is imported, it should be an option in the drop down menu in the SSL Service, allowing you to select the proper SSL certificate to use. This will allow you to run SSL on the web server.

WinBox Service

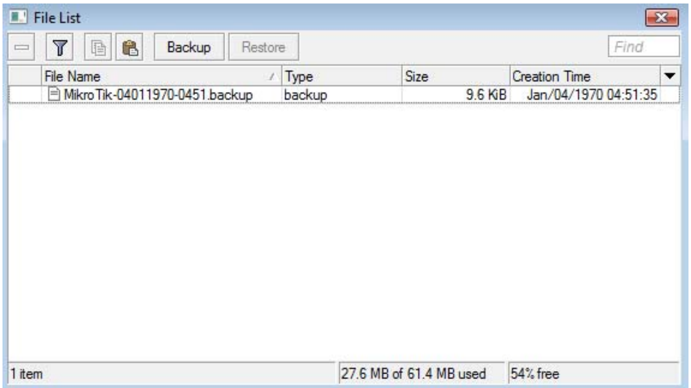
WinBox by default runs on port 8291. Inside the IP Services system, you can change this port as well as change where it is available. Normally, I would secure this with Firewall rules, like other items. I typically though, leave it on the default port.

Working with Files

RouterOS offers two different ways to manage files on your Router. The original way for several versions, was to simply FTP files up and down via the FTP service. You can connect via a standard FTP client, using your admin username and password that you setup on the router, and then transfer the files as you need. The files that you would typically transfer are packages or RouterOS NPK version files. You would also commonly transfer hotspot files as well. This method is quick and painless, but does require you to have a FTP client program loaded on your computer.

The best way to transfer files though, is through WinBox.   WinBox allows you to transfer files and even entire directory structures. This works quite well, and does not have an extra port or non-secure protocol to transfer.

To view your files in WinBox, simply click Files.

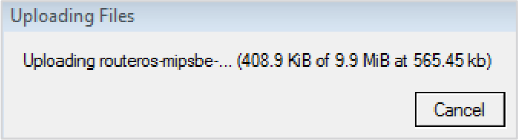


File Name	Type	Size	Creation Time
MikroTik-04011970-0451.backup	backup	9.6 KiB	Jan/04/1970 04:51:35

1 item | 27.6 MB of 61.4 MB used | 54% free

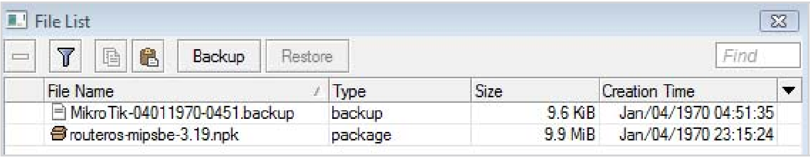
As you can see, you have information, such as how many items that is inside your file system, as well as space information on the bottom of the window. You can also select an object and delete it by using the minus button at the top.

Getting files into the file system of RouterOS is simple. You can use the FTP service to upload or download files as needed. But RouterOS and your WinBox application is smarter than that. You can simply drag and drop files from a folder on your desktop, right into the file list window!



Uploading routeros-mipsbe-... (408.9 KiB of 9.9 MiB at 565.45 kb)

Below, you can see that we have uploaded an .npk file. This is a MikroTik Package file that allows your RouterOS to either install or upgrade the OS or packages. You can simply drag and drop it from your file system right into the file list window.



File Name	Type	Size	Creation Time
MikroTik-04011970-0451.backup	backup	9.6 KiB	Jan/04/1970 04:51:35
routeros-mipsbe-3.19.npk	package	9.9 MiB	Jan/04/1970 23:15:24

Clicking the Backup button created the .backup file. You can now simply, click and drag onto your desktop or into a file folder. It will then download from RouterOS.

When downloading and uploading files in RouterOS through WinBox, you will typically need an IP or Layer 3 connection. Sometimes the Layer 2 connection can be a bit flaky, and unreliable. I have seen instances where it will stall, stop and hang on some computers. I would recommend putting an IP address on your Router, then connecting with the IP address through WinBox and then uploading your files.

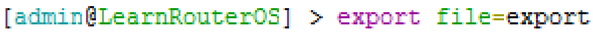
Backup / Restore

In the File List window, you also have Backup and Restore options. Backing up RouterOS is as simple as clicking the Backup button. When you click the Backup button, you will see that there is a .backup file created. This is the backup file for your RouterOS. Restoring this file is as simple as uploading the file, selecting the file and clicking on Restore.

There are a few things that you should know about backups that I would like to share. The .backup files are the best way to do backups in general. They will restore on the same hardware platform without issues, however, if you have an older platform, and the chances of you replacing that older platform with a newer one in the event of a failure is high, and then I would suggest also making a text backup. **The .backup files are not editable, they are a binary file that is proprietary to RouterOS, so you can't see inside them, view the configuration etc.** If you have a unit that you wish to make a change to, you can create a backup file and make the change. Reverting is as simple as uploading the file and doing the restore.

Creating Editable Text Backup Files

Creating editable backup files is very easy, but you can't do it in the graphic interface. You will need to start a terminal window. Do this by selecting New Terminal on the left side of WinBox. At the command prompt, type *export file=exportfilename*. You can change the export file name to whatever you wish.



```
[admin@LearnRouterOS] > export file=export
```

Once you export the file, you can go to the file listing, and see that there is an export.rsc.

export.rsc	script	11.6 KiB

Now you can take this file, just like a backup file or other files, and download it in WinBox. If you open this file, in any text editor, you will see:

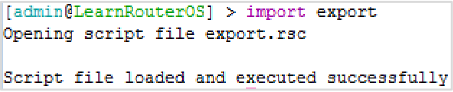
```
/interface bridge
add admin-mac=00:00:00:00:00:00 ageing-time=5m arp=enabled
auto-mac=yescomment="" disabled=no forward-delay=15s max-
message-age=20s mtu=1500 name=bridge1 priority=0x8000 protocol-
mode=stp transmit-hold-count=6

/interface Ethernet set 0 arp=enabled auto-negotiation=yes
comment="" disabled=no full-duplex=yes mac-
address=00:0C:42:32:22:17 mtu=1500 name=Ethernet 1
speed=100Mbps set 1 arp=enabled auto-negotiation=yes
bandwidth=unlimited/unlimited comment="" disabled=no full-
duplex=yes mac-address=00:0C:42:32:22:18 master-port=none
mtu=1500 name=Ethernet 2 speed=100Mbps

set 2 arp=enabled auto-negotiation=yes
bandwidth=unlimited/unlimited comment="" disabled=no full-
duplex=yes mac-address=00:0C:42:32:22:19 master-port=none
mtu=1500 name=Ethernet 3 speed=100Mbps

/interface vlan add arp=enabled comment="" disabled=no
interface=Ethernet 2 mtu=1500 name=vlan100.2 vlan-id=100
```

This is the command line representation of the programming and configuration that you have on your RouterOS. You can take sections of this, and paste them into the terminal window to copy configuration. Doing this for the entire script will not work. However, since you can read the configuration, you can use this to base other configurations and/or reconfigure other units.

Note: the above italic text was edited for formatting. Your output could contain the same information; just the formatting may be different.

Importing Scripts

Once you get real good at reading and doing command-line interfaces, you can start creating scripts, or RSC files that you can bring right into RouterOS. You will need to



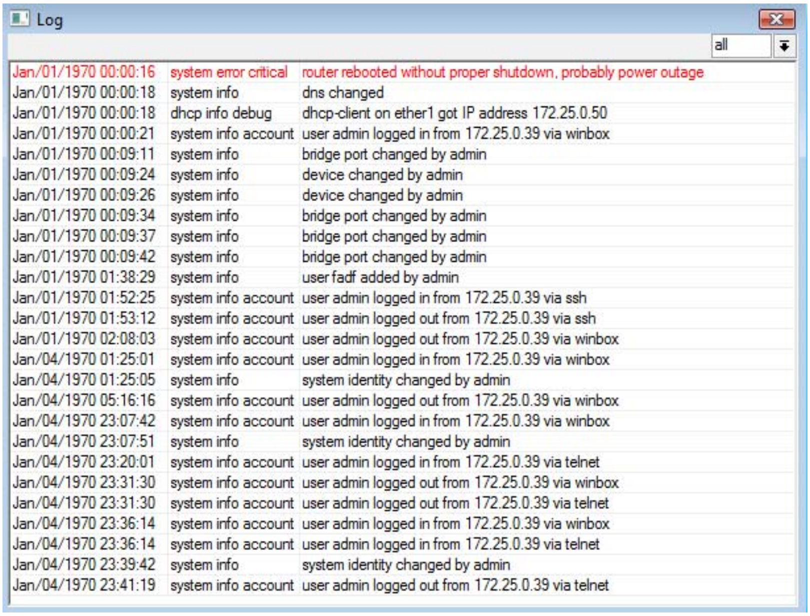
```
[admin@LearnRouterOS] > import export
Opening script file export.rsc

Script file loaded and executed successfully
```

create this file, and of course test and test again. Once you have it just the way you want it, then go ahead and upload the file. Of course you can simply paste it right into the terminal window, but you can also import the file in the command line. To use this feature, you simply type *import filename*. You will need to be at the root in the command line interface for this to work.

Logging

Just like with other routing systems, you have logging capabilities. You will use this to review access to the router, make changes and even show packets that you may be dropping or changing. We also have options to send your logging data out to a Syslog server, like the one contained in MikroTik's The Dude application, or other standardized Syslog servers. Debugging information also can help you diagnose issues, such as Radius, and hotspot.

To access your log in WinBox, simply click Log on the left menu.

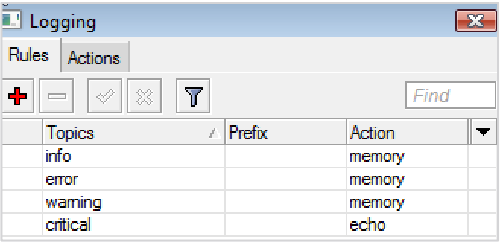


```
Jan/01/1970 00:00:16  system error critical  router rebooted without proper shutdown, probably power outage
Jan/01/1970 00:00:18  system info            dns changed
Jan/01/1970 00:00:18  dhcp info debug        dhcp-client on ether1 got IP address 172.25.0.50
Jan/01/1970 00:00:21  system info account    user admin logged in from 172.25.0.39 via winbox
Jan/01/1970 00:09:11  system info            bridge port changed by admin
Jan/01/1970 00:09:24  system info            device changed by admin
Jan/01/1970 00:09:26  system info            device changed by admin
Jan/01/1970 00:09:34  system info            bridge port changed by admin
Jan/01/1970 00:09:37  system info            bridge port changed by admin
Jan/01/1970 00:09:42  system info            bridge port changed by admin
Jan/01/1970 01:38:29  system info            user fadf added by admin
Jan/01/1970 01:52:25  system info account    user admin logged in from 172.25.0.39 via ssh
Jan/01/1970 01:53:12  system info account    user admin logged out from 172.25.0.39 via ssh
Jan/01/1970 02:08:03  system info account    user admin logged out from 172.25.0.39 via winbox
Jan/04/1970 01:25:01  system info account    user admin logged in from 172.25.0.39 via winbox
Jan/04/1970 01:25:05  system info            system identity changed by admin
Jan/04/1970 05:16:16  system info account    user admin logged out from 172.25.0.39 via winbox
Jan/04/1970 23:07:42  system info account    user admin logged in from 172.25.0.39 via winbox
Jan/04/1970 23:07:51  system info            system identity changed by admin
Jan/04/1970 23:20:01  system info account    user admin logged in from 172.25.0.39 via telnet
Jan/04/1970 23:31:30  system info account    user admin logged out from 172.25.0.39 via winbox
Jan/04/1970 23:31:30  system info account    user admin logged out from 172.25.0.39 via telnet
Jan/04/1970 23:36:14  system info account    user admin logged in from 172.25.0.39 via winbox
Jan/04/1970 23:36:14  system info account    user admin logged in from 172.25.0.39 via telnet
Jan/04/1970 23:39:42  system info            system identity changed by admin
Jan/04/1970 23:41:19  system info account    user admin logged out from 172.25.0.39 via telnet
```

In the log, you have the date/time, as well as what system generated the log and the actual event information.

Setting Logging Rules

Logging options are setup in System → Logging.



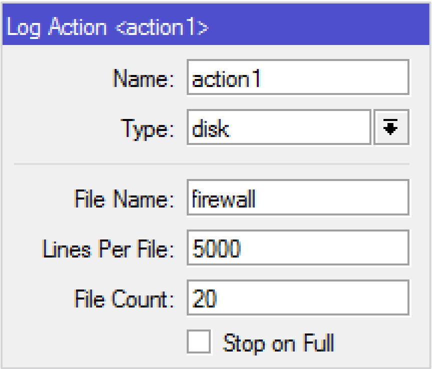
Topics	Prefix	Action
info		memory
error		memory
warning		memory
critical		echo

Here you can setup how your logs are stored, where they go, and what you wish to log. Most of the topics that are included in RouterOS are really debugging information. Normally you would not need to see all of the Radius information on a Radius request, however, seeing this information, may show you that your Radius server is not responding, or show that there is not a profile on your RouterOS that corresponds to the one sent in Radius.

Under your Logging Rules, you have objects that you can add, remove, disable and enable just like any other object in RouterOS. The default logging options are listed above. This is what your RouterOS system will come with on a fresh load. These are the minimum that I would have on a Router. The ones that I would use normally are Radius and hotspot logging.

Logging to Disk/File

You can also log data out to disk files. These will be stored on your system disk. You will have options such as what name of the files name, number of many lines per file as well as how many files. You can also stop the logging once the file count has been reached.



Name: action1
Type: disk
File Name: firewall
Lines Per File: 5000
File Count: 20
□ Stop on Full

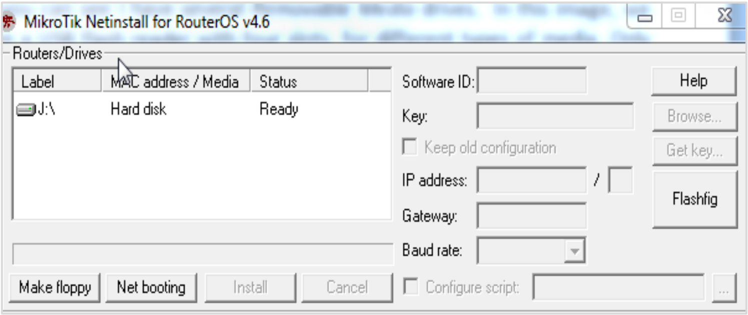
Once you setup your logging action, it will create a *filename*.0.txt file in your files menu; this is the log file that is currently being written to. You can then download these as you wish.

Flashfig

Flashfig was introduced in a v4.6 of NetInstall. This utility is designed to allow you quickly program RouterOS systems. This program, applies a "configuration" or script within typically 2-3 seconds! This is a windows application, which runs inside the NetInstall utility. It is important to note that this utility requires Flashfig support on the RouterBOARD. All RouterBOARDs support this, however, by default; only boards manufactured after the March 2010 have it enabled by default.

Enabling Flashfig on RouterBOARDs.

There are two ways to enable Flashfig on your RouterBOARDs. The simplest, is to load RouterOS, and connect to a terminal window. Inside your terminal window, you can simply select the boot device.



Label	MAC address / Media	Status
J:\	Hard disk	Ready

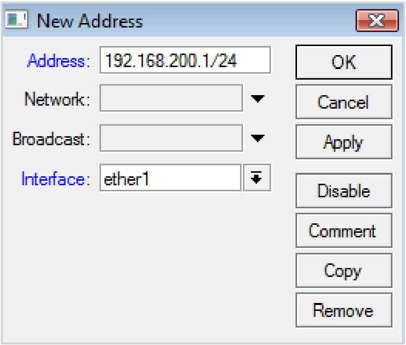
The Flashfig program can be accessed right in NetInstall as shown above.

Basic RouterOS Setup

There are a few features of RouterOS that you need to be aware of. These features are commonly used in many configurations, and before we dive into them, you will need to know where you can find them and how to configure them.

Configuring IP Addresses

We are dealing with a Router right? Well then we will need some IP addresses to go on our Router. Now, we are not going to get into talking about sub netting and TCP/IP right here in this book, but we are going to at least get you on the Internet with some basic IP information.



New Address

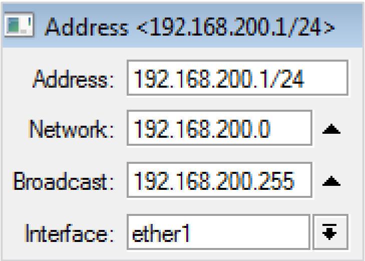
Address: 192.168.200.1/24

Network:

Broadcast:

Interface: ether1

We are going to start by configuring an IP address. To access your list of IP addresses, you will click on IP → Addresses (imagine that!). You will add IP Addresses to RouterOS just like any other object list in WinBox. Click the plus sign and you will be on your way. To configure your IPs, you will need three pieces of information. One is the IP Address itself. The second is the subnet mask, or in our case, the CIDR, and the third is the interface you wish to place that IP on.



Address <192.168.200.1/24>

Address: 192.168.200.1/24

Network: 192.168.200.0

Broadcast: 192.168.200.255

Interface: ether1

I get plenty of comments like "what the heck the /24 is on the end of this IP"? There are two ways of displaying an IP and subnet mask. Most people are accustom to typing in the IP address, in this case *192.168.200.1*, and then typing a subnet mask that looks like this, *255.255.255.0.* The above IP and mask, *192.168.200.1/24,* is the exact same thing as putting in all of those 255's. Using the Triple-255-dot-zero is called the Dotted-Decimal Notation method. Another method, the one that RouterOS uses, is the CIDR, or Classless Inter-Domain Routing method. This method uses the

"/24" to notate how many of the subnet mask bits are "on". If you convert 255.255.255, to decimal, and count the ones, you will get 24, hence, that's where the "/24" comes from. Both methods are perfectly valid, but RouterOS prefers the CIDR method.

You also may have noticed that I have not entered a network or broadcast address. One nice thing about RouterOS is that, based on your IP address and subnet mask, it will calculate your network and broadcast addresses for you. Once you hit apply, it will fill in those fields for you. I do recommend that you allow RouterOS to do this for you as it will prevent human-error issues normally.

An important note to remember is that when you ***change*** your IP address and CIDR notation on an existing IP Address entry, unless you use the up arrow next to each field to clear the broadcast and network addresses, WinBox will not automatically recalculate your network and broadcast addresses.

IP Subnetting

I wanted to do a quick subnetting review. In this book you will see that I refer to private IPs and public IPs. If you know what they are, then you are doing well, but if you don't, here is what you need to know.

IP addresses basically start from 0.0.0.0 and go through 255.255.255.255. That's a lot of IP addresses, however, there are blocks of IP addresses that will never be used on the Internet as a whole. These blocks are used for different things, including private IP space. The Internet Assigned Numbers Authority (IANA) Reserved Private Network ranges are as follows:

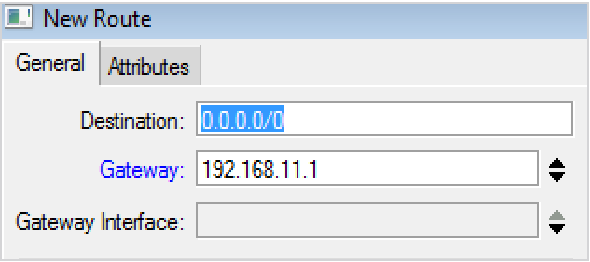
Prefix	Range	Total IPs
10.0.0.0/8	10.0.0.0 - 10.255.255.255.0	16.777 Million
172.16.0.0/12	172.16.0.0 - 172.31.255.255	1.048 Million
192.168.0.0/16	192.168.0.0 - 192.168.255.255	65,536

These blocks are set aside just for private network use in RFC 1918 and RFC 4193. The most common block is the /16 of 192.168.0.0. This entire block is very common in home routers. You can use these blocks on your internal network, or private network, without fear of them being used on the Internet.

Everything else is considered public IPs. These are IPs that are routed somewhere around the Internet. These are public, routable IP addresses, and these addresses typically allow for direct connections between point A and B. Private IP addresses are not publicly routable. You will need to have some form of Masquerading or other translation from your private IPs to your public IPs for you to get on the Internet. You also can not send data from a public IP address on the internet to a private IP address. Those private IP addresses are not listed in the global routing table.

Default Routes

A default route catches all traffic that the router does not have a route for, and tells the router that this is the gateway of last resort. To put it another way, unless another route more specific route is specified, the router will use this "default" gateway. RouterOS uses a default destination-address of *0.0.0.0/0* for its default gateway. To setup your default route, you will need to set this gateway. To access your Routing-Table, you will click on IP → Routes, again very straight forward. This will gives you access to the routing table, and allows you to click the plus sign and create a new route.



Destination: 0.0.0.0/0
Gateway: 192.168.11.1
Gateway Interface:

Above, you will see the destination of all zeros, or 0.0.0.0/0. This means all networks with any subnet mask, what we call a default route. You will need to enter the default gateway address for your network in the Gateway field.

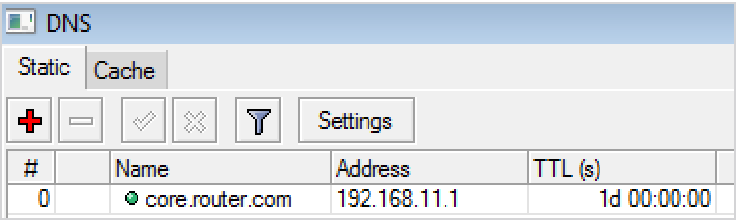
DNS Caching / Service

Once you get your IP addresses on your router, then you will need to have some form of DNS. Depending on your provider, they may have given you the DNS server's IP addresses; in which case you can enter that right into your DHCP-Server or your client computers. However, RouterOS does have the ability to do DNS caching. This allows everyone that uses your MikroTik

RouterOS router as their DNS server, to cache and to provide faster DNS lookups compared to going out over the Internet for these lookups.

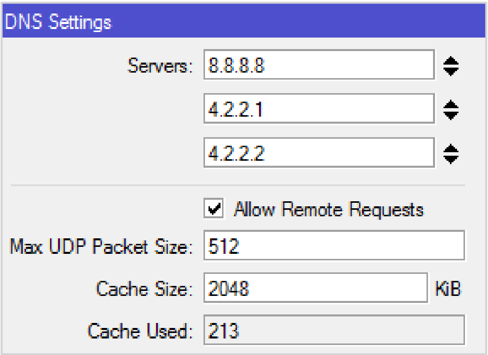
There has been some debate on if this method of caching is actually faster than just using a regular DNS server. The results that I have found are that as your DNS lookup queries hits your RouterOS, it and it does have the information that you need in cache; your DNS lookups take only are a few milliseconds verses 30-50 milliseconds just for the round trip time up to the next public DNS server.

You will access your DNS system, by clicking on IP → DNS.



#		Name	Address	TTL (s)
0		● core.router.com	192.168.11.1	1d 00:00:00

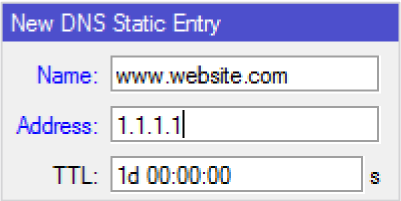
Once you get into your DNS system, you will click the Settings button to setup your upstream DNS servers. Versions prior to v4.6 had two DNS servers; these were listed as primary and secondary DNS Servers. After v4.6 you have the option of just a single "servers" field, where you can add many DNS servers as you like.



Servers: 8.8.8.8
4.2.2.1
4.2.2.2
☑ Allow Remote Requests
Max UDP Packet Size: 512
Cache Size: 2048 KiB
Cache Used: 213

If you use DHCP-Client (refer to the DHCP-Client section for more information), and the use-peer-dns option is selected, then the DNS servers under your DNS settings options will be replaced by any learned DNS servers from the DHCP-Client. You will also need to check the *Allow Remote Request* box as well. This allows your RouterOS to respond to DNS requests on its interfaces. If this option is not checked, then the RouterOS system will only use the DNS servers for internal lookups, and not respond to remote DNS requests.

Static DNS Entries



New DNS Static Entry

Name: www.website.com

Address: 1.1.1.1

TTL: 1d 00:00:00 s

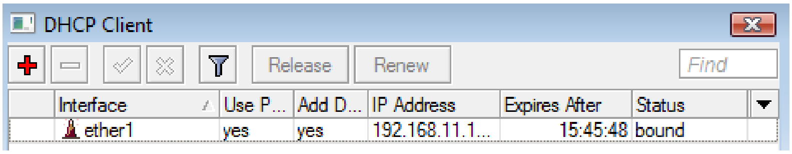
RouterOS supports the ability to answer DNS queries with a built in DNS server. This server, as outlined above, is not a full blown DNS server. It will however allow you to specify some records for resolution. These static entries take priority over other “learned” DNS entries from other DNS severs. For example, if you use website.com, and create a static entry like the one pictured here, if someone goes to www.website.com, and they are using your RouterOS system for their DNS server, they will receive 1.1.1.1 regardless what is on the public internet.

You can also use asterisks (*), and common expressions inside your static DNS entries. So instead, of using www.website.com, you can use *.website.com. This would match all of the following - www.website.com, ftp.website.com, or just website.com.

DHCP-Client

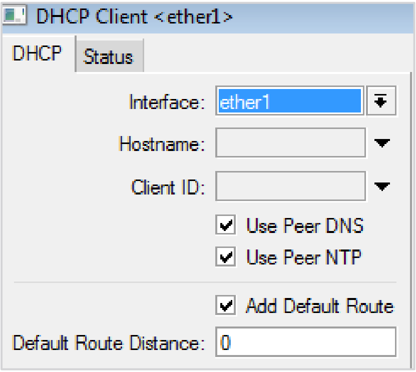
Sometimes, your Internet provider will allow you to obtain all of your IP settings automatically via DHCP or Dynamic Host Configuration Protocol. RouterOS has both a DHCP-Server and Client built in and will allow you to easily get the configuration that is necessary from your network or provider with ease. DHCP-Client will obtain not only your IP address, but your subnet mask, your DNS settings, NTP Server, and your default route. This makes it very easy to configure hosts quickly on a network. Most businesses will use DHCP to issue IPs out to clients that don't need to have a static IP address.

To access the DHCP-Client system, you will need to click on IP → DHCP-Client.



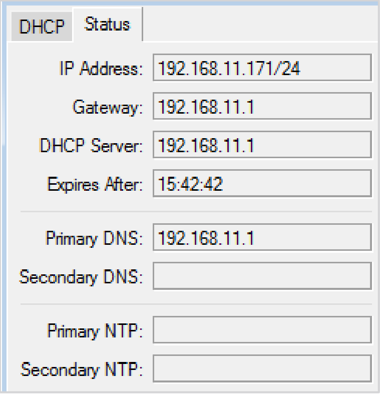
Interface	Use P...	Add D...	IP Address	Expires After	Status
ether1	yes	yes	192.168.11.1...	15:45:48	bound

Above you will see the DHCP client, running on Ethernet 1. On the right are the options that you have to select when you add a DHCP-Client. The main item that you will need to select is the interface you wish to run your DHCP-Client on. The other options, such as Hostname and Client ID are typically not used for our purposes. However, we do want to make sure we enable the Peer DNS, Peer NTP and Default Route. We are also not going to make any changes to the Route Distance here as well.



Interface: ether1
Hostname:
Client ID:
☑ Use Peer DNS
☑ Use Peer NTP
☑ Add Default Route
Default Route Distance: 0

Note that on the top menu bar of our RouterOS item list, we also have two extra buttons. One is a release and one is for renewing IPs. You will select the DHCP-Client under your item list that you wish to use, and then you can release or renew an IP address as you wish by using these buttons.



IP Address: 192.168.11.171/24
Gateway: 192.168.11.1
DHCP Server: 192.168.11.1
Expires After: 15:42:42
Primary DNS: 192.168.11.1
Secondary DNS:
Primary NTP:
Secondary NTP:

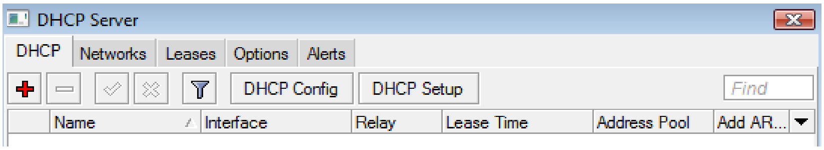
The default route distance is used to select what the distance cost is for the default route that is learned by this DHCP Client. This is useful if you need several DHCP-Clients, and wish to prefer a specific default route from one client. Refer to the static routing section and distances for more information about the distance feature.

Above, there is an image of the DHCP-Client status. This shows the IP address, gateways, DHCP-Server address, and DNS and NTP information that we obtained, and how long it is valid for.

DHCP-Server

Just like the above, RouterOS also has the ability to become a DHCP-Server, and handing out IP configurations for client usage. You can have multiple DHCP-Servers on different interfaces handing out different IP scopes for you, as well as have DHCP-Clients running on other interfaces. You can only have one DHCP system on one interface. With your DHCP-Server, you can give all of the necessary information to your clients without having to manually configure each one.

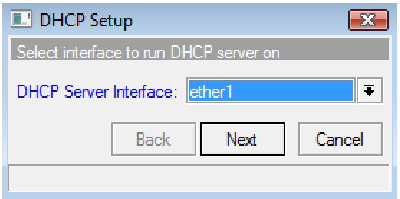
One important note is that you cannot run a DHCP-Server on an interface that is part of a bridge group. You can add a DHCP-Server to a bridge interface, but not to the interface that is part of the bridge group.



Name	Interface	Relay	Lease Time	Address Pool	Add AR...

To access the DHCP-Server menu, you will click IP → DHCP-Server. DHCP-Servers are not complicated to setup, but there are a number of functions and pieces of information that must be obtained and setup for them to work. Due to this fact, RouterOS has created a wonderful DHCP Setup button that we can use to quickly setup a DHCP-Server based on an interface. I do recommend that you go ahead and setup your IP address on the interface that you are going to put the DHCP-Server on. This will add that range and subnet to the DHCP-Server setup wizard.

DHCP-Server Wizard

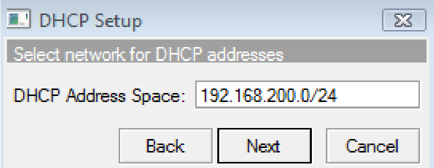


DHCP Setup

Select interface to run DHCP server on

DHCP Server Interface: ether1

Let's run through the wizard so that you understand all of the information and questions that RouterOS asks during the setup wizard. Start by clicking the DHCP Setup button. It will ask what interface you wish to run the DHCP-Server on. Remember, DHCP-Servers run on an interface. You typically will only have one DHCP-Server per network as well. Select the interface and select next.

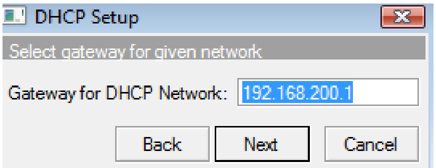


DHCP Setup

Select network for DHCP addresses

DHCP Address Space: 192.168.200.0/24

You will be asked you for your DHCP Address space. This typically will be filled in for you if you have your IP already on the interface that you selected. This is basically the subnet that the DHCP-Server will run on.

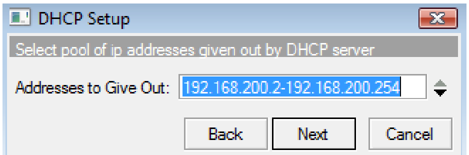


DHCP Setup

Select gateway for given network

Gateway for DHCP Network: 192.168.200.1

When you click Next, you will be asked to select your Gateway for the DHCP network. This typically is going to be your Router, if it is the default gateway. This is the IP address that will be given to the DHCP Clients as their default gateway. Click Next to continue.



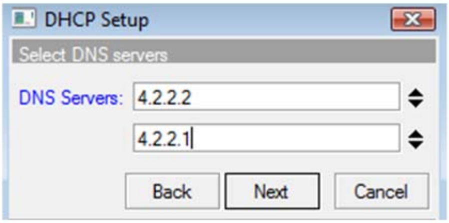
DHCP Setup

Select pool of ip addresses given out by DHCP server

Addresses to Give Out: 192.168.200.2-192.168.200.254

Now, you will specify the IP Addresses to give out. This is a pool of IPs that will be given to your clients as they request them. By default, RouterOS will specify all of the IPs in the subnet other than the IP of your router. In this case, our router is 192.168.200.1, so by default the router gives out 192.168.200.2 through 192.168.200.254.

If you are running a business network you may need to have some IPs that is statically assigned. I typically will use 2 through 50 for static items, such as printers, servers etc. You can set this up however you wish. Also, if I know I will not have more than 100 dynamic devices on the network at once, I will set the DHCP address range to be something like 100-200.


DHCP Setup

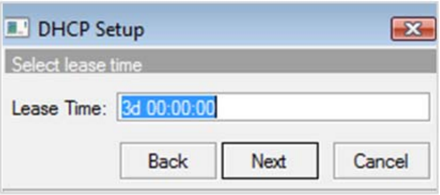
Select DNS servers

DNS Servers: 4.2.2.2

4.2.2.1

The next section is the DNS Server setup. As we said in the DHCP-Client section, we can hand out the DNS Servers that we wish our clients to use. Here we can enter the DNS Servers to hand out. This could be the local IP of our MikroTik, so we could enter in 192.168.200.1 as our primary DNS server and add a secondary upstream server if we wished.

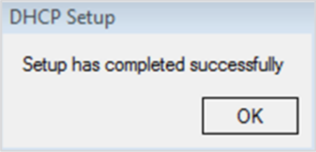
The final stage is to setup the lease time. This is the time that the client will keep that IP and information. Once this expires, the client must perform another DHCP request. They very well may get the same IP address back however, if there is a break in time after the lease expired is up and the computer does another DHCP request, then the original that IP may have gone back into the pool of available addresses and been handed out to another DHCP Client.


DHCP Setup

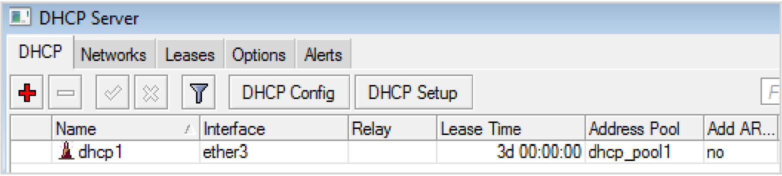
Select lease time

Lease Time: 3d 00:00:00

There are a number of thoughts to assigning the lease time. Typically DHCP traffic is minimal, so a shorter lease time often is sometimes preferred. If you are setting up a network that will have lots of transient users, or users that come and go often, then you may wish to reduce the lease time way down to something like 2 or 3 hours. This way you won't run out of IPs. If you have desktop computers that don't move around much, then you can have a longer lease time. I would always recommend on a lower lease time rather than a higher one, as the worst it can do is to require the DHCP client to renew their lease more often. This does not generate much traffic and doesn't affect clients.

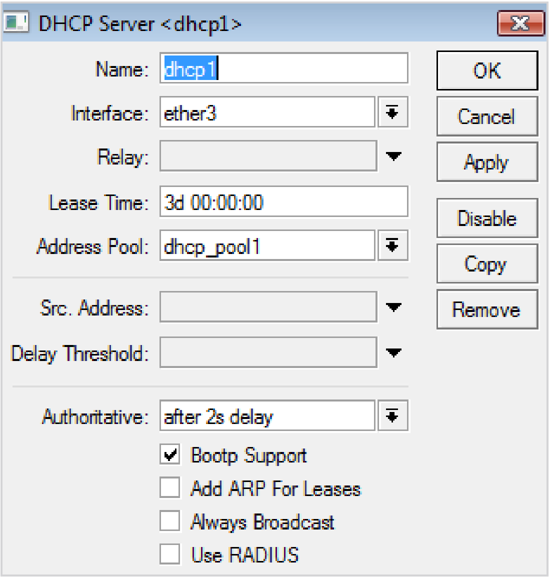

DHCP Setup

Setup has completed successfully

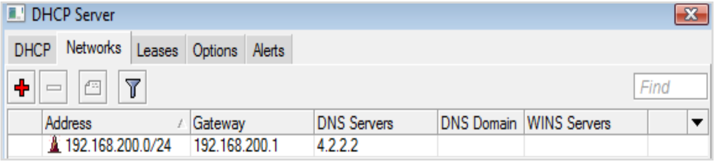


Name	Interface	Relay	Lease Time	Address Pool	Add AR...
dhcp1	ether3		3d 00:00:00	dhcp_pool1	no

Once this wizard is completed your DHCP-Server should be working. One reason it might show up red, is that you placed it on an interface that is part of a bridge group, or the interface is not running. Double-clicking on the DHCP-Server object will allow you to change the interface settings, as well as the lease time and the pool of available IP addresses it will use. You also have the options here to select of adding ARPs for the leases that you have, as well as the ability to use Radius.

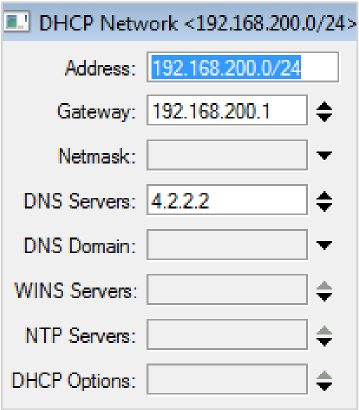


Name: dhcp1
Interface: ether3
Relay:
Lease Time: 3d 00:00:00
Address Pool: dhcp_pool1
Src. Address:
Delay Threshold:
Authoritative: after 2s delay
☑ Bootp Support
□ Add ARP For Leases
□ Always Broadcast
□ Use RADIUS



Address	Gateway	DNS Servers	DNS Domain	WINS Servers
192.168.200.0/24	192.168.200.1	4.2.2.2		

Under the Networks tab of your DHCP-Server, you will see all of the network settings. As you can see in the image below, you have options for your Gateway information, DNS Servers, and even other information such as DNS Domain, and WINS Servers, if you have them on this network.

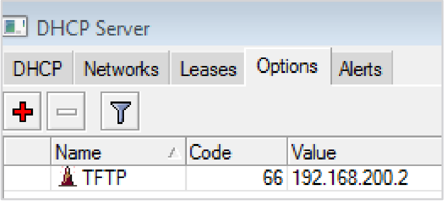


DHCP Network <192.168.200.0/24>

Address: 192.168.200.0/24
Gateway: 192.168.200.1
Netmask:
DNS Servers: 4.2.2.2
DNS Domain:
WINS Servers:
NTP Servers:
DHCP Options:

Double clicking on the DHCP Network object, will allows you to change these options for your DHCP networks. If you wish to specify NTP Servers you can do that as well right here inside your DHCP network.

Other DHCP Options, such as TFTP Servers, are setup here. The tab under DHCP-Server called Options, allows you to specify what Options you wish to use. You will first create these Options along with their code and value, and then under your DHCP Network settings, you will be able to specify that this Network has this DHCP Option. In this example, the TFTP name would show up in the DHCP Options section. You can specify several DHCP options as needed.



Name	Code	Value
TFTP	66	192.168.200.2

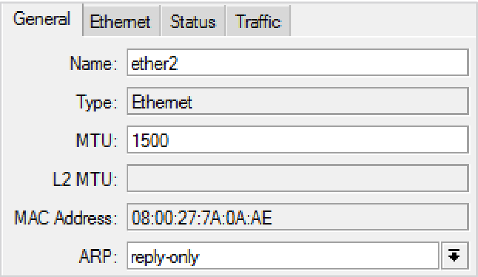
The DHCP Setup Wizard does quite a few things; real quick, let’s review them here.

- What does the DHCP-Server Setup Create?
- DHCP-Server Interface
  - What Interface to run on
  - Lease time to use
- IP Pool to use
- DHCP Network Settings
  - What Gateway to hand out,
  - DNS Server
- Other DHCP network options
- IP Address Pool
- Creation of a pool of IP addresses to hand out.

Using “Add ARP for Leases”

A feature in RouterOS called “Add ARP for Leases” allows your DHCP-Server to hand out IP addresses to MAC addresses on your network, and it then adds ARP entries for these MAC addresses in your ARP table. Normally, this is not needed; however there is a reason to do this in specific instances.

So what would be one of these instances? This feature can be used as a security measure in some cases. What you would do is Setup your interface, the same one your DHCP-Server is running on, to only reply to ARP requests. This option is called “reply-only” under your interface ARP settings. This allows devices that are looking for the MAC of an IP address that is on your router, to get a reply; however, RouterOS never enters their MAC/IP address combination into its ARP Table. This prevents your RouterOS box from talking to devices on your network, because there is no ARP entry for them.



Name: ether2
Type: Ethernet
MTU: 1500
L2 MTU:
MAC Address: 08:00:27:7A:0A:AE
ARP: reply-only

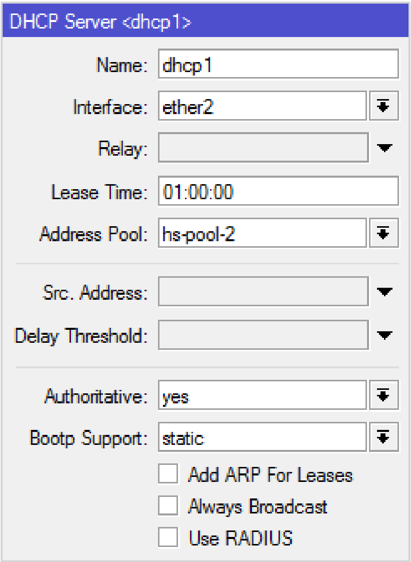
So how does this work with our DHCP-Server? Normally, the ARP mode is enabled on your interfaces. This means that any devices can put a IP in the address range that the router has, and the router will communicate with it. It will cause when the device asks who has an IP that is on the router, the router replies giving its MAC address. The Router has to do the same, it asks, who has some IP, and then the client will respond via its MAC. With the interface in "enabled" ARP mode, this would be the normal operations. However, with the interface in "reply-only" mode, the device can get the MAC of the router, but the router does not attempt to find MACs for other IP addresses. The only way the router knows what MAC goes to what IP is by the DHCP-Server assigning a MAC a specific IP and then entering that as a ARP in the routers ARP table.

So by using a combination of interface ARP Reply-Only and your DHCP-Servers “Add ARP for Leases” options, your router will only talk to devices that have received IP addresses from your DHCP-Server. Other IP addresses, even if it is in the proper subnet, would not get two way communication from your router.

Prevention of Rouge DHCP-Servers (er, rather, help prevent)

Wireless ISPs that run bridged or Layer 2 networks, have issues with what we refer to as Rouge DHCP-Servers. Simply put, these are DHCP-Servers that are operating in the same broadcast domain as valid DHCP-Servers. You should only have one DHCP-Server per broadcast domain, but when you add customers whose networking experience is somewhat limited, you can end up by having customers plug their home routers in backwards, or uses switches in-line with their home routers. These actions can cause issues on your network, regardless if the customer knows they are causing it.

Some hardware manufacturers have ways of preventing this. Typically this is done by not allowing DHCP-Discovery packets to pass through the wireless interface to the Ethernet interface. RouterOS can prevent this the same way, by ensuring that the DHCP-Discovery packet cannot go toward the client address. This is done simply would by creating by a filter rule preventing that type of traffic through your bridge.



DHCP Server <dhcp1>

Name: dhcp1
Interface: ether2
Relay:
Lease Time: 01:00:00
Address Pool: hs-pool-2

Src. Address:
Delay Threshold:

Authoritative: yes
Bootp Support: static
□ Add ARP For Leases
□ Always Broadcast
□ Use RADIUS

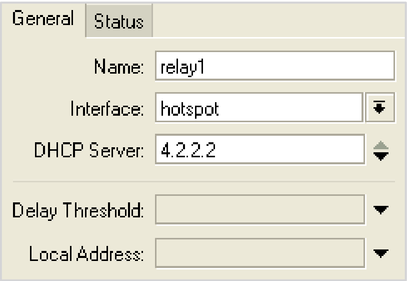
RouterOS also has another standardized feature, called Authoritative. This basically sets your server as the "Authoritative" server on the broadcast domain. Another way of looking at this is; is this DHCP-Server the only one DHCP-Server for the network? If it is, then you should setup your DHCP-Server to be Authoritative.

By default, the Authoritative setting is after-2sec-delay. This causes your DHCP-Server to wait two seconds before doing anything with the client's request. Typically we don't want to wait. We, as the administrators of the network, know that this is our only DHCP-Server and it should be the one to answer the DHCP requests.

With this said, is this a cure-all for rouge DHCP-Server operations on any network. Nope. Will it help out? Of course it will. Preventing your clients from receiving other clients DHCP Requests would also help as well. In my opinion, your clients should never have access to your Layer 2 network. They should always be segmented by a router preventing the issue entirely. This though this may not be practical for every network, but in most cases it should be a goal for many administrators.

DHCP-Relay

DHCP-Relay is a proxy that is able to receive a DHCP request and send it to a real DHCP-Server. To setup DHCP-Relay, you will click IP → DHCP-Relay and create a relay. The interface is the interface that the DHCP-Relay can run on. The DHCP-Server is the IP address that we pass the DHCP-Server request to.



Name: relay1
Interface: hotspot
DHCP Server: 4.2.2.2
Delay Threshold:
Local Address:

On your DHCP-Server side, you create a DHCP-Server with the proper IP Pool, but in the DHCP-Server options, you will place the IP address of your DHCP-Relay in the relay field. This will make that server respond to only relay requests only.

The IP address of the DHCP-Relay is the IP address that your DHCP-Relay sends the requests from. This typically will be the forward-most-facing interface from your DHCP-Relay to your DHCP-Server. This can cause us issues if the DHCP-Relay router undergoes a routing change that causes it to use another interface. In this case, the IP address that the relay uses to send out the relay requests would change to the new interface. To fix this issue, we use the local address to specify what IP address the DHCP-Relay uses to send out the requests to the DHCP-Server. This should match the IP address in the DHCP-Server under the DHCP-Relay IP address field.

Real World Usage of DHCP-Relay

The DHCP-Relay feature allows you to use a single DHCP central server where all leases are handed out. In many networks, we end up having a central (core) DHCP-Server, and all of the other sites would have a central DHCP-Relay system set up to relay those requests to the centralized server. This gives you the ability to monitor all of your DHCP-Leases in a centralized location, and the ability to make the entries static or to remove them, all from the one central location. Scripting can be used to automate tasks, if needed from the single central location instead, vs. all of at all your DHCP-Relay locations.

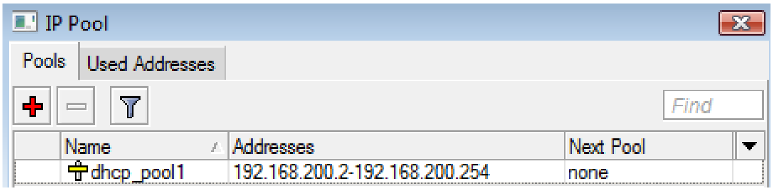
Keep in mind that you will have to have IP networks addresses properly configured defined for all of your DHCP-Servers, as well as for your DHCP-

Relays addresses defined for this feature those systems to work properly. If you have a fully routed network as well as redundancies, you will need to implement a loop back system to assign your SRC-Addresses in your DHCP-Relays too. See the loopback interface section for more information on how this works.

IP Pools

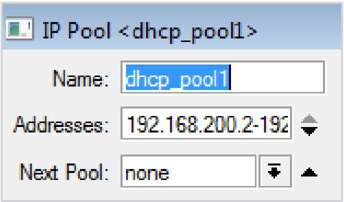
The wizard also creates IP Pools; these are pools of IP addresses that your RouterOS System can use to assign IP addresses out of. These IP Pools are automatically created for you, but you should know where it they are created.

To access your IP Pools, click IP → Pools under WinBox.



Name	Addresses	Next Pool
dhcp_pool1	192.168.200.2-192.168.200.254	none

You will notice that there is already a DHCP Pool created. Double-clicking the pool will access the individual pool information. Here you can change the address range that it gives out. There is also an option for the Next Pool. This option specifies what pool to go to once this pool is out of IP addresses.



Name: dhcp_pool1

Addresses: 192.168.200.2-192

Next Pool: none

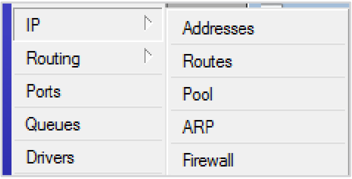
Masquerading - NAT

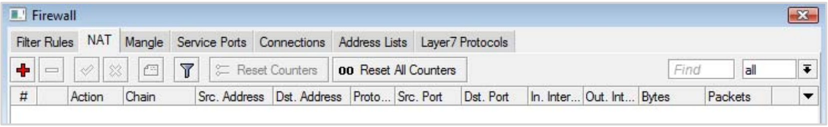
The NAT or Network Address Translation system inside RouterOS is very advanced. We will focus on is just one function called Masquerading in this section. This feature allows is a many-to-one translation of IP addresses. An example would be; you have 100 computers on a private network. You are assigned a single IP address from your Internet provider, and you need all 100 clients to get to the Internet. By using Masquerading, you will be translating these 100 client addresses all into one IP address. Lots of consumer routers will call this function NAT, but NAT actually does quite a bit more than just masquerading, and may not require masquerading to function. We will refer to NAT and Masquerading them as separate items.

We don't need to go into the interworking knowledge of how Masquerading is accomplished, however, it is important to know that, from the Internet's perspective, looking toward at your 100 clients, all that will be seen is just that single public IP address that we were assigned. All of the traffic will appear to be coming from that single IP address even though we have 100 clients behind it. This is important to understand as the outside world does not have any direct access to any of the individual devices behind that Masquerade. It hides those private addresses, and because of that, no other IP addresses on the Internet can connect directly to your private addresses.

Configuration of basic Masquerading

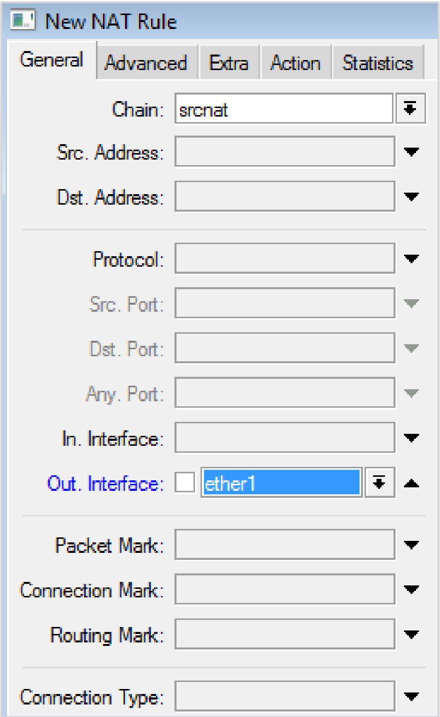
To start, you will need to access the NAT section of RouterOS. This is located under your IP Firewall system. Click IP → Firewall, and then under the Firewall options, you will need to click the NAT Tab. This is pictured to the right and below.





Here's how to need to create a basic Masquerade. We will assume our Internet connection is on Ethernet 1, and our private network is on Ethernet

2. Just like other sections of RouterOS, we will click the plus sign to create a new object. In this case though, we will call these objects rules. The reason for this is that we now have an order in which the rules are processed. In the above window, we have a # field to the far left. This is the rule number. **RULES ARE PROCESSED BY ORDER NUMBER**.

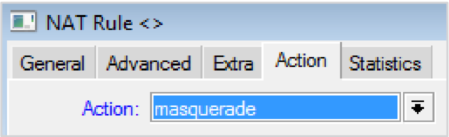


Chain: srcnat
Src. Address:
Dst. Address:
Protocol:
Src. Port:
Dst. Port:
Any. Port:
In. Interface:
Out. Interface: ☐ ether1
Packet Mark:
Connection Mark:
Routing Mark:
Connection Type:

These objects are rules. What is the goal of rules? It is simple; to match data. You will be building rules that will match data in some way. Since our Internet connection is on Ethernet 1, we are going to setup this rule to match on our Out Interface using Ethernet 1. We use a chain of src-nat. We will discuss NATing and the chains later in the NAT section of the book further.

Once we have said that we are looking for traffic that is going out Ethernet 1, we now need an action.

We click on the Action Tab and then select the action of Masquerade. This says, once the rule is matched, and then performs the action of Masquerading.



Action: masquerade

Home Router

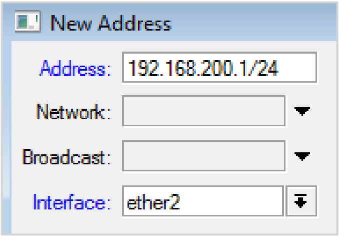
One common method of setting up RouterOS; as well as a great introduction to some of the common features of RouterOS, is setting it up as a generic home router. There are a few functions we will need to perform here:

- **Our Goal**
    - To allow several computers on a private home network, to gain access to the Internet through a single Internet connection.
- **What We Know**
    - DHCP Internet connection
    - Several computers on our home network
    - Internet Connection is on Ethernet 1
    - Private computers will be on Ethernet 2
- **Features We Will need to Use**
    - DHCP-Client
        - To get the IP information from our Internet provider.
    - DHCP-Server
        - To assign private addresses to the computers inside our network.
    - Masquerading
        - To translate the many private IPs on our private network computers inside our network to the single public IP address that we will receive from our Internet provider.
- **Here are the Steps We Will Take:**
    - Login to RouterOS
    - Set your Private IP on Ethernet 2.
    - Setup DHCP-Client to run on Ethernet 1
    - This will obtain your Default Route and DNS information
    - Setup DHCP-Server on Ethernet 2.
        - DNS information will be filled in since we obtained it from our DHCP-Client.
    - Default Gateway will be our router
    - Setup Masquerading on the Ethernet 1 Interface
  - Create rule, out Ethernet 1, action Masquerade

Home Router Walkthrough

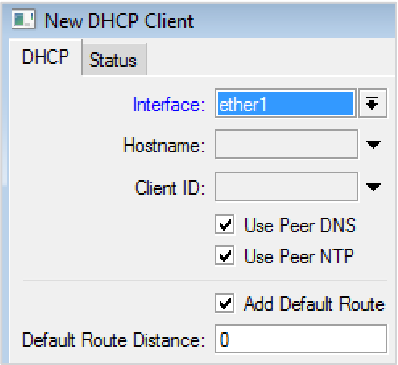
Step 1: Login to your router

Step 2: Set your Private IP on Ethernet 2. We will use 192.168.200.1/24 for your private range. Click IP → Addresses → Plus Sign. Add IP address to Ethernet 2.



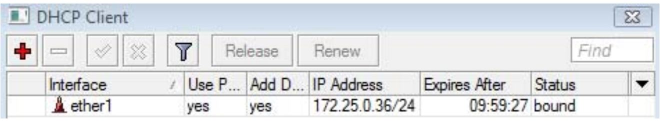
Address: 192.168.200.1/24
Network:
Broadcast:
Interface: ether2

Step 3: Setup DHCP-Client on Ethernet 1. Click IP → DHCP-Client → Plus Sign. Select interface Ethernet 1. We will use all of the peer information as well as the default route from our provider, so leave these checked.



Interface: ether1
Hostname:
Client ID:
☑ Use Peer DNS
☑ Use Peer NTP
☑ Add Default Route
Default Route Distance: 0

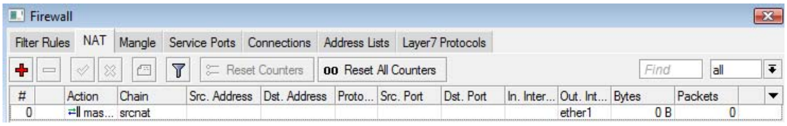
Verify that you obtained an IP address



Interface	Use P...	Add D...	IP Address	Expires After	Status
ether1	yes	yes	172.25.0.36/24	09:59:27	bound

Step 4: Setup DHCP-Server on Ethernet 2. Click IP → DHCP-Server → DHCP Setup Button. Follow the DHCP Setup Wizard. Select Ethernet 2 as the interface, the address space will be filled in as we have already placed the IP on the interface. The gateway will be the IP of your RouterOS system that you placed on Ethernet 2. Leave the defaults values for the addresses to give out. The DNS servers that show up in the DNS section will be the ones that you obtained from your Internet provider. The final step is to leave the lease time at 3 days and finish out the configuration. Since we have followed this process in the DHCP section above, I will not outline it here.

Step 5: Setup Masquerading on data going out Ethernet 1. Select IP → Firewall → NAT tab → Click the Plus sign. The chain will be src-nat. The out interface will need to be set to Ethernet 1; and then click on the action tab, drop down the actions and select Masquerading.



#	Action	Chain	Src. Address	Dst. Address	Proto...	Src. Port	Dst. Port	In. Inter...	Out. Int...	Bytes	Packets
0	⇄ mas...	srcnat							ether1	0 B	0

Once this is done, you can now plug a computer or device into Ethernet 2, obtain an IP address, and then browse the Internet.

Common Wireless Configurations

So you got your first RouterOS system running, either x86 or RouterBOARD. Let's go through the basic information that you will need to setup wireless configurations on your RouterOS. Some of the material in this section, we will talk address only briefly, but in the RouterOS Features section, we will go in to much more depth. Here are some basic configurations in this section.

Bridged Access Point Configuration

To create a bridged access point, there are only a few things that you will need to configure. First, open your router and connect to it. Create a bridge group, and then add both your Ethernet port and your wireless interface to the bridge group that you just created. Once this is done, you will need to setup a basic IP address for management as well as a default route. The IP address should be on the bridge group interface.

Now that you have your bridge and management IP setup, simply configure your wireless interface. For Point-to-Multipoint, select AP-Bridge, select the best channel for your usage after you, as well as setup the band that you wish to operate in. Then create a SSID that you wish to use. The last thing you should do is setup a security profile inside your wireless interface if you wish to have security on your wireless network!

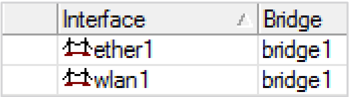
Some people have asked about the NAT and DHCP settings. Simply put, if you are allowing your access point to be a simple bridge, I would configure those features on your router below your access point instead of vs. on your access point. There is no need to have more services on the access point. If you wish to have bridged CPEs you will need to go ahead and setup your WDS settings. I would recommend dynamic WDS mode along with the default bridge as your bridge group you created originally. For more information on WDS, please refer to the WDS section.

CPE – Client Premise Equipment Configuration

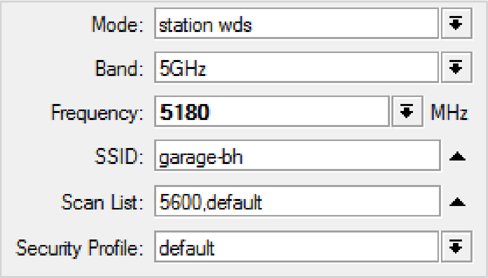
CPEs are mainly used at a subscriber's home for internet access by wireless ISPs (WISPs). These configurations may not be the best for your business however the bridged client and access point configuration be used is to create a Layer 2 back haul link between buildings or towers. For WISPs, I will always recommend the routing/NAT configuration for a CPE instead of the client bridges configuration. The reason is that it keeps the clients from connecting gear on your network, remember we don't want to allow your clients direct Layer 2 access to your network; you will be providing them a private subnet that is just for them. If they plug in something incorrectly, create a bridging loop, or if they have a virus that creates a broadcast storm, your network should be protected due to the router being in-line with the client before they get to your wireless network.

Bridged Client

To create a bridge client, the proper way is to use WDS. You will have to configure your access point for WDS. This is done simply by adding the WDS mode and default bridge to your wireless access point. See the Bridged Access Point Configuration section. Once this is done, then you can configure your CPE; otherwise it will not work. Note; refer to the WDS section for more detailed information on WDS configuration and operation.

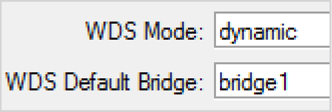


Interface	Bridge
ether1	bridge1
wlan1	bridge1



Mode: station wds
Band: 5GHz
Frequency: 5180 MHz
SSID: garage-bh
Scan List: 5600,default
Security Profile: default

First, create a bridge group on your RouterOS System. Add both the Ethernet and the wireless interface to the bridge group. Next, configure your wireless interface mode for a mode of station-wds. This mode will form a station relationship along with WDS to your access point. Configure the proper SSID or scan for the proper SSID. You may have to configure your security profile to be the same as your access point security profile in order for the unit to register to the AP. You will also need to configure the WDS settings; I would recommend dynamic WDS. Now, along with adding the bridge group you first created as the default WDS bridge group.



WDS Mode: dynamic
WDS Default Bridge: bridge1

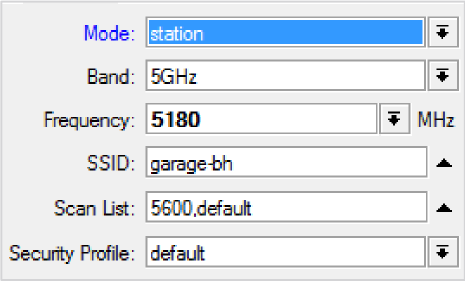
Once you complete these configurations this and the CPE associates to the access point, you should see a WDS interface that displays the MAC address of your CPE on your access point. Also, in the CPE bridge group you should see that a WDS interface has been dynamically created on your CPE as well. This will bridge the Ethernet interface and the wireless interface by providing a true bridge

How to Use Pseudobridge Mode

To create a CPE that uses Pseudobridge mode, follow the instructions in the Bridged Client section above. However, you do not have to have your access point and CPE with WDS enabled. Simply set your mode to station Pseudobridge or station Pseudobridge-clone mode. Once you do this, and you have added your wireless interface and Ethernet port into a bridge group, you are done! Remember though, this is not per the 802.11x RFC specification.

Routed / NAT CPE

The best way to configure a CPE is to have either a Routed or a NAT mode. These really are two different modes. Routing allows you to configure a publicly or privately routable IP address that can then be routed through your network. This typically is the best option, however, for customers that do not need publicly or privately routed subnets, and this includes most residential and/or business customers, you can simply do NAT on their CPE and use a single IP address on the wireless interface. By doing this, you create a separate broadcast domain for your clients, and prohibit broadcasts and ARPs from going across your wireless network!



Mode: station
Band: 5GHz
Frequency: 5180 MHz
SSID: garage-bh
Scan List: 5600,default
Security Profile: default

To do configure a CPE in Router/NAT mode, you will need to configure your CPE's wireless interface in station mode, just like any other client device. First, place an appropriate IP address and default route on the wireless interface. This should allow the CPE RouterOS device to ping out to the rest of the network and maybe even to the Internet. Second, you will create a private subnet on the Ethernet interface. I commonly use 192.168.200.1/24 on the Ethernet interface as most home and businesses do not use this on their networks. Go through the DHCP-Server setup on your Ethernet interface; this will hand out IP addresses to clients connected via Ethernet. The last step is to create a Masquerade rule going out the wireless client interface. I typically will set the source address of 192.168.200.0/24 along with the out interface of the wireless interface name. The action will be Masquerade using a Source NAT rule. This will Masquerade all of the private

IPs, or (192.168.200.x IP addresses) out using the single public/private IP on your network. That's it! If you wish you can also setup the DNS Client to accept DNS requests, and cache them right on your CPE device as well!

For businesses that need public IPs on their equipment you would configure a subnet, depending on the number of IPs required by the customer, on the Ethernet interface of your CPE. You would also have to have the proper routing setup on your network, access points, etc. For a customer that just needs a few IPs, we would typically route a /29 to the IP address on the WLAN interface from our Access Points or tower routers. Then on the Ethernet port, we would place the first usable IP of that /29 as the routers IP. This functions just like the previous configuration, but we don't need any NAT functions or rules. The Router does what it does best, route.

RouterOS and IP

In this section, we will start discussing the features of RouterOS. We will go down the WinBox section lists discussing each feature, how it works, configuration options, and give examples for your future use! Inside RouterOS and WinBox, there are a number of places that are “duplicated”. For example, you can configure the wireless interface settings of an individual radio card inside the interface section, but you can also get the same configuration settings inside the wireless interface section as well.

Why Routing

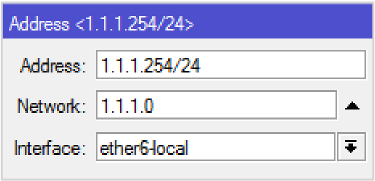
MikroTik RouterOS is of course, a router! Let’s get into the basic routing functions of RouterOS! First off, I like to spend a moment to answer a common question that I get all of the time. When I bridge, configuring IP addresses is easy and things work, so why should I route? Let me answer this question with two comments. First, bridging and IP addressing is very easy to manage, and run. However, it is not a matter of “if it will fail”, it’s just a matter of “when it will fail”. Second, my company motto is, “Friends don’t let Friends Bridge Networks!”

With that said, what are the technical reasons why you should route? The Internet is routed for a reason. Failures cause topology changes, and, just like the Internet, you should have routed traffic so when there is a network outage, your traffic will be able to fail over to other connections and links. In other words, by routing, you can take advantage redundant connectivity. Bridging will allow some redundancy, however, typically at the expense of turning OFF some links. Preventing a bridging loop requires disabling ports, therefore entire links are wasted. With routing you can have some traffic, go over a primary connection, and other traffic go over secondary connections, so you actually use more of the infrastructure that you have.

I also like to keep traffic that should be local, well, local. Every device on my entire network, from core routers, etc. doesn’t need to know about all 500 other devices on the network. ARP entries should be limited to just those needed to communicate. This also has another benefit, and that is to be able to better handle ARP and broadcast storms. This is done by limiting the size of your broadcast domain. Due to this, you also limit the effects of these types of issues to a much smaller part of your network. Backbone connections should not be affected by (and congested by) excessive ARP traffic or by broadcast storms. Read the VLAN section as well as it will describe further information about how VLANs do not keep traffic separate on physical networks.

IP Addresses

All IP addresses in RouterOS are setup and configured in the IP → Addresses menu or the IPv6 → Addresses. Unlike Cisco and other manufacturers, that place the configuration of the IP address on the interface itself, we configure the IP address on the Address menu.


Address <1.1.1.254/24>

Address: 1.1.1.254/24

Network: 1.1.1.0

Interface: ether6-local

Note that in v4 and many of the beta and RC version of v5, you will have both network and broadcast addresses shown in here. In v5rc9+ you should just see the network section instead of vs the network and subnet. The network address is calculated by the CIDR subnet block. RouterOS no longer uses not use Dotted Decimial notation, only CIDR.

Interface ARP – Address Resolution Protocol Settings

ARP or Address Resolution Protocol basically changes MAC addresses to IP addresses. A PC or device will say, via MAC broadcast, 'I need to communicate with 192.168.1.1'. There will be some other device that replies via MAC, or Layer 2 communications. This other device will say, "I am responsible for 192.168.1.1 and here is my MAC. This communication is done at the second layer of the OSI Model. This IP address-to-MAC address translation is retained in the ARP list. See the following section for more information about the ARP list.

ARP: enabled

Depending on the interface that you have, you typically will have only a few ARP options. The main options, that you have will be enabled, the default setting, disabled, proxy-arp and reply-only.

99.99% of the time, enabled is perfectly fine. This is the default option, which will reply to ARP requests. When a device is looking for an IP that your RouterOS has on its interface, this router will reply to that device saying it is responsible for that IP. Also, if it does not know the MAC of an IP, it will send an ARP request out to find the MAC for that IP.

So what are the uses for the other ARP modes? Some administrators will use the ARP disabled mode as a form of security. With ARP in the disabled mode, your router will not send out ARP requests, or reply to ARP broadcasts. Because of this, you will have to add ARP entries in your ARP list manually. On the other device, you will also have to do this as well. This requires manual ARP entries on both devices, but works quite well to increase security. With that said, some devices don't have manual ARP tables therefore you can't make a manual ARP Entry.

The Reply-Only ARP mode, will allows for this. In this mode, your RouterOS will reply to ARP requests, but will not send out ARP requests. So your remote device will send out the ARP request, and your RouterOS will reply, but your RouterOS will not know what IP is on the remote device. You will have to enter a manual ARP entry so that your RouterOS knows what IP belongs to what MAC.

In a previous chapter, there is an example of using Reply-Only ARP settings along with your DHCP-Server. This would be a practical example of using the Reply-Only ARP setting. Refer to that section called "Using 'Add ARP for Leases" for more information about this feature.

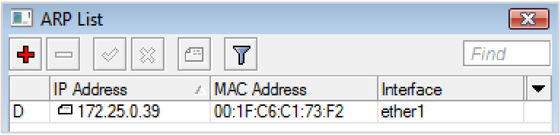
Proxy ARP

The last mode is Proxy-ARP. Smaller ISPs and WISPs will use this mode to deliver public IPs across their private network. This is not extremely common, but it can occasionally be useful work. The interface that you have setup for Proxy-ARP, takes any ARP requests it receives and forwards them on to other interfaces. If another device on one of the other interfaces has the specified IP, the ARP request is replied to. However, RouterOS translates this. The interface with Proxy-ARP will say that it has the desired IP, but it then translates the IP it to the other interface's MAC. So think of Proxy-ARP it as a Masquerading of MAC and IPs at Layer 2.

The most common example of a need for Proxy-ARP this is if you have several servers behind a RouterOS Router, you can place a public IP on a server. The gateway is on the public side of the router. Because the public interface has Proxy-ARP turned on, it forwards the ARP requests from the gateway, and sends them through to the server. This allows the server or servers to have a public IP even though the interfaces are not bridged.

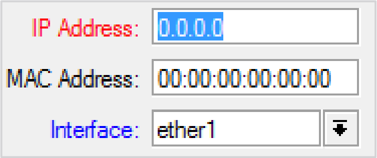
As a professional recommendation, I typically stay away from the above method. See the NATing section for information on how to NAT public IP addresses instead of vs. using Proxy-ARP. However, in a pinch, Proxy-ARP this will work. If possible, I would rather have routed subnets. When you have trouble on your network, look at Proxy-ARP first, and you will waste less time troubleshooting.

ARP List / Table



	IP Address	MAC Address	Interface
D	172.25.0.39	00:1F:C6:C1:73:F2	ether1

This table stores the entire MAC to IP translation list that a network device will need to communicate at Layer 3. You can access the ARP List by going to IP → ARP. Note that there is have a "D" next to the entry in this unit. That means this was a dynamically created entry. We get dynamic entries when your MikroTik sends out ARP requests and receives the replies from the remote devices.



IP Address: 0.0.0.0
MAC Address: 00:00:00:00:00:00
Interface: ether1

You can add manual ARP entries, by clicking the plus sign. You can also enable/disable just like other item lists, as well as add comments to each static entry. You also have the ability to use the find function to find MAC or IPs. To add devices, you will need the IP address and the associated MAC address, and the interface that the MAC address is found on of.

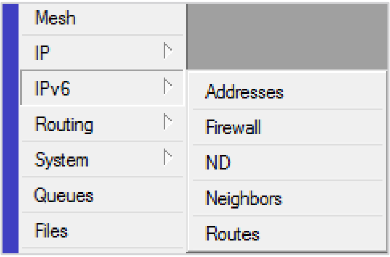
MikroTik also gives you tools such as Ping, MAC Ping, etc., to help you verify ensuring that your static entries are correct. Another feature that you should be aware of is the Make Static option. This option will let you select a dynamic ARP entry, and easily convert it to a static entry.

IPv6

As you may know, the internet as we know it is running out of IPv4 address space, and, by the time you read this, it may already have ran out of IPv4 addresses. In this book we are not covering, how, what, and when IPv6 came about, how to implement it, or best practices. However, it is important to note that RouterOS software is fully IPv6 capable. If you wish to know where and why IPv6, there are plenty of books that describe the best practices, options, header information, as well as the agreed upon specifications of the protocol.

What we do cover here is the IPv6 support in RouterOS, how it works, what works, and the specific IPv6 features that RouterOS contains. However, MikroTik is always adding new IPv6 features and support, so there is no point in listing it here. Most major usages for IPv6 are already implemented, but refer to www.MikroTik.com for more information on what actual feature set is supported and tested.

The IPv6 Package

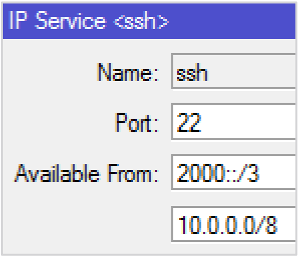


In order for your RouterOS system to support IPv6, you will have to install and/or enable the IPv6 package. In most system the package is already installed, but not enabled. By enabling the package and rebooting your router, it will enable the IPv6 feature set.

A number of IPv6 commands including the IPv6 Firewall, routes, and IP addresses are all under a new menu item called IPv6. I know; creative wasn't it? It also will show you your IPv6 neighbors as well.

Dual Stack

By installing the IPv6 package, you will have the ability to run 'dual stack'. In other words, your RouterOS system can route both IPv6 and IPv4 packets. This goes for all of the RouterOS services if you configure telnet or WinBox to answer, it will answer on any IP address, or IPv6 address on the router (unless Firewalled). You also add abilities to specify src-addresses and available forms throughout RouterOS with IPv6 addresses as well.



IP Service <ssh>

Name: ssh

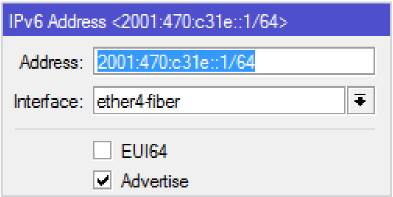
Port: 22

Available From: 2000::/3

10.0.0.0/8

IPv6 Addresses

As you can see, IPv6 addresses are quite a bit different than v4 addresses, but inside IPv6 → Addresses you will have the ability to setup the IPv6 address of your RouterOS system. There are two options here, one is the advertise feature. This enables the stateless address configuration for that subnet on that interface. This option is set by default if the subnet size is a /64. This advertises to hosts using ICMPv6 and allows auto configuration of IP addresses



IPv6 Address <2001:470:c31e::1/64>

Address: 2001:470:c31e::1/64

Interface: ether4-fiber

□ EUI64

☑ Advertise

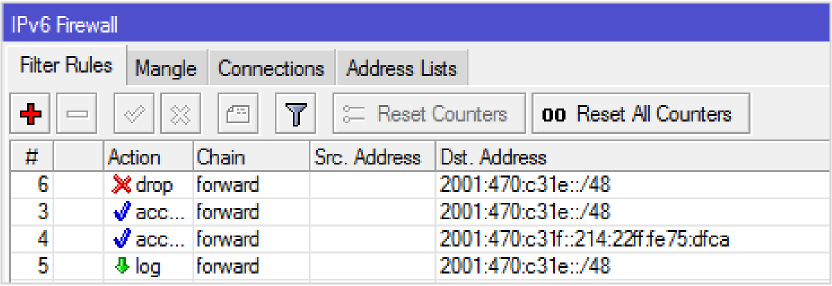
The second option is EUI64. With this option, the last 64 bits of the address will automatically be generated and updated by using the interface identifier. How and why this process works, is really outside the scope of this book, again, refer to a IPv6 resource or book for more information.

IPv6 Stateless Autoconfiguration

IPv6 offers and stateless Auto configuration feature. This allows configuration of the individual nodes connected to the LAN without a DHCP-Server. Devices plugged into the network will automatically receive and configure IPv6 address and gateway. This can only be done on /64 subnets and you must configure the IPv6 address to advertise.

IPv6 Firewall

The IPv6 Firewall is virtually the same Firewall as the standard IPv4 Firewall with the exception that we can use the IPv6 addresses! This Firewall is separate from the IPv4 Firewall, as it only processes IPv6 packets. The standard IP Firewall system, will only process on IPv4 packets. The IP stacks are separated inside RouterOS. Check out the dual stack section for more information about this.



IPv6 Firewall

#		Action	Chain	Src. Address	Dst. Address
6		✗ drop	forward		2001:470:c31e::/48
3		✓ acc...	forward		2001:470:c31e::/48
4		✓ acc...	forward		2001:470:c31f::214:22ff:fe75:dfca
5		log	forward		2001:470:c31e::/48

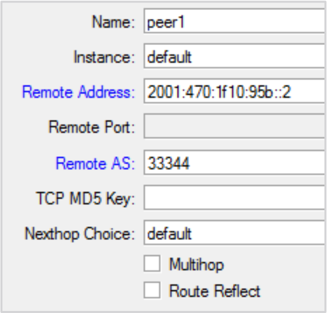
We also have the IPv6 Connections, Mangle and the IPv6 Address-list here under the IPv6 Firewall just like the standard IP Firewall. Counters, and actions are identical; the address-list simply runs IPv6 addresses but is otherwise is unchanged in configuration.

IPv6 Dynamic Protocols

The IPv6 package we also has added support for IPv6 in BGP, OSPF, and RIP.

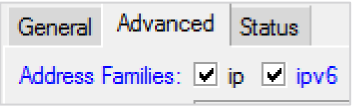
IPv6 BGP

In BGP, it was designed originally with multiple address families; therefore migration to IPv6 is very simple and straight-forward. In this case, you can BGP peer with either an IPv6 address or and an IPv4 address, and do your announcements running BGP over IPv4. You also can use an IPv6 remote address as well.



Name:	peer1
Instance:	default
Remote Address:	2001:470:1f10:95b::2
Remote Port:	
Remote AS:	33344
TCP MD5 Key:	
Nexthop Choice:	default
	□ Multihop
	□ Route Reflect

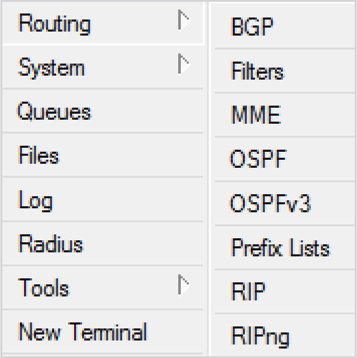
If you wish to make announcements for IPv6, you will need to have a BGP peer that supports the IP family of IPv6. Note this is shown to the right. As you add networks to advertise, you will have the ability to simply type either an IPv4 address or an IPv6 address.



Address Families: ☑ ip ☑ ipv6

IPv6 OSPF v3

OSPF, unlike BGP, was never really designed to be able to support address families, or other similar systems. OSPF required many changes to the protocol to support IPv6. It required so many changes that they ended up creating an entirely new version of OSPF. OSPFv3 is the new version of OSPF; OSPFv2 is the standard IPv4 OSPF and they are completely separated. In V3 uses the same systems as v2, (for example: LSAs, and flooding) but also adds the IPv6 addresses, as well as improving on the protocol as well. You will need to consult and OSPFv3 resource to get more information on the fixes that were accomplished in v3.



To configure OSPFv3 you will need to use the OSPFv3 options under routing. The standard OSPF section is for v2 and IPv4.

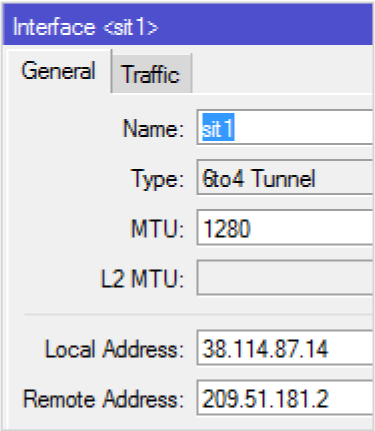
Once inside OSPFv3, you will configure OSPF using the same configuration interface that you are used to with v2. One of the major differences though, is OSPFv3 does not support the Networks tab. This was done away with in v3. OSPFv3 runs on interfaces rather than subnets or networks. So the networks tab is completed unneeded in V3. The

IPv6 RIP – RIPng

Just like OSPFv3, RIP for IPv6 created a new version of RIP. RIPng is the new version that supports IPv6 addresses. The configuration is the same as RIP other than you must configure the RIPng interface instead of adding networks, again just like OSPFv3, as RIPng is interface-based not IP subnet-based.

IPv6 to 4 Tunnels – 6to4

RouterOS supports 6to4 tunneling. This is a technology that was designed to aid in the transition from IPv4 to IPv6. What this does is simply encapsulate IPv6 packets inside IPv4 packets. This creates a tunnel, or link to be able to transmit IPv6 across network infrastructure that does not support IPv6. Note though, that this tunnel does not have any type of security features.


Interface <sit1>

Name: sit1
Type: 6to4 Tunnel
MTU: 1280
L2 MTU:
Local Address: 38.114.87.14
Remote Address: 209.51.181.2

To create a 6to4 tunnel, simply go to the interface list, and add a 6to4 Tunnel. You will need both the local and remote address. That's it. You would configure this on the other end as well, with the addresses swapped.

Remember that we are using 'Dual Stack' so the routing processes of IPv6 and IPv4 are separated. In other words, we have an IPv4 default route and an IPv6 default route. Once we have the tunnel established, then we would place an IPv6 address to the tunnel, and then create a route (possibly your default IPv6 route) to point at the tunnel. This will route your IPv6 data out the 6to4 tunnel.

Deploying IPv6 Within Your ISP

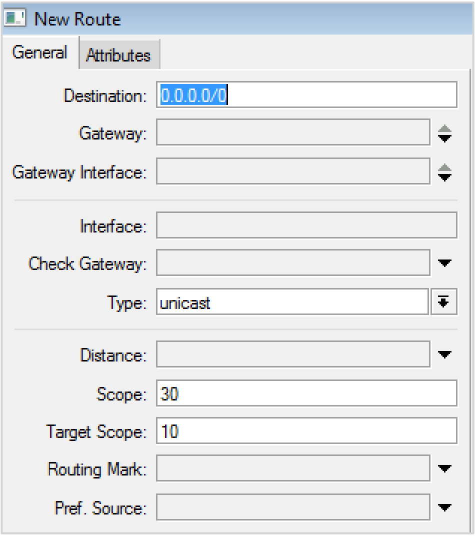
There are basically two major ways to allow your WISP or ISP to run with IPv6 and to provide IPv6 services to your customers. The first way is to run dual-stack, as explained in the previous dual-stack section; we have the ability to run both IPv4 and IPv6 at the same time. This means that you have to have both an IPv4 and IPv6 address on all of your interfaces. You maintain two separate routing tables, one for v4 and one for v6. Think of it as running two different versions of IP, because, you are! This method works fine, but adds additional work for you as an administrator. You must keeping up two sets of IPs on every device and every interfaces. This method is the best way to run both versions of IP.

The second way is to create a 6-to-4 Tunneling server at your network edge and let each of your clients create a tunnel back to your core. This allows you to run IPv6 over your existing IPv4 network. This works quite well if you have a few IPv6 clients that you wish to enable. This also allows you to maintain the IPv4 network, including your dynamic routing, and keep your IPv6 clients happy. For the short term, this will work for most ISPs that are slowly deploying IPv6 on their backbone. As you roll out your dual-stack network, you can easily convert from the 6-to-4 tunnels.

Routing and Routes

Adding Routes is a very simple process. Click the plus sign while you are viewing the routing table. The new route window will give you plenty of options. Remember, your default route will have a destination of 0.0.0.0/0. You can specify the gateway by typing the IP address in the Gateway box.

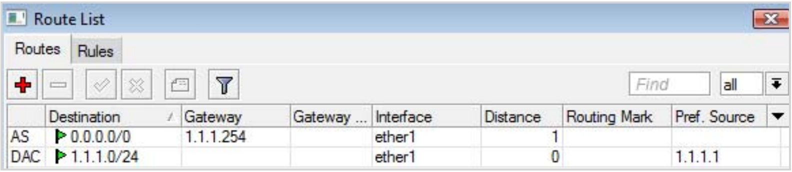
In v3 of RouterOS there is an option to specify a gateway interface. You can do this on tunnel connections, PPPoE connections, and on another interface that is setup with a /30 subnet. The reason for this, is that there are only two other IP addresses to use in the /30 subnet. Your router will have one, and the other will be used as the Gateway. Interface routing makes it very easy to route, just by using just an interface name instead of needing vs. having to know the IP address.



Destination: 0.0.0.0/0
Gateway:
Gateway Interface:
Interface:
Check Gateway:
Type: unicast
Distance:
Scope: 30
Target Scope: 10
Routing Mark:
Pref. Source:

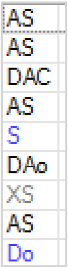
Static Routing

RouterOS offers a very simple interface for creating static routes. To access the Routing interface, simply click IP → Routes.



	Destination	Gateway	Gateway ...	Interface	Distance	Routing Mark	Pref. Source
AS	0.0.0.0/0	1.1.1.254		ether1	1		
DAC	1.1.1.0/24			ether1	0		1.1.1.1

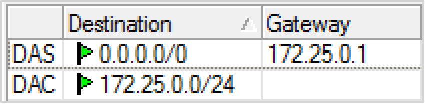
The IP Routes list will show all of your Route options you have. The first column is very important. This column shows the status of each of your routes.



AS
AS
DAC
AS
S
DAo
XS
AS
Do

S = Static Route
A = Active Route
C = Connected Route
o = OSPF Route
X = Disabled Route
r = RIP Route
b = BGP Route
d = Dynamic Route

There are also items that are blue in color. Blue items are routes that are valid, but are not active. This typically means there is a static route that is taking priority, or another route that has a lower cost. It also could mean that the gateway check has failed; therefore the route is inactive due to not being able to get to the gateway.



	Destination	Gateway
DAS	▶ 0.0.0.0/0	172.25.0.1
DAC	▶ 172.25.0.0/24	

There are two types of routes that I wish to cover in more detail. DAS routes are always interesting. How can it be a dynamic active static route? This statement contradicts itself. The reason for this is that this is a route that was received via the DHCP-Client system. When the DHCP-Client has the check box to receive the default route, a DAS route is created. It's static due to the fact that it was entered in via the DHCP-Client, but since the client entered it from a dynamic source, DHCP in this case, it is also dynamic. There are also DAC routes. These are dynamically active connected routes. This basically says this subnet is directly connected to the router. It is added dynamically due to adding an IP to the router, and as long as the interface is up and running, it will be active!

Checking Gateways

Check Gateway gives you the ability to verify that the gateway is available. It has two options, ping or ARP. If the router does not have an ARP for the gateway, the gateway will be considered to be unavailable. This makes the route turn Blue; it would normally be available and active, but in this case, without an ARP table entry, the gateway is considered not on-line or otherwise unavailable. In most cases you would use the ping option first, as this pings the gateway watching for it not to respond. However, if you have a router or gateway that does not respond to pings, you can use the ARP option.

The check gateway with ping has some interworking's that you should be aware of. When you specify check gateway on an IP address, the router will check the gateway by pinging it. It will ping, every 10 seconds; if you use ARP it sends an ARP request at the same interval. After two timeouts, it cannot ping the gateway after two attempts, the gateway is considered unreachable. If it receives a reply from the gateway, then it is considered reachable and the timeout counter is reset. Once the gateway is considered down, any route that uses that gateway would be considered unreachable! Once you put check-gateway on one route, it would be used for all other routes using that same gateway. It don't flood that same device with pings from every route, RouterOS is smart enough to know to only send one ping every 10 seconds even if there are 400 routes with the same gateway.

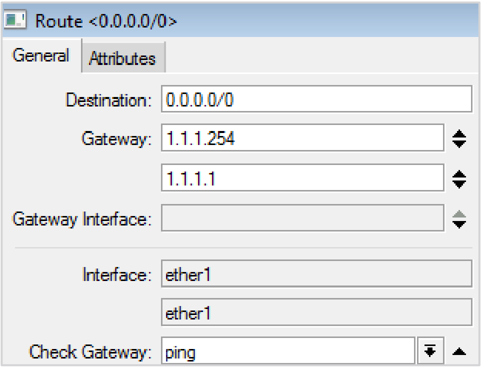
Using Distances

The distance metric is also very useful to us. This is the distance that your static route has, sometimes called “cost”. If there are of two different Internet connections and one with has a distance of 2 and one with a distance of 1, the route with a distance of 1 will be preferred over the route with a distance of 2. The route one with the shorter distance is preferred. The route with the longer distance will be shown in blue. The longer route is still a valid route, but there is another route that is preferred, just due to its sorter distance.

ECMP – Equal Cost Multiple Path

RouterOS also offers a real easy way to balance traffic across multiple gateways. This process is called ECMP or Equal Cost Multiple Path. ECMP system basically says that two gateways are of the same cost. In the following example you will see that we have specified multiple gateways therefore. Because of this, we will balance our connections going out between both gateways.

Notice that I said the word balance “connections” out the gateways; this will not balance “bandwidth”. ECMP balances connections by matching source/ and destination IPs up. Once computer 1 establishes, a connection, that source/destination IP information will stay on one interface until that connection is completed. However, that same computer could establish another connection that may go out the second gateway. Let's put this into an example of web page surfing. Normally, when you open a webpage, this opens up from 5-20 connections, just to get the images, etc. Now let's say that some of those images are on different servers, etc. Some of those connections could go out one gateway and some could go out another. This normally is not a problem for HTTP traffic. But HTTPS depends on the source IP address you are coming from to ensure encryption. ECMP will break HTTPS, so make sure you don't use ECMP for HTTPS traffic.



Route <0.0.0.0/0>

Destination: 0.0.0.0/0
Gateway: 1.1.1.254
1.1.1.1
Gateway Interface:
Interface: ether1
ether1
Check Gateway: ping

In my experience, ECMP works for only for traffic going out the same provider. I have had this turned on with one cable provider and one DSL provider, and the differences in round-trip time usually caused issues. Webpage timeouts and other weird issues occurred. On the other hand, if you have three DSL lines from the same provider ECMP usually works quite well, however, I typically put only port 80 or HTTP traffic through EMCP. For other traffic, I use routing marks and route accordingly.

When using the Check Gateway option with EMCP, note that the gateway or route will not turn blue if it is inactive, but it will be marked as inactive and not used, even though it does not indicate this.

If you have an unbalanced connection, for example, one gateway has 2 Mbps and the second has 1 Mbps, you can list the same gateway multiple times. In this example, if you had this setup you would place the gateway with the 2 Mbps service in the list twice, and the 1 Mbps gateway only once. This would be a 2:1 difference. You will need to calculate the difference in your gateways and figure up how many entries you will need for each to make it balance as much as possible.

Policy Based Routing

Policy based routing is one of the many great features of RouterOS. This feature allows you to create multiple routing tables on one router. This is great if you have multiple connections, and/or wish to control how traffic flows. Basically you will have a policy rule that will match data in some way, and then determine what routing table to use. With policy-based routing, you can send data from one customer through an anti-virus/anti-spyware box and other customer's data directly out to the Internet. I have also used this as well in enterprise applications where I want remote site users to go to the main site for Internet access instead of going out their local gateway.

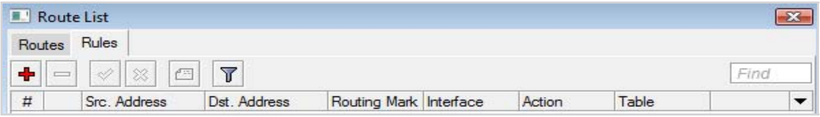
There are a few basic ideas that you have to understand when you are doing policy based routing. First, you have to identify the traffic that you wish to change the routing on. Typically we will use our Mangle system to identify the traffic and then place a routing-mark on the packets and/or connection. Second, you have to specify where you want the identified traffic to go. We will create a second routing table with separate routes on it. This table is distinguished by a routing mark. The last part is a routing rule, which basically says; if you have this routing mark, then process those packets on said routing table.

Since version 4 though, MikroTik implemented a change in which you can effect policy based routing in two steps. RouterOS identifies the routing-mark inside the Mangle system with the routing-mark of the routing tables. So there is no need for a routing rule, as the routing-mark from the Mangle and the routing-mark on the routing table identifies where to send the traffic. You may still use the routing rules to modify this default behavior though.

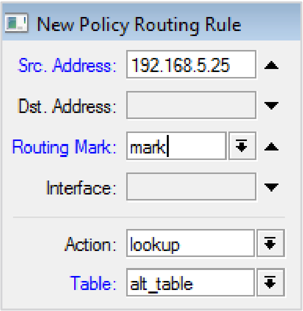
So the steps in the process include you need are: traffic identification, use of a separate routing table, and use of a policy (a routing rule). In v4+ you can remove the policy or routing rule unless needed.

Routing Policies

There are two ways to identify your traffic; one is using routing marks under the Mangle system. The second is to directly identify the traffic under your routing policies. To access your routing policies, or (more commonly called "routing rules") click on IP → Routes → Rules tab. This is your routing rules section.

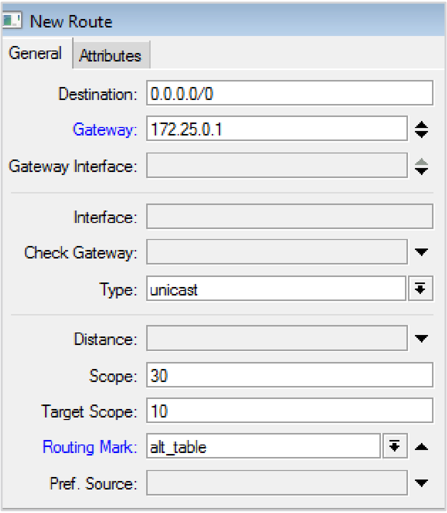


Under your routing rules, you can create rules that identify traffic directly. Typically, I will use a Source Address. I identify the traffic by specifying this address, but you can also use a routing mark. Remember, these are rules, and any rules in RouterOS are processed from the top down in ordered fashion. Each rule is attempting to match traffic. If you specify both a Source Address and a routing mark, both will have to match for it to work.



Src. Address: 192.168.5.25
Dst. Address:
Routing Mark: mark
Interface:
Action: lookup
Table: alt_table

Under the action section, we are going to do a lookup, but you can also drop or make that traffic unreachable. The lookup action simply says use the listed table to find the proper routing.

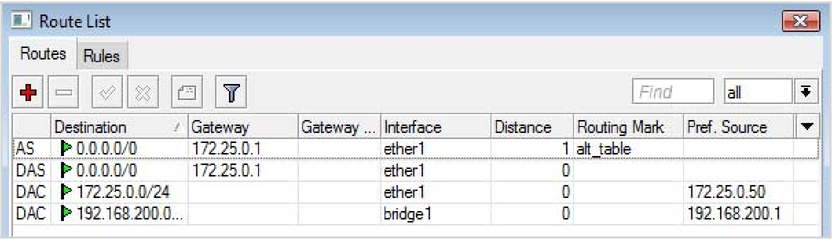


Destination: 0.0.0.0/0
Gateway: 172.25.0.1
Gateway Interface:
Interface:
Check Gateway:
Type: unicast
Distance:
Scope: 30
Target Scope: 10
Routing Mark: alt_table
Pref. Source:

To access your routing table, click on IP → Routes, and select the Routes tab. Under this tab we have a dropdown on the far right. Normally, this will say all; however, we can add as many routing tables as we wish to and give them any name. In our example, we add a default route out to a second table, and we will call it alt_table, like the example above.

Note, we have added the normal destination for a default route, and setup a proper gateway. Below, we changed our Routing mark to be alt_table.

To the left we see this rule has been added. Note, that we clicked on the right drop down and selected just the alt_table. This allows us to only see the alt_table rules. Sometimes you may get this item as a blue entry. The reason for this is that you may not have a routing policy defined yet for that table. In this case, we created one before we added our new route, so it becomes active right away.



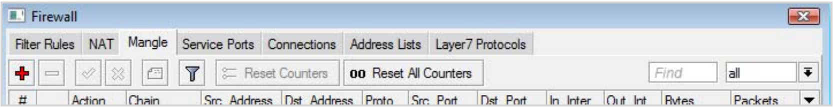
	Destination	Gateway	Gateway ...	Interface	Distance	Routing Mark	Pref. Source
AS	▶ 0.0.0.0/0	172.25.0.1		ether1	1	alt_table	
DAS	▶ 0.0.0.0/0	172.25.0.1		ether1	0		
DAC	▶ 172.25.0.0/24			ether1	0		172.25.0.50
DAC	▶ 192.168.200.0...			bridge1	0		192.168.200.1

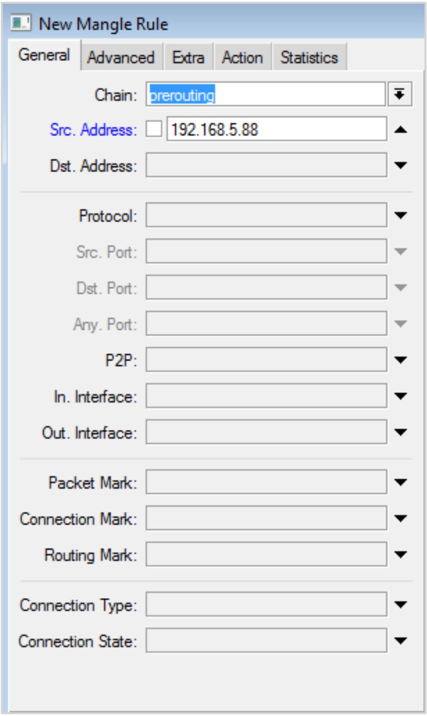
In the above list, note that we have two default routes! This occurs because the first one has a routing mark that directs traffic to another routing table.

QUICKTIP: When using multiple routing tables, secondary tables (tables that are not the main table) will failover to the main table if there is not a matching route. However, if you use a default route in your secondary table, it will never fail to the main table.

Using Mangle to Route Traffic

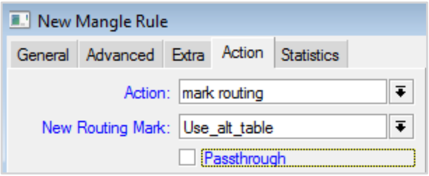
Earlier, we said another way to identify traffic is by using routing marks in Mangle. To access your Mangle system, you will click on IP → Firewall → Mangle Tab.





Chain: prerouting

Src. Address: ☐ 192.168.5.88



Action: mark routing

New Routing Mark: Use_alt_table

☐ Passthrough

Your Mangle system is very powerful. You can use any of the ways shown wish to identify traffic here. These are rules and just like any rule in MikroTik they are processed in order. Once the packet has been matched, it may or may not continue to be processed by other rules. In our case, once we match it, we will typically stop the processing by specifying a routing mark.

To do this, click on the Action tab, and select "mark-routing" as the action; you can then define the New Routing Mark as you see fit. Uncheck the Passthrough option; so that once data matches the rule, it will stop processing other rules.

Once you have your routing mark, then you can setup your policy rule to use your routing mark, and then lookup on the corresponding table.

Firewalling

RouterOS has a full featured Firewall. The Firewall will allow you to permit or deny different types of traffic, based on a set of rules. Firewalls are used to not only prevent unauthorized access to your router, and your network, but they also can be used to prevent unwanted or unnecessary data from flowing through your network.

Traffic Identification

First off, I love dealing with Firewalls, I spend more time working on Firewall rules, management, and coming up with creative ways to achieve to the desired result. The most important function in Firewalling is, ”traffic identification”. Just like with many other features, your Firewall deals with traffic coming to, from and through your router! There are all kinds of traffic and being able to identify the traffic you want is sometimes the hardest part. Think of picking out that nice red sedan that you want, out of 20,000 cars as they go down a 10 lane highway! This becomes hard to watch for, and you have to know how to identify it. This is ever harder when the bulk of the cars are red!

So first, let’s talk a bit more about traffic identification. You can identify traffic in a number of ways. With RouterOS you can use your Firewall to identify traffic by what interface it either arrives or leaves on. This is a very broad approach.

TCP/IP as you know has a number of protocols. The most common ones are TCP and UDP, but there are others commonly used, such as GRE and ICMP. If we identify traffic by protocol, now we know what “highway” they are coming and going on. This again, is still quite broad. So we go deeper, and look at what port they are using. Both TCP and UDP have 65,000+ ports each, so figure you have 65,000+ lanes to each highway. That is a huge number of lanes to watch, so we need to further narrow it down further.

Sometimes we have flags to help us. Between all of these highways in and out, each (protocols), and then each port or lane, we have narrowed the traffic down quite a bit. But what happens if we have a flag? If we have a flag with a number on it, that’s what DSCP or TOS bits do for us. So now we can identify for just cars on this highway, going out of our city, in this lane, that are red, and have a flag that has the number 46.

RouterOS offers the ability to build rules based on many different variables all at the same time. This allows you to specify “I want this car, with this flag, on this road, going to this highway on this lane”. Not only can you identify packets through set rules like this, but you can also setup methods to match only a percentage of traffic, connection counts, and many other methods as well. We also have a tool called Torch; that we will cover this in the tools section, which will help you further identify the traffic as well.

Once you have identified the desired traffic, now you can do something with it! That’s the goal! This same process is used when you identify traffic through the Firewall filters or through your Mangle system.

Rules

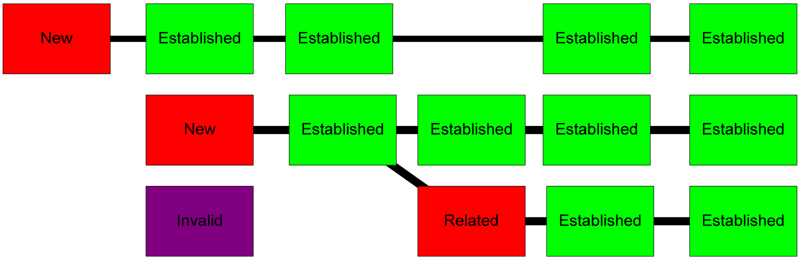
RouterOS uses rules in its Firewall and Mangle systems. It also uses rules in several other places. Rules are an ordered list that is typically processed in order. In the Firewall/Mangle system, these rules are used to match data, once the data is matched; the rule has an action assigned to it. In reading this section and others, you will see me mention "drop rule" and "log rule". These are simply an entry in your Firewall system that is assigned an action of Drop or Log. They are types of rules, but RouterOS does not distinguish between these types. It's just an entry that needs to be processed and once data matches that entry, it will perform the action specified.

Understanding Connection States

You need to understand connection states when configuring for your Firewall rules. There are four types of connection states in RouterOS: Invalid, New, Established and Related. Normally, when a connection is established between point A and point B, it goes through two connection states. One is the New connection state. This connection state means that the connection is being created. Once the connection is created, the state becomes Established. The bulk of your data movement and packets are going to be with Established connections.

An Established connection, sometimes, calls upon another connection state to do something else while the original connection continues on. A good example of this would be your web browser. The first connection obtains the HTML code for the page. During this connection, it will call upon other Related connections to obtain images, graphics and sound. Each one of these separate connections never goes through a New connection state but rather a Related connection state. These Related connections, then enter the Established state once the connection is running.

Below is a chart of common processes of a connection.



It is important to understand connection states with the RouterOS Firewall because as it gives you the ability to understand how connections are created. By understanding how these connections are made, you can create a better Firewall, not only one that protects your devices but also one that is efficient as well. We typically assume that invalid packets are hack attempts, as they are packet injections into the connection stream that are out of order or do not match the connection.

Firewall Efficiencies

Creating your Firewall efficiently is very important, especially on RouterBOARD products. Remember that every packet is processed though the Firewall and your Firewall should limit the number of rules that any packet needs to go though. The packet is processed though the Firewall in order, so how can we maximize the efficiency of your Firewall rules?

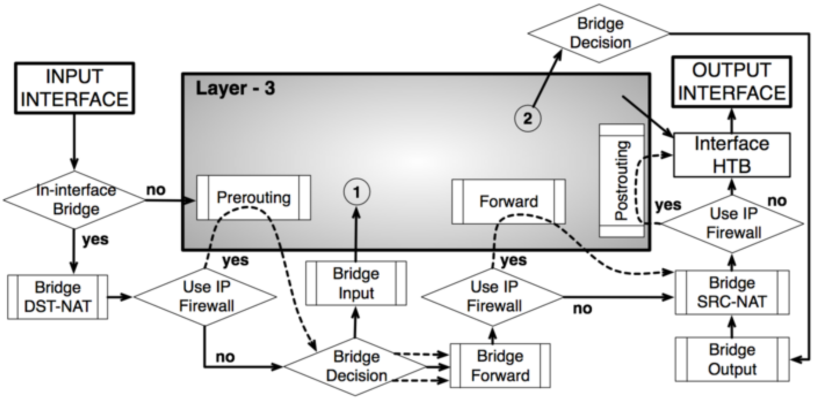
The single largest step is to process on the New-type connection states only. These would be New, and Related connection states. To do this, the top two most rules will typically be accept rules for both Established and Related connections. The ideology behind this is, if you never let the connection become Established, then there will be no further data. If you did let it become Established, why process rules on the rest packets? You allowed the connection state to become established. If you did not want the connection, why allow the new connection state packets? In regards to the related connections, again, you allowed the original connection, why do you need to do anything but allow your related connections?

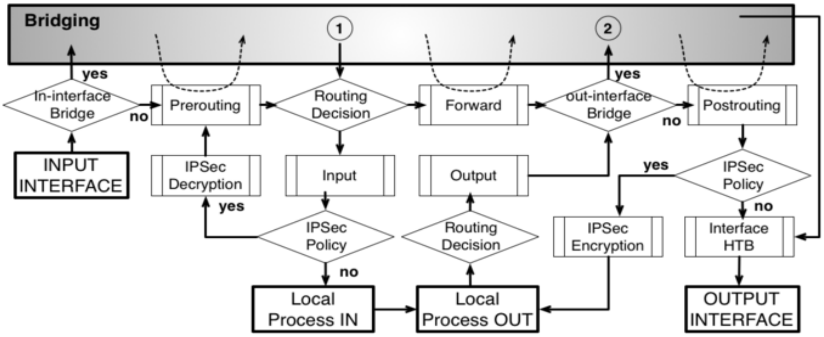
With this said, by creating two accept rules at the top of your chain, you will limit the number of rules your packets have to pass though. The rest of your Firewall would only process on New connections. So how much more efficient is this? Is it worth doing? Let's look at a single connection, this would be a download of a ISO image, we will say around 550 Megabytes. In this case, the first packet would be the request for the connection, if we did not have the Established and Related Accepts, this packet would flow though the Firewall and look for anything that would match the data. If not found, the default action is accept and the connection would then change to an Established state. Now in the Established state the file begins to download. As the download progresses, every packet in the download rules though ALL of the Firewall rules, and then since there is no rule that matches, is accepted. If we had the two rules at the top of our chain, the first packet would go through those two, and run through all of the Firewall rules, and eventually be accepted. The rest of the packets, once in the Established state would be then accepted on one of the first two rules and no other rules would be processed. 99% of the entire 550 megabyte transfer would be considered Established and would be accepted by the first rule; no other rule processing would be done.

Here are some good suggestions about connections states. First, drop your invalid connections, as typically they are hack attempts. Process your rules based on New connection states, and allow your Related and Established connections. This will minimize your amount of CPU usage that you use, while as well as still accomplishing the Firewalling features that you need.

Packet Flow in RouterOS

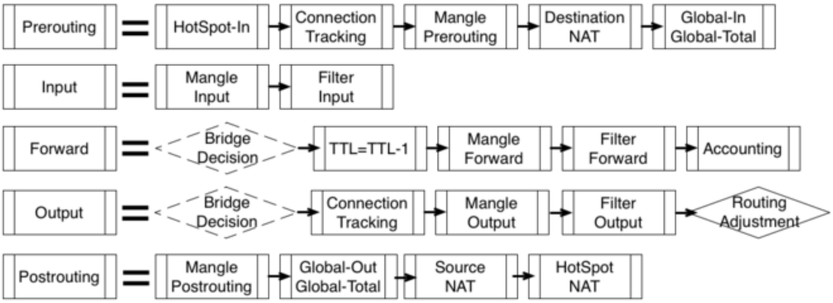
When you start building Firewall rules, you will need to understand how packets flow inside the RouterOS system. This packet flow is important; depending on several factors, the packets may use different chains etc. Below is the RouterOS packet flow diagram. You will need this to understand how packets flow. This is even more important if you wish to use several RouterOS features, like QoS, Hotspot, and other features. Many thanks to MikroTik as these diagrams are directly from their Wiki.



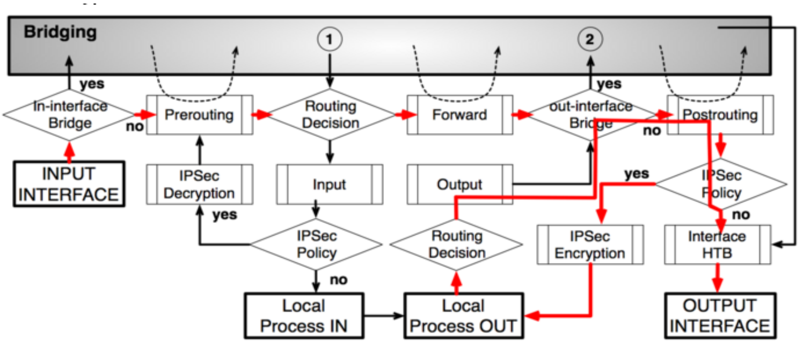


Mangle Packet Flow

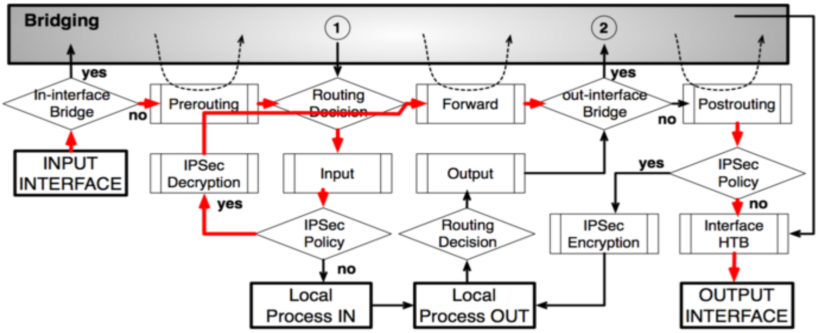
Below is the Mangle packet flow. Use this if you wish to know what RouterOS facilities are used on each of your Mangle chains.



IPSec Encryption/Decryption



Above is a diagram on the packet flow for IPSec Encryption; on the next page we have a diagram on how IPSecDecryption occurs.

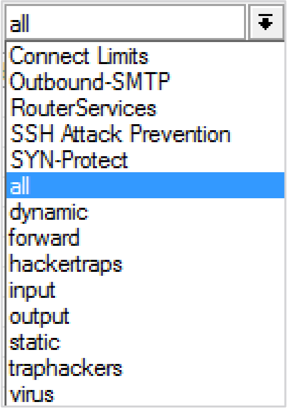


Packet Flow with MESH

MikroTik has a MESHing protocol called HWMP+, (see that section for more information) so titled; however, we have been talking about packet flow though RouterOS. At the time of this writing, the MESH system does NOT pass though the IP Firewall system. You cannot use IP Firewall and/or Mangle using this system.

Chains

Before we start working with the Firewall, we need to discuss chains. RouterOS uses chains for segmenting the different types of traffic that your router has. There are three built-in chains; these are chains that are always present in any RouterOS system. You can also create new chains for greater manageability as well. All of your rules under each chain are processed in order.



RouterOS also makes it easy to manage these chains by providing a drop-down box in the right side of the Firewall filter rules. You can manage each portion of your RouterOS Firewall more easily by grouping rules into chains, and then calling those chains from the built-in chains.

Input Chain

The input chain processes on data that is going to the router. If you have five IP addresses on the router, then any packets coming into the router for one of those IP addresses would be processed on the input chain. Use your input chain to provide access rules that allow services and authorized users onto your router and the IPs associated with the router.

Output Chain

The output chain is for data that is generated from the router. Things such as pings from the router, and ping replies from the router. Creating tunnels, using the web proxy system, and other outbound connections would be controlled here.

Forward Chain

The forward chain is used to process on packets and data that flow through the router. Most of the Firewall rules that you create will be in this chain, as this would protect customers, and networks behind your router.

Other Chains

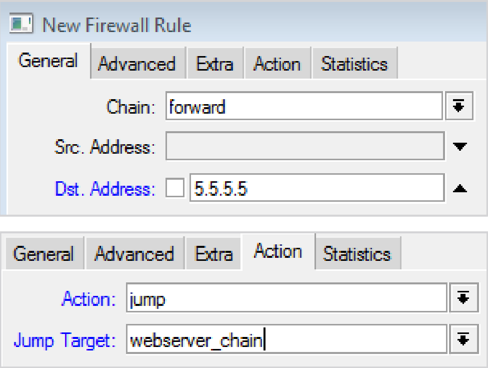
Just because you already have the three chains, does not mean you can't add more. You can create chains with any name you wish, just simply by changing the name of the chain under each rule. You will need to jump to these chains to be able to use them from one of the built-in chains.

The main purpose of these other chains, is to allow you to specify rules chains, and jump to them from the main built-in chains. This gives you the ability to provide rules based off another rule. For example, you can setup your forward chain to say if the packets are destined for your web server IP address, to send them to a chain called, web_server. Then, under the web_server chain, you can apply all of the Firewalling that you wish to, as needed. This allows you to have a completely different set of Firewall rules for one individual IP address vs. all of the other forwarding rules.

Also, as mentioned before at the beginning of the session, you can also improve manageability of your Firewall by grouping functions of your Firewall into chains, and then calling on those chains from the built-in chains.

Jumping to Chains

By default, you have the three built-in chains, input, output and forward. For organization and other reasons, you build other chains that you create names for. To use these other chains, you have to jump to them from one of the built-in chains. Remember, all data flows through the three built-in chains based on the type of traffic. For you to jump to another chain that you created, let's say your web_server chain, you will have to create a rule with a jump action under your built-in chains.



Chain: forward

Src. Address:

Dst. Address: □ 5.5.5.5

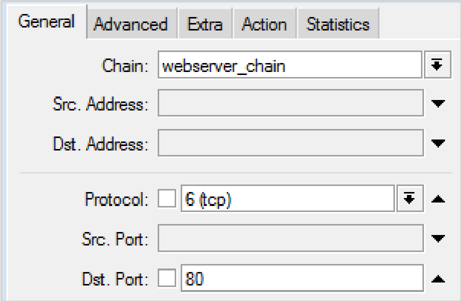
Action: jump

Jump Target: webserver_chain

So, we will assume that you have a web_server with a public IP address being routed through your RouterOS system. This web server's IP address will be 5.5.5.5 in our case. Since we are routing through our router, we will need to apply Firewall rules in the forward chain, or jump from the forward chain to our web_server chain. Since we only want to send data that is going to our web server to the web_server chain, we will apply a new rule that matches only the web server data, and then specifies a jump to our web_server chain.

We have created a rule and told it to jump to our web_server chain. Note that we have added the web server IP address as our Dst-address field as our web server IP address. This is so that the only data that will jump to this chain is data that is going to our web server IP address. We then go into our action tab, and tell the system to jump to another chain.

Now, inside the webserver_chain we can create other rules, and since we have only brought traffic that is destined for the 5.5.5.5 IP address into the webserver_chain, we don't have to specify that information again. In this rule, we say that if they are using TCP port 80, HTTP traffic, we will apply a rule called accept. This action will accept or allow the packet. Once the accept rule has matched the data, that packet will not process any further in the chain.



Chain: webserver_chain

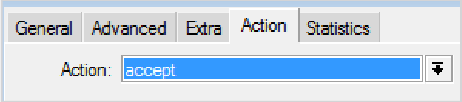
Src. Address:

Dst. Address:

Protocol: □ 6 (tcp)

Src. Port:

Dst. Port: □ 80



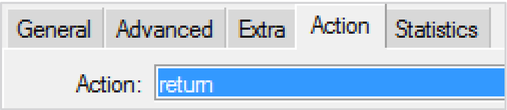
Action: accept

Since this is a web server, you may need TCP/443 opened up for secure HTTP or HTTPS traffic. If you don't, the last rule in this chain would be a drop rule. Drop is basically a deny rule. Any other traffic that was not matched against the accept rules, will be dropped. This data will not make it past your Firewall to the web server.

Returning from Chains

Once you have jumped to another chain, you have the option to return to the original chain. In our example above with the web server, there is no need to return as we have processed all of the rules based on the IP address of the web server, and anything not accepted was dropped.

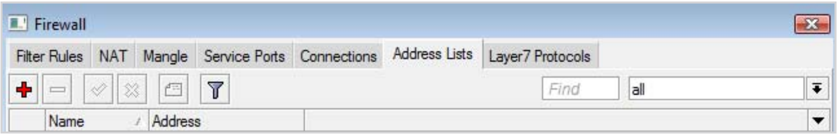
One example of jumping and needing to return is a connection limit chain. You jump to the connection limiting chain to apply a few connection limiting rules, just as an organizational method; at the bottom of your connection limit chain, you can apply a rule called return. This returns you to right under where you jumped from. You can jump from one chain to another to another if you wish, and every time you return, you return to the point from which you jumped from. Most of the time, I create the rules on my "other" chain, and then I would create a rule at the bottom of the other chain for the return. Your return action rule typically will not have any matching options, rather it matches all packets that flowed through your chain and did not match other rules.



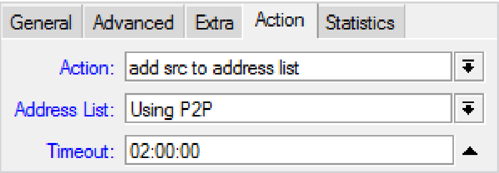
Action: return

Address-lists

Address-lists are an extremely powerful feature of RouterOS. The Address-lists give you the ability to provide a list of addresses, a single address, or an address range or subnet, which you can use in other parts of RouterOS. In the Firewall filters section, under the advanced tab, you will have the ability to match based on address these lists. To access the address-list section, click on IP → Firewall → and then the Address-lists tab.



You can add many entries in the address-list as well as have many different lists. You can also have RouterOS create Dynamic Address-lists. These lists are created when a Firewall rule is matched. Once matched, these addresses are added to the address-list of your choosing for a specified time. Note however, that these dynamic entries are just stored in RAM, so if you reboot your router, it will clear out dynamic entries. In contrast, static entries however are stored on the disk.



Action: add src to address list

Address List: Using P2P

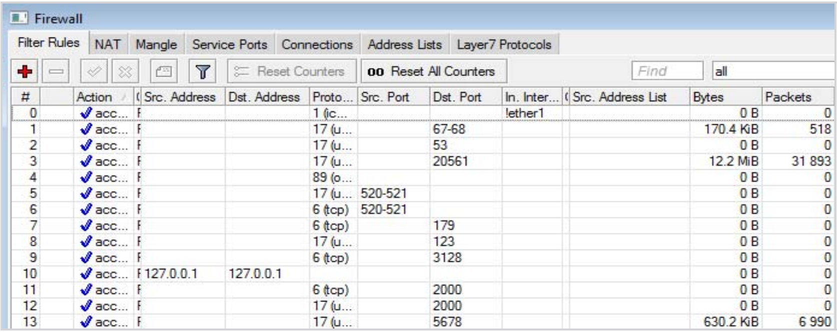
Timeout: 02:00:00

An example of using the address-list feature is to generate an address-list dynamically get created as users use Peer to Peer applications. This will create an address-list dynamically giving you the list of those users who are using Peer to Peer applications. The timeout value in your Firewall rule will determine how long the users will stay on the address-list. You can specify both Source and Destination address-lists as needed, and can create an address-list with any name.

Another use for an address-list is to create a list of possible hackers. The methodology that you will use is “detected, add, block”. First, you detect that a hack attempt is taking place. You do this by specifying Firewall rules that identify this type of traffic. Once identified, you will add their IP address is added to an address-list. Now that you have an IP address-list with these IP addresses on it, you can create another rule. I typically decide that once an address is on this address-list, even if they were doing a Port scan, or attempting a SSH brute force attack, it doesn’t matter. Now you are on the list, so I have another Firewall rule that blocks all data if you are on that address-list. Once you are listed, they can’t get past my Firewall until the timeout value has been reached, and they drop off the list. Please refer to the section under Firewalling called “SSH Attack Prevention” for more information about if you are interested in understanding this process.

How to Match Data

There are many ways to match data in RouterOS. In the following sections we will talk about common ways to match data. Keep in mind that you can use these in both the IP Firewall system, as well as the IP Mangle system to match data. The rules in the IP Firewall as well as the Mangle system are processed in order. Make sure you process them from the top down. I typically will accept at the top and drop at the bottom.



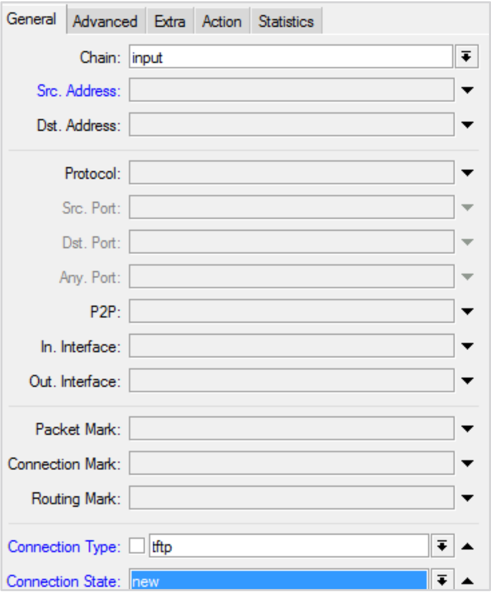
#		Action	(	Src. Address	Dst. Address	Proto...	Src. Port	Dst. Port	In. Inter...	(	Src. Address List	Bytes	Packets
0		✓ acc...	F			1 (ic...			!ether1			0 B	0
1		✓ acc...	F			17 (u...		67-68				170.4 KiB	518
2		✓ acc...	F			17 (u...		53				0 B	0
3		✓ acc...	F			17 (u...		20561				12.2 MiB	31 893
4		✓ acc...	F			89 (o...						0 B	0
5		✓ acc...	F			17 (u...	520-521					0 B	0
6		✓ acc...	F			6 (tcp)	520-521					0 B	0
7		✓ acc...	F			6 (tcp)		179				0 B	0
8		✓ acc...	F			17 (u...		123				0 B	0
9		✓ acc...	F			6 (tcp)		3128				0 B	0
10		✓ acc...	F	127.0.0.1	127.0.0.1							0 B	0
11		✓ acc...	F			6 (tcp)		2000				0 B	0
12		✓ acc...	F			17 (u...		2000				0 B	0
13		✓ acc...	F			17 (u...		5678				630.2 KiB	6 990

Right on the general tab, you can configure the basic IP matching rules. You have the ability to match based on source and destination addresses, protocol, as well as source port, destination ports, or any ports. You can also match based on your In or Out interfaces as well.

The ‘any port’ option basically says match the packet regardless if its source or destination port number, as long as one of them is the ‘any’ port. You can also use packet marks to match data as well in the Firewall rules, but make sure to follow the packet flow diagram to know how and where to put these packet marks and Firewall rules. The packet flow diagram can be found a few chapters back in the “Packet Flow in RouterOS” section.

In the Advanced tab, you have even more options for matching common TCP/IP data. We talked about building and making those source and destination address-lists dynamically under the Address-lists section. Once you have IP address-lists created, you can then match based on those lists under the Advanced tab, or you can say NOT this address-list, by checking the ! box.

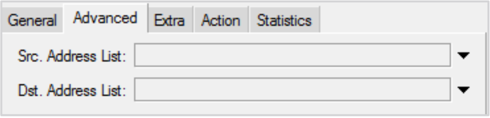
There are a number of other methods of matching data in your Firewall and Mangle. Keep in mind that you can combine fields and types to create a match. For example you wished, you can say you want all TCP/80 traffic with a source address of the source address-list called “local IPs”. You can also match data based on packets that have a ToS bit of 4 and come in your Internet Ethernet port. The key is to put all of your Firewall rules together to make it do whatever you want!



Chain: input

Connection Type: □ tftp

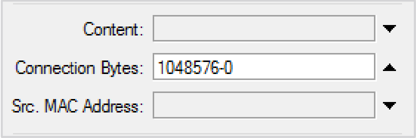
Connection State: new



Connection Bytes

Normally I would reserve not talking about connection bytes as it’s sometimes difficult to properly communicate. Connection bytes only work on TCP connections first of all. This rule gives you the ability to match based on the amount of data a connections transferred by a connection of data.

A really good usage for this is looking for extended downloads. The example in the graphic shows a rule looking for connections that have gone over 1Mbps of transferred data. Of course, you can change this number to whatever quantity you want to have. Once the connection goes over 1Mbps, it will start matching this rule. The rule is in Bytes so calculate accordingly.

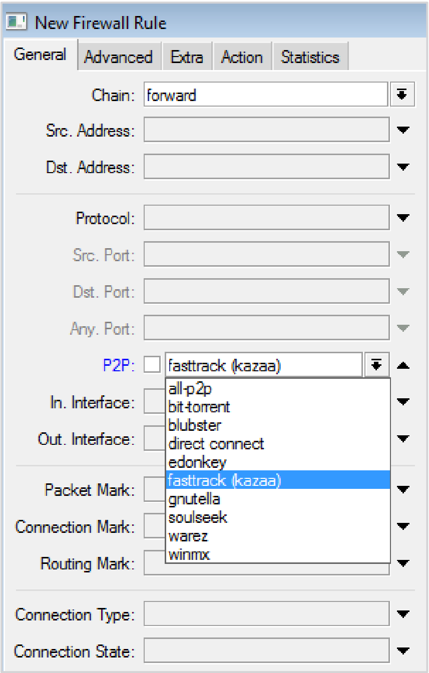


Connection Bytes: 1048576-0

Now that you have this rule, you can do something with it. Sometimes I will use a connection or packet mark, with a special rule, that puts the connection into an extended download queue. Everyone in that queue can then contend fight over the bandwidth in that one queue for these extended downloads. Normally though, the bandwidth limit it would not be 1 Mbps; it would be something like 200+ Mbps for most of my configurations, but the bandwidth limit can be any value that you choose is a preference.

Built-In Peer to Peer Filtering

RouterOS is used by many ISPs (Internet Service Providers) and WISPs (Wireless Internet Service Providers). In many cases P2P or Peer to Peer (P2P) applications can be disruptive to some services by creating many packets per second and using up quite a bit of Access Point bandwidth. RouterOS gives you the ability to create your Firewall rules that filter P2P packets. These filters are extremely optimized Layer-7 filters. Firewall rule functions are constantly being updated in newer by RouterOS versions, therefore due to this; the latest RouterOS version provides will give you better matching compared to older versions.



Chain: forward

P2P: □ fasttrack (kazaa)

You can select several different types of P2P or you can select all-p2p, to match all of the types. Once you have used this filter option to match P2P data, you can do whatever you wish with it. You can apply other options to do connection limiting on this, or you can even drop the traffic. In the address-list section, we mentioned that you can also use this filter to add P2P users to an Address-list, and then base other rules off of that.

This P2P feature is also in the simple queue section and allows you to assign bandwidth limits to P2P applications as well. This is discussed in more depth in the Traffic Management section.

Layer-7 Filters

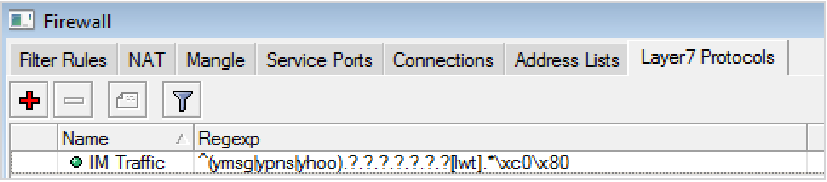
Normally, when you identify traffic by, you are using port, protocol, IP addresses, etc. However, some applications use common ports that are used for other types of data. Some instant messenger applications will use TCP port 80 to connect with IM servers. TCP port 80 is more commonly used for HTTP traffic. This IM data is virtually impossible to match and catch without affecting other types of traffic. That's where the Layer-7 Firewalling abilities of RouterOS come in to help out.

When you apply most RouterOS Firewall filters, you are really only looking at the first 40 bits of data, or the TCP header data. The header contains your IP addresses, port numbers as well as options like TOS, etc. This is less than 2% of the data contained in many packets. Because of to this, we can process rules very quickly. However, when we start doing Layer-7 or application layer filtering, RouterOS must start looking at the entire packet. Therefore the amount of data that must be we processed goes from 20 bytes to the entire 1500 byte or larger packet. Since we are now processing the entire packet, we can look for data inside the packet that is common to a specific application.

```
^(ymsg|ypns|yhoo).?.?.?.?.
?.?.?[lwt].*\xc0\x80
```

If we use the IM or instant messaging traffic that we talked about earlier as an example, we can match data based on a Layer-7 filter that defines what the packet must contain. If the packet it does contain IM data, the packet it will match the filter. The following to the right example illustrates you an example of matching based on packet content.

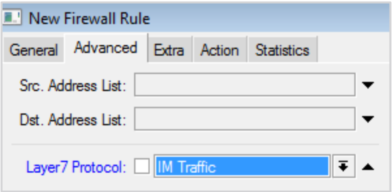
To match via this we have to first define what the Layer-7 filter will be matching. This is done in the Layer-7 tab of the Firewall. To get to this, click the IP → Firewall → Layer-7 Protocols tabs.



Name	Regexp
IM Traffic	^(ymsg|ypns|yhoo).?.?.?.?.?.?.?[lwt].*\xc0\x80

In the above example, we have defined a Layer-7 Protocol and given it a name. This name can be anything but you will use it in your Firewall rule.

When you create that Firewall rule, you will use the Advanced tab, and select the Layer-7 Protocol that you created in the Layer-7 Protocol tab. Once you do this, you can then define an action based on that protocol.



Src. Address List:

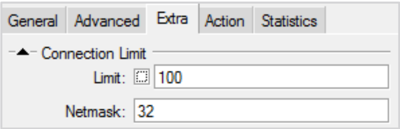
Dst. Address List:

Layer7 Protocol: □ IM Traffic

With Layer-7 processing you will consume quite a bit more CPU time, as you are processing the entire packet. You'll need to do your own testing, but assume 2 to 3 times more CPU for the same amount of data when you are doing Layer-7 processing. I prefer to leave this type of processing in the core routers, where CPU power is plentiful.

Connection Limiting

In the Extra tab under your Firewall rules, you also have a feature called Connection Limiting. This feature is very simple to use. I use it quite a bit to limit P2P applications from creating too many connections. I also use it to prevent a residential client from becoming a spammer.



Limit: □ 100

Netmask: 32

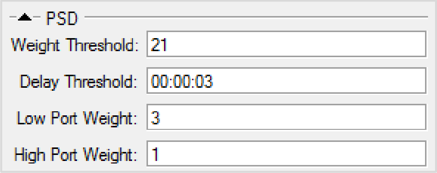
To use Connection Limiting feature, you will need to select the TCP protocol because as it is a connection-based protocol in the TCP/IP suite. In most cases, I will also select either a Source Address subnet or a Source Address-list listing on my network, so that I apply Connection Limiting this to only my network instead of to the entire Internet. Once this is done, specify the number of connections in we can apply the limits. The limit field is for the number of connections. If you want to limit your customers to 100 connections, you would enter that number in the limit field. The netmask field defines the size in what size of the subnet to apply this limit too. If you defined a /8 in your source address, and then defined a netmask of 32, you would end up giving every IP address on your /8 100 connections. If you changed this to a netmask of 24, that would allow 100 connections per /24 network under your /8. In most cases you will use a netmask of 32 to say every IP address receives 100 connections.

Connection Limiting Chart

What it actually does.	**Limit Field**	**Netmask**
100 TCP connections per individual IP or /32 address range	100	32
100 TCP connections per /24 address range	100	24
50 TCP Connections per /29 address range	50	29

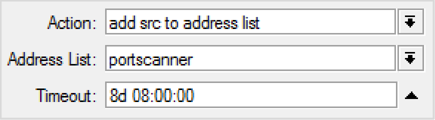
I get a lot of questions that ask on what to set the connection limits too. I have found that a limit of 25-30 connections per residential account is typically a good number. On business connections we typically let them run without a limit. This is typically because they are paying a premium for their connection and we don't know how many PCs or devices they have behind their single IP address. If though, you use the rule of thumb of 20 connections per workstation, you typically will not have issues.

Port Scan Detection



PSD
Weight Threshold: 21
Delay Threshold: 00:00:03
Low Port Weight: 3
High Port Weight: 1

RouterOS offers the ability to match against port scanning activities. It does this by providing a cost or weight when someone attempts to open a port. There are three weight variables included in the PSD, or (port scan detection) system. The most important is the low port weight. This is for ports under 1024. It is typically normal that ports below 1024 are more commonly scanned as most basic Internet services are provided below port 1024. RouterOS allows you to define a weight for these ports, and then a weight for high ports, or ports above 1024. Typically the weight on the high ports would be less than the lower ports.



Action: add src to address list
Address List: portscanner
Timeout: 8d 08:00:00

You then specify a delay threshold and a weight threshold. When the weight threshold is exceeded within the specified delay threshold a match to the rule takes place. Remember, this is a rule, so you are matching data based on the rule. Once the data is matched, you have to specify what do with it. I will typically place an action to add the source IP address to an address-list, for example, typically "port scanners" or some other easily recognizable name. Then I typically will place another rule that will drop all traffic from the "port scanners" address-list.

Ingress Priority and TOS / DSCP

I put these two features together even though they are separate items, because but they both deal with priorities. The ingress priority is a function of WMM (Wireless Multi-Media) or VLAN priorities. If you set priorities with VLAN or WMM you will be able to match data based on those priorities.

The DSCP or TOS bit is a priority-based number that is included in the IP header information of the packet. Because this information is transmitted with the packet you can perform QoS and other data matching very easily. In some cases, you can do ingress marking by letting your edge routers identify the traffic and place TOS bits on all of the traffic. Then, in your core network, or within your backbone, you can process based just on the TOS bits. If you want to change priorities, you can do it by simply by changing the TOS bit.

Since RouterOS can change TOS bits as they pass though the router, we can do anything we wish, including removing the TOS bits. Some networks drop all of these as they pass through their network, however, typically TOS bits are designed to help identify critical traffic, such as VoIP or other latency sensitive applications. I typically would use them to help identify VoIP Traffic on networks, but keep in mind that any network that the packets bits pass though, could have changed the TOS bits.

Random

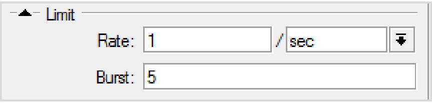
Using random can be fun. For example, that when I want to aggregate my wife, I can use the random command to drop 30% of her web traffic. Trust me; I get a response from her in about two minutes from enabling that rule. I don't know which is worse, the fact that I actually use that rule, or the fact I have it just disabled in my router, ready for action.

As the random switch implies, it allows you to setup a random matching ability. Besides aggravating the wife with it, it can also can do some good. If you have an application that you wish to test based on with a simulated poor connection, you can randomly drop packets based on a percentage of the packets. This will simulate the performance of a T1 or other type of link that has packet loss on it.

Limit/DST Limit

This function allows you to effectively limit packet rates based on time. Blocking denial-of-service (DOS) or Ping-Of-Death (PoD) attacks works quite well when you limit the packets per second. You can also use this limit system to limit the amount of logging messages per second, and other functions as well.

The configuration is very simple. You have a rate option, which is the maximum average packet rate. This is normally measured in pps or (packets per second). You have the option though of changing the time variable, so you can say, per second, or per minute, if needed. You also have a burst option; and this is how many packets to allow in a burst.



Limit

Rate: 1 / sec

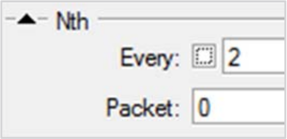
Burst: 5

The thing to keep in mind with the limit value is that it does not match data until the data rate exceeds the rate that you specify. Once it goes over the rate, then it matches data. Once the data is matched, remember, you will then have an action. The rule, matches data once the limit value is exceeded. Once that occurs, the rule matches and then you can configure your rule to perform some kind of action. This could be drop the excess data or use an address-list to perform some other actions. Remember, that if you specify this rule without other options, then it will ONLY look for the rates, not rates based on Src-Address or other options.

The DST Limit further limits packets per second but this time it limits per IP/port. In the limit system, if you place a limit of 40 pps for an entire /24 subnet, then that is exactly what you would get. The entire subnet would have a limit of 40 pps. If you change the rule from a limit to a dst-limit you can revise that limit to 10 pps per destination IP and port. Instead of the entire /24 subnet being limited to 40 pps, an individual computer can have 10 pps per port. If

Nth

Nth is a value that you can use to match Nth amount of data. In version 3 of RouterOS this is handled differently than v2.x. It is now possible to match 50% of your data with only one rule. The key to understand Nth, is that you are matching what packet out of what packet count. If you create a rule matching every 2nd packet with a packet number of 0, then this rule will assume we are dividing up all matching packets into two streams, furthermore, the rule specifics that we are going to match on packet 0 of those streams. So our rule shows every 2, packet 0, so as packets go by RouterOS divides them up by two. The first packet would be considered stream 0, the second packet would be considered stream 1, and then the third packet would be back to stream 0, and so on. This rule also has the option of Packet 0, so we are only matching packets that fall into the stream 0. We would need another rule, with the same Every field, but the packet would be packet 1 to match the REST of the data that the first rule does not match.



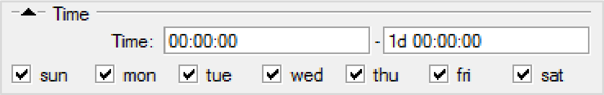
Every: □ 2
Packet: 0

Once we match the rules, we can then preform some action with them, such as adding them to an address-list. The major use of the Nth was to help load balance connections out multiple gateways; however, this has been replaced with the PCC function. Please refer to the PCC (Per-Connection-Classifer) section for more information.

Time

The time field is exactly what it sounds like. It lets your rule match at various times of the day! This works great if you wish to allow access to some sites or change bandwidth allocations at different times of the day. The rule works just by matching the time to the system clock. Remember, on RouterBOARDs, they do not keep the clock set after a reboot, so make sure they can get to a NTP Server to reset their time. For x86 applications, you won't have to worry about this.

Simply specify the start time and the duration. Then use the check boxes to specify on what day or days to apply the rule applies for with the check boxes. I typically use this in combination with other items. I normally have one rule that has time options checked, and then below that rule another option with no time rules. If it is during the specified time frame, the first rule will match and take the correct action. If that rule does not match due to it being out of the matching time frame time, then it will fail over to the second rule.



Time

Time: 00:00:00 - 1d 00:00:00

☑ sun ☑ mon ☑ tue ☑ wed ☑ thu ☑ fri ☑ sat

Firewall Actions

Inside your Firewall there are many different actions. Some just accept data, some deny data, and others can change data. In the sections to follow, I will discuss the different types of Firewalling actions that you can use in RouterOS.

Accept

Accept is very simple operation; it accepts" or allows data traffic. By default RouterOS "allows all". In other words, with no Firewall rules, everything is allowed. There is nothing blocked, no data is not passed.

The typical usage for accept rules is to allow very specific data. One common practice with Firewalling, especially in the enterprise, is to deny everything but what is needed. You create accept rules for all of the types of traffic that you will allow. Then at the end of the chain, place a rule that denies everything. As data flows through the Firewall rules, if it matches against one of the accept rules, the packet is matched and no further rule processing is done. However, if the packet gets to the bottom of the list the "deny all" rule will block that traffic.

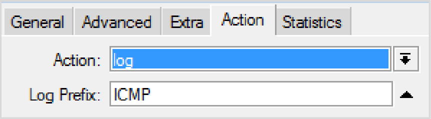
Drop

When you drop data, this means that you are denying it. You have matched that packet, and that packet is no longer processed. It does not forward through the router; it is not processed by the RouterOS system in any way. The packet is basically ignored as if it was never received it.

I commonly use drop with a connection limiting rule. Once there are xx number of connections, then the connection limit rule starts to match data. If I used an accept rule data over the connection limits I had specified would be accepted and continue on, nothing would really change and no connections would be denied. By making that rule a deny rule, once an IP goes over that connection limit, it will start dropping or denying connections above the connection limit; only data above the connection limit would be dropped. An example would be if we limited a customer to 50 TCP connections, as they attempt to establish the 51st, the new TCP connection request packet would be dropped, and the customer would not be able to establish the 51st connection.

Logging

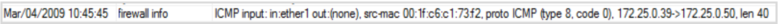
Inside your Firewall you can also perform logging actions on your data. This logging action allows you to identify traffic. The best use for this is to see what kind of data is hitting a drop rule. Right before your "drop all" rule, place a log rule. This will log all data that makes it to the "drop all" log rule, and unlike most other actions, the log rule will let the packet continue to process down the rule list. The next rule though, is a drop all rule. This way you will get information on what your drop rule is dropping. The Log Prefix information is simply that, an informational prefix that is appended to all of the logs generated.



Action: log

Log Prefix: ICMP

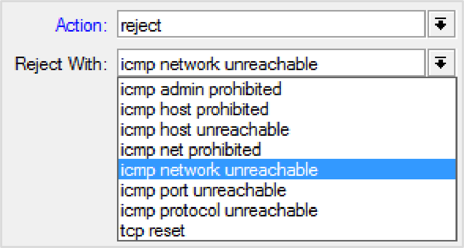
The logging rule places these logs into memory as 'Firewall info' types. You can then use your logging actions to be able to see these in your logs, or send the logs to a Syslog system. You can get more information on how to send to a Syslog server, by reading the logging section in the "Nuts and Bolts" chapter. Below you can see what the output is inside your log memory. Note that we have the ICMP prefix that we defined above. This information will help you identify specific traffic as it passes through your Firewall.



```
Mar/04/2009 10:45:45   firewall info          ICMP input: in:ether1 out:(none), src-mac 00:1f:c6:c1:73:f2, proto ICMP (type 8, code 0), 172.25.0.39->172.25.0.50, len 40
```

Reject

The reject action is solely for ICMP packets. This rule stops the ICMP packets, and then sends a reject message back. Once you select the reject action, you can then specify what reject message to respond with.



Action: reject

Reject With: icmp network unreachable

Tarpit

The Tarpit action is used to simply tick off hackers! Yep, that’s right. When hackers attempt to open connections, either for DOS or other types of attacks, they send a TCP SYN packet. This basically says, “open a connection”. What The Tarpit action does is replies to this, with a SYN/ACK, saying the connection is open; however, Tarpit doesn’t open a connection; it simply then drops everything else. To the hacker, the TCP connection is open, and there is no response to the hacker’s “close connection” packets. This keeps these connections open on the hackers system, and consumes resources their system resources. The end result is, the hacker has many of open connections, that don’t respond, and in the end, makes them mad!

Protecting Your Router

I would like to go over some common ways to protect your router. Some of these are common-sense measures, but I like to cover them anyway. Here is a check list that you can use to ensure that your router is secure.

- Change your Admin password!
- Add another user to the system and disable the Admin account!
- Select a good, strong password.
- Disable services that are not needed.
- If you don't use Telnet, FTP or SSH, turn them off!
- Input Chain: Only allow Established and Related Connections in your Firewall.
- Input Chain: Identify port scanners and massive SYN attacks.
- Add Source IPs to address-lists.
- Input Chain: Drop hackers and PSD IPs from the dynamic address-lists.
- Input Chain: Only allow services that you are using on your router in your Firewall.
- Limit ICMP pings to something manageable.
- Drop excessive pings.
- Allow only services you use:
    - WinBox
    - SSH, Telnet, or FTP
- Input Chain: Only allow management connections from trusted IP addresses.
- Build an Address-list for management IPs.
    - Drop other packets not from management IP
- Input Chain: Log other data that makes it past your standard rules.
- Input Chain: Drop everything else besides what you allow.

The basic idea behind these suggestions is to disable services you don't need, block DOS and PSD IPs once you identify them, allow only traffic from management IP subnets, and drop everything else!

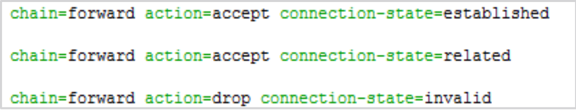
Protecting Networks

Common Firewall Options

Putting the Firewall to use for you and your customers can be challenging. RouterOS offers so many options that building your Firewall may seem like a daunting task. As a router administrator, putting it all together can be hard work.

First off, we want to prevent the unwanted traffic that we don't need on our network. TCP-based connections are the place to start. We will block invalid connections, but allow those Established and Related connections. This will keep us from processing excessive data by focusing on our efforts to block the initial creation of the undesired connections. This also will keep our CPU time down. To provide basic Firewalling, I would also look for port scanners in your forward chain just like in the input chain. Add those detected users to an address-list and block them.

Next, work to prevent some data crossing your network. Both TCP and UDP ports 135-139 are commonly used for worms and viruses. These are the ports that are used by NetBIOS traffic, and in my opinion, should never traverse a public network. A common example of this is two users on a network, file sharing directly in Windows. The NetBIOS ports should be blocked on most ISP networks. Not blocking these allows for hackers, viruses, etc., to get into your customer computers. TCP/UDP 445 is also NetBIOS ports and should be treated as such.



```
chain=forward action=accept connection-state=established

chain=forward action=accept connection-state=related

chain=forward action=drop connection-state=invalid
```

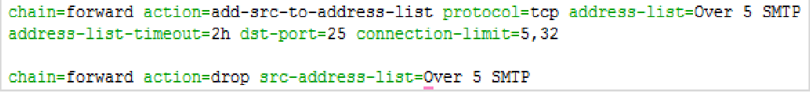
What about viruses? There are a number of virus scripts, a script; a set of terminal commands that would automatically enter a number of Firewall rules into your router simply by pasting them into a terminal window, on the Internet. You will need to look over these before adding them to your system, as some of them may have undesirable affects. So be sure you know what you are doing and what it will affect. I have seen some of these scripts block common ports that are regularly used. So be very careful when applying something that you did not make. I do have a small section on pre-made Firewalls that I would read over prior to implementing a pre-made script.

In our input chains, we also limited ICMP packets and TCP SYN packets when they come into our network. We can do the same thing with our forward chain, helping protect our customers from POD, (Ping of Death) attacks, as well as DOS attacks that flood systems with connection attempts. I typically will put a TCP SYN limit of 300-400 per second per IP. You can also prevent large pings from going through your router completely as well. Preventing large ICMP (ping) packets from flowing though you router will help with PoD attacks, but some customers may use large pings to troubleshoot connectivity.

SPAM Prevention

As an Internet provider, you may wish to prevent network users from sending out SPAM. This is a difficult task, as there may be legitimate mail servers operating on your network. Even with this, there are a number of methods that you can control SPAM on your network. The first way is simple connection limiting. Most mail system will send outbound mail via TCP port 25. This is the SMTP port.

Because mail servers commonly use port 25 for mail, identifying mail traffic is fairly easy. If you apply a connection limit per IP just on port 25, this will be the first step. Residential users typically will never need more than five TCP port 25 connections out. They typically send a single message via a single connection; therefore, any residential user going over this limit very well may have been infected with a virus or worm that causes their computer to send out SPAM.   A rule that prohibits over five connections, and then adds the source IP to an address-list will allow you to identify spammers' user. Set this rule up with for an hour or so timeout period on the address-list. Create a second rule to block all port 25 outbound access based on that address-list. Once a computer is infected and attempting to open more than 5 connections out, it will be placed on the list, and all SMTP traffic will be blocked for an hour. After that, if that address continues to attempt to open more than five simultaneous connections, they will simply get added again.

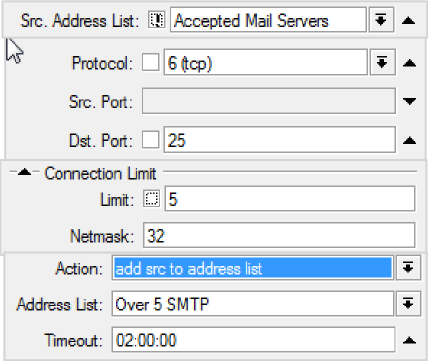


```
chain=forward action=add-src-to-address-list protocol=tcp address-list=Over 5 SMTP
address-list-timeout=2h dst-port=25 connection-limit=5,32

chain=forward action=drop src-address-list=Over 5 SMTP
```

With the above rules in place we have effectively eliminated the possibility of sending lots of SPAM and e-mail out quickly. What happens if you have a real mail server on your network? Well there are two ways of dealing with this.

Real mail servers may send out quite a few messages very quickly as they are legitimately serving many users. A retail business that I worked with had about 175 users, but they could send out a staggering amount of e-mail in some cases.

Most mail servers though will limit the number of outbound threads, or connections. One hundred seems to be a good number for simultaneous connections on most mail servers. The simplest method of allowing legitimate mail servers processing is to first create an "accepted" mail server list. You create an address-list containing that is for approved mail servers. When you create your standard 5 connection limit SMTP rule, as discussed in the previous paragraphs, you will add an extra option, this is on the advanced tab. The option is the Src-Address-List, where you will specify your accepted mail server's address-list. You will also use the !, or NOT function. This says this rule will only match, if the protocol is TCP, dst-port is 25, they already have 5 connections open, AND they are NOT on the accepted-mail-server's list. If the src-ip is on the list accepted mail server address-list, the rule will never match, so you don't have to worry about them getting on the list.

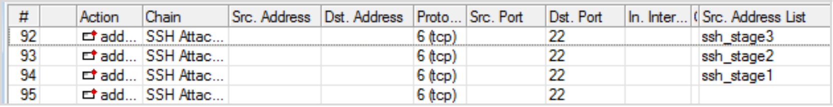


Src. Address List: ! Accepted Mail Servers
Protocol: □ 6 (tcp)
Src. Port:
Dst. Port: □ 25
Connection Limit
Limit: □ 5
Netmask: 32
Action: add src to address list
Address List: Over 5 SMTP
Timeout: 02:00:00

Brute Force Attacks

These attacks send an entire dictionary of words to a SSH, telnet or FTP Server trying to discover the password that will let them in. One of the ways I have found to effectively limit this type of attack very effectively is to create several different address-lists dynamically. These lists, will allow only so many SSH, FTP or Telnet login attempts before it blocks the attacking IP for a set amount of time.

To do this, you will simply create a rule that says "if this is a new connection attempt on one of these ports; add the IP address to a stage 1 list". Normally, most actual users will not go past this, and the stage 1 list only keeps the IP there for maybe a minute or two. With telnet, you will have several attempts to type in the correct username and password, but if you type the wrong one(s) the router will terminate the connection. Now you can create another connection and try again. This is where your stage 2 rule comes in. This rule says, "If your IP is already on the stage 1 list, and you are attempting to make a connection, add the IP to the stage 2 list; this time for five or six minutes". Again, as the user, you will have several attempts to connect with FTP and telnet. The third rule is the big one. "If you are making made a third connection attempts, and you are already on the Stage 2 list; your IP will be added to a stage 3 list". This time however, your IP will be retained on the stage 3 for several hours to days. There is also another rule that says if you are on this stage 3 list, drop all of your traffic preventing you from getting past the Firewall to do any other attempts.



#		Action	Chain	Src. Address	Dst. Address	Proto...	Src. Port	Dst. Port	In. Inter...	(	Src. Address List
92		add...	SSH Attac...			6 (tcp)		22			ssh_stage3
93		add...	SSH Attac...			6 (tcp)		22			ssh_stage2
94		add...	SSH Attac...			6 (tcp)		22			ssh_stage1
95		add...	SSH Attac...			6 (tcp)		22			

This is a set of rules that I usually use. I would jump if it is port 22 for SSH or port 23. You don’t have to actually specify the port here, because you can do this in the jump rule as well. Note that we ordered them backwards; we wanted the first rule to look at the stage 3 list while the second rule looks at the stage 2 list, and the third rule looks at the stage 1 list. So on and so forth. The idea behind this is that we create several opportunities and have several stages, to let someone that may be valid to login. Once they have made an excessive number of attempts in a small time period, we assume that they are attempting to hack the router, and then block them for a long time.

DOS/POD Attacks

There are two other types of attacks that we commonly see. One is a DOS or Denial of Service attack. This attack typically sends thousands, or more, connection requests, or (TCP SYNs) to a single IP address. This IP may be a web server, or some other connection based protocol. Even though all of these connections requests may be made and are valid, the issue is that the server can only handle so many of them. After a while the server will be overloaded with SYN requests and will have so many connections open that the server is overwhelmed and performance slows down.

First off we need to identify these types of connections. We can setup a rule to match TCP connections, with a SYN flag that are in the “new connection” state. This identifies all of these connections that are attempting to be opened up. Next, we place a limit on the number of packets per second we wish to allow. A good number would be 300-400 of these types of packets per second. Once we identify that a single IP has gone over this limit, we can then add the remote IP to an address-list, then we can use another rule to block access.

The second attack that is common is called a POD or Ping of Death attack. This attack uses many computers around the world, to ping you (or an IP behind your router) with large packets. The sheer number of ICMP packets coming in typically consumes all of the available bandwidth. Further, the large number of requests per second consumes CPU power. If you have a small Internet connection, under 50-100 Mbps, a PoD or DoS attack will typically result in slow service, or performance because your bandwidth is consumed and legitimate traffic is slowed before it gets to your RouterOS system. With the POD attack, simply limiting the size of pings can lessen the impact. Secondly, prohibiting or limiting pings totally also will also help.

Firewalling Examples – Using Multiple Rules to do what YOU want!

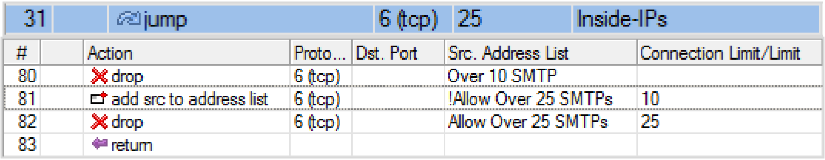
I have said that RouterOS is an infinitely configurable router. But do you know what the only problem with RouterOS is? Simple, it’s an infinitely configurable router! Many system administrators today simply check a box to enable a Firewall, or selecting a link and turn on VoIP QoS. These types of clickable configurations don’t match data effectively, don’t guarantee QoS, and in general are a shotgun approach compared to the precision control that RouterOS features can provide. The challenge is getting the RouterOS system that you have to make it do exactly what you want! Sometimes it’s not as simple as just checking a single check box or enabling a feature. Sometimes you may need to build Firewall rules to do something and then, followed those by three more “if-then” statements. One rule eventually will block that hacker, but you have to make it do it!

Using Pre-Built Firewall Scripts

Be careful of prebuilt Firewall Scripts. Make sure you know exactly what that script does PRIOR to its installation in your network. Why? In some cases, that Firewalling script may do things that may adversely affect your network. Firewall rules that might be fine on the original creators network, may not work properly on your network. Always use caution when implementing Firewall scripts. You should use the mantra of, "If you don't know what it will do, don't use it!"

Multiple SMTP Outbound Limits

In this example, we have created a set of rules that allow us to have two different outbound limits for outbound SMTP traffic. There are actually four steps in this system. The first is a forward chain jump. We jump to a SMTP chain so that we can process further inside a custom chain, providing us organization inside our Firewall rules.



31		jump	6 (tcp)	25	Inside-IPs

#		Action	Proto...	Dst. Port	Src. Address List	Connection Limit/Limit
80		drop	6 (tcp)		Over 10 SMTP	
81		add src to address list	6 (tcp)		!Allow Over 25 SMTPs	10
82		drop	6 (tcp)		Allow Over 25 SMTPs	25
83		return				

This list does a number of things. One, it checks a list called "Allow 25 SMTP", for IP addresses, if they are on this list; they will get 25 SMTP connections. If they are not on this list, it will give them 10 SMTP connections, once they go over 10; they get added to an address-list called "Over 10 SMTP" for 2 hours. Once they are on this list they will have all SMTP blocked for however long they are on the list.

This gives you two levels of SMTP, one default and one that allows up to 25 connections. Of course you can modify that list or the number of connections to suit your preferences.

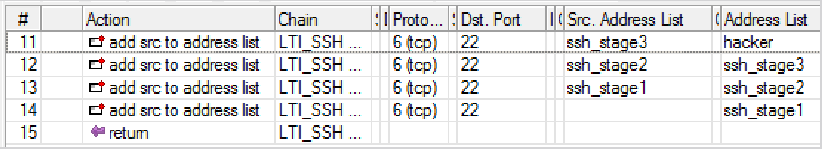
Here is the order of events:

- Conditional Jump from Forward Chain to SMTP Chain
- Conditions
- *Source Address-list:* Inside IP addresses – Lists all inside IP addresses.
- Protocol: TCP
- Dst Port: 25
- *Jump to:* SMTP Chain
- If they are on the "Over 10 SMTP" Address-list, drop all traffic
- If they have over 10 connections, and are NOT on the "Allow 25 SMTP" Address-list, add their IP to the source address-list of "Over 10 SMTP"

- If they are on the "Allow 25 SMTP" address-list, then drop any connections over 25.
- Return to forward Chain

SSH Brute Force Attack Prevention

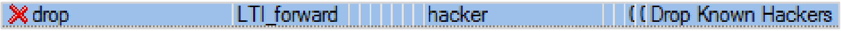
Again, there are a number rules that are needed in this case. Just like the above example, we will have either an input or forward chain rule that jumps TCP Port 23 packets over to our SSH Attack chain.



#		Action	Chain	Proto...	Dst. Port	Src. Address List	Address List
11		add src to address list	LTI_SSH ...	6 (tcp)	22	ssh_stage3	hacker
12		add src to address list	LTI_SSH ...	6 (tcp)	22	ssh_stage2	ssh_stage3
13		add src to address list	LTI_SSH ...	6 (tcp)	22	ssh_stage1	ssh_stage2
14		add src to address list	LTI_SSH ...	6 (tcp)	22		ssh_stage1
15		return	LTI_SSH ...				

In the above example, we show what rules we would add to have a multi-stage SSH attack-prevention Firewall. This could work for FTP as well as Telnet services as well. If we go through the rules in order:

- First new SSH Connection
    a. Is SRC IP on ssh_stage3 address-list? No.
    b. Is SRC IP on ssh_stage2 address-list? No.
    c. Is SRC IP on ssh_stage1 address-list? No.
    d. Add SRC IP to the ssh_stage1 address-list for 1 minute.
- Second SSH connection
    a. Is SRC IP on ssh_stage3 address-list? No.
    b. Is SRC IP on ssh_stage2 address-list? No.
    c. Is SRC IP on ssh_stage1 address-list? YES!
        i. Add SRC IP to the ssh_stage2 address-list for 1 minute.
- Third SSH Connection
    a. Is SRC IP on ssh_stage3 address-list? No.
    b. Is SRC IP on ssh_stage2 address-list? YES!
        i. Add SRC IP to the ssh_stage3 address-list for 1 minute.
- Fourth SSH Connection
    a. Is SRC IP on ssh_stage3 address-list? YES
        i. Add SRC IP to the hacker address-list for 10 days.



drop	LTI_forward	hacker	Drop Known Hackers

There is also a rule that looks for SRC IPs on the hacker address-list. Once found, we drop all traffic to and from that SRC IP address, in our forward and input chains. This rule, blocks those attempted hackers from even getting to our RouterOS or to customers.

Using Mangle

The MikroTik Mangle system is used for several different tasks. Marking of data, by using connection, packet, or routing marks are but one task of the Mangle system. You can also modify some fields in the IP header of TCP/IP packets. These modifications can include TOS and TTL fields. Take a look at the Firewall section for examples to understand on how to match data. Again, the key to your Mangle is matching data. Once you match data you can perform an action on it.

Chains

The Mangle system uses chains, just like your Firewall system. It is important to understand how these chains work together. Depending on how your data flows, you will use different chains. Make sure you look at the packet flow section in the Firewall system.

Pre-routing

This is the most common location for your Mangle rules. Pre-routing will process data as it flows through your router, but more importantly, it processes prior to your routing decision. So you can apply marks prior to the router to determine what route to take. 99% of your Mangle rules will go here.

Post-routing

Post-routing is typically for packets leaving your router that you wish to Mangle. Good usages for your Mangle system here is when you are changing your TCP MSS size, or making other packet changes. Another use possibility is to change the TOS bit of the packet.

Input

The input chain in Mangle is the same as the input chain in the Firewall system. Input rules are for packets that are destined for your router. They input into the router. An example of this would be ICMP ping packets that are pinging your router. Your router receives them in on the input chain, and then responds to them on the output chain.

Forward

The forward chain is the same as the forward chain in the Firewall system. This chain processes on packets that are passing or flowing though the router. The DST or SRC IP is not an IP address on the router, but rather the router passes the packet though the router, in one interface and out another.

Output

The output chain is just like the output on your Firewall rules. This is for packets that are generated by your router, and sent out an interface.

Using Marks

While using your Mangle system, one of the key features is marking. You will typically use the ability to mark data simply to identify it for use in other RouterOS features. There are several different features that RouterOS will uses marks for. Policy based routing, along with traffic management, and queues are just a few. Each of these RouterOS features uses a mark to identify traffic. Since these marks are just used to identify traffic for other RouterOS features, they do not travel outside of your Router. They don’t go between RouterOS systems, nor are you changing the packet or data in anyway. The two marks that you will commonly use are Packet marks and Routing Marks.

Packet Marks

When you are identifying data to either use in Firewall rules, or in the queuing system, you are going to be marking the packets with a packet mark. This is the most common type of mark that you will use. The goal is to identify traffic and then place a mark on that traffic for other RouterOS facilities. To identify traffic, you will use the Firewall-like options in the

Mangle to match data. Once the data is matched you will have an action type to mark your packets. This places a virtual mark, only inside the RouterOS system, that you can use in your queuing system, as well as your Firewalling system.

Routing Marks

Routing marks allow you to mark data in a way to apply routing policies and rules to the data. The data that you match can have a routing and packet mark at the same time. Just like packet marking, you will match your data, and then specify a routing mark to give to that packet. This routing mark has only one real purpose. This is to allow you to identify traffic in the routing rules section of RouterOS. Identifying traffic allows you to apply different routing tables to different types this type of traffic.

Unlike the actual routing rules section, in Mangle, you can apply routing marks to packets. This means you can use all of the advanced features of your Mangle to match data. An example of this is to send non-latency sensitive data out a connection that has higher latency. HTTP traffic could be sent out a secondary connection to off-load traffic from the primary low-latency connection. You could then apply another routing mark for more traffic, say SMTP or mail traffic, and send it out a connection that is just for mail traffic. The routing rules only give you the options to match based on source or destination IP addresses. You also can specify routing marks creating in the Mangle system in your routing rules. This allows you to tell your routing system to use a routing mark that was created in your Mangle system.

Connection Marks

Connection Marks are used to increase the processing capacities of your RouterOS. It’s important to understand how connections are created and how connection states in RouterOS are handled. Refer to the connection state section if you need more information.

Using connection marks, is very simple, you need to match the data, preferably when its connection state is still new, by using a Mangle rule. The Mangle rule then places a connection mark on that connection. All other packets that come from that new connection will also have a connection mark on them. This then allows you to place a routing or packet mark on the packets that have a connection mark on them. Processing all of the packets based on the mark is faster than matching the data every time. The connection mark allows you to do complicated or high capacity matching without the high CPU overhead of processing and analyzing every packets header information. It's just simply faster to process based on the connection mark.

So the question that has to be asked is; when do you use connection marks? Typically I have found unless you are really starting to push the RouterOS system and hardware that you have, I find it simpler to mark with packet or routing marks directly, instead of indirectly with a connection mark. If you are having CPU issues with the RouterOS device, either you are pushing too much data or you have lots of rules, you can start to use connection marks to improve the performance of your router and CPU. If you are starting to drive your RouterOS system to this level, then you should think about replacing the hardware with something a bit faster.

Change TOS Bit / DSCP

Using this option, you can change the TOS/DSCP bit of a packet. This is very useful for identifying traffic in your Mangle system. Unlike packet and routing marks, TOS bit changes travel with the packet as it leaves your router. This allows you to identify traffic on some routers and put a bit number on it for further identification across your network or on other networks. I will typically use a TOS bit change this to match data, then I match and prioritize the data across backbone routers, by using only using the TOS bits instead of matching the data again by some other method. This is one of the few actions that work well on your post-routing chain.

Change MSS

This allows you to change your MSS or Maximum Segment Size field of your IP header. MSS is the largest amount of data that a device can handle in a single unfragmented piece. The number of bytes in the MSS plus the header information must not add up to be more than above the number of bytes in the MTU or Maximum Transmission Unit.

Typical usage for changing the TCP MSS is to set a packet size on an outgoing interface, which your data will leave on. An example of this is a PPPoE-Client connection. You will typically have a 1500 byte packet size with Ethernet, but when you add the header information that PPPoE has to have, you end up with a 1460 maximum packet size. By changing your MSS, you are specifying those packets that are going through your routing and leaving on said interface need to be fragmented to the MSS size.

Of course you can use your post-routing chain for changing your MSS, however, for optimized processing; you can also specify the packet size in the advanced tab of your Mangle rule, specifying packets that are oversized for your interface. So if you are changing your MSS to 1460, instead of processing on all of the packets that are larger than that, you can specify 1461-1500 packet size. This way it will only change the MSS on packets that need to be fragmented.

Clear DF

This simply clears the DF, or do-not-fragment, bit of the packet. This bit if set to 1, specifies do not fragment. This basically says that this packet should not be fragmented. By using this option, you are clearing this bit and resetting it to 0, showing that the packet can be fragmented.

Set Priority

This sets a new-priority parameter on the packet that is sent out through a link that can carry the priority. This is just for VLAN and WMM-Enabled Wireless interfaces.

Strip IPv4 Options

This does exactly what it says; it strips the IPv4 Option fields from IP packets. These may include any of the following options: *loose-source-routing, no-record-router, no-router-alert, no-source-routing, no-timestamp, router-alert, strict-source-routing, timestamp*.

NAT – Network Address Translation

Network Address Translation or NAT, is a very useful feature in many routers. Unlike many other routers, RouterOS offers a full featured NAT system. Too many times, consumer routers offer a NAT feature, but it actually is a small feature set of the NAT system. Usually, this feature is Masquerading, or many to one translation. RouterOS will allow you to do both inbound and outbound NAT as well as redirection and other functions. We will cover the basic usages of your NAT system in this chapter.

NAT has two main sides, and inside and outside network. You can perform NAT on many different IPs, and RouterOS does not restrict you to private versus public routable IP addresses. The inside IPs are typically being NATed by a many-to-one rule, called a masquerade rule. Incoming connections, unless they were called by an inside request, are typically dropped. However, you can perform inbound NATing or dst-nat with RouterOS as well. This will take the public IPs that you have, and translate them to an inside IP address. In most cases, the IP addresses on the inside will be private IP addresses, while the outside will contain public IP addresses.

To access the NAT system in RouterOS, you will click on IP --> Firewall --> NAT tab. Use the chart below to remember what chain to use:

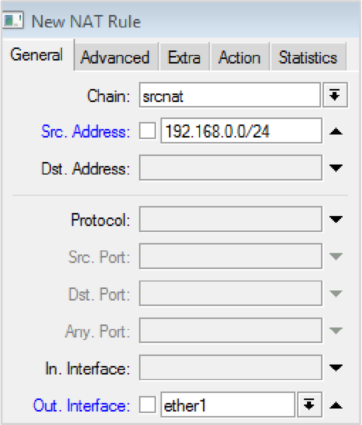
SRC-NAT	***Outbound Traffic*** **Many Privates to Single Public** **Private IP to Public IP**
DST-NAT	*Inbound Traffic* Public IP to Private IP

Chains

The NAT system has two built in chains. Just like the Firewall chains, NAT rules must belong to a chain of some type. All NAT rules will start with either a src-nat or a dst-nat chain. Src-nat rules are rules that perform actions that come from the NATed network. As the data passes through the router, the source IP address is replaced by the new IP address on the outside of the NAT system. Dst-nat rules are data that come from the public side of your network and are translated to the private side; think inbound routing.

Masquerading

This feature is misnamed quite a bit. Lots of routers, especially the lower cost home routers and other types of CPEs will label this feature as NAT. Masquerading is a many-to-one network address translation system. This allows many IP addresses to be translated into a single IP address. The most common usage is to translate many private IP addresses, such as a 192.168.0.0/24 subnet, into the single IP address that you received from your Internet provider.



Chain: srcnat
Src. Address: □ 192.168.0.0/24
Dst. Address:
Protocol:
Src. Port:
Dst. Port:
Any. Port:
In. Interface:
Out. Interface: □ ether1

To implement a masquerading system, you will need to define at least one of two options. What is the source IP subnet, or the outbound interface? You must have one of those to perform the action of masquerade. When you do this, all data going either out that interface or from the defined source addresses will be translated to the IP address on that interface.

One of the little details that you need to understand is that when you do masquerading, the outgoing IP address on the out interface that is used for translation is the first IP that was added to that outgoing interface. Also, there is no reason you cannot masquerade public IPs, or virtually any IP address you wish with RouterOS. Many routers will only let you masquerade private addresses. I do this when a networks primary Internet connection with public IPs goes down and all we have is a DSL or cable connection. We masquerade customers with those public IPs out the DSL or cable interface letting them get on-line, but not use their public IPs.

To create a basic masquerade rule, see the above graphic, click on IP → Firewall → NAT, and create a new rule. Enter either a source address or an out interface, preferably both, and then click on Action, and use the action drop down box to select masquerade.

Setup basic Masquerading:

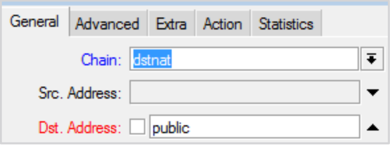
- IP → Firewall → NAT
- New Rule

- Add either Source Address Subnet or Out-Interface
- You can specify both if you wish
- Select Action Tab
- Use the action drop-down to select masquerade as the action.

PPPoE-Client and other types of Tunnels and Masquerading

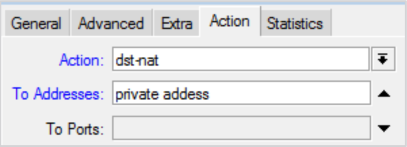
When you create a PPPoE-Client, you may get a public IP once the connection comes up. This is just like any other interface with RouterOS. If you have private IPs that you wish to masquerade out a PPPoE-Client connection, your outbound interface will have to be the PPPoE-Client, as that is the actual interface that you are sending data out. This remains true when you have other types of tunnels, or wireless interfaces as well.

Inbound NAT



Chain: dstnat
Src. Address:
Dst. Address: □ public

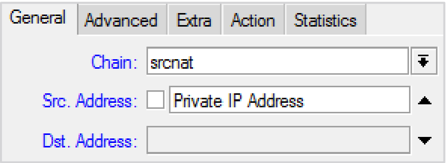
Inbound NAT or dst-nat commonly is used to take a public IP address, and forward it into an internal private IP address. You do not have to forward an entire IP address and all protocols and ports though. You can simply have several rules to only forward specific protocols and ports. This typically will be used in conjunction with your outbound NAT rule as well. The reason for this is that data that comes in through the inbound NAT system will typically need to reply on the same IP address that the request was sent too.



Action: dst-nat
To Addresses: private address
To Ports:

To create your basic NAT, think of data that is coming in via a public IP address. We need to get this data to a private IP address. This private could be a customer IP, or a server. In this case, you will create a DST-NAT rule, which will use the destination IP address of the public IP. If you don't wish to specify a specific port or protocol, you will then need to specify what action you wish the rule to perform. You will select dst-nat as your action, to perform destination NAT, and then the to-address field will be the private IP that you will be forwarding that public IP to.

Outbound NAT

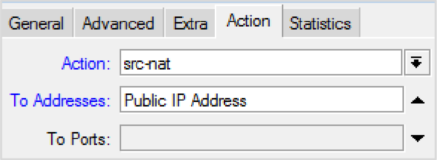


Chain: srcnat

Src. Address: □ Private IP Address

Dst. Address:

Outbound NAT uses your source IP address to translate to a specific public IP address. Instead of looking from the outside in, you will be looking at this rule from the inside private address going out, hence outbound NAT. RouterOS calls this src-nat, or source NAT. This feature is very basic. It says, if data comes from *xyz* private IP address, then perform src-nat on it, and translate it to *abc* public IP and send that data out to the Internet. This will allow a private IP address to show up as a very specific public IP address. You will have to have this public IP address on your public interface of your RouterOS system.



Action: src-nat

To Addresses: Public IP Address

To Ports:

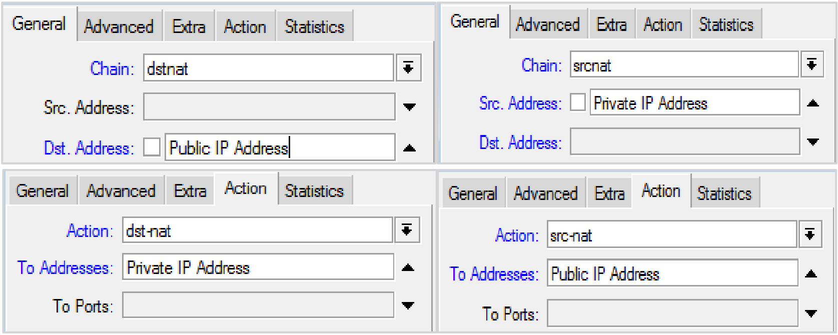
To create this rule, simply specify the src-nat chain, then you will specify the source address of the Private IP address on the inside of your network. Use the action tab, perform the action of src-nat, and the To Addresses field will be the Public IP address you wish that private to show up as.

One way to check this feature is to have the private IP of a computer or router, and browse to a website that checks the public IP address that you are coming from and displays it. Whatismyip.com is one of these, as well as another called IPChicken.com. This will display the IP address that you are coming from. If you are simply doing your masquerade, it will be the first IP address on your outgoing interface. Once you specify this src-nat rule, you should be able to reload that webpage and see the IP address that you put on the to-address field in the action tab. This will make the computer, or device, that is on the private address appear to come from its own public IP address. You can only have one src-nat rule per public IP address, as this is a 1:1 relationship.

Performing a One-to-One NAT – Assigning a Public IP to a Private

Doing a 1:1 NAT, allows you to assign a public IP address to a private address on the inside of your network. In some routers, the functionally of this is limited, however, with RouterOS, this function works perfectly. To do this, you will have to create two different rules. One is an outbound NAT. This takes all the traffic that the private IP address generates and sends it out an individual public IP address. No other traffic will be generated from this public IP without it coming from the private IP.

The second rule is the inbound NAT rule. This sends all packets that are destined to the public IP address and forwards them into the private IP. We do not define any ports or protocols, so that all data is passed through. The only thing that is changed is the source and destination IP addresses to allow forwarding through the private network.



Chain: dstnat
Src. Address:
Dst. Address: □ Public IP Address

Chain: srcnat
Src. Address: □ Private IP Address
Dst. Address:

Action: dst-nat
To Addresses: Private IP Address
To Ports:

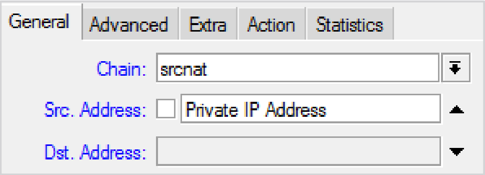
Action: src-nat
To Addresses: Public IP Address
To Ports:

Above, I have provided screenshots of the rules that are required. Remember, your dst-nat is inbound, so your DST-Address field will be your public, and that dst-nats to your private. Your src-nat is outbound so your src address is your private IP address and you are translating it to a public IP address. With this method, you have sent a public IP to a private IP on a 1:1 basis. This will forward in all protocols and all ports to the private IP address.

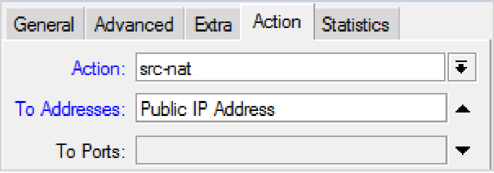
Selective Port Forwarding

When you do port forwarding from a public IP to a private, you typically do a 1:1 NAT like the above section. However, you don't necessarily have to forward in all of the protocols and ports. You can selectively send these in as needed. Keep in mind that this will require more rules than a standard 1:1 NAT.

First, you will still need to have the src-nat rule to send data from your private IP out the public. So create your src-nat rules as the outbound NAT section describes. You need to do this because when a request comes in to your public IP, we need to have the server reply from the public IP address. So you will need to create this rule.

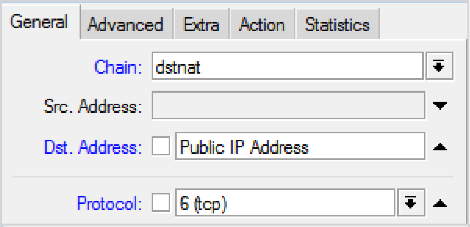


Chain: srcnat
Src. Address: □ Private IP Address
Dst. Address:

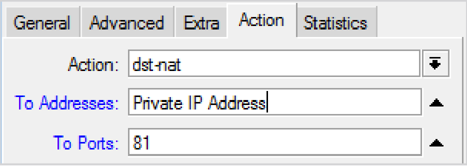


Action: src-nat
To Addresses: Public IP Address
To Ports:

Once that is done, now you need to create your inbound NAT rule. This is done with the same information as the inbound NAT system described in the prior sections. You will make a change though, now you will select what protocol and/or ports that you will need to send in. So the example is if you are sending in web or HTTP traffic to a web server, you will need to select the protocol of TCP and a destination port of 80. On the action tab, you typically will not need to specify a port, as you are receiving on port 80 so it will forward to port 80.



Chain: dstnat
Src. Address:
Dst. Address: □ Public IP Address
Protocol: □ 6 (tcp)



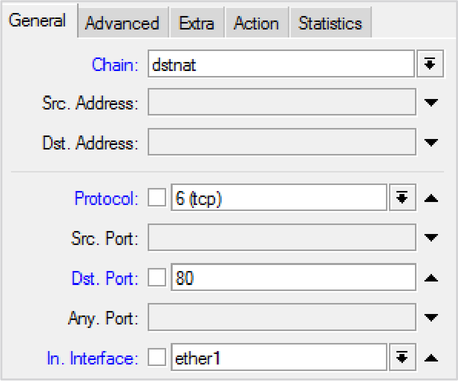
Action: dst-nat
To Addresses: Private IP Address
To Ports: 81

You can also change the port during the translation on the inside or private network as well. Once you specify the To Addresses or the private IP where your server is, you also have the ability to change the To Ports. If you wished, you can run your web server on port 81 on the private IPs, but port 80 on the public IP will be translated to port 81 on the inside.

For every other protocol and port that you wish to send to your private IP, you will need another rule. If you wished to send TCP port 25 into that same private IP, you will need to create another rule that uses DST port 25 protocol TCP to send inside to that private IP. Any protocols and ports that you do not forward in with a dst-nat rule will end up hitting your router. I would suggest putting a deny rule on the input chain for this IP address so that your router does not get any requests. I say this, because if you do not forward in port 23, the telnet port, then you could have users hitting this public IP that is normally assigned to a server with a private IP, but they actually get the router login prompt since it is not dst-nat to your private IP address.

Inbound NAT with DHCP Public IP Address

Sometimes you will only have a single public IP address and you obtain that IP via a dynamic method, such as DHCP or PPPoE. When you get this public IP, you do not necessarily know what the IP address is going to be, therefore, how can you do your dst-nat rules for IP addresses you don't know? So how do you do this? Well, since we can't use a destination address and we typically will only have one IP address on the interface, we will specify the in interface instead.



Chain: dstnat

Src. Address:

Dst. Address:

Protocol: □ 6 (tcp)

Src. Port:

Dst. Port: □ 80

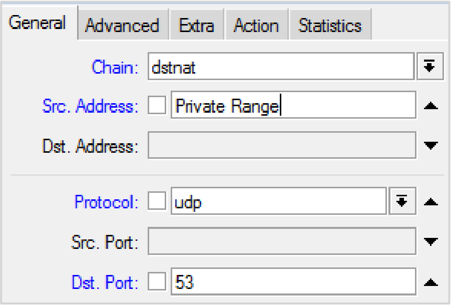
Any. Port:

In. Interface: □ ether1

In the example to the right, you will see that we are sending in TCP port 80 via a dst-nat rule. Since we don't know the IP address to put in, we can use the interface that we know the information is going to come in on. This interface may be an Ethernet port that we have DHCP Client turned on, or it could be a PPPoE-Client Interface.

Redirect

Redirect is the same thing as a dst-nat action, but it does not need a to-address field. It always redirects to the incoming interface IP on the router. What this is typically used for is redirecting traffic to router facilities and features. Two really good functions of this feature is for redirecting web traffic to your web proxy system, and/or redirecting DNS requests to the local DNS caching system on RouterOS.



Chain: dstnat

Src. Address: □ Private Range

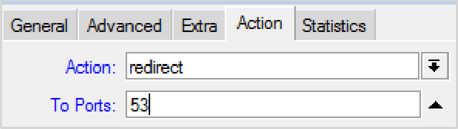
Dst. Address:

Protocol: □ udp

Src. Port:

Dst. Port: □ 53

An example of the transparent redirect for DNS would be a rule that would match against DNS traffic; so UDP port 53. I typically would add either a source address of your private range, or if you have multiple local address ranges, you can create an address-list with these and match on that. Then your action would be to redirect to the local port 53. This will send DNS traffic from your local subnet to your caching system on your RouterOS system. Be sure to enable remote requests in your DNS system!

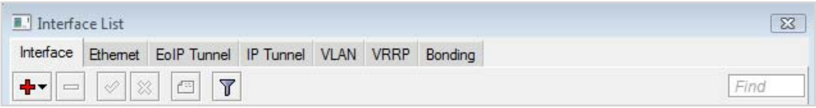


Action: redirect

To Ports: 53

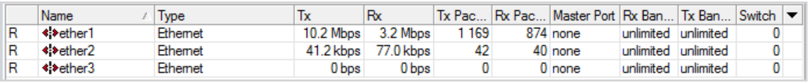
Basic Interfaces

The interface sections will allow you access to all RouterOS interfaces. You will be able to configure not only Ethernet, but also Wireless Interfaces, Tunnel Interfaces, VRRP, Bonding and VLAN interfaces within the interface settings. We will cover Tunnels such as EoIP and IP tunnels in the tunneling section.



Ethernet

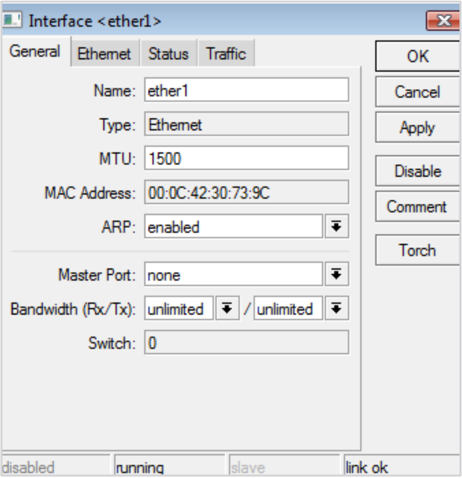
Inside the Ethernet interface settings, you will see a listing of just hard Ethernet interfaces. This will not include other interfaces associated with Ethernet interfaces, such as PPPoE or VLANs, but just actual, physical Ethernet interfaces.



	Name	Type	Tx	Rx	Tx Pac...	Rx Pac...	Master Port	Rx Ban...	Tx Ban...	Switch
R	ether1	Ethernet	10.2 Mbps	3.2 Mbps	1 169	874	none	unlimited	unlimited	0
R	ether2	Ethernet	41.2 kbps	77.0 kbps	42	40	none	unlimited	unlimited	0
R	ether3	Ethernet	0 bps	0 bps	0	0	none	unlimited	unlimited	0

Inside this section you will see the type of Ethernet interface, the current TX and RX data rates, as well as the TX and RX number of packets. Depending on your version and hardware, you may have some other options. On the RouterBOARD 400 Series, you may have options such as switch port, master port and TX/RX Bandwidth limits.

Upon double-clicking the Ethernet interface, you will be presented with the actual interface configuration options. These are pictured to the left.



Name: ether1
Type: Ethernet
MTU: 1500
MAC Address: 00:0C:42:30:73:9C
ARP: enabled
Master Port: none
Bandwidth (Rx/Tx): unlimited / unlimited
Switch: 0

disabled running slave link ok

You will be able to configure what the name of your interface is, however on Ethernet interfaces, I would also recommend leaving the Ethernet 1, 2 names, and then adding more description to the name, instead of renaming the entire interface. The reason I recommend this is simple. Most Ethernet interfaces are numbered in some way, if you rename them without the numbers, then later, it may be difficult to find out what interface belongs to what name. I have seen this a number of times, and typically it's simpler not to rename the entire interface.

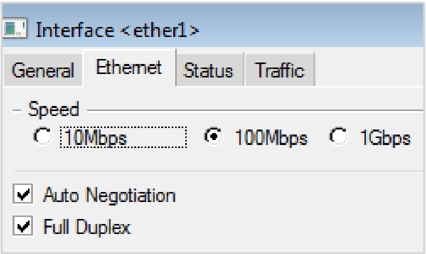
You do have some other options on naming interfaces; you can also add a comment by clicking on the comment button. This will let you put in information and comments for that interface. Comments such as, "cable goes to the third floor," or other type of descriptive comments can help you identify an individual port instead of renaming the entire interface. The name of the interface will also be used in other places in RouterOS, so I would not put in a long interface name as well.

RouterOS also allows you to change your MTU, or (Maximum Transmission Unit) packet size. For Ethernet interfaces you will typically leave these alone, however, if you have a long distance fiber link etc., you can change this to include super frames, and allow for more throughput if necessary. This is very uncommon though.

Switch Controls

On most Ethernet interfaces you will not have options to control bandwidth or options for master and switch ports. These are typically on the RouterBOARD 100 and 400 series devices. This gives you options to setup Ethernet ports into a hardware-based switch's. By selecting what master port you are using and the switch number, you can place several interfaces in a hardware switch. There are a few benefits with using this method of switching. The first is that it will give you faster throughput, typically close to the 100Mbps the Ethernet port is capable of processing, but you will also lower your CPU load as well. This is due to you using the onboard chip instead of using the CPU and software bridging.

Ethernet Speed and Negotiation / MDI-X



Speed

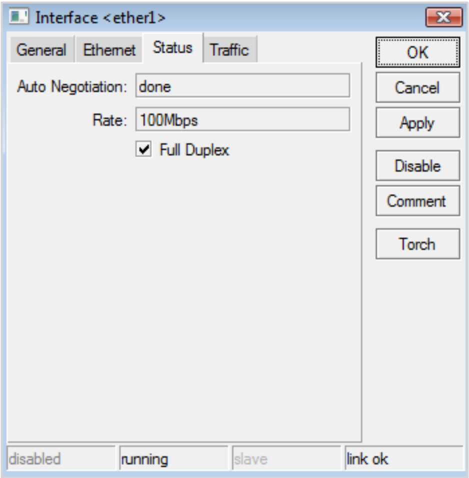
○ 10Mbps ● 100Mbps ○ 1Gbps

☑ Auto Negotiation

☑ Full Duplex

RouterOS supports full Auto-Negotiation on its Ethernet Interfaces. Most network and computer related devices will support this as well. Auto-Negotiation allows the Ethernet ports to negotiate how fast they can communicate. The options are 10Mps, 100Mps or 1Gbps. RouterOS also now supports 10GigE interfaces as well if you have the hardware to run it.

Below you will see the Ethernet tab of an Ethernet interface. Here you can manually select the Ethernet speed as well as enable or disable either Auto - Negotiation or Full-duplex Operation. Full-duplex operation means that the interface can both send and receive data at the same time. This is the default for most Ethernet connections. If you have a device that only runs at 10Mbps Half-Duplex, then you should manually configure your interface, by checking the 10Mbps option, and unchecking both Auto Negotiation as well as Full-duplex. This will tell RouterOS to ONLY run the selected Ethernet port in the 10Mbps Half-Duplex mode only.



Auto Negotiation: done

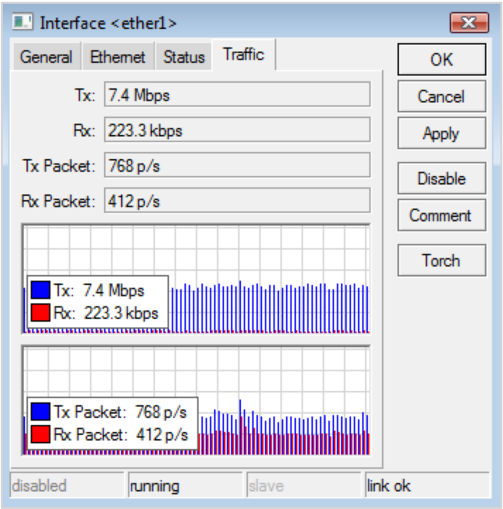
Rate: 100Mbps

☑ Full Duplex

disabled | running | slave | link ok

In the Ethernet status window, you will see the current status of your Ethernet port. In this example, we have performed auto-negotiation. We have a link rate of 100Mbps, and are running Full-duplex.

Note at the bottom of the window, we have a few other indicators. If the interface was disabled, the disabled text would be black instead of grey. We show that the interface is running and it's not a slave to another interface. This is typically used in bonding applications. You also will see that it shows that the Link is OK, showing that the Ethernet interface has a link indicator.

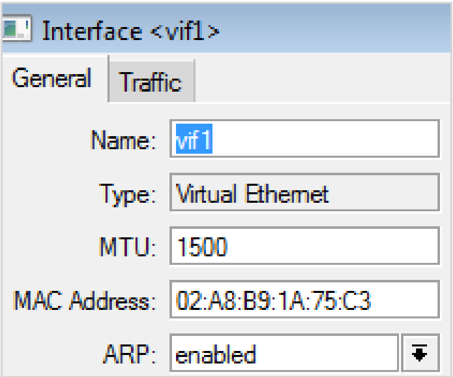


Tx: 7.4 Mbps
Rx: 223.3 kbps
Tx Packet: 768 p/s
Rx Packet: 412 p/s

In the Traffic tab, you have one of the best features of RouterOS. This is the real-time traffic graphing. In this case, we are looking at an Ethernet interface, and its current traffic. We have TX/RX data rates, both in number and graph form, as well as TX/RX packets per second, again, in both graph and text form.

The standard buttons on the right side of your interface are very common to many types of interfaces. You can disable the interface, as well as comment on the interface, and use Torch. We will cover Torch in the Tools section.

Virtual Ethernet Interfaces



Interface <vif1>

Name: vif1

Type: Virtual Ethernet

MTU: 1500

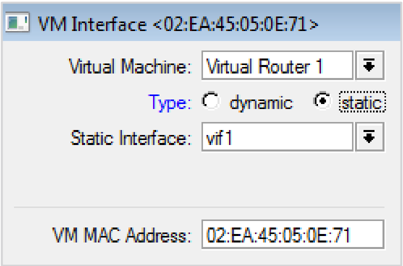
MAC Address: 02:A8:B9:1A:75:C3

ARP: enabled

Virtual Ethernet interfaces are used in conjunction with virtual routers. An example of usage for these interfaces is to interconnect a virtual router to an interface on the actual RouterOS. Think of it as an Ethernet cable between the virtual router and the physical router, just, well, virtual!

To create a virtual Ethernet interface, click on interfaces → Plus → Use the drop down to select Virtual Ethernet.

Once you create the interface, you will just need to name it. Once named, it will become active and now you can configure your virtual router to use it. Using MetaRouters, you will create your MetaRouters using the normal procedure. Once created, then you can assign interfaces. Click on the MetaRouter → Interfaces tab. This will allows you to assign interfaces to your virtual routers.



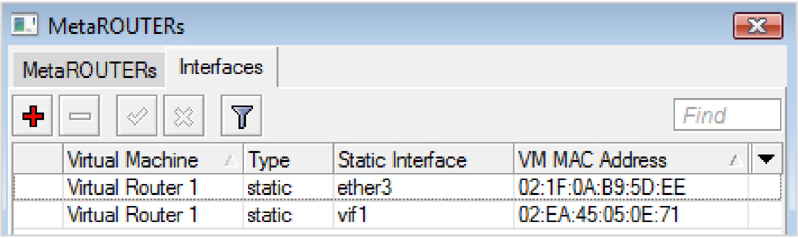
VM Interface <02:EA:45:05:0E:71>

Virtual Machine: Virtual Router 1

Type: ○ dynamic ● static

Static Interface: vif1

VM MAC Address: 02:EA:45:05:0E:71



Virtual Machine	Type	Static Interface	VM MAC Address
Virtual Router 1	static	ether3	02:1F:0A:B9:5D:EE
Virtual Router 1	static	vif1	02:EA:45:05:0E:71

As you can see, we have assigned Ethernet 3, the physical interface of our main router, as a static interface to our Virtual Router. We also assigned vif1 to this same router. This will make our virtual router have two interfaces. One that is connected to the Ethernet 3 interface of our physical router, and the second, is our Virtual Ethernet Interface interconnecting both our physical router and our virtual router.

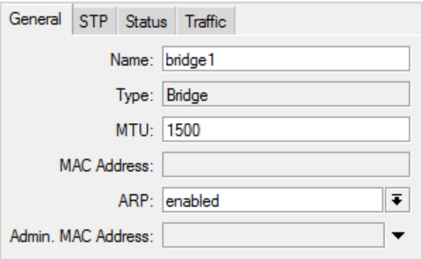
This interface, just like every other one in RouterOS, can be masqueraded, Firewalled, and otherwise controlled just like if it was a real interface.

Bridge Interfaces

MikroTik fully supports bridging of many types of interfaces. You can bridge Ethernet ports together, making them function as a software switch. You can also easily bridge an Ethernet port and a wireless radio in Access Point mode as well. There are also a number of tunnels that support bridging.

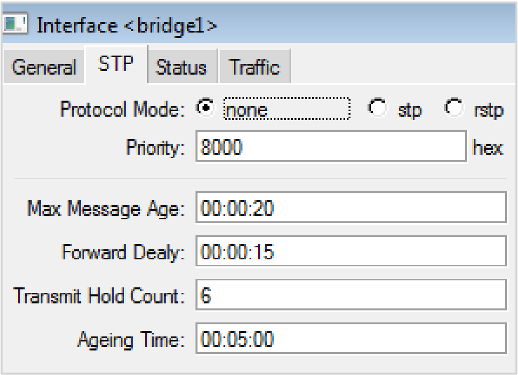
There are some reasons for bridging; one would be to control data flowing through a network. You can bridge two Ethernet interfaces and then control, block, and manage traffic as it goes through your RouterOS device. You can also bridge VLAN traffic as well.

Creating your bridge starts by creating a bridge interface. This interface is now the single interface that your traffic will flow through. To create this interface, select Bridge → Add Interface → Configure Bridge Options.



Name: bridge1
Type: Bridge
MTU: 1500
MAC Address:
ARP: enabled
Admin. MAC Address:

In most cases, a simple bridge will do. No options are necessary other than maybe changing the name of your bridge. The first tab inside your bridge settings will allow you to set the name of your bridge interface, as well as the MAC and Admin MAC address if you wish. You can also setup your ARP information as well here as well.



Protocol Mode: ● none ○ stp ○ rstp
Priority: 8000 hex
Max Message Age: 00:00:20
Forward Dealy: 00:00:15
Transmit Hold Count: 6
Ageing Time: 00:05:00

The STP section of your bridge is for Spanning Tree -Protocol. STP is designed to prevent bridging loops. These occur when you have several different paths that a Layer 2 frame can pass through. Think of it as two switches plugged together. STP gives your data a single path, like a single cable, between the two switches. Now, add another cable. What often occurs is that packets enter on one switch, go out through the first cable, go through the second switch and then go back out the second cable, back to the first switch. That same packet then goes back out the first cable, so on and so forth. The packet keeps going around and around endlessly. Eventually this will use up all bandwidth and CPU power in the switches, causing the network to basically come to a complete halt, or at best, becomes so slow that the network is not usable. This is also called a network loop.

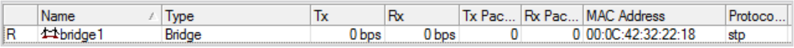
STP of course, is designed to prevent this. RouterOS supports two different versions of STP, the standard STP and RSTP. RSTP stands for Rapid Spanning Tree Protocol. In most cases the default settings for STP and RSTP is fine. The main thing to note is that when you enable STP and turn on an interface, you have to understand that there is a forward delay. This delay, defaulted to 15 seconds, basically enables the interface, but does not allow any transmission. It waits during the forward delay and listens to see if enabling that port would cause a loop. If enabling the port would create a bridge loop, then (R)STP will leave that port disabled. This prevents the bridging loop.

RSTP does the same thing, but it does not wait, it listens and looks to prevent a loop quickly before it becomes an issue. Also topology changes happen within seconds or less instead of in 30-50 seconds that STP requires for a change. Also, RSTP maintains backup details regarding the discarding status of ports. This will help avoid the timeouts if the current forwarding ports were to fail.

A few things to note; if the port is in forwarding status; data is flowing across that port. Disabled status means that the port has been disabled due to loop detection. Listening means that it is trying to figure out if it can bring that port to a forwarding status without creating a loop. Backup ports mean that the port is disabled but considered a backup if necessary.   The last mode is designated port; this is also a forwarding port.

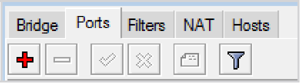
Bridge Ports

Once you create your bridge interface, now you will need to add ports to it. Below is a bridge that is running STP.

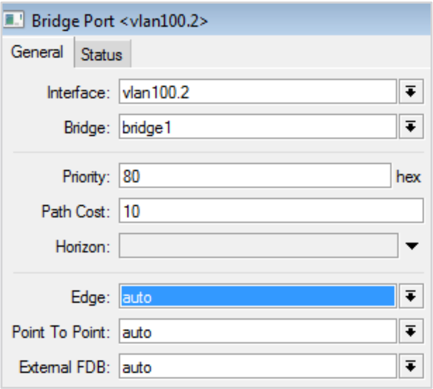


	Name	Type	Tx	Rx	Tx Pac...	Rx Pac...	MAC Address	Protoco...
R	bridge1	Bridge	0 bps	0 bps	0	0	00:0C:42:32:22:18	stp

The Ports tab is where you will add new ports to your bridge interface.



ONCE YOU ADD INTERFACES TO BRIDGE GROUPS, FEATURES SUCH AS HOTSPOT AND DHCP WILL NO LONGER FUNCTION. YOU MUST ASSIGN THESE FEATURES TO THE BRIDGE INTERFACE VS A PORT MEMBER.

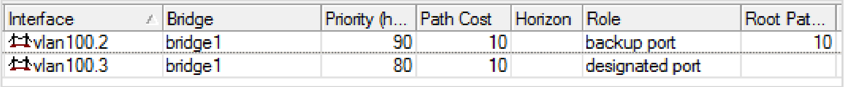


Interface: vlan100.2
Bridge: bridge1
Priority: 80 hex
Path Cost: 10
Horizon:
Edge: auto
Point To Point: auto
External FDB: auto

By clicking the plus tab you can add objects to your ports. You will need to select what interface you wish to add to your bridge group and what bridge group you wish to add that port to. Typically the options that are included are perfectly fine.

One instance when you would wish to change these bridge port options is if you wished to prefer one link over the other. An example of this is if you have a high capacity fiber or wireless link, say over 100 Mbps, and right alongside of it, you have a low cost 30+ Mbps wireless link. Of course you will wish to use the higher capacity link normally, but with Spanning Tree, it does not detect which link is faster. However, we do have options inside our interface that allows us to prefer a link. You would have to configure this on both sides of your link as well.

The simplest thing to do with this is just to increase the priority of the interfaces that are on the slower link. The default is a priority of 80. An increase to 90 will make the primary link, if working, to be preferred.



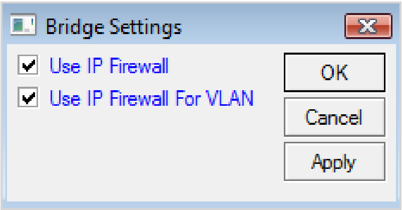
Interface	Bridge	Priority (h...	Path Cost	Horizon	Role	Root Pat...
vlan100.2	bridge1	90	10		backup port	10
vlan100.3	bridge1	80	10		designated port	

Note in the above example, we changed the priority to 90 on our vlan100.2 interface. Since both are running, we have preferred our vlan100.3 interface.

We have an example of setting up an Active/Failover Backup link with priority in the Quick Reference Guide.

Bridge Settings / Using IP Firewall

In RouterOS version 2.9.x, by default, the system will pass data through the bridge and through your IP-based Firewall. This IP Firewall is located under IP → Firewall → Filters. However, in Version 3.x MikroTik added a bridge setting feature. This feature is designed to eliminate the processing time and CPU needed to process the bridge interfaces if you don't need a bridge Firewall. However, quite a few MikroTik users wish to use the IP-based Firewall filters and rules on their bridged traffic. This new feature, allows you to process bridge packets on your IP Firewall. If you run VLANs as well, you will have options for running your IP Firewall for bridged VLANs.



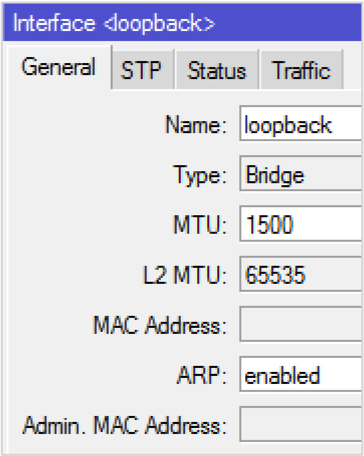
Bridge Settings

☑ Use IP Firewall

☑ Use IP Firewall For VLAN

Bridge Loopbacks

In other sections, we have recommended you find the loopback section. Here we will describe in detail why it's important to use loopback addresses in detail. A loopback is simply a bridge group with no ports, which is always running, and is part of your OSPF or other dynamic routing protocol network. This interface typically has a single /32 address assigned to it.



Interface <loopback>

Name: loopback

Type: Bridge

MTU: 1500

L2 MTU: 65535

MAC Address:

ARP: enabled

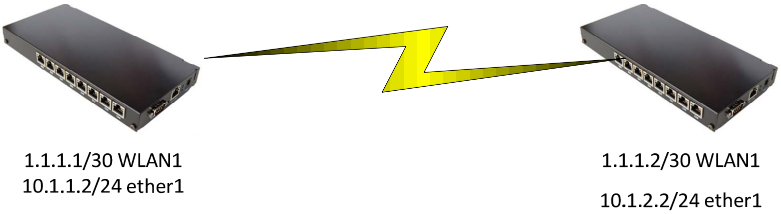
Admin. MAC Address:

Management

There are a number of usages for loopback addressing. The first that we will discuss is for management of your network. For example, we can use a subnet; say 10.255.255.0/24, as a routed loopback management network. All communications with your routers, such as WinBox, etc., would come to and from the management subnet. This allows you to build Firewall rules protecting your routers from access from other subnets that customers may be one, or other devices. Your customers will never see this management subnet IP addresses in a traceroute but the management addresses will be there for your use only. Having these IP addresses on each one of your routers will assist you in securing your network. Since you only will have only a single IP for management, you now can have only a single subnet to remember, 10.255.255.x instead of all of the subnets out on your network. You will know that .45 is your core router at POP 4 and .55 is your core router at POP 5.

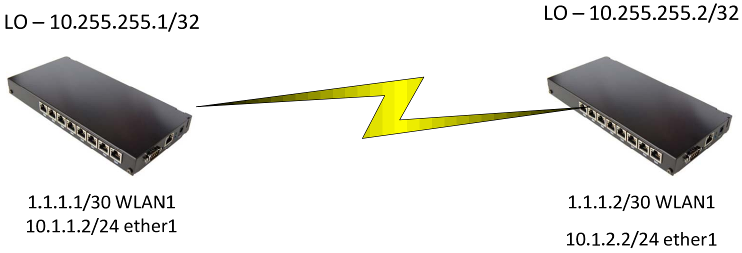
OSPF Loopback

During a network outage, IP addresses from one side of a link may not be available anymore. For instance, the near side of the primary link has a /30 on it, the first IP, let's use .1, is your router on the near side of the link. The .2 is on the far side of the link but there is a wireless issue preventing communication. This causes a subnet, in our case the 1.1.1.0/30 subnet to become split. In OSPF, there will be two routers advertising that they are connected to the 1.1.1.0/30 subnet, but those two routers will have "different" parts of the subnet available, therefore we have an issue.



In this example, we have a break in the network. A single subnet is split. If we were coming from a subnet on the left side of our network diagram, we would only be able to see the 1.1.1.1 address, not the 1.1.1.2 address. The reason for this is that once we reach the 10.1.1.2 router, it thinks it's directly connected to the network. When we send a request to 1.1.1.2, it ends up going out the 1.1.1.1 router as a ARP, and since the link is broken, that ARP never gets a reply. In the end, we can't connect to the 1.1.1.2 IP to manage it. In our example we assume that we have disabled the running check, in other words, the wireless interface does not think that it is down, but the wireless connection is down. This happens frequently when using other radios, such as Ubiquiti or Motorola, the wireless link actually goes down but regardless of the running check, the Ethernet interface is still running. This creates said condition.

The other condition that could occur is that both wireless interfaces show down, and that entire subnet is removed from the routing table. Again, we cannot manage the 1.1.1.2 IP address.



Using the example above, we have added Loopback addresses. To create this, we simply create a bridge with no ports associated with it, and then add the /32 IP addresses accordingly to this interface. Now, using our first example, we would be using the 10.255.255.2 IP to manage our router, not the 1.1.1.2. Since that interface never goes down, it's always in our routing table. Since we are using a single IP in a /32 subnet on an interface that never goes down, that IP is always available to us, regardless of the routing changes. If we did not use management loopback addresses, if we were using the IP address of our primary backhaul to manage our router, and that backhaul went down, then that IP address that we was using would become unavailable. This means, we cannot manage our router. The management loopback address not being tied to an interface that can go up or down, will be accessible, as the dynamic routing protocol has rerouted access to that via a backup link. Note that if you don't have another path to that, then of course you will not be able to get to the management IP.

Loopback SRC-Address

In other sections we have suggested to looking at the loopback section. One example we will use is our DHCP-Relay system. When the DHCP-Relay system sends a relay request, the default action is to send it from the IP address closest to the destination IP. If routing changes, so too could this source IP address. Seeing that the DHCP-Server has a DHCP-Relay IP address entered into it, if that IP changed, then this would break your DHCP-Relay system, as it would be sending from an IP address that the DHCP-Server does not have in its relay system, breaking your DHCP-Relay system, and making you scratch your head!

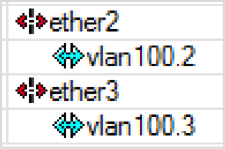
In these cases, it would be a good idea to create your loopback address and specify that loopback address as the Src-Address for your DHCP-Relay system. This prevents routing changes from breaking your DHCP-Relay system, as the source IP address will always be your loopback. The reverse is also true. I would suggest using a loopback on your DHCP-Server, with the correct Src-Address specified as well.

Virtual LAN (VLANs)

RouterOS Supports Virtual LANs, or VLANs. These are used to separate traffic inside an individual Ethernet segment. VLANs will reduce the number of devices in a broadcast domain; however, this does not reduce the physical size of the broadcast domain. The biggest use is to separate logical networks, such as a data network and a network with just VoIP phones on it.

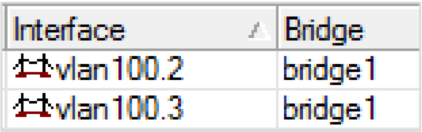
RouterOS though, does not manage VLANs as you are used to with switches. RouterOS is typically an end-point. You can run up to 4095 VLANs, each with its own unique VLAN IDs. You can also do Q-in-Q, or VLANs inside VLANs. One example of this is to have a managed Ethernet switch and you can run VLANs through a single cable. Then you break out those VLANs to untagged ports, giving you many routable interfaces on your managed switch. Then if you need another VLAN on top of one going through the switch, you can simply configure the new VLAN!

Wireless has some restrictions with VLANs. It is not possible to have VLANs on a station wireless card while in a bridge. You can run a VLAN on a station wireless card if that's the termination point of a VLAN, not bridged through. So if you need to end a connection with a VLAN, and put a IP on that VLAN interface to route through, you can do that, but you cannot add that VLAN to another bridge group and bridge it through the wireless station interface.



ether2
vlan100.2
ether3
vlan100.3

Bridging your VLAN traffic between interfaces is very simple. In the image to the right, you will see two Ethernets that have VLAN 100 configured on each. Then you will need to create a bridge interface, see bridging, and add each of the VLAN interfaces to your bridge group as shown in the image on the right. Once you get these added to the bridge group, data will flow from one VLAN to the next without issues.

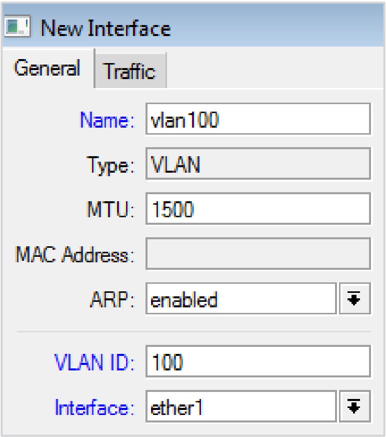


Interface	Bridge
vlan100.2	bridge1
vlan100.3	bridge1

Something to note with this type of setup, is that you can use different VLAN IDs. So if VLAN 100 is on Ethernet 1, you could bridge VLAN 200 on Ethernet 2, or bridge VLAN 100 on Ethernet 1 with VLAN 300 on Ethernet 1. You can also add Firewall rules based on your bridged interface.

VLAN Configuration

VLAN configuration is super simple. All we need to know is what the VLAN ID that you wish to run. If you wish to run VLAN 100 on Ethernet 2 then that is the extent of the configuration that you will need.



Name: vlan100
Type: VLAN
MTU: 1500
MAC Address:
ARP: enabled
VLAN ID: 100
Interface: ether1

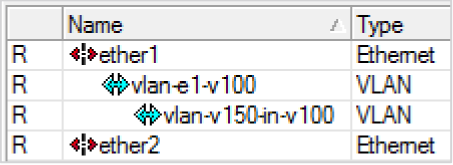
To add a VLAN, Select Interface → Add Interfaces → VLAN → Enter the Name of the VLAN, and change the VLAN ID.

Something that you need to be aware of is that MikroTik will default to VLAN1, and VLAN2 names. They increment for each VLAN you add. It is more common to change the name of the interface to match your VLAN. Something that I do is to name the interface, VLAN100.Ethernetnumber, so if you placed VLAN 100 on interface Ethernet 1, then the name for your VLAN interface would be VLAN100.1. I do this because you cannot have two interfaces with the same interface name.

Overall, VLAN configuration with RouterOS is very simple. Select what interface you want the VLAN to be on and what VLAN ID. RouterOS does treat these as separate interfaces, so you can apply NAT rules, Firewall filters, and other rules to VLAN interfaces just like other types of interfaces.

802.1QinQ – VLAN inside VLAN – Q&Q

QinQ is an addition to the original 802.1Q standard. RouterOS does support VLANs inside VLANs. The concept is simple enough, the ability to add one VLAN inside another VLAN. Normally you simply place a VLAN on an Ethernet or wireless interface; in this case, you are simply adding a VLAN on another VLAN interface.

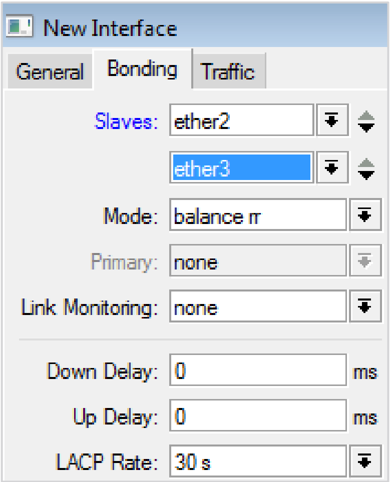


	Name	Type
R	ether1	Ethernet
R	vlan-e1-v100	VLAN
R	vlan-v150-in-v100	VLAN
R	ether2	Ethernet

In the example provided, we simply have VLAN 100 on Ethernet 1. Then, you add another VLAN 150, with the interface set to the ordinal VLAN 100 interface you already created. This allows multiple VLAN tags inside a single VLAN.

Bonding

Bonding will allow you to aggregate several interfaces into a single virtual link. You will end up getting higher data rates as well as possibly providing failover. Typically you would bond only Ethernet interfaces, however, you can bond other types of connections, including tunnels, and wireless interfaces.

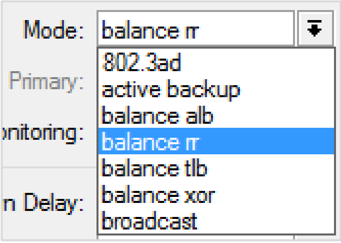


Slaves: ether2
ether3
Mode: balance rr
Primary: none
Link Monitoring: none
Down Delay: 0 ms
Up Delay: 0 ms
LACP Rate: 30 s

To create a bonded interface, you will simply click interfaces → Plus sign → Bonding. You can also use the bonding tab under interfaces as well.

When creating one of the bonding interfaces, you will need to select what interfaces are "slaves", or under the bonding interface. These interfaces will become part of the bonding group. You can select two or more interfaces. An example of using more than two interfaces was bonding six GigE interfaces on PowerRouter 732 units. After doing this, and using Jumbo Frames, bandwidth tests showed 5.9 Gbps of data able to pass between two units using bonded interfaces.

There are several different modes. The default mode of *balance rr*, is a round-robin load balancing of the data across each slave. This will provide load balancing as well as fault tolerance. This mode, typically gives the best results as long as the links are balanced. An example would be two GigE interfaces. It works the best if the connections have the same latencies.



Mode: balance rr

The *802.3ad mode* is the IEEE dynamic link aggregation standard mode. Using this mode, the interfaces will be aggregated in a group where each slave shares the same speed. This mode is typically what you would use to increase overall speed into a switch, as long as the switch supports the IEEE 802.3ad standard. This provides load balancing and fault tolerance. Using this mode you will have the ability to bond two GigE channels into a single switch, giving you close to a combined total of two Gbps instead of the 1 Gbps of a single GigE Ethernet cable. I have used this mode quite a bit, and the end performance is great.

Active-backup is only designed to allow for a backup link. One slave will be running at a time and there is no load balancing. Even though this gives you a good way to fail over to a second connection, I would recommend using dynamic routing or other methods instead of using active-backup. It does work but there are better ways of providing a failover from a primary to secondary connection. I have used *active-backu*p as a replacement for STP.

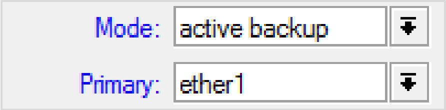
If you wish to balance outbound traffic according to the load on each slave, then *balance-tlb* is for you. This mode, will balance the outbound traffic, but the receiving data comes in by the current slave. If a slave fails, then another will take the MAC address of the failed slave. This does not require any special switch support. I typically don't use this mode.

Adaptive load balancing is what *balance-alb* is. It includes the balance-tlb functionality but also balances the receive data. Note that for this to work; you will have to have device driver support for setting the MAC address. If not, it will not work. This mode does not require any special switch support. I typically do not use this mode.

The balance-xor mode uses a XOR policy for transmission, but only provides failover. I typically do not use this mode.

The *broadcast* mode sends data out all slave interfaces at once. This will provide fault tolerance, but on some slower systems, it can cause slowdowns on the speed of the connection. I really never have used this mode, as I typically need more throughput.

On the modes that have a primary and secondary connection, such as active-backup, there is also an option to specify the primary connection. In this case, I have selected active-backup, and then selected that Ethernet 1 is my primary connection. This mode works quite well even if

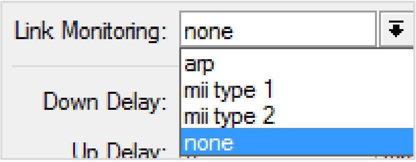


Mode: active backup

Primary: ether1

you don’t have balanced links. So your primary connection may be high-end giving hundreds of Mbps of throughput, however, your secondary may be a much smaller connection. This will fail over, but no considerations are made for the slower connection. All it cares about is, is the primary up, if not bring up the backup.

Of course, you have to have a way to detect that you have a failed link in any of these methods with fault tolerance. The link monitoring type will help you with this. There are basically three types of monitoring. ARP is the most common. It simply uses the existing ARP entries to determine if the remote interface is reachable. MII, or Media Independent Interface, basically allows the media interface to be changed or redesigned without changing the MAC hardware. There is a hardware and driver requirement that must be met for these modes to function. Most Mini-GBICs and other such devices support MII. Type one uses this standard to determine link-status of the slave interfaces. If you can unplug a slave interface and it still shows up, this means it is not supported with MII. Type 2 uses MII type2 to determine the link status. This would be used if type1 is not supported by the interface.

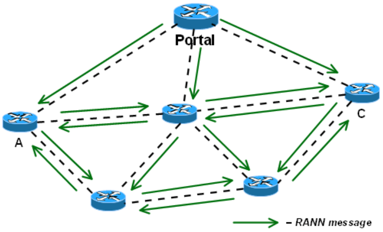


Link Monitoring: none

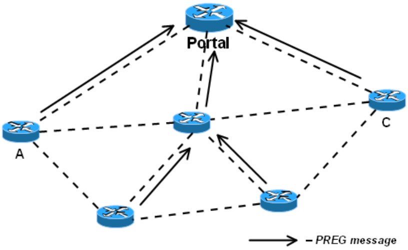
A few notes about bonding. Most of these methods require the latency of the connections to be similar, as well as the speed of the connections. If you are trying to balance across different types of connections, I would suggest using another method. Trial and Error sometimes will help you with this, set up and test across your links to see how they will perform when you are trying to balance across multiple links. If you are more worried about failures and redundancy, link failure detection will work much better on higher-end hardware such as 3COM and Intel NICs compared to other less expensive cards. An example would be some Intel cards will detect the failure and switch over in less than a second, while other, less expensive cards may require up to 20 seconds!

MESH

RouterOS offers a MESH Protocol, what is called HWMP+. We are covering it in the interfaces section, because even though most people think MESH is a wireless standard, and it does not have to run on wireless interfaces! Before we dive into MESH systems, note that the RouterOS implementation is HWMP+, not the HWMP IEEE 802.11s draft standard, so it is not compatible with the HWMP standard. It is however, based on the HWMP, or Hybrid Wireless Mesh Protocol standard. It is typically used instead of either STP or RSTP in a Layer-2 network to ensure loop-free optimal routing.



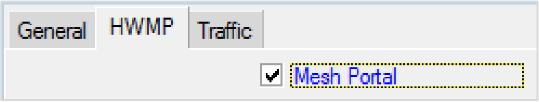
To access the MESH configuration, you will simply click on the MESH button on the left side of WinBox. To use the MESH configuration in RouterOS, the first thing you need to think of is that this is configured as a bridge. Look back up in the bridge system if you need to. You are going to create a mesh interface, and then add ports under that mesh interface. Since the ports are under the mesh interface, they will have the mesh or HWMP+ system running.



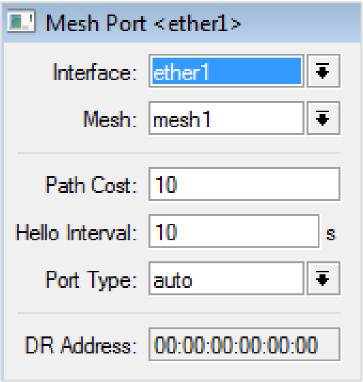
We are going to cover what is called a proactive mode of HWMP+. This mode only has one additional feature, and that is a portal. This portal is typically an entry or exit point to the mesh network. In most cases, this could be a hotspot controller or the gateway router to the Internet. By configuring a portal, the network will send a RANN message out into the mesh network, saying that it's basically the default route. As other mesh devices reply with PREG or Path Registration messages, it will build a routing tree with the root of the tree as the portal. Think of the portal as the default gateway. Also, if other nodes do not know where to send data, they will default to the portal device.

With all of that said, you do not have to have a portal mode device, however, this is better for a mesh network where most of the communication occurs between devices, not out to the Internet. Instead of a device sending RANN messages out, all of the devices send out the PREQ messages looking for other devices and destinations. Clients of an access point do not have to respond to these messages as the device that they are connected to will answer them for those clients and send PREP, or path response messages back for the clients.

When you start to build your network, keep in mind that wireless stations can't be bridged, so you will need to use the WDS setup. You can either set it up statically or dynamically. One thing I like to do with WDS is to take advantage of Dynamic WDS, but put a high access-list signal value. I don't want any device that doesn't have a good strong signal to form a WDS bridge with my devices. This eliminates low signal transmissions, and other devices that simply will not form a good quality link too!



☑ Mesh Portal



Interface: ether1

Mesh: mesh1

Path Cost: 10

Hello Interval: 10 s

Port Type: auto

DR Address: 00:00:00:00:00:00

Once I create my mesh interface, I typically create them with defaults. The only option I normally change is the mesh portal device. You do this by clicking the HWMP tab on the mesh interface and then check the Mesh Portal option.

Once you have your mesh interface up and running, simply add your mesh ports. You can add Ethernet interfaces as well as wireless interfaces. You can also add bridge interfaces if you wish to. I typically will not use the bridge interface because I want the mesh to take care of everything. Also, you can set the port type if you wish; however, RouterOS is really good with the auto type. You can set a port to the type of port that it is, either an Ethernet, wireless, or WDS.

As the mesh builds, it will determine different MACs and devices. As it builds it creates a FDB, for forwarding database. This will labels devices as outsiders if they are not part of the mesh network. These may are clients, etc. Local types are the MACs that belong to the local device. Direct types are MACs that are a wireless client on an interface that is in the mesh network. You will also have MESH MACs. These are devices that are reachable over the mesh network; it may be either internal or external to the mesh system tough.

MACs that are another mesh router directly connected to your router are called neighbors. Unknown MACs are addresses that belong to an unknown device and if that device is reachable over the mesh network, then they are changed to a larval device, but are still unknown.

The mesh system is not real difficult to manage or to run; the whole point is that of it is a self-aware Layer 2 bridged networks with many interconnection points. If one link fails, it will reroute around the failed link. This will also give you the best routing of data to its end point, thus making it better than RSTP as RSTP, those protocols are only for loop prevention. Mesh calculates the best route by simply using the link metrics -think of OSPF, just for a Layer 2 network. However, with WDS links, the metric is updated dynamically depending on actual link bandwidth. This is influenced by wireless signal and the current data transfer rate. The idea is that it will use the better quality links first, before the lower quality links.

Switches and MESH

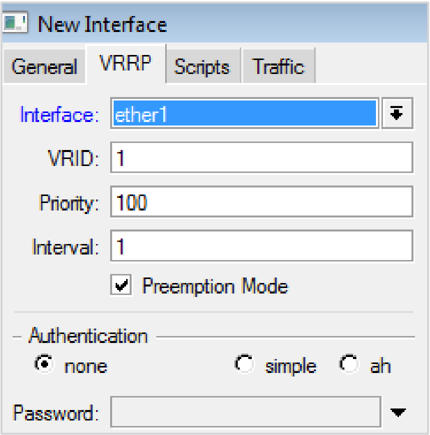
Just like anything good, there are a few configurations that you will have issues with. One of them can occur when is by simply placing a switch between two mesh nodes. Hubs do not have this issue, but the end result is that the switch can cause data to be lost and devices not to get their data. I have found the best way of getting around this, is to use a RouterBOARD 493 and simply set all of the ports as mesh ports. This will allow the mesh to use this node as a mesh device and prevent the lost MAC issue that can occur with a switch.

VRRP

VRRP or Virtual Router Redundancy Protocol is a RFC standard protocol that is used to combine several routers into a Virtual Router Group, or VR. This group's purpose is to have router redundancy. Each of the Virtual Router Nodes will have a virtual IP configured along with a virtual MAC address. One of the nodes will have the virtual IP as its real IP. This node will be the owner, and will only be replaced if the power becomes unavailable. The other routers will be backups; when they do not see a number of broadcasts that normally come from the owner at the advertisement intervals, they start an election process and one of the backup routers becomes the master router and assumes the virtual IP as their own.

Before we configure VRRP, it is important to understand how this system works and what its limitations are. The reason I say this, is because typically when I think about using VRRP, I ended up using dynamic routing to route around a failed interface or router. This typically works better, and allows you more options. But, there may not be an ability to do this in your network design etc., hence, VRRP.

To configure VRRP, you have to create a VRRP interface; this is done on the interface menu. Click Interfaces → Add → VRRP. This will start you off with a new interface. The VRID is your Virtual Router ID number. You will also need to setup a priority if you wish to have one router to be primary and another one secondary. I would also suggest using some form of authentication. Also, you will need to have the same interval on all of your routers, otherwise other routers will ignore the received advertisement packets and it simply will not work.



Interface: ether1

VRID: 1

Priority: 100

Interval: 1

☑ Preemption Mode

Authentication: ● none ○ simple ○ ah

Password:

There are three types of VRRP routers. The Master is the router that is currently being used as the IP. It would be the unit that you would be using to go through normally. The Backup, of course, is the Backup unit, and you can have multiple Backups these if you wish. When the Master is no longer available, then the Backup router with the highest priority will become the new Master. Now, if the original unit comes back on line, if it has a higher priority, it will automatically become the new Master, so your traffic will switch over to that higher priority unit. You may not wish this to occur, so you can turn on Preemption mode.

The Preemption mode ignores higher priority routers and does not switch over just because a higher priority backup router comes on-line. But the third type of VRRP Router is an Owner. An Owner router is by default the Master router. The owner needs to have a priority of 255 and its virtual IP is the same as its real IP. It will own the IP address. When this unit comes back on-line, regardless of the preemption mode, it will become the Master.

Since you created a VRRP interface, you will need a virtual IP. This IP will be placed on the VRRP interface, but you will need to have a /32 on it. What you will do is create a real IP; this is the IP that the routers communicate between on. This IP would be 172.25.0.1/24 on Ethernet 1. Your backup router would be 172.25.0.2/24. Then, you would configure your VRRP IP, the virtual IP. Place this on the VRRP interface; and the IP address would be something like 172.25.0.254/32. Your default gateway on your network can be the .254, but the other IPs will ensure that the two VRRP routers can communicate on the network.

Testing this is simple, by unplugging the master router; you will note that the IP and gateway does not change, nor does the ARP entry for the .254 or Virtual IP. The second router simply uses the same MAC and IP when it changes from Backup to Master status.

Another consideration that you will need to understand is that the backup router will need to ensure that you have the right configuration on it for it to route, send data, etc. This means that the backup router must have more than just the VRRP IP on it. Remember, VRRP is only to backup an IP address, not the entire router. You will need to have dynamic routing, default gateways, etc., on the backup unit, just like the master. That information does not carry over with the VRRP configuration. You will need to have all of the IP addresses on all of your interfaces setup with VRRP. You will also need to copy the configuration from your primary unit often to ensure you have the same configuration on the Backup router. Then if your primary router completely goes off-line, your Backup will work for you. You will also need to put some thought into what happens if one interface goes down on your primary router and not the entire router as well!

Wireless and RouterOS

RouterOS started with Wireless networking. MikroTik itself makes M-PCI radio cards including the R52, R52H, R5H and even the newer R52N radio cards. RouterOS also has full support for a number of radio cards other than MikroTik cards ones as well. There are many different modes of operations, radio frequencies and other abilities inside RouterOS. MikroTik also has proprietary high-performance wireless protocols. Most of the radio cards that you will find will meet the are IEEE 802.11x standard.

WIC – Wireless Interface Cards

Most of your wireless cards will be M-PCI; however there is support for a few PCI and PCI-E wireless cards. We will focus on the more common M-PCI cards as these are made by MikroTik and are designed to go onto the RouterBOARD hardware. The newest Radio card is the R5N. This radio card can run in both 2.4 and the 5 gigahertz spectrum. It will run standard IEEE modes such as A, B, G, and even the pre-standard of N.

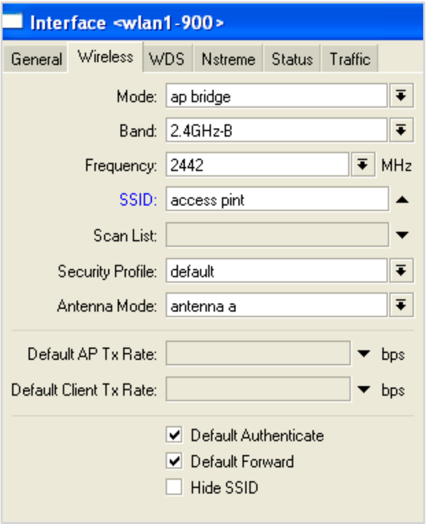
Depending on the wireless radio card that you select, you may have options for only 5 GHz only or only 2.4 GHz because as some radio cards are designed for only those specific frequencies. There are also down-conversion radio cards available. These cards use the 802.11a standard and then down convert the frequencies from the 5 GHz band, to other bands. The XR9 and XR7 are cards that do this from Ubiquiti Networks. There are other cards available as well available.

Reset Configuration Button

After you make changes to the advanced configuration in the wireless interface, you may decide that the configuration is not working for you. In v3 of RouterOS, you can now simply reset the entire wireless interface back to the default settings as if you just put the card in for the first time and powered up your RouterOS system. Remember though, all of your configuration will be lost, and if you are using the wireless interface to connect to your RouterOS system, it will be disabled and reset, so you may not be able to get back into the system again.

Basic Configuration of Wireless Interface Cards

To create a basic Access Point, you will need to click on Interfaces → Double-click on your Wireless Interfaces → Click on your Wireless Tab. Once here, you will have all of the settings that you will need to do the basic configuration. You will need to setup your radio mode, band and frequency, as well as the SSID or service set identifier. If you have a security profile that you have already configured, you will select that from the security profile drop down. This will get you going quickly!



Interface <wlan1-900>

Mode: ap bridge
Band: 2.4GHz-B
Frequency: 2442 MHz
SSID: access pint
Scan List:
Security Profile: default
Antenna Mode: antenna a
Default AP Tx Rate: bps
Default Client Tx Rate: bps
☑ Default Authenticate
☑ Default Forward
☐ Hide SSID

Default Options

There are three check boxes that you may or may not need to configure depending on your needs. The Default Authenticate box allows MACs that are not in the access list to connect. If it is in the access list, it will perform the action listed in that rule. The same goes for the Default Forward check box. By default, both of these will be on. If you do not wish your clients to directly communicate to each other, via the access point, then I would suggest turning off default forward.

Hiding the SSID

The Hide SSID check box tells the access point not to transmit its SSID in beacon frames. It will also not respond to an “empty” SSID request as well. IF your SSID is not hidden, then your access point will transmit beacon frames periodically. When you open your laptop and look for wireless networks, the listing that you see is generated by the beacon frames that were received by your laptop. The wireless access point will also respond to “empty” SSID request, if you do not have the Hide SSID option turned on.

Note that hiding the SSID is not a security measure. When clients attempt to connect by typing in the exact SSID, the SSID is transmitted in clear-text, and hence is not secure. Anyone sniffing the air can see these SSID transmissions and will have your SSID.

Default TX Rates

MikroTik has proprietary wireless frame data that is transmitted with MikroTik wireless devices. This data is typically ignored by most other devices, however for MikroTik devices; we can specify default transmittal rates both on the Access Point and on the Client. These fields will set these options for you by default. These default AP and Client Rates will be overridden if they are specified by an access list policy.

Scan List

The scan list is not normally used; however, if you have a RouterOS device with super channel license, you will have the ability to put an access point on a non-standard frequency center. The scan list will give your client devices the ability to scan the inputted channels for your SSID. When you put in the scan list, you will type the frequencies to scan separated by spaces to scan for your SSID.

Basic / Advanced Configuration Modes

RouterOS has quite a few wireless options inside your wireless interface. Most of these options do not need to be changed under normal operations, however, if you know what you are doing, there is an advanced mode available for you to use. Once you open your wireless interface, click on the Advanced mode button. This will add the Data Rates, Advanced, and Tx-Power Tabs. It will also show other information such as frequency mode, country, DFS, and WMM options in your wireless tab.

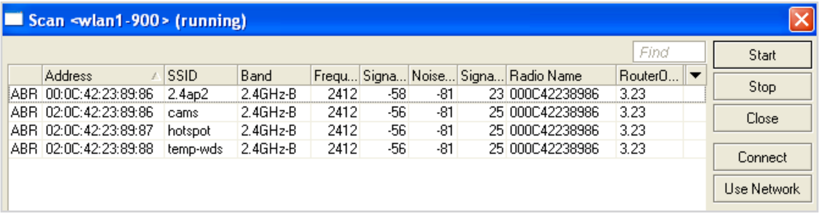
Wireless Tools

RouterOS is a very powerful router and gives you plenty of tools and abilities with your wireless interfaces. There are a number of tools right inside the wireless interface settings that will help you.

Note that using these tools will disable normal wireless operations. If you were configured and operating as it's an access point, anyone connected will be disconnected while you are using these tools.

Scanning

Scanning will allow you to basically see any broadcasting SSIDs that are within range of the wireless interface card. You will need to setup your band prior to scanning, and if you are using Nstreme, you will need to enable that as well before you will see Nstreme enabled SSIDs.



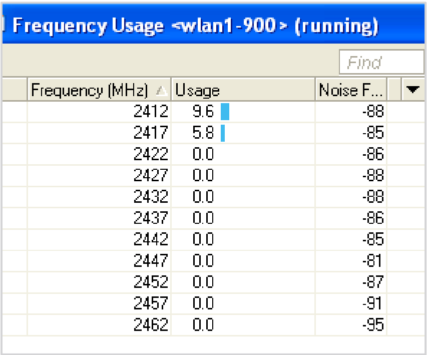
Scan <wlan1-900> (running)

	Address	SSID	Band	Frequ...	Signa...	Noise...	Signa...	Radio Name	RouterO...
ABR	00:0C:42:23:89:86	2.4ap2	2.4GHz-B	2412	-58	-81	23	000C42238986	3.23
ABR	02:0C:42:23:89:86	cams	2.4GHz-B	2412	-56	-81	25	000C42238986	3.23
ABR	02:0C:42:23:89:87	hotspot	2.4GHz-B	2412	-56	-81	25	000C42238986	3.23
ABR	02:0C:42:23:89:88	temp-wds	2.4GHz-B	2412	-56	-81	25	000C42238986	3.23

The Scan table will also give shows you the MAC addresses of the access points. If there is MikroTik proprietary extensions are transmitted, you will see the as radio names and RouterOS version. You will also see the signal strengths, SNR and noise floor information. By selecting on one of these access points and then clicking on you can then select Connect, your wireless interface will to automatically change your wireless mode to station with the correct SSID and frequency.

Frequency Usage

The frequency usage tool processes the will take all packets and data received by your wireless interface and displays you the noise floor for each channel as well as and show you the percentage of usage based on packets/data in the air. Even if they are encrypted, you can see how much a channel is being used, and based on that, you can make an educated decision on what channel to use. Typically, you want to use the channel that has a zero or low usage.



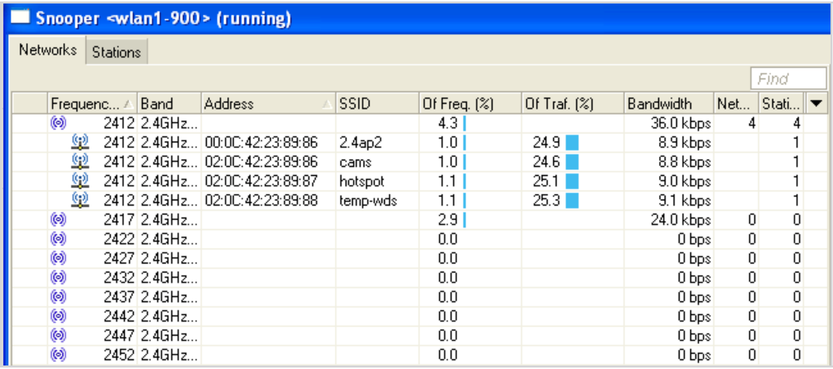
Frequency (MHz)	Usage	Noise F...
2412	9.6	-88
2417	5.8	-85
2422	0.0	-86
2427	0.0	-88
2432	0.0	-88
2437	0.0	-86
2442	0.0	-85
2447	0.0	-81
2452	0.0	-87
2457	0.0	-91
2462	0.0	-95

Sniffing

The Sniffer is another wireless tool. This is basically the same as the packet sniffer tool (described later in the book), but instead of having to be connected to a wireless interface, this pulls packets out of the air. How? Well wireless is wireless; just because you are not connected to an access point does not mean that there are not packets floating in the air! This system simply listens on all channels all SSIDs and reads packets as it receives them. It will listen on multiple channels, as it does not look at only the channels that are defined by specific SSIDs. You can also use this in conjunction with a streaming server. This use is covered in the Packet Sniffing section under RouterOS tools more.

Snooping

Using the snooper tool you, will be able to see all of the wireless stations, and access points and statistical over-the-air data information about each one as that data is moved around in the air. From the image below you can see what kind of data you can collect, including how much data, the packets, what SSID, band, frequency, and channel usage percentages they are using including the actual bandwidth being used by each device!

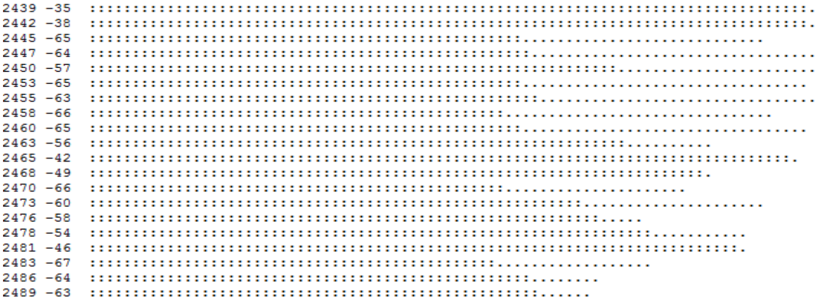


Frequenc...	Band	Address	SSID	Of Freq. (%)	Of Traf. (%)	Bandwidth	Net...	Stati...
2412	2.4GHz...			4.3		36.0 kbps	4	4
2412	2.4GHz...	00:0C:42:23:89:86	2.4ap2	1.0	24.9	8.9 kbps		1
2412	2.4GHz...	02:0C:42:23:89:86	cams	1.0	24.6	8.8 kbps		1
2412	2.4GHz...	02:0C:42:23:89:87	hotspot	1.1	25.1	9.0 kbps		1
2412	2.4GHz...	02:0C:42:23:89:88	temp-wds	1.1	25.3	9.1 kbps		1
2417	2.4GHz...			2.9		24.0 kbps	0	0
2422	2.4GHz...			0.0		0 bps	0	0
2427	2.4GHz...			0.0		0 bps	0	0
2432	2.4GHz...			0.0		0 bps	0	0
2437	2.4GHz...			0.0		0 bps	0	0
2442	2.4GHz...			0.0		0 bps	0	0
2447	2.4GHz...			0.0		0 bps	0	0
2452	2.4GHz...			0.0		0 bps	0	0

Spectrum Analyzer

In RouterOS v4.3, MikroTik added the ability to perform a spectrum snapshot for R52N and R2N radio cards. This tool will allows you to see what frequencies are in use regardless of the type of transmission. Using this feature, you can find other radios and RF equipment that does not use the 802.11x standards. These could be wireless video senders, or Trango™, or Motorola Canopy™ radios as well. It really don't matter, it's a true spectrum analyzer integrated right into your RouterOS system.

To use the spectrum analyzer this, you must use a command line mode, however, since Dude v3.6, MikroTik added the ability to take this information and translate it into a graphical form. See the Dude -Spectrum Analyzer section for more information on using the Dude to perform these analyses.



```
2439 -35  ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::.
2442 -38  ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::.
2445 -65  :::::::::::::::::::::::::::::::::::::::::::::::::::............................
2447 -64  ::::::::::::::::::::::::::::::::::::::::::::::::::::.................................
2450 -57  ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::.......................
2453 -65  :::::::::::::::::::::::::::::::::::::::::::::::::::.................................
2455 -63  :::::::::::::::::::::::::::::::::::::::::::::::::::::................................
2458 -66  :::::::::::::::::::::::::::::::::::::::::::::::::...............................
2460 -65  :::::::::::::::::::::::::::::::::::::::::::::::::::.................................
2463 -56  :::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::..........
2465 -42  ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::.
2468 -49  ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::.
2470 -66  :::::::::::::::::::::::::::::::::::::::::::::::::.....................
2473 -60  ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::.....................
2476 -58  ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::.....
2478 -54  ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::...........
2481 -46  ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::.
2483 -67  ::::::::::::::::::::::::::::::::::::::::::::::::..................
2486 -64  ::::::::::::::::::::::::::::::::::::::::::::::::::::........
2489 -63  :::::::::::::::::::::::::::::::::::::::::::::::::::::......
```

The command for using the spectrum analyzer spectral scan is simply *spectral-scan* then the interface number. Refer to the on-line documentation for all of the commands associated with this. An ASCII graph, like the one above, will be displayed showing the signal received regardless of the transmission type.

Air/Data Rates and Performance

I wanted to make sure I said something about air and data rates. I have customers calling me asking how fast an access point is, or what the maximum speed of a wireless point-to-point link is. When I tell them, they will say “But I am connected at 54 Mbps”. So let’s clarify this information!

We will start with 802.11b. The maximum air-rate that you can get is 11 Mbps. But the actual data transfer rate is right at 6 to 7 Mbps, depending on the type of traffic. UDP traffic will be on the higher side. However, that assumes only one wireless client! As you add more and more clients, the data rate per client goes down. You have to keep lowering total possible data rates. What that means is, with an 802.11b access point, the absolute highest bandwidth you can get is when only a single client is connected to that access point, and it would be around 6 Mbps. Then, as you add more clients, each additional client uses up a bit more access point time, so that actual throughput drops a bit more with each client you add.

Now let’s talk about 802.11a/g. If you have a data connection at 54 Mbps air-rate, the maximum data rate will be around 30-40 Mbps in one direction. As you drop your channel width mode, from standard 20 MHz channels down to 10 or 5 MHz channels the maximum you also will get a cut in throughput goes down proportionally to the decrease in channel width. Assuming you can get 30 Mbps of data though a 802.11g connection, and then you drop from a 20 MHz to a 10 MHz channel, you would lose roughly half of your data throughput, going down to around 15 Mbps of throughput. Drop that down to a 5 MHz channel size, and you will cut it from 15 Mbps to around 7.5 Mbps.

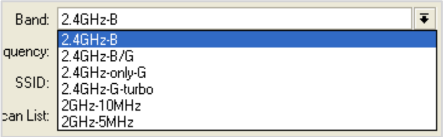
Access Point Time

Another question I get is about access point time. For example, how many clients can I put on a single access point? This is really a measure of modulation type and connection rates. For example, if you have a client connected at the 1 Mbps connect rate, they have to use 11 times more of the access point time to transfer the same 500k of data compared to a client connected at 11 Mbps.

My recommendations is, for normal usage, 30-40 clients on a B/G access point is high, but if most are using 11 Mbps connections on most of your clients, then you can get upwards of 50-60 clients! On 802.11a, I would say this is a bit more, assuming again, good data rates, upwards of 60-70 clients should be possible. This will also depends on how much bandwidth you are giving each connection as well. If you are a wireless ISP, then selling 4 Mbps downstream Internet access on an 802.11b access point only allows you to sell a few connections on one access point!

Bands

There are a number of bands that RouterOS can operate in. The IEEE standards typically apply unless you are using RouterOS with a super channel license. You have 802.11b or b/g modes, very common, but you can also turn off the CSMA protocol with 802.11b and just run g-only with all the 802.11g air rates. You can also run g-Turbo mode as well. This uses a 40 MHz channel size instead of a 20 MHz channel size. Doing this in 2.4 GHz typically will reduces the number of non-overlapping channels to about two, but will give you higher than expected data rates. In a wireless ISP scenario, I would stick to the smaller channel sizes.



Band: 2.4GHz-B

You also have options for 2.4 GHz at 10 MHz and 5 MHz channel sizes. The reason for these options is to have more channels available, and also to reduce interference. For every ½ cut in channel size, your signal-to-noise ratio (SNR) will improve by receive around 3dBi, just due to the fact that you are now only listening to only one half of the frequency range. This does not show up in signal strength improvement, only in SNR.

In 5 GHz with 802.11a, you have the same options, both A-Turbo using 40 MHz channel sizes as well as smaller 10 MHz and 5 MHz channel sizes.

Wireless Operational Modes

RouterOS offers a number of different wireless operational modes for wireless interfaces. No longer are you limited to a device being only an access point or just a station. RouterOS allows you to select between these modes simply by changing a drop down box! Note that we will discuss Nstreme Dual Slave mode in the Nstreme Dual section instead of in the operation modes section.

AP-Bridge (P2MP Access Point) Mode

You may be familiar with one of the most common radio modes; one is the AP-Bridge mode in RouterOS. This is the standard point-to-multipoint access point mode. This will allow a number of clients to connect at the same time, providing computers the ability to connect to an access point. In this mode you do have the ability to add the radio card to a bridge group. The IEEE standard will allow bridging of a wireless radio card with other types of interfaces as long as the wireless card is running as an access point. You can enable WDS support though for this type of interface.

WDS-Slave Mode

The WDS-Slave mode is basically an access point, however, it connects to an AP-Bridge radio cards and forms a WDS (wireless distribution system) connection. The only difference between this and the AP-Bridge mode is that if the primary radio, the one in AP-Bridge mode changes channels, the access point in the WDS-Slave's will change channels accordingly.

Bridge (P2P Access Point) Mode

This mode is the same as the AP-Bridge mode with one major exception. It will allow only one station to connect. This means it’s very well suited for point-to-point wireless links where there will only be one station connection.

Of course you can add this to a bridge and/or use WDS to bridge though if you wish too. What is nice about this mode is that any other attempts to register to the radio is completely ignored and not processed.

Station (Wireless Client) Modes

The other most common mode is the Station mode. In this mode, the radio card acts as a client device. You would use this if you wished to connect to an access point, and act as a client device. Most of the CPEs (Customer Premise Equipment) that WISPs would install are be set to this mode. When you set this mode, you will typically need to do some form of routing, or masquerading as the IEEE specs do not allow bridging of a wireless interface in station mode.

If you are required to do bridging; the proper way, per IEEE RFC, is to use the Station-WDS mode. This mode, along with an AP-Bridge radio running WDS (Wireless Distribution System) is the proper way to create a true bridged link. If you are linking a number of stations and wish them to be bridged, there is little performance loss when using this method, as long as there is only one access point. See the “Using WDS” section for more information on how to set this up.

Another way of bridging is to use the Station-Pseudobridge modes, and yes, I say “modes” because as there are two; Station-Pseudobridge mode and the Station-Pseudobridge-Clone mode. These modes will both allow you to add the wireless interface to a bridge group and run properly run. These are both non-standard, as in they are not per the IEEE RFC. To make this work, MAC NAT is performed for devices behind other interfaces of the bridge group with the Pseudobridge mode. In the Station-Pseudobridge mode, MAC NAT will be performed with the wireless radio cards MAC. All clients and devices behind the wireless card will appear as coming from the MAC of the radio card. In the Station-Pseudobridge-Clone mode, a device will transmit, and then the wireless card will take that MAC and use it, or you can specify a MAC to use in the settings of the wireless interface card.

WMM – Wi-Fi Multimedia

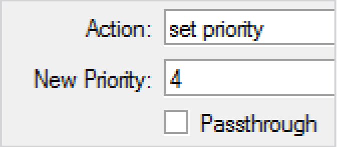
WMM Support: enabled

WMM is actually WME or Wireless Multimedia Extensions per the Wi-Fi Alliance interoperability certification. This is based on the IEEE 802.11e standard, and provides based QoS over an 802.11x network. WMM prioritizes traffic based on four different ACs, or access categories. This though, does NOT guarantee throughput! RouterOS does support WMM on its wireless interfaces by simply enabling it, disabling it, or requiring it. If you set the WMM Support to “required”, then a CPE will not be able to connect without using WMM. If you have WMM simply enabled on the wireless interface, then the client could or could not have WMM, but still could connect. So you will need to be careful with this feature if you do have clients that do not support WMM.

WMM has four priorities as mentioned. According to the WMM spec, priorities 1 and 2, are for background data, priorities 0 and 3 are for best effort traffic, priorities 4 and 5 are meant for video, and 6 and 7 are reserved for voice. WMM is used in when setting priority of the data flowing through your RouterOS system. These priorities can come from several places; one is WMM packets that are transmitted over a wireless interface, and the second is the “ingress priority” from packets that come into your RouterOS system though a VLAN. As this priority is in the VLAN header, other packets, for example routed packets not over a wireless link, would be treated as an ingress priority of 0.

Setting WMM Priority’s

The two methods to set WMM priority; is by using DSCP (Differentiated Services Code Point) or the Set Priority field in the Mangle system. You can set priorities by either using the IP Firewall, Mangle system, or you can use the bridge Firewall filter rules as well. Even though you get an ‘ingress priority’ through either a VLAN or WMM packet, note that you still need to set the priority as the ingress value is not copied to the priority value by default. You will need to ensure you have the proper rules to do this. You also need to remember that priorities only go out on either WMM wireless interface or a tagged VLAN, that’s it.



Action: set priority

New Priority: 4

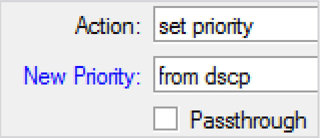
□ Passthrough

Setting the priority by using your Mangle is simple enough; you will use the Firewall/Mangle system to identify the traffic (see the Traffic Identification section), and then your action will be to set priority. Remember there are only 4, four, priority types.

The second way is to manage priority is to use the DSCP bits. This is the preferred way as the DSCP bits are attached to your IP packets and are transmitted regardless of the medium. Also, once you set the DSCP bit, it will travel through your network and WMM will automatically use this DSCP bit to assign priorities. The method of deciding what DSCP bit to setup for what priority gets a bit complicated. For this, we have the chart below:

WMM Traffic	WMM Value	DSCP Bit(s)
Best Effort	0	0 through 7
Background	1	8 through 15
Background	2	16 through 23
Best Effort	3	24 through 31
Video	4	32 through 39
Video	5	40 through 47
Voice	6	48 through 55
Voice	7	55 through 63

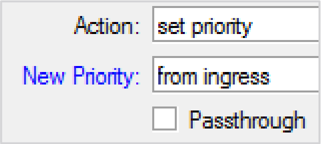
Even though you have the DSCP bit set, you will need to tell your RouterOS system to use the DSCP for the priority. All you have to do is create a single Mangle to set the priority based on DSCP, or based on ingress (if you are receiving it from a VLAN). As you can see, you simply create a blank Mangle rule to set the priority from the DSCP. You also can set this to "from ingress"' if needed as shown.



Action: set priority

New Priority: from dscp

□ Passthrough



Action: set priority

New Priority: from ingress

□ Passthrough

Security Profiles (Securing your Wireless Connection)

To understand wireless security, you have to understand why wireless has more security issues than other types of connections, such as wired connections. The main reason is simple; the transmission is in the open air; it is not limited to the inside of a cable. For instance, if you setup a point-to-point link between buildings a few hundred feet apart, you are sending and receiving data from the buildings. You could stand in-between these buildings and see all of the data that is being transmitted from both ends if there is no security enabled. What if you were a mile behind one of the antennas? You would get (at least) the transmission from the far end of the link, (depending on power levels). You are a mile away and receiving the wireless transmissions. With an Ethernet cable running between the buildings you eliminate the other RF energy in the air and you would have to have physical access to the cable to be able to tap it. With wireless, even if they are not connected and directly communicating with the access point, you can still watch data flow through the air! I hope your data is encrypted!

With this said, I think you can see how wireless is considered to be very insecure. However, with the proper encryption and security practices, you can secure your wireless signals and prevent unauthorized computers from connecting or viewing your data. Without a connection, an intruder won't be able to transmit data to the access points, however, this doesn't prevent them from listening to the air and possibly pulling data as it goes between a station and access point.

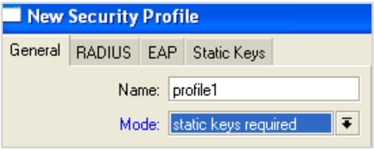
The way RouterOS works is that you will define security profiles with a form of encryption. These security profiles can then be setup on your wireless interface. Simply define the profile, setup WPA and the shared key, and then you will change the drop down on the wireless interface to the profile that you configured. Once you do this, the wireless interface will be using the security profile that you setup.

MAC Authentication

I will start off this section by saying, "MAC does NOT provide security on your network". By using MACs to control access, you are telling the access point that you must have such-and-such MAC address to connect to the access point. Keep in mind that this is not encryption, so data is in the air is still unencrypted by using this. Second, I want to tell you that MAC does NOT provide security on your network. Even in RouterOS it is very simple to spoof a MAC address and there are plenty of applications out there for even the average Joe to spoof a MAC. MAC level security is just not going to do anything for your wireless network security.

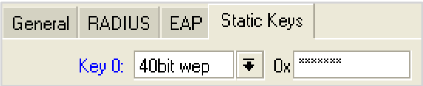
WEP (Wired Equivalent Privacy)

WEP is an IEEE standard to secure wireless networks. This uses a shared key to encrypt data between the access point and the client device. To setup WEP on RouterOS, you will need to setup static keys in the Security Profiles. You will setup your mode for static keys. If you make WEP optional, which means that clients don't need to have to have WEP to connect, but if WEP it is required, then you have to have the WEP key to continue. Then, under the static keys option, you can select if you wish to use a 40 or 128 bit WEP key. This is the key that you will share with your clients to allow them to connect.



Name: profile1

Mode: static keys required



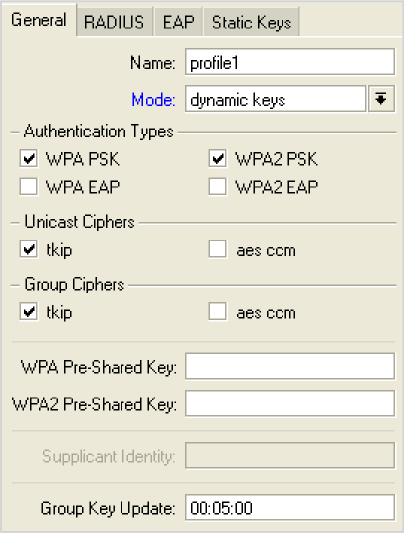
Key 0: 40bit wep 0x ********

You can also select a transmit key. This allows you to connect to the Access point without the key and then the key is given to you so that you can communicate securely using WEP. You will need the mode as static keys optional so that they can connect and get the key before they start using the key. Keep in mind that this method transmits your WEP key over the air as well.

With that said, my recommendation is to NOT use WEP. WEP is outdated; it was originally created in 1997. With any Linux-based laptop, it usually takes for the most part it takes about 20 seconds to break WEP. It's considered very easy to break and should not be used if you are wishing to have a well-secured wireless network.

WPA / WPA2

WPA or Wi-Fi Protected Access was created once several weaknesses were found in the WEP system. These weaknesses were considered serious and you should consider WPA as a replacement to WEP. Keep in mind that they are not backwards compatible. WPA2 is considered a replacement for WPA since there were issues with the TKIP key stream found in WPA.



Name: profile1
Mode: dynamic keys

Authentication Types
☑ WPA PSK ☑ WPA2 PSK
□ WPA EAP □ WPA2 EAP

Unicast Ciphers
☑ tkip □ aes ccm

Group Ciphers
☑ tkip □ aes ccm

WPA Pre-Shared Key:
WPA2 Pre-Shared Key:

Supplicant Identity:

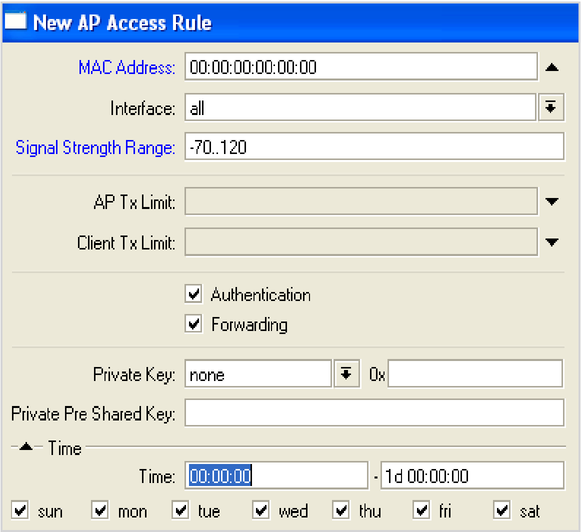
Group Key Update: 00:05:00

Typically you will deploy WPA the same as you would as WEP. Go to your security profiles and, when creating a new security profile, select the mode as “dynamic keys”. You can then choose if you wish to use PSK (pre-shared keys) or EAP (Extensible Authentication Protocol). Most users will use PSK, as this relies on a shared secret that you will give to the clients connecting.

You will also be able to select what kind of ciphers as well. Most users will be fine using TIKP ciphers; however, if you are security conscious, you can use the AES-CCM ciphers as well. RouterOS can run WPA and WPA2 at the same time, and if you wish you can specify different shared keys for each method. Once you setup your security profile, you can then enable it on your wireless interface by selected it in the security profile dropdown in your wireless interface.

Access Lists

I like to talk about access lists right next to the security profiles section because they are related. Your access list allows you to do is setup a number of rules based on MAC address, signal strengths, shared keys and time. These rules will allow you to specify if the radio in question has the ability to connect, has the ability to talk to other clients connected to the same AP, and what shared key to use based on the current time.



New AP Access Rule

MAC Address: 00:00:00:00:00:00

Interface: all

Signal Strength Range: -70..120

AP Tx Limit:

Client Tx Limit:

☑ Authentication

☑ Forwarding

Private Key: none 0x

Private Pre Shared Key:

Time

Time: 00:00:00 - 1d 00:00:00

☑ sun ☑ mon ☑ tue ☑ wed ☑ thu ☑ fri ☑ sat

These rules, like other ordered lists in RouterOS, run from the top down, in order. Once a rule is matched, the processing stops. This allows you to setup times when the rule may or may not match, for example, to allow any MAC to connect during lunch hour, but at other times, only allowing a few MACs to connect. Remember that this is not just MAC authentication only. If you have WPA2 running, then you will still need that WPA2 shared key. But you can also specify that a specific MAC address must have this specific pre-shared key. This will allow you to setup different pre-shared keys for each MAC that you have connected to your access point.

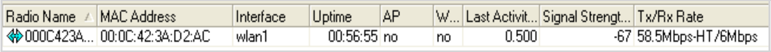
The Signal Strength also limits the Access Point to only allowing clients with strong enough signals to give good quality connections. An example is in 802.11b, a -70 dBm signal is typically required to have an 11 Mbps air rate connection. With a rule like the one above, every MAC must have at least between a -70 dBm a 120db signal to be able to connect. A client with a -80 dBm signal would NOT be able to connect. Remember though, if you create a rule like this, you then need to specify that anyone that doesn’t match that rule would not be authenticated by un-checking the authentication check box.

You can also limit a customer’s forwarding ability. This prevents the client from talking through the access point to another client directly connected to the same access point, or client-to-client communications. This does not prevent a client on access point A from communicating to a client on access point B.

The TX Limits are for MikroTik CPE or clients. They can be in any wireless mode that connects them to the access point. Once they are connected, you can add an AP and Client TX limit. This will limit the TX speed of the access point sending to the client as well as limit the clients transmit speed sending to the access point! This information is embedded in the MikroTik proprietary wireless frame extensions, and will not work with most other non-RouterOS clients.

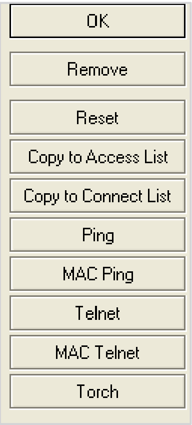
Registration Table

The wireless registration table is exactly what it sounds like. It is a listing of wireless registrations or connections to your radio card. If your wireless interface is in station mode, it's registered to the access point so you will see a registration. Inside the registration table, you will see information about your wireless connections, such as up time, signal strengths and air rates.



Radio Name	MAC Address	Interface	Uptime	AP	W...	Last Activit...	Signal Strengt...	Tx/Rx Rate
000C423A...	00:0C:42:3A:D2:AC	wlan1	00:56:55	no	no	0.500	-67	58.5Mbps-HT/6Mbps

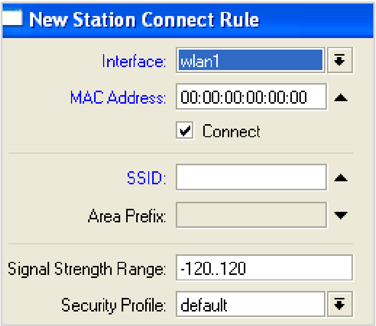
Double-clicking the registrations in your list, you can see much more detailed information about your connection. The RouterOS proprietary information such as RouterOS version, CCQs and the P Throughput are all listed here. The P Throughput field is a "possible throughput" that RouterOS will calculate based on a number of factors. It will also show the CCQ information on both directions and the Signal Strength in both directions of your link.



Also inside each registration, are a number of support tools that you can use just on the wireless registration. One of the fields you will see is the Last IP. This is simply the last IP packet that has traveled through the interface. It is not the "IP Address" of the link, or a side of the link, just simply an IP that has flowed through the link. Remember this when using Layer 3 tools such as Ping, telnet, and Torch. You will also have options to copy the MAC information to either your access or connection lists here as well.

Connection Lists

Connection lists are the exact opposite of the access lists. Access lists, if you read back a few pages, are for controlling access to an access point. Connection lists are for telling your station what and how to connect to access points! To use them, remember that they are an ordered rule list, just like anything else in RouterOS. What you can do is either setup by MAC or SSID, as well as another feature called areas. We will discuss that in the next section. You start creating this ordered list, with multiple SSIDs, and Signal Strengths.



New Station Connect Rule

Interface: wlan1

MAC Address: 00:00:00:00:00:00

☑ Connect

SSID:

Area Prefix:

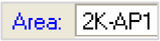
Signal Strength Range: -120..120

Security Profile: default

An example is that you can say connect to SSID ‘Tower1’ only if the signal strength is above -70 dBm. If it drops below that signal level, then it will disconnect and start searching for another connection. The 802.11x standards will not drop a working access point for another with a stronger signal unless the signal of the currently connected access point drops below the allowed range. So once tower 1 drops below that -70 dBm level, it will disconnect and start looking for something else.

Another thing that is handy is that you can have different security profiles associated with different rules and/or SSIDs. Some wireless ISPs will use a standard load on a CPE device, loaded up with SSIDs, security profiles and signal strengths settings for installers. This way, once they point it at a tower, it will automatically connect with the right security profile for that access point, assuming they have enough signal strength. This creates a simple method of having installers performing installations without having to have the installers knowing all of the security keys and other settings.

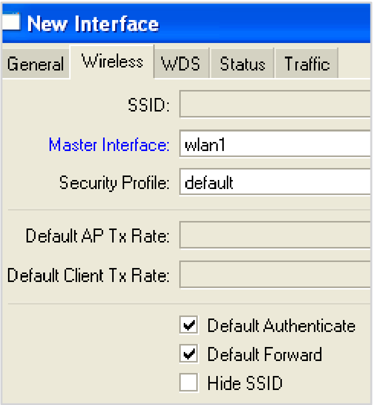
Area / Area Prefixes


Area: 2K-AP1

Inside the advanced tab of your wireless interface settings, there is a value called Area. This area value is matched up with the connection lists area prefix. An example of this, is that if all of your wireless towers and access points have an Area set that starts with “2K”, then you can create a connect rule that has an area prefix of “2K” as something to attempt to connect to. This area prefix allows you to either match the entire area on the wireless interface or just the beginning of the area. If your area is “2K-AP2” on your wireless interface, and your connect list just has “2K” as the area prefix, it will match. Of course it will have to match the rest of the values in your connect rule as well.

Virtual Access Points

RouterOS has a great feature called Virtual APs, or Virtual Access Points. These are treated as new interfaces with separate SSIDs and security profiles on the same radio and channel. Since this is considered a completely new interface, you can run separate IP space, separate DHCP-Servers, and even run other services such as hotspots, all while running a single radio card.



SSID:
Master Interface: wlan1
Security Profile: default
Default AP Tx Rate:
Default Client Tx Rate:
☑ Default Authenticate
☑ Default Forward
□ Hide SSID

To create a virtual-AP, simply click on Interfaces → Add → Virtual AP. The main three settings you will need are the new SSID, what actual radio card you will be transmitting this new virtual SSID from, and the security profile. All of the other settings are the same as any other wireless interface. You can form WDS connections as well with a virtual AP. Once you create this interface, you will either need to place it into a bridge group, or place IPs and other Layer 3 services on it for it to work, just like if you had a new wireless interface card installed.

You will need to be careful however, because, even though you have two separate SSIDs, they are still on the same channel as the master wireless interface. They share the available bandwidth of the frequency that you have placed them connected on.

VAPs and VLANs

Many customers ask about running several different VLANs for different SSIDs and networks. A common practice is to run a specific SSID that would be for corporate, or private traffic, as well as another SSID that would be for general public use. In these cases, you may want to VLAN each SSID to separate the traffic. This is very simple to do, but many people have become accustomed to seeing a check box to enable VLAN on this interface. RouterOS handles this inside the VLAN and bridging system of RouterOS.

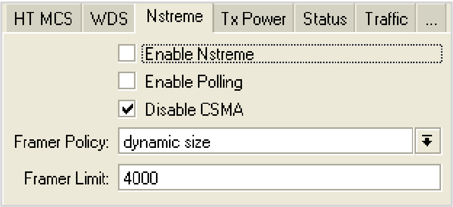
To give you an example, let's use VLAN 100 for our private traffic and VLAN 200 for our public traffic. On all of our RouterOS access points we have created and SSID called 'private' and another called 'public'. In our case, we will have our WLAN1 interface as our 'private' SSID, and we created a Virtual AP, under our WLAN1, with our 'public' SSID.

To complete the task above, we will need to complete the following steps:

1) Create a 'private' and 'public' bridge group
2) Create the VLAN 100 and VLAN 200 interfaces on the correct Ethernet port.
3) Add both the VLAN 100 and the private WLAN interface to the private bridge group.
4) Add both the VLAN 200 and the public VAP interface to the public bridge group.

This sounds simple enough. Remember, VLANs are basically bridges, so we are just going to create a bridge, with the correct VLAN and the correct interface. Data coming in the public VAP interface has only one port to go out, that would be the VLAN interface, when it passes though the bridge group it adds the VLAN tag and sends it out the VLAN interface, properly tagged as that VLAN. The same is done for our private interface; the only difference is that we are using the actual WLAN interface instead of a VAP.

Nstreme



□ Enable Nstreme
□ Enable Polling
☑ Disable CSMA
Framer Policy: dynamic size
Framer Limit: 4000

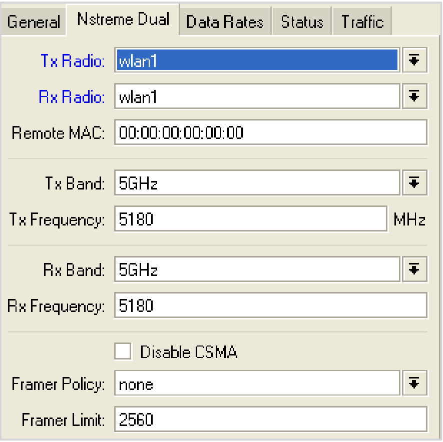
Nstreme is a proprietary extension of the 802.11x design that MikroTik created to overcome some of the limitations and increase the performance of wireless links. This is only supported with MikroTik RouterOS running on both ends. The goal is to increase performance typically at the cost of latency. Nstreme does a few things, including compression, polling, and no limits on distance. It will also combine frames similar to the way M3P (Mikrotik Packet Packing Protocol) does as well.

To enable Nstreme on your access point, you will simply need to check the Enable Nstreme button on the Nstreme tab of your wireless interface. Here you can also set your framer policy, limits, polling as well as the ability to disable CSMA. Once you check this box on your access point, if you had clients connected, they will be disconnected. You will need to check the corresponding box on your clients as well so that they will connect. Also remember that when you are scanning with Nstreme enabled, you are looking only for Nstreme enabled access points, not standard a/b/g/n access points.

In typical usages, Nstreme mode will provide higher data throughput, however, typically latency increases a bit. This is mostly due to the compression that occurs in the link. There is also no limit in the ACK timeout values, so you can go greater distances compared to running standard 802.11x. We have seen 52 Mbps connections using 5 GHz Turbo modes and Nstreme, however, with 802.11n; this performance is upwards of 70 Mbps half-duplex. This is not what you will get all of the time, typical link performance and path analysis should be done to determine what your actual throughput is expected to be.

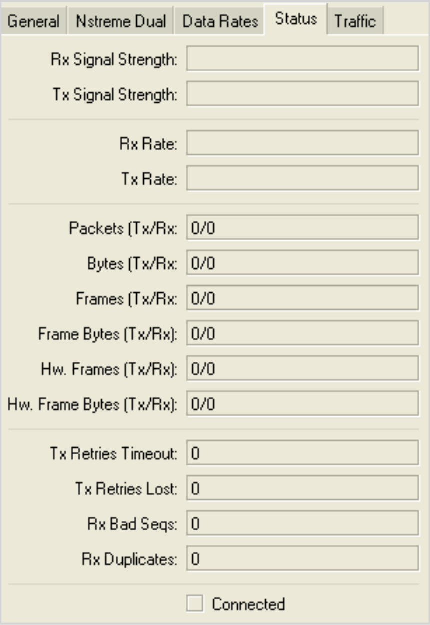
Nstreme Dual

Dual Nstreme uses two wireless interface cards to create a full-duplex wireless link. You will need to setup your two wireless interfaces to N-Nstreme-Dual-Slave mode. Once you set your radio cards to this mode, everything with the exception of the Tx power settings are for the most part ignored by the system and are then controlled by a new Nstreme dual interface that you will create. To create this, click on Interfaces → Plus sign → Nstreme Dual Interface.



Tx Radio: wlan1
Rx Radio: wlan1
Remote MAC: 00:00:00:00:00:00
Tx Band: 5GHz
Tx Frequency: 5180 MHz
Rx Band: 5GHz
Rx Frequency: 5180
□ Disable CSMA
Framer Policy: none
Framer Limit: 2560

This will create your new interface. This interface will use two radio cards to provide full-duplex throughput. It does this by enabling one card for only receive only and the other card to only transmit. On the Nstreme Dual tab of your new interface, you will need to setup what radio card is receiving and what one is transmitting. These would be the TX and RX radio settings.



Rx Signal Strength:
Tx Signal Strength:
Rx Rate:
Tx Rate:
Packets (Tx/Rx: 0/0
Bytes (Tx/Rx: 0/0
Frames (Tx/Rx: 0/0
Frame Bytes (Tx/Rx): 0/0
Hw. Frames (Tx/Rx): 0/0
Hw. Frame Bytes (Tx/Rx): 0/0
Tx Retries Timeout: 0
Tx Retries Lost: 0
Rx Bad Seqs: 0
Rx Duplicates: 0
□ Connected

One of the most common mistakes that are made in the Nstreme is that once configured, you will have frequencies for TX and RX; these frequencies are flipped on your remote system. So for example, if you transmit at 5180 MHz then you have to receive at 5180 MHz on the remote side. It is a common mistake to set 5180 MHz as the transmit frequency at the remote end. Another mistake is to forget where to get your Remote MACs from. The remote MAC is the MAC address of the far end's Nstreme Dual interface. This is only created once you do the initial configuration of the interface. Click on General of your Nstreme Dual interface, and you should see a MAC Address. This MAC would go into the other end's remote MAC address field.

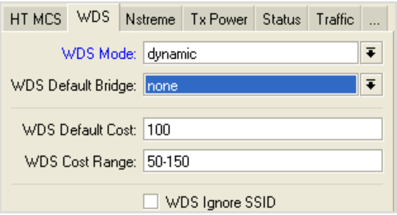
Just like other wireless interfaces, you can set the frequencies, disable CSMA, and set your framer policies and configuration of your data rates. Your Nstreme Dual interface will take care of what radio card transmits where. Inside your status tab you will have all of the information that you would need to diagnose your connection and perfect it. This information will include your signal strengths, retries and timeouts. The connected check box at the bottom shows you if you are actually connected or not. If you use a dual-polarity dish or antenna, and you have about 20-25 dBi difference from what your signal is and what your link path analysis showed you should have, and then you may need to swap your Tx/Rx radio cards. This would be to simply change what card transmits and what card receives.

Using WDS (Wireless Distribution System)

WDS or (Wireless Distribution System) is designed to create custom wireless coverage areas by using multiple access points that can pass data between each other just as if there was a wire connected between them. The access points will need to be on the same SSID and same channel as well as use the same band and channel size. There are two types of WDS, one is dynamic and the other is static. When you put a radio into WDS mode, you will select what mode type and typically what bridge group to add the WDS interfaces into. Per RFC specs, if you wish to bridge a wireless link, you must use WDS. This would include if you are wishing to send VLANs across a bridged wireless link.

WDS Bridged Wireless Link

To create a WDS bridged wireless link, on the access point, configure the wireless interface to use the proper modes and frequencies that you wish to use. You will need either an AP-Bridge or bridge mode interface to work with the WDS system. Then on the WDS tab you will need to configure the WDS type, dynamic or static, and the default bridge group. Typically you will create a bridge group and add your Ethernet interface on it. You do not need to add the wireless interface because when the client connects the interface is created. The system will either dynamically add the new WDS interface to the bridge group or you will manually add it once the WDS interface is created. This will depend on if you have configured the WDS-Default-Bridge option.



WDS Mode: dynamic

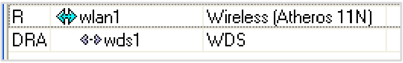
WDS Default Bridge: none

WDS Default Cost: 100

WDS Cost Range: 50-150

□ WDS Ignore SSID

On the remote end, you can use Station-WDS. If it is simply a point-to-point link, then I would use Bridge on your main site and Station-WDS on the other. On the Station-WDS side, you would also create a bridge group, add your Ethernet interface and then configure your Station-WDS wireless interface with the proper WDS type and default bridge group. When the link comes up, the wireless interface card should be dynamically added to the bridge group. You should also notice a dynamic WDS interface created on the bridge side, as well as it being added to the bridge group as well.



```
R      wlan1      Wireless (Atheros 11N)
DRA      wds1     WDS
```

Static WDS Bridges

If you wish to use static WDS entries, you will need to setup your wireless interface WDS settings to static mode, and then you will have to have the remote MAC of the radio cards you wish to form a WDS link to.

WDS Bridged Access Points

Of course the point of using WDS is to have multiple access points within an area without connecting them with wires. To create a custom coverage area like this, you will simply set the modes to AP-Bridge, and then as you add more systems with the same SSID, Channel and band, they will dynamically create WDS links to any other systems with the same configuration that are within range.

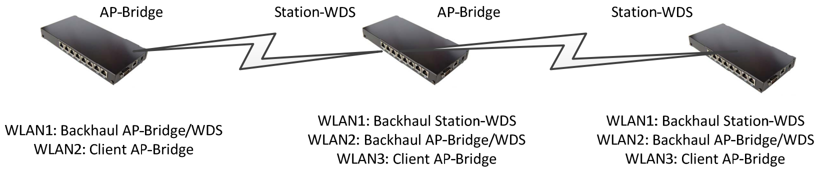
Two major issues that you will come across by using this method is a) the slow performance and b) bridging loops. When you first bring up a new unit with dynamic WDS enabled, if it can see four other systems with the same configuration, it will attempt to form a WDS connection and interface. Even though this is what you normally want, the issue that often arises is bridging loops. The interfaces are not smart enough to make a determination if WDS link 1 is better than link 2, and therefore link number 2 should be disabled. You will need to use some form of spanning tree protocol on all of your access points to ensure a bridging loop does not occur. Using WDS like this is just simply not practical in many cases. MikroTik created the HWMP+ Meshing system to overcome these limitations, please see the "MESH" section for more information.

The second issue is the slow performance of a WDS system like this. As you go further out and go through more and more access points, your performance degrades very quickly. For example, if we assume that we have a chain of WDS enabled access points, the performance of a client connected to the main or first access point would likely be at 11 Mbps, assuming they have a good of a connection. The second access point though, even though it shows an 11Mbps connection, has only one half of the performance. This happens because the access point is using up one half of the air time to retransmit the client’s data back to the first access point. So now the possible performance is only 5.5 Mbps given a perfect world (no interference, perfect signal qualities, etc.). Now, when we go to the third access point and the actual performance here is cut by one half again. So now the maximum possible performance is 2.75 Mbps. This is the air rate, so if we are using

802.11b, the actual throughput data rate would be under 1.5 Mbps. This may be enough for your solution however; this example is for that's given a perfect world with one client. As you start adding additional clients to the network, this performance can drop very quickly.

WDS Bridged Access Points - Dual Radios

The performance issues that are listed in the above section can be fixed by using dual radios in all of your systems. The main access point will have Station-WDS wireless clients registered to it, and then in the same bridge group you can have another radio with can be a different SSID and channel that will be for clients as well as the next WDS link. If you need to link more access points you would have a third card with a third SSID, which would run AP-Bridge mode with WDS. Repeaters would then have a radio card configured in Station-WDS mode, as well as an access-point for clients. This eliminates the need for the same radio card to do the retransmission to the next node allowing a dedicated radio do this.


AP-Bridge
Station-WDS
AP-Bridge
Station-WDS
WLAN1: Backhaul AP-Bridge/WDS
WLAN2: Client AP-Bridge
WLAN1: Backhaul Station-WDS
WLAN2: Backhaul AP-Bridge/WDS
WLAN3: Client AP-Bridge
WLAN1: Backhaul Station-WDS
WLAN2: Backhaul AP-Bridge/WDS
WLAN3: Client AP-Bridge

This will eliminate the bridging issues, as each backhaul Station-WDS only connects to a specific AP-Bridge. Clients have their own radio cards to connect to on each node, therefore, when they are transmitting, their access point simply receives that data, and then sends it over to the backhaul radio card to transmit back to the main site. No radio care in this configuration actually retransmits what it receives; therefore we have removed the performance issue. Since the Station-WDS nodes are not connecting to multiple AP-Bridges, we can build our network and don't have to worry about dynamically created bridge loops.

Wireless Link Optimization / Best Practices

Over the years of deploying wireless networks, there are a number of best practices that I can recommend. There are a number of things that you can do to improve your wireless links. In general, these optimizations apply to point-to-point wireless links that are fixed on both ends, for example, a tower-to-tower wireless link.

Keep it Simple First

When first setting up a wireless link, start with the most basic configuration possible; set one end in Bridge mode and the other in Station mode. Do NOT put a security profile in place; leave it as an open access point. Don't worry, we will secure it later. The idea here is that you will keep the configuration as simple as possible until the wireless characteristics of the link are optimized.

Hardware Selection

Hardware selection plays an important role in the design of your network. It is ALWAYS better to have a larger antenna and a lower power radio instead of a higher-power radio with a smaller antenna. Antennas amplify in both directions; they have both receive and transmit gain. If you are using a 20 dBi antenna that means the signal that it receives will be increased by 20 dBi AND the power from your radio card at 12 dBi then is also increased by 20 dBi as it leaves the antenna. If you can use a 60 mw radio card and a 20 dBi antenna, it will be better than using a 600 mw radio card and a 12 dBi antenna. Again, plan to use lower transmit power and a higher-gain antenna. Finally is that when designing backhaul links, don't skimp on the CPU power. Go with AH RouterBOARDs instead of the cheapest thing possible!

Radio and Antenna Coax and Selection

From your radio to your antenna, you want the least amount of loss possible. I recommend using integrated radios, where the radio is built right into the antenna, whenever possible. This eliminates transmission-line loss as much as possible. You simply need to run outdoor, UV-rated Cat 5 Ethernet cable to your radio and that's it. It will provide power and data and the antenna is integrated. If you can't use an integrated radio, due to distance or wind loading on your tower, then you will need to either use the shortest possible length of coax as possible or use high quality coax. An example is that I typically would not go more than 20-30 feet with LMR400 cable on 5.8 GHz before I started considering at putting in the use of 5/8 inch Heliax™ coax. We are dealing with milliwatts of power, not watts, so even a few dBi of loss is substantial. Contact your hardware vendor, for more information on specific line-loss levels of each cable offered. You also can find datasheets from cable manufactures to show your transmission-line loss at specific frequencies, so that you can choose the proper cable.

Antenna Alignment

Now that you have the equipment selected, the antennas and coax installed etc., now we need to go ahead and align the link. Again, start with the simplest configuration, no security, a simple SSID, etc. If possible, get the units to connect on the ground before you put up the link. Then go ahead and put up your link. You should have done your link planning prior to putting up the link. You will know in advance what signal levels (in dBm) to expect on each end. If you have a 10-mile link with 19 dBi antennas and 320 mw radios, you should have close to a -70 dBm signal on both ends. When you align the links, first start with your horizontal or side to side azimuth, and once you maximize the signal as much as possible that way, optimize the antenna alignment vertically until you get the best possible signal vertically.

Find Possible Interference

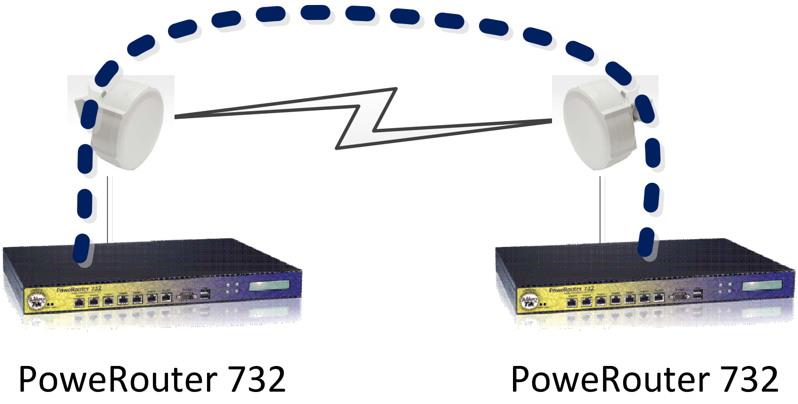
You will now also need to do an interference study. This is simple enough, if you have a spectrum analyzer (if you don’t you need one), hook it to one of your antennas, and let it record the max values across your entire 5 GHz range (if that’s what you are using). You should record this for as long as possible, but the minimum time would be 30-45 minutes. After this recording is done, look at it, then choose the cleanest channel(s). Hopefully, you will have a number of channels to choose from. Now that you have the channels, select a channel that is the cleanest. The less interference, the better! Go ahead and hook your radios back up.

Signal Issues

Now that you have your antenna aligned, what is the signal strength? Is it within +/- 2 dBm of the planned link quality? If it is not, then there may be something wrong. Is it 20 dBm off? If it is, then you may have an antenna in the wrong polarization! Typically the difference between vertical and horizontal polarization is around 20 dBm, and if you are using a single antenna and you are 20 dBm off from what you calculated, then guess what, turn one of your antennas 90 degrees! What if your signal is fluctuating wildly, more than +/- 5 dBm? You may have Fresnel zone issues. Check your link path again to make sure there are no obstructions. Moving one end up or down 5-10 feet may provide you with relief from this issue.

Secure your Link and Testing

When you have a good quality link, your CCQ should be higher than 90% most of the time, now it's time to optimize that link. First, if you are going to run security or encryption, now is the time to go ahead and place your security profile on the link. Next, do some tests, remember don't do bandwidth tests from your link radios. You should have some other bandwidth testing RouterBOARDs or other high-end systems on both sides of your link for testing (see image below). The reason for this is that moving data across the wireless link takes CPU power of your wireless devices, you don't want to add more CPU time by generating data to do the bandwidth test with, and you need to have separate bandwidth testing devices doing this processing.



Minimize Rate-Flapping

Your bandwidth tests will tell you if you have good throughput. If the throughput and CCQ vary during your bandwidth tests, you will need to look at your air rates. If they are changing from 54 Mbps to 48 Mbps to 36 Mbps and so on all of the time while you are doing your transfer, you may have a rate-flapping issue. What you want to do is lock in your data rate to the highest rate that the system is able to maintain consistently. If you move data and it stays 90% of the time at 48 Mbps, then unselect the 54 Mbps air rate. Changing from 48 to 54 Mbps takes time and causes latency issues as well as performance issues. If during your transfer it will sometimes, less than 10% of the time, drop to 36, I would leave the 36 Mbps data rate in there, but remove the lower data rates. This will prevent you from constantly changing data rates and adding latency and jitter to your link. Another suggestion is to increase the hardware retries to 10, to see if you can prevent rate flapping more. This is done in the advanced menu options.

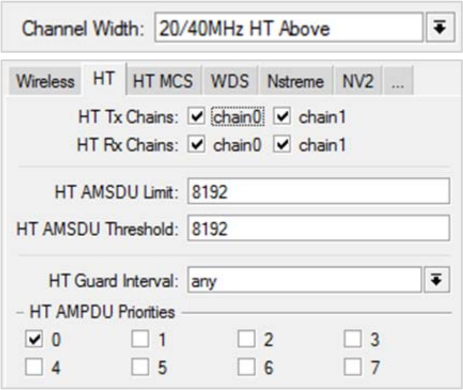
Using Nstreme

Now that you don't have a bunch of jitter in your link, now try enabling Nstreme using the default settings. Remember to do it on your far end first, and then do the end you are at, so that you don't lose connectivity. When the connection has been reestablished with Nstreme, duplicate your bandwidth tests and see if Nstreme will give you more throughput. Sometimes it does, but I have seen times that due to other factors (such as interference) etc., it does not.

Try some variations of Nstreme as well, such as larger frame sizes, and dynamic frame sizes. The goal is to find the best settings for your link and every link will be different. What works on one link may not work on another! One last option for more throughput, is to try using turbo channel sizes. If you can't use Nstreme, try turning on M3P (MikroTik Packet Packing Protocol) as well on both interfaces, this could save you quite a bit of bandwidth as well. There is already some compression done with Nstreme and I have found M3P doesn't really help much with Nstreme turned on.

Improving 802.11n Performance

The much touted 802.11n protocol boasts extremely fast speeds when compared to 802.11a. 802.11n is, designed for MIMO (Multiple-In-Multiple-Out) operations. One of the first things to remember when trying to get an 802.11n link is that it requires at least two antennas. The current generation of radio cards supports 2x2 antenna modes, giving you the ability to attach two antennas to a single radio card.



Channel Width: 20/40MHz HT Above

HT Tx Chains: ☑ chain0 ☑ chain1
HT Rx Chains: ☑ chain0 ☑ chain1

HT AMSDU Limit: 8192
HT AMSDU Threshold: 8192

HT Guard Interval: any

HT AMPDU Priorities
☑ 0 □ 1 □ 2 □ 3
□ 4 □ 5 □ 6 □ 7

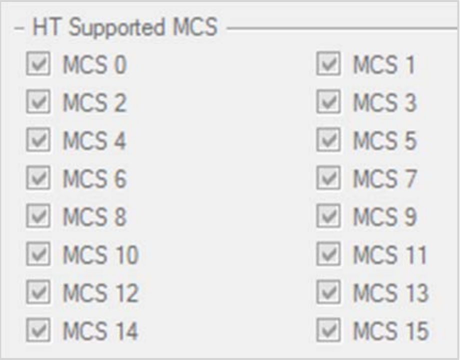
Rate flapping is a big reason for a link to become unstable and slow. In many cases, I would also suggest turning off the 802.11a/g data rates, so that you operate using only on 11n data rates. By selecting only the n rates, you can hopefully continue to optimize your n link. A typical fastest setup for 11n links also includes setting an HT Channel Width (allows for 40 MHz operation), and using all HT TX and RX chains, as shown above.

In these configurations, I have seen upwards of 60 Mbps TCP and 220 Mbps UDP transfer rates between two 802.11n radio cards. I also note that when using dual polarization antennas with good signal strength. It's very common for a pair of MikroTik SXT radios with -60 to -60 dBm signal strengths to max out with 92-95 Mbps of TCP throughput.

AMSDU (aggregated frames and sending using block acknowledgement) or (Frame Aggregation)

The HT AMPDU (aggregated frames and sending using block acknowledgement) is used to allow the 802.11n standard to cut overhead by sending multiple frames per single access to the wireless medium. This is accomplished by combining frames together into a larger frame, kind of like what Nstreme does already. This system can increase the maximum frame transmission size up to 8k. By default RouterOS comes set with only 0 enabled. By enabling more priorities it is possible to increase throughput, but may increase latency. In most cases, where I need a balance of performance and low latency, I leave the AMSDU alone.

802.11n and Data Rates

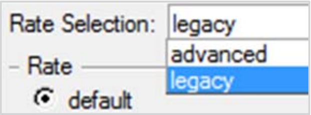


HT Supported MCS

☑ MCS 0 ☑ MCS 1
☑ MCS 2 ☑ MCS 3
☑ MCS 4 ☑ MCS 5
☑ MCS 6 ☑ MCS 7
☑ MCS 8 ☑ MCS 9
☑ MCS 10 ☑ MCS 11
☑ MCS 12 ☑ MCS 13
☑ MCS 14 ☑ MCS 15

802.11n is a bit different when it comes to data rates. In 802.11a/b/g the radio negotiates to the highest possible data rate without errors. As your link preforms retransmits it will eventually drop down your data rate and retry your transmission until it can communicate reliably without retransmits. 802.11n does this as well, but handles the extra spatial stream (or extension channels) differently. By default, RouterOS supports MCS's (Modulation and Coding Schemes) from 0 through 15. This is all MCS's that support 2x2 spatial streams. This is basically the MIMO operation to use the second antenna.

The issue comes in when we don't have a good signal in 802.11n. The example we will use is when we have a signal that only allows us to get to MCS 5. This MCS is using one antenna, not MIMO, but since we have 6 and 7 enabled, we will NEVER attempt to connect at MCS 8-15! We never get to a MCS that offers two spatial streams or MIMO. 802.11n goes by MCS just like b/g/n does; it goes from MCS 0 to MSC 1, then to MSC 2 and so on. But you must note that MCS 5 and MCS 12 is basically the same thing, one with two spatial streams and one with only one.



Rate Selection: legacy

Rate

● default

In v5.6, MikroTik made a change to their rate selection system. They added a new option in the data rates section of your wireless configuration. This rate selection option, allows you to use the old original legacy method of rate selection. The details and information described in the above section all apply with this legacy mode of operation. The advanced mode eliminates the need to disable the higher single stream MCS's in order to be able to get to the dual stream MCS's. It is also supposed to help with the b/g/n mode and getting above 11 Mbps modulations as well.

The 802.11 Hidden Node Issue

802.11 uses a carrier-sense multiple access (CSMA) system to sense when other radios are transmitting on the channel. See the CSMA section under the NV2 topic for more information on CSMA. Using CSMA, every radio clients connected to the same tower, listens for any source of radio-frequency energy on the channel. If no other signals are heard, the radio (it could be either a client radio or an access-point radio) client then starts transmitting to the access point.

Originally, the 802.11 protocol was never designed for long range wireless communications, the way we use it today for fixed wireless operations. It was designed for in-office and short-range use, typically with omnidirectional antennas on all devices, both the access point and clients. When two laptops are talking via Wi-Fi, or (802.11) they will use CSMA. For example, if laptop A is uploading a file to the access point and laptop B wants to start transmitting as well, it uses CSMA to listen to see if the frequency is clear to talk. In our instance, Laptop B would hear both laptop A, and the access point therefore it waits until Laptop A has finished before it to transmits. In the typical our large, outdoor scale wireless networks though, laptop B (or in this case, client B), can't hear laptop/client A because there may be some physical topology in the way such as buildings, trees, or even hills.

Simply put, Client B can't hear client A, therefore client B starts transmitting. Client A is a hidden node to client B, and hence the hidden node issue. If Client B can't hear Client A, Clint B transmits at the same time that Client A is transmitting, and we end up with collisions at the access point receiver. This reduces overall network throughput, and increases network latency. The performance of the entire network is reduced. In some cases this can cause significant issues with access points on fixed wireless networks.

So how do you fix such an issue? For many operators, using 802.11 protocols that include CSMA is simply going to be an issue. You can increase the power and/or transmitting gain of your clients so that they all can hear each other, however, in most terrains, this would be not only impractical, but impossible. You can also move the client's antennas in such a way as all of them can see each other, but I doubt that your clients will be willing to move due to your hidden node issue.

MikroTik has two fixes; the first is a polling protocol called Nstreme. Clients must poll the access point in order to transmit, and the access point then controls who talks when. This does work, but may decreases efficiency, as every time a client wishes to transmit it must poll. The latest workaround is NV2. NV2 uses timeslots and the access point scheduler to give clients times when they can transmit. Each client must wait their turn and, based on the bandwidth needs of the client and other factors, the timeslots are dynamically allocated. In this method, the access point controls when the clients can transmit, thus preventing the hidden node issues.

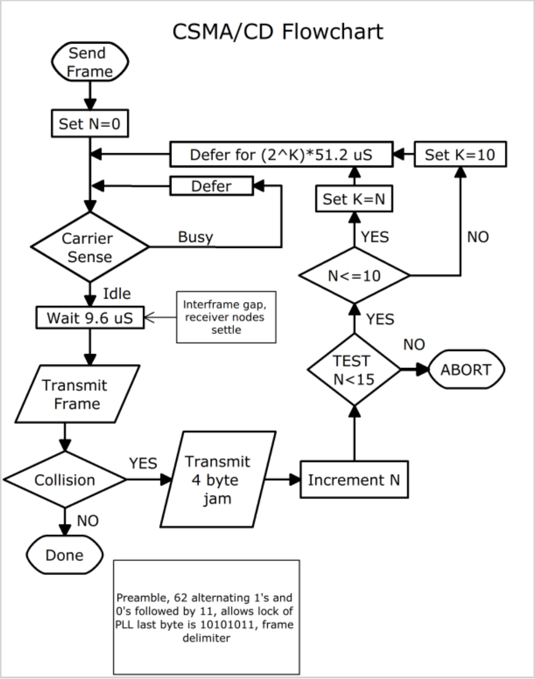
Keep in mind that any type of polling or NV2 protocol is a departure from 802.11 standards. We are no longer using 802.11 as the protocol; we are using something new, something hopefully better to prevent our hidden node issues and increase overall network performance. With RouterOS, both Nstreme and NV2 are proprietary, so you will have to have both RouterOS access points and RouterOS client radios with Nstreme (or NV2) enabled on all radios. If you have other clients that follow the 802.11x standard, those clients will not be able to connect to the access point when it changes from the 802.11 standard to either Nstreme or NV2.

NV2

NV2 is a new wireless time-division multiple-access (TDMA) protocol introduced in v5 of RouterOS. This protocol is proprietary, so you will need to have RouterOS on both ends of the connections. This system uses TMDA or Time Division Multiple Access technology instead of the original CSMA system that 802.11 devices uses.

CSMA – Carrier Sense Multiple Access

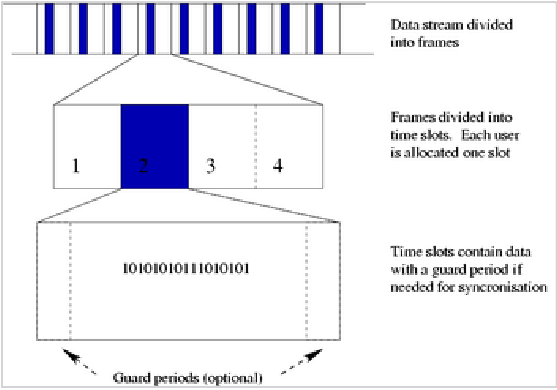
It is important to understand the differences between the protocols and access methods when we start talking about NV2. I wanted to make sure I put some information in to help you understand these differences. Using CSMA, we use the first two words, "carrier-sense" literally mean to listen before transmit – to listen, we use carrier sensing to determine if another carrier is present before the transmitter can transmit. The radio listens for another "carrier", for example, another radio transmitting. If any carrier or signal from another radio is not heard, then the transmitter can start transmitting.



In short-range indoor networks where every station can hear every other station this works quite well. In large outdoor wireless networks such as WISPs, we have an issue called "Hidden Node". This issue is described in more detail in the previous chapter. This becomes a major problem when CSMA Wi-Fi network protocols are used at long range. Clients connected to the same access point can't hear each other due to obstructions, such as buildings, hills, trees, etc. Therefore, the carrier sensing simply can't hear if another radio is transmitting. This causes many issues, including collisions, that overall reduce the network throughput.

TDMA – Time Division Multiple Access

NV2 is the new access protocol that uses TDMA. Instead of sensing transmissions, which we know has problems in long range 802.11 networks; the available air time is divided into different time slots. Each user, gets pre-assigned time slot, and transmits only during their timeslot thereby, preventing collisions on the network. In outdoor, long-range wireless networks, the kind that RouterOS uses, this is a big advantage.


Data stream divided into frames
Frames divided into time slots. Each user is allocated one slot
1
2
3
4
10101010111010101
Time slots contain data with a guard period if needed for syncronisation
Guard periods (optional)

NV2 of course, uses the TDMA method of access, removing some of the limitations of the older CSMA style networks.

NV2 Hardware & TDMA

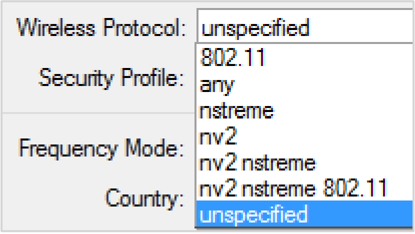
To use NV2, one of the first things you need to remember is that you must have a newer Atheros-based radio card, anything starting from the AR5212 will work. NV2 does not work with the older AR5211 and AR5210 chips; you may have to upgrade your hardware in order for NV2 to work. You can use both 802.11n and older 11a or 11 b/g devices; however they just simply must use the newer chipsets.

As we said, NV2 uses TDMA, where access control is done by the access point. The NV2 access point will divides up the time into a fixed time slot periods. This is done dynamically, based on the queue state on both the access point and the clients. The Timeslot allocation is done in both directions, both for downlink and uplink. Uplink or (client to access point data) is further divided between the connected clients based on their bandwidth requirements. Each period or cycle, the access point will transmit a schedule that tells the clients how and when they should transmit and how much time they have.

Because of the access point schedules the transmission times for all connected clients, as well as leaving some "unspecified' time for new clients, we eliminate the hidden node issue that is common with CSMA networks therefore network performance is increased. We also increase performance in several ways as well; network overhead is reduced because of the propagation delay overhead, this would be the per-frame ACKs, as NV2 also does frame aggregation. NV2 also has advantages over the original Nstreme protocol because, we are not polling each client and that reduces the polling overhead. There is also, more control over the latency, with adjustable period sizes and QoS features.

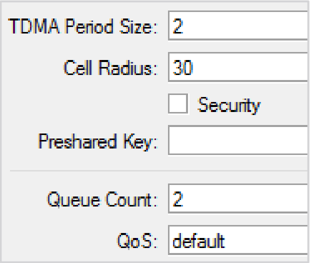
Configuration of NV2

One of the key improvements that RouterOS implemented since v5rc1 is a new interface setting called *wireless-protocol.* I know that does not sound exciting, however this setting allows us to control what wireless protocol to use, and guess what, we use it to set up NV2 if we wish to have it.



Wireless Protocol: unspecified

So to setup an access point for NV2, first, you will change its wireless protocol to NV2. Other options such as "*nv2 Nstreme 802.11*", allow the client, to search for a NV2 access point first, if not found then move to a Nstreme access point, and then finally to a 802.11 access point. There are other options such as the NV2-Queue-Count, etc., that we will not discuss here.



TDMA Period Size: 2

Cell Radius: 30

□ Security

Preshared Key:

Queue Count: 2

QoS: default

Securing NV2

Securing your NV2 access point is a bit different than with your original 802.11 system. A new TAB in your wireless interface called NV2 gives you all of your NV2 Options, including your security settings. Before NV2, there was a security profile under your wireless tab that allowed you to specify what security protocol you wish to use, however, using NV2, we have a security option to enable security, as well as a Pre-shared key to use. The security profile on the wireless tab is NOT used in NV2.

NV2 implements its own security using the pre-shared key that you have placed on the NV2 tab. The NV2 security system includes hardware-accelerated data encryption using AES-CCM with 128-bit keys, and pre-shared key authentication. It will periodically update the group keys, and also has a four-way handshake for key management very similar to 802.11i. To configure security, simply enable the security by checking the security box and then setup a pre-shared key in the NV2 tab. These settings are for NV2 ONLY.

NV2 QoS

NV2 also has QoS built into it; it will work with your Firewall/Mangle, VLAN priorities and/or MPLS EXP bits. The built in QoS scheduler will allow you to specify what data to prioritize. It also has a variable number of priority queues. The QoS policies that are defined are controlled by the access point and the clients adopt the policy from the access point.

You will configure the NV2 QoS system by using the QoS settings under the NV2 tab. The *NV2-QoS* setting specifies what kind of frame priority it should be using. By default, the system will use built-in rules and provide QoS based on the built-in QoS policy algorithm. This algorithm selects queues based on the packet type and size. If the built-in rules don't match, then it will use the frame priority field.

The second option is the Frame-Priority field. This bypasses the built-in QoS schedule for your own. You must set the Frame-Priority field by processing it in either Firewall rules, ingress priority or by the frame forwarding process, such as MPLS EXP bits. The queue is selected by the frame priority done by the 802.1D recommendations.

Migration to NV2

With the introduction of the wireless-protocol setting, we have a new feature that will allow us to migrate to NV2 quicker. Using the wireless-protocol setting we can specify what wireless protocol to use, but we also have the ability to select any. With the ANY option, we have the ability to quickly migrate from 802.11 to NV2 quickly and effectively.

The simple procedure is as follows:

- Upgrade your access point to a version that supports NV2, but do not turn it on.
- Upgrade your clients to a version that supports NV2, but these times, configure your clients with either ANY, or NV2-nstreme-802.11 options.
- Configure any security-related settings on both the clients and the access point.
- Change your wireless protocol to NV2 on your access point
    - If you have an issue, simply change it back to 802.11 and all clients will connect to the 802.11 access point.
- Tune NV2 for settings.
- Implement QoS.

Troubleshooting Wireless Links

Low Signal

When you design your wireless link the first time then, and based on your path analysis you should have calculated your link budget. This link budget will tell you what signal strengths you should have at each side of the link. This typically is accurate to 1 to 2 dBi! If your signal is not within a few dBi, in other words, if it is in excess of 4 dBi off, I would recommend looking at your link further. The number one reason for low signal is antenna miss-alignment. Follow the instructions in the New Link section to check your antenna alignment. The second reason is for low signal is usually Fresnel zone encroachment. Again, your path analysis should show if you have something in the Fresnel zone that can block part of affect your signal. Finally, on a new link, simply a bad radio card or antenna connector can cause low signal levels. I highly recommend using a pre-built, known-good pair of RouterBOARDs that you can hook up to your antennas that are tested with signal strengths to ensure that you don’t have this issue in the field.

On an old link, if your signal level has gone down, then you would need to recheck your antenna alignment. Antennas can move over time, usually due to lose bolts over time.

Wandering/Fluctuating Signal

A wandering signal or fluctuating signal would be +/- 4dBi of signal level change within a few minutes. If you just installed your link and the signal is changing wildly, again, 2dBi +/-; and then I would go ahead and look for possible causes. If this is a new link, then I would first look for Fresnel zone issues. Move one side of the link up 5-10 feet and see if that improves the stability of the signal. If this is an old link that just started to have this issue, investigate and see if the issue could have started with a recent rain or freezing weather. If so, then chances are you have a water-intrusion issue. Remember, those N connectors and cabling ends needs to be wrapped extremely well to prevent water from getting into them.

Bad CCQ

Bad CCQ can be a result of a low or wandering signal, so check those first. When your CCQ starts going down, see what air rate you are connected at, again, Fresnel-zone issues could be the culprit. Make sure trees have not grown in your path, or buildings built in your path. Don't laugh, it happens! If your CCQ is low even with minimal or no data running across the link, then this typically is always a signal issue, something about the signal is creating a change in the link quality, and troubleshooting that is the first step. Interference is also a good possibility, as a new link could have gone up and may be causing many retransmits on your link. Use in a spectrum analyzer and test for this once you have done everything else. Remember you want to use a larger antenna with a tighter antenna pattern to minimize interference from other links.

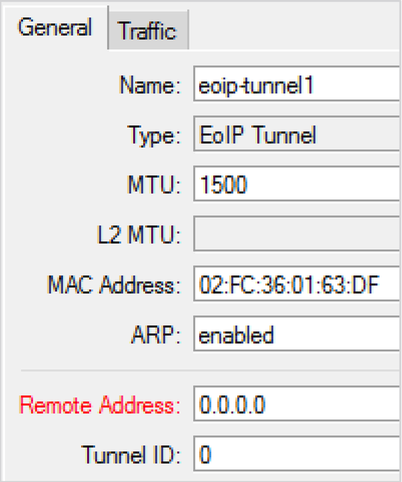
Tunnels

RouterOS offers many different types of tunneling options. Some of these you can bridge and some you cannot. Tunnels that you can bridge are Layer 2 tunnels. My experience though, shows that you will always have a better performing network if you use Layer 3 tunnels. Tunnels you will route through reduce your network overhead and minimize the size of broadcast domains. Also, Layer 3 tunnels provide routing capabilities, so you can really control traffic on each segment, as well as provide queuing, traffic shaping and QoS.

Some tunnels also encrypt traffic, and that encryption can be simple or very advanced. RouterOS can do from MPPE 128 Stateless encryption, very common for home VPN connections, to AES-256 bit encryption. Some of the tunnels however, do not encrypt traffic or have an option not to encrypt traffic. I use a rule of thumb to keep encryption to a minimum; this also keeps the load off of your RouterOS CPU as well. An example would be for most site-to site-traffic, which does not deal with private personal data and/or credit card information; I would suggest just using the MPPE 128 encryption. Typically this provides enough encryption to keep that private data private. IF you are transmitting credit card information, first it should be encrypted by whatever method you are transmitting it before it hits any types of tunneling, but you may wish to bump that up to something like 3DES or AES-128. But if you want the most encryption you can get, you can do an IPsec tunnel inside an encrypted L2TP tunnel. So, you encrypt with AES-256 or 3Des, and then hit the tunnel, that encrypts the already encrypted data with MPPE 128.

EoIP

EoIP or Ethernet-over-IP (EoIP) tunnels are proprietary to RouterOS. These give you a very quick, unsecured method of creating a Layer 2 tunnel. To create an EoIP tunnel, you simply need two MikroTik systems that can communicate directly with each other. EoIP will use IP Protocol 47, more commonly referred to as GRE for the communication between the two sites. EoIP is not a replacement for WDS in wireless bridging as well.

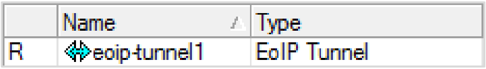


Name: eoip-tunnel1
Type: EoIP Tunnel
MTU: 1500
L2 MTU:
MAC Address: 02:FC:36:01:63:DF
ARP: enabled
Remote Address: 0.0.0.0
Tunnel ID: 0

Even though EoIP is not encrypted, it can run on top of other tunnels. An example would be an encrypted MPPE 128bit PPTP tunnel, as well as any other connection that uses TCP/IP. To use a PPTP tunnel, first setup a PPTP tunnel and set it to use encryption. Now create your EoIP tunnels, and use the remote address of the PPTP interface on both ends. This will force the tunnel to go through the PPTP tunnel, thus, encrypting it. This method does work, however, look in the PPTP section, as you can now simply bridge the PPTP interfaces instead of setting up two tunnels.

To create an EoIP tunnel click Interfaces → Plus Sign → EoIP Tunnel. This will create a new interface that you can apply filters, queues, and setup routing on. The only two items that you need in the interface settings is your Remote address, this would be the remote IP address of the remote end, and the tunnel ID number. This number must be the same on both ends. Once you create the two ends, now you have a tunnel. You can at this point, place IPs on each end, and setup routing. You can route across an EoIP tunnel if you wish, but most people would use it for what it is intended for, and that is for a transparent bridge.

One thing that I want to point out, and one reason I do not use EoIP tunnels much, is that the interface, regardless of its actual status, always shows running. This means that you will not have a state change, or other identification that

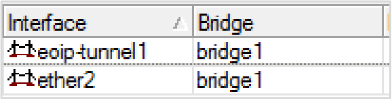


	Name	Type
R	eoip-tunnel1	EoIP Tunnel

shows that the interface is down. It never goes down, and hence, anything that is based on the interface never changes or fails over due to this fact. Also, unless you pass data to the other side you will not know if the link it working or not.

Bridging an EoIP Tunnel

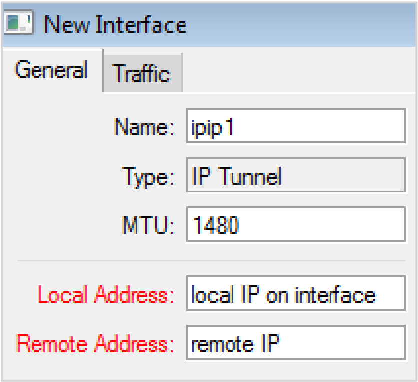
Creating a bridge on an EoIP tunnel is super easy. Since the interface was designed to be a bridge, you will only have to add it to a bridge group, to bridge it. In the example to the right, you will see that an EoIP tunnel interface is in the same bridge group with an Ethernet port.



Interface	Bridge
eoip-tunnel1	bridge1
ether2	bridge1

One major issue that you may have with EoIP links is MTU. Typically when you bridge Ethernet across the Internet, if you have a good Ethernet connection, you won't have issues; however, if you go through things like a PPPoE-Client, you may have to adjust the packet sizes of your tunnel. By default your tunnel MTU will be 1500, and this is fine for Ethernet, but may not be optimum for use over the Internet. MTU issues are often difficult to troubleshoot. Common signs are HTTPS and other very specific websites are not working (assuming you are going through the EoIP tunnel to get to the Internet) as well as large ping packets are not getting though. To fix this, you will simply need to change the MSS size on large packets to be smaller than the max MTU that the devices between your two routers can support.

IPIP



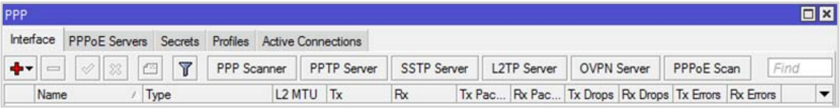
Name: ipip1
Type: IP Tunnel
MTU: 1480
Local Address: local IP on interface
Remote Address: remote IP

IPIP is IP inside IP. It simply encapsulates IP packets inside other IP packets. IPIP, unlike EoIP, is a standardized tunnel type and is used by other router vendors. IPIP, like EoIP, is very simple to setup and can run inside another tunnel if you require encryption, but does not offer encryption by itself. IPIP also, does not show an interface state. Once you create the interface, it will always show as "up" regardless of the other side of the tunnel. You will have to implement other kinds of checking, such as pinging or ARP to verify that this tunnel is running.

To create the IPIP interface, click on Interfaces → Plus Sign → IP Tunnel. Once you get the new interface screen up, you will have two IP addresses to enter. One is the local IP address of your router. Typically this IP is the IP address of the closest interface to the remote router. This could be any address on that interface though. The remote address is the IP address of the remote router. Once you create both ends, I would place IP addresses on them, and ping across the tunnel to verify its operation. You will need to route data across the IPIP Tunnel as it is not designed to bridge.

PPP System

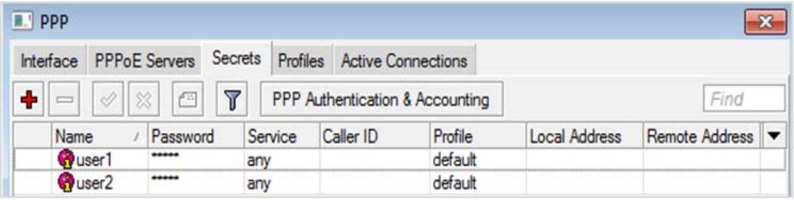
RouterOS offers a full PPP Server/Client system. This point-to-point system includes other protocols as well, such as PPTP, L2TP, PPPoE, and even OpenVPN. It also supports the PPP Server and Client. To access the PPP system, click PPP in WinBox.



As you can see, we have quite a few options here. The important thing here is that there are a number of tabs that are common to several different systems. The secrets, profiles and active connections tabs are all shared by the PPP System and each of protocols uses these tabs. The PPP System uses four authentication modes, as well depending on the protocol and service. What is important to note is that the PAP method is not encrypted or secured, when in doubt, disable this method.

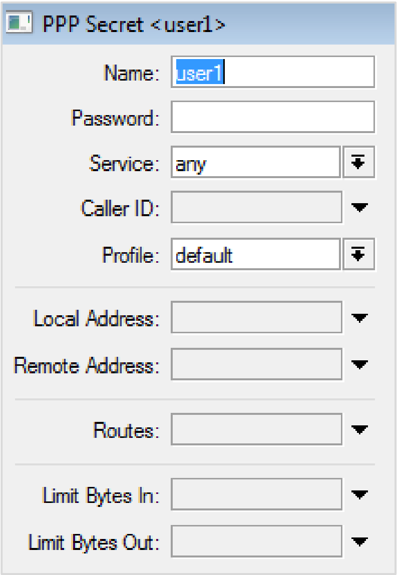
PPP Secrets

The PPP Secrets section is for the creation of PPP shared-user accounts. These accounts are basically a local authentication database for the PPP protocols. These accounts have many options where you can setup what username/password they have, what service they use, as well as if they must call from a specific IP address. It also gives you options for the local and remote address, but this can be specified inside the profile that they use. We also have the ability to add a route when this PPP secret is used. This can be used if you are using an IP pool in the profile. You will not know the IP address that will be assigned to the PPP user, but regardless, using the route here, will add a route to the IP that the PPP user has been assigned.



Name	Password	Service	Caller ID	Profile	Local Address	Remote Address
user1	*****	any		default		
user2	*****	any		default		

Also on the secrets tab, you will have an option for PPP Authentication and Accounting. By clicking on this button you can access get into the Radius information for PPP. By enabling Radius and accounting information here, the PPP system will use a Radius server to attempt authentication of the PPP user. By default, the system will always look at its own local database first before sending it out to the Radius system. This Radius system could be a billing system, or even Internet Authentication Services with Active Directory. There is more to configure however, you will need to setup a Radius server with the PPP service as well for this to work.



PPP Secret <user1>

Name: user1

Password:

Service: any

Caller ID:

Profile: default

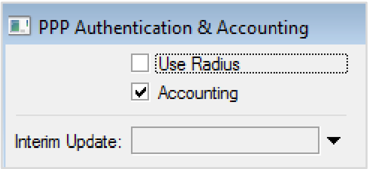
Local Address:

Remote Address:

Routes:

Limit Bytes In:

Limit Bytes Out:



PPP Authentication & Accounting

☐ Use Radius

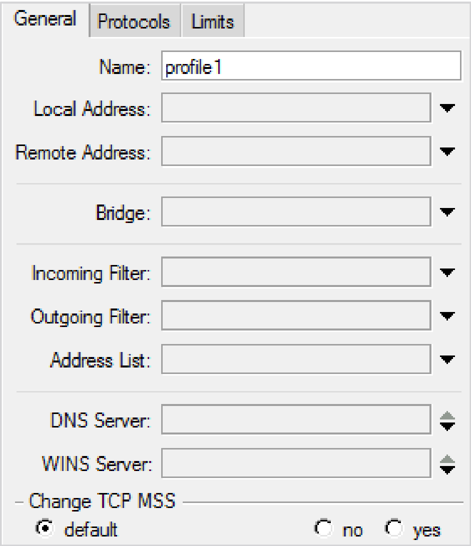
☑ Accounting

Interim Update:

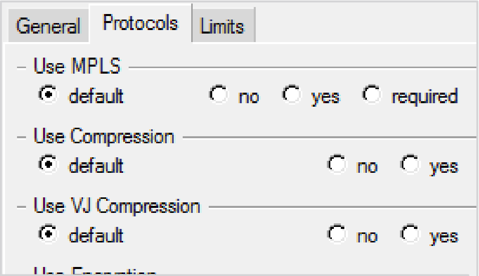
PPP Profiles

Once you create PPP Secrets with usernames and passwords, you also have the ability to point that user to a PPP Profile. The profiles are used to group common items that PPP clients need into one profile. An example would be for PPTP VPN clients. These clients need to get an address from a pool of IP addresses, and specific DNS servers for your active directory system. You can also wish to require them to encrypt your data via MPPE128.

To configure your PPP Profiles, you will click on PPP → then in the PPP windows select the PPP Profiles tab. Two profiles come by default and can't be removed. The default, allows no encryption and the default-encryption forces encryption. You can create as many of these as you wish. The remote address is where you typically will specify the IP Pool you wish the clients using the profile to get their IPs. In the case of a Windows Server System, you can add your DNS servers to point them to the Active Directory DNS server. You can also configure if you want compression and encryption.



Name: profile1
Local Address:
Remote Address:
Bridge:
Incoming Filter:
Outgoing Filter:
Address List:
DNS Server:
WINS Server:
Change TCP MSS
● default ○ no ○ yes

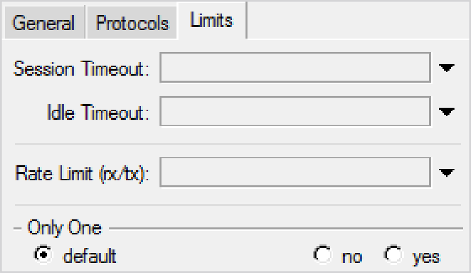


Use MPLS
● default ○ no ○ yes ○ required

Use Compression
● default ○ no ○ yes

Use VJ Compression
● default ○ no ○ yes

Use Encryption

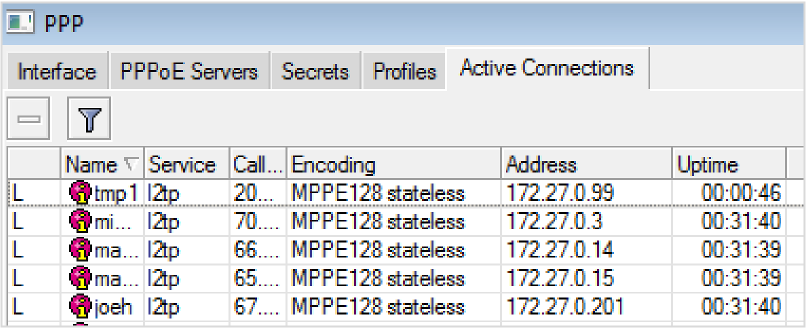


Session Timeout:
Idle Timeout:
Rate Limit (rx/tx):
Only One
● default ○ no ○ yes

Another option here is the ability to change your TCP MSS or Maximum Segment Size (MSS). This is mostly important if you are using PPPoE and need to reduce your packet size to allow for the PPPoE header information.

PPP Active Connections

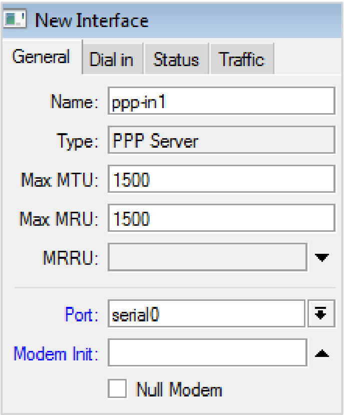
The active connections tab is very straight forward. It will show all of your current active sessions for your PPP System. This includes PPPoE, PPTP, as well as L2TP connections. You have the ability here to highlight one of these and click the minus; this would disconnect that PPP User. They may come right back if their system auto redials right away, but this would be how you could remove a client from being connected.



	Name	Service	Call...	Encoding	Address	Uptime
L	tmp1	l2tp	20...	MPPE128 stateless	172.27.0.99	00:00:46
L	mi...	l2tp	70....	MPPE128 stateless	172.27.0.3	00:31:40
L	ma...	l2tp	66....	MPPE128 stateless	172.27.0.14	00:31:39
L	ma...	l2tp	65....	MPPE128 stateless	172.27.0.15	00:31:39
L	joeh	l2tp	67....	MPPE128 stateless	172.27.0.201	00:31:40

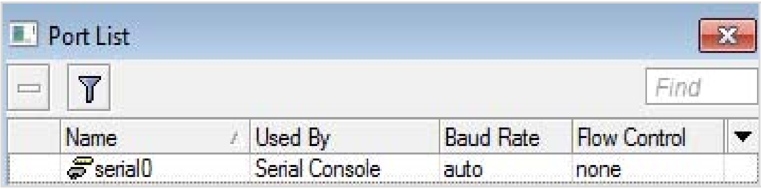
PPP Server

The PPP Server and Client are used to create PPP connections. The main usage for the PPP Server is to be able to establish a PPP connection using a modem of some type. Typically a dial-in modem would be used. To do this, you will have to create a PPP Server. Click on Interfaces → Plus Sign → PPP Server to create this new interface. Once there, you will have to specify the Port and modem init. You can also specify if you are going to use a null modem cable or not. Typically for a modem, you would not.



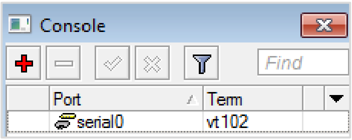
Name: ppp-in1
Type: PPP Server
Max MTU: 1500
Max MRU: 1500
MRRU:
Port: serial0
Modem Init:
☐ Null Modem

One of the challenges when using this method is that the existing serial0 is typically used for the console. If you are using a RouterBOARD or other hardware with only one serial port, you will have to remove the console from the serial port.

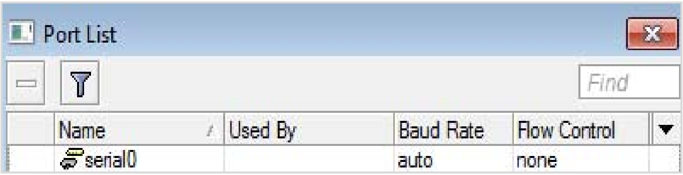


Name	Used By	Baud Rate	Flow Control
serial0	Serial Console	auto	none

Above you see the port list; you get this by clicking the PORT button in WinBox. This port button shows you your serial interfaces and ports. Note in this example, it shows that serial0 is in use by the "serial console"; this is your console port for RouterOS! We will have to remove the serial console in order to use the Serial interface for PPP Server. The process for this is to click on System → Console. This will give you the console options. Note in the screen shot to the left, we have the serial0 port is using the emulation type vt102. We will simply need to remove this, so that our serial port will be unused. Highlight the serial0 item here and then click the minus to remove it.

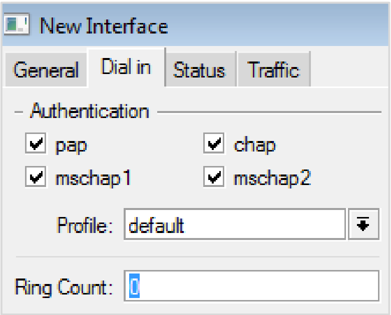


Port	Term
serial0	vt102



Name	Used By	Baud Rate	Flow Control
serial0		auto	none

Now you will see that the port list does not have a "used by" value next to your serial interface. This shows that you have freed the port. Now finish configuring your PPP Server interface. You will need to configure your modem init string. Typically this would be ATZ, to issue a modem reset, and then the default configuration of your modem would be set to auto answer. If you do not have the default configuration set you can also use ATA0, however, refer to your user manual for exact auto answer commands.



Authentication

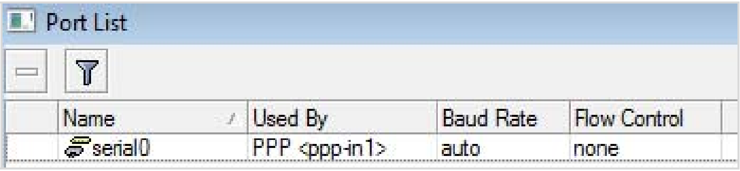
☑ pap ☑ chap

☑ mschap1 ☑ mschap2

Profile: default

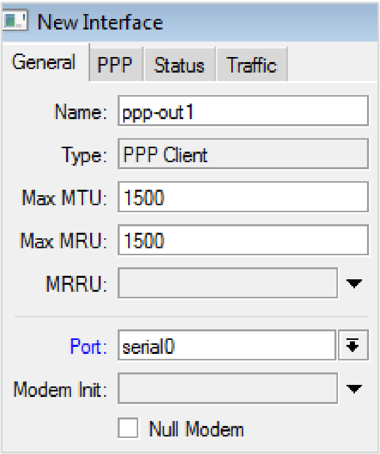
Ring Count: 0

Under the Dial-in tab, you can specify what profile you wish to use. Think of this, not as a serial console, but as a method to get an IP connection via a modem. The profile will specify the IP information as well as other information, and you can use Radius to login as well. Since you are using this mostly for remote access, I would use a local PPP secret to connect. Something else to keep in mind is that you can specify a MRRU; you enable MP or Multilink-PPP. This will allow you to use several serial ports to bond speeds if you wish. I typically use PPP Servers to allow out-of-band access to your routers via phone lines, so typically you do not need greater speed than the phone line allows. Once you configure this interface and apply it, you will see that your serial port is in use by your ppp-in interface.



Name	Used By	Baud Rate	Flow Control
serial0	PPP <ppp-in1>	auto	none

PPP Client



Name: ppp-out1

Type: PPP Client

Max MTU: 1500

Max MRU: 1500

MRRU:

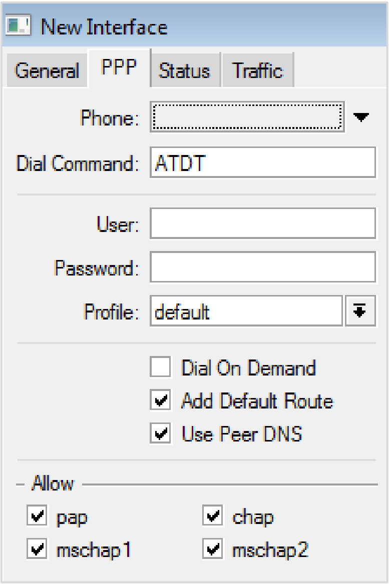
Port: serial0

Modem Init:

□ Null Modem

The PPP Client is used to dial some connections. ISDN modems would be another example of using PPP Client. The PPP client will have to have a free port as well, just like the PPP Server, however, with the PPP client; you can also use other forms of modems, such as 3G or Cellular data cards. First, you will need to free up the serial port. Once that is done, then you can create your PPP Client. Do this by clicking on Interfaces → Plus Sign → PPP Client.

If you do wish to use a modem init string, you will need to place it in here. Most modems attention and reset command will be ATZ, however, refer to the modem manual for the proper commands. If you are using a null-modem cable, you would specify it here as well.



Phone:

Dial Command: ATDT

User:

Password:

Profile: default

□ Dial On Demand

☑ Add Default Route

☑ Use Peer DNS

Allow

☑ pap ☑ chap

☑ mschap1 ☑ mschap2

On the PPP tab you will have the rest of your options. Specifically the phone number you wish to dial, the dial command of your modem, and the login information. If you specify the Dial On Demand option, this will only dial out once a request has been made. You will typically need to specify both a User and Password as well as a method of authentication to be sent. Remember, PAP is unsecured, so when in doubt, don't use it. If you are using this as your Internet access, you will need the default and peer DNS.

One of the common usages for this is with an out-of-band cellular data card. I use these cards along with the PPP client, to have out-of-band access to core routers. This works quite well, and will give you a backup method to get into your router. When I do this, I do not use the peer DNS or default route options, but I do leave the dial on Demand unchecked as I want the connection to be up all of the time. I also will use a tunnel of some type as these types of connections typically do not offer static IPs. I force this tunnel I force out the PPP connection, so that I have remote management IPs for all of the core routers via tunnels to my main connection or main office.

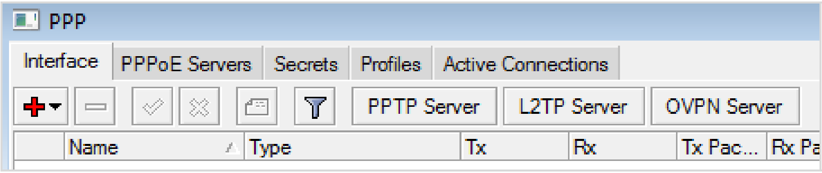
Using PPP Client with a Cellular USB Card

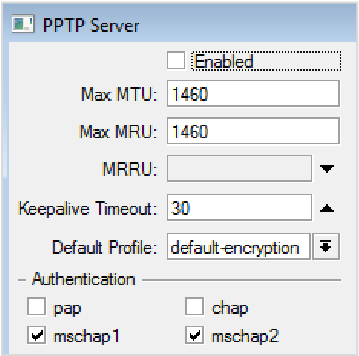
Start by referring to the PPP Client section, as this will give you some insight on how I use these connections. Since we are using the USB port, we will need to ensure that RouterOS knows how to handle this card, including if the drivers included in RouterOS include the drivers for your card. Refer to MikroTik’s website and the list of supported hardware if you are unsure if your card will work.

Most carriers will offer configuration guides to get connected without using their software, typically it’s just a simple PPP connection. In the US, Sprint requires simply has to have a dial command using the phone number of #777. Other carriers will differ so you will need to have the correct dial-in number for them. Information can be found either by contacting the carrier or on-line.

L2TP/PPTP Servers

I combine the L2TP and PPTP systems together, because the setup is virtually identical. Each protocol is a bit different; both use GRE protocol 47 to establish the connections; however the PPTP system is TCP-based while L2TP uses a UDP stream. Which is better? I get asked that quite a bit. PPTP is more common, and due to it using TCP, it should be more reliable, however, I have seen better performance with L2TP connections on lossy or other high-latency applications. If I had to make a recommendation, I would use PPTP.




PPTP Server

□ Enabled

Max MTU: 1460

Max MRU: 1460

MRRU:

Keepalive Timeout: 30

Default Profile: default-encryption

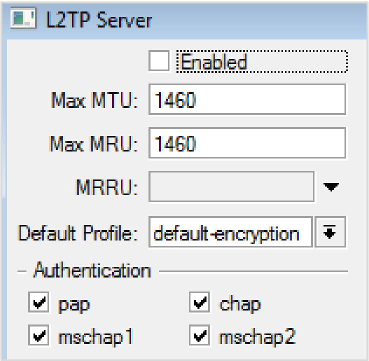
Authentication

□ pap □ chap

☑ mschap1 ☑ mschap2

To setup either of the PPTP or L2TP servers, you will need to enable them. Under your PPP menu, you will have options for both servers. The options are virtually identical with the exception of the keepalive timeout on the PPTP server. You will need to Click on the enable check box to start the server. Here, it also allows you to specify what authentication method to use you wish to allow and what the default profile to use. By enabling this, you effectively turn this on for all IPs on the router.

As you can see, the configuration for the L2TP Server is very close to the PPTP Server configuration. Remember that by enabling this server, you turn it on basically on every interface and every IP that comes into the router.



L2TP Server

☐ Enabled

Max MTU: 1460

Max MRU: 1460

MRRU:

Default Profile: default-encryption

Authentication

☑ pap ☑ chap

☑ mschap1 ☑ mschap2

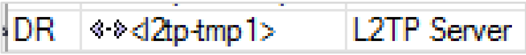
These servers will use the username/passwords through the PPP Secrets system, or Radius. If there is profile information in the local user account, it will be used it. If you do not have any profile information in the Radius server, then the default profile for the server will be used. Hence the need for the default profile on the server. You can also select what type of authentication that you wish to have on the L2TP server as well.

Windows PTPP VPN Users

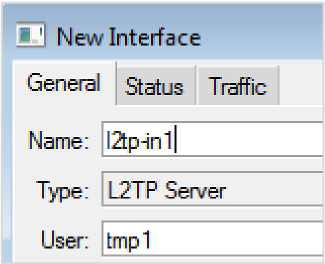
Since Windows 98, we have had a built -in PPTP VPN client. By default, this client uses by default PPTP, but on newer versions it can also use L2TP. The Windows VPN client will connect to the PPTP server of RouterOS without issues. You will have to issue an IP and hand out DNS to the client, but this type of VPN connection is extremely common. Most of the time, when a user says they VPN into their office or work, they are using a PPTP connection. There is no special configuration in Windows for this to work.

L2TP/PPTP Server Interfaces

When you enable the L2TP and PPTP servers, there is no interface that is created, and you normally do not need these interfaces. As clients connect and disconnect, the interfaces will be automatically created and removed. These interfaces will be dynamic, and normally this works perfectly fine. For VPN users, that is using PPTP this is very common and does not present any type of issue. However for Site-to-site communications, it's sometimes desirable to create Firewall and NAT rules based off of the interface. Using the L2TP/PPTP server interfaces, you can create a static interface that comes up and down depending on if the proper user is connected. This will give you the ability to create rules based on that interface. If you don't do this, what will happen is your rules will work, until that user disconnects. When that occurs the interface is no longer there, and your rules become invalid. Even when the client reconnects, and a new interface is created, that new interface is not matched and your rules will remain invalid.



DR <-> <l2tp-tmp1> L2TP Server



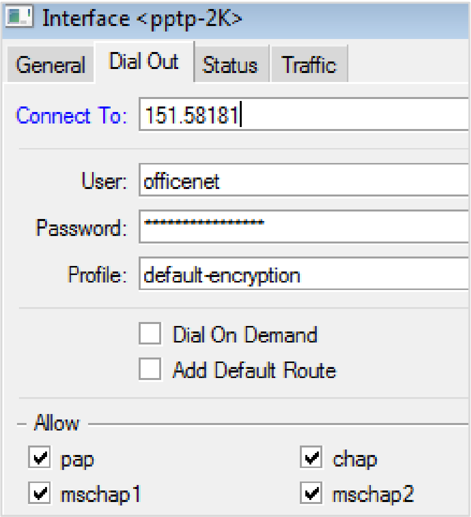
Name: l2tp-in1

Type: L2TP Server

User: tmp1

To create a static interface, simply click on Interfaces in WinBox→ select the Interface Tab → click the Plus drop down Box → Select either PPTP or L2TP Server. This will create a new interface, and the only option is the name and the user. The user is the username that you wish to associate with the server interface. The remote site would use a username/password stored in PPP Secrets normally to connect to this server. That username would go into the user box. Once created, if there is already a connection up that is dynamic, shown by the D next to Interface, then you will need to remove that active connection. Once removed, that connection should come back and when it does it will put an R, for Running, next to your new server interface. This shows you that that server interface is running. If that client disconnects, the interface will go hard down. This gives you a state change for your routing protocols, but does not remove the interface so that your rules will still work when that interface comes back up.

L2TP/PPTP Client



Connect To: 151.58181

User: officenet

Password: *****************

Profile: default-encryption

☐ Dial On Demand

☐ Add Default Route

Allow

☑ pap ☑ chap

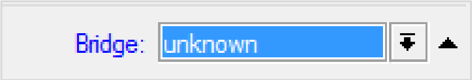
☑ mschap1 ☑ mschap2

Unlike enabling the PPTP or L2TP Servers, the clients are interfaces. When you create one of these client interfaces, you will have to put in all of the information necessary to have that interface establish a connection to the server. In this case, on the Dial-Out tab, you will find the IP address that you will need to connect to, as well as the username/password and the profile that you want to use on the client. You also can set the Authentication method you wish to send and as well as install a default route.

Note that you also have a profile here. The profile on the client side is typically used to direct compression, and encryption information. The default profile that comes installed with RouterOS will tell the client to follow the profile that the server has installed. Typically the server side profile will contain other information, such as IP addresses, what kind of encryption/compression is required, and other variables. The client is typically just receives this information and follows.

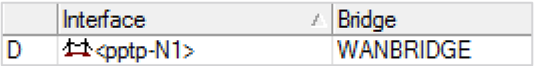
Bridging PPTP

RouterOS has begun to offer the ability to bridge your PPTP VPN connection. This will allow you to create a direct Ethernet bridge, and allow you to pass Layer 2 Traffic across your encrypted tunnel. This only works in PPTP and not in L2TP. You will start by simply creating your VPN just like you would if you would route your tunnel. Create your profiles on both sides, with one exception. In this bridging profile you will need to select a bridge. This bridge is the bridge that when your PPTP tunnel comes up, it will automatically add the PPTP tunnel into your bridge group for you. You will need to select this on both sides of your PPTP link. Once this is done, enter your PPP Secret, and create your interfaces. I would suggest using PPTP Server to create a static interface on your server side. When the interfaces come up, they should drop the PPTP interface into the bridge group dynamically, and you should be able to pass traffic across your tunnel.



Bridge: unknown

In some versions of RouterOS you may see unknown interfaces under the bridge group as dynamic entries. These dynamic entries may be red. Upgrading to the latest version should fix any of these entries. Even if they say “unknown”, they will typically still work. If you check in the command line, it very well may show correctly. This simply was a display issue in WinBox vs. an actual issue with the protocol.



	Interface	Bridge
D	<pptp-N1>	WANBRIDGE

For those of you who have tried this, don’t forget that you are going to have to increase your MRRU to 1600, I typically also decrease my MTU and MRU to 1200 or 1400. You will also need to setup your MRRU in your PPTP server as well for it to be able to pass 1500-byte packets.

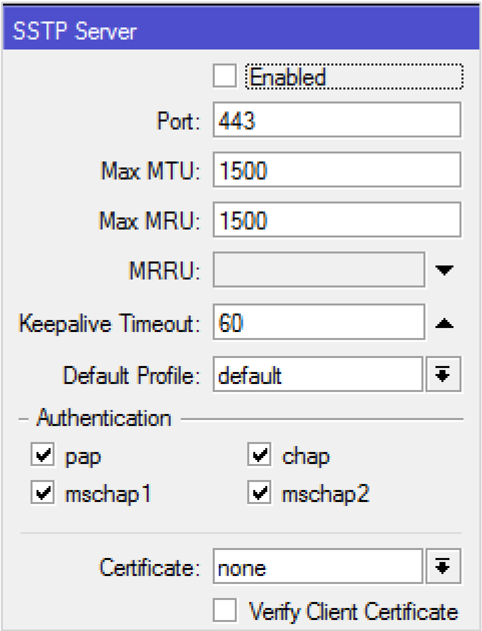
SSTP

SSTP or Secure Socket Tunneling Protocol allows you to transport a PPP tunnel inside a SSL channel. This also gives you the ability to bypass many proxy server and/or Firewalls as it uses TCP port 443 for communication. RouterOS typically follows standards; however, it did add the ability to communicate on the SSL port without valid certificates, or any certificates for that matter. This only can be done however, with MikroTik at both ends, both client and server.

SSTP Server

The server setup is very simple; just like PPTP or L2TP, you will have to enable the tunnel interface. You will be able to enable this under the PPP section of RouterOS under the SSTP server button.

In the SSTP Server menu, here you can enable the server, what port to use, the default profile as well as authentication. Note that we also have the Certificate option here to select your certificate. You will have to import a certificate into the certificate system before it will be an option here.



SSTP Server

□ Enabled

Port: 443

Max MTU: 1500

Max MRU: 1500

MRRU:

Keepalive Timeout: 60

Default Profile: default

Authentication

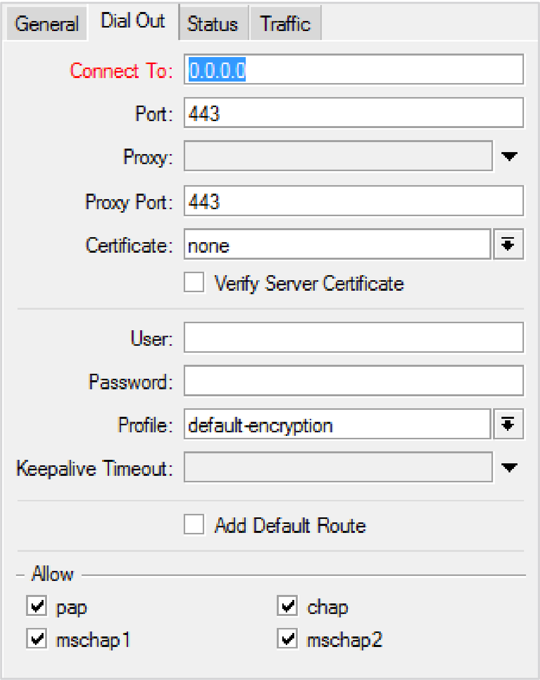
☑ pap ☑ chap

☑ mschap1 ☑ mschap2

Certificate: none

□ Verify Client Certificate

SSTP Client



Connect To: 0.0.0.0

Port: 443

Proxy:

Proxy Port: 443

Certificate: none

□ Verify Server Certificate

User:

Password:

Profile: default-encryption

Keepalive Timeout:

□ Add Default Route

Allow

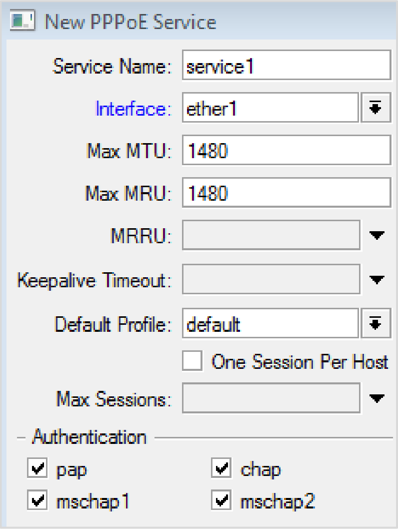
☑ pap ☑ chap

☑ mschap1 ☑ mschap2

The SSTP client is very similar to the PPTP/L2TP system. You will configure the new interface to connect to your server IP, what port, if you are or wish to relay though a proxy server, as well as client certificate to use. Note that you have the option to verify the server certificate as well. You will also still need your username/password and profile. Remember that this is basically a PPTP or L2TP system inside an SSL connection, so most of the properties are the same as any other PPP tunnel.

PPPoE Server

RouterOS offers a very powerful PPPoE server. An example of this is that we have run 2600 active sessions through one router with peaks upward of 200+ Mbps of throughput. That's a lot of encryption, traffic and data for one piece of hardware. PPPoE server is a Layer 2 protocol, so the only thing that you need for this service to work is the username and password. This of course, can come from the local PPP database and works just like a VPN tunnel, though with the exception that you need to have that Layer 2 connectivity for the connection to run. Since this is Layer 2 traffic, there can be no routers between each site, but you can place protected ports in-line; just remember you have to have two way communications between the client and the server. Since this system uses the PPP secrets and profiles, it can also use a Radius server as well.



New PPPoE Service

Service Name: service1

Interface: ether1

Max MTU: 1480

Max MRU: 1480

MRRU:

Keepalive Timeout:

Default Profile: default

□ One Session Per Host

Max Sessions:

Authentication

☑ pap ☑ chap

☑ mschap1 ☑ mschap2

Being that PPPoE Servers run via Layer 2, you can add them to a bridge group, Ethernet port or wireless interface. To add them, you simply need to click on the PPPoE Servers tab under PPP. Here you can add the PPPoE Service to your interface as you needed. You can select what authentication methods to allow as well as what default profile you wish to use. Just like in the PPPTP and L2TP services, this will be for users that do not have a profile from Radius. The One Session per Host field will enforce that only one connection can come from each MAC. This is useful to prevent several connections from one MAC address.

PPPoE Server Interfaces

Just like with PPTP and L2TP, typically when the user connects, it creates a dynamic interface. This interface is removed upon disconnect of the PPPoE session. You can create a PPPoE Server interface for you to apply rules to by simply clicking on Interfaces → Interfaces Tab → Plus Sign Drop Down → PPPoE Server. Inside this new interface put the username that the user will use to connect via PPPoE. This will create a static interface that is not running when the user is not connected, and will show running when the user is connected.

PPPoE Server, Dynamic Routing and /32 Subnets!

The PPPoE Server in RouterOS creates dynamic interfaces as PPPoE-Clients come up. Since you are creating a point-to-point interface, you can assign your customer a /32 subnet. This is a single IP, and then a private IP on the server side. Your customer may get 199.1.5.2, a public IP on their PPPoE-Client, but the default gateway would be 10.0.1.1, a private IP. Their subnet mask will be a /32 or 255.255.255.255, giving them only one direction to go, their remote address. When you do this, you can assign public IPs out to your customers again, with a single /32 subnet. If you add in a dynamic routing protocol, such as OSPF, as soon as the interface comes up, that's a state change, so that subnet will be advertised. Within a few seconds, and sometimes quicker, that new route can appear in your edge router. That edge router having the large block of publics routed to it, now knows how to get to that individual /32 address on your network via private IPs.

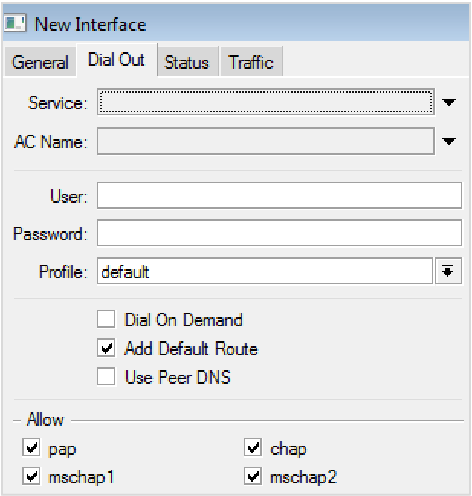
Using this method, you can assign public IPs all over your network without subnetting them down into smaller chunks. You can also have any IP on any tower. If your customer moves and they now connect to a new tower, their same username/password can give them the same IP address even though they are on another segment of your network. This allows you to give out public addresses without losing ANY to routing. Of course, this will increase your routing table size, but in many cases, the size of the routing table will not affect performance. You also will not have to deal with subnetting blocks of IPs out to towers etc., as you can use a Radius system to push a pool of addresses out to your clients, all controlled by your centralized Radius system!

Connection Balancing and/or Failover with PPPoE Servers

Since PPPOE is a Layer 2 system using PPPOE-Discovery packets, it is possible to balance between two or more PPPoE severs. If you add two servers to the same Layer 2 broadcast domain, you will end up with a very close balance. It will only balance the number of PPPoE connections, so it won't know that the majority of the traffic is going through one server or the other. If you have 200 PPPoE sessions, and add two, when they call come up, you will end up with around 100 on each one.

This really is more of a balancing between the connections that are coming into the PPPoE system on a single broadcast domain. However, in the event of a hardware failure on one of the units, the other will respond to all requests for reconnection. When the first unit comes back on-line, you will need to bump users off the second one for some of the connections to come back on the first one.

PPPoE-Client



Service:

AC Name:

User:

Password:

Profile: default

☐ Dial On Demand
☑ Add Default Route
☐ Use Peer DNS

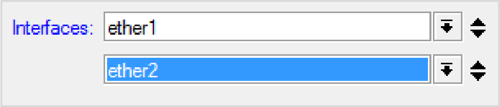
Allow
☑ pap ☑ chap
☑ mschap1 ☑ mschap2

The PPPoE-Client, just like the PPPoE Server, is a Layer 2 protocol. Because of this, it runs on an interface. The PPPoE-Client will obtain all of the information that it normally will needs to access the Internet or network. It will receive your IP address, subnet information, and default gateway, and you can also receive DNS information. Of course you have a few options -to get some of this information or all of it via the options for a default route and to get DNS information. You also have to specify the PPP Profile that you wish to use. Remember there are two default profiles in every system, both default and default-encrypted. If you require encryption on the PPPoE Server side, you will have to use the default-encrypted profile in order to connect; else it will attempt to connect and then just disconnect.

Even if you are using PPPoE-Client on a wireless or Ethernet interface, remember the PPPoE-Client is an interface. If you are doing masquerading, many people forget to change the masquerade rule to have an out interface of the PPPoE-Client instead of using the Ethernet or wireless interface. The reason for this, again, is that the PPPoE-Client is an interface, and you are no longer going out the WLAN1, instead, you are going out the PPPoE-Client interface.

I do get questions about the Service and AC Name. The service name is the name of the PPPoE service on the PPPoE Server interface. This name normally goes unnoticed, as most PPPoE-Clients look for any PPPoE service, regardless of its name. Usually, the goal is, “get them on-line quickly”. However, if you do have the time to kill, you can use the service name under the PPPoE-Client and setup the client to only use one PPPoE Service name. This could be used if you need multiple concentrators in a given broadcast domain due to speed and/or processor restrictions. My suggestion is to leave this to a single PPPoE Server per segment, and ensure that you have enough performance. If you have a failure, it’s simple enough to activate another server and get everyone back online quickly instead of having another parameter to configure in the client or server.

Multi-Link or MLPPPoE



Interfaces: ether1

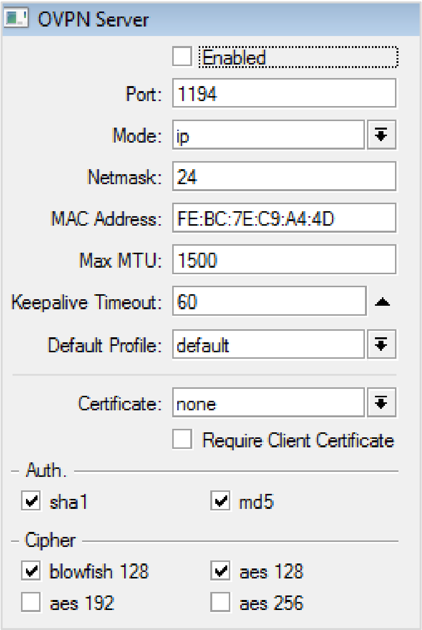
ether2

RouterOS also offers Multi-Link PPPoE. This service does, is gives you the ability to bond multiple PPPoE-Clients into one large pipe. To enable this feature simply specify multiple interfaces to run your PPPoE-Client on; doing so will automatically attempt to make a PPPoE connection on both interfaces. The PPPoE Server that you are connecting to must support MLPPP. You will need to contact your provider to be able to verify if their system supports MLPPP. Other than this, you will gain about 95% of the additional connection bandwidth because is typically some additional overhead, but all in all, you will experience a decent speed gain. Also, this method is a true bonding, so if you have 2 x 2Mbps/6Mbps Internet connections, then you will actually get on a single TCP connection of around 4Mbps/12Mbps.

OpenVPN

OpenVPN is an open source virtual private network (VPN) or virtual private network designed to create point-to-point or point-to-multi-client tunnels with strong encryption. It was designed to work across NAT and Firewalls as well. RouterOS supports both OpenVPN Server and Client. What is nice about OpenVPN is that it functions just like a PPTP or L2TP tunnel instead of an IPSec tunnel. If you are interested in getting all of the security that you need with the encryption of IPSec, and the ease of creation like a PPTP or L2TP tunnel, then OpenVPN is for you. It's based on SSL certificates and offers 3DES, as well as AES encryption capabilities. On top of all of those great features, it has been ported to virtually every operating system you can think of, including Linux, OpenBSD, Windows, Vista, and even MacOS.

Inside of OpenVPN there are two different modes, TUN and TAP. These are the common names in Linux and Windows operating systems; however, RouterOS has changed these names to what they really mean. TUN is for IP routing, and TAP is bridge mode, or in RouterOS, Ethernet. To create a bridged tunnel between two locations using OpenVPN, then you will use the TAP mode. If you wish to route across your tunnel (what I like to call "The Right Way"), then you will use the TUN mode, or IP.


OVPN Server

□ Enabled
Port: 1194
Mode: ip
Netmask: 24
MAC Address: FE:BC:7E:C9:A4:4D
Max MTU: 1500
Keepalive Timeout: 60
Default Profile: default
Certificate: none
□ Require Client Certificate

Auth.
☑ sha1 ☑ md5

Cipher
☑ blowfish 128 ☑ aes 128
□ aes 192 □ aes 256

OpenVPN Server

The Server portion starts out just like any other PPP tunnel. You will need to define a profile, and then create a VPN user under the PPP Secrets section. Then, you will need to enable the OpenVPN Server. IF you read the PPTP Server section, then you will know there are three buttons in the Interfaces tab of the PPP menu. The last one is our OpenVPN Server. So to get to this you would click PPP → Interfaces Tab → OpenVPN Server button.

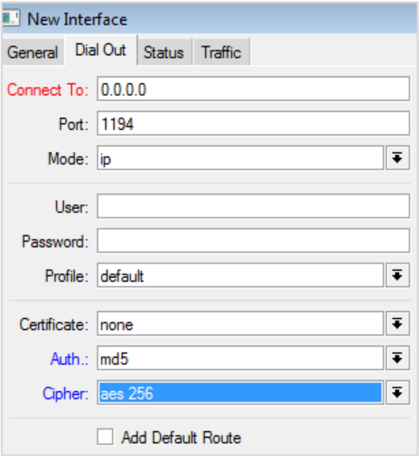
Of course you will need to enable the interface, and then pick what mode you wish your OpenVPN server to operate in. Select the Profile that you wish to use as well as the server-side certificate. If you need to install a certificate, refer to the certificates section of the book. You will also have options for what type of authentication encryption you wish to use, and what cipher. I typically will use AES-128, but if I need to ensure the data is secure, use AES-256.

OpenVPN Server Interface

This is a repeat of PPTP, L2TP, and PPPoE Server Interfaces. This functions just like the rest of the tunnels, so please refer to them for more information.

OpenVPN Client

The OpenVPN Client is an interface like the rest of the Tunneling Systems. On this interface, you can apply routes, Firewalls and rules too. Here, instead of checking boxes to allow ciphers and authentication methods, we have drop down boxes to select these. You will also need to have the correct certificate installed, the correct profile, and mode. These settings really will mirror the server side, but you will need a correct username and password as well. Once you enter all of the correct information correct, the link will connect up - just like any other tunnel. The difference now is that you can run AES encryption and have strong authentication and cipher methods.



Connect To: 0.0.0.0
Port: 1194
Mode: ip
User:
Password:
Profile: default
Certificate: none
Auth.: md5
Cipher: aes 256
□ Add Default Route

I really like using OpenVPN, because it gives me the security of IPSEC, and when dealing with financial or private information, this high security is a must. Moreover though, is that it creates an interface. This interface is “SIMPLE” in comparison to routing, Firewalling and do other common IP tasks too. If the data goes to the interface, it will be encrypted, so the method I use to send data over that encrypted tunnel is just like the rest of the tunnels in RouterOS. This makes it simpler for me to push encrypted traffic as needed.

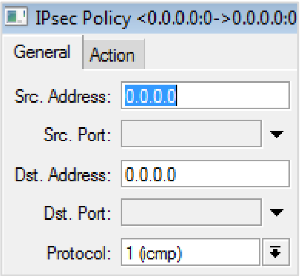
OpenVPN TAP / Bridging Mode

Just like PPTP, you can bridge using high quality cipher with OpenVPN. The method for creating the bridge is the same. You will need the bridge group's setup in your profiles and, but you will also have to change the mode on both ends to Ethernet, or TAP.

IPSec

Internet Protocol Security or (IPSec) is an entire protocol suite to encrypt, and secure IP communications. This suite is an open standard, so it can be used for cross platform security, ex. You can have a connection from RouterOS to Cisco, and on the same RouterOS system have another connection to a Juniper or other IPSec router or Firewall. IPSec has long since been regarded as the defacto standard in data encryption technology. There are entire books dedicated to IPSec, and therefore, we will not cover all of the technology here, the ins and outs. I will assume that you know the basics of IPSec.

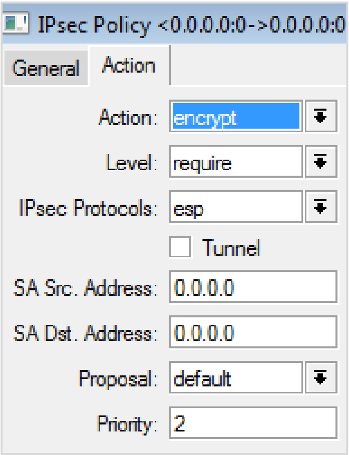
So let's start with where IPSec is matched. When you have data that you wish to encrypt, after performing any SRC-NAT rules (if needed), and right before the interface queue; the policy database for IPSec will be looked at. This policy is where we start with IPSec. The SPD or Security Policy Database (SPD) is created under the IP → IPSEC → Policies tab. These security policies tell your router what to do with data – how to encrypt it. Should we do nothing, or should we encrypt in some way. There are two parts to this; the first is the packet matching. Just like Firewall and Mangle rules, you have to match your data. If you wish to encrypt the data, first you must match it and then you can encrypt it, the second part, that's the action. RouterOS gives you options to discard, or drop the data at this step, encrypt the data, or doing nothing and continue on with the packet as if there is no IPSec for that packet. This gives you a number of ways to filter and match data.



IPsec Policy <0.0.0.0:0->0.0.0.0:0

General | Action

Src. Address: 0.0.0.0
Src. Port:
Dst. Address: 0.0.0.0
Dst. Port:
Protocol: 1 (icmp)



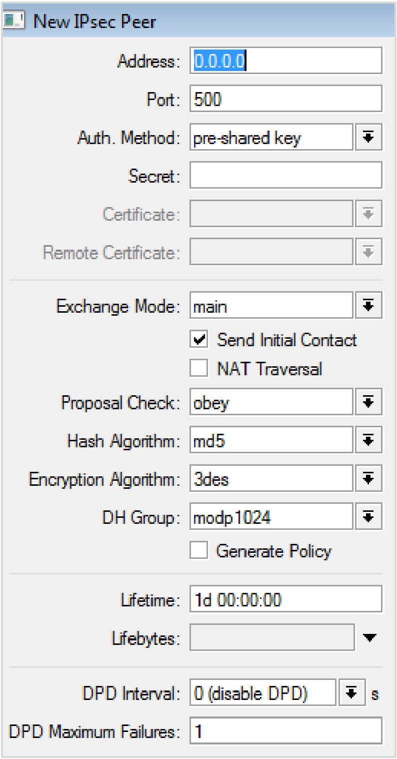
IPsec Policy <0.0.0.0:0->0.0.0.0:0

General | Action

Action: encrypt
Level: require
IPsec Protocols: esp
□ Tunnel
SA Src. Address: 0.0.0.0
SA Dst. Address: 0.0.0.0
Proposal: default
Priority: 2

All of this data matching does not do you any good unless you have some security; this is where the SA or Security Association (SA) comes into play. Each rule will have been associated with SAs that specify what and how the packets get encrypted. On top of all of this security you can even have multiple rules, using their own SAs, or using a common SA. The level field controls this. If you specify the “use” level, it will send the packet unencrypted, but if you specify “require”, that means you must have a SA for that data to go through, and this will use the IKE domain (something we will talk about in a few lines), to go ahead and create a SA. The last option is unique, this means that there will have to be a new SA just for data matched in this rule, and that SA cannot be shared with other policies!

IKE Domain



New IPsec Peer

Address: 0.0.0.0
Port: 500
Auth. Method: pre-shared key
Secret:
Certificate:
Remote Certificate:
Exchange Mode: main
☑ Send Initial Contact
□ NAT Traversal
Proposal Check: obey
Hash Algorithm: md5
Encryption Algorithm: 3des
DH Group: modp1024
□ Generate Policy
Lifetime: 1d 00:00:00
Lifebytes:
DPD Interval: 0 (disable DPD) s
DPD Maximum Failures: 1

The Internet Key Exchange is the system that provides the “keying material” for the ISAKMP framework. ISAKMP stands for the Internet Security Association and Key Management Protocol. This basically provides a means for authentication and automatic management of the SAs we talked about before. 99% of the time the IKE is not doing much. But if traffic is caught by a policy and there is no SA, then that policy will notify the IKE and it will establish a connection to the remote side of the link. The other time it is running is when it responds to said requests from a remote connection. When it does this it has two phases of operations.

Phase 1 is when the two sides agree on what algorithms they will use to send IKE information and then they exchange that “keying material” between each other. All of the SAs that will be generated will start from this material, so it has to be the same on both sides.

Phase 2 is when the peers establish one or more SAs. These SAs have a value that determines when they will become inactive. SAs can be based on a lifetime value, (a timed SA), or a life bytes value; it that remains active until a certain amount of data has been transferred, or both! Once either of these two values runs out, the SA will become invalid. These values also have two additional values, a soft value and a hard value. Once the soft value has been reached, the IKE domain is contacted again, and an attempt is made to create a new SA before the first one reaches its second timer. The second is the hard value, once the SA reaches this value, the SA is invalid. Hopefully there is a new SA already in place, or else, the data will have to wait for that, or be dropped.

If you wish to have even more security, you can regenerate the keying material every time the phase 2 operation starts. Even though the SA has been created and the lifetime or life bytes value is getting ready to expire, now we will create an entirely new key is created, to generate new SAs from that are totally different than the originals in phase 1. This can be very CPU intensive, and I would only recommend it on x86 systems!

IPsec Peers

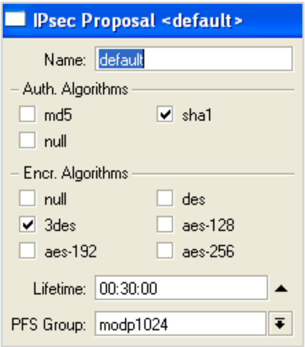
Once you have created your policy, you will need to create a peer. This peer gives your system all of the information that is necessary to create a connection. The peers are located on the Peers tab under IPSec. The peer you will need the basic information, such as the remote IP address and the port that you wish to use. Typically, you will start with a pre-shared key, this is a secret that will be entered on both sides, and will be the starting point for the keying material as well as the SAs. Make this a strong key; use upper and lowercase letters, numbers and some symbols if at all possible. You can also use a certificate to generate this material as well instead of a pre-shared key; however the key is the most common.

In this section you will also set your exchange mode. I use the ”main” exchange mode 99% of the time, and unless you know what you are doing with IPSec, I would suggest not changing this. The option for the initial contact allows this peer to tell the IKE to start a peering conversation. The NAT-Traversal option will only works in some cases. This basically enables the Linux NAT-T system that helps to solve IPSec incompatibility with NAT routers between peers. This only works with the ESP protocol. My results are mixed, but typically this will not help much if you do have a NAT system running. The proposal check is a lifetime/life byte check, that determines how it should act if these values are different from on one router to another. I would suggest ensuring that they are identical on both peers to ensure proper operations. These are also set here in the peer options.

The rest of the options will allow you to set the kind of encryption and proposals you wish to use. These values will need to be identical on both peers of course. You can also check the box to generate policies. This creates SAs based on traffic that may go across a tunnel. It will generate these SAs dynamically as traffic passes, creating a simple way of encrypting traffic, without having to create a lot of complex IPSec policies.

Proposals

The Proposal is basically the start of the conversation. The Proposal starts a secure channel between the two IPSec peers and allows them to communicate securely even during the start of the conversation. When configuring Proposals, you will need to have the same information on both ends. The Authentication Algorithm allows the two sides to authenticate against each other. The Encryption Algorithm is the method of encryption.



IPsec Proposal <default>

Name: default

Auth. Algorithms

□ md5 ☑ sha1

□ null

Encr. Algorithms

□ null □ des

☑ 3des □ aes-128

□ aes-192 □ aes-256

Lifetime: 00:30:00

PFS Group: modp1024

The Lifetime and PFS (Perfect Forward Secrecy) Groups are also specified here in the IPSec Proposal menu well; these will need to match on both ends of your IPSEC tunnel.

Encryption Performance

Since we are talking about encryption, now is a good time to discuss the different types of encryption and the performance that each provides of each. Most people have heard about triple DES, or (3DES). This is a very common high-security encryption method that is widely supported. However this 3DES is fairly slow in most cases. Performance and encryption using this method takes quite a bit of CPU time and I would recommend at least a high-end RouterBOARD or even better, an x86 system. The AES-256 encryption method is a Department of Defense (DoD) standard. It offers better encryption and faster encrypting/decrypting routines than 3DES. If I had an option of using AES or a form of DES encryption, I would use AES.

Choosing a Tunnel Type

Choosing your tunnel type can be confusing. Between all of the acronyms and security options, you have a daunting task. The following chart summarizes and compares the tunneling options for you. So I wanted to break down the information so that you can choose what you wish to use. Below is a chart that shows what kind of encryption, what board you may need, as well as other information that you may find helpful in your choice.

Tunnel Name	Protocol Used	Functional Layer	Setup Complicated?	Private Data?	Max Encryption	Minimum Hardware
PPTP	TCP	2 or 3	No	No	MPPE 128	400AH+
L2TP	UDP	Layer 3	No	No	MPPE 128	400AH+
IPIP		Layer 3	No	No	None	400
EoIP	Protocol 47	2 or3	No	No	None	400
OpenVPN	TCP or UDP	2 or 3	Yes	Yes	DES – AES256	RB1000+
IPSec	UDP	Layer 3	Yes	Yes	DES – AES256	RB1000+

A few other points that you want to remember are; IPSEC and OpenVPN will require quite a bit of CPU power. OpenVPN is not difficult to setup, but it is more time consuming than setting up PPTP tunnels. If you are in the need to ensure that you are providing maximum protection for private data, things like complete customer financial data, credit card numbers that are not already encrypted, or bank information, then encrypted with something stronger than the MPPE. However, if you are not transporting sensitive information then, use PPTP or L2TP, as these are much simpler to setup, and troubleshoot!

Traffic Control

MikroTik RouterOS offers a very advanced method of controlling traffic, as well as many different ways to control traffic. You can queue traffic and control it based on individual IP addresses, giving each IP address its own queue, its own bursting abilities all based on up and down speeds, or a total speed. You can also evenly distribute traffic among IPs on given subnets by using the PCQ queuing method, all with a single IP. RouterOS, though, does not stop there; you can identify traffic by types, including protocols, ports, source or destination IP addresses, peer-to-peer traffic as well as by using stateful packet inspection or Layer-7 identification rules. With all these capabilities, this you can build an extremely sophisticated queuing system that can provide advanced quality of service, (QoS) for your customer’s data, based on any method you wish!

Compared to using RouterOS, providing advanced QoS is difficult on many systems, and they only allow you to identify specific types of traffic. Many of switches look only at simple Type of Service (ToS) bits, however, that may not be the only method you wish to use to prioritize traffic. With RouterOS you can tailor QoS for other latency-sensitive applications, such as terminal services, remote applications, and even telnet sessions can also be prioritized inside RouterOS. The definition of QoS is to provide Quality of Service to some form of data, and with RouterOS you can define what that data is, and how it acts!

The first step when you start building your queuing system is to understand that you must identify traffic. You can use many different methods inside RouterOS to identify traffic for your queuing system. These methods can be as simple as specifying an IP address or an entire subnet range. As you get more advanced and wish to really start providing more than just bandwidth limiting and queuing, but QoS, then you will need to start identifying traffic based on protocol and ports, and if necessary, Layer 7 traffic characteristics. For most people, managing bandwidth for a specific IP address is the most common use of QoS.

In this section, we are going to talk about how MikroTik does its queuing, the methods of queuing available, and how to ensure QoS with applications, controlling Peer-2-Peer traffic and help you understand how bursting works as well.

Identifying Queue Data

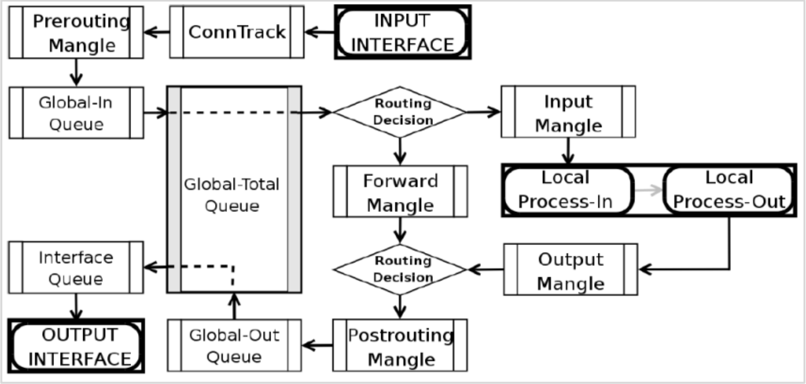
Normally we would go into a long section on how to identify data; however, we already covered this in our Firewalling and Mangle section. However, it’s important to note that if you wish to identify data by using ports and protocols, you will need to create packet marks so that the queuing system has something to identify the traffic with. RouterOS does allow for IP addresses and subnets inside the simple queue system without using Mangle to identify traffic. Your situation and needs will dictate how you wish to identify traffic, and you can identify traffic based on both IPs and Mangle packet marks, the trick is to put both of these methods together. I would like you to refer to the Mangle section for more information on how to identify traffic using your Mangle system.

Hierarchical Token Bucket– HTB

MikroTik uses a system called HTB or Hierarchical Token Bucket to provide all of the queuing and bandwidth control inside RouterOS. This is a common Algorithm, it allows bursting of data, and controls when data can be transmitted by controlling the outbound data flow. All QoS implementations inside RouterOS will be based on this system. This system uses a hierarchical Queue structure by creating three virtual HTB queues. These queues are Global-In, Global-Total, and Global-Out. However, there is also a queue created for every interface, but remember this is only for outbound data. We typically can't control data coming in, however, data flowing through the router, has two control points. Data from our LAN going out our WAN has a control point, as it goes out our WAN connection. Data from our WAN going to our LAN ha a control point as it goes out our LAN connection. Using this method we can control all aspects of data as it flows through our RouterOS system.

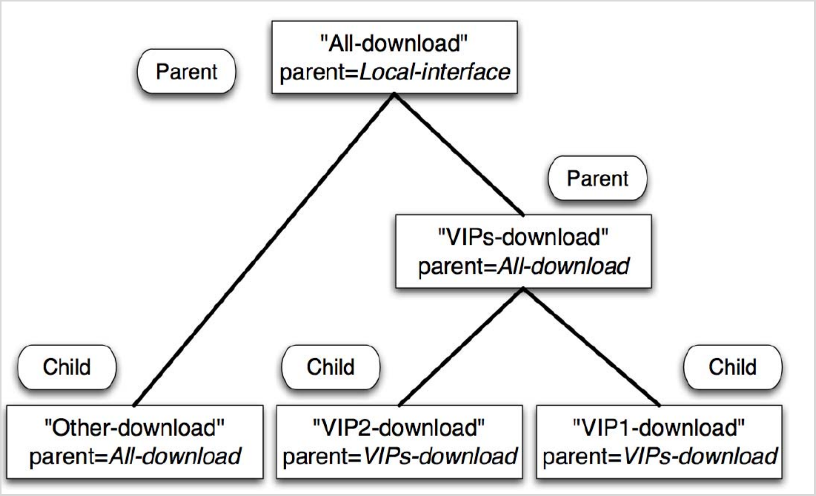
HTB Packet Flow

As packets flow through our router, it will flow through all three global HTB queues, but it will also pass through the interface HTB queue as well. So for data going through our router, it passes through a total of four HTB queues. Data to our router will only use the Global-In and Global-Total queue, so it only passes through two queues. Data that our Router generates will pass through the Global-Out, Global-Total as well as the interface HTB queue. You can see this on the image below.



HTB Queue Tree Structure

As far as bandwidth is concerned, HTB has a few rules that it follows. As we said, HTB forms a hierarchical queue structure, so you have queues that are parents of other queues, and queues that are parents of other parents. Once a queue has a single child, or queue under it, then it is considered a parent queue. Now the hard part, no matter how many parent queues there are or the number of levels of parent queues, all child queues treated as equal. You need to use the child queues for your actual traffic. SO you match traffic in your child queues. Your parent queues are strictly for distributing that traffic. Of course, child queues cannot receive more traffic than the parent has as well. See the image below for a better understanding of this.



HTB and Rate Limiting

HTB has two rate limits, the limit-at and max-limit rate. You may have heard of CIR and MIR though, and these relate to the limit-at and max-limit rates in RouterOS. The CIR, Committed Information Rate, or limit-at rate in RouterOS is considered a guaranteed amount of bandwidth. This is what you will say your customer is guaranteed, providing that there is enough bandwidth available. Keep in mind that even though you have a limit-at of 1 Mbps for each of your 10 customers, if you only have 5 Mbps of Internet bandwidth, then you really can’t guarantee that bandwidth. But if you have 20 Mbps, and other customers that don’t have a limit-at rate at all, they are not guaranteed any bandwidth, your customers with the limit-at will receive the bandwidth and then the customers with only a MIR or max-limit will get what's left over. The Max-limit is defined as; during a best case data can flow up to this limit, assuming that there is bandwidth available.

There are a few rules as well for the bandwidth distribution using your queues. First is that your max-limit of the parent must be either greater than or equal too, >=, the sum of all of your child limit-at's, and the max-limit of all of your child's must be less than or equal to the max-limit of your parent.

Queue Types

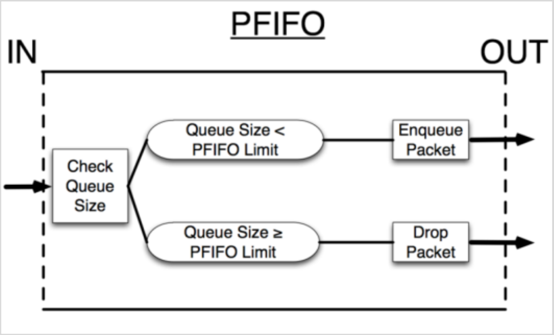
There are a number of different types of queues. RouterOS supports four different types of queues; FIFO, RED, SFQ and PCQ. To help you decide on what you will use, the chart below will assist you!

Queue Type	Reason to Use	Pros / Cons
FIFO	Use for simple bandwidth limiting and control. Simplest and fastest.	*Pros:* Very quick, low CPU overhead. *Cons:* Provides only two priorities.
RED	Have never found any.	*Pros:* Still very quick. *Cons:* Never had a need for the random feature.
SFQ	Gives you up to 16 queue levels, a must if you are wishing to provide QoS.	*Pros:* Provides up to 16 priority levels, and works great for providing QoS configuration. *Cons:* Highest in CPU cost.
PCQ	Use if you wish to share bandwidth equally among many users.	*Pros:* VERY FAST, one queue can serve hundreds of clients. *Cons:* Dividing this into sub queues of different types of traffic and QoS becomes difficult.

To configure your queue types, you will need to go into the queue types tab under queues click on Queues → then the Queue Type Tab. Here you can specify queue names along with their types and their configurations.

FIFO Queues

FIFO or First-In-First-Out Queuing does exactly what it says. Data leaves the queue in the same order that it entered the queue. As data comes in it goes out. FIFO does not reorder packets based on priorities as they flow through the router, however, for basic bandwidth shaping and traffic limiting, it works! There are actually two types of FIFO queues in RouterOS, -byte and packet FIFO queues. They both work the same way however they operate just on different types of data. A packet FIFO queue works on entire packets while a byte FIFO queue works on bytes of data.

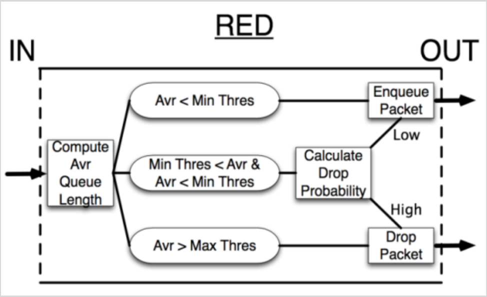


The way this works, is simple, as data comes in, it flows through a queue, Think of FIFO the queue as a bucket of water with a hose going in and a valve for a drain. As data flows into this bucket, the bucket is constantly draining at the rate that you specified in the queue. So if you have a drain that allows can fit 1 Mbps of data through, then that would be its max-limit. As data comes into the bucket, it drains back out at 1 Mbps. Sometimes, data comes into the bucket faster than it can drain out. This is normal, so what happens then? The bucket eventually becomes full depending on the queue size of the bucket. If you have a queue size of 10 packets, then once 10 packets come in to the bucket, it’s full. As more and more packets come in, again, you can only drain the bucket at the max-limit rate, and then eventually the bucket will overflow. Those bits or packets that “spill out” are lost, in our case, dropped. Now, the data stream has lost data. TCP/IP corrects this; by slowing down the speed at in which data is sent and eventually, you end up with an actual data rate very close to the max-limit that the bucket is draining at.

This is the default behavior for most queues and as you create queues in your queue tree or simple queues (discussed further on in this section) you will have many buckets that are filling and draining all at the same time.

RED Queues

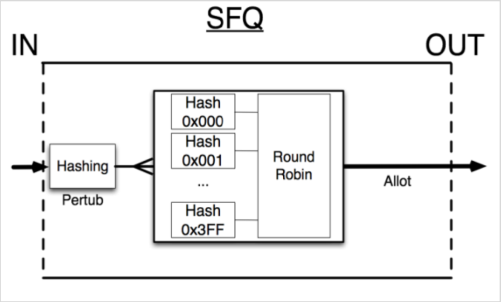
When I first started writing this section, I questioned if I should even put this queue type in the book. I have never really found a good use for Random Early Detection (RED) Queues. These queues function just like a FIFO queue, with only one exception. RED queues allow an additional possibility that packets coming into the bucket could be dropped randomly. The idea behind this to prevent what is called Global Sync. We won't get into a detailed description of that here, but basically it occurs when all of the data going through the router fills up the queues. Each stream slows down and then, all at the same time, tries to speed up again. With that additional random probability of dropping data even though the bucket or queue is not full, RED fixes this issue.



I typically do not use RED in production networks; there just simply does not seem to be a need for this in most cases. However, your situation may warrant such a queuing system; therefore it is built into RouterOS.

SFQ Queues

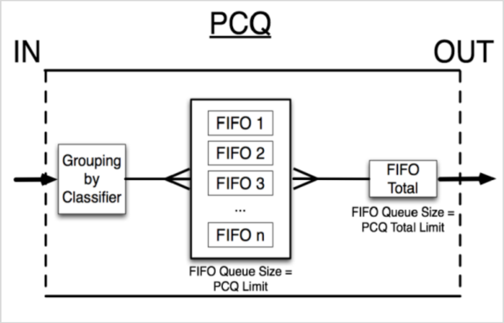
Stochastic Fairness Queuing (SFQ) is the way to go if you are looking for great QoS implementation. This system will take advantage of priorities, max-limits and limit-at's in your queues. It works by using a hash value from up to four different classifiers, typically but not limited to using both source and destination addresses for most types of implementations. Then it divides that traffic into 1024 sub-streams, and then performs round-robin between each of those sub-streams. Even though this queue uses the most CPU time, it is absolutely great for traffic prioritization



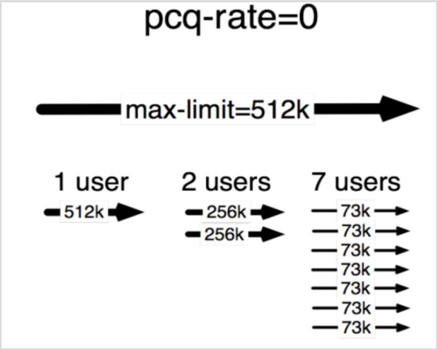
and QoS implementations with RouterOS. With SFQ queuing system, you can guarantee data rates, provide the QoS type of services based on types of data, as well as ensure VoIP quality.

PCQ Queues

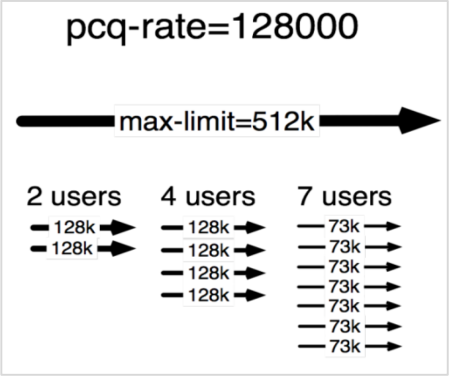
Per-Connection Queuing (PCQ) is a MikroTik specific queue type. This was designed to simply distribute traffic evenly across a large subnet and then provide the ability to limit each sub-stream that is created while maintaining a super-low CPU requirement. PCQ works by taking classifiers, and then based on those, forming sub-streams. Each of those are basically an individual FIFO queue. In most WISPs and ISP implementations, the idea is to have an entire subnet have the same max-limit for each individual IP address, or to share an amount of limited bandwidth with all IPs evenly.



Using PCQ is very simple; however there are a number of things you will need to understand. There are two limits to your PCQ queues, a max-limit which shows the overall queue bandwidth, and a pcq-rate. The pcq-rate is the rate to give the individual sub-queues. If you leave that, the pcq-rate, as zero, there will be no individual queue limit. This will allow us to evenly distribute the max-limit of the PCQ queue regardless of the number of sub-streams.



So how does that work? Let's assume that you have a max-limit of 512k. As you add more and more users to the network, they will get grouped by classifier, and separated into each of their sub-streams. If you have two users downloading, each can only get 256k as there is only a total of 512k available. As more users come on-line and start moving data, as with the image to the left, the bandwidth for each user goes down and is split evenly!



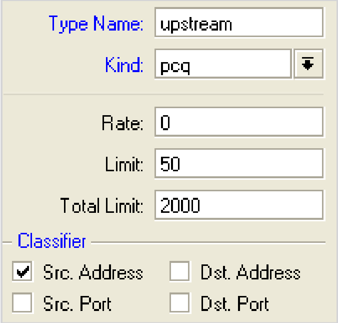
If we specify a pcq-rate, now we are adding that individual rate limit for each of our sub-queues. So in the example above, when we had two users, they could use 256k each, however now that we have 128k pcq-rate, each user cannot use more than 128k individually.

When configuring your PCQ system you also need to look at the limit and total-limit settings of your PCQ Queues. If you have a limit of 50 (50 packets per sub-stream limit) and a total-limit of 2000 (all sub-streams have a combined limit of 2000 packets), then it would only take 40 users before the entire queue is filled. You can do the math but a total-limit of 2000 divided by a limit of 50 equals that 40 user's number. You should have at least 10-20 packets available for each user, so you will need to increase the total-limit number as your user count grows. If you set a limit of 50, and figure 20-25 packets per user and if you have 300 users, then you would need a total-limit of around 7500.

PCQ RAM Requirements

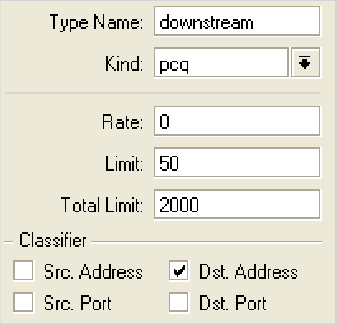
As the number of users this grows, you will need to take into account RAM usage as well. RAM is used as queues are being used, once the buckets are full, the queues have to store that data someplace, and this is in RAM. PCQ allocates RAM based on your total number of packets stored. You can figure a max usage of 1500 byte packets times your total limit, there is overhead as well. PCQ uses about 4.2 Megabytes of RAM if you have a total-limit of 2000, and around 10.5 Mbps of RAM for a total-limit of 5000. If you take our example above and figure a total-limit of 7500, that would be around 15.7 Mbps. Take your total-limit and divide that by around 470 or so. That will get you a good number for RAM.

Using PCQ

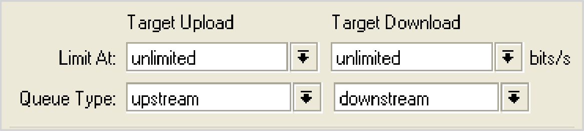

Type Name: upstream
Kind: pcq
Rate: 0
Limit: 50
Total Limit: 2000
Classifier
☑ Src. Address ☐ Dst. Address
☐ Src. Port ☐ Dst. Port

Now that you understand how PCQ works, I want to describe how to configure PCQ! First, we need to create two different PCQ Queue types, up and down. This will help us identify traffic that is considered up, or going out to the Internet from our customers, and traffic going down, or to our customers.

First go to the Queues → Queue Type tab and create a new queue. This one will be our upstream queue. This queue we will use our source address as our classifier. We will then create a second queue type, called downstream, again, note that we are setting this up as a PCQ kind. This time, the downstream PCQ will use a destination address as its classifier.


Type Name: downstream
Kind: pcq
Rate: 0
Limit: 50
Total Limit: 2000
Classifier
☐ Src. Address ☑ Dst. Address
☐ Src. Port ☐ Dst. Port

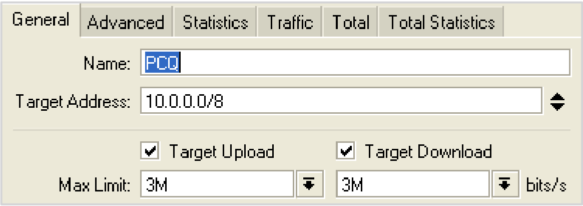
Now that we have both of these queue types created, now we can identify our traffic and limit them. In this case, we are not specifying a PCQ-Rate, we are leaving the rate fields in each one of these PCQ types as 0, and therefore we are not limiting each individual customer to a specific rate. If you want to limit your upload and download rates per customer, or in our case per IP, you would do that in the PCQ Type rate field.



	Target Upload	Target Download	
Limit At:	unlimited	unlimited	bits/s
Queue Type:	upstream	downstream	

Once we have our PCQ types and rates per customer, now we need to setup a rule to match data from our customers, and then setup max-limits that that queue can pull. If you don't set a max-limit, then the PCQ will assume that you have 100 Mbps, or whatever your Ethernet connection is, and will not divide up the bandwidth accordingly. You have to setup a rule that knows how much bandwidth you wish to divide evenly between all of your customers!

Now, we'll create a simple queue rule. Under the Advanced tab, we select your upload and download queue types to our new PCQ Queue types that we created. We Specify the target address so that you know what data you are aiming for, in this case our; in this case, private subnets. , and then specify a Max-Limit so that the PCQ system knows when to start dividing the bandwidth up at.



Name: PCQ

Target Address: 10.0.0.0/8

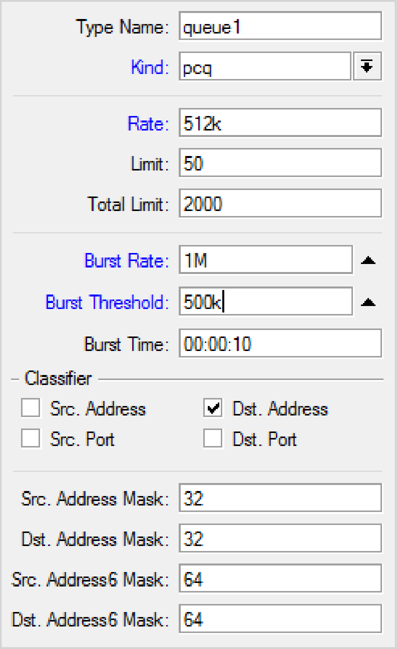
☑ Target Upload ☑ Target Download

Max Limit: 3M 3M bits/s

PCQ Bursting

MikroTik added PCQ Bursting in v5rc5 (version 5 release candidate 5). PCQ bursting functions just like the standard PCQ system and is configured the same way, as v3 and v4 PCQ is configured. I even left the older version screenshots in the original PCQ system section, however now we have options to allow bursting inside the PCQ system.

As far as options, it's very close to the standard bursting option in the simple queues. For more information on the bursting options, I would suggest reading up on the bursting section. The options are the same as the standard bursting system. You still have the classifier, and now as well as now you have a burst rate, and a burst threshold and a as well as burst time.



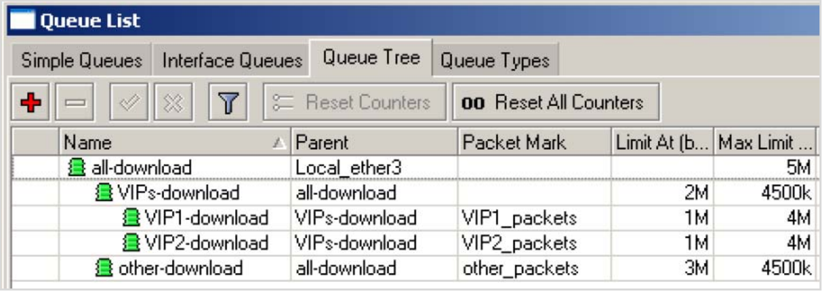
Type Name: queue1
Kind: pcq
Rate: 512k
Limit: 50
Total Limit: 2000
Burst Rate: 1M
Burst Threshold: 500k
Burst Time: 00:00:10

Classifier
☐ Src. Address ☑ Dst. Address
☐ Src. Port ☐ Dst. Port

Src. Address Mask: 32
Dst. Address Mask: 32
Src. Address6 Mask: 64
Dst. Address6 Mask: 64

Queue Trees

The Queue tree is an implementation of Hierarchical Token Bucket (HTB). To get to this, click on Queues → Queue Tree tab. The queue tree only works in one direction, so you will need to create two queues, one for up and one for down if you want working to control traffic both ways. Inside the queue tree, all queues are processed at the same time, so they are much faster than simple queues, even though you have to have two of them to perform the same task. Something you can do inside the queue tree that you cannot in the simple queues is providing double-queuing. By using your Mangle system, you can mark packets and process them on the queue tree. You don't have to mark twice, one for each direction. If you mark web traffic for instance, data that is going out your WAN interface is (your up traffic) then the data going out your LAN interface is your down traffic. You can specify speeds here as well as your priorities.



Name	Parent	Packet Mark	Limit At (b...	Max Limit ...
all-download	Local_ether3			5M
VIPs-download	all-download		2M	4500k
VIP1-download	VIPs-download	VIP1_packets	1M	4M
VIP2-download	VIPs-download	VIP2_packets	1M	4M
other-download	all-download	other_packets	3M	4500k

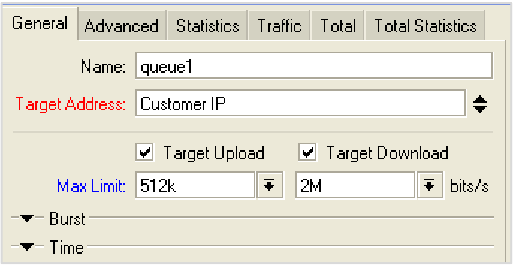
As you can see from the above image, you can setup multiple parent queues, typically though, you will setup the main parents on the actual interface. You do this by specifying the parent as the interface you are going out. You would then need to create a second set of rules, just like the one above, however, this time; you would create an all-upload queue with a parent of the WAN connection. It is also important to note that any simple queues that match traffic that would normally be matched by your queue tree, will take that traffic and not allow the queue tree from processing, as simple queues are processed before the queue tree.

Simple Queues

Simple queues are designed to make your life simple by providing a single queue for individual and/or multiple IP address and subnets. The simplest configuration of simple queues is to put a target address as your customer IP, or the IP that you wish to control bandwidth on, then you can set both max-limits and limit-at data rates. The simple queue will actually create between zero and three queues, possibly creating Global-Total, Global-In and/or Global-Out queues. These are actually created in the queue tree but you won't see them. If you create several rules in your simple queues, and then click on your queue tree, you will see something that the bottom like "0 of 2", or "0 of 8" etc. It will be a blank list, but the "8" or second number is the number of dynamic hidden queues created based on your simple queues.

Limiting Total Throughput for IP or Subnet

To create a simple queue that will limit an IP or subnet to a specific speed, the simplest method is to create a simple queue, select the target address of the customer IP and then select their speed Max-Limit. Inside the Target address field you can click on the down arrow and put in a second or third IP address. You can also put in a subnet range as well, something like 192.168.1.0/24. Your Max-Limit field will effectively limit the target addresses up and down bandwidth speed. Specify the speed in bits, so (512k would be 512000). You can also specify the speed either in Kilobits by typing "512k" or in Megabits "2M".



Name: queue1

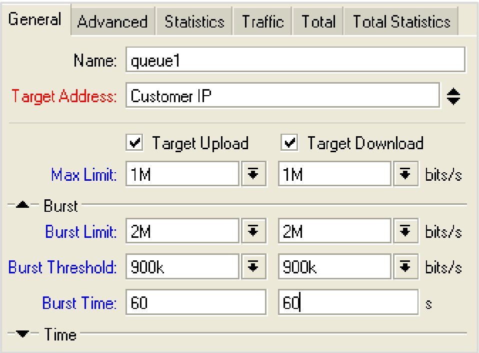
Target Address: Customer IP

☑ Target Upload ☑ Target Download

Max Limit: 512k 2M bits/s

Bursting

Once you have setup your customer with their simple queue, you can also do bursting. Bursting allows you to specify several options; that will give your target address the ability to receive a higher data rate for a short period of time. This works very well when you have traffic that is short and bursty. Web traffic for the most part is this way, - you load the webpage, and once the page is done loading, you basically sits there moving no data while you read the page. Bursting in this case, is perfect, giving your customers a faster web surfing experience overall. Downloads can be done this way too, if you have a large download that is a few hundred Megabytes, you can download that at 2 Mbps for a while, but once your queue no longer allows you to burst the download would slow.



Name: queue1

Target Address: Customer IP

☑ Target Upload ☑ Target Download

Max Limit: 1M 1M bits/s

Burst

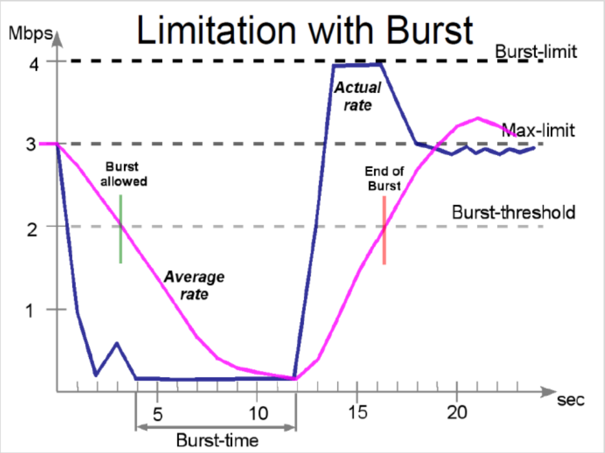
Burst Limit: 2M 2M bits/s

Burst Threshold: 900k 900k bits/s

Burst Time: 60 60 s

Time

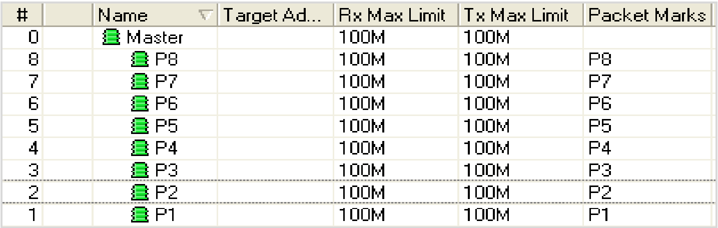
The example to the right shows you how you can setup bursting for your customer. In this case the customer will receive a burst of 2 Mbps for roughly 30 seconds. That assumes that for the last 60 seconds they have not transferred any data. Bursting is a tricky subject and I have some graphs that will help as well.



Bursting works by looking at a variable called the average data rate. This is not something that you set, but something that is calculated inside the router. It is important to understand how this is calculated though so you can understand how bursting actually works. RouterOS calculates the average data rate by taking the burst time, in our example above, its 60 seconds, and dividing that up into 16 chunks and then averaging those 16 segments together. If the customer has not moved any traffic or data in the past 60 seconds (in this case), then their average data rate will be basically zero. As the customer starts a download, their actual data rate goes up. Once it hits the 1M Max-Limit, we then do a comparison. Is the average data rate over the burst threshold? If it is not, in our case, the average data rate was basically zero, so bursting is allowed. The customer then starts to receive 2 Mbps of bandwidth. As their download progresses, the average data rate for the customer over time, goes up. Since we are dealing with 1 Mbps and 2 Mbps, it is safe to assume that around the 30 second mark, the average rate will go over the burst threshold. Once the average data rate goes over that burst threshold, the queue no longer allows bursting, and the customer's actual data rate is slowed to 1 Mbps.

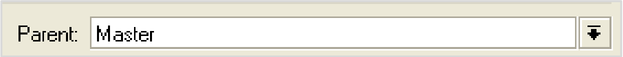
Creating Queue Priorities with Parents

This is where some people get lost on understanding how the Parent and Child queues work. You create your Parent queues with one purpose, to manage traffic. You don't want them to "match" traffic. Then you create Child queues to actually match traffic. However, with that said, you can use your Parent queues to match traffic on-top of your Child queues. Finally, On top of this; rule order is important as well!



#		Name	Target Ad...	Rx Max Limit	Tx Max Limit	Packet Marks
0		Master		100M	100M	
8		P8		100M	100M	P8
7		P7		100M	100M	P7
6		P6		100M	100M	P6
5		P5		100M	100M	P5
4		P4		100M	100M	P4
3		P3		100M	100M	P3
2		P2		100M	100M	P2
1		P1		100M	100M	P1

By using SFQ queue types, as well as using Parent queues, you can start to create quality of service, QoS, systems. In the previous image, you will see a basic core router QoS system. What this does, is simply identify data via the packet marks, and then apply them to the Master queue. The Master queue has plenty of bandwidth, so we are not limiting bandwidth except at 100 Mbps. In a single clock cycle, packets that are P1 will go out before packets with a status of P8. These are arbitrary



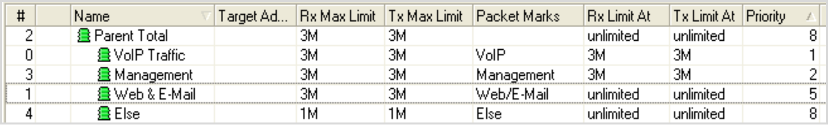
Parent: Master

identifications; we use the Mangle system to mark packets and identify them based on the type of traffic. To set queues as sub-queues, simply click on the Advanced tab of your queue and set a Parent queue. In this case, we use the Master queue as the Parent. This means the sub-queue will share bandwidth with the Master queue.

Ensuring Bandwidth Allocations– VoIP

Now that we know how to setup Parent queues, and setup basic QoS systems, I want to talk about how to create a queuing system that will ensure bandwidth to applications that need it. More importantly, this is a QoS system, which will allow you to properly prioritize bandwidth usage and ensure quality VoIP calls. That's what most of us wish to do, but it's not as simple as that. I love looking at other consumer grade routers and network devices that have a check box that says "Prioritize VoIP". What it doesn't ask is what kind of VoIP. How does it know what the VoIP data is? What ToS bit?

First, to ensure bandwidth you have to be able to identify your traffic. In the case of a VoIP system, you may have a ToS bit, or, if you run your own VoIP system, then you have IPs! I like this even better because it gives us a simple way to identify traffic.



#		Name	Target Ad...	Rx Max Limit	Tx Max Limit	Packet Marks	Rx Limit At	Tx Limit At	Priority
2		Parent Total		3M	3M		unlimited	unlimited	8
0		VoIP Traffic		3M	3M	VoIP	3M	3M	1
3		Management		3M	3M	Management	3M	3M	2
1		Web & E-Mail		3M	3M	Web/E-Mail	unlimited	unlimited	5
4		Else		1M	1M	Else	unlimited	unlimited	8

A basic VoIP QoS system is shown above. This example assumes that we create the necessary Mangle rules. We identify traffic going to and from our VoIP server as VoIP traffic; we also identify management traffic, Web and E-Mail as well as anything else. We identify management traffic because things like OSPF packets are very important as well as WinBox traffic and maybe something like SSL. You could also use your Mangle to identify traffic from management subnets, and prioritize them as well. Web and E-mail is typically what we want to be as fast as possible, so we prioritize that as well above other types of traffic. Last, we always need to have Else queue identifying anything else going through our router. Here, we also limit our Else queue to one-third of the total bandwidth that we have.

The number one thing that everyone forgets is to setup some form of total limit. For example, something that recognizes Internet connection is 3 Mbps, etc. If you don’t do this, then when your VoIP traffic goes up to 1 Mbps, RouterOS does not know to pull bandwidth from other queues. In our case, we have specified that VoIP has the number 1 priority, and that it can use all 3 Mbps of bandwidth if needed. In most cases, this would not occur, but we don’t limit the bandwidth to something small.

You also need to remember that you will need to create Mangle rules that apply to your network. Rules that someone else creates are is not necessarily what you want for your network! In our case, we identify traffic and change ToS bits on the packets, this way our core routers do not have to process more than a few rules to be able to apply the QoS system.

Creating Advanced Queues

Double Queuing

Double queuing is a method of queuing data twice! Yep, I said twice. Why would you want to do this? Well, the idea is to provide quality of service for different types of traffic while still maintaining overall speed restrictions for individual IP addresses. So even if your customer has a 1 Mbps up and 1 Mbps down connection, you can ensure the VoIP and Web traffic are at a higher priority than their P2P traffic.

To double queue this, you will need to mark your data twice. The first mark is done in your pre-routing chain. This is where you will mark data based on traffic. You would identify web traffic, point-to-point, email, and VoIP here. Once you do this, you would then create an HTB queue with the Parent of the Global-In HTB queue. This will then allow you to specify each mark under that Global-In with its correct priority.

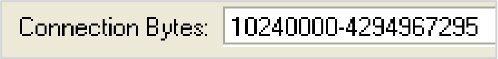
The second step is to mark your data again, typically you would use an address-list to identify customers at several different speed packages, and then, mark the packets based on their relationship to the address-list their IP is on. You will do that in the forward Mangle chain. Then you create interface HTB Queues, one for your WAN and one for your LAN interface, and setup PCQ rules to limit the marked packets accordingly.

With this type of configuration, you will need to have as few Mangle rules as possible, as having lots of Mangle rules will create load on your system. This system will allow you to have both customer queuing and have traffic prioritization queuing on the entire system as well.

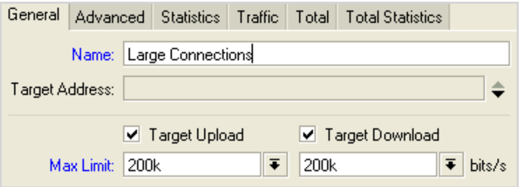
Large Transfer Queues

I have some customers that have issues with large downloads. Customers will come in and start huge downloads that run for hours. That is not usually an issue when you have plenty of Internet bandwidth, however, in some cases your bandwidth is limited and the large downloaders can slow down all the users on your network. This affects your network more if you are allowing them a substantial amount of bandwidth. If you use PCQ systems, normally this should balance out your data and customers, so one customer will not negatively affect the rest of your users. Regardless of the reasons, I find it interesting that we now have the option to can limit large downloads to slower speeds separately from your customers normal queuing. The example I will use here is a customer that starts a large download, for example, let's identify that by a download that has went over 10 Mbps of data. Then, we can separate that data out from the customer's individual queue and group all of these large downloads together into one small pool.

To do this, we have two steps. The first step is to identify that large data download. Do this in your Mangle system. You should identify the connection, and then mark the packets accordingly. The way to do this is by using the connection bits option under the Advanced tab of your Mangle. In our case, 10 Mbps is roughly 10,240,000 bits. Once we get a connection that goes over that in bits, we can identify that connection and then do a packet mark. Once we get that packet mark, we can create a simple queue, which is higher in the rule list than all of our customer's individual queues. The reason we move the rule above our individual queues, is because we want this data to match before the other queues take effect. Once a connection goes above that 10 Mbps that we specified, that connection all of a sudden matches a different queue, the Large Connection queue that we just created. This then will put all of the customer's with large connections into one queue with very limited bandwidth.



Connection Bytes: 10240000-4294967295



Name: Large Connections

Target Address:

☑ Target Upload ☑ Target Download

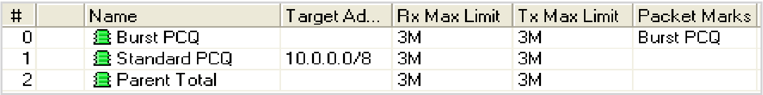
Max Limit: 200k 200k bits/s

Setting Multiple PCQ Rates

In this section we describe how to create a bursting system using PCQ prior to when we had an actual PCQ bursting system. This system does not work 100% like the actual PCQ bursting in RouterOS v5. See the PCQ Bursting section for the most up to date information using RouterOS v5's PCQ Bursting system.

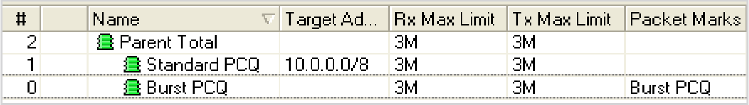
We have covered quite a few different ways of limiting traffic and described how to do customer bursting on individual queues, however, what if you are doing PCQ and you wish to burst? Bursting using PCQ is not the same as bursting with individual queues, but it still works quite well. The way we do this is the same way as limiting large downloads; we simply packet mark the data that is under a specific connection bytes. In our Mangle, we will specify connections that are between say from 0 and 1,0240,000 bytes and mark that data with a packet mark. We will then allow bursting on these it gives connections under 10 Megabytes the ability to burst. To allow the bursting, well we simply create a second set of PCQ types with higher PCQ-rates than our normal rates. So, we will have a burst-up, burst-down, standard-up and standard-down queue types. The standard up and down PCQ types may have a PCQ rate limit at 512k, while the burst queue types may have limits of 2 Mbps.

Setting up your simple queues is complicated though using this method. We will assume a 512k PCQ for normal rates, and 2M PCQ for our bursting. We also will assume we have a 3M Internet connection. First, we must have our queues in order! Remember rule order is important here. We want to separate the burstable data by using a packet mark, but that needs to be higher than the standard PCQ rule so that when it has a burst packet mark, it will not match our standard PCQ rule.



#		Name	Target Ad...	Rx Max Limit	Tx Max Limit	Packet Marks
0		Burst PCQ		3M	3M	Burst PCQ
1		Standard PCQ	10.0.0.0/8	3M	3M	
2		Parent Total		3M	3M	

Now that we have our order of importance, note that we have a Parent total rule. This is going to be a Parent of both of our PCQs; we have to know how much bandwidth we can allot as we only have a 3M Internet connection, so we need to still limit that.

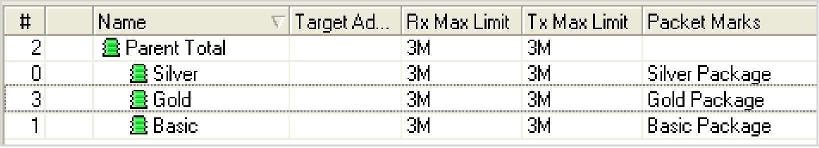


#		Name	Target Ad...	Rx Max Limit	Tx Max Limit	Packet Marks
2		Parent Total		3M	3M	
1		Standard PCQ	10.0.0.0/8	3M	3M	
0		Burst PCQ		3M	3M	Burst PCQ

Here's what happens -What occurs is that as new connections are being created, until they are at 10 Mbps of data transferred, the customer will be able to get data transfers up to 2M. This gives them quick access to small and short connections, but once they go over that 10 Mbps transfer rate per connection, they it then drops down to the standard PCQ rate and they no longer get that burstable speed. This is not as good as the actual bursting of data in the simple queues; however, it is an alternative if you are using PCQ.

Using Multiple Data Packages and PCQ

Using Multiple PCQ packages for different types of customers is as simple as creating multiple PCQ rules. The difference is instead of identifying traffic using IP addresses in your simple queues, you will need to use your Mangle system and mark traffic based on their package. The simplest way is to use address-lists of your customer IP addresses based on their package. You mark their data based on their IP address and then pass that mark to each of your Simple queues. Each simple queue has separate PCQ queue types with different PCQ rates according to your packages.



#		Name	Target Ad...	Rx Max Limit	Tx Max Limit	Packet Marks
2		Parent Total		3M	3M	
0		Silver		3M	3M	Silver Package
3		Gold		3M	3M	Gold Package
1		Basic		3M	3M	Basic Package

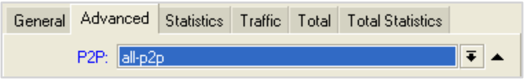
Each one of your customers is assigned to a specific get different PCQ bandwidth packages. You can also create a queue that has a limit-at that guarantees bandwidth over other queues. Keep in mind that you will also need to specify the SFQ queue type in your Parent. If you do this, you may have business customers that are guaranteed bandwidth while your other customers have no guaranteed bandwidth. You can simply change the priority as well in the queue to ensure your higher-priority customers get higher allocations of bandwidth as your Parent’s bandwidth becomes scarce.

Controlling P2P (Peer-to-Peer) Traffic

Many WISPs want to control peer-to-peer traffic. There are a few reasons for this; one is simply that P2P can create traffic when users are not at home, or at their PC. The bandwidth that is used is acting as a server, allowing other users to pull that data off of your customer's computer. Many broadband companies do not allow servers on broadband connections due to high data usage. An additional issue, especially for WISPs is access point capacity. Having many small connections in use simultaneously, streams a large amount of packets per second to your access point. This uses, using up access point capacity, and in the end slowing all users down.

RouterOS offers ways to help control P2P applications from eating up lots of bandwidth as well as ways to limit the number of connections that P2P applications can open up. The primary way to manage P2P is through the P2P matching rules built right into RouterOS. This allows you to create a simple queue that matches P2P traffic. This matching is actually done via Layer-7 stateful packet inspection; however, the method in which this is implemented is in the RouterOS system itself. Think of it as very, very highly-optimized Layer-7 filtering. Since this is done in the OS itself, we cannot change the matching parameters. Also, typically newer versions of RouterOS will use the latest matching capabilities, so if you wish to capture more P2P traffic, you will need to upgrade to the latest versions.

We will start by using a simple queue to control P2P traffic that flows through our RouterOS system. We will create a simple queue with only the P2P option selected. Inside this option you have the ability to select several different types of P2P systems. From Bit-torrent, to Kazaa and even edonkey P2P systems can be matched. You also have the option for all-p2p; this is the one that I typically would use. This gives you the ability to match your data based on this filter.



P2P: all-p2p

Rule order is important, if you list your customer base by IP address and have individual queues for them or a PCQ rule for all of your IPs, you typically will need to process this P2P rule before the others. Remember in the simple queue system, once the data is matched it will not be processed anymore. As an example, if you have queues for customer xyz based on their IP address, and then below that in rule order you have your P2P rule, the xyz customer will always get their allotted bandwidth on the first simple queue that was matched instead of being separating out the P2P data. If you move the P2P queue above the actual customer IP matching rule, any P2P that the customer uses will be matched first, and the rest of the data will be matched inside their normal queue. Doing this though, you will give the customer the bandwidth in the P2P rule for their P2P plus the bandwidth in their non-P2P queue. It is possible for the customer to pull more than their entire queue since they can also pull the amount of bandwidth in the P2P queue separately from their standard queue.

When I do configure P2P queues though, I match P2P across all data going through the RouterOS system; this bandwidth will get shared with all of the users on that system. If you have a P2P queue with a max-limit of 1M and individual queues of 1M, the chances of an individual pulling 1M of P2P is slim when all P2P is being matched for all customers and being grouped into that single P2P queue.

Limiting / Changing P2P and the Consequences

I have been asked many times to provide some form of summarization on the question of the legality of controlling P2P on your network is legal. First off, as it says in the beginning of this book remember, I am not a lawyer and do not claim to be. Recent events recently talk about controlling and changing the way P2P works on a private network. The fact that you control P2P on your network is not an issue typically, but how you control it is. In the cases in question, the network was not only controlling this data, it was changing and injecting its own responses into it. They were changing the data, and the way the application worked was to add these responses artificially making the application think the connection was closed.

This created a controversy as end users wanted their data unmodified and this process meant that other applications could be modified in some way to the benefit of the network provider. Even though the network operators said they have the right to control data on their network to ensure fair use for all users; it still was seen as an invasion.

Hotspots

A hotspot is a network access method that allows access to network resources based on some form of authentication. Most people incorrectly think of a hotspot being a Wi-Fi access point only, but hotspots can run on any TCP/IP medium including Ethernet, as well as wireless access points.   The main goal for hotspots is to allow users that are authorized to gain access to network resources (in most cases this is the Internet) over what would typically be an unsecured medium. Typically wireless hotspots are unsecured wireless access points. In some cases users can gain authorization to use the network resources by paying a fee. Most hotspot owners want users to pay for hotspot services (Internet access) by allowing enough usage to get the users to process a credit card in payment for Internet access.

Wireless and Hotspots

Chances are you have paid for Internet access at a hotspot location. It has become very common to have wireless hotspots. Setting them up is very easy as well with RouterOS. I wanted to touch though on one of the biggest common mistakes I see businesses and engineers doing with RouterOS and when it comes with wireless hotspots. Most of the hotspot users are going to be using some form of laptop or PDA to connect to your wireless access point. These devices have low power output, and a low gain antenna. For some reason, hotspot companies love to deploy high power radio cards in an effort to get more coverage per access point. Simply put, don't do this. There is no reason to place high power radio cards into an area where laptops are going to be the primary clients. The best method to provide the maximum coverage is low power radios with the largest antennas possible.

If you place high powered cards in your access point, you will be yelling at a client. That client, then whispers back to your access point.   Your access point may or may not be able to hear the client. If you are outputting high power Wi-Fi plus a quality high-gain antenna, the client will see a signal level that is good, but the response from the laptop will be very weak creating a false sense of the coverage area you actually have. Remember that your antennas have gain that both increases your transmit power as well as amplifies what the antenna hears.

Paid Hotspots

As a business owner, I like hotspots. The reason is because they can make me money. In areas that I already have Internet bandwidth available, I can place a paid hotspot system using RouterOS into an area that has many transient users, or users that come and go. They are willing, and allow them to pay for Internet services with a credit card and gain access to the Internet. The best part about these types of hotspots is that I don’t have to talk to the customer, take a credit card over the phone, have a 24-hours sales/support line, or do anything more than typically setup the system. The funds are deposited right into my account, so I don’t even have to take a check to the bank!

Paid hotspots are very common today, any place that people gather who would like to have Internet connections are a potential place for a hotspot. Hotels, and coffee shops are greats places, as well are restaurants, truck stops and rest areas.

Free Hotspots

Regardless of what many think, free hotspots can make money, and yes I did just use the phrase “make money” and the word free in the same sentence! Most free hotspots are not the main attraction. An example is a coffee shop or restaurant that puts in a free hotspot system to attract more coffee drinkers and business people. The idea is that now they can stay connected to their office with their laptops to their office even though they are having a coffee or lunch! These hotspots may exist solely for free as an added extra to your meal.

In the case of the coffee shop or restaurant, it’s a hard case to make money on a free hotspot, but if you have a hotel, gas station, truck stop or rest area, you can make money with a free hotspot! The idea is simple; you sell ads to businesses in the area that someone may be interested in! An example of this is a hotel that has a pizza shop that will deliver pizza to the hotel. When the hotel guest starts their web browser, they will get a splash page that contains the pizza shop ads before they are allowed on the Internet. Of course, this could be a great benefit to the local pizza place!

You can also boost the pizza business though paid hotspots, by offering the pizza place an ad on the splash page as well as FREE access to their website. The end user does not have to pay for Internet access to get on the pizza company’s website. This is another way to add value to your hotspot solutions as well.

RouterOS and Hotspots

RouterOS fully supports hotspots in many different ways. It offers an integrated security system for small hotspots as well as “trial users” to allow users limited access for specific amounts of time. You can also use a centralized Radius server to allow network access in addition to or instead of the built in security. User accounting, bandwidth controls, Firewalling, login or splash pages are provided. A walled-garden system allows access to resources without authentication. Automatic and transparent changing of any IP to a valid address is also supported.

Definitions

There are some definitions that you should know about before we get into the configuration of RouterOS with a hotspot system. We will cover those quickly so that you can get started!

Splash Page

The splash page is the initial page that RouterOS will display if a user is not authenticated. A new user will connect to the network, and upon starting their web browser, they will be redirected to the splash page. RouterOS supports customization of the splash page. This page is stored locally on the hotspot router, typically as login.html. There is also a redirect.html that points to the login.html file if you wish to do some form of redirection instead of displaying the page from RouterOS. RouterOS does have a built in web server to deliver these pages, however, there is no server side processing built in, so these pages should be simple html and client side application code. The default folder for the hotspot splash page, html and images, is called hotspot.

You can upload and change these files just like any other files in RouterOS. You can simply drag and drop them using WinBox, or you can FTP them as well. Common usages for your splash page is to present a login so that your users can login, links to websites that may be in your walled-garden, as well as links to sign-up systems to get a username/password for Internet access.

You may also have links to your business and; contact information for support may be listed as well.

Walled-Garden

These are resources that you are going to specifically allow users with no authentication to access. An example of this is that pizza shop's website we talked about in the free hotspot section. Items that you list in RouterOS walled-garden can be accessed by users without authentication to your hotspot. RouterOS has two walled-gardens; one is an IP walled garden, designed for you to enter IPs, protocols and ports into for allowing access. The second is the standard walled-garden. This one allows you to enter hostnames, and DNS names into the system to allow un-authenticated access.

Bindings

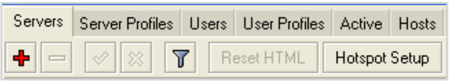
RouterOS offers an IP Binding system. This allow you to setup one-to-one NAT translation, allows you to bypass login/authentication requirements to specific hosts as well as allows you to block specific hosts and subnets from your hotspot system.

Hotspot Interface

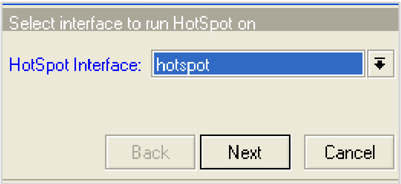
Hotspots run at Layer 2 in the OSI model, therefore they are applied to an interface. When you apply a hotspot to an interface, once the setup is complete, you will assume that all devices, MAC addresses, and IPs behind that interface must authenticate somehow. For this reason if you place a hotspot sever on the interface that you are currently running on, you will typically be disconnected from the RouterOS interface until you authenticate.

Setup of a Hotspot Interface in RouterOS

Setting up a hotspot interface on RouterOS is very simple. The first step is to place an IP on the interface as well as a subnet. In our example we will use 10.5.5.1/24 as the IP address on our hotspot interface. Place this IP on your interface. Next we will configure the hotspot. RouterOS makes a wizard that I recommend that you use. You will find this in the IP → Hotspot menu options.

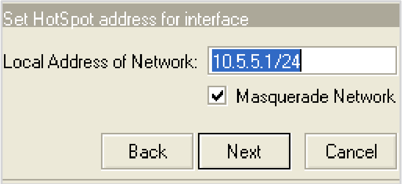


Here you will find a Hotspot Setup button. This wizard does a number of things that you will need to have or your hotspot will not function.


Select interface to run HotSpot on

HotSpot Interface: hotspot

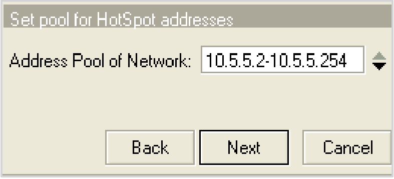
Step one is to select what interface you wish to use for your hotspot network. Remember, once this interface is completed with its configuration, it assumes everyone needs to be authenticated.


Set HotSpot address for interface

Local Address of Network: 10.5.5.1/24

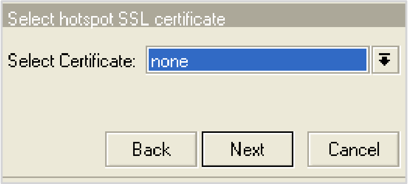
☑ Masquerade Network

Next, you will select the local address of your hotspot network. This address is the IP address of your hotspot network. In addition, you can choose if you wish to masquerade the hotspot network. During the setup process, it will add the correct NAT rule if you wish it to.


Set pool for HotSpot addresses

Address Pool of Network: 10.5.5.2-10.5.5.254

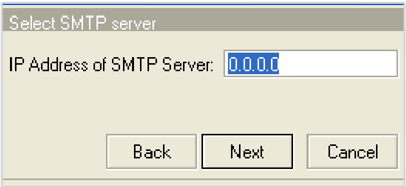
Here it creates an address pool to hand out DHCP addresses. You can select the IPs in the pool. Normally, you can accept the defaults here, however, sometimes if you have more access points out there that you would like to manage, you may reserve some IPs for these other devices on your hotspot network.



Select hotspot SSL certificate

Select Certificate: none

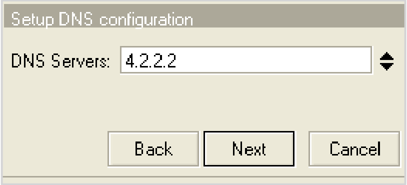
Next, it asks you if you have a certificate that you wish to use for the hotspot. Typically I don't use this, as I don't take any data that needs to be secure locally on the router.



Select SMTP server

IP Address of SMTP Server: 0.0.0.0

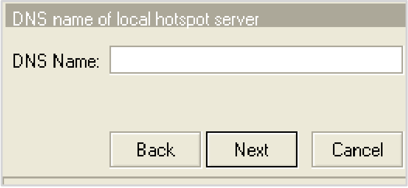
The SMTP address is the mail server you wish to redirect all TCP port 25 traffic too. This was common a while back, however, recently, I do not do this.



Setup DNS configuration

DNS Servers: 4.2.2.2

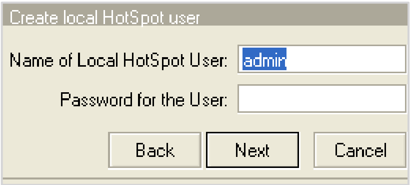
These are the DNS server IP addresses your hotspot system will use. Remember that DNS is a very important part of your hotspot system. List your DNS servers here.



DNS name of local hotspot server

DNS Name:

The DNS name is a name that you wish your customers to be redirected to by the hotspot for the delivery of the splash page. This DNS name does not have to be a publicly valid DNS name, as the hotspot system will add this DNS name in your DNS caching server automatically for you.

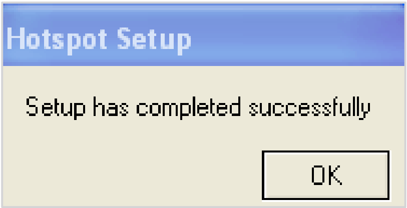


Create local HotSpot user

Name of Local HotSpot User: admin

Password for the User:

The last part of the setup process is to create a local username/password for authentication to the hotspot. If you don't do this, then there would be no username/password for you even to login with. You can delete it later, but it does get created with the wizard.



Hotspot Setup

Setup has completed successfully

Once all of those steps are completed, you should have a functional hotspot system on a single interface. The wizard does quite a few different functions inside a single simple to use setup wizard. It creates a hotspot server and server profile with your hotspot address and DNS name information. It creates your initial hotspot user as well. Then it creates a DHCP-Server with the correct DNS information, and DHCP IP pool to use as well. It also creates the IP pool that the DHCP-Server uses as well. Finally, it also enters static DNS information in if you entered a plopped in the DNS name in that step; otherwise, the hotspot IP address is used.

As you can see, there are a lot of functions that need to occur to configure a hotspot system in RouterOS system. I do recommend using the wizard because it does all of the things you need it to in one easy-to-use setup wizard.

Configuration of Servers and Server Profiles

The Servers and Server Profiles tabs give you the options that you will need to administrate your hotspot. The server's option shows you what the name of your hotspot is, the interface that it is running on, as well as the address pool and server profile that the server should use. One example of using the Server Profiles option here, is that one week you could have a conference hotel that normally has paid Internet access normally, and then a group of MikroTik people come in for a MUM where MikroTik pays for free Internet access. By having two different hotspot profiles, you can make a major change that includes the splash page, the method of authentication, and so on by changing a single option under the Servers tab.

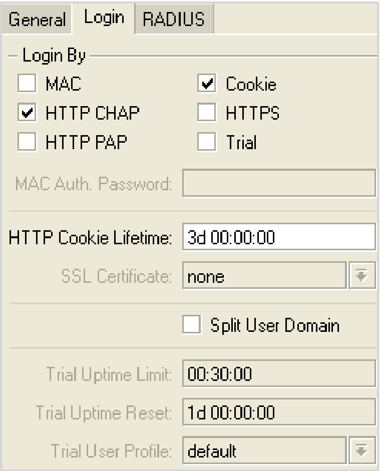
Inside the Servers options as well, you have the ability to reset the HTML code to the default RouterOS splash page. The folder that is reset is actually listed in the hotspot server profile that the server is using. Another option is the idle timeout. This option is important as this will prevent users from appearing to remain on-line even though they have actually departed. The default for this option is five minutes, and I have found this to be far short of what it should be. I typically set this for upwards of 20 minutes.

Under your hotspot Server Profiles, you have quite a few options. Here it will list the IP address of your hotspot, and the DNS name if you specified one in the wizard. You do NOT have to use a DNS name. If you don't, the hotspot will simply redirect to the hotspot IP address via IP vs. DNS name. You also have options for the hotspot folder that the splash page can be taken from. The rate-limit field is for the entire hotspot interface! If you have a 10 Mbps Internet connection and you don't want the hotspot to use more than 2

Mbps of that, this is where you would configure such a limit. This supersedes any individual user limits as well. The options to force users through an HTTP proxy and SMTP servers are listed towards the bottom on the General tab.

Hotspot Login Methods

RouterOS supports a number of login methods that you can use to authenticate users. These are configured in the hotspot Server Profiles under the Login tab. You can have several login methods if you wish at the same time. The MAC method uses the MAC address on the network to try to authenticate. Since the MAC is just a single line of letters and numbers, you can also specify a MAC Auth Password. The system will use the MAC address as the username and the password that you specify here together. These can authenticate through the local database or through Radius.



Login By

☐ MAC ☑ Cookie
☑ HTTP CHAP ☐ HTTPS
☐ HTTP PAP ☐ Trial

MAC Auth. Password:

HTTP Cookie Lifetime: 3d 00:00:00

SSL Certificate: none

☐ Split User Domain

Trial Uptime Limit: 00:30:00

Trial Uptime Reset: 1d 00:00:00

Trial User Profile: default

The default method for Logins is HTTP CHAP. The splash page that comes with RouterOS contains code for the browsers to CHAP encrypt the username/password. That along with the splash page allows users to type in their username/password. This is the simplest of hotspot logins and is supported by RouterOS. HTTP PAP is the same method, however, the username/passwords are sent in plain text. The HTTPS method is the same as HTTP PAP with the exception that you have a SSL certificate installed into RouterOS that the hotspot uses to create a secure connection with the users browser. I typically don’t use this method as the HTTP Chap method works quite well. I also am not taking any sensitive information via the web server on the RouterOS System, so I don’t think having to have a SSL page is necessary.

Hotspot Cookies

The cookie method is really an extension of the other HTTP methods, including the HTTPS method. Once the user logs in via their username and password, the MikroTik will generate a cookie to give to their browser. This cookie is good for the HTTP Cookie Lifetime value. This cookie has information in it to identify the user. If the users logs out or leaves and then comes back and connects to the network within the cookie lifetime, the browser delivers this cookie automatically to RouterOS and the hotspot system and if the user account is still valid, this cookie will log them in automatically; no need to type the username and password again. You can also look at any assigned cookies, as well as delete them via the Cookies tab under the hotspot interface.

Using Trial Users

Trial users are a special user inside the RouterOS system. The trial users feature allows users to perform a single-click login. This link on the splash page tells RouterOS to use their MAC address and create a username called T-*MAC ADDRESS.* This user is automatically created and they are allowed on-line. Typically, you will limit the user's ability to get on-line using this trial user feature by using the trial uptime limit. An example of using this is to allow anyone who wishes to use the Internet free access for one hour a day. When the trial user is created, they would have an uptime limit of 1 hour. This user can get on-line for one hour and then is presented the splash page again so that they can login. If they try to use the trial user link again, the code in the HTML will display that their time has been used. Once that user account is created, and then the user either logs off, or runs out of time, a second timer starts. This is the trial uptime reset timer; the default is one day. This timer would then remove that user account, along with the uptime limit after it has been reached.

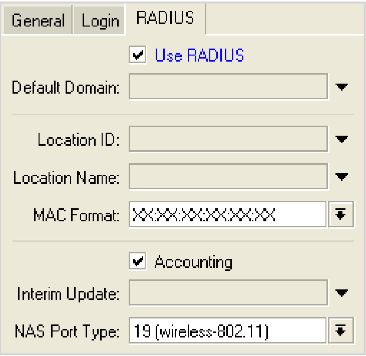
Most people do not completely understand how this feature works. We will follow a user for a moment. Assuming the uptime-limit is one hour and the uptime reset timer is set for one day, the user creates this account by clicking on the trial user link at 1 PM. The user uses the Internet for the full hour, and is then presented with the splash page again. There may be an option for that user to create a paid account, but let's assume that the user decides not to use this option. The next morning the user turns on their laptop and tries to connect, but their uptime limit has still been reached. The trial uptime reset timer starts when the user is logged out, at 2 PM. So their user account will not be reset till 2 PM the following day since we have a 24-hour reset timer. As you can see, this may not be the desired result. One method of fixing this is to simply shorten their uptime reset timer to around 8 hours. If they login at 8AM in the morning and, use an hour, by 5 PM they would be able to get on-line again. The second option is to have a script run around midnight that will delete all of the T-users in the database. You can find this script on the MikroTik WIKI. Describing the details is outside of the context of this book. If your script runs every night at midnight, then all trial users will be deleted, or reset, and the next day they will again be able to use their free service for one hour.

The last option in the trial users section is the trial user profile. This is the profile that the trial users should use when they login. This profile will deliver the user information, rate-limits, and filters that the trial user should use. Since the trial users are not identified via their own username/password combo, the trial user profile would be used for any trail user accounts that get created. This delivers the trail users speeds, and other options that normal users would have defined in their own profiles.

Your business model may dictate that you offer free internet access with restrictions, and then remove those restrictions if a user pays for internet access. We can use the trail users for users who do not wish to pay for internet access, but by using our trail user profiles, we can set filters for the trial users that prohibit most Internet activity with the exception of web surfing. We can also set the trail user profiles to ensure slower than normal access. We could even add websites that we don’t want the free Internet users (trial users) to get to. This gives incentive for the trail users, or our free users, to purchase internet access.

Hotspots with Radius

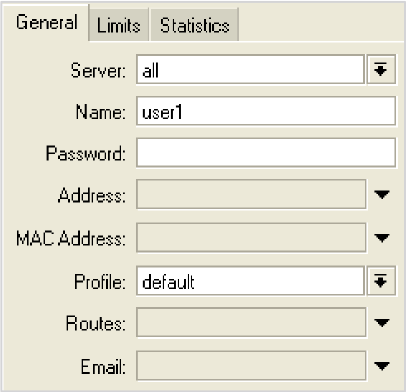
RouterOS has two methods of authenticating hotspot users. The internal method is to use the built-in user database that RouterOS offers. This is the Users and User Profiles tabs inside the RouterOS' Hotspot system. The built-in database is good for up to a few hundred users at most. Once you go past that you will need to go to an external database of some type. RouterOS supports Radius servers. Under the Server Profiles, you can click on the Radius tab to setup Radius authentication. Here you will have various Radius options as well as the ability to send Radius accounting information as well. The MAC format is used when you are doing MAC-based authentication with your hotspot. This tells RouterOS how to format the MAC address to send to the Radius server. Of course you will have to have your Radius server configured; we will discuss that in the Radius Server section.



☑ Use RADIUS
Default Domain:
Location ID:
Location Name:
MAC Format: XX:XX:XX:XX:XX:XX
☑ Accounting
Interim Update:
NAS Port Type: 19 (wireless-802.11)

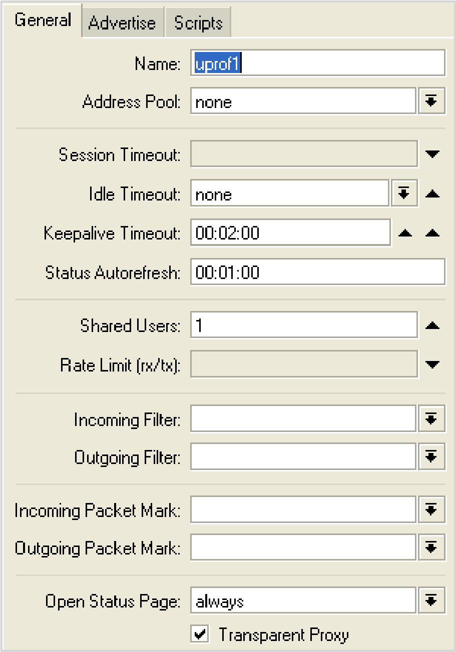
Internal Hotspot User Management

RouterOS does have a built-in user management system. This system is designed for a small number of users; I would recommend under a few hundred users. This user-management system is built into two separate portions. One is the Users tab, and the second is the User Profiles tab. The Users tab is what you use to create the actual user, set up-time limits, the user password, what user profile that user will use, as well as other identifying options. The User Profile tab specifies information such as idle timeouts, (if different than the server timeouts), the number of users that can share the profile, rate-limits, filters, packet marks, scripts and any advertisements.



Server: all
Name: user1
Password:
Address:
MAC Address:
Profile: default
Routes:
Email:

Under the Users section, you can specify if this user can login to all of the hotspots on the RouterOS system or just a specific one. You will define the username/password that the user will use here as well. The address/MAC information is used to also identify the user. Not only does the username and password have to be correct but the user must have the proper MAC and/or IP address as well to authenticate. You can also specify a route, when the user logs in, a route would be automatically created.



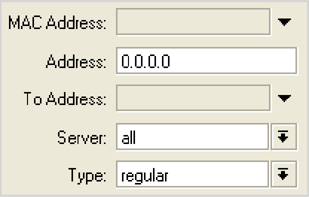
Name: uprof1
Address Pool: none
Session Timeout:
Idle Timeout: none
Keepalive Timeout: 00:02:00
Status Autorefresh: 00:01:00
Shared Users: 1
Rate Limit (rx/tx):
Incoming Filter:
Outgoing Filter:
Incoming Packet Mark:
Outgoing Packet Mark:
Open Status Page: always
☑ Transparent Proxy

Under the User Profiles, you have a lot of options here. They include changing your idle timeout from the value in the server settings, the rate-limit that the user will have, and any other filters or packet marks you wish to impose on users using this profile. The shared users are the number of active hotspot users with the same name that are allowed. This is a security feature to prevent username sharing. You can also force them through your web proxy system by checking the Transparent Proxy option.

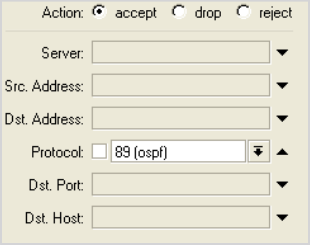
The advertising system is also located under the User Profiles. The way this works is that you will specify an "advertise" URL. This is a page with the ad that you wish to display. This will be displayed on the advertising interval time. If the advertisement is not loaded within the advertisement timeout value, then network access is restricted until that advertising URL is displayed.

The advertising system uses a pop-up to display advertisements, and due to this fact, may not work for every user the same way. Many users have a pop-up blocker running that would disable the pop-ups from coming up; and this could lead to confusion by the end user. I recommend spending some time in front of your computer behind a hotspot with advertising to really get a feel for what your users might experience before you deploy it.

Using IP Bindings



MAC Address:
Address: 0.0.0.0
To Address:
Server: all
Type: regular

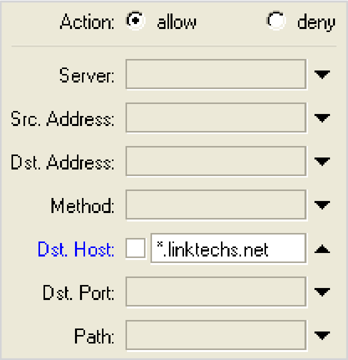


Action: ● accept ○ drop ○ reject
Server:
Src. Address:
Dst. Address:
Protocol: □ 89 (ospf)
Dst. Port:
Dst. Host:

IP Bindings is a way to setup one- to-one NAT translations; it's also used to bypass hotspot clients without authentication. You can also use it to block specific hosts or subnets as well.

The most common feature I use IP bindings for is to bypass other access points and routers IPs addresses for management. If you have a separate bridged access point behind the hotspot interface with an IP on it, you can't even ping it from your hotspot router! You will need to bypass it to be able to ping it, SSH or telnet into it. You can specify not only the IP address as well as the MAC address. Make sure that the IP address is not part of a DHCP pool that could be given out to other users as well, because that could create addressing problems that will mess up the works. The best way of doing this is to attempt to ping your device from your hotspot RouterOS system. This will create a host for the device that you are trying to ping. This will capture both the MAC address and IP address from that host. Then use your host list to copy those addresses into your bindings system; this will prevent typos with the MAC and IP addresses.

Creating Walled Garden Entries



Action: ● allow ○ deny
Server:
Src. Address:
Dst. Address:
Method:
Dst. Host: □ *.linktechs.net
Dst. Port:
Path:

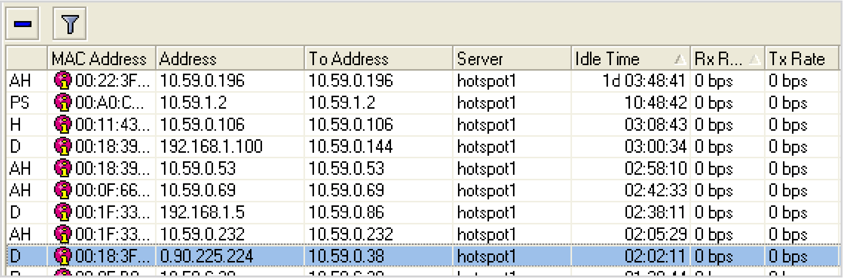
The walled garden gives you the ability to allow unauthenticated users to gain access to specific resources. These resources are defined in the walled garden. RouterOS has two different walled garden systems. The standard walled garden is used to bypass HTTP and HTTPS resources. If you want to allow customers to access to www.linktechs.net for example, you would put www.linktechs.net in the dst-host field of a walled garden entry. You can also allow sub domains by putting in *.linktechs.net as well.

The second walled garden that RouterOS supports is the IP walled garden. This is used for Protocols and ports, as well as IP addresses. Things like DNS requests, WinBox, and so on, would be defined here. In the screen shot, I have configured the rule to accept OSPF packets into the router from the hotspot interface.

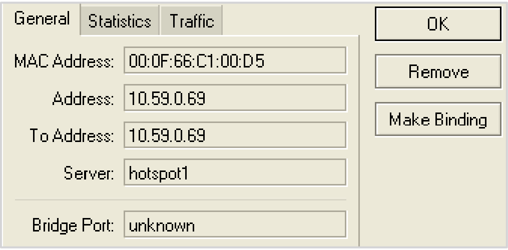
You could also specify a specific server IP addresses here, or maybe you wish to allow pings regardless if they are authenticated or not. One thing I do quite often is to allow both TCP and UDP Port 53 requests. Hotspot uses DNS to allow for the hotspot splash page requests, and DNS resolution is required, so I open that up as well.

Viewing Hotspot Hosts and Active Users

The Hotspot Hosts tab is a wealth of information. Here you will see all of the hosts that are on the hotspot network. You can see their MAC address information, IP address, and also see if they have a translated address. If you look down in the image below, you will see users with addresses that are not part of the 10.59.x.x network; these are customers that have static IPs in their PC. RouterOS will use a feature called Universal Client to renumber and perform one-to-one NAT. Note the second To Address column.



	MAC Address	Address	To Address	Server	Idle Time	Rx R...	Tx Rate
AH	00:22:3F...	10.59.0.196	10.59.0.196	hotspot1	1d 03:48:41	0 bps	0 bps
PS	00:A0:C...	10.59.1.2	10.59.1.2	hotspot1	10:48:42	0 bps	0 bps
H	00:11:43...	10.59.0.106	10.59.0.106	hotspot1	03:08:43	0 bps	0 bps
D	00:18:39...	192.168.1.100	10.59.0.144	hotspot1	03:00:34	0 bps	0 bps
AH	00:18:39...	10.59.0.53	10.59.0.53	hotspot1	02:58:10	0 bps	0 bps
AH	00:0F:66...	10.59.0.69	10.59.0.69	hotspot1	02:42:33	0 bps	0 bps
D	00:1F:33...	192.168.1.5	10.59.0.86	hotspot1	02:38:11	0 bps	0 bps
AH	00:1F:33...	10.59.0.232	10.59.0.232	hotspot1	02:05:29	0 bps	0 bps
D	00:18:3F...	0.90.225.224	10.59.0.38	hotspot1	02:02:11	0 bps	0 bps

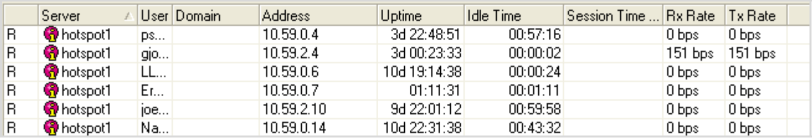


MAC Address: 00:0F:66:C1:00:D5
Address: 10.59.0.69
To Address: 10.59.0.69
Server: hotspot1
Bridge Port: unknown

On the left we have letters that identify what the host is currently doing. "A" is for an active host, this would be a host that has been authenticated. "D" is dynamic hosts; these are typically customers that the universal client had to dynamically assign them a valid address to get them to work.

One of the best tools you can have for managing your hotspot is right here. If you double click the hosts, you will have several tabs as well as traffic, statistics plus one major button I use all of the time. The Make Binding button gives you the ability to take this host and create a binding from it. This will ensure that you don't typo either the MAC or the IP address.

The active list is users authenticated via either Radius or the internal database. This list will show you their uptime, idle time, along with the username that they have used to authenticate with, as well data rates etc. Here you can bump users from being logged in as well. Keep in mind that if you are using cookies, you will also need to remove their cookie before bumping them from the active user list; otherwise, they may sign right back in using their cookie.



	Server	User	Domain	Address	Uptime	Idle Time	Session Time ...	Rx Rate	Tx Rate
R	hotspot1	ps...		10.59.0.4	3d 22:48:51	00:57:16		0 bps	0 bps
R	hotspot1	gjo...		10.59.2.4	3d 00:23:33	00:00:02		151 bps	151 bps
R	hotspot1	LL...		10.59.0.6	10d 19:14:38	00:00:24		0 bps	0 bps
R	hotspot1	Er...		10.59.0.7	01:11:31	00:01:11		0 bps	0 bps
R	hotspot1	joe...		10.59.2.10	9d 22:01:12	00:59:58		0 bps	0 bps
R	hotspot1	Na...		10.59.0.14	10d 22:31:38	00:43:32		0 bps	0 bps

Running Multiple-Subnets Behind a Hotspot Interface

Yes you can have routable subnets behind a hotspot interface and behind other routers; however, in your hotspot server configuration you need to make a few changes. You will lose some of the security that occurs with the hotspot server configuration. Normally hotspots use not only the IP address of the client, the username/password of the client connection, but also their MAC addresses as well. RouterOS provides security by using all of these together. However, when you place other routers behind your hotspot interface, the only MACs you will see is the MAC address of the forward facing interface from the router. You will end up with 20-30 of the same MAC address, one for each of the IPs on the second subnet. By default though, RouterOS will not allow this. To correct this, you will need to make two changes.

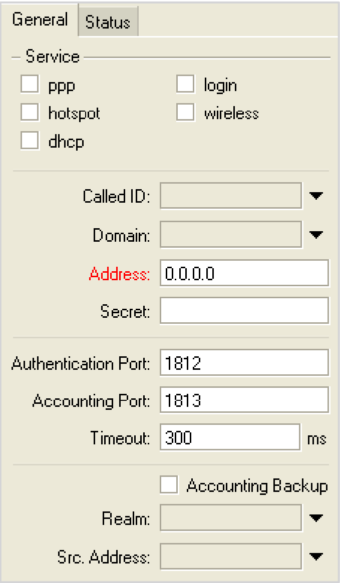
The first change to make this work is to change the address pool under your hotspot server settings to none. The reason for this is that the IP addresses under the other subnet are valid; therefore there is no reason to do NAT translation of those. You can do this if you wish, however, you will need to watch your IP pool as you now have another entire subnet using the same IP pool, and you can quickly run out of IP addresses if you are not careful. The second, and more important change, is the address-per-mac setting. This is still in the hotspot server settings. This settings prevents a MAC address from getting more than so many IP addresses. Normally, this is defaulted to two addresses, but as I said, with another subnet behind your hotspot, you will always see the MAC of the router that the other subnet is connected on. You will typically need to change this number to something really high or disable it.

Running Dynamic Routing (RIP/OSPF) Behind a Hotspot Interface

This can be done as well, however, since all IP addresses behind the hotspot interface have to be authenticated in some way. When running some form of dynamic routing protocol, there can be a problem that occurs because the other routers are not bypassed, or authenticated in some way, the dynamic routing protocol packets are not accepted and are dropped. This is the correct thing that the hotspot should do on the hotspot interface. It is correct because the routers are not authenticated, therefore, the dynamic routing updates that they send out, are dropped. This is not the desired effect, but the Hotspot is doing what it's supposed to be doing by default.   There are two methods to solve this. The quickest is to simply bypass (in IP-Bindings) the IP/MAC of the neighboring router. This will allow data to flow to and from that router without issues. The second is to simply allow the routing protocols transport method through. In the case of OSPF, you would allow the OSPF protocol in the walled garden.

Radius Client

RouterOS fully supports Radius standards. The Radius setup is twofold. The first is configuring the Radius server under the Radius option. Here, you will define Radius servers and the information for those servers. You will need the IP address of the Radius server, and the secret. If the authentication or accounting port is different than the standards, then you can change it here as well as other information. The timeout value is important though, as this is the time that the RouterOS Radius client will wait for the Radius server to reply. If you have a fast Radius server, then 300 ms should be fine. Keep in mind that if you ping your Radius server from the MikroTik Radius client, and that RTT (round-trip-time) is 100 ms, then the server has an additional 200 ms to respond. Sometimes you may find it simpler to increase this to around 1000 ms.



Service

□ ppp □ login
□ hotspot □ wireless
□ dhcp

Called ID:
Domain:
Address: 0.0.0.0
Secret:
Authentication Port: 1812
Accounting Port: 1813
Timeout: 300 ms

□ Accounting Backup

Realm:
Src. Address:

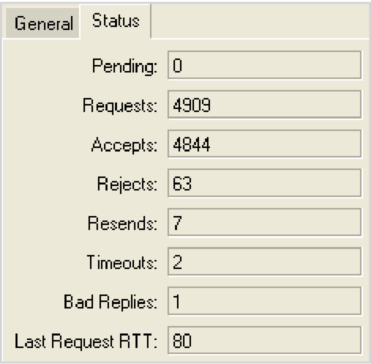
The service checkboxes in the radius client defines what service the Radius server is responsible for. You can have the same Radius server doing authentication for multiple services at the same time. If you check PPP as a service, that means this Radius client can be used for PPP Authentication attempts. If you select DHCP, then the said server can be used for DHCP Authentication attempts. These are basically what services in RouterOS this Radius client can be used for. If you had two separate Radius servers, one for router logins and one for hotspot logins, then you would have two Radius clients, one with the service of login and one with the service of hotspot. Even if the login service Radius client was listed first in the Radius clients section, the hotspot system would not use that Radius server to authenticate against.

The second portion is the service that you are configuring to use Radius. We cover this in each individual section, so if you wanted to configure your hotspot system, you will have to configure the Radius check box under the hotspot profile. For your PPP service, you must configure your PPP system to use Radius just like any other service. Once both of these are done, then you can start using your Radius client!

Multiple Radius Servers

When you setup multiple Radius servers with the same service, there is an order that occurs. The first Radius server will be used based on the ordered list. However, if that Radius server DOES NOT RESPOND within the time-out value, then it will go to the next Radius server for that service on the list. Note that if the Radius server DOES respond, regardless if the response is a deny or accept, RouterOS will not try another Radius server. The Radius server must NOT respond, for RouterOS to move through the list.

Troubleshooting Radius Client Issues



Pending: 0
Requests: 4909
Accepts: 4844
Rejects: 63
Resends: 7
Timeouts: 2
Bad Replies: 1
Last Request RTT: 80

Troubleshooting Radius client issues is typically very simple. There are only a few things that can go wrong. Looking at the Status window, you will see the number of Requests, Accepts, and Rejects. If you are getting Rejects or Accepts, then that means the Radius server is responding to your request. This would show that everything is working correctly, at least with Radius client to server communication. If the Radius server is getting Timeouts, you will have to look at several things. First, is to check the IP and Secret of the Radius server. If those look correct, check the Radius server to ensure that the proper Radius client IP is listed. Remember, your Radius server will default to use the IP on the interface closest to your Radius server.

If you are getting some Accepts and Rejects, but also are getting Timeouts, check the last request RTT time. This is the turnaround time it took to get a response back from your Radius server. In the image above it is taking 80 ms. If your timeout value is under 80, then you will get more timeouts, however, if you see higher times, you may need to simply bump your timeout value accordingly to give your server time to respond.

MPLS

MPLS or Multiprotocol Label Switching (MPLS) is designed to be a high-performance method to carry data from one network node to the next using labels. Using MPLS you can create virtual links between distant nodes, and encapsulate several different network protocols including Layer 2 packets. This ability allows you to create MPLS links that are bridged or routed links. One of the more complex configurations is the ability for you to create a separate Layer 3 OSPF network inside yours for your customer/client. So not only do you maintain your own IGP tables for your own routing, but you also can have separate routing tables for your customer's sites! Sounds fun, huh!

MPLS operates at what most people consider Layer 2.5. It actually operates between Layer 2 and Layer 3. It was designed so that it can carry both Layer 2 or circuit based information, and packet-switching clients. You also can deploy MPLS in different technologies a well, anything from Ethernet, wireless, ATM and even SONET. You can also encapsulate those types of packets as well transmitting them as a circuit if you wish.

One of the major advantages when MPLS was first conceived was that MPLS label switching can occur inside the switch chip of a switch. All of a sudden this created a much faster and less expensive device in comparison to an equivalent router. Switches can switch millions of packets per second, and typically can do that at wire speed on all of the ports. So the idea was to create a system that the switch chips can handle, hence, lowering the cost of ownership.

A second advantage of using MPLS labels is that you also cut down on the lookup time. Typically when a packet is forwarded via IP based routing; we have to look at the IP header information in the IP packet. This header information in an IPv4 packet averages around 40 bytes. This header has to be examined on every packet as it goes through the router. Using MPLS, the only thing that has to be examined is a 4 byte header, the MPLS packet label. So if you think about it, the MPLS lookup can be upwards of 10 times more efficient than standard IP routing due to simply the lesser amount of data that the router has we have to look at. This number is not realistic, but MPLS lookups are going to be faster than the IP lookups.

RouterOS supports many features of MPLS, and using MPLS will allows you to create both Layer 2 and Layer 3 tunnels across your network, increase the performance of your network and create virtual networks inside your network.

Getting Started with MPLS

One of the first things that you must realize is that you still need a routed network for MPLS to run over! Yes; that right! MPLS forms the labels across a routed network, so you may wish to read the dynamic routing, and OSPF sections to understand how to properly dynamically route your network. As I said, this is required to get MPLS running. In our example, we will assume that you have several devices all running with OSPF.

Remember MPLS is designed to get you from one point to another point on your MPLS cloud, or MPLS network; it’s really not designed to get you connected to the Internet. Not saying that it can’t do that, but it’s not designed to replace your default route and/or your OSPF network. However, But to get from point A to point B on your network, MPLS can reduce the network load and speed up the process of getting from point A to point B.

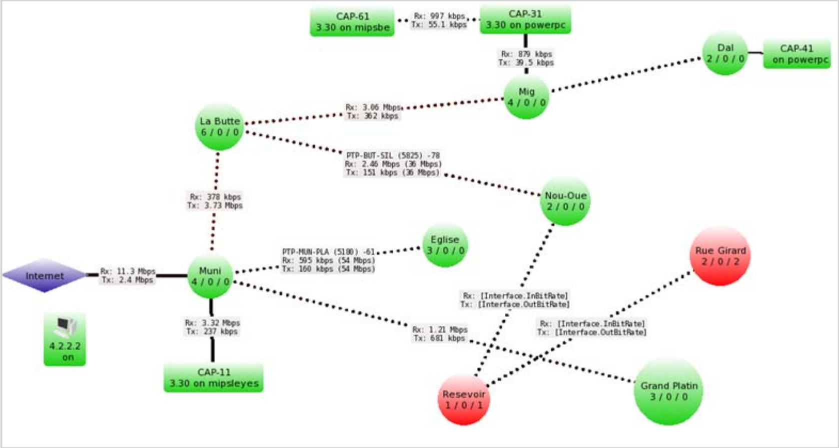
MPLS is actually a collection of items and terms. So you need to make sure you know the terms and how they are used. This is a very confusing portion of MPLS for many people. The First, MPLS, again, is a collection of terms. By itself it uses LDP or Label Distribution Protocol, that each LER or Label Edge Routers exchange label information. These labels are generated and exchanged with each other, as more and more devices are added, each device exchanges the information it knows with its neighbor and therefore a forwarding table is created. By exchanging LDP information and building the MPLS forwarding table, you end up with MPLS forwarding packets based on labels, vs. routing. This is the first step with MPLS.

So just by turning on LDP and allowing your routers to communicate and exchange LDP information, you are now running MPLS. This should be just a faster "routing" system, by using the labels instead of the entire IPv4 header information.

The second part of MPLS is actually called VPLS, but you have to be running a MPLS network, i.e. exchanging that LDP information prior to deploying VPLS. VPLS or Virtual Private LAN Service provides you the tunneling over the MPLS network. The main feature of this is to create a private LAN services over your existing layer 3 networks using the MPLS label switching to ensure fast connectivity. In its simplest form, the VPLS circuit is simply two MPLS routers forming an EoIP Tunnel. EoIP though is processor intensive where MPLS should have greatly reduced CPU usage in comparison. We will discuss VPLS in much more detail later in the chapter.

MPLS Configuration

MPLS configuration is quite simple, let's assume we have an existing network. The image below is going to be our test network. In this network, we have a single internet connection and multiple sites. These sites have, in some cases a number of access points, and each access point will be tunneled back via Ethernet layer 2 tunnels to the core. The core router, at the MUNI site, will host a PPPoE server connecting all customers. Assume in this case that we already have a layer 3 routed network, this can be via OSPF or RIP, but will be routed.

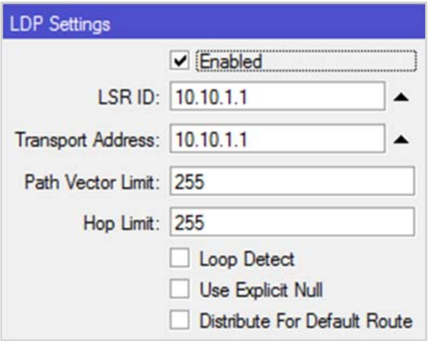


The first thing to get started with configuration of this network, is to setup and start using LDP, this will create our forwarding table via LDP exchanges, and give us the ability to use MPLS as our forwarding protocol instead of IP forwarding.

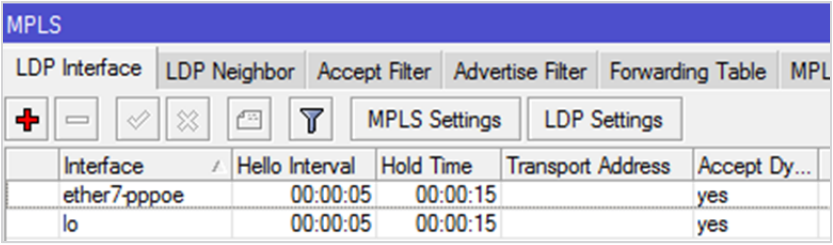
Before we start with configuration of LDP and MPLS, we will need to know that when we build our tunnels later, as well as good general practice, all of our routers should be configured with a loopback address. If you have read the OSPF section, you will know that it’s a good idea to use loopback addresses on all of your routers for remote access and monitoring. VPLS will make better use of these loopback addresses for tunnel establishment.

Configuration of LDP

To get started, on each router we will need to configure LDP. Simply by enabling it in the LDP Settings menu, will get us started, however, you should use the loopback address mentioned in the above section for your LSR ID and transport address.



☑ Enabled
LSR ID: 10.10.1.1
Transport Address: 10.10.1.1
Path Vector Limit: 255
Hop Limit: 255
□ Loop Detect
□ Use Explicit Null
□ Distribute For Default Route



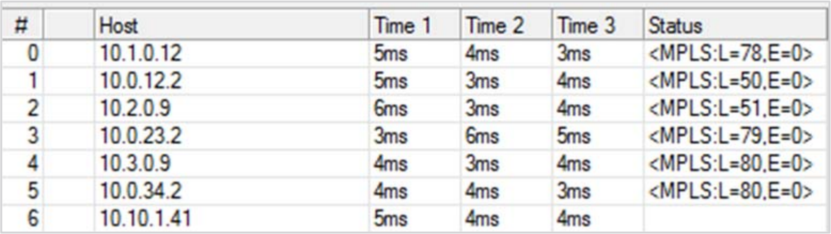
Interface	Hello Interval	Hold Time	Transport Address	Accept Dy...
ether7-pppoe	00:00:05	00:00:15		yes
lo	00:00:05	00:00:15		yes

The second thing you need to do is to configure the LDP interface. This is simply a list of interfaces that your LSR will communicate to other LSRs on. Simply put, if you want to talk LDP on an interface, you need to have that interface listed here.

Testing MPLS Configuration

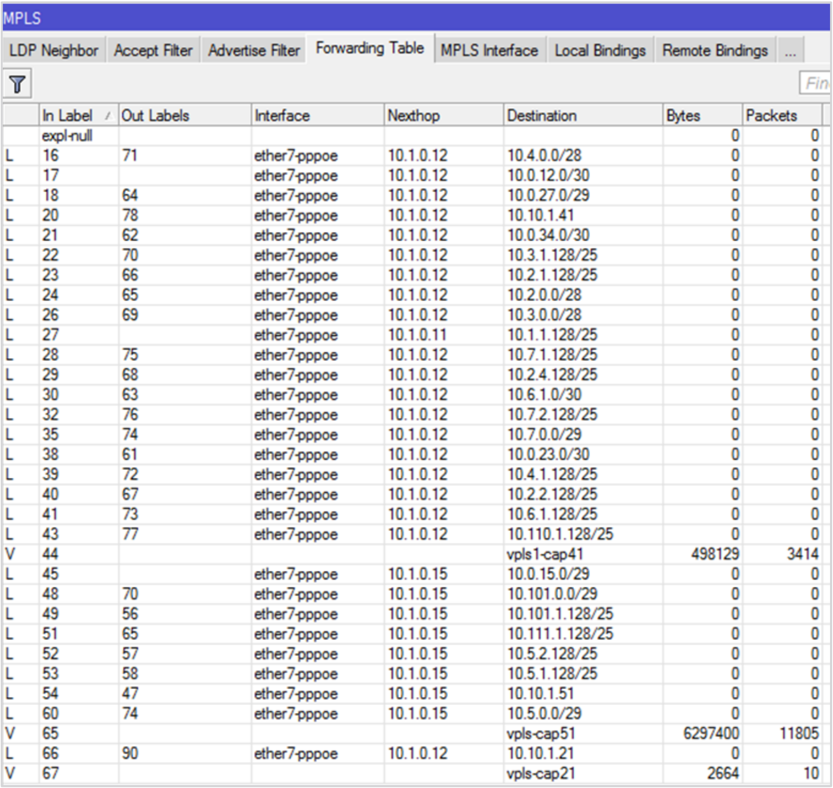
So if you have done the two steps above on all routers between point A and B, then you should be using MPLS to forward packets instead of standard IP forwarding. However, you will need to check this, and sometimes this can be difficult. With RouterOS though we can see MPLS labels very quickly, we will simply do a trace route to the CAP-41s loopback address.

Note that we have MPLS labels showing up under our stats, these MPLS labels show that we are using MPLS forwarding.



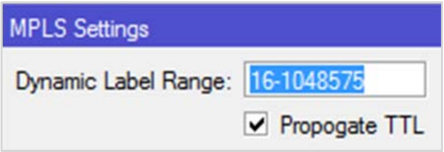
#		Host	Time 1	Time 2	Time 3	Status
0		10.1.0.12	5ms	4ms	3ms	<MPLS:L=78,E=0>
1		10.0.12.2	5ms	3ms	4ms	<MPLS:L=50,E=0>
2		10.2.0.9	6ms	3ms	4ms	<MPLS:L=51,E=0>
3		10.0.23.2	3ms	6ms	5ms	<MPLS:L=79,E=0>
4		10.3.0.9	4ms	3ms	4ms	<MPLS:L=80,E=0>
5		10.0.34.2	4ms	4ms	3ms	<MPLS:L=80,E=0>
6		10.10.1.41	5ms	4ms	4ms	

Something else you can look at is the MPLS → Forwarding Table in WinBox. As you add more and more LDP neighbors and they exchange data, just like an IP Routing table, your forwarding table in MPLS will grow.



	In Label	Out Labels	Interface	Nexthop	Destination	Bytes	Packets
	expl-null					0	0
L	16	71	ether7-pppoe	10.1.0.12	10.4.0.0/28	0	0
L	17		ether7-pppoe	10.1.0.12	10.0.12.0/30	0	0
L	18	64	ether7-pppoe	10.1.0.12	10.0.27.0/29	0	0
L	20	78	ether7-pppoe	10.1.0.12	10.10.1.41	0	0
L	21	62	ether7-pppoe	10.1.0.12	10.0.34.0/30	0	0
L	22	70	ether7-pppoe	10.1.0.12	10.3.1.128/25	0	0
L	23	66	ether7-pppoe	10.1.0.12	10.2.1.128/25	0	0
L	24	65	ether7-pppoe	10.1.0.12	10.2.0.0/28	0	0
L	26	69	ether7-pppoe	10.1.0.12	10.3.0.0/28	0	0
L	27		ether7-pppoe	10.1.0.11	10.1.1.128/25	0	0
L	28	75	ether7-pppoe	10.1.0.12	10.7.1.128/25	0	0
L	29	68	ether7-pppoe	10.1.0.12	10.2.4.128/25	0	0
L	30	63	ether7-pppoe	10.1.0.12	10.6.1.0/30	0	0
L	32	76	ether7-pppoe	10.1.0.12	10.7.2.128/25	0	0
L	35	74	ether7-pppoe	10.1.0.12	10.7.0.0/29	0	0
L	38	61	ether7-pppoe	10.1.0.12	10.0.23.0/30	0	0
L	39	72	ether7-pppoe	10.1.0.12	10.4.1.128/25	0	0
L	40	67	ether7-pppoe	10.1.0.12	10.2.2.128/25	0	0
L	41	73	ether7-pppoe	10.1.0.12	10.6.1.128/25	0	0
L	43	77	ether7-pppoe	10.1.0.12	10.110.1.128/25	0	0
V	44				vpls1-cap41	498129	3414
L	45		ether7-pppoe	10.1.0.15	10.0.15.0/29	0	0
L	48	70	ether7-pppoe	10.1.0.15	10.101.0.0/29	0	0
L	49	56	ether7-pppoe	10.1.0.15	10.101.1.128/25	0	0
L	51	65	ether7-pppoe	10.1.0.15	10.111.1.128/25	0	0
L	52	57	ether7-pppoe	10.1.0.15	10.5.2.128/25	0	0
L	53	58	ether7-pppoe	10.1.0.15	10.5.1.128/25	0	0
L	54	47	ether7-pppoe	10.1.0.15	10.10.1.51	0	0
L	60	74	ether7-pppoe	10.1.0.15	10.5.0.0/29	0	0
V	65				vpls-cap51	6297400	11805
L	66	90	ether7-pppoe	10.1.0.12	10.10.1.21	0	0
V	67				vpls-cap21	2664	10

MPLS Settings -Propagate TTL



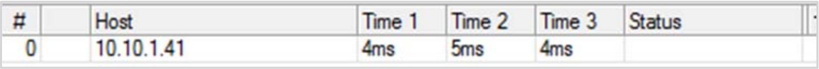
MPLS Settings

Dynamic Label Range: 16-1048575

☑ Propogate TTL

The MPLS settings are very basic, but it includes a KEY feature. The dynamic label range is exactly what it sounds like; it is simply the range of label numbers that the LDP system can apply to create dynamic label entries.

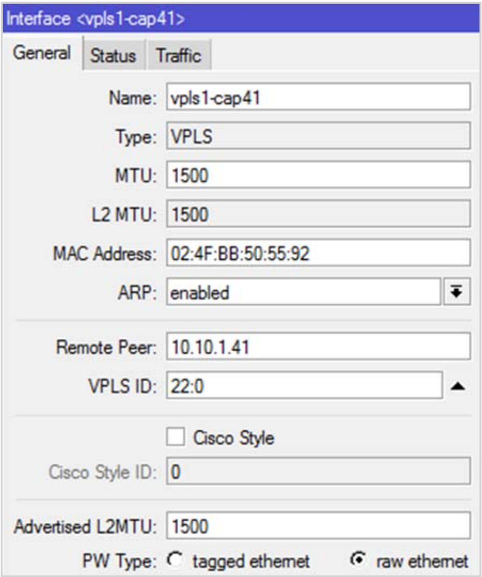
That's not the feature we are interested in, the Propagate TTL is though. This feature basically does what it says, does it propagate the TTL or not. By default this will be checked, and the above trace route is what you will see when tracing from the router, or from a non-LDP enabled port. However, if you un-check this, any of your MPLS forwarded hops the TTL will NOT be propagated, showing only ONE hop for ALL of your MPLS. The image below is the same router, pinging the same end router, but with TTL turned off.



#		Host	Time 1	Time 2	Time 3	Status
0		10.10.1.41	4ms	5ms	4ms	

VPLS

VPLS or Virtual Private LAN Services, create a point to point or point-to-multipoint private network. Think creating end points, such as offices etc., and each of these end-points are part of the same LAN. VPLS creates these types of tunnels over MPLS. In our example network for the MPLS section, we are creating VPLS tunnels over MPLS to allow PPPoE-Clients to contact the PPPoE server at the core of the network. In this case, we will simply need point-to-point VPLS tunnels from each access point back to the core router.



Interface <vpls1-cap41>

Name: vpls1-cap41

Type: VPLS

MTU: 1500

L2 MTU: 1500

MAC Address: 02:4F:BB:50:55:92

ARP: enabled

Remote Peer: 10.10.1.41

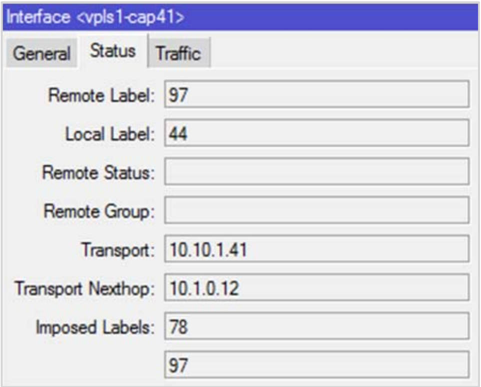
VPLS ID: 22:0

□ Cisco Style

Cisco Style ID: 0

Advertised L2MTU: 1500

PW Type: ○ tagged ethernet ● raw ethernet



Interface <vpls1-cap41>

Remote Label: 97

Local Label: 44

Remote Status:

Remote Group:

Transport: 10.10.1.41

Transport Nexthop: 10.1.0.12

Imposed Labels: 78
97

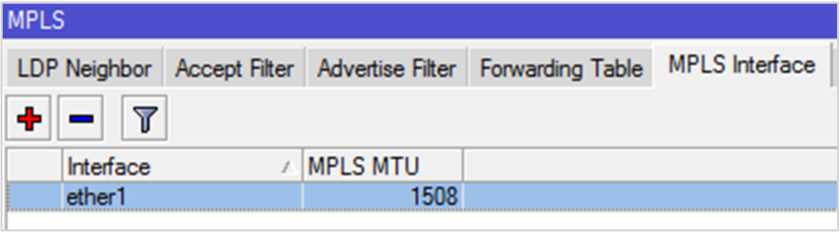
MPLS → VPLS is where you will configure these tunnels. Each tunnel you must connect at each end, i.e. you will configure each end of the tunnel, just like an EoIP or PPTP tunnel. Also note that these tunnels are not encrypted nor do they operate outside of your MPLS network, i.e. they can't go across the internet without an interconnect system. Again, in our example, we are delivering these tunnels to a PPPoE server ON our network, so these tunnels will work.

On each end we will create a VPLS configuration. We will use the loopback address for the remote peer as well as the same VPLS ID. This ID is for identifying multiple VPLS tunnels with the same endpoint, but actually separating them as needed. In many cases these settings are all that is needed.

These interfaces are created and are stateful, i.e. they only become running once they have connected and are actually working. Under the status tab you will note that there are a number of labels and transport addresses that have been determined once the interface is running.

Since the PW type is 'raw Ethernet' this tunnel acts as a layer 2 bridge, and typically would be added to a bridge group as a bridge port. This will take data from one bridge port and pass it over the VPLS tunnel interface. This is the simplest form of VPLS tunneling.

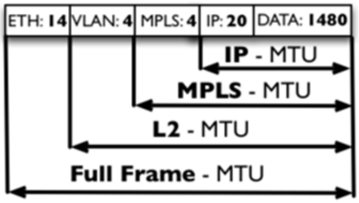
In the past, we have seen some trouble with creating VPLS point-to-point tunnels with the MPLS interfaces set to ALL. We simply change these to any of the actual interfaces to be used, and typically this issue has been corrected. An example of an issue that could occur is that one side of the VPLS tunnel says up, but the other does not, i.e. not running. In this event, on the side that is not running, change the MPLS interfaces to just the interfaces that you need.   You can see the example in the image below, we simply removed the ALL interface and changed it to the interface that we talk MPLS on, and in our case we only talk MPLS on Ethernet 1.



Interface	MPLS MTU
ether1	1508

MPLS/VPLS MTU Issues

In many cases you can have MTU issues with MPLS especially if you are adding more items on top, such as VPLS etc. Some RouterBOARDs also don't have large L2MTU support, therefore are limited in the amount of MTU you can pass.



In this image, we have a good example of L2MTU sizes. 1500 is the standard MTU for Layer 3 type of traffic plus we add the 14 bytes for the Ethernet frame information. This gives you a L2MTU (L2 includes the Ethernet frame information) that is required. Remember, MPLS is a Layer 2.5 protocol, so you should not have to modify the MTU as that’s a Layer 3 setting.

If you are using VPLS, you assume, the 1480 in data, 20 for IP, two MPLS headers, at 4 bytes each, a VPLS header and then the Ethernet fame, giving you a 1526 byte L2MTU. With that said as long as all MPLS routers have a L2MTU on all interfaces at 1526 or higher, that means VPLS will be able to send full 1500 byte packets without fragmentation. But, don't distress over this, as if you set the MPLS Interface MTU to 1508, the system will fragment those packets for you, and so data will flow.

One major concern though, is with non-VPLS traffic, just straight MPLS forwarding. In this event, if you are passing MPLS data, you have 1500 bytes for the IP/data, another 8 for two MPLS headers and 14 for your Ethernet; this gives you a L2MTU of 1522 bytes. The current line-up of RouterBOARDs support at least a 1522 L2MTU, some more, the older RB100 series did not. Assuming this, most people will be fine; however, if you go through a device that does not have a 1522 L2MTU, MPLS, being a layer 2.5 protocol, does not have a way to correct this error, and therefor drops the data. This becomes more aggregated with VLAN / VPLS forwarding. In these cases L2MTU needs to be upwards of 1530 bytes due to the size of the packets. 1500 bytes just in the IP / data, plus 4 for VLAN information, two MPLS 8 byte headers, one 4 bytes VPLS header, and this is where it stinks, two Ethernet headers, because the VLAN information must be encapsulated inside the VPLS packet. So simply, 1480 (data), 20 (ip), 4 (vlan), and 14 (Ethernet) is just the data with the VLAN and Ethernet information giving you 1518 bytes before you get to the MPLS/VPLS information. We add 8 (two MPLS), 4 (VPLS), plus your 14

(Ethernet) giving you a grand total of 1530. Now we get into the world where a number of RouterBOARDs interfaces and other devices can't support.

For L2MTU's for RouterBOARD products please refer to MikroTik's WIKI site for the latest information.

MPLS RSVP TE -- MPLS Resource Reservation Protocol - Traffic Engineering

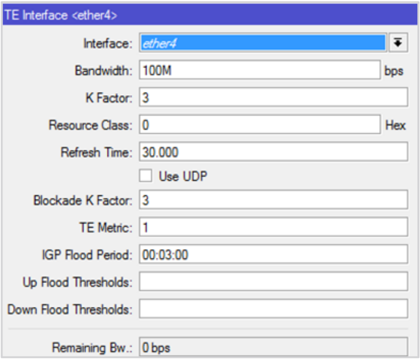
RSVP TE is one of the major reasons large providers use MPLS. To sum up what this system does is, well, difficult. MPLS already provides the automatic forwarding though the shortest path of the network, RSVP TE adds metrics to the Traffic Engineering portion. The example would be if you have a primary link that is 100 Meg and a secondary that is 50 Meg, if you built all of your traffic with TE, each time you put a new TE on that circuit, it would deduct the bandwidth required from the total link. Seeing that the 100 is faster, all the clients would be on that. Say you have nine 10 Meg clients, that gives you only 10 Meg free. This doesn't say that the TE's are using their 10 Meg, but it shows that 90 Meg is committed too. Now when you add a new client at 20 Meg, the TE system understands that there is no bandwidth available for that client on that circuit, and moves them automatically to the 50 Meg, as that’s the next best choice with available bandwidth.

As an alternate, you can build primary and secondary path, your secondary or third path can be a dynamic path creation as well. The example would be to use the above, but instead of using just the primary circuit first, the second client you can run though the second circuit unless it fails then the traffic goes to the secondary, third and so on. This gives you the ability to use multiple links at the same time, regardless of the path costing that you have placed via OSPF or other dynamic routing system.

Remember that RSVP TEs are not uni-directional. Because of this, it is possible to send traffic over one TE out from A to B. Since the path is from point A to B, you don't have to configure a TE from point B to A, but if you want the traffic to take a specific path back, then you would need to create the TE on the other side. In most cases, you would build a TE in both directions.

MPLS Traffic Engineering Interface

The first step to creating a RSVP TE system is to define the speeds of the network. You will need to create a MPLS → Traffic Eng → Interface item for any interface you are going to be doing TE on. This can be as simple as defining the BW on the interface. This is the interface setting, setting its bandwidth for the most part. This says how much bandwidth there is going out this interface.



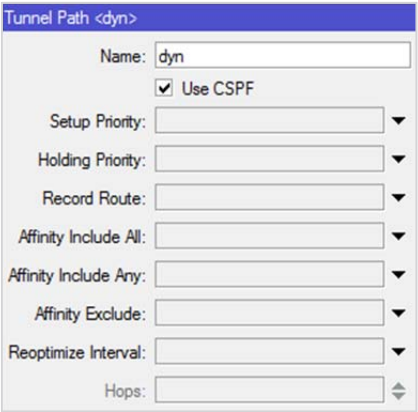
TE Interface <ether4>

Interface:	ether4
Bandwidth:	100M bps
K Factor:	3
Resource Class:	0 Hex
Refresh Time:	30.000
	□ Use UDP
Blockade K Factor:	3
TE Metric:	1
IGP Flood Period:	00:03:00
Up Flood Thresholds:	
Down Flood Thresholds:	
Remaining Bw.:	0 bps

This is a very big item that gets confused. This, even though it is an interface, is not the actual Traffic Engineering interface; it is the traffic engineering interface settings!

RSVP TE Tunnel Paths

The second thing that you need to create is your tunnel path. The first tunnel path you create should be you dynamic path. This is simply a tunnel path with the use CSPF on. CSFP stands for Constrained Shortest Path First. So it will use the shortest path, unless there is a reason not to, an example would be where there is not enough bandwidth to meet the TE requirements.



Tunnel Path <dyn>

Name: dyn

☑ Use CSPF

Setup Priority:

Holding Priority:

Record Route:

Affinity Include All:

Affinity Include Any:

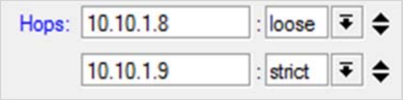
Affinity Exclude:

Reoptimize Interval:

Hops:

For auto-path creation, this dynamic tunnel will use the MPLS Traffic Engineering Interfaces to determine the proper path, typically this will be the shortest unless there is not enough bandwidth on the primary link between site A and B, then the dynamic path will automatically path out to the next best etc. The idea though here, is that when you start reserving bandwidth, you will not end up with links that are full and have high latency, the additional path will be though another method that has available bandwidth that you reserve.

Tunnel Path Hops



Hops: 10.10.1.8 : loose

10.10.1.9 : strict

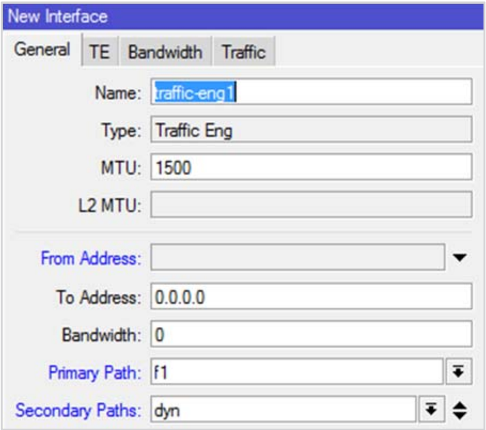
The hop in the TE path is the hops strictly or loosely required for the path to take. Unlike standard routing, where you have to define all routes and gateways from point A to point B, RSVP TE you can just define required addresses to go through, typically you're LSR IPs. In our example, we loosely require that the path goes through 10.10.1.8, meaning that it is acceptable to have other hops between the previous hop (in our case none) and the defined hop. Then we are strict that we then use 10.10.1.9, or we require that there are no other hops between the previous hop, and this hop. If you start with a strict, then chances are you will have to define the full path.

Traffic Engineering Interface

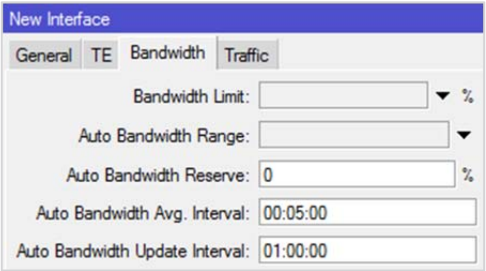
The last step to create your RSVP TE tunnel is to actually create the, well, tunnel! Click on Interface → Traffic Eng to create a new interface for your RSVP TE. Here you can specify the from and to address, as well as how much bandwidth you need. You also will need to define the primary and secondary paths. I would always have the last path as a dynamic path, just in case your others fail.

In the bandwidth tab we have the actual bandwidth limit, this is specified by percentage. If you enter 10 Meg in the bandwidth, then 100% in the limit, then the speed of this TE would be 10 Meg max. Remember, TEs are built into a single direction, so this is traffic from this TE router to its remote end.

On the TE tab, if you are using Dynamic paths, there is an option for optimization interval. This is important cause with TE's the traffic stays on the path unless optimized. In this case you can setup a new optimization interval for it to see if there is a better path available.



Name: traffic-eng1
Type: Traffic Eng
MTU: 1500
L2 MTU:
From Address:
To Address: 0.0.0.0
Bandwidth: 0
Primary Path: f1
Secondary Paths: dyn



Bandwidth Limit: %
Auto Bandwidth Range:
Auto Bandwidth Reserve: 0 %
Auto Bandwidth Avg. Interval: 00:05:00
Auto Bandwidth Update Interval: 01:00:00

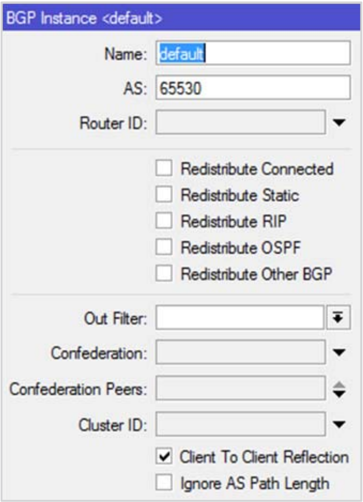
<u>*RSVP TE Auto-Bandwidth*</u>

In the image above, there are a number of options for auto-bandwidth. Even though you can define what the MAX bandwidth is and what the actual limit is as well, you also change this. In another example we use a 100 Meg circuit with nine (9) 10 Meg customers on it. Depending on the type of customers, most will not be using their full 10 Meg, but normally, each customer reserves 10 Meg, therefore only one more customer could fit on that single circuit.

TE Auto-Bandwidth, fixes this by allows you to sample the actual bandwidth used and change the bandwidth reservation on the paths of the TE! This allows for oversubscription of the paths by way of changing the actual reservation of bandwidth. There are a few downsides though, keep in mind that you are not reserving the bandwidth, therefore it can't be guaranteed, due to you adjusting bandwidth, when your client is using more bandwidth, the auto-bandwidth feature has to ramp up to his needs even if that customer paid for 10 Meg, it's possible that they won't get it when they request it. However, the upsides to this are that they simply get the bandwidth that they have been averaging and therefore you can oversell your lines. Circuits are not sitting unused due to the large amount of overhead that the reservations hold.

BGP based VPLS

RouterOS does support BGP VPLS, or BGP based VPLS. BGP based VPLS really should be looked at like a centralized, router, or team of routers running private BGP sessions to each of the VPLS links. The VPLS system uses a single VPLS tunnel between the client and one of the BGP VPLS routers. Typically, you would configure the BGP routers at the core to be a route reflector. What this does, is reflect the routes learned by each peer to the others. So again, VPLS circuits come to a central location and are then decimated via the BGP System.



BGP Instance <default>

Name: default
AS: 65530
Router ID:

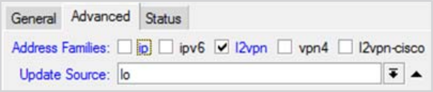
□ Redistribute Connected
□ Redistribute Static
□ Redistribute RIP
□ Redistribute OSPF
□ Redistribute Other BGP

Out Filter:
Confederation:
Confederation Peers:
Cluster ID:

☑ Client To Client Reflection
□ Ignore AS Path Length

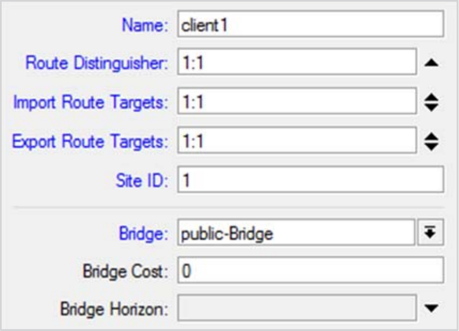
This is just the start of what you can do with BGP based MPLS systems. To start with this, you will create a private BGP instance and AS. You will also configure that BGP core router to offer client-to-client reflection. This instance will provide the client to client data that is needed between the VPLS clients. You can also configure other BGP core routers like this, and configure iBGP between them to offer multiple peer points on your network if you wish.

Next you will start creating BGP peers on these core routers, but instead of forming a IP or IPv6 peer, we will be creating L2VPN peers. Remember that we will need to use our loopback addresses for sourcing. One extra thing you will need to do is to ensure that you "route Reflect" at each client. This is in addition to the client-to-client-reflection on the BGP core. Once you configure both ends, you should get BGP establishment. I.e. the BGP session between the client site and the core BGP routers should come up and show established.



Address Families: □ ip □ ipv6 ☑ l2vpn □ vpn4 □ l2vpn-cisco
Update Source: lo

One the cores, you will need to create the bridge interfaces for your VPLS configuration. You will do this by going into the MPLS → VPLS → BGP VPLS section on your core BGP router. Here we will create BGP VPLS instances; these basically identify what is the client, and allow us to set the route targets, distinguisher as well as the site ID. Also, you will configure the Bridge in which the VPLS systems will be configured. We don't need to make VPLS interfaces with the BGP VPLS system, that’s the point of the BGP; it will create those as necessary. The client systems would be configured the same way, however, the site ID would be different for each site.



Name: client1
Route Distinguisher: 1:1
Import Route Targets: 1:1
Export Route Targets: 1:1
Site ID: 1
Bridge: public-Bridge
Bridge Cost: 0
Bridge Horizon:

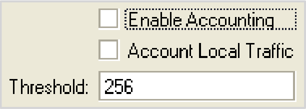
If you did the job correctly, dynamic VPLS tunnels would be created, automatically, for each site. The tunnels are added to the proper bridge group as well. As more sites come on line, more VPLS tunnels should be added to each remote site by reflecting the VPLS tunnel information and allowing the remote sites to establish a VPLS tunnel with each of their remote systems. This is about as close as you can get to auto VPLS creation!

RouterOS Extras

In this section, we will cover extra features that are important to note in RouterOS, however, they typically are not included in many routers on the market. These are extra features of RouterOS; they don't fit into the interfaces sections, but things like Proxy services, NTP, and IP Pools are all covered here.

IP Accounting

IP → Accounting is a method to track both the number of packets and the number of bytes based on IP pairs. When enabled, the IP accounting system starts tracking IP pairs based on the source and destination IP addresses. Data that is dropped in the router are not counted, only data flowing through the router. You can of course, enable or disable local traffic, or traffic sent or received by the router itself.



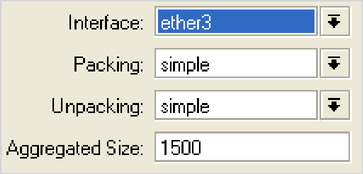
□ Enable Accounting
□ Account Local Traffic
Threshold: 256

Once you enable IP Accounting, you will start to build a list, of these IP pairs along with their corresponding packet and byte counts.   The Threshold is how many IP pairs can be created; the maximum number is 8192. Once the accounting system has reached 8192 IP pairs, anything left over or unmatched will go into an Uncounted counter. You can take a snapshot; this does two things. One, it displays the IP Pairs along with their counters. Two, it clears out the table.

Most people will use this with some form of data-collection application. You would normally enable the web-access system, and when your data collection-application connects to http://routerip/accounting/ip.cgi on the router, a snapshot is taken and the information is presented. You do have the ability both in the Firewall as well as in the IP Accounting Web-Access menu to limit what IPs can run this web application.

M3P – MikroTik Packet Packing Protocol

Try it; say it three times real fast! The MikroTik Packet Packing Protocol optimizes data usages of links that have high overheads. Some types of data have quite a bit of overhead per packet and M3P will optimize throughput on these high-overhead links. For example, VoIP packets that are very small packets, (around 100 bytes) and they could be combined into one larger packet and transmitted faster and quicker. In many cases this will increase the overall usable bandwidth on a link. This feature is very simple to setup, as you simply enable it for the interface with the settings you wish to have on both sides of your link.

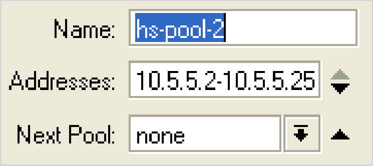


Interface: ether3
Packing: simple
Unpacking: simple
Aggregated Size: 1500

To access this system, click on IP → Packing. Add an interface with the options. The more complex the packing process, the more CPU time you will use. I would suggest starting just with just the simple packing types and see what your CPU does before doing compression etc.

IP Pools

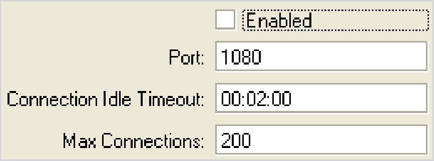
IP → Pools are your IP pools for both your DHCP and for other systems like the universal client in your hotspot. Most of the time, your IP pools are setup by other processes such as when you setup your DHCP-Servers. The pool is exactly what it sounds like. It provides gives us a list of IPs that different services can use. RouterOS will also have a Used Addresses tab to list addresses that are currently in use.



Name: hs-pool-2
Addresses: 10.5.5.2-10.5.5.25
Next Pool: none

Socks

Socks is a proxy server that is designed to relay TCP-based applications across Firewalls. This service is not commonly used anymore; however, it's worth saying that RouterOS does provide one. To configure this, click IP → Socks. In the settings, you will enable it this as well as and set the port, timeout and the Max Connections. Under the Access tab, you will create access rules, to allow access to your Socks server.

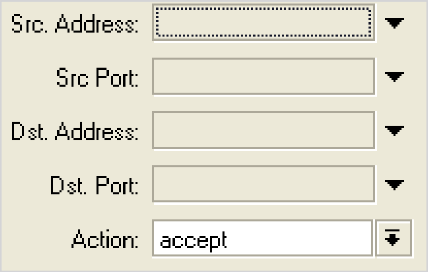


□ Enabled

Port: 1080

Connection Idle Timeout: 00:02:00

Max Connections: 200



Src. Address:

Src Port:

Dst. Address:

Dst. Port:

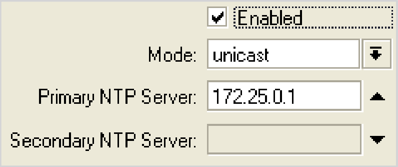
Action: accept

NTP

Since RouterBOARD do not have a clock, RouterOS does support the Network Time Protocol (NTP). This system allows clients to sync their time with a time server. It is very simple to use. RouterOS since v3 has placed the NTP Client in as part of the base system combined package, but the NTP server is a separate package to install. In version 5 of RouterOS, by default, the NTP Client and Server are not installed, but the SNTP client is installed.

Client

To configure the NTP Client, all you have to do is enable it, specify what mode to operate in and the IPs of the NTP servers if applicable. One example could be us.pool.ntp.org. A domain will resolve to an IP address if the DNS is working correctly on the RouterOS. In version 5, you will need to install the NTP package to get the NTP client. The settings between the NTP client and the SNTP client are shared.



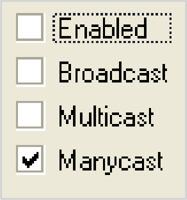
☑ Enabled

Mode: unicast

Primary NTP Server: 172.25.0.1

Secondary NTP Server:

Server



□ Enabled
□ Broadcast
□ Multicast
☑ Manycast

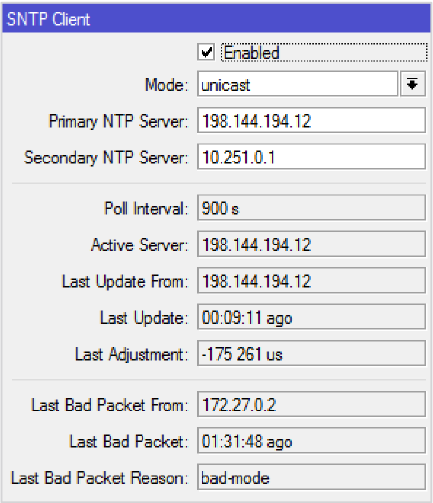
The NTP server on RouterOS needs to be installed as a separate package, however, it is very simple to setup. The NTP Server responds to NTP clients requests. The only option is what kinds of requests it should respond to. It will reply with the date and time from its clock.

Troubleshooting NTP

NTP is a very simple protocol. For the NTP-Client to work, of course it must be able to reach a NTP server. Start by standard troubleshooting the connection; make sure that you can ping the NTP server. You can also try with another device that you know can get to your NTP server. If you are using a public NTP server, or one on the Internet, then you should verify that you have full Internet connectivity. Verify that your NTP port is also open. NTP uses UDP port 123 for communications, so make sure you have that operational and unblocked as well.

SNTP

In version 5 of RouterOS, MikroTik has relocated they have removed the NTP Client and Server to the NTP package, and has have installed by default now the installed the Simple Network Time Protocol SNTP client by default. This is implemented per the RFC standards, and (per the RFC) does not support the manycast mode. When you install the NTP package, the settings from the SNTP Client are shared with the NTP Client package and the SNTP Client is automatically disabled.

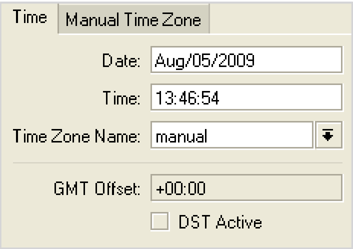


SNTP Client

☑ Enabled

Mode:	unicast
Primary NTP Server:	198.144.194.12
Secondary NTP Server:	10.251.0.1
Poll Interval:	900 s
Active Server:	198.144.194.12
Last Update From:	198.144.194.12
Last Update:	00:09:11 ago
Last Adjustment:	-175 261 us
Last Bad Packet From:	172.27.0.2
Last Bad Packet:	01:31:48 ago
Last Bad Packet Reason:	bad-mode

Clock



Date: Aug/05/2009
Time: 13:46:54
Time Zone Name: manual
GMT Offset: +00:00
□ DST Active

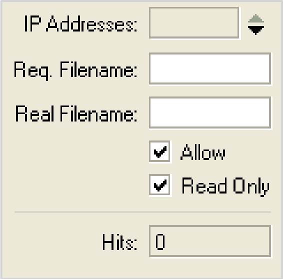
The clock is the time and date system for your RouterOS system. RouterBOARD systems do NOT remember the date/time after a reboot or power cycle. X86 systems typically have a battery that will remember the time. To set your clock manually, as well as setup your time zone, click System → Clock. Here you can setup your Time Zone, as well as a manual zone if you wish.

System Identity

The System Identity is a means to identify the RouterOS system. It does not do anything but label the RouterOS system. To set the Identity, click System → Identity and type in the new Identity. The identity does show up on WinBox Discovery as well as the IP Neighbors Discovery systems.

TFTP Server

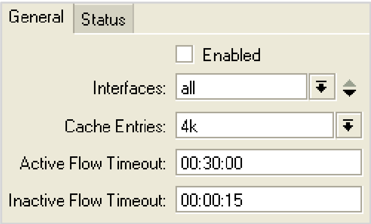
In Version 3.21, a TFTP server was introduced into RouterOS. To access the TFTP Server, simply click IP → TFTP. Here you can click the plus sign and add what IPs addresses that are allowed to read from the server, as well as and set what file names that you wish to have. It also has options allowing if the file to be written to or not.



IP Addresses:
Req. Filename:
Real Filename:
☑ Allow
☑ Read Only
Hits: 0

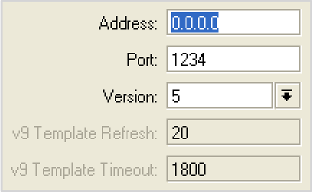
Traffic-Flow

Traffic-Flow is a system that can provide stats based on packets that pass through your router. What is more important is that this data can be collected by using some kind of NetFlow traffic capture software or device. The amount of data that is generated is very low, but it streams that data to your capture and analysis software. Most of these types of software's can help you identify performance issues with your network, what kind of data is moving, when it moves, help you to identify traffic patterns, as well as look at individual IPs and subnet ranges and generate usage reports based on those. Describing these applications in more detail is outside the scope of this book; however, RouterOS has the ability to stream that data to these applications. To access the net-flow system, click on IP → Traffic-Flow.



□ Enabled
Interfaces: all
Cache Entries: 4k
Active Flow Timeout: 00:30:00
Inactive Flow Timeout: 00:00:15

The first thing to do is to enable and configure the basic information about your Traffic-Flow system. Under the traffic flow settings button, you can enable and disable the system, as well as specify what interface, how much data to cache, and specify timeout values. Next, you will need to create a traffic flow target. This is where your RouterOS system will send that data. You can specify several targets if you wish by IP address, as well as their port and what type of NetFlow data you would like to send to them.

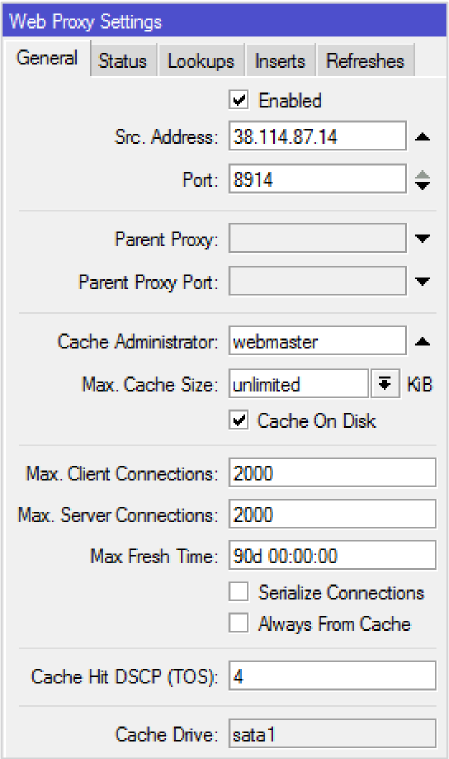


Address: 0.0.0.0
Port: 1234
Version: 5
v9 Template Refresh: 20
v9 Template Timeout: 1800

Once this is done, you should see data moving under your settings status tab, and that's all that you will need to configure under RouterOS. Your NetFlow software will need to be configured correctly to accept that data stream as well as how to perform the analysis of that data.

Web Proxy

The Web Proxy system provides a content-caching system for web traffic. Web caching your data can result in increased web surfing performance, as well as significant bandwidth savings. I have seen a web caching system running on a PowerRouter 732 that had a hit rate over 45.9%. That means for every 1 GB of data passed through the web proxy system, 1.459 GB of data was passed through to the customer. So you saved almost ½ of a GB of data that would have otherwise been moved through your Internet connection.



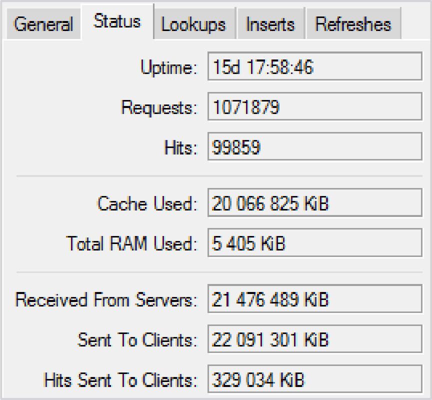
☑ Enabled
Src. Address: 38.114.87.14
Port: 8914
Parent Proxy:
Parent Proxy Port:
Cache Administrator: webmaster
Max. Cache Size: unlimited KiB
☑ Cache On Disk
Max. Client Connections: 2000
Max. Server Connections: 2000
Max Fresh Time: 90d 00:00:00
☐ Serialize Connections
☐ Always From Cache
Cache Hit DSCP (TOS): 4
Cache Drive: sata1

The web caching system is configured in RouterOS by going to IP → Web Proxy. Under the Access tab, use you will have the Web Proxy settings button to configure your web proxy options. You will need to enable the web proxy system, and specify a port. Also keep in mind that there are hackers and other people out on the Internet looking for open proxies, so you will need to secure this. There is really no standard proxy port, even though 8080 is commonly used. I typically will use some random port number.

The Cache Administrator will be displayed if there is a cache issue or non-existent pages. You also can set the max caching size, and if you are using a disk to cache with, make sure you check the box for caching on disk. If you do not check the Cache On to Disk box, RouterOS will use your RAM for your caching system. The drive that you will use to cache with is defined in the Store system; see that section for more information on the Store system.

The Server and Client connection configuration is are important if you have a large system with many connections. Simply stated, this is how many connections the web proxy system will use. The Max Fresh Time is very important if you are interested in caching as much data as possible. When pages get delivered through the web proxy system, they may or may not contain fresh time data. This time is how long the web browser and caching systems should hold the page without requesting a new copy. If the currently-stored page has not timed out, it will be used, however, many pages will not have a timeout value in the HTML code. When this occurs the Max Fresh Time is the default timeout value used.

The cache hit DSCP is also a great tool. It allows you to be able to identify data coming from the caching system that was not retrieved from the Internet. This allows you a number of options, but the one I like to do is to specify an unlimited queue to deliver data from the caching system as fast as possible outside of the customer's normal queues.

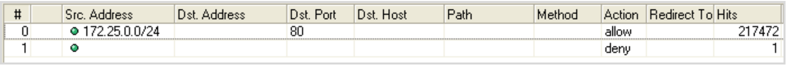


Uptime: 15d 17:58:46
Requests: 1071879
Hits: 99859
Cache Used: 20 066 825 KiB
Total RAM Used: 5 405 KiB
Received From Servers: 21 476 489 KiB
Sent To Clients: 22 091 301 KiB
Hits Sent To Clients: 329 034 KiB

The Status tab of your web proxy settings will shows you all of the stats that you will need to evaluate how your web proxy system is working. The most important statistical piece of information is how much data you have saved by using the web caching system. This number would be calculated by diving the number of hits sent to clients into the sent to clients value. In our image, it's a very low number, around .03%. However, as more and more users use this system, the ratio will increase. Also show here is the amount of data in the web caching system. In the image, 20.066 GB of data is stored.

Web Proxy Access List

The access list is a list of IPs, and or ports and protocols that can use the web caching system. This is used to secure your web proxy system. I recommend a setup like the following image:

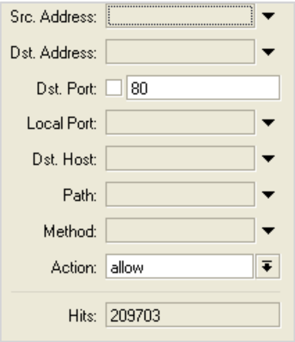


#	Src. Address	Dst. Address	Dst. Port	Dst. Host	Path	Method	Action	Redirect To	Hits
0	● 172.25.0.0/24		80				allow		217472
1	●						deny		1

In this image, we have the private address of the local subnet on port 80 only as allowed, everything else is denied. This way a hacker cannot use my web proxy system to relay though. This is very important to do; otherwise, any bandwidth saving that you may get through the web proxy system could be used up by someone outside of your network stealing your bandwidth.

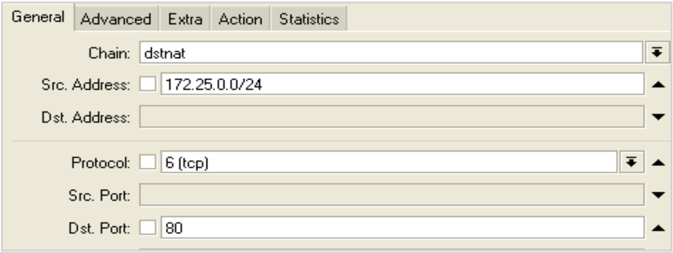
Cache and Direct Web Proxy Tabs

The Cache tab says what can be cached. In most cases, I simply place a rule to cache all port 80 traffic. But you can specify IPs, and paths that you may not wish to cache. The Direct tab allows you to specify something that the web proxy will allow the client to make a direct connection on. No caching will be done. You can also specify items that should go through another proxy server if you wished.

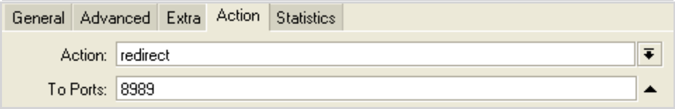

Src. Address:
Dst. Address:
Dst. Port: □ 80
Local Port:
Dst. Host:
Path:
Method:
Action: allow
Hits: 209703

Transparent Web Caching

By default, you can simply specify in your web browser to use a proxy server. However, using other systems in RouterOS, such as the NAT system, we can transparently send customers data to a proxy server without them even knowing about it. To setup a transparent proxy rule, you will need to go to IP → Firewall → NAT. Inside this we will Create a rule that will take data from our customers, using TCP/80 as a destination, and NAT them to our proxy system.



Chain: dstnat
Src. Address: □ 172.25.0.0/24
Dst. Address:
Protocol: □ 6 (tcp)
Src. Port:
Dst. Port: □ 80

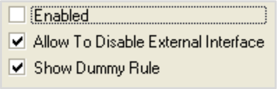


Action: redirect
To Ports: 8989

In this rule, you can see that we are taking IPs from our private network, and then redirecting them to the proxy port. This rule effectively reroutes their HTTP traffic through the web proxy system regardless of what their settings are in their browser.

Universal Plug and Play - UPnP

Universal Plug and Play applications communicate with your RouterOS system to open and forward ports through NAT that are necessary for the application or UPnP system to function correctly. For the ISP or WISP, I would not configure UPnP on any core devices, normally, only customer CPEs would be normal for us to use UPnP on. To configure UPnP, simply click IP → UPnP. The UPnP Settings button will allow us to enable or disable the feature and allows us to use some security features of UPnP. Specifically, we have the ability to disable the ability for an UPnP Device or software to disable the external interface. I don't know why you would need to do this, but it is an option. UPnP is very insecure, as no passwords or authentication are used. UPnP is used to simply make holes in your NAT system easy for software and devices.

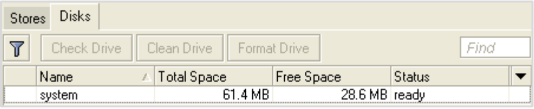


□ Enabled
☑ Allow To Disable External Interface
☑ Show Dummy Rule

Once enabled, you will need to add two interfaces at least, one external and one internal. This way UPnP knows what is inside and outside of your network. After you have completed that, there is no more configuration needed for UPnP. If the device is working correctly, you should see dynamic NAT rules created for specific ports to forward to internal addressing.

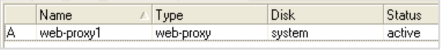
Store System

RouterOS introduced in Version 3.15 a new file store system. This system is designed to help you properly manage the locations of stored data along with the ability to help you manage storage mediums such as hard disks and flash drives. This new system is called Stores. This system can be accessed from the System → Stores (version 3.25 and higher versions.) On 3.15 to 3.24 it is accessed by Stores on the main menu. Inside this menu you have two options; one is for your disks. This will show your disks, if they are ready and in-use, or if they are unknown and need to be formatted. You can also do the formatting of your disks here. One thing to note is that the formatting option on large drives can take some time and CPU. I recommend doing the formatting in a non-production RouterOS system and then attach your router to the production system. Most small flash drives though do not take long. The Hard disks larger than upwards of 60+ Gigabytes of storage take a considerable amount of time.



Name	Total Space	Free Space	Status
system	61.4 MB	28.6 MB	ready

You also have a tab called Stores. The Stores menu allows you to specify what goes where. The example is that if you have user manager enabled and web proxy, you can specify that the user manager data is to remain on your primary system disk, and the web proxy data is to be on your secondary disk. You also have the different statuses. An example is that you can have the user manager data on your system disk, but another copy, (a backup copy) on the USB flash drive that you plugged in. The main copy is on-line on the system disk and if something occurred, you would have a backup copy on the USB disk.



	Name	Type	Disk	Status
A	web-proxy1	web-proxy	system	active

CALEA

The CALEA, or (Communication Assistance for Law Enforcement Act) requires routers in the US to be able to intercept and log network traffic based on a CALEA order. I will not go into the legal issues with this, as always on these types of matters, consult with a lawyer. What I will cover here, is how to install, configure and use the CALEA system as it appears in the v3.28 of RouterOS.

There are two sections to the CALEA package, even though it is one actual package. One is called the Client, or what I like to think of as the interceptor. This is the device that intercepts and relays the data; it is located at the point where you wish to capture data flowing to and from the end “target” or customer. The second portion is the Server system; it receives data from the inceptor, and stores it. As of v5rc9, the entire CALEA package only has command line interfaces.

Before you get into this section, please note that I am not a lawyer. Please consult with legal counsel. Any information stated here is for informational purposes only.

CALEA Interceptor

This portion of the CALEA system is used to intercept the data. This could be at an access point, core router, or elsewhere in your network. My best practices usually leave this at the AP that the client connects to or the core router to the Internet, assuming there is no other possible route. You typically will want this to be where it can capture ALL of the data possible from your target. This can be at a small router, or AP, as the package is very small and does not take much CPU. However, keep in mind that as an interceptor it sends the customer data from itself to the CALEA server. This creates more traffic on your network. If you intercept at your AP, and send data to a server RouterOS system near your network edge, every bit of data both transmitted and received by the interceptor will be sent across your network. If a customer that can move 1024k/2048k up/down, then that means you can have up to 3096k being sent from that interceptor to your server!

Intercepting Data

Once you have determined where the interceptor is going to go, you need to determine if that device is a bridge or a router. There are two different places to intercept packet flow in RouterOS depending on if that unit is a bridge or a router. For routers, you will use***: /ip Firewall calea.*** For bridges, you will need to go to: ***/interface bridge calea***.

In the IP Firewall system, you will specify a sniff-target, and a sniff-target-port. These two items are the target for your data to be sent too, or the CALEA server. You will also need to specify what data you are collecting; typically this would be either a SRC-MAC or SRC-IP address. The following is a valid add line if you are sending to the router 1.1.1.1, on port 300, and capturing data from the IP of 2.2.2.2 :***/ip Firewall calea add src-address=2.2.2.2 sniff-target=1.1.1.1 sniff-target-port=300 chain=forward action=sniff.*** Note here, that this rule only catches data to the 2.2.2.2 ip address. To get data going the other way, you will need to configure another rule to match data based on the dst-address of 2.2.2.2. If you are bridging, The bridge side of things you could use the same filter, but you would need to specify the MAC Protocol as being IP, and/ or you could match data in some other way, such as src-mac.

There are two actions, sniff and sniff-pc. The sniff-pc is used for packet-cable protocol only. Typically you would use the sniff action only.

Once the interceptor is setup, you are basically streaming a packet sniffer with specific data to and from your IP address and/or MAC. Once this is done, you will need to configure the CALEA server side

CALEA Server

This is the portion that actually stores the sniffed data from your interceptor. Note that depending on the speed at which data can be sent and received, you may need to have a quite a large disk or disk array.

> ***As a side note here, I typically use a Virtualized RouterOS system as the CALEA server, typically with an attached SAN, or other system with multiple TBs of storage. I have also seen customers place 3-5 2TB hard drives in a windows system, created a virtual Raid 5 array, and then used that disk as CALEA server storage.***

To configure your CALEA server, you will need to go into the command line at a different point in RouterOS. This system will be in the ***/tool calea*** section. We will create a server in response to the interceptor that we created in the interceptor section.

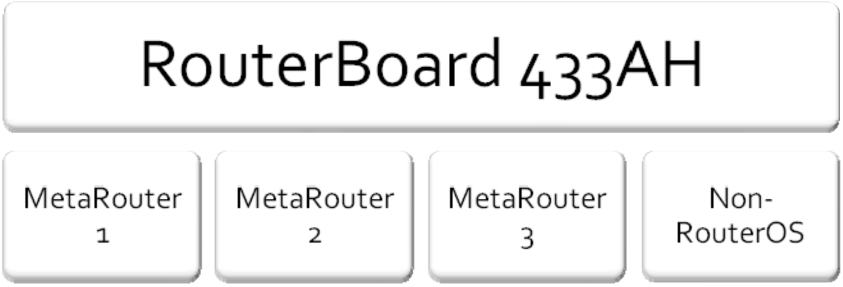
```
/tool calea add action=pcap intercept-port=300 intercept-ip=3.3.3.3
```

Note here that the intercept-ip and port should be the IP address that the interceptor is coming from (note this is the forward-most facing interface of that router towards the CALEA Server), and the port should be the port configured on the interceptor. The action of PCAP specifies that we should store that data in PCAP format. We also have several other options here that we need to note, ***pcap-file-stop-size*** is how large to let each file to get before starting a new file. The other is very important; this is the ***pcap-file-hash-method***. This specifies the hash algorithm that the data file should have. These files, the hash files, should be maintained by you in an effort to ensure that the data being presented is the data that is actually collected.

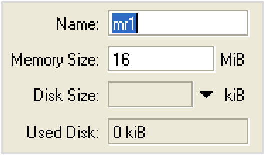
MetaRouters

MetaRouters are virtualized routers that operate inside of your RouterOS system. This can be useful to allow a customer or individual access to their own private router, with their own IPs and Firewall settings, without not actually having to purchase the hardware to do so. At the time of writing this, MetaRouters work only on RouterBOARD 400 series boards. You are limited as well in the number of MetaRouters that can run on one system. This is mostly due to CPU and RAM restrictions. These MetaRouters run underneath the main RouterOS system, and use the license that the main RouterOS uses.

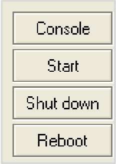
In the diagram below, you will see that you can have multiple MetaRouters below an individual RouterBOARD 400 series product.



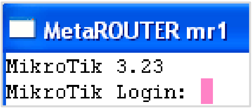
To create a MetaRouter in your system, you will click on the MetaRouters tab on the left side of your WinBox Application. Here you will get two tabs, one is for the actual MetaRouters, and the second is for the interfaces to these routers. You can associate both virtual interfaces from your hardware-based RouterOS system to individual MetaRouters. You can also associate physical interfaces to interfaces inside your MetaRouter as well. When you create your MetaRouter, it's as simple as clicking the Plus sign, and assigning a name to the router. Once done, the system will start the MetaRouter virtualized under your RouterOS hardware.



Name: mr1
Memory Size: 16 MiB
Disk Size: kiB
Used Disk: 0 kiB



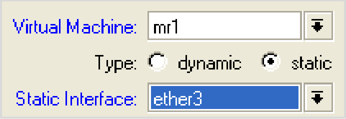
Notice that inside the MetaRouter options, you can also reboot, shutdown and start these, as well gain



```
MikroTik 3.23
MikroTik Login:
```

console access to the MetaRouter.

Once you have created your MetaRouter, you can then start assigning interfaces to it under the interfaces menu. Again, simply click the plus sign and assign your physical interface to the virtual machine. This is a static interface assignment. The other type is a dynamic; upon starting the MetaRouter, the interface attaches dynamically to a bridge group on the physical router.



Virtual Machine: mr1

Type: ○ dynamic ● static

Static Interface: ether3

I have used MetaRouters primarily for testing mostly. I have never really had a need yet to create such a small virtual router. Keep in mind that you only have so much processing power in one of these small 400-series boards. Make sure you do not need to move a lot of data though these types of systems.

Non-RouterOS MetaRouters

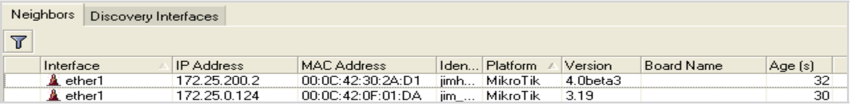
RouterOS started to offer the ability for MetaRouters on the 400 Series RouterBOARDs to run Non-RouterOS software.

RouterOS Tools

RouterOS has many tools, and these tools give you the ability to monitor data, log it, view it and perform many other critical tests to help you optimize, troubleshooting and properly maintain your network.

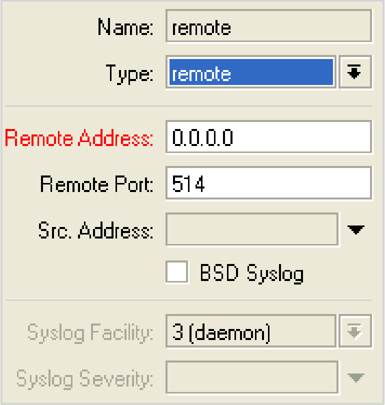
Neighbors

RouterOS uses discovery packets sent out all interfaces to discover neighboring RouterOS and Cisco IOS systems. To access this, click IP → Neighbors. This discovery process is done via MNDP or MikroTik Network Discovery Protocol. It will learn information about the neighboring devices as well, such as IP address on the neighboring interface, MAC, Identity and versions. Since RouterOS offers MAC-Telnet ability, you can simply double click on a discovered device and MAC Telnet to a neighbor. You can also turn on or off the discovery protocol by using the Discovery Interfaces tab. MNDP uses UDP Protocol 5678 and broadcasts every 60 seconds. After 180 seconds, it only discards routers that have been removed after 180 seconds.



Interface	IP Address	MAC Address	Iden...	Platform	Version	Board Name	Age (s)
ether1	172.25.200.2	00:0C:42:30:2A:D1	jimh...	MikroTik	4.0beta3		32
ether1	172.25.0.124	00:0C:42:0F:01:DA	jim_...	MikroTik	3.19		30

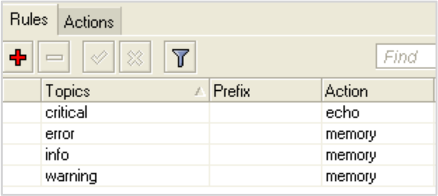
Logging



Name:	remote
Type:	remote
Remote Address:	0.0.0.0
Remote Port:	514
Src. Address:	
□	BSD Syslog
Syslog Facility:	3 (daemon)
Syslog Severity:	

The RouterOS logging systems is quite extensive. You have options to create log files, send logs to Syslog servers, or echo data to the local log or console. Under System → Logging you can setup the different types of actions, giving you the ability to send your log data elsewhere. The remote action sends data to a remote Syslog server. Under the Action tab you can create several Syslog servers; however you will need to specify the remote address as well as the remote port for this to work.

Once you have setup your logging actions, you will then need to setup logging rules. The rules specify what type of topic the log is about and what action to perform. When you wish to perform debugging to help figure out why some application is not working correctly, you can enable more logging through the Topics. There are many Topics, and you should refer to the Command Reference Manual for more information on each of these Topics.



Topics	Prefix	Action
critical		echo
error		memory
info		memory
warning		memory

Logging Actions

There are a number of logging actions that RouterOS offers. In this section I will simply cover what each one does.

The first is Echo – This simply sends what would be displayed to the console screen. The second (and the most common logging action) is the Memory action. This simply displays what should be logged into the /log submenu showing up on the log. By default, there is a limit to the size of the memory; this is by the number of lines. You can setup this limit here as well. The third is Disk. This creates a log file that is a specified number of lines, as well as how many files you can have. There is also an option for stopping of the log files when full. This option simply stops the log file from writing once you have reached the number of lines and the number of files specified. The e-mail option is the last one covered in this section. This option allows you to send an e-mail with the information contained in the log every time the rule applies. Be careful with this one as you can get a large number of e-mails very quickly.

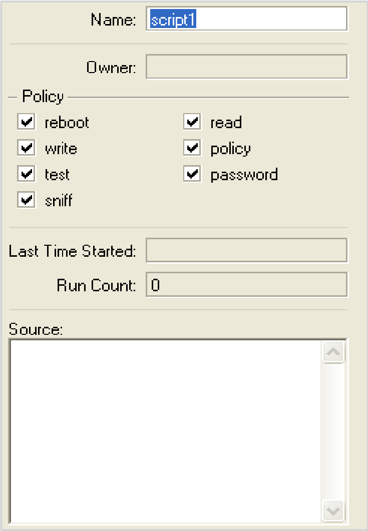
Remote allows you to send logging out to a syslog server, and is covered in the previous section.

System Configuration Reset

To reset the RouterOS configuration you must use a command line option; the command is */system reset.* When you issue this command it will ask you to confirm. Once confirmed, the system will reboot and come back up in a blank state, just like if the OS was just installed. All password and configuration is wiped out. This reset does not remove any licensing information.

Scripting

RouterOS offers a full scripting system to automate and perform complex tasks. Under System → Scripting, you can create scripts, run them, and modify the source code as needed. Programming and scripting commands are outside of the context of this book. Once you have created your scripts though, you can then schedule them through System → Schedule.



Name: script1

Owner:

Policy

☑ reboot ☑ read
☑ write ☑ policy
☑ test ☑ password
☑ sniff

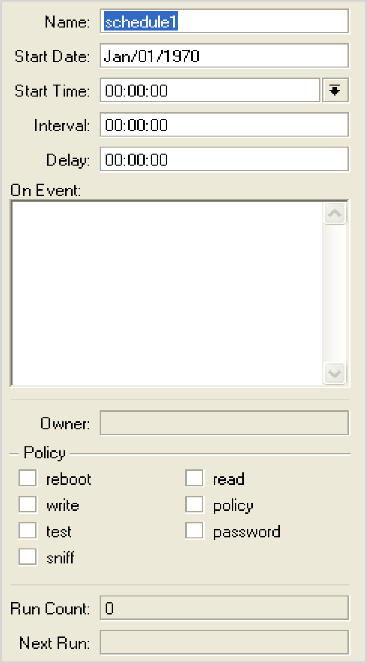
Last Time Started:

Run Count: 0

Source:

Scheduler

The Scheduler allows you to schedule start dates, times and rerun intervals as well as delays upon starting. You can setup a start time of startup as well, with a delay. I like to do this when it is necessary to run a script upon startup, but I need a minute or two for all of the services and connections to come up before I start running the script. It will also list the number of times to run. Remember though, that your clock will need to be set if you want the Scheduler to run at the correct time.



Name: schedule1

Start Date: Jan/01/1970

Start Time: 00:00:00

Interval: 00:00:00

Delay: 00:00:00

On Event:

Owner:

Policy

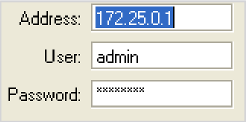
☐ reboot ☐ read
☐ write ☐ policy
☐ test ☐ password
☐ sniff

Run Count: 0

Next Run:

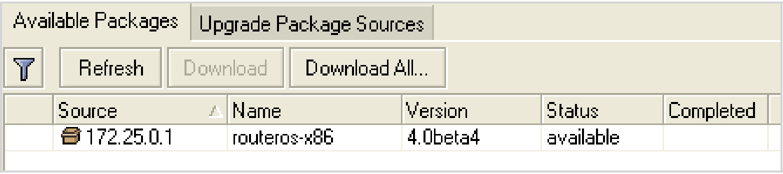
Auto Upgrades

The Auto Upgrades section allows you to perform RouterOS software upgrades quickly. To get to this menu section, click System → Auto Upgrade. Here, you will need to first define a package source. This system will FTP into the package source and obtain a file list and based on that file list show packages that may be available to upgrade the RouterOS version that you are on. It will take into account variables like the current RouterOS version, as well as the processor type.



Address: 172.25.0.1
User: admin
Password: *********

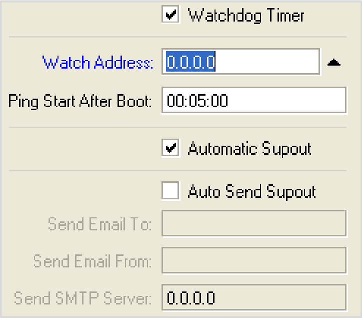
Once you have defined your package sources under your Package Source tab, you can then click on your available package list, and select refresh. This will force the system to go out to the FTP servers and download these lists. Remember that you must have the correct username/password in the package source as well as the ability to FTP into that source for this to work. Once the list is downloaded, then you should have a listing for your Processor type.



Source	Name	Version	Status	Completed
172.25.0.1	routeros-x86	4.0beta4	available	

As you can see here, we have a package that is available. Clicking on this and then clicking download will simply download the package from the FTP server, and nothing more. The Download All button will give you options to download beta packages as well as rebooting after the download is complete.

Watchdog



☑ Watchdog Timer

Watch Address: 0.0.0.0

Ping Start After Boot: 00:05:00

☑ Automatic Supout

□ Auto Send Supout

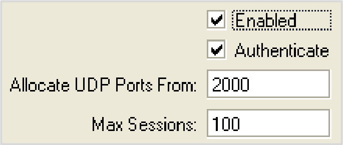
Send Email To:

Send Email From:

Send SMTP Server: 0.0.0.0

The watchdog system runs in two different modes, both a hardware and a software mode. RouterBOARDs have a hardware based watchdog that this feature uses, but if you are on an x86 system, the watchdog runs as an software-based service. By enabling the watchdog timer, RouterOS pings the watched address. Once the ping has failed several times, the system will reboot. Prior to rebooting, you can have the system create a supout, as well as e-mail the supout if you have defined the e-mail system. The E-Mail system is covered a bit later in this chapter. After the system reboots, the watchdog waits the amount of time in the Ping Start After Boot time. Once that times has passed it will attempt to ping again, this is to prevent the watchdog from constantly rebooting and gives us time to login to the radio and disable the watchdog if necessary.

Bandwidth Test Server



☑ Enabled

☑ Authenticate

Allocate UDP Ports From: 2000

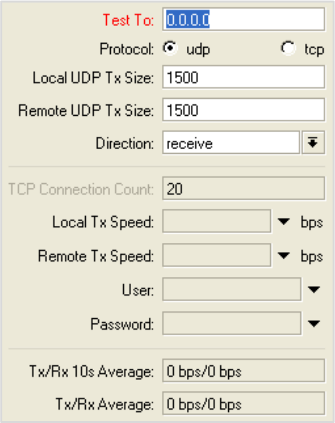
Max Sessions: 100

Usually testing bandwidth is a complicated process by using some form of public bandwidth test site or using a Linux application like IPERF. RouterOS though, offers the ability to run a bandwidth test server for your clients to connect to and perform bandwidth tests. These tests will be covered more in the Test Client section. To configure the options for the bandwidth test server, you will click Tools → BTest Server. Here we have an option for our BTest Server settings. We can enable or disable the server, specify the maximum number of bandwidth test sessions as well as specify if we require authentication. Authentication is user authentication through the RouterOS users system. For example, the admin user that comes defaulted on the RouterOS system would be a user that would be able to perform a bandwidth test.

Note that the performance of the bandwidth test server is not only subject to the network connection that you have but also the power of the CPU on the board. For instance, a RouterBOARD 100 series board cannot generate enough traffic to test even a 100Mbps Ethernet connection, whereas the 600 or 1000 RouterBOARD would be able to do so. The PowerRouter 732 can generate more than 5 gigabit of bandwidth. If you have a wireless link that you wish to test, generating the traffic on the boards that are managing the wireless connection is not the best method; you should use some other type of fast RouterOS device to do these tests with. Also note that you can use the bandwidth test tool for windows that MikroTik provides free of charge on their website as both a bandwidth test server and client.

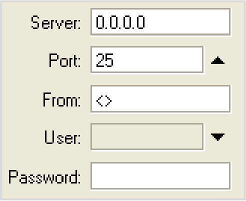
Bandwidth Test Client

The bandwidth test client is the client side of the bandwidth test application built into RouterOS. From here you can start bandwidth tests to bandwidth test servers. You will need at least the IP address of the bandwidth test server, and if you need to authenticate to the server, you will need to place your username and password in as well. You have options to specify packet size in the UDP mode, as well as the direction, send, receive or both that you wish the bandwidth test to run. You also have options to limit the speeds and change to TCP-based connections. In v3.25 and higher versions of RouterOS you also have the option to create several TCP connection streams instead of just a single one, thereby giving you the ability to test over 600 Mbps.



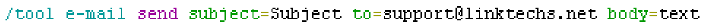
Test To: 0.0.0.0
Protocol: ● udp ○ tcp
Local UDP Tx Size: 1500
Remote UDP Tx Size: 1500
Direction: receive
TCP Connection Count: 20
Local Tx Speed: bps
Remote Tx Speed: bps
User:
Password:
Tx/Rx 10s Average: 0 bps/0 bps
Tx/Rx Average: 0 bps/0 bps

E-Mail System


Server: 0.0.0.0
Port: 25
From: <>
User:
Password:

MikroTik included in a function that allows you to be able to send e-mails based on events inside RouterOS. The E-Mail tool configures the default options for these types of services. Scripting does have the ability to use the command line tools to send e-mails for alerts and notifications from the RouterOS system. When you click Tools → E-Mail, you will get your e-mail settings. In version 3.21, MikroTik put in SMTP authentication. The server would be your outbound SMTP server, and its corresponding port numbers. Then you can specify the e-mail address as well as the username/password to send authenticated e-mails to your mail server.

To send an e-mail out, you must use the command line.

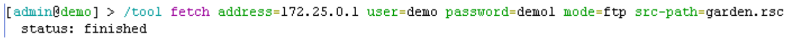


```
/tool e-mail send subject=Subject to=support@linktechs.net body=text
```

Using the tool e-mail command you can specify the body, to and subject lines. As long as your e-mail system is configured correctly with your outbound SMTP Server, the email should go out without issues. These commands inside scripting will allow you to send e-mails when someone logs in to your hotspot as a trial user, or other task.

Using Fetch Commands

Fetch is a command line tool that allows you to fetch or get files from both FTP and HTTP Servers. As long as you can connect to the server in question, you can pull a file into the system drive of your RouterOS system. This is useful to obtain a script or new RouterOS version from a centralized server system.

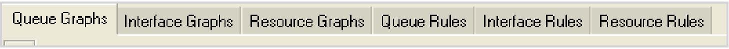


```
[admin@demo] > /tool fetch address=172.25.0.1 user=demo password=demo1 mode=ftp src-path=garden.rsc
  status: finished
```

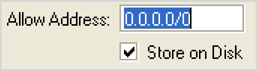
As you can see, you can get files form FTP or HTTP, just simply enter the address, if there is the username/password and the well as what mode you wish to try to download the file in. The src-path is the folder and file that you wish to download. If you specify a dst-path, you can put the file on your removable storage card or in another folder if you wish. .

Graphing

The graphing system of RouterOS allows you to quickly and effectively use RouterOS to show usage over time. RouterOS supports graphing your interfaces, simple queues, as well as the resources of your RouterOS system. To enable graphing, you will click Tools → Graphing. Here you will see several tabs, the graph tabs are the actual graphs to be accessed inside RouterOS, and the rules are the ability to specify who and how you can access those graphs.

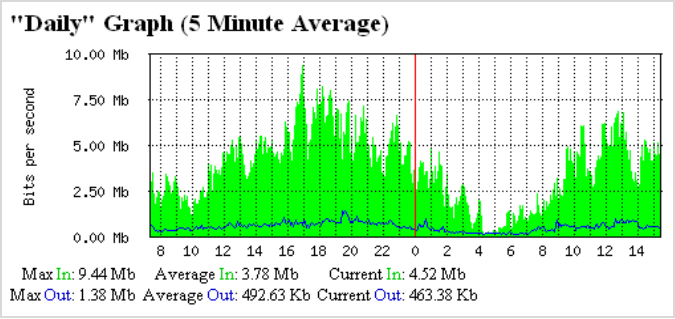


Inside your rules, you will have the option to turn on graphing for each of the different types of rules. You also can turn on the allowed addresses to view this graph. With x86 RouterOS systems, you can store data to a disk and the graphs will be saved even after a reboot. RouterBOARD products do not store this information regardless if you have the store-on-disk selected. You will get four different graphs, 1 hour, 1 week, 1 month and 1 year graphing.



Allow Address: 0.0.0.0/0

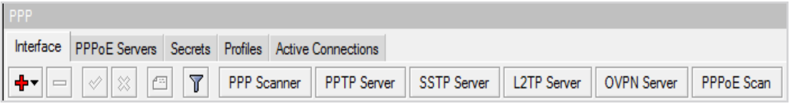
☑ Store on Disk

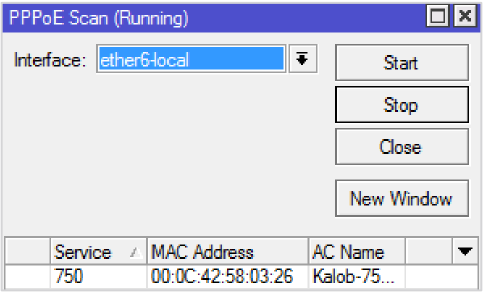


You can also go to http://routerip/graphs and see a web-based version of the graphs as well. These work quite well and record quite a bit of information for you to review and see on each queue and interface. Remember, if you change your www service port under IP Services, you will need to use that port number when trying to view your graphs.

PPPoE Scan

This feature was introduced in v3.21. You can access this feature via the command line using */interface pppoe-client scan*. In v5 of RouterOS, you can also access this via the WinBox system. Click on the PPP → PPPoE Scan.





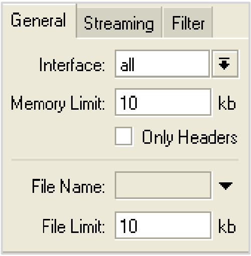
Interface: ether6-local

Service	MAC Address	AC Name
750	00:0C:42:58:03:26	Kalob-75...

As you can see now we have a PPPoE Scan option in WinBox. Inside the PPPoE Scan function, we can scan for existing PPPoE Servers on the network. In our case, we have a single PPPoE server. This shows the MAC address and the AC Name, and service.

Packet Sniffer

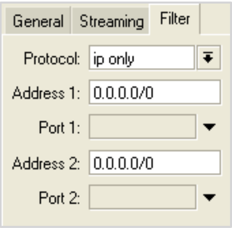
The packet sniffer is located under Tools → Packet Sniffer in the WinBox menus. Once here, you can setup your packet sniffer settings to get started. The interface is required, as well as specifying the memory limit. If you check only headers, you will get quite a bit less data than if you check the entire packet. You can also save that data into a data file if you wish by specifying a File Name and File Limit.



Interface: all
Memory Limit: 10 kb
□ Only Headers
File Name:
File Limit: 10 kb

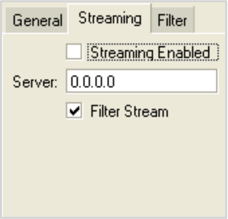
Once you have selected your general information, you may wish to filter your data so that you only look at frames, or only IP information using the Filters menu. You can also filter based on ports and IP addresses or subnets. This can cut down on the data recorded so that you don't have to sort through so much of it later.

Streaming Packet Sniffer Data



Protocol: ip only
Address 1: 0.0.0.0/0
Port 1:
Address 2: 0.0.0.0/0
Port 2:

Streaming is another option in the packet sniffer settings. This is very useful because instead of capturing the data to files or memory of your router. The Streaming tab allows you to send a stream of the captured data to a locally-connected workstation. I use this to capture data to another packet analysis program like Etherreal or Wireshark. Other programs could be used though as well.

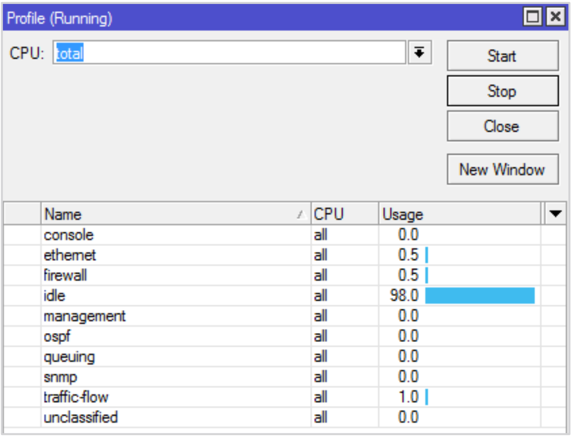


□ Streaming Enabled
Server: 0.0.0.0
☑ Filter Stream

Simply enable streaming in the Streaming tab of the packet sniffer settings and setup the IP address for that stream to be sent to. By selecting the filter stream, it will filter out packets that the router creates itself to send to you. I always use Filter Stream.

Profiler

In version 5 release candidate 1, RouterOS introduced a new tool called Profiler. This basically gives you the ability to monitor CPU usage per task, just like Windows Task Manager does. In Profiler you have options for what CPU you wish to monitor (for multi-core systems that is), as well as what Service is using the CPU. Typically, you should see something like the image to the left. Of course, you can see if your wireless system is taking up a lot of CPU, or your queuing, your Firewall, or other task or services. You can use this to see what tasks are taking up the most of your CPU time.

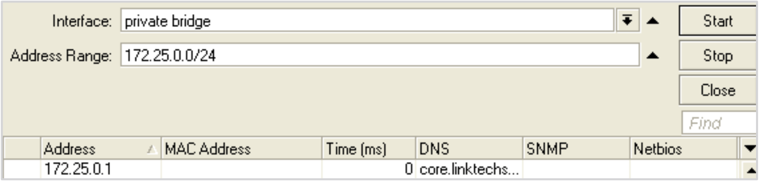


CPU: total

Name	CPU	Usage
console	all	0.0
ethernet	all	0.5
firewall	all	0.5
idle	all	98.0
management	all	0.0
ospf	all	0.0
queuing	all	0.0
snmp	all	0.0
traffic-flow	all	1.0
unclassified	all	0.0

IP Scan

The IP Scan tool scans an IP subnet and returns devices that can be pinged as well as any information that it can obtain from that device. To run IP Scan, click on Tools → IP Scan. Then select what interface you wish to run the scan on, and the address range you wish to scan. When you run it, IP Scan will show the IPs that respond, the MAC addresses, response time, DNS name if any, as well as SNMP and NetBIOS data.



Interface: private bridge

Address Range: 172.25.0.0/24

Address	MAC Address	Time (ms)	DNS	SNMP	Netbios
172.25.0.1		0	core.linktechs...		

Dynamic Routing

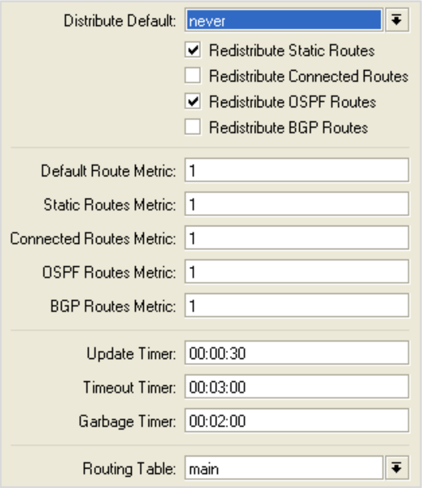
In v3+ a RouterOS license level of 4 is required for dynamic routing. This is on all platforms including x86.

There are many books on OSPF, BGP and RIP.I will run through the basic setup; however we will not discuss routing techniques, and most troubleshooting inside this book as those topics are outside our scope here.

If Installed vs. Always

Most of the protocols listed below use both an “if installed” option and an “always” option. Always distributes the route, well always, regardless of how it was learned. It could be a static route, or it could be a learned dynamic route. If installed, means only if the route was learned do we distribute it.

RIP



Distribute Default: never

☑ Redistribute Static Routes
□ Redistribute Connected Routes
☑ Redistribute OSPF Routes
□ Redistribute BGP Routes

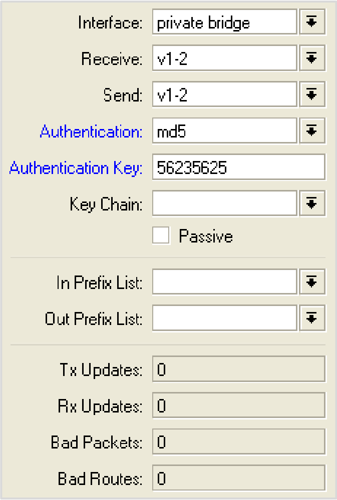
Default Route Metric: 1
Static Routes Metric: 1
Connected Routes Metric: 1
OSPF Routes Metric: 1
BGP Routes Metric: 1

Update Timer: 00:00:30
Timeout Timer: 00:03:00
Garbage Timer: 00:02:00

Routing Table: main

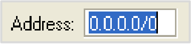
RIP or Routing Information Protocol, even though outmoded by newer protocols, is still in RouterOS. The hop count limit of 15 limits the network size as well as the speed of the routing updates. There are a few version of RIP, as well including RIPv1, RIPv2, and RIPng. RouterOS does support all three of these RIP versions.   Typical RIP updates go out every 30 seconds, so there is also a time delay in most routing updates. RIP is also considered an IGP or Interior Gateway Protocol and is not used on the Internet for routing. Note that most RIP systems have been replaced by OSPF or BGP in most modern networks.

To configure RIP, you need to define your RIP networks and interfaces. You can also further define how RIP distributes routes by configuring your RIP settings. To access the RIP menu, click on Routing → RIP in WinBox. The RIP settings window will look like the image to the right. Here you can change different route metrics, and timers, as well as configure what routes to distribute.



Interface: private bridge
Receive: v1-2
Send: v1-2
Authentication: md5
Authentication Key: 56235625
Key Chain:
□ Passive
In Prefix List:
Out Prefix List:
Tx Updates: 0
Rx Updates: 0
Bad Packets: 0
Bad Routes: 0

Once you have your RIP settings configured, you need to define how RIP talks on what interface. Under the Interfaces tab, you will need to add the interfaces that you wish to run RIP on. You can specify what version you wish to send and receive as well as specify an authentication key to prevent unknown devices from injecting routes. Then click on your networks tab and add the networks you wish to have RIP distribute normally. You can define all networks by entering a 0.0.0.0/0 if you wish.

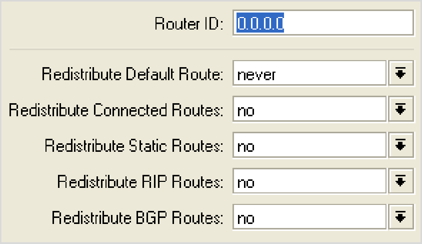


Address: 0.0.0.0/0

OSPF

Open Shortest Path First, (OSPF) is the primary IGP used in most networks today. It is also a link-state protocol. OSPF will send routing updates as interface states changes. If you unplug an Ethernet cable, OSPF sends an update. When you plug it back in, OSPF sends another update, and therefore it can make routing changes very quickly. OSPF uses protocol 86 to communicate between routers. To configure OSPF, click on Routing → OSPF in your WinBox application.

Inside your Interfaces tab you can also click the OSPF Settings button. Here, just like RIP, you will have the options about what routes to distribute, as well as the ability to change the default metrics.


Router ID: 0.0.0.0
Redistribute Default Route: never
Redistribute Connected Routes: no
Redistribute Static Routes: no
Redistribute RIP Routes: no
Redistribute BGP Routes: no

I recommend securing your OSPF network, using at least the MD5 password. The only other thing you need to do, just like RIP, is to configure the networks that you wish to distribute via the Networks tab. OSPF can separate systems into areas. By default, you will have a backbone area; and to start you can simply use that. If you have not defined any interfaces, and you are not using any type of security, then you may notice that you have Dynamic interfaces listed. These are OSPF neighbors that you have started communications with already. To secure these communications, you will need to add your interface and select the proper security method. I do recommend doing this to prevent bad and unknown routes from being injected into your OSPF network.

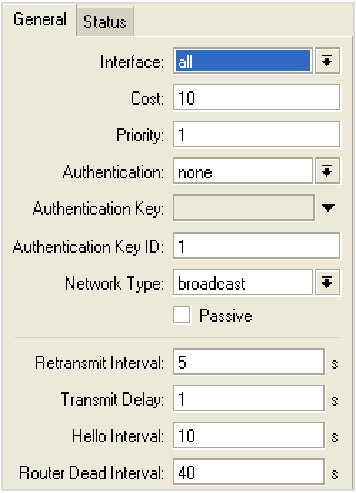
OSPF Loopback

By default the RouterID will be the largest IP address on the router, however, if you wish to manually configure these, using the loopback address would be the best. This will allow you to create a management subnet with a specific subnet as well as keep your RouterIDs from overlapping when using many different subnets on your network.

See the Loopback section for more information on using Loopback addressing on your OSPF network for network monitoring, as well as management on your network. The Loopback section will describe why this is important.

Changing Path Costs

OSPF's interface settings also allow you to change your path cost. You can specify that an interface has a higher path cost compared to another interface by using this method. If you have two links, and you wish to prefer the faster connection, you can simply make the cost on the slower link higher. If you want to do this both ways, you will have to do this on both interfaces so that traffic only flows on the faster connection and will only fail over to the slower connection if the primary fails.



Interface: all
Cost: 10
Priority: 1
Authentication: none
Authentication Key:
Authentication Key ID: 1
Network Type: broadcast
□ Passive
Retransmit Interval: 5 s
Transmit Delay: 1 s
Hello Interval: 10 s
Router Dead Interval: 40 s

OSPF Full-duplex Links

In the above text we described having two connections, one faster and used as the primary link and a slower backup link. If you have two links that are about the same speed but perform only in half-duplex, such as wireless links, you can set these links up to create a full-Full-duplex link right in OSPF. To do this, you will have two interfaces on side A and two interfaces on side B. On side A, you will increase the cost of interface two, and on side B you will increase the cost of interface one. Traffic going from A to B will use link one, but when traffic on side B goes back to A, that router will send it out the lower-cost link of link two.

This creates a full-duplex link and can be used with wireless interfaces as well. What is nice about using this method is that both links are still capable of doing two-way communications, so if one link fails, you still have connectivity, just not a full-duplex link. This would have marginal performance increase on a full-duplex circuit.

You also will need to take into account Firewalling with your OSPF full-duplex links. Seeing that you will only see one side of the communications, you will need to make sure that the routers or devices that you are running through are not blocking or stopping invalid packets. This is one of those rare instances where the invalid packets are actually legitimate traffic. Since the connection-tracking system can only see one direction of traffic, it will see OSPF full-duplex packets as invalid.

BGP

We do have full support for BPG within RouterOS. BGP or Boarder Gateway Protocol (BGP) is the key protocol on the Internet. It supplies inter-domain routing across the Internet and if you are going to multi-home to several providers then you will need to run BGP somewhere. Why should you run BGP with several providers is a question I get asked quite often! When you are running all private IPs behind your core router, then BGP is not really necessary. You can change providers, gateways and connections without much hassle. But when you end up with your own IP addresses, and your own AS (Autonomous System) number, you will need to eventually run BGP.

If you are running with a single Internet provider, BGP will not help your business that much, however, once you go with multiple providers, getting your own IPs and AS is the way to go. Now, you have your own IPs, they don't belong to your provider, they are yours. It doesn't matter what provider you wish to use (as long as they will establish a BGP session with you) and you can use your own IPs without issues. When you start running multiple providers, you can start load balancing, and shape your traffic across them.

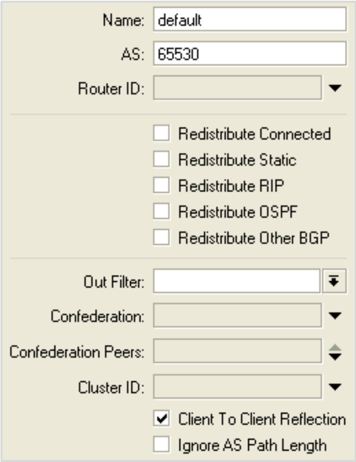
To get started you will need several things. First, you will need to configure the default BGP instance. This is basically changing the instance AS number to the one you have been assigned, then creating a BGP peer with the next router. Once you do that, everything else is modifying what routes are seen by each peer, as well as changing and modifying route information for your internal routing protocol. Most networks that I work with will run several BGP peers to multiple providers. This provides redundancy, but also allows us to load balance and provide symmetry across your network. If one peer goes down, the entire network, along with all of your public IPs are still reachable and able to use the Internet through the single peer.

Please keep in mind that there are entire books about BGP, how to optimize BGP, provide load balancing and symmetry and failover. Refer to other reference materials for more advanced configuration of BGP between providers.

Instances

When you start with RouterOS, you will need to create what is called an “instance”. RouterOS will have a default instance that you will need to edit to start off. Changing your AS number is the main task when starting. If you don’t define the Router ID, it will use the highest IP address on the router, however, this is typically not needed. To configure your RouterOS BGP Instance options, click on Routing → BGP → Instances tab. Then you can double click on the default Instance.

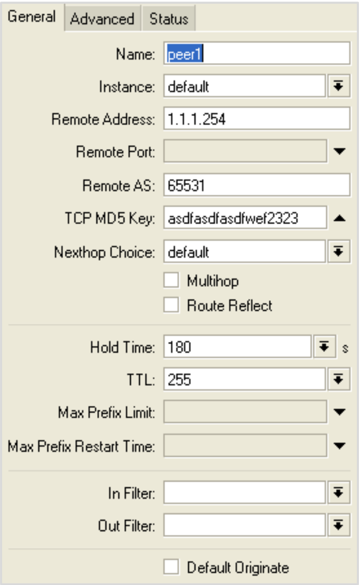
You will also have the options of redistributing your routes learned from other routing protocols. This is important because it will allow you to distribute routes that are running on the inside of your network. I recommend also setting up an out filter here as well because you should not wish to distribute IPs that are not yours or IPs that are not valid, such as private addresses.



Name: default
AS: 65530
Router ID:
□ Redistribute Connected
□ Redistribute Static
□ Redistribute RIP
□ Redistribute OSPF
□ Redistribute Other BGP
Out Filter:
Confederation:
Confederation Peers:
Cluster ID:
☑ Client To Client Reflection
□ Ignore AS Path Length

Peers

The second step after configuring your instance is to configure a BGP peer. This is simpler than it sounds; keep in mind that you will have to have IP connectivity. Most providers will assign a /30 or /29 for routing between your network and them. One of those IPs will be your router and one will be theirs. Theirs will normally also be the BGP peer router as well. You can also use BGP multi-hop as well to provide a BGP peer. We will cover that a bit further in the chapter.



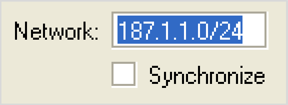
Name: peer1
Instance: default
Remote Address: 1.1.1.254
Remote Port:
Remote AS: 65531
TCP MD5 Key: asdfasdfasdfwef2323
Nexthop Choice: default
□ Multihop
□ Route Reflect
Hold Time: 180 s
TTL: 255
Max Prefix Limit:
Max Prefix Restart Time:
In Filter:
Out Filter:
□ Default Originate

To create a BGP peer, start by going to the Peers tab in your BGP configuration. Here, you can click the plus button and create a new BGP peer. There are only a few items that you need to have to create a peer.

Specify the instance that you will be using, the remote IP for your peer, as well as BGP port to be used, the remote AS, and the MD5 key. Once you do this, you should be able to establish a BGP session. This is very fast and secure and you should have routes quickly.

Your provider may have other settings, including route reflects, different hold times and TTLs. You typically will need to work with your provider or peer to ensure connectivity. Inside the peer you have options for your in and out filters. These again, are used to filter routes that come in and out of your BGP session.

Networks

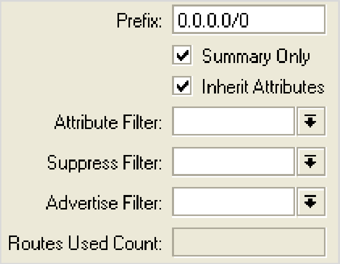


Network: 187.1.1.0/24

□ Synchronize

The networks in RouterOS are a listing of IP prefixes that will be advertised to your peers. If you have not placed filters in your BGP system, and you type in a network here, BGP will advertise this network. The synchronize box, will first ensure that some part of the network was learned via an IGP (Interior Gateway Protocol), this would be via OSPF or RIP. For instance, if you put in the above network, 187.1.1.0/24 and check the Synchronize box, unless you have some 187.1.1.x subnet in your routing table, it will not be advertised. If you only have an 187.1.1.0/30 it will be advertised though.

Aggregates



Prefix: 0.0.0.0/0

☑ Summary Only

☑ Inherit Attributes

Attribute Filter:

Suppress Filter:

Advertise Filter:

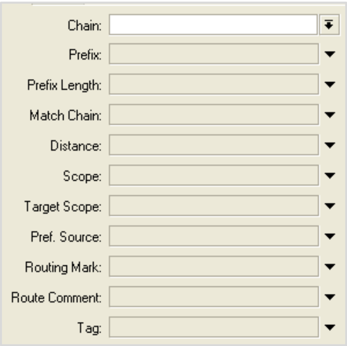
Routes Used Count:

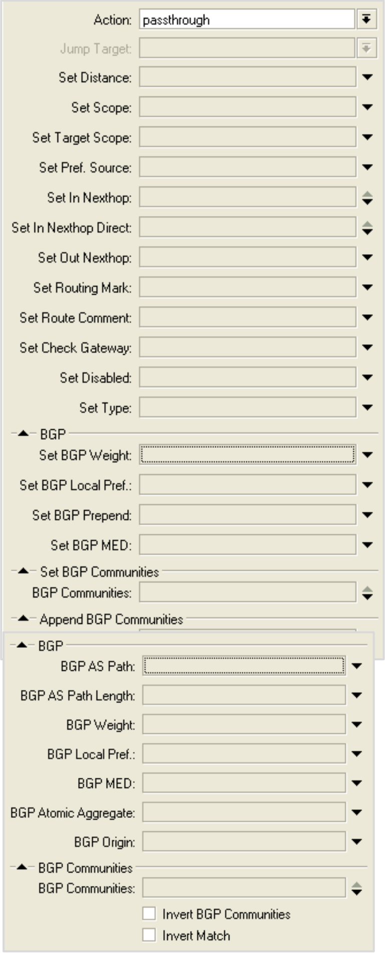
BGP Aggregates are meant to summarize or only send specific prefixes instead of sending the entire routing table. If you have an entire /24 subnet dedicated to /30 subnets, you don't want to advertise 64 /30s. You just need to advertise the entire /24 or more to the Internet. In this event, you would use the BGP aggregates to summarize the networks into one /24.

Here you can also specify Suppression and Attribute filters, as well as Advertising filters. If you check the box to Inherit Attributes, any BGP attributes that were learned from the smaller subnets will be carried over into the summarization.

Routing Filters

Inside dynamic routing we can use filters to filter and change routing as we wish. This is all done in the routing filters. To get to the routing filters, click Routing → Filters. Here, just like other filters in the Firewall / Manage and Filters section, we can define multiple chains. These chains are used when defining in, out, attribute, suppression, and advertising filters in your dynamic routing protocols. Instead of specifying a source IP we are defining prefixes and then prefix lengths.




Action: passthrough

There are plenty of options here to match data. This is just like matching data with your Firewall or Mangle rules, but now you are matching routes or prefixes! You can also match by BGP information as well, such as communities, MEDs, or even AS paths. You also have the option to invert your matches as well.

Once you get your matches, you can then perform an action. Most of your actions are going to be passing through, because you want the data to run through your router, unless you wish to drop or discard routes. There are many options including BGP and community options. Most of these options though will be through BGP sessions. The ones that I commonly use are the BGP Prepends and Local Pref as well!

BGP Configurations

In this section we will discuss several different common configurations that providers frequently use BGP in. Since BGP is really designed for multi-homed networks or networks with more than one upstream provider, these are what we will focus on.

Default/Summary/Full Routing Table

When we talk BGP to another peer, we typically have options specifying what kind of routes we receive from them. The three most common are default, summary and full routes. Most of the time, the default, or 0.0.0.0/0 route is the only route that most configurations need. Summary routes are exactly that, a summarization of the major network blocks, greatly reducing the size of the routing table and the time it takes for your router to receive those routes from your peer. The largest routing update that we can receive is a full Internet routing table; this is upwards of 400,000 routes! This can take considerable time to receive and in many cases is simply not needed.

Single Router – Primary and /Backup Upstream Providers

I have seen configurations like this for providers that have started with T1s, and have since has since moved to MetroE or fiber service. In most cases, when they move to the new circuit, it's typically with another provider. They want to be able to use their T1s or older and slower circuit in case of a primary circuit outage.

In this scenario, you would form two peers, one with each provider. You will also only need default routes from your providers. Since you are using one provider as primary, you simply will ensure that that provider's default route is lower cost than the secondary provider. The secondary provider will also send only the default route, however its cost will be higher. You can modify this cost using your routing filters.

Setting the default route cost inside your network is fairly simple thing; however, the inbound traffic requires a bit more configuration. The simplest method to still advertise your routes to your backup provider, but not use it, is to simply add prepends. 5 to 10 prepends is more than enough. Prepend simply adds your AS several times, however many times you prepend into the routing table. This causes other routers to see the prepended route as a higher cost. In many cases your network is within 5-6 AS numbers away from everyone else. If you prepend at least 6 times, everyone on the global table should see your primary provider as the lowest cost route, and will use it.

To further expand on prepending, assume that provider A, AS 532, you do not prepend on and provider B, AS 555, you prepend four times to. If a user connected to another network is on AS (anonymous system) 114. AS 114 both directly connects to AS 532 and 555. You directly peer with 532 and 555 as well. Your AS number will be 111. In this case, from AS 114, it will receive two routes for your prefix, one from each peer. The one from AS 532 will show a AS path of 532 114. The prefix from AS 555 will have an AS path of 555 114 114 114 144. The question is which is shorter? In this case, of course the path though AS 532 is shorter; therefor the one AS 114 will use to send data to. Now let's assume some event causes your peer with provider A to go down. The global routing table is updated, and now AS 114 only receives on route with an AS path of 555 114 114 114 144. Even though this is five AS's away it’s the only path, therefore it will be used.

Once your primary provider goes off-line, the BGP peer to your primary provider will go down; hence the advertisements will be removed from your primary provider and all of your inbound traffic will start to come in your secondary provider. Your outbound traffic will follow suit as well, as the primary BGP default route will be removed from your routing table and the only one left, even though it is a higher cost, will be used.

Single Router – Same Speed Upstream Providers, Load Balance Traffic

Another simple BGP configuration is when network interconnects with two other providers. In this case your internet providers are at the same location and connected to the same router. By forming BGP peers with each of your providers, you can announce your IP address space out to the world, and in the event that one provider fails, your IP space will still be globally accessible though the second provider.

In most typical networks you will have some form of IGP (interior gateway protocol), such as OSPF or RIP, running on the inside of your network. On your edge device, you will form two BGP peers, one with each of your providers. In this case, you typically will want to receive the full internet routing table from each provider. The reason why you would wish to do this is because you would typically want to have your outbound traffic going out the provider that the destination is the closest too.

As an example of this; let's assume you have two providers, A and B. You also have a customer that creates a VPN on your network out to another site. That site has a connection with provider B. In this case, you would want the outbound traffic from this customer to go out provider B and the inbound traffic from that VPN to come in provider B. By receiving full routing tables, your single router can determine what IP blocks are closer on provider A and what IP blocks are closers on provider B. It will automatically send traffic out using lowest cost route, either using provider A or B. We can massage the outbound traffic, by adding route filters for specific IP blocks to either increase or decrease the cost we calculate upon receiving those blocks from each provider.

We also have to watch our traffic and our upstream providers to ensure this occurs. If you peer with a smaller local company, and then peer with a large international provider, you will typically find that most of your traffic will go out the larger provider. This is not necessarily bad, but you will find that your smaller provider's bandwidth is be less used.

How we get to the inbound data is determined by or our advertisements of our IP space, or prefixes, to the global routing table. Again, if you do not change any settings, all of your blocks will be advertised at the same cost. Again, this is not necessarily bad or wrong. If you use two international providers, this may work out great and inbound traffic will be fairly balanced between the two. Regarding our example of one provider being a small local provider and the other provider a large international provider, again, you will find that your inbound traffic will prefer your larger provider's connection. If you are using two connections from the same provider you typically will see a good balance though.

If you have two providers, one the large international provider and one that is a smaller local provider and you advertise your prefixes to each equally, with no prepends on either, then you typically will see the majority of your inbound traffic coming in on the peer with the international provider. The larger provider is simply better connected to other AS's. Now, assume that the larger provider is being preferred so much that you are overloading your connection, but you have an abundance of bandwidth available with the smaller provider. Simply add smaller blocks without prepends to the smaller provider, and then advertise only the large block out to the larger provider.

This will get them coming in through your smaller provider and thus shifting inbound traffic away from your larger provider.

Let's use a /22 as an example of shifting inbound load. In this /22 we have four /24 subnets. If most of our traffic is coming in the larger provider, we need to offload some of that traffic to the smaller provider. We started by simply having the entire /22 advertised with no prepends on either provider. This is fine as it gives us our redundancy. Now we will add one or two of the /24s to be advertised out the smaller provider. Since the /24s are more specific prefixes, they will be used instead of the larger /22. 99% of the /24s that we advertised would come in through the smaller provider. If we lost connectivity to the smaller provider, the /22 would reroute traffic though the major provider as it’s still a valid route.

Is modifying the inbound/outbound costs what you really want to do? If you are looking to balance your connections or if there are different costs involved between providers, then there may be business reasons to do this. If you have two 50 Mbps Internet connections and you are starting to see periods of time where your connection to the larger provider is being maxed out, then you may wish to push some traffic out your smaller provider before purchasing more bandwidth from the larger provider. However, is this really going to provide your customers with the fastest possible connections and best routing path? You should answer this is a question that you will have to answer both technically and in your and from a business perspective.

Two Connections – Diverse Peering Locations

Having two, or more, providers at different peering locations on your network provide another challenge to your network operations and business practices. You have two high speed connections at different points on your network, and you are paying for them so you want to utilize them, but you also want to provide the fastest service possible and have the redundancy that BGP can give you. There are two distinct options here, and a question that you have to ask about your network. The question is; do you have a high speed connection between your two diverse on-network locations?

High Speed Connection Between Peers

If you do have a high speed connection, and if it is figure 75% as large as of your upstream provider’s bandwidth to be a good number, then you can have the best of both worlds; -diverse connections but also proper route selection.

In this situation you will be able to receive full routing tables from each provider, and then form a BGP peer between your two routers, typically over a VPN or tunnel type of connection. By doing this, your routers learn of each provider's routes and costs, and just like in the single router examples, routing takes place on a lowest cost basis. Unlike OSPF however, the connection between the two routers inside the same AS is not really counted in the cost path. If your customers outbound connection hits provider A's edge router, but that router identifies that the IP block the customer wants wishing to go to is on provider B's network, router A will send that data over the VPN or tunnel link to router B to be routed out to provider B. This would works in reverse as well!

This gives you the ability to use the closest and lowest cost routing path on outbound connections. Inbound routing would take place the same way; however, it would not look at your internal path costs! If your public IP space is advertised as the same cost on both providers, then inbound traffic will take the lowest-cost path. Now this is where we get to undesirable paths. For instance, if inbound data comes in on provider B, but the end IP address is routed to an IP that is at the datacenter where your provider A connection is. You are now transporting that data across your network and using taking up more resources. This goes back to the question of if you have a high speed connection between your BGP peering locations. If you do then this may not be of major significance.

No High Speed Connection Between Peers

A large number of providers do not have a large, on-network connection between their BGP peering points. A network may already be transporting lots of traffic and we may not have the transport resources to carry traffic from the provider B peering point to a peering point location near provider A. Now, we need to come up with another method, or in this case, a scheme to balance this out as much as possible. The most common method is to break your network up into segments. These segments will have specific /24 subnets assigned to each of them. These subnets will be then advertised at the lowest cost out the nearest peer. They will also be advertised out the other peer(s) that we have but at a much higher cost. By doing this, we still expect that some traffic will come from provider B destined for provider A's IP blocks, but most of the traffic will come in provider A. If provider A goes down, the IP blocks will still be accessible though the higher-cost advertisements out provider B.

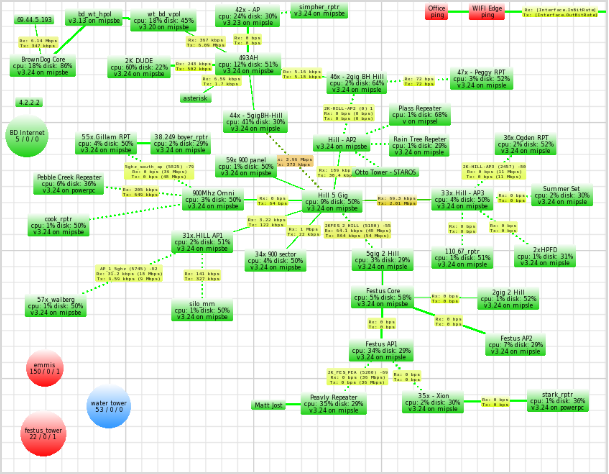
We also need to split and manage our outbound traffic. By using an IGP, such as OSPF, we can split our network between the two providers. Adding costs to links between the sections of our network. This then allows half of our network to have a default route of provider A, and the other half to have a default route to go out provider B. We will advertise the /24s that are on provider A's side of the network out provider A without any additional costs, but also advertise them out provider B, but with added cost. We will also do the same thing for the /24s on the provider B side of the network.

When we do this, splitting of our network, if an entire provider goes down, our outbound traffic, (even though going over a high cost OSPF link) will still go to the provider that is still up and running. The peer that is still connected will still be advertising all of the IP space, some at high cost and some at low cost. We also do not need to form a BGP peer between our routers, nor do we need to get full tables, because we do not need to know about anything else. We simply send the outbound data to the closest default gateway. The process of balancing of the two connections is simply a mechanism to break the network into multiple segments. Each segment will have their own provider, and under normal operations, both outbound and inbound traffic will take the closest peer to that segment. Only in the event of a failure would traffic go out the other provider.

The Dude NMS

MikroTik had a need for a centralized monitoring and management application to manage RouterOS systems. They started to develop an application called “The Dude”. There is an entire story on how The Dude got its name, but I won’t bore you with that! What we are going to cover is installing, configuring, and running The Dude system.

The Dude is more than just a RouterOS management tool. Yes; you can use The Dude to perform upgrades quickly with just a few mouse clicks, but it is also a very powerful NMS or network monitoring system (NMS). You can monitor the up and down status of virtually any kind of network device. You can setup SNMP probes that delivering detailed network information right onto your desktop. It is multi-user capable via both its own Dude Client interface or through Dudes web-based interface. Network alerts, e-mails and SMS are all parts of the Dude package. You can also run Dude on any Windows PC and even on some RouterBOARDs as a package. With all of these features, you would expect The Dude to be expensive; RouterOS has made this useful tool completely free. Even if you don’t run RouterOS, you can still use it to monitor networks, track bandwidth usages and manage devices.

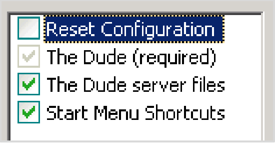


Installation

Installing The Dude is very easy regardless if it is on a Windows PC, Linux box, or a RouterBOARD. You will need to download the package for the platform that you wish to use it on from MikroTik's website at http://www.MikroTik com/thedude.php. Here you can download the Dude for Windows, or the optional RouterOS packages. With the RouterOS packages, remember that you will need to ensure you get the right processor version for your RouterBOARD product.

Windows Installation

Installing the Windows version of The Dude is just like installing any other Windows application. You will Download the windows installation file, and run it. Agree to the setup terms, and select the components to install. Dude really has two main components, the Server and Client. In the Windows installation, you can install both the Server and the Client at the same time. The required component is the Client, and the Server files will allow you to run a Dude Server on your PC. Dude does have the capabilities to run the Server as a service under just about any of your Windows versions, but that is configured in the Server settings. If you check the Reset Configuration box, this will wipe the configuration data files and let you start over. I normally never need this as there is also a Reset Configuration option inside the Client application.



☐ Reset Configuration
☑ The Dude (required)
☑ The Dude server files
☑ Start Menu Shortcuts

After the selection of the components to install, you will then select what folder you wish to install in. Here is a little trick that I like to do. Keep in mind that I use The Dude everyday on many different networks. We have many different Dude Servers and versions out on many different networks. I have to be able to quickly change between different Dude versions and Servers all of the time. When I select the installation folder for the Dude, I install it in a folder with the version information. For example, for The Dude v3 RC2, I installed it in a folder called Dude3rc2. This way I can have different versions running at the same time as well.

RouterOS Installation

Installation of the Dude on RouterOS is as simple as a package installation. RouterOS will have a NPK file that you will simply copy to the root folder and reboot the router; however there are some restrictions that I would recommend you use. You can install Dude on a 100 series RouterBOARD if you wished, however, due to memory and disk space constraints, I would highly recommend against doing so. Dude is a decent sized package and it does use RAM. After it monitors and collects data for a while, I have seen Dude installations that have become quite large. I have used RouterBOARD 433Ahs with a one or two GB Micro-SD card for storage of Dude Data. I have seen Dude data exceed 800 MB before, so make sure you have the extra storage space. Something else to take into consideration is that even if you think you have enough storage, you may not because when you make a backup of the Dude application, it creates a XML file, which it has to be (yes; you guessed it) stored somewhere before you can download it.

Dude Agents

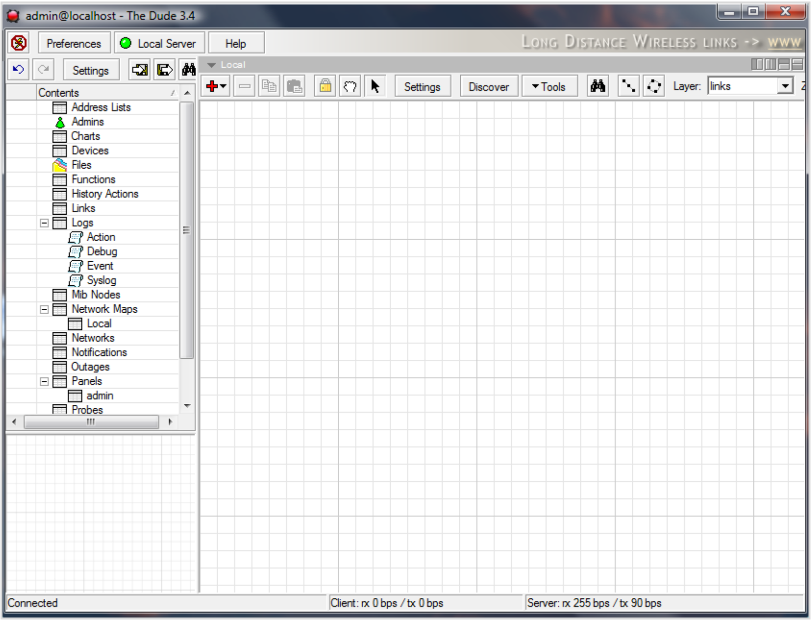
A Dude agent is a Dude Server acting on behalf of the primary Server. No data and configuration is stored on this other than a username/password to secure that Dude Server. Your primary Server will be programmed to use the agent to get to subnets that are not normally accessible by the primary Dude Server. For example, if you may have multiple hotspot networks behind different types of broadband connections, and these hotspots may share the same common IP structure. In this case, if you had a single Dude Server, you would normally only be able to ping and monitor devices with public IPs for the most part. However, with a Dude agent, your primary Dude Server can request the Dude Agent that has both a public IP and a private IP to ping the private IP. Since the only private IPs the agent can ping are the ones local to itself, you can monitor the entire private subnet behind the NAT with the Agent. If anything ever happened to the Agent box, nothing is lost, as the entire configuration is located on the primary Dude Server!

Installation of a Dude Agent

Well, there is none! You will simply install the Dude service into RouterOS. I would also login to that Dude Server and put a username/password on it as well as secure it with the Firewall on the RouterOS system, however, that’s it. Now you will simply make calls from the primary Dude Server to the agent.

Dude Layout

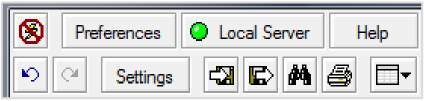
Once you perform the initial installation, you should get an application like the following:

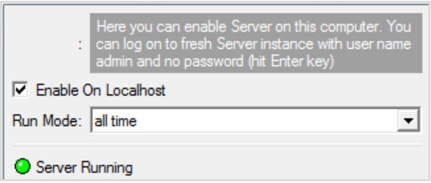


This is the initial screen area for the Dude application. In the upper left, we have the settings, Server and other command buttons. Along the left we have our contents to get into all of the sections of the Dude application; below that, a quick reference map window. The main application screen to the right is where your maps will go!

Running a Server

The Dude application installs the Dude Server, if you checked it, and the green indicator light shows that you have a local server running by the green indicator light. If you click Local Server, this will give you your options for the Dude Server. If you uncheck the "Enable On Localhost" box, the Dude Server will stop running. If you give it a second or two, you will note that the green indicator will change to grey, showing that the server application is no longer running in the background.





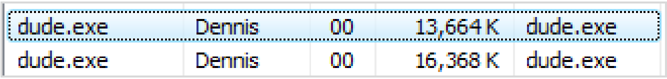
Here you can enable Server on this computer. You can log on to fresh Server instance with user name admin and no password (hit Enter key)

☑ Enable On Localhost

Run Mode: all time

Server Running

There are several server running modes. The default mode is for the server to start with the client and stay running until the computer is rebooted, however, this does NOT start the server when you start your computer. The second mode is "only when local client is running", and will do exactly what it says! When you start the Dude client application, the server will run, and when you close the client, it will stop the server. The last mode is "As a Service". This mode installs a "The Dude" Service into Windows XP, or greater allowing the Dude service to start with the workstation or server in question.

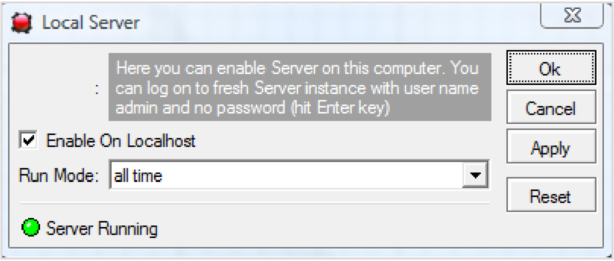


dude.exe	Dennis	00	13,664 K	dude.exe
dude.exe	Dennis	00	16,368 K	dude.exe

I want to point out, that upon the installation of the Dude and the Dude server files; you will have a local server running. When you execute the Dude application, it actually starts two copies of the Dude.exe file. One is the server and one is the client that you are using to communicate with the server. exe that is running in the background.

Resetting Configuration

Inside the Local Server dialog box there is a reset button. This button resets the Dude configuration back to just after the installation, clearing out anything that you may have configured or installed.



Local Server

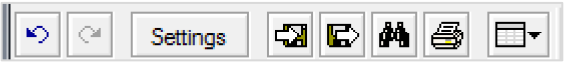
Here you can enable Server on this computer. You can log on to fresh Server instance with user name admin and no password (hit Enter key)

☑ Enable On Localhost

Run Mode: all time

● Server Running

Menus and Options



The left context menu has the management features of your Dude System. The first ↶ is an undo command for The Dude. If you delete something, you can undo this. There is also an undo contents list that shows the commands that you can undo. The ↷ is a redo command option in case you wish to redo the undo!

The settings button here goes to the main server configuration. I will cover that in the next section.

These buttons are very important as they are your export and import commands. The export button, on the right, tells the Dude to generate a XML file with all of your Dude data. It stores this on the disk, and then prompts you to download the file, or save it somewhere. When doing exports, you need to keep in mind that the XML file contains everything! What is everything? First and foremost are the devices, what and how you are monitoring them, what map they are on and how to notify you. It also includes the server configuration that you have. Any background images that you put in your Dude application, they are in the XML file, as well as other files images, and even RouterOS NPK files that you have in for upgrading from the Dude Server are included in the XML file! Think about how big that file can get big really quickly.

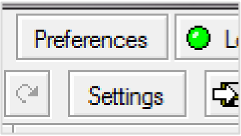
The import button, the one on the left, does the exact opposite of the export; it takes an import XML file and imports it into the system. Once done, the Dude server will restart and apply that configuration. This sometimes can take a few minutes on RouterBOARDs. You will see their CPU jump to 100% for several minutes and will be unable to connect to the Dude during this time. I assume that the Dude application is loading up the XML file during this time.

Above the Map window you have more commands. The or (plus sign), just like in RouterOS, allows you to add a number of items into your map. Things like your Devices, links, other maps and so forth. You also have a (minus button) to remove objects that are selected, -again, just like inside RouterOS. Like most windows applications, you also have the copy and paste commands here as well. The lock prevents movement of your devices and links on the map. The hand tool allows you to click to drag your map instead of scrolling, and the pointer, allows you to select objects in you map.

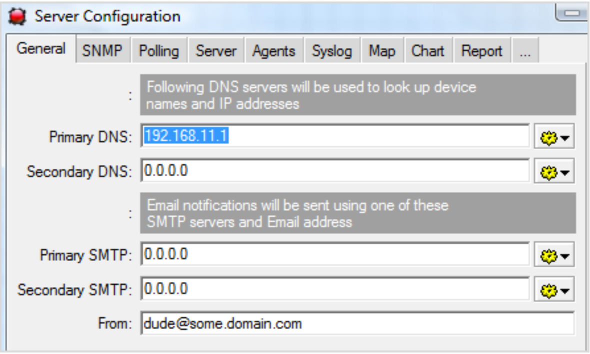
The Settings button above the map is the map settings button. We will discuss this later as well. The Discover button activates the discovery tools, and the tools section allows you to export the map to an image file and helps you automatically lay out devices.

Server Configuration

The Dude application has a lot of configuration options. I will cover the ones that are not cosmetic. To get to the server configuration, click on the Settings button right below the Preferences. This will get you into the server configuration for your Dude Server.



Inside the Server Configuration Options you will have a number of tabs. Be sure to see that the structure of configurations, tabs and menus are very close to that of RouterOS. The movement throughout both The Dude and RouterOS is similar. The first tab we get is the General tab. This gives your Dude server the primary and secondary DNS and SMTP server. The From option is the mail account that the e-mail should appear to come from. At the time of the writing of this book, RouterOS has an authenticated e-mail system, however the Dude does not. So, for your Dude system to be able to send e-mails, you MUST have a mail server that will accept un-authenticated SMTP e-mails. You can do this by having a mail server that will take all mail from the IP address of your Dude box, and/or accept all mail from specific e-mail addresses regardless of authentication. I would opt for the IP address for security reasons.



: Following DNS servers will be used to look up device names and IP addresses

Primary DNS: 192.168.11.1

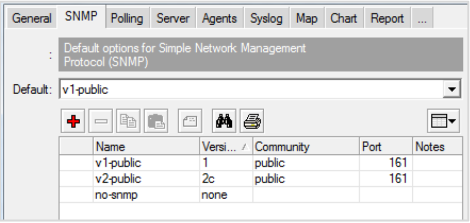
Secondary DNS: 0.0.0.0

: Email notifications will be sent using one of these SMTP servers and Email address

Primary SMTP: 0.0.0.0

Secondary SMTP: 0.0.0.0

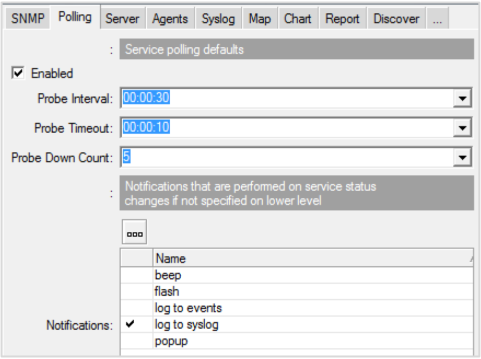
From: dude@some.domain.com



: Default options for Simple Network Management Protocol (SNMP)

Default: v1-public

Name	Versi...	Community	Port	Notes
v1-public	1	public	161	
v2-public	2c	public	161	
no-snmp	none			



: Service polling defaults

☑ Enabled

Probe Interval: 00:00:30

Probe Timeout: 00:00:10

Probe Down Count: 5

: Notifications that are performed on service status changes if not specified on lower level

Notifications:

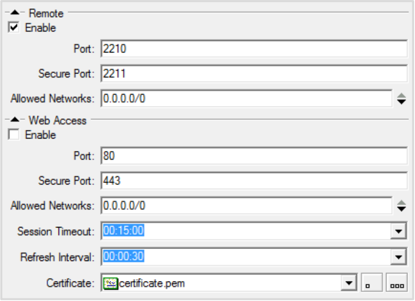
	Name
	beep
	flash
	log to events
✓	log to syslog
	popup

The next tab is the SNMP tab. The Dude allows you to have several different SNMP community strings active at once. Inside this tab, you can create several SNMP profiles. Within each one, you can select what version of SNMP you wish to use, what community string, as well as what port. You can also tag on notes in case you have an odd device out there and want someone to be able to remember what that SNMP profile is for.

The Polling tab allows you to setup the default polling times and notification events for new devices. You can enable or disable the polling options and well as control how often, when to consider the probe timed out, and how many probes must time out before you get an alert. The bottom section is your notifications. This allows you to set the default notifications for new devices. I typically would configure the notifications first when building my Dude server, as I want the notification options on all of the devices that I add anyways, however as you grow this may not be an option for your network.

Configuration of Dude Servers

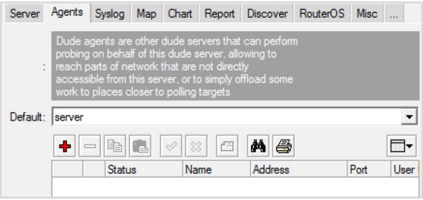
The Server tab is very important. This controls the remote server as well as the web server application. The default is to allow remote connections to your Dude server. These connections are other Dude client's attempting to connect to the Dude server. The web server portion allows you to access the basic Dude information, device up/down status and maps on a web server port. I use the web server portion to allow users access to maps as well as up/down status for devices, but I don't want them to edit data or have to install the Dude client software. I have used this very successfully with call centers to allow their agents to check on network status with a click of a button instead of again, having to have that client software loaded on each PC.



Remote
☑ Enable
Port: 2210
Secure Port: 2211
Allowed Networks: 0.0.0.0/0

Web Access
□ Enable
Port: 80
Secure Port: 443
Allowed Networks: 0.0.0.0/0
Session Timeout: 00:15:00
Refresh Interval: 00:00:30
Certificate: certificate.pem

Dude Agents



Dude agents are other dude servers that can perform probing on behalf of this dude server, allowing to reach parts of network that are not directly accessible from this server, or to simply offload some work to places closer to polling targets

Default: server

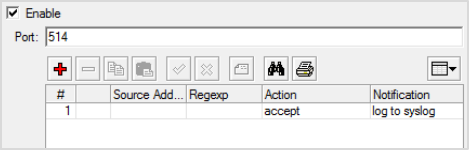
	Status	Name	Address	Port	User

Dude agents are Dude servers that function as an agent of a primary Dude server. These Dude servers allow you to relay your Dude probes though them. One of the ways I use Dude servers for is to get by a NAT system. To setup Dude agents, you simply install a Dude server. It can be on a PC system or on a RouterBOARD. Configure a password to secure it, and then you will add it a as a Dude Agent in your server configuration. You will need the IP, name, port as well as a username/password to connect to it. Here it will show the status of the agent, and if it is on-line or not. You can configure a number of them as you need.

With Dude agents, the Dude server that is not an agent acts as the central database. No information pertaining to the relay of probes though the Dude Agent is stored on the agent; all configurations are stored on that central Dude server.

Dudes Syslog Server

Dude also operates a Syslog server to handle all of your logging needs. This database can get quite large sometimes; however it does work quite well. To enable your Syslog server, go to the Syslog tab under the Server Configuration options. By default, the Syslog server is enabled; however, I do recommend that you secure it with a better set of rules. These rules are just like Firewall rules, except they are only are for data coming in on your Syslog server.

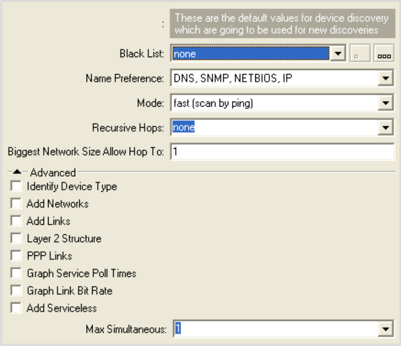


☑ Enable

Port: 514

#		Source Add...	Regexp	Action	Notification
1				accept	log to syslog

Dude Discovery Services



: These are the default values for device discovery which are going to be used for new discoveries

Black List: none

Name Preference: DNS, SNMP, NETBIOS, IP

Mode: fast (scan by ping)

Recursive Hops: none

Biggest Network Size Allow Hop To: 1

Advanced

□ Identify Device Type

□ Add Networks

□ Add Links

□ Layer 2 Structure

□ PPP Links

□ Graph Service Poll Times

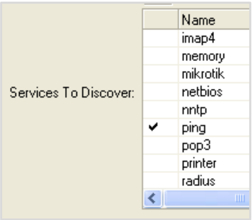
□ Graph Link Bit Rate

□ Add Serviceless

Max Simultaneous: 1

In the Discover tab you will have all of the discovery options that the Dude offers. As I start this section, it was hard to figure out what I wanted to say, so let me put it this way. If you know the layout of your network, then there should be no reason for you to need to do a discovery process. I would highly recommend that if you can avoid the discovery process, do so. Even though you have many network devices that can be discovered, the discovery process can lead to a messy map. I prefer to build maps and Dude systems from scratch to ensure that I know exactly what is on there and where devices and links go. Also, this helps you understand your network as you build it.

There are other features of the discovery service that may be useful. Specifically, I like to leave the service discovery on, but limit it to the specific types of probes that I wish it to discover. Specifically, pings are what I am mostly interested in, however sometimes CPU comes into play as well.



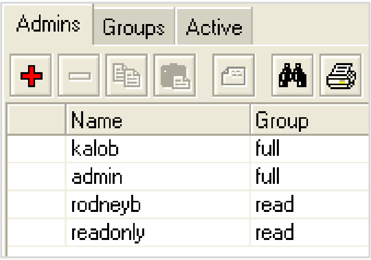
Services To Discover:

	Name
	imap4
	memory
	mikrotik
	netbios
	nntp
✓	ping
	pop3
	printer
	radius

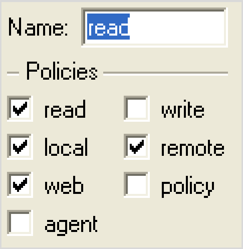
Note that in the screenshot above, I have turned off most of the discovery services, and I do not let these services run on to a big network. The reason for doing something like this is to prevent the discovery service from starting to run, as during its process, it sends out hundreds of connections and probes looking for devices etc. On a small network, this is okay, but on larger networks it can cause huge outages and have messy results. In the services to discover, I will leave on ping on. That way I can discover the ping probe as I add many devices, however, if you wish, you can discover other services as well, as this service discovery process can be pointed at a single device while you are creating it on the map. On the rest of the options, such as the device types, I typically turn these off as well. Keep in mind that these are my personal preferences and yours may vary.

Admins

The Admins section of the Dude is the user management system. This allows users to login to the Dude application, view network status, and login abilities to the web access if applicable. Just like RouterOS's user management system, you have groups where you can specify rights and policies. You can control policies, the ability to login remotely, via the web, login locally, and specify as well as if they have read/write actions as well are all policies that you can control. The agent policy is to allow that user to use this Dude server as an agent. If the user account that you are trying to use in the server configuration for agents is not setup with the agent policy then the agent relaying will not work.



Name	Group
kalob	full
admin	full
rodneyb	read
readonly	read

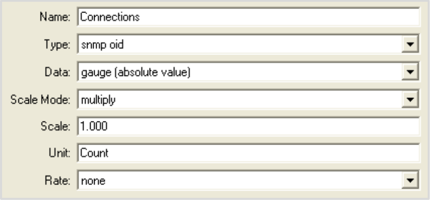

Name: read

Policies

☑ read ☐ write
☑ local ☑ remote
☑ web ☐ policy
☐ agent

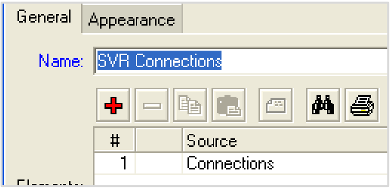
Charts

Charting in the Dude is done by specifying values units, and a scale based on some data source. Dude already has many data sources as you are collecting data from the devices that you are monitoring. However, you can use SNMP (Simple Network Management Protocol) OID (Object Identifier) and functions to collect the data. Building functions is outside the scope of this book. I will share a graph that I built to monitor TCP connections on a Windows server. This may not be something that you can use, but you may be able to modify it for your needs.


Name: Connections
Type: snmp oid
Data: gauge (absolute value)
Scale Mode: multiply
Scale: 1.000
Unit: Count
Rate: none

To start, we create a new data source, to get here, open your Charts pane, and then you should have two tabs at the top; one is for your Charts and one is for your data sources. On the Data Source tab, click plus to add a new source. We will name this, and then make the type SNMP OID. Our data is going be an absolute value; i.e. the value that is returned is what we wish to see. The scale mode in my case is Multiply, but we have a scale value of 1, so even though the calculation is done, it does not modify the number. The unit will be connection counts.

Next, we will fill in where to get the information. In this case, the bottom half of our connection information contains the address of the server, the Dude agent, the SNMP Profile (remember we have to be able to pull that SNMP information), as well as the OID that we wish to view, plus how often we wish to pull that data.



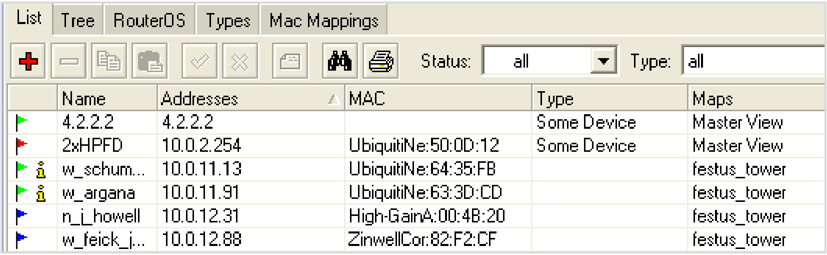
Name: SVR Connections

#	Source
1	Connections

Next, we have to create our chart. Simply Click the chart tab, then click plus and create a chart with the chart name you wish to have. Once the chart is created we get the chart Elements box. Here we are going to add our connections data source that we just created. By doing this, we should get a chart of our server connections; remember you will have to wait a while to be able to see some data, as it only polls at the intervals that you put in the data source.

Devices

The Devices pane gives you a lot of information about your individual devices. Devices are objects that you wish to monitor, and this pane will give you detailed information about each one. What is also nice is that, Dude has the ability to covert the MAC to a brand, so in the image below, you will see several MAC addresses but with the brand of device that is connected. This can be helpful to determine what gear the customer may be using.

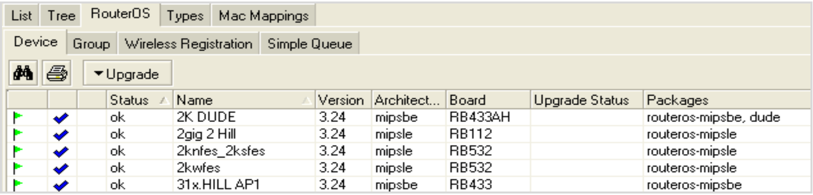


Status: all Type: all

	Name	Addresses	MAC	Type	Maps
⚑	4.2.2.2	4.2.2.2		Some Device	Master View
⚑	2xHPFD	10.0.2.254	UbiquitiNe:50:0D:12	Some Device	Master View
⚑ ℹ	w_schum...	10.0.11.13	UbiquitiNe:64:35:FB		festus_tower
⚑ ℹ	w_argana	10.0.11.91	UbiquitiNe:63:3D:CD		festus_tower
⚑	n_j_howell	10.0.12.31	High-GainA:00:4B:20		festus_tower
⚑	w_feick_j...	10.0.12.88	ZinwellCor:82:F2:CF		festus_tower

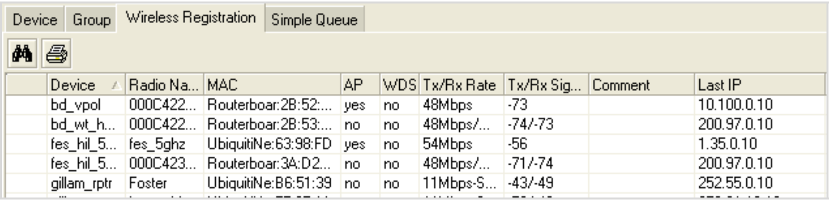
Inside here, we have also have a lot of information as well. You can add notes to each device. This is extremely useful when you are swapping radios or creating a service history on the device. The tree view is not as useful; however the RouterOS tab can get you some wonderful information. As you can see below, under the RouterOS tab, you will get information on your

RouterOS devices, such as the name, version board and what packages are installed.



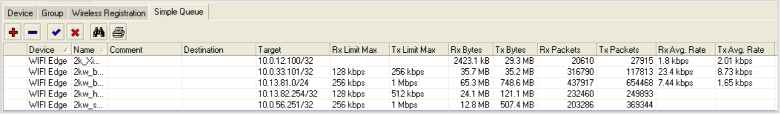
			Status	Name	Version	Architect...	Board	Upgrade Status	Packages
⚑	✓		ok	2K DUDE	3.24	mipsbe	RB433AH		routeros-mipsbe, dude
⚑	✓		ok	2gig 2 Hill	3.24	mipsle	RB112		routeros-mipsle
⚑	✓		ok	2knfes_2ksfes	3.24	mipsle	RB532		routeros-mipsle
⚑	✓		ok	2kwfes	3.24	mipsle	RB532		routeros-mipsle
⚑	✓		ok	31x.HILL AP1	3.24	mipsbe	RB433		routeros-mipsbe

Also you will see a Status; this means that the username/password that is stored in the Dude for the device allows us to connect to the device to get information. If you can't connect, you cannot get this information. Dude is constantly checking this so this gives you a good way to find out what Dude devices needs to be updated to ensure you have the proper password for them.



	Device	Radio Na...	MAC	AP	WDS	Tx/Rx Rate	Tx/Rx Sig...	Comment	Last IP
	bd_vpol	000C422...	Routerboar:2B:52:...	yes	no	48Mbps	-73		10.100.0.10
	bd_wt_h...	000C422...	Routerboar:2B:53:...	no	no	48Mbps/...	-74/-73		200.97.0.10
	fes_hil_5...	fes_5ghz	UbiquitiNe:63:98:FD	yes	no	54Mbps	-56		1.35.0.10
	fes_hil_5...	000C423...	Routerboar:3A:D2...	no	no	48Mbps/...	-71/-74		200.97.0.10
	gillam_rptr	Foster	UbiquitiNe:B6:51:39	no	no	11Mbps-S...	-43/-49		252.55.0.10

Above you can see another useful tab in Dude. This is the Wireless Registrations of your network. Dude is pulling data from every device that you can monitor Dude is pulling data on! And due to this, we have a bunch of data as it pertains to your wireless registrations. Here we have all of the wireless registrations from all the devices that you have listed! Along with their signal levels, data rates and IPs, you can add any comments you wish to make about on them. You can also double click on the wireless registration, and it will give you your registration information, just like if you were in RouterOS!

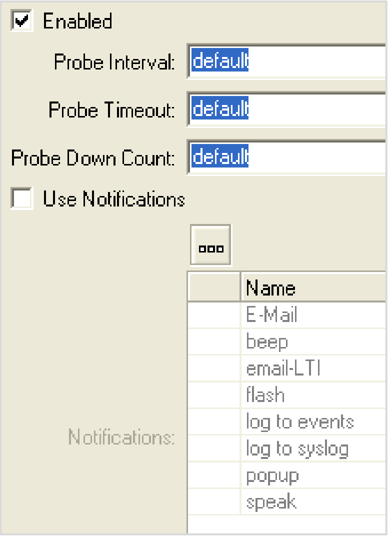


	Device	Name	Comment	Destination	Target	Rx Limit Max	Tx Limit Max	Rx Bytes	Tx Bytes	Rx Packets	Tx Packets	Rx Avg. Rate	Tx Avg. Rate
	WIFI Edge	2k_Xi...			10.0.12.100/32			2423.1 kB	29.3 MB	20610	27915	1.8 kbps	2.01 kbps
	WIFI Edge	2kw_b...			10.0.33.101/32	128 kbps	256 kbps	35.7 MB	35.2 MB	316790	117813	23.4 kbps	8.73 kbps
	WIFI Edge	2kw_b...			10.13.81.0/24	256 kbps	1 Mbps	65.3 MB	748.6 MB	437917	654468	7.44 kbps	1.65 kbps
	WIFI Edge	2kw_h...			10.13.82.254/32	128 kbps	512 kbps	24.1 MB	121.1 MB	232460	249893		
	WIFI Edge	2kw_s...			10.0.56.251/32	256 kbps	1 Mbps	12.8 MB	507.4 MB	203286	369344		

This image shows the Simple Queue tab under RouterOS. We can see queues, data rates, and limits setup for each of our RouterOS devices. What is even better is YOU CAN CHANGE THEM! If you double-click on a simple queue, you will update the simple queue on the device that you double clicked! How easy is that?

Device Options

If you double-click your device, you will get a big dialog box that will show lots of information. You will have options to set the device Name and IP address, as well as what type of device it is, if you should poll that IP address through an agent, what SNMP profile to use as well as username/password information for RouterOS devices.



☑ Enabled

Probe Interval: default

Probe Timeout: default

Probe Down Count: default

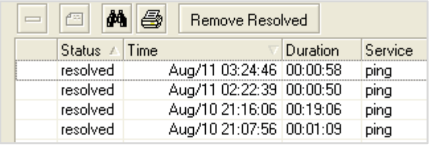
□ Use Notifications

Notifications:

Name
E-Mail
beep
email-LTI
flash
log to events
log to syslog
popup
speak

Under the Polling tab, you have options to specify how often to probe the services that you have selected on the next tab, as well as what notification options this device should use. By default, your polling options will be defaulted, this means whatever network map the device is on, it will inherit the polling settings of that map.

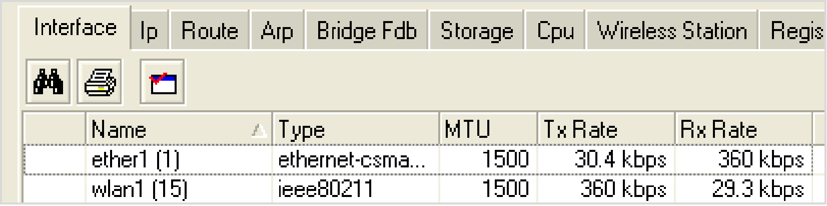
The Services tab is exactly what it sounds like; it is the services that the Dude will probe in an attempt to monitor the service. You can have multiple services on one device. If a single or multiple services are down, but at least one service is up, then the device will be considered partially down. Typically this means that the device will be a different color on your maps, for example, yellow instead of red, showing that the device is reachable just that some services are not responding. If all of the services are down, then the device will show as red, indicating all services have failed.



Status	Time	Duration	Service
resolved	Aug/11 03:24:46	00:00:58	ping
resolved	Aug/11 02:22:39	00:00:50	ping
resolved	Aug/10 21:16:06	00:19:06	ping
resolved	Aug/10 21:07:56	00:01:09	ping

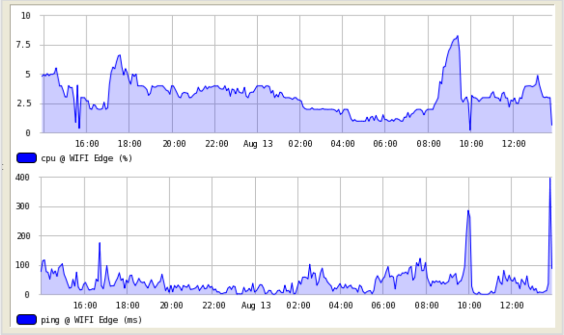
The Outages tab is a wonderful history of the outages that the Dude has reported on. What is really nice is that you can place notes on each outage so that you know the reason and why the outage occurred. Note that in the image we have the time/date and the duration of the outages just for this device.

Under the SNMP tab, you will have all of the information that the Dude has probed for, typically this can be quite a bit of data. In the image below this text, you will see IPs, Routes, ARP table entries, CPU usage, simple queues, and more. This data is nice to be able to see inside the Dude, however, it is mostly for show in the Device. The RouterOS tab is the exact same thing as the SNMP tab as well, but it may show more RouterOS-specific information including the packages, files and neighbors of the RouterOS device, but it is also information for you to view.



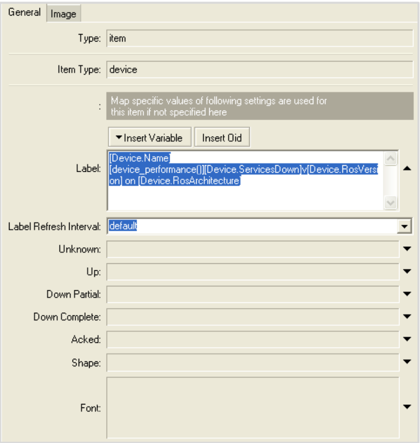
Name	Type	MTU	Tx Rate	Rx Rate
ether1 (1)	ethernet-csma...	1500	30.4 kbps	360 kbps
wlan1 (15)	ieee80211	1500	360 kbps	29.3 kbps

The History tab of the device can give you detailed history information and graphing of the services. Depending on what kind of services you are monitoring, it can present a number of different types of graphs. For example, for a device that is monitored for DNS, ping and CPU; the DNS and ping graphs will be response times. How long did the device take to ping, and how long did the device take to respond to a DNS query? The CPU graph though will be a % graph showing how much of the CPU has been used. Note in the image to the left that we have both response times in pings to this device as well as CPU usage in the top graph. Also, you can use your scroll wheel, if your mouse has one, and place the mouse cursor over the graph. By scrolling you can change from the past hour graphs to the past day, week, month and year.



Device Appearance

On top of all of the options you have in the device properties, you also have options about how the device appears. To get to these options, you can right-click on the device, and then select Appearance. Here, you will get a dialog box showing a bunch of options including a Label Name. This Label Name you can include SNMP OIDs as well as a number of variables. I have seen devices with the number of current registrations listed here. I have also set up these to monitor other types of access points and we included the channel and antenna polarization.



Type: item

Item Type: device

: Map specific values of following settings are used for this item if not specified here

Label: [Device.Name] [device_performance()][Device.ServicesDown]v[Device.RosVersion] on [Device.RosArchitecture]

Label Refresh Interval: default

Unknown:

Up:

Down Partial:

Down Complete:

Acked:

Shape:

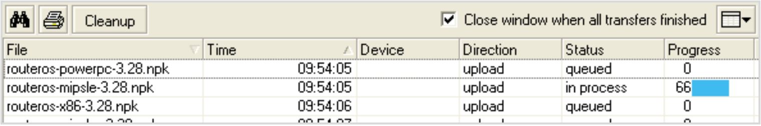
Font:

Files

The Dude system contains a file system with two different areas. The All section is for files such as images, and graphics for you to use with devices, and background maps, etc. The second section, Packages, is much more important. These are for is to be able to upload packages, (RouterOS packages) to be able to force devices on your network to do upgrades. There are two upgrade paths in Dude; one simply transfers the file to the RouterOS device, and the second not only transfers but reboots the unit to perform the upgrade as well.

Transferring Files within Dude

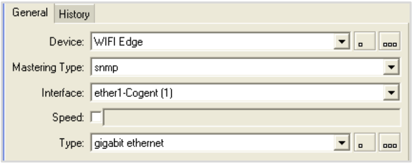
Uploading or downloading files to and from The Dude is a simple drag and drop action. Simply highlight the files you wish to move and drag and drop them into the Packages window. At that point the system will upload or download the file as necessary. You will also get the transfer window, pictured below showing files that need to be transferred and are in process, as well as what was completed.



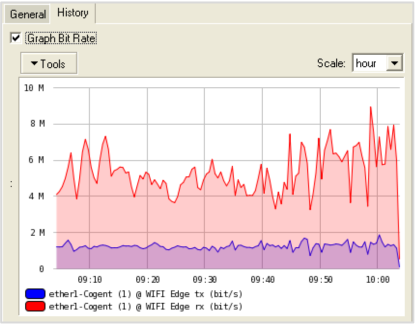
☑ Close window when all transfers finished

File	Time	Device	Direction	Status	Progress
routeros-powerpc-3.28.npk	09:54:05		upload	queued	0
routeros-mipsle-3.28.npk	09:54:05		upload	in process	66
routeros-x86-3.28.npk	09:54:06		upload	queued	0

Links



Device: WIFI Edge
Mastering Type: snmp
Interface: ether1-Cogent (1)
Speed: □
Type: gigabit ethernet

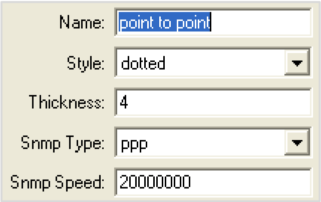


☑ Graph Bit Rate

🟦 ether1-Cogent (1) @ WIFI Edge tx (bit/s)
🟥 ether1-Cogent (1) @ WIFI Edge rx (bit/s)

The Links pane shows links that you created on your maps. You can double-click the links on this page, and view each one of your links just like if you were on your map. You can also the history of the link or make changes to the link.

Under the General tab you have the options to setup monitoring of your link. The monitoring and how you set this up is very important. If you set it up with a mastering type of RouterOS, you must have the proper username/password in the device to be able to monitor the interface. If you select the mastering type of RouterOS and no interfaces are listed, that simply means either the polling has not finished getting that information, or more likely the username and/or password is not correct to get that information. The other mastering type is SNMP. This is virtually universal for all types of links, and can monitor other devices not just RouterOS. Here, you will have to ensure that your system has SNMP turned On. RouterOS defaults to Off, and the community string is working as well. There is still a probe interval that occurs, but within a few minutes you should be able to see the interfaces.



Name: point to point
Style: dotted
Thickness: 4
Snmp Type: ppp
Snmp Speed: 20000000

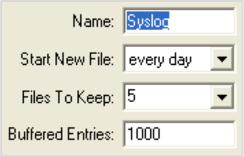
The Link Types tab, allows you to setup link types, and default speeds. If you have a number of identical links, this can be useful to setup and create the same types several times without retyping the same information. You can also set what type of line and the thickness of the lines that you wish to have.

Link Speed Setting

The reason for setting the link speed is that the Dude will take that link speed into consideration when it monitors the link. As the link grows in bandwidth usage, and approaches the link speed, the link, line and text will start to turn red to show that the link is approaching capacity. Something I do is to use this feature with backup links. Specifically, I will set the link speed to something very small, say a few hundred kbps. Something that I know if the link actually starts getting used, that link will immediately become red and be something that sticks out while glancing at the Dude to see that the link is being used. Normally there is an outage too, a radio down however, due to dynamic routing, it's possible for your radios not to be down, but just a link. Hence you need a method to determine that traffic is moving over a link that is not normally used.

Logs

The logging system of the Dude actually contains four logging systems. There is the Action log, Debug log, Event log, and if you are running, a Syslog. The Action log will list manual operations that are performed by an administrator. This could be you changing a link speed, or adding/editing a device. The debug log are changes that occur in the system, and the event log is network events, such as a device failing.

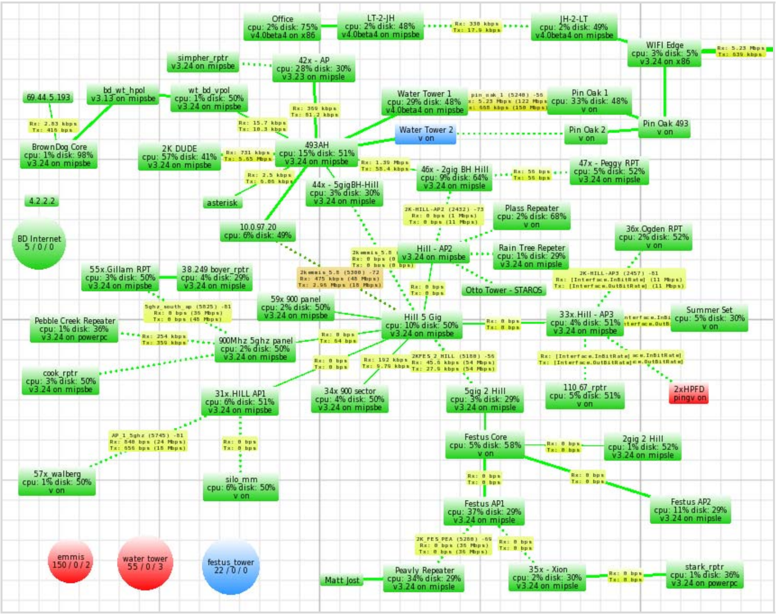


Name: Syslog
Start New File: every day
Files To Keep: 5
Buffered Entries: 1000

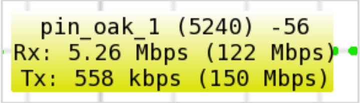
All of your logs have a Settings button that will allow you to setup how many buffered entries to keep, entries that are in memory but not committed to disk, how often to start new files and how many files to keep of back logs.

Network Maps

Network maps are at the heart of Dude. Many other NMS systems will list devices and their up/down status. Dude does this inside the Devices tab if that is all that you wish to have. However the real beauty of Dude is the ability to create a network map, with devices positioned as they really are on the network. You can add a map as an image behind your devices and actually lay out your network just like it is physically. This graphical representation of your network, along with link lines, bandwidth usages, CPU and Registration counts, can all be added to your network maps. This is the power of Dude, giving you the graphical layout of your network. As well as At-a-glance status indicators show, green for good links and devices and red for downed devices and overloaded links!



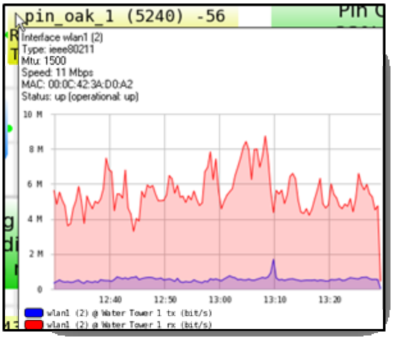
Above is a Dude map of one of the networks I operate. We have a repeater that is down on the right side. A few customers that were taken down during a storm show in red, but otherwise, all nodes are green!



```
pin_oak_1 (5240) -56
Rx: 5.26 Mbps (122 Mbps)
Tx: 558 kbps (150 Mbps)
```

You can create several Dude maps inside one system; each map can be linked to each other maps by creating submaps. They are only considered submaps if they are being linked from another map; otherwise, they are just maps. You can lay out these maps however you wish, and then draw links between each device. If you have SNMP or RouterOS devices, you can also place link information, such as bandwidth used, as well as other information like frequency, current signal and the air rate into the map.

You can also get the graphs for the link by simply putting your mouse over the link. This will give you a pop-up of the past hours' worth of traffic. This can be very useful if you are looking for traffic patterns, or sudden increases or decreases in bandwidth usage, on a link.

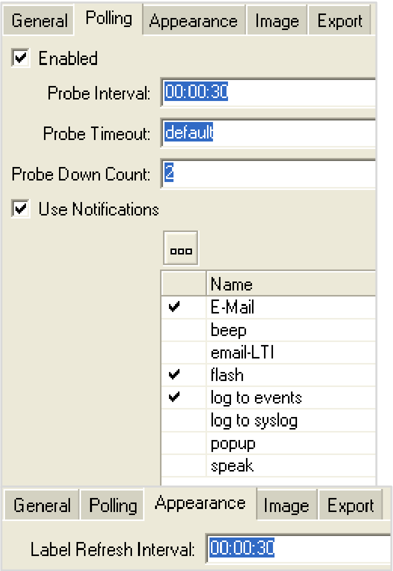


pin_oak_1 (5240) -56

Interface wlan1 (2)
Type: ieee80211
Mtu: 1500
Speed: 11 Mbps
MAC: 00:0C:42:3A:D0:A2
Status: up (operational: up)

Pin O
10 M
8 M
6 M
4 M
2 M
0
12:40
12:50
13:00
13:10
13:20
🟦 wlan1 (2) @ Water Tower 1 tx (bit/s)
🟥 wlan1 (2) @ Water Tower 1 rx (bit/s)

Map Settings



☑ Enabled

Probe Interval: 00:00:30

Probe Timeout: default

Probe Down Count: 2

☑ Use Notifications

	Name
✓	E-Mail
	beep
	email-LTI
✓	flash
✓	log to events
	log to syslog
	popup
	speak

Label Refresh Interval: 00:00:30

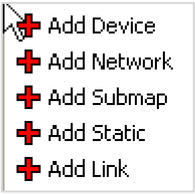
Each network map has its own Settings page. The Settings link is at the top of the Dude, over the right map pane. Inside the map Settings, we have the ability to setup defaults for the map here. The Polling section allows you to setup how often to probe the services, how long to wait till the probe times out and then how many timeouts can occur before considering the device to be down. You also have the options here to setup what notification settings you want all of your devices to default too.

The Appearance tab allows you to change the color, looks of the map, including the default colors, and background as well as how often to refresh the labels on the map. Depending on what information you have on the device labels, this may not make much difference; however, it may be important if you had CPU usage and disk information as well as other information listed on the label. The Image tab allows you to setup a background image for your map; you can scale it, and tile if you need too.

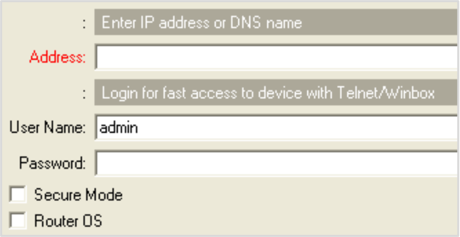
The Export tab allows you to export the map to an image file at specific intervals. You can select what kind of image type as well as at what intervals to export these files.

Adding Devices to your Maps

To add a device to your map, you can click the Plus sign in the map, or even simpler, just right click in a blank part of the map. When you do this you will get a menu option to create different objects. You can add devices, networks, submaps, static items and links. The ones I will use are submaps, links, and devices. In this example we are going to create a device. So click on Add Device and you will see the Add Device dialog box.



The Add Device dialog box, asks for the IP address of the device that you wish to monitor, as well as the username/password for RouterOS devices. You can also select to use the secure WinBox mode here. If it is a RouterOS device, check the box so that Dude knows this and will do the RouterOS probing. Once completed, click the Next button to proceed.



: Enter IP address or DNS name

Address:

: Login for fast access to device with Telnet/Winbox

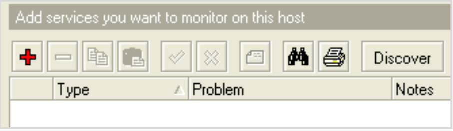
User Name: admin

Password:

□ Secure Mode

□ Router OS

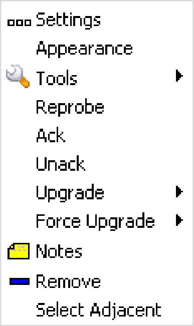
The next box is for the services that you wish to monitor. Remember, in the discovery section I suggested that you only check the services that you would possibly wish to monitor, such as PING or DNS. Here, we have an option for discovery, the Discover button. When you click this, Dude will perform probes on the IP address that you entered in on the first screen and try to detect the services that are running at the IP address in question. This is very useful if you only have only a few services that you wish to monitor. You can however click on the plus sign and add individual services that you may wish to use.



Add services you want to monitor on this host

Once you have finished this, now you should have a device listed on your network map with the name of the IP address that you entered in on the Add Device dialog. If you double click the device window you will have options to give this a more meaningful name, as well as other options, including the abilities to change the agent , SNMP profile, username/password as well as notification, services, and other historical information.

Working with Devices



Once you have created devices, there are a number of tools and options that you can use to help manage your device. By placing your mouse over the device and right-clicking, you will get a context menu. This menu gives your device settings, (the same as double-left-clicking), the appearance options, (discussed in the Devices section), as well as other tools. The Tools menu is the extension of the tools pane on the left side. By selecting the tools, it will pop-out another list of tools. By clicking on WinBox, your WinBox application will automatically use the username and password in the device along with the IP to login to your RouterOS system. If you have other tools, such as MSTSC or pathping, you can also access them there as well.

The Reprobe command tells The Dude to issue a reprobe on the services that are on the device. If you device shows down, but your probe interval is two minutes, you can reprobe the device to see if it is back up and running quickly. Once you have a down device, you may know that it will be down for some time. Inside your notifications section, you may have reoccurring notifications, sending out that this device is down every 30 minutes. By you ACKing the device, it will turn the device blue. No more probe requests or checks will be done on the device until it is unacked, but it also will not send out any more notifications. The idea is that you are acking or acknowledging that the device is down. When you unack it, it tells the system that you now wish it to start the checks again, most likely you have fixed the issue, therefore it will turn green!

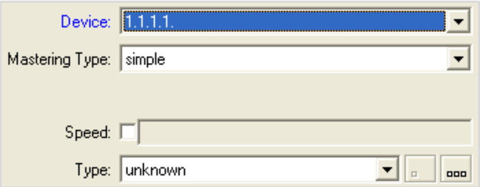
Upgrades

The Dude offers two ways to upgrade your RouterOS systems. One is a forced upgrade and another is just an upgrade. The forced upgrade not only transfers the file to the RouterOS system, but also performs the upgrade by rebooting the RouterOS system once the file has been uploaded. The standard upgrade method does not reboot the router, but does transfer the file. To access this menu item, right click on your device in Dude and then simply select upgrade or force upgrade, and then the appropriate version that you wish to upgrade too. Remember you will have to upload the NPK files to your Dude server. Once uploaded, as long as you have the appropriate CPU versions uploaded, they should appear in your Upgrade Context menu.

Creating Links

Links as described in the link section can be used to show bandwidth usage and stats on a link between two devices. To create a link, right-click on your network maps, and then select Add Link. Next, click and HOLD on one of the two devices you wish to create a link from and to, and drag your mouse from the first device to the second device, releasing once you get to the second device. This will create a link!

Upon creating that link you will see an Add Link dialog box appear. This is for the mastering information about the link. If this is a RouterOS link and you have SNMP turned on, you can get SNMP data right away. We discuss the mastering types, link speed and types in the links section further.



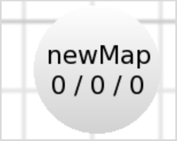
Device: 1.1.1.1

Mastering Type: simple

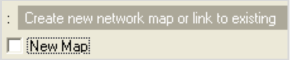
Speed: □

Type: unknown

Creating and Linking to Submaps

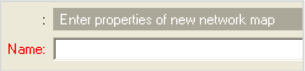


Remember that all of your maps are submaps, however when you have a single map, you will need to create a second map to link to. This is done very simply by starting to link to a submap; fortunately, as there is a simple option to help you create the submap. To start, just like when you created devices and links, we will need to right click on the background of the current map. This provides us the option to add a submap. The very first option it asks if this is going to be a new map. If so, check it, if not click Next. Remember, though you have to have more than one map to be able to link to the second, so I typically just create my submaps by creating new maps to link to.



: Create new network map or link to existing

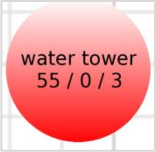
□ New Map



: Enter properties of new network map

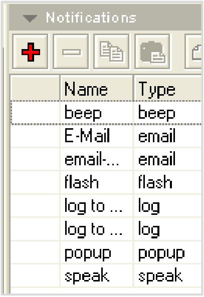
Name:

Once you click Next, if you are creating a new map, it will ask you for the name of the map. If you are not creating a new map, you will have a dropdown of the existing maps to choose from. As you can see, we have created a new submap, and that circle is now clickable. If you double-click, Dude will automatically open up the map that you just created. The new map that you just created also will have a submap already in it to return you to the map that you linked in from!



You may notice that there are numbers in some of my maps. As you add devices, your maps will change colors and give you numbers indicating, the total number of devices on the map, the number of partially down devices and the number of down devices on that map. Since the Water Tower map has three devices down, note that it's not green, but red.

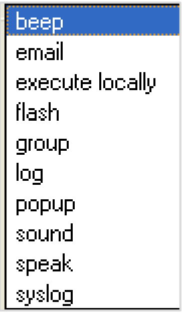
Notifications



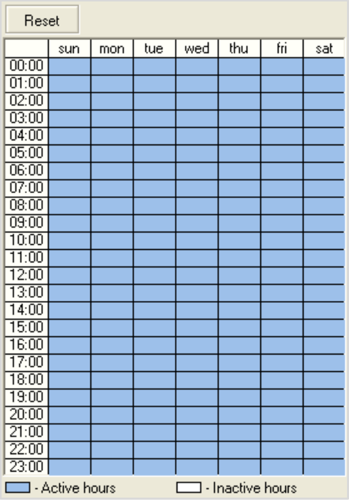
Name	Type
beep	beep
E-Mail	email
email-...	email
flash	flash
log to ...	log
log to ...	log
popup	popup
speak	speak

The Dude has a number of capabilities when it comes to Notifications. First, I wanted to go over the places where you configure Notifications. This is where you can set what notifications go with what device or devices. Initially out, there are map defaults, so by default, anything on your map unless otherwise changed, would have the Notifications that you place on them. Second, you have each and every device, you can setup towers in town A to only contact the tower climber in town A and the climber in town B to get notifications from tower B failures. This allows you to really custom tailor your notifications to whom and what you wish to. You can also setup times, so on Saturdays this person may get a page or text message and on Sundays another person may get them. On top of all of that, you still have the server defaults to setup. Inside your server configuration you can also setup server default as well, so there are a number of places to setup notifications.

Dude works by allowing you to configure different notification names with different notification types. An example would be the two tower climbers in two towns I talked about in the above paragraph. One notification name may be Tower A and one may be Tower B. Tower A would have the e-mail address of the tower climber in town A, and Tower B would have the e-mail address of the climber in town B. You can also create groups of notifications, so you can place items with names of your techs, say, Bob and Jim. By then using the groups, you can choose to notify a group of people.

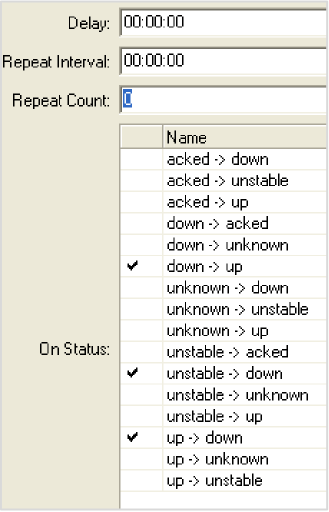


A number of built-in notification types are included. The one that is most commonly used is e-mail. With e-mail delivered and pushed right to your phones, it's hard not to simply use your PDA as a notification device. However, most phones that are not PDAs also get SMS or text messaging. In the US (and I would assume other places), most wireless companies provide an e-mail-to-text gateway, typically your phone number at their domain. Simply sending an e-mail with a short subject will be delivered quickly as a text message.



	sun	mon	tue	wed	thu	fri	sat
00:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
01:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
02:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
03:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
04:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
05:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
06:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
07:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
08:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
09:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
10:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
11:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
12:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
13:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
14:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
15:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
16:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
17:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
18:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
19:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
20:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
21:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
22:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦
23:00	🟦	🟦	🟦	🟦	🟦	🟦	🟦

🟦 - Active hours □ - Inactive hours



Delay: 00:00:00

Repeat Interval: 00:00:00

Repeat Count: 0

On Status:

	Name
	acked -> down
	acked -> unstable
	acked -> up
	down -> acked
	down -> unknown
✓	down -> up
	unknown -> down
	unknown -> unstable
	unknown -> up
	unstable -> acked
✓	unstable -> down
	unstable -> unknown
	unstable -> up
✓	up -> down
	up -> unknown
	up -> unstable

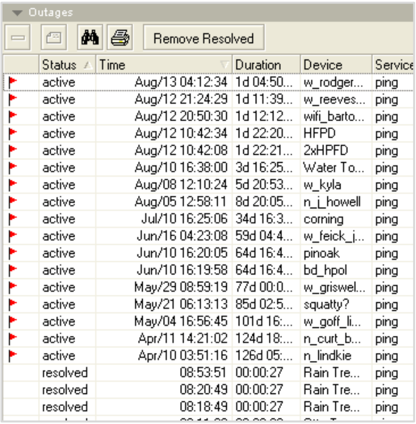
The Dude does have the ability though to simply log, beep, flash the device that went down, provide a pop-up notification window, as well as send data to a Syslog server. All of these are simple and easy to use, but the fun ones are the sounds effects. Dude offers two of them. Once is just a simple WAV file that it will plays. As you start creating more notifications, you can have different sound effects. A customer of mine uses this and the Dude PC hooked to their overhead paging system. If they hear a specific sound effect they know exactly what area and what tower has an issue, while another tower or location, would have its own sound effect. The second sound that Dude can make is actual speech! Yes; Dude can speak to you though a text-to-speech engine. This engine is typically part of your OS, so if your OS doesn't support it, then you may have an issue, however most Windows systems will have this capability. Under the speak type, you can have it say, "Alert, this device is now down!" You can include variables like the probe in the device name in this to be able to more easily identify what device is down!

Inside each of your notification types, you also have a notification schedule. This schedule will help you turn on and off the notifications, so that only during specific days and times will specific notifications work. This can be good or bad, so make sure you have the proper people that need to be notified in the active hour's schedule.

Last is the Advanced tab. This tab will allow you to input a delay. This delay is how long to wait for the device to come back up before sending out the notification. This may be something that is useful if you have a link or other device that is sometimes unplugged or otherwise the device is prone to going down for reasons outside of your control. Normally, as a network engineer you would want to fix this, gluing the power plug in might be simple enough. However, when it comes to home users, you may wish to wait five or ten minutes before you start performing notifications that a $20 customer is down.

The Repeat Interval and Repeat Count is a valuable tool. These will allow you to resend the notification, if the device is still down, every so often, as well as configure and how many times it should send this. Sometimes people forget that a device is down as they might be working on another issue. If we keep alerting them, they will be reminded that the outage is still there.

Outages

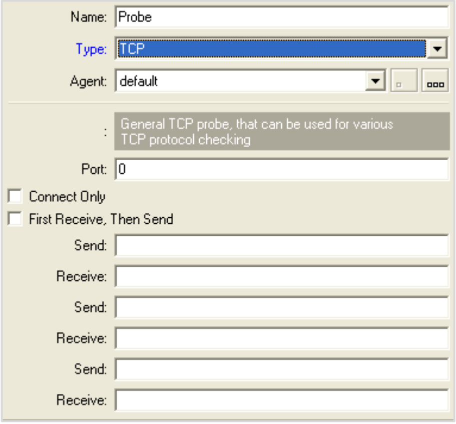


	Status	Time	Duration	Device	Service
⚑	active	Aug/13 04:12:34	1d 04:50...	w_rodger...	ping
⚑	active	Aug/12 21:24:29	1d 11:39...	w_reeves...	ping
⚑	active	Aug/12 20:50:30	1d 12:12...	wifi_barto...	ping
⚑	active	Aug/12 10:42:34	1d 22:20...	HFPD	ping
⚑	active	Aug/12 10:42:08	1d 22:21...	2xHPFD	ping
⚑	active	Aug/10 16:38:00	3d 16:25...	Water To...	ping
⚑	active	Aug/08 12:10:24	5d 20:53...	w_kyla	ping
⚑	active	Aug/05 12:58:11	8d 20:05...	n_j_howell	ping
⚑	active	Jul/10 16:25:06	34d 16:3...	corning	ping
⚑	active	Jun/16 04:23:08	59d 04:4...	w_feick_j...	ping
⚑	active	Jun/10 16:20:05	64d 16:4...	pinoak	ping
⚑	active	Jun/10 16:19:58	64d 16:4...	bd_hpol	ping
⚑	active	May/29 08:59:19	77d 00:0...	w_griswel...	ping
⚑	active	May/21 06:13:13	85d 02:5...	squatty?	ping
⚑	active	May/04 16:56:45	101d 16:...	w_goff_li...	ping
⚑	active	Apr/11 14:21:02	124d 18:...	n_curt_b...	ping
⚑	active	Apr/10 03:51:16	126d 05:...	n_lindkie	ping
	resolved	08:53:51	00:00:27	Rain Tre...	ping
	resolved	08:20:49	00:00:27	Rain Tre...	ping
	resolved	08:18:49	00:00:27	Rain Tre...	ping

The Outages pane will show you your current outages, when they started and how long they have been active. In Dudes web interface, this gives some people a good place to start where they can see a list of outages that need to be addressed. You can also add notes to each of these outages. These correspond to the Outages tabs in each individual device, except the outages pane will list all outages across your Dude system. You can also use three different drop downs on the upper right side to filter these, including only active, only pings and then also have the ability to only watch a specific map.

Probes

Probes are functions that the Dude does to check if services are up and running. Common functions have been configured for you however; you may wish to modify them. The basic probes that I will cover here are the TCP/UDP and SNMP probes. The other types, such as functions, are really outside the scope of this book. These require programming logic, functions and if then equals that are simply are more complicated than what most people wish to accomplish.



Name: Probe
Type: TCP
Agent: default

General TCP probe, that can be used for various TCP protocol checking

Port: 0

□ Connect Only
□ First Receive, Then Send

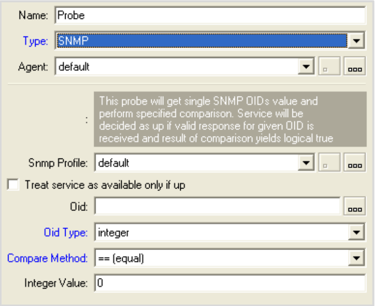
Send:
Receive:
Send:
Receive:
Send:
Receive:

Creating TCP/UDP Probes

We will start with TCP and UDP probes. Simply put, these perform an action by opening the specific TCP or UDP port for communication. You specify the port number and you can have it only attempt to open that port and obtain a connection. That is what the Connect Only check box is for. If the service on the port specified sends data to you as a client first, then you will need to check the box to First Receive, Then Send first and then send. The idea here is that you can carry on a conversation with the program on that port, to the extent that you know it is running. In most cases, simply connecting is fine; however, some people wish to actually issue a function. An example would be to issue a normally valid command to a SMTP server. If that command fails then typically there is an issue with the server. Both TCP and UDP settings are virtually identical.

SNMP Probes

SNMP probes are very simple. The probe does an SNMP request to your device. The probe contains an OID value; this is a single item inside the SNMP table. Then your device will respond with the value for the OID. Once that value is returned, based on the other settings in your probe, that value will be compared to the integer value you entered. There are many different ways to compare that value to the integer value you entered, and based on that, the probe will return either an up or down status.



Name: Probe

Type: SNMP

Agent: default

: This probe will get single SNMP OIDs value and perform specified comparison. Service will be decided as up if valid response for given OID is received and result of comparison yields logical true

Snmp Profile: default

□ Treat service as available only if up

Oid:

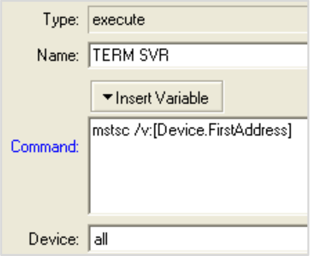
Oid Type: integer

Compare Method: == (equal)

Integer Value: 0

Tools

The Tools pane allows you to add and control tools that you can access by right clicking the device. There are a number of built-in tools, including WinBox, telnet, and snmpwalk however; one tool that I have found useful is MSTSC, or terminal services. I do use Dude to monitor windows servers and having the ability to right click on the device and term serv right into the server makes it very simple. MSTSC uses a command line of *MSTC /v:address.* It is very simple to build this tool. I click the plus, to add a new tool, and then give it a name. Now I simply enter the command line, along with the address variable.

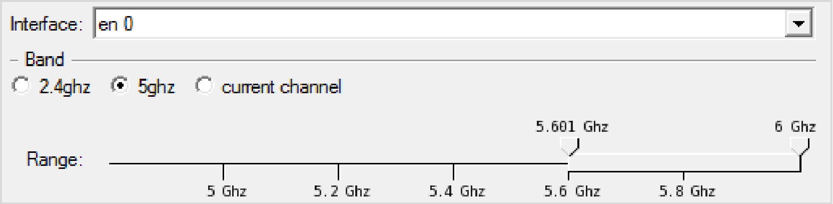


Type: execute
Name: TERM SVR
Command: mstsc /v:[Device.FirstAddress]
Device: all

If you wanted to build a SSH tool, simply add another tool, name it, and then make sure the SSH application is in your path, or you will need to specify that path. In my case, I use putty. So, my command line would be very simple: putty *address.* That's it. If you have other tools you can enter them here. Whatever you think would help you, and you can place in here, assuming that there is a command line interface for it.

Spectrum Scan

A new feature of Dude v3.6 is the spectrum scanner. This is an graphically-improved version of the included spectral scanner introduced in v4.3 on R52N and R2N radio cards. The feature, in v4.3+ creates a spectral snapshot from the radio card and displays in the command line view of RouterOS. Even though this view is good, MikroTik took this and improved upon it inside Dude. Inside here, you can simply select the band you wish to scan, hold times, sample times, interface and device right from a simple graphic interface.



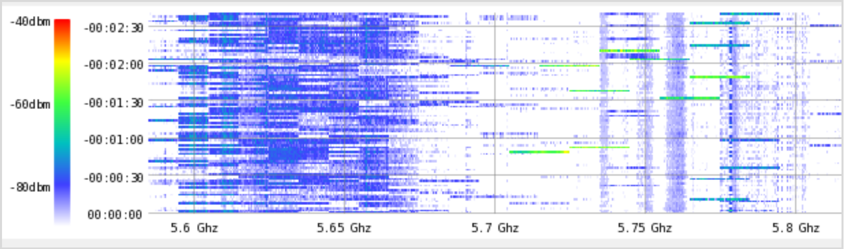
Interface: en 0

Band

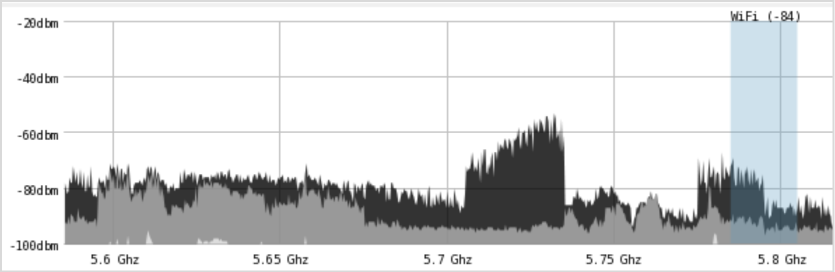
○ 2.4ghz ● 5ghz ○ current channel

Range:

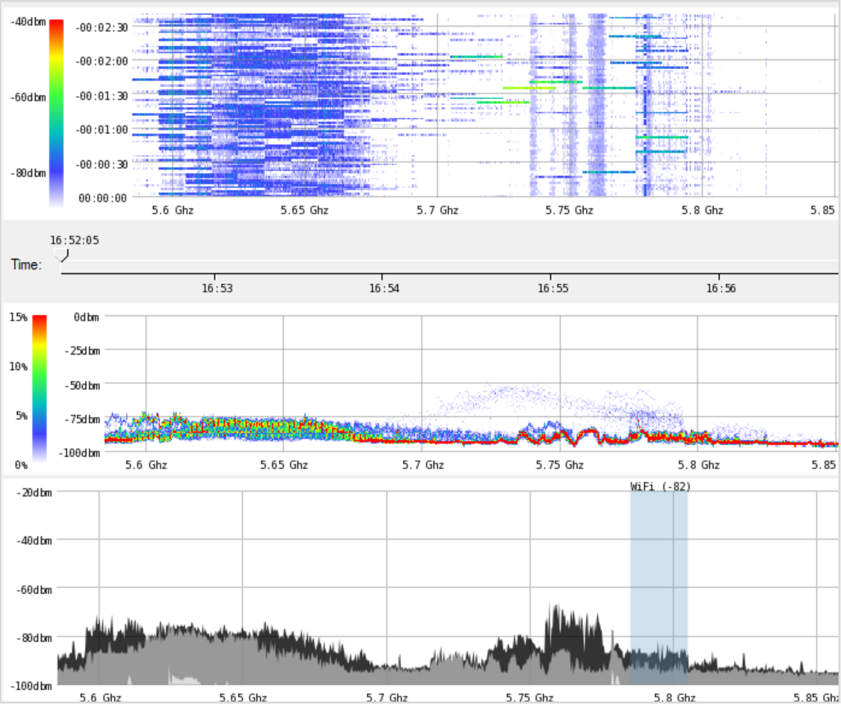
As you can see above, you can select the band, and frequency range that you wish to use. You get three different graphical images of the spectrum that you are viewing; below is the waterfall display.



You also get the actual graph showing the 30-second peak, current maximum, and current average.



Notice above, the scanner also selects what it considers and can identify as a Wi-Fi signal. You also can use the density graph as well, or show all three at the same time if you wish.



User Manager

Recently MikroTik has developed a system called User Manager. The main purpose for this was to eliminate the bulky and slow database system inside RouterOS and provide a fast and efficient way to run large user databases. As time went on, User Manager grew to a much slicker system. The common use for User Manager now is as a Radius server and as a, user management and payment gateway for hotspot systems. Even though it could also be used for other Radius purposes, this is the most common usage that I use User Manager for. On top of that, the User Manager system comes with your RouterOS license!

Using User Manager as a hotspot gateway allows users to create a user account, pass through some type of payment gateway, and then come back and uses their username/password that they created to login. Typically this is used in a hotspot environment allowing users to pay for Internet access time and to get on the Internet without administration intervention or action. There are other systems out there, but, for the cost you can't go wrong using the User Manager system. Did I mention it's FREE?

Hardware / License Requirements

User Manager has to run on a RouterOS system, so you have to have some form of license. It is important to note here that there are license restrictions to the number of users that User Manager will allow you to run.

License Level	**3**	**4**	**5**	**6**
Number of Active Users	10 Users	20 Users	50 Users	Unlimited

For installations I do with RouterOS and User Manager as a hotspot payment gateway, I will use a Level 6 license. Other factors though also play into the hardware that I select for the installation of User Manager. The minimum hardware that I will use is a RouterBOARD 433AH. There are some reasons for this. The first, one is RAM; you need at least 32 MB of RAM for User

Manager and the second reason is disk space. Even though the 493AH has a sizable NAND, I prefer to be able to use external storage, so I go with the 433AH board with an add-on 2 GB Micro-SD card. If you are interested in the exact hardware I use for User Manager Installations with under 200 active users, I would suggest visiting my homepage at http://www.linktechs.net. There you will find the PowerSpot 400 system. This is a completed 433AH with the Micro-SD Card installed and formatted, User Manager installed on the Micro-SD card, and a Level 6 RouterOS license installed.

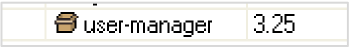
Once you go over the 200-300 user mark, I would suggest going with a RouterBOARD 1000 or PowerRouter 732 to ensure that you have fast user lookups. Having fast response times is critical in some cases, so make sure you are not over tasking your hardware. I also typically do not put both the User Manager software running Radius and a number of users on a 433AH along with having that 433AH performing my routing, hotspot server, etc. If it is just for a single site with a few Mbps of throughput, then this may be fine as long as the number of users do not get higher than 50 or so active at one time.

Reference Version of User Manager

The first edition of this book, v3 of User Manager was used. Many things have already changed to increase your options and customizability in v4beta though-out v5. The following sections are based on v3 of the user manager, as this is still stable and working. After the major User Manager Sections, a new v5 User Manager section has been created with the changes and updates that v5 offers.

Installation of User Manager

Installing User Manger is as simple as adding another package to RouterOS. I would ask you to refer to the package installation procedures in this book to understand how to do this. Simply put though, drag and drop the User Manager .npk file into your RouterOS system, and reboot your router. Upon rebooting, you should see the User Manager package installed.

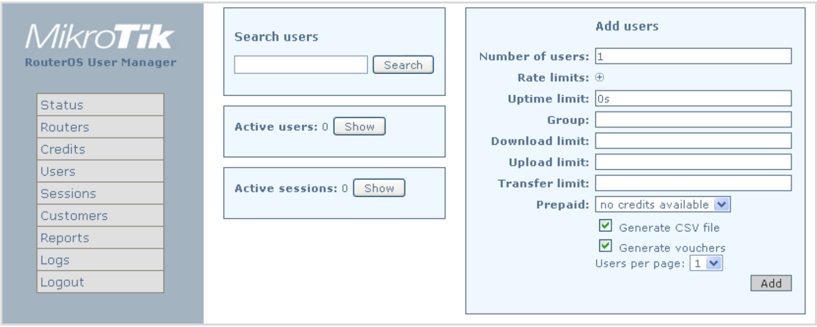


user-manager 3.25

Configuration of User Manager

First Time Access

To access your User Manager system for the first time you will need to verify that your WWW service on your RouterOS system is running. You will also need to verify the port. The default in RouterOS is port 80, and by default, the service is enabled. If this has not been modified, then you can simply open your web browser to http://*ipaddress*/userman for the admin interface. The default username/password is admin and admin.



Active users: 0

Active sessions: 0

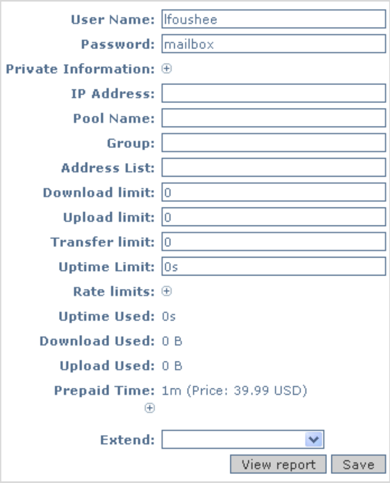
Number of users: 1
Rate limits: ⊕
Uptime limit: 0s
Group:
Download limit:
Upload limit:
Transfer limit:
Prepaid: no credits available
☑ Generate CSV file
☑ Generate vouchers
Users per page: 1

There are a few things to note on here; the /userman page is meant for managing your User Manager system. There is a user level access page at /user. This would allow users to access their accounts, add time, make a payment, and so on etc. Also, most users are used to not using a port number, so in many cases, I would leave your RouterOS WWW service on port 80 so that users as well as you have simple access to the management pages.

Understanding Concepts and Definitions

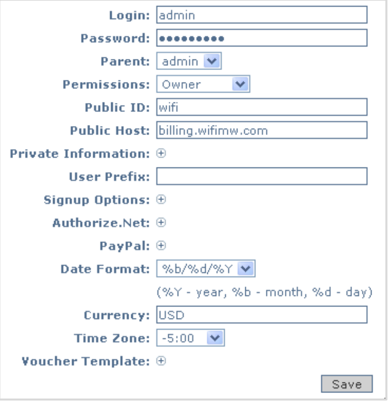
I prefer to explain this a bit differently than MikroTik does. Without understanding some of this, you may get lost, so read carefully!

Users are your end users, the people who create an account, pay for time and use the username/password to gain access. They typically will never use the /userman interface, as they have their own /user interface with User Man. In the Users section of the admin interface, here you can setup usernames/passwords, add them to a pool, or group, setup limits as well as see what prepaid time they have purchased.



User Name:	lfoushee
Password:	mailbox
Private Information:	⊕
IP Address:	
Pool Name:	
Group:	
Address List:	
Download limit:	0
Upload limit:	0
Transfer limit:	0
Uptime Limit:	0s
Rate limits:	⊕
Uptime Used:	0s
Download Used:	0 B
Upload Used:	0 B
Prepaid Time:	1m (Price: 39.99 USD) ⊕
Extend:	

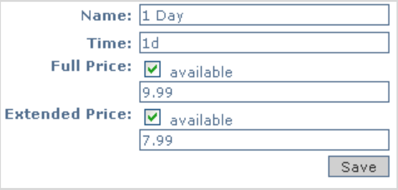
Customers are people who are selling services. I or you would be considered a customer as we sell Internet services to users. You can have levels of customers that share the same packages and routers. Inside the customer's configuration, you will have information such as the signup Options, the authorize.net and PayPal information, as well as the currency and time zone information as well.



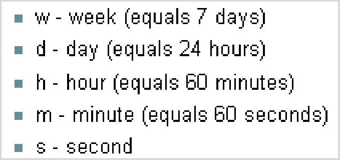
Login:	admin
Password:	●●●●●●●●●
Parent:	admin
Permissions:	Owner
Public ID:	wifi
Public Host:	billing.wifimw.com
Private Information:	⊕
User Prefix:	
Signup Options:	⊕
Authorize.Net:	⊕
PayPal:	⊕
Date Format:	%b/%d/%Y (%Y - year, %b - month, %d - day)
Currency:	USD
Time Zone:	-5:00
Voucher Template:	⊕

Subscribers are customers. The difference is they are the "top" customer. They have their own authorize.net, account, their own routers, and their own pricing. To set a customer to be considered a subscriber, you set the customer's account to have a parent of itself. That's it, there is nothing different. Subscribers are nothing more than a customer that has itself set as a parent. Note in the image above, the customer name is admin, and the parent is admin.

Credits are time plans. Each credit belongs to a subscriber. If you have multiple subscribers in the system, each of them can have their own credits. Inside your credits you have the ability to say how much time they get for a specific price. Also note in there that the constants that RouterOS uses are not the same; for example, there is no "m" constant for month in User Manager. The image to the right shows the constants that User Manager uses.

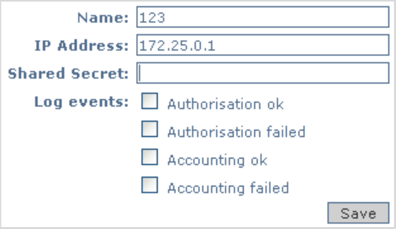


Name: 1 Day
Time: 1d
Full Price: ☑ available
9.99
Extended Price: ☑ available
7.99



- w - week (equals 7 days)
- d - day (equals 24 hours)
- h - hour (equals 60 minutes)
- m - minute (equals 60 seconds)
- s - second

Routers are devices that will make a Radius query against the User Manager database. They are Radius clients where the User Manager system is the Radius server. Inside here you have a few options. The name of the router, the IP that the client request is going to come from, the shared secret as well as the logging options that you wish to have enabled. Note that when you have an active system, logging every login etc, can take up considerable disk space.



Name: 123
IP Address: 172.25.0.1
Shared Secret:
Log events: □ Authorisation ok
□ Authorisation failed
□ Accounting ok
□ Accounting failed

NOTE: At the time of writing this book, v3 of User Manager does not allow wildcards, or subnets in the IP address field. You must have the exact IP address for this to work. If you wish to get around this, create a PPTP tunnel and setup policy-based routing on your remote site to use the tunnel for your requests. Regardless of the public IP that your Radius client has, it will always have the same tunnel IP.

Basic Configuration Settings

Many people get nervous because of all of the settings with User Manager. In this section we will setup a base system quickly and effectively for a business to sell Internet access from a hotspot.

RouterOS Settings!

Yes, before you begin, of course your RouterOS system has to have an Internet connection, but more importantly, we have to configure your RouterOS system to ensure that User Manager works! Yes, there is configuration inside RouterOS that has to be done or your User Manager system WILL NOT WORK. These are the requirements:

- Internet access from RouterOS
- Correctly Set Date/Time and Time Zone
    a. You will need to use a NTP client if you are on a RouterBOARD product, as they don't keep their time upon a reboot
- Correctly configure E-Mail tool
    a. This is to send out the e-mail notifications
- If using Authorize.net, a SSL installed.

That's it, there are not many requirements, but they are the requirements needed to make the system work. First off, could we not run without a properly configured e-mail tool? Nope, because User Manager sends out the e-mails and needs configuration to accomplish this. Then what is the clock for? When sending a request to either Authorize.net or PayPal, the system generates a hash based on the time and date to secure the communication between the User Manager system and the payment gateway. If the payment gateway receives data that is from 1970 (the default date on RouterBOARDs), the system will reject it as bad data, and you will never get a card to process!

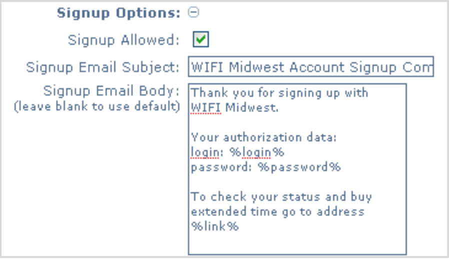
The SSL portion is a configuration requirement; you must have installed a SSL certificate on your RouterOS system. This is to setup authorize.net information, as that information is your transaction key and API login information.

User Manager Settings

So to start, you will need to login to your User Manager system. Your base user 'admin' is already a subscriber, so we will simply use that user to create everything. The first task is to configure the payment gateway information, and sign-up information as well as the admin user with all of the proper settings.

Configuration of the First Subscriber

To configure the admin customer, click Customers and then click View. This will show all of installed customers. Click on the Admin User to pull up the customer data. Here we will setup a password to secure our configuration settings. Next I suggest creating a public ID. Typically this is only used when you have more than one subscriber; however, I like to configure it as well. This can be a simple piece of text; in many cases I will just use Wi-Fi as the public ID. The public host information is more important; this is the return IP or URL that your payment processor will return result data to, so it has to be either a public IP or a valid URL on the Internet. The User Prefix option is for when if you want to have multiple subscribers on the system. Each user would get a prefix to identify what subscriber they belong too.



Signup Options: ⊖

Signup Allowed: ☑

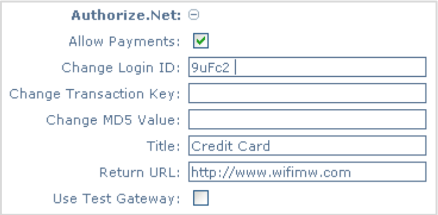
Signup Email Subject: WIFI Midwest Account Signup Con

Signup Email Body: (leave blank to use default)

```
Thank you for signing up with
WIFI Midwest.

Your authorization data:
login: %login%
password: %password%

To check your status and buy
extended time go to address
%link%
```

Next we will fill out the Private Information section. These fields are not required, but you can fill them out if you wish. The Signup-up Options configuration is next.

The Signup-Up Options is a drop down, so you will need to click the plus sign. This allows users to sign up for service themselves, so you will wish to check this box to allow signups. Also here, you can go ahead and configure your Signup-Up E-mail as necessary.



Authorize.Net: ⊖

Allow Payments: ☑

Change Login ID: 9uFc2

Change Transaction Key:

Change MD5 Value:

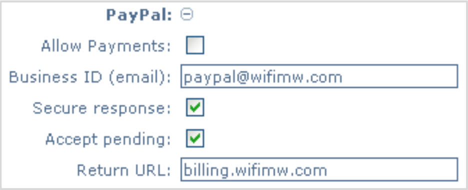
Title: Credit Card

Return URL: http://www.wifimw.com

Use Test Gateway: □

Authorize.net is an on-line credit card processing system. To access this section, User Manager will require you to be in HTTPS mode. If you have not logged in via https://*ipaddress*/userman, now you will have to do so now otherwise you cannot expand the authorize.net section. If you are logged in, then we can continue to do the authorize.net setup. Once you do get your dropdown, you will be able to enter your transaction key, login ID and MD5 value in that you received or created with Authorize.net. The title field is what the user will see as a payment method, and starting in v3.24 you should also have the return URL field. This is what webpage to return the user to after a payment has been processed. Typically you would set this to something that the user would not have access to until they login. Upon finishing the payment they are taken back to User Manager with the information from your payment gateway showing that they paid. Then they get directed to the return URL, showing them the login screen so that they can login.

The PayPal method is a bit simpler, as you do not have to be in HTTPS to access the PayPal configuration. It's very simple; enter what is the PayPal payment address, or e-mail. Fill in the appropriate Allow Payments, Secure response and Accept pending checkboxes. Should you allow PayPal payments, do you require a secure response and/or accept pending payments. The Return URL is the same as the Authroize.net system.



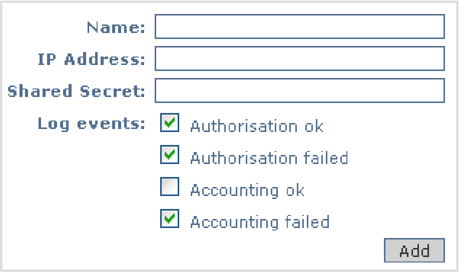
PayPal: ⊖
Allow Payments: □
Business ID (email): paypal@wifimw.com
Secure response: ☑
Accept pending: ☑
Return URL: billing.wifimw.com

In regards to what system do I prefer; I think the authorize.net system is more business oriented, and is simply more professional. I also have trouble with PayPal as they have in the past changed their system and it required an update for User Manager to use PayPal again.

The final section is your time zone and currency. User Manager uses a three digit currency code, so for the US you would use "USD". I would also suggest setting your time zone here as well.

Configuration of your Routers

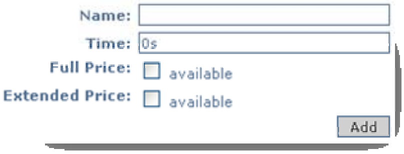
Next we will configure your routers. Since this is a small system, we will just do one. Click on Routers, and you should get another menu to either View or Add. Select Add. Enter the name of the router you wish to add, the IP address and the shared secret. Enable whatever logging you wish to have from that site. On the IP address, remember this is the forward facing interface towards your User Manager System. You also CANNOT use subnets or IP address ranges in here, it must be exact.



Name:
IP Address:
Shared Secret:
Log events: ☑ Authorisation ok
☑ Authorisation failed
☐ Accounting ok
☑ Accounting failed

Configuration of Credits

Credits say how much time the end user receives if they pay xx amount. Remember that in the Time field, there is no “m” for month, so if you wish to give a month access you will need to use 4w, for four weeks. The full price is the price that the user will pay upon creating their initial user account, and the extended price is the price to add time to their existing account. If you wish you could give existing users a discount.



Name:
Time: 0s
Full Price: ☐ available
Extended Price: ☐ available

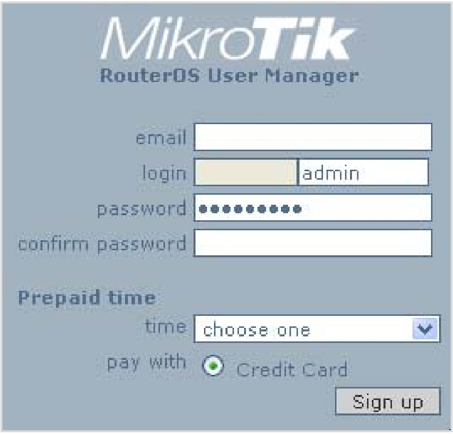
Now that you have all of the necessary information inside your User Manager, you should be able to have a user get to the sign-up page and signup for an account, pay on-line, and then come back to sign in and use the Internet!

User Sign-Ups

For users to sign up for service, they will need to follow a link from your splash page to get them to create an account. The signup link is as follows:

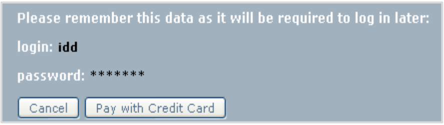
http://*urlorIPofUserManager/*user?signup=*publicID*

When users click the sign-up link from your splash page, this is where they should be taken. Remember, that you will need to allow this URL and/or IP in your walled garden. This page will allow your users to enter their e-mail address, create a new login and password, and select how much prepaid time they wish. Since this system has authorize.net configured, they will pay with a credit card.



login: admin
password: ●●●●●●●●●
time: choose one
pay with: ● Credit Card

Next they will click the sign-up button. This will take the user to a page that will remind them to remember their username and password and a button to pay with credit card. By the user clicking this button, User Manager delivers the customer to Authorize.net for payment. User Manger does not process or store credit card information. It passes them off to the respective websites for your payment processor, and they process and take the credit cards over secure HTTPS sites. There is typically no need for you to have your own SSL as you never take personal information.



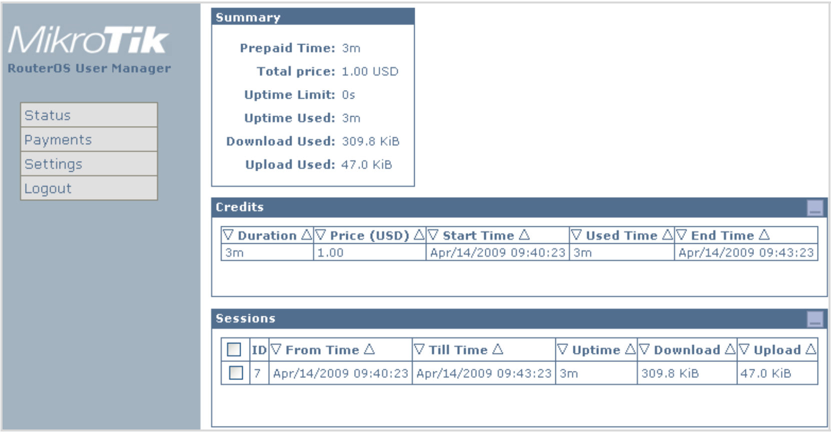
Please remember this data as it will be required to log in later:

login: idd

password: *******

User Sign-In Page

The users also have a page where they can sign in, and update their account, and add more time. This page is http://*ipaddress*/user.



Summary

Prepaid Time: 3m
Total price: 1.00 USD
Uptime Limit: 0s
Uptime Used: 3m
Download Used: 309.8 KiB
Upload Used: 47.0 KiB

Credits

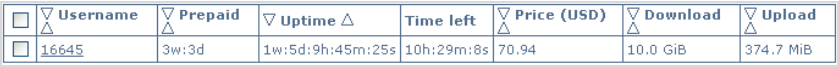
▽ Duration △	▽ Price (USD) △	▽ Start Time △	▽ Used Time △	▽ End Time △
3m	1.00	Apr/14/2009 09:40:23	3m	Apr/14/2009 09:43:23

Sessions

□	ID	▽ From Time △	▽ Till Time △	▽ Uptime △	▽ Download △	▽ Upload △
□	7	Apr/14/2009 09:40:23	Apr/14/2009 09:43:23	3m	309.8 KiB	47.0 KiB

Active Sessions

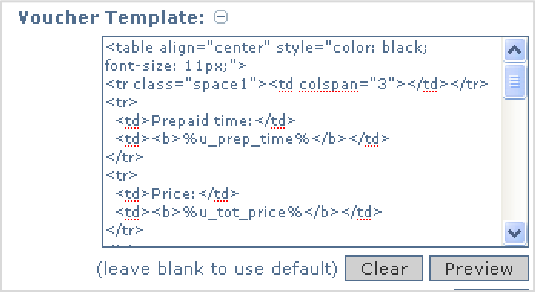
The active sessions/users page will show you the users that are currently logged in. When they logging out, the Radius system should receive accounting information updates, showing how much time they used, as well as data transfers information



□	▽ Username △	▽ Prepaid △	▽ Uptime △	Time left	▽ Price (USD) △	▽ Download △	▽ Upload △
□	16645	3w:3d	1w:5d:9h:45m:25s	10h:29m:8s	70.94	10.0 GiB	374.7 MiB

Vouchers

The User Manager system also allows you to create vouchers. These would be some form of card, or paper that you can sell in a retail business to customers. These cards will contain username/passwords that have a specific amount of session time. You could give out free 1-hour vouchers; every username/password is different so you would not have to worry about other users freeloading on your network. But you could also sell 1-week vouchers as well.



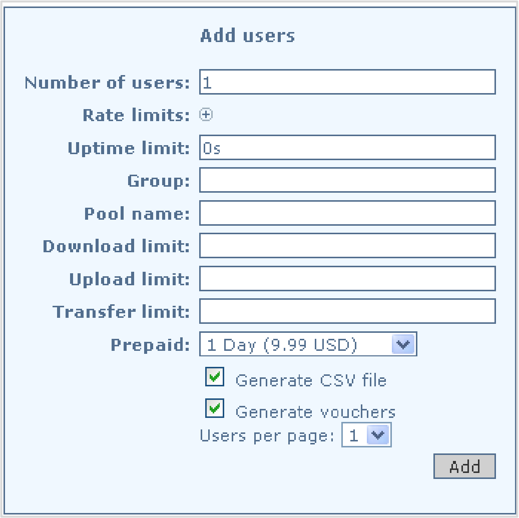
Voucher Template: ⊖

```
<table align="center" style="color: black;
font-size: 11px;">
<tr class="space1"><td colspan="3"></td></tr>
<tr>
  <td>Prepaid time:</td>
  <td><b>%u_prep_time%</b></td>
</tr>
<tr>
  <td>Price:</td>
  <td><b>%u_tot_price%</b></td>
</tr>
```

(leave blank to use default) Clear Preview

Before generating these, you should take a look at your subscriber information. At the very bottom you have Voucher Template. This template allows you to setup how your vouchers will look when you print them up.

Once you are happy with the way your voucher will look; now you can go ahead and generate them. To do this, on the main status page of the User Manager admin interface, you use the Add Users section on the right.



Add users

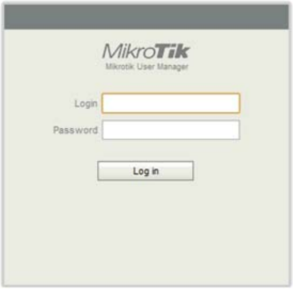
Number of users: 1
Rate limits: ⊕
Uptime limit: 0s
Group:
Pool name:
Download limit:
Upload limit:
Transfer limit:
Prepaid: 1 Day (9.99 USD)
☑ Generate CSV file
☑ Generate vouchers
Users per page: 1

As you can see, you can specify rate limits, the number of vouchers you wish to generate, as well as limits, and how much prepaid time users have. You can generate both a CSV file that you can merge with your own template, or you can actually generate vouchers per your subscriber template. Once created these users are in the database, and you can print these out and give these username and passwords out as you wish.

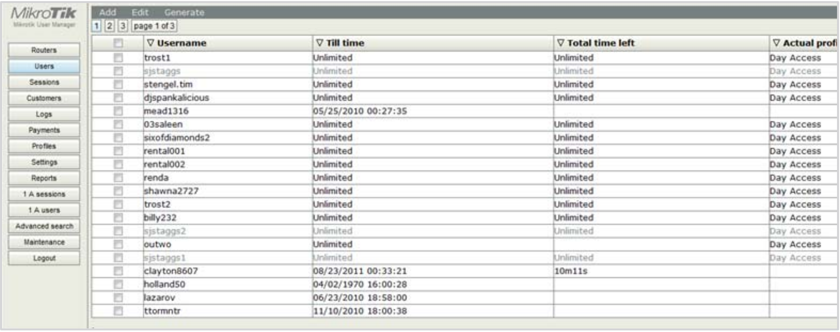
V5 of User Manager

In this section we will describe and show screenshots of the newer version of User Manager. At the time of this writing we are using RouterOS v5.6. Some of the major changes are simply layout.

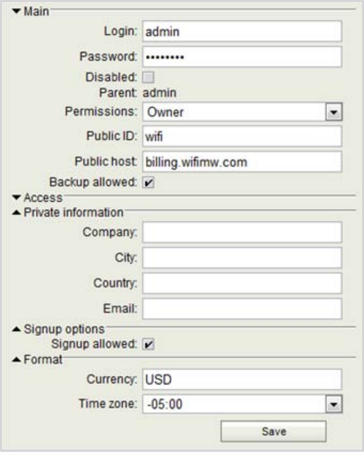
Layout



The User Manager now has a new look in the management and login side of things, as well as additional options.



□	▽ Username	▽ Till time	▽ Total time left	▽ Actual profi
□	trost1	Unlimited	Unlimited	Day Access
□	sjstaggs	Unlimited	Unlimited	Day Access
□	stengel.tim	Unlimited	Unlimited	Day Access
□	djspankalicious	Unlimited	Unlimited	Day Access
□	mead1316	05/25/2010 00:27:35		
□	03saleen	Unlimited	Unlimited	Day Access
□	sixofdiamonds2	Unlimited	Unlimited	Day Access
□	rental001	Unlimited	Unlimited	Day Access
□	rental002	Unlimited	Unlimited	Day Access
□	renda	Unlimited	Unlimited	Day Access
□	shawna2727	Unlimited	Unlimited	Day Access
□	trost2	Unlimited	Unlimited	Day Access
□	billy232	Unlimited	Unlimited	Day Access
□	sjstaggs2	Unlimited	Unlimited	Day Access
□	outwo	Unlimited		Day Access
□	sjstaggs1	Unlimited	Unlimited	Day Access
□	clayton8607	08/23/2011 00:33:21	10m11s	
□	holland50	04/02/1970 16:00:28		
□	lazarov	06/23/2010 18:58:00		
□	ttormntr	11/10/2010 18:00:38		



▾ Main

Login: admin

Password: ••••••••

Disabled: □

Parent: admin

Permissions: Owner

Public ID: wifi

Public host: billing.wifimw.com

Backup allowed: ☑

▾ Access

▴ Private information

Company:

City:

Country:

Email:

▴ Signup options

Signup allowed: ☑

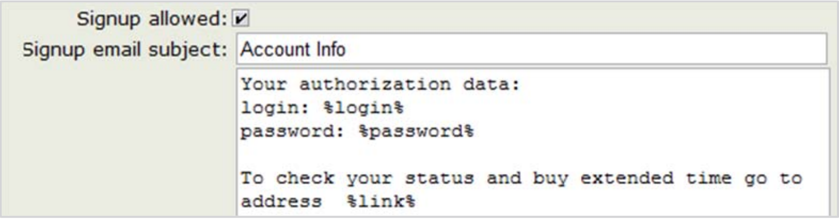
▴ Format

Currency: USD

Time zone: -05:00

The customers, users, routers sections are basically the same, we still have the public ID and host as well, and the ability to allow for signups. We also have the format information, such as our time zone and currency that we need.

Once you have selected that the signup is allowed under your customer, you will also need to enable signup under the settings section. Here you will be able to create the e-mail that will be sent out with the account information to the end user.



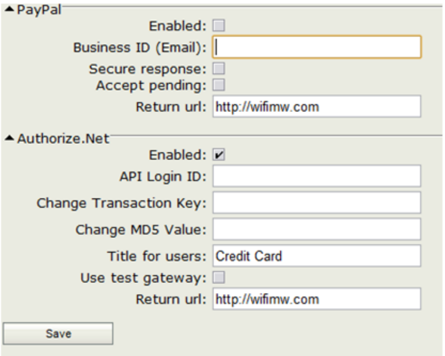
Signup allowed: ☑

Signup email subject: Account Info

```
Your authorization data:
login: %login%
password: %password%

To check your status and buy extended time go to
address  %link%
```

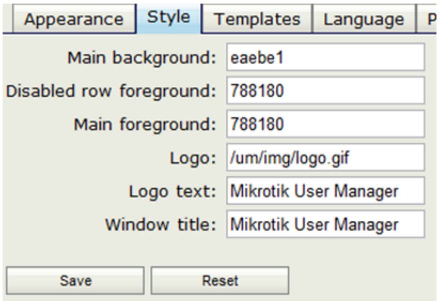
Payments Page

The signup system is a bit different, as the configuration for Authorize.net and PayPal have been moved to the settings page. On the settings page we have a tab called payment gateways and under that tab we have all of the necessary information for receiving payments. All of this information is the same as in v3 of User Manager.


▲ PayPal
Enabled: □
Business ID (Email):
Secure response: □
Accept pending: □
Return url: http://wifimw.com
▲ Authorize.Net
Enabled: ☑
API Login ID:
Change Transaction Key:
Change MD5 Value:
Title for users: Credit Card
Use test gateway: □
Return url: http://wifimw.com

Styles and Customization

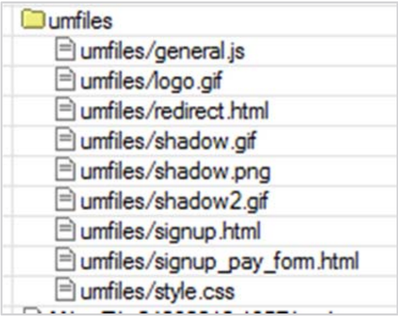
In v5 we also have the ability to change colors using HEX codes, as well as the ability to upload a new image into our RouterOS files menu, and select it via the Logo field. We can also change the logo text and window title here. This also will replace the MikroTik logo in the customer management window as well.



Main background: eaebe1
Disabled row foreground: 788180
Main foreground: 788180
Logo: /um/img/logo.gif
Logo text: Mikrotik User Manager
Window title: Mikrotik User Manager

User Manager HTML Customization

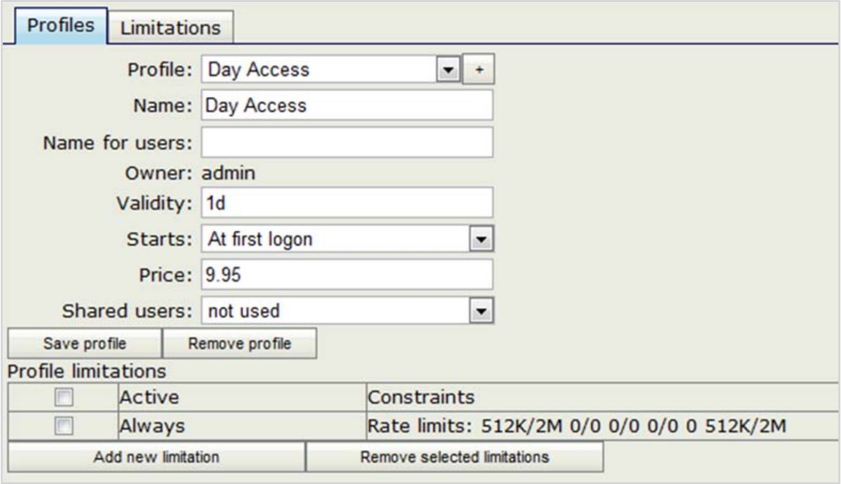
In v5 we also have full control over the HTML files that are associated with the User Manager system. This is one of the major reasons to start using the v5 system. How to customize these is outside the scope of this book.


umfiles
umfiles/general.js
umfiles/logo.gif
umfiles/redirect.html
umfiles/shadow.gif
umfiles/shadow.png
umfiles/shadow2.gif
umfiles/signup.html
umfiles/signup_pay_form.html
umfiles/style.css

The major file that you will edit is under "/umfiles/signup.html." This file is the main signup page that you can customize as you wish to. Please refer to the MikroTik website for all of the details, coding, as well as variables that you can use for the customization of the HTML on your signup page

Profiles

In the v3 system, we did not have profiles; this is a new feature in v5. Profiles replace the credit system in v3. The profile system is used just like the credit system, it is setup to specify how much each profile costs, how long it can be used for, but now we also have options to specify limitations.



Profile: Day Access
Name: Day Access
Name for users:
Owner: admin
Validity: 1d
Starts: At first logon
Price: 9.95
Shared users: not used

Profile limitations

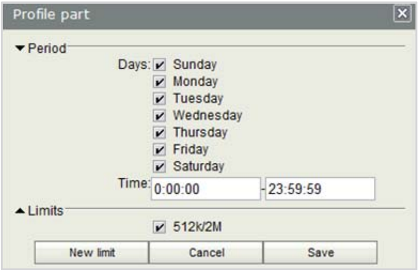
□	Active	Constraints
□	Always	Rate limits: 512K/2M 0/0 0/0 0/0 0 512K/2M

We do have a few new options as well, and these are extremely important as they fix issues in v3. In v3, when the client was given a credit for 10 hours, the client could login and logout as much as they want, over months of time if they wished, as long as their actual used time was under 10 hours.

In v5, we have a Validity field; this field just like the uptime field in v3, specifies how long the user account is valid. Now though, we have a Starts field. This field has two options; at first logon and now. The now option says that their validity time starts when the profile is applied to the user account. If they create an account, and pay for service that has a validity of 1 day, then upon their payment being processed and then the profile being applied to their account, they will have 24 hours to use the service. That includes anytime that they are not on-line. They simply have 24 hours of access starting from the point they signed up. The at first logon option, simply starts the validity timer when they first login and are authorized though the User Manger system.

Limitations

Limits are applied to limitations, and limitations are applied to profiles. The reason is that you could have different limitations based on time. The basic limitation section defines the times the limits apply to. By clicking the New Limit button, you can create new limits that can be applied to this limitation.

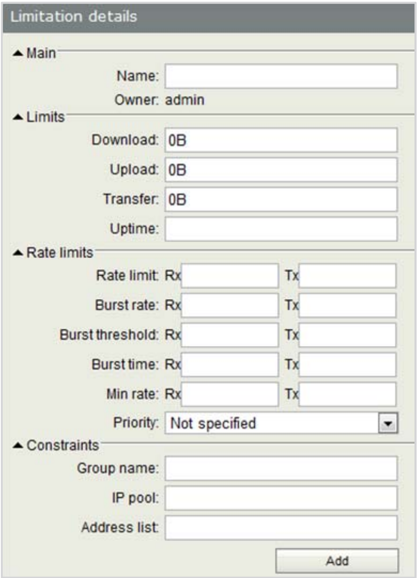


Days: ☑ Sunday
☑ Monday
☑ Tuesday
☑ Wednesday
☑ Thursday
☑ Friday
☑ Saturday

Time: 0:00:00 - 23:59:59

☑ 512k/2M

In the New Limit box, you can create a name for the limit and specify how you wish to limit the profile. You can choose to limit it by transfer, uptime, downloads, or uploads, and/or by a rate limits that may or may not include bursting. You also may assign Constraints to the profile as well. These could be from what IP Pool the authenticated user will pull their address from, or what address list to add the IP address they have too.

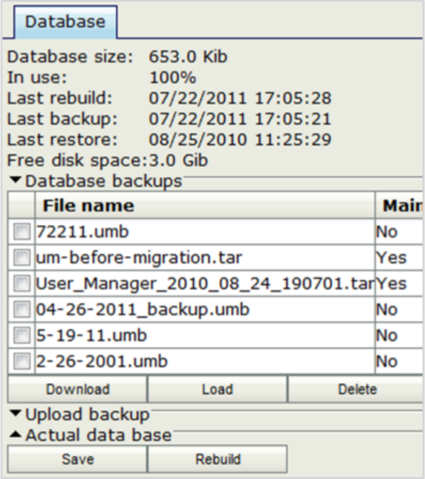


Name:
Owner: admin
Download: 0B
Upload: 0B
Transfer: 0B
Uptime:
Rate limit: Rx Tx
Burst rate: Rx Tx
Burst threshold: Rx Tx
Burst time: Rx Tx
Min rate: Rx Tx
Priority: Not specified
Group name:
IP pool:
Address list:

In most cases, I use the validity field in the profile to limit the customers available time on-line. I typically sell 1 day or weekly packages and do not want the customer to be able to access it outside of their validity; therefor I do not use the uptime anymore. The transfer limits are nice to cap the downloaders, but make sure you disclaim this in your sign-up process. The rate limits are the same exact system as a simple queue, the User Manger system sends this information via Radius attributes, and RouterOS builds a simple queue for the user with this information.

Database Maintenance

In v5, we also have a web based database maintenance section. It's simply called Maintenance. In side this system, we have the ability to see how large our User Manager database is, the ability to save the actual database, and download it for safe keeping as well as the ability to upload saved databases, and do basic file maintenance on the existing backup files. You also can run a rebuild on the database.



Database size: 653.0 Kib
In use: 100%
Last rebuild: 07/22/2011 17:05:28
Last backup: 07/22/2011 17:05:21
Last restore: 08/25/2010 11:25:29
Free disk space:3.0 Gib

	File name	Mair
□	72211.umb	No
□	um-before-migration.tar	Yes
□	User_Manager_2010_08_24_190701.tar	Yes
□	04-26-2011_backup.umb	No
□	5-19-11.umb	No
□	2-26-2001.umb	No

V5 Signup

In v5 the signup page link is different, the new link is:

http://routerdnsname/user/signup/publicid

Command Line Interface

The command line interface is arranged just like the WinBox interface is organized prior to version 3.25. In v3.25 and higher, MikroTik changed the WinBox interface to accommodate small resolution laptops, and net books, but they did not change the command line interface.

If you have used DOS at all, then you should feel comfortable with the command line interface. The directory structure is just like the menu in WinBox, except only thing is you don't have to put in CD to change directories, and "?" always gives you options.

```
[admin@CORE] >
```

Upon logging into the command line, you will get the username@systemidentity of the RouterOS system you are using.

To change to a different sub menu, let's use IP → Addresses. To put an IP address on an interface, we will simply use the menu names.

```
[admin@CORE] >ip address
[admin@CORE] /ip address>
```

Note that the command line interface also changes to show what menu option you are in. I will now change to just the ip submenu

```
[admin@CORE] /ip address>..
[admin@CORE] /ip>
```

To change to the upper menu, I simply added the dot dot and hit enter. This will let you go up a menu item. Let's change to see the wireless registrations.

```
[admin@CORE] /ip>/interface wireless registration-table
[admin@CORE] /interface wireless registration-table>
```

Note here that I used a forward slash in front to change to another menu that is not underneath the IP ADDRESS menu that I was in before. I could also have used a forward slash by itself, hit enter, and then typed the rest of the menu out. Typing the long line of menu items can be time consuming though, so let's change to another menu, our IP → Firewall → Address-List Menu.

```
[admin@CORE] /interface wireless registration-table>/ip fir add
[admin@CORE] /ip Firewall address-list>
```

Here, I used the forward-slash to start out with, but note that some of the menu items are not completely typed out. If you type the first few letters of the menu item and there is no other menu item that would match the first few letters, that is all you need. You can also check your work by hitting the TAB button. For example if I typed in */ip fire add* and then hit the TAB key, it would auto fill with */ip fire address-list* for me. This will work on multiple levels, so on the Firewall menu item, I could have hit TAB then typed ADD and then hit TAB again.

Now let's look at some options inside a menu. So switch over to the IP → Firewall → NAT menu, and list all of the NAT rules.

```
[admin@CORE] /ip Firewall address-list>/ip fir nat
[admin@CORE] /ip Firewall nat>print
Flags: X - disabled, I - invalid, D - dynamic
 0 X ;;; place hotspot rules here
     chain=unused-hs-chain action=passthrough

 1     chain=dst-nat  action=dst-nat  to-addresses=172.25.0.5  protocol=tcp
dst-address=99.184.190.92
     dst-port=25,143,80,443,53

 2     chain=dst-nat  action=dst-nat  to-addresses=172.25.0.5  protocol=udp
dst-address=99.184.190.92 dst-port=53
```

First we changed to the proper menu, and then issued a print command. In many cases you can just type PR as well. This lists out any of the rules, if they are valid, dynamic etc, and lists what they do. Now we will change item two by specifying a different to-address.

```
[admin@CORE] /ip Firewall nat>set 2 to-addresses=172.25.0.99
```

I used the set command to set a parameter in that specific rule number. If we wished to create a rule, we would use the add command, and to remove, we simply use the remove command. You can also move items from one spot to another by using the move command. To move item 2 to 1, you would type *move 2 1* and that's it.

99% of the commands in the command line interface are done this way. It is very simple to use. Remember that you can always use a question mark to find out what menu and options you have in any given location in the command line interface.

Command Line Hotkeys

CTRL-X = Safe Mode – Enter and Exit
CTRL-D = Exit Safe and discard all changes

CTRL-V = Hotlock Mode – Auto finishes commands

Quick Reference Guide

You want a super quick reference guide that explains how to do common features in RouterOS? This is it! Step by Step instructions on how to get common tasks done quickly!

NetInstall of RouterBOARD Products

- Download NetInstall Utility
- Download necessary NPK files, ensure compatibility with RouterBOARD CPU
- Set Network card on PC for static IP
- Ensure no Firewall, or network security software applications are running.
- Run NetInstall Utility
- Configure Net Booter with IP address inside subnet of your PCs static IP Network
- Connect NULL Modem cable to serial port on RouterBOARD
- Start Terminal Software – 115200 baud rate
- Power on RouterBOARD
- Press any key to enter RouterBOARD BIOS setup
- Select Boot Device
- Select Boot from Ethernet once, then NAND
- Exit BIOS setup
- Upon Reboot of RouterBOARD, RouterOS Software Remote Installation will be loaded
- In NetInstall, select your RouterBOARD MAC, typically called nstreme device
- In NetInstall browse to the package you wish to install
- Select other options, such as keeping old configuration, default baud speed as well as default script if necessary
- Press Install
- RouterBOARD will install, will prompt to press any key to reboot after installation
- RouterBOARD will boot to NAND, generate SSH keys, start services and show login prompt!

NetInstall your Flash / DOM / Hard Disk

- Download NetInstall Utility
- Download necessary NPK files, ensure compatibility with RouterBOARD CPU
- Run NetInstall Utility
- In NetInstall, select your Drive letter – Careful not to select a drive with data on it!
- In NetInstall browse to the package you wish to install
- Select other options, default baud speed as well as default script if necessary – You cannot keep old configurations
- Press Install
- Device will show Installation is Complete
- Insert storage device into your new RouterOS system and power on
- Upon startup, the RouterOS system will finish the installation
- The RouterOS system will reboot following the installation, generate the SSH keys, start services and show a login prompt.

Creating a Active/Backup Bridged Auto-Fail Link

- Physical Links
- Each side will need to have a RouterBOARD, and individual ports for each link, plus an extra Ethernet for your client data.
- Link one, the one we wish to prefer, will be plugged into Ethernet 1 on both RouterOS units on each end
- Link two, will be plugged into Ethernet 2 on both ends.
- The cable going to the rest of the network will be on Ethernet 3 on each RouterOS.
- Setup Bridges on both ends, with STP or RSTP.
- Setup Ethernet 1, 2 and 3 as bridge ports
- Increase the Priority for Ethernet 2, on both sides to 90+, or another higher number than the default that is created on the other ports.
- Once this is setup, your Ethernet 1 should be your designated port, and Ethernet 2 will be the backup port.

Setup Transparent Web Proxy System

- Setup Proxy Settings including if you wish to store on disk and the proxy port.
- Secure your Proxy system using the access lists.
- Tell your Proxy system to cache port 80 data.
- Create DST-NAT rule to redirect outbound TCP/80 Connections to the Proxy port.

Redirect Non-Paying Customer

- This requires a external web server
    - Must answer to the IP address not host header information
    - Configure the 404 error message as well to be your customer message. Ensure you have your contact information and hours.
- Address-lists
    - Create Address-List called Overdue_Customer
- Firewall NAT rules
    - Create DST-NAT rule that matches the Source Address-list of Overdue_Customer ,then redirect TCP DST Port 80, to the IP of the web server, port 80
- Filter Rules
    - Create forward chain rule to jump to Overdue chain for customers on SRC address-list of Overdue_Customer.
        - Overdue Chain
        - Allow TCP and UDP Port 53
            - Redirects only work in IPs, must have DNS resolution
        - Allow Port 80 to your web sever
        - Allow Port 80 to any other sites you wish them to have access too. Maybe authorize.net payment site or PayPal.
        - Deny all rule

Per Connection Load Balancing

- Assume we have three Internet circuits that we wish to balance across.
- Assume they are even bandwidth each
- Add IP addresses from each connection to proper interface
    - Add each of your Internet connection IPs to each of your interfaces, if they SHARE the same subnet (regardless if they are NATTING or not) you can simply set them up on one interface, but make the gateways on your modems .2, .3. and .4, while your router is .1
- Create Connection Marks
    - Add pre-routing Mangle rules to mark connection using the PCC options of source address and port. The first will be 3/0 the second rule will be 3/1 and last will be 3/2.
    - There will be three rules
    - Have each rule have a src-address of your private IP network.
- Create Routing Marks
    - Create three routing marks, one based on each of your connection marks created previously
- Create Routing Rules
    - Create three rules, using the routing marks; each mark performs an action of lookup on three different tables. In our example we will name them C1,C2 and C3
- Create Routing tables
    - On table C1, add the default route of your first connection
    - On table C2 add the default route of your second connection
    - ON table C3 add the default route of your third connection
- Create NAT Rules
    - Create a NAT rule out either the single interface or multiple interfaces required to get to the Internet from your private LAN

Create a Private VPN

- Assume 192.168.0.x/24 is the private LAN
- Add IP Pool for PPTP users
    - 192.168.200.2-192.168.200.200 should give them 198 users.
- Configure PPTP server
    - Enable server under server options, set the default profile to default-encrypted
    - Modify the default-encrypted profile to include the local address of 192.168.200.1 and the remote address of the PPTP Pool
    - Modify the DNS servers for the private DNS servers inside the network.
    - Can also modify anything else necessary.
    - Add PPTP Secrets using default-encryption profile
- Add NAT rules
    - Add a source-address src-nat rule to masquerade out your internet connection for the new 192.168.200.0/24 subnet.
    - This will allow Internet access while connected to the VPN.
- Add DNS (optional)
    - If you have a public IP address, create a DNS name for this, have your users point their VPN to vpn.businessname.com, so that if there is an IP change later, you just have to change DNS.

Appendix

Features Only Available via Command Line Interface

- Export Command – Used to create text configuration export. Running this command an under a sub-section of RouterOS will process only that section and sections under that section. If you run the command under /ppp, it will export all commands under the /ppp context, including /ppp secrets, /ppp profiles, and so on. Issuing this command right in the root of the command line interface will result in a full configuration export into a text readable format.
- Import Command – Used to process .rsc or other script files without pasting them into RouterOS. This allows you to process a script file after uploading it to the File List.
- /Tool Fetch Command – Used to fetch files from HTTP and FTP Websites
- /Tool E-Mail Send – Used to send E-mails via the e-mail system. The e-mail server settings can be specified in the command line, but you can also specify them in WinBox under Tools → E-Mail
- /system note Command – Used to tag a note on the command line Interface. Upon entering the command line interface, the text placed in the Router with the note command will be displayed.
- /interface wireless set *item* disable-running-check=yes/no – By disabling the running check OSPF never sees an interface state change. The only way it knows if the link is down is the by the dead router detection. Sometimes this is upwards of 60-90 seconds. If your running check is set to no, then as soon as an interface drops, i.e. wireless connection drops for a moment, OSPF will issue a state change. Useful if you have a connection that likes to drop for a moment.
- /ip Firewall calea – Interceptor portion of the CALEA Package.
- /tool calea – Server and capture portion of CALEA Package

Index

6to4, 114
802.1QinQ, 184
802.3ad, 185
About the Author, 12
Accept, 145
Access List, 55
Access Lists, 209
Access Point Time, 201
Accessing RouterOS, 42
Accounting, 334
Active Users, 68
Active-backup, 186
Add ARP for Leases, 91
Address Lists, 134
Address Resolution Protocol, 107
advertise, 111
Agents, 387
aggregated frames and sending using block acknowledgement, 226
Air/Data Rates, 200
AMSDU, 226
Antenna coax, 222
AP-Bridge, 202
API, 70
Area Prefixes, 213
ARP, 107
ARP List, 109
at first logon, 427
Auto Upgrades, 356
Backup / Restore, 73
Bad CCQ, 237
balance-alb, 186
balance-tlb, 186
balance-xor, 186
Bands, 201
Bandwidth Test Client, 358
Bandwidth Test Server, 357
Basic / Advanced, 196
Basic Firewall, 53
Basic RouterOS Setup, 80
BGP, 368, 372
BGP Aggregates, 370
BGP based VPLS, 331
BGP Networks, 370
Bindings, 299
Bonding, 185
boot from Ethernet once, 34
boot from its NAND, 34
boot from the NAND, 34
bootp, 35
Bridge, 176, 202
Bridge Loopbacks, 180
Bridge Ports, 177
Bridged Access Point Configuration, 101
Bridged Client, 102
Bridging PPTP, 255
Brute Force Attacks, 151
Bursting, 285
CALEA, 346
CALEA Interceptor, 346
CALEA Server, 348
Carrier Sense Multiple Access, 231
CD Installation, 38
Cellular USB Card, 251
Chains, 131, 156, 161
Change MSS, 159
Change TOS Bit, 159
Changing Path Costs, 367
Checking Gateways, 117
Choosing a Tunnel Type, 269
CIDR, 80
Classless Inter-Domain Routing, 80
Clear DF, 160

Clock, 338
Command Line Interface, 430, 440
Common Wireless Configurations, 101
Communication Assistance for Law Enforcement, 346
Configuration of basic Masquerading, 96
Configuration of LDP, 320
Configuration of NV2, 233
Configuration Reset, 354
Configuring IP Addresses, 80
Connection Bytes, 137
Connection Limiting, 141
Connection Limiting Chart, 141
Connection Lists, 212
Connection Marks, 158
Connection States, 125
Controlling P2P, 293
CPE – Client Premise Equipment Configuration, 101
Credits, 16
CSMA, 231
Database Maintenance, 429
Default Options, 195
Default Routes, 82
Default TX Rates, 196
Default User and Password, 43
Default/Summary/Full Routing Table, 372
Deploying IPv6 within your ISP, 114
Device Appearance, 394
Device Options, 392
DHCP Relaying, 94
DHCP Server, 56
DHCP Server Wizard, 87
DHCP-Client, 85
DHCP-Server, 86
DNS Caching, 82
Dotted-decimal Notation, 80
Double Queuing, 289
Drop, 146
DST Limit, 143
Dual Stack, 111
Dude, 378
Dude - Admins, 388
Dude - Charts, 389
Dude - Discovery Services, 388
Dude - Files, 394
Dude - Links, 395
Dude - Logs, 396
Dude - Map Settings, 399
Dude - Menus and Options, 383
Dude - Reset Configuration, 382
Dude - Server Configuration, 384
Dude - Spectrum Scan, 409
Dude - Upgrades, 401
Dude - Working with Devices, 401
Dude Agents, 380
Dude Devices, 390
Dude Layout, 381
Dynamic Routing, 364
E1, 24
ECMP, 118
E-Mail System, 359
Encryption Performance, 268
Ensuring Bandwidth Allocations, 287
EoIP, 241
EoIP Tunnel, 242
Ethernet, 170
Ethernet Speed and Negotiation, 172
Extended Frequency, 27
Failover with PPPoE Servers, 259
Fetch, 359
Fiber Optic Interfaces, 24
FIFO Queues, 276
Files, 71
Firewall Actions, 145
Firewall Efficiencies, 126
Firewalling, 124
Firewalling Examples, 153

Flashfig, 78
Flashfig on RouterBOARDs, 78
Forward, 157
Forward Chain, 131
Free Hotspots, 297
Frequency Usage, 198
FTP, 70
Global Sync, 277
Graphing, 360
Hard Disk Installation, 40
Hidden Node Issue, 229
Hiding the SSID, 195
Hierarchical Token Bucket, 272
Home Router Walkthrough, 99
Hotspot Cookies, 303
Hotspot Definitions, 298
Hotspot Interface, 299
Hotspot Login Methods, 303
Hotspots, 296
Hotspots with Radius, 306
How to Match Data, 136
HTB, 272
HTB and Rate Limiting, 273
HTB Packet Flow, 272
HTB Queue Tree Structure, 273
HWMP, 188
HWMP+., 188
Identifying Queue Data, 271
If Installed vs. Always, 364
IKE, 266
Importing Scripts, 74
Improving 802.11n, 226
Inbound NAT, 163
Inbound NAT with DHCP Public IP Address, 167
Ingress Priority, 142
Input, 157
Input Chain, 131
Instances, 369
Intercepting Data, 347
Interfaces, 170
Internal Hotspot User Management, 306
Introduction, 10
IP Addresses, 80
IP Bindings, 308
IP Pools, 95
IP Scan, 363
IP Subnetting, 81
IPIP, 243
IPsec, 265
IPSec Encryption/Decryption, 129
IPsec Peers, 267
IPsec Proposals, 268
IPv6, 110
IPv6 Addresses, 111
IPv6 BGP, 112
IPv6 Dynamic Protocols, 112
IPv6 Firewall, 112
IPv6 OSPF v3, 113
IPv6 Package, 110
IPv6 to 4 Tunnels, 114
Jumping to Chains, 132
L2TP, 252
Large Transfer Queues, 289
Layer 7 Filters, 139
Licensing, 26
Limit, 143
Limitations, 428
Limiting Total Throughput, 284
Link Technologies, Inc, 13
Links, 402
Local Users, 66
Logging, 76, 146, 353
Logging Actions, 353
Logging Rules, 77
Logging to Disk/File, 77
Loopback SRC-Address, 182
Lose your RouterOS License, 28
Low Signal, 236
M3P, 335
MAC Authentication, 206
Mangle, 156

Mangle Packet Flow, 129
Mangle to Route Traffic, 123
Masquerading, 96, 162
MCS's, 228
MESH, 188
MetaRouters, 349
methods of accessing a RouterOS System?, 43
Migration to NV2, 235
MikroTik Packet Packing Protocol, 335
MIMO, 228
MPLS, 316
MPLS Configuration, 319
MPLS Traffic Engineering, 327
MPLS/VPLS MTU, 325
Multi-Link, 261
Multiple Data Packages and PCQ, 292
Multiple PCQ Rates, 290
Multiple Radius Servers, 313
Multiple SMTP Outbound Limits, 154
Multiple-Subnets Behind a Hotspot, 310
Multiprotocol Label Switching, 316
NAT, 96
Neighborhood Viewer, 43
Neighbors, 352
Net booting, 33
NetFlow, 339
NetInstall, 40
Network Address Translation, 161
Network Maps, 397
Non-Paying Customer, 436
Non-RouterOS MetaRouters, 350
Notifications, 403
Nstreme, 216
Nstreme Dual, 217
Nth, 144
NTP, 336
NTP Client, 336
NTP server, 337
NV2, 231
NV2 Hardware, 232
NV2 QoS, 234
One-to-One NAT, 165
OpenVPN, 262
OpenVPN Client, 263
OpenVPN Server, 262
OpenVPN Server Interface, 263
OpenVPN TAP, 264
OSPF, 366
OSPF Full-duplex, 367
OSPF Loopback, 180, 366
Other Chains, 131
Outages, 406
Outbound NAT, 164
Output, 157
Output Chain, 131
Packet Flow, 128
Packet Flow with MESH, 130
Packet Marks, 157
Packet Sniffer, 361
Paid Hotspots, 297
Parents, 286
PCQ Bursting, 282
PCQ Queues, 278
PCQ RAM Requirements, 279
Peer to Peer Filtering, 138
Peers, 369
Per Connection Load Balancing, 437
POD Attacks, 152
Policy Based Routing, 120
Pools, 335
Port Scan Detection, 142
Postrouting, 156
PPP, 244
PPP Active Connections, 247
PPP Client, 250
PPP Profiles, 246
PPP Secrets, 244

PPP Server, 247
PPPoE, 258
PPPoE and Masquerading, 163
PPPoE Client, 55, 260
PPPoE Scan, 361
PPPoE Server, Dynamic Routing, 259
PPTP, 252
PPTP Client, 254
PPTP Server Interfaces, 253
Pre-Built Firewall Scripts, 153
Prerouting, 156
Private VPN, 438
Probes, 406
Profiler, 362
Profiles, 427
Propagate TTL, 322
Protecting Networks, 149
Protecting Your Router, 148
Proxy ARP, 108
Pseudobridge Mode, 103
Queue Trees, 283
Queue Types, 275
Quick Reference Guide, 434
Radius Client, 312
Radius RouterOS Users, 69
Random, 143
Rate Flapping, 224
rate selection, 228
Real World Usage of DHCP Relay, 94
rebuild, 429
RED Queues, 277
Redirect, 168, 436
Reference Version of User Manager, 413
Registration Table, 52, 211
Reject, 147
Reply-Only, 108
Returning from Chains, 133
RIP, 364
RIPng, 113
Rouge DHCP-Servers, 92
Routed / NAT CPE, 103
RouterBoard Devices, 18
RouterOS Extras, 334
RouterOS Hardware, 18
RouterOS Installation, 30
RouterOS Reference Version, 15
RouterOS Services, 69
RouterOS Tools, 352
RouterOS User Groups, 67
Routing, 52
Routing Filters, 371
Routing Marks, 158
Routing Policies, 120
RSTP, 177
RSVP TE Auto-Bandwidth, 329
RSVP TE Tunnel, 328
Rules, 125
Running Dude Server, 382
Safe Mode, 62
Scan List, 196
Scanning, 197
Scheduler, 355
Scripting, 355
second edition of Learn RouterOS., 16
Secure Shell Access, 45
Securing NV2, 233
Securing Services, 70
Security Profiles, 205
Selective Port Forwarding, 166
Server Profiles, 302
Set Priority, 160
Setting WMM Priority’s, 204
Setup of a Hotspot, 300
Setup Transparent Web Proxy System, 436
SFQ Queues, 277
Signal Issues, 223
Simple Queues, 54, 284
Sniffing, 198
SNMP, 407

Snooping, 199
SNTP, 337
Socks, 336
Solar Power, 22
SPAM Prevention, 150
Special Thanks, 16
Spectrum Analyzer, 199
Splash Page, 298
SSH, 45, 70
SSH Brute Force Attack, 155
SSH Keys, 68
SSTP, 256
SSTP Client, 257
SSTP Server, 256
Stateless Autoconfiguration, 111
Static DNS Entries, 84
Static Routing, 116
Static WDS Bridges, 220
Station, 203
Store, 345
STP, 176
Strip IPv4 Options, 160
Styles and Customization, 426
Submaps, 402
Supported x86 Hardware, 25
Switch Controls, 171
Switches and MESH, 190
Syslog, 387
System Configuration Reset, 354
System Identity, 338
System Options, 53
T1, 24
Tarpit, 147
TDMA, 232
Telnet, 45
Text Backup Files, 73
TFTP Server, 338
The Dude, 378
The PowerRouter 732, 23
Time, 145
Time Division Multiple Access, 232
Tools, 408
Traffic Control, 270
Traffic Engineering, 329
Traffic Identification, 124
Traffic-Flow, 339
Transferring Files within Dude, 394
Transparent Web Caching, 342
Transparent Web Proxy, 436
Troubleshooting NTP, 337
Troubleshooting Radius, 314
Troubleshooting Wireless Links, 236
Tunnel Path Hops, 328
Tunnels, 240
Upgrades, 57
UPnP, 344
User Management, 66
User Manager, 412
User Manager - Concepts and Definitions, 415
User Manager - Layout, 424
User Manager database, 429
User Manager HTML Customization, 426
User Sign-In Page, 421
User Sign-Ups, 421
Using Distances, 118
Using IP Firewall, 178
Using Marks, 157
Using Multiple Rules to do what YOU want!, 153
Using NetInstall, 31
Using Nstreme, 225
Using PCQ, 280
Using Trail Users, 304
Using WinBox, 59
V5 of User Manager, 424
V5 Signup, 429
Validity, 427
VAPs and VLANs, 214
Virtual Access Points, 214
Virtual Ethernet, 174

Virtual Ethernet Interfaces, 174
Virtual LAN, 183
VLAN Configuration, 184
VLANs, 183
Vouchers, 422
VPLS, 323
VRRP, 191
Walled Garden, 308
Walled-Garden, 299
Wandering/Fluctuating Signal, 236
Watchdog, 357
WDS, 219
WDS Bridged Access Points, 220
WDS Bridged Wireless Link, 219
WDS-Slave, 202
Web Proxy, 340
Web Proxy Access List, 341
WebBox, 48
WebFig, 58
What is RouterOS, 14
Why Routing, 106
WIC – Wireless Interface Cards, 194
Wi-Fi Multimedia, 203
WILL LOSE YOUR LICENSE., 40
WinBox Interface Options, 64
WinBox Menus, 63
WinBox Service, 71
Windows PTPP VPN Users, 253
Wired Equivalent Privacy, 207
Wireless Distribution System, 219
Wireless Interfaces, 51
Wireless Link Optimization, 222
Wireless Multimedia Extensions, 203
Wireless Operational Modes, 202
Wireless Tools, 197
WMM, 142, 160, 203
WPA, 208
WPA2, 208
WWW, 71
X86 Based RouterOS Systems, 23
x86 Hardware, 25